

Лабораторная работа №1 «Шифрование сообщений с помощью средств GNU Privacy Guard»

Был установлен пакет GnuPG

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --version
gpg (GnuPG) 2.2.4
libgcrypt 1.8.1
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Home: /home/dumpling/.gnupg

Поддерживаются следующие алгоритмы:

С открытым ключом: RSA, ELG, DSA, ECDH, ECDSA, EDDSA

Симметричные шифры: IDEA, 3DES, CAST5, BLOWFISH,
AES, AES192, AES256, TWOFISH, CAMELLIA128,
CAMELLIA192, CAMELLIA256

Хеш-функции: SHA1, RIPEMD160, SHA256, SHA384, SHA512,
SHA224

Алгоритмы сжатия: Без сжатия, ZIP, ZLIB,
BZIP2

Был создан новый ключ

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --gen-key
```

```
. . .
gpg: ключ 33***073D2886C8 помечен как абсолютно доверенный
gpg: сертификат отзыва записан в
'/home/dumpling/.gnupg/openpgp-revocs.d/571536F562A**73D2886C8.rev'.
открытый и секретный ключи созданы и подписаны.
```

```
pub   rsa3072 2020-09-05 [SC] [годен до: 2022-09-05]
       57153**8F6E5C5833D11F073D2886C8
uid           Ksenia Rogova <ksenia.rogova99@gmail.com>
sub   rsa3072 2020-09-05 [E] [годен до: 2022-09-05]
```

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --list-keys
/home/dumpling/.gnupg/pubring.kbx
```

```
-----
pub   rsa1024 2019-07-08 [SC]
       5522**4C17AF8EBEE7BC0FBBBD38682
uid           [ абсолютно ] Ksenia Rogova <ksenia.rogova99@gmail.com>
sub   rsa1024 2019-07-08 [E]

pub   rsa1024 2019-07-09 [SC]
       E9E9***15AC85F7072E451E7DBB0D16
uid           [ абсолютно ] Ksenia Rogova <ksenia.rogova99@gmail.com>
sub   rsa1024 2019-07-09 [E]

pub   rsa3072 2020-09-05 [SC] [годен до: 2022-09-05]
       571536***8F6E5C5833D11F073D2886C8
uid           [ абсолютно ] Ksenia Rogova <ksenia.rogova99@gmail.com>
sub   rsa3072 2020-09-05 [E] [годен до: 2022-09-05]
```

Данный ключ был успешно удален

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --delete-secret-keys 33**3D2886C8
```

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --delete-keys 33**73D2886C8
```

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --list-keys
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 2 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 2u
/home/dumpling/.gnupg/pubring.kbx
-----
pub   rsa1024 2019-07-08 [SC]
      55**4C17AF8EBEE7BC0FBBD38682
uid           [ абсолютно ] Ksenia Rogova <ksenia.rogova99@gmail.com>
sub   rsa1024 2019-07-08 [E]

pub   rsa1024 2019-07-09 [SC]
      E9E**215AC85F7072E451E7DBB0D16
uid           [ абсолютно ] Ksenia Rogova <ksenia.rogova99@gmail.com>
sub   rsa1024 2019-07-09 [E]
```

Еще раз создадим ключ и проанализируем папку ~/.gnupg:

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ ll ~/.gnupg/
итого 44
drwx-----  4 dumpling dumpling  4096 сен  5 21:12 ./
drwxr-xr-x 119 dumpling dumpling 12288 сен  5 17:48 ../
drwx-----  2 dumpling dumpling  4096 сен  5 21:12 openpgp-revocs.d/
drwx-----  2 dumpling dumpling  4096 сен  5 21:12 private-keys-v1.d/
-rw-r--r--  1 dumpling dumpling  5756 сен  5 21:12 pubring.kbx
-rw-r--r--  1 dumpling dumpling  3805 сен  5 21:05 pubring.kbx~
-rw-r-----  1 dumpling dumpling   676 сен  5 21:05 sshcontrol
-rw-----  1 dumpling dumpling  1480 сен  5 21:12 trustdb.gpg
```

В каталоге openpgp-revocs.d/ записываются сертификаты отзыва. Сертификаты отзыва можно назвать системой безопасности и страховкой — если парольная фраза была забыта/утеряна или она была скомпрометирована, пользователь может опубликовать сертификат отзыва, чтобы проинформировать других пользователей, что данный открытый ключ более недействителен.

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ ll ~/.gnupg/openpgp-revocs.d/
итого 24
drwx-----  2 dumpling dumpling  4096 сен  5 21:12 ./
drwx-----  4 dumpling dumpling  4096 сен  5 21:12 ../
-rw-----  1 dumpling dumpling  1732 июл  9 2019
552241**4C17AF8EBEE7BC0FBBD38682.rev
-rw-----  1 dumpling dumpling  2109 сен  5 20:58
5715**8F6E5C5833D11F073D2886C8.rev # созданный и удаленный ключ в данной
лабораторной
-rw-----  1 dumpling dumpling  2109 сен  5 21:12 A8**B6480028ED8C6F589.rev #
созданный ключ для данной лабораторной
-rw-----  1 dumpling dumpling  1732 июл  9 2019
E9E***5AC85F7072E451E7DBB0D16.rev
```

каталог `private-keys-v1.d/` предназначается для хранения секретных ключей. Каждый ключ хранится в файле с именем, составленным из кода ключа и суффикса `key`. Для всех файлов в этом каталоге следует проводить резервное копирование, тщательнейшим образом сохраняя резервные копии в надежном месте.

```
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ ll ~/.gnupg/private-keys-
v1.d/
итого 32
drwx----- 2 dumpling dumpling 4096 сен  5 21:12 ./
drwx----- 4 dumpling dumpling 4096 сен  5 21:12 ../
-rw----- 1 dumpling dumpling 1607 сен  5 21:12
3B54**288260A7E9BE02591C8E1BB588.key
-rw----- 1 dumpling dumpling  710 июл  9  2019
3E94**914B0720A841CD734183F0C.key
-rw----- 1 dumpling dumpling 1623 сен  5 21:12
8889**8F5D728CFF3C3843FF4408.key
-rw----- 1 dumpling dumpling  710 июл  9  2019
928FA***113EEB75C5EED4ED2AB081E19A.key
-rw----- 1 dumpling dumpling  710 июл  9  2019
F56****D3290867C7916E3EB50CED776FA4.key
-rw----- 1 dumpling dumpling  710 июл  9  2019
FE9912A7***117650EC68CD8D87F.key
```

Файл `sshcontrol` используется, когда включена поддержка протокола SSH (см. параметр `--enable-ssh-support`). В протоколе SSH используются только присутствующие в этом файле ключи. Для этого файла следует проводить резервное копирование.

Для добавления в этот файл новых записей можно применять средство `ssh-add`; их можно также добавлять вручную. Строки комментария, на которые указывает знак «`#`» в начале, а также пустые строки игнорируются. Запись начинается с необязательных пробелов с последующим кодом ключа из 40 шестнадцатеричных цифр с последующим необязательным сроком действия буфера в секундах и другим необязательным полем для произвольных признаков. Ненулевой срок действия отменяет глобальное исходное значение, задаваемое с помощью `--default-cache-ttl-ssh`.

Поддерживается только один признак, `confirm`. Если этот признак установлен, при каждом использовании ключа будет вызываться программа ввода пароля для подтверждения. Признак автоматически устанавливается, если новый ключ загружается в `gpg-agent` с помощью параметра `-c` команды `ssh-add`.

Создадим файл для кодирования-декодирования:

```
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ vim original.txt
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ cat original.txt
There is some text. And it's going to be encoded
(and probably decoded back)
```

So...

Hello World!

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 -a -r
A8A****A216AB6480028ED8C6F589 -e original.txt
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ cat original.txt.asc
-----BEGIN PGP MESSAGE-----
```

```
hQGMAzWemME/VFnIAQv/X0iBWy5BE18vm1fz6tTxwNxp7Zh10eCkNxEy9Gp+qACG
ejwHTGgMI/cLTEDxsCFN6px0iJL03o+xD0hwsB6gWcUUXKzvgJMdbITvL8053bGb
AUteCVohIzV5pwmnYB2W4qT6geNdU9HxzBLJByPGfoMrr2MuYfsbfglAh+J0m+T6
XMV+ionHEBCTHL2eaXn43L2Hjg2X1X73WDS6LG5HiN+mYLsaPzT3bDwPRpKesHhZ
mnJSyNDPkbbczuY/wLxxzCRkxCerwkfMEy3aA0F8UPQCYlczERRjVVL/rRAmD1hI
u+9aEWNmT6gje****0YhZXjA++ajFEy0MeZr9/dcIKHfMy23fSskoZuG/3nzsVLw
30vEu6oeaEAQhQw07dSLjgPV09yYVi3Y7ZtaymynSykYDB0ofU3Dnr4WnmQxAUM0
bYWafaqJn5R6+1fY5ho3qIIMxBgcB4Fqu8oP5r0t/G93lNmTlm5wu5CjYEieQj1z
3sAbJwzxxERYA++MZvmL0p4B4fdeT8edybLkiCJDg7N0anOLByk98SxVS2XYR60p
4XLSFJeLfFoyiys0R3SiDHxudUiWF60qFihkpgy81cLin3RURGM8zu8WVJDqRhwm
eoF1PB9KpFMdC3niosKT9ZvzJiKJq0FJktxBmj1L3tLevtW2Ql1Us23WgwMn4BZl
2xQVnssRrEdttdHc1E5AtUE4MxofKao2msDLUsz/qQ==
=GCjz
```

-----END PGP MESSAGE-----

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 -r
A8A2B0****B7A216AB6480028ED8C6F589 -d original.txt.asc > decrypted.txt
gpg: зашифровано 3072-битным ключом RSA с идентификатором 359E9**459C8,
созданным 2020-09-05
```

"Ksenia Rogova <ksenia.rogova99@gmail.com>"

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ cat decrypted.txt
There is some text. And it's going to be encoded
(and probably decoded back)
```

So...

Hello World!

Подпишем:

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ cp original.txt
original2.txt
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 -r CC84***5BFE6A --
clearsign original2.txt
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ cat original2.txt.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512
```

There is some text. And it's going to be encoded
(and probably decoded back)

So...

Hello World!

-----BEGIN PGP SIGNATURE-----

```
iQGzBAEBCgAdFiEEqKKwGW2n9Ti3oharZiACjtjG9YkFAl/AHF8ACgkQZiACjtjG
9YkTNAwA*****7C1gXitqAc2CajBsfcRGJUVtdh8uSy0EtvH30H0Z+LMvfTMIA2A
fcafU00h2/Us/M7DUWDRuFWMv8XMEcimgl+HHNBmZ67ERChARSzo1/8Y/au5CLR7
WgMQiqkNU7HVMhm4JRYfeUIM9CPpo3z+2gwQ0lJhZHLN0cAPdWEJ24z/0kQVpqsI
cr35PlGJwRDDxepTZD4jE8CCpeAnF/R2fDDL8yrFaDH8qQrG0D2oLu3ZGNkDrpQ
xyPNKESJ40qztUc9QiNzF6BhoppSsmBm1hDjpDcLtrpozIayHQ0a7YEA9EKdEAjZ
geb8J7qJ+s*****xdVrzqiIXGjH7Dwn8tL0aKyveNM+awUNxOWsAV8Hz4C6nUxm1
```

```
dLtggdBvo/jS08fal+hp4cuI9fBvJ0fiUEb4KYIInt0+eL8F6Cvmz2WhMNsZdmfK
X+1AlmPr1YVHGM20g9/QPURdp+bDkqakpvIZNNQ9IWFNE12iOMZa5j5mcybv2Zkg
2PYlu3NO
=ZtfK
-----END PGP SIGNATURE-----
```

Лабораторная работа №2 «Импорт и экспорт ключей. Цифровая подпись»

Экспортируем открытый ключ в файл:

```
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ vim mydoc.txt
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --export --armor
CC847734955BFE6A > mykey.asc
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ cat mykey.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
mQGNBF/AGegBDACZEHC0jSR+AKzLDjYBnJ3qyiEzsGBdVfK/PUT1PT874rIgxQ+B
L2rtdGfQlnLQNO+AzzXNn8JgUTm/0qbgj+ekoIEq/urp50FGMLi6wpuFJ6ISXuQj
YJqP/Zl74phipPx4z0Bu1TM83oZDG2wSj20ISmb1JINv6VFRDSuAF1pNRkuhrVtB
ob8hJ*****qUWiTwPY3DF4ddAH9wsM07omBk18iHWeDu8QguOmIgvkotL1CLdzYs
hyKdWfwj7Cma9uiAZ1oFyhEkCYIhECrEkzGs5RGW8zfYsYEmQd6DliUhl8UBSpQs
Q+EmQyWVX7j6SLY6is0aekMb7aDjr28boueDwOHLyJTe1pslPjTCh4YOH7nQf
PudnZtvKJFgjHooCez4XP5iMxfMVT+oHRJ/fG7R7a/vKNsag3vqtUDM1hg9mhApz
/s0hEkms4D/j1dAK4AARM0Ug0h8PgTR5/aCP3m1320nwbDVYrJv+eibxCJdRQCRC
pAeeT36tA0Ju9mMAEQEAAbQpS3NlbnlhIFJvZ292YSA8a3NlbnlhLmR1bXBsaW5n
QGdtYWlsLmNvbT6JAdQEEwEKAD4WIQT3Dwni0r9LHD8gwHbMhHc0lvv+agUCX8AZ
6AibAwUJA8JnAAULCQgHAgYVCgkICwIEFgIDAQIeAQIXgAAKCRDMhHc0lvv+ahN
...
```

Создадим отсоединённую ЭЦП файла mydoc.txt в текстовом формате:

```
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --sign --detach-sign
--default-key CC847734955BFE6A --armor mydoc.txt
gpg: "CC84***955BFE6A" используется в качестве основного секретного ключа для
подписи
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ cat mydoc.txt
Hello, World!
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ cat mydoc.txt.asc
-----BEGIN PGP SIGNATURE-----
```

```
iQGzBAABCgAdFiEE9w8J4jq/Sxw/IMB2zIR3NJVb/moFAl/AITsACgkQzIR3NJVb
/mpwqAv/***tMsDscSdC7HVTWNNqVrHXUboE9afIXJc2ztmYqUn9bUy0ehbFJlc6
4hwoobqlwmKhrzETyrJxje3gKkF8wzUuuZ0eAVPGksBg/o51AIwsRNlqrzZDm2b6
gGEV/0lJB005R0v6zBI0FJPfmmwN6G3/15rcBFM9Dp/mF3S6nKZkb6t174558v30C
BsPsACDtttooilA3Tj4W47DfXqbHqxmNMg0ZoJtbMV3lC9dp35ICCj6vh7yf39az
PUQY06Vbd2qaHHVB6zDXA2wpJj8frs753haS0s8HC+TPAmDL/CGji3G4EK00Ly9P
hGsHJDULXJ00QY4ah7G8m9z+bTJhpaNtGP/3u65Ei0s/zwFs/gCaaRC25TCQRL3
Syfu6WP00uMwf*f5Al6Sn5p0m0W3XURDIeVH8npGBnVD127pVXHy332peh3xGcr
+UVkVA+RFZiBUHhr743dqcj4ka0qmffrYho61bK1Lp6TNHTA0cD79sGCrcp8t7H
VvZ/4Yjq
=ktpc
-----END PGP SIGNATURE-----
```

Создадим отсоединённую подпись в двоичном формате:

```
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --sign --detach-sign
--default-key CC8***34955BFE6A mydoc.txt
gpg: "CC8***4955BFE6A" используется в качестве основного секретного ключа для
подписи
dumppling@thinkpad-1380:~/Unix-Labs/network_security$ cat mydoc.txt
```

```

mydoc.txt      mydoc.txt.asc  mydoc.txt.sig
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ cat mydoc.txt.sig
0#0###
##!#0 0:0K#? 0vW40[0j#_0"
      #w40[0j#0
          0D0[h0u000N0Yp00##0&g00070000'#0:0
0p#00000r0##00#0C000V000000nXi00nz00,a000#090050Q<0Mc0s0##0#ST0x}f0#0000接t
0$00e0~v;000#YS00/0Uİ00061#0RB00#rp00000g樓p1!0000g0"x00#0B=0#0\
_c0Wj}0##00000#00.00#0<p6000=#H0#R#a#$00000?
0W#r#00t00v0##D0m#000#0#&0000a0Ko0'8s0~00]00050J#0#0u000dw0Er%20=00"0ne70P0#000\
0_00_0g00#0#V010j0U0*00P0q0>e80U0N6#0d#00N0i000w0E{00?Z0#d

```

Создадим встроенную в файл подпись в текстовом формате:

```

dumpling@thinkpad-l380:~/Unix-Labs/network_security$ gpg2 --sign --default-key
CC8***4955BFE6A --armor mydoc.txt
gpg: "CC8***4955BFE6A" используется в качестве основного секретного ключа для
подписи
Файл 'mydoc.txt.asc' существует. Записать поверх? (y/N) y
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ cat mydoc.txt.asc
-----BEGIN PGP MESSAGE-----

```

```

owEB5AEb/pANAwAKAcYEdzSVW/5qAawdYglteWRvYy50eHRfwCKFSGVsbG8sIFdv
cmxkIQqJAbMEAAEKAB0WIQT3DwniOr9LHD8gwHbMhHc0lVv+agUCX8AihQAKCRDM
hHc0lVv+anuL***lGKty/cF6BERqHUete6guxqOMfxMYBPcX5g6LAuC+u3kh5C3c
5AAQ6oC1Chjt0Hu13U8w2xaZf0XLPAr9lhtK1nd2rlGMAJHB/yUeV4JhJXG/oAj+
Nv+faOr2jbOprMJ1UzD9bsiQawzWi3E/TzmZd0eeha1sj/TzQF+WlPGi/www/eIi
TVt4L2fR9kzwT5i8dgmptwARuQHclJyhU/m0BG8fp08lNRKf1pQsE7G3BgqvWU1P
RLWDX0Ro/p7V8X006SoQv4GM5rCxXnttAPSJ/NyoCg1vr0Doz9zMVyAqFNKS43oB
HLUoQMkain/k+3EZphxH8qzchtGaedeT8volQzcelWeXsLX9EPIATtdWP+Lo70Y/
LvsoVq/Z2KC/bAUm7zUgc844BuBNNqxUww077ZbUcL/+wc02QREmBU4HoAvSgfMb
cxGn9rEgjkaI1XU1LzaU0+wDNgXjETb7YxNLHZXSnhjFfmP/cIMRcKaATBw90uh1
Tsa1BLElklzRAMBk=
=1k1Y
-----END PGP MESSAGE-----

```

Создадим встроенную в файл подпись в двоичном формате:

```

dumpling@thinkpad-l380:~/Unix-Labs/network_security$ gpg2 --sign --default-key
CC847734955BFE6A mydoc.txt
gpg: "CC8***4955BFE6A" используется в качестве основного секретного ключа для
подписи
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ cat mydoc.txt.gpg
###0###00
#w40[0j#0#b mydoc.txt_0"0Hello, World!
0#0###
##!#0 0:0K#? 0vW40[0j#_0"0
      #w40[0j#a
          0,f000^00#}0
              0]/&00000
V8##00*:00#]M0S000#09#0*L00uE0~000H0
000V>n00m00#005 0000ce0B$*`#0a0#0#00]hQD00emL0r0##0KA0V0#0#0断4f)00?ahmL#0H0[
0|T0P2#0s0"]k0~00Z#0|Rf#)|000D0h00L000
#NR90&00yr1K000##0U0000A0B00TMv0|"#c000q00[N;#$000;G00
%p00:.WXX000400~000U00rh]00#0iYG000I~000}NY%}'=000#0P00      0000#0      0H#0000!
o00fY0

```

При создании встроенных подписей содержимое файла-источника целиком включается внутрь, поэтому использовать данный формат не желательно из-за

дублирования и значительного размера. Поэтому отсоединённая ЭЦП является самым популярным вариантом подписи.

Импортируем открытый ключ из файла:

```
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ gpg2 --import mykey.asc
gpg: ключ CC84***34955BFE6A: "Ksenia Rogova <ksenia.dumpling@gmail.com>" не
изменен
gpg: Всего обработано: 1
gpg: неизмененных: 1
```

Установим доверие импортированному ключу, т.к. в противном случае не сможем проверить подпись:

```
dumpling@thinkpad-l380:~/Unix-Labs/network_security$ gpg2 --edit-key
CC8***4955BFE6A
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Секретный ключ доступен.

```
sec  rsa3072/CC8***955BFE6A
      создан: 2020-11-26  годен до: 2022-11-26  назначение: SC
      доверие: абсолютное  достоверность: абсолютное
ssb  rsa3072/2B8***8B01C525C
      создан: 2020-11-26  годен до: 2022-11-26  назначение: E
[  абсолютно ] (1). Ksenia Rogova <ksenia.dumpling@gmail.com>
```

```
gpg> trust
sec  rsa3072/C***4955BFE6A
      создан: 2020-11-26  годен до: 2022-11-26  назначение: SC
      доверие: абсолютное  достоверность: абсолютное
ssb  rsa3072/2B8***B01C525C
      создан: 2020-11-26  годен до: 2022-11-26  назначение: E
[  абсолютно ] (1). Ksenia Rogova <ksenia.dumpling@gmail.com>
```

Укажите, насколько Вы доверяете данному пользователю в вопросах проверки достоверности ключей других пользователей (проверяет паспорт, сверяет отпечатки ключей из разных источников и т.п.)

- 1 = Не знаю или не буду отвечать
- 2 = НЕ доверяю
- 3 = Доверяю ограниченно
- 4 = Полностью доверяю
- 5 = Абсолютно доверяю
- m = вернуться в главное меню

Ваше решение? 5

```
sec  rsa3072/CC8***4955BFE6A
      создан: 2020-11-26  годен до: 2022-11-26  назначение: SC
      доверие: полное  достоверность: абсолютное
ssb  rsa3072/2B8***8B01C525C
      создан: 2020-11-26  годен до: 2022-11-26  назначение: E
[  абсолютно ] (1). Ksenia Rogova <ksenia.dumpling@gmail.com>
Учтите, что показанная достоверность ключа может быть неверной,
```

пока Вы не перезапустите программу.

```
gpg> quit
```

Проверим ранее подписанный файл:

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ gpg2 --verify mydoc.txt.sig
gpg: предполагается, что подписанные данные находятся в 'mydoc.txt'
gpg: Подпись сделана ПТ 27 ноя 2020 00:45:36 MSK
gpg:
gpg: ключом RSA с идентификатором
F70**BF4B1C3F20C076CC847734955BFE6A
gpg: Действительная подпись пользователя "Ksenia Rogova
<ksenia.dumpling@gmail.com>" [абсолютное]
```

Лабораторная работа №3 «Анализатор сетевого трафика Wireshark»

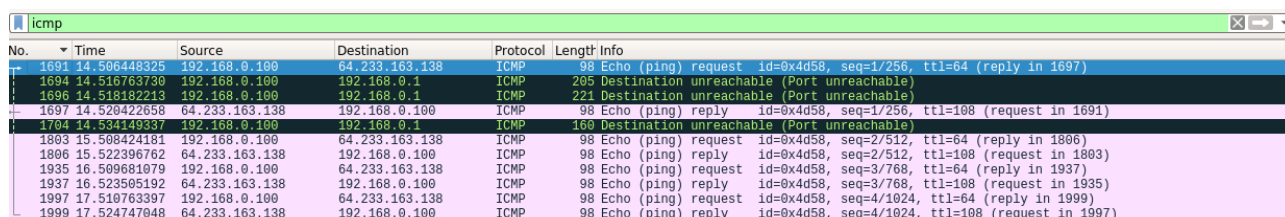
Сначала обозначим все интерфейсы, которые присутствуют на устройстве

```
dumpling@thinkpad-1380:~/Unix-Labs/network_security$ nmcli device status
DEVICE                TYPE      STATE      CONNECTION
enx9cebe8ba2b15       ethernet  подключено Проводное соединение 2
wlp2s0                wifi      подключено LogovoTarakanov
docker0               bridge    подключено docker0
enp0s31f6             ethernet  недоступен --
vmnet1                ethernet  не настроенно --
vmnet8                ethernet  не настроенно --
lo                    loopback  не настроенно --
```

Проанализируем трафик с помощью утилиты Wireshark. Выберем нужный интерфейс (ethernet-подключение в данном случае), начнем слушать трафик. Параллельно запустим 4 пакета утилитой ping:

```
dumpling@thinkpad-1380:~$ ping -c 4 google.com
```

и в Wireshark отфильтруем трафик ICMP:



No.	Time	Source	Destination	Protocol	Length	Info
1691	14.506448325	192.168.0.100	64.233.163.138	ICMP	98	Echo (ping) request id=0x4d58, seq=1/256, ttl=64 (reply in 1697)
1694	14.516763730	192.168.0.100	192.168.0.1	ICMP	205	Destination unreachable (Port unreachable)
1696	14.518192218	192.168.0.100	192.168.0.1	ICMP	221	Destination unreachable (Port unreachable)
1697	14.520422658	64.233.163.138	192.168.0.100	ICMP	98	Echo (ping) reply id=0x4d58, seq=1/256, ttl=108 (request in 1691)
1704	14.534149337	192.168.0.100	192.168.0.1	ICMP	160	Destination unreachable (Port unreachable)
1803	15.508424181	192.168.0.100	64.233.163.138	ICMP	98	Echo (ping) request id=0x4d58, seq=2/512, ttl=64 (reply in 1806)
1806	15.522396762	64.233.163.138	192.168.0.100	ICMP	98	Echo (ping) reply id=0x4d58, seq=2/512, ttl=108 (request in 1803)
1935	16.509681079	192.168.0.100	64.233.163.138	ICMP	98	Echo (ping) request id=0x4d58, seq=3/768, ttl=64 (reply in 1937)
1937	16.523505192	64.233.163.138	192.168.0.100	ICMP	98	Echo (ping) reply id=0x4d58, seq=3/768, ttl=108 (request in 1935)
1997	17.510763397	192.168.0.100	64.233.163.138	ICMP	98	Echo (ping) request id=0x4d58, seq=4/1024, ttl=64 (reply in 1999)
1999	17.524747048	64.233.163.138	192.168.0.100	ICMP	98	Echo (ping) reply id=0x4d58, seq=4/1024, ttl=108 (request in 1997)

Можно увидеть, что на каждый эхо-запрос успешно возвращался эхо-ответ. Wireshark предоставляет подробную информацию о пакете, расписывая каждый из полей. Для ICMP пакетов схема следующая:



С помощью утилиты ping можно редактировать содержимое данных с помощью ключа -p (pattern). Например, команда `ping -c 4 google.com -p ff` отправит 4 пакета, внутри которых данные будут состоять из одних единиц. Это видно в Wireshark:

```

▶ Frame 2096: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▶ Ethernet II, Src: BizlinkK_ba:2b:15 (9c:eb:e8:ba:2b:15), Dst: b0:95:75:50:4b:a3 (b0:95:75:50:4b:a3)
▶ Internet Protocol Version 4, Src: 192.168.0.100, Dst: 64.233.162.139
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x369b [correct]
  [Checksum Status: Good]
  Identifier (BE): 20259 (0x4f23)
  Identifier (LE): 9039 (0x234f)
  Sequence number (BE): 4 (0x0004)
  Sequence number (LE): 1024 (0x0400)
  [Response frame: 2097]
  Timestamp from icmp data: Nov 19, 2020 22:32:39.000000000 MSK
  [Timestamp from icmp data (relative): 0.398701799 seconds]
▼ Data (48 bytes)
  Data: 5e15060000000000ffffffffffffffffffffffffffffffffffff...
  [Length: 48]
0000 b0 95 75 50 4b a3 9c eb e8 ba 2b 15 08 00 45 00 ..uPK...+...E.
0010 00 54 e5 b2 40 00 40 01 b0 75 c0 a8 00 64 40 e9 .T..@.@. .u...d@.
0020 a2 8b 08 00 36 9b 4f 23 00 04 57 c8 b6 5f 00 00 ...6.0# .W...
0030 00 00 5e 15 06 00 00 00 00 00 ff ff ff ff ff ff .A.....
0040 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
0050 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
0060 ff ff ..

```

Проанализируем ARP-пакеты. Попросим маршрутизатор сказать, какое устройство находится по адресу 192.168.0.101 (это еще один ПК в домашней сети) и снимем трафик в Wireshark:

```

dumpling@thinkpad-1380:~$ arping -I enx9cebe8ba2b15 192.168.0.101

```

5860	47.518616479	BizlinkK_ba:2b:15	AsustekC_e1:4d:3a	ARP	42	Who has 192.168.0.101? Tell 192.168.0.100
5861	47.519043532	AsustekC_e1:4d:3a	BizlinkK_ba:2b:15	ARP	60	192.168.0.101 is at 10:bf:48:e1:4d:3a
▶ Frame 5860: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
▶ Ethernet II, Src: BizlinkK_ba:2b:15 (9c:eb:e8:ba:2b:15), Dst: AsustekC_e1:4d:3a (10:bf:48:e1:4d:3a)						
▼ Address Resolution Protocol (request)						
Hardware type: Ethernet (1)						
Protocol type: IPv4 (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: request (1)						
Sender MAC address: BizlinkK_ba:2b:15 (9c:eb:e8:ba:2b:15)						
Sender IP address: 192.168.0.100						
Target MAC address: AsustekC_e1:4d:3a (10:bf:48:e1:4d:3a)						
Target IP address: 192.168.0.101						

На системе был поднят FTP-сервер, была попытка коннекта к серверу:

```
dumppling@thinkpad-1380:~$ systemctl status vsftpd
```

```
● vsftpd.service - vsftpd FTP server
```

```
Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since Wed 2020-11-18 00:07:52 MSK; 5s ago
```

```
Process: 15481 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
```

```
Main PID: 15487 (vsftpd)
```

```
Tasks: 1 (limit: 4915)
```

```
CGroup: /system.slice/vsftpd.service
```

```
└─15487 /usr/sbin/vsftpd /etc/vsftpd.conf
```

```
dumppling@thinkpad-1380:~$ ftp 127.0.0.1
```

Снятые сниффером пакеты показывают информацию о логине и пароле в незащищенном виде:

No.	Time	Source	Destination	Protocol	Length	Info
1138	9.756439098	127.0.0.1	127.0.0.1	FTP	88	Response: 220 (vsFTPd 3.0.3)
1877	15.516118006	127.0.0.1	127.0.0.1	FTP	83	Request: USER dumppling
1879	15.516305593	127.0.0.1	127.0.0.1	FTP	102	Response: 331 Please specify the password.
2352	19.703889105	127.0.0.1	127.0.0.1	FTP	87	Request: PASS Password123!
2745	23.320440665	127.0.0.1	127.0.0.1	FTP	90	Response: 530 Login incorrect.
2747	23.320582535	127.0.0.1	127.0.0.1	FTP	74	Request: SYST
2749	23.320674456	127.0.0.1	127.0.0.1	FTP	106	Response: 530 Please login with USER and PASS.
3052	25.951951409	127.0.0.1	127.0.0.1	FTP	74	Request: QUIT
3053	25.952126700	127.0.0.1	127.0.0.1	FTP	82	Response: 221 Goodbye.

SSH соединения передают все данные в зашифрованном виде:

```
dumppling@thinkpad-1380:~$ ssh root@127.0.0.1 -p 22
```

```
root@127.0.0.1's password:
```

ssh						
No.	Time	Source	Destination	Protocol	Length	Info
949	7.033130174	127.0.0.1	127.0.0.1	SSHv2	109	Client: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3)
957	7.049562376	127.0.0.1	127.0.0.1	SSHv2	109	Server: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3)
959	7.050604577	127.0.0.1	127.0.0.1	SSHv2	1428	Client: Key Exchange Init
960	7.051693112	127.0.0.1	127.0.0.1	SSHv2	1148	Server: Key Exchange Init
963	7.058919022	127.0.0.1	127.0.0.1	SSHv2	116	Client: Diffie-Hellman Key Exchange Init
965	7.074084843	127.0.0.1	127.0.0.1	SSHv2	520	Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=172)
969	7.080939054	127.0.0.1	127.0.0.1	SSHv2	84	Client: New Keys
981	7.124853737	127.0.0.1	127.0.0.1	SSHv2	112	Client: Encrypted packet (len=44)
983	7.124993920	127.0.0.1	127.0.0.1	SSHv2	112	Server: Encrypted packet (len=44)
984	7.125137574	127.0.0.1	127.0.0.1	SSHv2	128	Client: Encrypted packet (len=60)
985	7.126063785	127.0.0.1	127.0.0.1	SSHv2	120	Server: Encrypted packet (len=52)
1661	11.988813790	127.0.0.1	127.0.0.1	SSHv2	216	Client: Encrypted packet (len=148)
2088	14.613945339	127.0.0.1	127.0.0.1	SSHv2	120	Server: Encrypted packet (len=52)
2288	16.228918331	127.0.0.1	127.0.0.1	SSHv2	152	Client: Encrypted packet (len=84)
2290	16.229290149	127.0.0.1	127.0.0.1	SSHv2	120	Server: Encrypted packet (len=52)

Telnet на localhost в данной системе тоже ходит через SSH:

```
dumppling@thinkpad-1380:~$ telnet 127.0.0.1 22
```

```
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
```

```
Escape character is '^['.
```

SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3

ssh					
No.	Time	Source	Destination	Protocol	Length Info
584	6.515257233	127.0.0.1	127.0.0.1	SSH	109 Server: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3)
1236	11.646430766	127.0.0.1	127.0.0.1	SSH	82 Client: Encrypted packet (len=14)
1238	11.646582076	127.0.0.1	127.0.0.1	SSH	87 Server: Encrypted packet (len=19)

Лабораторная работа № 4 «Аудит защищенности сети сканером Nmap»

Проведем сканирование локальной сети. Для начала определим подсеть:

```
dump1ing@thinkpad-1380:~$ ip a s
```

```
. . .
7: enx9cebe8ba2b15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
state UP group default qlen 1000
    link/ether 9c:eb:e8:ba:2b:15 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global dynamic noprefixroute
        valid_lft 6417sec preferred_lft 6417sec
    inet6 fe80::b07a:f485:a03f:d05a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Быстро просканируем сеть:

```
dump1ing@thinkpad-1380:~$ nmap -T4 192.168.0.0/24
```

Starting Nmap 7.60 (<https://nmap.org>) at 2020-11-18 00:55 MSK

Nmap scan report for _gateway (192.168.0.1)

Host is up (0.0091s latency).

Not shown: 996 closed ports

PORT	STATE	SERVICE
------	-------	---------

22/tcp	open	ssh
--------	------	-----

53/tcp	open	domain
--------	------	--------

80/tcp	open	http
--------	------	------

1900/tcp	open	upnp
----------	------	------

Nmap scan report for thinkpad-1380 (192.168.0.100)

Host is up (0.00018s latency).

Not shown: 993 closed ports

PORT	STATE	SERVICE
------	-------	---------

21/tcp	open	ftp
--------	------	-----

22/tcp	open	ssh
--------	------	-----

80/tcp	open	http
--------	------	------

111/tcp	open	rpcbind
---------	------	---------

443/tcp	open	https
---------	------	-------

902/tcp	open	iss-realsecure
---------	------	----------------

2049/tcp	open	nfs
----------	------	-----

Nmap scan report for thinkpad-1380 (192.168.0.102)

Host is up (0.00017s latency).

Not shown: 993 closed ports

PORT	STATE	SERVICE
------	-------	---------

21/tcp	open	ftp
--------	------	-----

22/tcp	open	ssh
--------	------	-----

80/tcp	open	http
--------	------	------

111/tcp	open	rpcbind
---------	------	---------

443/tcp	open	https
---------	------	-------

```
902/tcp open iss-realsecure
2049/tcp open nfs
```

Nmap done: 256 IP addresses (3 hosts up) scanned in 2.50 seconds

Можно увидеть, что обнаружено три хоста — маршрутизатор и два интерфейса рабочей станции — беспроводное и проводное устройство. В данной домашней сети точно больше устройств, чем вывел nmap. Попробуем найти их IP- и MAC-адреса, предварив команду sudo:

```
dump1ing@thinkpad-l380:~$ sudo nmap -PR 192.168.0.0/24
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2020-11-19 22:53 MSK
Nmap scan report for _gateway (192.168.0.1)
Host is up (0.00056s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
1900/tcp  open  upnp
MAC Address: B0:95:75:50:4B:A3 (Unknown)
```

```
Nmap scan report for 192.168.0.101
Host is up (0.00060s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
6881/tcp  open  bittorrent-tracker
49163/tcp open  unknown
MAC Address: 10:BF:48:E1:4D:3A (Asustek Computer)
```

```
Nmap scan report for thinkpad-l380 (192.168.0.100)
Host is up (0.000029s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
443/tcp   open  https
902/tcp   open  iss-realsecure
2049/tcp  open  nfs
```

```
Nmap scan report for thinkpad-l380 (192.168.0.102)
Host is up (0.000027s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
443/tcp   open  https
902/tcp   open  iss-realsecure
2049/tcp  open  nfs
```

Nmap done: 256 IP addresses (4 hosts up) scanned in 201.41 seconds

Попробуем запустить Nmap с другими ключами. Пусть утилита выберет случайным образом 5 хостов и просканирует их на наличие запущенных на них веб-серверов (порт 80). Перебор хостов отключен опцией -Pn, т.к. посылка пары предварительных запросов с целью определения доступности хоста является нецелесообразной, когда интересует всего один порт на каждом хосте.

```
dumppling@thinkpad-l380:~$ nmap -v -iR 5 -Pn -p 80
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2020-11-19 23:14 MSK
Initiating Parallel DNS resolution of 5 hosts. at 23:14
Completed Parallel DNS resolution of 5 hosts. at 23:14, 0.16s elapsed
Initiating Connect Scan at 23:14
Scanning 5 hosts [1 port/host]
Completed Connect Scan at 23:14, 2.00s elapsed (5 total ports)
Nmap scan report for 1.179.194.249
Host is up.
```

PORT	STATE	SERVICE
80/tcp	filtered	http

```
Nmap scan report for 95.162.68.144
Host is up.
```

PORT	STATE	SERVICE
80/tcp	filtered	http

```
Nmap scan report for host86-152-12-173.range86-152.btcentralplus.com
(86.152.12.173)
Host is up.
```

PORT	STATE	SERVICE
80/tcp	filtered	http

```
Nmap scan report for pool-173-70-163-150.nwrknj.fios.verizon.net
(173.70.163.150)
Host is up.
```

PORT	STATE	SERVICE
80/tcp	filtered	http

```
Nmap scan report for 91.137.1.81
Host is up.
```

PORT	STATE	SERVICE
80/tcp	filtered	http

```
Read data files from: /usr/bin/./share/nmap
Nmap done: 5 IP addresses (5 hosts up) scanned in 2.21 seconds
```

Лабораторная работа №5 «Утилиты Netcat и Cryptcat»

На виртуальной машине (с Arch на борту) утилита ncат стала слушать 7000 порт:

```
[analyst@secOps ~]$ ncat -lvp 7000
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::7000
Ncat: Listening on 0.0.0.0:7000
Ncat: Connection from 192.168.0.100.
Ncat: Connection from 192.168.0.100:36860.
Hi Arhc-Linux host!
Hello, Ubuntu!
```

С основного хоста (с Ubuntu на борту) с помощью утилиты nc было создано соединение по адресу виртуальной машины (192.168.0.113) на порт 7000:

```
dumpling@thinkpad-l380:~$ nc 192.168.0.113 7000
Hi Arhc-Linux host!
Hello, Ubuntu!
```

Можно видеть, что обмен сообщениями проходит успешно. Организуем передачу файла:

```
[analyst@secOps ~]$ cat output.txt
cat: output.txt: No such file or directory
[analyst@secOps ~]$
[analyst@secOps ~]$ ncat -lvp 7000 > output.txt
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::7000
Ncat: Listening on 0.0.0.0:7000
Ncat: Connection from 192.168.0.100.
Ncat: Connection from 192.168.0.100:36916.

^C
[analyst@secOps ~]$ cat output.txt
Hi, this is Ubuntu!
I hope you feel well and everything is fine.
[analyst@secOps ~]$
```

```
dumpling@thinkpad-l380:~$ nano send_file.txt
dumpling@thinkpad-l380:~$ cat send_file.txt
Hi there! It's Ubuntu.
I hope everything is fine and you feel well.
dumpling@thinkpad-l380:~$
dumpling@thinkpad-l380:~$ nc 192.168.0.113 7000 < sender_info.txt

dumpling@thinkpad-l380:~$
```

Перехватим незашифрованный и зашифрованный трафик сниффером. Развернем на основном хосте слушающий сервер и отправим с виртуальной машины через NetCat два сообщения:

```
[analyst@secOps ~]$ ncat 192.168.0.100 7000
Hi there!
It's arch
```

Найдем эти пакеты в Wireshark:

192.168.0.113	192.168.0.100	TCP	76 45084 → 7000 [PSH, ACK] Seq=1 Ack=1 Win=29696 Len=10
192.168.0.113	192.168.0.100	TCP	76 45084 → 7000 [PSH, ACK] Seq=11 Ack=1 Win=29696 Len=10

```

9c eb e8 ba 2b 15 64 5d 86 88 04 e6 08 00 45 00  ....+d] .....E.
00 3e cf 74 40 00 40 06 e9 1f c0 a8 00 71 c0 a8  .>.t@.@. ....q..
00 64 b0 1c 1b 58 5c c7 d6 8e 95 89 c5 43 80 18  .d...X\.. ....C..
00 3a 73 eb 00 00 01 01 08 0a e2 e7 26 b4 67 f2  .:s..... ..&.g.
50 87 48 69 20 74 68 65 72 65 21 0a             P.Hi the re!.

9c eb e8 ba 2b 15 64 5d 86 88 04 e6 08 00 45 00  ....+d] .....E.
00 3e cf 75 40 00 40 06 e9 1e c0 a8 00 71 c0 a8  .>.u@.@. ....q..
00 64 b0 1c 1b 58 5c c7 d6 98 95 89 c5 43 80 18  .d...X\.. ....C..
00 3a 37 f3 00 00 01 01 08 0a e2 e7 48 cb 67 f2  .:7..... ..H.g.
63 5a 49 74 27 73 20 61 72 63 68 0a             cZIt's a rch.

```

Теперь отправим сообщения с помощью утилиты Cryptcat:

```

dumpling@thinkpad-l380:~$ cryptcat 127.0.0.1 7000
Hi! It's encrypted now!

```

Видим, что сообщение зашифровано:

Wireshark · Packet 12 · Loopback: lo

▶ Frame 12: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0

▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

▼ Transmission Control Protocol, Src Port: 35340, Dst Port: 7000, Seq: 33, Ack: 1, Len: 16

Source Port: 35340

Destination Port: 7000

[Stream index: 0]

[TCP Segment Len: 16]

Sequence number: 33 (relative sequence number)

[Next sequence number: 49 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

1000 = Header Length: 32 bytes (8)

▶ Flags: 0x018 (PSH, ACK)

Window size value: 512

[Calculated window size: 65536]

[Window size scaling factor: 128]

Checksum: 0xfe38 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

▶ [SEQ/ACK analysis]

▶ [Timestamps]

TCP payload (16 bytes)

TCP segment data (16 bytes)

0000	00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00E.
0010	00 44 21 18 40 00 40 06 1b 9a 7f 00 00 01 7f 00	.D!..@..
0020	00 01 8a 0c 1b 58 bd 4d 32 69 8b d9 49 b3 80 18X·M 2i·I...
0030	02 00 fe 38 00 00 01 01 08 0a e5 45 de e3 e5 45	...8....E..E
0040	de e3 66 74 ff d6 c4 3a 57 cb be 5b b8 33 37 aa	..ft...: W·[·37·
0050	09 69	.i

Лабораторная работа №6 «Сетевое экранирование. Применение правил iptables»

Выведем iptable правила на хосте:

```

dumpling@thinkpad-l380:~$ sudo iptables -L

```

```

Chain INPUT (policy ACCEPT)
target      prot opt source                destination

Chain FORWARD (policy DROP)
target      prot opt source                destination
DOCKER-USER all  --  anywhere              anywhere
DOCKER-ISOLATION-STAGE-1 all  --  anywhere              anywhere
ACCEPT      all  --  anywhere              anywhere    ctstate
RELATED,ESTABLISHED
DOCKER      all  --  anywhere              anywhere
ACCEPT      all  --  anywhere              anywhere
ACCEPT      all  --  anywhere              anywhere

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination

Chain DOCKER (1 references)
target      prot opt source                destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target      prot opt source                destination
DOCKER-ISOLATION-STAGE-2 all  --  anywhere              anywhere
RETURN      all  --  anywhere              anywhere

Chain DOCKER-ISOLATION-STAGE-2 (1 references)
target      prot opt source                destination
DROP        all  --  anywhere              anywhere
RETURN      all  --  anywhere              anywhere

Chain DOCKER-USER (1 references)
target      prot opt source                destination
RETURN      all  --  anywhere              anywhere

```

Заблокируем входной трафик:

```

dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -j DROP
dumpling@thinkpad-1380:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target      prot opt source                destination
DROP        all  --  anywhere              anywhere

```

На хосте входной трафик заблокирован, проверим это утилитой ping:

```

dumpling@thinkpad-1380:~$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3060ms

```

Добавим правило, пропускающее веб-трафик и блокирующее весь остальной входящий трафик. Сначала запретим весь трафик и добавим в исключения lo интерфейс:

```

dumpling@thinkpad-1380:~$ sudo iptables -P INPUT DROP
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -i lo -j ACCEPT

```

Разрешим TCP- и UDP-трафик:

```

dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p tcp -m state --state

```



```
ESTABLISHED,RELATED -j ACCEPT
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p udp -m state --state
ESTABLISHED,RELATED -j ACCEPT
```

Разрешим DNS:

```
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p tcp --dport 53 -j ACCEPT
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p tcp --sport 53 -j ACCEPT
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p udp --dport 53 -j ACCEPT
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p udp --sport 53 -j ACCEPT
```

Разрешим HTTP и HTTPS:

```
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

```
dumpling@thinkpad-1380:~$ sudo iptables -L INPUT -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
0 0 ACCEPT all -- lo any anywhere anywhere
0 0 ACCEPT tcp -- any any anywhere anywhere state RELATED,ESTABLISHED
0 0 ACCEPT udp -- any any anywhere anywhere state RELATED,ESTABLISHED
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:domain
0 0 ACCEPT tcp -- any any anywhere anywhere tcp spt:domain
0 0 ACCEPT udp -- any any anywhere anywhere udp dpt:domain
0 0 ACCEPT udp -- any any anywhere anywhere udp spt:domain
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:http
0 0 ACCEPT tcp -- any any anywhere anywhere tcp dpt:https
3057 342K DROP all -- any any anywhere anywhere
```

Проверим трафик:

```
dumpling@thinkpad-1380:~$ ping -c 4 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.075 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.079 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.074 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.074/0.076/0.079/0.009 ms
```

```
dumpling@thinkpad-1380:~$ ping -c 4 google.com
PING google.com (173.194.222.100) 56(84) bytes of data.
64 bytes from lo-in-f100.1e100.net (173.194.222.100): icmp_seq=1 ttl=108 time=12.8 ms
64 bytes from lo-in-f100.1e100.net (173.194.222.100): icmp_seq=2 ttl=108 time=13.6 ms
64 bytes from lo-in-f100.1e100.net (173.194.222.100): icmp_seq=3 ttl=108 time=16.4 ms
64 bytes from lo-in-f100.1e100.net (173.194.222.100): icmp_seq=4 ttl=108 time=13.6 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 12.825/14.164/16.453/1.372 ms
```

Лабораторная работа №7 «Сетевое экранирование. Работа с iptables»

Запретим ICMP ping запросы извне:

```
dumpling@thinkpad-1380:~$ sudo iptables -A INPUT -p icmp --icmp-type echo-
request -j DROP
```

```
[analyst@secOps ~]$ ping -c 4 192.168.0.102
PING 192.168.0.102 (192.168.0.102) 56(84) bytes of data.

--- 192.168.0.102 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3062ms
```

Разрешим ICMP ответы:

```
dumpling@thinkpad-l380:~$ export SERVER_IP=192.168.0.102
dumpling@thinkpad-l380:~$ sudo iptables -I INPUT -i wlp2s0 -p icmp --icmp-type 8
-s 0/0 -d $SERVER_IP -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
dumpling@thinkpad-l380:~$ sudo iptables -I OUTPUT -p icmp --icmp-type 0 -s
$SERVER_IP -d 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Посмотрим на добавленные правила:

```
dumpling@thinkpad-l380:~$ sudo iptables -L INPUT -v
Chain INPUT (policy ACCEPT 181 packets, 54022 bytes)
  pkts bytes target     prot opt in     out     source            destination
    0      0 ACCEPT     icmp -- wlp2s0 any      anywhere          thinkpad-
1380      icmp echo-request state NEW,RELATED,ESTABLISHED
    4   336 DROP       icmp -- any     any      anywhere          anywhere
icmp echo-request
dumpling@thinkpad-l380:~$ sudo iptables -L OUTPUT -v
Chain OUTPUT (policy ACCEPT 219 packets, 23174 bytes)
  pkts bytes target     prot opt in     out     source            destination
    0      0 ACCEPT     icmp -- any     any      thinkpad-l380     anywhere
icmp echo-reply state RELATED,ESTABLISHED
```

Проверим пинг с удаленного хоста:

```
[analyst@secOps ~]$ ping -c 4 192.168.0.102
PING 192.168.0.102 (192.168.0.102) 56(84) bytes of data.
64 bytes from 192.168.0.102: icmp_seq=1 ttl=64 time=0.407 ms
64 bytes from 192.168.0.102: icmp_seq=2 ttl=64 time=0.372 ms
64 bytes from 192.168.0.102: icmp_seq=3 ttl=64 time=0.321 ms
64 bytes from 192.168.0.102: icmp_seq=4 ttl=64 time=0.353 ms

--- 192.168.0.102 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3062ms
rtt min/avg/max/mdev = 0.321/0.363/0.407/0.033 ms
```

Заблокируем трафик с адреса удаленного хоста (важно добавить правило в начало списка):

```
dumpling@thinkpad-l380:~$ sudo iptables -I INPUT 1 -s 192.168.0.113 -j DROP
dumpling@thinkpad-l380:~$ sudo iptables -L INPUT -v
Chain INPUT (policy ACCEPT 16 packets, 1600 bytes)
  pkts bytes target     prot opt in     out     source            destination
    0      0 DROP      all -- any     any      192.168.0.113     anywhere
    4   336 ACCEPT     icmp -- wlp2s0 any      anywhere          thinkpad-
1380      icmp echo-request state NEW,RELATED,ESTABLISHED
    0      0 DROP      icmp -- any     any      anywhere          anywhere
icmp echo-request
```

```
[analyst@secOps ~]$ ping -c 4 192.168.0.102
PING 192.168.0.102 (192.168.0.102) 56(84) bytes of data.

--- 192.168.0.102 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3059ms
```

На локальном хосте поднимем netcat и заставим его слушать 7000 порт. Отправим сообщение «Hello» с виртуальной машины, потом внесем правило, ограничивающее трафик с 700 порта и попробуем отправить второе сообщение:

```
dumpling@thinkpad-l380:~$ nc -lvp 7000
Listening on [0.0.0.0] (family 0, port 7000)
Connection from 192.168.0.113 56562 received!
Hello
```

```
dumpling@thinkpad-l380:~$ sudo iptables -I INPUT 1 -p tcp --dport 7000 -j DROP
```

```
[analyst@secOps ~]$ ncat 192.168.0.102 7000
Hello
Hello again
```

Заблокируем трафик по MAC-адресу:

```
dumpling@thinkpad-l380:~$ sudo iptables -A INPUT -m mac --mac-source 08:00:27:88:b3:29 -j DROP
```

```
[analyst@secOps ~]$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:88:b3:29 brd ff:ff:ff:ff:ff:ff
[analyst@secOps ~]$ ping -c 4 192.168.0.102
PING 192.168.0.102 (192.168.0.102) 56(84) bytes of data.

--- 192.168.0.102 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3077ms
```

И разрешим для этого MAC-адреса TCP-соединение через 7000 порт:

```
dumpling@thinkpad-l380:~$ sudo iptables -I INPUT 1 -p tcp --destination-port 7000 -m mac --mac-source 08:00:27:88:b3:29 -j ACCEPT
```

```
[analyst@secOps ~]$ ncat 192.168.0.102 7000
Hello
```

```
dumpling@thinkpad-l380:~$ nc -lvp 7000
Listening on [0.0.0.0] (family 0, port 7000)
Connection from 192.168.0.113 56564 received!
Hello
```