МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ САХА (ЯКУТИЯ)
МУНИЦИПАЛЬНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКИЙ ЛИЦЕЙ № 24 г. НЕРЮНГРИ
678960 Республика Саха (Якутия), г.Нерюнгри, пр. Ленина 12/1 тел.6-21-37, факс 7-68-18

НОУ «ИНТЕЛЛЕКТ 21 ВЕК»

Беспроводная точка доступа с многоуровневым алгоритмом защиты данных

Выполнил:
Куницын Кирилл 9А
Руководители проекта:
Дёминов Сергей Иванович, учитель технологии
Годизов Олег Александрович, учитель робототехники

Нерюнгри 2018 г.

Содержание

Абстракт

Разработан многоуровневый алгоритм защиты передачи пакета данных, передающихся через беспроводной маршрутизатор. В создании алгоритма использованы несколько отдельных средств шифрования данных, которые объединены в одну систему, то есть, прописаны последовательно в одну консоль и работают параллельно через защищённые каналы связи.

В качестве основного механизма защиты служит луковичная маршрутизация (onion routing -TOR), которая позволяет направить трафик через несколько анонимных и безопасных серверов. Поскольку каждая технология имеет свои уязвимости и OR не исключение, во избежание потери данных, вся информация заранее шифруется, предварительно проходя еще один этап с использованием сетевого протокола SOCKet Secure-5 (SOCKS5) с параллельным использованием технологии DNS-прокси. Данный алгоритм обеспечивает процедуру авторизации, что позволяет открыть полный доступ к информации только легитимным пользователям, тем самым защищая их от третьих лиц.

Введение

Сегодня Интернет превратился в неотъемлемую часть нашей жизни. От общения с банком до покупок в путешествии, каждый аспект нашей жизни связан с глобальной сетью.

В настоящее время такое понятие, как кибератака является очень распространенным, и оно становится все большей головной болью для многих ИТ-компаний и предприятий. В последние годы некоторые крупные имена, такие как Google, The New York Times, Facebook и т. д., стали жертвами взломов.

Большинство информации в сетях передаётся и хранится в открытом виде. Например, при входе на форум, вводится логин и пароль, написали сообщение — и логин, и пароль, и сообщение передаётся в открытом виде, в качестве простого текста. Причём, в передачи данных участвуют много узлов и почти на каждом из них возможен перехват данных. Это возможно как в локальной сети начинающим хакером, который скачал программу для пентестинга беспроводных сетей и сумел подобрать пароль от вашего Wi-Fi, это возможно на уровне городского провайдера, где сидит продвинутый и не в меру любопытный администратор, это возможно на последующих узлах вплоть до хостинга того форума, где вы общаетесь.

Чтобы хоть как-то защититься от этого, популярные сайты (почтовые службы, социальные сети и другие) обзавелись сертификатами, смысл их в том, что обмен данными между сайтом и вами теперь происходит в зашифрованном виде. Т.е. теперь начинающий хакер, продвинутый администратор и другие лица по цепочке не смогут так легко перехватить ваши данные.

Для защиты персональных данных пользователя существует множество алгоритмов защиты, каждый из которых в свою очередь имеет ряд уязвимостей. Поэтому целесообразно создание многоуровневого алгоритма защиты персональных данный, который объединяет в себе несколько технологий обеспечения безопасности.

Разработка алгоритма защиты

В разрабатываемом алгоритме защиты будут использованы следующие технологии:

- Tor;
- Socks5;
- DNS proxy.

Tor (The onion routing) — Технология луковичной маршрутизации, которая является распределенной системой серверов, между которыми трафик проходит в зашифрованном виде. На последнем узле в цепочке передаваемые данные проходят процедуру расшифровки и передаются целевому серверу в открытом виде.

В качестве аналога технологии Tor была рассмотрена технология i2p. I2p(invisible internet project) - проект, создания анонимной компьютерной сети, работающей поверх глобальной сети Интернет. i2p является более сложным решением для рядовых пользователей, при этом имеет ряд недостатков, таких как отсутствие децентрализованности серверов и недостаточно хорошие методы шифрования.

В ходе сравнения двух технологий было принято решение использовать Tor, поскольку он является более удобным и более функциональным методом обеспечения безопасности.

Технология Tor уязвима к атаке, в процессе которой подменяется последний узел в сети и трафик, проходящий через неё, может попасть к злоумышленнику, проводившему атаку. Для того чтобы не передавать трафик в открытом виде даже на последнем узле, трафик предварительно шифруется с помощью технологии Socks5.

Socks5 – сетевой протокол, который позволяет пересылать пакеты от клиента к серверу через прокси-сервер, шифруя данные в процессе передачи. В данном случае используется протокол с возможностью авторизации, что позволяет получить доступ к данному серверу только легитимным пользователям.

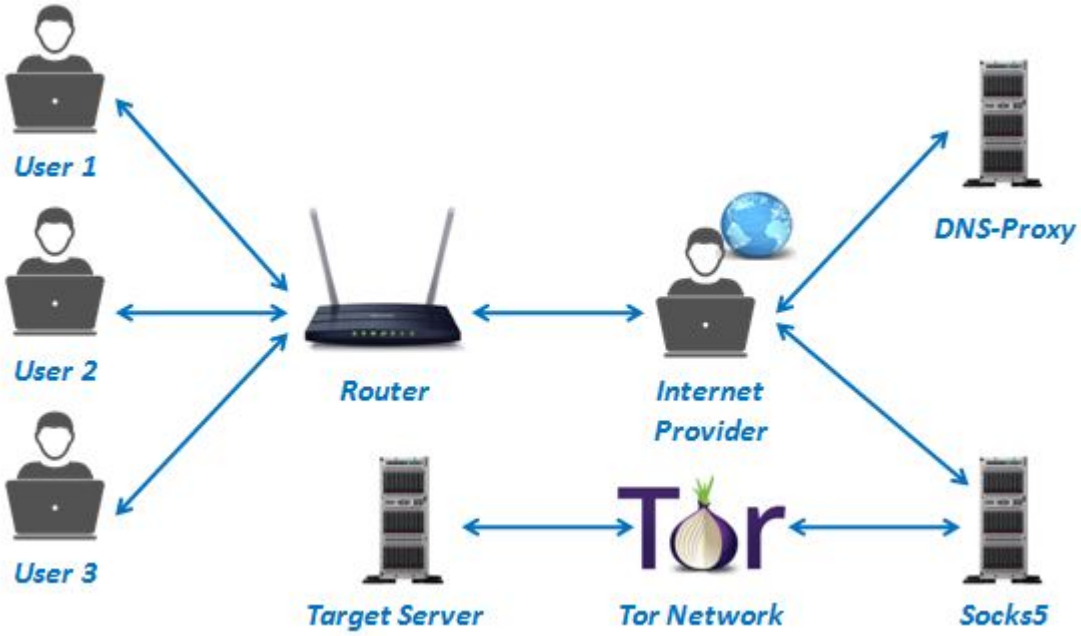
Аналогом для протокола Socks5 может служить VPN-сервер. VPN (Virtual Privat Network) – это сервер, который также шифрует данные при передаче, но в отличии от выбранной технологии Socks5, появляется необходимость аренды VPN-сервера, что в свою очередь ведет к дополнительным финансовым затратам. При этом настройка данной технологии возможна самостоятельно, но требует больших временных затрат и специализированных навыков в данной сфере.

Технологии Tor вместе с технологией Socks5 обеспечивают безопасность передачи данных пользователя, исключая большинство возможных атак.

Разрабатываемый многоуровневый алгоритм защиты пользовательских данных позволяет исключить вариант подмены целевого сервера. За данную функцию отвечает технология DNS proxy.

DNS (Система доменных имен) - это протокол, который интерпретирует буквенное доменное имя в IP-адрес. Говоря простым языком, его основная функция — превратить удобное для пользователя имя домена в IP-адрес.

Данный протокол подвержен нескольким видам атак, таких как DNS-spoofing и Fast Flux DNS, в целях предотвращения данных атак, используется технология DNS proxy, которая позволяет задействовать частный DNS-сервер, тем самым защищая пользователя от утечки данных через систему доменных имён. Схема работы многоуровневого алгоритма защиты данных представлен на рисунке 1.



Для реализации разработанного многоуровневого алгоритма защиты и для предоставления безопасного доступа к глобальной сети Интернет нескольких пользователей одновременно, требуется устройство, которое может выступать в роли точки доступа с возможностью построения маршрута пользователя к сети Интернет с использованием алгоритма шифрования и защиты данных.

Выбор элементной базы и расчет стоимости портативного роутера

В качестве аппаратной платформы для создания портативной беспроводной точки доступа с системой защиты персональных данных был выбран микрокомпьютер Raspberry Pi 3B, который оптимально подходит для создания подобного рода проектов.

Raspberry Pi– одноплатный компьютер, работающий на Raspbian (-unix подобной операционной системе), разработанный Дэвидом Брэбеном в 2011 году для студентов, главной особенностью которого должна была стать мощная аппаратная начинка и бюджетная цена. Он достаточно быстро обрел популярность, благодаря своей простоте использования, широким предоставляемым возможностям, невысокой цене, и сейчас является самым популярным одноплатным компьютером в мире.

На сегодняшний день множество исследовательских проектов в сфере технического творчества, таких как игровые приставки, серверы для хранения информации, медиаплееры и многие другие, проектируются на основе одноплатного микрокомпьютера Raspberry Pi.

Выбранная нами модель Raspberry Pi 3B отличается встроенным WIFI адаптером, Bluetooth, Ethernet портом, а также невысокой ценой и современными техническими характеристиками. Сравнение одноплатных микрокомпьютеров Raspberry Pi различных моделей представлено в таблице 1.

Таблица 1. Сравление различных версий микрокомпьютеров Raspberry Pi.

Версия	Процессор	Частота	Ядер	ОЗУ	GPIO	USB	Ethernet	Wi-Fi
B	ARM1176JZ-F	700 МГц	1	512 Мб	26 пинов	2 порта	есть	нет
2B	ARM Cortex-A7	900 МГц	4	1 Гб	40 пинов	4 порта	есть	нет
3B	ARM Cortex-A53 x64	1,2 ГГц	4	1 Гб	40 пинов	4 порта	есть	802.11n

Корпус роутера изготовлен из PLA пластика с применением технологии 3d печати. PLA-пластик (полилактид) - является биоразлагаемым, биосовместимым, термопластичным алифатическим полиэфиром, структурная единица которого - молочная кислота. Данный материал достаточно прочный, при ударах трескается, но не рассыпается на осколки, легче, чем сталь, алюминий и многие другие материалы, не подвержен коррозии, что облегчает эксплуатацию устройства.

В таблице 2 указан перечень всех компонентов, входящих в состав портативного роутера с указанием приблизительной стоимости каждого элемента.

Таблица 2.Перечень элементов

№	Наименование	Кол-во, шт.	Цена, руб.	Итого, руб.
1	Микрокомпьютер Raspberry Pi 3B	1	1910	1910
2	Корпус из PLA-пластика	1	100	100
3	Карта памяти microSDHC 8Gb	1	390	390
4	Кабель USB – microUSB	1	190	190
Итого				2590

Стоимость проекта суммарно составляет примерно 2590 рублей, что вполне сопоставимо со стоимостью хорошего WiFi роутера, но готовые решения на рынке не способны обеспечить достаточную степень анонимности и защиты персональных данных рядовых пользователей глобальной сети Internet от злоумышленников. Созданный в ходе исследовательской работы портативный WiFi роутер со встроенной многоуровневой системой защиты данных не имеет аналогов среди заводских моделей ведущих фирм производителей.

Установка программного обеспечения и настройка портативной беспроводной точки доступа

Для начала на сайте официального производителя скачиваем образ операционной системы Raspbian и создаем загрузочную карту памяти micro SD с помощью программы Win32DiskImager (Рисунок 2, Рисунок 3)

Устанавливаем карту памяти в одноплатный микрокомпьютер Raspberry Pi 3B, подключаем монитор, для вывода данных и работы с оболочкой через цифровой порт HDMI, Ethernet кабель и питание через Micro USB (Рисунок 4).

Далее через терминал операционной системы Linux с помощью apt-get (пакетный менеджер) устанавливаем следующие пакеты и утилиты: (Рисунок 5)

- top;
- psmisc;
- git;
- make;
- gcc;
- python3-pip;
- hostapd;
- iptables-persistent;
- wvdial;
- tor;
- tor-arm;
- dnsmasq;
- netdiag;
- tcpdump;
- pppoeconf.

Затем с помощью консоли скачиваем и устанавливаем утилиту для работы с proxy-сервером из репозитория на github (Рисунок 6).

Далее с помощью консоли скачиваем и устанавливаем утилиту для работы dns proxy-сервером из репозитория на github (Рисунок 7).

Настраиваем фаервол для работы веб-приложений (Рисунок 8).

Прописываем настройки установленных утилит в конфигурационных файлах dnsproxy, dnsmasq, resolv, interfaces, hostapd, wvdial, sysctl. (Рисунок 9, Рисунок 10)

Затем с помощью терминала запускаем установленные и настроенные утилиты, добавляем в автозагрузку скрипт (hostapd) для поднятия точки доступа при включении Raspberry (Рисунок 11).

Далее после загрузки пакетов и настройки необходимых утилит приступаем к Разработке web-панели для изменения параметров WiFi роутера.

Установим зависимости для python3 с помощью пакетного менеджера pip3 (Рисунок 12): Flask, requests, pysocks.

Верстаем интерфейс для web-панели (Рисунок 13, Рисунок 14).

Программируем web-панель на основе фреймворка Flask (Рисунок 15).

Исследования

Сравнить созданный прототип с заводским аналогом. В качестве аналога был выбран WiFi роутер D-link DIR-615, который входит в перечень самых популярных решений для домашнего пользования. Сравнение изготовленного прототипа и заводского аналога представлено в таблице 3.

Таблица 3. Сравнение роутера с заводским аналогом

Наименование	Оптимальный радиус действия (м)	Размеры (мм)	Доп. функции	Цена
Raspberry Pi 3B	~20 метров	95x75x23	4 USB, Bluetooth, HDMI	2590руб.
D-Link DIR 615	~20 метров	175x123x31	4 LAN, WPS	1300руб.

Вывод: Прототип на базе микрокомпьютера Raspberry Pi 3B по заявленным дальности действия и уровню сигнала не уступает заводскому аналогу, при этом имеет более компактные габариты и возможность подключения периферийных устройств, для расширения возможностей и функционала.

Провести замеры скорости Wi-Fi сигнала на различных дистанциях и сравнить со стандартными моделями WiFI роутеров различных производителей. Замеры производились при помощи приложения NetSpot. (Рисунок 16, Рисунок 17, Рисунок 18, Рисунок 19). Полученные в ходе исследования данные представлены в таблице 4.

Таблица 4. Замеры скорости WiFi

Наименование	Дальность устройства от источника сигнала (м)			
	5	10	15	20
Raspberry Pi 3B	64%	47%	33%	23%
Dlink DIR-615	63%	51%	23%	13%
ZyXEL Keenetic	53%	55%	34%	25%

Вывод: Прототип на базе микрокомпьютера Raspberry Pi 3B по уровню сигнала на различных дистанциях проведения замеров не уступает заводскому аналогу.

Проверить анонимность пользователя, подключенного к глобальной сети Интернет, через портативную беспроводную точку доступа с многоуровневой системой защиты, используя алгоритм тестирования подключения через следующие ресурсы:

- www.2ip.ru/privacy (Рисунок 20);
- https://proxy6.net/privacy (Рисунок 21).

Вывод: Средства проверки анонимности по средствам тестирования подключения пользователя к глобальной сети Интернет не смогли определить реальное местонахождение пользователя или его фактический ip адрес.

Создание элементов корпуса при помощи процедуры 3d-печати

В ходе работы в 3D-редакторе «SolidWorks» были разработаны объемные модели корпуса для Raspberry Pi 3B.

SolidWorks — программный комплекс для автоматизации работ промышленного предприятия на этапах конструкторской и технологической подготовки производства. Обеспечивает разработку объемных моделей изделий любой степени сложности и назначения.

Элементы корпуса представлены на рисунке ниже (Рисунок 22, Рисунок 23).

Заключение

В результате исследовательской работы, на основе микрокомпьютера Raspberry PI 3Вбыла создана беспроводная точка доступа (роутер) с разработанным многоуровневым алгоритмом защиты, который способен шифровать входящий и исходящий трафик, а также обеспечивать анонимность пользователя, при работе в глобальной сети Интернет. Разработанная в исследовательском проекте беспроводная точка доступа (роутер), помимо высокой степени защиты данных, обладает компактными размерами, возможностью питания от 5 вольт в отличие от заводских аналогов.

Список использованной литературы

1. https://ru.wikipedia.org/wiki/Raspberry_Pi
2. http://raspberrypi.ru/
3. https://www.raspberrypi.org
4. https://ru.wikipedia.org/wiki/Tor
5. https://www.altlinux.org/Hostapd
6. http://dmitrysnotes.ru/raspberry-pi-3-prisvoenie-staticheskogo-ip-adresa
7. https://ru.wikipedia.org/wiki/DHCP
8. https://2ip.ru/

Приложения

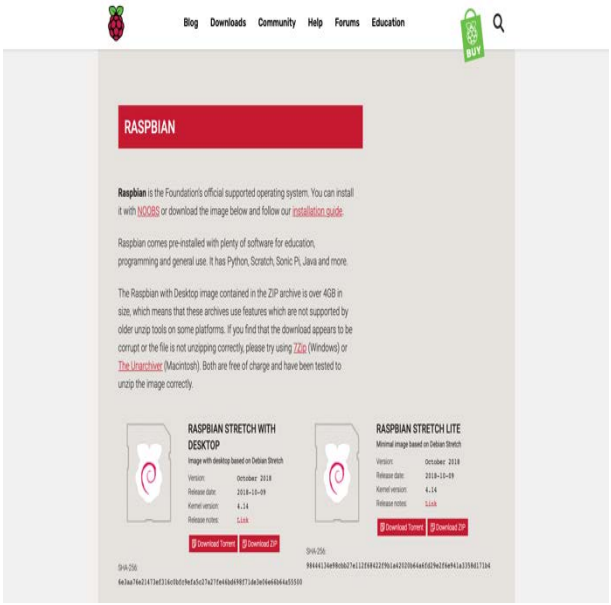


Рисунок 2. Скачивание образа системы

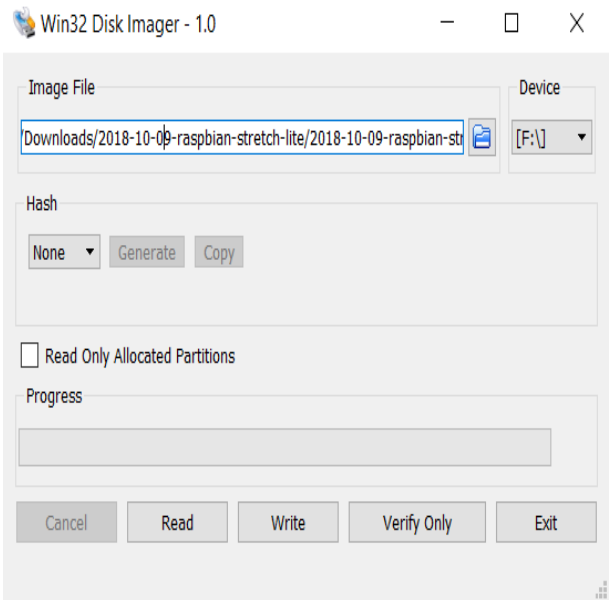


Рисунок 3. Запись образа на карту памяти

Рисунок 4. ОС Rasbian Lite

```
apt-get update
apt-get upgrade -y
apt-get install -y mc htop psmisc git make gcc python3-pip hostapd iptables-persistent wvdial tor tor-arm dnsmasq netdiag tcpdump
```

Рисунок 5. Установка пакетов утилит

```
cd ~
git clone https://github.com/z3APA3A/3proxy
cd 3proxy/ && make -f Makefile.Linux
mkdir -p /lib/3proxy/
mv src/TransparentPlugin.ld.so /lib/3proxy/
mv src/PCREPlugin.ld.so /lib/3proxy/
mv src/TrafficPlugin.ld.so /lib/3proxy/
mv src/StringsPlugin.ld.so /lib/3proxy/
mv src/3proxy /usr/bin/
cd ~
rm -rf 3proxy
```

Рисунок 6. Установка proxy

```
cd ~
git clone https://github.com/jtripper/dns-tcp-socks-proxy
cd dns-tcp-socks-proxy && make
mkdir -p /etc/dnsproxy
mv dns_proxy.conf /etc/dnsproxy/
mv dns_proxy /usr/bin/
cd ~
rm -rf dns-tcp-socks-proxy
```

Рисунок 7. Установка dns proxy

```
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j REDIRECT --to-ports 22
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 5000 -j REDIRECT --to-ports 5000
iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j REDIRECT --to-ports 6666
iptables-save > /etc/iptables/rules.v4
```

Рисунок 8. Настраиваем Firewall

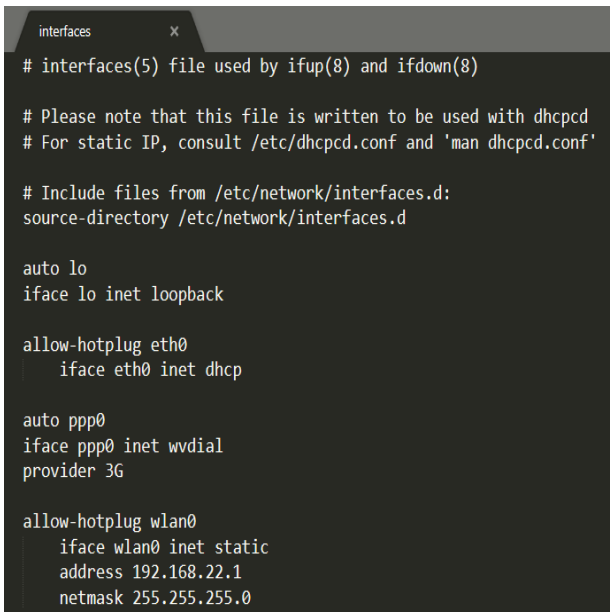


Рисунок 9. Interfaces

Рисунок 10. hostapd.conf

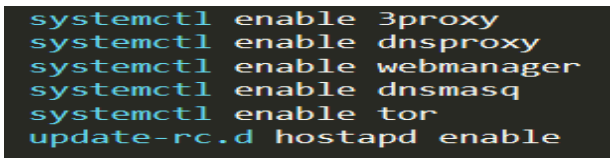


Рисунок 11. Запускаем утилиты

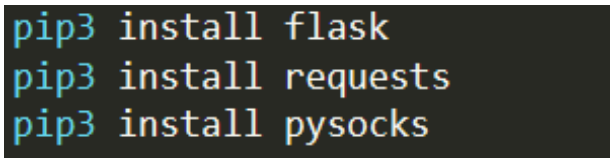


Рисунок 12. Установка зависимостей python

index.html

```
{% include 'header.html' %}

<div id="content">
      <table>
    <tr>
      <th>System status :</th>
      <th>{{uptime}}</th>
    </tr>
    <tr>
      <td>Curent Mode</td>
      <td>{{mode}}</td>
    </tr>
    <tr>
      <td>Status TOR</td>
      <td>{{tor_status}}</td>
    </tr>
    <tr>
      <td>Status 3proxy</td>
      <td>{{proxy_status}}</td>
    </tr>
    <tr>
      <td>Status DNS proxy</td>
      <td>{{dnsproxy_status}}</td>
    </tr>
    <tr>
      <td>Status 3G</td>
      <td>{{vwdial_status}}</td>
    </tr>
</table>

</div>

{% include 'footer.html' %}
```

Рисунок 13. Index.html

main.css

```
body{
      height:100%;
      background:#2A2E33;
      color:#CCCFD4;
      font-family:verdana,tahoma,arial;
}

#menu {
      width: 800px;
      margin-left: auto;
      margin-right: auto;
      margin-top: 20px;
      font-weight:bold;
      font-size:10pt;
      font-family:verdana,tahoma,arial;
      text-decoration: none;
    }

#content {
      margin-top: 100px;
      margin-right: auto;
      margin-left: auto;
      width: 800px;
      height: 400px;
      text-align: center;
}

a {
      color: whitesmoke;
      text-decoration: none;
      font-weight:bold;
}

#copyright {
      margin-top: 50px;
      text-align: center;
}

table {
      border-collapse: collapse;
}
```

Рисунок 14. Main.css

```
import os
import time
import subprocess
import config

def check_tor():
    try:
        tor = subprocess.check_output(['systemctl', 'is-active', 'tor'])
        return 'Active'
    except subprocess.CalledProcessError:
        return 'Down'

def check_3proxy():
    try:
        proxy3 = subprocess.check_output(['systemctl', 'is-active', '3proxy'])
        return 'Active'
    except subprocess.CalledProcessError:
        return 'Down'

def check_dnsproxy():
    try:
        dnsproxy = subprocess.check_output(['systemctl', 'is-active', 'dnsproxy'])
        return 'Active'
    except subprocess.CalledProcessError:
        return 'Down'

def check_wvdial():
    try:
        wvdial = int(subprocess.check_output(["pidof", "-s", "wvdial"]))
        return 'Connected'
    except subprocess.CalledProcessError:
        return 'Down'

def get_pid_wvdial():
    try:
        wvdial = int(subprocess.check_output(["pidof", "-s", "wvdial"]))
        return wvdial
    except subprocess.CalledProcessError:
        return 'none'
```

Рисунок 15. Control.py

Рисунок 16. Уровень WiFi сигнала на расстоянии 5 метров

Рисунок 17. Уровень WiFi сигнала на расстоянии 10 метров

Рисунок 18. Уровень WiFi сигнала на расстоянии 15 метров

Рисунок 19. Уровень WiFi сигнала на расстоянии 20 метров

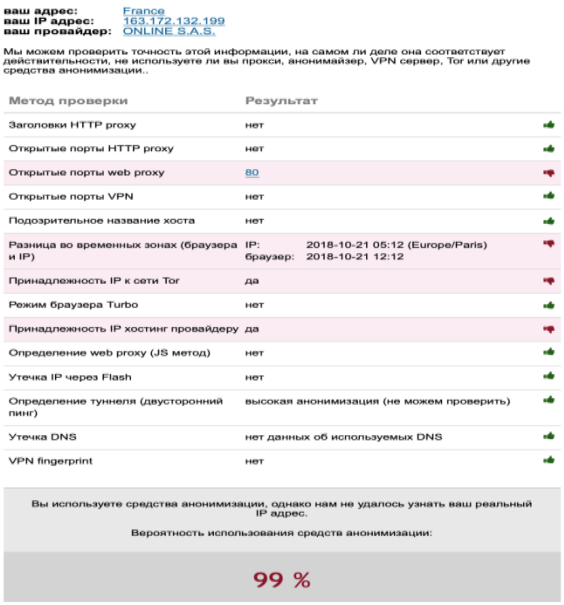


Рисунок 20. Проверка анонимности(2ip.ru)

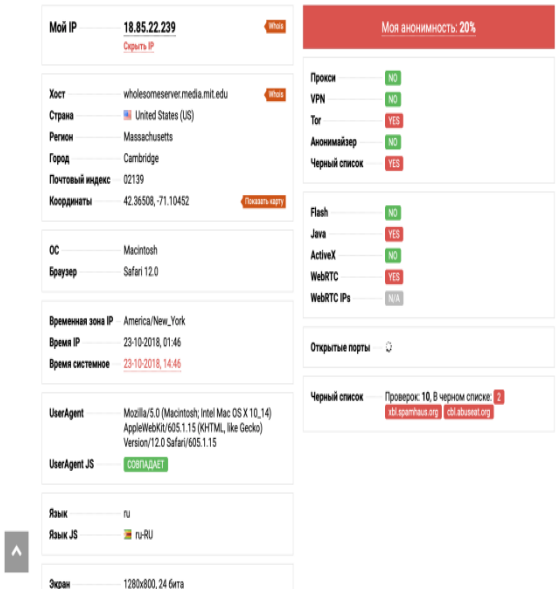


Рисунок 21. Проверка анонимности(proxy6.net)

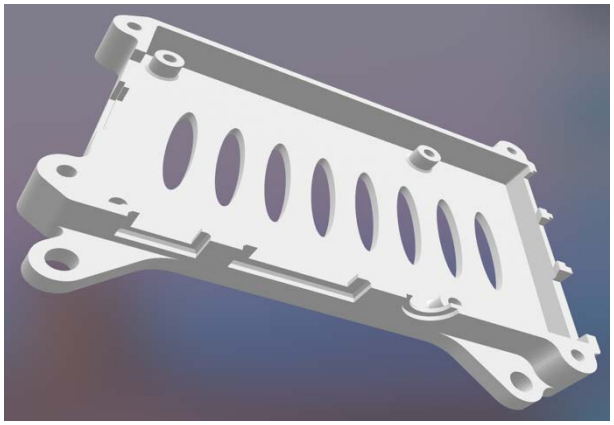


Рисунок 22. Задняя крышка

Рисунок 23. Задняя крышка