Bluetooth Tutorial

2008.1.21

본 문서는 Dr GrEeN님의 Blutooth Tutorial을 기반으로 작성된 문서임을 밝힙니다.

rich4rd

『rich4rd.lim@gmail.com』

목차.

1. 소개 및 역사
    1.1 블루투스란 ?
    1.2 블루투스 역사

2. 준비물

3. 기본 명령어 및 실습 준비

4. 공격에 쓰이는 툴들

5. RAW 모드로 변환하기

6. 마무리

1. 소개 및 역사

1.1 블루투스란 ?

블루투스(Bluetooth)는 1994년 에릭슨이 최초로 개발한 개인 근거리 무선 통신(PANs)을 위한 산업 표준이다. 나중에 블루투스 SIG에 의해 정식화되었고, 1999년 5월 20일 공식적으로 발표되었다. 블루투스 SIG에는 소니 에릭슨, IBM, 노키아, 도시바가 참여하였다.

IEEE 802.15.1 규격을 사용하는 블루투스는 PANs(Personal Area Networks)의 산업 표준이다. 블루투스는 다양한 기기들이 안전하고 저렴한 비용으로 전 세계적으로 이용할 수 있는 라디오 주파수를 이용해 서로 통신할 수 있게 한다. 블루투스라는 이름은 10세기의 덴마크 왕 헤럴드 블루투스에서 유래했는데, 대립국면에 있는 파벌들과 협상하는데 있어서 특히 유명했다. 다시 말해, 다른 장치들끼리 통신할 수 있게 하는 이 기술에 적합한 이름이다.

블루투스는 ISM 대역인 2.45GHz를 사용한다. 버전 1.1과 1.2의 경우 속도가 초당 723.1 킬로비트에 달하며, 버전 2.0의 경우 EDR(Enhanced Data Rate)을 특징으로 하는데, 이를 통해 초당 2.1 메가비트의 속도를 낼 수 있다.

블루투스는 유선 USB를 대체하는 개념이며, 와이파이(Wi-Fi)는 이더넷(Ethernet)을 대체하는 개념이다. 암호화에는 SAFER(Secure And Fast Encryption Routine)+을 사용한다. 장치끼리 믿음직한 연결을 성립하려면 키워드를 이용한 페어링(pairng)이 이루어지는데, 이 과정이 없는 경우도 있다.

1.2 블루투스의 역사

블루투스 1.0과 1.0B

1.0과 1.0B는 많은 문제점을 가졌고 다양한 제조사들이 그들 제품끼리 상호 호환성을 가지게 하는 데에 큰 어려움을 겪었다. 1.0과 1.0B는 또한 핸드셰이킹 과정에서 블루투스 하드웨어 장치 주소(BD_ADDR)를 반드시 전송해야 하므로 프로토콜 수준에서의 익명 표현(rendering anonymity)을 할 수 없었는데, 이는 블루투스 환경에서 제공되기로 계획된 소비 확대 정책(Consumerium) 같은 서비스를 제공하는 데에 큰 결점이었다. 블루투스는 2.4Ghz의 주파수를 사용하고 무선 랜 802.11b/g 또한 2.4ghz대의 주파수를 사용한다. 같은 주파수를 사용하니 만큼 동시사용에 따른 충돌은 피할 수 없는데 양 제품의 초기 보급 시에는 그것에 대한 우려의 목소리가 높았다. 하지만 블루투스의 버전업과, 연결 특성상 큰 문제는 발생하지 않았다. 블루투스는 해당 주파수 대역에서 비어있는 채널을 찾아 데이터를 전송하기 때문에 간섭이 일어난다 하더라도 금세 다른 빈 곳으로 전송하게 된다. 음성기기의 사용 시 아주 잠깐의 딜레이가 발생하지만 체감하기 어렵다. 하지만 두 기기의 거리가 1cm 이하로 접근할 경우 간섭이 발생할 가능성이 있다. 무선랜과 블루투스 두 기능을 동시에 가지고 있는 기기의 경우에는 두 장비가 하나의 안테나를 사용하게 되는데, 서로 번갈아가며 데이터를 전송하는 사용하는 방식을 이용해, 애초부터 그 간섭을 최대한 줄이게끔 되어있다.

블루투스 1.1

2002년 802.15.1 IEEE 표준으로 승인 되었고, 1.0B의 많은 문제점들을 수정하였다. 이 외에 비 암호화 채널(non-encrypted channels)을 지원하였고, Signal Strength Indicator (RSSI)를 수신 받을 수 있게 되었다.

블루투스 1.2

이 버전은 1.1버전과 호환이 되며 주요 향상 점은 다음과 같다: 빠른 접속과 가까운 거리에서의 주파수 간섭 및, 먼 거리에서의 분산스펙트럼(frequency-hopping spread spectrum)에 대비하였다. 실제 전송 속도는 1.1과 같은 721kbit/s이다. 패킷의 오류나 재전송에 따른 음성이나 음원신호의 quality손실을 막는 Extended Synchronous Connections (eSCO)를 지원하게 되었고, three-wire UART를 위한 Host Controller Interface (HCI)를 지원하게 되었다. 2005년 802.15.1 IEEE 표준으로 승인 되었다.

블루투스 2.0

2004년 10월에 표준화가 된 이 버전은 1.1과 호환되게 하였다. 주된 향상 점은 3.0Mbit/s의 Enhanced Data Rate (EDR)를 지원하게 된 점이다. 이로써 다음의 효과를 가지게 되었다 : 평균 3배, 최대 10배의 전송속도 향상(실제 전송 속도 2.1Mbit/s)과 Duty Cycle감소에 의한 저 전력 소비, 또한 multi-link scenarios의 단순화로 사용 가능한 대역폭이 증가 되었다. 이론상의 전송 속도는 3.0Mbit/s이고, 실제 data전송 속도는 2.1Mbit/s이다. Special Interest Group (SIG)에 표준화가 된 "Bluetooth 2.0 + EDR"은 많은 업체들이 사용하는 EDR과 표준화되지 않은 "Bluetooth 2.0"를 포함한다. 앞에 명시된 기술을 보여주는 HTC TyTN pocket PC phone과 다른 Bluetooth 2.0 without EDR의 기술은 추가적인 문제점들을 수정한 버전 1.2와 거의 같다. 많은 제품들이 Bluetooth 2.0을 지원한다고 명시하지만 실제로 EDR을 지원하는지는 명확하게 표기하지 않아 문제가 되고 있다.

블루투스 2.1

1.1버전과 완벽하게 호환이 되는 핵심 표준화 버전인 Bluetooth 2.1은 Bluetooth SIG에 의해 2007년 8월1일 제정되었다. 이 기술은 다음의 특징을 가지고 있다: 확장된 inquiry 응답: 접속하기 전에 좀 더 나은 필터링을 위해 inquiry procedure동안 더 많은 정보를 제공한다. 여기서 정보라 함은 장치의 이름, 장치가 지원하는 서비스 목록, 날짜나 시간, 공유정보와 같은 것들을 포함한다. Sniff subrating 기술: 저 전력 모드일 경우, 특히 asymmetric data flows로 연결되어 있을 경우 전력소비를 줄일 수 있다. Human interface devices (HID)장치들이 가장 이익이 될 것으로 예상되는데 최소 3배에서 최대 10배까지 battery의 수명을 증가시킬 수 있다. Encryption Pause Resume: 암호를 재설정 했을 경우, 장치 간에 더욱 강력한 암호화로 최소 23.3시간 이상의 연결을 유지할 수 있다. 안전하고 간편한 공유: 보안의 강화와 사용 시간의 증가로 인해 Bluetooth장치간의 공유 기술이 근본적으로 향상 되었다. 이것은 앞으로의 Bluetooth의 사용에 있어 큰 기여를 할 것으로 기대된다. NFC cooperation: NFC radio interface이 사용가능할 경우, 자동적으로 안전하게 접속할 수 있게 된다. 예를 들어, 수 센티미터 이내로 헤드셋을 NFC를 포함한 Bluetooth 2.1 phone로 가져가기만 하여도 접속할 수 있게 되고 또 다른 예로는, 휴대 전화나 디지털 카메라로 찍은 사진을 디지털 액자에 가깝게 가져가는 것만으로 디지털 사진을 디지털 액자로 업로드 할 수 있게 된다.”

2. 준비물

(1) 블루투스 동글(Bluetooth dongle)

블루투스 장비들과의 통신이 가능하고 더 나아가서 스니핑까지 가능하게 할 수 있는 USB장비입니다.

(2) 블루투스 장비

핸드폰, 해드셋, 컴퓨터 등등 많은 예가 있겠습니다.

(3) Backtrack과 같은 Bluetooth 실습이 가능한 환경

3. 기본 명령어 및 실습 준비

(1) hciconfig을 이용해서 동글을 등록합니다.

```
bt / # hciconfig
hci0:   Type: USB
        BD Address: 00:00:00:00:00:00 ACL MTU: 0:0 SCO MTU: 0:0
        DOWN
        RX bytes:0 acl:0 sco:0 events:0 errors:0
        TX bytes:0 acl:0 sco:0 commands:0 errors:0

bt / # hciconfig hci0 up
bt / # hciconfig
hci0:   Type: USB
        BD Address: 00:15:83:C9:78:3F ACL MTU: 384:8 SCO MTU: 64:8
        UP RUNNING
        RX bytes:85 acl:0 sco:0 events:9 errors:0
        TX bytes:30 acl:0 sco:0 commands:8 errors:0
```

(2) hciconfig -a는 자세한 동글의 기본정보를 제공합니다.

```
bt / # hciconfig -a
hci0:   Type: USB
        BD Address: 00:15:83:C9:78:3F ACL MTU: 384:8 SCO MTU: 64:8
        UP RUNNING
        RX bytes:374 acl:0 sco:0 events:14 errors:0
        TX bytes:51 acl:0 sco:0 commands:10 errors:0
        Features: 0xff 0xff 0x8f 0xfe 0x9b 0xf9 0x00 0x80
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy:
        Link mode: SLAVE ACCEPT
        Name: 'EDRClassone'
        Class: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Ver: 2.0 (0x3) HCI Rev: 0x7a6 LMP Ver: 2.0 (0x3) LMP Subver: 0x7a6
        Manufacturer: Cambridge Silicon Radio (10)
```

(3) hcitool을 이용해서 근처의 블루투스 장비를 검색합니다.

```
bt / # hcitool scan hci0
Scanning ...
        00:12:56:DB:EC:D6
```

※ 제가 테스트에 이용된 장비의 소유주 임을 말씀드립니다.

(4) 해당 블루투스 장비에 대한 기본정보를 획득합니다.

```
bt / # sdptool browse 00:12:56:DB:EC:D6
Browsing 00:12:56:DB:EC:D6 ...
Service RecHandle: 0x10000
Service Class ID List:
  "PnP Information" (0x1200)

Service Name: Y
Service RecHandle: 0x10001
Service Class ID List:
  "Intercom" (0x1110)
  "Generic Telephony" (0x1204)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "TCS-BIN" (0x0005)
    uint16: 0x2df5
Profile Descriptor List:
  "Intercom" (0x1110)
    Version: 0x0100

Service RecHandle: 0x10002
Service Class ID List:
  "Headset Audio Gateway" (0x1112)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
Profile Descriptor List:
  "Headset" (0x1108)
    Version: 0x0100

Service RecHandle: 0x10003
Service Class ID List:
  "Handfree Audio Gateway" (0x111f)
  "Generic Audio" (0x1203)
```

(5) 블루투스 설정

/etc/bluetooth/hcid.conf 파일을 다음과 같이 수정합니다.

```
#
# HCI daemon configuration file.
#

# HCId options
options {
        # Automatically initialize new devices
        autoinit yes;
```

```
        # Security Manager mode
        #   none - Security manager disabled
        #   auto - Use local PIN for incoming connections
        #   user - Always ask user for a PIN
        #
        security auto;

        # Pairing mode
        #   none  - Pairing disabled
        #   multi - Allow pairing with already paired devices
        #   once  - Pair once and deny successive attempts
        pairing multi;

        # Default PIN code for incoming connections
        passkey "1234";
}

# Default settings for HCI devices
device {
        # Local device name
        #   %d - device id
        #   %h - host name
        name "device1";

        # Local device class
        class 0x000000;

        # Default packet type
        #pkt_type DH1,DM1,HV1;
```

```
        # Inquiry and Page scan
        iscan enable; pscan enable;

        # Default link mode
        #   none   - no specific policy
        #   accept - always accept incoming connections
        #   master - become master on incoming connections,
        #            deny role switch on outgoing connections
        lm accept,master;

        # Default link policy
        #   none     - no specific policy
        #   rswitch - allow role switch
        #   hold     - allow hold mode
        #   sniff    - allow sniff mode
        #   park     - allow park mode
        lp rswitch,hold,sniff,park;
        auth enable;
        encrypt enable;
}
```

(6) 블루투스를 재시작 합니다.

```
bt bluetooth # bash /etc/rc.d/rc.bluetooth restart
Stopping Bluetooth subsystem:  pand dund rfcomm hidd sdpd hcid.
Starting Bluetooth subsystem:  hcid passkeys.
```

(7) 해당 블루투스 장비에 대한 rfcomm bind를 설정합니다.

```
bt bluetooth # mknod -m 666 /dev/rfcomm0 c 216 1
bt bluetooth # mknod -m 666 /dev/rfcomm1 c 216 2
bt bluetooth # mknod -m 666 /dev/rfcomm2 c 216 3
```

이렇게 3개의 bind를 생성하였습니다.

(8) sdptool을 이용해서 해당 channel들을 등록합니다.

```
bt bluetooth # sdptool add --channel=1 GENERIC
bt bluetooth # sdptool add --channel=2 HANFREE
bt bluetooth # sdptool add --channel=3 OBEX
bt bluetooth #
```

☆ 이제 블루투스 장비 공격실습을 위한 준비를 끝났습니다. :-)

4. 공격에 쓰이는 툴들

구글링을 하시다 보면 요즘은 블루투스 해킹이 많이 알려진 만큼 생각보다 많은 툴들이 존재함을 알 수 있습니다. 그 중에서 몇가지 툴을 소개하겠습니다. 본 문서에서는 공격시연에 대한 내용은 담지 않았습니다.

(1) Bluesnafer

블루투스 장비의 OBEX Push profile을 통해서 장비에 접속하게 됩니다. 동영상을 보시면 아시겠지만 저장된 번호 확인 및 삭제, 전화걸기 또한 일명 블루재킹이 가능합니다. 악의적인 공격자는 블루재킹을 이용해서 바이러스나 트로잔같은 악성 프로그램을 블루투스 장비에게 보낼 수 있습니다.

※ 참조: http://en.wikipedia.org/wiki/Bluesnarfing

```
bt bluetooth # bluesnarfer
bluesnarfer: you must set bd_addr
bluesnarfer, version 0.1 -
usage: bluesnarfer [options] [ATCMD] -b bt_addr

ATCMD     : valid AT+CMD (GSM EXTENSION)

TYPE      : valid phonebook type ..
example   : "DC" (dialed call list)
            "SM" (SIM phonebook)
            "RC" (recevied call list)
            "XX" much more

-b bdaddr : bluetooth device address
-C chan   : bluetooth rfcomm channel

-c ATCMD  : custom action
-r N-M    : read phonebook entry N to M
-w N-M    : delete phonebook entry N to M
-f name   : search "name" in phonebook address
-s TYPE   : select phonebook memory storage
-l        : list aviable phonebook memory storage
-i        : device info
bt bluetooth #
```

(2) BlueBugger

BlueBugger 또한 Bluesnafer와 비슷하게 접근합니다.

```
bt bluetooth # bluebugger

bluebugger 0.1 ( MaJoMu | www.codito.de )
-----------------------------------------

Usage: bluebugger [OPTIONS] -a <addr> [MODE]

        -a <addr>      = Bluetooth address of target

        Options:
        --------
        -m <name>      = Name to use when connecting (default: '')
        -d <device>    = Device to use (default: '/dev/rfcomm')
        -c <channel>   = Channelto use (default: 17)
        -n             = No device name lookup
        -t <timeout>   = Timeout in seconds for name lookup (default: 5)
        -o <file>      = Write output to <file>

        Mode:
        -----
        info                    = Read Phone Info   (default)
        phonebook               = Read Phonebook    (default)
        messages                = Read SMS Messages (default)
        dial <num>              = Dial number
        ATCMD                   = Custom Command (e.g. '+GMI')

        Note: Modes can be combined, e.g. 'info phonebook +GMI'

* You have to set the target address

bt bluetooth #
```

(3) BlueStab 기법

주로 노키아와 파라소닉폰을 대상으로 많이 실습됐던 BlueStab는 서비스거부 공격 기법입니다.

5. RAW 모드로 변환하기

Cambrige Silicon Radio 칩셋의 USB동글을 FTS4BT 스니핑 동글로 바꿔보겠습니다. bluz 유틸리티로 펌웨어를 수정하게 되는데 다시 기존 모드로 돌아올 수 없음을 말씀드립니다. 기존 펌웨어 백업을 위해서는 dfutool을 이용 합니다.

(1) USB 동글의 칩셋 확인하기

```
bt bluetooth # hciconfig hci* revision
hci0:   Type: USB
        BD Address: 00:15:83:C9:78:6E ACL MTU: 0:0 SCO MTU: 0:0
        HCI 19.2
        Chip version: BlueCore4-ROM
        Max key size: 128 bit
        SCO mapping:  HCI
bt bluetooth #
```

USB동글 칩셋에는 BlueCore-4 Rom과 BlueCorer-4 External이 있습니다. 칩셋이 어느것이던 RAW 모드로 바꾸는데는 상관없습니다. :-)

(2) 동글을 등록하고 USB동글의 특정 값를 수정합니다.

```
bt bluetooth # hciconfig hci0 up
bt bluetooth # bccmd psget -s 0x0000 0x02bf
USB product identifier: 0x0002 (1)
```

어떤 동글을 쓰느냐에 따라 다를 수 있지만 동글에는 byte를 읽기위한 몇 부분들이 있습니다. 일반적으로 "Default" (0x0000), "param" (0x0008), "psi" (0x0001), "psf" (0x0002) 그리고 "psrom" (0x0004)입니다.

(3) 이번엔 Vender ID를 수정합니다.

```
bt bluetooth # bccmd psget -s 0x0000 0x02be
USB vendor identifier: 0x0a12 (2578)
```

(4) 이젠 "psf" (0x0002)를 수정해야 합니다. 새로운 ID를 만든 후 확인합니다.

```
bt bluetooth # bccmd psset -s 0x0000 0x02bf 0x0002
bt bluetooth # bccmd psget -s 0x0000 0x02bf
USB product identifier: 0x0002 (2)
```

(5) 계속적인 hciconfig 명령어를 통해서 스니핑이 가능한 RAW모드임을 확인 할 수 있습니다. (TX와 RX가 계속적으로 증가함을 확인합니다.)

```
bt bluetooth # hciconfig hci0
hci0:   Type: USB
        BD Address: 00:15:83:C9:78:6E ACL MTU: 0:0 SCO MTU: 0:0
        UP RUNNING RAW
        RX bytes:525 acl:0 sco:0 events:0 errors:0
        TX bytes:378 acl:0 sco:0 commands:31 errors:0

bt bluetooth # hciconfig hci0
hci0:   Type: USB
        BD Address: 00:15:83:C9:78:6E ACL MTU: 0:0 SCO MTU: 0:0
        UP RUNNING RAW
        RX bytes:537 acl:0 sco:0 events:0 errors:0
        TX bytes:381 acl:0 sco:0 commands:32 errors:0

bt bluetooth # hciconfig hci0
hci0:   Type: USB
        BD Address: 00:15:83:C9:78:6E ACL MTU: 0:0 SCO MTU: 0:0
        UP RUNNING RAW
        RX bytes:549 acl:0 sco:0 events:0 errors:0
        TX bytes:384 acl:0 sco:0 commands:33 errors:0

bt bluetooth # hciconfig hci0
hci0:   Type: USB
        BD Address: 00:15:83:C9:78:6E ACL MTU: 0:0 SCO MTU: 0:0
        UP RUNNING RAW
        RX bytes:561 acl:0 sco:0 events:0 errors:0
        TX bytes:387 acl:0 sco:0 commands:34 errors:0
```

6. 마무리

블루투스는 현재도 많이 취약한 것으로 알고 있습니다. 물론 블루투스가 대중화되서 편하게 통화할 수 있는 핸드폰이라든지 우리 삶을 좀 더 윤택하게 만들어 주고 있습니다. 하지만 앞으로 다소 어려울 수는 있겠으나 꼭 빠른 개발만이 아닌 보안까지 신경써서 새로운 제품 혹은 기술이 탄생하기를 바랍니다. 해킹시연에 대해서는 부족한 문서이지만 조금이나마 도움이 되시기를 바랍니다. 읽어주셔서 감사합니다.