

Сессионное задание

Задача 1. Шерлок Холмс и LVM+RAID	2
Постановка задачи	2
Объяснение решения.....	2
Решение	3
Аппаратные сбои	5
Аппаратный сбой 1. Система на первом RAID0 (SSD1+SSD2), без LVM.....	5
Аппаратный сбой 2. Система на втором RAID1 (HDD5+HDD6), без LVM.....	6
Аппаратный сбой 3. Система на RAID0 (SSD1+SSD2) + RAID1 (HDD5+HDD6) через LVM.....	6
Аппаратный сбой 4. Система на RAID0 (SSD3+SSD4) — «жёлтый массив».....	7
Аппаратный сбой 5. Система на RAID1 (HDD) внутри LVM	7
Программные сбои	8
Программный сбой 1. RAID (mdadm) не собирается автоматически (битые/потерянные метаданные)	8
Программный сбой 2. Неправильное имя массива (md1 ожидался, а собрался md127) / несоответствие имени	8
Программный сбой 3. LVM-метаданные повреждены / VG не найден	9
Программный сбой 4. VG «partial»: LVM отказывается активировать root из-за «missing PV»	9
Программный сбой 5. Initramfs не содержит поддержки mdraid/LVM или нужных конфигов	9
Программный сбой 6. Несоответствие UUID (fstab/GRUB указывают старые идентификаторы)	10
Программный сбой 7. Гонка инициализации (udev/порядок появления устройств, таймауты)	10
Ответ.....	11
Задача 2. Найди 10 отличий: Nginx/Apache	12
Постановка задачи	12
Решение	13
Nginx.conf	13
Все ошибки в <code>nginx.conf</code>	13
Исправленный файл конфигурации <code>nginx.conf</code>	14
Apache.conf	14
Все ошибки в <code>apache.conf</code>	14
Исправленный файл конфигурации <code>apache.conf</code>	15

Задание 3. Задача 3. Shell-scripting vs «Однострочники»	16
Постановка задачи	16
Решение	16

Задача 1. Шерлок Холмс и LVM+RAID

Постановка задачи

Условие

На сервере 6 дисков (4 SSD, 2 HDD):

Диски SSD #1 и SSD #2 объединены в RAID 0.

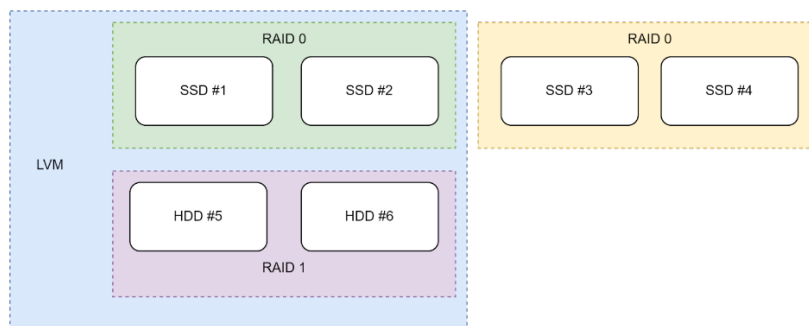
Диски SSD #3 и SSD #4 объединены в RAID 0.

Диски HDD #5 и HDD #6 объединены в RAID1.

Вопрос

Система на сервере перестала запускаться. Имея вводные из условия, с каким или какими дисками произошла проблема, если точно поломка в нем или них? Почему?

Ответ поясните. Вспомните об особенностях настройки LVM.



Объяснение решения

Так как в задаче четко не указано, что проблема в дисках, рассмотрим как аппаратные, так и программный сбои. Фразу «Имея вводные из условия, с каким или какими дисками произошла проблема, если точно поломка в нем или них?» читаю так, что в чем поломка, в дисках или нет, не понятно. Фраза «Если точно поломка в нем или в них?» может быть трактована неоднозначно. Что значит: Если? Если надо четко дать задачу, то так и надо написать

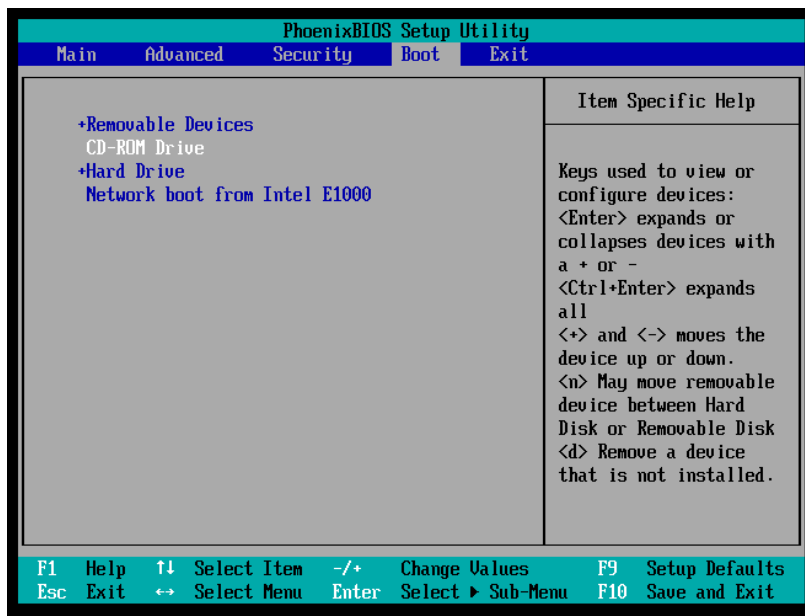
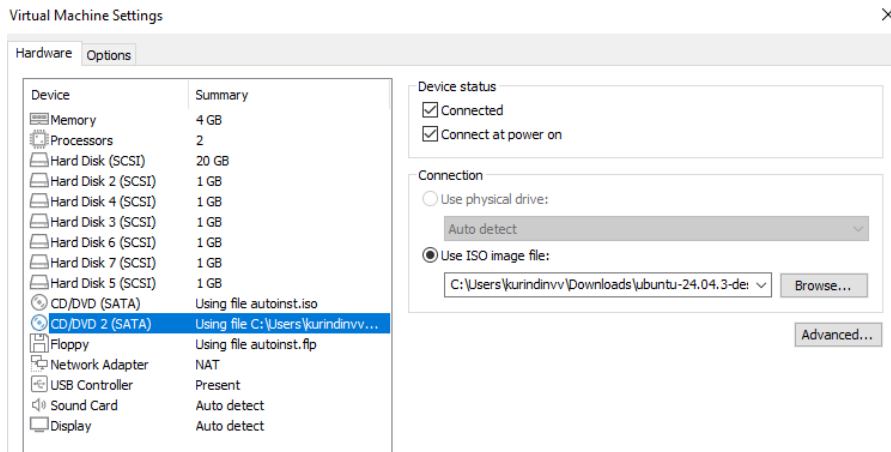
Предусловия: поломка в дисках. Тем не менее , прошу обратить внимание, что в ответе будут указаны и аппаратные сбои.

Решение

Настройка VMware для запуска с Live USB

Так как система не запускается, то нам надо загрузиться с live USB и попытаться начать выяснять , в чем реально у нас проблема.

Для этого сначала подправим bios –загрузки.



Потом надо не забыть постаивть CD-ROM Drive после hard drive.

Вот,что я получил, загрузившись с live usb на своей виртуалке с Ubuntu.

```

ubuntu@ubuntu:~$ lsblk -f
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
fd0
loop0
  squash 4.0                                0 100% /rofs
loop1
  squash 4.0
loop2
  squash 4.0
loop3
  squash 4.0                                0 100% /snap/bare/5
loop4
  squash 4.0                                0 100% /snap/core22/2045
loop5
  squash 4.0                                0 100% /snap/firefox/6565
loop6
  squash 4.0                                0 100% /snap/firmware-updater/167
loop7
  squash 4.0                                0 100% /snap/gtk-common-themes/1535
loop8
  squash 4.0                                0 100% /snap/snapd-desktop-integration/315
loop9
  squash 4.0                                0 100% /snap/thunderbird/769
loop10
  squash 4.0                                0 100% /snap/snap-store/1270
loop11
  squash 4.0                                0 100% /snap/snapd/24792
loop12
  squash 4.0                                0 100% /snap/gnome-42-2204/202
loop13
  squash 4.0                                0 100% /snap/ubuntu-desktop-bootstrap/413
sda
├─sda1
├─sda2
│   └─ext4 1.0                                5990bb5d-32d0-44b3-a10b-3f805f8e03e5
sdb
├─sdb1
│   ├──linux_ 1.2 0                                5f1086e0-29af-a85e-82b9-52fb2367f845
│   ├──nd127
│   │   ├──LVM2_m LVM2                                j0n9Yr-kR28-z3oH-8CU0-2gLI-sv9R-F0whAx
│   │   └─vg_raids-lv_data
│   │       └─ext4 1.0 data_lv                                77efb2f9-0fbd-40c1-afd9-33bd1407023d
└─sdc
  ├──sdc1
  │   ├──linux_ 1.2 0                                5f1086e0-29af-a85e-82b9-52fb2367f845
  │   ├──nd127
  │   │   ├──LVM2_m LVM2                                j0n9Yr-kR28-z3oH-8CU0-2gLI-sv9R-F0whAx
  │   │   └─vg_raids-lv_data
  │   │       └─ext4 1.0 data_lv                                77efb2f9-0fbd-40c1-afd9-33bd1407023d
  └─sdd
    ├──sdd1
    │   ├──linux_ 1.2 0                                5f1086e0-29af-a85e-82b9-52fb2367f845
    │   ├──nd127
    │   │   ├──LVM2_m LVM2                                j0n9Yr-kR28-z3oH-8CU0-2gLI-sv9R-F0whAx
    │   │   └─vg_raids-lv_data
    │   │       └─ext4 1.0 data_lv                                77efb2f9-0fbd-40c1-afd9-33bd1407023d
    └─sde LVM2_m LVM2                                D0rFyY-bA3k-DATP-rkuI-Lypp-8x0C-vnLWKS
      ├──vg_lstack-lv_a
      │   ├──linux_ 1.2 1                                ca9567ee-2150-315f-1626-cb713bb400fd
      │   └─nd120
      │       ├──ext4 1.0 data_nd1                                cbea7053-e44f-45b3-b9db-4b39e38c2092
      └─sdf LVM2_m LVM2                                ckrItx-Cqu3-jbQ0-15fd-ns34-hx0u-FZxsIE
        ├──vg_lstack-lv_b
        │   ├──linux_ 1.2 1                                ca9567ee-2150-315f-1626-cb713bb400fd
        │   └─nd120
        │       ├──ext4 1.0 data_nd1                                cbea7053-e44f-45b3-b9db-4b39e38c2092
        └─sdg LVM2_m LVM2                                0aExyc-M0Yc-M51n-G4W3-iubc-0tAd-nP09Kc
          ├──vg_lstack-lv_c
          │   ├──linux_ 1.2 1                                ca9567ee-2150-315f-1626-cb713bb400fd
          └─nd120
              ├──ext4 1.0 data_nd1                                cbea7053-e44f-45b3-b9db-4b39e38c2092

```

Далее видим результаты выполнения предыдущих домашних заданий

```
ubuntu@ubuntu:~$ sudo -i
root@ubuntu:~# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [linear] [raid0] [raid1] [raid10]
md126 : active (auto-read-only) raid5 dm-1[3] dm-2[4] dm-0[0]
        1841152 blocks super 1.2 level 5, 256k chunk, algorithm 2 [3/3] [UUU]

md127 : active (auto-read-only) raid5 sdb1[0] sdd1[3] sdc1[4]
        2091008 blocks super 1.2 level 5, 256k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

```
root@ubuntu:~# lsblk -f | grep -E "ext4|xfs"
└─sda2                ext4          1.0                5996bb5d-32d0-44b3-a16b-3f805f8e03e5
   └─vg_raid5-lv_data  ext4          1.0                data_lv             77efb2f9-6fbd-40c1-afd9-33bd1407623d
      └─vg_raid5-lv_data  ext4          1.0                data_lv             77efb2f9-6fbd-40c1-afd9-33bd1407623d
         └─vg_raid5-lv_data  ext4          1.0                data_lv             77efb2f9-6fbd-40c1-afd9-33bd1407623d
            └─md126         ext4          1.0                data_md1            cbea7053-e44f-45b3-b9db-4b39e38c2092
               └─md126         ext4          1.0                data_md1            cbea7053-e44f-45b3-b9db-4b39e38c2092
                  └─md126         ext4          1.0                data_md1            cbea7053-e44f-45b3-b9db-4b39e38c2092
```

Подготовка

`sudo -i` #сразу становимся под root тут.

возможно потребуется установить утилиты (Ubuntu/Debian Live):

```
command -v mdadm >/dev/null 2>&1 || { apt-get update && apt-get install -y mdadm; }
```

```
command -v lvs >/dev/null 2>&1 || { apt-get update && apt-get install -y lvm2; }
```

```
mkdir -p /mnt/target
```

```
mdadm --assemble --scan --readonly || true
```

```
vgchange -ay --partial || true
```

```
lsblk -f
```

```
cat /proc/mdstat
```

Далее идут уже мои теоретические изыскания, потому что проверить на практике систему под рейд я смог из-за ограниченного количества времени.

Аппаратные сбои

Аппаратный сбой 1. Система на первом RAID0 (SSD1+SSD2), без LVM

Признаки в выводе:

```
mount -o ro /dev/md0 /mnt/target || echo "Не монтируется /dev/md0"
```

если /boot отдельный:

```
BOOT_DEV=$(awk ' $2==" /boot" {print $1;exit}' /mnt/target/etc/fstab); [ -n "$BOOT_DEV" ]
&& mount -o ro "$BOOT_DEV" /mnt/target/boot || true
```

```

awk '$2=="/{print}' /mnt/target/etc/fstab

grep -E 'root=(UUID=|/dev/md|/dev/mapper/)' /mnt/target/boot/grub/grub.cfg | tail -n1

pvs; vgs; lvs -a -o +devices

umount /mnt/target/boot 2>/dev/null || true

umount /mnt/target 2>/dev/null || true

```

Вывод: система на массиве RAID0 (SSD1+SSD2). Очень рискованная схема: поломка любого одного SSD = падение ОС.

Результат: поломка диска тут может привести к неработоспособности системы

Аппаратный сбой 2. Система на втором RAID1 (HDD5+HDD6), без LVM

Признаки в выводе:

```

mount -o ro /dev/md1 /mnt/target || echo "Не монтируется /dev/md1"

BOOT_DEV=$(awk '$2=="boot"{print $1;exit}' /mnt/target/etc/fstab); [ -n "$BOOT_DEV" ]
&& mount -o ro "$BOOT_DEV" /mnt/target/boot || true

awk '$2=="/{print}' /mnt/target/etc/fstab

grep -E 'root=(UUID=|/dev/md|/dev/mapper/)' /mnt/target/boot/grub/grub.cfg | tail -n1

pvs; vgs; lvs -a -o +devices

umount /mnt/target/boot 2>/dev/null || true

umount /mnt/target 2>/dev/null || true

```

Вывод: система на зеркале HDD (RAID1).

Результат: поломка диска тут ни на что не повлияет, если не вылетят оба диска

Аппаратный сбой 3. Система на RAID0 (SSD1+SSD2) + RAID1 (HDD5+HDD6) через LVM

Признаки в выводе:

```

lvs -a -o lv_path,lv_name,vg_name,devices --noheadings

ROOT_LV=$(lvs --noheadings -o lv_path,lv_name | awk 'tolower($2) ~ /^(^|[-_])root$/{print $1; exit}')

mount -o ro "$ROOT_LV" /mnt/target

BOOT_DEV=$(awk '$2=="boot"{print $1;exit}' /mnt/target/etc/fstab); [ -n "$BOOT_DEV" ]
&& mount -o ro "$BOOT_DEV" /mnt/target/boot || true

```

```

awk '$2=="/{print}' /mnt/target/etc/fstab

grep -E 'root=(UUID=|/dev/md|/dev/mapper/)' /mnt/target/boot/grub/grub.cfg | tail -n1

lvs -a -o lv_name,devices | grep -E '(root|lv_root)'

umount /mnt/target/boot 2>/dev/null || true

umount /mnt/target 2>/dev/null || true

```

Вывод: система размазана по обоим массивам. Сбой одного SSD из RAID0 ломает всё.

Результат: поломка диска тут может привести к неработоспособности системы

Аппаратный сбой 4. Система на RAID0 (SSD3+SSD4) — «жёлтый массив»

Признаки в выводе:

```

mount -o ro /dev/md2 /mnt/target || echo "Не монтируется /dev/md2"

BOOT_DEV=$(awk '$2=="boot"{print $1;exit}' /mnt/target/etc/fstab); [ -n "$BOOT_DEV" ]
&& mount -o ro "$BOOT_DEV" /mnt/target/boot || true

awk '$2=="/{print}' /mnt/target/etc/fstab

grep -E 'root=(UUID=|/dev/md|/dev/mapper/)' /mnt/target/boot/grub/grub.cfg | tail -n1

pvs; vgs; lvs -a -o +devices

umount /mnt/target/boot 2>/dev/null || true

umount /mnt/target 2>/dev/null || true

```

Вывод: система на RAID0 из SSD3+SSD4. Тоже крайне ненадёжно: один диск — и ОС падает.

Результат: поломка диска тут может привести к неработоспособности системы

Аппаратный сбой 5. Система на RAID1 (HDD) внутри LVM

Признаки в выводе:

```

lvs -a -o lv_path,lv_name,vg_name,devices --noheadings

lvs -a -o lv_name,devices | grep -E '(root|lv_root)'

ROOT_LV=$(lvs --noheadings -o lv_path,lv_name | awk 'tolower($2) ~ /^(^|[-_])root$/{print $1; exit}')

mount -o ro "$ROOT_LV" /mnt/target

```

```
BOOT_DEV=$(awk ' $2==" /boot" {print $1;exit}' /mnt/target/etc/fstab); [ -n "$BOOT_DEV" ]  
&& mount -o ro "$BOOT_DEV" /mnt/target/boot || true
```

```
awk ' $2==" /" {print}' /mnt/target/etc/fstab
```

```
grep -E 'root=(UUID=|/dev/md|/dev/mapper/)' /mnt/target/boot/grub/grub.cfg | tail -n1
```

```
umount /mnt/target/boot 2>/dev/null || true
```

```
umount /mnt/target 2>/dev/null || true
```

Вывод: формально система в LVM, но фактически она сидит только на RAID1. В этом случае отказ SSD не ломает загрузку.

Результат: поломка одного диска тут ни на что не повлияет

На текущий момент имеем, что проблемы с невозможностью загрузить систему мы имеем при:

- Аппаратном сбое 1 «Система на первом RAID0 (SSD1+SSD2)», без LVM ,
- Аппаратном сбое 3 «Система на RAID0 (SSD1+SSD2) + RAID1 (HDD5+HDD6) через LVM»,
- Аппаратном сбое 4 «Система на RAID0 (SSD3+SSD4) — «жёлтый массив»

Далее смотрим программные проблемы

Программные сбои

Программный сбой 1. RAID (mdadm) не собирается автоматически (битые/потерянные метаданные)

Признаки в выводе:

```
mdadm --detail --scan
```

```
cat /proc/mdstat
```

```
lsblk -f | grep -E 'md[0-9]|sd[a-z]'
```

Вывод: диски видны, но массив не собран из-за метаданных/конфига mdadm.

Результат: система не загружается, даже при «живых» дисках (root на недоступном md-устройстве).

Программный сбой 2. Неправильное имя массива (md1 ожидался, а собрался md127) / несоответствие имени

Признаки в выводе:

```
# что ожидает установленная система:
```



```
grep -E 'root=(UUID=|/dev/md|/dev/mapper/)' /mnt/target/boot/grub/grub.cfg
```

что реально собрано в live:

```
cat /proc/mdstat
```

```
mdadm --detail --scan
```

Вывод: несоответствие между тем, что ожидает загрузчик/`fstab`, и тем, как ядро/`udev` назвали массив.

Результат: система не находит `root`-устройство и уходит в `emergency shell`.

Программный сбой 3. LVM-метаданные повреждены / VG не найден

Признаки в выводе:

```
cat /proc/mdstat
```

```
pvs; vgs; lvs -a -o +devices
```

Вывод: RAID в порядке, но LVM-слой сломан (метаданные VG/LV повреждены/утрачены).

Результат: система не загружается, т.к. `root-LV` недоступен при целых массивах.

Программный сбой 4. VG «partial»: LVM отказывается активировать root из-за «missing PV»

Признаки в выводе:

```
vgs
```

```
lvs -a -o +devices | grep '(missing)'
```

```
vgchange -ay --partial
```

Вывод: LVM считает группу неполной (ссылается на отсутствующий PV/UUID), даже если RAIDы собраны — несогласованность метаданных VG и фактических устройств.

Результат: `root` не активируется; загрузка невозможна без вмешательства.

Программный сбой 5. Initramfs не содержит поддержки mdraid/LVM или нужных конфигов

Признаки в выводе:

```
KVER=$(ls /mnt/target/boot/initrd.img-* 2>/dev/null | sed 's|.*|initrd.img-||' | tail -nl)
```

```
[ -n "$KVER" ] && lsinitramfs "/mnt/target/boot/initrd.img-$KVER" | \
grep -E ' (^etc/mdadm|/lvm|/dm-|^usr/lib/modules/.*kernel/drivers/md/)
```

Вывод: образ initramfs собран без модулей/конфигов для mdraid/LVM, поэтому на стадии initrd корневое устройство недоступно.

Результат: система не находит root на ранней фазе и падает в initramfs shell.

Программный сбой 6. Несоответствие UUID (fstab/GRUB указывают старые идентификаторы)

Признаки в выводе:

```
# UUID из установленной ОС:

awk '$2=="{"{print $1}' /mnt/target/etc/fstab

# Сопоставляем с реальностью в live:

blkid

# Для md/LVM дополнительно:

mdadm --detail --scan

pvs -o+pv_uuid; vgs -o+vg_uuid; lvs -a -o+lv_uuid
```

Вывод: после пересборки/миграции изменились UUID массива/раздела/VG, а конфигурационные файлы остались старые.

Результат: root не монтируется из-за неверных ссылок по UUID.

Программный сбой 7. Гонка инициализации (udev/порядок появления устройств, таймауты)

Признаки в выводе:

```
# если журналы persistent:

journalctl --directory=/mnt/target/var/log/journal -b -p err | \

grep -E 'Timed out waiting for device|A start job is running'

# иначе смотрим dmesg старта, если сохранён:

grep -i 'Timed out waiting for device' /mnt/target/var/log/* 2>/dev/null
```

Вывод: устройства md/LVM появляются с запозданием относительно попытки монтирования root (порядок/тайминги systemd-udev).

Результат: загрузка может срывать/зависать, хотя аппаратные ресурсы и метаданные в порядке.

Ответ

Аппаратный сбой 1. Система на первом RAID0 (SSD1+SSD2), без LVM

- Отказ одного SSD приводит к полной недоступности массива.
- Система не загружается.
- При выходе из строя двух SSD результат аналогичен (массив полностью потерян).

Аппаратный сбой 2. Система на втором RAID1 (HDD5+HDD6), без LVM

- Отказ одного HDD не нарушает работоспособность массива.
- При отказе двух HDD массив становится недоступен, система не загружается.
- Возможный частный случай: если загрузчик установлен только на одном HDD, отказ именно этого диска делает систему не загружаемой.

Аппаратный сбой 3. Система на RAID0 (SSD1+SSD2) + RAID1 (HDD5+HDD6) через LVM

- Отказ одного SSD в RAID0 делает недоступной часть LVM, система не загружается.
- При одновременном отказе двух HDD массив RAID1 теряется, LVM становится частично недоступен, система не загружается.

Аппаратный сбой 4. Система на RAID0 (SSD3+SSD4) — «жёлтый массив»

- Отказ одного SSD приводит к полной недоступности массива.
- Если на массиве размещены системные разделы (/ , /boot), система не загружается.
- Если массив использовался только для данных, отказ не влияет на загрузку.
- При отказе двух SSD результат аналогичен (массив полностью потерян).

Аппаратный сбой 5. Система на RAID1 (HDD5+HDD6) внутри LVM

- Отказ одного HDD не нарушает работоспособность массива.
- При отказе двух HDD массив становится недоступен, система не загружается.
- При установке загрузчика только на один HDD возможна недоступность системы при отказе именно этого диска.

И учитываем, что при любом из перечисленных 7-ми программных сбоев, система также не загрузится.

Задача 2. Найди 10 отличий: Nginx/Apache

Постановка задачи

Условие

Даны два конфигурационных файла настройки веб-серверов Nginx и Apache. Необходимо найти 10 ошибок и написать исправленные конфигурационные файлы в поле ответа.

nginx.conf

```
server {  
    listen 80;  
    server_name example.com;  
  
    root /var/www/html;  
  
    location / {  
        index index.htm;  
    }  
  
    location /old-page {  
        return 302 /new-page;  
    }  
  
    location /proxy {  
        proxy_pass http://backend_server;  
    }  
}
```

apache.conf

```

<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/wrong_path

    <Directory /var/www/html>
        Options Indexes FollowSymlinks
        AllowOverride all
        Require all denied
    </Directory>

    ErrorLog ${APACHE_MAIN_LOG_DIR}/error.log

    Redirect temp /old-page /new-page

    ProxyPass /proxy http://backend_server
    ProxyPassReverse /proxy http://backend_server

    DirectoryIndex
</VirtualHost>

```

Решение

Nginx.conf

Все ошибки в nginx.conf

1. `index index.htm;`
 Указан только `index.htm`, а стандартный `index.html` не упомянут → при наличии `index.html` будет 403/404.
 Исправление: `index index.html index.htm;`
2. `return 302 /new-page;`
 Используется относительный путь, т.е. редирект зависит от контекста. Лучше указывать абсолютный URL.
 Исправление: `return 302 http://example.com/new-page;`
3. `proxy_pass http://backend_server;`
 Недопустимое имя хоста. Подчёркивание в DNS нельзя.
 + тут пропущена буква `p` в `http`.
 Исправление: `proxy_pass http://backend-server;` или IP.
4. `location /proxy { proxy_pass http://backend-server; }`
 Ошибка в склейке путей: при отсутствии хвостового `/` к апстриму уйдёт `/proxy/test`, а не `/test`.
 Исправление:

```
location /proxy/ {
```

```
    proxy_pass http://backend-server/;
}
```

5. Нет заголовков для бэкенда (proxy_set_header)

Без Host, X-Real-IP, X-Forwarded-For, X-Forwarded-Proto бэкенд может работать некорректно.

Исправление: добавить proxy_set_header ...

Исправленный файл конфигурации nginx.conf

```
server {
    listen 80;

    server_name example.com;

    root /var/www/html;

    location / {
        index index.html index.htm;
    }

    location /old-page {
        return 302 http://example.com/new-page;
    }

    location /proxy/ {
        proxy_pass http://backend-server/;

        # Рекомендуемые заголовки для корректной работы бэкенда
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Apache.conf

Все ошибки в apache.conf

1. DocumentRoot /var/www/wrong_path

Несуществующий путь, а в <Directory> указан /var/www/html.
Несогласованность.

Исправление: DocumentRoot /var/www/html

2. **Require all denied**
Полный запрет доступа → сайт отдаст 403.
Исправление: `Require all granted`
3. **DirectoryIndex пустой**
Директива без значения вызовет ошибку.
Исправление: `DirectoryIndex index.html`
4. **ErrorLog \${APACHE_MAIN_LOG_DIR}/error.log**
Такой переменной нет. В Ubuntu/Debian используют `${APACHE_LOG_DIR}`.
Исправление: `${APACHE_LOG_DIR}/error.log`
5. **Options Indexes FollowSymLinks**
`Indexes` включает листинг каталогов — небезопасно.
Исправление: `Options FollowSymLinks`
6. **ProxyPass /proxy http://backend_server**
 - Неверное имя хоста (подчёркивание).
 - Нет хвостового / → URLы будут склеиваться неправильно.Исправление:

`ProxyPass /proxy/ http://backend-server/`
`ProxyPassReverse /proxy/ http://backend-server/`
7. **Redirect temp /old-page /new-page**
Здесь редирект тоже относительный → лучше указывать абсолютный URL.
Исправление: `Redirect temp /old-page http://example.com/new-page`
8. Дополнительно отмечу, что названия `apache2.conf` нет, есть, как у меня, например (Fedora), `/etc/httpd/conf/httpd.conf` или Debian/Ubuntu — `/etc/apache2/apache2.conf`
9. Также есть опечатки в `http://backendserver` целых две

Исправленный файл конфигурации `apache.conf`

```
<VirtualHost *:80>

    ServerName example.com

    DocumentRoot /var/www/html

    <Directory /var/www/html>

        Options FollowSymLinks

        AllowOverride All

        Require all granted

    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log

    Redirect temp /old-page http://example.com/new-page
```

```

ProxyPass /proxy/ http://backend-server/

ProxyPassReverse /proxy/ http://backend-server/

DirectoryIndex index.html

</VirtualHost>

```

Задание 3. Задача 3. Shell-scripting vs «Однострочники»

Постановка задачи

Написать shell-скрипт (однострочник), который проверяет права доступа на SSH-ключи, хранящиеся в директории ~/.ssh/. Не забудьте, что на системе может быть несколько пользователей, а также root. Скрипт должен проверить, что приватные ключи имеют права доступа 600, а публичные — 644. Если права на файл отличаются от ожидаемых, информация об этом должна быть записана в лог-файл ssh_permission_errors.log в формате “<название файла> <права доступа>”.

Пример лог файла:

```

/home/user/.ssh/id_rsa 644
/home/user/.ssh/id_rsa.pub 777

```

Решение

Так как у меня никаких этих ключей на машине нет, создадим тестовые ключи и специально сделаем неправильные права.

```

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ echo "fake_private" > ~/.ssh/id_rsa
echo "fake_public" > ~/.ssh/id_rsa.pub

```

Добавим левых пользователей

```

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ LOG="$HOME/ssh_permission_errors.log"; : >"$LOG"; if ! systemctl is-active --
quiet sshd && ! systemctl is-active --quiet ssh; then echo "SSH is not enabled (sshd/ssh service is inactive)"; else found=0; for d in /root /home/* "$HOME" /tmp/tes
tuser*; do [ -d "$d/.ssh" ] || continue; found=1; sudo find "$d/.ssh" -maxdepth 1 -type f -name "*.pub" ! -perm 0644 -printf "%p %m\n"; sudo find "$d/.ssh" -maxdepth
1 -type f \( -name "id_*" -o -name "*.key" -o -name "identity" -o -name "*.pem" \) ! -name "*.pub" ! -perm 0600 -printf "%p %m\n"; done >>"$LOG"; if [ "$found" -eq
0 ]; then echo "No .ssh directories found for any users"; else n=$(wc -l <"$LOG"); [ "$n" -eq 0 ] && echo "All SSH keys have correct permissions" || { echo "Permissi
on issues found: $n (see log: $LOG)"; cat "$LOG"; }; fi; fi
Permission issues found: 7 (see log: /home/vladimir-kuryndin/ssh_permission_errors.log)
/home/vladimir-kuryndin/.ssh/id_rsa.pub 777
/home/vladimir-kuryndin/.ssh/id_rsa 644
/home/vladimir-kuryndin/.ssh/id_rsa.pub 777
/home/vladimir-kuryndin/.ssh/id_rsa 644
/tmp/testuser1/.ssh/id_rsa.pub 777
/tmp/testuser1/.ssh/id_rsa 644
/tmp/testuser3/.ssh/mykey.key.pub 700

```

Собственно, финальный скрипт.

```

DIRS="/root /home/* /tmp/testuser*"; LOG="$HOME/ssh_permission_errors.log"; : >"$LOG"; if
command -v systemctl >/dev/null 2>&1 && ! systemctl is-active --quiet sshd && ! systemctl is-active --
quiet ssh; then echo "SSH is not enabled (sshd/ssh service is inactive)"; else found=0; for d in $DIRS; do [
-d "$d/.ssh" ] || continue; found=1; sudo find "$d/.ssh" -maxdepth 1 -type f -name "*.pub" ! -perm 0644
-printf "%p %m\n"; sudo find "$d/.ssh" -maxdepth 1 -type f \( -name "id_*" -o -name "*.key" -o -name
"identity" -o -name "*.pem" \) ! -name "*.pub" ! -perm 0600 -printf "%p %m\n"; done >>"$LOG"; if [
"$found" -eq 0 ]; then echo "No .ssh directories found for any users"; else sort -u -o "$LOG" "$LOG";

```



```
n=$(wc -l <"$LOG"); [ "$n" -eq 0 ] && echo "All SSH keys have correct permissions" || { echo "Permission issues found: $n (see log: $LOG)"; cat "$LOG"; }; fi
```

Проверка на Ubuntu, где создавались тестовые пользователи и фейковые ключи.

```
vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:~/usr/local/src/ModSecurity$ DIRS="/root /home/* /tmp/testuser*"; LOG="$HOME/ssh_permission_errors.log"; :>"$LOG"; if command -v systemctl >/dev/null 2>&1 && ! systemctl is-active --quiet sshd && ! systemctl is-active --quiet ssh; then echo "SSH is not enabled (sshd/ssh service is inactive)"; else found=0; for d in $DIRS; do [ -d "$d/.ssh" ] || continue; found=1; sudo find "$d/.ssh" -maxdepth 1 -type f -name "*.pub" ! -perm 0644 -printf "%p %m\n"; sudo find "$d/.ssh" -maxdepth 1 -type f \( -name "id_*" -o -name "*.key" -o -name "identity" -o -name "*.pem" \) ! -name "*.pub" ! -perm 0600 -printf "%p %m\n"; done >>"$LOG"; if [ "$found" -eq 0 ]; then echo "No .ssh directories found for any users"; else sort -u -o "$LOG" "$LOG"; n=$(wc -l <"$LOG"); [ "$n" -eq 0 ] && echo "All SSH keys have correct permissions" || { echo "Permission issues found: $n (see log: $LOG)"; cat "$LOG"; }; fi; fi
Permission issues found: 5 (see log: /home/vladimir-kuryndin/ssh_permission_errors.log)
/home/vladimir-kuryndin/.ssh/id_rsa 644
/home/vladimir-kuryndin/.ssh/id_rsa.pub 777
/tmp/testuser1/.ssh/id_rsa 644
/tmp/testuser1/.ssh/id_rsa.pub 777
/tmp/testuser3/.ssh/nykey.key.pub 700
vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:~/usr/local/src/ModSecurity$ nano /home/vladimir-kuryndin/ssh_permission_errors.log
vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:~/usr/local/src/ModSecurity$
```

Проверка на Fedora, где не было экспериментов с ssh

```
vladimirkuryndin@fedora:~$ DIRS="/root /home/* /tmp/testuser*"; LOG="$HOME/ssh_permission_errors.log"; :>"$LOG"; if command -v systemctl >/dev/null 2>&1 && ! systemctl is-active --quiet sshd && ! systemctl is-active --quiet ssh; then echo "SSH is not enabled (sshd/ssh service is inactive)"; else found=0; for d in $DIRS; do [ -d "$d/.ssh" ] || continue; found=1; sudo find "$d/.ssh" -maxdepth 1 -type f -name "*.pub" ! -perm 0644 -printf "%p %m\n"; sudo find "$d/.ssh" -maxdepth 1 -type f \( -name "id_*" -o -name "*.key" -o -name "identity" -o -name "*.pem" \) ! -name "*.pub" ! -perm 0600 -printf "%p %m\n"; done >>"$LOG"; if [ "$found" -eq 0 ]; then echo "No .ssh directories found for any users"; else sort -u -o "$LOG" "$LOG"; n=$(wc -l <"$LOG"); [ "$n" -eq 0 ] && echo "All SSH keys have correct permissions" || { echo "Permission issues found: $n (see log: $LOG)"; cat "$LOG"; }; fi; fi
No .ssh directories found for any users
```