Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Высшая школа кибербезопасности и защиты информации**

**Отчет**

**о научно-исследовательской работе**

**Исследование способов выявления ботнетов на мобильных устройствах**

Выполнил

студент гр. 3651003/50801 А.А.Корольков

Руководитель Е.Ю. Павленко

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Санкт-Петербург

2020

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc33729635)

[1. Классификация угроз мобильных устройств 5](#_Toc33729636)

[1.1. Угрозы На Уровне Приложений 6](#_Toc33729637)

[1.2. Веб-Угроз 8](#_Toc33729638)

[1.3. Угрозы Сетевого Уровня 9](#_Toc33729639)

[1.4. Угрозы Физического Уровня 10](#_Toc33729640)

[2. Определение ботнетов 11](#_Toc33729641)

[2.1. Основные модели ботнетов 13](#_Toc33729642)

[2.2. Способы использования ботнетов и типы ботнет-атак. 15](#_Toc33729643)

[3. Классификация ботнетов 23](#_Toc33729644)

[3.1. Интересные способы взаимодействия ботов с ботмастером 33](#_Toc33729645)

[4. Мобильные ботнеты 35](#_Toc33729646)

[4.1. Особенности и сравнение существующих мобильных ботнетов 35](#_Toc33729647)

[4.1.1. Переупаковка приложений 38](#_Toc33729648)

[4.1.2. Получение команд 39](#_Toc33729649)

[4.1.3. Обмен сообщениями 40](#_Toc33729650)

[4.1.4. Кража данных 40](#_Toc33729651)

[4.1.5. Root Exploit 42](#_Toc33729652)

[4.2. Этапы работы мобильного ботнета 44](#_Toc33729653)

[5. Методы выявления ВПО на платформе Android 48](#_Toc33729654)

[5.1. Deep learning algorithms 52](#_Toc33729655)

[5.2. Анализ существующих систем обнаружения ВПО 56](#_Toc33729656)

[5.2.1. Рассмотрение подходов к обнаружению на примере DL-Droid 59](#_Toc33729657)

[5.2.2. Сравнение глубоко обучения с другими методами машинного обучения. 60](#_Toc33729658)

[ЗАКЛЮЧЕНИЕ 62](#_Toc33729659)

[СПИСОК ИСТОЧНИКОВ: 63](#_Toc33729660)

# ВВЕДЕНИЕ

Мобильные устройства предлагают множество возможностей и большой объемов личных и конфиденциальных данных. В настоящее время большинство мобильных устройств предлагают больше вычислительных мощностей и памяти, чем многие персональные компьютеры имели несколько лет назад. Любое мобильное устройство имеет три основные функции, такие как приложения, хранение данных и подключение к сети. Эти ключевые функции делают Android устройства привлекательным инструментом для создателей вредоносных программ для атаки организации или отдельных устройств. Открытость Android делает устройства на нем привлекательной целью для киберпреступников, и на протяжении многих лет существует ряд угроз, с которыми сталкиваются мобильные устройства, такие как шпионские программы, ботнеты, уязвимые приложения, угрозы конфиденциальности, Drive-by-Download уязвимостей, фишинг, вредоносные программы, сетевые эксплойты, эксплойты браузера и Wi-Fi sniffing. Ботнет является одной из самых опасных угроз, с которыми сталкиваются мобильные устройства в последнее время.

Вредоносное ПО используется для повреждения подключенных к Интернету устройств и сбора чувствительных к раскрытию данных от частных лиц или использует шпионские программы для доступа к наиболее конфиденциальной информации на зараженном устройстве. Шпионские программы собирают всю эту информацию специально для рекламных целей. Угрозы конфиденциальности могут быть вызваны теми приложениями Android, которые не могут быть вредоносными по своей природе, но используют конфиденциальную информацию, полученную незаконно от ничего не подозревающих пользователей Android. Уязвимые приложения — это те, которые содержат недостатки, которые могут вызвать вредоносные атаки и вредоносные действия. Фишинговые мошенничества — это те, которые используют электронные письма жертвы для отправки зараженных вирусом ссылок на устройства, подключенные к интернету. В режиме drive-by-download заражения устройства загружают приложение при доступе к веб-сайту. В то время как браузер-эксплойты извлекает выгоду из уязвимостей в веб-браузерах мобильных устройств или приложениях, запущенных браузером, таких как flash player, PDF reader и многое другое, в сетевых эксплойтах киберпреступники используют недостатки операционной системы Android для преступной. Когда данные передаются с одного устройства на другое, подключенное по Wi-Fi, поскольку многие приложения не используют надлежащие правила безопасности, это приводит к краже данных, известной как Wi-Fi sniffing.

# Классификация угроз мобильных устройств

Существуют различные типы угроз для мобильных устройств, которые могут плохо повлиять на мобильные устройства, такие как вирусы и шпионские программы, которые могут заразить персональные компьютеры (ПК). Эти угрозы можно разделить на четыре широкие категории: уровень приложений, веб-уровень, сетевой уровень и физический уровень (см. Рисунок 1).

## Угрозы На Уровне Приложений

Угрозы на уровне приложений основаны на приложениях, которые являются основной функционала каждого мобильного устройства. Эти угрозы являются наиболее широко обсуждаемыми угрозами в литературе, которая представляет угрозы прикладного уровня как наиболее широко распространенную угрозу. Поскольку приложения, работающие на этих мобильных устройствах, доступны на сторонних магазинах приложений, ясно, что они могут быть целевыми направлениями для нарушений безопасности мобильных устройств (Faruki et al., 2015). Вредоносные программы — это приложения для Android, которые выполняют вредоносные действия, могут внедрять вредоносный код в мобильное устройство, которые отправляют нежелательные сообщения и позволяют противнику удаленно управлять устройством.

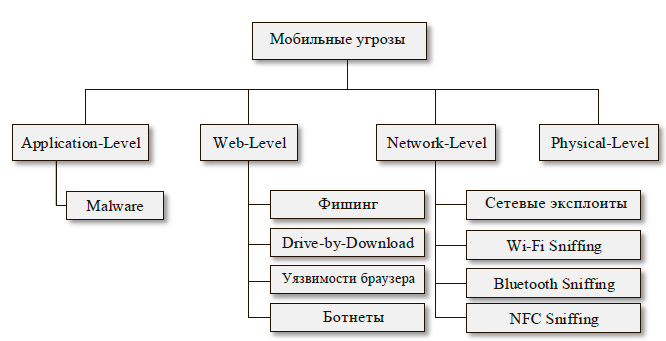


Рисунок 1 Общая классификация мобильных угроз

**Вредоносная программа(Malware**). Это специально разработано для повреждения устройств, на которых они выполняются, или сети, по которой он взаимодействует. Вредоносное ПО в основном устанавливается на устройства жертв для выполнения незаконных действий без ведома владельца. Диапазон вредоносных программ варьируется; это может быть просто всплывающая реклама или настолько опасно, что программа вызывает повреждения ОС или устройства. Кража конфиденциальных учетных данных владельца и заражение новых уязвимых устройств являются основными целями вредоносных программ. Наиболее распространенными вредоносными программами являются крадущие доступ к финансам, шифрующие данные и рекламные вредоносные программы .

## Веб-Угроз

Угрозы безопасности и конфиденциальности для мобильных устройств от веб-сайтов не редки. Наиболее опасными угрозами на веб-уровне являются фишинговые мошенничества, скачиваемые файлы и эксплойты браузера.

Фишинговые мошенничества являются ключевой угрозой на веб-уровне, которая использует электронную почту или другие приложения социальных сетей для отправки ссылок пользователю на фишинговый сайт, предназначенный для обмана пользователей и кражи конфиденциальной информации, такой как учетные данные пользователя. Фишинг является одной из семи основных угроз безопасности, выявленных Лабораторией Касперского (Kaspersky, 2015).

**Drive-by-Download**. Это скачанный файл, который относится к потенциально вредоносным приложениям, который устанавливается на устройство пользователя без разрешения. Пользователь может даже не знать, что программное обеспечение было установлено. Drive-by-downloads — это разновидность вредоносных программ, обычно обнаруживаемых на скомпрометированных веб-страницах. Просто посещая веб-страницу файл начинает загружаться и затем устанавливается в фоновом режиме на компьютере или мобильном устройстве без предупреждения пользователя.

**Уязвимостей браузера**. Это вредоносный код, который использует часть программного обеспечения или уязвимостей ОС для нарушения безопасности браузера. Эксплойты браузера выполняют эти вредоносные действия без уведомления владельца устройства.

**Ботнет** — это сеть подключенных к интернету зараженных устройств (ботов) под контролем ботмастера (киберпреступника) для выполнения киберпреступной деятельности без ведома владельца устройства. Существует два типа ботнетов: традиционные ботнеты и мобильные ботнеты. Эта работа посвящена мобильным (Android) ботнетам, но также будут рассмотрены общие теоретические сведенья о ботнетах.

## Угрозы Сетевого Уровня

Угрозы сетевого уровня могут возникать из-за подключения мобильных устройств к сотовым/мобильным сетям, локальным беспроводным сетям или ближней полевой связи (NFC). Сетевые эксплойты, Wi-Fi sniffing, Bluetooth и NFC являются основными типами сетевых угроз.

**Сетевые эксплойты**. Сетевые эксплойты используют недостатки в мобильной операционной системе или другом программном обеспечении, которое работает в локальных или сотовых сетях. После подключения они могут перехватывать подключения и данные и находить способ внедрения вредоносного программного обеспечения на телефоны пользователей без их ведома.

**Wi-Fi sniffing** захватывает данные, когда они перемещаются между устройством и точкой доступа Wi-Fi. Большинство приложений Android не используют надлежащие меры безопасности при отправке незашифрованных данных по сети. Киберпреступник может легко прочитать данные. Публичные места, такие как кафе, рестораны и книжные магазины могут не иметь защищенного соединения и вполне вероятно, что любой человек может перехватить ваши пакеты.

**Bluetooth**. Люди, которые оставляют BT всегда включенным, оставляют себя уязвимыми для связи со сторонними устройствами. Такие атаки как Blue jacking могут привести к попыткам фишинга и распространению вредоносных программ или вирусов.

**Коммуникации ближнего поля (NFC).** NFC позволяет обмениваться изображениями, приложениями и другими данными между двумя устройствами без предварительного сопряжения. Для этой цели оба устройства используют функцию, которую Google называет Android Beam.

## Угрозы Физического Уровня

**Угрозы физического уровня** более важны, чем другие упомянутые угрозы. Поскольку мобильные устройства являются небольшими, портативными и ценными, это делает их физическую безопасность более важной. Кража и неуместное использование устройств являются общей проблемой среди пользователей этих устройств. Эти устройства ценны не только потому, что их перепродают на нелегальном рынке, но и, что более важно, потому что они содержат конфиденциальные и личные данные. Большинство пользователей мобильных устройств используют свой телефон для банковских операций, социальных коммуникаций и многого другого, в то время как конечные пользователи всегда подключены к этим счетам, что делает украденный более уязвимым для преступной деятельности. Кроме того, потерянное или украденное устройство может быть использовано для получения доступа к секретным данным, хранимым на них.

# Определение ботнетов

Ботнет – это сеть компьютеров, зараженных вредоносной программой поведения Backdoor. Backdoor позволяет киберпреступникам удаленно управлять зараженными машинами (как каждой в отдельности, так и всей сетью целиком) без ведома пользователя. Такие программы называются ботами, от сокращенного английского слова robot. Злоумышленник, контролирующий сеть ботов, именуется ботмастером.

Самым первым массовым ботнетом принято считать сеть, созданную злоумышленниками в 2004 году с использованием почтового червя Beagle. Эта вредоносная программа инфицировала порядка 230 000 компьютеров по всему миру, работающих под управлением Microsoft Windows. Beagle рассылал свои копии по электронной почте, а благодаря наличию встроенного руткит-модуля умел скрывать собственное присутствие в системе и завершать процессы некоторых антивирусных программ, вследствие чего его обнаружение и удаление было непростой задачей. [1]

Управление компьютером, который заражен ботом, может быть прямым и опосредованным. В случае прямого управления злоумышленник может установить связь с инфицированным компьютером и управлять им, используя встроенные в тело программы-бота команды. В случае опосредованного управления бот сам соединяется с центром управления или другими машинами в сети, посылает запрос и выполняет полученную команду.

В любом случае хозяин зараженной машины, как правило, даже не подозревает о том, что она используется злоумышленниками. Именно поэтому зараженные вредоносной программой-ботом компьютеры, находящиеся под тайным контролем киберпреступников, называют еще зомби-компьютерами, а сеть, в которую они входят, – зомби-сетью. Чаще всего зомби-машинами становятся персональные компьютеры домашних пользователей.

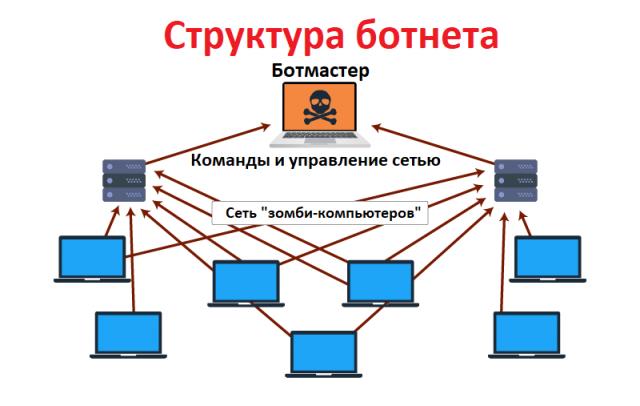


Рисунок 2- структура ботнета

Главными отличиями бота от любого другого ВПО, являются возможность получения указаний от ботмастера, а так же команда обновления собственной версии, что позволяет расширить функционал от ничего не умеющего бота, до кейлогера, спамера, вируса ворующего пароли и т.д. и т.п. Данная черта является для бота самой главной, так как позволяет оперативно улучшать меры противодействия обнаружению или уничтожить себя, в случае необходимости.

## Основные модели ботнетов

Структура ботнета часто принимает одну из двух форм: модель «клиент-сервер» или одноранговую модель (P2P, «Peer-to-peer»). Более подробно способы взаимодействия будут рассмотрены в пункте классификации ботнетов.

**Модель «клиент-сервер»**

В структуре бот-системы «клиент-сервер», создается базовая сеть, в которой один сервер выступает в роли ботмастера. Ботмастер контролирует передачу информации от каждого клиента для установки команд и управления над клиентскими устройствами.

Модель «клиент-сервер» работает с помощью специального программного обеспечения и позволяет ботмастеру сохранять постоянный контроль над зараженными устройствами.

Эта модель имеет несколько недостатков: ее можно легко обнаружить, и она имеет только одну контрольную точку. В этой модели, если сервер уничтожен, ботнет погибает.

**Одноранговая модель (PTP, Peer-to-peer)**

Чтобы преодолеть недостаток использования одного централизованного сервера, были созданы ботнеты современного типа. Они взаимосвязаны в форме одноранговой структуры.

В модели бот-сети P2P каждое подключенное устройство работает независимо как клиент и как сервер, координируя между собой обновление и передачу информации между ними.

Структура бот-сети P2P сильнее из-за отсутствия единого централизованного управления.

## Способы использования ботнетов и типы ботнет-атак.

Рассмотрим более распространение примеры использования ботнетов:

Распределенный отказ в обслуживании.

Ботнет может использоваться для атаки распределенного отказа в обслуживании (DDoS) с целью уничтожения сетевых подключений и сервисов.

Это делается путем перегрузки вычислительных ресурсов. Наиболее часто применяемыми атаками, являются атаки TCP SYN и UDP.

DDoS-атаки не ограничиваются только веб-серверами, но могут быть направлены на любую службу, подключенную к Интернету. Серьезность атаки может быть увеличена путем использования рекурсивных HTTP-потоков на веб-сайте жертвы, что означает, что боты рекурсивно следуют по всем ссылкам в HTTP-пути.

Эта форма называется «паутиной», и практикуется для более эффективного увеличения нагрузки.  


Рисунок 3 –DDOS через ботнет

Одна из крупнейших DDoS-атак ботнета 2018 года была связана с IoT и использовала вирус ботнета Mirai.

Вирус предназначался для десятков тысяч менее защищенных интернет-устройств и контролировал их, превращая в ботов для запуска DDoS-атаки.

Mirai породил множество производных запросов и продолжил расширяться, делая атаку более сложной.

Спам и мониторинг трафика.

Ботов можно использовать в качестве анализатора для выявления наличия конфиденциальных данных на зараженных машинах или компьютерах-зомби. Они также могут найти бот-сети конкурента (если они установлены на той же машине), и могут быть своевременно взломаны. Некоторые боты предлагают открыть прокси-сервер SOCKS v4 / v5 (универсальный протокол прокси для сети на основе TCP / IP). Когда прокси-сервер SOCKS включен на скомпрометированной машине, его можно использовать для различных целей, например, для рассылки спама.

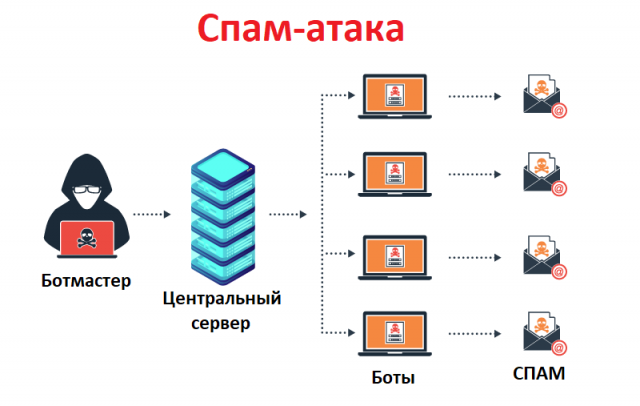
Боты используют анализатор пакетов для отслеживания информации или данных, переданных скомпрометированной машиной. Сниффер может получить конфиденциальную информацию, такую как имя пользователя и пароль.  
  


Рисунок 4 – спам атака через ботнет

**Keylogging**

С помощью кейлоггера ботмастеру легко получить конфиденциальную информацию и украсть данные пользователя. Используя программу кейлоггер, злоумышленник может собрать только те ключи, которые набраны в последовательности клюевых слов, таких как PayPal, Yahoo и другое.

Разновидность шпионского ПО, идентифицируемая как OSX / XSLCmd, перенесенная из Windows в OS X, включает в себя возможность ведения блогов и захвата экрана.

Массовая кража личных данных.

Различные виды ботов могут взаимодействовать для совершения крупномасштабной кражи личных данных, что является одним из наиболее быстро растущих преступлений. С помощью ботов можно маскироваться под известный бренд и просить пользователя предоставить личные данные, такие как пароль банковского счета, данные кредитной карты, данные о налогообложении.

Массовая кража личных данных может быть осуществлена с использованием фишинговых писем, которые вынуждают жертв вводить учетные данные для входа на веб-сайты, как eBay, Amazon или даже известные коммерческие банки.

Злоупотребление платой за клик.

Программа Google Ads позволяет веб-сайтам показывать рекламные объявления Google и тем самым зарабатывать на них деньги. Google платит деньги владельцам веб-сайтов на основании количества кликов, полученных от рекламы. Скомпрометированные машины используются для автоматического нажатия на ссылки, увеличивая количество фейковых кликов.

Adware

Рекламное программное обеспечение используется для привлечения пользователей за счет рекламы на веб-страницах или в приложениях. Они появляются на компьютерах без ведома или разрешения пользователей, а оригинальная реклама заменяется мошенническим рекламным ПО, которое заражает систему любых пользователей, кто на нее нажимает.

Рекламное программное обеспечение выглядит как безвредная реклама, но использует шпионское ПО для сбора данных браузера.

Вычисления.

Сеть ботов обладает высокой вычислительной мощностью, благодаря чему, множество ботов может использоваться для решения сложных задач. В последнее время популярность и ценность криптовалют продолжают расти, в связи с чем, в сети появились боты, занимающиеся майнингом криптовалют. Проникая в компьютер пользователя, бот использует ресурсы машины в своих целях. Чем больше зараженных машин, тем больше валюты зарабатывает злоумышленник. Мощность графического процессора может использоваться во время простоя компьютера, что позволяет сохранять скрытность.

Анонимный доступв **интерн**ет.

Злоумышленники могут обращаться к серверам в интернете, используя зомби-машины в качестве proxy-серверов и от имени зараженных машин совершать действия в сети, от просмотра запрещенных сайтов, до киберпреступлений.

Продажа и аренда ботнетов.

Злоумышленникам и нечистым на руку бизнесменам совсем не обязательно своими силами создавать ботнет «с нуля». Ботнеты самых разных размеров и производительности они могут купить или арендовать у хакеров – например, обратившись на специализированные форумы.

Стоимость готового ботнета, равно как и стоимость его аренды, напрямую зависит от количества входящих в него компьютеров. Наибольшей популярностью готовые ботнеты пользуются на англоязычных форумах.

Аренда почтового ботнета со скоростью рассылки порядка 1000 спамовых писем в минуту (при 100 находящихся в онлайне зомби-машинах) обойдется примерно в 2000 долларов в месяц.

Маленькие ботнеты, состоящие из нескольких сотен ботов, стоят от 200 до 700 долларов. При этом средняя цена одного бота составляет примерно 50 центов. Более крупные ботнеты стоят больших денег.

Зомби-сеть Shadow, которая была создана 19-летним хакером из Голландии, насчитывала более 100 тысяч компьютеров, расположенных по всему миру, продавалась за 25 000 евро. За эти деньги можно купить небольшой домик в Испании, однако преступник из Бразилии предпочел приобрести ботнет.

Фишинг.

Адреса фишинговых страниц могут довольно быстро попасть в черные списки. Ботнет дает возможность фишерам быстро менять адрес фишинговой страницы, используя зараженные компьютеры в роли прокси-серверов. Это позволяет скрыть реальный адрес веб-сервера фишера.[4]

Создание поискового спама.

Еще один вариант использования ботнетов – повышение популярности сайтов в поисковых системах. Работая над поисковой оптимизацией, администраторы ресурсов стараются повысить позицию сайта в результатах поиска, поскольку чем она выше, тем больше посетителей зайдет на сайт через поисковые системы и, следовательно, тем больше будет выручка владельца сайта, например от продажи рекламных площадей на веб-страницах. Многие компании платят веб-мастерам немалые деньги, чтобы они вывели сайт на первые позиции в «поисковиках». Владельцы ботнетов подсмотрели некоторые их приемы и автоматизировали процесс поисковой оптимизации.

Когда вы видите в комментариях к своей записи в «Живом Журнале» или удачной фотографии, выложенной на фотохостинге, множество ссылок, созданных неизвестным вам человеком, а иногда и вашим «френдом», – не удивляйтесь: просто кто-то заказал раскрутку своего ресурса хозяевам ботнета. Специально созданная программа загружается на зомби-компьютер и от имени его владельца оставляет на популярных ресурсах комментарии со ссылками на раскручиваемый сайт. Средняя цена на нелегальные услуги поискового спама – порядка 300 долларов в месяц.

Кликфрод.

Ботмастер может получать прибыль от управления кликами ботов на онлайн объявления (то есть посылать HTTP запросы на веб-страницы рекламодателя) с целью личной или коммерческой выгоды. Кликфрод может использоваться для повышения рейтинга веб-сайтов в поисковых системах. Например, ботнет Clickbot.A, использующийся для выполнения малозаметных атак мошеннического клика, симулирует поведение большого числа обычных пользователей.

Распространения другого ВПО.

Распространение других нежелательных программ, например, рекламное/шпионское программное обеспечение. Ботнеты являются хорошей платформой для распространения множества других форм вредоносного ПО.

Абузоустойчивый хостинг.

Зараженные компьютеры могут использоваться для размещения на них различного запрещенного контента. В случае блокировки подобных сайтов, ботмастер просто переносит сайт с одного зараженного хоста на другой.

# Классификация ботнетов

Классификация по архитектуре

Существует множество способов управления зараженными хостами и передача команд каждой машине. Все зависит от конкретных предпочтений бот-мастера. В зависимости от типа используемого протокола командным центром может выступать:

* **Веб-сервер** - управление осуществляется через веб-интерфейс. В настоящее время самый распространенный способ (кстати, именно его использует нашумевший Zeus).
* **Instant Message среда** - передача команд по одному из IM-протоколов (ICQ, jabber, MSN и т.п.). Используется в в бот-сетях с небольшим количеством хостов.
* **IRC** - командный центр находится на одном из IRC-каналов. Морально устаревший метод осуществления контроля. В настоящее время практически не используется из-за высокой степени вероятности изолирования (перехвата) командного центра.
* **Twitter-среда** - управление ботнетом посредством передачи команд в твиттер-аккаунте. Довольно экзотический способ, но имеет право на существование в условиях повсеместной распространенности социальных сетей и веб-сервисов, предоставляющих свои API. Кстати, в данном случае можно не задумываться о том, что командный центр может упасть из-за нагрузки своей же «армии», ведь большинство данных проектов рассчитаны на огромную аудиторию и имеют соответствующие средства масштабируемости.
* **TCP/IP-based** – управление посредством протоколов, базирующихся на стеке TCP/IP. Под эту категорию попадают все остальные способы, основанные на передаче команд по экзотическим и самописным протоколам.

Классификация по протоколам связи

Для передачи боту команд хозяина ботнета необходимо, как минимум, установить сетевое соединение между зомби-компьютером и компьютером, передающим команду. Все сетевые взаимодействия основаны на сетевых протоколах, определяющих правила общения компьютеров в сети. Поэтому существует классификация ботнетов, основанная на используемом протоколе общения. По типу используемых сетевых протоколов ботнеты делятся на следующие группы.

**IRC-ориентированные:** это один из самых первых видов ботнетов, где управление ботами осуществлялось на основе IRC (Internet Relay Chat). Данный протокол связи уже не используется и считается вымершим. Каждый зараженный компьютер соединялся с указанным в теле программы-бота IRC-сервером, заходил на определенный канал и ждал команды от своего хозяина. Сообщения, получаемые ботами от ботмастера, легко перехватывались, нередки были случаи, когда контроль над IRC-сервером переходил в другие руки и ботнет попросту “угоняли”.

**IM-ориентированные:** не очень популярный вид ботнетов. Отличается от своих IRC-ориентированных собратьев только тем, что для передачи данных используются каналы IM-служб (Instant Messaging), например AOL, MSN, ICQ и др. Невысокая популярность таких ботнетов обусловлена сложностями, возникающими при создании отдельного аккаунта IM-службы для каждого бота. Дело в том, что боты должны выходить в Сеть и постоянно присутствовать онлайн. Поскольку большинство IM-служб не позволяют входить в систему с разных компьютеров, используя один и тот же аккаунт, у каждого бота должен быть свой номер IM-службы. При этом владельцы IM-служб всячески препятствуют любой автоматической регистрации аккаунтов. В результате хозяева IM-ориентированных ботнетов сильно ограничены в числе имеющихся зарегистрированных аккаунтов, а значит и в числе ботов, одновременно присутствующих в Сети. Конечно, боты могут использовать один и тот же аккаунт, выходить в онлайн один раз в определенный промежуток времени, отсылать данные на номер хозяина и в течение короткого промежутка времени ожидать ответа, но это все весьма проблематично: такая сеть реагирует на команды очень медленно.

**Веб-ориентированные:** наиболее популярная ветвь ботнетов, ориентированная на управление через www. Бот соединяется с определенным веб-сервером, получает с него команды и передает в ответ данные. Такие ботнеты популярны в силу относительной легкости их разработки, большого числа веб-серверов в Интернете и простоты управления через веб-интерфейс.

**Другие:** кроме перечисленных выше существуют и другие виды ботнетов, которые соединяются на основе своего собственного протокола, базируясь лишь на стеке протоколов TCP/IP: используют лишь общие протоколы TCP, ICMP, UDP. [2]

Классификация ботнетов по топологии и способу управления

Класифицируя ботнеты по топологии и схеме управления можно выделить 3 основных вида:

1. **Ботнет с единым центром управления:** в ботнетах с такой архитектурой все зомби-компьютеры соединяются с одним центром управления, или C&C (Command&Control Centre). C&C ожидает подключения новых ботов, регистрирует их в своей базе, следит за их состоянием и выдает им команды, выбранные владельцем ботнета из списка всех возможных команд для бота. Соответственно, в C&C видны все подключенные зомби-компьютеры, а для управления централизованной зомби-сетью хозяину сети необходим доступ к командному центру.

Ботнеты с централизованным управлением являются самым распространенным типом зомби-сетей. Удобство централизованных схем объясняется наличием единого центра, к которому обращаются боты с целью получения задания. Не нужно беспокоиться о своевременном получении команды конкретным ботом. Факты получения, выполнения, успешного/неуспешного завершения задачи легко фиксируются, что позволяет вести детальную статистику. Однако централизованная топология остается актуальна лишь для небольших бот-сетей по следующим причинам:

* плохая масштабируемость (с ростом числа зараженных хостов растет нагрузка на командный центр и увеличивается вероятность осуществления атаки типа «отказ в обслуживании» на сервер, передающий задания);
* централизованное управление (высокая вероятность изолирования командного центра, что немедленно «парализует» весь ботнет).

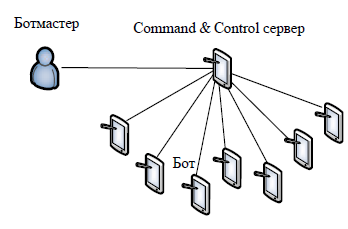


Рисунок 5 – Схематичное изображение централизованной системы управления

1. **Децентрализованные ботнеты, или P2P-ботнеты:** (от англ. “peer-to-peer”, что означает «соединение типа “точка-точка”»). В случае децентрализованного ботнета боты соединяются не с центром управления, а с несколькими зараженными машинами из зомби-сети. Команды передаются от бота к боту: у каждого бота есть список адресов нескольких «соседей», и при получении команды от кого-либо из них он передает ее остальным, тем самым распространяя команду дальше. В этом случае злоумышленнику, чтобы управлять всем ботнетом, достаточно иметь доступ хотя бы к одному компьютеру, входящему в зомби-сеть.

На практике построение децентрализованного ботнета не очень удобно, поскольку каждому новому зараженному компьютеру необходимо предоставить список тех ботов, с которыми он будет связываться в зомби-сети. Поэтому ботмастеру легче использовать комбинированный тип связи, боты обращаются к C&C, где они получит список ботов-«соседей», а затем уже переключаются на взаимодействие через P2P-подключения. Такая смешанная топология также относится к типу P2P, хотя на отдельном этапе боты используют C&C. Бороться с децентрализованными ботнетами гораздо сложнее, поскольку в действующем ботнете центр управления отсутствует. Децетрализованная топология полностью лишена недостатков централизованной и, в силу особенностей своей архитектуры, обеспечивает большую «живучесть» бот-сети. Но, как всегда, за большими плюсами имеются свои минусы:

* Peer-to-peer схема предполагает уведомление каждого бота о существовании других зараженных машин. Эта процедура является довольно «палевной», так как необходимо хранить на каждой зараженной рабочей станции огромный (мы рассматриваем большие ботнеты) файл со списком IP всех ботов сети и в реальном времени его обновлять, если требуется доставка команд каждой «боевой единице».
* Обновление списка и получение команды требуют дополнительно открытых портов на зараженной машине, что увеличивает вероятность обнаружения ботнета.
* Значительное время затрачивается на передачу задания от хоста к хосту (P2P) и, соответственно, растет общее время его выполнения.
* Трудность ведения статистических данных: сколько ботов получили/выполнили задание.

В большинстве случаев обилие недостатков и сложность реализации P2P-ботнетов являются решающим фактором в пользу выбора централизованной топологии.

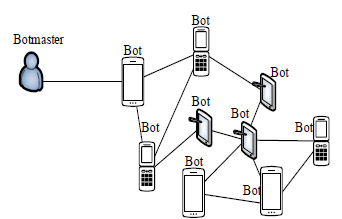


Рисунок 6 – схема децентрализованного ботнета

1. **Гибридная архитектура ботнета.**

Гибридная архитектура представляет собой сочетание централизованной и децентрализованной архитектуры, как показано на рисунке 7. Гибридная архитектура состоит из двух типов ботов, а именно, слуги и клиента. Боты подключаются к гибридному ботнету в качестве клиента или слуги. Мониторинг и обнаружение ботнетов с гибридной архитектурой проще, чем обнаружение ботнетов с централизованной и децентрализованной архитектурой.

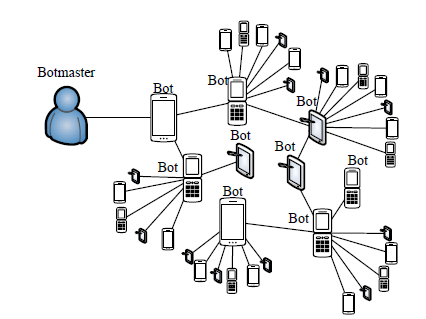


Рисунок 7- Схема гибридной ботнет сети

Сравнение трех типов ботнетов

Подводя итоги по типам ботнетов была составлена таблица, изображенная ниже. В нем отражены характеристики топологии, шанса обнаружения, задержки в передаче сообщений, и надежности

|  |  |  |  |
| --- | --- | --- | --- |
| Архитектура | Централизованная | Децентрализованная | Гибридная |
| Топология | Звезда | P2P | Смешанная, случайная |
| Обнаружение | Среднее | Средне-низкое | Высокое |
| Задержка сообщений | Очень низкая | Средняя | Высокая |
| Надежность | Низкая | Средняя | Высокая |

Таблица 1 -сравнение типов ботнетов

Классификация ботнетов по способам распространения

Для заражения используются спам-рассылки, постинг сообщений на форумах и в социальных сетях и другие приемы; часто бот наделяется функцией самораспространения, как вирусы или черви.

**Социальная инженерия:** чтобы заставить потенциальную жертву установить бот, используют приемы социальной инженерии. Например, предлагают посмотреть интересное видео, для чего требуется скачать специальный кодек, либо на почту приходит фальшивое письмо от налоговой службы.

**Drive-by-download:** при посещении пользователем зараженной веб-страницы на его устройство через различные «дыры» в приложениях – прежде всего в популярных браузерах – загружается вредоносный код. Для эксплуатации слабых мест используются специальные программы – эксплойты. Они позволяют не только незаметно загрузить, но и незаметно запустить вирус или бот. Такой вид распространения вредоносного ПО наиболее опасен, ведь, если взломан популярный ресурс, заразятся десятки тысяч пользователей.

**Самораспространение:** бот можно наделить функцией самораспространения по компьютерным сетям. Например, он может распространяться путем заражения всех доступных исполняемых файлов или путем поиска и заражения уязвимых компьютеров сети.

## Интересные способы взаимодействия ботов с ботмастером

**Бонеты на основе Tor:** в 2013 году, количество активных клиентов Tor составляло около 500 тыс. Но в определенный момент за 2 недели число внезапно возросло. В течение недели число онлайн-клиентов Tor утроилось, а на начало сентября составило 2,5 млн. Члены Tor Project начали выяснять причину такого внезапного наплыва новых пользователей. Они связались с разработчиками операционных систем, чтобы узнать, не начал ли кто-либо из них встраивать Tor в свои дистрибутивы, но ничего такого не нашли. Также они рассмотрели возможность того, что причиной роста послужила популяризация Tor среди журналистов, политических активистов и других типичных пользователей Tor после разоблачений деятельности АНБ, но по всем признакам это не могло вызвать такой сильный эффект. В ходе дальнейшего расследования они пришли к выводу, что миллионы новых клиентов Tor были элементами ботнета, чьи владельцы недавно решили по какой-то причине использовать сеть Tor.

В июле было обнаружено два различных семейства вредоносных программ, использующих скрытый протокол сервиса TOR для взаимодействия с командным C&C сервером. Такой сервис хорошо подходит при организации скрытого канала связи, но в то же время является довольно медленным способом кражи больших объемов данных с зараженного компьютера. Для киберприступников этот путь использования скрытой службы протокола общения с C&C является предпочтительным при получении обновлений о данных конфигурации или загрузки дополнительных вредоносных модулей. С технической точки зрения подобные ботнеты довольно сложны для расследований, как и представляет сложность обнаружение истинного местоположения C&C. [3]

**Бонеты в twitter среде:**боты, не имеющие командного центра. Управляются с помощью твиттов от ботмастера. Бот использует Twitter API для получения команды из ленты твиттера и выполняет их на клиентской машине. Огромным плюсом является простота разработки и полная уверенность в работоспособности сервера, т.к. соцсеть рассчитана на огромную аудиторию и имеют соответствующие средства масштабируемости.

# Мобильные ботнеты

## Особенности и сравнение существующих мобильных ботнетов

Для определения возможных характеристик ботнета были выбраны для оценки следующие ботнеты Android: Zitmo, BaseBridge, AnServerBot, DroidDream, DroidKungFu, Geinimi, Nickispy, Pjapps, RootSmart, FakePlayer, SMSspacem, TigerBot и ADRD [6]. Мы загрузили приведенные выше примеры вредоносных программ с мобильного хранилища contagio: общей библиотеки вредоносных программ для исследовательских целей. Мы оценили образцы вредоносных программ на основе разрешений, чтобы определить общие характеристики ботнета. К общим характеристикам относятся: переупаковка приложения / программного обеспечения, получение команд, обмен сообщениями, кража информации, приложения, найденные на сторонних рынках приложений, разрешения, загрузка дополнительного контента и обновление устройства. Некоторые из этих идентифицированных характеристик тесно связаны с традиционными функциями ботнета, такими примерами являются команды приема, кражи характеристик информации и загрузки контента. Поэтому можно использовать эти идентифицированные характеристики для обнаружения бот-сетей на устройствах Android. В таблице 2 показаны вредоносные действия некоторых ботнетов Android.

|  |  |
| --- | --- |
| **Название** | **Возможности действий** |
| Zitmo | - Зараженные SMS-сообщения  - Мобильные банковские атаки  - Кражи TAC  - Незаконные сделки |
| AnserverBot | Кража личных данных |
| Geinimi | - Отправить электронную почту и SMS,  - Совершать телефонные звонки,  - Обновить адрес C & C,  - Добавлять новые ярлыки приложений |
| PjApps | - Отправить смс  - Кража личных данных  - установить новое приложение,  - Открыть URL в браузере телефона |
| DroidDream | - Кража личных данных  - Скачать вредоносные приложения  - Попытки подключиться к корневому устройству. |
| RootSmart | - Кража личных данных |
| DroidKungFu | - Скачать вредоносные приложения  - Кража IMEI |
| SMSspacem | - Отправьте SMS на все контакты на телефоне, содержащие ссылку HTTP. |
| FakePlayer | - Манипулирование доставкой контента  - Отправка смс |
| ADRD | - Отправить информацию IMEI / IMSI на удаленный сервер.  - Получать команды с удаленного сервера.  - Скачать пакеты.  - Получить запись URL. |
| Smishing.D | -Обнаружение ввода сообщений  -Подмена банковских URL  -Кража чувствительной информации  -Кража паролей |
| Morder.A | -Отслеживание местоположения  -Утечка контактов и информации с памяти устройства  -Удаление и загрузка файлов  -Фотографирование на камеру  -Запись звонков  -Исполнение shell команд |
| MazarBOT.A | -Отправка платных СМС -Кража СМС через настроенный бекдор в устройстве |

Таблица 2. Примеры Android ботнетов и их дейстивй

### Переупаковка приложений

Все мобильные операционные системы, такие как Android, iPhone предоставляют пользователю возможность загружать приложения из своих официальных магазинов. Но в некоторых операционных системах, таких как Android, пользователи также могут загружать приложения из неофициальных магазинов. Распространение вредоносного кода для управления ботнетом обычно принимает форму приложения. Авторы бот-сетей выполняют реверс-инжиниринг этих законных приложений и перепаковывают их с вредоносным кодом вместе с исходным кодом. Это самый распространенный способ распространения бот-сетей. «Pjapps» — это пример приложения, содержащего вредоносную программу, которая отображает традиционные функции ботнета. Pjapps поставляется с приложениями, доступными в неофициальных магазинах приложений сторонних производителей. Вредоносные программы позволяют открыть бэкдор на зараженном устройстве и, таким образом, получать команды от удаленного сервера [7]. Другой такой пример - «OPFAKE». Более 500 вариантов OPFAKE можно загрузить из некоторых сторонних магазинов приложений. Он содержит вредоносный рекламный код и отправляет сообщения на номера премиум-класса.

### Получение команд

Важной характеристикой ботнета является возможность получать команды с удаленного сервера. Методы командования и управления (C & C) используются в качестве интерфейса для отправки команд ботмастеров целевым устройствам и получения от них ответов [8]. Современные методы, используемые мобильными ботнетами, очень похожи на эти традиционные методы. Первый вариант - отправлять команды напрямую с сервера управления и команд (C & C) боту по мере необходимости. Другой вариант - позволить боту регулярно связываться с сервером C & C и спрашивать, доступны ли новые команды. Любой контакт с удаленным сервером является явным признаком того, что в устройстве присутствует мобильный ботнет. AnserverBot - один из примеров такого мобильного ботнета. В дополнение к обнаружению и отключению решения безопасности на зараженном устройстве AnserverBot регулярно проверяет свою подпись, чтобы проверить его целостность, чтобы защитить себя от любых изменений. «TigerBot» — это бот, который управляется с помощью SMS вместо любых веб-технологий. Он обнаруживает сообщения C & C и скрывает их от владельцев мобильных устройств. Вместо сбора личных данных, таких как SMS-сообщения, он записывает голосовые звонки и даже окружающие звуки.

### Обмен сообщениями

Основной мотив традиционного ботнета заключается в получении финансовой выгоды. Мобильные ботнеты используют SMS-сообщения для сбора денег, отправляя сообщения на номера с премиальным тарифом. Эти номера премиум-класса являются телефонными номерами, используемыми для определенной услуги, и тарифы на них превышают обычные телефонные звонки. Посылая SMS-сообщения через равные промежутки времени на такие номера, ботнет может принести значительные суммы денег. «Zitmo» — это бот, который заражает различные мобильные операционные системы, такие как Symbian, Windows Mobile, BlackBerry и Android. Он в основном перехватывает SMS-сообщения, которые банки отправляют клиентам путем кражи мобильных номеров аутентификации транзакций (TAC). «Android.Bmaster» заработал миллионы долларов с помощью премиальных SMS.

### Кража данных

Ботнет отправляет информацию о персональных данных на удаленный сервер. Этот вид активности обычно происходит после установки вредоносного приложения. Обычно информация, собираемая ботнетами, может включать: номер IMEI (международный идентификатор мобильного оборудования), номер IMSI (международный идентификатор мобильного абонента), местоположение GPS, номер телефона, модель устройства, список контактов, установленные приложения, адреса электронной почты, истории браузера и версию SDK. , «Geinimi» - это первое вредоносное ПО, отображающее традиционные функции ботнетов. Эта вредоносная программа собирает личную информацию об устройстве и передает эту информацию на удаленный сервер. «PjApps.A» собирает IMEI, IMSI, номер телефона, сервисный центр SMS, информацию ICCID и отправляет ее на удаленный сервер. «TigerBot.A» отправляет SMS и IMEI на удаленный сервер. Еще одной фундаментальной характеристикой ботнета является осведомленность о местоположении, то есть способность определять географическое положение.

Возможность дополнительный загрузки контента. Последняя особенность ботнетов Android - возможность загрузки дополнительного контента. Этот контент, обычно злонамеренный по своей природе, помогает и повышает производительность ботнета. Дополнительный контент либо загружается приложением динамически, либо запрашивается у пользователя выполнить необходимую загрузку. Вредоносное ПО «DroidDream» загружает другое приложение на зараженные цели. Эти новые загрузки не позволяют удалить «DroidDream».

### Root Exploit

С ростом бот-сетей их мотивация сместилась в сторону эксплойтов, которые могут улучшить функциональность вредоносных программ. Хорошо известный эксплойт — это эксплойт «rage against the cage», который позволяет пользователю получить root-доступ на устройствах. Такие эксплойты открывают новые возможности для развития мобильных ботнетов. Вредоносное ПО, называемое RootSmart, способно получить root-доступ на устройствах с некоторыми версиями операционной системы Android. «DroidDream» заразил более 50 приложений в официальном магазине Android. Благодаря таким функциям, как кража данных, загрузка содержимого, он также получает корневой доступ к устройству. Конечная цель вредоносного ПО «DroidDream» заключалась в создании ботнета, затрагивающего почти 200 000 пользователей [9].

Самый простой способ загрузить приложения — это зайти в официальный магазин, такой как Android Market. Но есть и целый ряд независимых магазинов мобильных приложений. Эти магазины продают приложения на разных платформах и предлагают некоторые уникальные функции, которые вы не можете найти в официальных магазинах приложений. Помните, что пользователи Android могут покупать приложения в сторонних магазинах приложений в дополнение к официальным магазинам на своих телефонах. Эти магазины содержат огромное количество вредоносных программ для Android.

Каждое приложение Android может определять разрешения, которые контролируют доступ к конфиденциальным ресурсам и функциям. Существует 130 определенных для Android разрешений [10], среди которых 122 разрешения доступны сторонним приложениям. Разрешения должны быть включены в файл AndroidManifest.xml. Разрешения Android используются для информирования пользователей о рисках безопасности установки приложений. Но исследователи проанализировали, что уведомления о разрешениях Android в основном игнорируются пользователями. Структура для определения разрешений в файле манифеста выглядит следующим образом:

<manifest xlmns: android ...> ... < uses-permission android: name =” android.permission.INTERNET” /> <application></ manifest >

Ботнеты Android обычно используют следующие разрешения:

- READ\_CONTACTS

- READ\_PHONE\_STATE

- ACCESS\_FINE\_LOCATION

- WRITE\_CONTACTS

- SEND\_SMS

- WRITE\_SMS

- READ\_SMS

- RECEIVE\_SMS

- READ\_PHONE\_STATE

- INTERNET

- ACCESS\_GPS

- INSTALL PACKAGES

- WRITE\_INTERNAL\_STORAGE

- EXTERNAL STORAGE

Следовательно, файл AndroidManifest.xml предоставляет важную информацию пользователю о конкретном приложении и содержит идентифицируемые характеристики ботнета Android. Некоторые разрешения подробнее рассмотрим в следующих пунктах.

## Этапы работы мобильного ботнета

Недавно обнаруженные мобильные ботнеты становятся все сложнее. Каждый новый вариант этих вредоносных программ использует несколько серверов C & C для запуска команд на целевом устройстве. В будущем мобильные ботнеты только продолжат развиваться, поскольку основной мотивацией таких угроз является финансовая выгода, отправляя сообщения на премиальные номера или читая сообщения из банков. Что касается мер безопасности, то существуют некоторые контрмеры против угроз, создаваемых мобильными ботнетами. Чтобы выяснить, является ли приложение безопасным или имеет функции ботнета, характеристики, описанные в предыдущем разделе, могут оказаться полезными методами обнаружения. В этом разделе будет обсуждаться модель разработки мобильного ботнета (то есть этапы, на которых бот-мастер разрабатывает ботнет), и на каждом этапе выявляются характеристики ботнета. Ниже приведены этапы разработки ботнета Android:

* Заражение
* Распространение
* Выполнение полезной нагрузки

Жизненный цикл ботнета начинается с фазы заражения, когда ботмастер модифицирует легитимное приложение, чтобы освободить место для вредоносного кода бота. Ботмастер получит исходный код легального приложения, используя методы обратного инжиниринга. Затем он внесет изменения в код, добавив вредоносное содержимое. Это приведет к переупаковке приложения, что является первой характеристикой ботнета. Фаза распространения, которая является следующей по очереди, полностью посвящена распространению этого переупакованного приложения. В случае компьютерных устройств зараженный код может распространяться по электронной почте, файлообменникам или вредоносным URL-адресам и т. Д. [9] .

Но в случае мобильных устройств наиболее распространенным и эффективным средством передачи информации для ботмастеров являются сторонние магазины приложений. Для устройств Android доступно множество магазинов приложений сторонних производителей. Разработчики могут загружать свои приложения в эти магазины очень легко без каких-либо проблем с безопасностью. Следовательно, ботмастер загружает перепакованное приложение в хранилище приложений для распространения. Эта фаза приводит к седьмой характеристике ботнета (сторонние магазины приложений).

Последним этапом модели разработки Android-ботнета является этап выполнения, на котором ботнет выполнит свою задачу. Назначение ботнета может иметь несколько возможностей, включая атаки типа «отказ в обслуживании», кражу информации, обмен SMS-сообщениями или получение команд и т. Д. Остальные характеристики ботнета появляются на последнем этапе. Для идентификации вредоносных программ в приложении Android выполняется обратный инжиниринг. В обратном инжиниринге некоторые инструменты ApkTool, Dex2Jar, AndroGuard и т. Д. Используются для получения исходного кода, а затем для сканирования каждой строки кода, чтобы определить, является ли это вредоносным ботнетом или нет. Это может занять много времени, если аналитик по безопасности оценивает каждую строку кода. Но общие характеристики ботнета могут легко помочь обнаружить мобильный ботнет. Таким образом, характеристики ботнета могут использоваться во время процесса обнаружения ботнета. Процесс обнаружения ботнета описывает шаги, которые может предпринять аналитик по безопасности, чтобы определить, является ли определенное приложение безопасным или вредоносным по отношению к ботнетам. Шаги, выполняемые в процессе обнаружения ботнета [12] , включают: поиск, исследование и идентификация. Чтобы найти возможные вредоносные приложения, аналитик по безопасности использует седьмую характеристику ботнета и выбирает приложение из сторонних хранилищ приложений. После загрузки приложения аналитик безопасности может определить, является ли исследуемое приложение переупакованным приложением или нет. В случае переупаковки приложений путем изучения разрешений, определенных в AndroidManifest.xml, это может привести к выявлению возможных угроз, создаваемых приложением. Например, следующие основные разрешения Android являются индикаторами для ботнета Android:

* INTERNET, ACCESS\_NETWORK\_STATE RECEIVE\_BOOT\_COMPLETED: Приложение запрашивает доступ в интернет, и автозапуск после загрузки устройства
* INTERNET\_RECEIVE SMS, SEND\_SMS: Приложение может принимать команды из вне, и отсылать смс
* ACCESS\_WIFI\_STATE, CHANGE\_ WIFI\_STATE: Приложение может управлять сетевыми настройками.
* INTERNET, READ\_ PHONE\_STATE, READ\_CONTACTS: Приложение может украсть ваши личные данные
* INTERNET, SEND\_SMS: Приложение может отсылать СМС и другую информацию.
* INTERNET, INSTALL\_PACKAGES: Приложение может устанавливать сторонние программы.

Наличие этих наборов разрешений не всегда указывает на вредоносное ПО, однако это поможет аналитику по безопасности сосредоточиться на коде (вызовах API) вышеупомянутых характеристик, а не оценивать все структуры кода. Оценив конкретную область кода, могут ли аналитики безопасности сделать вывод, представляет ли приложение какие-либо угрозы, связанные с ботнетами, и какое вредоносное поведение выполняет приложение? Хотя эти характеристики ботнета очень полезны, но пользователи всегда должны загружать приложения из официального магазина Android (Google Play), поскольку он очень безопасен. Кроме того, во время установки приложения пользователь должен обращать внимание на разрешения, запрашиваемые этим приложением. Тогда самым старым из возможных способов защиты является использование приложений защиты от вредоносных программ для мобильных устройств. Пользователи должны регулярно обновлять свои антивирусные программы для выявления новых угроз.

# Методы выявления ВПО на платформе Android

Мобильные вредоносные программы требуют сложного анализа. Одним из критических моментов мобильных телефонов является то, что они представляют собой систему постоянно получающую события, которая позволяет вредоносным программам реагировать на приходящие SMS, установку положений и т. д., повышая сложность автоматизированных методов анализа вредоносных программ. Кроме того, приложения могут использовать сервисы и действия и интегрировать различные языки программирования (например, Java и c++) в одном приложении. Основываясь на способах классификации приложений, анализ может быть статическим, динамическим и гибридным, объединяющий статический и динамический анализ. Каждый из этих анализов имеет свои сильные и слабые стороны.

Статический анализ сканирует части приложения, фактически не выполняя их. Этот метод включает в себя сигнатурный анализ, проверку разрешений приложения и анализ его отдельных компонентов. Анализ на основе сигнатур ищет особенности и выделяет отличительные знаки для идентификации конкретных вредоносных программ. Следовательно, он не может распознать вариацию или неопознанную вредоносную программу. Анализ на основе разрешений распознает запросы приложения для распознавания вредоносных программ. Методы, основанные на анализе компонент, декомпилируют приложение для анализа и проверки байт-кодов значимых компонентов (активити, сервисы и т.д.). Основными недостатками статического анализа являются отсутствие реального выполнения и условий, которые могут возникнуть только во время выполнения программы. Кроме того, существуют проблемы в случае обфускации кода или динамической загрузки кода.

Метод динамического анализа включает выполнение приложения либо на виртуальной машине, либо на физическом устройстве. Во время выполнения ПО за поведением приложения наблюдают и могут вмешиваться в его выполнение. Динамический анализ ближе в анализу реального поведения программы, чем статический анализ. Пути выполнения кода во время анализа, являются подмножеством всех доступных путей. Основной целью анализа является достижение высокого уровня покрытия кода, поскольку каждое возможное событие должно быть активировано для наблюдения за любым возможным вредоносным поведением [31]. Основные недостатки динамического анализа заключаются в том, что динамический анализ требует значительных ресурсов по сравнению со статическим, что препятствует его распределению на сотовых телефонах. Кроме того, динамический анализ малоэффективен при низком охвате кода. В последнее время вредоносное ПО пытается распознать выполнение на эмуляторе и другие платформы динамического анализа и воздерживаются от демонстрации их полезной нагрузки. Следовательно, некоторые динамические анализаторы уязвимы для уклонения ВПО от анализа [23].

Третий тип анализа называется гибридным анализом, который объединяет статический и динамический анализ. Благодаря комбинации статического и динамического анализа, из анализа можно извлечь больше функций. Использование соответствующих функций для анализа даст гораздо лучший результат. Гибридный анализ может быть использован для преодоления недостатков статического и динамического анализа, таких как сложность анализа при методах запутывания кода и ресурсоемкая «песочница» динамического анализа. Хотя все методы обычно используются при анализе и обнаружении вредоносных программ, эти методы также применимы в качестве основы для анализа и обнаружения мобильных ботнетов. Следовательно, гибридный анализ является оптимальным выбором при обнаружении ботнетов.

Многие инструменты были разработаны для извлечения различных динамических и статических функций из файлов пакета приложений для Android (APK). В исследовании [14] была представлена Kirin, службу безопасности для Android, которая предоставляет правила безопасности для приложений во время установки. В другой работе [15] было предложено машинное обучение с использованием моделей байесовской классификации для обнаружения мобильных вредоносных программ наряду со статическим анализом кода с использованием функций, таких как разрешения, вызовы API и системные команды Linux.

Между тем, динамический анализ не проверяет исходный код, а выполняет приложения, отслеживает и регистрирует каждую соответствующую операцию выполнения. В исследовании [16] было предложено Andromaly, приложение для обнаружения вредоносных программ на базе Android, с использованием машинного обучения. Он был протестирован с использованием 88 параметров для описания поведения приложения, такого как использование процессора и памяти, энергопотребление, загрузка сети и количество запущенных процессов. Хотя этот подход дает хороший результат, он не способен обнаружить вредоносное ПО, которое избегает системных вызовов с правами root.

Правильный выбор контролируемых функций при обнаружении ботнета оказывает значительное влияние на результаты экспериментов. Наиболее полное исследование по выбору функций для мобильных вредоносных программ сделано в [17]. Они выделили все методы выбора функций для обнаружения мобильных вредоносных программ и разбили все доступные функции на четыре группы: статические, динамические, гибридные и метаданные приложения.

Так же, в исследовании [18] было предложено обнаружение мобильных ботнетов путем проверки аномального сетевого потока через виртуальную частную сеть (VPN) с частотой обнаружения 94,6%.

## Алгоритмы глубокого обучения

Выбор правильного алгоритма классификации в соответствии с целью имеет важное значение, учитывая его влияние на производительность и точность обнаружения. В Таблице 3 кратко представлены часто используемые алгоритмы глубокого обучения для обнаружения вредоносных программ на Android.

Талица 3. Часто используемые алгоритмы машинного обучения.

|  |  |  |
| --- | --- | --- |
| Алгоритм | Сильные стороны | Слабые стороны |
| Глубокая нейронная сеть (DNN) общий метод глубокого обучения, используемый для классификации. Состоит из более чем 2 скрытых слоев. | Успешна применяется во многих приложениях | Процесс обучения модели очень долгий. |
| Ограниченные машины Больцмана (RBM) используются в качестве генеративных моделей различных видов данных, которые могут изучать дисперсию вероятности по определенному расположению ее входных данных. | -Позволяет понять совпадает ли образец с нормальным распределением  -Может использоваться для извлечения наборов для обучения других сетей | -Сложно хорошо обучить  -Вычисление вероятностей занимает много времени |
| Сверточная нейронная сеть (CNN) CNN состоит из трех слоев: сверточного слоя, слоя субсэмплинга (объединяющего слоя) и полностью связанного слоя. Сверточный слой использует процедуру свертки для выполнения разделения весов. Слой субсэмплинга стремится уменьшить размерность входных объектов. Он может быть применен с помощью процедуры среднего пула или процедуры максимального пула. После этого на верхний слой помещается множество полностью связанных слоев и слой SoftMax для классификации и распознавания. Глубокая сверточная нейронная сеть обычно содержит несколько сверточных слоев и слоев субсамплинга для обучения объектов на выборках больших изображений. | -Меньше нейронных связей, чем в стандартной нейронной сети  -Множество вариаций CNN уже реализовано | -Обычно требует большую обучающую выборку размеченных изображений |
| Глубокая сеть доверия (DBN) — это порождающая графическая модель, или, иначе, один из типов глубинных нейронных сетей, состоящая из нескольких скрытых слоев, в которых нейроны внутри одного слоя не связаны друг с другом, но связаны с нейронами соседнего слоя | Возможен послойный подход к обучению модели | Требует больших системных ресурсов во время обучения. |
| Рекуррентные нейронные сети (*RNN*) — вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки. | -Моделирование временных зависимостей  -Возможность запоминать события | Процесс обучения страдает от проблемы исчезающего градиента (большие изменения значений параметров для ранних слоев не оказывает большого влияния на результат) |
| Глубокий автоэнкодер (DAE) представляет собой своего рода DNN. Это сеть позволяющая применять обучение без учителя при использовании метода обратного распространения ошибки. | - Применяется для извлечения признаков / уменьшения размерности.  - Много вариантов реализации | -Требуется фаза подготовки  -Требуется дополнительный анализ полученных результатов |

## Анализ существующих систем обнаружения ВПО

Существует множество работ, описывающих применение статических методов обнаружения ВПО, ниже приведены несколько из них.

Исследователь W. Li [23] реализовал систему идентификации вредоносных программ, использующую метод глубокого обучения, который использует как опасные вызовы API, так и рискованные комбинации разрешений в качестве функций для построения модели DBN, которая может автоматически распознавать вредоносные программы. Результаты их тестов показали, что модель получила точность 90%.

Yi Zhang [24] разработал DeepClassifyDroid на базе CNN. Предлагаемая система выполняет статический анализ для достижения пяти различных характеристик. Система, протестированная на наборе данных, содержит в общей сложности 10 770 приложений, в том числе 5546 вредоносных и 5224 доброкачественных. Результаты показали, что точность подхода составила 97,4%.

Ниже приведены несколько работ по динамическому анализу для выявления ВПО.

H. Liang [25] разработал методы обработки естественного языка для анализа вредоносных программ Android. Они разработали модель, которая рассматривает последовательности системных вызовов как тексты и рассматривает функцию обнаружения вредоносных программ как извлечение темы текста. Предлагаемый подход был протестирован на наборе данных из 14 231 приложения. Результаты показали, что точность была 93,16%.

S. Hou и соавторы [26] предложил автоматическую систему обнаружения вредоносных программ для Android Deep4MalDroid. Представление извлеченных системных вызовов Linux в виде графов и группы автокодировщиков (SAEs) была применена для сканирования общих шаблонов вредоносных программ и, таким образом, для определения новых неизвестных вредоносных приложений. Экспериментальные результаты на наборе данных из 3000 приложений показали, что наилучшая достигнутая точность составила 93,68%.

Ниже рассматриваются системы гибридного анализа.

Z. Yuan разработал Droid-Sec [27], который является первой попыткой применить глубокое обучения в области обнаружения вредоносного ПО для Android. Авторы используют многочисленные функции для создания моделей сетей DBN, которые способны отличить вредоносные программы от доброкачественных. При обучении модели на выборке из 250 вредоносных и 250 доверенных приложений результат обнаружения оказался на уровне 96%.

Z. Yuan и его команда разработали Droid-Detector [28] на базе DBN. Предлагаемый подход был протестирован на большом несбалансированном наборе данных, включающем 20000 образцов доброкачественных и вредоносных программ. Результаты показали хорошую производительность DBN с точностью 96,76%.

R. Vinayakumar и его команда [29] предложили использовать Long Short-Term Memory сети (LSTM), которые представляет собой особый вид рекуррентной нейронной сети. Авторы извлекли динамические и статические характеристики и использовали набор данных из 1738 приложений для оценки производительности модели, которая достигла 93,9% в динамическом анализе и 97,5% в статическом анализе.

Но к сожалению исходные коды рассмотренных решений не были выложены в открытый доступ.

### Рассмотрение подходов к обнаружению на примере DL-Droid

Разработчики DL-Droid, представили систему глубокого обучения для обнаружения вредоносных приложений для Android с помощью динамического анализа с использованием генерации входных данных с учетом предыдущих состояний. В исследовании [30] представлены эксперименты, выполненные с более чем 30 000 приложениями (доброкачественными и вредоносными) на реальных устройствах. Кроме того, исследователями были также проведены эксперименты по сравнению производительности обнаружения и покрытия кода совместно с методом генерации пользовательского ввода с широко используемым подходом без сохранения состояния с использованием системы глубокого обучения. Для создания входных данных для обучения модели динамического анализа разработчики использовали DynaLog.

DynaLog предназначен для автоматического установки большого количества приложений для Android, их последовательного запуска с помощью эмулятора (виртуального устройства Android "AVD") или реального телефона, а также регистрации и извлечения нескольких динамических функций (т. е. вызовов API, действий/событий). С динамическим анализом Android-приложений, генерация тестового пользовательского ввода необходима для того, чтобы обеспечить достаточное покрытие кода, чтобы вызвать вредоносное поведение. DynaLog способен использовать различные методы генерации тестовых входных данных: без учета состояния (случайный ввод) (с помощью разработчиков Monkey tool), и с отслеживанием состояния (с помощью DroidBot), и гибридный (который сочетает в себе инструменты генерации входных данных без состояния и с сохранением состояния Alzaylaee). Фактически, большинство существующих платформ динамического анализа для обнаружения вредоносных программ на Android используют подход без сохранения состояния (основанный на инструменте Monkey tool). Разработчики DL Droid использовали только случайную и генерацию входных данных с учетом состояний.

### Сравнение глубоко обучения с другими методами машинного обучения.

В DL Droid используется бинарный классификатор написанный с использованием H2O, который в настоящее время поддерживает только многослойный персептрон (MLP).  Для оценки эффективности различных классификаторов в системе подсчитывается матрица ошибок и сравнивается производительность между DL Droid и семью популярными подходами машинного обучения. Классификаторы выбранные для сравнения: метод опорных векторов (SVM Linear), машина опорных векторов с ядерными функциями (SVM RBF), Наивный байесовский классификатор (NB), простой логистический классификатор (SL), частичные деревья решений (PART), случайный лес (RF) и дерево решений J48.

TPR –вероятность верно полодительного результата

TNR–вероятность верно отрицательного

FPR–вероятность ложно положительного результата

FNR–вероятность ложно отрицательного результата

P = T P /(T P + F P)

Для сравнения с методами машинного обучения исследователями была выбрана модель персептрона с тремя слоями по 200 нейронов в каждом, как наиболее результативная. Результаты сравнения можно увидеть в таблице 4.

Таблица 4. Результаты сравнения методов машинного обучения.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **TPR** | **TNR** | **FPR** | **FNR** | **P** |
| NB | 0.62 | 0.855 | 0.145 | 0.38 | 0.765 |
| SL | 0.761 | 0.933 | 0.067 | 0.239 | 0.87 |
| SVM Linear | 0.758 | 0.938 | 0.062 | 0.242 | 0.872 |
| SVM RBF | 0.758 | 0.944 | 0.056 | 0.242 | 0.876 |
| J48 | 0.855 | 0.954 | 0.046 | 0.145 | 0.917 |
| PART | 0.861 | 0.955 | 0.045 | 0.139 | 0.92 |
| RF | 0.88 | 0.971 | 0.029 | 0.12 | 0.938 |
| DL(200,200,200) | 0.9776 | 0.9086 | 0.0914 | 0.0224 | 0.9482 |

# ЗАКЛЮЧЕНИЕ

Популярность мобильных устройств Android сделала их легкой мишенью для потенциальных атак. Были рассмотрены подходы к обнаружению ВПО в мобильных устройствах и их реализации различными разработчиками. Наиболее эффективным подходом в большинстве случаем оказывается гибридный анализ приложений с использованием модели глубокого обучения. Но из-за особенностей и сложностей реализации, таких как большое время, необходимое для обучения и проверки приложения становится, недостаток покрытия кода и возможность обнаружения виртуализации со стороны ВПО очень сложно создать гибридный анализатор, пригодный для применения на реальных устройствах.

# СПИСОК ИСТОЧНИКОВ:

1. Dr/Web Ботнеты —Url: <https://www.drweb.ru/pravda/issue/?number=80> — (Дата обращения: 15.01.2020).
2. Анатомия ботнета—Url: <http://www.nestor.minsk.by/kg/2010/02/kg00205.html>— (Дата обращения: 15.01.2020).
3. Ботнеты на основе TOR —Url: https://habr.com/ru/company/eset/blog/187962/ — (Дата обращения: 15.01.2020).
4. Ботнеты: новый характер угроз —Url: <https://www.cisco.com/web/RU/downloads/Botnets.pdf> — (Дата обращения: 15.01.2020).
5. H. Binsalleeh, M. Mannan, A. Youssef, and M. Debbabi, "Detection of malicious payload distribution channels in DNS," —Url: <https://www.researchgate.net/publication/269271799_Detection_of_malicious_payload_distribution_channels_in_DNS> — (Дата обращения: 15.01.2020).
6. Current android malware, Forensics blog, —Url: http://forensics.spreitzenbarth.de/android-malware/ — (Дата обращения: 15.01.2020).
7. C.A. Castillo, Android malware past, present, and future, (McAfee), —Url: <http://www.mcafee.com/us/resources/white-papers/wpandroid-malware-past-present-future.pdf> — (Дата обращения: 15.01.2020).
8. M. Eslahi, R.Salleh and N.B Anuar, MoBots: A New Generation of Botnets on Mobile Devices and Networks, International Symposium on Computer Applications and Industrial Electronics (ISCAIE) —Url: <https://www.researchgate.net/publication/261348553_MoBots_A_new_generation_of_botnets_on_mobile_devices_and_networks> — (Дата обращения: 15.01.2020).
9. Y. Zeng, On detection of current and next-generation botnets —Url: <https://www.researchgate.net/publication/295262190_On_Detection_of_Current_and_Next-Generation_Botnets> — (Дата обращения: 15.01.2020).
10. K. W. Y. Au, Y. F. Zhou, Z. Huang, P. Gill, and D. Lie, Short paper: a look at smartphone permission models —Url: <https://www.researchgate.net/publication/254007208_Short_paper_A_look_at_smartphone_permission_models> — (Дата обращения: 15.01.2020).
11. Github:androguard —Url: https://github.com/androguard/androguard — (Дата обращения: 15.01.2020).
12. H. PIETERSE AND M. S. OLIVIER, Android botnets on the rise: Trends and characteristics, In: Information Security for South Africa —Url:<https://www.researchgate.net/publication/261451446_Android_botnets_on_the_rise_Trends_and_characteristics> — (Дата обращения: 15.01.2020).
13. Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. —Url:<http://www.covert.io/research-papers/security/A%20survey%20on%20automated%20dynamic%20malware-analysis%20techniques%20and%20tools.pdf> — (Дата обращения: 15.01.2020).
14. Enck, W., Ongtang, M., & McDaniel, P. (2009). On lightweight mobile phone application certification. —Url: <https://www.researchgate.net/publication/221609170_On_Lightweight_Mobile_Phone_Application_Certification> — (Дата обращения: 15.01.2020).
15. Yerima, S. Y., Sezer, S., McWilliams, G., & Muttik I. (2013). A new android malware detection approach using Bayesian classification. —Url: <https://arxiv.org/ftp/arxiv/papers/1608/1608.00848.pdf> — (Дата обращения: 15.01.2020).
16. Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., & Weiss, Y. (2012). Andromaly’: A behavioral malware detection framework for android devices. —Url:<https://www.researchgate.net/publication/220616039_Andromaly_A_behavioral_malware_detection_framework_for_android_devices> — (Дата обращения: 15.01.2020).
17. Feizollah, A., Anuar, N. B., Salleh, R., & Wahab, A. W. A. (2015). A review on feature selection in mobile malware detection. —Url: <https://www.researchgate.net/publication/332719349_Neuro-Evolutionary_Feature_Selection_to_Detect_Android_Malware> — (Дата обращения: 15.01.2020).
18. Choi, B., Choi, S. K., & Cho, K. (2013). Detection of mobile botnet using VPN. —Url: <https://www.researchgate.net/publication/261435085_Detection_of_Mobile_Botnet_Using_VPN> — (Дата обращения: 15.01.2020).
19. Yoshimi-Botnet —Url: https://github.com/moloch--/Yoshimi-Botnet— (Дата обращения: 15.01.2020).
20. Android Botnet dataset—Url: <https://www.unb.ca/cic/datasets/android-botnet.html>— (Дата обращения: 15.01.2020).
21. Android Botnet dataset—Url: <https://www.impactcybertrust.org/dataset_view?idDataset=932>— (Дата обращения: 15.01.2020).
22. Collection of android malware samples—Url: <https://github.com/ashishb/android-malware>— (Дата обращения: 15.01.2020).
23. An Android Malware Detection Approach Using Weight-Adjusted Deep Learningм — Url: https://ieeexplore.ieee.org/document/8390391— (Дата обращения: 15.01.2020).
24. A Novel Android Malware Detection Approach Based on Convolutional Neural Network — Url: https://dl.acm.org/doi/10.1145/3199478.3199492— (Дата обращения: 15.01.2020).
25. An end-to-end model for Android malware detection— Url: https://ieeexplore.ieee.org/document/8004891 — (Дата обращения: 15.01.2020).
26. Deep4MalDroid: A Deep Learning Framework for Android Malware Detection Based on Linux Kernel System Call Graphs— Url: https://ieeexplore.ieee.org/document/7814490— (Дата обращения: 15.01.2020).
27. Droid-Sec: Deep Learning in Android Malware Detection— Url: https://www.researchgate.net/publication/266660090\_Droid-Sec\_Deep\_Learning\_in\_Android\_Malware\_Detection — (Дата обращения: 15.01.2020).
28. Droiddetector: Android malware characterization and detection using deep learning— Url: https://ieeexplore.ieee.org/abstract/document/7399288— (Дата обращения: 15.01.2020).
29. Detecting Android malware using Long Short-term Memory (LSTM) — Url: https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs169424 — (Дата обращения: 15.01.2020).
30. DL-Droid: Deep learning based android malware detection using real devices — Url: https://www.sciencedirect.com/science/article/pii/S0167404819300161 — (Дата обращения: 15.01.2020).
31. Common Weaknesses of Android Malware Analysis Frameworks — Url: https://ayeks.de/post/2015-06-16-common-weaknesses-of-android-malware-analysis-frameworks/ — (Дата обращения: 15.01.2020).
32. Accurate mobile malware detection and classification in the cloud — Url: https://www.researchgate.net/publication/282773817\_Accurate\_mobile\_malware\_detection\_and\_classification\_in\_the\_cloud — (Дата обращения: 15.01.2020).