Taming the Blue Beast

A Survey of Bluetooth-Based Threats

As Bluetooth finds its way into millions of devices worldwide, it also becomes a prime target for hackers. The author presents a taxonomy for threats against Bluetooth-enabled devices, describes several of these threats, and identifies steps for threat mitigation.

John Paul Dunning
Virginia Tech

Albert walks into his favorite coffee shop to study for the next day's midterm. After placing his order, he takes a seat near the window and begins to pull out his notes when his phone vibrates. A message appears on the screen: "Do you wish to accept Edge3_update_5132.cab from ComMobile?" Assuming this to be an update for his new phone, he gladly accepts, hoping for some cool new apps. What he gets is a nasty piece of malware wreaking havoc on his phone.

Linda quickly finishes up a conversation with her husband as the rest of the team arrives to talk about the big merger. Placing her cell phone and headset on the conference table, she stands up to begin her presentation, unaware that a hacker bent on corporate espionage is listening in on the whole presentation.

Randy decides to leave his phone in the office so it doesn't interrupt him during his lecture. Coming back after class, he checks his phone to find he has an unusually high number of messages. Listening to his voice mail, he soon realizes that his phone has been used to prank-call several people in his contact list.

The Common Thread

The common thread running through all these attacks is Bluetooth (see the "Bluetooth Basics" sidebar). The first is a clever social engineering attack in which the hacker determined Albert's phone type and service provider from information gathered by a Bluetooth device-discovery scan of Albert's phone. The hacker used this information to customize an attack to convince Albert to install a program that wiped his phone's flash memory. Linda was the victim of an extended-range Bluetooth radio from several blocks away connecting to her Bluetooth headset by guessing the default PIN. This lets the hacker record everything picked up by the headset's microphone from farther away than would ordinarily be possible. In the third attack, the hacker was able to access telephony services on Randy's phone and place calls to people in his contact list.

In its decade of public use, hackers and researchers have discovered several security risks to Bluetooth-enabled devices. You might wonder why you haven't heard more about these security breaches. One reason is that most Bluetooth attacks go undetected or unreported. Unlike standard Internet-based networks, Bluetooth generally isn't monitored by intrusion-detection systems. Bluetooth attacks often target mobile and embedded devices that have few (if any) security features. Furthermore, the attacks are usually on a much smaller scale than attacks that make the news. Rather than millions of stolen credit-card numbers or a 10,000-node botnet used for mass spamming, Bluetooth attacks affect only a small number of devices within a limited proximity to the attacker. This makes it difficult to assess the real damage caused by hackers abusing the technology.

Bluetooth Security in a Nutshell

Bluetooth's original design supported a high level of information protection through stealth, frequency hopping, authentication, and encryption.[1]

Stealth is one of its best security features. Devices can hide in a network and refuse connections through discoverable and connectible modes. In discoverable mode, devices reply to inquiries, letting other devices discover their existence in a local area. In nondiscoverable mode, devices do not listen for inquiry scans and thus never announce their presence. In connectible mode, devices listen for requests to their Bluetooth address. In nonconnectible mode, they do not allow other devices to initiate connections.

Bluetooth broadcasts at radio frequencies between 2.402 and 2.480 GHz, supporting frequency hopping between 79 different channels. A timing sequence coordinates the frequency hopping, which occurs 1,600 times per second. This not only helps prevent signal jamming but also makes third-party monitoring of all traffic-specific frequencies much more difficult.

In 2007, a major overhaul of Bluetooth security led to some differences in security between legacy and current versions. Specifically, Bluetooth has four different security modes for authentication and encryption. The first three apply to legacy versions, whereas the fourth applies to current versions. Security mode 1 requires no authentication or encryption. Mode 2 uses authentication and encryption only for individual service communication, such as file transfer or synchronization. Mode 3 enforces authentication and encryption before devices can fully establish a link, thus encrypting all traffic. Mode 4, the newest security mode, uses secure simple pairing (SSP) to create service-level security, similar to security mode 2.

Bluetooth authentication performs device (not user) verification, which must be completed before devices can establish a connection. Legacy Bluetooth versions use a personal identification number (PIN) for initial authentication. Devices agree upon an alphanumeric string of up to 16 characters to authenticate pairing. Current SSP-enabled Bluetooth versions can use passkey entry, which prompts the user to agree or type in a specific six-digit number, which the slave device provides. The passkey entry is an improvement over the PIN because the encryption process doesn't use the authentication number.

Bluetooth encryption uses a stream cipher. The encryption algorithm pulls information from the master device's address, clock time, and an encryption key. The process encrypts only the payload, not the entire packet.

Bluetooth Threats

Bluetooth's built-in security features and its limited use in its early years helped avoid much attention in the hacker community. Bluetooth hacking gained momentum in 2003 with the release of BlueSnarfing (trifinite.org/trifinite_stuff_bluesnarf.html), a technique that facilitates direct access to some models of cellular phones. Since then, dozens of tools and methods have emerged to exploit different areas of Bluetooth technology.

Bluetooth Basics

Bluetooth is a wireless communication technology for short-range communications. First released in 1998, Bluetooth was designed for low power consumption and moderate data transfer rates over short ranges:

- Class 1: 100 meters
- Class 2: 10 meters
- Class 3: 1 meter

The technology forms a mobile ad hoc network, or *piconet*, between two or more wireless devices. The connecting devices establish a master–slave relationship in which the master device is in charge of the network. Devices are identified by a unique 48-bit string *address*; a user- or manufacturer-assigned human-readable *name*; and a *class*—identifying the device type, such as a cell phone, computer, printer, or video game console.

The Bluetooth Special Interest Group is an industry consortium that specifies and licenses the technology. Major specification versions and release dates are as follows:

- Version 1.0: 2001
- Version 2.0 + EDR (enhanced data rate): 2004
- Version 3.0 + HR (high speed): 2009
- Version 4.0: 2009

For more information, including the specifications, see www.bluetooth.com.

Classifying these threats can assist in determining threat severity, precautionary methods, and reactionary strategies. Understanding the similarities in threats also helps in applying previous knowledge to new discoveries. A Bluetooth Threat Taxonomy (Abott) provides a framework for classifying all Bluetooth-based threats. Aboott consists of nine distinct classes, many of which are already part of cybersecurity's standard terminology. Specifically, the Aboott classifications are surveillance, range extension, obfuscation, fuzzer, sniffing, denial of service (DoS), malware, unauthorized direct data access (UDDA), and man in the middle (MITM).

Table 1 lists the classifications and some example attacks and methods. Each attack appears in only one classification, based on its predominant characteristic, although a single attack can fall under several classifications.

Surveillance

Surveillance is used to acquire specific details about a device to assess possible vulnerable vectors. Often, these tools and methods cause no adverse effects to the target device.

Blueprinting (trifinite.org/trifinite_stuff_blueprinting.html) is a surveillance method designed for device

Table 1. Bluetooth attacks.

Attack classification	Threats
Surveillance	Blueprinting, bt_audit, redfang, War-nibbling, Bluefish, sdptool, Bluescanner, BTScanner
Range extension	BlueSniping, bluetooone, Vera-NG
Obfuscation	Bdaddr, hciconfig, Spooftooph
Fuzzer	BluePass, Bluetooth Stack Smasher, BlueSmack, Tanya, BlueStab
Sniffing	FTS4BT, Merlin, BlueSniff, HCIDump, Wireshark, kismet
Denial of service	Battery exhaustion, signal jamming, BlueSYN, Blueper, BlueJacking, vCardBlaster
Malware	BlueBag, Caribe, CommWarrior
Unauthorized direct data access	Bloover, BlueBug, BlueSnarf, BlueSnarf++, BTCrack, Car Whisperer, HeloMoto, btpincrack
Man in the middle	BT-SSP-Printer-MITM, BlueSpooof, bthidproxy

fingerprinting. It uses the device address, available services, and other information to profile the interface, device, and host operating system. Many computer, PDA, and cell phone models ship with the same Bluetooth interface type, so it's possible to determine the device model and interface from this information. Vendors configure many devices to perform a select number of Bluetooth services out of the box. Upon request, the device will list all the services it offers. Attackers can use this service information to profile the device and get information on potential vulnerable vectors.

Surveillance methods can also assess Bluetooth protocol-service multiplexers (PSMs) and RFComm channels. Bluetooth provides services that run on specific PSMs and RFComm channels,[1] similar to TCP/IP network ports. An attacking device can query a target device to reveal all offered services, but the target device determines what information to divulge. A device can be configured to withhold reporting certain unsecured services running on specific PSMs or RFComm channels. Bt_audit (trifinite.org/trifinite_stuff_btaudit.html) scans all PSMs and RFComm channels to determine if a target device has any undisclosed ports that could potentially lead to the discovery of unsecured services.

Location tracking is also possible for Bluetooth devices. War-nibbling is a technique for gathering information on Bluetooth-enabled devices in a specific location. An attacker can profile devices in a given area and possibly detect who is present based on the presence of their Bluetooth-enabled device.[2]

Bluefish takes surveillance of Bluetooth devices one step further. When a Bluefish system detects a new device, it records the Bluetooth device information and takes a photograph in the suspected direction of the device. Each time the device reenters the range of the computer running Bluefish, the process is repeated. This process can not only find time and place patterns but also associate pictures of the device owner with a device's presence.[3]

Range Extension

The specification for many wireless technologies limits their range of operation. The limitations are in place to prevent interference and to bind power expenditure (among other reasons). Extending a device's range might be against US Federal Communication Commission (FCC) rules, but attackers can use it to conduct attacks from a distance.

Bluetooone (trifinite.org/trifinite_stuff_bluetooone.html) extends a Bluetooth interface's range far beyond its standard scope. The method involves attaching a high-gain antenna to the standard Bluetooth radio to extend ranges from meters to kilometers. The yagi-directional antenna, as used in this method, gives the Bluetooth interface a small-angle, long-range boost, allowing many of the attacks discussed here to be conducted from a discreet distance.

Obfuscation

Attackers can use obfuscation to achieve a level of anonymity for launching an attack. For example, a hacker can masquerade as a device with another valid identity or create an entirely fictitious identity.

Bluetooth device addresses are assumed to be unique static identities. However, bdaddr (www.bluez.org) can change device addresses on certain Bluetooth chip sets by modifying the Bluetooth interface's firmware. By permanently resetting the interface device address, bdaddr nullifies the assumption of the device address as a unique identifier.

Hciconfig (www.bluez.org) is an application that lets users change most of its publicly provided Bluetooth information, including name and class. Used in combination with bdaddr, hciconfig lets attackers clone device addresses, names, and classes, thereby letting a laptop mask itself as a cell phone, automobile, mobile headset, and so on. Spooftooph (www.hackfromacave.com/projects/spooftooph.html) simplifies this process by automatically scanning for devices in range and cloning their Bluetooth device information according to the user's selection.

Fuzzer

Bluetooth packets follow a strict formatting standard.[1] Input that doesn't follow the format can result in buffer overflow, unauthorized data access, and application or system failure. Fuzzing is a technique used to test application input handling. Fuzzers operate by submitting nonstandard input to an application to achieve malicious results.

Bluetooth Stack Smasher (BSS)[4] and BluePass[5] are tools for assembling and sending packets to a target device. They help craft packets that test an application's ability to handle standard and nonstandard input data. Interpreting the malformed packets can cause unwanted results, such as buffer overflows, increased device activity, Bluetooth stack crash, and even device unresponsiveness.

BlueSmack (trifinite.org/trifinite_stuff_bluesmack.html) uses a Logical Link-Control and Adaptation Protocol (L2CAP) echo request, similar to an Internet Control Message Protocol (ICMP) ping. An attacker can abuse the echo request by changing its size to 600 bytes or greater. Some protocol stacks can't properly handle echo requests over a certain size, which can render the victim device's Bluetooth services unusable.

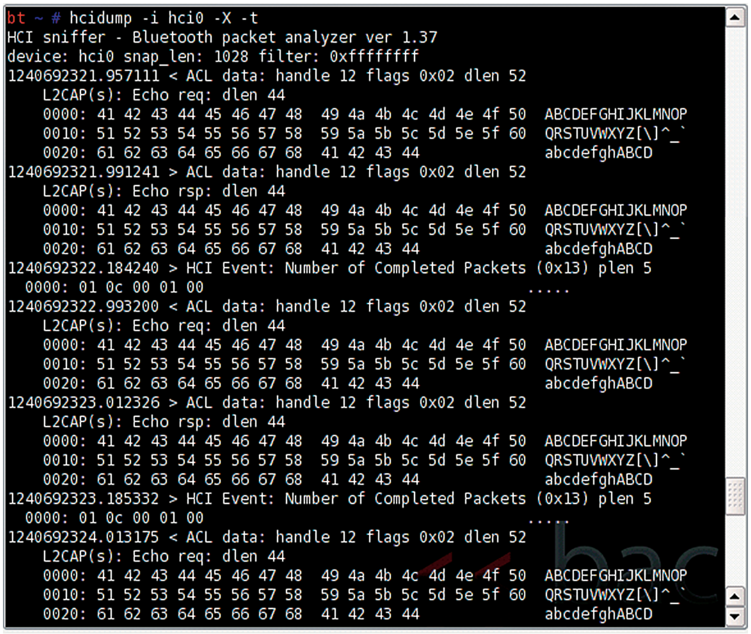
Sniffing

Sniffing is the process of capturing traffic in transit, much like eavesdropping on a phone line. Because Bluetooth broadcasts traffic wirelessly over RF, it's vulnerable to outside monitoring on specific frequencies.

Two commercially available Bluetooth sniffers are Frontline FTS4BT[6] and Lecroy Merlin.[7] These tools combine specialized hardware and software to monitor Bluetooth traffic by matching the connection's frequency hops and then capturing data in that frequency range. They log the sniffed data to a local file, which users can later view and analyze.

BlueSniff takes a different approach, using a modified Universal Software Radio Peripheral (USRP2) motherboard to monitor all 79 channels at the same time. It monitors each channel's traffic as binary data and reassembles the data into standard Bluetooth traffic for further analysis.[8]

HCIDump (www.bluez.org) is a utility that can capture and read raw Bluetooth traffic by monitoring local Bluetooth interfaces and capturing data from sniffed traffic. This tool assists attackers in discovering weaknesses in protocols and services. Figure 1 is an example output of a Bluetooth ping in hexadecimal and ASCII format.



```
bt ~ # hcidump -i hci0 -X -t
HCI sniffer - Bluetooth packet analyzer ver 1.37
device: hci0 snap_len: 1028 filter: 0xffffffff
1240692321.957111 < ACL data: handle 12 flags 0x02 dlen 52
    L2CAP(s): Echo req: dlen 44
    0000: 41 42 43 44 45 46 47 48  49 4a 4b 4c 4d 4e 4f 50  ABCDEFGHIJKLMNOP
    0010: 51 52 53 54 55 56 57 58  59 5a 5b 5c 5d 5e 5f 60  QRSTUVWXYZ[\]^_`
    0020: 61 62 63 64 65 66 67 68  41 42 43 44              abcdefghABCD
1240692321.991241 > ACL data: handle 12 flags 0x02 dlen 52
    L2CAP(s): Echo rsp: dlen 44
    0000: 41 42 43 44 45 46 47 48  49 4a 4b 4c 4d 4e 4f 50  ABCDEFGHIJKLMNOP
    0010: 51 52 53 54 55 56 57 58  59 5a 5b 5c 5d 5e 5f 60  QRSTUVWXYZ[\]^_`
    0020: 61 62 63 64 65 66 67 68  41 42 43 44              abcdefghABCD
1240692322.184240 > HCI Event: Number of Completed Packets (0x13) plen 5
  0000: 01 0c 00 01 00                                     .....
1240692322.993200 < ACL data: handle 12 flags 0x02 dlen 52
    L2CAP(s): Echo req: dlen 44
    0000: 41 42 43 44 45 46 47 48  49 4a 4b 4c 4d 4e 4f 50  ABCDEFGHIJKLMNOP
    0010: 51 52 53 54 55 56 57 58  59 5a 5b 5c 5d 5e 5f 60  QRSTUVWXYZ[\]^_`
    0020: 61 62 63 64 65 66 67 68  41 42 43 44              abcdefghABCD
1240692323.012326 > ACL data: handle 12 flags 0x02 dlen 52
    L2CAP(s): Echo rsp: dlen 44
    0000: 41 42 43 44 45 46 47 48  49 4a 4b 4c 4d 4e 4f 50  ABCDEFGHIJKLMNOP
    0010: 51 52 53 54 55 56 57 58  59 5a 5b 5c 5d 5e 5f 60  QRSTUVWXYZ[\]^_`
    0020: 61 62 63 64 65 66 67 68  41 42 43 44              abcdefghABCD
1240692323.185332 > HCI Event: Number of Completed Packets (0x13) plen 5
  0000: 01 0c 00 01 00                                     .....
1240692324.013175 < ACL data: handle 12 flags 0x02 dlen 52
    L2CAP(s): Echo req: dlen 44
    0000: 41 42 43 44 45 46 47 48  49 4a 4b 4c 4d 4e 4f 50  ABCDEFGHIJKLMNOP
    0010: 51 52 53 54 55 56 57 58  59 5a 5b 5c 5d 5e 5f 60  QRSTUVWXYZ[\]^_`
    0020: 61 62 63 64 65 66 67 68  41 42 43 44              abcdefghABCD
```

Figure 1. HCIDump packet capture. The output captures a Bluetooth ping flood. In Bluetooth pinging, an L2CAP echo request packet is sent to another device and awaits a reply. A Bluetooth ping flood is a form of DoS designed to send a high volume of packets to a target device to saturate the communication channel. The left column shows the data in hexadecimal format; the right column shows it in ASCII.

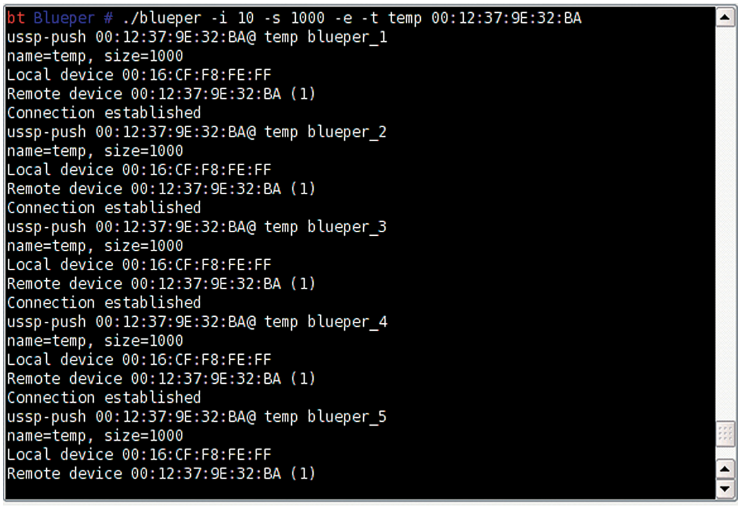
Denial of Service

The DoS classification applies to attacks that deny resources to a target. These attacks often target communication channels, but they can relate to any service the device uses, including the processor, memory, disk space, battery life, and system availability.

Intentionally saturating a communication channel or preventing communication over it is known as jamming. Any technology that uses wireless communication is subject to this type of attack. Bluetooth is no exception to this rule, but it handles jamming better than most wireless technologies by using adaptive frequency hopping (AFH) to avoid wireless interference. AFH identifies channels that cause collisions in packet transfer and routes around them. To effectively jam Bluetooth with AFH enabled, an attack must block all 79 channels at the same time.

Bluejacking is a technique for abusing the vCard feature on mobile phones. vCards are similar to business cards—users send them as short formatted messages between Bluetooth-enabled phones. Accepting vCards often requires no interaction on the receiver's end, opening a way for attackers to send anonymous messages without any credentials. The attack can be used to frighten users with suspicious-looking messages on their mobile devices.[9]

Blueper (www.hackfromacave.com/blueper.html) is designed to abuse Bluetooth file transfer on select mobile devices. It floods the target with file transfer requests. One possible result is a rather benign annoyance to the user through a continual stream of pop-up



```
bt Blueper # ./blueper -i 10 -s 1000 -e -t temp 00:12:37:9E:32:BA
ussp-push 00:12:37:9E:32:BA@ temp blueper_1
name=temp, size=1000
Local device 00:16:CF:F8:FE:FF
Remote device 00:12:37:9E:32:BA (1)
Connection established
ussp-push 00:12:37:9E:32:BA@ temp blueper_2
name=temp, size=1000
Local device 00:16:CF:F8:FE:FF
Remote device 00:12:37:9E:32:BA (1)
Connection established
ussp-push 00:12:37:9E:32:BA@ temp blueper_3
name=temp, size=1000
Local device 00:16:CF:F8:FE:FF
Remote device 00:12:37:9E:32:BA (1)
Connection established
ussp-push 00:12:37:9E:32:BA@ temp blueper_4
name=temp, size=1000
Local device 00:16:CF:F8:FE:FF
Remote device 00:12:37:9E:32:BA (1)
Connection established
ussp-push 00:12:37:9E:32:BA@ temp blueper_5
name=temp, size=1000
Local device 00:16:CF:F8:FE:FF
Remote device 00:12:37:9E:32:BA (1)
```

Figure 2. Blueper attack. An attacker configured Blueper to upload 10 files with a file size of 1,000 Kbytes to the device with address 12:37:9E:32:BA. This figure shows that five of the 10 uploads have completed.

messages for file transfer requests. A more detrimental result is data written to a target device disk without user interaction or previous authentication, causing some devices to temporarily halt execution or crash. Figure 2 is an example of this attack.

Malware

Malware is a malicious form of software, often self-replicating, that carries out various activities such as data mining, accessing personal files, password theft, file corruption, and system reconfiguration. Commonly known malware subsets include viruses, worms, and Trojan horses.

The Caribe[10] and CommWarrior[11] worms propagate though Bluetooth communication, infecting cell phones running Symbian OS. The targeted device's user receives a message to accept the incoming file. Based on the worm file type, once downloaded, the worm can bypass the normal user prompt for execution, install itself in hidden directories on the host device, and set itself to autorun. It then begins to search for Bluetooth devices in range and propagates itself.

Unauthorized Direct Data Access

UDDA attacks gather private information for unauthorized entities by penetrating devices through loopholes in security, allowing unauthorized access to privileged information.

Some attacks—for example, BlueBug (trifinite.org/trifinite_stuff_bluebug.html) and BlueSnarf++ (http://trifinite.org/trifinite_stuff_bluesnarf.html)—facilitate unauthorized access to certain cell phone models, letting attackers view contacts, text messages, pictures, call records, and so on. They can also send a command to a victim device on a covert channel, thus avoiding user detection. UDDAs also use phone features such as short message service (SMS), Internet connection, and telephony to gain complete control of a device through its Bluetooth connection. The attacker is then free to place phone calls, copy contact lists, and reconfigure call forwarding.

Attackers can also potentially obtain a Bluetooth PIN. BTCrack[12] and btpincrack (http://openciphers.sourceforge.net/oc/btpincrack.php) use a brute-force method to crack the PIN. They capture packets in the pairing process and compare them with attacker-crafted packet parameters, which they generate by enumerating PINs for encrypting standard packet content. The time it takes to break a PIN is directly proportional to its length—for example, on a standard desktop it takes milliseconds to crack a four-digit PIN but several thousand years to crack a 16-digit PIN.

Another way of discovering a device's PIN is to directly pair with a device using common default passwords. Many manufacturers ship devices, such as headsets and computer mice, with default static passwords that are universal to a particular model and generally short and simple, such as 0000 or 1234. CarWhisperer (trifinite.org/trifinite_stuff_carwhisperer.html) automates the access to Bluetooth-enabled devices with default settings (specifically, headsets and hands-free units) by guessing the default PIN. Once connected, the attacker can extract audio from or inject it into the target device.

Man in the Middle

MITM attacks place an attacking device between two connected devices to act as a relay (the attacker uses obfuscation to hide the attacking device). Previously paired devices send their information to the attacking device, which then relays it to its intended destination.

Security features in current Bluetooth versions are designed to thwart many MITM attacks. However, the BT-SSP-Printer-MITM attack shows possible vulnerabilities in the newer Bluetooth standards.[13] This attack focuses on the Just Works connection option in security mode 4, which lets devices pair without authentication. The BT-SSP-Printer-MITM attack sets the attacker's device as a relay point between the user device and a printer. When the user device connects to the printer using the Just Works method, the attacker breaks the connection by using some form of DoS. The user, feeling frustrated that the printer isn't working, deletes the association and attempts to reestablish communication. The attacker's device then poses as both the user device and the printer in an attempt to connect to both devices and act as a relay. This gives the attacker access to all data sent between the user device and the printer.

Table 2. Threat levels.

Attack classification	Threat level
Surveillance	*Low*: Generally harmless on its own. Its main purpose is to gather information, facilitating the use of other tools.
Range extension	*Low*: Generally harmless on its own. Its main purpose is to give attackers a safe range from which to conduct attacks.
Obfuscation	*Low*: Generally harmless on its own. Its main purpose is to hide the attacker's identity.
Fuzzer	*Medium*: Bluetooth isn't often used for critical communication, so fuzzer breakdowns of those communication channels often result in only frustration and inconvenience.
Sniffing	*Medium*: Sniffing can be useful in extracting data from unencrypted traffic (which some devices use by default), but traffic is usually encrypted. Each packet is encrypted individually using a different key stream, making decryption difficult for an attacker.
Denial of service	*Medium*: Bluetooth isn't often used for critical communication, so DoS breakdowns of those communication channels often result in only frustration and inconvenience.
Malware	*Medium*: These attacks can be malicious, but the wide range of Bluetooth devices limits their threat to a small number of devices. The short range of Bluetooth communication also hinders the spread of malware.
Unauthorized direct data access	*High*: This classification is possibly the most detrimental because of the effectiveness of some attacks and the seriousness of information theft.
Man in the middle	*High*: MITM attacks are more easily conducted against devices using security mode 1 or the Just Works setting in security mode 4. However, effective MITM methods in use today are dangerous because they bypass authentication and gain access to all transferred data.

Threat Levels

All threats are not created equal. Different attack classifications warrant different threat levels—for example, surveillance and range-extension methods can be viewed as benign when not combined with more serious attacks such as UDDA and MITM.

Returning to the threat scenarios presented in this article's introduction, we can view the UDDA attack that took control of Randy's phone to place prank calls as a malicious nuisance. The execution of malware on Albert's phone was more serious, resulting in the loss of personal information and the phone itself. The combination of range extension, surveillance, and UDDA on Linda's headset was potentially more detrimental. As a tool of corporate espionage, it could have ruined her company. These examples illustrate that the type of attack and context greatly affect the threat level.

Aboott groups threats to facilitate a better understanding of existing and 0-day attacks. The threat level is part of this understanding. It depends on the potential harm the attack can inflict. Table 2 summarizes the practical dangers of Aboott classifications.

Threat Mitigation

With all these Bluetooth hacking methods available, you might be wondering, "Should I use Bluetooth at all?" In my opinion, the answer in many cases is "Yes, if proper security is in place." Bluetooth is a wonderful technology with many practical applications. In general, devices with properly configured security settings are safe from most Bluetooth threats. Most weaknesses come from lax default security settings, poor software development practices, and users' lack of understanding about Bluetooth security.

The attack on Randy's phone, for example, was successful only because of the manufacturer's poor development practices. The social engineering attack on Albert's phone was successful because he didn't understand common Bluetooth file transfers. Linda's headset was breached because it was powered on while not in use and retained a manufacturer-assigned default PIN for authentication. Security steps could have prevented all these attacks. Table 3 presents steps for users, manufacturers, and the specification to reduce attack threats.

Bluetooth security has several systemic problems that can't be mitigated. First, because it transmits data wirelessly, a third party can monitor the data within a limited range. Also, because Bluetooth doesn't rely on a centralized communication medium, such as the Internet, no third-party entities can verify device addresses, names, or classes. Users must be responsible for device security. Many low-resource devices also cause problems because they can't install updates or patches. Exploits developed for these devices will be effective as long as the device is in use. Users must consider these systemic problems before implementing Bluetooth on any security-critical systems.

Table 3. Steps to mitigate Bluetooth attack threats.

Responsible entity	Action	Explanation
User	Disable Bluetooth when not in use.	Bluetooth is often used for short-term interdevice connections. When not in use, the best defense against attacks is to disable Bluetooth through hardware or software controls.
User	Disable unused services.	Many systems let users specify which services to enable/disable. For example, users might want to enable the audio gateway on a mobile phone but disable file transfer.
User	Place Bluetooth devices in nondiscoverable mode when not pairing.	A device should only be discoverable during initial pairing. Afterward, devices will be able to locate each other without being in discoverable mode. Devices in nondiscoverable mode are much more difficult for an attacker to find.
User	Place Bluetooth devices in security mode 2, 3, or 4, requiring authentication and encryption for communication.	This often involves selecting a settings option such as "enable encryption" or "authentication required." These settings help prevent connection from unauthorized devices and make it more difficult to extract data from sniffed traffic.
User	Avoid using Just Works.	The Just Works association model doesn't protect against MITM attacks. It also facilitates device connections without any form of authentication.
User	Use alphanumeric PINs, 12 digits or greater in length.	This helps prevent brute-force password guessing and makes it almost impossible for attackers to extract the password from cracking attempts on sniffed traffic.
User	Never accept files or messages from untrusted devices.	Files and messages can carry attacks against a device. Attackers can easily spoof the device name, so it's best to use a second factor of verification, such as a verbal conversation, before accepting a connection.
User	Never accept pairing with untrusted devices.	So many services are available on Bluetooth that it can be difficult to determine what you're agreeing to when a message is presented for action. Pairing is also permanent unless partnerships are later deleted. Pairing with an untrusted device can provide access to all Bluetooth services enabled on the local device.
User	Change PINs semifrequently.	This is good practice with any form of authentication. Most Bluetooth authentication occurs just once, so changing PINs can help prevent previously trusted devices from regaining access to a device without user notification.
Manufacturer	Make input validation a high priority during development of Bluetooth-related tools.	This basic principle applies to all software development. Software relating to the use of Bluetooth should be rigorously tested to prevent buffer overflows and illegal directory traversals.
Manufacturer	Disable all unnecessary protocol-service multiplexers (PSM) and RFComm channels.	Closing all unused PSMs and RFComm channels helps prevent attackers from gaining access to standard device services and back doors left open from testing.
Manufacturer	Disregard traffic not formatted to Bluetooth specification.	This will help prevent fuzzing and enforce the Bluetooth standards.
Manufacturer	Test all products with applicable hacking tools for vulnerabilities.	Using the tools such as those discussed in this article can help reveal vulnerabilities during production before the product goes on the market.
Specification	Offer two-factor authentication.	Because initial authentication often occurs only once, a second factor of authentication is warranted for devices that might have multiple users or be at risk for theft. This second form of authentication could be required for each pairing and/or service use.

Bluetooth threats will likely become more prevalent as the technology's growing popularity draws even more of the hacker community's attention. Imagine a self-propagating worm infecting phones in Washington, DC, or a Bluetooth sniper listening in on Wall Street conversations. A hacker's greatest advantage would be the lack of public concern for Bluetooth as a threat vector.

Better understanding the potential for such threats can greatly diminish their effectiveness. The Abootț classifications and mitigation steps can help users and device manufacturers assess the current state of their

Bluetooth-enabled devices and implement mitigation techniques accordingly. □

References

1. *Bluetooth Specification, v. 3.0 + HS*, Bluetooth SIG, Apr. 2009; www.bluetooth.com/Bluetooth/Technology/Building/Specifications/Default.htm.
2. O. Whitehouse, "War Nibbling: Bluetooth Insecurity," white paper, @stake Inc., Oct. 2003; www.wardriving.ch/hpneu/blue/doku/atstake_war_nibbling.pdf.
3. A. Reiter, "Bluefish Software Finds Bluetooth Devices, Takes Photo of Area," blog entry, 6 Jan. 2005; www.cameraphonereport.com/2005/01/bluefish_softwa.html.
4. P. Betouin, "[Infratech – release] version 0.6 of Bluetooth Stack Smasher," secuobs.com, 2 May 2006; www.secuobs.com/news/print05022006-bluetooth10.shtml.
5. G. Me, "Exploiting Buffer Overflows over Bluetooth: The BluePAss Tool," *Proc. 2nd IFIP Int'l Conf. Wireless and Optical Communications Networks* (WOCN 05), IEEE Press, 2005, pp. 66–70.
6. "FTS4BT Bluetooth Protocol Analyzer and Packet Sniffer," product data sheet, Frontline Test Equipment, 2010; www.fte.com/products/FTS4BT.aspx.
7. "CATC Merlin II," product data sheet, LeCroy, 3 Nov. 3 2003; www.lecroy.com/files/pdf/LeCroy_MerlinII_Datasheet.pdf.
8. D. Spill and A. Bittau, "BlueSniff: Eve Meets Alice and Bluetooth," *Proc. Usenix Workshop on Offensive Technologies* (WOOT 07); www.usenix.org/event/woot07/tech/full_papers/spill/spill.pdf.
9. A. Becker, "Bluetooth Security & Hacks," unpublished paper, 16 Aug. 2007; http://gsyc.es/~anto/ubicuos2/bluetooth_security_and_hacks.pdf.
10. "Cabir," F-Secure Corporation, 2004; www.f-secure.com/v-descs/cabir.shtml.
11. M. Hines, "CommWarrior Guns for Nokias," CNET News.com, 8 Mar. 2005; news.zdnet.co.uk/security/0,1000000189,39190552,00.htm.
12. T. Karygiannis and L. Owens, *Wireless Network Security 802.11, Bluetooth and Handheld Devices*, special publication 800 48, US Nat'l Inst. of Standards and Technology, 2002.
13. K. Haataja and K. Hypponen, "Man-in-the-Middle Attacks on Bluetooth: A Comparative Analysis, a Novel Attack, and Countermeasures," *Proc. 3rd Int'l Symp. Communications, Control, and Signal Processing* (ISCCSP 08), IEEE Press, 2008, pp. 1096–1102.

***John Paul Dunning** is a graduate student at Virginia Tech working as research assistant in the university's Information Security Lab. He also founded and maintains www.hackfromacave.com, a Web site devoted to offensive and defensive security. His research interests are in wireless and portable security technologies. Dunning has a BS in computer science from Virginia Tech. Contact him at jpvt40@vt.edu.*