Roger Skyline 1

Установка Virtual Box

(http://rus-linux.net/MyLDP/vm/VirtualBox-networking.html)

(https://faist.ru/post/2017/06/29/Установка-Linux-Debian-в-VirtualBox)

Создание VM

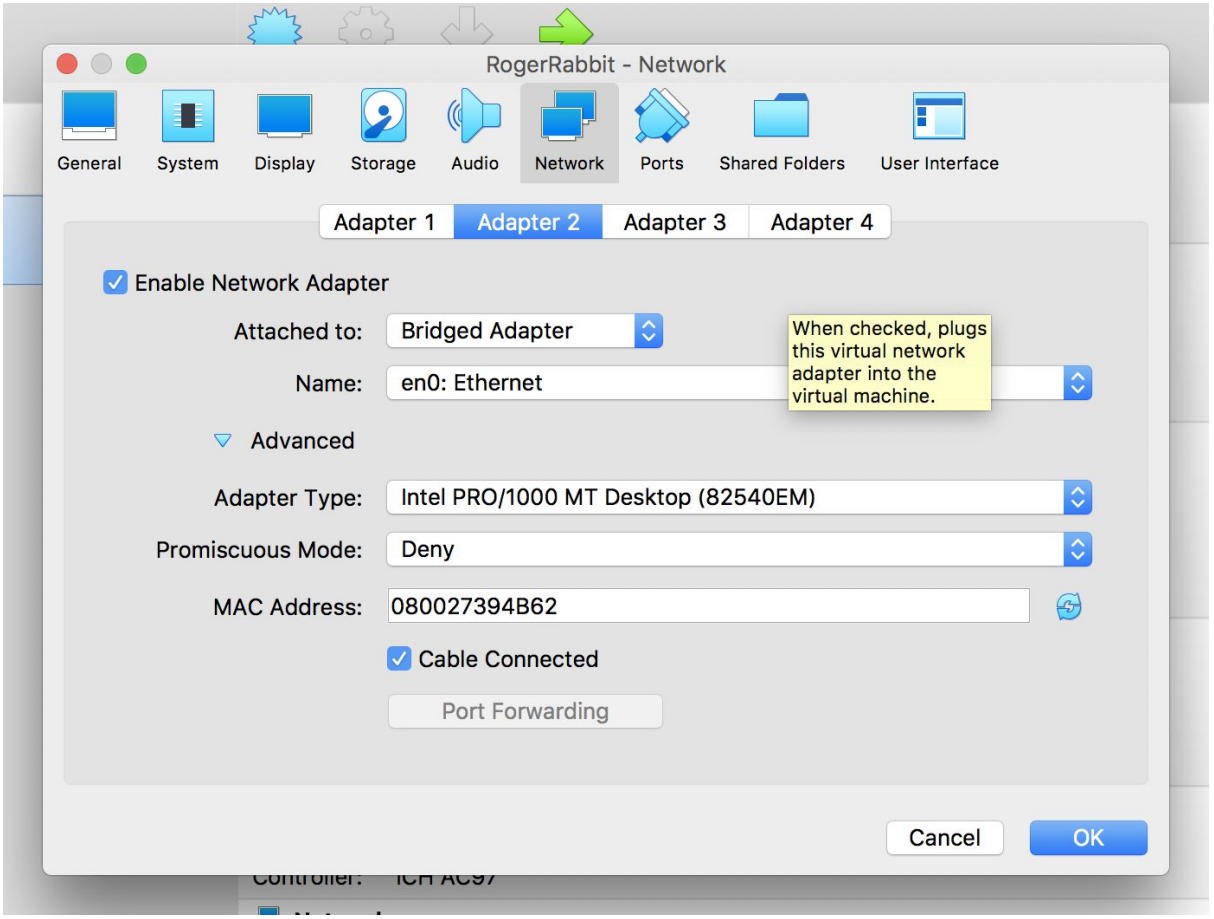
Machine Folder: /Volumes/Storage/goinfre/eharrag-

Type: Linux

Version: Debian (64-bit)

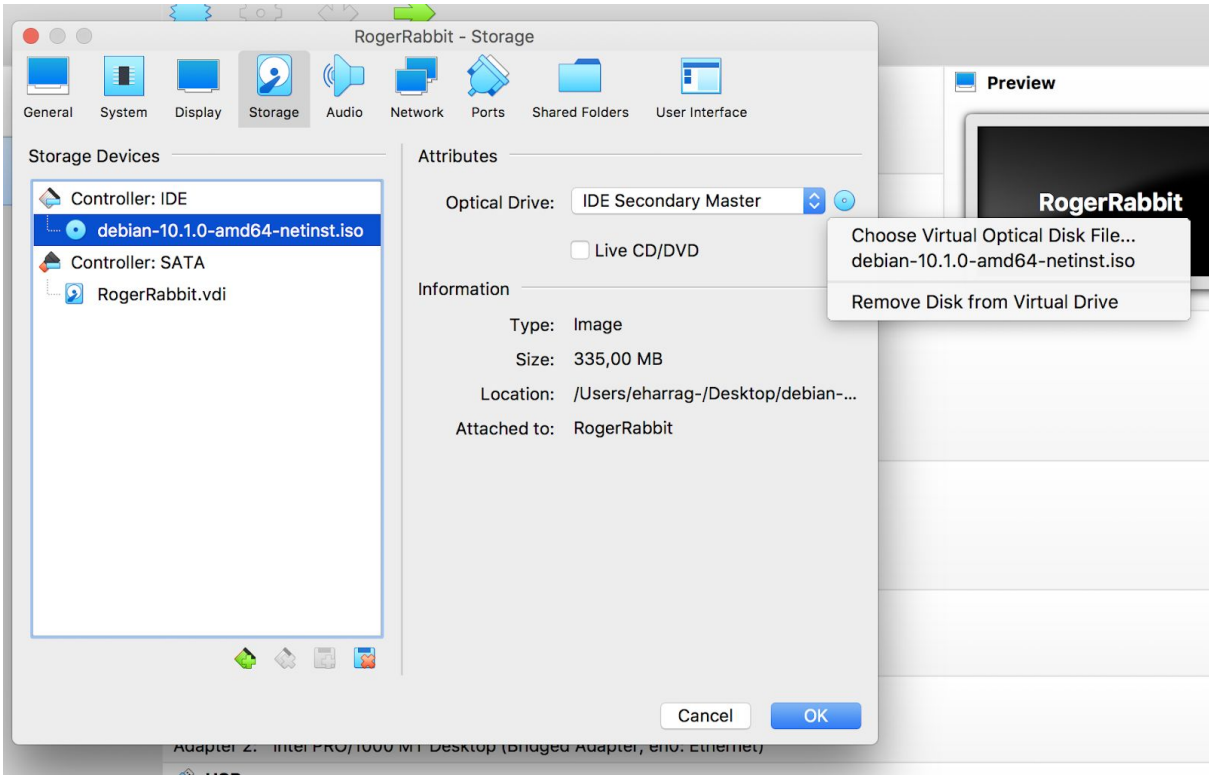
В настройках Network меняем Adapter 1 -> Attached to: `NAT` на `Bridged Adapter`.

Либо используем оба (Adapter 1 - NAT, Adapter 2 - Bridged).



Debian Version

Скачиваем с сайта Debian версию debian-10.1.0-amd64-netinst или debian-10.1.0-i386-xfce-CD-1. Выбираем в Storage -> Optical Drive.



Устанавливаем Debian

Region

Language : english

Zone : Russian Federation

Keyboard : American english

Admin

Hostname : debian

Domain : eharrag-

Password root : xxxxxx

Non-admin account

Full name : eharrag-

Username : eharrag-

Password : xxxx

Partition disks

Type : Manual 1st partition : 4.5 gb (либо же 4.6, чтобы получились нужные нам 4.2), primary, beggining, ext4, mounted on / (root)

2nd partition : 1gb, primary, beggining, swap

3rd partition : rest, primary, ext4, mounted on /home

Package manager

Country : Russian Federation

Mirror : ftp.ru.debian.org

Config

HTTP proxy : no

Software selection : Выбрать [SSH server] [Standard system utilities]

Instal the GRUB boot loader : yes

Подготовка

Connect to VM with previous login/password

```
$> su
$> apt-get update -y && apt-get upgrade -y
$> apt-get install sudo vim ssh openssh-client openssh-server nmap ufw iptables-persistent fail2ban apache2 portsentry sendmail sendmail-cf sendmail-bin
```

https://losst.ru/ustanovka-i-nastrojka-servera-apache ТУТ НАПИСАНО КАК ВКЛЮЧИТЬ ЕГО АВТОЗАГРУЗКУ https://losst.ru/nastrojka-fail2ban-centos-7 https://www.kubuntu.ru/node/5128 https://losst.ru/kak-uznat-mac-adres-v-linux

```
$> visudo
```

После sudo & %sudo добавить пользователя

```
eharrag-   ALL=(ALL:ALL) ALL (or NOPASSWD:ALL)
```

и/или

```
%eharrag-   ALL=(ALL:ALL) ALL (or NOPASSWD:ALL)
```

Output:

```
#

# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbi$

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
eharrag-    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
%eharrag-    ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

```
$> adduser eharrag- sudo [your_non-admin_user]
$> exit
```

Устанавливаем статический IP

1. Необходимо внести изменения в файл `/etc/network/interfaces` и задать “primary network”

```
$> cat vim /etc/network/interfaces
```

Output:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
#The loopback Network interface
auto lo
iface lo inet loopback
```

```
# Аутпут может быть “пустым”, как в примере. А может быть заполнен для enp0s3, enp0s8 и т.д.
# Если заполнен, то скорректировать “dhcp” на “static”. Если нет, то добавить ручками. Так как в настройках VM выбраны Adapter 1 - NAT, Adapter 2 - Bridged, то enp0s3 для NAT, enp0s8 для Bridged.
```

```
$> sudo ifconfig
```

Output:

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe1c:8959  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1c:89:59  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 590 (590.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 39  bytes 4942 (4.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.20.237  netmask 255.255.252.0  broadcast 192.168.23.255
        inet6 fe80::a00:27ff:feeb:7d97  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:eb:7d:97  txqueuelen 1000  (Ethernet)
        RX packets 8476  bytes 3902810 (3.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 268  bytes 37636 (36.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 10  bytes 474 (474.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 474 (474.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
$> sudo vim /etc/network/interfaces
```

```
ДОБАВИТЬ после имеющихся строк:

# The primary network interface
allow-hotplug enp0s8
auto enp0s8
iface enp0s8 inet static
address 192.168.20.237
```

```
netmask 255.255.255.252
gateway 192.168.20.1
```

The primary network interface - интерфейс для обычной сети
The second network interface - интерфейс для SSH
allow-hotplug enp0s8 - перезапуск интерфейса при падении
auto enp0s8 - запуск интерфейса при включении системы
netmask 255.255.255.252 (252 - то самое \30)
gateway 192.168.20.1
http://it-example.ru/сетевые-маски-2/
A CIDR /30 (or 1 static IP) - 255.255.255.252

```
$> sudo reboot
or
$> sudo service networking restart
```

Как проверить: IP вводим в адресную строку. Должна появиться страница Apache.

Настройка SSH

1. Под рутом вносим изменения в файл конфигурации SSHD.

```
$> sudo vim /etc/ssh/sshd_config
```

2. В строке 13 прописываем новый порт. Было “Port 22”.

   Стало “Port 22222”.

   Рекомендую ознакомиться с заметкой ниже:

Note: The Internet Assigned Numbers Authority (IANA) is responsible for the global coordination of the DNS Root, IP addressing, and other Internet protocol resources. It is good practice to follow their port assignment guidelines. Having said that, port numbers are divided into three ranges: Well Known Ports, Registered Ports, and Dynamic and/or Private Ports. The Well Known Ports are those from 0 through 1023 and SHOULD NOT be used. Registered Ports are those from 1024 through 49151 should also be avoided too. Dynamic and/or Private Ports are those from 49152 through 65535 and can be used. Though nothing is stopping you from using reserved port numbers, our suggestion may help avoid technical issues with port allocation in the future.

3. **На виртуальной машине!** Вносим изменения в файл конфигурации SSH.

```
$> sudo vim /etc/ssh/ssh_config
```

- В строке 32 меняем “yes” на “no”: `PermitRootLogin no`

- В строке 56 меняем “yes” на “no”: `PasswordAuthentication no`

Раскомментим обе строки (удалить символ # в начале каждой строки)

Установка доступа SSH с публичным ключом.

1. Для начала генерируем публичный и приватный rsa-ключи. **На локальном компьютере** вводим команду:

```
$> ssh-keygen -t rsa
```

Эта команда генерирует 2 файла `id_rsa` и `id_rsa.pub`

- **id_rsa:** Приватный ключ, храним в безопасности, любим, холим, лелеем.
- **id_rsa.pub:** Публичный ключ, отправляем на сервер (в виртуальную машину).

```
$> cat ~/.ssh/id_rsa.pub
$> cat ~/.ssh/authorized_keys
```

2. Для копирования ключа на сервер используем команду the ssh-copy-id command. (ssh-copy-id username@remote_host).

```
$> ssh-copy-id eharrag-@192.168.20.237 -p 22222
or
$> ssh-copy-id -i ~/.ssh/id_rsa.pub eharrag-@192.168.20.237 -p 22222
```

```
➜ ~ ssh-copy-id eharrag-@192.168.20.237 -p 22222
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/eharrag-/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
eharrag-@192.168.20.237's password:

Number of key(s) added:        1

Now try logging into the machine, with:   "ssh -p '22222' 'eharrag-@192.168.20.237'"
and check to make sure that only the key(s) you wanted were added.
```

Ключ автоматически добавляет в `~/.ssh/authorized_keys` на виртуальной машине.

3. Для сохранения изменений перезапускаем SSH.

```
$> sudo /etc/init.d/ssh restart
or
$> sudo service sshd restart
```

4. Теперь мы можем подключаться по SSH с использованием публичного ключа.

```
$> ssh eharrag-@192.168.20.237 -p 22222
```

Настройка Firewall

С использованием IPTABLES

sudo iptables -L - просмотр правил файрвола

Правила файрвола:

iptables -p INPUT drop - политика INPUT по дефолту на DROP

iptables -p input drop - политика FORWARD по дефолту на DROP

iptables -p OUTPUT drop - политика OUTPUT по дефолту на ACCEPT

sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT - разрешение только установленных соединений или связанных с установленными

sudo iptables -A INPUT -f -j DROP - сброс ферментированных пакетов

sudo iptables -A INPUT -p tcp -m recent --rcheck --seconds 60 --hitcount 2 --name scan --mask 255.255.255.255 --rsource -j DROP - подсчет и ограничение подключений по tcp для защиты от сканирования портов при 2 подключениях, которые не прошли по правилам

sudo iptables -A INPUT -p udp -m recent --rcheck --seconds 60 --hitcount 1 --name scan --mask 255.255.255.255 --rsource -j DROP - подсчет и ограничение подключений по ump для защиты от сканирования портов при 1 подключении, которое не прошло по правилам

sudo iptables -A INPUT -p tcp --dport 22222 -m recent --rcheck --seconds 200 --hitcount 5 --name scan_ssh --rsource -j DROP - подсчет и ограничение подключений по ssh

sudo iptables -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -m limit --limit 40/sec --limit-burst 40 -j ACCEPT - разрешение на подключение по http, https не больше 40 за секунду

sudo iptables -A INPUT -p tcp --dport 22222 -m recent --set --name scan_ssh --rsource - подсчет подключений по SSH

sudo iptables -A INPUT -p tcp -m multiport --dports 22222 -m conntrack --ctstate NEW -j ACCEPT - разрешение на подключение по ssh

sudo iptables -A INPUT -p tcp -m multiport --dports 80,443 -m connlimit --connlimit-above 10 --connlimit-mask 32 -j DROP - разрешение на 10 подключений с уникального ip

sudo iptables -A INPUT -p tcp --dport 22222 -m connlimit --connlimit-above 2 -j DROP - не больше двух подключений по SSH

sudo iptables -A INPUT -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s --limit-burst 2 -j DROP - OR RETURN защита от спама определенными типами пакетов

sudo iptables -A INPUT -p tcp ! --dport 5458 -m recent --set --name scan --rsource - подсчет подключений не по SSH

sudo iptables -A INPUT -p udp -m recent --set --name scan --rsource - подсчет подключений по udp

Либо же с использованием UFW.

https://community.vscale.io/hc/ru/community/posts/208348529-Настройка-фаервола-в-Ubuntu-с-помощью-утилиты-UFW?page=1#community_comment_208858805

ufw - упрощенный вариант манипулирования iptables.

---

1. Убедимся, что ufw - enable

```
$> sudo ufw status
```

Если нет, то:

```
$> sudo ufw enable
```

2. Устанавливаем правила:

   SSH : `sudo ufw allow 50683/tcp`

   HTTP : `sudo ufw allow 80/tcp`

   HTTPS : `sudo ufw allow 443`

Или:

- sudo ufw default deny incoming - выключить входящие запросы
- sudo ufw default allow outgoing - включить выходящие запросы
- sudo ufw allow 8822/tcp - открыть порт для ssh
- sudo ufw allow http - открыть порт для http.
- sudo ufw allow https - открыть порт для https

Настройка Fail2ban

Настройка Denial Of Service Attack с помощью fail2ban

https://www.dmosk.ru/instruktions.php?object=fail2ban

Пишет логи странных айпишников и добавляет нужные правила в iptables.

Посмотреть логи:

```
$> sudo vim /var/log/auth.log
```

Что происходит с fail2ban сейчас:

```
$> sudo tail -f /var/log/fail2ban.log
```

Вносим конфиги в:

```
$> sudo vim /etc/fail2ban/jail.conf
or
$> sudo vim /etc/fail2ban/jail.d/defaults-debian.conf
```

Добавляем:

```
[ssh-iptables]

enabled  = true
filter   = sshd
action   = iptables[name=SSH, port=ssh, protocol=tcp]
logpath  = /var/log/secure
maxretry = 3
```

Или

```
[sshd]
enabled = true
port    = 42
logpath = %(sshd_log)s
backend = %(sshd_backend)s
maxretry = 3
bantime = 600

#Add after HTTP servers:
[http-get-dos]
enabled = true
port = http,https
filter = http-get-dos
logpath = /var/log/apache2/access.log (доступ к файлу на веб-сервере)
maxretry = 300
findtime = 300
bantime = 600
action = iptables[name=HTTP, port=http, protocol=tcp]
```

(Опционально) Add http-get-dos filter

```
$> sudo cat /etc/fail2ban/filter.d/http-get-dos.conf
```

Output:

```
[Definition]
failregex = ^<HOST> -.*"(GET|POST).*
ignoreregex =
```

(Опционально) if you want to allow ping you can add the following lines in `/etc/ufw/before.rules`:

```
# Allow ping
-A ufw-before-output -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-output -p icmp --icmp-type source-quench -j ACCEPT
-A ufw-before-output -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-output -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-output -p icmp --icmp-type echo-request -j ACCEPT
```

В завершение перезагружаем Firewall и Fail2ban

```
Чтобы изменения вступили в силу:
$> sudo ufw reload
$> sudo service fail2ban restart
or
$> systemctl restart fail2ban
```

Защита сканирования портов

portsentry - утилита защищающая от сканирований

-l или --listening - посмотреть только прослушиваемые порты
-p или --program - показать имя программы и ее PID
-t или --tcp показать tcp порты
-u или --udp показать udp порты
-n или --numeric показывать ip адреса в числовом виде

https://www.youtube.com/watch?v=24M8dStXARg

http://aidalinux.ru/w/PortSentry_-_средство_противодействия_сканированию_портов

http://www.smeegul.kiev.ua/portsentry.html

https://www.lissyara.su/articles/freebsd/security/portsentry/kreilly

1. Вносим конфиги portsentry

```
$> sudo vim /etc/default/portsentry
```

Сначала меняем следующий строки

```
TCP_MODE="atcp"
UDP_MODE="audp"
```

После этого вносим изменения в `/etc/portsentry/portsentry.conf`

```
$> sudo vim /etc/default/portsentry
```

```
BLOCK_UDP="1"
BLOCK_TCP="1"
```

Комментируем (ставим в начале строки #) текущее KILL_ROUTE и раскомментим следующее:

```
KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP"
```

Комментируем (ставим в начале строки #) строку:

```
KILL_HOSTS_DENY="ALL: $TARGET$ : DENY
```

Добавляем в /etc/portsentry/portsentry.ignore.static:

```
$> sudo vim /etc/portsentry/portsentry.ignore.static
```

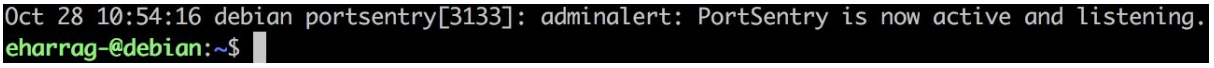
```
192.168.20.237/255.255.255.252
```

2. Перезагружаем сервис для сохранения изменений

```
$> sudo service portsentry restart
or
$> sudo /etc/init.d/portsentry restart
```

- В системных логах (/var/log/syslog) должно появиться:

debian portsentry[3133]: adminalert: PortSentry is now active and listening.



После данных манипуляций защита от сканирования установлена.

Пробуем сканить IP и порты с помощью nmap

```
$> nmap [IP_VM_TO_TEST] -p [PORT from 1 to 65535]
```

```
eharrag-@debian:~$ nmap 192.168.20.237 -p 2845
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-28 09:56 MSK
Nmap scan report for 217-67-187-54.in-addr.mastertelecom.ru (192.168.20.237)
Host is up (0.00010s latency).

PORT     STATE  SERVICE
2845/tcp closed bpcp-trap

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Nmap should not be able to scan your ports and IP trying to scan should be banned (see logs portsentry and iptables). State - closed!

Пробуем DOS attack с помощью SlowLoris

Скачиваем SlowLoris: https://github.com/nicolasvienot/rogerskyline/tree/master/SlowLoris

На локальном компьютере вводим команды:

```
$> perl ./slowloris.pl -dns 192.168.20.237

$> perl slowloris.pl -dns [STATIC IP VM] -port [SSH PORT VM]
$> perl slowloris.pl -dns [STATIC IP VM] -port 80
$> perl slowloris.pl -dns [STATIC IP VM] -port 443
$> perl slowloris.pl -dns [STATIC IP VM] -port 25
```

Slowloris should not be able to send any packets and IP should be banned from VM (see logs portsentry and iptables)

Пока работает Лорис, пробовать открыть сайт + netstat -n, смотреть подключения.

Стопим сервисы, в которых не нуждаемся

sudo service --status-all - проверка статусов всех сервисов
sudo systemctl disable application - убрать сервис из автозагрузки
service application stop - отключение сервиса

https://pingvinus.ru/note/services-systemctl

```
$> sudo systemctl disable console-setup.service
$> sudo systemctl disable keyboard-setup.service

Может быть еще
$> sudo systemctl disable apt-daily.timer
$> sudo systemctl disable apt-daily-upgrade.timer
$> sudo systemctl disable syslog.service
```

Scripts

Update Packages

Create a script that updates all the sources of package, then your packages and which logs the whole in a file named /var/log/update_script.log. Create a scheduled task for this script once a week at 4AM and every time the machine reboots.

Создаем файл `update_script.sh` и в нем указываем:

```
$> sudo vim update_script.sh
```

```
#!bin/bash

sudo apt-get update -y | sudo tee -a /var/log/update_script.log
sudo apt-get upgrade -y | sudo tee -a /var/log/update_script.log

exit 0
```

Monitor Crontab Changes

• Make a script to monitor changes of the /etc/crontab file and sends an email to root if it has been modified. Create a scheduled script task every day at midnight.

Создаем файл `cronMonitor.sh` и в нем указываем:

```
$> sudo vim cronMonitor.sh
```

* Почту указываем на свой вкус и цвет.

```
#!/bin/bash

cat /etc/crontab > ~/crontab_save/new

DIFF=$(diff ~/crontab_save/new ~/crontab_save/crontab)

if [ "$DIFF" != "" ]; then
        echo "Crontab has changed, sending mail!"
    sudo sendmail eharrag-@student.21-school.ru  < /etc/crontab
        rm ~/crontab_save/crontab
        mv ~/crontab_save/new ~/crontab_save/crontab
else
        echo "No changes on crontab!"
```

```
fi

exit 0
```

Добавляет задание в cron

```
$> sudo crontab -e
$> sudo vim /etc/crontab
```

https://crontab.guru - расшифровка звездочек

В открывшемся файле добавляем следующие строки:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
@reboot sudo ~/update.sh
0 4 * * 6 sudo ~/update.sh
0 0 * * * sudo ~/cronMonitor.sh
```

* **Проба пера (отправляем себе e-mail через терминал)**

Create a file email.txt with content

```
$> sendmail eharrag-@student.21-school.ru < /Users/eharrag-/email.txt
```

Web server && SSL

Deploy a Web application reachable on the machine IP's

Генерируем SLL-сертификат с помощью команды

```
$> sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

```
eharrag-@debian:/$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -subj "/C=FR/ST=IDF/O=42/OU=Project-roger
/CN=192.168.20.237" -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
Generating a RSA private key
...........+++++
.....................................+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
eharrag-@debian:/$
```

Прописываем конфиги в `/etc/apache2/conf-available/ssl-params.conf` :

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLHonorCipherOrder On

Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff

SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"

SSLSessionTickets Off
```

Вносим изменения в `/etc/apache2/sites-available/default-ssl.conf` :

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin eharrag-@student.21-school.ru
                ServerName      192.168.20.237

                DocumentRoot /var/www/html

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                SSLEngine on

                SSLCertificateFile          /etc/ssl/certs/apache-selfsigned.crt
                SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

                <FilesMatch "\.(cgi|shtml|phtml|php)$">
                                SSLOptions +StdEnvVars
                </FilesMatch>
                <Directory /usr/lib/cgi-bin>
                                SSLOptions +StdEnvVars
                </Directory>

        </VirtualHost>
</IfModule>
```

Добавляем в `/etc/apache2/sites-available/000-default.conf` соответствующие строки:

```
<VirtualHost *:80>

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        Redirect "/" "https://192.168.20.237/"

        ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Запускаем новые конфиги следующими командами:

```
$> sudo a2enmod ssl
$> sudo a2enmod headers
$> sudo a2ensite default-ssl
$> sudo a2enconf ssl-params
По просьбе команд вводим:
$> systemctl reload apache2
или
$> systemctl restart apache2
```

Содержание сайта можно изменить в /var/www/index

```
$> sudo vim /var/www/index/index.html
```

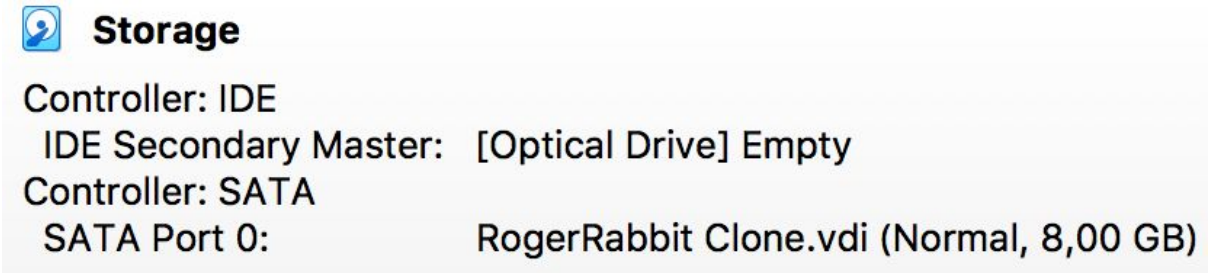
> “Что посеешь, то и пожнешь.”

Submission and peer-evaluation

Make your work on your repo as usual. For obvious reasons you do not return your virtual machine but a checksum of your disk image. You can do this using a command like: shasum < disk.vdi. Keep your disk image somewhere so that you can make your peer-evaluation.

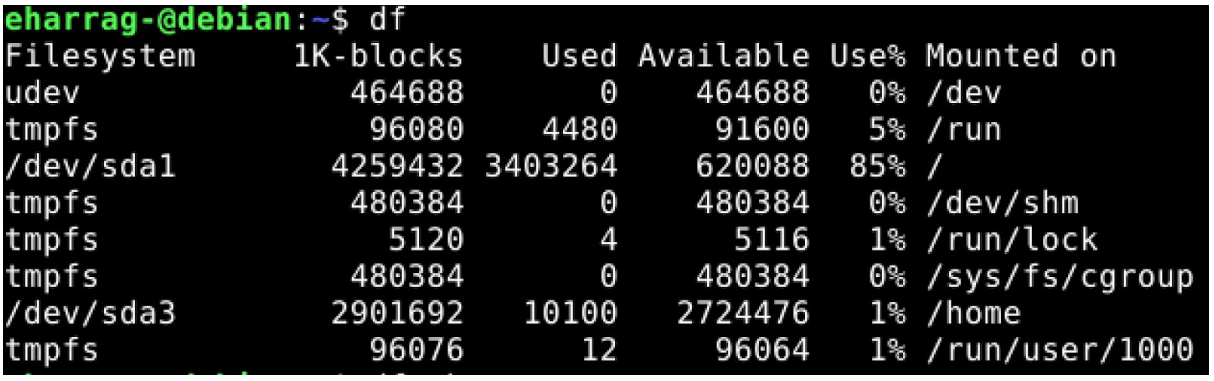
Checklist

1.




```
$> df
```

https://system-admins.ru/komandy-dlya-proverki-razmera-diska-v-linux/

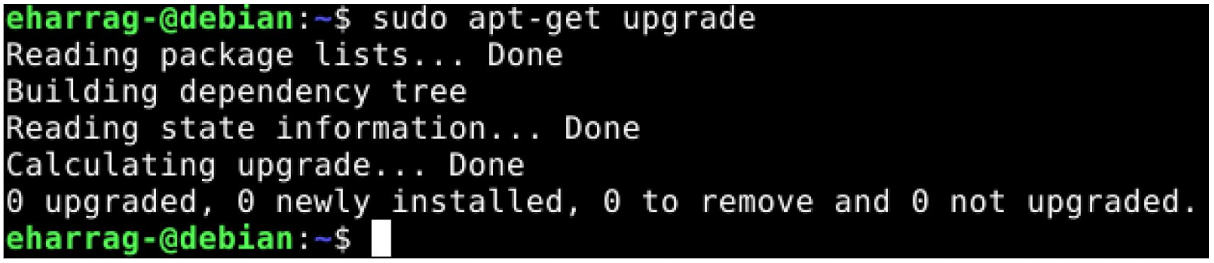


To make sure that our versions are up-to-date, let's update and upgrade the system with apt-get:

```
$> sudo apt-get update -y && sudo apt-get upgrade -y
```

The -y flag will confirm that we are agreeing for all items to be installed, but depending on your version of Linux, you may need to confirm additional

prompts as your system updates and upgrades.




```
$> apt list --installed
```

2.

```
$> sudo adduser [USER]
$> sudo adduser [USER] sudo

$> visudo
```

После sudo & %sudo добавить:

[USER] ALL=(ALL:ALL) ALL (or NOPASSWD:ALL)

and

[USER] ALL=(ALL:ALL) ALL (or NOPASSWD:ALL)

С виртуальной машины:

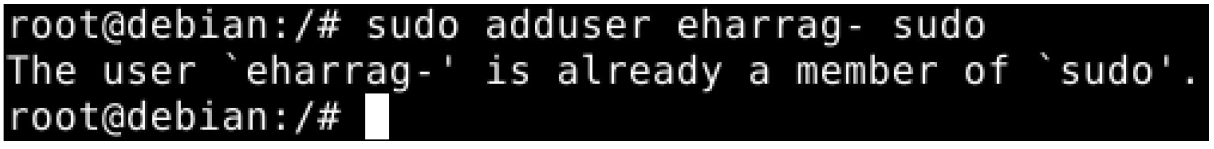
```
$> sudo vim /etc/ssh/sshd_config
```

- New line 13 - Port 22222 лучше выбирать от 46000
- Edit line 32 like: `PermitRootLogin no`
- Edit line 56 like: `PasswordAuthentication no`

```
$> sudo /etc/init.d/ssh restart
или
$> sudo service sshd restart
```

С локальной:

```
$> ssh-copy-id [USER]@192.168.20.237 -p 22222
или
$> ssh-copy-id -i ~/.ssh/id_rsa.pub [USER]@192.168.20.237 -p 22222
$> ssh [USER]@192.168.20.237 -p 22222
```




```
$> sudo vim /etc/network/interfaces
```

Output

```
# The primary network interface
allow-hotplug enp0s8
auto enp0s8
iface enp0s8 inet static
address 192.168.20.237
netmask 255.255.255.252
gateway 192.168.20.1
```

The primary network interface - интерфейс для обычной сети
The second network interface - интерфейс для SSH
allow-hotplug enp0s3 - перезапуск интерфейса при падении
auto enp0s3 - запуск интерфейса при включении системы
netmask 255.255.255.252 (252 - то самое \30)

```
$> sudo iptables -L
```

```
$> sudo fail2ban -V - проверка установленного ф2б
```

```
$> sudo netstat -ntulp - открытые порты
```

https://losst.ru/kak-posmotret-otkrytye-porty-v-linux

```
$> sudo service --status-all - активные сервисы
```

Scripts:

```
$> sudo vim update_script.sh
```

```
#!bin/bash

sudo apt-get update -y | sudo tee -a /var/log/update_script.log
sudo apt-get upgrade -y | sudo tee -a /var/log/update_script.log

exit 0
```

```
 $> sudo vim cronMonitor.sh
```

```
#!/bin/bash

cat /etc/crontab > ~/crontab_save/new

DIFF=$(diff ~/crontab_save/new ~/crontab_save/crontab)

if [ "$DIFF" != "" ]; then
        echo "Crontab has changed, sending mail!"
    sudo sendmail eharrag-@student.21-school.ru  < /etc/crontab
        rm ~/crontab_save/crontab
        mv ~/crontab_save/new ~/crontab_save/crontab
else
```

```
        echo "No changes on crontab!"
fi

exit 0
```

Task to cron

```
$> sudo vim /etc/crontab
```

https://crontab.guru - расшифровка звездочек

В открывшемся файле добавить строки:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
@reboot sudo ~/update.sh
0 4 * * 6 sudo ~/update.sh
0 0 * * * sudo ~/cronMonitor.sh
```

SSL

https://losst.ru/sozdanie-sertifikata-openssl

Создание самоподписанного сертификата openssl из существующего закрытого ключа:

```
$> sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Опция -new говорит, что нужно запросить информацию о csr у пользователя. Чтобы браузер доверял ключу нужно этот же сертификат импортировать в список доверенных.

Apache web server package

https://losst.ru/ustanovka-i-nastrojka-servera-apache

Deployment Part

http://192.168.20.237/

```
$> sudo vim /var/www/index/index.html
```