\*\*\*CHROOT (недостаточная изолция)

// создаем папку

mkdir GB

//вводим команду смены ченч рут

//где GB папка,

//а /bin/bash интерпретатор кот мы хотим использовать

sudo chroot GB /bin/bash

\*\*\*\*\*\* получили ошибку так как нет исполняемого файла в папке назначения

// создаем папку

mkdir GB/bin

//копируем исполняемый файл в GB/bin

cp /bin/bash GB/bin

//вводим команду смены ченч рут

//где GB папка,

//а /bin/bash интерпретатор кот мы хотим использовать

sudo chroot GB /bin/bash

\*\*\*\*\*\* получили ошибку так как нет библиотек

//смотрим зависимости командой

ldd /bin/bash

//копируем все зависимости

mkdir GB/lib

mkdir GB/lib64

cp /lib/x86\_64-linux-gnu/libtinfo.so.6 GB/lib

cp /lib/x86\_64-linux-gnu/libc.so GB/lib

cp /lib/x86\_64-linux-gnu/libc.so.6 GB/lib

cp /lib64/ld-linux-x86-64.so.2 GB/lib64/

sudo chroot GB /bin/bash

\*\*\*ГОТОВО!\*\*\*

\*\*\*копирование ls\*\*\*

whereis ls - смотрим где исполняемый файл

mkdir ./GB/usr - создаем папку

mkdir ./GB/usr/bin - создаем папку

cp /usr/bin/ls ./GB/usr/bin/ - копируем файл по нужному пути

ldd /usr/bin/ls -смотрим зависимости

cp /lib/x86\_64-linux-gnu/libselinux.so.1 ./gb/lib/ -копируем зависимости

cp /lib/x86\_64-linux-gnu/libc.so.6 ./gb/lib/ -копируем зависимости

cp /lib/x86\_64-linux-gnu/libpcre2-8.so.0 ./gb/lib/ -копируем зависимости

cp /lib64/ld-linux-x86-64.so.2 ./gb/lib64/ -копируем зависимости

sudo chroot GB /bin/bash -заходим в chroot

ls -l -смотрим

\*\*\*ГОТОВО!\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*ПРОСТРАНСТВО ИМЕН\*\*\*\*\*\*

//механизм изоляции процссов друг от друга//

PID изоляция идентификатора процессов (уникальные номера процессов)

Network сетевая изоляция

User изоляция пользователей и групп

Mount переопределение точек монтирования

IPC(Inter Process Communication)изоляция межпроцесного взаимодействия

UTS изоляция именя хоста и доменного имени

\*\*\*PID

//посмотреть максимальное количество PID

cat /proc/sys/kernel/pid\_max

//показать запущенные процессы

top

\*\*\*NET

//просмотр лупбэк запущенной программы

strace ./program

\*\*\*MOUNT

\*\*\*UID (User Identifier)

\*\*\*UTS (Unix Time-Sharing)

uname -> Linux

uname -a -> вся информация включая версию ядра

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*РАБОТА С ИЗОЛЯЦИЕЙ IP\*\*\*

ip a - просмотр всех сетевых соединений

sudo ip netns add test1 -создание нового неймспейса test1 (для сетевого интерфейса в данном случае)

sudo ip netns exec test1 bash -изоляция неймспейса и вход в него

ip a -просмотр изолированных соединений

sudo ip link add veth0 type veth peer name veth1 -создание двух связанных сетевых интерфейсов

ip a -их просмотр

sudo ip link set veth1 netns test1 -добавление сет.интерфейса в неймспейс test1

sudo ip addr add 10.0.0.1/24 dev veth0 -добавление ip адреса сетевому интерфейсу veth0

sudo ip link set dev veth0 up -поднять сетевой интерфейс (включить)

sudo ip netns exec test1 bash -входим в неймспейс

sudo ip addr add 10.0.0.2/24 dev veth1 -добавляяем ip адрес сетевому интерфейсу

sudo ip link set dev veth1 up -поднимаем интерфейс

ping 10.0.0.1 -пингуем сетевой интерфейс в неймспейсе

//компиляция в исполняемый файл

gcc example.c -o example.sh

//запуск скомпелированного файла

./example.sh