Slide 1

“I Can Hear You Now”
Eavesdropping on Bluetooth Headsets

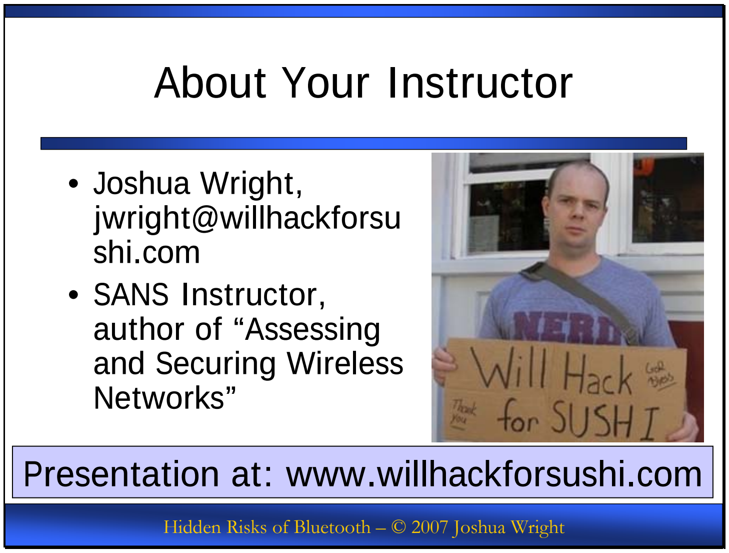
Joshua Wright
jwright@willhackforsushi.com

Welcome to this session titled “I Can Hear You Now: Eavesdropping on Bluetooth Headsets”. My name is Joshua Wright, and I'm the author of this material, as well as a fervent wireless security researcher and analyst. For the past several years, I've been studying the growth of Bluetooth technology and use, with a careful eye toward the risks that Bluetooth can expose us to.

This presentation will examine the risks associated with Bluetooth technology, demonstrating some new attacks against Bluetooth technology. During this presentation, it is my goal to help you understand the risks associated with Bluetooth technology and wireless headsets.

During this presentation, feel free to ask any questions you may have, though I reserve the right to defer questions if it looks like we won't be able to make it through all the material in the time allotted. If you prefer to email me with any questions, please feel free to do so by contacting me at jwright@arubanetworks.com.

Slide 2



About Your Instructor

- Joshua Wright, jwright@willhackforsushi.com
- SANS Instructor, author of “Assessing and Securing Wireless Networks”

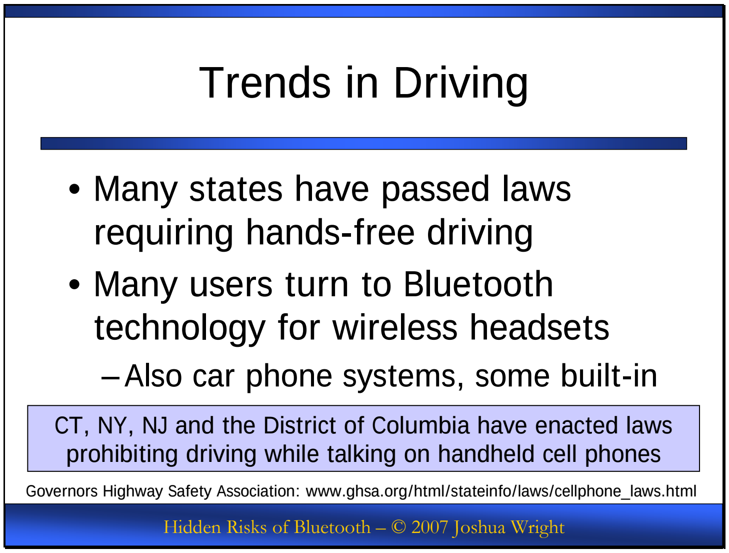
NERD
Will Hack for SUSHI
God Bless
Thank you

Presentation at: www.willhackforsushi.com

About Your Instructor

My name is Joshua Wright, and I’ll be presenting today on an attack I’ve been developing against Bluetooth headsets. I’m a SANS instructor, and the author of the SANS Institute Assessing and Securing Wireless Networks course, where we spend 5 days investigating wireless threats against IEEE 802.11, WiMAX and Bluetooth networks, followed by a step-by-step day of designing and deploying a secure wireless infrastructure. Throughout the class, students will leverage the SANS Wireless Auditing Toolkit (SWAT) for lab exercises, which includes a high-gain wireless card and panel antenna, GPS and accompanying software on a Linux bootable CD.

Slide 3



Trends in Driving

- Many states have passed laws requiring hands-free driving
- Many users turn to Bluetooth technology for wireless headsets
  - Also car phone systems, some built-in

CT, NY, NJ and the District of Columbia have enacted laws prohibiting driving while talking on handheld cell phones

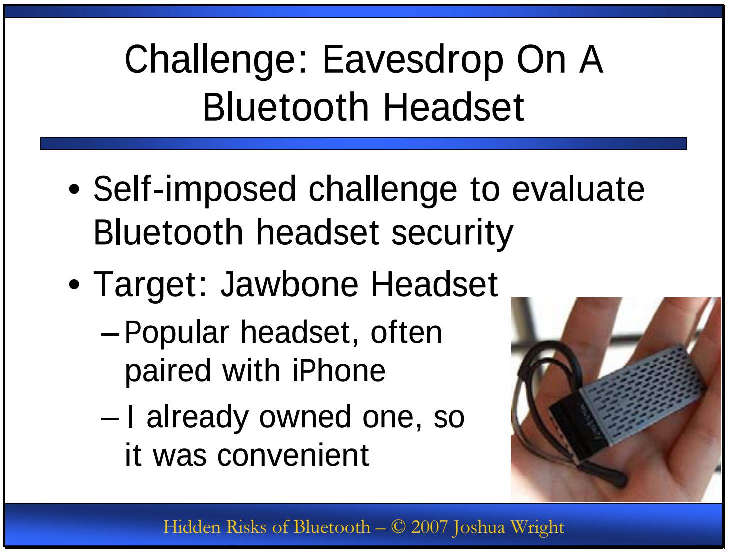
Governors Highway Safety Association: www.ghsa.org/html/stateinfo/laws/cellphone_laws.html

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Trends in Driving

As the popularity of mobile phones has risen, many states have passed legislation regarding the use of mobile phones while driving. Several states, including Connecticut, New York, New Jersey and the District of Columbia have passed laws that prohibit the use handheld phones while driving. This activity, and otherwise just good driving sense, have prompted many consumers to turn to Bluetooth hands-free devices for in-car use. In a slightly smaller portion of the population, Bluetooth headsets have even turned into a fashion item, with designer headsets growing in popularity, with customizable options such as color options.

Slide 4



Challenge: Eavesdrop On A Bluetooth Headset

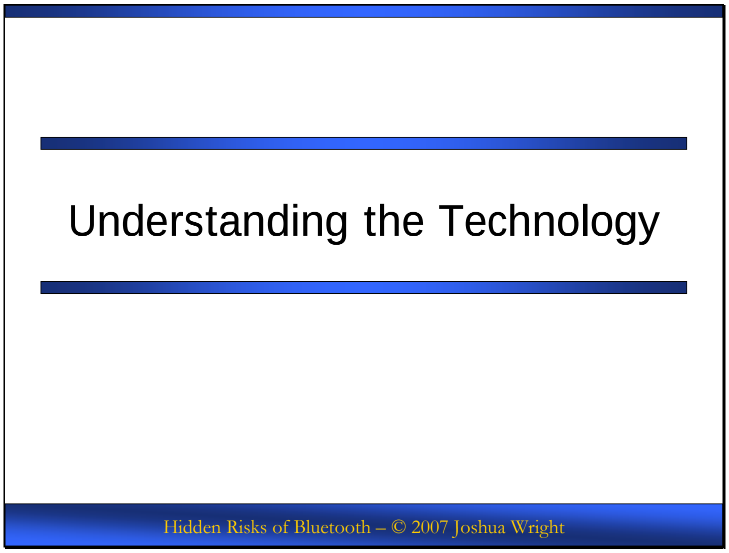
- Self-imposed challenge to evaluate Bluetooth headset security
- Target: Jawbone Headset
  - Popular headset, often paired with iPhone
  - I already owned one, so it was convenient

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Challenge: Eavesdrop On A Bluetooth Headset

As part of a self-imposed challenge, I wanted to evaluate weaknesses in Bluetooth headset devices. My selected target is the popular Jawbone headset. This particular headset is often sold with the Apple iPhone in AT&T/Cingular stores or online.
The Jawbone headset has two buttons for controlling the device, both of which are “hidden” for esthetic reasons. The device operates like many Bluetooth headset devices, where the device is by default in non-discoverable mode with a fixed PIN of “0000” that cannot be changed.

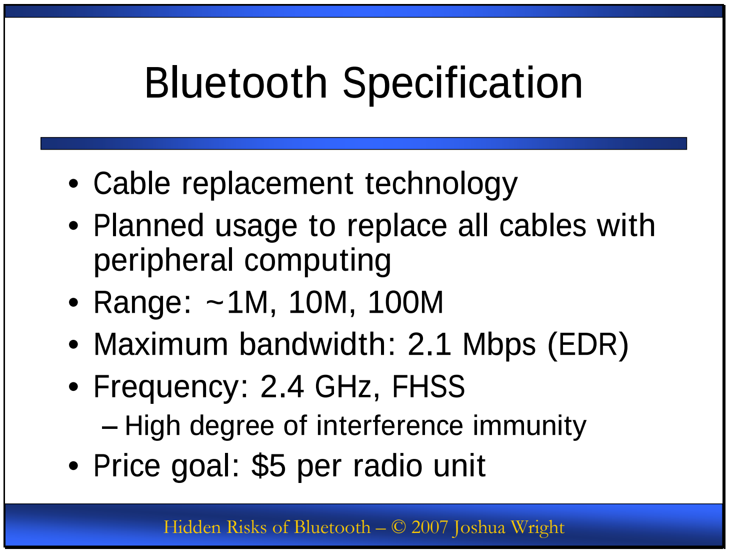
Slide 5



Understanding the Technology

This page left intentionally blank.

Slide 6



Bluetooth Specification

- Cable replacement technology
- Planned usage to replace all cables with peripheral computing
- Range: ~1M, 10M, 100M
- Maximum bandwidth: 2.1 Mbps (EDR)
- Frequency: 2.4 GHz, FHSS
  - High degree of interference immunity
- Price goal: $5 per radio unit

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Bluetooth Specification

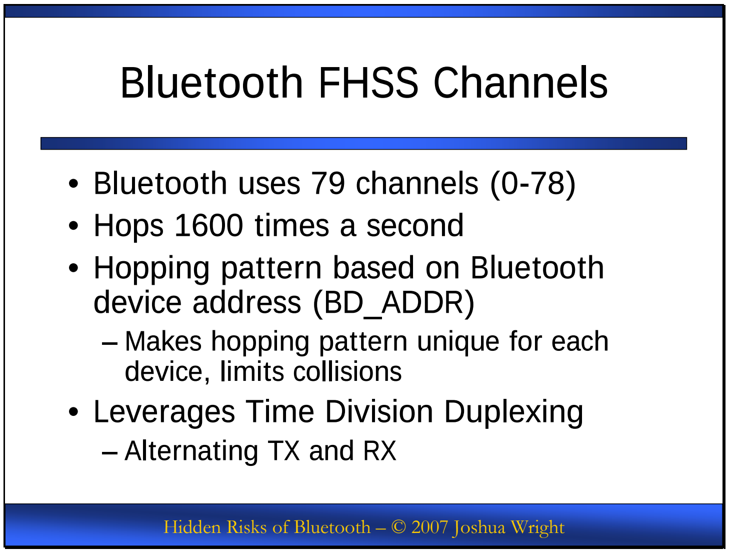
Simple stated, Bluetooth is designed as a cable replacement technology. I'm sure we're all familiar with the heartache of traveling to a location, getting our your gear only to discover that you're missing some obscure cable to connect A to B, or perhaps the attractive cluster of cables behind your desk connecting all of our peripheral devices together. Bluetooth is designed to place all peripheral cabling, designed to emulate existing cabling systems (such as serial devices, or network connections).

Bluetooth transmitters come in three varieties; Class 1, 2 and 3 devices. A Class 3 device is designed to transmit at a range of approximately 1 meter with a transmit power of 1mW (0 dBm). Class 2 devices transmit at a range of 10 meters with a transmit power of 2.5 mW (4 dBm). Class 1 devices are the most powerful transmitters with a range of 100 meters and a transmit power of 100 mW (20 dBm), rivaling the transmit power and distance of IEEE 802.11b transmitters.
Class 2 devices are the most popular variety of Bluetooth transmitter in phones and headsets providing a useful mix of low-power consumption and useful range. Class 1 devices are very popular as USB dongles for laptops and other peripherals.

The maximum bandwidth of a Bluetooth Enhanced Data Rate (EDR) dongle is a little over 2 Mbps, which isn't fast by modern WLAN networking standards, but is suitable for most cable replacement needs. Using the 2.4 GHz spectrum, Bluetooth devices implement Frequency Hopping Spread Spectrum (FHSS) by rapidly hopping through a large channel set. With channel hopping, Bluetooth devices have a high level of interference immunity, since they are only subject to a narrow-band jammer when they hop in the jammed frequency range; later frequency hops are error-free and unaffected by the jammer.

An important consideration in the evaluation of Bluetooth security is that the pricing goal of a Bluetooth radio was set at $5/USD. This goal is important for widespread adoption of Bluetooth technology, but limits the options available to Bluetooth engineers for strong cryptography and other security mechanisms.

Slide 7



Bluetooth FHSS Channels

- Bluetooth uses 79 channels (0-78)
- Hops 1600 times a second
- Hopping pattern based on Bluetooth device address (BD_ADDR)
  - Makes hopping pattern unique for each device, limits collisions
- Leverages Time Division Duplexing
  - Alternating TX and RX

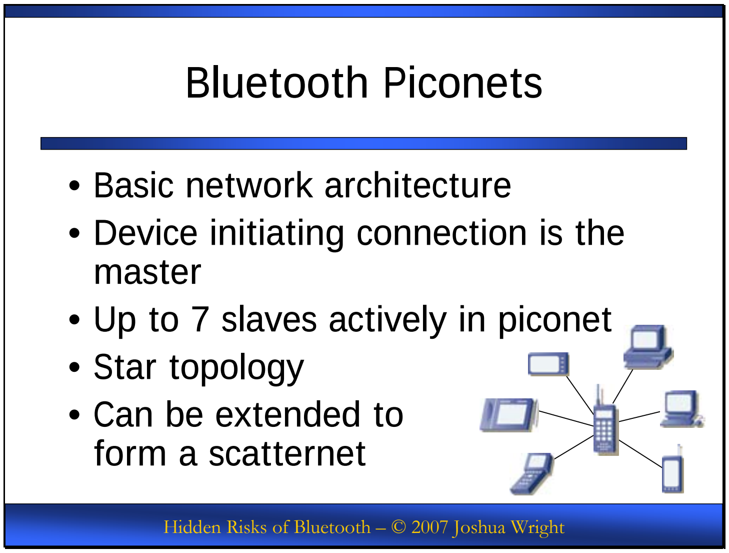
Hidden Risks of Bluetooth – © 2007 Joshua Wright

Bluetooth FHSS Channels

A Bluetooth transmitter uses Frequency Hopping Spread Spectrum to channel hop in a set of 79 channels, ranging from 2.402 GHz to 2.480 GHz. Two devices communicating leverage a Time Division Duplexing strategy, where each device takes a turn transmitting and receiving traffic between channel hops.

Bluetooth networks implement FHSS with very a very rapid channel hopping strategy, where devices channel hop at 1600 hops a second under normal circumstances, and as fast as 3200 hops a second when initially connecting. In order to avoid collisions with other Bluetooth transmitters in the same area, Bluetooth uses pseudo-random generation algorithm to identify the frequency hopping pattern between two devices. The hopping pattern is based on the Bluetooth Device Address information, a unique identifier that is assigned to every Bluetooth transmitter.

Slide 8



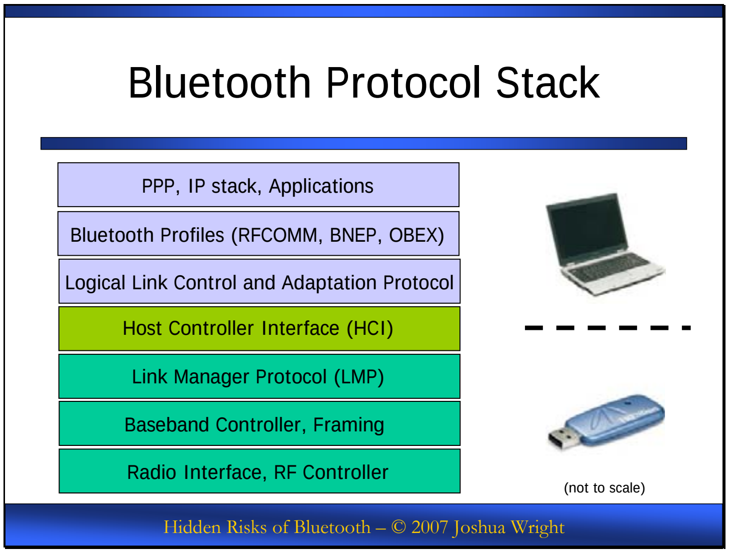
Bluetooth Piconets

- Basic network architecture
- Device initiating connection is the master
- Up to 7 slaves actively in piconet
- Star topology
- Can be extended to form a scatternet

Bluetooth Piconet

Bluetooth networks are not limited to one-to-one communication; a groups of Bluetooth devices can come together to form a piconet of up to 7 devices. In each piconet there is one device classified as the master which is responsible for network synchronization and authorization. The FHSS channel hopping pattern is derived from the master's BD_ADDR information.
All other devices in the piconet are considered slave devices. In order to be certified as a Bluetooth device, manufacturers must implement devices to accommodate the role of either master or slave, such that any device can initiate the piconet or join the piconet. Bluetooth piconets can be extended as well, where any device that is participating as a slave can be the master of another piconet at the same time, linking the two piconets together and forming a scatternet.

Slide 9



Bluetooth Protocol Stack

PPP, IP stack, Applications

Bluetooth Profiles (RFCOMM, BNEP, OBEX)

Logical Link Control and Adaptation Protocol

Host Controller Interface (HCI)

Link Manager Protocol (LMP)

Baseband Controller, Framing

Radio Interface, RF Controller

(not to scale)

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Bluetooth Protocol Stack

In order to accommodate the relatively low cost of a Bluetooth radio, the Bluetooth protocol stack is designed in a layered approach. The layered approach allows different portions of the protocol stack to be designed independently and tightly integrated into hardware, which is important for lightweight devices such as headsets that have limited memory availability.

At the bottom of the stack if the RF controller or the radio interface. This layer handled functionality including channel changing, synchronization of receive and transmit functions, and data modulation and demodulation.
The baseband controller is responsible for assembling the Bluetooth packet header information, and for applying error checking and data whitening (removing DC bias) functions.
The link manager protocol (LMP) is responsible for establishing and tearing down a piconet, as well as the discovery and enumeration of remote Bluetooth devices. The LMP layer is also responsible for negotiating security such as device authentication and link encryption.

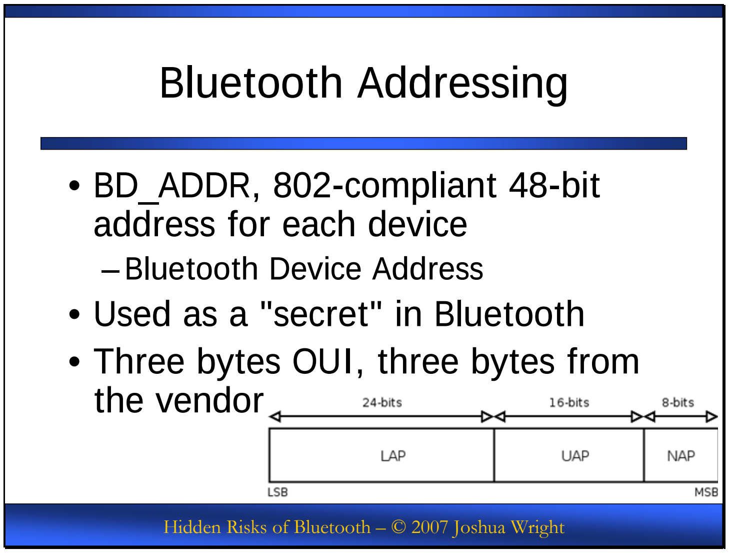
In standard Bluetooth hardware, the lower three layers are implemented in firmware or in hardware, and are generally not accessible to Bluetooth developers. Instead, Bluetooth developers interact with the host controller layer (HCI) which resides at a boundary between the host operating system (such as Windows or Linux) and the Bluetooth hardware. The HCI layer abstracts Bluetooth functionality such as establishing a connection, or probing for available devices.

Above the HCI layer is the Logical link Control and Adaptation Protocol (L2CAP) is an abstraction layer above the HCI and LMP layers, handling application functionality for upper-layer protocols. L2CAP is responsible for managing connectivity between multiple

applications using the same Bluetooth interface simultaneously, as well as packet fragmenting and reassembly and QoS functions.
Above the L2CAP layer are Bluetooth profiles. The Bluetooth profiles implement useful functions for Bluetooth devices, including the ability to emulate a serial connection (Radio Frequency Communication/RFCOMM), provide network connectivity between multiple Bluetooth devices (Bluetooth Network Encapsulation Protocol/BNEP) and arbitrary file exchanges (Object Exchange Protocol/OBEX).

Finally, above the Bluetooth profiles layer, the standard operating system functions are used. For example, the IP stack may interact with the BNEP profile to access the LAN over Bluetooth, or the PPP process may use the RFCOMM profile as a virtual serial port for a simulated dial-up connection.

Slide 10



Bluetooth Addressing

- BD_ADDR, 802-compliant 48-bit address for each device
  - Bluetooth Device Address
- Used as a "secret" in Bluetooth
- Three bytes OUI, three bytes from the vendor

Hidden Risks of Bluetooth – © 2007 Joshua Wright

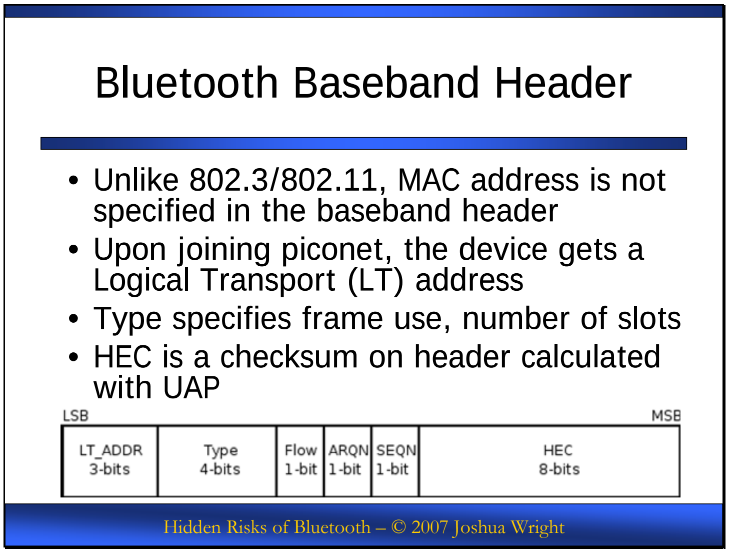
Bluetooth Addressing

We've discussed how every Bluetooth device has its own BD_ADDR, or Bluetooth Device Address. The BD_ADDR information is an IEEE 802-compliant, 48-bit MAC address that is allocated by the manufacturer to the Bluetooth device. Like standard IEEE 802 addresses, this address is made up of the organizationally unique identifier (OUI) portion of the address which is allocated to the vendor, and three additional bytes that are allocated to devices by the manufacturer.

In the Bluetooth specification, the BD_ADDR information is divided into three components:
LAP; The Lower Address Part or LAP is the last three bytes of the BD_ADDR. The LAP represents the MAC address bytes allocated by the vendor to the device.
UAP; Upper Address Part or UAP is the last byte of the OUI allocated to the vendor.
NAP; Non-significant Address Part or NAP is the first two bytes of the OUI allocated to the vendor.

In Bluetooth networks, the BD_ADDR information is treated as a secret. In order to connect to the piconet, the slave must have knowledge of the BD_ADDR of the master; if the BD_ADDR is not known, the slave cannot connect to the piconet.

Slide 11



Bluetooth Baseband Header

- Unlike 802.3/802.11, MAC address is not specified in the baseband header
- Upon joining piconet, the device gets a Logical Transport (LT) address
- Type specifies frame use, number of slots
- HEC is a checksum on header calculated with UAP

LSB MSB

LT_ADDR 3-bits	Type 4-bits	Flow 1-bit	ARQN 1-bit	SEQN 1-bit	HEC 8-bits

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Bluetooth Baseband Header

In the standard Ethernet and WLAN frame headers, the MAC addresses of the source and destination devices are present, allowing anyone who captures this information to be able to identify the transmitter and the receiver. In a Bluetooth frame, the MAC address (BD_ADDR) information is not present since the transmission of two 48-bit MAC addresses for each frame is considered too much overhead for Bluetooth. Since Bluetooth is a TDD architecture, there is no need for specifying a source and destination address, since traffic is only transmitted between the master and one or more slave devices.

Instead of using the entire MAC address in the baseband header, Bluetooth devices are allocated a Logical Transport address (LT_ADDR) when they connect to the piconet. The LT_ADDR is the first three bits in the Bluetooth baseband header, and is used to identify the slave device that should process the frame information from the master device. Because this field is only three bits in length, it can only represent 8 unique devices addresses; the LT_ADDR zero is reserved for frames being sent to the broadcast address. Since the master does not have a LT_ADDR, seven other values are possible, representing the maximum number of slave devices in the piconet.

The type field in the baseband header represents the type of packet being transmitted. The type field can be used to identify asynchronous traffic, synchronous traffic, or special management frames.

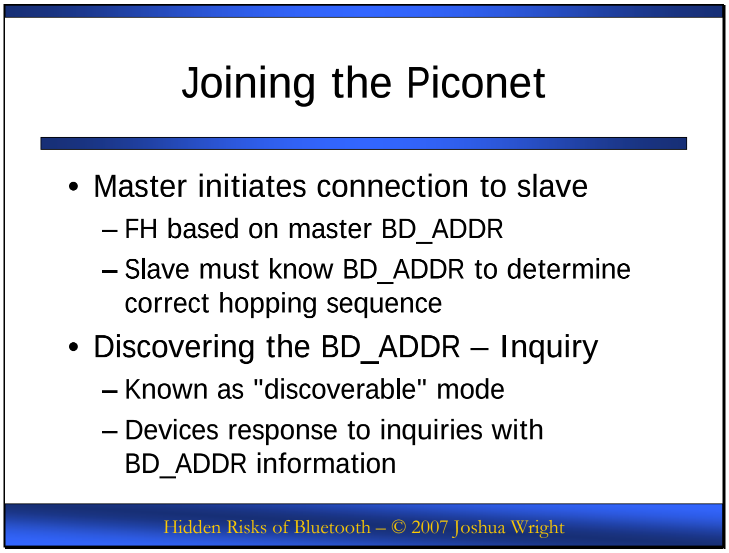
The flow field is used to implement flow controls with the transmission of real-time data where the data recipient can stop the transmission of data by sending a frame with the flow bit set.

The ARQN field implements the Automatic Repeat reQuest Number acknowledgement mechanism, where the receiver can send a frame to positively or negatively acknowledge a frame with checksum information.

The SEQN field is a simple flip-flop bit for ordering the packet stream. Each frame that is transmitted flips the SEQN bit before the next packet is transmitted.

The HEC field is the header error correction checksum. The HEC is calculated over the baseband header contents to ensure it was not accidentally corrupted in transit.

Slide 12



Joining the Piconet

- Master initiates connection to slave
  - FH based on master BD_ADDR
  - Slave must know BD_ADDR to determine correct hopping sequence
- Discovering the BD_ADDR – Inquiry
  - Known as "discoverable" mode
  - Devices response to inquiries with BD_ADDR information

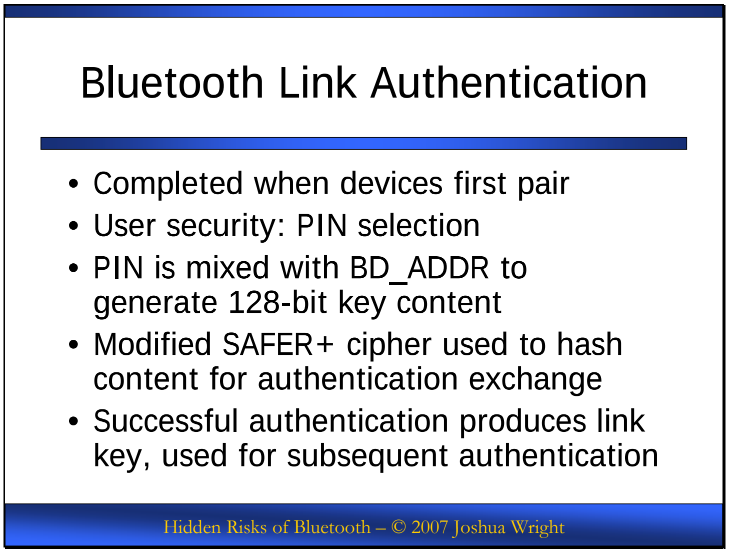
Hidden Risks of Bluetooth – © 2007 Joshua Wright

Joining the Piconet

When two Bluetooth devices wish to communicate, a piconet is formed. The device that initiates the connection to another device is elected the master of the piconet network, and is responsible for managing the piconet and any security practices used in the network. The master and slave devices use the BD_ADDR of the master device to generate the frequency hopping pattern and start hopping to communicate with each other.

Since the BD_ADDR of the master is required to identify the hopping pattern, knowing this address is mandatory for a device that wishes to participate in the piconet. A device manufacturer could accommodate a human-interface device (HID) to allow users to manually enter the BD_ADDR information on the slave, but this is not useful for devices without HID interfaces. As an alternate communication mechanism, Bluetooth includes a feature where a device can probe other devices for their BD_ADDR information. Known as inquiry mode or discoverable mode, a device that is not currently participating in the piconet can probe for other Bluetooth devices in the area and learn their BD_ADDR information in the process.

Slide 13



Bluetooth Link Authentication

- Completed when devices first pair
- User security: PIN selection
- PIN is mixed with BD_ADDR to generate 128-bit key content
- Modified SAFER+ cipher used to hash content for authentication exchange
- Successful authentication produces link key, used for subsequent authentication

Hidden Risks of Bluetooth – © 2007 Joshua Wright

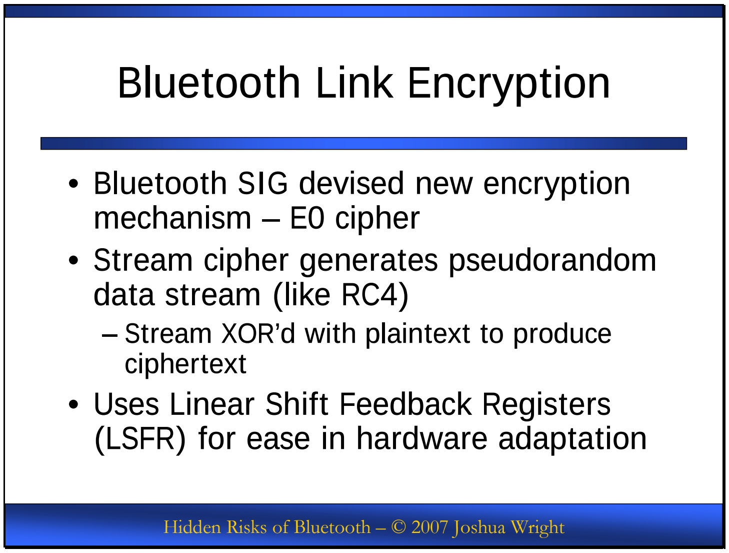
Bluetooth Link Authentication

When security mode 3 is in use, Bluetooth encrypts all traffic before transmitting it over the air, and authenticates the identity of the slave device to the master. For practical implementations, the authentication component is based on a PIN value that is hard-coded into the device, or is selected by the user.

When two devices pair for the first time, the PIN is used to authenticate the user, and is then used with the BD_ADDR of the master device to generate a 128-bit encryption key known as the link key. The PIN is not transmitted in plaintext across the Bluetooth connection, rather, it is protected using a classic challenge/response protocol that leverages a modified version of the SAFER+ cipher.

After authentication, the link key is stored on both devices and is used for subsequent connections to authenticate both parties. This is beneficial to Bluetooth security, since the PIN is only ever used in the authentication process during the initial pairing; later connections use the link key for authentication.

Slide 14



Bluetooth Link Encryption

- Bluetooth SIG devised new encryption mechanism – E0 cipher
- Stream cipher generates pseudorandom data stream (like RC4)
  - Stream XOR'd with plaintext to produce ciphertext
- Uses Linear Shift Feedback Registers (LSFR) for ease in hardware adaptation

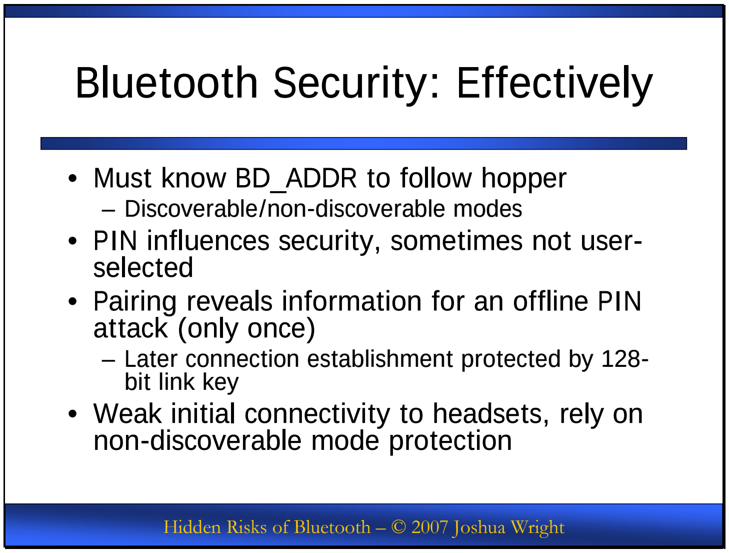
Hidden Risks of Bluetooth – © 2007 Joshua Wright

Bluetooth Link Encryption

Due to the demands for inexpensive, lightweight Bluetooth devices, the Bluetooth Special Interest Group (SIG) designed their own encryption mechanism known as the E0 cipher for encrypting Bluetooth traffic with mode 3 security. E0 is a stream cipher like the RC4 cipher (RC4 is also used in the TLS, WEP and TKIP protocols), generating a pseudorandom data stream (known as the pseudo-random generation algorithm or PRGA) that is XOR'd with the plaintext data to produce ciphertext. This has the advantage of being fast and simple to implement, since the decryption routine is the same as the encryption routine; when decrypting, the ciphertext is XOR'd with the PRGA to produce plaintext.

The E0 cipher was selected and implemented to be simple to offload into hardware. E0 is based on a linear shift feedback register (LSFR), which can be easily implemented in inexpensive hardware. This is as opposed to a more complex cipher like the Advanced Encryption System which requires significant processing capabilities to implement.

Slide 15



Bluetooth Security: Effectively

- Must know BD_ADDR to follow hopper
  - Discoverable/non-discoverable modes
- PIN influences security, sometimes not user-selected
- Pairing reveals information for an offline PIN attack (only once)
  - Later connection establishment protected by 128-bit link key
- Weak initial connectivity to headsets, rely on non-discoverable mode protection

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Bluetooth Security: Effectively

Before we finish this part of the module, let's summarize some of the Bluetooth security functions that are available today:

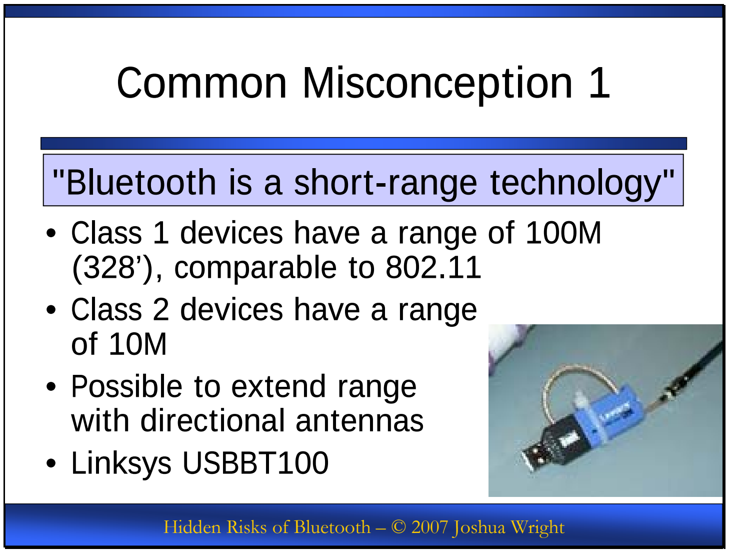
Frequency Hopping makes sniffing difficult, since the sniffer must be able to hop along in synchronization with the transmitters.
In order to know the frequency hopping pattern, devices must know the BD_ADDR information. If the master device is in discoverable mode, it can be queried by an unauthenticated device for the BD_ADDR information.
The selection of a PIN influences the security of the network, but the selection of the PIN is not always possible without a man-machine interface.
During the initial pairing exchange, the PIN is vulnerable to an offline brute-force PIN enumeration attack. This is only a vulnerability the first time two devices pair however. Successive connections between the devices leverage a previously established and stored 128-bit link key and does not utilize PIN information.
Bluetooth headset devices often have a default PIN of “0000” or “1234”. The significant feature that makes these devices more resilient to attack is the fact that they are deployed in non-discoverable mode. Since an attacker does not know the BD_ADDR of the device, they are unable to establish a connection.

Slide 16

Common Bluetooth Misconceptions

This page left intentionally blank.

Slide 17



Common Misconception 1

"Bluetooth is a short-range technology"

- Class 1 devices have a range of 100M (328'), comparable to 802.11
- Class 2 devices have a range of 10M
- Possible to extend range with directional antennas
- Linksys USBBT100

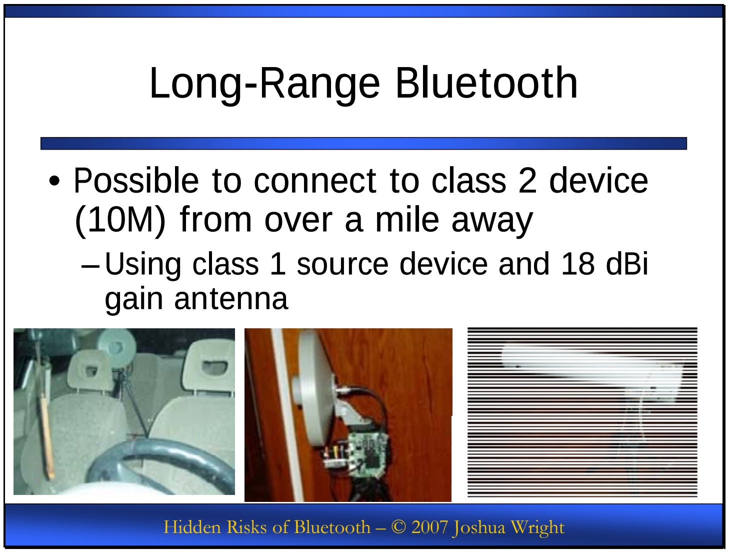
Hidden Risks of Bluetooth – © 2007 Joshua Wright

Common Misconception 1

"Bluetooth is a short-range technology"

Many organizations disregard the security of Bluetooth networks as a concern because they consider Bluetooth to be a short-range technology. Bluetooth technology is not limited to short-range connections however, with class 1 Bluetooth devices transmitting at 100mW, which is approximately 100 meters or 328 feet, comparable to the range of an 802.11b WLAN device. Class 1 devices are most commonly implemented in devices where power is plentiful, such as laptop and desktop systems.
In contrast, class 2 devices transmit at 2.5 mW with a range of approximately 10 meters or 32 feet. Class 2 devices are the most common Bluetooth transmitters for their fair range with less power requirements than class 1 devices. Most mobile phones and Bluetooth headsets are class 2 devices.
Since Bluetooth devices operate in the 2.4 GHz spectrum, they use the same commodity antennas designed for WLAN devices. While vendors don't design Bluetooth dongles with external antenna connectors, some Bluetooth dongles such as the Linksys USBBT100 can be modified to accommodate an external antenna connector.

Slide 18



Long-Range Bluetooth

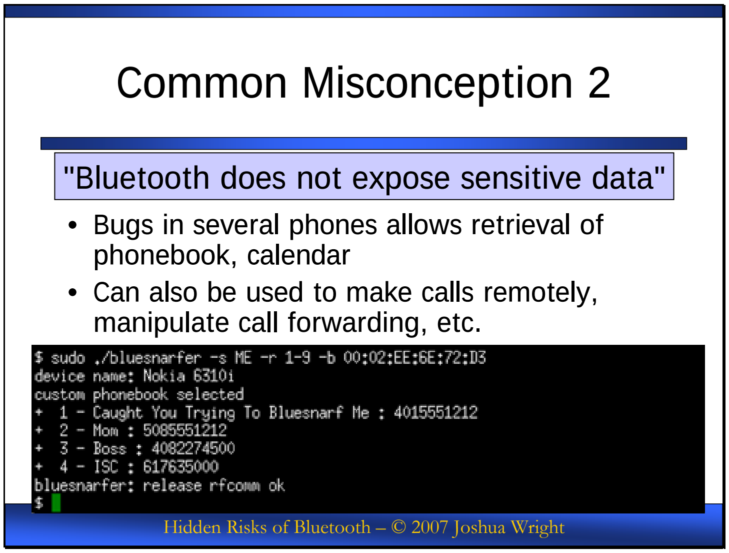
- Possible to connect to class 2 device (10M) from over a mile away
  - Using class 1 source device and 18 dBi gain antenna

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Long-Range Bluetooth

Armed with a class 1 dongle and a high-gain 2.4 GHz antenna, it is possible for an attacker to connect to a class 2 device (designed for a range of 10 meters) from a distance of over a mile. This allows an attacker to exploit even short-range Bluetooth devices from a significant distance.

Slide 19



Common Misconception 2

"Bluetooth does not expose sensitive data"

- Bugs in several phones allows retrieval of phonebook, calendar
- Can also be used to make calls remotely, manipulate call forwarding, etc.

```
$ sudo ./bluesnarfer -s ME -r 1-9 -b 00:02:EE:6E:72:D3
device name: Nokia 6310i
custom phonebook selected
+  1 - Caught You Trying To Bluesnarf Me : 4015551212
+  2 - Mom : 5085551212
+  3 - Boss : 4082274500
+  4 - ISC : 617635000
bluesnarfer: release rfcomm ok
$
```

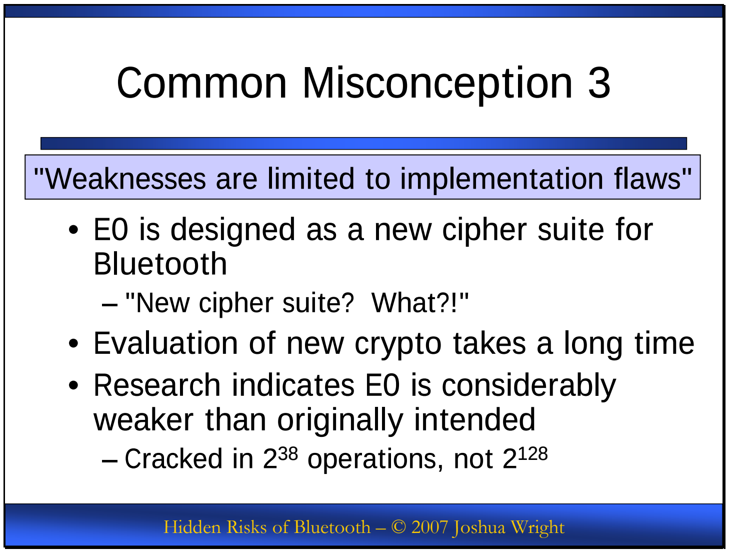
Common Misconception 2

"Bluetooth does not expose sensitive data"

Another misconception about Bluetooth networks is that they don't represent a mechanism to expose any sensitive data. Consider the attack known as the BlueSnarfing attack. Targeting several popular Nokia and Ericsson phones, the BlueSnarfing attack leverages a flaw where phones expose the RFCOMM profile on an undocumented service that allows an attacker to connect to the device without authentication. Using the serial connectivity provided by the RFCOMM profile, an attacker can execute arbitrary AT commands to manipulate the remote device, including the ability to retrieve, modify and delete phonebook and calendar entries.

The bluesnarfer tool implements this attack, where the attacker can specify a remote phonebook (stored numbers, recent outgoing calls, recent incoming calls, etc) and retrieve, modify or delete the results. If the attacker connects to the RFCOMM service manually with a terminal emulator tool (either on Windows or Linux systems), they can enter manual AT commands, such as initiating a call ("ATDT911"), forwarding all calls to a specified number ("AT+CCF911") or redial the last number called ("ATDL"). Even more potentially useful information is available for the attacker, including the Electronic Serial Number of the phone ("AT+CGSN").

Slide 20



Common Misconception 3

"Weaknesses are limited to implementation flaws"

- E0 is designed as a new cipher suite for Bluetooth
  - "New cipher suite? What?!"
- Evaluation of new crypto takes a long time
- Research indicates E0 is considerably weaker than originally intended
  - Cracked in 2^{38} operations, not 2^{128}

Hidden Risks of Bluetooth – © 2007 Joshua Wright

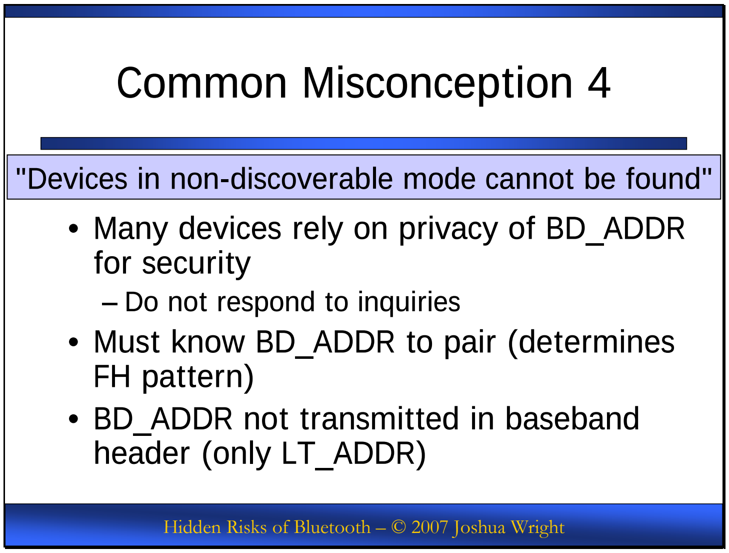
Common Misconception 3

"Weaknesses are limited to implementation flaws"

In the design of the Bluetooth specification, the Bluetooth SIG invented their own encryption mechanism, known as the E0 cipher. It is generally frowned upon in the cryptography community when someone invents their own encryption mechanism, since it can take many years to fully understand the implications of the cipher and potential weaknesses. Recent research into the E0 cipher from the LASEC Security and Cryptography Labs has revealed that while E0 was designed to provide 128-bit security levels, it has sufficient weaknesses such that it can be compromised with 2^{38} operations, instead of 2^{128}.

The research paper highlighting this weakness in the E0 cipher is available at http://lasecwww.epfl.ch/pub/lasec/doc/LMV05.pdf, with presentation slides from the International Association for Cryptologic Research available at http://www.iacr.org/conferences/crypto2005/p/16.pdf.

Slide 21



Common Misconception 4

"Devices in non-discoverable mode cannot be found"

- Many devices rely on privacy of BD_ADDR for security
  - Do not respond to inquiries
- Must know BD_ADDR to pair (determines FH pattern)
- BD_ADDR not transmitted in baseband header (only LT_ADDR)

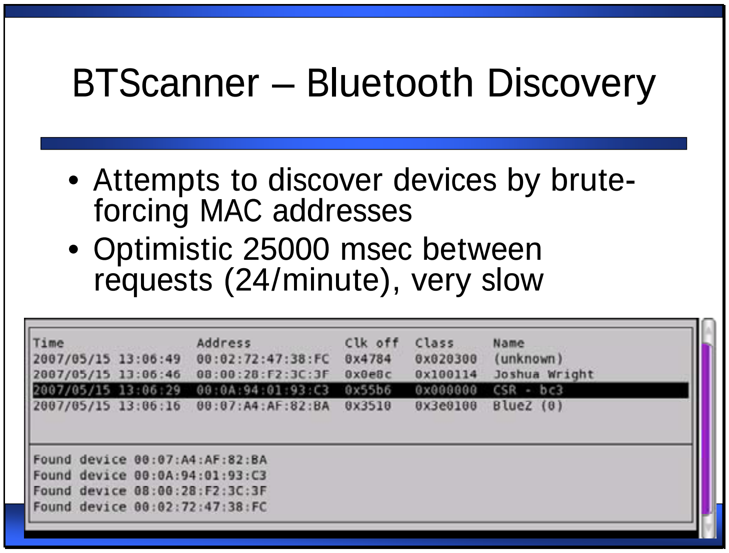
Hidden Risks of Bluetooth – © 2007 Joshua Wright

Common Misconception 4

"Devices in non-discoverable mode cannot be found"

Many devices rely on the secrecy of the BD_ADDR information for security. Bluetooth headsets, for example, do not have a mechanism for a user to specify a PIN value to as an authentication mechanism, and solely rely on not disclosing the BD_ADDR information in discoverable mode to protect the device.
We've established that the BD_ADDR information must be known by the slave device to pair with the master in a piconet, and that sniffing the wireless network does not reveal the BD_ADDR information in the baseband header. Current research suggests that this is an acceptable form of protection for securing Bluetooth headsets, as long as the BD_ADDR for the device remains a secret.

Slide 22



BTScanner – Bluetooth Discovery

- Attempts to discover devices by brute-forcing MAC addresses
- Optimistic 25000 msec between requests (24/minute), very slow

```
Time                 Address            Clk off   Class      Name
2007/05/15 13:06:49  00:02:72:47:38:FC  0x4784    0x020300   (unknown)
2007/05/15 13:06:46  08:00:28:F2:3C:3F  0x0e8c    0x100114   Joshua Wright
2007/05/15 13:06:29  00:0A:94:01:93:C3  0x55b6    0x000000   CSR - bc3
2007/05/15 13:06:16  00:07:A4:AF:82:BA  0x3510    0x3e0100   BlueZ (0)

Found device 00:07:A4:AF:82:BA
Found device 00:0A:94:01:93:C3
Found device 08:00:28:F2:3C:3F
Found device 00:02:72:47:38:FC
```

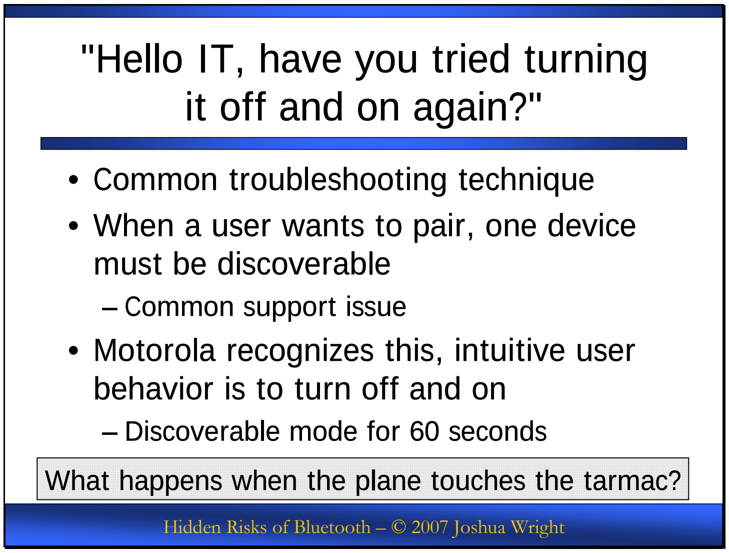
BTScanner – Bluetooth Discovery

One tool designed to identify the BD_ADDR information for a target device is BTScanner. Designed for Linux systems, BTScanner attempts to brute-force the 48-bit MAC address of a device in non-discoverable mode by issuing repeated connection requests to sequentially-selected BD_ADDR's.

Optimistically, BTScanner must spend 25,000 msec between each successive request for the BD_ADDR guess, which means that a single Bluetooth dongle can make 24 request each minute. This makes BTScanner a very slow tool when attempting to enumerate a large range of potential MAC addresses.

One mechanism to accelerate the BTScanner attack is to use more than one Bluetooth dongle in parallel. Using lots of Bluetooth dongles simultaneously, BTScanner can accelerate the attack, though even with 10 Bluetooth dongles, the attack is only guessing at 240 BD_ADDR a minute. Practically, if the attacker knows the first three bytes of the BD_ADDR from the manufacturer of the device (a Motorola headset, for example, will have a consistent three-byte OUI prefix), and has to enumerate the last three bytes of the BD_ADDR, BTScanner will need to make 16.7 million requests. At a rate of 240 requests each minute, the scan can take over 48 days to complete!

Slide 23



"Hello IT, have you tried turning it off and on again?"

- Common troubleshooting technique
- When a user wants to pair, one device must be discoverable
  - Common support issue
- Motorola recognizes this, intuitive user behavior is to turn off and on
  - Discoverable mode for 60 seconds

What happens when the plane touches the tarmac?

"Hello IT, have you tried turning it off and on again?"

As a helpdesk support operator for Microsoft Windows software, the first and probably the most important troubleshooting technique is to ask the user "Have you tried turning it off and on again?" This is a common troubleshooting technique, that even novice computer users have become accustomed to.

When two Bluetooth devices need to pair for the first time, one of the devices must be in discoverable mode. If both devices are not in discoverable mode by default (the preferred security configuration), the end-user will be unable to pair the devices and may resort to troubleshooting techniques to rectify the situation. Recognizing this, Motorola has adapted several of its Bluetooth devices such that when the device boots, it is in discoverable mode for a short period of time, usually 60 seconds. This adapts well to the common troubleshooting method of "have you tried turning it off and on again", since if the user decides to reboot the device when troubleshooting a pairing problem, the device will be available in discoverable mode for a short time.

However, this troubleshooting feature can have undesirable circumstances. When boarding a plane, passengers are advised to turn off all electronic devices, including mobile phones. However, as soon as the plane touches down, passengers are allowed to turn their phones on again. During this time, a curious attacker can easily obtain the BD_ADDR information for many devices, by taking advantage of this small window of discoverable mode behavior. The attacker does not need to exploit devices on the plane; if he has collected BD_ADDR information, he can use these addresses to attack his fellow passengers in the airport while waiting for connecting flights.

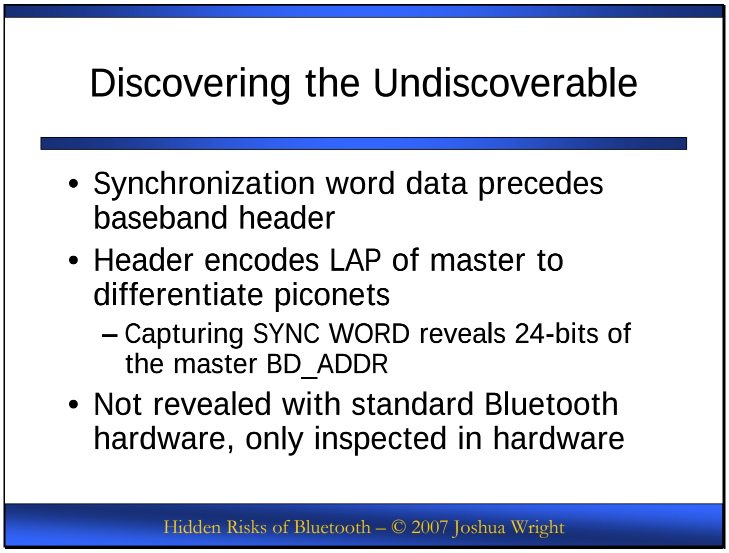
Slide 24

SAPPORO
LR3
DESIGNED FOR THE EXTRAORDINARY
Make your Bluetooth handset discoverable and get the whole story now.
750
Me

LR3 - Designed for the Extraordinary

Bluetooth has been adapted in metropolitan areas to deliver marketing and advertising information to devices as well. This picture of the corner of 7th and West 49th Street in New York City, NY, where a billboard advises passers-by to make their "… Bluetooth handset discoverable and get the whole story now", about the Land Rover LR3 vehicle. Once placed into discoverable mode, the billboard will beam an interactive application to the person walking by. However, an attacker in the same location can now also take advantage of this opportunity, and identify BD_ADDR information from people as they follow the instructions on the billboard.

Slide 25



Discovering the Undiscoverable

- Synchronization word data precedes baseband header
- Header encodes LAP of master to differentiate piconets
  - Capturing SYNC WORD reveals 24-bits of the master BD_ADDR
- Not revealed with standard Bluetooth hardware, only inspected in hardware

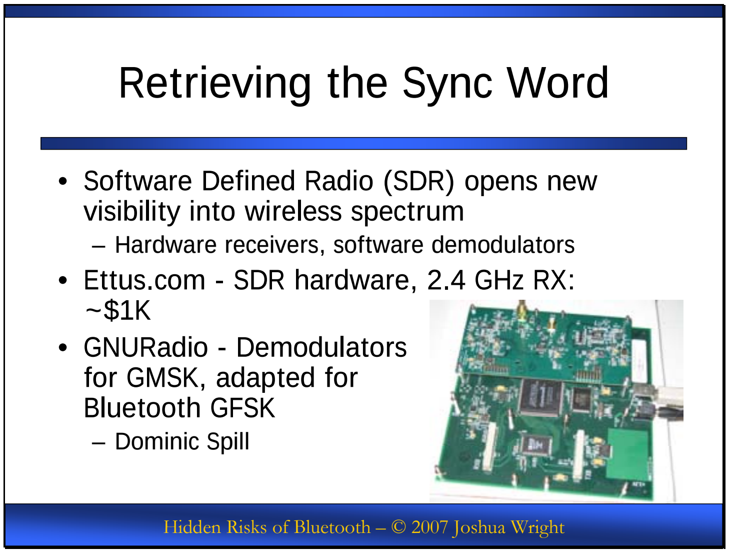
Discovering the Undiscoverable

In order to connect to a Bluetooth device, knowledge of the BD_ADDR is needed. While the secrecy of this information prevents other devices from connecting in an unauthorized manner, this is certainly a weak authentication mechanism.

For each single-frame slot transmitted by a Bluetooth piconet member includes a data preamble before the beginning of the baseband header known as the sync word. The sync word is used to differentiate traffic from multiple piconets by the receiving station, where the transmitter embeds the LAP (last three bytes) of the master device into the sync word. Capturing the sync word data reveals 24-bits of the BD_ADDR of the master of the piconet.

Knowledge of the LAP can be very valuable for the attacker, since it discloses a significant portion of the BD_ADDR for the piconet. This information cannot be retrieved by a standard Bluetooth dongle however (nor can it be returned with any of the commercial Bluetooth sniffers on the market) as it is only processed in hardware with the receipt of a frame, and it is not passed up to the host operating system.

Slide 26



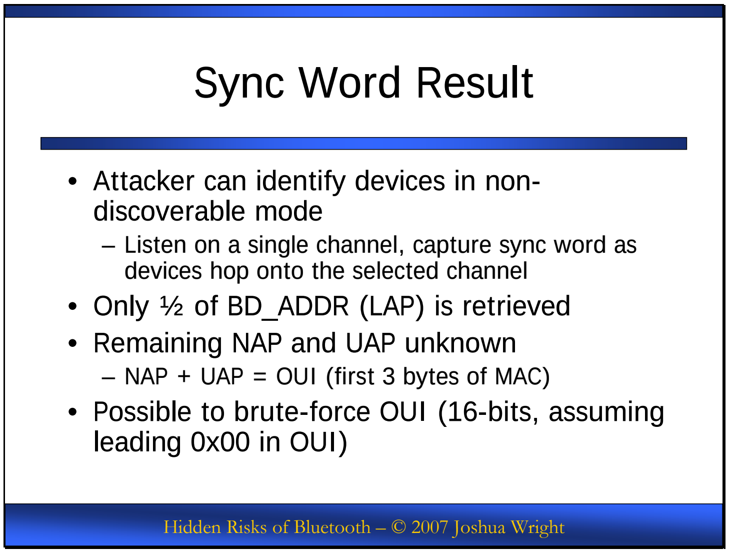
Retrieving the Sync Word

- Software Defined Radio (SDR) opens new visibility into wireless spectrum
  - Hardware receivers, software demodulators
- Ettus.com - SDR hardware, 2.4 GHz RX: ~$1K
- GNURadio - Demodulators for GMSK, adapted for Bluetooth GFSK
  - Dominic Spill

Retrieving the Sync Word

Tools to capture the sync word are not commonly available, as standard Bluetooth dongles are not designed to allow users to interact with the hardware beyond the interfaces exposed at the HCI later.
To overcome this limitation and have access to sniff Bluetooth frames, development board such as the University Software Radio Peripheral (USRP) can be used to write custom demodulators to retrieve Bluetooth data including the sync word. Much of the work to demodulate Bluetooth traffic is already complete through the GNURadio project, requiring MAC-layer processing to identify the start of Bluetooth data.
On August 1st 2007, Dominic Spill from the University College London published a paper at the Usenix Woot07 conference, where he debuted software to implement a minimally-featured Bluetooth stack using the USRP. While the USRP is unable to frequency hop in synchronization with the other Bluetooth transmitters (the USRP hardware is incapable of hopping at a rate of 1600 hops/second), it is possible to listen on a single frequency, and demodulate (decode) Bluetooth traffic as it is transmitted on the selected channel.

Slide 27



Sync Word Result

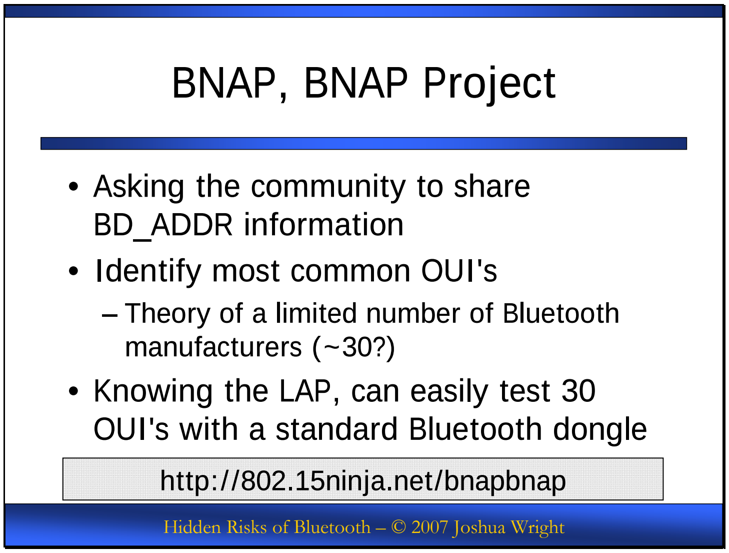
- Attacker can identify devices in non-discoverable mode
  - Listen on a single channel, capture sync word as devices hop onto the selected channel
- Only ½ of BD_ADDR (LAP) is retrieved
- Remaining NAP and UAP unknown
  - NAP + UAP = OUI (first 3 bytes of MAC)
- Possible to brute-force OUI (16-bits, assuming leading 0x00 in OUI)

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Sync Word Result

Using tools such as the USRP and GNURadio, an attacker can sniff on a single frequency to identify Bluetooth devices that are currently transmitting, even when in non-discoverable mode. The sync word content reveals 3-bytes of the BD_ADDR of the master (LAP), leaving only the first three bytes of the BD_ADDR (the OUI, or the NAP and UAP information) to be determined.
Assuming the leading byte of the BD_ADDR is 0, an attacker can adapt a tool like BTScanner to brute-force the remaining 2-bytes. This represents 65,536 possible addresses, which would take approximately 2 days with a single Bluetooth dongle to complete.
However, since we are examining the OUI information, it may not be necessary to test all BD_ADDR possibilities.

Slide 28



BNAP, BNAP Project

- Asking the community to share BD_ADDR information
- Identify most common OUI's
  - Theory of a limited number of Bluetooth manufacturers (~30?)
- Knowing the LAP, can easily test 30 OUI's with a standard Bluetooth dongle

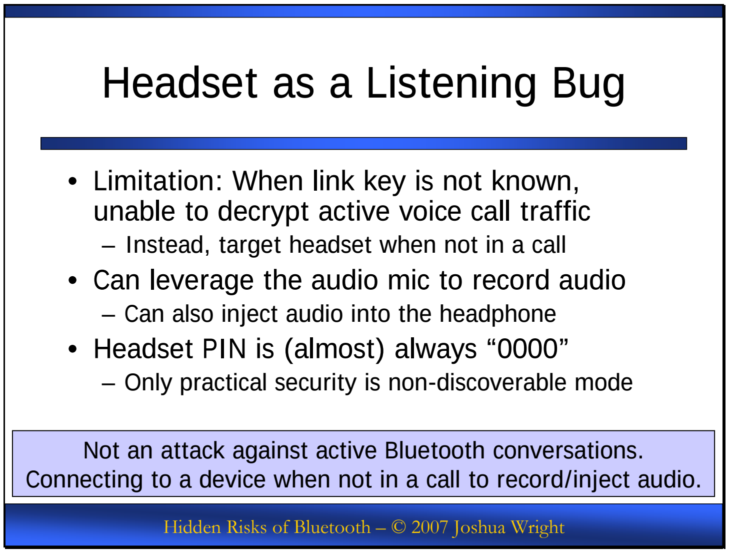
http://802.15ninja.net/bnapbnap

Hidden Risks of Bluetooth – © 2007 Joshua Wright

BNAP, BNAP Project

The BNAP, BNAP project was started to collect information about how vendors allocate Bluetooth addresses to devices. When the LAP is known, the attacker can reduce the amount of keyspace to search to discover the BD_ADDR by limiting their tests to well-known OUI's that have been used for Bluetooth device allocations. While a list of all the IEEE OUI's is available, there was no list of the OUI's being used by Bluetooth vendors. The BNAP, BNAP project asks the community to share the first several bytes of BD_ADDR information from any Bluetooth device. Using this information, we can examine how vendors are allocating Bluetooth device addresses, and identify the most common OUI's based on the frequency of submissions.

Slide 29



Headset as a Listening Bug

- Limitation: When link key is not known, unable to decrypt active voice call traffic
  - Instead, target headset when not in a call
- Can leverage the audio mic to record audio
  - Can also inject audio into the headphone
- Headset PIN is (almost) always “0000”
  - Only practical security is non-discoverable mode

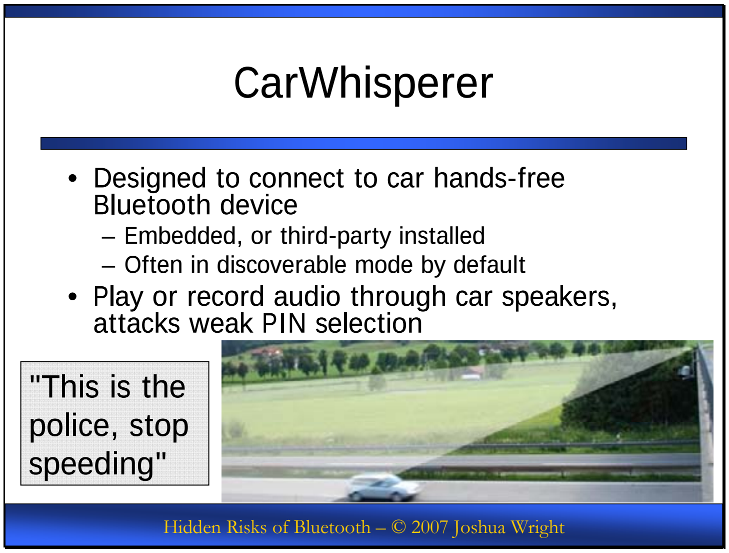
Not an attack against active Bluetooth conversations.
Connecting to a device when not in a call to record/inject audio.

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Headset as a Listening Bug

With the ability to identify the BD_ADDR of the master device, it is often possible to connect to the headset directly, leveraging the static, fixed PIN information for authentication. However, passive eavesdropping and decryption requires knowledge of the 128-bit link key that was generated when the two devices first paired. Knowledge of just the PIN information is not sufficient to capture and decrypt an active voice call.
An alternative attack is to exploit the headset when it is not actively engaged in a call, using the headset microphone to record any audio content, and potentially play arbitrary audio information through the headset to the wearer. Since the PIN on headset devices is commonly “0000”, it is trivial for an attacker to connect to the headset and send and receive the same kind of data that would normally be exchanged with a Bluetooth phone.

Slide 30



CarWhisperer

- Designed to connect to car hands-free Bluetooth device
  - Embedded, or third-party installed
  - Often in discoverable mode by default
- Play or record audio through car speakers, attacks weak PIN selection

"This is the police, stop speeding"

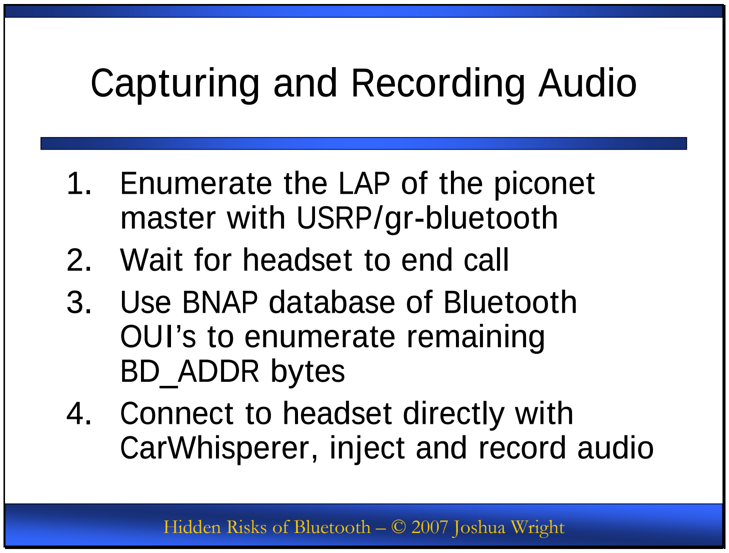
CarWhisperer

Many cars are shipping with Bluetooth technology built-in, often with the Bluetooth stack in discoverable mode by default with a simple, static PIN such as 1234 or 0000. The CarWhisperer tool was designed to demonstrate weaknesses in automotive Bluetooth installations, automating the process of connecting to an automobile's Bluetooth stack and playing selected audio files through the car's stereo speakers.

In testing this tool, the Trifinite group, a worldwide group of Bluetooth security researchers, found a long stretch of highway and a bridge overlooking the highway. Positioned on the bridge with a laptop running CarWhisperer and a high-gain directional antenna, the Trifinite group was able to connect to vulnerable Bluetooth stacks, and play an audio message in the car, helpfully informing the drivers that their car was vulnerable to Bluetooth attacks.

A limitation of the CarWhisperer is that is was only able to target devices who were in discoverable mode. Bluetooth hands-free systems deployed in cars that disable discoverable mode have not been vulnerable to the CarWhisperer attack, unless the BD_ADDR of the device was known through some other discovery mechanism.

Slide 31



Capturing and Recording Audio

1. Enumerate the LAP of the piconet master with USRP/gr-bluetooth
2. Wait for headset to end call
3. Use BNAP database of Bluetooth OUI's to enumerate remaining BD_ADDR bytes
4. Connect to headset directly with CarWhisperer, inject and record audio

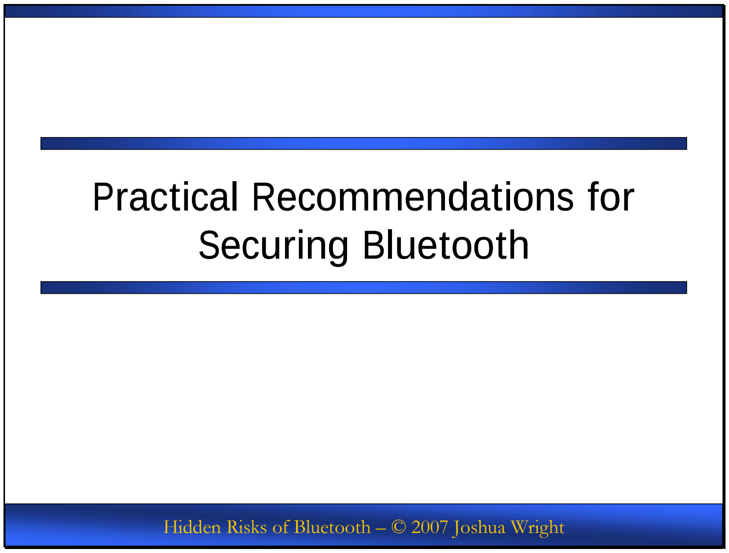
Hidden Risks of Bluetooth – © 2007 Joshua Wright

Capturing and Recording Audio

Out attack develops as follows:

Identify the presence of Bluetooth traffic using the USRP with the gr-bluetooth package.
Capture the sync word data to enumerate the LAP of the piconet master.
Wait for the headset to the end the call.
Use a modified version of BTScanner to enumerate the remaining BD_ADDR bytes.
Instead of brute-forcing the entire unknown 24-bit range, leverage the known Bluetooth OUI's from the BNAP, BNAP project to identify the full BD_ADDR.
Connect to the headset directly using the CarWhisperer tool, injecting and recording audio information.

Slide 32



Practical Recommendations for Securing Bluetooth

Practical Recommendations for Securing Bluetooth

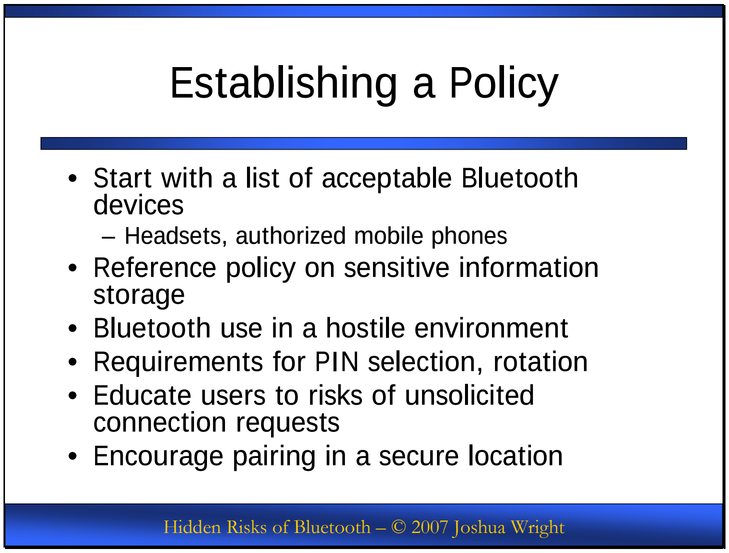
Now that we've examined different attacks and vulnerabilities affecting Bluetooth networks, let's examine some practical advice for securing these networks.

Slide 33

“The Good Old Days”

This page left intentionally blank.

Slide 34



Establishing a Policy

- Start with a list of acceptable Bluetooth devices
  - Headsets, authorized mobile phones
- Reference policy on sensitive information storage
- Bluetooth use in a hostile environment
- Requirements for PIN selection, rotation
- Educate users to risks of unsolicited connection requests
- Encourage pairing in a secure location

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Establishing a Policy

A security policy for your organization with regard to the use and deployment of Bluetooth technology is an excellent, low-cost first step in addressing the issues. Bluetooth policies should cover the following areas:

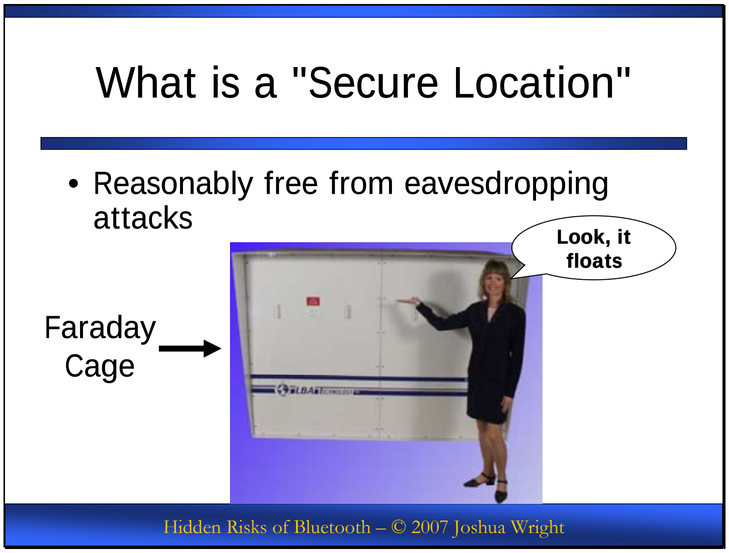
Acceptable Bluetooth devices: Identify a list of acceptable Bluetooth devices in your organization. Most organizations who wish to allow the use of Bluetooth technology will want to allow the use of headsets and Bluetooth-enabled phones, but may wish to forbid the use of Bluetooth on desktop and laptop systems where sensitive information is stored. Other organizations may wish to establish a "No Bluetooth" policy for strict controls on this ad-hoc technology.
Reference your policy on sensitive information storage: Some employees may be storing sensitive information on mobile Bluetooth devices, such as phones, PDA's and other devices. It is important to reference any existing policies on where sensitive information can be stored (for example, can confidential information be stored on USB drives that are easily lost or stolen?), requiring any Bluetooth devices to abide by this policy as well.
Identify how Bluetooth can be used in a hostile environment: Some organizations may wish to forbid the use of Bluetooth technology in hostile environments, such as trade shows, since they may be at greater risk to any number of attacks.
Requirements for PIN selection: What is the minimum PIN length that is required for Bluetooth devices? This length should be based on a reasonable amount of time that an adversary may be within range of an victim to implement an attack. For example, a 10-character PIN can be brute-forced with BTCrack in less than 14 hours, with probability on the side of the attacker, assume less than 7 hours. If this is a reasonable amount of time for exposure to an attacker, then an 10-character PIN may be acceptable for your environment. The PIN selection does not adequately defend against a targeted attack where the adversary may crack the PIN and then return to exploit the victim, so it is moot

to force a longer PIN selection. Also consider a PIN rotation policy, requiring that Bluetooth users change their PINs at specified intervals.

Educate users to risks of unsolicited connection requests: Advise users not to accept unsolicited connection requests on their Bluetooth devices, as this may open up an opportunity for the attacker to exploit the target device.

Encourage pairing in a secure location: During the pairing Bluetooth devices is when they are most vulnerable to attack. Advise users not to pair Bluetooth devices is open public areas (such as coffee shops) or other potentially hostile environments.

Slide 35



What is a "Secure Location"

- Reasonably free from eavesdropping attacks

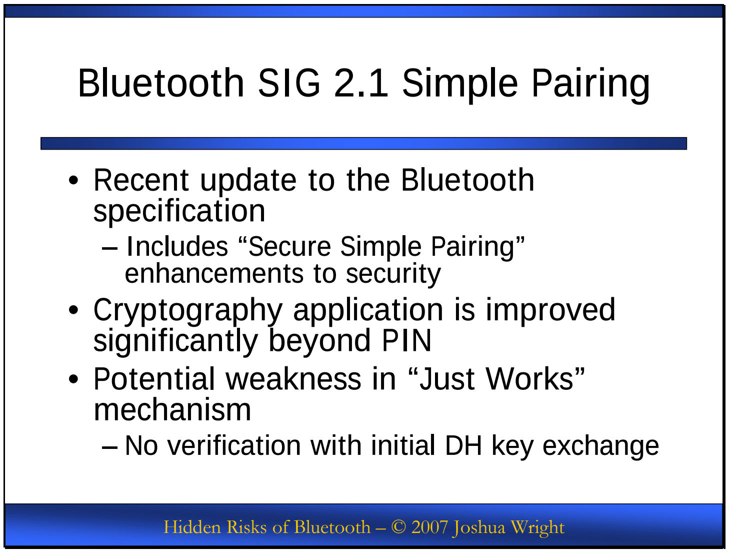
Faraday Cage →

Look, it floats

Hidden Risks of Bluetooth – © 2007 Joshua Wright

This page intentionally left blank

Slide 36



Bluetooth SIG 2.1 Simple Pairing

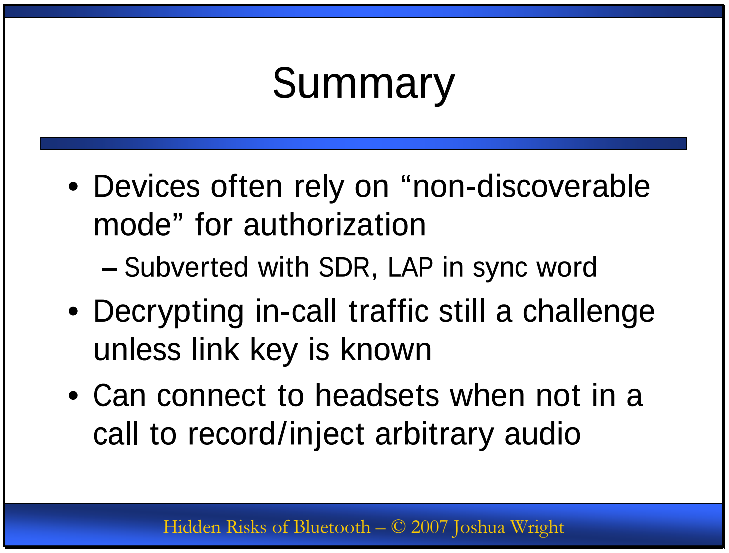
- Recent update to the Bluetooth specification
  - Includes “Secure Simple Pairing” enhancements to security
- Cryptography application is improved significantly beyond PIN
- Potential weakness in “Just Works” mechanism
  - No verification with initial DH key exchange

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Bluetooth SIG 2.1 Simple Pairing

In June 2007, the Bluetooth Special Interest Group ratified the Bluetooth 2.1 specification, which introduced several enhancements to the Bluetooth security model. Replacing the PIN and link key derivation authentication exchange are several authentication options, each suitable to a different class of device.
For Bluetooth headsets without display capabilities, the “Just Works” authentication exchange is used. Since there is no ability to display on the headset, and limited user input (e.g. one button that can be used for yes or no responses), the initial pairing between devices is automatic, assuming the remote entity is legitimate. Despite the fact that the headsets use Diffie Hellman (DH) key exchange, the lack of verification here could allow an attacker to connect to the headset in an unauthenticated fashion, similar to the attack described in this presentation. As there are no known Bluetooth 2.1 adapters shipping yet, the potential weakness described here is still untested.

Slide 37



Summary

- Devices often rely on “non-discoverable mode” for authorization
  - Subverted with SDR, LAP in sync word
- Decrypting in-call traffic still a challenge unless link key is known
- Can connect to headsets when not in a call to record/inject arbitrary audio

Hidden Risks of Bluetooth – © 2007 Joshua Wright

Summary

In this module, we've examined the Bluetooth specification and the capabilities and features of Bluetooth devices including the layered stack model, three classes of devices with varying transmit capabilities, and three security models. Understanding how Bluetooth operates is necessary for understanding that vulnerabilities exist in both the implementation of Bluetooth stacks, and in the design of the protocol as well.

Many Bluetooth devices with limited form-factors are unable to change the default PIN value, relying on the non-discoverable mode feature as an authorization mechanism. This is not a strong authorization approach however, as 24-bits of the BD_ADDR can be retrieved from the sync word in a single frame.

While decrypting traffic during a call is still a challenge due to the use of a 128-bit link key used for encryption, it is possible to connect to another headset directly once the BD_ADDR is known, and the device is no longer in an active call. This allows the headset to be used as a remote audio bug device, and potentially greater mischief with audio playback.

I hope you enjoyed this session on my adventure in exploiting a Bluetooth headset. I welcome any comments or questions on this material. Please contact me at jwright@willhackforsushi.com.