

# Accounting department server



## Contents

Problem statement.....	3
Preliminary observation.....	5
Hardware.....	5
System installation and disk layout.....	7
BIOS configuration.....	9
Basic server configuration.....	9
Protecting GRUB2.....	10
Setting the system domain name.....	11
Disabling IPv6.....	11
Disabling Magic SysRq.....	12
GRE tunnel support.....	12
Creation of the users.....	12
Configuring access control lists.....	14
Disabling unnecessary services.....	15
Configuring the list of repositories.....	16
Installing auxiliary software.....	17
Configuring the UDEV subsystem.....	18
Configuration.....	18
Fail protection (auto-resume).....	21
Verification of performance and stability.....	21
Configuring network.....	22
Network interfaces configuration.....	23
Routing.....	26
Network bandwidth management.....	28
Checking the backup channels operability.....	29
DNS/DHCP service.....	30
Configuration.....	30
Fail protection (auto-resume).....	35
Verification of performance and stability.....	36
Network time service (ntp).....	40
Configuration.....	40
Fail protection (auto-resume).....	43
Verification of performance and stability.....	44

SSH Remote Management Service.....	44
Configuration.....	45
Fail protection (auto-resume).....	46
Verification of performance and stability.....	46
NUT configuration.....	48
Configuration.....	48
Fail protection (auto-resume).....	53
Verification of performance and stability.....	54
RAID1 configuration.....	59
Configuration.....	59
Fail protection (auto-resume).....	62
Verification of performance and stability.....	63
Samba configuration.....	64
Configuration.....	64
Fail protection (auto-resume).....	67
Verification of performance and stability.....	68
Setting up a backup system.....	71
About additional services.....	74
AppArmor/SELinux.....	74
Password protection (fail2ban).....	74
Log rotation (logrotate).....	75
Network time system (ntp).....	75
Mail service.....	75
Automatic shutdown of the system.....	77
Network filter (nftables) installation and configuration.....	77
Network filter installation.....	77
Network environment and potential threats.....	77
Network filter structure.....	79
Packet forwarding and tunnel support.....	81
Starting and checking the network filter.....	82
Completing the installation.....	97
Server usage and maintenance.....	97
Login procedure.....	97
When updating the system.....	97
Hardware status monitoring.....	98
Information about network adapters.....	98
Connecting clients.....	101
Disk array maintenance.....	103
Replacing an outdated local repository key.....	105

## Problem statement

It is required to create a school accounting server according to the following network scheme:

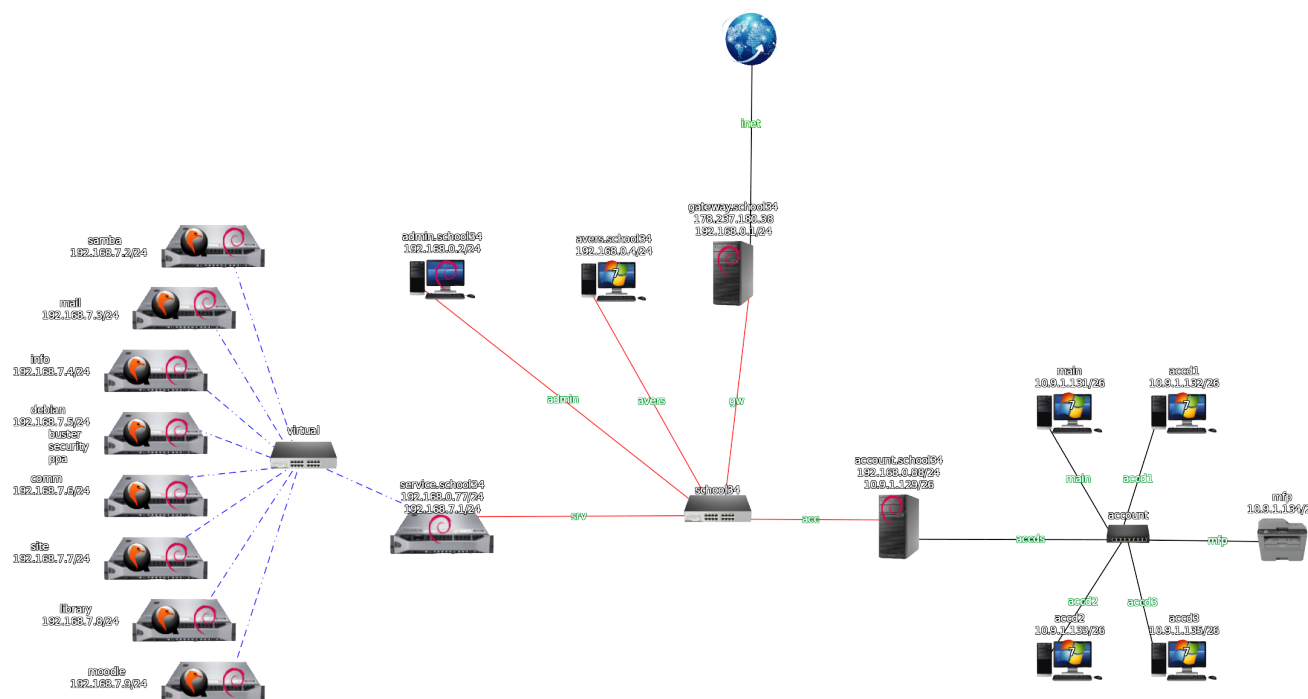


Figure 1: A fragment of the network diagram

This diagram shows the part of the school network visible to the accounting server (server.account.school34), indicating the domain names and IPv4 addresses of some nodes. Most of the network is hidden, because it should not be visible to either accounting workstations or its server. The figure also shows a virtualization server dedicated to the service.school34 domain. A number of virtual machines are deployed that provide various network services to the school's network clients on the basis of this server. The AIS server "Avers" and the computer of the system administrator should be noted in the school's network kernel.

The server should provide DHCP services only to the specified client devices on its internal interface, issuing clients with permanent IPv4 addresses with client identification via the link layer address (lladdr, MAC address). In this case, the addresses should be issued in the subnet 10.9.1.128/26, taking into account that the address 10.9.1.129 is occupied by the server itself.

The server should provide domain name service services to clients by forwarding requests about the service.school34 domain (192.168.7.0/24) to server.service.school34, about the school34 domain to gateway.school34, and all other requests to an external DNS server. At the same time, the external DNS server must be permanent with minimal filtering (YandexDNS), the use of a filtering DNS server (for example, SkyDNS) is not required.

The server should provide clients with the network time protocol service, using `gateway.school34` for its own synchronization, which is also a school-level time server with `ntp.school34` alias.

The server should be accessible for remote management only from the computer of the system administrator (admin.school34).

The server should provide the ability to quickly switch to a backup mobile Internet connection and back by accounting staff without the participation of the system administrator, while the set

of mobile devices should be strictly limited and known to the server in advance. With such a switch, routing and network filtering rules should also be automatically rebuilt.

For file exchange the server should provide a network directory via the smb protocol (Samba, CIFS). This directory should be available for reading and writing to all accounting workstations.

To organize a backup system, the server should have a software RAID1 array, hardware-provided with two hard drives of the same model. A file system should be formed on this array to support the backup directory.

The backup system should consist of two network directories and work as follows. The files that should be archived are placed in the network directory located on the system hard disk. At the end of the working day, these files are combined into an archive, which is transferred (transferred, not copied) to the RAID array. By the way the list of files placed in the archive is appended to the end of the file, which plays the role of a directory catalogue for backup storage. This file is also stored on a RAID array, has a simple format (html) and can be obtained by the client to search for the desired file and the archive containing it. The contents of the RAID array are also accessible via the smb protocol, but read-only. This scheme of work allows to protect data from malware that can get into accounting workstations, for example, from cryptographers. Of course, workstations managed by Windows OS are protected by antiviruses and firewalls, but an additional line of defense will not be superfluous, given that when a new cryptographer appears, there is a risk of its penetration into the protected network before his signature is entered into the antivirus databases. As for backup management, due to the specifics of accounting in a particular organization, it is performed manually, i. e. the accountants themselves decide what and when they need to save on a long-term basis.

The firewall in its work should be based on the default prohibiting policy. Practically any outgoing connections from accounting workstations to any external networks, connections established in response to them, as well as connections within the GRE tunnel necessary for software operation on one accounting workstation strictly defined in advance should be allowed. At the same time, the network MFP should not be able to send packets outside the accounting network. Also, the accounting network should not have a connection with the subnets of the organization that are not displayed in the diagram above, and the server should have a static set of ARP records for all client devices and available servers of the school.

As noted above, when switching to a backup Internet connection and back, changes in routing tables and filtering rules should be performed automatically and in a timely manner. If necessary, changes should also be made to the ARP tables.

Finally, the power supply system of the server and network equipment should be organized in a suitable way. The server should receive power from an uninterruptible power supply and have an information communication channel with it so that when the power is turned off in the organization's power grid, the UPS can promptly notify the server of a low battery level and initiate a correct shutdown of the server. At the same time, the accounting switch must also be connected to the same UPS. Thus, in conjunction with the UPS of workstations, the operability of almost the entire accounting subnet is ensured, with the exception of MFP. This allows to stop the network software operation normally.

Low-level server protection measures adequate to the task should also be taken. At the same time, file system encryption should be abandoned due to the physical location of the server within the premises of the accounting department and in order to avoid unjustified performance

degradation. It also does not provide disk quotas usage for the simple reason that network directories are placed on separate file systems and are managed on behalf of a single user.

## Preliminary observation

The following conventions are used in fragments of terminal sessions and when specifying commands to be executed: commands executed during system configuration are highlighted in bold, the system response or excerpts from configuration files and scripts do not have such highlighting. All commands are given with a full command prompt, which reflects the user on whose behalf the command is being executed and the current working directory. Both of these factors are significant in most of the listed commands and listings.

At last, the original document was written in Russian, all commands and scripts were also executed in system with Russian localization. So it is not surprising that some messages obtained from system are localized too. There is no any translations for such messages in this document because of two reasons. Firstly, such localized fragments are not essential and can't preclude understanding the sense of the commands. Secondly, localized fragments serve as remark about the document origination.

## Hardware

As a platform for the implementation of the accounting server, a stationary computer was selected from the existing machine park with some improvements. Further information about the hardware of this computer is given in an abbreviated form:

```
root@server:/# lshw
```

```
server
```

```
description: Desktop Computer
product: G41MT-S2P
vendor: Gigabyte Technology Co., Ltd.
width: 64 bits
capabilities: smbios-2.4 dmi-2.4 smp vsyscall32
configuration: boot=normal chassis=desktop uuid=00000000-0000-0000-0000-
1C6F65C2C56F
```

```
*-core
```

```
description: Motherboard
product: G41MT-S2P
vendor: Gigabyte Technology Co., Ltd.
```

```
...
```

```
*-cpu
```

```
description: CPU
product: Pentium(R) Dual-Core CPU E5500 @ 2.80GHz
vendor: Intel Corp.
physical id: 4
bus info: cpu@0
version: Pentium(R) Dual-Core CPU E5500 @
slot: Socket 775
size: 1286MHz
capacity: 4GHz
width: 64 bits
clock: 200MHz
capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic sep
```

```
mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 s
s ht tm pbe syscall nx x86-64 constant_tsc arch_perfmon pebs bts rep_good nopl cpuid
aperfperf pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr
pdcm xsave lahfm lm pti tpr_shadow vnmi flexpriority dtherm cpufreq
```

```

*-cache:0
  description: L1 cache
  physical id: a
  slot: Internal Cache
  size: 64KiB
  capacity: 64KiB
  capabilities: synchronous internal write-back
  configuration: level=1
*-cache:1
  description: L2 cache
  physical id: b
  slot: External Cache
  size: 2MiB
  capacity: 2MiB
  capabilities: synchronous internal write-back
  configuration: level=2
*-memory
  description: System Memory
  physical id: 18
  slot: System board or motherboard
  size: 2GiB
*-bank:0
  description: DIMM 400 MHz (2,5 ns)
  physical id: 0
  slot: A0
  size: 2GiB
  width: 196 bits
  clock: 400MHz (2.5ns)
*-bank:1
  description: DIMM [empty]
  physical id: 1
  slot: A1
*-bank:2
  description: DIMM [empty]
  physical id: 2
  slot: A2
*-bank:3
  description: DIMM [empty]
  physical id: 3
  slot: A3
*-pci
...
  *-network
    description: Ethernet interface
    product: RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller
    vendor: Realtek Semiconductor Co., Ltd.
    physical id: 0
    bus info: pci@0000:02:00.0
    logical name: enp2s0
    version: 06
    serial: 18:d6:c7:00:ea:6c
    size: 1Gbit/s
    capacity: 1Gbit/s
    width: 64 bits
    clock: 33MHz
    capabilities: pm msi pciexpress msix vpd bus_master cap_list ethernet
physical tp mii 10bt 10bt-fd 100bt 100bt-fd 1000bt 1000bt-fd autonegotiation
    configuration: autonegotiation=on broadcast=yes driver=r8169
duplex=full ip=192.168.0.88 latency=0 link=yes multicast=yes port=MII speed=1Gbit/s
    resources: irq:17 ioport:ce00(size=256) memory:fdeff000-fdefffff
memory:fddfc000-fddfffff
  *-network DISABLED

```

```

description: Ethernet interface
product: AR8151 v1.0 Gigabit Ethernet
vendor: Qualcomm Atheros
physical id: 0
bus info: pci@0000:03:00.0
logical name: enp3s0
version: c0
serial: 1c:6f:65:c2:c5:6f
capacity: 1Gbit/s
width: 64 bits
clock: 33MHz
capabilities: pm msi pciexpress vpd bus_master cap_list ethernet
physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
configuration: autonegotiation=on broadcast=yes driver=atl1c
driverversion=1.0.1.1-NAPI latency=0 link=no multicast=yes port=twisted pair
resources: irq:18 memory:fdcc0000-fdcffff ioport:df00(size=128)
...
*-ide
description: IDE interface
product: NM10/ICH7 Family SATA Controller [IDE mode]
vendor: Intel Corporation
physical id: 1f.2
bus info: pci@0000:00:1f.2
logical name: scsi0
version: 01
width: 32 bits
clock: 66MHz
capabilities: ide pm isa_compat_mode bus_master cap_list emulated
configuration: driver=ata_piix latency=0
resources: irq:19 ioport:1f0(size=8) ioport:3f6 ioport:170(size=8)
ioport:376 ioport:f800(size=16)
*-disk
description: ATA Disk
product: WDC WD3200AAKS-0
vendor: Western Digital
physical id: 0.0.0
bus info: scsi@0:0.0.0
logical name: /dev/sda
version: 3A01
serial: WD-WMAT10439126
size: 298GiB (320GB)
capabilities: partitioned partitioned:dos
configuration: ansiversion=5 logicalsectorsize=512 sectorsize=512
signature=6b449e09
...

```

As comments to the information provided, it should be noted the following:

The server has a central processor, the volume and characteristics of RAM, hard drives, network adapters and other systems that are quite sufficient to perform the tasks set.

This excerpt does not reflect the hard drives used to form a RAID array, they will be described separately in the corresponding section.

## System installation and disk layout

The system is installed without any graphical shell, but with standard system utilities and ssh service.

The system hard disk is divided into sections as follows (in the output below, lines that do not relate to the hard disk are omitted, and the commands themselves are executed on an already configured system with installed utilities):

```
root@server:/# /usr/sbin/fdisk -l /dev/sda
```

```
Disk /dev/sda: 298,1 GiB, 320071851520 bytes, 625140335 sectors
```

```
Disk model: WDC WD3200AAKS-0
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: dos
```

```
Disk identifier: 0x6b449e09
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	39063551	39061504	18,6G	83	Linux
/dev/sda2		39063552	46876671	7813120	3,7G	82	Linux swap / Solaris
/dev/sda3		46876672	85938175	39061504	18,6G	83	Linux
/dev/sda4		85938176	625139711	539201536	257,1G	83	Linux

```
root@server:/# /sbin/parted -l /dev/sda
```

```
Model: ATA WDC WD3200AAKS-0 (scsi)
```

```
Disk /dev/sda: 320GB
```

```
Sector size (logical/physical): 512B/512B
```

```
Partition Table: msdos
```

```
Disk Flags:
```

Number	Start	End	Size	Type	File system	Flags
1	1049kB	20,0GB	20,0GB	primary	ext4	boot
2	20,0GB	24,0GB	4000MB	primary	linux-swap(v1)	
3	24,0GB	44,0GB	20,0GB	primary	ext4	
4	44,0GB	320GB	276GB	primary	ext4	

```
root@server:/# mount -l
```

```
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro) [rootfs]
```

```
/dev/sda3 on /var type ext4 (rw,nosuid,nodev,relatime) [var]
```

```
/dev/sda4 on /srv/samba type ext4 (rw,nosuid,nodev,noexec,relatime) [samba]
```

```
root@server:/# df -h
```

Файловая система	Размер	Использовано	Дост	Использовано%	Смонтировано в
udev	965M	0	965M	0%	/dev
tmpfs	196M	5,5M	191M	3%	/run
/dev/sda1	19G	885M	17G	6%	/
tmpfs	980M	0	980M	0%	/dev/shm
tmpfs	5,0M	0	5,0M	0%	/run/lock
tmpfs	980M	0	980M	0%	/sys/fs/cgroup
/dev/sda3	19G	256M	18G	2%	/var
/dev/sda4	253G	61M	240G	1%	/srv/samba
tmpfs	196M	0	196M	0%	/run/user/1000

This disk layout scheme was chosen for the following reasons:

The DOS table is the simplest and sufficient for the server in question, especially taking into account the year of the hardware release. For the swap partition 4 GB are allocated, the rest of the disk space is divided into three partitions, so that the /var directory of changeable data and network directories are located in separate file systems. This is done as part of the basic measures to protect the server. Most of the disk is allocated specifically for network directories, while the minimum sufficient (with some margin) disk space is allocated for the rest of the system components.



Kernel attempts to get access to the floppy disk drive were detected when booting the system:

```
root@server:/# dmesg | grep fd0
[  24.880285] print_req_error: I/O error, dev fd0, sector 0
[  47.704340] print_req_error: I/O error, dev fd0, sector 0
[  74.660564] print_req_error: I/O error, dev fd0, sector 0
root@server:/# lsmod | grep floppy
floppy                86016  0
```

Since there is no such disk drive in the system, and the possibility of disabling the controller present on the motherboard is not provided in the BIOS of the system, the most correct solution is to disable the corresponding kernel module. To do this, it is enough to add this module to the "black" list and update the initramfs image, after which it would be necessary (including for verification) to reboot the system. At the same time, due to the scripts called when updating the boot image, the value of the PATH environment variable should be supplemented. All these actions are demonstrated in the following fragment of the terminal session:

```
root@server:/# echo "blacklist floppy" > /etc/modprobe.d/blacklist.conf
root@server:/# PATH=$PATH:/sbin
root@server:/# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-4.19.0-16-amd64
root@server:/# reboot
```

After the reboot, it is possible to ensure that there are no messages about problems, as well as that the kernel module is not loaded, using the following commands:

```
root@server:/# dmesg | grep fd0
root@server:/# lsmod | grep floppy
```

## BIOS configuration

The BIOS of the motherboard installed in the system allows to perform the following settings essential for the server operation:

- password protection of the access to the BIOS settings;
- turning on the PME Event Wake Up (turning on the system by internal timer at the beginning of the working day);
- AC Back Function Full-On (automatic switching on of the system after returning the lost power supply);
- disabling the USB Keyboard/Mouse/Storage Function, which does not interfere with the operation of the uninterruptible power supply.

## Basic server configuration

This section describes the measures applicable to many similar servers. In particular, the methods of minimizing the number of services, configuring the network subsystem and some security systems are considered.

## Protecting GRUB2

Bootloader protection does not seem to be redundant, even though the server is physically located in a room with limited access. This protection is easily implemented and does not lead to any increase in the server load in its normal operation mode.

First of all, it is need to create a separate configuration file in which the parameters implementing this protection would be accumulated. It should be located in the `/etc/grub.d` directory, and its name should start with such a numeric constant that it would be processed last. The contents of this file must match the template provided in the `/etc/grub.d/40_custom` file. Thus, in the system under consideration, the file has the name `/etc/grub.d/40_custom` and is filled in as follows:

```
#!/bin/sh
exec tail -n +3 $0
set superusers="root"
password_pbkdf2 root PBKDF2
```

The PBKDF2 literal is just a stub for the password checksum, which gives access to editing boot parameters on behalf of the root user.

When this file is ready, access to this file should be restricted and the stub should be replaced with a real checksum via executing the following commands (after the first command the password for GRUB should be entered twice):

```
root@server:/etc/grub.d# PBKDF2=$(grub-mkpasswd-pbkdf2 | grep -o grub.\*)
root@server:/etc/grub.d# sed s/PBKDF2/$PBKDF2/ -i /etc/grub.d/42_custom
root@server:/etc/grub.d# chmod u+x,go-rw /etc/grub.d/42_custom
```

Then some lines in the configuration file `/etc/default/grub` should be changed to the following:

```
GRUB_TIMEOUT=1
GRUB_DISABLE_RECOVERY="true"
```

Thus, all auxiliary items are removed from the main GRUB menu, and an interval of one second is set to initiate the transition to editing boot parameters.

The penultimate step is to update the loader menu using the commands:

```
root@server:/etc/grub.d# PATH=$PATH:/sbin
root@server:/etc/grub.d# update-grub
```

The first of them changes the value of the PATH system variable in the current session to provide the operation of the second, which forms the final configuration file `/boot/grub/grub.cfg`, which is used directly by the loader. Despite the fact that it is forbidden to edit this file manually, because all changes into it are lost when the update-grub command is called again, there is no other way out, since the currently available GRUB2 configuration mechanisms do not provide any other way for booting the system in normal mode without entering the set password.

To make boot without password possible it is just needed to add the `--unrestricted` key to the GRUB2 menu item used by default, which would lead this line to a appearance like:

```
menuentry 'Debian GNU/Linux' --unrestricted --class debian --class gnu-linux ...
```

Such changings should be performed every time the GRUB2 menu is updated, including if it happens during the installation of updates.

## Setting the system domain name

Taking into account the fact that IPv6 protocol is not supposed to be used in the system, to configure the full domain name, first **/etc/hosts** file should be filled, then the system name without the domain part should be placed in the **/etc/hostname** file and the command to change the host name in the current session should be executed. All the necessary commands, as well as the contents of the configuration files are presented in the following fragment of the terminal session. The last command in it demonstrates the configured full domain name of the system (FQDN):

```
root@server:/# cat /etc/hosts
127.0.0.1      localhost.localdomain localhost
127.0.1.1      server.account.school34 server
root@server:/# cat /etc/hostname
server
root@server:/# hostname -f
server.account.school34
```

## Disabling IPv6

Disabling IPv6 support in the system kernel can be performed both for the current session and on a permanent basis. To disable it on a permanent basis, the contents of the **/etc/sysctl.conf** file should be changed to a form that does not contradict the following lines in it:

```
#Disabling IPv6
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
```

To disable it only in the current session, it is enough to execute commands:

```
root@server:/# echo '1' > /proc/sys/net/ipv6/conf/all/disable_ipv6
root@server:/# echo '1' > /proc/sys/net/ipv6/conf/default/disable_ipv6
root@server:/# echo '1' > /proc/sys/net/ipv6/conf/lo/disable_ipv6
```

To check the success the network subsystem should be rebooted via command

```
root@server:/# systemctl restart networking
```

Then in the output of the command

```
root@server:/# ip address show
```

no IPv6 should be present:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether 18:d6:c7:00:ea:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.88/24 brd 192.168.0.255 scope global enp2s0
        valid_lft forever preferred_lft forever
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether 1c:6f:65:c2:c5:6f brd ff:ff:ff:ff:ff:ff
```

```
inet 10.9.1.129/26 brd 10.9.1.191 scope global enp3s0
    valid_lft forever preferred_lft forever
```

**Note:** Here is the output with a fully configured network subsystem. The description of the settings is given below, here it is enough to note only the absence of IPv6 addresses for interfaces.

## Disabling Magic SysRq

Disabling Magic SysRq may seem redundant or even harmful: for example, in case of a system failure, it would not be possible to stop it or flush the contents of buffers to disk. However, due to the physical availability of the server to users, there is, albeit a ghostly, probability that some attacker will press such key combination. This can lead to server shutdown and denial of customer service. At the same time, protection is organized so simply that it is not rational to neglect it.

To disable the "magic" combinations in the current session, the command is enough:

```
root@server:/# echo '0' > /proc/sys/kernel/sysrq
```

To disable it on a permanent basis, it is enough to bring the `/etc/sysctl.conf` file to a form that does not contradict the next line from it

```
kernel.sysrq=0
```

## GRE tunnel support

Tunneling using the GRE protocol is used to run one of the client applications on one of the workstations. To ensure the functioning of such a tunnel, it is necessary to ensure the loading of the necessary kernel modules and firewall support. Only measures related to kernel modules are described here, the network filter settings are given further in the corresponding section.

There are two approaches to loading modules: when the system is turned on and during its operation as needed. Based on the nature of the functioning of the network in question, it is most rational to apply the first approach. To do this, it is enough to provide additions to the `/etc/modules` file :

```
#Loading modules needed for GRE tunnel
ip-gre
ip-nat-pptp
```

## Creation of the users

Further, when describing the ssh secure remote login service, it is indicated that logging in remotely on behalf of the superuser is blocked. In order to still be able to manage remotely a regular user should be created, on whose behalf the login will be performed. After opening a session in the system under such account, there is, if necessary, an opportunity to raise root privileges with the help of su. Restrictions on system resources for this user are not currently being introduced, however, based on the experience operating the system operating, this can be done in the future.

The creation of such user (let his/her login be administrator) can be performed both at the system installation stage and after, for example, using the command

```
root@server:/# /usr/sbin/adduser administrator
```

Besides creating a user it is necessary to pay attention to several aspects of his work in the system.

Firstly, by default the newly created home directory of this user is readable by everyone in the system. If this is unacceptable the situation should be corrected using the command

```
root@server:/# chmod 0750 /home/administrator
```

leaving, thus, full access to the directory for the user himself and read access for members of his group (nobody in the system besides himself anyway). For everyone else access would be closed.

The second aspect is the sudo usage. This command allows to execute commands with administrative privileges without logging in as a superuser. This approach is fraught with some danger: an attacker, knowing the username and password of such user, can get very extensive rights in the system (if sudo policies are configured) or almost unlimited if sudo settings are left default. To solve this problem, it is the easiest (and most optimal on such server) to use the following approach: do not use sudo at all, forcing the user to enter (and, accordingly, know) the superuser password in order to increase his/her abilities in the system.

Thus, in order to deprive the user of the possibility of using sudo, it is enough to exclude him from the wheel group, if he is a member of it. This can be done with the usermod command, leaving the administrator user only in his (eponymous) group. The success of the changes can be verified by executing the groups command on behalf of this user strictly after re-logging in. The following is a fragment of a terminal session demonstrating all these operations:

```
root@server:/# /usr/sbin/usermod -G administrator administrator
root@server:/# su guest
administrator@server:/$ groups
administrator
administrator@server:/$ exit
exit
root@server:/#
```

Also, to organize the file server work, it is useful to create a separate user (and in general, users) on whose behalf all operations with network directories and their contents would be performed. This user should not be able to log in, he/she does not need a command interpreter and a home directory. The creation of such user is performed by the command:

```
root@server:/# /sbin/adduser buh --shell /dev/null --no-create-home --disabled-login
```

For the same purposes it is necessary to increase the maximum allowable number of simultaneously open file descriptors for the user due to the fact that by default in Linux this value is 1024, whereas in Windows it is 16384. Since all clients of the server in question are running Windows, and the smbd system service also sets this value in the system when it starts, it is rational to set it manually in advance and document this change. To do this it is needed to enter the following lines in the `/etc/security/limits.conf` file:

```
* - nofile 16384
root - nofile 16384
```

After that the system should be restarted. Of course, it is possible to change this parameter in the current session of the system and reboot later. To change the limit only in the current session it is enough to run the command:

```
root@server:/# ulimit -n 16384
```

To check the changes both in the current session and after restarting the system a command is used:

```
root@server:/# ulimit -Hn -Sn
open files          (-n) 16384
open files          (-n) 16384
```

## Configuring access control lists

Many daemons use the files **/etc/hosts.allow** and **/etc/hosts.deny** as sources of information about who is allowed to use the services of these daemons. These configuration files are part of the tcp wrappers mechanism. In addition, a tcpd daemon can be installed on the system, which performs such check itself before transmitting the network packet initiating the connection to the target daemon.

To find out whether a some service daemon uses this mechanism, it is usually enough to make sure that the daemon executable file is built using the libwrap library. The following fragment of the terminal session demonstrates that the sshd daemon uses this library, but dnsmasq, ntpd, NUT components and smbd do not:

```
root@server:/# ldd /sbin/upsd | grep libwrap
root@server:/# ldd /sbin/upsdrvctl | grep libwrap
root@server:/# ldd /sbin/upsmon | grep libwrap
root@server:/# ldd /sbin/upssched | grep libwrap
root@server:/# ldd /sbin/smbd | grep libwrap
root@server:/# ldd /sbin/ntpd | grep libwrap
root@server:/# ldd /sbin/dnsmasq | grep libwrap
root@server:/# ldd /sbin/sshd | grep libwrap
libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007fa52ea65000)
```

However, despite the fact that NUT subsystem programs do not use the libwrap library, they nevertheless use these configuration files.

To check whether the tcpd daemon is installed on the system the command can be used

```
root@server:/# dpkg -l | grep tcpd
```

As can be seen from the empty response of the command, this daemon is not installed.

In general, such check is performed by a firewall, and the tcp wrappers mechanism is considered obsolete and has given way to network screens. Nevertheless, given the simplicity of configuration and the low demand for resources, it makes sense to make entries in the specified configuration files for all the daemons used by the system, thereby leaving the decision to use this mechanism to the developers of daemons and the distribution maintainers. In other words, if they decide to enable support in a new version of the software, the customized system will be automatically ready for such changes. In addition, no software is immune from developer errors and vulnerabilities. The use of this mechanism may prove to be an additional line of protection when the network filter is compromised.

It does not make any sense to install tcpd into this system because ssh uses this mechanism on its own, and the other daemons simply do not accept connections on the external interface.

Knowing that ssh is allowed to connect only from the system administrator's computer with the address 192.168.0.2, using the uninterruptible power supply protocol only from the loop interface, and using the ntpd, smbd/nmbd and dns/dhcp protocols only from the internal and

loop interfaces, the **/etc/hosts.allow** file should be brought to the following state (comments omitted for brevity):

```
ntpd : 10.9.1.128/26 127.0.0.1
nmbd : 10.9.1.128/26 127.0.0.1
smbd : 10.9.1.128/26 127.0.0.1
dnsmasq : 10.9.1.128/26 127.0.0.1
sshd : 192.168.0.2
upsd : upsmon@localhost
```

In turn, the **/etc/hosts.deny** file prohibits any other connections to the server (comments are also not given):

```
ALL: ALL
```

To ensure that ssh does not accept connections from foreign hosts it is needed to restart this service with the command

```
root@server:/# systemctl restart sshd
```

and to try to connect via ssh from any host other than the subnet server. Obviously, at the time of the experiment, the firewall should not block such a connection.

Of course, such security parameters should be configured before the server is connected to the network, and finally checked immediately after, if possible, protecting the network segment being checked by other means (for example, by a firewall of a higher-level server of the organization).

## Disabling unnecessary services

Observing the rule of necessary sufficiency, you should disable and/or delete all unused services, for example: rsh, telnet, rpcbind. To check whether these services are installed, you can use one of (or a combination of) the following (and possibly some additional) commands:

```
root@server:/# dpkg -l | grep rpcbind
root@server:/# which rsh
root@server:/# ss -l | grep telnet
```

None of these methods is universal in the sense that it clearly shows the presence or absence of a particular service. As an example, the following fragment of a terminal session can be given:

```
root@server:/# which rsh
/usr/bin/rsh
root@server:/# ss -Htlp
LISTEN 0 16 127.0.0.1:nut 0.0.0.0:* users:(("upsd",pid=477,fd=4))
LISTEN 0 50 10.9.1.129:netbios-ssn 0.0.0.0:* users:(("smbd",pid=1411,fd=34))
LISTEN 0 32 127.0.0.1:domain 0.0.0.0:* users:(("dnsmasq",pid=465,fd=9))
LISTEN 0 32 10.9.1.129:domain 0.0.0.0:* users:(("dnsmasq",pid=465,fd=7))
LISTEN 0 128 192.168.0.88:ssh 0.0.0.0:* users:(("sshd",pid=481,fd=3))
LISTEN 0 50 10.9.1.129:microsoft-ds 0.0.0.0:* users:(("smbd",pid=1411,fd=33))
root@server:/# dpkg -l | grep rsh
root@server:/# ls -l /usr/bin | grep rsh
lrwxrwxrwx 1 root root 21 мая 4 18:27 rsh -> /etc/alternatives/rsh
root@server:/# ls -l /etc/alternatives | grep rsh
lrwxrwxrwx 1 root root 12 мая 4 18:27 rsh -> /usr/bin/ssh
lrwxrwxrwx 1 root root 28 мая 4 18:27 rsh.1.gz -> /usr/share/man/man1/ssh.1.gz
root@server:/#
```

Analyzing the fragment above, we can note the following:

The first command shows that the rsh executable file is present in the system, but the second command shows that the rsh daemon is not listening to any tcp ports. The following command informs you that there is not any package installed in the system whose name mentions rsh. When viewing the properties of an executable file in detailed mode, it turns out that it is a symbolic link to another file, which is also a symbolic link to an ssh executable file. Thus, rsh is not installed in this system, instead an alias for ssh is created.

Among other things, a telnet package containing only the client part was installed in the system, but since it is not required for the system to work, the package should be deleted:

```
root@server:/# apt-get purge telnet
```

After restarting the system, you can make sure that the system now listens only to the ports it needs to perform its tasks (including services, the installation and configuration of which are described later in this document):

```
root@server:/# ss -tulp
```

Netid	State	Local Address:Port	Peer Address:Port
udp	UNCONN	127.0.0.1:domain	0.0.0.0:* users:(("dnsmasq",pid=465,fd=8))
udp	UNCONN	10.9.1.129:domain	0.0.0.0:* users:(("dnsmasq",pid=465,fd=6))
udp	UNCONN	0.0.0.0:bootps	0.0.0.0:* users:(("dnsmasq",pid=465,fd=4))
udp	UNCONN	192.168.0.88:ntp	0.0.0.0:* users:(("ntpd",pid=451,fd=22))
udp	UNCONN	10.9.1.129:ntp	0.0.0.0:* users:(("ntpd",pid=451,fd=18))
udp	UNCONN	127.0.0.1:ntp	0.0.0.0:* users:(("ntpd",pid=451,fd=17))
udp	UNCONN	0.0.0.0:ntp	0.0.0.0:* users:(("ntpd",pid=451,fd=16))
udp	UNCONN	10.9.1.191:netbios-ns	0.0.0.0:* users:(("nmbd",pid=1421,fd=17))
udp	UNCONN	10.9.1.129:netbios-ns	0.0.0.0:* users:(("nmbd",pid=1421,fd=16))
udp	UNCONN	0.0.0.0:netbios-ns	0.0.0.0:* users:(("nmbd",pid=1421,fd=14))
udp	UNCONN	10.9.1.191:netbios-dgm	0.0.0.0:* users:(("nmbd",pid=1421,fd=19))
udp	UNCONN	10.9.1.129:netbios-dgm	0.0.0.0:* users:(("nmbd",pid=1421,fd=18))
udp	UNCONN	0.0.0.0:netbios-dgm	0.0.0.0:* users:(("nmbd",pid=1421,fd=15))
udp	UNCONN	0.0.0.0:54728	0.0.0.0:* users:(("dnsmasq",pid=465,fd=10))
tcp	LISTEN	127.0.0.1:nut	0.0.0.0:* users:(("upsd",pid=477,fd=4))
tcp	LISTEN	10.9.1.129:netbios-ssn	0.0.0.0:* users:(("smbd",pid=1411,fd=34))
tcp	LISTEN	127.0.0.1:domain	0.0.0.0:* users:(("dnsmasq",pid=465,fd=9))
tcp	LISTEN	10.9.1.129:domain	0.0.0.0:* users:(("dnsmasq",pid=465,fd=7))
tcp	LISTEN	192.168.0.88:ssh	0.0.0.0:* users:(("sshd",pid=481,fd=3))
tcp	LISTEN	10.9.1.129:microsoft-ds	0.0.0.0:* users:(("smbd",pid=1411,fd=33))

In the presented fragment, the Recv-Q and Send-Q columns have been removed solely for formatting reasons due to their insignificant informativeness in the context of the presentation.

## Configuring the list of repositories

Immediately after installing the system to be able to install additional software it is needed to fill in the list of repositories used by the system, to register the public keys of local repositories and to configure the network adapters of the system. The description of the network interface settings is given below to preserve the harmony and integrity of the presentation. There are no obstacles to setting up a list of repositories and installing additional software after setting up a wired network.

The configuration file **/etc/apt/sources.list** containing a list of repositories should be given the form:



```
deb http://debian.service.school34/buster/ buster main contrib non-free
deb http://debian.service.school34/security/ buster/updates main contrib
deb http://debian.service.school34/ppa/ buster main contrib
```

Then the public part of the local repository key (ppa) should be obtained, for example, by downloading the repository\_key.asc file from it with the command:

```
root@server:/# cd /tmp && wget http://debian.service.school34/ppa/repository_key.asc
```

To use this key, the gnupg package must be present in the system. If there is none, i. e. the command

```
root@server:/tmp# dpkg -l | grep gnupg
```

returns an empty response, then it should be installed. To do this, run the following commands:

```
root@server:/tmp# apt-get update
root@server:/tmp# apt-get install gnupg
```

At the same time, the first of these commands will surely issue a warning that the key of the local ppa repository cannot be verified and therefore this repository would not be used.

Next, the key file of the local repository should be registered in the package management system with the command:

```
root@server:/tmp# apt-key add repository_key.asc
```

After that it is already possible to update information about packages stored in the specified repositories:

```
root@server:/tmp# apt-get update
```

There is no need to import the keys of the other repositories because they are mirrors of the official repositories and are installed with the system.

## Installing auxiliary software

The Midnight Commander file manager (mc package) and the tree, lshw and parted utilities can be very useful when managing the server. To install them, just run the command:

```
root@server:/# apt-get install mc tree lshw parted
```

To monitor the status of the server, the means of viewing data from hardware sensors are very useful. These tools are part of the lm-sensors package, which must be installed, and then the process of the sensors available in the system detection must be carried out. This can be done by running the following commands and following the instructions that appear:

```
root@server:/# apt-get install lm-sensors
root@server:/# sensors-detect
```

After that the data of these sensors can be obtained using the command

```
root@server:/# sensors
```

Moreover, this command can be executed even with the rights of an unprivileged user.

Another mean of monitoring the state of the system is the S.M.A.R.T technology – assessment of the state of the hard disk (or other data storage) by the built-in self-diagnostic equipment.

To use this technology, you need to install the smartmontools package, which contains utilities that allow you to receive data from the hard disk, interpret and display it, as well as perform disk testing. Also there is the smartd daemon included in the package (and the corresponding systemd service too), which automatically monitors the status of the disk and informs the system administrator. This is described in more detail in the section "Using the server".

You can install these software tools using the command:

```
root@server:/# apt-get install smartmontools
```

Their usage is also shown in the section "Using the server".

On systems with multiple network interfaces (as in the case under consideration) or if you need full control over the interface hardware, the ethtool utility is useful. To install it, just run the command:

```
root@server:/# apt-get install ethtool
```

## Configuring the UDEV subsystem

In the system under consideration, to ensure reliability and safety of operation, it will be necessary to fine-tune the udev subsystem, which notifies the system software about events occurring with devices, manages access rights to them, and can also create additional links for devices in the `/dev` system directory and rename network interfaces. According to the task statement, udev means should provide protection from connecting extraneous usb devices (other connectors for connecting removable media are not provided in the system hardware) to protect against unauthorized copying of data, as well as connecting a limited number of previously known smartphones in usb-modem mode to organize a backup Internet channel with automatic switching to this channel when connecting a smartphone or modem and back to the wired interface when disconnecting the USB device. Also, the usb bus interacts with an uninterruptible power supply, which requires not only authorization of this device, but some additional actions for the correct selection of the UPS driver.

## Configuration

Further, the description of the udev setup will be performed on the example of one test smartphone, while in the actual setup the number and models of smartphones would change.

First of all, you should collect all the necessary information about the allowed devices. To do this, each such device in turn should be connected to the system and detected by the command:

```
root@server:/# lsusb
Bus 002 Device 004: ID 22b8:2e76 Motorola PCS
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 005: ID 0764:0501 Cyber Power System, Inc. CP1500 AVR UPS
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

After that the command like

```
root@server:/# udevadm info -a --name=/dev/bus/usb/002/004 >> \
/etc/udev/rules.d/35-usb-filter.rules
```

would help to enter a detailed summary of the device in the file **/etc/udev/rules.d/usb-filter.rules**. This file has the extension **.rules** according to the requirements of the **systemd-udevd** daemon, but at the moment it does not contain the rules of operation for this daemon, but the information necessary to build them.

It is important to note that the smartphone should be switched to USB modem mode before collecting information about it, because most smartphones change their device ID depending on the operating mode.

The further set of rules is based on the principle of prohibiting everything except what is explicitly allowed, therefore, for the correct functioning of UPS and USB modems (or smartphones acting in their role), permissive rules must be drawn up not only for the devices themselves, but also for USB hubs to which they can be connected. In this case, any USB ports of the system are allowed to be used, so rules must be drawn up for each USB hub. To collect information about hubs, you can use the commands similar to the above:

```
root@server:/# udevadm info -a --name=/dev/bus/usb/001/001 >> \
/etc/udev/rules.d/35-usb-filter.rules
root@server:/# udevadm info -a --name=/dev/bus/usb/002/001 >> \
/etc/udev/rules.d/35-usb-filter.rules
root@server:/# udevadm info -a --name=/dev/bus/usb/003/001 >> \
/etc/udev/rules.d/35-usb-filter.rules
root@server:/# udevadm info -a --name=/dev/bus/usb/004/001 >> \
/etc/udev/rules.d/35-usb-filter.rules
root@server:/# udevadm info -a --name=/dev/bus/usb/005/001 >> \
/etc/udev/rules.d/35-usb-filter.rules
```

It should be noted that as a result of executing these commands, the received data will be redundant both in terms of the composition of device attributes and in displaying information about device No. 1 on bus No. 2 (Linux Foundation 2.0 root hub). The fact is that when forming a summary about a smartphone, it also includes a summary of the entire chain of devices through which this smartphone is connected.

Next, you should convert the information for the rules into the rules themselves. To do this, the file **/etc/udev/rules.d/35-usb-filter.rules** should be brought to a form similar to the accuracy of specific devices with the following:

```
#Enable only known devices
#Skip not USB
SUBSYSTEM!="usb", GOTO="usb_filter_end"
#Skip all action except add or change
ACTION!="add|change", GOTO="usb_filter_end"

#System usb 2.0 hubs
ATTR{product}=="EHCI Host Controller", ATTR{idVendor}=="1d6b",
ATTR{idProduct}=="0002", ATTR{authorized}="1", GOTO="usb_filter_end"

#System usb 1.1 hubs
ATTR{product}=="UHCI Host Controller", ATTR{idVendor}=="1d6b",
ATTR{idProduct}=="0001", ATTR{authorized}="1", GOTO="usb_filter_end"

#UPS Cyber Power
ATTR{product}=="UPS VALUE", ATTR{idVendor}=="0764", ATTR{idProduct}=="0501",
ATTR{authorized}="1", GOTO="usb_filter_end"
```

```

#Smartphone No 1
ATTR{product}=="SAMSUNG_Android", ATTR{serial}=="JVNEV983VK3",
ATTR{idVendor}=="04e8", ATTR{idProduct}=="6863", ATTR{authorized}="1",
GOTO="usb_filter_end"

#Smartphone No 2
ATTR{product}=="SAMSUNG_Android", ATTR{serial}=="bnjkfbnk33093nvk3b",
ATTR{idVendor}=="04e8", ATTR{idProduct}=="6863", ATTR{authorized}="1",
GOTO="usb_filter_end"

#Smartphone No 3
ATTR{product}=="HRY-LX1", ATTR{serial}=="VKJV80924NBKV892K", ATTR{idVendor}=="12d1",
ATTR{idProduct}=="108a", ATTR{authorized}="1", GOTO="usb_filter_end"

#Smartphone No 4
ATTR{product}=="ATOLL-AB-IDP_SN:2AD1B6A8", ATTR{serial}=="vk2924vn",
ATTR{idVendor}=="2717", ATTR{idProduct}=="ff80", ATTR{authorized}="1",
GOTO="usb_filter_end"

#Disable all other usb devices
SUBSYSTEMS=="usb", ACTION=="add|change", ENV{DEVTYPE}=="usb_device",
ATTR{authorized}="0"

LABEL="usb_filter_end"

```

The first lines of this file indicate that it handles only add (connection) or change events for usb subsystem devices, other events and subsystems are ignored by this set of rules. Then, for each device, a number of attributes are defined that play the role of a unique device identifier. In the context under consideration, such an identifier is a tuple consisting of the device name (ATTR{product}), its serial number (ATTR{serial}) and manufacturer identifiers (ATTR{idVendor}) and the device (ATTR{idProduct}). By linking together this data with the permission to use the device (ATTR{authorized}="1"), a rule for udevd is formed. A small exception has been made for the uninterruptible power supply: the serial number of the device is not taken into account. A ban is set at the end of the file (ATTR{authorized}="0") to connect any other devices.

Based on the data from this file, you can form a second file **/etc/udev/rules.d/77-net-alias.rules**:

```

#Making common alias for usb interfaces
#Skip not net
SUBSYSTEM!="net", GOTO="net_alias_end"
#Skip all action except add or change
ACTION!="add|change", GOTO="net_alias_end"

#Smartphone No 1
ENV{ID_VENDOR_ID}=="04e8", ENV{ID_MODEL_ID}=="6863", NAME="ethusb",
GOTO="net_alias_end"
#Smartphone No 2
ENV{ID_VENDOR_ID}=="04e8", ENV{ID_MODEL_ID}=="6863", NAME="ethusb",
GOTO="net_alias_end"
#Smartphone No 3
ENV{ID_VENDOR_ID}=="12d1", ENV{ID_MODEL_ID}=="108a", NAME="ethusb",
GOTO="net_alias_end"
#Smartphone No 4
ENV{ID_VENDOR_ID}=="2717", ENV{ID_MODEL_ID}=="ff80", NAME="ethusb",
GOTO="net_alias_end"

LABEL="net_alias_end"

```

This file is structurally similar to the previous one and is built according to recommendations given in **/usr/share/doc/udev/README.Debian.gz**. A naming scheme for udev configuration files was selected according to this file too. So the file `77-net-alias.rules` has the prefix 77 in order to be processed after the system file `75-net-descriptor.rules`, but before the system `80-net-setup-link.rules`. The file `35-usb-filter.rules` also has the prefix 35 to be processed before other files of the usb subsystem. Finally, the third file **/etc/udev/rules.d/99-ups.rules** should be processed by one of the latter and prevent the capture of the UPS operating under the USB HID protocol by the corresponding HID driver, thereby leaving the possibility for NUT to take control of the device. This is described in more details below in the section "Configuring NUT". This file contains only one rule:

```
SUBSYSTEMS=="usb", DRIVERS=="usbhid", ACTION=="add", ATTR{idVendor}=="0764", \
ATTR{idProduct}=="0501", ATTR{authorized}="0"
```

After editing the configuration files finished, restart the `udev` daemon and make sure that it recognizes the new configuration:

```
root@server:/# systemctl restart systemd-udev
root@server:/# systemctl status systemd-udev
• systemd-udev.service - udev Kernel Device Manager
  Loaded: loaded (/lib/systemd/system/systemd-udev.service; static; vendor preset:
  enabled)
  Active: active (running) since Thu 2021-05-06 10:34:47 MSK; 8s ago
    Docs: man:systemd-udev.service(8)
          man:udev(7)
 Main PID: 791 (systemd-udev)
   Status: "Processing with 15 children at max"
    Tasks: 1
  Memory: 1.1M
   CGroup: /system.slice/systemd-udev.service
           └─791 /lib/systemd/systemd-udev

мая 06 10:34:47 server systemd[1]: Starting udev Kernel Device Manager...
мая 06 10:34:47 server systemd[1]: Started udev Kernel Device Manager.
```

To ensure that the rules work not only when devices are hot-connected, but also when the system boots, it is necessary to update the `initramfs` image:

```
root@server:/# PATH=$PATH:/sbin
root@server:/# update-initramfs -u
```

Changing the `PATH` environment variable was required to ensure the operability of scripts called during `update-initramfs`.

## Fail protection (auto-resume)

The management of services in the system is handled by the boot manager and `systemd` services, the latter ensures (with proper configuration) the automatic resumption of services in case of a crash. In the system under consideration, such setup for the `systemd-udev` service was performed initially by the operating system distribution maintainers.

## Verification of performance and stability

Check of the service reliability could be done according to the following scenario:

- shutdown the system;

- connect to it a smartphone (in USB modem mode), a USB keyboard and a USB drive;
- start the system and make sure that the ethusb network interface is detected in the system, but neither the keyboard nor the drive are detected;
- disconnect all the devices from the system and connect them again, making sure that the result remains unchanged.

After turning on the system and connecting to it remotely via ssh (described below), the desired result is observed: the keyboard receives power via the USB bus, but pressing its buttons is ignored by the system, the flash drive is connected to the system, but it is not recognized as a block device, the smartphone is identified as ethusb network interface. This can be figured out from the next fragment of the terminal session:

```
root@server:/# lsusb
Bus 005 Device 004: ID 8564:1000 Transcend Information, Inc. JetFlash
Bus 005 Device 003: ID 22b8:2e25 Motorola PCS
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 002: ID 1a2c:4c5e China Resource Semico Co., Ltd
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
root@server:/# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda      8:0    0 298,1G  0 disk
├─sda1    8:1    0  18,6G  0 part /
├─sda2    8:2    0   3,7G  0 part [SWAP]
├─sda3    8:3    0  18,6G  0 part /var
└─sda4    8:4    0 257,1G  0 part /srv/samba
root@server:/# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0: <BROADCAST,MULTICAST> mtu 1500 qdisc htb state DOWN mode DEFAULT group
default qlen 1000
    link/ether 18:d6:c7:00:ea:6c brd ff:ff:ff:ff:ff:ff
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT group default qlen 1000
    link/ether 1c:6f:65:c2:c5:6f brd ff:ff:ff:ff:ff:ff
4: gre0@NONE: <NOARP> mtu 1476 qdisc noop state DOWN mode DEFAULT group default qlen
1000
    link/gre 0.0.0.0 brd 0.0.0.0
5: gretap0@NONE: <BROADCAST,MULTICAST> mtu 1476 qdisc noop state DOWN mode DEFAULT
group default qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
6: erspan0@NONE: <BROADCAST,MULTICAST> mtu 1464 qdisc noop state DOWN mode DEFAULT
group default qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
7: ethusb: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN
mode DEFAULT group default qlen 1000
    link/ether 36:e1:49:c8:cb:1c brd ff:ff:ff:ff:ff:ff
```

After disconnecting the devices and reconnecting them to a running system the result is identical.

## Configuring network

This section describes only the configuration of the system network interfaces, the description of firewall system is given below. Network setup can be divided into the following steps:

- installation of necessary software;
- configuring network interfaces;
- configuring packet routing and forwarding;
- configuring the bandwidth management system;

For the full operation of one of the network cards in the system under consideration, it is needed to install the firmware-realtek package. This can be done immediately after configuring the wired network interface or by transferring the package manually from a removable media. With a configured network interface and a list of repositories, you can perform the installation with the command:

```
root@server:/# apt-get install firmware-realtek
```

The rest necessary tools for network configuration (iproute2, tc) are installed with the system.

## Network interfaces configuration

First of all, you need to find out the exact set of the network interfaces available to the system, as well as make sure that all the software necessary for their operation is installed and that the interfaces themselves are switched to the most productive operating modes. The following fragment of the terminal session shows that there are seven network interfaces in the system at the time: a loopback, two wired, three virtual ones connected to the GRE tunnel, and a smartphone in USB modem mode:

```
root@server:/# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0: <BROADCAST,MULTICAST> mtu 1500 qdisc htb state DOWN mode DEFAULT group
default qlen 1000
    link/ether 18:d6:c7:00:ea:6c brd ff:ff:ff:ff:ff:ff
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT group default qlen 1000
    link/ether 1c:6f:65:c2:c5:6f brd ff:ff:ff:ff:ff:ff
4: gre0@NONE: <NOARP> mtu 1476 qdisc noop state DOWN mode DEFAULT group default qlen
1000
    link/gre 0.0.0.0 brd 0.0.0.0
5: gretap0@NONE: <BROADCAST,MULTICAST> mtu 1476 qdisc noop state DOWN mode DEFAULT
group default qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
6: erspan0@NONE: <BROADCAST,MULTICAST> mtu 1464 qdisc noop state DOWN mode DEFAULT
group default qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
7: ethusb: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN
mode DEFAULT group default qlen 1000
    link/ether 36:e1:49:c8:cb:1c brd ff:ff:ff:ff:ff:ff
```

Before configuring network interfaces to increase the stability of the server in case of switching errors on the network equipment, it is necessary to make some additions to the `/etc/sysctl.conf` file:

```
#Securing arp
net.ipv4.conf.all.arp_ignore=2
net.ipv4.conf.all.arp_announce=2
```

The string `arp_ignore` means that an arp request for ip address resolution is served only if it came to an interface running under this address, and the ip address of the sender of the request are in the same subnet.

The string `arp_announce` means that an outgoing udp packet can only be sent from the interface that contains the advertised ip address.

Thus, if, for example, a host with an address corresponding to the accounting subnet appears in the external network, it will not create an address conflict for the server.

These changes will be applied after the server is restarted. To accept them in the current session, execute commands like:

```
root@server:/# echo 2 > /proc/sys/net/ipv4/conf/all/arp_ignore
root@server:/# echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
```

Network interfaces are configured by entering the necessary information in the `/etc/network/interfaces` file. In this case it has the following form (with some insignificant abbreviations):

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# Main external interface
auto enp2s0
iface enp2s0 inet static
    address 192.168.0.88/24
    gateway 192.168.0.1
    pre-up /etc/network/neighbors kernel
    post-up /etc/network/routes up
    post-down /etc/network/routes down

# Main internal interface
auto enp3s0
iface enp3s0 inet static
    address 10.9.1.129/26
    pre-up /etc/network/neighbors account

# Reserve external interface
allow-hotplug ethusb
iface ethusb inet dhcp
    pre-up ip link set enp2s0 down
    pre-up /etc/network/dns usb-pre-up
    post-up /etc/network/dns usb-post-up
    pre-down /etc/network/dns usb-pre-down
    post-down ip link set enp2s0 up
    post-down ip route add default via 192.168.0.1
    post-down /etc/network/dns usb-post-down
```

Thus, the network interface acting as an external one (relative to clients) receives an address in the core of the network, and the internal one receives an address in the accounting subnet.

To apply the new configuration file, restart the network subsystem with the command:

```
root@server:/# systemctl restart networking
```



Then you should ensure that the restart is successful after the output of the command:

```
root@server:/# systemctl status networking
```

```
• networking.service - Raise network interfaces
  Loaded: loaded (/lib/systemd/system/networking.service; enabled; vendor preset:
enabled)
  Active: active (exited) since Tue 2021-05-11 11:43:22 MSK; 2h 3min ago
    Docs: man:interfaces(5)
 Main PID: 704 (code=exited, status=0/SUCCESS)
   Tasks: 0 (limit: 2315)
  Memory: 0B
   CGroup: /system.slice/networking.service
```

```
мая 11 11:43:21 server systemd[1]: Starting Raise network interfaces...
```

```
мая 11 11:43:22 server systemd[1]: Started Raise network interfaces.
```

Finally, you can see the new network settings of the system in the output of the command:

```
root@server:/# ip address show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 1c:6f:65:c2:c5:6f brd ff:ff:ff:ff:ff:ff
    inet 10.9.1.129/26 brd 10.9.1.191 scope global enp3s0
        valid_lft forever preferred_lft forever
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc htb state UP group
default qlen 1000
    link/ether 18:d6:c7:00:ea:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.88/24 brd 192.168.0.255 scope global enp2s0
        valid_lft forever preferred_lft forever
4: gre0@NONE: <NOARP> mtu 1476 qdisc noop state DOWN group default qlen 1000
    link/gre 0.0.0.0 brd 0.0.0.0
5: gretap0@NONE: <BROADCAST,MULTICAST> mtu 1476 qdisc noop state DOWN group default
qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
6: erspan0@NONE: <BROADCAST,MULTICAST> mtu 1464 qdisc noop state DOWN group default
qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

As can be seen from this output, there are three network interfaces in the system, considering the loop, they use exactly the addresses that were assigned to them, while IPv6 addresses are not assigned. Thus, the network interface settings fully comply with the requirements set above. Also, the output indicates the currently inactive GRE tunnel, which was mentioned earlier.

In addition, four scenarios are executed before and during the launch of network interfaces: adding static entries to link-level address tables, configuring routing, configuring bandwidth for client connections, and configuring the domain name service (only when starting or stopping a backup Internet connection).

All scripts are located in the same directory **/etc/network** as the main configuration file of network interfaces, and are available for reading, writing and execution only to the superuser. These files creation and their attributes setting can be performed by commands (only the described files are displayed in the output of the last one):

```

root@server:/# cd /etc/network
root@server:/etc/network# touch shaping routes neighbours dns
root@server:/etc/network# chmod 0700 shaping routes neighbours dns
root@server:/etc/network# ls -l
итого 28
-rwx----- 1 root root    0 мая 11 14:25 dns
...
-rw-r--r-- 1 root root 1167 мая 11 14:03 interfaces
...
-rwx----- 1 root root    0 мая 11 14:25 neighbours
-rwx----- 1 root root    0 мая 11 14:25 routes
-rwx----- 1 root root    0 мая 11 14:25 shaping

```

At the moment, the script for making static arp entries looks like:

```

#!/bin/sh

if [ $1 = "kernel" ]
then
    ip neighbour add 192.168.0.1 dev enp2s0 lladdr 00:0c:26:a7:b9:39 nud permanent
    ip neighbour add 192.168.0.2 dev enp2s0 lladdr 90:2b:34:48:08:b5 nud permanent
    ip neighbour add 192.168.0.77 dev enp2s0 lladdr 54:be:f7:28:4b:40 nud permanent
elif [ $1 = "account" ]
then
    ip neighbour add 10.9.1.131 dev enp3s0 lladdr 90:2b:34:a3:83:44 nud permanent
    ip neighbour add 10.9.1.132 dev enp3s0 lladdr 50:e5:49:33:de:d9 nud permanent
    ip neighbour add 10.9.1.133 dev enp3s0 lladdr 90:2b:34:a0:24:7b nud permanent
    ip neighbour add 10.9.1.135 dev enp3s0 lladdr 90:2b:34:96:16:53 nud permanent
    ip neighbour add 10.9.1.134 dev enp3s0 lladdr f8:0d:ac:78:8a:f1 nud permanent
fi
exit 0

```

As the hardware on the virtualization server, main gateway, or any other node specified in this scenario is replaced, their hardware addresses must be replaced.

The output of the following command executed after restarting the network subsystem demonstrates the fact of successful entry of static hardware addresses entries:

```

root@server:/# ip neighbour show
192.168.0.1 dev enp2s0 lladdr 00:0c:26:a7:b9:39 PERMANENT
192.168.0.2 dev enp2s0 lladdr 90:2b:34:48:08:b5 PERMANENT
192.168.0.77 dev enp2s0 lladdr 54:be:f7:28:4b:40 PERMANENT
10.9.1.132 dev enp3s0 lladdr 50:e5:49:33:de:d9 PERMANENT
10.9.1.134 dev enp3s0 lladdr f8:0d:ac:78:8a:f1 PERMANENT
10.9.1.131 dev enp3s0 lladdr 90:2b:34:a3:83:44 PERMANENT
10.9.1.133 dev enp3s0 lladdr 90:2b:34:a0:24:7b PERMANENT
10.9.1.135 dev enp3s0 lladdr 90:2b:34:96:16:53 PERMANENT

```

The scenario for configuring the domain name service is described below in the section "DNS/DHCP Service".

## Routing

It should be noted that the communication of devices in the local network of the organization is implemented by splitting the local network into isolated subnets protected by their firewalls, simultaneously acting as servers of such subnets. One of these servers (for the accounting subnet) is the system in question. At the same time, these servers are responsible for routing packets and performing address translation (NAT). From the routing point of view, each such

server should, on the one hand, provide clients with the shortest route to the target node of the network, and on the other hand, not provide a route to subnets which are not supposed to interact with the subnet in question by the logic of the network and organization. In this case, the accounting network server should only provide clients with routes to the network core (including the main gateway and the AIS "Avers" server) and to the network of virtual servers. A default route through the organization's main gateway should also be provided. The server should not perform protection against attempts to get into other subnets using routing tables of other servers, for example, the main gateway, because protection against such attacks is on duty of this gateway firewall itself. Further, in the section "Installing and configuring the network filter (nftables)", measures are described to counteract the use of this server for the same purpose.

Regarding the issue of address translation, it should be noted that NAT has no effect on routing within the local network. When accessing external resources, symmetric NAT is performed on the main gateway of the organization. Within a local network such transformation can have both positive and negative features in its application. The negative aspects include the increased load on the machine performing this conversion, and the inability to find out the source IP address of the client in the logs of external (relative to the subnet) servers. The same impossibility in some cases should be considered a virtue. For example, using NAT on the accounting subnet server, you can achieve a fairly reliable concealment of information about the number of hosts in this subnet, their internal addresses, ports used by each of them, etc. As a result, it will become impossible to determine from which of the accounting department computers the connection to the local information server was made. When applying the same mechanism to a wireless subnet, the result will be hiding the traces of the attacker client in the logs of the same info server. That is why NAT is not used in the wireless network, but it is used in the accounting subnet.

Additional network routes should be added to the routes created by default based on network interface settings in order to send packets along the shortest path. To do this it is enough to bring the previously mentioned routing configuration scenario to the form:

```
root@server:/# cat /etc/network/routes
#!/bin/sh

if [ $1 = "up" ]
then
    ip route add 192.168.7.0/24 via 192.168.0.77 dev enp2s0
elif [ $1 = "down" ]
then
    ip route del 192.168.7.0/24 via 192.168.0.77 dev enp2s0
fi
exit 0
```

After restarting the network subsystem using the following commands you can verify both the availability and the operability of the added route:

```
root@server:/# ip route show
default via 192.168.0.1 dev enp2s0 onlink
10.9.1.128/26 dev enp3s0 proto kernel scope link src 10.9.1.129
192.168.0.0/24 dev enp2s0 proto kernel scope link src 192.168.0.88
192.168.7.0/24 via 192.168.0.77 dev enp2s0
root@server:/# traceroute 192.168.7.1
traceroute to 192.168.7.1 (192.168.7.1), 30 hops max, 60 byte packets
1  server.service.school34 (192.168.7.1)  0.625 ms  0.581 ms  0.559 ms
root@server:/# ping -c 3 debian.service.school34
```

```

PING debian.service.school34 (192.168.7.5) 56(84) bytes of data.
64 bytes from debian.service.school34 (192.168.7.5): icmp_seq=1 ttl=63 time=4.51 ms
64 bytes from debian.service.school34 (192.168.7.5): icmp_seq=2 ttl=63 time=1.56 ms
64 bytes from debian.service.school34 (192.168.7.5): icmp_seq=3 ttl=63 time=1.12 ms

--- debian.service.school34 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 1.116/2.394/4.513/1.509 ms

```

The routing table presented by the first command also contains routes created based on network interface settings. Together, they make up a complete set of routes that the server needs to work.

## Network bandwidth management

The bandwidth setting for client connections is performed by means of the traffic control mechanism and the tc utility based on the following considerations:

- traffic is divided into three classes: connections to the server of the Avers system, connections to the internal virtualization server and connections to Internet resources
- for traffic to the server of the Avers system, a bandwidth of 20 Mbit/s is guaranteed with the possibility of its expansion up to 50 Mbit/s
- for traffic to external Internet resources, a bandwidth of 2 Mbit/s is guaranteed with the possibility of its expansion up to 10 Mbit/s
- 100 Mbit/s bandwidth is guaranteed for traffic to internal servers with the possibility of its expansion up to 1000 Mbit/s
- the bandwidth for traffic to the server of the Avers system is limited from above to 50 Mbit/s in order to limit the maximum load on this server from accounting workstations, since the data provided by this server is in demand by accounting occasionally, and other subnets create a significant load for this server
- the bandwidth to external Internet resources is limited from above by the width of the external channel of the school
- the bandwidth for traffic to the internal servers of the school is limited from above by the bandwidth of the wired network interface of the described system
- overcommitting, i. e. setting the total guaranteed bandwidth exceeding the hardware capabilities of the interface, is not allowed

The bandwidth setup scenario for transit network connections based on the above considerations takes the form:

```

#!/bin/sh

#Сбрасываем текущее состояние
tc qdisc delete dev enp2s0 root
#Ставим корневую дисциплину
tc qdisc add dev enp2s0 root handle 1: htb default 13
#Создаём корневой класс
tc class add dev enp2s0 parent 1: classid 1:1 htb rate 1000mbit ceil 1000mbit
#Создаём подклассы: avers, service, inet
tc class add dev enp2s0 parent 1:1 classid 1:11 htb rate 20mbit ceil 50mbit
tc class add dev enp2s0 parent 1:1 classid 1:12 htb rate 100mbit ceil 1000mbit
tc class add dev enp2s0 parent 1:1 classid 1:13 htb rate 2mbit ceil 10mbit
#Настраиваем дисциплины для подклассов
tc qdisc add dev enp2s0 parent 1:11 handle 10:0 sfq perturb 10
tc qdisc add dev enp2s0 parent 1:12 handle 20:0 sfq perturb 10
tc qdisc add dev enp2s0 parent 1:13 handle 30:0 sfq perturb 10
#Настраиваем фильтры для классификации трафика (по умолчанию - inet)

```

```
tc filter add dev enp2s0 protocol ip parent 1:0 prio 1 u32 match ip dst 192.168.0.4
flowid 1:11
tc filter add dev enp2s0 protocol ip parent 1:0 prio 1 u32 match ip dst
192.168.7.0/24 flowid 1:12
```

First of all, the script deletes the current root discipline along with all its components and connects htb (Hierarchical Token Bucket) as the root discipline, while specifying that all non-classified (default) traffic should be processed using Class 1:13 disciplines (Internet traffic). Then a root class is created, where all traffic will fall (this is necessary for the implementation of borrowing). In this class, the bandwidth is limited according to the hardware capabilities. Next, three subclasses are created to provide required channel width separation. After that, three sfq (Stochastic Fairness Queueing) disciplines are created, one for each class. Finally, two filters are created: the first classifies traffic to the AIS "Avers" server, the second – to the virtualization server. Unclassified traffic is considered to be the traffic directed to the Internet.

After running the script, you can check the traffic control subsystem status with the following commands:

```
root@server:/# /sbin/tc qdisc show
qdisc noqueue 0: dev lo root refcnt 2
qdisc pfifo_fast 0: dev enp3s0 root refcnt 2 bands 3 priomap 1 2 2 2 1 2 0 0 1 1 1 1
1 1 1 1
qdisc htb 1: dev enp2s0 root refcnt 2 r2q 10 default 0x13 direct_packets_stat 0
direct_qlen 1000
qdisc sfq 30: dev enp2s0 parent 1:13 limit 127p quantum 1514b depth 127 divisor 1024
perturb 10sec
qdisc sfq 10: dev enp2s0 parent 1:11 limit 127p quantum 1514b depth 127 divisor 1024
perturb 10sec
qdisc sfq 20: dev enp2s0 parent 1:12 limit 127p quantum 1514b depth 127 divisor 1024
perturb 10sec
root@server:/# /sbin/tc class show dev enp2s0
class htb 1:11 parent 1:1 leaf 10: prio 0 rate 20Mbit ceil 50Mbit burst 1600b cburst
1600b
class htb 1:1 root rate 1Gbit ceil 1Gbit burst 1375b cburst 1375b
class htb 1:13 parent 1:1 leaf 30: prio 0 rate 2Mbit ceil 10Mbit burst 1600b cburst
1600b
class htb 1:12 parent 1:1 leaf 20: prio 0 rate 100Mbit ceil 1Gbit burst 1600b cburst
1375b
root@server:/# /sbin/tc filter show dev enp2s0
filter parent 1: protocol ip pref 1 u32 chain 0
filter parent 1: protocol ip pref 1 u32 chain 0 fh 800: ht divisor 1
filter parent 1: protocol ip pref 1 u32 chain 0 fh 800::800 order 2048 key ht 800
bkt 0 flowid 1:11 not_in_hw
match c0a80004/ffffffff at 16
filter parent 1: protocol ip pref 1 u32 chain 0 fh 800::801 order 2049 key ht 800
bkt 0 flowid 1:12 not_in_hw
match c0a80700/ffffffff00 at 16
```

## Checking the backup channels operability

Of course, this check should be performed after configuring the firewall, because its rules may prevent the backup channel usage if the firewall is not fully configured. The check itself boils down to the possibility of accessing the Internet from a client computer both when a smartphone is connected to the server in usb modem mode, and when working on the main Internet channel.

## DNS/DHCP service

### Configuration

Install the dnsmasq package with all dependencies to implement this service:

```
root@server:/# apt-get update
root@server:/# apt-get install dnsmasq
```

Then you should bring the configuration file **/etc/dnsmasq.conf** to the form:

```
### General configuration ###

listen-address=127.0.0.1
listen-address=10.9.1.129
cache-size=500
domain-needed
bind-dynamic
filterwin2k
no-resolv
no-poll
no-hosts
bogus-priv

server=77.88.8.7@enp2s0
server=/school34/192.168.0.1@enp2s0
server=/0.168.192.in-addr.arpa/192.168.0.1@enp2s0
server=/service.school34/192.168.0.77@enp2s0
server=/7.168.192.in-addr.arpa/192.168.0.77@enp2s0

local=/account.school34/
domain=school34,192.168.0.0/24
domain=service.school34,192.168.7.0/24
domain=account.school34,10.9.1.128/26

mx-target=mail.service.school34
localmx

### DHCP configuration ###

no-dhcp-interface=lo
no-dhcp-interface=enp2s0
dhcp-range=10.9.1.128,static,infinite
dhcp-option=6,10.9.1.129
dhcp-option=42,10.9.1.129

#log-queries
#log-facility=/var/log/dnsmasq.log

dhcp-host=90:2b:34:a3:83:44,main,10.9.1.131
dhcp-host=50:e5:49:33:de:d9,accd1,10.9.1.132
dhcp-host=90:2b:34:a0:24:7b,accd2,10.9.1.133
dhcp-host=90:2b:34:96:16:53,accd3,10.9.1.135
dhcp-host=f8:0d:ac:78:8a:f1,mfp,10.9.1.134

### DNS configuration ###

address=/server.account.school34/10.9.1.129
ptr-record=129.1.9.10.in-addr.arpa,server.account.school34
mx-host=server.account.school34,mail.service.school34,50
txt-record=server.account.school34,"account department subnet server"
```

```
address=/samba.account.school34/10.9.1.129
ptr-record=129.1.9.10.in-addr.arpa,samba.account.school34
mx-host=samba.account.school34,mail.service.school34,50
txt-record=samba.account.school34,"account department file server"
```

```
address=/ntp.account.school34/10.9.1.129
ptr-record=129.1.9.10.in-addr.arpa,ntp.account.school34
mx-host=ntp.account.school34,mail.service.school34,50
txt-record=ntp.account.school34,"account department time server"
```

```
address=/main.account.school34/10.9.1.131
ptr-record=131.1.9.10.in-addr.arpa,main.account.school34
mx-host=main.account.school34,mail.service.school34,50
txt-record=main.account.school34,"senior accountant workstation"
```

```
address=/accd1.account.school34/10.9.1.132
ptr-record=132.1.9.10.in-addr.arpa,accd1.account.school34
mx-host=accd1.account.school34,mail.service.school34,50
txt-record=accd1.account.school34,"accountant workstation"
```

```
address=/accd2.account.school34/10.9.1.133
ptr-record=133.1.9.10.in-addr.arpa,accd2.account.school34
mx-host=accd2.account.school34,mail.service.school34,50
txt-record=accd2.account.school34,"accountant workstation"
```

```
address=/accd3.account.school34/10.9.1.135
ptr-record=135.1.9.10.in-addr.arpa,main.account.school34
mx-host=accd3.account.school34,mail.service.school34,50
txt-record=accd3.account.school34,"accountant workstation"
```

```
address=/mfp.account.school34/10.9.1.134
ptr-record=134.1.9.10.in-addr.arpa,mfp.account.school34
mx-host=mfp.account.school34,mail.service.school34,50
txt-record=mfp.account.school34,"account department network MFP"
```

Thus, the dnsmasq service uses the following order of operation:

The service accepts requests only on the local loop and internal client interfaces. The cache of DNS records has been increased from 150 (by default) to 500 records, since the amount of RAM in the system does not prevent this and such increase has a positive effect on the performance of the service. The amount of 500 records was chosen as the most optimal according to the experience of the operating of the whole network, in which the network served by the described system is a part.

The bind-dynamic parameter instructs the service to respond to changes in the state of network interfaces. Thus, it is possible to start the service when the network interface of the system is disabled with automatic "picking up" of this interface when it starts.

The service does not use the system configuration files of the built-in DNS client in its work. All the parameters necessary for its operation are specified in the main configuration file of the service given above.

The service knows three local domains (school34, service.school34 and account.school34). Each of them (except for the account.school34 domain served by the system itself) has its own DNS server, which is accessed through the external interface of the system. The resolution of DNS queries about other non-routable addresses on the Internet is blocked by the bogus-priv parameter. DNS queries about external hosts are resolved using YandexDNS.

The email service parameters are set so that the default email address for all subnet clients is the address of the organization's local mail server (mail.service.school34).

The DHCP protocol in the wireless subnet distributes addresses to clients in the range 10.9.1.130 - 10.9.1.190 (62 addresses, respectively, no more than 62 clients at a time). Requests for an address are obviously accepted only from the internal interface of the system. In addition to providing the address, the server provides the client with DNS services (dhcp-option 6) and network time service (dhcp-option 42). It should be noted that the address is issued only to clients registered in the configuration file, and the address is always the same to the client, i. e. the address setting is dynamic, and the address itself is static.

The service keeps logs in the file **/var/log/daemon.log**, which briefly reflects information about the start of the service, its settings, etc. Detailed logging of the service is disabled in this configuration, however, by commenting out the corresponding two lines in the above file, you can enable logging. At the same time, it should be remembered that logging, even taking into account the rotation of log files (see "About additional services"), can serve as a direction for attacking the server: sending a huge number of meaningless requests can overflow the server's disk space. In the system under consideration, there is no need for detailed logging of what is happening, which is why it is disabled.

Upon completion of editing the configuration file to run the service it remains only to restart it with the command

```
root@server:/# systemctl restart dnsmasq
```

You can verify the success of the restart by the output of the command

```
root@server:/# systemctl status dnsmasq
```

```
• dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
  Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
enabled)
  Active: active (running) since Tue 2021-05-11 08:00:27 MSK; 1h 35min ago
  Process: 370 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
  Process: 381 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited,
status=0/SUCCESS)
  Process: 389 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf
(code=exited, status=0/SUCCESS)
  Main PID: 388 (dnsmasq)
    Tasks: 1 (limit: 2315)
    Memory: 3.0M
    CGroup: /system.slice/dnsmasq.service
            └─388 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7
/etcdnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-new --local-service --trust-
anchor=.,20326,8,2,e06d44b80b8f1d39a95c0b0d7c65d08458e8
```

```
мая 11 08:00:26 server systemd[1]: Starting dnsmasq - A lightweight DHCP and caching
DNS server...
мая 11 08:00:27 server dnsmasq[370]: dnsmasq: syntax check OK.
мая 11 08:00:27 server systemd[1]: Started dnsmasq - A lightweight DHCP and caching
DNS server.
```

The service runs on behalf of the dnsmasq system user. The process ID (PID) is stored by default in the **/run/dnsmasq/dnsmasq.pid** file. During operation the dnsmasq process reacts to the SIGUSR1 signal by sending messages about current clients, statistics of their requests, etc. to the system log. The following command allows you to send this signal to the desired process and open a sample from the system log containing all mentions of dnsmasq for the last hour:



```
root@server:/# kill -s SIGUSR1 $(cat /run/dnsmasq/dnsmasq.pid) && journalctl -u dnsmasq --since -1h
```

This completes the configuration of the main mode of DNS/DHCP service, but you should consider changing the configuration of the service when switching to a backup Internet connection via a USB modem (smartphone). In this mode, the accounting subnet and its server are disconnected from the organization's local network, which means that the resources of this local network are no longer available. Therefore, dnsmasq in this mode of operation should not know anything about the local network. The easiest way to implement this behavior of the service is to restart it when changing the external interface using an alternative configuration file **/etc/dnsmasq.usb.conf**:

```
### General configuration ###

listen-address=127.0.0.1
listen-address=10.9.1.129
cache-size=500
domain-needed
bind-dynamic
filterwin2k
no-resolv
no-poll
no-hosts
bogus-priv

server=77.88.8.7@ethusb

local=/school34/
local=/account.school34/
domain=account.school34,10.9.1.128/26
selfmx

### DHCP configuration ###

no-dhcp-interface=lo
no-dhcp-interface=ethusb
dhcp-range=10.9.1.128,static,infinite
dhcp-option=6,10.9.1.129
dhcp-option=42,10.9.1.129

#log-queries
#log-facility=/var/log/dnsmasq.log

dhcp-host=90:2b:34:a3:83:44,main,10.9.1.131
dhcp-host=50:e5:49:33:de:d9,accd1,10.9.1.132
dhcp-host=90:2b:34:a0:24:7b,accd2,10.9.1.133
dhcp-host=90:2b:34:96:16:53,accd3,10.9.1.135
dhcp-host=f8:0d:ac:78:8a:f1,mfp,10.9.1.134

### DNS configuration ###

address=/server.account.school34/10.9.1.129
ptr-record=129.1.9.10.in-addr.arpa,server.account.school34
txt-record=server.account.school34,"account department subnet server"

address=/samba.account.school34/10.9.1.129
ptr-record=129.1.9.10.in-addr.arpa,samba.account.school34
mx-host=samba.account.school34,mail.service.school34,50
txt-record=samba.account.school34,"account department file server"
```

```
address=/ntp.account.school34/10.9.1.129
ptr-record=129.1.9.10.in-addr.arpa,ntp.account.school34
mx-host=ntp.account.school34,mail.service.school34,50
txt-record=ntp.account.school34,"account department time server"
```

```
address=/main.account.school34/10.9.1.131
ptr-record=131.1.9.10.in-addr.arpa,main.account.school34
txt-record=main.account.school34,"senior accountant workstation"
```

```
address=/accd1.account.school34/10.9.1.132
ptr-record=132.1.9.10.in-addr.arpa,accd1.account.school34
txt-record=accd1.account.school34,"accountant workstation"
```

```
address=/accd2.account.school34/10.9.1.133
ptr-record=133.1.9.10.in-addr.arpa,accd2.account.school34
txt-record=accd2.account.school34,"accountant workstation"
```

```
address=/accd3.account.school34/10.9.1.135
ptr-record=135.1.9.10.in-addr.arpa,main.account.school34
txt-record=accd3.account.school34,"accountant workstation"
```

```
address=/mfp.account.school34/10.9.1.134
ptr-record=134.1.9.10.in-addr.arpa,mfp.account.school34
txt-record=mfp.account.school34,"account department network MFP"
```

Difference between main and alternative configurations should be noted:

Firstly, the name of the external interface has been changed, and the DHCP protocol is also disabled on it. Secondly, only one external name server (Yandex DNS) remained available. Finally, the logic of working with mail (MX) records has been changed: in the absence of a single mail server, the final destination for each client of the network is himself.

Restart of the service is on duty the files **/etc/network/interfaces** and **/etc/default/dnsmasq**. The last file contains the default dnsmasq settings file name and allows you to easily change the path to the main configuration file. In the network interface configuration file, when starting a backup connection, perform such a substitution, and when it stops, return to its original state. Such substitution is performed using the **/etc/network/dns** script:

```
#!/bin/sh

if [ $1 = "usb-pre-up" ]
then
    systemctl stop dnsmasq
elif [ $1 = "usb-post-up" ]
then
    sed s/"#DNSMASQ_OPTS="/"/DNSMASQ_OPTS="/ -i /etc/default/dnsmasq
    echo "nameserver 127.0.0.1" > /etc/resolv.conf
    systemctl start dnsmasq
elif [ $1 = "usb-pre-down" ]
then
    systemctl stop dnsmasq
    sed s/"DNSMASQ_OPTS="/"/#DNSMASQ_OPTS="/ -i /etc/default/dnsmasq
elif [ $1 = "usb-post-down" ]
then
    echo "nameserver 127.0.0.1" > /etc/resolv.conf
    systemctl start dnsmasq
fi
exit 0
```

So before starting the backup interface, you should stop the dnsmasq service, and immediately after you need to replace the configuration file of this service, specify in the **/etc/resolv.conf** file that the server itself should continue to use its own service, and not access the dns server provided by the mobile provider, and restart the dnsmasq service. Before disabling the backup interface, you should stop the domain name service and return to the main configuration file. Finally, after disconnecting the backup connection, it remains to restart the service.

*Note 1:* In the **/etc/network/interfaces** file, the order of commands invoked when starting or stopping the backup interface is significant. Both the main and backup dnsmasq configurations use the bind-dynamic parameter, which allows the service to start working even if the network interfaces it needs are not running yet. In this situation, dnsmasq automatically "picks up" newly launched network interfaces as they are ready. However, this behavior does not prevent the possibility of capturing a backup connection from the main configuration and vice versa. Observing the order of commands in the **/etc/network/interfaces** file allows you to avoid this.

*Note 2:* Debian provides subdirectories **/etc/network/\*.d** for hosting such scripts. However, solely for stylistic reasons, the approach proposed by Debian with grouping actions by the events that cause them was not applied on the system in question, but its own approach with grouping actions by network interfaces was used.

At the same time, a small correction should be made to the **/etc/default/dnsmasq** file: since the alternative dnsmasq configuration is described in the file, **/etc/dnsmasq.usb.conf**, then the line

```
#DNSMASQ_OPTS="--conf-file=/etc/dnsmasq.alt"
```

in the file **/etc/default/dnsmasq** should change to:

```
#DNSMASQ_OPTS="--conf-file=/etc/dnsmasq.usb.conf"
```

After completing the service configuration, you should tell the system to use its own service for its own needs. To do this, bring the configuration file **/etc/resolv.conf** to the following content:

```
nameserver 127.0.0.1
```

The you should restart the network subsystem via command:

```
root@server:/# systemctl restart networking
```

This is required to send DNS queries generated by the system itself about local servers directly to the virtualization server, and not to redirect them there by the main gateway.

## Fail protection (auto-resume)

To organize the automatic recovery of the service after a failure, you can use the means of the systemd initialization system.

First of all, you should check the current parameters of the service with the command:

```
root@server:/# systemctl show dnsmasq
Type=forking
Restart=no
PIDFile=/run/dnsmasq/dnsmasq.pid
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
```

As can be seen from the response of the command (it is given here in a strongly abbreviated form, containing only important lines in the current context), the restart of the service is disabled.

Then you should edit the service using the command:

```
root@server:/# systemctl edit dnsmasq
```

A text editor would be opened in which certain edits need to be made. If the changes are confirmed to be at text editor exit, a file `/etc/systemd/system/dnsmasq.service.d/override.conf` is created in the system, these corrections are stored there. The next time the service starts, they would override the original parameter values. To solve the problem of restarting the service in case of failures, the following contents of this file are sufficient:

```
[Service]
Restart=on-failure
```

This changes in the service are saved even when installing system updates.

After making the changes, restart the service and make sure that the parameter changes are taken into account:

```
root@server:/# systemctl restart dnsmasq
root@server:/# systemctl show dnsmasq
Type=forking
Restart=on-failure
PIDFile=/run/dnsmasq/dnsmasq.pid
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
```

## Verification of performance and stability

Checking of the service operability is trivial: the client is connected (of course, temporarily included in the configuration file) via the DHCP protocol and a number of DNS queries are performed on it. The following fragment of a terminal session performed on a client device running the Xubuntu 20.04 live disk demonstrates the fact of connecting to the network, obtaining an ip address, network mask, network routes and gateway addresses. Testing is performed using a small auxiliary script located in the `/tmp` directory:

```
xubuntu@xubuntu:/# cat /tmp/dnstest.sh
#!/bin/bash -x
ip address show
ip route show
host samba.account.school34
host avers.school34
host mail.service.school34
host yandex.ru
host mcst.ru
```

The `-x` key of the interpreter allows to display the executed commands during the script operation, in the following fragment of the terminal session these commands are highlighted in bold. Of course, the script is made executable.

```
xubuntu@xubuntu:/tmp$ ./dnstest.sh
+ ip address show
```

```

...
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 60:eb:69:b0:45:e2 brd ff:ff:ff:ff:ff:ff
    inet 10.9.1.136/26 brd 10.9.1.191 scope global noprefixroute enp2s0
        valid_lft forever preferred_lft forever
    inet6 fe80::ce14:beee:a6db:a8bc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
group default qlen 1000
...
+ ip route show
default via 10.9.1.129 dev enp2s0 proto dhcp metric 100
10.9.1.128/26 dev enp2s0 proto kernel scope link src 10.9.1.136 metric 100
169.254.0.0/16 dev enp2s0 scope link metric 1000
+ host samba.account.school34
samba.account.school34 has address 10.9.1.129
samba.account.school34 mail is handled by 50 mail.service.school34.
+ host avers.school34
avers.school34 has address 192.168.5.51
+ host mail.service.school34
mail.service.school34 has address 192.168.7.3
mail.service.school34 mail is handled by 50 mail.service.school34.
+ host yandex.ru
yandex.ru has address 213.180.193.56
yandex.ru has IPv6 address 2a02:6b8:a::a
yandex.ru mail is handled by 10 mx.yandex.ru.
+ host mcst.ru
mcst.ru has address 84.201.189.147
mcst.ru mail is handled by 50 tretyak2.mcst.ru.
xubuntu@xubuntu:/tmp$ ./dnstest.sh
+ ip address show
...
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 60:eb:69:b0:45:e2 brd ff:ff:ff:ff:ff:ff
    inet 10.9.1.136/26 brd 10.9.1.191 scope global noprefixroute enp2s0
        valid_lft forever preferred_lft forever
    inet6 fe80::ce14:beee:a6db:a8bc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
group default qlen 1000
...
+ ip route show
default via 10.9.1.129 dev enp2s0 proto dhcp metric 100
10.9.1.128/26 dev enp2s0 proto kernel scope link src 10.9.1.136 metric 100
169.254.0.0/16 dev enp2s0 scope link metric 1000
+ host samba.account.school34
samba.account.school34 has address 10.9.1.129
samba.account.school34 mail is handled by 50 mail.service.school34.
+ host avers.school34
Host avers.school34 not found: 3(NXDOMAIN)
+ host mail.service.school34
Host mail.service.school34 not found: 3(NXDOMAIN)
+ host yandex.ru
yandex.ru has address 213.180.193.56
yandex.ru has IPv6 address 2a02:6b8:a::a
yandex.ru mail is handled by 10 mx.yandex.ru.
+ host mcst.ru
mcst.ru has address 84.201.189.147
mcst.ru mail is handled by 50 tretyak2.mcst.ru.
xubuntu@xubuntu:/tmp$

```

Some of the command responses are reduced in the uninformative part in the presented fragment. At the same time, from the server side, the following lines can be seen in the log file `/var/log/dnsmasq.log` (also with abbreviations):

```
Jun 21 14:41:49 dnsmasq[1273]: query[A] samba.account.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: config samba.account.school34 is 10.9.1.129
Jun 21 14:41:49 dnsmasq[1273]: query[AAAA] samba.account.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: config samba.account.school34 is NODATA-IPv6
Jun 21 14:41:49 dnsmasq[1273]: query[MX] samba.account.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: config samba.account.school34 is <MX>
Jun 21 14:41:49 dnsmasq[1273]: query[A] avers.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded avers.school34 to 192.168.0.1
Jun 21 14:41:49 dnsmasq[1273]: reply avers.school34 is 192.168.5.51
Jun 21 14:41:49 dnsmasq[1273]: query[AAAA] avers.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded avers.school34 to 192.168.0.1
Jun 21 14:41:49 dnsmasq[1273]: query[MX] avers.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded avers.school34 to 192.168.0.1
Jun 21 14:41:49 dnsmasq[1273]: query[A] mail.service.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded mail.service.school34 to 192.168.0.77
Jun 21 14:41:49 dnsmasq[1273]: reply mail.service.school34 is 192.168.7.3
Jun 21 14:41:49 dnsmasq[1273]: query[AAAA] mail.service.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded mail.service.school34 to 192.168.0.77
Jun 21 14:41:49 dnsmasq[1273]: query[MX] mail.service.school34 from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded mail.service.school34 to 192.168.0.77
Jun 21 14:41:49 dnsmasq[1273]: query[A] yandex.ru from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded yandex.ru to 77.88.8.7
Jun 21 14:41:49 dnsmasq[1273]: reply yandex.ru is 213.180.193.56
Jun 21 14:41:49 dnsmasq[1273]: query[AAAA] yandex.ru from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded yandex.ru to 77.88.8.7
Jun 21 14:41:49 dnsmasq[1273]: reply yandex.ru is 2a02:6b8:a::a
Jun 21 14:41:49 dnsmasq[1273]: query[MX] yandex.ru from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded yandex.ru to 77.88.8.7
Jun 21 14:41:49 dnsmasq[1273]: query[A] mcst.ru from 10.9.1.136
Jun 21 14:41:49 dnsmasq[1273]: forwarded mcst.ru to 77.88.8.7
Jun 21 14:41:54 dnsmasq[1273]: reply mcst.ru is 84.201.189.147
Jun 21 14:41:54 dnsmasq[1273]: query[AAAA] mcst.ru from 10.9.1.136
Jun 21 14:41:54 dnsmasq[1273]: forwarded mcst.ru to 77.88.8.7
Jun 21 14:41:54 dnsmasq[1273]: reply mcst.ru is NODATA-IPv6
Jun 21 14:41:54 dnsmasq[1273]: query[MX] mcst.ru from 10.9.1.136
Jun 21 14:41:54 dnsmasq[1273]: forwarded mcst.ru to 77.88.8.7
...
Jun 21 14:46:30 dnsmasq[1598]: started, version 2.80 cachesize 500
Jun 21 14:46:30 dnsmasq[1598]: compile time options: IPv6 GNU-getopt DBus i18n IDN
DHCP DHCPv6 no-Lua TFTP conntrack ipset auth DNSSEC loop-detect inotify dumpfile
Jun 21 14:46:30 dnsmasq-dhcp[1598]: DHCP, static leases only on 10.9.1.128, lease
time infinite
Jun 21 14:46:30 dnsmasq[1598]: using local addresses only for domain
account.school34
Jun 21 14:46:30 dnsmasq[1598]: using local addresses only for domain school34
Jun 21 14:46:30 dnsmasq[1598]: using nameserver 77.88.8.7#53(via ethusb)
Jun 21 14:46:30 dnsmasq[1598]: cleared cache
Jun 21 14:46:33 dnsmasq[1598]: query[A] samba.account.school34 from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: config samba.account.school34 is 10.9.1.129
Jun 21 14:46:33 dnsmasq[1598]: query[AAAA] samba.account.school34 from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: config samba.account.school34 is NODATA-IPv6
Jun 21 14:46:33 dnsmasq[1598]: query[MX] samba.account.school34 from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: config samba.account.school34 is <MX>
Jun 21 14:46:33 dnsmasq[1598]: query[A] avers.school34 from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: config avers.school34 is NXDOMAIN
Jun 21 14:46:33 dnsmasq[1598]: query[A] avers.school34 from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: config avers.school34 is NXDOMAIN
Jun 21 14:46:33 dnsmasq[1598]: query[A] mail.service.school34 from 10.9.1.136
```

```

Jun 21 14:46:33 dnsmasq[1598]: config mail.service.school34 is NXDOMAIN
Jun 21 14:46:33 dnsmasq[1598]: query[A] mail.service.school34 from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: config mail.service.school34 is NXDOMAIN
Jun 21 14:46:33 dnsmasq[1598]: query[MX] yandex.ru from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: forwarded yandex.ru to 77.88.8.7
Jun 21 14:46:33 dnsmasq[1598]: query[AAAA] mcst.ru from 10.9.1.136
Jun 21 14:46:33 dnsmasq[1598]: forwarded mcst.ru to 77.88.8.7
Jun 21 14:46:36 dnsmasq[1598]: reply mcst.ru is NODATA-IPv6
Jun 21 14:46:40 dnsmasq[1598]: exiting on receipt of SIGTERM

```

From the above fragment it can be seen that at 14:41 the test script was executed when connected to the Internet via the main connection, and at 14:46 the daemon was restarted due to the transition to a mobile Internet connection, followed by a re-execution of the script. In the first case, positive responses were received to all requests except for IPv6 requests, keeping in mind the fact that the rejection of IPv6 in the network was agreed earlier. In the second case, requests for other subnets of the organization other than the one under consideration were not only rejected, but there were not even attempts to redirect these requests to higher DNS servers. The results obtained fully correspond to the required operating mode of the service.

Checking the system's resilience to failures is demonstrated in the following fragment of the terminal session (the responses of some commands are reduced in an uninformative to the context part):

```

root@server:/# ps -ef | grep dnsmasq
dnsmasq  1100      1  0 13:34 ?          00:00:00 /usr/sbin/dnsmasq -x ...
root      1111    652  0 13:37 pts/0      00:00:00 grep dnsmasq
root@server:/# kill -9 1100
root@server:/# ps -ef | grep dnsmasq
dnsmasq  1121      1  0 13:37 ?          00:00:00 /usr/sbin/dnsmasq -x ...
root      1130    652  0 13:37 pts/0      00:00:00 grep dnsmasq
root@server:/# systemctl status dnsmasq
• dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
  Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/dnsmasq.service.d
           └─override.conf
  Active: active (running) since Tue 2021-05-11 13:37:41 MSK; 18s ago
  Process: 1113 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
  Process: 1114 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited,
status=0/SUCCESS)
  Process: 1122 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf
(code=exited, status=0/SUCCESS)
  Main PID: 1121 (dnsmasq)
    Tasks: 1 (limit: 2315)
   Memory: 1.3M
    CGroup: /system.slice/dnsmasq.service
            └─1121 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 ...

```

In this fragment, the PID of the current dnsmasq process (1100) is found out, then this process is intentionally terminated, after which the PID of the dnsmasq process is found out again. From the fact that PID = 1121 exists and differs from the previous one, it follows an unambiguous conclusion that the service was restarted after the crash. This is also confirmed by the output of the last command.

# Network time service (ntp)

## Configuration

First of all, you need to install the ntp and ntpdate packages with all the dependencies:

```
root@server:/# apt-get update
root@server:/# apt-get install ntp ntpdate
```

These packages include a number of programs:

- ntpd daemon,
- ntpq – standard program for queries,
- ntpdc – advanced query program,
- ntpdate – client program for setting the time in the system via ntp,
- sntp – a simple network client
- and others (key generators, utility, debugging, simulators, etc.)

Before configuring ntp.conf, you should first configure the time zone (file **/etc/localtime**), which is the most correct way is to use the command:

```
root@server:/# dpkg-reconfigure tzdata
```

To configure the exact time service you should bring the configuration file **/etc/ntp.conf** to the form:

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

server ntp.school34      iburst prefer
server ntp1.vniiftri.ru iburst
server ntp2.vniiftri.ru iburst
server ntp3.vniiftri.ru iburst

interface ignore all
interface listen 192.168.0.88
interface listen 10.9.1.128/26

restrict default ignore
restrict ntp.school34      noquery notrap
restrict ntp1.vniiftri.ru noquery notrap
restrict ntp2.vniiftri.ru noquery notrap
restrict ntp3.vniiftri.ru noquery notrap
restrict 127.0.0.1
restrict 10.9.1.128 mask 255.255.255.192 kod notrap nomodify nopeer noquery limited
```

Explanation of the configuration file options:

driftfile is a file for recording the frequency correction of the hardware clock, updated by the daemon once an hour;

interface is a parameter describing interaction with the network interfaces of the system; of several such directives, the last suitable one is applied to the network packet.

server – an external server used for synchronization, several such servers can be set in case one or more of them are unavailable; the prefer additional parameter means that in case of availability the priority of use should be given to this server.



iburst – send 8 packets in 2 seconds instead of one, this allows you to synchronize faster (in a few seconds instead of a few minutes), but is not suitable for networks with low bandwidth;

kod - kiss of death - send a response packet with a notification in response to a packet that violates the limits on the load on the service;

notrap – not to implement the functionality of determining the position of the host over IPv6;

nomodify - reject requests that try to change the state of the server, only requests that only receive a response are allowed;

nopeer - reject unauthorized requests to establish communication, does not apply to packages that do not establish communication, i. e. to serve clients, but not synchronize with them;

restrict default - this line sets default restrictions, here by default all packages except those explicitly indicated below are ignored;

restrict - entering a restriction on a host or network (a host can be specified by both a name and an address, it makes no sense to specify a host with multiple ip addresses, such as, for example, `debian.pool.ntp.org`, because in this case, the default rule will be triggered), the network is set by its address and mask, and then the keys follow, limiting this network or host;

limited - reject synchronization requests if the traffic limits set by the `discard` command are exceeded (by default, the minimum interval between packets is 1 second, and the average is 3 seconds), if the `kod` flag is also set, then a response packet is sent;

noquery - reject requests from `ntpq` and `ntpd`, the exact time service is not affected;

Thus, the configuration file forces `ntpd` to work according to the following rules:

- 1. synchronization is allowed only with predefined ntp servers
- 2. by default, all packets sent to the server are ignored, except explicitly allowed ones
- 3. External ntp servers are not allowed to access the local one with service requests or use IPv6
- 4. connections from the server itself are allowed to do everything, i. e. locally you can make any requests to the ntp server, manage it, monitor its status
- 5. local clients are allowed only from the specified subnet, they are limited in bandwidth, cannot influence the time server and send service requests to it, they are only allowed to receive the exact time from the server

It remains only to restart the exact time subsystem with the command

```
root@server:/# systemctl restart ntp
```

and make sure that the service is in a working state, using the command

```
root@server:/# systemctl status ntp
```

If there are no errors, the output of this command will be something like:

```
• ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2021-05-12 11:54:46 +03; 6s ago
    Docs: man:ntpd(8)
  Process: 2487 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited,
status=0/SUCCESS)
 Main PID: 2493 (ntpd)
   Tasks: 2 (limit: 2315)
```

```
Memory: 1.3M
CGroup: /system.slice/ntp.service
└─2493 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 108:113
```

```
мая 12 11:54:46 server systemd[1]: Started Network Time Service.
мая 12 11:54:46 server ntpd[2493]: proto: precision = 0.070 usec (-24)
мая 12 11:54:46 server ntpd[2493]: Listen and drop on 0 v6wildcard [::]:123
мая 12 11:54:46 server ntpd[2493]: Listen and drop on 1 v4wildcard 0.0.0.0:123
мая 12 11:54:46 server ntpd[2493]: Listen normally on 2 lo 127.0.0.1:123
мая 12 11:54:46 server ntpd[2493]: Listen normally on 3 enp3s0 10.9.1.129:123
мая 12 11:54:46 server ntpd[2493]: Listen normally on 4 enp2s0 192.168.0.88:123
мая 12 11:54:46 server ntpd[2493]: Listening on routing socket on fd #21 for
interface updates
мая 12 11:54:46 server ntpd[2493]: kernel reports TIME_ERROR: 0x2041: Clock
Unsynchronized
мая 12 11:54:46 server ntpd[2493]: kernel reports TIME_ERROR: 0x2041: Clock
Unsynchronized
```

It should be noted that the clock is not synchronized (Clock Unsynchronized), which is expected, because the service has just started, and synchronization takes some time. The following commands executed two minutes after restarting the service demonstrate synchronization success:

```
root@server:/# /sbin/ntpdate -q localhost
server 127.0.0.1, stratum 2, offset 0.000005, delay 0.02567
12 May 11:58:07 ntpdate[2518]: adjust time server 127.0.0.1 offset 0.000005 sec
root@server:/# ntpq -p
      remote           refid      st t when poll reach   delay   offset  jitter
=====
*ntp3.vniiftri.r .MRS.          1 u   56  256  377   30.848  -0.296   2.041
+ntp1.vniiftri.r .MRS.          1 u  190  256  375   30.600  -0.207   0.298
+ntp2.vniiftri.r .MRS.          1 u  256  256  377   30.672  -0.338   3.232
```

Particularly, it is indicated here that the system clock is synchronized with the local server with an accuracy of 0.000005 seconds, while the local server has a stratum level of 2, the deviation of its clock from the clock of the higher-level ntp.school34 server is 0.296 milliseconds, and the variance of deviations according to the results of several recent requests was 2.041 milliseconds.

With the settings set, the ntpd daemon nevertheless listens to all system interfaces, but ignores packets according to its configuration file. The daemon also listens to IPv6 addresses. This can be seen in the output of the command:

```
root@server:/# ss -ulp | grep ntp
UNCONN    0      0      192.168.0.88:ntp    0.0.0.0:*    users:(("ntpd",pid=385,fd=23))
UNCONN    0      0      10.9.1.129:ntp     0.0.0.0:*    users:(("ntpd",pid=385,fd=19))
UNCONN    0      0      127.0.0.1:ntp      0.0.0.0:*    users:(("ntpd",pid=385,fd=18))
UNCONN    0      0      0.0.0.0:ntp        0.0.0.0:*    users:(("ntpd",pid=385,fd=17))
UNCONN    0      0      [::]:ntp           [::]:*       users:(("ntpd",pid=385,fd=16))
```

To forcibly disable IPv6 protocol support by the daemon, you should bring the configuration file **/etc/default/ntp** to the form:

```
NTPD_OPTS='-4 -g'
```

After restarting the service, you can make sure that IPv6 is no longer in use:

```
root@server:/# systemctl restart ntp
root@server:/# ss -ul | grep ntp
UNCONN    0      0      192.168.0.88:ntp    0.0.0.0:*    users:(("ntpd",pid=815,fd=19))
```

```
UNCONN    0    0    10.9.1.129:ntp    0.0.0.0:*    users:(("ntpd",pid=815,fd=18))
UNCONN    0    0    127.0.0.1:ntp    0.0.0.0:*    users:(("ntpd",pid=815,fd=17))
UNCONN    0    0    0.0.0.0:ntp    0.0.0.0:*    users:(("ntpd",pid=815,fd=16))
```

The service keeps its system log in the file **/var/log/daemon.log**, in which other services write their messages too. To get a subset of messages related to the ntp service for a certain time interval (for example, 08:00-10:30 of the current day), use the command

```
root@server:/#journalctl -u ntp --since 08:00 --until 10:30
```

## Fail protection (auto-resume)

To organize the automatic recovery of the service after a failure, you can use the means of the systemd initialization system.

First of all, you should check the current parameters of the service with the command:

```
root@server:/# systemctl show ntp
Type=forking
Restart=no
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
```

As can be seen from the response of the command (it is given here in a strongly abbreviated form, containing important lines in the current context), restarting the service is disabled.

Then you should edit the service using the command:

```
root@server:/# systemctl edit ntp
```

A text editor will be opened in which certain edits need to be made. If the changes are confirmed to be saved when exiting the text editor, a file **/etc/systemd/system/ntp.service.d/override.conf** would be created in the system to store these corrections. The next time the service is started, they override the original parameter values. To solve the problem of restarting the service in case of failures, the following contents of this file are sufficient:

```
[Service]
Restart=on-failure
```

These changes in the service are saved even when installing system updates.

After making the changes, restart the service and make sure that the parameter changes are taken into account:

```
root@server:/# systemctl restart ntp
root@server:/# systemctl show ntp
Type=forking
Restart=on-failure
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
```

## Verification of performance and stability

Checking the service operability is trivial: a request for time adjustment is made from a Windows client connected to the internal network.

Checking the system's resilience to failures is demonstrated in the following fragment of the terminal session (responses of some commands are reduced in an uninformative part applied to the context):

```
root@server:/# systemctl status ntp
• ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/ntp.service.d
           └─override.conf
  Active: active (running) since Thu 2021-05-13 09:56:43 +03; 5min ago
  Docs: man:ntpd(8)
  Process: 861 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited,
status=0/SUCCESS)
  Main PID: 867 (ntpd)
    Tasks: 2 (limit: 2315)
  Memory: 1.3M
  CGroup: /system.slice/ntp.service
          └─867 /usr/sbin/ntpd -p /var/run/ntpd.pid -4 -g -u 108:113
root@server:/# kill -9 867
root@server:/# systemctl status ntp
• ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/ntp.service.d
           └─override.conf
  Active: active (running) since Thu 2021-05-13 10:02:13 +03; 5s ago
  Docs: man:ntpd(8)
  Process: 878 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited,
status=0/SUCCESS)
  Main PID: 884 (ntpd)
    Tasks: 2 (limit: 2315)
  Memory: 1.2M
  CGroup: /system.slice/ntp.service
          └─884 /usr/sbin/ntpd -p /var/run/ntpd.pid -4 -g -u 108:113
```

In this fragment, the status of the network time service is checked, the PID of the current ntpd process (867) is found out from the response message, then this process is intentionally terminated, after which the status of the service is checked again and the PID of the newly launched ntpd process (884) is found out. The operability of the service is also confirmed by repeated successful synchronization with the server of the same client.

## SSH Remote Management Service

Ssh installation can be performed both at the system installation stage and after it using the commands

```
root@server:/# apt-get update
root@server:/# apt-get install ssh
```

Ssh is used on this server as the only tool for remote login and administration. Given the position of the server in the network, ssh accepts connections only on the external (inaccessible to clients) interface. Nevertheless, significant measures should be taken to protect the remote

administration system. These measures are divided into three echelons: a network filter, the basic configuration of the server and, in fact, the configuration of the ssh service itself.

The configuration of network filter is described below, and the basic configuration of the server is described above when describing access lists and static arp records.

Regarding the configuration of the service itself, two directions should be distinguished: protection against unauthorized access and ensuring trouble-free operation.

## Configuration

The main configuration file `/etc/ssh/sshd_config` of the service should look like:

```
Port 22
ListenAddress 192.168.0.88
AddressFamily inet
Protocol 2
PermitRootLogin no
AllowUsers administrator
PasswordAuthentication yes
PubkeyAuthentication no
KerberosAuthentication no
HostbasedAuthentication no
IgnoreRhosts yes
PermitEmptyPasswords no
X11Forwarding no
```

This ensures the following order of operation of the ssh server:

Port 22 – the server accepts messages on port 22 over tcp.

Protocol 2 – ssh version 2 protocol is used.

AddressFamily inet – IPv4 protocol is allowed, IPv6 usage is disabled.

PermitRootLogin no – remote login on behalf of the superuser is prohibited.

AllowUsers administrator – Remote login is allowed only for the administrator user.

PasswordAuthentication yes – password authentication is allowed only, other methods are prohibited because not used.

PermitEmptyPasswords no – the use of empty passwords is prohibited.

X11 Forwarding no – traffic transmission over the X11 protocol is not allowed because the protocol is not used by the server.

After bringing this file to the specified form, restart the ssh service, then make sure that it is successfully started and accepts connections only over IPv4. This can be done with the following commands:

```
root@debian:/# systemctl restart sshd
root@debian:/# systemctl status sshd
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2021-05-05 11:58:06 MSK; 10s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 3838 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 3839 (sshd)
```

```
Tasks: 1 (limit: 2315)
Memory: 1.1M
CGroup: /system.slice/ssh.service
└─3839 /usr/sbin/sshd -D
```

```
мая 05 11:58:06 server.account.school34 systemd[1]: Starting OpenBSD Secure Shell
server...
мая 05 11:58:06 server.account.school34 sshd[3839]: Server listening on 192.168.0.88
port 22.
мая 05 11:58:06 server.account.school34 systemd[1]: Started OpenBSD Secure Shell
server.
root@debian:/# ss -Htulp | grep ssh
tcp LISTEN 0 128 192.168.0.88:ssh 0.0.0.0:* users:(("sshd",pid=3839,fd=3))
```

The service keeps its system log in the file **/var/log/daemon.log**, in which other services write their messages too. To get a sample of ssh-related messages for a certain time interval (for example, for the current day), use the command

```
root@server:/#journalctl -u ssh --since today
```

## Fail protection (auto-resume)

The management of services in the system is handled by the boot manager and systemd services, which ensures (with proper configuration) the automatic resumption of services in case of a crash. In the system under consideration such setup for the ssh service was performed initially by the maintainers of the operating system distribution.

## Verification of performance and stability

To check the status of the service, you can use the command

```
root@debian:/# systemctl status ssh
```

To check which ports, addresses and interfaces ssh listens to, you can use the command

```
root@debian:/# ss -Htulp | grep ssh
```

The following fragment of the terminal session demonstrates attempts to log in remotely from an authorized host under different accounts:

```
administrator@admin:~$ ssh administrator@server.account.school34
administrator@server.account.school34's password:
Last login: Thu May 13 09:43:44 2021 from 192.168.0.2
```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in **/usr/share/doc/\*/copyright**.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

```
administrator@server:~$ logout
Connection to server.account.school34 closed.
administrator@admin:~$ ssh root@server.account.school34
root@server.account.school34's password:
Permission denied, please try again.
root@server.account.school34's password:
Permission denied, please try again.
root@server.account.school34's password:
```

```
root@server.account.school34: Permission denied (password,keyboard-interactive).
administrator@admin:~$
```

As you can see from this fragment, the login as administrator was successful, and the login as root was not executed even despite entering the correct password.

When trying to log in from a host other than the allowed one (both from the internal network and from the external one), the client receives a message (with the firewall of the server in question disabled):

```
xubuntu@xubuntu:~$ ssh administrator@server.account.school34
ssh_exchange_identification: read: Connection reset by peer
```

This is the result of access control lists work. When the firewall is enabled, the client receives a different message:

```
xubuntu@xubuntu:~$ ssh administrator@server.account.school34
ssh: connect to host administrator@server.account.school34 port 22: Connection timed out
```

However, in both cases, the connection to the server is denied.

One of the ways to check for automatic resumption is given in the following terminal session:

```
root@server:/# date
Чт мая 13 11:21:07 +03 2021
root@server:/# ps -ef | grep ssh
root      420      1  0 08:00 ?        00:00:00 /usr/sbin/sshd -D
root      774      420  0 09:43 ?        00:00:00 sshd: administrator [priv]
adminis+  776      774  0 09:43 ?        00:00:00 sshd: administrator@pts/0
root      970      794  0 11:21 pts/0    00:00:00 grep ssh
root@server:/# kill -9 420
root@server:/# ps -ef | grep ssh
root      774      1  0 09:43 ?        00:00:00 sshd: administrator [priv]
adminis+  776      774  0 09:43 ?        00:00:00 sshd: administrator@pts/0
root      972      1  0 11:22 ?        00:00:00 /usr/sbin/sshd -D
root      975      794  0 11:22 pts/0    00:00:00 grep ssh
root@server:/# systemctl status sshd
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2021-05-13 11:22:08 +03; 14s ago
  Docs: man:sshd(8)
        man:sshd_config(5)
  Process: 971 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 972 (sshd)
  Tasks: 4 (limit: 2315)
  Memory: 10.1M
  CGroup: /system.slice/ssh.service
          └─774 sshd: administrator [priv]
            └─776 sshd: administrator@pts/0
              └─777 -bash
                └─972 /usr/sbin/sshd -D

мая 13 11:22:08 server systemd[1]: Starting OpenBSD Secure Shell server...
...
мая 13 11:22:08 server systemd[1]: Started OpenBSD Secure Shell server.
root@server:/#
```

Here, the system date and time are displayed first, then a list of running processes in the system is obtained, filtered by the mention of ssh. As you can see from this list, the sshd process with pid

420 is running in the system. The following command kills this process. Next, it is demonstrated that the process is indeed killed, but a new process with PID 972 is already running and the service has been successfully restarted. At the same time, it should be noted that the experiment was performed via ssh connection from the system administrator's computer, but the connection did not break.

## NUT configuration

The power supply of the server in question is carried out using a linear interactive uninterruptible power supply Cyber Power BR650ELCD-RU. To ensure proper handling of power outages, the NUT (network UPS tools) system should be installed and configured on the server.

### Configuration

You should start by installing NUT. The system consists of three components: the UPS driver, the daemon and the client. Command

```
root@server:/# apt-get install nut
```

installs all necessary packages. Upon completion of its execution, you should decide on the UPS driver. To do this, first of all, you need to find a reference file for uninterruptible power supply drivers as part of the nut-server package:

```
root@server:/# dpkg -L nut-server | grep driver
/lib/systemd/system/nut-driver.service
/usr/share/nut/driver.list
```

Next, you should collect information about the UPS used using commands (the output is partially reduced in the uninformative part):

```
root@server:/# lsusb
```

```
...
Bus 003 Device 007: ID 0764:0501 Cyber Power System, Inc. CP1500 AVR UPS
```

```
root@server:/# lsusb -d 0764:0501 -v
```

```
Bus 003 Device 079: ID 0764:0501 Cyber Power System, Inc. CP1500 AVR UPS
```

```
Device Descriptor:
```

```
  bLength                18
  bDescriptorType         1
  bcdUSB                  1.10
  bDeviceClass             0
  bDeviceSubClass          0
  bDeviceProtocol          0
  bMaxPacketSize0          8
  idVendor                0x0764 Cyber Power System, Inc.
  idProduct               0x0501 CP1500 AVR UPS
  bcdDevice               0.01
  iManufacturer           3 CPS
  iProduct                1 UPS VALUE
  iSerial                 0
  bNumConfigurations      1
```

```
Configuration Descriptor:
```

```
  bLength                9
  bDescriptorType         2
  wTotalLength            0x0022
  bNumInterfaces          1
```



```

bConfigurationValue      1
iConfiguration           0
bmAttributes              0xc0
    Self Powered
MaxPower                  50mA
Interface Descriptor:
    bLength               9
    bDescriptorType       4
    bInterfaceNumber      0
    bAlternateSetting     0
    bNumEndpoints         1
    bInterfaceClass       3 Human Interface Device
    bInterfaceSubClass    0
    bInterfaceProtocol    0
...

```

Already from this output, basing on the interface class, it can be assumed that the UPS is controlled by the `usbhid-ups` driver. Command output

```

root@server:/# cat /usr/share/nut/driver.list | grep CP1500
"Cyber Power Systems"  "ups"  "2"  "CP1500AVRLCD"  "USB"  "usbhid-ups"

```

confirms this assumption. It also states that the support level is 2, which means fragmented support for the open protocol.

Next, you should proceed with configuring the NUT components, taking into account the information provided in the driver's manual

```

root@server:/# man 8 usbhid-ups

```

First, you will need to ensure that the system responds correctly to UPS detection. Since this device belongs to the HID (Human Interface Device) class, when detected, it is captured by the corresponding driver, making it impossible to use the NUT driver. To avoid this, it is necessary to inform the `udev` subsystem that the HID driver should not be used. This is exactly what was done in the section "Configuring the UDEV subsystem".

The next step is to fill the configuration file `/etc/nut/ups.conf`. In this file, sections should be created with a description of each device connected to the system in question (the only UPS in this case). The driver and the parameters necessary for its operation must be specified inside this section. On this server, the file has the form:

```

maxretry=3

[BR650ELCD]
    driver=usbhid-ups
    port=auto
    vendorid=0764
    productid=0501
    default.battery.voltage.high=13.6
    default.battery.voltage.low=10.5
    desc="Cyber Power BR650ELCD-RU"

```

At the beginning of the file the `maxretry` parameter is placed. It indicates the number of attempts to start the driver. This option is useful for managing slow devices. It is followed by the section header containing the name of the UPS for NUT. The name can be chosen freely enough, but it should be the same for several configuration files. The driver, the UPS connection port,

manufacturer and device identifiers, a text description of the UPS, as well as the voltages generated by the UPS battery at high and low charge levels are indicated inside the section.

The port parameter is required, but when connected via the USB bus, its value may not be constant even if the UPS is always connected to the same connector. In this case, the value `auto` is specified. At the same time, the search for the corresponding device is performed by other criteria, for example, by the identifiers of the device model and manufacturer. When applied to an existing power source it should be noted that its serial number is not specified in the device's memory (the value 0 in the device descriptor above for the `iSerial` field). Therefore, in general, connecting two such UPS to one server (the "Big Box" scheme in NUT terminology, used, for example, if the server has two power supplies) can be very problematic. However, this problem does not exist in the system under consideration. Regarding the voltage values, they were found out in advance from the experience of previous UPS operation, and their receipt in the configured system will be demonstrated below. During the initial setup of a new UPS, this data can be "guessed" based on the internal design of the UPS, its documentation, driver prompts during the first starts and the characteristics of the batteries used. So the existing device contains one battery with a nominal voltage of 12 V, so the initial estimate of the parameter values can be taken to be 12 V and 10 V for high and low charge, respectively.

After configuring the driver, you should configure the `upsd` daemon. Its configuration file `/etc/nut/upsd.conf` according to the existing statement of the problem should contain only one line:

```
LISTEN 127.0.0.1 3493
```

So the daemon will listen to the standard NUT port only on the local loopback interface. If the system administrator needs to remotely manage the daemon, then the connection to the system should be performed via `ssh` protocol. Of course, this prevents the creation of a centralized UPS monitoring system at the organization level, but the task of creating such system is not being set at the moment, besides, the accounting server should be as autonomous and protected as possible.

In addition, the daemon must be provided with a list of allowed users, their passwords and permissions. Here in the file `/etc/nut/upsd.users` should be brought up to the comments to the form:

```
[upsmon]
    password = upspasswd
    upsmon master
```

This means that the daemon knows the `upsmon` user with the `upspasswd` password, who has the right not only to receive data from the UPS, but also to manage it.

Next, you need to configure the client program, with which the accesses to the daemon will be performed. The simplest such program is `upsmon`, which is configured via the `/etc/nut/upsmon.conf` file. The following configuration file will be sufficient for the system in question:

```
RUN_AS_USER nut
MONITOR BR650ELCD@localhost 1 upsmon upspasswd master
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
POLLFREQ 5
POLLFREQUALERT 5
```

```
HOSTSYNC 15
DEADTIME 15
POWERDOWNFLAG /etc/killpower
RBWARNTIME 43200
NOCOMMWARNTIME 300
FINALDELAY 5
```

In this configuration file, most of the parameters have default values. Explanations to them can be easily found in

```
root@server:/# man 5 upsmon.conf
```

It should only be noted that the upsmon process is run on behalf of the nut system user and monitors the status of the BRL650ELCD UPS connected to the server and also managed by the process.

Finally, in the `/etc/nut/nut.conf` file, select the operating mode of the entire NUT system. In the situation under consideration this file should contain (except for comments) only the line

```
MODE=standalone
```

A number of observations should be made regarding the security of the system. In the upsmon.conf and upsd.conf files a password is specified in plain text that allows you to control the power supply of the system even if only locally. Therefore, these files should not be readable by anyone except the system user on whose behalf they work. At the same time, even for this user, they should be unavailable for recording. These measures have already been implemented by the maintainers of the operating system distribution, which is clearly visible from the following fragment of the terminal session:

```
root@server:/# ls -l /etc/nut
итого 24
-rw-r----- 1 root nut 1544 мая 31 10:12 nut.conf
-rw-r----- 1 root nut  208 мая 28 10:08 ups.conf
-rw-r----- 1 root nut   22 мая 31 10:02 upsd.conf
-rw-r----- 1 root nut 2184 мая 31 10:20 upsd.users
-rw-r----- 1 root nut  249 мая 31 10:32 upsmon.conf
-rw-r----- 1 root nut 3887 июн  1  2018 upssched.conf
```

After editing the configuration files is completed, it remains only to restart the three NUT component services in the specified order and make sure that this restart is successful:

```
root@server:/# systemctl stop nut-monitor
root@server:/# systemctl stop nut-server
root@server:/# systemctl stop nut-driver
root@server:/# systemctl start nut-driver
root@server:/# systemctl start nut-server
root@server:/# systemctl start nut-monitor
root@server:/# systemctl status nut-driver
• nut-driver.service - Network UPS Tools - power device driver controller
  Loaded: loaded (/lib/systemd/system/nut-driver.service; static; vendor preset:
  enabled)
  Active: active (running) since Mon 2021-05-31 15:57:09 +03; 17s ago
  Process: 25208 ExecStart=/sbin/upsdrvctl start (code=exited, status=0/SUCCESS)
  Main PID: 25210 (usbhid-ups)
  Tasks: 1 (limit: 2314)
  Memory: 856.0K
  CGroup: /system.slice/nut-driver.service
          └─25210 /lib/nut/usbhid-ups -a BR650ELCD
```

```

мая 31 15:57:09 server systemd[1]: Starting Network UPS Tools - power device driver
controller...
мая 31 15:57:09 server upsdrvctl[25208]: Using subdriver: CyberPower HID 0.4
мая 31 15:57:09 server upsdrvctl[25208]: Network UPS Tools - Generic HID driver 0.41
(2.7.4)
мая 31 15:57:09 server upsdrvctl[25208]: USB communication driver 0.33
мая 31 15:57:09 server upsdrvctl[25208]: cps_adjust_battery_scale: battery readings
will be scaled by 2/3
мая 31 15:57:09 server upsdrvctl[25208]: Network UPS Tools - UPS driver controller
2.7.4
мая 31 15:57:09 server usbhid-ups[25210]: Startup successful
мая 31 15:57:09 server systemd[1]: Started Network UPS Tools - power device driver
controller.
root@server:/# systemctl status nut-server
• nut-server.service - Network UPS Tools - power devices information server
  Loaded: loaded (/lib/systemd/system/nut-server.service; enabled; vendor preset:
enabled)
  Active: active (running) since Mon 2021-05-31 15:57:09 +03; 25s ago
  Process: 25211 ExecStart=/sbin/upsd (code=exited, status=0/SUCCESS)
Main PID: 25212 (upsd)
  Tasks: 1 (limit: 2314)
  Memory: 716.0K
  CGroup: /system.slice/nut-server.service
          └─25212 /lib/nut/upsd

мая 31 15:57:09 server systemd[1]: Starting Network UPS Tools - power devices
information server...
мая 31 15:57:09 server upsd[25211]: fopen /var/run/nut/upsd.pid: No such file or
directory
мая 31 15:57:09 server upsd[25211]: listening on 127.0.0.1 port 3493
мая 31 15:57:09 server upsd[25211]: listening on 127.0.0.1 port 3493
мая 31 15:57:09 server upsd[25211]: Connected to UPS [BR650ELCD]: usbhid-ups-
BR650ELCD
мая 31 15:57:09 server upsd[25211]: Connected to UPS [BR650ELCD]: usbhid-ups-
BR650ELCD
мая 31 15:57:09 server upsd[25212]: Startup successful
мая 31 15:57:09 server systemd[1]: Started Network UPS Tools - power devices
information server.
мая 31 15:57:16 server upsd[25212]: User upsmon@127.0.0.1 logged into UPS [BR650ELCD]

root@server:/# systemctl status nut-monitor
• nut-monitor.service - Network UPS Tools - power device monitor and shutdown
controller
  Loaded: loaded (/lib/systemd/system/nut-monitor.service; enabled; vendor preset:
enabled)
  Active: active (running) since Mon 2021-05-31 15:57:16 +03; 28s ago
  Process: 25215 ExecStart=/sbin/upsmon (code=exited, status=0/SUCCESS)
Main PID: 25217 (upsmon)
  Tasks: 2 (limit: 2314)
  Memory: 1.1M
  CGroup: /system.slice/nut-monitor.service
          └─25216 /lib/nut/upsmon
            └─25217 /lib/nut/upsmon

мая 31 15:57:16 server systemd[1]: Starting Network UPS Tools - power device monitor
and shutdown controller...
мая 31 15:57:16 server upsmon[25215]: fopen /var/run/nut/upsmon.pid: No such file or
directory
мая 31 15:57:16 server upsmon[25215]: UPS: BR650ELCD@localhost (master) (power value
1)
мая 31 15:57:16 server upsmon[25215]: Using power down flag file /etc/killpower

```

```
мая 31 15:57:16 server upsmon[25216]: Startup successful
мая 31 15:57:16 server systemd[1]: nut-monitor.service: Can't open PID file /run/nut/upsmon.pid (yet?) after start: No such file or directory
мая 31 15:57:16 server systemd[1]: nut-monitor.service: Supervising process 25217 which is not our child. We'll most likely not notice when it exits.
мая 31 15:57:16 server systemd[1]: Started Network UPS Tools - power device monitor and shutdown controller.
мая 31 15:57:16 server upsmon[25217]: Init SSL without certificate database
```

Saving the order of restarting services allows you to perform it without additional messages about temporary disconnections.

## Fail protection (auto-resume)

To organize the automatic restoration of the of services operability after a failure, you can use the means of the systemd initialization system. At the same time, the loss of communication with the UPS is not a failure, it is considered only an emergency termination of one of the processes, which, although extremely unlikely, can be fixed by the following actions.

First of all, you should check the current parameters of the services with the commands:

```
root@server:/# systemctl show nut-driver
Type=forking
Restart=no
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
root@server:/# systemctl show nut-server
Type=forking
Restart=no
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
root@server:/# systemctl show nut-monitor
Type=forking
Restart=no
PIDFile=/run/nut/upsmon.pid
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
```

As can be seen from the response of the commands (it is given here in a strongly abbreviated form, containing important lines in the current context), restarting services is disabled.

Then you should edit each service using the commands:

```
root@server:/# systemctl edit nut-driver
root@server:/# systemctl edit nut-server
root@server:/# systemctl edit nut-monitor
```

At the same time, a text editor will be opened in which certain edits need to be made. When confirming the saving of changes in the system, the corresponding files are created:

```
/etc/systemd/system/nut-driver.service.d/override.conf
/etc/systemd/system/nut-server.service.d/override.conf
/etc/systemd/system/nut-monitor.service.d/override.conf
```

The next time the service is started, they override the original parameter values. To solve the problem of restarting the service in case of failures, the following content is sufficient, the same for all files:

```
[Service]
Restart=on-failure
```

Such changes in the services are saved even when installing system updates.

After making the changes, restart the services and make sure that the parameter changes are taken into account:

```
root@server:/# systemctl stop nut-monitor
root@server:/# systemctl stop nut-server
root@server:/# systemctl stop nut-driver
root@server:/# systemctl start nut-driver
root@server:/# systemctl start nut-server
root@server:/# systemctl start nut-monitor
root@server:/# systemctl show nut-driver
Type=forking
Restart=on-failure
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
root@server:/# systemctl show nut-server
Type=forking
Restart=on-failure
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
root@server:/# systemctl show nut-monitor
Type=forking
Restart=on-failure
PIDFile=/run/nut/upsmon.pid
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
...
```

## Verification of performance and stability

To check the current status of the UPS, use the command

```
root@server:/# upsc BR650ELCD
Init SSL without certificate database
battery.charge: 100
battery.charge.low: 10
battery.charge.warning: 20
battery.mfr.date: CPS
battery.runtime: 2100
battery.runtime.low: 300
```

```
battery.type: PbAcid
battery.voltage: 13.6
battery.voltage.high: 13.6
battery.voltage.low: 10.5
battery.voltage.nominal: 12
device.mfr: CPS
device.model: UPS VALUE
device.type: ups
driver.name: usbhid-ups
driver.parameter.pollfreq: 30
driver.parameter.pollinterval: 2
driver.parameter.port: auto
driver.parameter.productid: 0501
driver.parameter.synchronous: no
driver.parameter.vendorid: 0764
driver.version: 2.7.4
driver.version.data: CyberPower HID 0.4
driver.version.internal: 0.41
input.transfer.high: 280
input.transfer.low: 180
input.voltage: 234.0
input.voltage.nominal: 230
output.voltage: 234.0
ups.beeper.status: enabled
ups.delay.shutdown: 20
ups.delay.start: 30
ups.load: 20
ups.mfr: CPS
ups.model: UPS VALUE
ups.productid: 0501
ups.realpower.nominal: 390
ups.status: OL
ups.test.result: Done and passed
ups.timer.shutdown: -1
ups.timer.start: 0
ups.vendorid: 0764
```

The output of this command clearly shows all the parameters of the uninterruptible power supply. So, for example, the current battery charge level is 100% (battery.charge), the low charge at which a power outage is initiated is considered to be a level of 10% (battery.charge.low), and the system should be warned about battery discharge from the level of 20% (battery.charge.warning). At the same time, power is currently supplied from the power grid (ups.status: OL), and the battery voltage at 100% charge is 13.6 V.

Further testing of the operability and stability of the NUT system can be divided into several experiments:

### **Short-term loss of communication with UPS**

A short-term loss of connection to the power supply can be simulated by physically disconnecting the USB cable from the UPS. At the same time, according to the current upsmon settings, a short-term shutdown is considered to be no more than 300 seconds.

5 seconds after disconnecting the cable, a broadcast message appeared in the system:

```
Broadcast message from nut@server (somewhere) (Wed Jun  2 15:14:11 2021):
Communications with UPS BR650ELCD@localhost lost
```

After connecting the cable, another such message appeared:

```
Broadcast message from nut@server (somewhere) (Wed Jun  2 15:14:21 2021):  
Communications with UPS BR650ELCD@localhost established
```

Obviously, the system reacted to both events. The same messages are also reflected in the system log:

```
root@server:/# cat /var/log/syslog | grep ups  
... Jun  2 15:14:08 server upsd[723]: Data for UPS [BR650ELCD] is stale - check driver  
Jun  2 15:14:11 server upsmon[726]: Poll UPS [BR650ELCD@localhost] failed - Data  
stale  
Jun  2 15:14:11 server upsmon[726]: Communications with UPS BR650ELCD@localhost lost  
Jun  2 15:14:16 server upsmon[726]: Poll UPS [BR650ELCD@localhost] failed - Data  
stale  
Jun  2 15:14:16 server upsd[723]: UPS [BR650ELCD] data is no longer stale  
Jun  2 15:14:21 server upsmon[726]: Communications with UPS BR650ELCD@localhost  
established
```

### Long-term loss of communication with UPS

The experiment is completely similar to the previous one, as well as its result: the lack of communication with the UPS for 15 minutes did not provoke a system shutdown.

### Emergency termination of NUT processes

The experiment should be carried out in the following order: find out the process IDs, then crash them and observe the broadcast messages that appear:

```
root@server:/# ps -ef | grep ups  
nut      444      1  0 10:00 ?        00:00:01 /lib/nut/usbhid-ups -a BR650ELCD  
nut      462      1  0 10:00 ?        00:00:00 /lib/nut/upsd  
root     465      1  0 10:00 ?        00:00:00 /lib/nut/upsmon  
nut      466     465  0 10:00 ?        00:00:00 /lib/nut/upsmon  
root     634     496  0 13:53 pts/0    00:00:00 grep ups  
root@server:/# kill -9 444
```

```
Broadcast message from nut@server (somewhere) (Thu Jun  3 13:54:15 2021):  
Communications with UPS BR650ELCD@localhost lost
```

```
Broadcast message from nut@server (somewhere) (Thu Jun  3 13:54:20 2021):  
Communications with UPS BR650ELCD@localhost established
```

```
root@server:/# kill -9 462
```

```
Broadcast message from nut@server (somewhere) (Thu Jun  3 13:54:35 2021):  
Communications with UPS BR650ELCD@localhost lost
```

```
Broadcast message from nut@server (somewhere) (Thu Jun  3 13:54:40 2021):  
Communications with UPS BR650ELCD@localhost established
```

```
root@server:/# kill -9 465
```

```
Broadcast message from nut@server (somewhere) (Thu Jun  3 13:54:55 2021):  
upsmon parent process died - shutdown impossible
```

From the above fragment of the terminal session, you can see that the driver and daemon processes restarted without any comments, and the monitoring process was started in two instances: parent and child. At the end of the parent process running on behalf of the superuser, the system began to issue warning messages that it was impossible to turn off the system when



the UPS battery was low. At the same time, this message is repeated in the terminal session every two minutes. Calling the command again

```
root@server:/# ps -ef | grep ups
nut      466      1  0 10:00 ?        00:00:00 /lib/nut/upsmon
nut      638      1  1 13:54 ?        00:00:07 /lib/nut/usbhid-ups -a BR650ELCD
nut      646      1  0 13:54 ?        00:00:00 /lib/nut/upsd
root     688    496  0 14:06 pts/0    00:00:00 grep ups
```

demonstrates that the upsmon child process continues to run on behalf of nut, but there is no new parent process running on behalf of root. At the same time, the systemd nut-monitor service remains active, though.

The forced restart of the service completely corrected the situation, as can be seen from the new upsmon process IDs.:

```
root@server:/# systemctl restart nut-monitor
root@server:/# systemctl status nut-monitor
• nut-monitor.service - Network UPS Tools - power device monitor and shutdown
controller
   Loaded: loaded (/lib/systemd/system/nut-monitor.service; enabled; vendor preset:
enabled)
   Drop-In: /etc/systemd/system/nut-monitor.service.d
            └─override.conf
   Active: active (running) since Thu 2021-06-03 14:12:21 +03; 2s ago
...
root@server:/# ps -ef | grep upsmon
root      705      1  0 14:12 ?        00:00:00 /lib/nut/upsmon
nut       706     705  0 14:12 ?        00:00:00 /lib/nut/upsmon
root      712    496  0 14:12 pts/0    00:00:00 grep upsmon
```

During the repeated experiment, the child process was crash-terminated, and the service was automatically restarted without any issues. At the same time, both the child and parent processes received new identifiers, i. e. they were also restarted.

Thus, we can conclude that the operability of the services has been confirmed, and measures to increase the stability of these services have been taken not only when configuring this server, but also when developing the NUT system itself. This explains the two upsmon processes, because the parent process does not directly perform useful work, but monitors the status of the running child process and, if necessary, informs the system, causing the entire service to restart. This separation of processes allows you to transfer almost all the risks of an emergency shutdown into the space of a child process.

### Switching to battery power and back without running out of battery

Imitation of this situation is trivial: it is enough to disconnect the UPS from the power grid and reconnect it after a short period of time. The following broadcast messages were received in the terminal of the system:

```
Broadcast message from nut@server (somewhere) (Thu Jun  3 14:31:18 2021):
UPS BR650ELCD@localhost on battery
```

```
Broadcast message from nut@server (somewhere) (Thu Jun  3 14:31:38 2021):
UPS BR650ELCD@localhost on line power
```

They clearly demonstrate the correct functioning of the system.

## Switching to battery power and running out of battery

For continuous monitoring of the UPS status during this experiment, it makes sense to run the command

```
root@server:/# watch upsc BR650ELCD
```

When the UPS was disconnected from the power grid, a broadcast message appeared after about two seconds

```
Broadcast message from nut@server (somewhere) (Fri Jun  4 13:56:18 2021):  
UPS BR650ELCD@localhost on battery
```

Then there was a uniform decrease in the battery charge level, the uninterruptible power supply periodically issued an audio alert, its indicator showed a load of 11%, which coincided with the data received from upsc. Finally, after 50 minutes of battery life, its charge was exhausted by 90%, the system decided to disconnect, as evidenced by the following broadcast messages:

```
Broadcast message from nut@server (somewhere) (Fri Jun  4 14:46:49 2021):  
Executing automatic power-fail shutdown
```

```
Broadcast message from nut@server (somewhere) (Fri Jun  4 14:46:49 2021):  
UPS BR650ELCD@localhost battery is low
```

```
Broadcast message from nut@server (somewhere) (Fri Jun  4 14:46:49 2021):  
Auto logout and shutdown proceeding
```

```
exit
```

```
Session terminated, killing shell... ...killed.  
Terminated
```

Thus, the system worked as expected, shutting down the server normally and saving some of the battery charge in order to prevent its deep discharge and the resource reduce.

## Return of power after system shutdown

After some time (about a minute) after the completion of the previous experiment, the UPS was reconnected to the power grid, after which the server automatically turned on. This result is expected, because the server shutdown occurred precisely because of the disappearance of the external voltage for the UPS.

## Return of main power during system shutdown

The next time the system was connected and disconnected with a low battery charge, the UPS was re-connected to the power grid precisely during the server shutdown. The result was also desirable: the UPS waited for the system to shut down, paused for a while and initiated the server boot.

## Return of main power when the system is normally switched off

The last experiment in the series consisted of a regular shutdown of the server without reference to power events, followed by disconnection and reconnection of an external power supply to the UPS. The server was not turned on, which was to be expected.

Thus, the interaction of all system components (server, UPS, power grid) meets the task statement.

## RAID1 configuration

Two Hitachi disks with a capacity of 500 GB each are allocated for the deployment of the RAID array in the system under consideration as can be seen from the following fragment of the terminal session, abbreviated in the non-informative part:

```
root@server:/# lshw
server
...
    *-disk:0
...
    *-disk:1
        description: ATA Disk
        product: HGST HTS725050A7
        physical id: 0.1.0
        bus info: scsi@0:0.1.0
        logical name: /dev/sdb
        version: A530
        serial: TF650AWE06VJ4V
        size: 465GiB (500GB)
...
    *-disk:2
        description: ATA Disk
        product: HGST HTS725050A7
        physical id: 1
        bus info: scsi@1:0.0.0
        logical name: /dev/sdc
        version: A530
        serial: TF650AWJ3UJ0TV
        size: 465GiB (500GB)
...
```

The disk with the serial number TF650AWE06VJ4V is installed in the upper slot in the server case, and the disk with the serial number TF650AWJ3UJ0TV, respectively, in the lower one. Knowing the physical location of the disks can somewhat speed up the process of replacing a failed disk. Despite the fact that spare hard drives of the same model are available at the time of system configuration, they are not installed in the server for the following reasons:

Firstly, the probability of failure of the disks is low. Secondly, the presence of a spare disk means additional power consumption and heat generation in the system throughout its operation. Thirdly, even though the spare disk will not be used to store data until one of the main ones fails, it will still start every time the system boots, and stop when the system is turned off, which will already be an aimless expenditure of its mechanical resource. In addition, the motherboard of the system in question does not support the "hot" connection/disconnection of disks, and therefore, if one of the disks fails, it will still need to be disconnected for system maintenance.

## Configuration

The RAID array configuration can be divided into several stages:

- installation of the necessary software
- array assembly and initialization
- setting up automatic array assembly at system boot

- creating a partition and file system on RAID
- setting up automatic mounting of a partition hosted on a RAID
- updating the initial boot disk of the system

First of all, you need to install the mdadm package:

```
root@server:/# apt-get install mdadm
```

Creating an array is performed by a command like:

```
root@server:/# PATH=$PATH:/sbin
root@server:/# mdadm --create /dev/md0 /dev/sdb /dev/sdc --level=1 --raid-devices=2
mdadm: partition table exists on /dev/sdb
mdadm: partition table exists on /dev/sdb but will be lost or
      meaningless after creating array
mdadm: Note: this array has metadata at the start and
      may not be suitable as a boot device. If you plan to
      store '/boot' on this device please ensure that
      your boot-loader understands md/v1.x metadata, or use
      --metadata=0.90
mdadm: partition table exists on /dev/sdc
mdadm: partition table exists on /dev/sdc but will be lost or
      meaningless after creating array
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Explanations should be given to the messages received during the creation of the array. So mdadm warns that there is a partition table on each of the disks, which will be lost when creating an array. The system also warns that the metadata of the array will be stored at the beginning of the disks, which means that in order to use the array as a boot partition, you should make sure that the system loader is able to cope with this. Obviously, saving partition tables on disks is not required, and the RAID array in the system in question is not a boot device. Therefore, the creation of the array is confirmed, the array is created and successfully launched, as can be seen from the output of the lsblk command:

```
root@server:/# lsblk
NAME      MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda         8:0    0 298,1G  0 disk
├─sda1      8:1    0  18,6G  0 part  /
├─sda2      8:2    0   3,7G  0 part  [SWAP]
├─sda3      8:3    0  18,6G  0 part  /var
└─sda4      8:4    0 257,1G  0 part  /srv/samba
sdb         8:16    0 465,8G  0 disk
└─md0       9:0    0 465,7G  0 raid1
sdc         8:32    0 465,8G  0 disk
└─md0       9:0    0 465,7G  0 raid1
```

Now you need to create a file system on the running array:

```
root@server:/# mkfs.ext4 /dev/md0
mke2fs 1.44.5 (15-Dec-2018)
Creating filesystem with 122063616 4k blocks and 30523392 inodes
Filesystem UUID: e401ae19-45a0-4766-aed0-fd34ebd42f8f
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000
Allocating group tables: done
```

```
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
```

Upon completion, it is very useful, although not strictly necessary, to perform a trial mounting of the system:

```
root@server:/# mkdir /srv/samba/archive
root@server:/# mount /dev/md0 /srv/samba/archive/
root@server:/# mount | grep md0
/dev/md0 on /srv/samba/archive type ext4 (rw,relatime)
```

Next, an entry about mounting the array must be entered in the **/etc/fstab** file, obviously so that mounting occurs automatically when the system boots. Despite the fact that the entry in this file may begin with the device name (**/dev/md0**), and not a unique identifier (UUID), for consistency with previous entries in it and unambiguous identification of the array, the UUID is used). To get this ID the following command can be used:

```
root@server:/# blkid /dev/md0
/dev/md0: UUID="e401ae19-45a0-4766-aed0-fd34ebd42f8f" TYPE="ext4"
```

Taking into account the logic of using the array, it makes sense to set fairly strict mount parameters, which eventually leads to such (the above UUID is shortened to accommodate the entry on the page in one line) entries in **/etc/fstab**:

```
UUID=e4...f8f    /srv/samba/archive ext4    nodev,nosuid,noexec    0    2
```

To fine-tune the array, first of all you need to fill in the **/etc/mdadm/mdadm.conf** file according to the manual

```
root@server:/# man 5 mdadm.conf
```

Most of the data needed for this can be obtained from the output of commands

```
root@server:/# /sbin/mdadm --examine --scan
root@server:/# lsblk
```

In the system under consideration, this file should be reduced to the form up to the comments:

```
HOMEHOST <system>
DEVICE /dev/sdb /dev/sdc
ARRAY /dev/md0 devices=/dev/sdb,/dev/sdc
AUTO homehost -all
MAILADDR root@localhost
```

The term **<system>** in the first line means that the hostname should be taken from the system settings. The **AUTO** string describes which arrays are to be automatically assembled when the system boots. In this case, only arrays should be collected whose metadata contains the same hostname as the one used by the system. In the **ARRAY** string, you should specify the minimum set of parameters required to build the array. On the system in question, only the disk names are sufficient.

After editing the file, you should update the image of the initial RAM disk (initrd), reboot the system, and after restarting, make sure using the mount command or the methods described below that the array and the file system on it are working properly:

```

root@server:/# PATH=$PATH:/sbin
root@server:/# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-4.19.0-16-amd64
root@server:/# reboot

```

When using RAID arrays, it is important not only to be able to create such an array, but also to have all the necessary tools to control its current state, access data when its components are damaged, and replace these components. The steps required for this are described further in the section "Using and maintaining the server".

## Fail protection (auto-resume)

To monitor the array state in real time, the mdadm process is continuously running in the system:

```

root@server:/# ps -ef | grep mdadm
root      301      1  0  08:00 ?                00:00:00 /sbin/mdadm --monitor --scan

```

This process is started by the mdmonitor service, which can be seen in the output of the following command:

```

root@server:/# systemctl status mdmonitor
• mdmonitor.service - MD array monitor
   Loaded: loaded (/lib/systemd/system/mdmonitor.service; static; vendor preset:
   enabled)
   Active: active (running) since Tue 2021-05-25 08:00:36 +03; 6h ago
 Main PID: 301 (mdadm)
    Tasks: 1 (limit: 2314)
   Memory: 476.0K
    CGroup: /system.slice/mdmonitor.service
            └─301 /sbin/mdadm --monitor --scan
...

```

By default, this service is not configured to restart automatically:

```

root@server:/# systemctl show mdmonitor
Type=simple
Restart=no
NotifyAccess=none
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
...

```

Therefore, you should edit the service (like the others previously described) using the command:

```

root@server:/# systemctl edit mdmonitor

```

A text editor would be opened, in which you need to make some edits. They would be saved in **/etc/systemd/system/mdmonitor.service.d/override.conf**. The next time the service is started, they override the original parameter values. To solve the problem of restarting the service in case of failures, the following contents of this file are sufficient:

```

[Service]
Restart=on-failure

```

It is worth noting that such changes in the service are saved even when installing system updates.

After making the changes, restart the service and make sure that the parameter changes are taken into account:

```
root@server:/# systemctl restart mdmonitor
root@server:/# systemctl show mdmonitor
Type=forking
Restart=on-failure
NotifyAccess=none
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
...
```

## Verification of performance and stability

Checking the operability of the service is trivial: successfully mounting a file system located on an array means its operability.

Checking the stability of continuous monitoring of the array state system is demonstrated in the following fragment of the terminal session (the responses of some commands are reduced in an uninformative applied to the context part):

```
root@server:/# systemctl status mdmonitor
• mdmonitor.service - MD array monitor
  Loaded: loaded (/lib/systemd/system/mdmonitor.service; static; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/mdmonitor.service.d
           └─override.conf
  Active: active (running) since Tue 2021-05-25 14:51:16 +03; 4s ago
Main PID: 12788 (mdadm)
  Tasks: 1 (limit: 2314)
  Memory: 404.0K
  CGroup: /system.slice/mdmonitor.service
          └─12788 /sbin/mdadm --monitor --scan
```

мая 25 14:51:16 server systemd[1]: Started MD array monitor.

```
root@server:/# kill -9 12788
```

```
root@server:/# systemctl status mdmonitor
• mdmonitor.service - MD array monitor
  Loaded: loaded (/lib/systemd/system/mdmonitor.service; static; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/mdmonitor.service.d
           └─override.conf
  Active: active (running) since Tue 2021-05-25 14:51:36 +03; 1s ago
Main PID: 12797 (mdadm)
  Tasks: 1 (limit: 2314)
  Memory: 412.0K
  CGroup: /system.slice/mdmonitor.service
          └─12797 /sbin/mdadm --monitor --scan
```

мая 25 14:51:36 server systemd[1]: Started MD array monitor.

In this fragment, the status of the service is checked, the PID of the current mdadm process (12788) is found out from the response message, then this process is intentionally terminated, after which the status of the service is checked again and the PID of the newly launched mdadm process (12797) is found out.

## Samba configuration

According to the task statement, as well as the disk layout and the structure of the RAID array, the Samba service should provide the operation of three network directories using the smb protocol. The first of them acts as a network folder for the rapid exchange of files, the second is intended for placing files in it, which should be archived and placed in the third directory at the end of the working day. At the same time, only the third archive directory is located on the RAID array. The RAID array is mounted on the system as **/srv/samba/archive**, and the other two directories are located on a 240 GB partition of the system hard disk mounted as **/srv/samba**, and are called **/srv/samba/shared** and **/srv/samba/backup**, respectively. At the same time, the archive catalog is not directly accessible to accounting staff for writing. It only allows them read access. The other two directories are available for both reading and writing. As noted in the task statement, the archive is written to the server based on the contents of the **/srv/samba/backup** directory at the end of the working day. This makes it impossible to damage or destroy the contents of the archive by accidental or intentional actions of personnel or software (for example, an encryption virus, which is very likely, given the fact that accountants' workstations are running MS Windows). However, this approach makes the problem of archive disk space overflow urgent. To solve it, as part of a script that performs daily archiving, a number of commands are provided to check the available disk space and notify the system administrator about the threat of disk overflow. At the time of writing this document, a temporary method of informing is being used by creating a warning text file in the **/srv/samba/backup** directory, as well as making an entry in the system log using the logger command. In the near future, as the organization's internal mail server is put into operation, the system administrator will be informed about events in the system (disk overflow, failures, other problems) via internal email. The use of public mail servers at the moment is regarded as a threat to the security of the system, and local mail delivery is disabled, because it can cause an overflow of the system's disk space, as described in more detail below.

Before describing the configuration of the service, it is appropriate to briefly highlight its internal structure and related issues. So the service consists of two daemons: **smbd** and **nmbd**. The first one is responsible for network access to files and printers, the second one is responsible for servicing NetBIOS requests. The configuration of these daemons is performed by a single configuration file. Detailed information about **samba**, **smbd**, **nmbd** and the configuration file can be obtained from the following built-in manuals:

```
root@server:/# man 7 samba
root@server:/# man 8 smbd
root@server:/# man 8 nmbd
root@server:/# man 5 smb.conf
```

In addition, it should be noted that the user **buh** is associated with this service. It should be understood that there are actually two such users. The system user **buh** is deprived of the command shell, has no password and no ability to log in. It is only needed to manage access rights in network directories. In the internal database of Samba users there is a user of the same name, the password for which is, on the contrary, assigned. This user and his/her password are meant to be used when connecting to the network directory from the client system.

## Configuration

First of all, you will need to install the packages:



```
root@server:/# apt-get install samba zip
```

And then bring the configuration file **/etc/samba/smb.conf** to the form:

```
[global]
workgroup = ACCOUNT
netbios name = samba
interfaces = enp3s0 lo
bind interfaces only = yes
hosts allow = 10.9.1.128/26 127.0.0.1
hosts deny = ALL
security = user
passdb backend = tdbsam
domain logons = no
domain master = no
```

```
[shared]
comment = Текущий обмен файлами
path = /srv/samba/shared
valid users = buh
force user = buh
force group = buh
read only = no
guest ok = no
```

```
[backup]
comment = Каталог для архивирования
path = /srv/samba/backup
valid users = buh
force user = buh
force group = buh
read only = no
guest ok = no
```

```
[archive]
comment = Архив
path = /srv/samba/archive
valid users = buh
force user = buh
force group = buh
read only = yes
guest ok = no
```

The given configuration file consists of several sections: global and describing each network directory separately. The global section specifies the system name for NetBIOS, the name of the workgroup for network discovery services of client operating systems, addresses and interfaces allowed for receiving requests, security mechanisms used and their parameters. For each network directory, in addition to the obviously necessary path, a number of parameters are also specified in the local directory tree that determine interaction with it: text comments are defined, the users who are allowed to access them, as well as the user and group on whose behalf all actions in these directories will be performed. A detailed description of each such parameter is meaningless here because their names are transparent, detailed, voluminous and well-structured documentation is provided by the above-mentioned built-in manuals.

The next step is to create these directories (except for the archive directory created when deploying the RAID array) and set the appropriate access rights to them (including archive):

```
root@server:/# mkdir /srv/samba/shared /srv/samba/backup
root@server:/# chown buh:buh -R /srv/samba/shared
```

```

root@server:/# chown buh:buh -R /srv/samba/backup
root@server:/# chown buh:buh -R /srv/samba/archive
root@server:/# chmod 0700 -R /srv/samba/shared
root@server:/# chmod 0700 -R /srv/samba/backup
root@server:/# chmod 0500 -R /srv/samba/archive

```

At the same time, it should be noted that the backup script run on behalf of the superuser, with corresponding access rights, still has the ability to create files in the archive directories.

Next, you will need to add the user buh to the Samba password database:

```

root@server:/# smbpasswd -a buh

```

After that, it is recommended to check the correctness of filling in the configuration file with the command

```

root@server:/# testparm

```

The response of this command almost completely duplicates the configuration file and therefore is not given here.

Finally, it remains to restart the smbd and nmbd services and make sure that this restart is successful:

```

root@server:/# service smbd restart
root@server:/# service nmbd restart
root@server:/# systemctl status smbd
• smbd.service - Samba SMB Daemon
  Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2021-06-08 15:27:52 +03; 2min 53s ago
    Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Process: 517 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile
 (code=exited, status=0/SUCCESS)
 Main PID: 522 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 4 (limit: 2314)
  Memory: 7.2M
   CGroup: /system.slice/smbd.service
           └─522 /usr/sbin/smbd --foreground --no-process-group
             └─525 /usr/sbin/smbd --foreground --no-process-group
               └─526 /usr/sbin/smbd --foreground --no-process-group
                 └─527 /usr/sbin/smbd --foreground --no-process-group

```

```

июн 08 15:27:52 server systemd[1]: Starting Samba SMB Daemon...
июн 08 15:27:52 server smbd[522]: [2021/06/08 15:27:52.848478,  0]
../lib/util/become_daemon.c:138(daemon_ready)
июн 08 15:27:52 server systemd[1]: Started Samba SMB Daemon.
июн 08 15:27:52 server smbd[522]:  daemon_ready: STATUS=daemon 'smbd' finished
starting up and ready to serve connections

```

```

root@server:/# systemctl status nmbd
• nmbd.service - Samba NMB Daemon
  Loaded: loaded (/lib/systemd/system/nmbd.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2021-06-08 15:05:56 +03; 24min ago
    Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
 Main PID: 435 (nmbd)
   Status: "nmbd: ready to serve connections..."
    Tasks: 1 (limit: 2314)

```

```
Memory: 14.5M
CGroup: /system.slice/nmbd.service
└─435 /usr/sbin/nmbd --foreground --no-process-group
```

```
июн 08 15:27:22 server nmbd[435]:
июн 08 15:27:22 server nmbd[435]: Samba name server SAMBA has stopped being a local
master browser for workgroup ACCOUNT on subnet 10.9.1.129
июн 08 15:27:22 server nmbd[435]:
июн 08 15:27:22 server nmbd[435]: *****
июн 08 15:27:39 server nmbd[435]: [2021/06/08 15:27:39.479661, 0]
../source3/nmbd/nmbd_become_lmb.c:397(become_local_master_stage2)
июн 08 15:27:39 server nmbd[435]: *****
июн 08 15:27:39 server nmbd[435]:
июн 08 15:27:39 server nmbd[435]: Samba name server SAMBA is now a local master
browser for workgroup ACCOUNT on subnet 10.9.1.129
июн 08 15:27:39 server nmbd[435]:
июн 08 15:27:39 server nmbd[435]: *****
```

## Fail protection (auto-resume)

To organize the automatic recovery of the `smbd` and `nmbd` daemons after a failure, you can, like other previously described services, use the means of the `systemd` initialization system.

From the following fragment of the terminal session demonstrating the initial state of Samba services, you can see that their automatic restart is not provided:

```
root@server:/# systemctl show smbd
```

```
Type=notify
Restart=no
PIDFile=/run/samba/smbd.pid
NotifyAccess=all
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
```

```
...
```

```
root@server:/# systemctl show nmbd
```

```
Type=notify
Restart=no
PIDFile=/run/samba/nmbd.pid
NotifyAccess=all
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
```

```
...
```

Then you should edit each service using the commands:

```
root@server:/# systemctl edit smbd
```

```
root@server:/# systemctl edit nmbd
```

At the same time, a text editor will be opened in which certain edits need to be made. When confirming the saving of changes in the system, the corresponding files are created:

```
/etc/systemd/system/smbd.service.d/override.conf
```

```
/etc/systemd/system/nmbd.service.d/override.conf
```

The next time the service is started, they override the original parameter values. To solve the problem of restarting the service in case of failures, the following content is sufficient, the same for both files:

```
[Service]
Restart=on-failure
```

These changes in the services are also saved when installing system updates.

After making changes, you should restart the services and make sure that the parameter changes are taken into account:

```
root@server:/# systemctl restart smbd
```

```
root@server:/# systemctl show smbd
```

```
Type=notify
Restart=on-failure
PIDFile=/run/samba/smbd.pid
NotifyAccess=all
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
...
```

```
root@server:/# systemctl restart nmbd
```

```
root@server:/# systemctl show nmbd
```

```
Type=notify
Restart=on-failure
PIDFile=/run/samba/nmbd.pid
NotifyAccess=all
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
...
```

## Verification of performance and stability

The system performance and stability check can be divided into the following steps:

- checking the availability of the server by name, ip address and network discovery tools
- checking the availability of the network directory for online file exchange for reading and writing
- checking the availability of the network directory for archiving for reading and writing
- checking the availability of the network archive for reading and unavailability for writing (deleting files)

A client running Microsoft Windows 7 was used for verification. The initial verification was performed by means of the graphical user interface and was successful. The following fragment of the terminal session demonstrates the discovery of network directories and step-by-step verification of each of them by means of the Windows command line. It should be noted that to check the archive directory in it by means of the server itself, a test file was created, which was also deleted by means of the server upon completion of testing.

Microsoft Windows [Version 6.1.7601]

(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

```
C:\Users\user>net view /all
```

```
Имя сервера          Заметки
```

```
-----
\\SAMBA                Samba 4.9.5-Debian
```

```
Команда выполнена успешно.
```

```
C:\Users\user>net use X: \\samba.account.school34\shared
```

```
Команда выполнена успешно.
```

```
C:\Users\user>dir X:\
```

```

Том в устройстве X имеет метку shared
Серийный номер тома: 9EDD-DA4F
Содержимое папки X:\
09.06.2021  10:12    <DIR>          .
18.05.2021  15:11    <DIR>          ..
                0 файлов                0 байт
                2 папок  256 781 373 440 байт свободно
C:\Users\user>echo sample file > X:\sample.txt
C:\Users\user>type X:\sample.txt
sample file
C:\Users\user>del X:\sample.txt
C:\Users\user>dir X:\
Том в устройстве X имеет метку shared
Серийный номер тома: 9EDD-DA4F
Содержимое папки X:\
09.06.2021  10:39    <DIR>          .
18.05.2021  15:11    <DIR>          ..
                0 файлов                0 байт
                2 папок  256 781 373 440 байт свободно
C:\Users\user>net use Y: \\samba.account.school34\backup
Команда выполнена успешно.
C:\Users\user>dir Y:\
Том в устройстве Y имеет метку backup
Серийный номер тома: 2A02-4E9E
Содержимое папки Y:\
18.05.2021  15:10    <DIR>          .
18.05.2021  15:11    <DIR>          ..
                0 файлов                0 байт
                2 папок  256 781 373 440 байт свободно
C:\Users\user>echo sample file > Y:\sample.txt
C:\Users\user>type Y:\sample.txt
sample file
C:\Users\user>del Y:\sample.txt
C:\Users\user>dir Y:\
Том в устройстве Y имеет метку backup
Серийный номер тома: 2A02-4E9E
Содержимое папки Y:\
09.06.2021  10:41    <DIR>          .
18.05.2021  15:11    <DIR>          ..
                0 файлов                0 байт
                2 папок  256 781 373 440 байт свободно
C:\Users\user>net use Z: \\samba.account.school34\archive
Команда выполнена успешно.
C:\Users\user>dir Z:\
Том в устройстве Z имеет метку archive
Серийный номер тома: D1E3-6E0D
Содержимое папки Z:\
09.06.2021  10:43    <DIR>          .
18.05.2021  15:11    <DIR>          ..
09.06.2021  10:43                24 sample.archive.txt
18.05.2021  15:06    <DIR>          lost+found
                1 файлов                24 байт
                3 папок  465 958 961 152 байт свободно
C:\Users\user>type Z:\sample.archive.txt
sample readonly archive
C:\Users\user>del Z:\sample.archive.txt
Z:\sample.archive.txt
Отказано в доступе.
C:\Users\user>echo sample file > Z:\sample.txt
Отказано в доступе.

```

During the above terminal session, the directories were connected as network drives X:\ , Y:\ , and Z:\. After that, the simplest text files were successfully created, read and deleted in the first two of them. For the last network directory (archive), reading the file in it was successful, but all other operations were not. Thus, the operability of the system is confirmed.

The system's resilience to failures of `smbd` and `nmbd` daemons is checked similarly to other services. To do this, process IDs are found out, these processes are forcibly terminated, and then the state of the services is checked. All these actions are reflected in the following fragment of the terminal session (responses of some commands are shortened in an uninformative part applied to the context):

```
root@server:/# systemctl status smbd
```

```
• smbd.service - Samba SMB Daemon
  Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/smbd.service.d
           └─override.conf
  Active: active (running) since Wed 2021-06-09 13:01:50 +03; 8min ago
  Docs: man:smbd(8)
        man:samba(7)
        man:smb.conf(5)
  Process: 880 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile
 (code=exited, status=0/SUCCESS)
  Main PID: 884 (smbd)
  Status: "smbd: ready to serve connections..."
  Tasks: 5 (limit: 2314)
  Memory: 8.1M
  CGroup: /system.slice/smbd.service
          └─884 /usr/sbin/smbd --foreground --no-process-group
            └─886 /usr/sbin/smbd --foreground --no-process-group
              └─887 /usr/sbin/smbd --foreground --no-process-group
                └─888 /usr/sbin/smbd --foreground --no-process-group
                  └─890 /usr/sbin/smbd --foreground --no-process-group
```

```
root@server:/# kill -9 884
```

```
root@server:/# systemctl status smbd
```

```
• smbd.service - Samba SMB Daemon
  Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/smbd.service.d
           └─override.conf
  Active: active (running) since Wed 2021-06-09 13:11:14 +03; 3s ago
  Docs: man:smbd(8)
        man:samba(7)
        man:smb.conf(5)
  Process: 904 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile
 (code=exited, status=0/SUCCESS)
  Main PID: 908 (smbd)
  Status: "smbd: ready to serve connections..."
  Tasks: 4 (limit: 2314)
  Memory: 7.2M
  CGroup: /system.slice/smbd.service
          └─908 /usr/sbin/smbd --foreground --no-process-group
            └─911 /usr/sbin/smbd --foreground --no-process-group
              └─912 /usr/sbin/smbd --foreground --no-process-group
                └─913 /usr/sbin/smbd --foreground --no-process-group
```

```
root@server:/# systemctl status nmbd
```

```
• nmbd.service - Samba NMB Daemon
  Loaded: loaded (/lib/systemd/system/nmbd.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/nmbd.service.d
           └─override.conf
  Active: active (running) since Wed 2021-06-09 13:14:47 +03; 38s ago
  Docs: man:nmbd(8)
```

```

man:samba(7)
man:smb.conf(5)
Main PID: 961 (nmbd)
Status: "nmbd: ready to serve connections..."
Tasks: 1 (limit: 2314)
Memory: 2.3M
CGroup: /system.slice/nmbd.service
└─961 /usr/sbin/nmbd --foreground --no-process-group
root@server:/# kill -9 961
root@server:/# systemctl status nmbd
• nmbd.service - Samba NMB Daemon
  Loaded: loaded (/lib/systemd/system/nmbd.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/nmbd.service.d
           └─override.conf
  Active: active (running) since Wed 2021-06-09 13:15:41 +03; 3s ago
    Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
Main PID: 967 (nmbd)
Status: "nmbd: ready to serve connections..."
Tasks: 1 (limit: 2314)
Memory: 2.3M
CGroup: /system.slice/nmbd.service
└─967 /usr/sbin/nmbd --foreground --no-process-group

```

## Setting up a backup system

The backup system described in detail in the task statement is implemented by several files. The key role is played by the shell script, which performs data archiving, appending a directory reference and turning off the computer at the end of the working day. This script is located in the **/usr/local/sbin** directory and looks like:

```

root@server:/# cat /usr/local/sbin/server-shutdown
#!/bin/bash

OLDPATH=$PATH
PATH=$PATH:/sbin

SED=$(which sed)
ZIP=$(which zip)
UNZIP=$(which unzip)

BACKUP_DIR=/srv/samba/backup
ARCHIVE_DIR=/srv/samba/archive
ARCHIVE_INDEX=$ARCHIVE_DIR/index.html
ARCHIVE_INDEX_TEMPLATE=/usr/local/share/index.template
ARCHIVE_NAME=archive_$(date +%Y_%m_%d).zip
ARCHIVE=$ARCHIVE_DIR/$ARCHIVE_NAME
ARCHIVE_REF=file:///samba.account.school34/archive/$ARCHIVE_NAME

HTML_OPEN=$(cat <<EOF
    <p>
        <a href=$ARCHIVE_REF><h2>Архив от $(date "+%d %B %Y")</h2></a>
        <details> <summary>Содержимое</summary>
    <pre>
EOF
)
HTML_CLOSE=$(cat <<EOF
    </pre>
    </details>

```

```

        </p>
</body>
</html>
EOF
)

if [[ -n $SED && -n $ZIP && -n $UNZIP ]];
then
    logger -s -t server_shutdown -p syslog.info "server shutdown begins"
    logger -s -t server_shutdown -p syslog.info "stopping smbd and networking"
    systemctl stop smbd
    systemctl stop networking
    NEEDED_BLOCKS=$(du --total $BACKUP_DIR/* | tail -1 | sed s/"[:space:]].*"//g)
    NEEDED_INODES=1
    AVAIL_BLOCKS=$(df --output=avail $ARCHIVE_DIR | tail -1 | sed s/" " //g)
    AVAIL_INODES=$(df --output=iavail $ARCHIVE_DIR | tail -1 | sed s/" " //g)
    logger -s -t server_shutdown -p syslog.info "it is needed up to $NEEDED_BLOCKS
        blocks and $NEEDED_INODES inodes, whereas $AVAIL_BLOCKS
        S blocks and $AVAIL_INODES inodes available"
    if [[ $AVAIL_BLOCKS -gt $NEEDED_BLOCKS && $AVAIL_INODES -gt $NEEDED_INODES ]];
    then
        if [[ -n $(ls $BACKUP_DIR) ]];
        then
            cd $BACKUP_DIR
            zip -mqT $ARCHIVE -r ./
            if [ -z $(ls $BACKUP_DIR) ];
            then
                if [ ! -f /srv/samba/archive/index.html ];
                then
                    logger -s -t server_shutdown -p syslog.info "archive index does
                        not exist, new one created"
                    cat $ARCHIVE_INDEX_TEMPLATE > $ARCHIVE_INDEX
                    chown buh:buh $ARCHIVE_INDEX
                    chmod 0744 $ARCHIVE_INDEX
                fi
                sed s/"<\body.*"//g -i $ARCHIVE_INDEX
                sed s/"<\html.*"//g -i $ARCHIVE_INDEX
                echo $HTML_OPEN >> $ARCHIVE_INDEX
                unzip -l $ARCHIVE >> $ARCHIVE_INDEX
                echo $HTML_CLOSE >> $ARCHIVE_INDEX
                logger -s -t server_shutdown -p syslog.info "archive created, archive
                    index updated"
            else
                logger -s -t server_shutdown -p syslog.warning "failed to create
                    archive, all data remain in backup directory"
            fi
        else
            logger -s -t server_shutdown -p syslog.info "backup directory is empty,
                nothing to move to archive"
        fi
    else
        logger -s -t server_shutdown -p syslog.warning "available blocks or inodes
            are not enough, can't move data to archive"
    fi
else
    logger -s -t server_shutdown -p syslog.warning "sed, zip or unzip missing, can't
        move data to archive"
fi

PATH=$OLDPATH
/sbin/shutdown -P now

```



To provide the operability of this script, you will also need a template file for the archive's html reference. It is located in the **/usr/local/share** directory and looks like:

```
root@server:/# cat /usr/local/share/index.template
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Архив бухгалтерии МБОУ СОШ №34</title>
<style>
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
a {
    text-decoration: none;
    color: #1C61B5;
}
a:link {
    text-decoration: none;
}
a:hover {
    text-decoration: none;
}
a:active {
    color: #000000;
    text-decoration: underline;
}
h1 {
    color: #0a0afa;
    font-size: 28px;
    margin: 0px;
}
h2 {
    font-size: 20px;
    margin-left: 30px;
    margin-bottom: 10px
}
p {
    font-size: 14px;
    margin-left: 30px;
    margin-top: 10px;
    margin-right: 30px;
    margin-bottom: 10px;
}
details {
    margin-left: 30px;
    margin-top: 10px;
    margin-right: 30px;
    margin-bottom: 10px;
}
pre {
    font-size: 12px;
    margin-left: 30px;
    margin-top: 10px;
    margin-right: 30px;
    margin-bottom: 10px;
}
</style>
</head>
```

```
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
  <h1 align="center">Архив бухгалтерии МБОУ СОШ №34</h1>
</body>
</html>
```

This file contains the entire html page, on the basis of which the archive reference will be formed, along with style sheets, so there are no external dependencies on this page, which means that it can be made available over the network without loss of functionality by means of a file server without the need to deploy a web server in the system.

The script itself, when running, checks for the necessary utilities, finds out the available and necessary disk space needed to perform archiving, in the absence of a reference file, creates it based on the specified template and stops the network service and the network directory service. The script then creates an archive (if the backup directory is not empty) and, upon successful completion of this operation, clears the archive directory and supplements the file reference. If the archive directory is empty, then the archive is not created, the index file does not change. At the end, the entire system is turned off. The actions of the script are reflected in the system log, which is demonstrated in the following excerpt:

```
root@server:/# cat /var/log/messages | grep server_shutdown
...
Jun 11 15:07:46 server server_shutdown: server shutdown begins
Jun 11 15:07:46 server server_shutdown: stopping smbd and networking
Jun 11 15:07:46 server server_shutdown: it is needed up to 760 blocks and 1 inodes,
whereas 455038044 blocks and 30523379 inodes available
Jun 11 15:07:46 server server_shutdown: archive created, archive index updated
...
```

As for creating an archive index file, this may be necessary in two cases: the first launch of the server and complete cleaning of the archive, for example, when filling it and transferring the contents to removable media, for example, optical disks.

The script should be run by the cron service, the implementation of this is described in the section "Automatic shutdown of the system".

## About additional services

### AppArmor/SELinux

The use of mandatory access rights systems in the server under consideration seems redundant, since the main service provided by the system is file storage, and the processes associated with it are executed on behalf of root. At the same time, they interact only with the internal network, whose clients are equal, and access from the external network to them is closed. Auxiliary processes work on behalf of their own system users and their access to directories containing user data is discretionary.

### Password protection (fail2ban)

Since, due to the nature of the server's functioning, ssh is the only remote login mechanism, it is quite safe to refuse to use fail2ban. The system is well protected by a firewall (described below) with an identification (both by ip and mac address) of the only host for it from which ssh login is allowed. The ssh server of the system itself has the same restrictions, in addition, measures have been taken to protect against arp attacks (static entries), remote login itself is allowed only for an

unprivileged user whose account is password protected, and privilege escalation requires knowledge of the superuser password.

As for the file sharing service, access to it is possible only from the internal network interface. Taking into account the physical location of both the server and client devices, and network switches and accounting subnet cables inside one room, it can be argued that the protection of this service by fail2ban is redundant.

## Log rotation (logrotate)

System logs can serve as a means of attacking the server in order to fill its disk space and thereby paralyze its operation. This is countered by the log file rotation system.

When logging in via ssh, the corresponding entry is entered in the `/var/log/auth.log` file. Given the ssh protection measures taken, it can be argued that an attack is possible only from a single not less secure host from the network core (the system administrator's computer).

The other services available to clients (dnsmasq, samba and ntp) do not keep their own logs of client requests with the configuration specified for them and, therefore, cannot be the direction for such attack.

As a result, it should be noted that the default system settings (rotation weekly) are quite applicable here.

## Network time system (ntp)

The network time server not only provides services to clients, but also monitors the accuracy of the system clock of the machine on which it is deployed. Therefore, it is impossible and meaningless to work with other clock synchronization tools in the described system. In particular, the simplest ntp client, which is part of systemd, turns out to be inoperable after installing and configuring ntpd and must be disabled:

```
root@server:/# systemctl status systemd-timesyncd
• systemd-timesyncd.service - Network Time Synchronization
   Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled; vendor
   preset: enabled)
   Drop-In: /usr/lib/systemd/system/systemd-timesyncd.service.d
            └─disable-with-time-daemon.conf
   Active: inactive (dead)
   Docs: man:systemd-timesyncd.service(8)
root@server:/# systemctl list-unit-files | grep timesyncd
systemd-timesyncd.service                                enabled
root@server:/# systemctl disable systemd-timesyncd
Removed /etc/systemd/system/dbus-org.freedesktop.timesync1.service.
Removed /etc/systemd/system/sysinit.target.wants/systemd-timesyncd.service.
root@server:/# systemctl list-unit-files | grep timesync
systemd-timesyncd.service                                disabled
```

## Mail service

A mail service, even a local one, can become a target for an attack: for example, an attacker can generate a huge shaft of emails locally in the system, thereby provoking file system overflow and service failure. Therefore, even the local mail service should be carefully configured or disabled altogether if it is not used.

The Mail forwarding agent (MTA) exim4 is installed in the system under consideration along with other components. In the future, it is planned to put into operation the internal mail server of the organization, which will allow many system services (for example, the hard disk health monitoring system) to notify the system administrator about problems.

From a security point of view, the MTA can be used to overflow user mailboxes, i. e. fill the disk space of the machine.

At this stage, it makes sense to completely remove the MTA from the system, especially since this does not lead to the removal of other system components. So, the following fragment of the terminal session, abbreviated in insignificant output, demonstrates the procedure for checking the possibility of deleting and the deleting itself of the MTA:

```
root@server:/# dpkg -l | grep exim
ii  exim4-base      4.92-8+deb10u5 amd64 support files for all Exim MTA (v4) packages
ii  exim4-config    4.92-8+deb10u5 all configuration for the Exim MTA (v4)
ii  exim4-daemon-light 4.92-8+deb10u5 amd64 lightweight Exim MTA (v4) daemon
root@server:/# apt-get -s purge exim4-daemon-light
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  exim4-base exim4-config libevent-2.1-6 libgnutls-dane0 libunbound8
Для их удаления используйте «apt autoremove».
Следующие пакеты будут УДАЛЕНЫ:
  exim4-daemon-light*
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 1 пакетов, и
12 пакетов не обновлено.
Purg exim4-daemon-light [4.92-8+deb10u5]
root@server:/# apt-get purge exim4-daemon-light
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  exim4-base exim4-config libevent-2.1-6 libgnutls-dane0 libunbound8
Для их удаления используйте «apt autoremove».
Следующие пакеты будут УДАЛЕНЫ:
  exim4-daemon-light*
...
Вычищаются файлы настройки пакета exim4-daemon-light (4.92-8+deb10u5) ...
root@server:/# apt autoremove
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты будут УДАЛЕНЫ:
  exim4-base exim4-config libevent-2.1-6 libgnutls-dane0 libunbound8
...
```

Since some system services use the mail system in their work, you should make sure that their performance is not impaired. So after a successful reboot of the system (in the sense of no error messages about starting services), you should check their status. Below is an abbreviated response of only one such verification command for brevity, since the other commands gave a similar result:

```
root@server:/home/administrator# systemctl status smartd
• smartd.service - Self Monitoring and Reporting Technology (SMART) Daemon
  Loaded: loaded (/lib/systemd/system/smartd.service; enabled; vendor preset:
enabled)
  Active: active (running) since Thu 2021-06-03 10:00:15 +03; 2min 57s ago
```

```

...
root@server:/home/administrator# systemctl status networking
...
root@server:/home/administrator# systemctl status mdmonitor
...
root@server:/home/administrator# systemctl status nut-monitor
...
root@server:/home/administrator# systemctl status nut-server
...
root@server:/home/administrator# systemctl status nut-driver
...
root@server:/home/administrator# systemctl status ssh
...
root@server:/home/administrator# systemctl status ntp
...
root@server:/home/administrator# systemctl status cron
...
root@server:/home/administrator# systemctl status dnsmasq
...
root@server:/home/administrator# systemctl status smbd
...

```

In the future, after the deployment of the internal mail server of the organization, the MTA can again be installed in the system and configured accordingly. Meanwhile, in its absence, the smartd daemon will, if necessary, create email files in the /tmp directory.

## Automatic shutdown of the system

Automatic shutdown of the system during non-working hours is also performed by means of the cron daemon, for which the following lines are entered in its configuration file **/etc/crontab**:

```

#Shutting down at night with backup archive creation
00 21 * * * root /usr/local/sbin/server-shutdown

```

They ensure that the system shuts down at 21:00 with a notification to users 10 minutes before that. Obviously, shutting down the system at night has three goals:

- energy saving;
- saving the resource of the platform hardware;
- countering long-term attacks on password selection.

## Network filter (nftables) installation and configuration

### Network filter installation

To manage firewall rules in modern versions of Debian it is recommended to use the nftables package, which will need to be installed with the command

```

root@server:/# apt-get install nftables

```

### Network environment and potential threats

Building a set of network filter rules should begin with finding out all the protocols and ports used by the system and clients. The list of network ports and protocols on which the system is ready to accept connections can be obtained as follows:

```
root@server:/tmp# ss -tuln
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
udp	UNCONN	0	0	0.0.0.0:67	0.0.0.0:*
udp	UNCONN	0	0	10.9.1.129:123	0.0.0.0:*
udp	UNCONN	0	0	192.168.0.88:123	0.0.0.0:*
udp	UNCONN	0	0	127.0.0.1:123	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:123	0.0.0.0:*
udp	UNCONN	0	0	10.9.1.191:137	0.0.0.0:*
udp	UNCONN	0	0	10.9.1.129:137	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:137	0.0.0.0:*
udp	UNCONN	0	0	10.9.1.191:138	0.0.0.0:*
udp	UNCONN	0	0	10.9.1.129:138	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:138	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:46417	0.0.0.0:*
udp	UNCONN	0	0	127.0.0.1:53	0.0.0.0:*
udp	UNCONN	0	0	10.9.1.129:53	0.0.0.0:*
tcp	LISTEN	0	50	10.9.1.129:445	0.0.0.0:*
tcp	LISTEN	0	16	127.0.0.1:3493	0.0.0.0:*
tcp	LISTEN	0	50	10.9.1.129:139	0.0.0.0:*
tcp	LISTEN	0	32	127.0.0.1:53	0.0.0.0:*
tcp	LISTEN	0	32	10.9.1.129:53	0.0.0.0:*
tcp	LISTEN	0	128	192.168.0.88:22	0.0.0.0:*

A number of observations should be made:

First, the network filter should close port 67, operating over the UDP protocol, for all incoming packets via the external interface. This port is used by the DHCP protocol, which, according to the task statement, should serve only clients of the internal network. This port is opened by the dnsmasq daemon, which also performs such check, but additional firewall protection will obviously not be superfluous.

Secondly, it is required to close ports 137 (netbios-ns) and 138 (netbios-dgm) for the udp protocol for external interfaces. This port should remain available only for the loop and internal interfaces. These ports are used to ensure the operation of network discovery mechanisms by Windows clients who do not need to know the structure of the organization's network other than their own subnet.

Thirdly, it would not be superfluous to close external interfaces for new connections using the network time protocol (port 123), without relying only on the stability of the ntpd daemon itself. At the same time, the ability of the daemon itself to initiate connections with higher-level network time servers would not be affected.

Fourth, to reduce the probability of denial of service attacks, you should limit the maximum number of packets per time unit for each service.

Finally, it is necessary to ensure maximum protection of the ssh service. It is critical for the security of the system and establishing connections with it should bbe allowed only by checking the ip address, hardware address, interface from which the packet came and observing the limit on the number of such packets per unit of time.

The issue of packet forwarding should be considered separately. In order to avoid attempts to scan the structure of the entire organization's network from potentially vulnerable client computers, it is necessary to leave the possibility of transit of network packets only to the AIS "Avers" server (via tcp port 8082), to the virtual subnet of the organization's servers (without port restrictions at this stage) and to the Internet.

At the same time, it should be understood that the Internet generally means any network address other than those indicated above, including addresses from other local subnets, about which the accounting subnet server does not know anything at all. It would seem that an attacker, knowing the structure of the organization's network, will be able to send a packet to one of the open ports to some computer, for example, in administrative subnet. However, due to the fact that the accounting subnet server does not have a route to this subnet in its routing table, and also due to the fact that it is impossible to direct the packet to the administrative subnet server through its address in the network core, the main gateway will remain the only route, the firewall of which will stop sending a dangerous packet. In addition, the firewall is provided on the server of the attacked subnet too.

You should also pay attention to port 46417, opened by the domain name service for communication with higher-level servers. Such connection does not have a fixed port and can and most likely would change every time the service is started.

The rest of the listed ports are open only at valid addresses and the firewall settings should only confirm them.

In addition, the network filter must also provide rules for the operation of GRE tunnels.

Finally, at this stage, for organizational reasons, it is not possible to build an accurate list of external resources outside the organization's network that can be accessed from client computers, just like the corresponding lists of ports and protocols, so a default trust policy has been adopted for such connections.

## Network filter structure

Based on the above and guided by the principle of prohibition by default, we can characterize the structure of the network filter as follows:

- communication with external networks is carried out using the NAT mechanism,
- defined filtering tables for IPv4 and IPv6 protocols,
- the filtering table defines chains of rules for incoming, outgoing and transit packets,
- the default packet reset policy is set for all IPv6 table chains,
- a default packet reset policy is defined for the incoming and transit chains of the IPv4 table, a trust policy is set for the outgoing chain by default,
- reception and transit of packets related to already established connections are allowed,
- finally, permissive rules have been introduced for the traffic described above.

As a result, the scenario of atomic loading of rules has acquired the form:

```
root@server:/# cat /etc/nftables.conf
#!/usr/sbin/nft -f
```

```
flush ruleset
```

```
define IFACE_KERNEL = enp2s0
define IFACE_ACCD = enp3s0
define IFACE_USB = ethusb
define IFACE_EXT = { $IFACE_KERNEL, $IFACE_USB }
```

```
#FIXME: remove entries from CLIENTS and CLIENT_LLADDRS for
#dhcp-host=54:be:f7:2a:69:05,test,10.9.1.136
```

```
define CLIENT_LLADDRS = { 90:2b:34:a3:83:44, 50:e5:49:33:de:d9, 90:2b:34:a0:24:7b,
```

```

90:2b:34:96:16:53 }
define MFP_LLADDR = f8:0d:ac:78:8a:f1
define ACCD_LLADDRS = { $CLIENT_LLADDRS, $MFP_LLADDR }
define ADMIN_LLADDR = 90:2b:34:48:08:b5

define CLIENTS = { 10.9.1.131, 10.9.1.132, 10.9.1.133, 10.9.1.135, }
define MFP = 10.9.1.134
define ACCD_HOSTS = { $CLIENTS, $MFP }
define KRISTA_CLIENT = 10.9.1.133
define KRISTA_SERVER = 213.222.245.118
define ADMIN_HOST = 192.168.0.2
define AVERS_HOST = 192.168.0.4
define SERVICE_NET = 192.168.7.0/24
define GATEWAY = 192.168.0.1

define SSH_PORT = 22
define NTP_PORT = 123
define DHCP_PORT = 67
define DNS_PORT = 53
define SAMBA_PORTS = { 137, 138, 139, 445 }
define KRISTA_PORT = 1723
define AVERS_PORT = 8082

table ip nat {
    chain prerouting {
        type nat hook prerouting priority 0;
        policy accept;
        iifname $IFACE_EXT tcp dport $KRISTA_PORT dnat $KRISTA_CLIENT:$KRISTA_PORT
    }
    chain postrouting {
        type nat hook postrouting priority 0;
        policy accept;
        oifname $IFACE_EXT masquerade
    }
}

table ip filter {
    chain input {
        type filter hook input priority 0;
        policy drop;
        iif lo accept
        ct state established,related accept
        iifname $IFACE_EXT icmp type echo-request limit rate 10/second accept
        iif $IFACE_ACCD ether saddr $CLIENT_LLADDRS ip saddr $CLIENTS icmp
type echo-request limit rate 50/second accept
        iif $IFACE_ACCD ether saddr $ACCD_LLADDRS ip saddr $ACCD_HOSTS udp
dport $NTP_PORT limit rate 300/second accept
        iif $IFACE_ACCD ether saddr $ACCD_LLADDRS udp
dport $DHCP_PORT limit rate 300/second accept
        iif $IFACE_ACCD ether saddr $CLIENT_LLADDRS ip saddr $CLIENTS udp
dport $DNS_PORT limit rate 300/second accept
        iif $IFACE_ACCD ether saddr $CLIENT_LLADDRS ip saddr $CLIENTS tcp
dport $DNS_PORT limit rate 300/second accept
        iif $IFACE_ACCD ether saddr $CLIENT_LLADDRS ip saddr $CLIENTS udp
dport $SAMBA_PORTS limit rate 200 mbytes/second accept
        iif $IFACE_ACCD ether saddr $CLIENT_LLADDRS ip saddr $CLIENTS tcp
dport $SAMBA_PORTS limit rate 200 mbytes/second accept
        iif $IFACE_KERNEL ether saddr $ADMIN_LLADDR ip saddr $ADMIN_HOST tcp
dport $SSH_PORT ct state new limit rate 1/second accept
    }
    chain forward {
        type filter hook forward priority 0;

```



```

        policy drop;
        ct state established,related accept
        ip saddr $CLIENTS oif $IFACE_KERNEL ip daddr $SERVICE_NET accept
        ip saddr $CLIENTS oif $IFACE_KERNEL ip daddr $AVERS_HOST tcp dport
$AVERS_PORT accept
        ip saddr $CLIENTS oif $IFACE_KERNEL rt nexthop $GATEWAY accept
        ip saddr $CLIENTS oifname $IFACE_USB accept
        ip saddr $KRISTA_CLIENT tcp sport $KRISTA_PORT ip daddr $KRISTA_SERVER tcp
dport $KRISTA_PORT accept
        ip saddr $KRISTA_SERVER tcp sport $KRISTA_PORT ip daddr $KRISTA_CLIENT tcp
dport $KRISTA_PORT accept
    }
    chain output {
        type filter hook output priority 0;
        policy accept;
    }
}

table ip6 filter {
    chain input {
        type filter hook input priority 0;
        policy drop;
    }
    chain forward {
        type filter hook forward priority 0;
        policy drop;
    }
    chain output {
        type filter hook output priority 0;
        policy drop;
    }
}

```

## Packet forwarding and tunnel support

First of all, it is necessary to ensure the forwarding of network packets between the interfaces of the server in question. To do this, you should bring the configuration file **/etc/sysctl.conf** to such a form that it contains the line

```
net.ipv4.ip_forward=1
```

Usually it's enough to just uncomment it. To enable packet forwarding in the current session without restarting the system, it is enough to run the command:

```
root@server:/# echo 1 > /proc/sys/net/ipv4/ip_forward
```

You will also need to add the following lines to the **/etc/sysctl.conf** file:

```
#Enabling netfilter conntrack helper
net.netfilter.nf_conntrack_helper=1
```

This connects the connection tracking mechanism necessary for the implementation of the GRE tunnel. For the current session (without restarting), this mechanism can be enabled with a command like:

```
root@server:/# echo 1 > /proc/sys/net/netfilter/nf_conntrack_helper
```

## Starting and checking the network filter

Automatic loading of firewall rules is provided by the nftables package installer and does not require manual intervention. You can check the current set of rules with the command

```
root@debian:/# /sbin/nft list ruleset
```

The output of this command, up to the comments, pre-cleaning instructions and substitution of the list of allowed ports, must coincide with the above scenario for loading firewall rules.

To make sure that the rules were loaded before the network interfaces were launched, you can view the contents of the `/var/log/daemon.log` file, an excerpt from which, with some abbreviations, but preserving the order of the presented ones, has the form:

```
Dec 10 14:05:41 server systemd[1]: Started udev Kernel Device Manager.
Dec 10 14:05:41 server systemd[1]: Started nftables.
Dec 10 14:05:41 server systemd[1]: Reached target Network (Pre).
...
Dec 10 14:05:41 server systemd[1]: Starting Raise network interfaces...
...
Dec 10 14:05:42 server systemd[1]: Started Raise network interfaces.
Dec 10 14:05:42 server systemd[1]: Reached target Network.
...
Dec 10 14:05:42 server systemd[1]: Reached target Network is Online.
```

You can also view the status of the service with the command:

```
root@server:/# systemctl status nftables
```

```
• nftables.service - nftables
  Loaded: loaded (/lib/systemd/system/nftables.service; enabled; vendor preset:
enabled)
  Active: active (exited) since Fri 2021-12-10 14:07:20 +03; 10s ago
    Docs: man:nft(8)
          http://wiki.nftables.org
  Process: 561 ExecStart=/usr/sbin/nft -f /etc/nftables.conf (code=exited,
status=0/SUCCESS)
 Main PID: 561 (code=exited, status=0/SUCCESS)
```

```
дек 10 14:07:20 server systemd[1]: Starting nftables...
дек 10 14:07:20 server systemd[1]: Started nftables.
```

The firewall check can be divided into a trivial check of the availability of authorized services to the client, including those located outside the accounting subnet, and a check by the nmap network scanner of the availability of certain server ports from four different directions:

- from the system administrator's computer
- from any computer from the core of the network
- from the client computer from the accounting subnet
- from an intruder computer artificially introduced into the accounting subnet

Testing is automated using a script `/tmp/nmap.sh`, the contents of which are transparently guessed from its output, since the executed commands are displayed with the prefix `user@host`.

Scan protocol from the system administrator's computer:

```
root@admin:/tmp# chmod a+x nmap.sh
root@admin:/tmp# ./nmap.sh
Пинг-сканирование
user@host:/# nmap -sP 192.168.0.88
```

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:31 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00021s latency).  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)  
Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds

#### **TCP SYN пинг**

**user@host:/# nmap -PS 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:31 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00020s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp open ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 7.75 seconds

#### **TCP ACK пинг**

**user@host:/# nmap -PA 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:31 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00021s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp open ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 12.51 seconds

#### **UDP пинг**

**user@host:/# nmap -PU 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:32 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00019s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp open ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 8.34 seconds

#### **Различные типы пинг-пакетов ICMP**

**user@host:/# nmap -PE 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:32 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00020s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp open ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 7.74 seconds

**user@host:/# nmap -PP 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:32 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.0011s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp open ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 12.51 seconds

**user@host:/# nmap -PM 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:32 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)

Host is up (0.00020s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp open ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 12.32 seconds

#### **Пинг с использование протокола IP**

**user@host:/# nmap -PO 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:32 MSK

Nmap scan report for server.account.school34 (192.168.0.88)

Host is up (0.00019s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 10.37 seconds

#### **ARP пинг**

**user@host:/# nmap -PR 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:32 MSK

Nmap scan report for server.account.school34 (192.168.0.88)

Host is up (0.00020s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 12.72 seconds

#### **TCP connect сканирование**

**user@host:/# nmap -sT 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:33 MSK

Nmap scan report for server.account.school34 (192.168.0.88)

Host is up (0.00023s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 4.68 seconds

#### **UDP сканирование**

**user@host:/# nmap -sU 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:33 MSK

Nmap scan report for server.account.school34 (192.168.0.88)

Host is up (0.00017s latency).

All 1000 scanned ports on server.account.school34 (192.168.0.88) are open|filtered

MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.46 seconds

#### **TCP NULL сканирование**

**user@host:/# nmap -sN 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:33 MSK

Nmap scan report for server.account.school34 (192.168.0.88)

Host is up (0.00019s latency).

All 1000 scanned ports on server.account.school34 (192.168.0.88) are open|filtered

MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.45 seconds

#### **TCP FIN сканирование**

**user@host:/# nmap -sF 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:33 MSK

Nmap scan report for server.account.school34 (192.168.0.88)

Host is up (0.00019s latency).  
All 1000 scanned ports on server.account.school34 (192.168.0.88) are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.44 seconds  
**TCP Xmas сканирование**  
**user@host:/# nmap -sX 192.168.0.88**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:34 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00017s latency).  
All 1000 scanned ports on server.account.school34 (192.168.0.88) are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.44 seconds  
**TCP ACK сканирование**  
**user@host:/# nmap -sA 192.168.0.88**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:34 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00019s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp unfiltered ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 12.23 seconds  
**TCP Window сканирование**  
**user@host:/# nmap -sW 192.168.0.88**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:34 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00020s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
22/tcp closed ssh  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 12.73 seconds  
**TCP сканирование Мэймона**  
**user@host:/# nmap -sM 192.168.0.88**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:35 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00018s latency).  
All 1000 scanned ports on server.account.school34 (192.168.0.88) are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.45 seconds  
**Сканирование протокола IP**  
**user@host:/# nmap -sO 192.168.0.88**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:35 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00015s latency).  
Not shown: 255 open|filtered protocols  
PROTOCOL STATE SERVICE  
1 open icmp  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 5.08 seconds  
**Определение версий ОС и служб в сочетании с TCP SYN сканирование**  
**user@host:/# nmap -O -sV -sS 192.168.0.88**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:35 MSK  
Nmap scan report for server.account.school34 (192.168.0.88)  
Host is up (0.00021s latency).  
Not shown: 999 filtered ports

```
PORT    STATE SERVICE VERSION
22/tcp  open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.18 - 2.6.22
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.43 seconds
```

The protocol of scanning from a network core computer (namely from the virtualization server):

#### **Пинг-сканирование**

```
user@host:/# nmap -sP 192.168.0.88
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2021-12-13 14:52 MSK
```

```
Nmap scan report for 192.168.0.88
```

```
Host is up (0.00016s latency).
```

```
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

#### **TCP SYN пинг**

```
user@host:/# nmap -PS 192.168.0.88
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2021-12-13 14:52 MSK
```

```
Nmap scan report for 192.168.0.88
```

```
Host is up (0.00017s latency).
```

```
All 1000 scanned ports on 192.168.0.88 are filtered
```

```
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)
```

```
Nmap done: 1 IP address (1 host up) scanned in 21.51 seconds
```

#### **TCP ACK пинг**

```
user@host:/# nmap -PA 192.168.0.88
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2021-12-13 14:52 MSK
```

```
Nmap scan report for 192.168.0.88
```

```
Host is up (0.00019s latency).
```

```
All 1000 scanned ports on 192.168.0.88 are filtered
```

```
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)
```

```
Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds
```

#### **UDP пинг**

```
user@host:/# nmap -PU 192.168.0.88
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2021-12-13 14:52 MSK
```

```
Nmap scan report for 192.168.0.88
```

```
Host is up (0.00014s latency).
```

```
All 1000 scanned ports on 192.168.0.88 are filtered
```

```
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)
```

```
Nmap done: 1 IP address (1 host up) scanned in 21.48 seconds
```

#### **Различные типы пинг-пакетов ICMP**

```
user@host:/# nmap -PE 192.168.0.88
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2021-12-13 14:53 MSK
```

```
Nmap scan report for 192.168.0.88
```

```
Host is up (0.00016s latency).
```

```
All 1000 scanned ports on 192.168.0.88 are filtered
```

```
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)
```

```
Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds
```

```
user@host:/# nmap -PP 192.168.0.88
```

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:53 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00013s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds

**user@host:/# nmap -PM 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:53 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00018s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds

**Пинг с использование протокола IP**

**user@host:/# nmap -PO 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:54 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.48 seconds

**ARP пинг**

**user@host:/# nmap -PR 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:54 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00016s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.48 seconds

**TCP connect сканирование**

**user@host:/# nmap -sT 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:54 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.39 seconds

**UDP сканирование**

**user@host:/# nmap -sU 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:55 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00016s latency).  
All 1000 scanned ports on 192.168.0.88 are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.48 seconds

**TCP NULL сканирование**

**user@host:/# nmap -sN 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:55 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00016s latency).  
All 1000 scanned ports on 192.168.0.88 are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds

**TCP FIN сканирование**

**user@host:/# nmap -sF 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:56 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
All 1000 scanned ports on 192.168.0.88 are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.48 seconds

#### **TCP Xmas сканирование**

**user@host:/# nmap -sX 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:56 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
All 1000 scanned ports on 192.168.0.88 are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds

#### **TCP ACK сканирование**

**user@host:/# nmap -sA 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:56 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds

#### **TCP Window сканирование**

**user@host:/# nmap -sW 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:57 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00023s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.47 seconds

#### **TCP сканирование Мэймона**

**user@host:/# nmap -sM 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:57 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
All 1000 scanned ports on 192.168.0.88 are open|filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 21.51 seconds

#### **Сканирование протокола IP**

**user@host:/# nmap -sO 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:57 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
Not shown: 255 open|filtered protocols  
PROTOCOL STATE SERVICE  
1 open icmp  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 3.73 seconds

#### **Определение версий ОС и служб в сочетании с TCP SYN сканирование**

**user@host:/# nmap -O -sV -sS 192.168.0.88**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2021-12-13 14:57 MSK  
Nmap scan report for 192.168.0.88  
Host is up (0.00017s latency).  
All 1000 scanned ports on 192.168.0.88 are filtered  
MAC Address: 18:D6:C7:00:EA:6C (Tp-link Technologies)  
Too many fingerprints match this host to give specific OS details



Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 25.12 seconds

Scan protocol from the client computer:

#### **Пинг-сканирование**

**user@host:/# nmap -sP 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:04 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00027s latency).

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.63 seconds

#### **TCP SYN пинг**

**user@host:/# nmap -PS 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:04 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00022s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT	STATE	SERVICE
53/tcp	open	domain
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 5.25 seconds

#### **TCP ACK пинг**

**user@host:/# nmap -PA 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:04 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00024s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT	STATE	SERVICE
53/tcp	open	domain
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 5.16 seconds

#### **UDP пинг**

**user@host:/# nmap -PU 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00028s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT	STATE	SERVICE
53/tcp	open	domain
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.66 seconds

#### **Различные типы пинг-пакетов ICMP**

**user@host:/# nmap -PE 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00023s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT	STATE	SERVICE
53/tcp	open	domain

139/tcp open netbios-ssn  
445/tcp open microsoft-ds  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 5.15 seconds

**user@host:/# nmap -PP 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00025s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT STATE SERVICE

53/tcp open domain

139/tcp open netbios-ssn

445/tcp open microsoft-ds

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 5.16 seconds

**user@host:/# nmap -PM 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00029s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT STATE SERVICE

53/tcp open domain

139/tcp open netbios-ssn

445/tcp open microsoft-ds

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 5.18 seconds

**Пинг с использование протокола IP**

**user@host:/# nmap -PO 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00020s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT STATE SERVICE

53/tcp open domain

139/tcp open netbios-ssn

445/tcp open microsoft-ds

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.66 seconds

**ARP пинг**

**user@host:/# nmap -PR 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00019s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT STATE SERVICE

53/tcp open domain

139/tcp open netbios-ssn

445/tcp open microsoft-ds

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.58 seconds

**TCP connect сканирование**

**user@host:/# nmap -sT 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00035s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT STATE SERVICE

53/tcp open domain  
139/tcp open netbios-ssn  
445/tcp open microsoft-ds  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.57 seconds

#### UDP сканирование

**user@host:/# nmap -sU 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00030s latency).

Not shown: 994 open|filtered udp ports (no-response)

PORT	STATE	SERVICE
------	-------	---------

53/udp	open	domain
--------	------	--------

67/udp	open	dhcps
--------	------	-------

123/udp	open	ntp
---------	------	-----

137/udp	open	netbios-ns
---------	------	------------

139/udp	closed	netbios-ssn
---------	--------	-------------

445/udp	closed	microsoft-ds
---------	--------	--------------

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 6.08 seconds

#### TCP NULL сканирование

**user@host:/# nmap -sN 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:05 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00019s latency).

All 1000 scanned ports on ntp.account.school34 (10.9.1.129) are in ignored states.

Not shown: 1000 open|filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 21.29 seconds

#### TCP FIN сканирование

**user@host:/# nmap -sF 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:06 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00019s latency).

All 1000 scanned ports on ntp.account.school34 (10.9.1.129) are in ignored states.

Not shown: 1000 open|filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 21.29 seconds

#### TCP Xmas сканирование

**user@host:/# nmap -sX 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:06 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00020s latency).

All 1000 scanned ports on ntp.account.school34 (10.9.1.129) are in ignored states.

Not shown: 1000 open|filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 21.31 seconds

#### TCP ACK сканирование

**user@host:/# nmap -sA 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:06 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00028s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT	STATE	SERVICE
------	-------	---------

53/tcp	unfiltered	domain
--------	------------	--------

139/tcp	unfiltered	netbios-ssn
---------	------------	-------------

445/tcp	unfiltered	microsoft-ds
---------	------------	--------------

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.95 seconds

#### **TCP Window сканирование**

**user@host:/# nmap -sW 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:06 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00023s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT	STATE	SERVICE
------	-------	---------

53/tcp	closed	domain
--------	--------	--------

139/tcp	closed	netbios-ssn
---------	--------	-------------

445/tcp	closed	microsoft-ds
---------	--------	--------------

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.29 seconds

#### **TCP сканирование Мэймона**

**user@host:/# nmap -sM 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:06 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00019s latency).

Not shown: 997 open|filtered tcp ports (no-response)

PORT	STATE	SERVICE
------	-------	---------

53/tcp	closed	domain
--------	--------	--------

139/tcp	closed	netbios-ssn
---------	--------	-------------

445/tcp	closed	microsoft-ds
---------	--------	--------------

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.77 seconds

#### **Сканирование протокола IP**

**user@host:/# nmap -sO 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:07 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00024s latency).

Not shown: 255 open|filtered n/a protocols (no-response)

PROTOCOL	STATE	SERVICE
----------	-------	---------

1	open	icmp
---	------	------

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 5.21 seconds

#### **Определение версий ОС и служб в сочетании с TCP SYN сканирование**

**user@host:/# nmap -O -sV -sS 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:07 UTC

Nmap scan report for ntp.account.school34 (10.9.1.129)

Host is up (0.00051s latency).

Not shown: 997 filtered tcp ports (no-response)

PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

53/tcp	open	domain	dnsmasq 2.80
--------	------	--------	--------------

139/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: ACCOUNT)
---------	------	-------------	---

445/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: ACCOUNT)
---------	------	-------------	---

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Linux 4.X|5.X

OS CPE: cpe:/o:linux:linux\_kernel:4 cpe:/o:linux:linux\_kernel:5

OS details: Linux 4.15 - 5.6, Linux 5.0 - 5.4

Network Distance: 1 hop

Service Info: Host: SAMBA

OS and Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 24.63 seconds

Protocol of scanning from the offending computer from the internal network:

#### **Пинг-сканирование**

**user@host:/# nmap -sP 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:11 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00018s latency).

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 13.14 seconds

#### **TCP SYN пинг**

**user@host:/# nmap -PS 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:11 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00015s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.31 seconds

#### **TCP ACK пинг**

**user@host:/# nmap -PA 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:12 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00016s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.30 seconds

#### **UDP пинг**

**user@host:/# nmap -PU 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:13 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00025s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.30 seconds

#### **Различные типы пинг-пакетов ICMP**

**user@host:/# nmap -PE 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:13 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00016s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.29 seconds

**user@host:/# nmap -PP 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:14 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00018s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.30 seconds

**user@host:/# nmap -PM 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:14 UTC

Nmap scan report for 10.9.1.129  
Host is up (0.00028s latency).  
All 1000 scanned ports on 10.9.1.129 are in ignored states.  
Not shown: 1000 filtered tcp ports (no-response)  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.30 seconds  
пинг с использование протокола IP  
**user@host:/# nmap -PO 10.9.1.129**  
Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:15 UTC  
Nmap scan report for 10.9.1.129  
Host is up (0.00017s latency).  
All 1000 scanned ports on 10.9.1.129 are in ignored states.  
Not shown: 1000 filtered tcp ports (no-response)  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.31 seconds  
**ARP пинг**  
**user@host:/# nmap -PR 10.9.1.129**  
Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:15 UTC  
Nmap scan report for 10.9.1.129  
Host is up (0.00017s latency).  
All 1000 scanned ports on 10.9.1.129 are in ignored states.  
Not shown: 1000 filtered tcp ports (no-response)  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.31 seconds  
**TCP connect сканирование**  
**user@host:/# nmap -sT 10.9.1.129**  
Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:16 UTC  
Nmap scan report for 10.9.1.129  
Host is up (0.00018s latency).  
All 1000 scanned ports on 10.9.1.129 are in ignored states.  
Not shown: 1000 filtered tcp ports (no-response)  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.21 seconds  
**UDP сканирование**  
**user@host:/# nmap -sU 10.9.1.129**  
Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:17 UTC  
Nmap scan report for 10.9.1.129  
Host is up (0.00028s latency).  
All 1000 scanned ports on 10.9.1.129 are in ignored states.  
Not shown: 1000 open|filtered udp ports (no-response)  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.34 seconds  
**TCP NULL сканирование**  
**user@host:/# nmap -sN 10.9.1.129**  
Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:17 UTC  
Nmap scan report for 10.9.1.129  
Host is up (0.00019s latency).  
All 1000 scanned ports on 10.9.1.129 are in ignored states.  
Not shown: 1000 open|filtered tcp ports (no-response)  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.28 seconds  
**TCP FIN сканирование**  
**user@host:/# nmap -sF 10.9.1.129**  
Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:18 UTC  
Nmap scan report for 10.9.1.129  
Host is up (0.00023s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.  
Not shown: 1000 open|filtered tcp ports (no-response)  
MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.30 seconds

#### **TCP Xmas сканирование**

**user@host:/# nmap -sX 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:18 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00018s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 open|filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.32 seconds

#### **TCP ACK сканирование**

**user@host:/# nmap -sA 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:19 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00028s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.28 seconds

#### **TCP Window сканирование**

**user@host:/# nmap -sW 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:19 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00017s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.30 seconds

#### **TCP сканирование Мэймона**

**user@host:/# nmap -sM 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:20 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00019s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 open|filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 34.31 seconds

#### **Сканирование протокола IP**

**user@host:/# nmap -sO 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:21 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00019s latency).

All 256 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 256 open|filtered n/a protocols (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 19.45 seconds

#### **Определение версий ОС и служб в сочетании с TCP SYN сканирование**

**user@host:/# nmap -O -sV -sS 10.9.1.129**

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-12-13 16:21 UTC

Nmap scan report for 10.9.1.129

Host is up (0.00016s latency).

All 1000 scanned ports on 10.9.1.129 are in ignored states.

Not shown: 1000 filtered tcp ports (no-response)

MAC Address: 1C:6F:65:C2:C5:6F (Giga-byte Technology)  
Too many fingerprints match this host to give specific OS details  
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at  
<https://nmap.org/submit/> .  
Nmap done: 1 IP address (1 host up) scanned in 37.53 seconds

Analyzing the network scanner verification protocols, we can draw the following conclusions:

When scanning the system from the system administrator's computer, only the ssh service port was available, which was to be expected. In addition, it turned out to be possible to check the availability of the system using the ICMP protocol and obtain its hardware address, which also does not contradict the task statement and the expected results. When determining software versions, the ssh service not only allowed you to determine your its own version, but also reported the version of the operating system. However, this should not be considered a disadvantage, since when checking the system from an outside computer, it was not possible to obtain this information from the network core.

When scanning the system by the subnet client, only ports of services allowed for use by clients were detected, the remote administration service (ssh) was not detected. When determining the software versions, it was found out that the DNS service was provided by dnsmasq version 2.80, running Linux OS with kernel version 4.15-5.6. The Samba service (versions 3.X – 4.X) and the ACCOUNT workgroup were also discovered. Such results should also be considered fully compliant with the requirements.

When scanning from an intruder computer artificially introduced into the accounting network with a manually assigned IP address, it was possible to find out only the link layer address of the server's internal interface and the fact that the server was working at the time of testing. No information about running services, software versions or operating system was received.

In addition to protecting the accounting subnet server itself, the firewall should also increase the security of the clients of this network. So an attempt to find a route from the virtualization server to one of the client devices with the address 10.9.1.134 failed, as did an attempt to detect this client by the ping utility:

```
root@virtserver:/# traceroute 10.9.1.134
traceroute to 10.9.1.134 (10.9.1.134), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  0.547 ms  0.501 ms  0.468 ms
 2  server.account.school34 (192.168.0.88)  0.415 ms  0.384 ms  0.359 ms
 3  * * *
...
30 * * *
root@virtserver:/# ping 10.9.1.134
PING 10.9.1.134 (10.9.1.134) 56(84) bytes of data.
From 192.168.0.1: icmp_seq=2 Redirect Host(New nexthop: 192.168.0.88)
...
^C
--- 10.9.1.134 ping statistics ---
21 packets transmitted, 0 received, 100% packet loss, time 482ms
```

*Note:* here the name of the virtualization server (server, server.service.school34) in the command prompt has been changed to avoid confusion with the name of the accounting subnet server (server, server.account.school34) to virtserver.



At the same time, the client has free access to the virtual servers located on this virtualization server and does not have access to devices in other subnets.

## Completing the installation

At the end of the installation, it makes sense to update the system, clear the local package cache and delete packages that the system no longer needs. This can be done by commands:

```
root@server:/# apt-get upgrade
root@server:/# apt-get autoremove
root@server:/# apt-get clean
```

## Server usage and maintenance

The scenarios, commands and techniques described below are intended to be used during the operation of the system to monitor its condition or change the modes of operation. The following description can be taken as a short guide (how to).

### Login procedure

Due to the ssh and network filter settings described above, the following procedure for remote login is provided: log in from the system administrator's computer as administrator, and then, if necessary, upgrade privileges to root locally. These actions are demonstrated in the following fragment of the terminal session:

```
administrator@admin:~$ ssh administrator@server.account.school34
administrator@server.account.school34's password:
Last login: Thu Oct 21 10:58:58 2021 from 192.168.0.2
```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

```
administrator@server:~$ su
Password:
root@server:/home/administrator#
```

### When updating the system

Since some measures to protect the loader were taken, including manual editing of the final configuration file **/boot/grub/grub.cfg**, which is used directly by the loader, it becomes necessary to re-perform such manual edits, including when updating the system, if during this update the specified configuration file was also reassembled. Usually such reassembly is performed by the `update-grub` command. If manual edits are not performed, it will be impossible to boot the system without entering a password for GRUB2.

To make boot possible, you just need to add the `--unrestricted` key to the GRUB2 menu entry used by default, which would lead this line to be something like:

```
menuentry 'Debian GNU/Linux' --unrestricted --class debian --class gnu-linux ...
```

## Hardware status monitoring

To get information from hardware sensors, it is enough to execute the sensors command on behalf of any user, for example:

```
administrator@server:/# sensors
coretemp-isa-0000
Adapter: ISA adapter
Core 0:      +35.0°C  (high = +76.0°C, crit = +100.0°C)
Core 1:      +36.0°C  (high = +76.0°C, crit = +100.0°C)
...
```

Here the output of the command is significantly shortened for brevity.

To get information from the built-in self-diagnosis equipment of the hard disk, you can use the following commands executed with superuser rights:

Search for all S.M.A.R.T-compatible devices:

```
root@server:/# /sbin/smartctl --scan
/dev/sda -d scsi # /dev/sda, SCSI device
/dev/sdb -d scsi # /dev/sdb, SCSI device
/dev/sdc -d scsi # /dev/sdc, SCSI device
```

Getting device information:

```
root@server:/# /sbin/smartctl /dev/sda -i
smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.19.0-16-amd64] (local build)
Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org
```

```
=== START OF INFORMATION SECTION ===
Model Family:      Western Digital Caviar Blue (SATA)
Device Model:      WDC WD3200AAKS-00B3A0
Serial Number:     WD-WMAT10439126
LU WWN Device Id:  5 0014ee 055cb8cb4
Firmware Version:  01.03A01
User Capacity:     320 071 851 520 bytes [320 GB]
Sector Size:       512 bytes logical/physical
Device is:         In smartctl database [for details use: -P show]
ATA Version is:    ATA8-ACS (minor revision not indicated)
SATA Version is:   SATA 2.5, 3.0 Gb/s
Local Time is:     Thu May 27 11:24:43 2021 +03
SMART support is:  Available - device has SMART capability.
SMART support is:  Enabled
```

Getting detailed related to S.M.A.R.T information about the device (the volume output of the command is omitted for brevity):

```
root@server:/# /sbin/smartctl /dev/sda -a
```

Getting even more detailed S.M.A.R.T related information about the device and not only (an even more voluminous output of the command is also omitted for brevity):

```
root@server:/# /sbin/smartctl /dev/sda -x
```

## Information about network adapters

During the server work, it may be necessary to obtain information about the operation of its network interfaces. The following commands may be useful, for example:

```

root@server:/# /sbin/ethtool enp2s0
root@server:/# /sbin/ethtool -i enp2s0
root@server:/# /sbin/ethtool -S enp2s0
root@server:/# /sbin/ethtool -p enp2s0 10

```

The first command allows you to get detailed information about the supported and current operating modes of the network adapter, the second displays information about the driver it uses, the third displays detailed interface statistics, and the last allows the LED of the network card to flash for 10 seconds (in this example), which is useful for matching the physical port of the server with its name in the system.

The difficulty lies in the fact that not all network adapters support these functions. So both interfaces of the system do not support LED indication, and the external card does not support statistics collection either.

However, the following fragment of the terminal session, given with some insignificant abbreviations, demonstrates that both cards are active and operate in full-duplex mode with a bandwidth of 1 GB/s:

```

root@server:/home/administrator# /sbin/ethtool enp2s0
Settings for enp2s0:
    Supported ports: [ TP MII ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Half 1000baseT/Full
    Supported pause frame use: Symmetric Receive-only
    Supports auto-negotiation: Yes
    Supported FEC modes: Not reported
    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Half 1000baseT/Full
    Advertised pause frame use: Symmetric Receive-only
    Advertised auto-negotiation: Yes
    Advertised FEC modes: Not reported
    Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                         100baseT/Half 100baseT/Full
                                         1000baseT/Full
    Link partner advertised pause frame use: Symmetric
    Link partner advertised auto-negotiation: Yes
    Link partner advertised FEC modes: Not reported
    Speed: 1000Mb/s
    Duplex: Full
    Port: MII
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: on
    Supports Wake-on: pumbg
    Wake-on: g
    Current message level: 0x00000033 (51)
                           drv probe ifdown ifup

    Link detected: yes
root@server:/home/administrator# /sbin/ethtool -i enp2s0
driver: r8169
version:
firmware-version: rtl_nic/rtl8168e-2.fw
expansion-rom-version:
bus-info: 0000:02:00.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no

```

```

supports-register-dump: yes
supports-priv-flags: no
root@server:/home/administrator# /sbin/ethtool -S enp2s0
NIC statistics:
  tx_packets: 2219
  rx_packets: 4690
  tx_errors: 0
  rx_errors: 0
  rx_missed: 0
  align_errors: 0
  tx_single_collisions: 0
  tx_multi_collisions: 0
  unicast: 2373
  broadcast: 2317
  multicast: 0
  tx_aborted: 0
  tx_underrun: 0
root@server:/home/administrator# /sbin/ethtool enp3s0
Settings for enp3s0:
  Supported ports: [ TP ]
  Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full

  Supported pause frame use: No
  Supports auto-negotiation: Yes
  Supported FEC modes: Not reported
  Advertised link modes:  Not reported
  Advertised pause frame use: No
  Advertised auto-negotiation: Yes
  Advertised FEC modes: Not reported
  Speed: 1000Mb/s
  Duplex: Full
  Port: Twisted Pair
  PHYAD: 0
  Transceiver: internal
  Auto-negotiation: on
  MDI-X: Unknown
  Supports Wake-on: pg
  Wake-on: d
  Current message level: 0x0000003f (63)
                        drv probe link timer ifdown ifup

  Link detected: yes
root@server:/home/administrator# /sbin/ethtool -i enp3s0
driver: atl1c
version: 1.0.1.1-NAPI
firmware-version:
expansion-rom-version:
bus-info: 0000:03:00.0
supports-statistics: no
supports-test: no
supports-eeprom-access: yes
supports-register-dump: yes
supports-priv-flags: no
root@server:/home/administrator# /sbin/ethtool -S enp3s0
no stats available
root@server:/home/administrator# /sbin/ethtool -p enp2s0 10
Cannot identify NIC: Operation not supported
root@server:/home/administrator# /sbin/ethtool -p enp3s0 10
Cannot identify NIC: Operation not supported

```

## Connecting clients

Finally, it is necessary to make a note about connecting client devices: only the client device itself determines the set of services that it is ready to use. The server can only offer these services. For example, the DHCP client of Windows devices does not use the hostname received from the server. Other devices may well ignore both the domain name and the proposed network time and domain name servers. In most cases, this does not pose any difficulties for the network itself. The only exceptions are attempts to bypass filtering by domain names or attempts to use third-party proxy servers, which, however, is opposed by the firewall.

Also, the functionality of the server does not include the ability to provide customers with information about the services of other servers on the local network. Thus, shortcuts on desktops, bookmarks in Internet browsers, etc. remain at the discretion of the customers themselves.

### Connecting clients to a network MFP

To connect clients to a network MFP, first of all, connect the client workstation and the network MFP to the network, and then install software for a multi-functional device downloaded from the manufacturer's official website on the client workstation. Then the MFP would be detected on the client computer automatically, at least for the HP LaserJet Pro MFP M428fdn considered in the current context. Separately, it should be noted that by default, scanning to a computer on this device is disabled. To enable it, you need to log in to the device's web interface at the following URL:

`https://mfp.account.school34`

and by going there to the "Settings" → "Security" → "Administrator Settings" section, set the "Scan from a computer or from a mobile device" flag.

This URL can also be used to create a shortcut on the accounting computers desktops.

Special attention should be paid to the possibility of installing your own SSL certificate for the MFP, certified by the local certification center of the organization. To do this, create a signature request and send it to the local control center. Then, after receiving the certificate, it should be installed in the MFP. At the same time, the private key and the corresponding signature request can be created both on the MFP itself and outside it. The second method in the considered network is more preferable, because the organization has its own certification center with centralized generation and storage of private keys, signature requests and certificates. In this case, all operations to create a private key, CSR and the corresponding certificate are performed in the local control center, and a ready-made certificate and private key are installed on the MFP. This is done via the device's web interface. In this case, the certificate and the private key must be presented in the form of a single encrypted file in PKCS#12 format.

Of course, the root certificate of the local certificate authority must be installed on the clients (in this case, on accounting computers). It should be remembered that some programs use the system certificate store, and some maintain their own, for example, the Mozilla Firefox browser.

### Connecting clients to a file server

It is most convenient to connect clients to the file server using shortcuts on the Desktop of client machines. As an alternative, you can consider connecting directories on a permanent basis as network drives. However, in this case, as the practice of previous operation of samba servers has

shown, the shared and backup network directories will not only be defenseless against an encryption virus that has penetrated any of the client workstations, but will also be easily detected by them. Although the use of shortcuts cannot guarantee anything, in practice it has repeatedly "protected" the network directory from a compromised workstation. In the system under consideration, shortcuts should have the following addresses of objects (in Windows notation):

```
\\samba.account.school34\shared
\\samba.account.school34\backup
\\samba.account.school34\archive
```

Also, for quick access to the index file, it is appropriate to create a separate shortcut for it with the address of the object:

```
file://///samba.account.school34/archive/index.html
```

This shortcut can be created not only in addition to the archive shortcut, but also instead of it, but this is already a matter of user comfort and no more.

As for the inclusion of workstations in the ACCOUNT group, this issue should be left to the discretion of the system administrator who maintains accounting workstations and the software installed on them. Network directories will remain accessible in any case.

### **Replacement of the accounting network node**

While replacing a workstation or a network MFP in the accounting subnet you should make the following edits on the server:

First, the client's link layer address (MAC address, lladdr) should be replaced in the **/etc/network/****neighbours** file.

Secondly, you should also replace this address in the corresponding dhcp-host... line in the **/etc/dnsmasq.conf** file.

Thirdly, the firewall configuration file **/etc/nftables** should be updated. In particular, it is required to replace the link and network layers addresses of the outgoing node with the corresponding addresses of the newly installed in the variables CLIENT\_LADDRS, CLIENTS and, possibly, MFP\_LLADDR and MFP (if the MFP is being replaced) or KRISTA\_CLIENT (if this particular workstation is being replaced).

Then it remains only to restart the services and make sure they are working properly:

```
root@server:/tmp# systemctl restart nftables
root@server:/tmp# systemctl restart networking
root@server:/tmp# systemctl restart dnsmasq
root@server:/tmp# systemctl status networking
...
root@server:/tmp# systemctl status dnsmasq
...
```

Of course, then you should configure the modified node: create network shortcuts, install root certificates, configure the connection to the MFP or install its certificate on a new MFP. All this should be done in accordance with the above instructions for the initial configuration of similar network nodes.

## Replacing an accounting employee's smartphone

The following operations should be performed in the case of an accounting employee smartphone replacement:

First, get the necessary information about the new smartphone with a series of commands of the form (up to the address of the phone on the USB bus):

```
root@server:/# lsusb
root@server:/# udevadm info -a /dev/bus/usb/005/005
root@server:/# udevadm info -q all -n /dev/bus/usb/005/005
```

Then, based on the received data, you should make edits to the files **/etc/udev/rules.d/35-usb-filter.rules** and **/etc/udev/rules.d/77-net-alias.rules**.

Next, you need to make the udev daemon to reload the rules using the command:

```
root@server:/# udevadm control --reload-rules
```

Finally, you should make changes to the initial RAM disk so that the smartphone connection is processed correctly even at the system boot stage. This can be done by commands:

```
root@server:/# PATH=$PATH:/sbin
root@server:/# update-initramfs -u
```

There is no need to make changes in other subsystems (for example, in the firewall rules).

## Replacing the node in the network core

In case of replacement of the system administrator's workstation, virtualization servers or the Avers system, as well as the main gateway the following edits should be made on the accounting server:

In the **/etc/network/neighbours** file, the link layer address of the changed node (MAC address, lladdr) should be replaced.

Then update the firewall configuration file **/etc/nftables**. In particular, it is required to replace the link (only for the system administrator's computer) and network layers addresses of the outgoing node with the corresponding addresses of the newly installed one in the variables ADMIN\_LLADDR, ADMIN\_HOST, AVERS\_HOST and GATEWAY.

Then it remains only to restart the services and make sure they are working properly:

```
root@server:/tmp# systemctl restart nftables
root@server:/tmp# systemctl status nftables
root@server:/tmp# systemctl restart networking
root@server:/tmp# systemctl status networking
```

## Disk array maintenance

Conditionally, we can distinguish three tasks of servicing a disk array on this system:

- monitoring the current state of the array
- access to data when one of the disks is damaged
- replacement of a damaged disk

There are several ways to track the state of an array.

Firstly, the mdmonitor systemd service, described when configuring the array, is continuously running in the system. It notifies root@localhost about problems by means of in-system email.

Secondly, the output of the command

```
root@server:/# cat /proc/mdstat
Personalities: [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb[0] sdc[1]
      488254464 blocks super 1.2 [2/2] [UU]
      bitmap: 0/4 pages [0KB], 65536KB chunk

unused devices: <none>
```

displays the current state of all arrays in the system. To view continuously changing (for example, when synchronizing disks) information from this file, you can use the following command

```
root@server:/# watch cat /proc/mdstat
```

Detailed information about the array can be obtained as follows:

```
root@server:/# mdadm --detail /dev/md0
```

Similarly, the detailed data about the disk included in the array are extracted as follows:

```
root@server:/# mdadm --examine /dev/sdb
```

The array integrity check can be performed using the following three consecutive commands:

```
root@server:/# echo check > /sys/block/md0/md/sync_action
root@server:/# watch cat /proc/mdstat
root@server:/# cat /sys/block/md0/md/mismatch_cnt
```

The first command initiates the check (the verification took more than one hour on the system in question). The second command allows you to monitor the progress of this check, the third one shows the result.

In case of failure of one of the disks, the output of the command

```
root@server:/# mdadm --detail /dev/md0
```

will show an inactive array with one working device on which there is a superblock. At the same time, in case of a disk failure at the boot stage, the system would try to assemble the array for some time, and then would switch to emergency mode.

If you need to access the array data before restoring its composition, then you need to rebuild and re-mount the array. These operations can be performed by the following series of commands:

```
root@server:/# mdadm --stop /dev/md0
root@server:/# mdadm /dev/md0 --assemble --run --verbose
root@server:/# mount /dev/md0 /srv/samba/archive/
```

To restore the composition of the array, turn off the system, remove the damaged and install a spare media, turn on the system. After that, by the output of the command

```
root@server:/# mdadm --detail /dev/md0
```



it can be concluded that the new disk is included in the array, but does not work, because there is no superblock on it. The disk turned out to be included in the array, because it received the same name in the system (for example, /dev/sdb) as the failed disk, since it is connected via the same bus to the same connector on the motherboard, has the same size and model. Therefore, you should unmount and stop the array, and then insert a new disk into its composition.

```
root@server:/# umount /dev/md0
root@server:/# mdadm --stop /dev/md0
root@server:/# mdadm --add /dev/sdb
```

After that, using the command

```
root@server:/# mdadm --detail /dev/md0 --verbose
```

you should ensure that there are already two working disks in the array, the newly added one is synchronized with the serviceable one. After waiting for the synchronization to finish

```
root@server:/# watch mdadm --detail /dev/md0 --verbose
```

it remains only to mount the array or reboot the system.

It should be explained why the available spare disk was not installed in the system and was not specified as a spare when configuring the RAID. This decision was made on the grounds that the disks used in the array are in good condition, the workload on them is relatively small, therefore the probability of their failure is insignificant. A spare disk may not be needed throughout the entire service life of the server, and installing it initially will waste its resource and electricity aimlessly. At the same time, regular monitoring of the server by the system administrator, the possibility of short-term decommissioning of the server and unhindered physical access of the administrator to it allow timely identification of the need to replace the disk and carry it out.

## Replacing an outdated local repository key

After the expiration of the local repository key, it must be replaced. To do this, it is enough to add a new key in exactly the same way as during the initial setup. But to maintain the cleanliness of the system, it is not superfluous to delete the old key. The following fragment of the terminal session (with some abbreviations of the system responses) demonstrates the necessary actions:

```
root@server:/# apt-key list
/etc/apt/trusted.gpg
-----
pub   rsa3072 2019-08-20 [SC] [просрочен с: 2021-08-19]
      FF63 9F36 C9A5 DE8B 16AF  F1E6 5647 BD17 421C B415
uid           [   просрочен   ] maintainer <maintainer@localhost>

...
root@server:/# apt-key del "FF63 9F36 C9A5 DE8B 16AF  F1E6 5647 BD17 421C B415"
OK
root@server:/# cd /tmp && wget http://debian.service.school34/ppa/repository_key.asc
...
root@server:/tmp# apt-key add repository_key.asc
OK
root@server:/tmp# apt-key list
/etc/apt/trusted.gpg
-----
pub   rsa4096 2021-10-08 [SC] [   родеи до: 2031-10-06]
      C395 1533 5648 35F7 8A0F  C094 057B A98A E75D EEFA
uid           [ неизвестно ] maintainer <debmaintainer@mail.service.school34>
```

sub rsa4096 2021-10-08 [E] [    годен до: 2031-10-06]

...

**root@server:/tmp# apt-get update**

Пол:1 http://debian.service.school34/buster buster InRelease [122 kB]

Пол:2 http://debian.service.school34/security buster/updates InRelease [65,4 kB]

Игн:3 http://debian.service.school34/ppa buster InRelease

Пол:4 http://debian.service.school34/ppa buster Release [1 655 B]

Пол:5 http://debian.service.school34/ppa buster Release.gpg [833 B]

Пол:6 http://debian.service.school34/ppa buster/main amd64 Packages [4 148 B]

Чтение списков пакетов... Готово