Exposing the BlackBerry Boogey Man

SECTor
Security Education Conference

Adam Meyers - SECTor 2010

Agenda

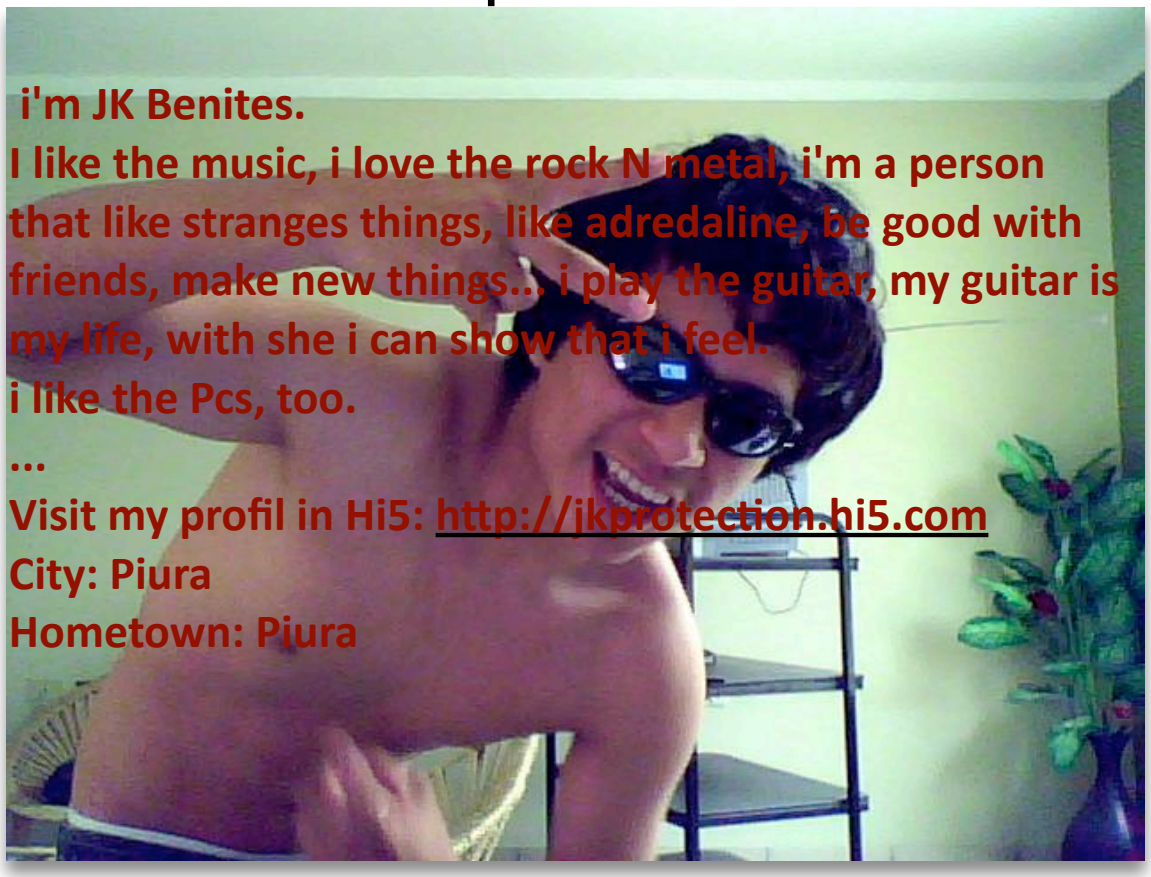
- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

Who are you, and what are you doing here?

- SRA
  - Leading provider of technology and strategic consulting services and solutions - including systems design, development and integration; and outsourcing and managed services.
  - Comprehensive cyber security practice integrating security architecture, risk assessments, and certification & accreditation. SRA's IA practice currently rated at NSA-CMM Level 3.
- Adam
  - Security Consultant
  - Penetration Test Team
  - Forensic Technician
  - Security Architect
  - Reverse Code Analysis

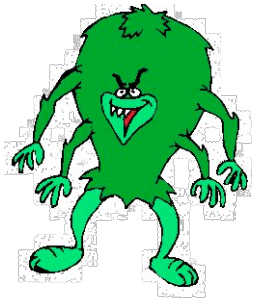
Hacker Fail

- Fall 2008 a promise is made
- Meet JK Benites
- This ‘genius’ left his name in the malware he wrote to steal banking credentials and ended up at a certain US Government Agency



BlackBerry Boogey Man

- “If you go to country X with a blackberry you are immediately owned”
- “We can decrypt in memory data with no password”
- “If you have a blackberry I can arbitrarily turn on the mic and listen to you”
- “Exploiting the blackberry is trivial”



Blackberry Truths

- Remote Exploitation is theoretically possible, though no publicly available examples are available – therefore probably not 'trivial'
- Remote Access Toolkits do exist
  - Tyler Shields – txsbbspy (publicly released code)
  - Flexispy (commercial)
  - Mobile Spy (commercial)
- Blackberry targeting has occurred
  - Etisalat
- The GoI, GoSA, GoX are trying to force RIM to provide monitoring capabilities

Caveats

- The scope of this presentation is Blackberry Enterprise devices and architecture – not blackberry stand alone devices (e.g. personal cell phones)
- This is all based on and using publicly available and **unclassified** information

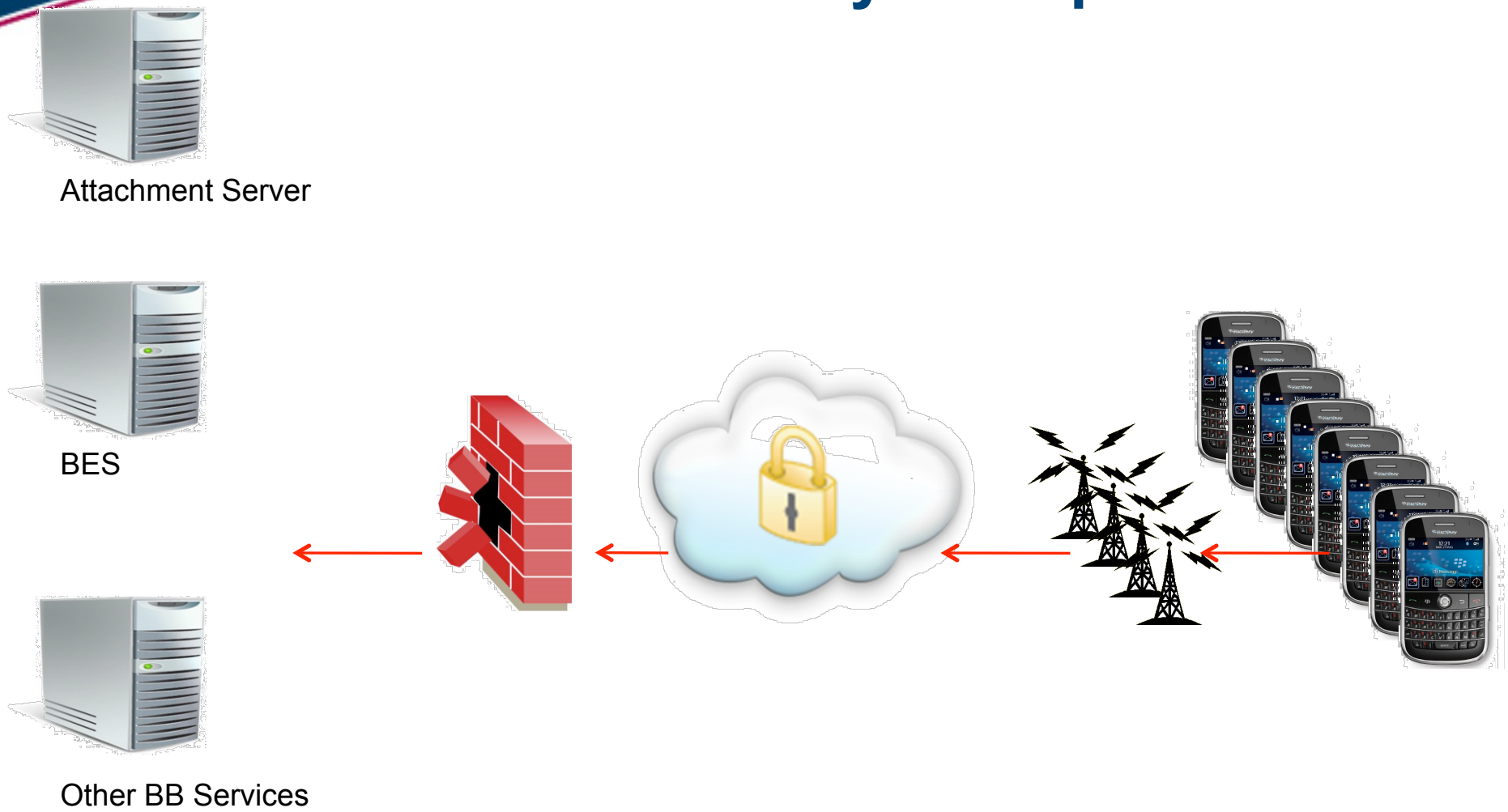
Acronyms

- BES – BlackBerry Enterprise Server
- MDS – Mobile Data Service
- MVS – Mobile Voice System
- JVM – Java Virtual Machine

Agenda

- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

BlackBerry Enterprise Architecture



BlackBerry Enterprise Components

- Attachment Service – Converts Attachments for handheld
- Collaboration Service – Integrate corporate IM (e.g. Microsoft Communicator)
- Configuration Database – Database for BES component configuration
- Controller – Monitor all components health
- Dispatcher – Compress and encrypt data for handheld
- Manager – Administration Console
- MDS Repository – Stores MDS runtime applications
- MDS Connection Service – Bridge handheld to intranet for application/browsing
- MDS Services – Manage MDS integration and device access/installation
- Messaging Agent – Interconnect to mail/calendar/contacts
- Policy Service – Manage IT Policies on devices and provisioning
- Router – Connect services to wireless network
- Synchronization Service – Organize data between devices and messaging server

Enterprise Components

- Management
  - controls mobile devices and can enforce enterprise policies regarding usage, third party software, bluetooth, password, etc.
  - Database (MSSQL) of devices, users, etc
- Components
  - Interconnect into Microsoft Exchange Server
  - Best Practice is to use a separate system
  - Interconnect between blackberry device and existing enterprise applications and systems

Agenda

- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

Blackberry API Development

Architecture

MDS Runtime

JavaME VM

Kernel

Java Virtual Machine

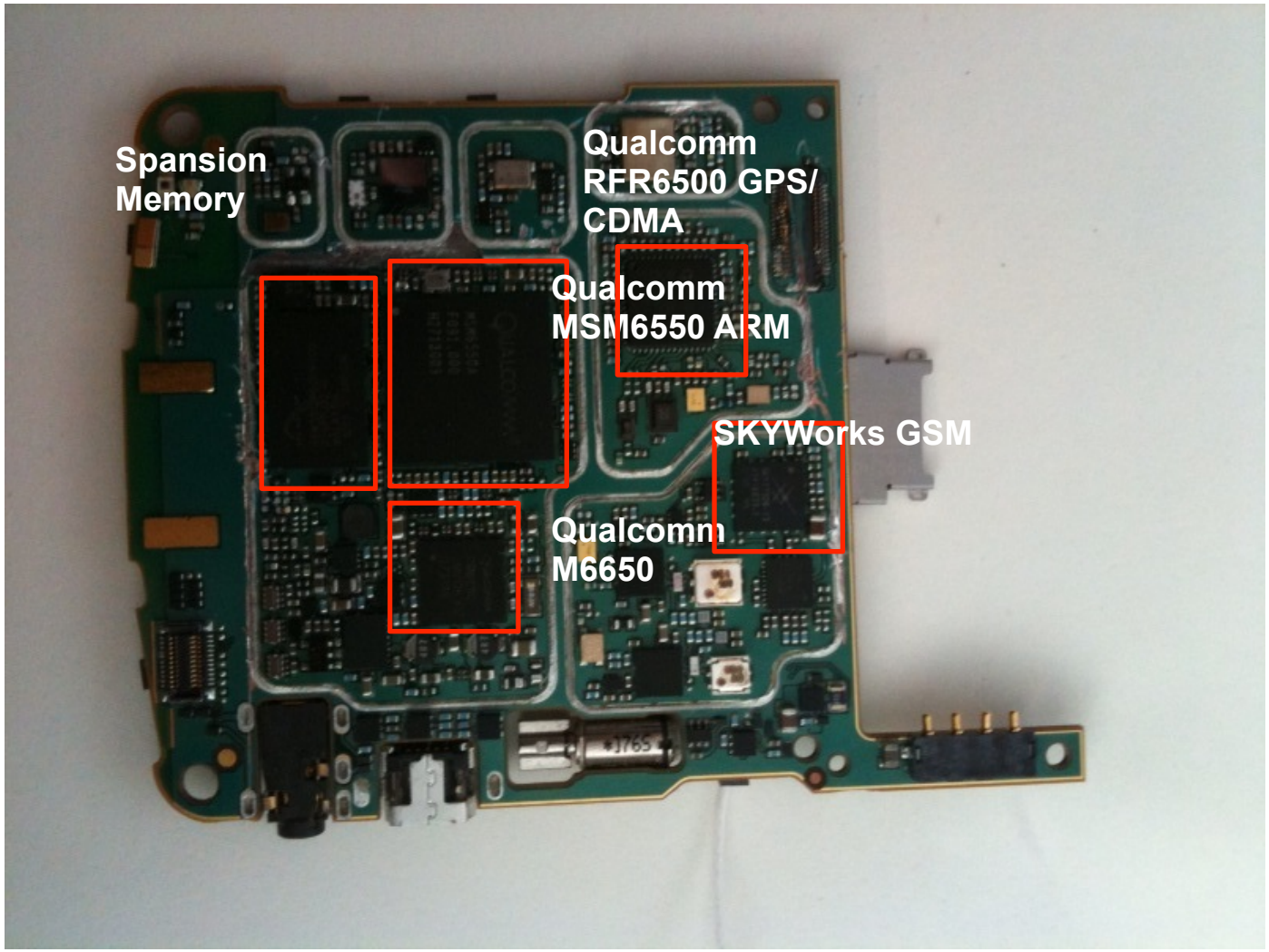
- JavaME VM
  - Connection Limited Device Configuration (CLDC)
  - Mobile Information Device Profile (MIDP)
  - Bberry API
- JVM Responsibility
  - Inter-Process Communications
  - Memory Management (Garbage Collection)
  - Process Management
  - Exception Handling

Agenda

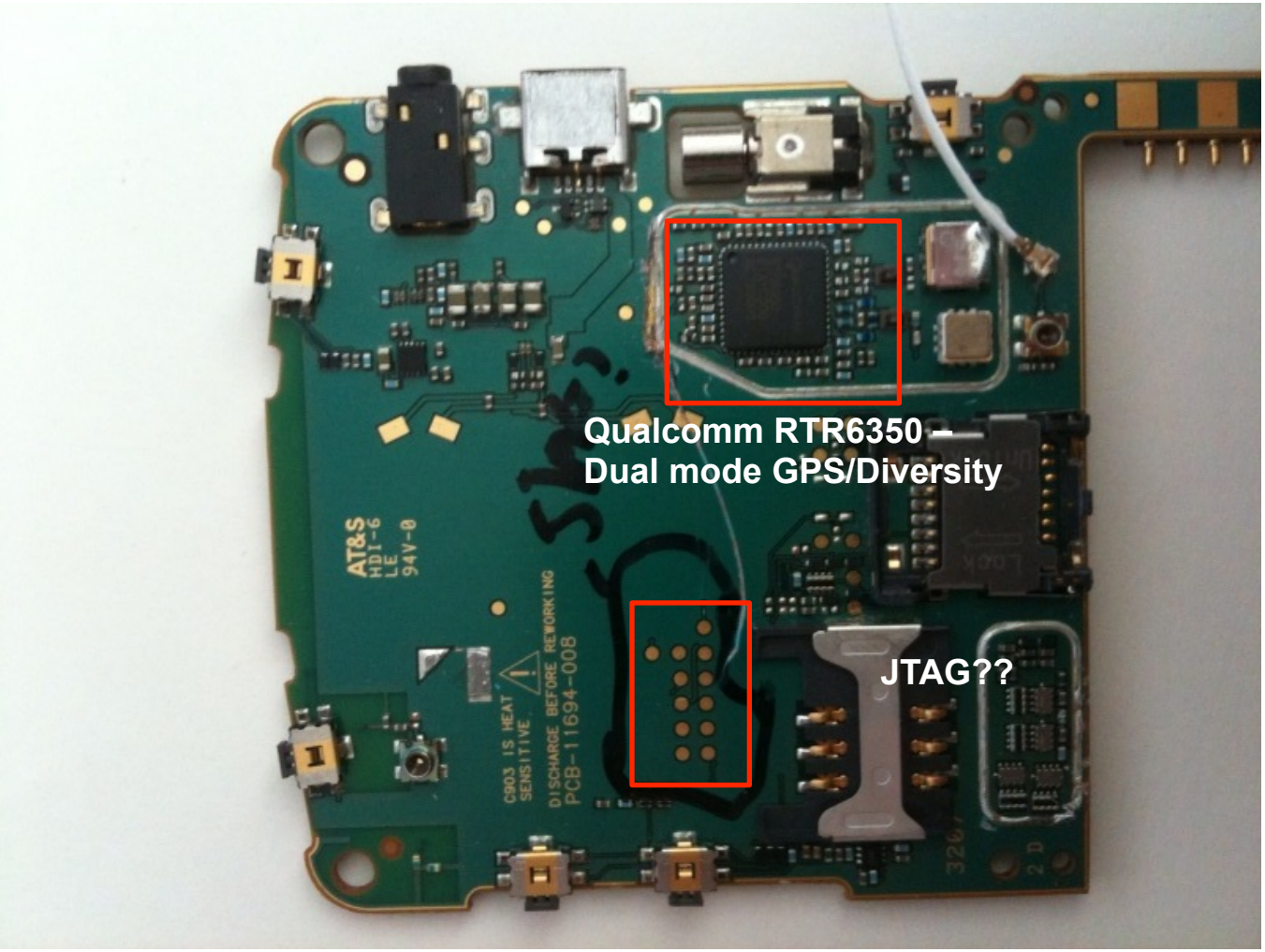
- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

Joint Test Action Group (JTAG)

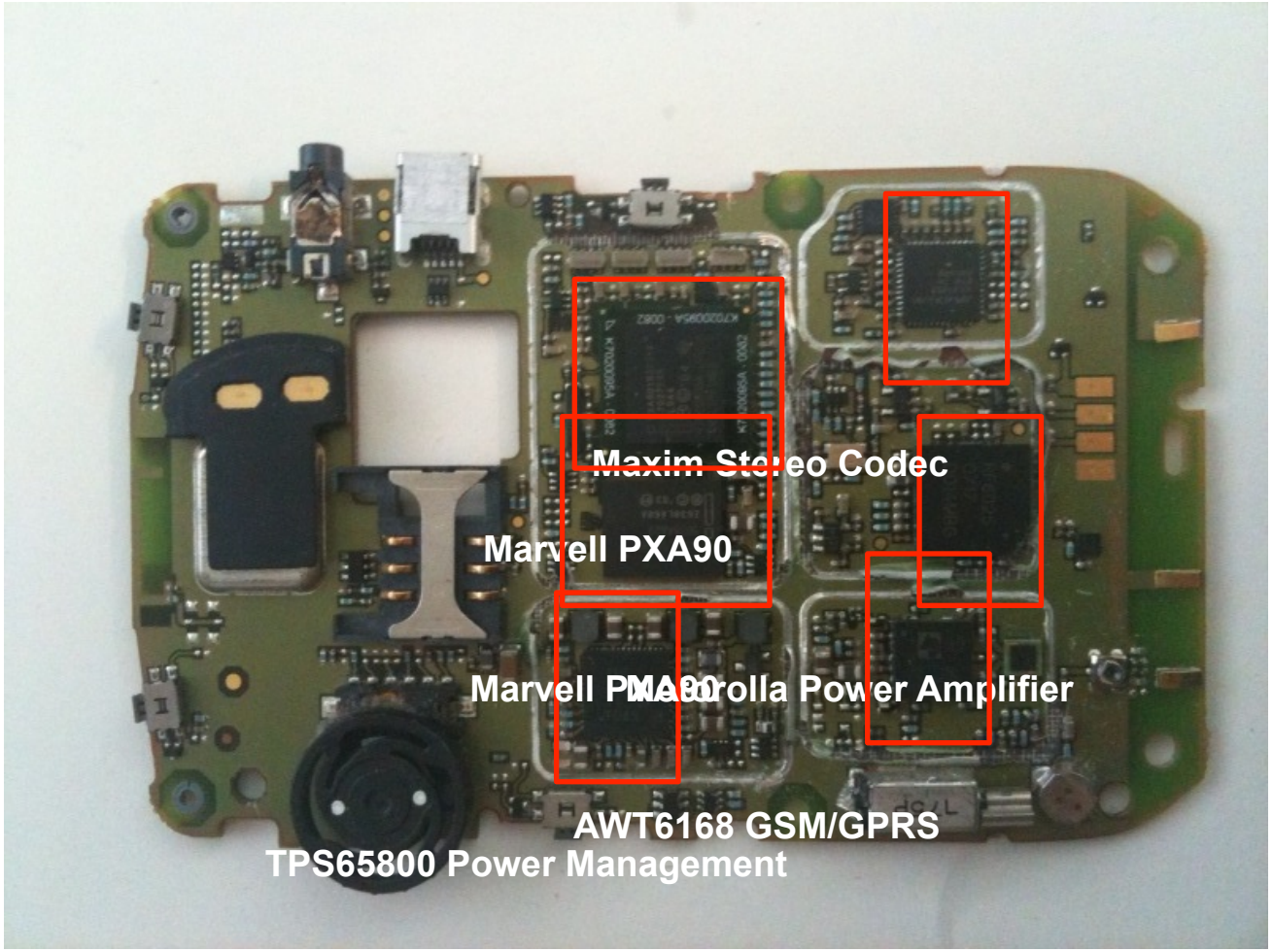
8830 PCB Front



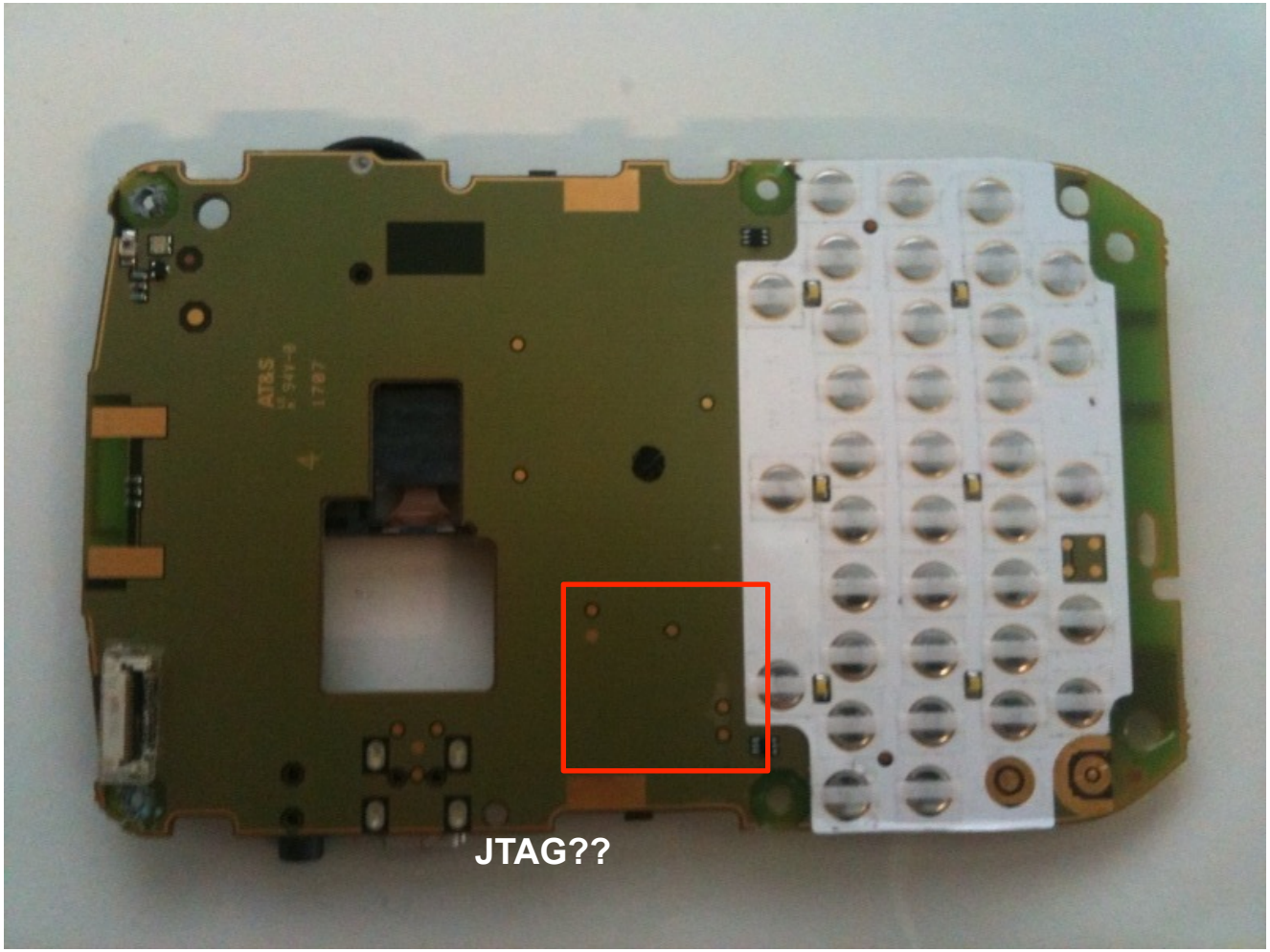
8830 PCB Back



8700 PCB Back



8700 PCB Front



JTAG Memory Dumping

:: MSL JTAG v.1.0.0 build 11 (pre-release 15/12/2008)

```
>HALT_AND_READ
Reseting TAP   : OK
Core IDCODE    : 0x200720E1, ARM926EJ-S, ARM
Halting Core   : OK (entry from ARM state)
Device WP      : OK (4 lines)
Device CPU     : MSM6800, Qualcomm.
CPU HW Rev.    : 0x200720E1 (known MSM6800 rev.)
Flash #1 ID    : 00EC:0036, Samsung K8F1208X0 NAND (1.8V), 8 Bit
MSM NAND CFG   : 512 MBit, 8 Bit, SP, R1, ECC
RAMBase Test   : OK
Current  PSR   : Supervisor Mode
>DUMP_NAND 0 0x80
Page #1, addr. 0x00000000, size 512 bytes
          0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000: 06 00 00 EA 38 00 00 EA 3E 00 00 EA 41 00 00 EA  ...Kj..K>..KA..K
00000010: 44 00 00 EA 47 00 00 EA 4A 00 00 EA 4D 00 00 EA  D..KG..KJ..KM..K
00000020: D1 F0 21 E3 00 80 A0 E1 01 90 A0 E1 02 A0 A0 E1  Cp!r.Ъ б.ђ б.  б
00000030: D3 F0 21 E3 CC 01 9F E5 00 10 90 E5 10 10 01 E2  Yp!rM.ue..ђe...в
00000040: 00 00 51 E3 09 00 00 1A 0D 00 A0 E1 0E 10 A0 E1  ..Qr...... б.. б
00000050: 0C 20 A0 E1 D1 F0 21 E3 00 B0 A0 E1 01 C0 A0 E1  .  бCp!r.' б.A б
00000060: 02 E0 A0 E1 D3 F0 21 E3 9C 01 9F E5 00 C0 A0 E1  .a бYp!rнь.ue.A б
00000070: 80 04 A0 E3 00 10 90 E5 50 19 C1 E3 00 10 80 E5  Ъ. r..ђeP.Бr..Ъe
Spare data:
          0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000200: 99 65 A5 99 55 FF C3 F0 FA 69 59 59 FF FF FF FF  ™eЕ™UяГръiYYяяяя
>FLASH_READ 0 0x100000 test
Reading NAND flash : chip #1, from 0x00000000, size 0x00100000
Flash read mode    : with loader, COMM = memory, buffer 256 Kb.
Sending  loader    : SENT
BAD blocks request : 1 BAD blocks: 0x26E8000
Output file        : 'C:\MSLJ\SavedMem\test.bin'
```

Read flash chip

flash_read 0 0x100000 test

Device
Name : Alltel UM150
CPU : Qualcomm MSM6800
RAM : 0x00100000 TCK R ..

on Halt
RAM base test Initialize (if TEST FAILED)

chip #1
Flash info
00EC:0036, Samsung
K8F1208X0 NAND (1.8V)
0x00000000, 0x04000000 (64 Mbyte)
GEO OK

Progress
Estimated time : ~ 0 min. 5 sec.
Stop running task

CORE : ARM926EJ-S | ARM | DEBUG | 000B:A000 | TCK : 2 MHz | SIR Err.: 0 | SDR Err.: 0

Winning with JTAG

- JTAG can provide access to bootloader
- JTAG can allow memory to be access from NOR/NAND
- JTAG may provide access to memory artifacts and other interesting things such as password hash

Agenda

- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

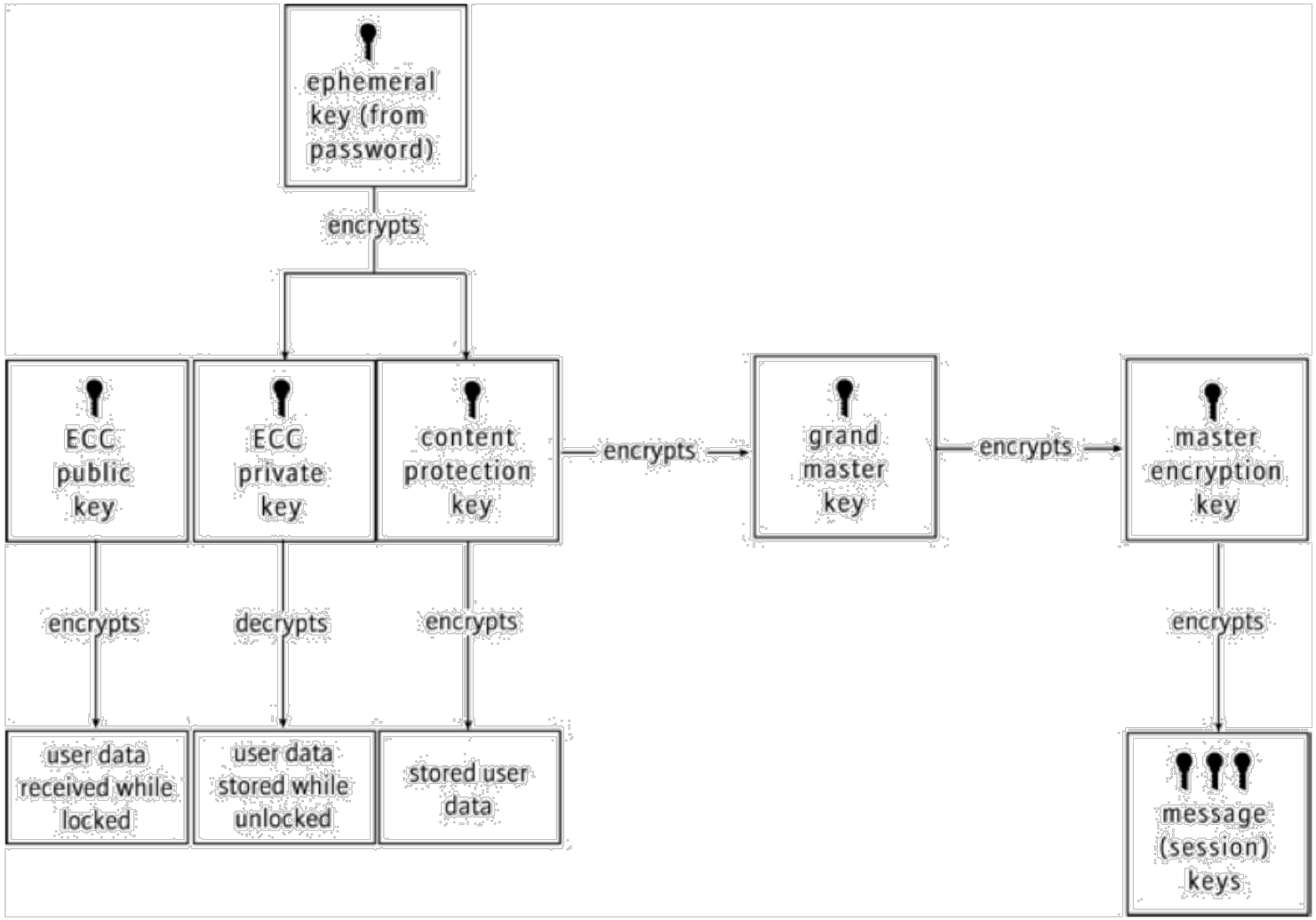
Boot Process

- BootROM code stored in flash memory
- BootROM is signed by RIM
- Processor can verify BootROM signature
  1) Processor runs ROM check code
  2) If ROM matches RSA public key then boot occurs if not then processor stops running

Crypto on Device

1) User enters password
2) Device derives ephemeral AES 256-Bit Key
3) Device uses ephemeral key to decrypt the content protection key and ECC private key
4) Decrypted keys moved to RAM
5) After 128 ECC encrypted items are accessed the device re-encrypts them with content protection key

Crypto Map


ephemeral key (from password)
encrypts
ECC public key
ECC private key
content protection key
encrypts
grand master key
encrypts
master encryption key
encrypts
decrypts
encrypts
encrypts
user data received while locked
user data stored while unlocked
stored user data
message (session) keys

Additional Security Features

- Message security
- Secure Memory Wipe
  - Default uses standard java garbage collection
  - Secure overwrites memory with 0x00 after collection

Agenda

- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

Cellular Phone Exploitation

- Variety of Operating Systems (OS)
  - Symbian
  - Iphone
  - Blackberry
  - Android
  - Windows Mobile (2.0, 2003, 2007)
  - Others
- Symbian and iPhone have been actively exploited

Attack Options

- BlackBerry Physical Access
- Cellular Communications Channels
- Browser
- Operating System
- MDS Runtime
- Social Engineering

Physical Access

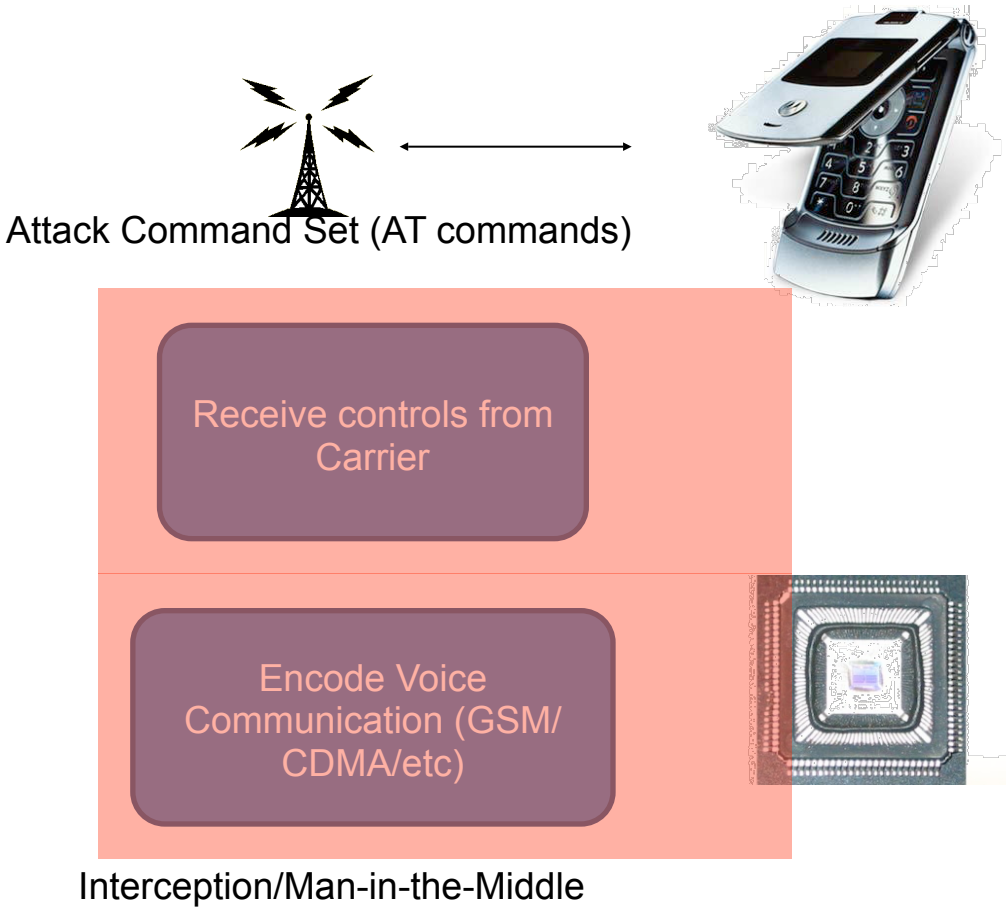
- Device Memory
  - Content protection (if enabled) uses 256-bit AES
    - Created using NIST approved PRNG
    - Derives key from password in accordance with PKCS #5
  - New data received while locked is encrypted using ECC public key of variable size
  - OS and non-protected data may be available
- External Storage (Media Card)
  - Blackberry supports external encryption

Device Forensics

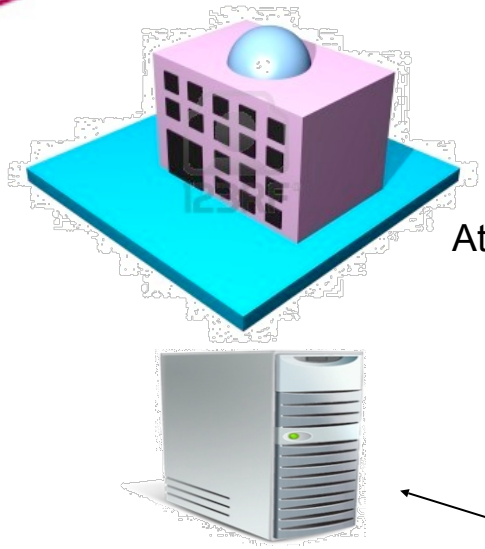
Application Databases (contacts, mail, etc)

Digital Storage (pictures, video, audio)

Cellular Phone Attack Surface



Blackberry Phone Attack Surface

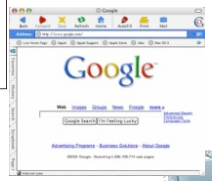


Attack Command Set (AT commands)

Receive controls from Carrier

Encode Voice Communication (GSM/ CDMA/etc)

Interception/Man-in-the-Middle



Browser

- Browser on mobile devices in general are subject to various attacks
  - Apple Safari PDF vulnerability = Jail Breaking
- Browsers can be targeted using social engineering for XSS attacks or remote exploitation
- Blackberry browser sucks (sorry)
- JavaScript does run on Browser so typical attacks using js can work
- Browser attack surface is very attractive for OS attack and JavaScript attacks targeting information, XSS

OS

- Blackberry OS is Java based, uses database to store application information and data
- Sensitive OS API are restricted by signed certificate to use
  - Blackberry Runtime API
  - Cryptographic API (includes certicom)
  - Blackberry Application API
- Certificate can be acquired at: https://www.blackberry.com/SignedKeys/
- 20 dollars

OS Permissions

- OS maintains permissions for all non-rim applications
- Permissions are broken into three categories
  - Connections
  - Interactions
  - User Data
- Permissions are Allow, or Deny
- New applications can be loaded via USB using Blackberry Desktop or OTA (Over The Air)
- Trusted Access?
- Default permissions - Tyler Shields - Veracode

OS

- Its Java
- OS handles Garbage Collection
- OS is optimized for speed and developers are encouraged to design using similar techniques
- Exposed API is limited

MDS Runtime

- Not installed by default
- Not very common
- Phasing out MDS on the device

Agenda

- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

Forensics Tools

- Paraben (Commercial)
- Amber BlackBerry Converter (Freely available)
- BlackBerry Desktop Manager
- Java_Loader.exe
- Others?

Wiping a Device

- Wiping does not remove 3rd party or additional software ‘cod’ files
- Reflashing the OS provides complete OS reload
- Wiping overwrites data stores with ‘0x00’

Agenda

- Introduction/Background
- Enterprise Architecture
- Device Architecture
- Hardware Overview
- Cryptographic Overview
- Attack Surfaces/Methodologies
- Forensics/Anti-Forensics
- Enterprise Security

IT Policy

- Blackberry Enterprise pushes “IT Policy”
- Template (think GPO) for device security
  - set default application permissions
  - set password enforcement
  - network settings

Conclusion

- Blackberry is extremely well architected to be secure
- Malicious tools do exist for remote access
- Social Engineering provides possible attack vector
- Exploitation is not easily accomplished
- If you have an AES decoder ring then nothing can be done
- Blackberry Enterprise Architecture needs to be carefully monitored and protected or all bets are off

Best Practices

- Secure and Enclave Servers
- Ensure Content Protection is enabled
- Generate Strong IT Policy
- Change the SQL password on the server
- Security Technical Implementation Guides
  - http://iase.disa.mil/stigs/content_pages/wireless_security.html

Sources

- http://www.blackberry.com/solutions/resources/Placing_the_BlackBerry_Enterprise_Solution_in_a_Segmented_Network.pdf
- http://docs.blackberry.com/en/admin/deliverables/12058/Using_a_segmntd_network_to_prevent_malware_855099_11.jsp
- http://docs.blackberry.com/en/admin/deliverables/3940/file_encryption_STO.pdf