KNOB-attack

Добрый день, уважаемые слушатели. Сегодня я хотел бы рассказать вам о KNOB атаке. Мы рассмотрим, как работает Bluetooth на техническом уровне, какие устройства подвержены KNOB атаке, как происходит атака, а также способы избежать данную атаку.

Слайд 0:

Об атаке

KNOB (Key Negotiation of Bluetooth) Attack – инцидент связан с ошибкой в процессе согласования ключей между устройствами Bluetooth. В спецификациях Bluetooth два устройства, устанавливающие между собой защищенное соединение, могут выбирать длину ключа в пределах от 1 до 16 байт. В случае однобайтового ключа взломать его можно достаточно быстро простым брутфорсом – системным перебором всех возможных комбинаций. Это и делает подобное соединение уязвимым и даёт возможность хакерам получить доступ к различным устройствам.

**Слайд брут форс:**

Brute-force переводится с английского как «грубая сила». Смысл этой атаки в том, чтобы перебрать все возможные варианты, пока один из них не окажется верным. Хакеры не пытаются найти тонкий подход к жертве, как, например, при фишинге, — только перебор, только хардкор.

**Слайд 2:**

Bluetooth BR /EDR - беспроводная технология ближнего действия, широко используемая многими продуктами. Я думаю, большинство из вас с ней знакомы. Bluetooth предоставляет механизмы безопасности для обеспечения аутентификации, конфиденциальности и целостности данных на канальном уровне. Безопасность и конфиденциальность Bluetooth подвергались атакам и исправлялись несколько раз, начиная с версии Blue tooth v1.0. Несколько успешных атак на этапе сопряжения привели к существенным пересмотры стандарта. Однако безопасности протокола согласования ключей шифрования уделялось мало внимания, из-за чего и появилась уязвимость, о которой сегодня пойдет речь. Но для начала давайте разберемся как вообще проходит этап соединения.

**Слайд 3:**

Этапы установки безопасного соединения:

1. Генерация ключа безопасности
2. Аутентификация
3. Безопасный обмен данными

**Слайд 4:**

Для безопасного простого сопряжения двух устройств на первом этапе используется метод эллиптических кривых Диффи-Хеллмана. Эта пара ключей должна быть сгенерирована вначале сопряжения и обычно используется до конца, и первое и второе устройство в любой момент времени могут отказаться от пары сгенерированных ключей и создать новую пару, начав при этом весь процесс сопряжения сначала.

**Слайд 5:**

Сам алгоритм состоит из этапов:

1. Выбираются общие параметры
2. Генерируются публичные ключи
3. Вычисляется секретный ключ К, (который как описано внизу слайда равен у обоих пользователей).

#### **Пример создания секретного ключа**

Допустим, Алиса и Боб хотят создать общий секретный ключ, используя алгоритм Диффи-Хеллмана. Они выполняют следующие шаги:

1. Алиса и Боб выбирают общие параметры: основание **g** (допустим, 5) и большое простое число **p** (допустим, 23).
2. Алиса генерирует свой секретный ключ **a** (допустим, 6) и вычисляет свой публичный ключ A:

A = g^a mod p = 5^6 mod 23 = 15625 mod 23 = 8.

1. Боб генерирует свой секретный ключ **b** (допустим, 9) и вычисляет свой публичный ключ B:

B = g^b mod p = 5^9 mod 23 = 1953125 mod 23 = 11.

1. Алиса и Боб обмениваются публичными ключами: Алиса отправляет свой ключ A (8) Бобу, а Боб отправляет свой ключ B (11) Алисе.
2. Алиса вычисляет общий секретный ключ s:

s = B^a mod p = 11^6 mod 23 = 1771561 mod 23 = 9.

1. Боб вычисляет общий секретный ключ s:

s = A^b mod p = 8^9 mod 23 = 134217728 mod 23 = 9.

Теперь Алиса и Боб имеют общий секретный ключ s, который равен 9. Этот ключ может быть использован для дальнейшего зашифрования и расшифрования сообщений между ними.

**Слайд 6:**

Перед рассмотрением этапа аутентификации рассмотрим понятие Энтропии

Информационная энтропия – это мера неопределённости или непредсказуемости информации, неопределённость появления какого- либо символа алфавита.

Иными словами, сколько вариаций ключа может быть. Чем больше бит ключ занимает, тем больше его вариаций может быть -> тем больше энтропия ключа.

В нашем случае энтропия – количество байт в ключе (Энтропия 1 = 1 байт).

**Слайд 7:**

На данном этапе pairing заканчивается и начинается последний этап инициализации bluetooth — Mutual authentication, или взаимная аутентификация. Основана она на схеме «запрос-ответ».

**Слайд 8:**

Каждый раз, когда для подключения Bluetooth требуется шифрование на канальном уровне, Алиса и Боб вычисляют ключ шифрования KC(его можно назвать ключ сессии) на основе

KL, BT ДОБАВЛЯЕТ, AURAND и ENR И.

KL — это ключ связи, установленный во время безопасного простого

сопряжения, а остальные параметры являются общедоступными. Предполагая идеальную

генерацию случайных чисел, энтропия KC всегда равна 16

байтам.

KC напрямую не используется в качестве ключа шифрования для текущего сеанса. Фактический ключ шифрования, обозначенный символом K’C(ключ шифрования) вычисляется путем уменьшения энтропии KC до N байт.

Данное N согласно спецификации может быть от 1 до 16 байт, на этом этапе и рождается уязвимость.

**Слайд 9-12:**

Шаги атаки

Первые два пакета – аутентификация, на ней генерируется ключ сессии.

Третий и четвёртый – запрос на защищённое соединение

Следующие три – договор о размере ключа

Последний – соединение установлено

Слайд причины:

Недоработка стандарта:

* + Возможность задать энтропию в 1 байт
  + Отсутствие защиты в момент согласования параметров соединения (можно было бы шифровать с помощью одного ключа
  + Непрозрачность процесса соединения для пользователя (можно было бы уведомлять о размере энтропии)

**Слайд идеи:**

**Слайд выводы:**