

# Домашнее задание (модуль 24)

## 24.5 Практическая работа

Цель практической работы

Применить полученные знания об информационной безопасности.

Что входит в работу

1. Провести сканирование при помощи rkhunter.
2. Заблокировать доступ по SSH при помощи iptables.
3. Запретить доступ к виртуальной машине.
4. Настроить выставление заголовка.
5. Внести изменения в конфигурацию БД.

Задание 1. Сканирование

Что нужно сделать

1. Установите на своей виртуальной машине rkhunter и запустите сканирование.
2. Изучите найденные проблемы (уровней error, warning и так далее).

Что оценивается

- Работа с утилитой для сканирования операционной системы на зловердный код и другие проблемы безопасности.
- Самостоятельная интерпретация результатов её работы.

Как отправить задание на проверку

Приложите скриншоты с запуском утилиты, опишите результат её работы в текстовом поле в свободной форме. (Насколько серьёзны найденные проблемы, что они значат.)

Ответ:

```
vgusev2007@skillbox-vgusev2007:~$ cat /etc/rkhunter.conf | grep WEB_CMD | tail -n 1
WEB_CMD=""
vgusev2007@skillbox-vgusev2007:~$
```

```
vgusev2007@skillbox-vgusev2007:~$ cat /etc/rkhunter.conf | grep MIRRORS_MODE | tail -n 1
MIRRORS_MODE=0
vgusev2007@skillbox-vgusev2007:~$ cat /etc/rkhunter.conf | grep UPDATE_MIRRORS | tail -n 1
UPDATE_MIRRORS=1
```

```
vgusev2007@skillbox-vgusev2007:~$ sudo rkhunter --update
[ Rootkit Hunter version 1.4.6 ]
```

#### Checking rkhunter data files...

Checking file mirrors.dat	[ Updated ]
Checking file programs_bad.dat	[ No update ]
Checking file backdoorports.dat	[ No update ]
Checking file suspscan.dat	[ No update ]
Checking file i18n/cn	[ Skipped ]
Checking file i18n/de	[ Skipped ]
Checking file i18n/en	[ No update ]
Checking file i18n/tr	[ Skipped ]
Checking file i18n/tr.utf8	[ Skipped ]
Checking file i18n/zh	[ Skipped ]
Checking file i18n/zh.utf8	[ Skipped ]
Checking file i18n/ja	[ Skipped ]

```
vgusev2007@skillbox-vgusev2007:~$ sudo rkhunter --check --skip-keypress
[ Rootkit Hunter version 1.4.6 ]
```

#### Checking system commands...

##### Performing 'strings' command checks

System checks summary  
=====

##### File properties checks...

Files checked: 145  
Suspect files: 1

##### Rootkit checks...

Rootkits checked : 498  
Possible rootkits: 0

##### Applications checks...

All checks skipped

The system checks took: 1 minute and 56 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.  
Please check the log file (/var/log/rkhunter.log)

```
vgusev2007@skillbox-vgusev2007:~$ sudo less /var/log/rkhunter.log | grep Warning
```

```
[21:19:52] /usr/bin/lwp-request [ Warning ]
[21:19:52] Warning: The command '/usr/bin/lwp-request' has been replaced by a script: /usr/bin/lwp-request: Perl script text executable
[21:21:20] Checking if SSH root access is allowed [ Warning ]
[21:21:20] Warning: The SSH configuration option 'PermitRootLogin' has not been set.
vgusev2007@skillbox-vgusev2007:~$
```

---

Первый warning говорит о том, что файл lwp-request должен быть бинарным файлом, а не скриптом.

Сперва читаем вообще, что это:

```
LWP-REQUEST(1p)                                User Contributed Perl Documentation

NAME
    lwp-request - Simple command line user agent

SYNOPSIS
    lwp-request [-afPuUsSedvhx] [-m method] [-b base URL] [-t timeout]
                [-i if-modified-since] [-c content-type]
                [-C credentials] [-p proxy-url] [-o format] url...

DESCRIPTION
    This program can be used to send requests to WWW servers and your local file system. The request is read from stdin. The content of the response is printed on stdout. Error messages are printed with a status value indicating the number of URLs that failed.

    The options are:

    -m <method>
        Set which method to use for the request. If this option is not used, then the method is default.
```

Ага... Похоже, очень старый аналог curl. Видимо, раньше это был бинарный файл, а теперь в ubuntu по крайней мере, это стало обычным сценарием на языке perl.

Гуглим:

rkhunter needs to know what package manager you are using.

Create or edit `/etc/rkhunter.conf.local` and add the following line:

```
PKG_MGR=DPKG
```

If you are not on Debian or Ubuntu, then change `DPKG` for your actual package manager.

Указываем, что наш менеджер пакетов dpkg. Таким образом, rkhunter, проанализирует информацию о всех файлах в deb пакетах, и «убедится», что скрипт вместо исполняемого файла, это нормально.

Исправляем первый Warning:

```
vgusev2007@skillbox-vgusev2007:~$ cat /etc/rkhunter.conf.local
PKG_MGR=DPKG
```

```
vgusev2007@skillbox-vgusev2007:~$ sudo rkhunter --check --skip-keypress
```

## System checks summary

=====

### File properties checks...

Files checked: 145

Suspect files: 0

### Rootkit checks...

Rootkits checked : 498

Possible rootkits: 0

### Applications checks...

All checks skipped

The system checks took: 2 minutes and 28 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.

Please check the log file (/var/log/rkhunter.log)

vgusev2007@skillbox-vgusev2007:~\$ █

---

```
vgusev2007@skillbox-vgusev2007:~$ sudo less /var/log/rkhunter.log | grep Warning
[21:31:53]    Checking if SSH root access is allowed          [ Warning ]
[21:31:53] Warning: The SSH configuration option 'PermitRootLogin' has not been set.
vgusev2007@skillbox-vgusev2007:~$ █
```

Отлично! Осталось только предупреждение на счет того, что нужно определить явно, что у нас может или не может входить Root пользователь. По умолчанию, может. Исправляем, и начинаем ходить не из-под root пользователя:

```
root@skillbox-vgusev2007:~# cp -r ~/.ssh/ /home/vgusev2007/
root@skillbox-vgusev2007:~# █
```

---

```
root@skillbox-vgusev2007:~# cat /etc/ssh/sshd_config | grep "PermitRootLogin no"
PermitRootLogin no
```

```
vgusev2007@skillbox-vgusev2007:~$ sudo rkhunter --check --skip-keypress  
[ Rootkit Hunter version 1.4.6 ]
```

Checking system commands...

```
■ Performing 'strings' command checks
```

```
System checks summary  
=====
```


```
File properties checks...  
  Files checked: 145  
  Suspect files: 0
```

```
Rootkit checks...  
  Rootkits checked : 498  
  Possible rootkits: 0
```

```
Applications checks...  
  All checks skipped
```

The system checks took: 2 minutes and 26 seconds

All results have been written to the log file: /var/log/rkhunter.log

No warnings were found while checking the system. 

```
vgusev2007@skillbox-vgusev2007:~$ sudo less /var/log/rkhunter.log | grep Warning  
vgusev2007@skillbox-vgusev2007:~$ █  
[01] 0:hash*
```

```
vgusev2007@skillbox-vgusev2007:~$ sudo less /var/log/rkhunter.log | grep error  
vgusev2007@skillbox-vgusev2007:~$ █
```

Никаких ошибок и предупреждений более нет, что и требовалось в задании.

## Задание 2. Блокировка

### Что нужно сделать

1. При помощи iptables заблокируйте доступ по SSH к вашей виртуальной машине.
2. Проверьте работу правила.
3. Сделайте так, чтобы эти изменения сохранялись после перезагрузки.

### Что оценивается

Корректное применение полученных знаний по работе с файрволом iptables.

### Как отправить задание на проверку

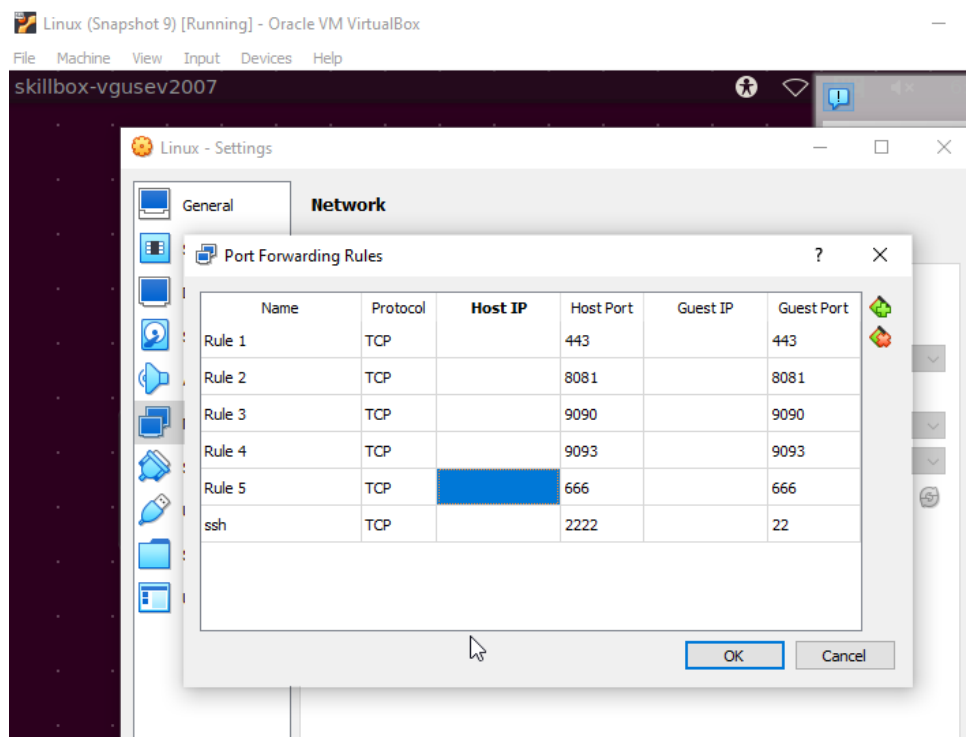
- Отправьте скриншоты использованных команд для настройки файрвола и настройки сохранения правил.
- Отправьте скриншот, на котором видно, что доступ по SSH действительно заблокирован.

```
vgusev2007@skillbox-vgusev2007:~$ sudo ufw status verbose
Status: inactive
vgusev2007@skillbox-vgusev2007:~$
```

Включаем ssh на нескольких портах:

```
vgusev2007@skillbox-vgusev2007:~$ sudo cat /etc/ssh/sshd_config.d/ports.conf
Port 22
Port 666
```

Пробрасываем NAT в VitrualBox, порт 666:



Пробуем достучаться по порту: 666

```
C:\Windows\system32\cmd.exe
```

```
C:\Users\admin>telnet 192.168.11.43 666
```

```
C:\Windows\system32\cmd.exe
```

```
SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.3
```

```
Invalid SSH identification string.
```

```
vgusev2007@skillbox-vgusev2007:~$ sudo ufw status  
Status: inactive
```

Разрешаем ssh на стандартном порту и применяем правила iptables:

```
vgusev2007@skillbox-vgusev2007:~$ sudo ufw allow ssh  
Rules updated  
Rules updated (v6)  
vgusev2007@skillbox-vgusev2007:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
vgusev2007@skillbox-vgusev2007:~$
```

Правило разрешающее ssh появилось:

```
vgusev2007@skillbox-vgusev2007:~$ sudo iptables -L | grep ssh  
ACCEPT      tcp    --  anywhere             anywhere             tcp dpt:ssh
```

Делаем проверку доступности ssh на втором порту:

```
C:\Windows\system32\cmd.exe
```

```
C:\Users\admin>telnet 192.168.11.43 666
```

ssh более недоступен:

```
C:\Telnet 192.168.11.43
```

Делаем перезагрузку:

```
vgusev2007@skillbox-vgusev2007:~$ uptime  
06:46:56 up 10:16, 4 users, load average: 0.00, 0.10, 0.15  
vgusev2007@skillbox-vgusev2007:~$ sudo init 6
```

```
vgusev2007@skillbox-vgusev2007:~$ uptime
06:48:48 up 1 min, 2 users, load average: 3.26, 0.91, 0.31
vgusev2007@skillbox-vgusev2007:~$ sudo iptables -L | grep ssh
[sudo] password for vgusev2007:
ACCEPT      tcp -- anywhere anywhere tcp dpt:ssh
vgusev2007@skillbox-vgusev2007:~$
```

После перезагрузки, порт 666 по-прежнему заблокирован:

```
C:\Windows\system32\cmd.exe
C:\Users\admin>telnet 192.168.11.43 666
Telnet 192.168.11.43
■
```

Проверим что у нас политика по умолчанию: DROP в INPUT:

```
vgusev2007@skillbox-vgusev2007:~$ sudo iptables -L -v | head -n 1
Chain INPUT (policy DROP 7 packets, 308 bytes)
```

Увеличим счетчик:

```
C:\Windows\system32\cmd.exe
C:\Users\admin>telnet 192.168.11.43 666
Telnet 192.168.11.43
```

```
vgusev2007@skillbox-vgusev2007:~$ sudo iptables -L -v | head -n 1
Chain INPUT (policy DROP 11 packets, 484 bytes)
```

Задание выполнено, а именно:

Работа политики DROP по умолчанию в INPUT – проверена, и работает.

Порт ssh под номером 666 – заблокирован, при этом порт 22 работает, что и позволяет мне выполнять домашнюю работу по ssh на порту 22, ввиду того что виртуальная машина VirtualBox, находится в тысячах километров от меня. – Проверена работа разрешающего правила iptables.

Так же, мы убедились, что после перезагрузки, все правила продолжают применяться, и нам не потребовалось использовать устаревшие решения, вроде: iptables-persistent, но, при этом, как и требуется в задании, мы используем iptables



### Задание 3. Запрет доступа

#### Что нужно сделать

1. Запретите весь доступ к виртуальной машине снаружи, кроме доступа по портам 80, 443 и по порту для SSH.
2. Проверьте, что доступ по SSH продолжил работать.
3. Продолжил ли работать apt-get update?
4. Подумайте о том, как это можно исправить.

#### Что оценивается

- Корректное применение полученных знаний по работе с файрволом iptables.
- Самостоятельное изучение документации к iptables.

#### Как отправить задание на проверку

- Приложите скриншоты использованных команд для настройки файрвола и настройки сохранения правил.
- Приложите скриншоты, на которых вы производите проверку работоспособности и правильного поведения правил.

#### Ответ:

Поскольку у нас политика таблица фильтра содержит для цепочки INPUT – требование DROP, убедимся, что при включении ufw у нас остаются доступными: http, https, ssh:

```
vgusev2007@skillbox-vgusev2007:~$ sudo ufw status numbered
Status: active
```

	To	Action	From
	--	-----	----
[ 1]	22/tcp	ALLOW IN	Anywhere
[ 2]	22/tcp (v6)	ALLOW IN	Anywhere (v6)

#### Упс... Кажется, нужно включить nginx профиль:

```
vgusev2007@skillbox-vgusev2007:~$ sudo ufw app list
Available applications:
CUPS
Nginx Full
Nginx HTTP
Nginx HTTPS
OpenSSH
Postfix
Postfix SMTPS
Postfix Submission
```

The connection has timed out

An error occurred during a connection to skillbox.hello.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

```
vgusev2007@skillbox-vgusev2007:~$ sudo ufw allow "Nginx Full"
Rule added
Rule added (v6)
vgusev2007@skillbox-vgusev2007:~$
```

### Смотрим какие порты у нас открылись:

```
vgusev2007@skillbox-vgusev2007:~$ cat /etc/ufw/applications.d/nginx | tail -n 4
[Nginx Full]
title=Web Server (Nginx, HTTP + HTTPS)
description=Small, but very powerful and efficient web server
ports=80,443/tcp

vgusev2007@skillbox-vgusev2007:~$ sudo ufw status numbered
Status: active
```

To	Action	From
--	-----	----
[ 1] 22/tcp	ALLOW IN	Anywhere
[ 2] Nginx Full	ALLOW IN	Anywhere
[ 3] 22/tcp (v6)	ALLOW IN	Anywhere (v6)
[ 4] Nginx Full (v6)	ALLOW IN	Anywhere (v6)

It works:



## Hello SkillBox

Проверяем, что доступ по ssh продолжает работать:

Проверяем, что доступ по ssh у нас имеется:

```
C:\Windows\system32\cmd.exe

C:\Users\admin>telnet 192.168.11.43 2222_
```

Да, всё отлично:

```
C:\ Telnet 192.168.11.43

SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.3
```

### Продолжил ли работать apt-get update?

```
vgusev2007@skillbox-vgusev2007:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://repo.mysql.com/apt/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
28 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Да, поскольку у нас политика OUTPUT разрешающая:

```
vgusev2007@skillbox-vgusev2007:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
```

### Почистим conntrack:

```
vgusev2007@skillbox-vgusev2007:~$ sudo conntrack -F
conntrack v1.4.6 (conntrack-tools): connection tracking table has been emptied.
vgusev2007@skillbox-vgusev2007:~$
```

### Посмотрим, что там есть:

```
vgusev2007@skillbox-vgusev2007:~$ sudo conntrack -L
tcp        6 431999 ESTABLISHED src=10.0.2.15 dst=10.0.2.2 sport=22 dport=54170 src=10.0.2.2 dst=10.0.2.15 sport=54170 dport=22 [ASSURED] mark=0 use=1
udp        17 29 src=10.0.2.15 dst=192.168.11.1 sport=39318 dport=53 src=192.168.11.1 dst=10.0.2.15 sport=53 dport=39318 mark=0 use=1
udp        17 29 src=127.0.0.1 dst=127.0.0.53 sport=48467 dport=53 src=127.0.0.53 dst=127.0.0.1 sport=53 dport=48467 mark=0 use=1
udp        17 29 src=10.0.2.15 dst=192.168.11.1 sport=46024 dport=53 src=192.168.11.1 dst=10.0.2.15 sport=53 dport=46024 mark=0 use=1
udp        17 29 src=127.0.0.1 dst=127.0.0.53 sport=34069 dport=53 src=127.0.0.53 dst=127.0.0.1 sport=53 dport=34069 mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 5 flow entries have been shown.
```

Есть ssh, есть запросы к dns

Выполним снова: apt update

```
vgusev2007@skillbox-vgusev2007:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://repo.mysql.com/apt/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 338 kB in 2s (179 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
28 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Да, всё работает, смотрим conntrack

```
vgusev2007@skillbox-vgusev2007:~$ sudo conntrack -L
udp      17 29 src=10.0.2.15 dst=192.168.11.1 sport=40405 dport=53 src=192.168.11.1 dst=10.0.2.15 sport=53 dport=40405 mark=0 use=1
tcp      6 68 TIME_WAIT src=127.0.0.1 dst=127.0.0.53 sport=38458 dport=53 src=127.0.0.53 dst=127.0.0.1 sport=53 dport=38458 [ASSURED] mark=0 use=1
udp      17 29 src=127.0.0.1 dst=127.0.0.53 sport=34537 dport=53 src=127.0.0.53 dst=127.0.0.1 sport=53 dport=34537 mark=0 use=1
tcp      6 431999 ESTABLISHED src=10.0.2.15 dst=10.0.2.2 sport=22 dport=54170 src=10.0.2.2 dst=10.0.2.15 sport=54170 dport=22 [ASSURED] mark=0 use=1
tcp      6 69 TIME_WAIT src=10.0.2.15 dst=65.9.55.48 sport=56600 dport=443 src=65.9.55.48 dst=10.0.2.15 sport=443 dport=56600 [ASSURED] mark=0 use=1
tcp      6 69 TIME_WAIT src=10.0.2.15 dst=23.201.249.199 sport=41566 dport=80 src=23.201.249.199 dst=10.0.2.15 sport=80 dport=41566 [ASSURED] mark=0 use=1
udp      17 29 src=10.0.2.15 dst=192.168.11.1 sport=49700 dport=53 src=192.168.11.1 dst=10.0.2.15 sport=53 dport=49700 mark=0 use=1
tcp      6 69 TIME_WAIT src=10.0.2.15 dst=91.189.91.39 sport=58268 dport=80 src=91.189.91.39 dst=10.0.2.15 sport=80 dport=58268 [ASSURED] mark=0 use=1
udp      17 29 src=127.0.0.1 dst=127.0.0.53 sport=60664 dport=53 src=127.0.0.53 dst=127.0.0.1 sport=53 dport=60664 mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 9 flow entries have been shown.
```

Почему?

**Потому что, в таблице conntrack появились разрешающие записи для INPUT! – Это делается автоматически, и делается операционной системой для удобства пользователей. – Если мы создали исходящее соединение, мы автоматически, должны разрешить входящий трафик для получения ответа.**

**Задание выполнено.** Все правила сохранены, и будут работать после перезагрузки. Все правила показаны, показана таблица conntrack. При том, задание выполнено при помощи современных инструментов входящих в состав ubuntu (с изучением материала).

## Задание 4. Выставление заголовка

### Что нужно сделать

1. Настройте выставление заголовка HSTS (HTTP Strict Transport Security) в Nginx.
2. Продемонстрируйте (например, при помощи команды curl), что этот заголовок действительно выставляется.

### Что оценивается

Применение полученных знаний про заголовки безопасности Nginx на примере HSTS.

### Как отправить задание на проверку

- Приложите скриншот с настройками Nginx.
- Приложите скриншот с проверкой применения настроек.

### Ответ:

```
vgusev2007@skillbox-vgusev2007:~$ curl -k -s -D- https://hello.skillbox/ |grep age
vgusev2007@skillbox-vgusev2007:~$ █
```

Пока заголовка нету, добавляем:

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

    # SSL configuration
    #
    listen 443 ssl default_server;
    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload";
}
```

```
vgusev2007@skillbox-vgusev2007:~$ cat /etc/nginx/sites-enabled/default |grep max-age
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload";
    █
```

---

```
vgusev2007@skillbox-vgusev2007:~$ sudo systemctl reload nginx.service
_
```

```
vgusev2007@skillbox-vgusev2007:~$ curl -k -s -D- https://hello.skillbox/ |grep age
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
_
```

## Задание 5. Изменения в конфигурации БД

### Что нужно сделать

Запустите скрипт `mysql_secure_installation` и внесите предложенные изменения в конфигурацию базы данных.

### Что оценивается

Применение полученных знаний о настройках безопасности MySQL.

### Как отправить задание на проверку

Пришлите скриншоты, на которых видна работа скрипта.

### Ответ:

```
vgusev2007@skillbox-vgusev2007:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary      file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2
[default] 0: y

Estimated strength of the password: 50
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
[default] 0: y

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
[default] 0: y
```

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
[default] 0:send*
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
[default] 0:send*
```

All done!

```
vgusev2007@skillbox-vgusev2007:~$
[default] 0:send*
```