



**Московский государственный университет имени М.В. Ломоносова**

Факультет вычислительной математики и кибернетики

Кафедра автоматизации систем вычислительных комплексов

Лаборатория безопасности информационных систем

Глущенко Майя Сергеевна

**Автоматизация поиска небезопасных настроек  
для сетевых сервисов**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Научный руководитель:**

М. Н. С.

А. А. Петухов

Москва, 2019

## **Аннотация**

В данной работе проводится исследование уровня защищенности устройств российского сегмента Интернета вещей в контексте доступа к их веб-интерфейсам через стандартный пароль.

Приводится обзор существующих подходов к решению задач, возникающих в процессе конструирования инструмента для автоматизированного поиска небезопасных настроек рассматриваемых устройств, предлагается использование различных методов определения стратегии подбора стандартных паролей. Выбор применяемого подхода зависит от реализации механизма обработки аутентификационных данных в каждом исследуемом устройстве.

Приведены результаты экспериментов, полученные с помощью сконструированного инструмента, а также проведен их анализ.

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
1.1	Цель работы . . . . .	4
1.2	Постановка задачи . . . . .	4
1.3	Актуальность задачи . . . . .	4
1.4	Замечания по терминологии . . . . .	5
<b>2</b>	<b>Анализ задачи</b>	<b>6</b>
2.1	Получение списка IP-адресов целевых устройств . . . . .	6
2.2	Получение списков стандартных паролей компаний-производителей и моде- лей целевых устройств . . . . .	6
2.3	Фингерпринтинг устройства. Сопоставление устройству списка стандартных паролей . . . . .	6
2.4	Унификация метода подбора паролей для различных административных па- нелей . . . . .	6
2.5	Проблема блокировки полей веб-формы. Определение критерия успеха для каждой рассматриваемой комбинации логина и пароля . . . . .	7
<b>3</b>	<b>Обзор существующих решений</b>	<b>9</b>
3.1	Методы получения списков исследуемых IP-адресов . . . . .	9
3.1.1	Сканирование диапазонов IP-адресов . . . . .	9
3.1.2	Поисковые системы . . . . .	11
3.1.3	Выводы . . . . .	12
3.2	Методы получения списков стандартных паролей для исследуемых устройств	12
3.3	Фингерпринтинг устройств . . . . .	13
3.4	Задача унификации методов подбора паролей для устройств . . . . .	17
3.5	Обход блокировки полей ввода веб-формы. Методы определения критерия успеха в процессе подбора комбинаций . . . . .	20
<b>4</b>	<b>Реализация прототипного инструмента</b>	<b>23</b>
4.1	Обоснование выбора инструментария . . . . .	23
4.2	Архитектура инструмента. Общая схема работы . . . . .	24
4.3	Детали реализации . . . . .	26
<b>5</b>	<b>Проведение эксперимента</b>	<b>28</b>
5.1	Маршрутизаторы . . . . .	28
5.2	Сетевые камеры . . . . .	30
5.3	Выводы . . . . .	31
<b>6</b>	<b>Результаты</b>	<b>33</b>
<b>7</b>	<b>Список литературы</b>	<b>34</b>

# 1 Введение

Тема Интернета вещей в течение последних нескольких лет набирает все большую популярность. Помимо внушительного количества современных научных исследований<sup>1</sup> существует множество тематических материалов<sup>2</sup> среди направленных на широкую аудиторию информационных ресурсов, заявляющих о невероятном потенциале Интернета вещей. Итак, термин IoT (Internet of things, Интернет вещей) включает в себя следующее: это понятие, объединяющее в себе технологии, устройства, платформы и приложения, так или иначе связанные с подключением этих устройств к сети. В группу устройств можно отнести, к примеру, бытовые приборы, открывающие новые возможности при сетевом подключении: умный холодильник самостоятельно закажет еду, датчик дыма пришлет уведомление, чтобы владелец вовремя вызвал пожарных. Сюда также можно отнести устройства, существование которых без доступа к сети не имело бы смысла: маршрутизаторы, коммутаторы, камеры видеонаблюдения и т.д.

Как правило, настройка и управление устройством осуществляется через 22, 23 и 80/443 сетевые порты, за которые отвечают сетевые протоколы telnet, ssh и http/https. Благодаря простоте в использовании самым популярным инструментом для управления стал веб-интерфейс, или административная панель. Чтобы обеспечить базовый уровень защищенности устройства, рекомендуется своевременно устанавливать обновления программного обеспечения и заменять стандартный пароль на более сложный. Однако большинство конечных пользователей не выполняют этих предписаний. Так, согласно исследованию британской компании Broadband Genie, 82% из 2,205 совершеннолетних респондентов никогда не меняли пароль доступа к настройкам своего маршрутизатора [1]. Не менее впечатляющим оказался тот факт, что 48% опрошенных не имели представления, зачем это нужно, в то время как 34% признались, что не знают, как поменять пароль администратора.

К сожалению, обладатели уязвимых устройств не только рискуют персональной безопасностью, но и поневоле могут причинять вред другим людям, не подозревая об этом. Так, в сентябре 2016 года мир узнал о ботнете Mirai после того, как с помощью него была произведена DDoS-атака (атака, в результате которой цель перегружается большим количеством интернет-трафика, что приводит к отказу сервиса) на американскую компанию Дун, которая предоставляет сетевую инфраструктуру и обслуживание DNS для ключевых американских организаций. Основным методом компрометации устройств со стороны вредоносного ПО был подбор паролей из списка наиболее популярных комбинаций. Ботнет Mirai использовал для подключения к IoT-устройствам сетевой протокол telnet, и если на устройстве стоял стандартный пароль, то оно заражалось вредоносным ПО и становилось частью ботнета Mirai. По оценке исследователей, Mirai успел инфицировать около 500,000 устройств по всему миру, что привело к отказу таких популярных сервисов, как Xbox Live, Spotify и Reddit [2].

---

<sup>1</sup> Научно-исследовательский ресурс Academia.edu содержит более тысячи статей в поисковой выдаче портала по ключевым словам “Internet of Things”.

<sup>2</sup> Поиск по ключевым словам “Интернет вещей” на одном из самых популярных в России IT-ресурсе Хабрахабр показывает более тысячи публикаций.

Несмотря на то, что с момента информационного взрыва по теме ботнета Mirai прошло уже почти три года, безопасность IoT-устройств остается актуальной проблемой. 24 февраля 2019 года исследователи компании Avast в рамках конференции Mobile World Congress 2019 провели эксперимент: они разместили 500 IoT-устройств в 10 странах мира с предварительно открытыми портами 22 и 23, за которые отвечают сетевые протоколы telnet и ssh. Его идея заключалась в том, чтобы зафиксировать общее число попыток подключения со стороны потенциальных хакеров. За четыре дня эксперимента это значение достигло 23,2 миллиона [3].

Все рассматриваемые выше исследования касаются преимущественно западных стран, из-за чего практически невозможно оценить ситуацию в аспекте защищенности устройств IoT на территории Российской Федерации.

## **1.1 Цель работы**

Целью данной работы является исследование уровня защищенности устройств IoT на территории Российской Федерации, проводимое с помощью разработанного инструмента для автоматизированного поиска небезопасных настроек.

## **1.2 Постановка задачи**

Для достижения поставленной цели необходимо решить следующие задачи:

1. Получить актуальную выборку IP-адресов сетевых устройств интересующих классов.
2. Получить списки стандартных паролей различных компаний-производителей и моделей сетевых устройств.
3. Разработать инструмент для сопоставления каждому устройству подходящего списка паролей.
4. Разработать инструмент для автоматизированного поиска небезопасных настроек и протестировать его работу на реальных сетевых сервисах.
5. Провести анализ полученных результатов.

## **1.3 Актуальность задачи**

Задача безопасности устройств IoT становится все более актуальной в связи с увеличением количества “умных” устройств в повседневной жизни каждого человека. По оценке аналитического агентства IDC, опубликованной в докладе “IDC Russia Semiannual Internet of Things Spending Guide”, в период с 2018 по 2022 год включительно инвестиции в оборудование, программное обеспечение, услуги и связь, привлеченные для создания решений Интернета вещей, будут расти в среднем на 18% ежегодно [4]. Количество кибератак неминуемо будет увеличиваться пропорционально росту рынка IoT-устройств. Уже сейчас вероятность стать целью хакерской атаки в 12,5 раз превышает риск быть ограбленным [5].

Вместе с этим растет разнообразие вредоносного ПО. Помимо постоянно мутирующего Mirai [6], на свет появляются такие вирусы, как VPNFilter [7], основная особенность которого заключается в умении “пережить” перезагрузку устройства. Для борьбы с такими вредоносами компании-производители выпускают обновления, которые устанавливаются автоматически, однако, это всего лишь временная мера, если для доступа к устройству по-прежнему будет использоваться стандартный пароль. В таком случае высока вероятность повторного заражения.

Как было отмечено выше, многие сетевые устройства находятся в группе риска, так как их владельцы недостаточно осведомлены насчет вероятных последствий.

#### 1.4 Замечания по терминологии

В работе используются следующие термины, требующие определения:

- *Административная панель* - веб-интерфейс, доступный по сетевым портам 80 и 443, с помощью которого осуществляется управление и настройка сетевого устройства.
- *Небезопасная настройка сетевого устройства (сервиса)* - понятие, подразумевающее, что устройство имеет стандартный пароль доступа и открытые 80 или 443 сетевые порты.
- *Целевое устройство* - в рамках данной работы то же самое, что и сетевое устройство;
- *Брутфорсинг* - метод определения учетной записи путем перебора возможных комбинаций логина и пароля.
- *Веб-приложение* - некоторое приложение, исполняемое на удаленном узле, реализующее архитектуру “клиент-сервер”, при этом взаимодействие между клиентом и сервером происходит по сетевому протоколу HTTP (HTTPS).
- *Веб-клиент* - приложение, реализующее взаимодействие пользователя с веб-приложением посредством отправки HTTP-запросов и обработки HTTP-ответов.
- *Фингерпринтинг* - сбор информации о целевом устройстве с целью его идентификации.

**Примечание:** в работе рассматривается защищенность сетевых устройств в аспекте небезопасных настроек административных панелей для конфигурации этих устройств.

## **2 Анализ задачи**

Для оценки эффективности существующих методов и инструментов автоматизированного поиска небезопасных настроек в сетевых сервисах был выделен набор задач, возникающих при реализации прототипа для конечной цели данного исследования.

### **2.1 Получение списка IP-адресов целевых устройств**

Для проведения эксперимента на реализуемом инструменте необходимо получить списки IP-адресов реальных сетевых устройств. На этом этапе в рамках контекста исследования можно выделить основные требования, которым эти списки должны удовлетворять:

- Все IP-адреса должны иметь открытые сетевые порты 80/443 (обладать административными панелями);
- Все IP-адреса должны соответствовать устройствам, территориально принадлежащим российскому сегменту IoT.

### **2.2 Получение списков стандартных паролей компаний-производителей и моделей целевых устройств**

Конструируемый инструмент должен проверять защищенность сетевого устройства путем поочередной подстановки комбинаций логина и пароля. Для этого необходимо сформировать списки этих комбинаций, при этом каждой компании, производящей устройства IoT, должен соответствовать свой список стандартных логинов и паролей.

Компании-производители должны быть выбраны в соответствии с рейтингом их популярности и количеством продаж на российском рынке IoT устройств. Это необходимо как для соответствия контексту исследования, так и для оценки зависимости между популярностью устройства и его базовой защищенностью.

### **2.3 Фингерпринтинг устройства. Сопоставление устройству списка стандартных паролей**

Для повышения эффективности подбора паролей для каждого устройства разрабатываемый инструмент должен уметь выбирать подходящий ему список комбинаций. Здесь можно поставить задачу фингерпринтинга сетевых устройств на уровне веб-интерфейса и его интеграцию в конструируемое приложение.

### **2.4 Унификация метода подбора паролей для различных административных панелей**

Каждому рассматриваемому устройству соответствует своя административная панель, доступ к которой можно получить, пройдя аутентификацию. В силу того, что административные панели устройств, принадлежащих разным компаниям, кардинально друг от друга отличаются из-за отсутствия общепринятых критериев разработки веб-интерфейсов

устройств IoT, каждая форма авторизации по-своему обрабатывает пользовательский ввод (1, 2). Разрабатываемый инструмент должен уметь подстраиваться под правила обработки входных параметров для каждого рассматриваемого веб-интерфейса.

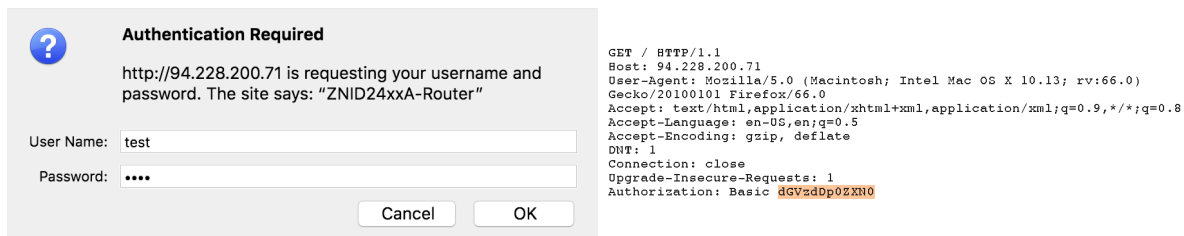


Рис. 1: Отправка на сервер логина и пароля test:test на примере веб-интерфейса маршрутизатора Zhone Technologies.



Рис. 2: Отправка на сервер логина и пароля test:test на примере веб-интерфейса маршрутизатора TP-Link.

## 2.5 Проблема блокировки полей веб-формы. Определение критерия успеха для каждой рассматриваемой комбинации логина и пароля

Для получения корректных результатов эксперимента программа должна уметь выделять подходящие комбинации логина и пароля. Для этого необходимо отследить поведение веб-приложения при подстановке заведомо ложных и верных комбинаций, а затем выделить критерии успешного подбора и реализовать соответствующий модуль в конструируемом приложении.

Кроме того, к этой подзадаче можно отнести проблему блокировки формы ввода в некоторых административных панелях после нескольких неудачных попыток подбора (3). Необходимо придумать и реализовать алгоритм обхода блокировки в тех веб-интерфейсах, где время ожидания ее снятия значительно влияет на увеличение интервалов простоя программы.



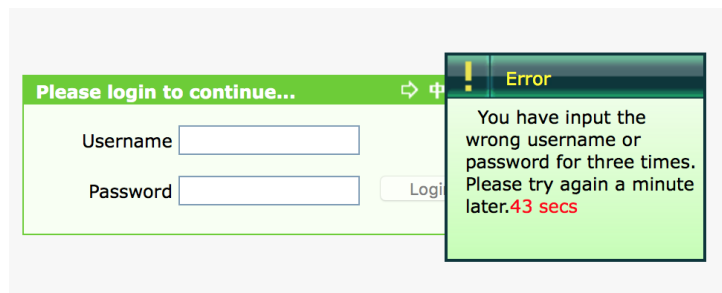


Рис. 3: Блокировка ввода данных после трех неудачных попыток подбора в веб-интерфейсе маршрутизатора ZTE.

## 3 Обзор существующих решений

Для каждой подзадачи, описанной в п. 2, приводятся разные варианты решений, которые используются как в существующих инструментах (см. подробнее п. 3), так и в реализованном в рамках данной работы инструментальном прототипе (см. подробнее п. 4).

### 3.1 Методы получения списков исследуемых IP-адресов

Как уже было отмечено, для проведения эксперимента необходимо получить первоначальные входные данные, то есть актуальные списки IP-адресов устройств российского сегмента IoT.

Ниже приводится обзор существующих инструментальных средств, которые описывают два ключевых подхода к решению подзадачи данной работы. Цель обзора: выделить наиболее подходящий способ построения требуемых списков с учетом контекста исследования.

#### 3.1.1 Сканирование диапазонов IP-адресов

- **Nmap** [8] - утилита с открытым исходным кодом, предназначенная для исследования сети и проверки ее безопасности. Помимо сканирования сетевых портов Nmap может идентифицировать сервис, слушающий открытый порт, и его версию. Чтобы использовать утилиту в рамках подзадачи данного исследования, необходимо провести первоначальную настройку параметров сканирования в соответствии с описанными в п. 2 критериями. Для выделения диапазонов российских IP-адресов можно воспользоваться онлайн-базой IpGeoBase<sup>1</sup>. Готовая к использованию команда Nmap будет выглядеть так:

```
$ nmap -p 80,443 --open -T4 109.94.192.0/24
```

-p 80,443: параметр настройки фильтра сканирования портов;

--open: параметр выдачи результата только по открытым портам;

-T4: параметр для более быстрого сканирования;

109.94.192.0/24: один из диапазонов IP-адресов компании Tele2 Россия, полученный через онлайн-базу IpGeoBase.

Данный подход имеет некоторые ограничения. Во-первых, Nmap работает по принципу синхронного сканера, то есть утилита отслеживает запросы на соединения и ожидает ответа. Если на TCP-запрос на соединение не приходит ответа, то утилита ждет ответ, пока не истечет все выделенное на запрос время. Так как заранее нельзя предугадать, какие именно IP-адреса из всего диапазона неактивны, общее время сканирования может быть очень велико. Во-вторых, суммарное количество IP-адресов, принадлежащих Российской Федерации, огромно. По оценке IpGeoBase количество IP-адресов в наибольших по численности населения городах России составляет более

---

<sup>1</sup><http://ipgeobase.ru>

30 миллиардов [9]. Кроме того, далеко не все рассматриваемые IP-адреса соответствуют устройствам IoT.

- **Zmap** [10] - асинхронный сканер сети с открытым исходным кодом. В отличие от Nmap утилита при отправке SYN-пакета не ждет, пока вернется ответ, а продолжает сканирование, ожидая ответы одновременно от всех хостов. За счет этого скорость сканирования по сравнению с Nmap значительно увеличивается. Готовая к использованию команда Zmap будет выглядеть так:

```
$ sudo zmap -p 80 130.193.32.0/24 -o output.csv
```

-p 80: параметр настройки фильтра сканирования портов;

130.193.32.0/24: один из диапазонов IP-адресов компании Яндекс, полученный через онлайн-базу IpGeoBase.

Несмотря на высокую скорость сканирования, данный подход имеет некоторые недостатки. Во-первых, утилите нельзя указать в качестве параметра сразу несколько портов, то есть для сканирования сетевых портов 80 и 443 ее придется запускать два раза. Во-вторых, по умолчанию утилита выводит лишь список IP-адресов с тем открытым сетевым портом, который был заранее указан при запуске программы. Для получения более подробной информации о сервисе необходимо использовать Zmap в связке с утилитой ZGrab [11], которая способна собирать ответы прикладного ПО сканируемых диапазонов.

- **Masscan** [12] - инструмент с открытым исходным кодом для сканирования больших сетей. Его высокая производительность достигается за счет асинхронной передачи пакетов, схожей с утилитой Zmap. По словам разработчика, скорость работы с теоретической производительностью до 10 миллионов пакетов в секунду позволяет просканировать все IPv4 публичное пространство менее чем за шесть минут. Кроме того, в отличие от Zmap, у инструмента есть функционал для получения базовой информации о сервисе в виде баннера, однако его возможности достаточно ограничены по сравнению с Zgrab и Nmap. Готовая к использованию команда Masscan будет выглядеть так:

```
$ sudo masscan 178.213.78.0/24 -p80,443 --banners
```

-p80,443: параметр настройки фильтра сканирования портов;

--banners: параметр для вывода базовой информации о сервисе;

178.213.78.0/24: один из диапазонов IP-адресов компании Lamoda, полученный через онлайн-базу IpGeoBase.

У рассматриваемого сканера есть ряд недостатков. Во-первых из-за того, что инструмент использует собственный сетевой стек TCP/IP, рекомендуется запускать сканирование на выделенном IP, чтобы избежать конфликта со стеком операционной системы. Во-вторых, из-за ограниченных возможностей вывода баннеров не всегда удается собрать желаемую информацию об устройстве (4).

```

Discovered open port 80/tcp on 192.168.1.1
Discovered open port 80/tcp on 192.168.1.40
Discovered open port 443/tcp on 192.168.1.40
Banner on port 443/tcp on 192.168.1.40: [ssl] TLS/1.0 cipher:0x35, HPE22E98
Banner on port 443/tcp on 192.168.1.40: [X509] MIICczCCAAdygAwIBAgIE007IWDANBgkqh
kiG9w0BAQQFADBNMREwDwYDVQQDEwhIUEUyMku50DESMBAGA1UEBxMjVmFuY291dmVybWERMwEQYDVQIQIE
wpXYXNoaW5ndG9uMQswCQYDVQQGEwJVUzELMAkGA1UEChMCsFAxDbANBgNVBAsTBkhQLU1QRzAeFw0xN
TEyMjEwODQ5MThaFw0zNTEyMTYwODQ5MThaMGcxETAPBgNVBAMTCeHQRRTk4MRIwEAYDVQQHEWlwY
W5jb3V2ZXIxZzARBgNVBAGTCldhc2hpbmd0b24xCzAJBgNVBAYTAlVTMQswCQYDVQQKEwJlUEDEPMA0GA
1UECmMGSAFATSVBHMIGdMA0GCSqGSIb3DQEBAQUAA4GLADCBhwKBgQDYA9m1RJa3N+jRaY30k3MnLkz9s
Jl6L7Jm2e0f7/ccZmmFNHjTUPu7S68S1FL6B5fThML2iWPSGQFPi0Zt0Atph9kLCRCfVmeQLjSwQmWJ
+ReZ9X42GnwsInrMZulB7za9RzkULJPTQuDsBIPBvWZ53JD3rtEXR9r0NYUy2CAswIBA6MuMwCwYDV
R0PBAQDAgTwMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjANBgkqhkiG9w0BAQQFAA0BgQAWs
URYT61js9VefKp5PfmHgxmIUIlhTfbXusCPyl6pEfZv/PeIy8JETrhLSdv31hInMa1+kcSmMe3clguPM
Peapw43hhIRoqUpiOKqFrYyQzNCKDvwm3BB0ENHkXxUBgtpo8sPUSqZiWYyPawbZtUfLRBf67XcP0RfC
93ibtfbng==
Banner on port 80/tcp on 192.168.1.40: [http] HTTP/1.1 505 HTTP Version Not Supp
orted\x0d\x0aServer: HP HTTP Server; HP Deskjet 3520 series - CZ275C; Serial Num
ber: CN4342G1DP05SZ; Stuttgart ia_pp usr hf Built:Mon Dec 21, 2015 09:49:18AM {S
AP1FN1552AR, ASIC id 0x00340104}\x0d\x0a\x0d

```

Рис. 4: Несмотря на обнаружение в домашней сети двух устройств с IP-адресами 192.168.1.1 и 192.168.1.40 Masscan вывел информацию только об одном из них - принтере HP Deskjet 3520.

### 3.1.2 Поисковые системы

В силу вышеописанных недостатков самостоятельное массовое сканирование требует взвешенного подхода, кроме того, для каждого из инструментов требуется проводить дополнительную аналитику результатов сканирования. С технической точки зрения можно облегчить задачу, воспользовавшись существующими коммерческими инструментами:

- **Shodan**<sup>1</sup> [13] - поисковая система, сканирующая весь Интернет в реальном времени, которая на основании полученных ответных баннеров делает выводы об устройствах и сервисах. Shodan предоставляет возможность фильтровать результаты поиска по различным критериям (порты, страны, операционные системы, география относительно введенных координат и т.д.). Например, чтобы получить результаты поиска по маршрутизаторам в рамках контекста исследования, необходимо сформировать такой запрос:

```
router country:ru port:80
```

Также есть возможность просмотреть ответные баннеры только лишь тех устройств, сканирование которых произошло после определенной даты. Это очень важно для данного исследования, так как позволяет сократить множество уже неактивных сервисов для входной выборки.

- **ZoomEye**<sup>2</sup> [14] - поисковик устройств IoT, созданный китайской компанией Knowsec Inc. Поисковик собирает информацию по широкому диапазону портов. Результаты поиска можно разделить по большому числу критериев (устройства, порты, операционные системы и т.д.) с демонстрацией содержимого прикладных баннеров. Для

<sup>1</sup><https://www.shodan.io>

<sup>2</sup><https://www.zoomeye.org>

получения результатов поиска в контексте исследования необходимо сформировать такой запрос:

```
device:"router"+country:"ru"+port:"80"
```

По сравнению с поисковой системой Shodan данный поисковик обладает некоторыми недостатками. Скорость обновления базы результатов сканирования ZoomEye ниже, чем у Shodan [15], из-за чего есть вероятность на этапе эксперимента встретить уже неактивные сервисы.

### 3.1.3 Выводы

В силу вышеперечисленных недостатков инструментов прямого сканирования и поисковика ZoomEye, было принято решение использовать поисковую систему Shodan для построения входных списков IP-адресов в рамках контекста данного исследования.

## 3.2 Методы получения списков стандартных паролей для исследуемых устройств

Как уже было упомянуто выше, для каждого устройства реализуемый инструмент должен проверять, присутствует ли в его административной панели стандартный пароль доступа. Для этого необходимо построить списки стандартных паролей компаний - производителей устройств, составляющих актуальную выборку для проведения эксперимента.

Существующие решения:

- Формирование списков паролей путем заимствования готовых списков, опубликованных в свободном доступе, либо используемых популярными приложениями для подбора паролей. Например, инструмент **Ncrack** [16] содержит несколько списков стандартных авторизационных данных, среди которых также присутствуют списки простых и популярных слов (qwerty, iloveyou и т.д.), часто выступающих в качестве паролей. Еще одним примером может служить инструмент подбора паролей **BruteX** [17], который использует словари стандартных паролей для популярных сервисов (ftp, postgres, ssh, telnet и т.д.), опубликованные на ресурсе <https://download.openwall.net/>. Среди находящихся в свободном доступе стандартных авторизационных данных также может служить база, размещенная на ресурсе <http://phenoelit.org/dpl/dpl.html>.

Данный подход имеет некоторые ограничения. Исследование сфокусировано на проверке защищенности устройств IoT путем перебора стандартных паролей. Это условие не позволяет использовать готовые списки инструментов подбора паролей, даже если там встречаются комбинации стандартных логинов и паролей, так как по сравнению с простыми словами их слишком мало. Кроме того, русскоязычные пользователи предпочитают использовать в качестве паролей русские слова в латинской раскладке. Это показывает крайнюю неэффективность использования готовых списков для

подбора паролей, так как многие из них составлены англоязычными исследователями. Во-вторых, при построении списков авторизационных комбинаций необходимо учитывать специфику российского рынка устройств IoT. Это условие не позволяет использовать публичные списки паролей сетевых устройств, так как большинство опубликованных сборных списков стандартных комбинаций принадлежат непопулярным в России иностранным компаниям.

- Самостоятельное построение списков с помощью документации, опубликованной на сайтах производителей. Для повышения эффективности подбора паролей было принято решение предварительно исследовать российский рынок устройств IoT и выделить наиболее популярные среди покупателей компании и модели. Для этого был рассмотрен сервис Яндекс.Маркет<sup>1</sup>, в котором по каждому типу рассматриваемых IoT устройств на основании поискового фильтра по популярности был выделен ряд компаний-производителей и их моделей. Таким образом были выделены наиболее приоритетные компании в рамках подзадачи построения списков стандартных паролей.

В рамках данного исследования были построены списки стандартных паролей с учетом специфики российского рынка устройств IoT.

### 3.3 Фингерпринтинг устройств

Для автоматизации поиска небезопасных настроек разрабатываемый инструмент должен уметь сопоставлять каждому устройству подходящий список стандартных паролей. Другими словами, на этом этапе стоит задача фингерпринтинга сетевых устройств, или сбор различной информации об этих устройствах с целью дальнейшей идентификации производителя и модели. Зачастую фингерпринтинг устройства подразумевает под собой выявление информации о программном обеспечении целевого устройства, его операционной системе и аппаратных компонентах для создания сигнатуры, которую называют “отпечатком”.

Фингерпринтинг сетевых устройств можно разделить на две основных категории:

- **Активный** - метод, заключающийся в том, чтобы послать запрос устройству и проанализировать полученный ответ. В одном случае устройству присылают заранее подготовленные пакеты данных, чтобы определить поведение устройства после получения ответа. В другом случае устройству присылают запросы на различные порты, чтобы определить расположенные на них сервисы и сделать выводы о системе в целом. Один из подходов активного фингерпринтинга описан в работе [18], где проводится анализ ответов исследуемых устройств, получаемых на сетевом, транспортном и прикладном уровнях, с целью извлечения характеристик для построения таблицы признаков для каждого устройства.

---

<sup>1</sup><https://market.yandex.ru>

- **Пассивный** - такой подход подразумевает возможность перехватывать трафик, идущий от анализируемого устройства, с целью его дальнейшего анализа для извлечения характеристик устройства. К примеру, в исследованиях [19] [20] [21] [22] рассматривается способ идентификации, основанный на использовании тактового искажения внутренних часов исследуемого устройства. Пассивный фидгерпринтинг нельзя проводить для удаленно расположенных устройств.

Выбор способа фидгерпринтинга устройств зависит как и от того, насколько подробную об исследуемом устройстве информацию нужно получить, так и от расположения самого устройства относительно машины, с которой проводится фидгерпринтинг. В Таблице 1 приведены основные сравнительные характеристики обоих методов.

Активный фидгерпринтинг	Пассивный фидгерпринтинг
Работает на удаленных устройствах.	Необходимо находиться в одной сети с устройством.
Взаимодействует с устройством напрямую.	Перехватывает и анализирует трафик.
Требуется разрешение на соединение от устройства.	Не требуется устанавливать прямое соединение с устройством.
Может быть обнаружен исследуемым устройством.	Не обнаруживается исследуемым устройством.

Таблица 1

Одним из первых инструментов, в которых были реализованы методы фидгерпринтинга, является утилита **Nmap** [8]. Утилита анализирует получаемые от устройства IP-пакеты для выявления информации о версии операционной системы хоста и типах запущенных на нем приложений и сервисов (5).

```
Nmap scan report for 162.168.10.3
Host is up (0.45s latency).

PORT      STATE SERVICE VERSION
80/tcp    closed http
443/tcp   closed https
MAC Address: A0:3B:E3:E4:26:47 (Apple)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: phone
Running: Apple iOS 6.X, Apple iPhone OS 1.X
OS CPE: cpe:/o:apple:iphone_os:6.1.4 cpe:/o:apple:iphone_os:1
OS details: Apple iOS 6.1.4 (Darwin 13.0.0), Apple iPhone mobile phone (iPhone OS 2.1)
Network Distance: 1 hop
```

Рис. 5: Фидгерпринтинг, проведенный утилитой *Nmap* для подключенного к домашней сети телефона.

Несмотря на все достоинства, *Nmap* нельзя использовать в данном исследовании в качестве инструмента для фидгерпринтинга, так как она показывает неверные результаты сканирования устройств малоизвестных компаний-производителей. Например, результат сканирования устройства с IP-адресом 94.228.200.124 предполагает с вероятностью 94%, что это маршрутизатор *Asus RT-AC66U* (6), однако это не так. Модель маршрутизатора

ZNID24xx-Router соответствует компании Zhone Technologies<sup>1</sup>.

```
mayaglushchenko@MacBook-Maya : ~  
$ sudo nmap -p 80 -A -T4 94.228.200.124  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-07 00:04 MSK  
Nmap scan report for union-tel.200.124.ru (94.228.200.124)  
Host is up (0.0067s latency).  
  
PORT      STATE SERVICE      VERSION  
80/tcp    open  tcpwrapped  
| http-auth:  
| HTTP/1.1 401 Unauthorized\x0D  
|_ Basic realm=ZNID24xx-Router  
Warning: OSScan results may be unreliable because we could not find at least 1 o  
pen and 1 closed port  
Aggressive OS guesses: Linux 2.6.18 (Debian 4.0, x86) (96%), OpenWrt White Russi  
an 0.9 (Linux 2.4.30) (95%), Asus RT-AC66U router (Linux 2.6) (94%), Asus RT-N10  
router or AXIS 211A Network Camera (Linux 2.6) (94%), Linux 2.6.18 (94%), Asus  
RT-N16 WAP (Linux 2.6) (94%), Asus RT-N66U WAP (Linux 2.6) (94%), Tomato 1.28 (L  
inux 2.6.22) (94%), AXIS 211A Network Camera (Linux 2.6.20) (94%), OpenWrt 0.9 -  
7.09 (Linux 2.4.30 - 2.4.34) (94%)  
No exact OS matches for host (test conditions non-ideal).  
Network Distance: 6 hops
```

Рис. 6: Результаты сканирования IP-адреса 94.228.200.124 утилитой Nmap.

В популярных инструментах для подбора паролей, таких как **Hydra** [23], **Medusa** [24], **Patator** [25] и **Ncrack** [16], автоматизированный фингерпринтинг не представлен. Для работы с приложениями, ориентированными преимущественно на таргетированный брутфорсинг, первым этапом собирается полезная информация о жертве, на основании которой формируется стратегия подбора паролей: строятся списки наиболее вероятных паролей, исследуется метод передачи аутентификационных данных и т.д. В таком случае фингерпринтинг не нужен.

Так как в рамках данного исследования реализуемый инструмент взаимодействует с устройствами на прикладном уровне, то разрабатываемый модуль фингерпринтинга должен посылать HTTP(S)-запросы и анализировать ответы, приходящие от устройств. На рисунках ниже показано, что характеристические признаки могут присутствовать как в заголовках, так и в самом теле ответа.

```
HTTP/1.1 401 Unauthorized  
Server: micro_httpd  
Cache-Control: no-cache  
Date: Mon, 05 Jan 1970 05:19:18 GMT  
WWW-Authenticate: XDigest realm="Eltex Router", domain="eltex.ru",  
nonce="NjdkOTI3NTE6Yzk1YWEmMTg6NzZhMDMyMGQ=", qop="auth", algorithm=MD5  
Content-Type: text/html; charset=UTF-8  
Connection: close
```

Рис. 7: Анализируя заголовок ответа, можно сделать вывод, что мы обратились к маршрутизатору компании Eltex.

<sup>1</sup><https://fccid.io/ANATEL/00494-13-07105/Manual/AE18F77E-5397-4277-B497-5B5290933794/PDF>



```

</svg>
<svg id="logo_keenetic" width="147" height="32.6" viewBox="-.
<style>.cest0{fill:#fff}</style>
<path class="cest0" d="M33.5 2.8V0H17.9v12.6h15.6V9.8H20.7
2.8V0H71.3v12.6h15.6V9.8H74.2V7.7h12.7V4.9H74.2V2.8zM89.2 0v2.
1.4.4.5.7 1.2.7 2.1v1.5h2.8v-1.4c0-1.5-.5-2.9-1.3-3.9-1-1.2-2.
0-3.5-.5-4.3-1.4-.4-.5-.7-1.2-.7-2.1s.2-1.6.7-2.1c.8-.9 2.2-1.
3.9s.5 2.9 1.3 3.9c1.3 1.6 3.5 2.3 6.4 2.3 3 0 5.1-.8 6.4-2.3:
</svg>
<svg height="38" width="200" id="logo_keenetic_18px" viewBo:

```

Рис. 8: В данном случае в теле ответа были обнаружены признаки того, что устройство принадлежит компании Keenetic.

Таким образом, для решения задачи сопоставления списков стандартных паролей было принято решение сконструировать модуль фингерпринтинга, реализующий активный подход, так как заранее неизвестно расположение рассматриваемых сетевых устройств относительно компьютера, на котором проводится исследование. Модуль должен поочередно анализировать заголовки и тело ответа каждого устройства для извлечения характеристических признаков. Кроме того, для расширения области поиска признаков модуль фингерпринтинга должен обрабатывать специально спровоцированные ответы с кодом 404, так как там потенциально может присутствовать ценная информация об устройстве. Ниже представлена схема работы модуля фингерпринтинга устройств для разрабатываемого инструмента.

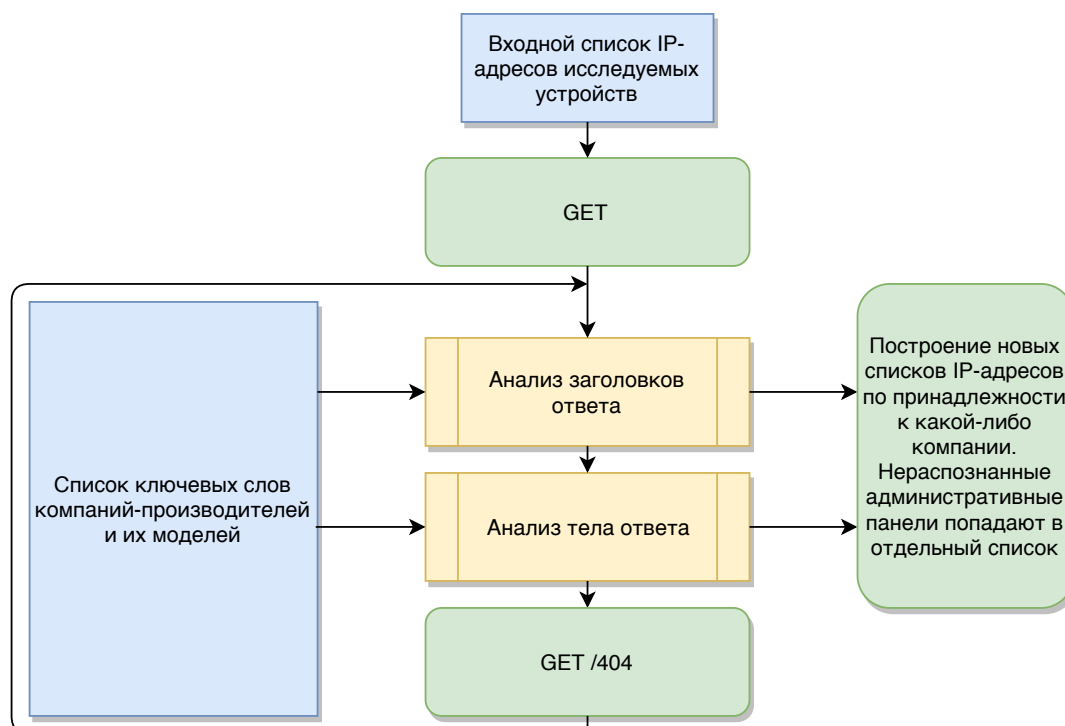


Рис. 9: Схема работы модуля фингерпринтинга.

### 3.4 Задача унификации методов подбора паролей для устройств

Все рассматриваемые устройства обладают административными панелями, для получения доступа к которой необходимо пройти аутентификацию, предоставив серверу логин и пароль. Разрабатываемый инструмент должен уметь подстраиваться под требуемый формат ввода аутентификационных данных.

Все рассматриваемые в рамках данного исследования административные панели устройств осуществляют аутентификацию по паролю с помощью следующих методов:

- **HTTP Аутентификация** - протокол, описанный в стандартах HTTP 1.0<sup>1</sup>/1.1<sup>2</sup>. Применительно к веб-приложениям он работает следующим образом:
  1. При обращении неавторизованного пользователя к защищенному ресурсу сервер отправляет ему статус “401 Unauthorized” и добавляет к ответу заголовок “WWW-Authenticate” с указанием схемы и параметров аутентификации.
  2. Браузер, получив такой ответ, показывает пользователю диалоговое окно, в котором предлагается ввести логин и пароль, после чего полученные данные отправляются на сервер, где производится решение о предоставлении доступа пользователю.
  3. При всех последующих обращениях к веб-приложению браузер будет добавлять заголовок “Authorization”, в котором передаются данные пользователя для аутентификации сервером.

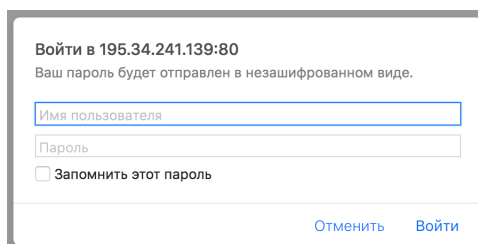


Рис. 10: Диалоговое окно, появляющееся при обращении к устройству.

В веб-интерфейсах рассматриваемых устройств, реализующих протокол HTTP Аутентификации, наиболее распространены следующие схемы аутентификации, различающиеся по уровню безопасности:

- **Basic:** схема, при которой пользовательские данные передаются в заголовке Authorization в незашифрованном виде, закодированном в base-64.
- **Digest:** схема, при которой сервер присылает пользователю уникальное значение nonce, в свою очередь браузер передает серверу MD5 хэш пароля, вычисленный

<sup>1</sup><https://tools.ietf.org/html/rfc1945>

<sup>2</sup><https://tools.ietf.org/html/rfc7235>

с помощью этого значения. Несмотря на то, что Digest схема - более безопасная альтернатива Basic схемы при незащищенных соединениях, она подвержена атаке Man in the middle с заменой схемы на Basic.

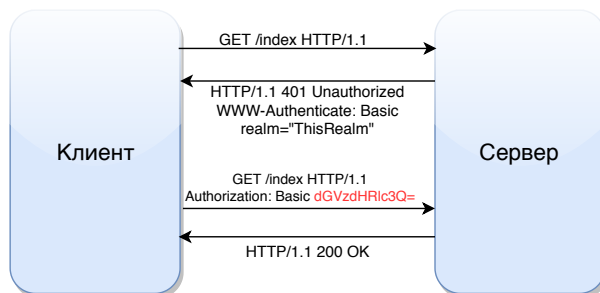


Рис. 11: Пример HTTP аутентификации с использованием Basic схемы.

- **Form Authentication** - для этого подхода нет определенного стандарта, поэтому все его реализации специфичны для конкретных веб-приложений. Для реализации этого метода в веб-приложение встраивается HTML-форма, в которую пользователь должен ввести аутентификационные данные, после чего они отправляются на сервер через HTTP-запрос. В случае успеха приложение генерирует токен сессии, который обычно помещается в куки браузера. При всех последующих запросах к веб-приложению токен сессии автоматически передается на сервер и позволяет приложению получить информацию о текущем пользователе для авторизации.

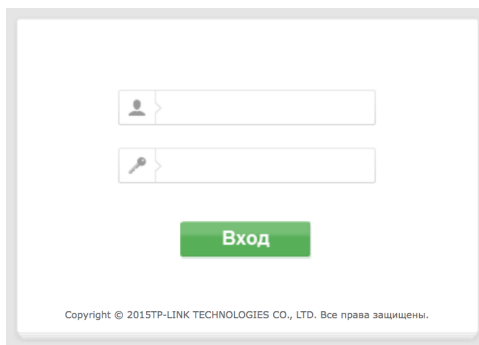


Рис. 12: Пример формы аутентификации для административной панели маршрутизатора TP-Link.

Кроме того, в рамках данного исследования среди рассматриваемых административных панелей присутствуют также “гибридные” веб-интерфейсы. Например, веб-интерфейс маршрутизаторов компании Eltex при обращении к нему возвращает статус “401 Unauthorized”, хотя на самом деле в нем реализован метод аутентификации через веб-форму:

### Авторизация

Имя пользователя

Пароль

```

HTTP/1.1 401 Unauthorized
Server: micro_httpd
Cache-Control: no-cache
Date: Mon, 06 May 2019 15:08:24 GMT
WWW-Authenticate: XDigest realm="Eltex Router", domain="eltex.ru",
nonce="OTUwOGI2Mjg6MThjZjUzM2Q6ODYwZjkxYTc=", qop="auth", algorithm=MD5
Content-Type: text/html; charset=UTF-8
Connection: close

```

Рис. 13: Форма авторизации и заголовки ответа, возвращаемые веб-интерфейсом маршрутизатора *Eltex*.

Как уже было упомянуто выше, конструируемый инструмент должен находить правильный подход для каждого веб-интерфейса при отправке данных для аутентификации. Здесь необходимо также учитывать то, что на этапе эксперимента могут попадаться нестандартные веб-интерфейсы, как в вышеописанном примере.

Существующие решения:

- Предварительный сбор информации о веб-интерфейсе с целью определения подходящего метода отправки запроса. В этом случае для веб-интерфейсов, реализующий протокол HTTP Аутентификации, уместно посылать данные, исходя из стандартов этого протокола. Для веб-интерфейсов со встроенной веб-формой аутентификации следует предварительно собрать информацию о самой форме, чтобы определить подходящий формат запроса. Как правило, в этом случае анализируют значения атрибутов веб-формы. Например, форма аутентификации маршрутизатора ZTE отправляет данные методом POST на страницу с формой для их обработки:

```

<form name="fLogin" id="fLogin" method="post" onsubmit="return false;" action="">
<input type="hidden" name="_lang" id="_lang" value="" disabled>
<input type="hidden" name="frashnum" id="frashnum" value="">
<input type="hidden" name="action" id="action" value="login">
<input type="hidden" name="Frm_Logintoken" id="Frm_Logintoken" value="">

```

Рис. 14: Форма отправки данных маршрутизатора *ZTE*.

Популярные инструменты для подбора паролей Patator [25], Hydra [23] требуют именно такой подход, но он не является автоматизированным, то есть вся ответственность за правильный выбор параметров подбора паролей лежит на операторе. Для автоматизации этого процесса можно проводить анализ тела ответа, выделяя подходящие теги и атрибуты. Однако, такой метод не всегда будет удобен, так как в исследуемых веб-интерфейсах может встречаться нестандартная реализация веб-формы:

```

<div class="loginBox">
  <div class="noteDiv">
    <span id="note"></span>
    <span id="tip"></span>
  </div>
  <div class="panelThre" align="center">
    <div align="center" class="picDiv" align="center">
      <ul>
        <li id="unLi" class="unLi"><input class="text" id="userName" type="input" maxlength="14"/></li>
        <li class="blank"></li>
        <li id="pwLi" class="pwLi"><input class="text" id="pcPassword" type="password" maxlength="14"/></li>
      </ul>
    </div>
    <label id="loginBtn" class="loginBtn" onclick="PCSubWin()"/></label>
  </div>
</div>

```

Рис. 15: Форма отправки данных маршрутизатора *TP-Link*.

- Использование headless-браузера для отправки данных формы аутентификации. Headless-браузеры преимущественно используются для тестирования веб-приложений. Они не тратят ресурсы компьютера на отрисовку содержимого веб-страницы, за счет чего потребляют мало памяти и достаточно быстро работают. Для управления headless-браузером используют программный интерфейс. Фактически здесь можно говорить о программной имитации поведения пользователя на странице формы аутентификации. Основное преимущество такого подхода заключается в том, что он избавляет разработчика от необходимости учитывать особенности каждой исследуемой веб-формы, что технически упрощает исходную задачу. Среди существующих headless-браузеров наиболее популярными являются **PhantomJS** [26] и **Headless Chrome** [27]. Среди них предпочтение обычно отдают второму, так как поддержка PhantomJS на сегодняшний день приостановлена.

В рамках данного исследования для подбора пароля в веб-интерфейсах, реализующих протокол NTTP Аутентификации, было принято решение использовать стандартный метод отправки данных, предписанный протоколом. Для отправки данных через веб-форму был выбран инструмент Headless Chrome.

### 3.5 Обход блокировки полей ввода веб-формы. Методы определения критерия успеха в процессе подбора комбинаций

В качестве метода противодействия брутфорсингу в некоторых административных панелях присутствует встроенный механизм блокировки полей ввода данных после определенного количества неудачных попыток. Обычно в таком случае веб-интерфейс показывает пользователю сообщение о блокировке и таймер:

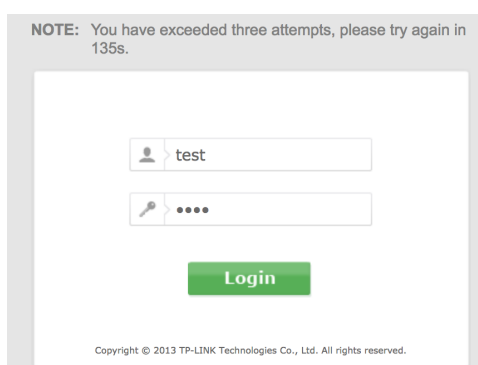


Рис. 16: После трех неудачных попыток форма ввода маршрутизатора TP-Link блокируется.

Для решения этой проблемы существуют следующие методы:

- После очередного ответа сервера проверять тело ответа на наличие сообщения о блокировке. Если сообщение появилось, то необходимо приостановить подбор паролей на

указанное время. Данный подход имеет некоторые ограничения. Во-первых, в рассматриваемых веб-интерфейсах устройств IoT время ожидания снятия блокировки варьируется от десяти секунд до двух часов, из-за чего значительно возрастает время простоя программы. Во-вторых, некоторые административные панели содержат сообщение о блокировке в исходном коде страницы по умолчанию, что гарантирует ложное обнаружение блокировок.

- Метод “горизонтального” брутфорсинга. При проверке большого числа IP-адресов административных панелей какой-либо компании удобно использовать следующий алгоритм: необходимо взять первую комбинацию логина и пароля и поочередно подставить ее в каждый веб-интерфейс. На этом этапе можно убрать из списка для проверки административные панели, которым эта комбинация подошла. Затем взять вторую комбинацию логина и пароля и повторить те же действия. Пока каждая комбинация обходит весь список IP-адресов, время блокировки веб-формы истекает естественным образом. Такой подход реализован в популярном инструменте для подбора паролей Patator [25].

Для решения проблемы блокировки формы ввода данных было принято решение реализовать в конструируемом приложении метод “горизонтального” брутфорсинга. Такой подход затрагивает только те веб-интерфейсы, в которых присутствует механизм блокировки. Во всех остальных административных панелях перебор паролей проводится прямым способом.

Кроме того, ключевой задачей в процессе разработки инструмента является определение критериев успеха подбора комбинации логина и пароля для всех исследуемых веб-интерфейсов. Здесь необходимо учитывать специфику подходов реализации методов передачи данных на сервер. Например, при реализованном протоколе HTTP Аутентификации достаточным условием верной комбинации будет статус ответа “200 OK”. Для веб-интерфейсов, реализующих передачу параметров через веб-форму, существует несколько способов выявления верной комбинации:

- Проверка тела ответа на наличие сообщения, указывающего на неправильную комбинацию. Если сообщение не найдено, пометить рассматриваемую комбинацию как подходящую. Несмотря на то, что такой подход используется в инструментах Hydra [23], Medusa [24], Patator [25], и довольно удобен для таргетированного брутфорсинга, он непригоден в рамках массового подбора паролей. Во-первых, чтобы он работал корректно, необходимо собрать словарь всех возможных сообщений об ошибочном вводе логина и пароля. Если принять во внимание тот факт, что многие веб-интерфейсы поддерживают несколько языков, задача становится сложной. Во-вторых, некоторые веб-интерфейсы при некорректном вводе попросту игнорируют его и никакие сообщения не выводятся.
- Определение успешного подбора путем проверки тела ответа на наличие поля ввода с типом password. Если такое поле встретилось один раз, то комбинация неверна. Если

такое поле встретилось ноль либо более одного раза, то либо комбинация оказалась успешной, либо административная панель изначально была без пароля.

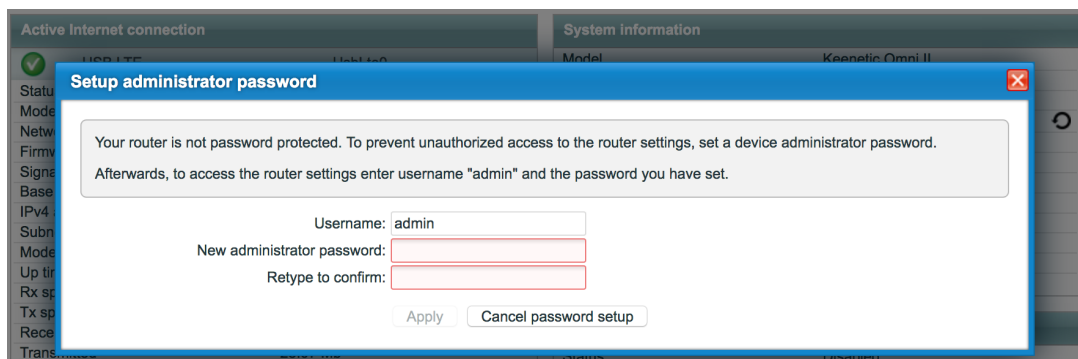


Рис. 17: Открытая административная панель маршрутизатора Zyxel содержит два поля ввода с типом *password*.

Конструируемый в рамках данного исследования инструмент для определения успешной подстановки входных данных для веб-форм реализует подход, основанный на анализе тела ответа для поиска полей ввода с типом *password*. Для веб-интерфейсов, реализующих протокол HTTP Аутентификации, критерием успешного ввода считается код ответа 200.

## 4 Реализация прототипного инструмента

В рамках данного исследования был реализован прототипный инструмент, реализующий методы решения задачи автоматизированного поиска небезопасных настроек, изложенных в п. 2.

### 4.1 Обоснование выбора инструментария

Приложение реализовано на языке Python с использованием библиотеки Requests [28] для подбора паролей в том случае, если в административной панели реализован протокол HTTP Аутентификации. Если же передача аутентификационных данных происходит с помощью веб-формы, то для таких случаев было принято решение использовать библиотеку Selenium WebDriver [29], которая предоставляет возможность использовать веб-клиент Headless Chrome для имитации поведения пользователя в сети. Этот легковесный веб-клиент позволяет взаимодействовать с веб-приложением через программный код, не затрачивая ресурсы на отображение страницы пользователю. Такая связка позволяет рассчитывать на лучшую совместимость и поддержку новых технологий со стороны веб-клиента (по сравнению с веб-клиентом PhantomJS, который в данный момент уже не поддерживается). Использование Headless Chrome через Selenium WebDriver и библиотеки Requests в связке с Python позволяет решить задачи программной реализации инструмента, обозначенные в п. 2 данной работы.



## 4.2 Архитектура инструмента. Общая схема работы

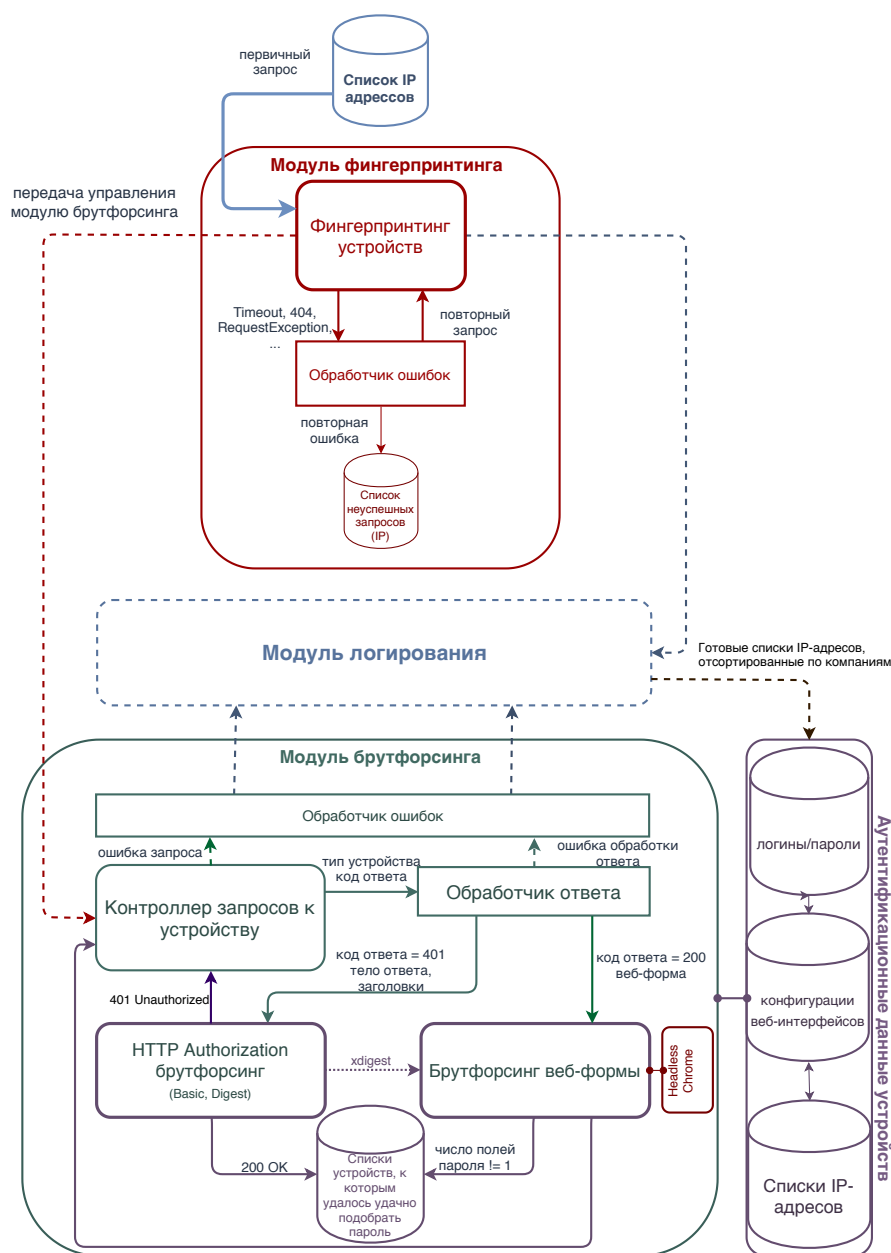


Рис. 18: Схема архитектуры инструмента.

Приложение имеет модульную архитектуру и состоит из следующих компонентов:

- **Модуль фотопечати** - модуль, определяющий принадлежность рассматриваемых устройств к какой-либо компании. Он предназначен для повышения эффективности подбора паролей на этапе эксперимента. На вход этому модулю подаются списки IP-адресов устройств, разбитых по классам, на которые посылаются запросы для определения характеристических признаков. Полученная информация анализируется и отправляется на вход модулю логирования для построения списков IP-адресов, сформированных по принадлежности к какой-либо компании.

- *Модуль логирования* - модуль, предназначенный для сбора подробной информации о каждом осуществленном запросе на указанные IP-адреса сетевых устройств. В задачи модуля входит преобразование информации о запросах в текстовый формат. Этот модуль формирует списки устройств по компаниям, списки устройств, к которым удалось подобрать правильную комбинацию логина и пароля, а также списки IP-адресов, запросы на которые завершились ошибкой или таймаутом.
- *Модуль брутфорсинга* - модуль, реализующий механизм подбора паролей на указанные IP-адреса. В задачи этого модуля входит осуществление всех запросов, посылаемых на рассматриваемые устройства, а также определение формата передачи аутентификационных данных. После передачи данных модуль определяет успешные комбинации логина и пароля и заносит соответствующие IP-адреса в новые списки. Кроме того, модуль решает задачи, возникающие в процессе подбора паролей: определение заблокированных полей формы, ожидание загрузки определенных элементов веб-страницы, переключение на “горизонтальный” метод брутфорсинга.
- *Аутентификационные данные устройств* - сюда входит вся необходимая об устройствах информация, без которой невозможно было бы проводить брутфорсинг: списки стандартных паролей компаний-производителей, конфигурационные файлы для определения типа устройства, его компании, времени блокировки формы, если в его административной панели присутствует этот механизм. К этому блоку относятся также списки IP-адресов устройств, разбитых по принадлежности к разным компаниям.

Общая схема работы приложения следующая:

1. Для рассматриваемого класса устройств проводится фингерпринтинг. По его завершении формируются списки IP-адресов устройств, построенные по принципу принадлежности к какой-либо компании, а также список устройств, запрос к которым завершился ошибкой. Для этого списка проводится повторный фингерпринтинг, так как зачастую ошибки запросов возникают из-за некачественного Интернет-соединения.
2. Каждый список IP-адресов определенной компании рассматриваемого класса подается на вход модулю брутфорсинга. Внутри модуля контроллер запросов отправляет первичный запрос на IP-адрес, чтобы определить формат передачи аутентификационных данных на сервер. В зависимости от результата выбирается стратегия дальнейшего подбора паролей.
3. Если в административной панели реализован протокол HTTP Аутентификации, то обработчик ответа анализирует содержимое заголовка “WWW-Authenticate” для определения схемы. Далее данные отправляются на сервер, после чего происходит анализ статуса ответа. Комбинация считается успешной, если в случае протокола HTTP Аутентификации сервер возвращает ответ “200 OK”. Если приходит статус “401 Unauthorized”, то пробуются следующая комбинация.
4. Если административная панель содержит веб-форму, то выбирается стратегия брут-

форсинга с помощью headless-браузера. Сначала происходит инициализация полей объекта устройства данными из соответствующего конфигурационного файла. На этом этапе проверяется, присутствует ли для этого устройства инструкция ожидания снятия блокировки после неудачных попыток. Если время ожидания не превышает разумное (десять секунд), то подбор паролей осуществляется в обычном режиме. Иначе выбирается стратегия “горизонтального” брутфорсинга.

5. Сначала проверяется, стоит ли вообще в административной панели пароль. Для этого делается запрос на указанный адрес и анализируется тело ответа. Если в нем присутствует ноль либо более одного поля ввода пароля, то административная панель считается открытой и помечается статусом “open”. Иначе выполняется поиск полей ввода логина и пароля, чтобы браузер смог “вписать” туда нужную комбинацию. Параллельно с этим программа проверяет, можно ли вообще в найденные поля записывать какие-либо значения, так как некоторые административные панели автоматически блокируют вход, если в панель уже кто-то зашел.
6. После заполнения полей нужными значениями браузер имитирует ввод клавиши “enter”. Это эффективнее, чем поиск кнопки отправки данных, так как не у всех административных панелей есть такая кнопка.
7. Далее анализируется тело ответа на наличие полей ввода пароля. Если такое поле одно, то предыдущая комбинация считается неудачной, и пробуется следующая. Иначе рассматриваемое устройство помечается как имеющее небезопасную настройку и его IP-адрес помещается в соответствующий список. Если поле ввода оказывается заблокировано после очередной попытки, программа ждет указанное время, после чего возобновляет работу.
8. На выходе программа возвращает результат - построенные списки IP-адресов сетевых устройств, для которых удалось подобрать стандартный пароль.

### 4.3 Детали реализации

Ниже описаны некоторые детали реализации приложения.

Для определения успеха рассматриваемой комбинации в случае веб-формы происходит поиск поля ввода пароля на странице. Программная реализация используемого headless-браузера такова, что он считает, что веб-страница загрузилась полностью, когда загрузилась лишь ее часть. Поэтому было принято решение реализовать механизм ожидания появления определенных элементов страницы, в частности тэгов frameset (многие административные панели содержат такой тэг, соответственно появление такого элемента может означать правильную комбинацию) и input с типом password.

Как уже было упомянуто выше, некоторые административные панели рассматриваемых устройств ведут себя нестандартным образом, в частности, некоторые возвращают статус “401 Unauthorized” при передаче параметров через веб-форму. Этой аномалии соответствуют некоторые маршрутизаторы компании Eltex. Они отличаются от других маршру-

рутизаторов той же компании тем, что в заголовке “WWW-Authenticate” для них указана схема XDigest. В этом случае подбор паролей осуществляется через headless-браузер. К обладателям нестандартных административных панелей также можно отнести некоторые маршрутизаторы компании Zyxel. При заходе на такой IP-адрес через программный код возвращается статус “200 OK”, так как сначала подгружается изначальная страница, которая перенаправляет на обработчик ввода данных, реализованный через протокол HTTP Аутентификации. В этом случае при обходе маршрутизаторов Zyxel сначала идет запрос на предполагаемый обработчик. Если такой запрос возвращает статус-код 401, дальнейший подбор паролей осуществляется исходя из схемы протокола HTTP Аутентификации. Если обработчика нет, то подбор паролей осуществляется через headless-браузер.

## 5 Проведение эксперимента

Эксперимент проводился среди двух классов сетевых устройств: маршрутизаторов и сетевых камер. Выбор этих классов обусловлен тем, что чаще всего именно эти устройства выставлены для доступа по сети в силу своей специфики.

Цель эксперимента: определить уровень защищенности сетевых устройств каждого рассматриваемого класса, чьи административные панели расположены на портах 80 и 443, и провести анализ полученных результатов.

### 5.1 Маршрутизаторы

Для проведения эксперимента среди маршрутизаторов российского сегмента IoT с помощью поисковой системы Shodan было получено 5000 случайных IP-адресов маршрутизаторов. Далее среди них проводился фингерпринтинг с целью определить распространенность маршрутизаторов определенной компании среди пользователей. По окончании фингерпринтинга были получены следующие результаты: из 5000 устройств удалось распознать 2421 устройство, причем запросы на IP-адреса, не входящие в списки распознанных, либо завершились ошибкой или таймаутом, либо в полученном ответе устройства не было никакой отличительной информации о его принадлежности к какой-либо компании. Более подробно результаты фингерпринтинга изложены в диаграмме ниже.

**Примечание:** представленная диаграмма не содержит количественного результата по компании TP-Link, так как устройств этой компании оказалось 1850. Из-за этого было принято решение не включать это значение в диаграмму, так как она потеряла бы свою наглядность.

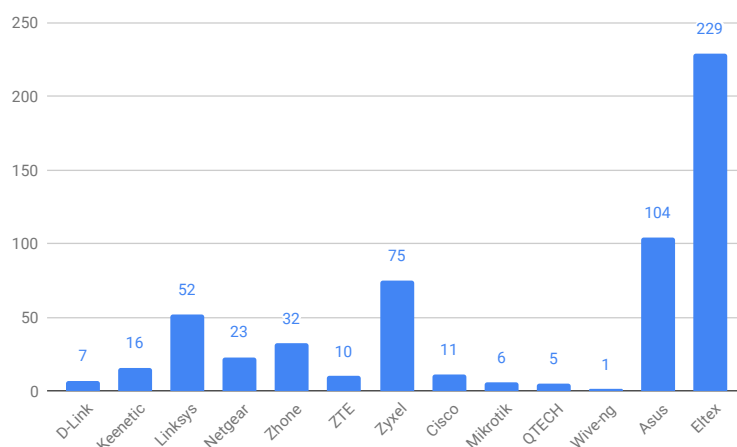


Рис. 19: Результаты работы модуля фингерпринтинга для маршрутизаторов.

Построенные списки устройств отправлялись на вход модулю брутфорсинга. Результаты поиска небезопасных настроек маршрутизаторов изложены в таблице ниже.

Компании	D-Link	Keenetic	Linksys	Netgear	Zhone	ZTE	Zyxel
Стандартные пароли	1	2	0	7	26	2	1
Всего	7	16	52	23	32	10	75

Компании	Cisco	Mikrotik	QTECH	Wive-ng	Asus	Eltex	TP-Link
Стандартные пароли	0	0	0	0	7	145	83
Всего	11	6	5	1	104	229	1850

Таблица 2

Для иллюстрации отношения выявленных незащищенных маршрутизаторов к их общему числу была построена диаграмма, показывающая процент выявленных незащищенных устройств для каждой рассмотренной компании:

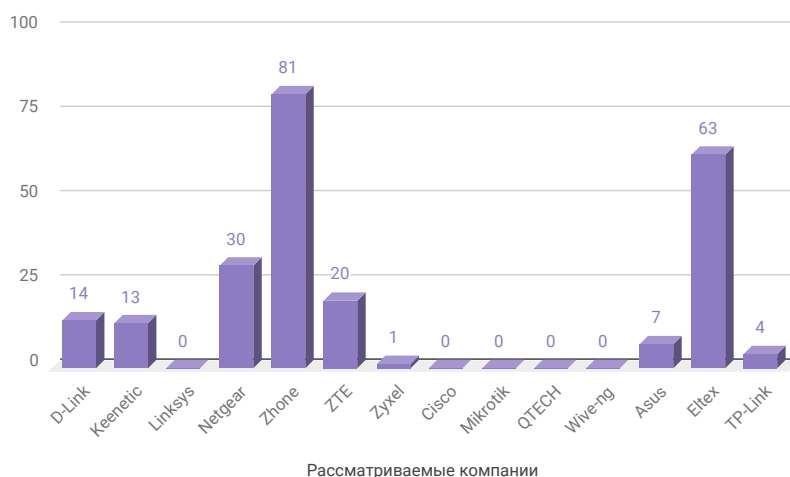


Рис. 20: Процент незащищенных маршрутизаторов по каждой компании.

Кроме того, были выявлены самые популярные стандартные пароли для вышеперечисленных компаний:

Компания	Asus	D-link	Eltex	Netgear	TP-Link	Zhone	ZTE	Zyxel
Логин и пароль	admin:admin	admin:admin	user:user	admin:password	admin:admin	admin:zhone	user:user	admin:admin

Таблица 3

## 5.2 Сетевые камеры

Эксперимент проводился среди сетевых камер российского сегмента IoT. С помощью поисковой системы Shodan для исследования было получено 1000 случайных IP-адресов сетевых камер. Далее для этой выборки эксперимент проводился аналогично всем этапам исследования маршрутизаторов. Фингерпринтинг смог определить 590 устройств из 1000. Подробные результаты представлены на диаграмме ниже. Построенные списки устройств

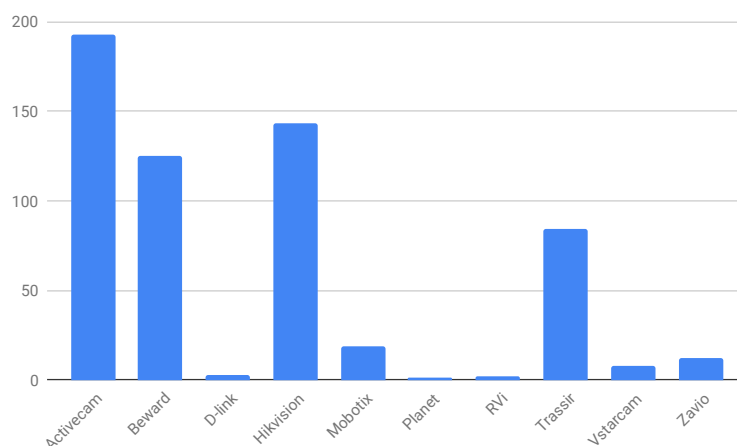


Рис. 21: Результаты работы модуля фингерпринтинга для сетевых камер.

отправлялись на вход модулю брутфорсинга. Результаты поиска небезопасных настроек сетевых камер изложены в таблице ниже.

Компании	Activecam	Beward	D-link	Hikvision	Mobotix	Planet	RVi	Trassir	Vstarcam	Zavio
Стандартные пароли	39	28	0	0	3	1	0	14	6	0
Всего	193	125	3	143	19	1	2	84	8	12

Таблица 4

Для иллюстрации отношения обнаруженных незащищенных сетевых камер к их общему числу была построена диаграмма, демонстрирующая процент этих устройств для каждой рассмотренной компании:

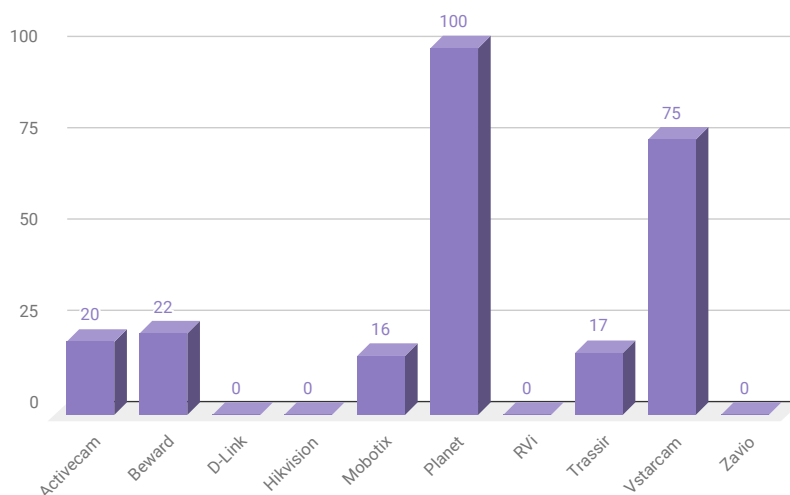


Рис. 22: Процент незащищенных сетевых камер по каждой компании.

Кроме того, были выявлены самые популярные стандартные пароли для вышеперечисленных компаний:

Компания	Activecam	Beward	Mobotix	Planet	Trassir	Vsrarcam
Логин и пароль	admin:admin	admin:admin	admin:meinsm	admin:admin	admin:admin	admin:888888

Таблица 5

### 5.3 Выводы

Полученные результаты экспериментов позволяют сделать следующие выводы:

1. Среди маршрутизаторов прослеживается следующая тенденция: популярные компании TP-Link, Zyxel, Asus имеют минимальный процент незащищенных устройств. Для некоторых малоизвестных компаний Wive-ng, QTECH это тоже верно, но это скорее связано с тем, что изначальное число устройств этих компаний для проверки было невелико, соответственно в этом случае результат нельзя экстраполировать.
2. Для маршрутизаторов прослеживается также обратная тенденция: наибольший процент незащищенных устройств имеют малоизвестные компании Zhone Technologies и Eltex. Вероятно, что это обусловлено тем, что устройства изначально имеют необычные стандартные настройки. Например, вторая по популярности стандартная комбинация для маршрутизаторов Eltex - admin:kW5i\_1bYC6os, которая с первого взгляда может показаться достаточно надежной.
3. Среди сетевых камер тех компаний, на которых был получен ненулевой результат, процент незащищенных устройств примерно одинаковый и составляет 16%-22%. Исключениями являются компании Planet и Vstarcam: изначальное количество устройств этих компаний было невелико, поэтому такой результат нельзя экстраполировать.



4. Наиболее популярной среди всех рассмотренных устройств оказалась комбинация `admin:admin`.

Среди маршрутизаторов удалось обнаружить 274 устройства со стандартным паролем, что составляет 11% от общего числа распознанных маршрутизаторов. Среди сетевых камер обнаружено 91 устройство со стандартным паролем, что составляет 15% от общего числа распознанных сетевых камер. Если проводить исследование на большей выборке устройств российского сегмента IoT (по запросу `'router country:ru port:80,443'` поисковая выдача Shodan показывает более 40 000 результатов, а по запросу `'http.favicon.hash:999357577 country:ru port:80,443'`, где первый параметр означает иконку камеры - более 31 000), то, экстраполируя полученные значения, можно получить более 9000 устройств, которые в перспективе могут стать роботами ботнета. Это довольно большое значение, если принять во внимание тот факт, что исследование проводилось с некоторыми ограничениями. Если расширить спектр рассматриваемых сетевых портов и дополнить списки стандартных паролей другими популярными комбинациями, то количество потенциальных устройств-роботов только вырастет.

Некоторые компании-производители осознают эти риски и конфигурируют свои устройства таким образом, что при его первом запуске требуется установить сложный пароль. В другом случае, когда из коробки вместе с устройством идет пароль по умолчанию, ответственность за защищенность устройства лежит на конечном пользователе.

## 6 Результаты

В рамках выполнения данного исследования были получены следующие результаты:

- Проведен обзор существующих методов решения задач, возникающих при реализации инструмента для поиска небезопасных настроек сетевых сервисов российского сегмента IoT.
- Предложен метод выбора стратегии подбора паролей, основанный на особенностях реализации веб-интерфейсов рассматриваемых устройств.
- Разработано инструментальное средство для автоматизированного поиска стандартных паролей доступа в рассматриваемых устройствах.
- С помощью разработанного инструмента проведены эксперименты среди двух классов сетевых устройств, а также выполнен анализ полученных результатов.

Поставленная задача была выполнена полностью. В качестве дальнейшего развития темы исследования и доработки инструмента можно предложить:

- Повышение качества фингерпринтинга за счет внедрения нейросетевых алгоритмов.
- Расширение спектра исследуемых сетевых портов.
- Реализация обхода сложных механизмов противодействия брутфорсинга (решение капчи, автоматизированный поиск скрипта-обработчика ввода).

## 7 Список литературы

- [1] **Matt Powell**, *Wi-Fi router security knowledge gap putting devices and private data at risk in UK homes*, Broadband Genie, 2018
- [2] **Brian Krebs**, *DDoS on Dyn Impacts Twitter, Spotify, Reddit*, KrebsonSecurity, 2016
- [3] **Avast Threat Intelligence Team**, *23.2 million potential attacks target 500 fake IoT-like devices deployed at Mobile World Congress*, Avast Blog, 2019
- [4] *Состояние рынка Интернета вещей в России в 2018 году и прогноз развития до 2022*, IDC, 2018
- [5] **Rob Sobers**, *The Likelihood of a Cyber Attack Compared*, Varonis Data Security Blog, 2018
- [6] **Ruchna Nigam**, *Mirai Compiled for New Processors Surfaces in the Wild*, Palo Alto Networks, 2019
- [7] *New VPNFilter malware targets at least 500K networking devices worldwide*, Talos, 2018
- [8] **Nmap**, *Network Mapping Tool*, <https://nmap.org>
- [9] *Топ-20 городов по количеству ip адресов*, <http://ipgeobase.ru/cgi-bin/AdvSearch.cgi>
- [10] **Zmap**, *Network Scanner*, <https://zmap.io>
- [11] **ZGrab**, *A Banner Grabber*, <https://github.com/zmap/zgrab2>
- [12] **Masscan**, *Mass IP port scanner*, <https://github.com/robertdavidgraham/masscan>
- [13] **Shodan**, *The search engine*, <https://www.shodan.io>
- [14] **ZoomEye**, *Cyberspace Search Engine*, <https://www.zoomeye.org>
- [15] *Использование offensive-методов для обогащения Threat Intelligence*, Инфосистемы Джет, 2018
- [16] **Ncrack**, *Network authentication cracking tool*, <https://github.com/nmap/ncrack>
- [17] **BruteX**, *Cracking Tool*, <https://github.com/1N3/BruteX>
- [18] **Kai Yang, Qiang Li, Limin Sun**, *Towards automatic fingerprinting of IoT devices in the cyberspace*
- [19] **T. Kohno, A. Broido, K. C. Claffy**, *Remote physical device fingerprinting*, IEEE Transactions on Dependable and Secure Computing, 2005
- [20] **S. Jana, S. K. Kasera**, *On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews*, IEEE Transactions on Mobile Computing, 2010

- [21] **F. Lanze, A. Panchenko, B. Braatz, A. Zinnen**, *Clock skew based remote device fingerprinting demystified*, IEEE Global Communications Conference, 2012
- [22] **A. S. Uluagac, S. V. Radhakrishnan, C. Corbett, A. Baca, R. Beyah**, *A passive technique for fingerprinting wireless devices with Wired-side Observations*, IEEE Conference on Communications and Network Security, 2013
- [23] **Hydra**, *Login Cracker*, <https://github.com/vanhauser-thc/thc-hydra>
- [24] **Medusa**, *A Brute Forcing Tool*, <https://github.com/jmk-foofus/medusa>
- [25] **Patator**, *A multi-purpose brute-forcer*, <https://github.com/lanjelot/patator>
- [26] **PhantomJS**, *Scriptable Headless Browser*, <http://phantomjs.org>
- [27] **Headless Chrome**, *A headless browser*, <https://developers.google.com/web/updates/2017/04/headless-chrome>
- [28] **Requests: HTTP for Humans**,  
*Python Library*, – <https://2.python-requests.org/en/master/>
- [29] **Selenium WebDriver API**,  
*Python Library* – <https://selenium-python.readthedocs.io/api.html>