

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
Факультет безопасности информационных технологий**

Дисциплина:

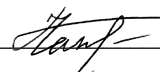
«Технологии и методы программирования»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Выполнили:

студенты группы N33491

Чапасов Пётр Константинович

_____ 
(подпись)

Проверил:

Ищенко Алексей Петрович

(подпись)

Санкт-Петербург

2022 г.

Техническое Задание

А) Выполняется в локальной операционной системе.

1. Создать текстовый документ (sys.tat), в котором будет содержаться «Системная информация».
2. Написать программу-инсталлятор sys_doc.exe для этого документа, которая под видом установки обновления (с отображением строки прогресса обновления) к какой-нибудь программе (например, Блокнот или Paint):
 - Запрашивает у пользователя папку (должен быть вариант использования существующей папки и вариант создания собственной) для копирования «Системной информации».
 - Записывает в папку файл с исполняемым кодом программы secur.exe (аналог требований к template.tbl из лабораторной работы №1), защищающей sys.tat.
 - Собирает (возможную) информацию о компьютере, на котором устанавливается программа.
 - Кодировывает эту информацию и записывает в файл sys.tat.
 - Подписывает её личным ключом пользователя программы и записывает подпись, например, в реестр Windows в раздел HKEY_CURRENT_USER\Software\Фамилия_студента как значение параметра Signature.
 - Запускает secur.exe для защиты sys.tat от несанкционированного доступа.
 - Прописывает запуск программы secur.exe при выполнении функции Open для sys.tat, чтобы защита срабатывала и после перезагрузки ОС (есть несколько способов такой «привязки»).
3. В саму программу защиты secur.exe включить следующий функционал:
 - Запрос у пользователя информации об имени раздела реестра с электронной цифровой подписью (фамилией студента).
 - Считывание подписи из указанного выше раздела реестра, которая проверяется с помощью открытого ключа пользователя.
 - Разрешение или запрет просмотра «Системной информации» в файле sys.tat в зависимости от правильности указания ключа.
4. При неудачной проверке работа защищаемой программы должна прекращаться с выдачей соответствующего сообщения.
5. Собираемая о компьютере информация включает в себя как минимум:
 - Имя пользователя,
 - Имя компьютера,
 - Конфигурацию компьютера (память и процессор, как минимум) и версию ОС.

Б) Выполняется в локальной сети (или виртуальной).

1. Создать скрипт, который удалённо и незаметно для пользователя (пользователь открывает какую-нибудь веб-страничку от создателя скрипта) собирает информацию о нём, его компьютере и системе (п.5 предыдущего задания) и записывает её на какой-либо локальный сетевой диск (доступный создателю скрипта) в папку с именем IP или Mac-адреса пользовательской машины.
2. Продумать доступ к этой информации (можно писать на удалённый диск).
3. Протестировать на 3-5 клиентах и получить статистику о них.

Выполнение работы

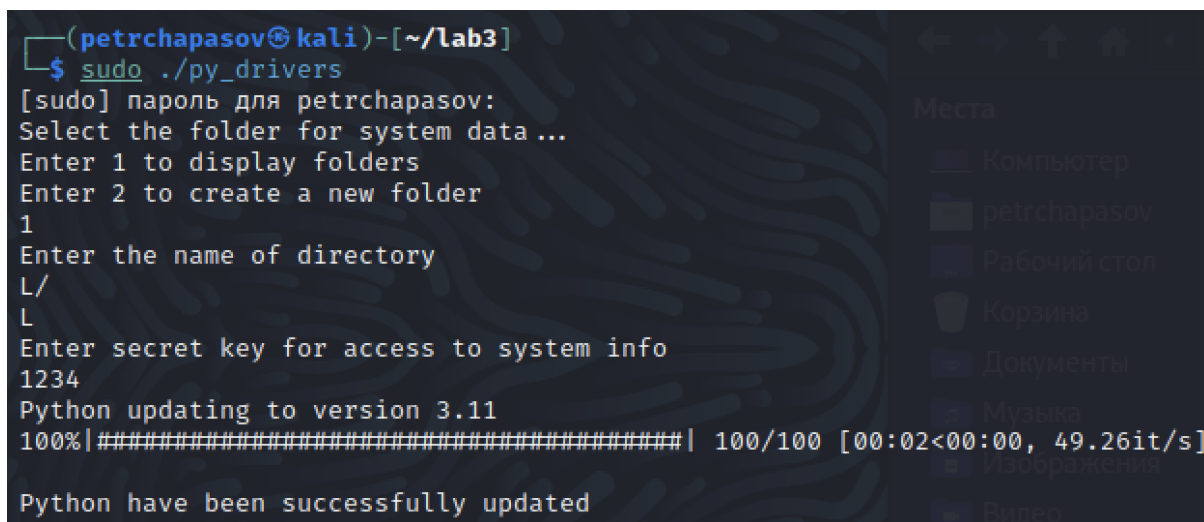
Вариант А

Инструкция по использованию:

Установка:

1. Запускаем программу-установщик *py_drivers*

\$ sudo ./py_drivers
2. Выбираем использовать существующий каталог - 1 (Рис. 1.1), или создать новый - 2 (Рис. 1.2)
3. Если выбрали существующий каталог, то указываем имя существующего каталога. Если создаем новый - вводим любое имя.
4. Вводим секретный ключ, который позволит нам посмотреть системную информацию

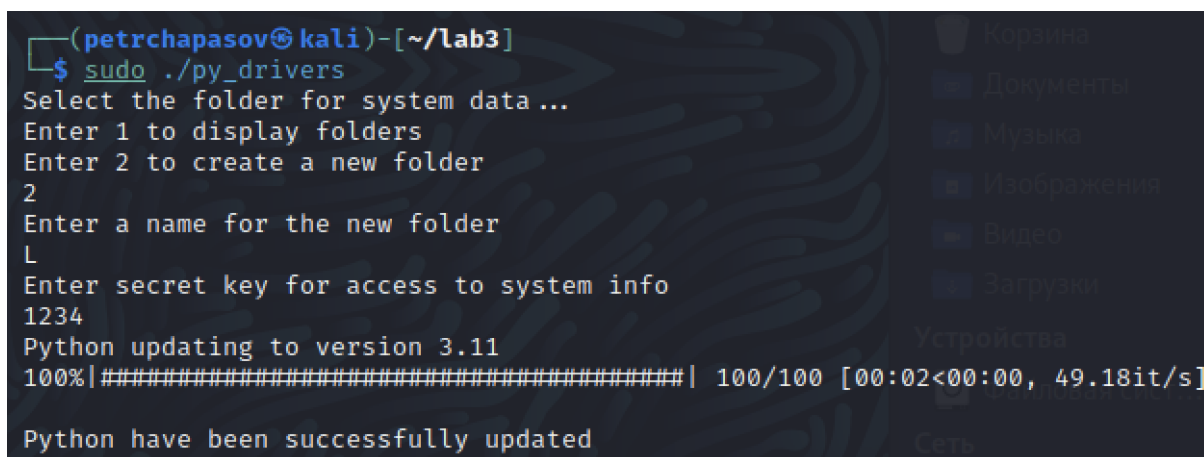


The screenshot shows a terminal window on the left and a file manager on the right. The terminal output is as follows:

```
(petrchapasov@kali)-[~/lab3]
$ sudo ./py_drivers
[sudo] пароль для petrchapasov:
Select the folder for system data ...
Enter 1 to display folders
Enter 2 to create a new folder
1
Enter the name of directory
L/
L
Enter secret key for access to system info
1234
Python updating to version 3.11
100%|#####| 100/100 [00:02<00:00, 49.26it/s]
Python have been successfully updated
```

The file manager on the right shows the contents of the ~/lab3 directory, including folders like Компьютер, petrchapasov, Рабочий стол, Корзина, Документы, Музыка, Изображения, and Видео.

Рисунок 1.1. Установка в существующий каталог



The screenshot shows a terminal window on the left and a file manager on the right. The terminal output is as follows:

```
(petrchapasov@kali)-[~/lab3]
$ sudo ./py_drivers
Select the folder for system data ...
Enter 1 to display folders
Enter 2 to create a new folder
2
Enter a name for the new folder
L
Enter secret key for access to system info
1234
Python updating to version 3.11
100%|#####| 100/100 [00:02<00:00, 49.18it/s]
Python have been successfully updated
```

The file manager on the right shows the contents of the ~/lab3 directory, including folders like Корзина, Документы, Музыка, Изображения, Видео, Загрузки, and Устройства.

Рисунок 1.2. Установка в новый каталог

Использование:

В результате установки, в указанной директории у нас появился исполняемый файл *secure* (Рис. 2.1). Также в этой директории есть два скрытых файла: *.key*, *.sys.tat* (Рис. 2.2). Для того, чтобы посмотреть *.sys.tat* необходимо запустить *seure*

1. Запускаем *secure*

```
$ sudo ./secure
```

2. Вводим полный путь до файла *.key*. В нашем случае выглядит так:

```
/home/petrchapasov/lab3/L/.key
```

3. Вводим наш секретный ключ, для того, чтобы посмотреть системную информацию хранящуюся в *.sys.tat*

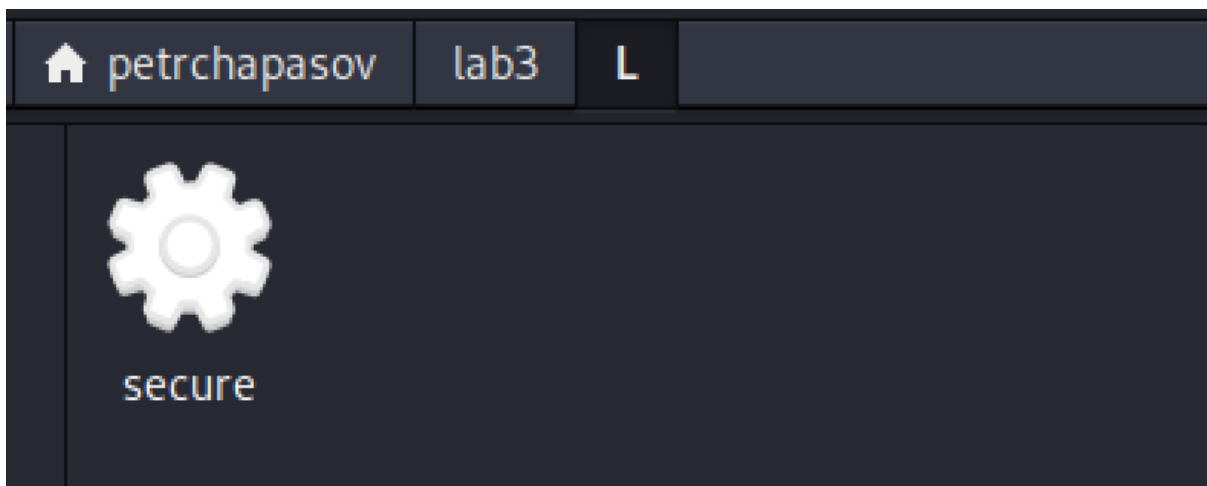


Рисунок 2.1. Результат установки

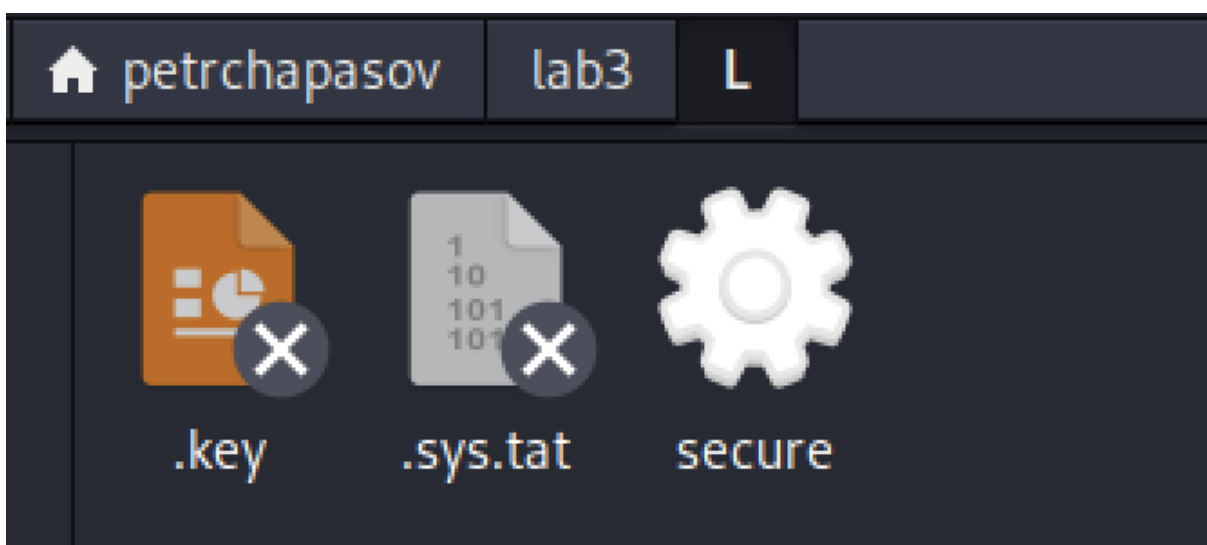


Рисунок 2.2. Все файлы директории


```

def loading():
    print('Python updating to version 3.11')
    for i in tqdm(range(100), ascii = True):
        time.sleep(0.02)
    print('\nPython have been successfully updated')

def install(dir, key):
    script =
'I2luY2x1ZGUgPGLvc3RyZWFTPgojaW5jbHVkZSA8ZnN0cmVhbT4KI2luY2x1ZGUgPHN0cmLuZz4KI2luY2
x1ZGUgPHVub3JkZXJlZF9tYXA+CiNpbmNsdWRLIDxiaXRzL3N0ZGMrKy5oPgp1c2luZyBuYW1lc3BhY2Ugc
3RkOwoKaW50IG1haW4gKGludCBhcmdjLCBjaGFyKiogYXJndikgCnsKICAgIHN0cmLuZyBzYWx0ID0gXCJz
YWx0XCi7CiAgICBzdHJpbmcgdXNlcmtleSA9IGFyZ3ZbMV07CiAgICBzdHJpbmcgc3RyID0gc2FsdCArIHV
zZXJrZXk7CiAgICBoYXNoIDxzdhJpbmc+IGhhc2hlcjSkICAgIHNpemVfdCBoYXNoID0gaGFzaGVyKHN0ci
k7CiAgICBjb3V0IDw8IGhhc2g7Cn0='
    k = base64.b64decode(script)
    cmd = k.decode("UTF-8")
    os.system(' echo "' + cmd + '" > ./' + dir + '/key.cpp')
    os.system(' g++ ./' + dir + '/key.cpp -o ./' + dir + '/key.exe')
    h = subprocess.check_output(['./' + dir + '/key.exe', key])
    os.system(' rm ./' + dir + '/key*')
    h = h.decode("UTF-8")
    os.system(' echo "' + h + '" > ./' + dir + '/.key')
    os.system(' sudo chmod ugo-rwx ./' + dir + '/.key')
    os.system(' sudo chattr +i ./' + dir + '/.key')

#-----

    script =
'I2luY2x1ZGUgPHN0ZGxpYi5oPgojaW5jbHVkZSA8aW9zdHJlYW0+CiNpbmNsdWRLIDxmc3RyZWFTPgojaW
5jbHVkZSA8c3N0cmVhbT4KI2luY2x1ZGUgPHN0cmLuZz4KI2luY2x1ZGUgPHVub3JkZXJlZF9tYXA+CiNpb
mNsdWRLIDxiaXRzL3N0ZGMrKy5oPgp1c2luZyBuYW1lc3BhY2Ugc3RkOwoKYm9vbCBFbmNyeXB0KHN0cmLu
ZyBmbGQsIHN0cmLuZyB1c2Vya2V5KTsKYm9vbCBLZXlfY29tcChzdHJpbmcgZmxkLCBzaXplX3QgaGFzaCk
7CmludCBibG9ja2VyKHN0cmLuZyBhKTsKaW50IHV0Ym9vY2t1c2hhdHJpbmcgYSk7CgppbnQgbWFPbiAoKQ
p7Cgljb3V0IDw8IFwiRW50ZXIgdGhlIGZ1bGwgaW5kZXBlbmRlbnQgcGF0aCB0byAua2V5XCIGPDwgZW5kb
DsKCXN0cmLuZyBmbGQgPSBcIlwiOwoJY2luID4+IGZsZDsKCQoJdW5ibG9ja2VyKGZsZCk7CglpZnN0cmVh
bSBmaWxkKGZsZCk7CglpZiAoIiwZpbGUuZ29vZCgpKQoJewoJCWNvdXQgPDwgXCJXcm9uZyBmb2xkZXIgm9
yIC5rZXkhXCIGPDwgZW5kbDsKCQlyZXR1cm4gMTsKCX0KCWZpbGUuY2xvc2UoKTsk'
    k = base64.b64decode(script)
    cmd = k.decode("UTF-8")
    os.system(' echo "' + cmd + '" >> ./' + dir + '/secure.cpp')

    script =
'ICAgIAIjb3V0IDw8IFwiRW50ZXIgdGhlIGtleSB0byBzZWUgc3lzaW5mb1wiIDw8IGVuZGw7CiAgICAJc3
RyaW5nIGtleTsKICAgIAIjaW4gPj4ga2V50wogICAgCWlmIchFbmNyeXB0KGZsZCwga2V5KSkKICAgIAI7C
iAgICAJCWNvdXQgPDwgXCJBY2Nlc3MgZGVuaWVkiVwiIDw8IGVuZGw7CiAgICAJCJWJsbn2N2ZXIoZmxkKtsK
CQl1bmJsb2N2ZXIoXCiuc3lzlLnRhdFwiKtsKCQlPZnN0cmVhbSBmaWxkFwiLnN5cy50YXRcIik7CgkJaWY
gKCFmaWxkLmdvb2QoKSkKCQl7CgkJCWNvdXQgPDwgXCJXcm9uZyBmb2xkZXIgm9yIHNLy3VyZS5leGUb3
IgZmlsZSBkb2Vzbid0IGV4aXN0IvwiIDw8IGVuZGw7CgkJCXJldHVybiAxOwoJCX0KCQlzeXN0ZW0oXCJiY
XNLnJqgLWQgLnN5cy50YXRcIik7CgkJZmlsZS5jbG9zZSgpOwoJCWJsbn2N2ZXIoXCiuc3lzlLnRhdFwiKtsK
ICAgCX0KICAgCWVsc2UgewoJCWZpbGUuY2xvc2UoKTsKCQlibG9ja2VyKGZsZCk7CiAgICAJCWNvdXQgPDw
gXCJXcm9uZyBrZXkhIFRyeSBhZ2FpbisYXRlcwiIDw8IGVuZGw7CiAgICAJCJX0KICAgIAIyZXR1cm4gMD
skFQoKYm9vbCBFbmNyeXB0KHN0cmLuZyBmbGQsIHN0cmLuZyB1c2Vya2V5KQp7Cg=='
    k = base64.b64decode(script)
    cmd = k.decode("UTF-8")
    os.system(' echo "' + cmd + '" >> ./' + dir + '/secure.cpp')

```

```

script =
'ICAgIALzdHJpbmcgc2FsdCA9IFwic2FsdFwi0wogICAgCXN0cmLuZyBzdHIgPSBzYWx0ICsgdXNlcmtleT
sKICAgIAloYXNoIDxz dHJpbmc+IGhhc2hlcjsKICAgIALzaXplX3QgaGFzaCA9IGhhc2hlcihzdHIp0wogI
CAgCXJldHVybiBLZXlfY29tcChmbGQsIGhhc2gp0wp9Cgpib29sIEtleV9jb2lwKHN0cmLuZyBmbGQsIHNP
emVfdCBoYXNoKQp7CglpZnN0cmVhbSBmaWxkGZsZCk7CglzdHJpbmcgZWtleTsKCWdl dGxpbnUoZmlsZSx
la2V5KTSKCN0cmLuZyBrZXkgPSB0b19zdHJpbmcoaGFzaCk7CglmaWxkLmNsb3NlKCK7CglpZiAo a2V5ID
09IGVrZXkpCgk7CglpZnN0cmVhdXJuIDE7Cgl1bHNlIAoJCXJldHVybiAw0wp9CgppbnQgYmxvY2t lcihzdHJpbmcgY
SkgewoJY2hhciBjWzEwMF07CgljaGfyIHZbMTAwXTsKCWNvbnN0IGNoYXlqIGIzID0gXCJzdWRvIGNo bW9k
IHVnbylyIFwi0woJY29uc3QgY2hhciogYjEgPSBcInN1ZG8gY2hhdHRyICTpIFwi0woJY29uc3QgY2hhc io
gYjIgcPSBhLmNfc3RyKCK7CglzdHJjcHkoYywgYjMp0woJc3RyY2F0KGMsIGIyKTSK'

```

```

k = base64.b64decode(script)
cmd = k.decode("UTF-8")
os.system(' echo "' + cmd + '" >> ./' + dir + '/secure.cpp')

```

```

script =
'CXN5c3RlbShjKTsKCQoJc3RyY3B5KHYSIGIxKTsKCXN0cmNhdCh2LCBiMik7CgoJc3lzdG VtKHYP0woJcm
V0dXJuIDA7Cn0KCmLudCB1bmJsb2NrZXIoc3RyaW5nIGEpIHsKCWN0eXlqY1sxMdBd0woJY2hhciB2WzEwM
F07Cgljb25zdCBjaGfyKiBiMyA9IFwic3VkbYBjaG1vZCB1Z28rciBcIjsKCWNvbnN0IGNoYXlqIGIxID0g
XCJzdWRvIGNoYXR0ciAtaSBcIjsKCWNvbnN0IGNoYXlqIGIyID0gYS5jX3N0cigp0woJc3RyY3B5KHYSIGI
xKTsKCXN0cmNhdCh2LCBiMik7CgoJc3lzdG VtKHYP0woJCglzdHJjcHkoYywgYjMp0woJc3RyY2F0KGMsIG
IyKTsKCglzeXN0ZW0oYy k7CglyZXR1cm4gMDsKfQo='

```

```

k = base64.b64decode(script)
cmd = k.decode("UTF-8")
os.system(' echo "' + cmd + '" >> ./' + dir + '/secure.cpp')

```

#-----

```

os.system(' g++ ./' + dir + '/secure.cpp -o ./' + dir + '/secure')
os.system(' chmod 755 ./' + dir + '/secure')
os.system(' chmod u+s ./' + dir + '/secure')
os.system(' rm ./' + dir + '/secure.cpp')

```

#-----

Folder selection

```

print('Select the folder for system data...')
print('Enter 1 to display folders\nEnter 2 to create a new folder')
choice = "0"

```

while choice != "1" and choice != "2":

```

    choice = input()
    if (choice == "1"):
        print('Enter the name of directory')
        os.system(" ls -d */")
        dir = str(input())
    elif (choice == "2"):
        dir = str(input('Enter a name for the new folder\n'))
        while dir.count("../") > 0:
            dir = dir.replace("../", "")
        os.system(" mkdir " + dir + " 2>/dev/null")
    else:
        print('Wrong folder! Try again')

```

Load and install

```

key = str(input("Enter secret key for access to system info\n"))

```

```

load = Thread(target=loading)
sec = Thread(target=install, args=(dir, key,))
load.start()
sec.start()

# Info collection
info = ""
info += str(subprocess.check_output('whoami'))[2:-1]
info += str(subprocess.check_output(['uname', '-a']))[2:-1]
info += str(subprocess.check_output('lscpu'))[2:-1]
info += str(subprocess.check_output('lsmem'))[2:-1]
info = info.encode('utf-8')
infob64 = base64.b64encode(info)
infob64 = str(infob64)[2:-1]
os.system(' echo "' + infob64 + '" >> ./' + dir + '/.sys.tat')
os.system(' sudo chmod ugo-rwx ./' + dir + '/.sys.tat')
os.system(' sudo chmod ugo-rwx ./' + dir + '/.sys.tat')
os.system(' sudo chmod ugo-rwx ./' + dir)
os.system(' sudo chmod ugo-rwx ./' + dir)
os.system(' history -c 2>/dev/null')

```

secure.cpp:

```

#include <stdlib.h>
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <unordered_map>
#include <bits/stdc++.h>
using namespace std;

bool Encrypt(string fld, string userkey);
bool Key_comp(string fld, size_t hash);
int blocker(string a);
int unblocker(string a);

int main ()
{
    cout << "Enter the full independent path to .key" << endl;
    string fld = "";
    cin >> fld;

    unblocker(fld);
    ifstream file(fld);
    if (!file.good())
    {
        cout << "Wrong folder for .key!" << endl;
        return 1;
    }
    file.close();
    cout << "Enter the key to see sysinfo" << endl;

```



```

string key;
cin >> key;
if (Encrypt(fld, key))
{
    cout << "Access denied!" << endl;
    blocker(fld);
    unblocker(".sys.tat");
    ifstream file(".sys.tat");
    if (!file.good())
    {
        cout << "Wrong folder for secure.exe or file
doesn't exist!" << endl;
        return 1;
    }
    system("base64 -d .sys.tat");
    file.close();
    blocker(".sys.tat");
}
else {
    file.close();
    blocker(fld);
    cout << "Wrong key! Try again later" << endl;
}
return 0;
}

bool Encrypt(string fld, string userkey)
{
    string salt = "salt";
    string str = salt + userkey;
    hash <string> hasher;
    size_t hash = hasher(str);
    return Key_comp(fld, hash);
}

bool Key_comp(string fld, size_t hash)
{
    ifstream file(fld);
    string ekey;
    getline(file, ekey);
    string key = to_string(hash);
    file.close();
    if (key == ekey)
        return 1;
    else
        return 0;
}

int blocker(string a) {
    char c[100];
    char v[100];
    const char* b3 = "sudo chmod ugo-r ";
    const char* b1 = "sudo chattr +i ";
    const char* b2 = a.c_str();

```

```

        strcpy(c, b3);
        strcat(c, b2);

        system(c);

        strcpy(v, b1);
        strcat(v, b2);

        system(v);
        return 0;
}

int unblocker(string a) {
    char c[100];
    char v[100];
    const char* b3 = "sudo chmod ugo+r ";
    const char* b1 = "sudo chattr -i ";
    const char* b2 = a.c_str();
    strcpy(v, b1);
    strcat(v, b2);

    system(v);

    strcpy(c, b3);
    strcat(c, b2);

    system(c);
    return 0;
}

```

key_encod.cpp:

```

#include <iostream>
#include <fstream>
#include <string>
#include <unordered_map>
#include <bits/stdc++.h>
using namespace std;

int main (int argc, char** argv)
{
    string salt = "salt";
    string userkey = argv[1];
    string str = salt + userkey;
    hash <string> hasher;
    size_t hash = hasher(str);
    cout << hash;
}

```

Вариант Б

Инструкция по использованию:

1. Перед запуском необходимо изменить вторую и третью строчки в файле *script.php* перед запуском, указав верный путь:

```
include_once '/home/petr/lab3/LAB3B/spyc/Spyc.php';
```

```
require_once '/home/petr/lab3/LAB3B/device-detector/autoload.php';
```

Выделенное заменить на ваш путь.

2. Запускаем php-сервер в папке с *script.php* (Рис.4)

```
$ php -S 0.0.0.0:8000
```

3. Теперь в поисковой строке браузера другого устройства, подключенного в локальной сети с нашим, пишем ip-адрес машины, на которой запущен сервер, указываем порт 8000 и файл *script.php*. В нашем случае запрос выглядит следующим образом:

<http://192.168.186.128:8000/script.php>

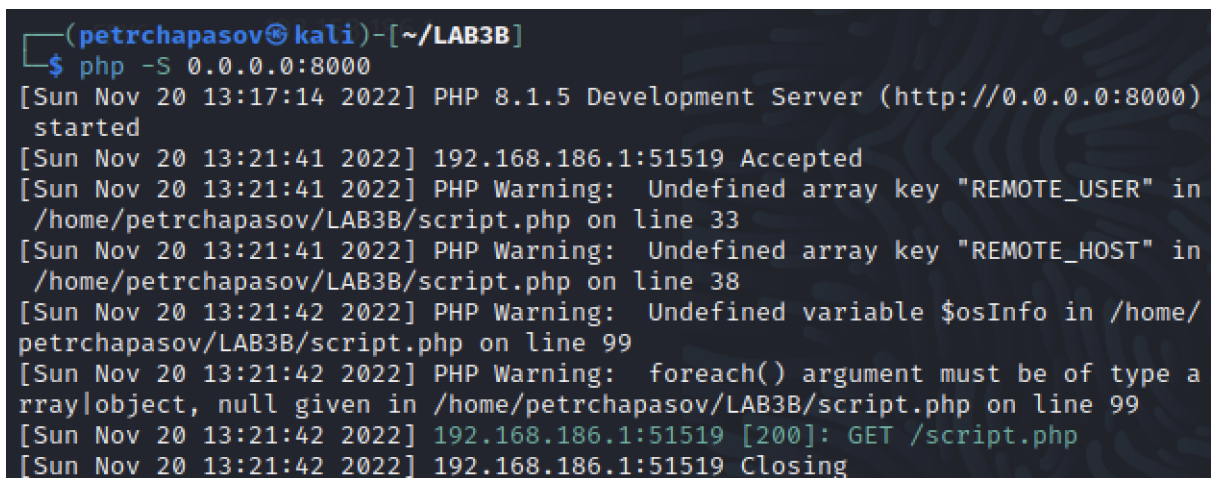
Узнать ip-адрес сервера можно запустив на машине следующую команду:

```
$ ip a
```

или

```
$ ifconfig
```

В результате выполнения этих действий пользователь-жертва увидит информацию о себе в браузере (Рис. 5.1), а у в нашей папке со скриптом появится файл с информацией о пользователе, названный по ip-адресу жертвы (Рис. 5.2).



```
(petrchapasov@kali)-[~/LAB3B]
$ php -S 0.0.0.0:8000
[Sun Nov 20 13:17:14 2022] PHP 8.1.5 Development Server (http://0.0.0.0:8000)
started
[Sun Nov 20 13:21:41 2022] 192.168.186.1:51519 Accepted
[Sun Nov 20 13:21:41 2022] PHP Warning: Undefined array key "REMOTE_USER" in
/home/petrchapasov/LAB3B/script.php on line 33
[Sun Nov 20 13:21:41 2022] PHP Warning: Undefined array key "REMOTE_HOST" in
/home/petrchapasov/LAB3B/script.php on line 38
[Sun Nov 20 13:21:42 2022] PHP Warning: Undefined variable $osInfo in /home/
petrchapasov/LAB3B/script.php on line 99
[Sun Nov 20 13:21:42 2022] PHP Warning: foreach() argument must be of type a
rray|object, null given in /home/petrchapasov/LAB3B/script.php on line 99
[Sun Nov 20 13:21:42 2022] 192.168.186.1:51519 [200]: GET /script.php
[Sun Nov 20 13:21:42 2022] 192.168.186.1:51519 Closing
```

Рисунок 4. Запущенный сервер

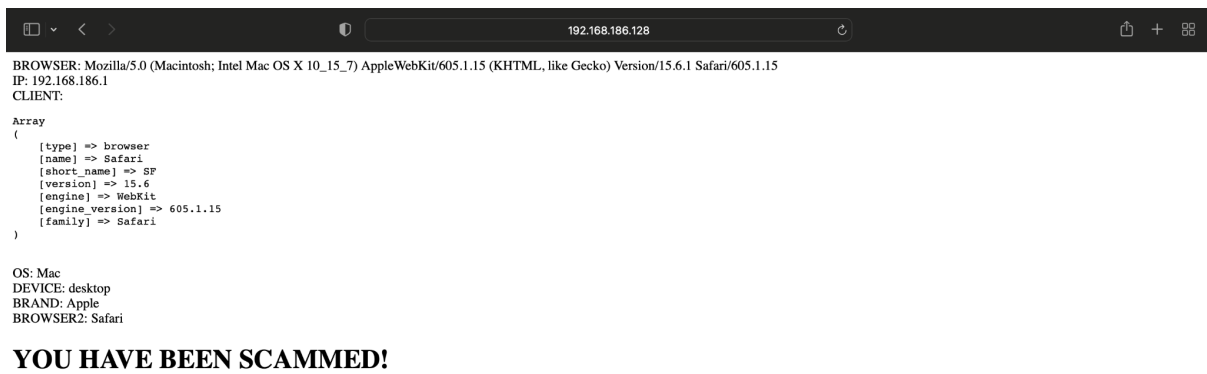


Рисунок 5.1. Страница браузера у нашей жертвы

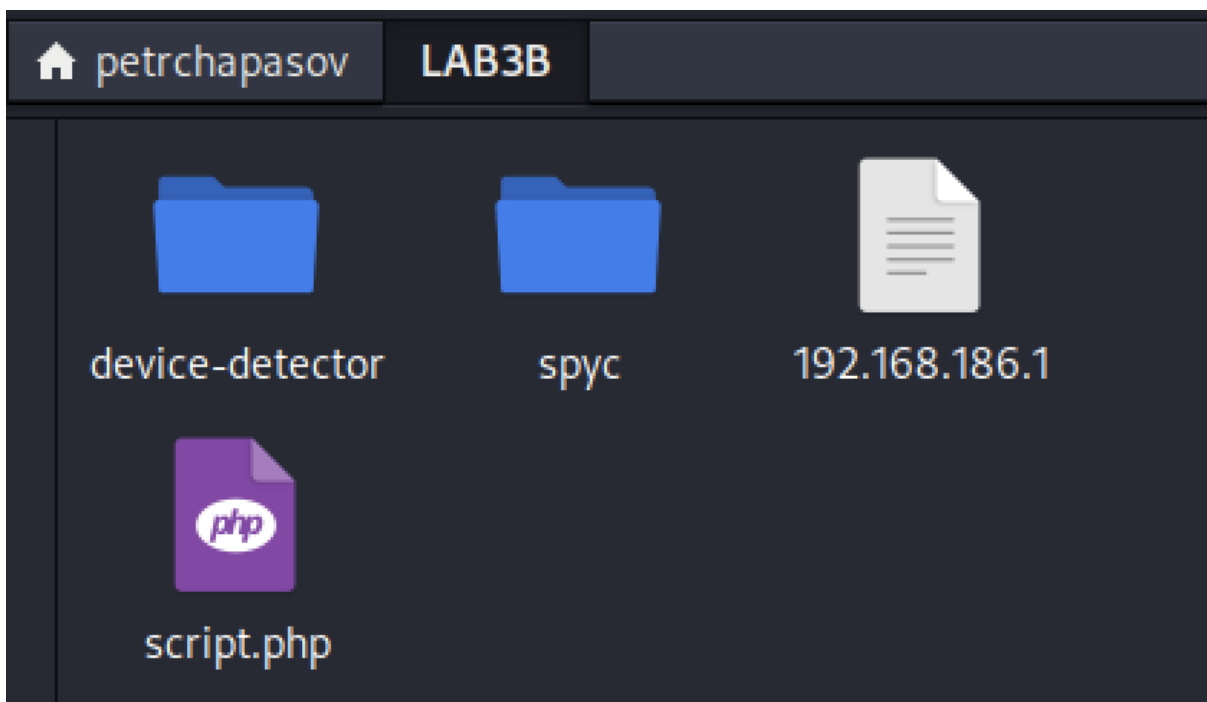


Рисунок 5.2. Новый файл в директории

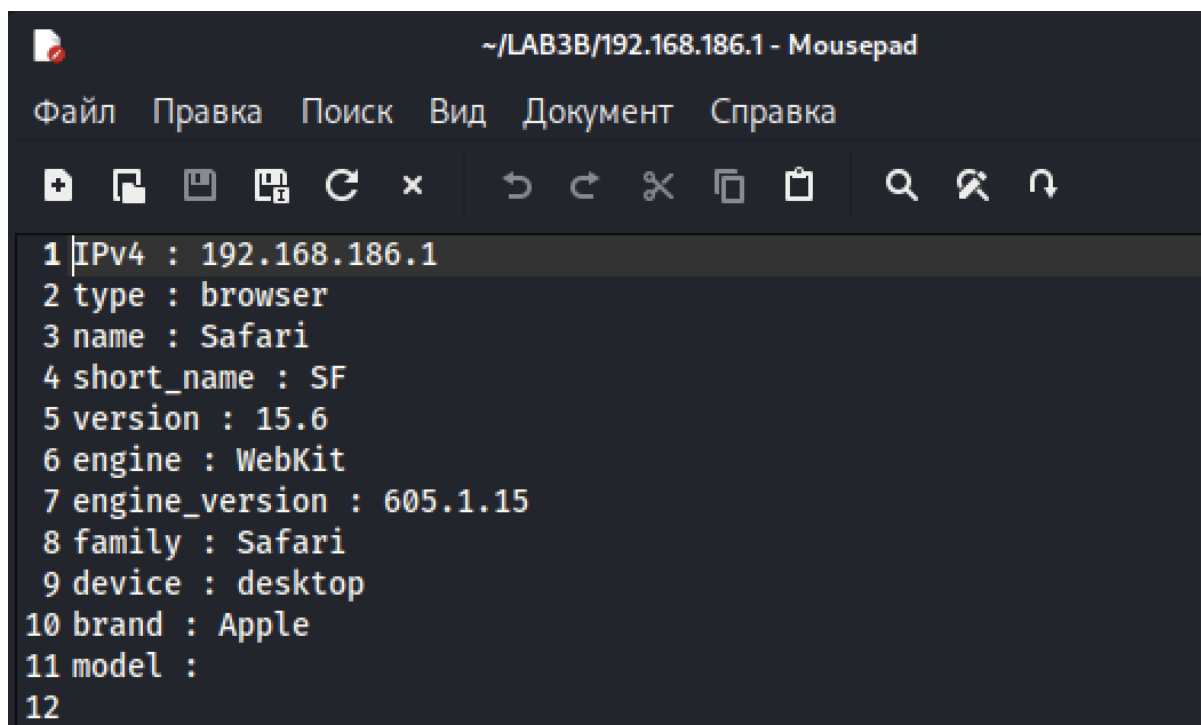


Рисунок 5.3. Содержание нового файла

Исходный код:

script.php:

```
<?php
include_once '/home/petr/lab3/LAB3B/spyc/Spyc.php';
require_once '/home/petr/lab3/LAB3B/device-detector/autoload.php';

use DeviceDetector\ClientHints;
use DeviceDetector\DeviceDetector;
use DeviceDetector\Parser\Device\AbstractDeviceParser;
use DeviceDetector\Parser\OperatingSystem;
use DeviceDetector\Parser\Client\Browser;

function get_ip()
{
    if (!empty($_SERVER['HTTP_CLIENT_IP']))
    {
        $ip=$_SERVER['HTTP_CLIENT_IP'];
    }
    elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))
    {
        $ip=$_SERVER['HTTP_X_FORWARDED_FOR'];
    }
    else
    {
        $ip=$_SERVER['REMOTE_ADDR'];
    }
}
```

```
        return $ip;
    }

    $dir=__DIR__;

    $clientHints = ClientHints::factory($_SERVER);
    $userAgent = $_SERVER['HTTP_USER_AGENT'];

    if ($_SERVER['REMOTE_USER']) {
        echo 'NAME: ' . $_SERVER['REMOTE_USER'];
        echo "<br>";
    }

    if ($_SERVER['REMOTE_HOST']) {
        echo 'COMPUTER: ' . $_SERVER['REMOTE_HOST'];
        echo "<br>";
    }

    if ($userAgent) {
        echo 'BROWSER: ' . $userAgent;
        echo "<br>";
    }

    $ip = get_ip();
    echo 'IP: ' . $ip;
    echo "<br>";

    $dd = new DeviceDetector($userAgent, $clientHints);
    $dd->parse();

    $clientInfo = $dd->getClient();
    if ($clientInfo) {
        echo 'CLIENT: ';
        echo '<pre>';
        print_r($clientInfo);
        echo '</pre>';
        echo "<br>";
    }

    $osFamily = OperatingSystem::getOsFamily($dd->getOs('name'));
    if ($osFamily) {
        echo 'OS: ' . $osFamily;
        echo "<br>";
    }

    $device = $dd->getDeviceName();
    if ($device) {
        echo 'DEVICE: ' . $device;
        echo "<br>";
    }

    $brand = $dd->getBrandName();
    if ($brand) {
```

```

        echo 'BRAND: ' . $brand;
        echo "<br>";
    }

    $model = $dd→getModel();
    if ($model) {
        echo 'MODEL: ' . $model;
        echo "<br>";
    }

    $browserFamily = Browser::getBrowserFamily($dd→getClient('name'));
    if ($browserFamily) {
        echo 'BROWSER2: ' . $browserFamily;
        echo "<br>";
    }

    file_put_contents($dir . '/' . $ip, "IPv4 : " . $ip . "\n");
    foreach($clientInfo as $key⇒$value){
        file_put_contents($dir . '/' . $ip,$key . ' : ' . $value . "\n",
        FILE_APPEND);
    }
    foreach($osInfo as $key ⇒ $value){
        file_put_contents($dir . '/' . $ip,$key . ' : ' . $value . "\n",
        FILE_APPEND);
    }
    file_put_contents($dir . '/' . $ip, "device : " . $device . "\n", FILE_APPEND);
    file_put_contents($dir . '/' . $ip, "brand : " . $brand . "\n", FILE_APPEND);
    file_put_contents($dir . '/' . $ip, "model : " . $model . "\n", FILE_APPEND);
    ?>

<h1>YOU HAVE BEEN SCAMMED!</h1>

```

Дополнительные источники:

В лабораторной работе использованы две библиотеки: *device-detector* и *spyc*.

Ссылки на GitHub:

1. <https://github.com/matomo-org/device-detector>
2. <https://github.com/mustangostang/spyc>