MWR InfoSecurity

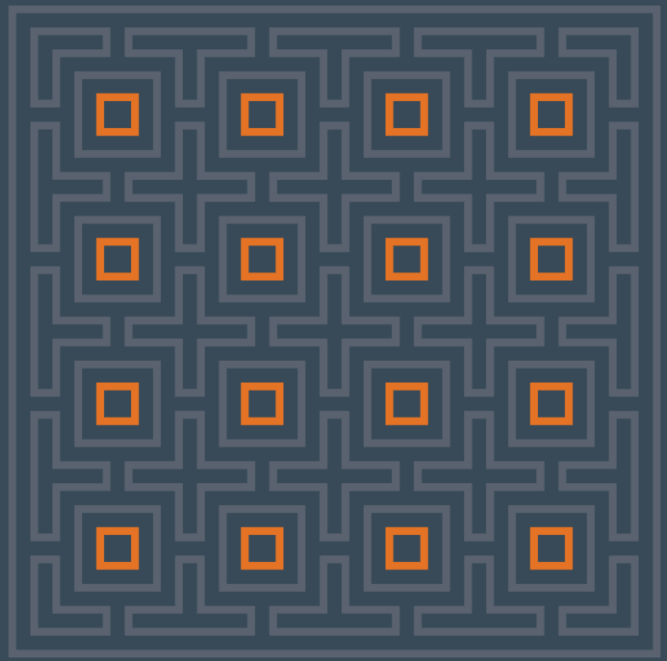
Public

EXTERNAL

Is Blackberry Dead?

An Introduction to Blackberry 10 Security (BB10 - QNX)

Alex Plaskett - 2013



Introduction

- Technical Research Into BB10 (QNX Platform)
- Application Security Features
- Enterprise Features Introduction

Blackberry Background

Worldwide Smartphone Sales to End Users by Operating System in 4Q12 (Thousands of Units)

Operating System	4Q12 Units	4Q12 Market Share (%)	4Q11 Units	4Q11 Market Share (%)
Android	144,720.3	69.7	77,054.2	51.3
iOS	43,457.4	20.9	35,456.0	23.6
Research In Motion	7,333.0	3.5	13,184.5	8.8
Microsoft	6,185.5	3.0	2,759.0	1.8
Bada	2,684.0	1.3	3,111.3	2.1
Symbian	2,569.1	1.2	17,458.4	11.6
Others	713.1	0.3	1,166.5	0.8
Total	**207,662.4**	**100.0**	**150,189.9**	**100.0**

Source: Gartner (February 2013)

Blackberry 7

BB7 Features

- Blackberry Proprietary OS
- Java Applications
- CESG Approved (RESTRICTED)
- No Modern Exploit Mitigations – DEP/ASLR (pwn2own 2011)
- Never publically rooted

Blackberry Playbook

Playbook Features

- QNX (6.5 sp1) based
- Rooted via Samba (Dingleberry)
- Backups were unsigned!

Blackberry 10

BB10 Features

- QNX 8.0
- Playbook Similarities
- Not rooted (yet.. ☺)

theguardian

News | US | World | Sports | Comment | Culture | Business | Money

News › Technology › BlackBerry

BlackBerry software ruled not safe enough for essential government work

CESG rejects BB10 software in new Z10 handset, dealing blow to Canadian firm in key market

CESG's response:

"Discussions with BlackBerry are ongoing about the use of the BlackBerry 10 platform in government. ***We have not yet performed an evaluation of the security of the platform****, but we expect to be issuing Platform Guidance in the summer. This will cover a number of platforms, including BlackBerry 10 (and the use of 'Balance').*

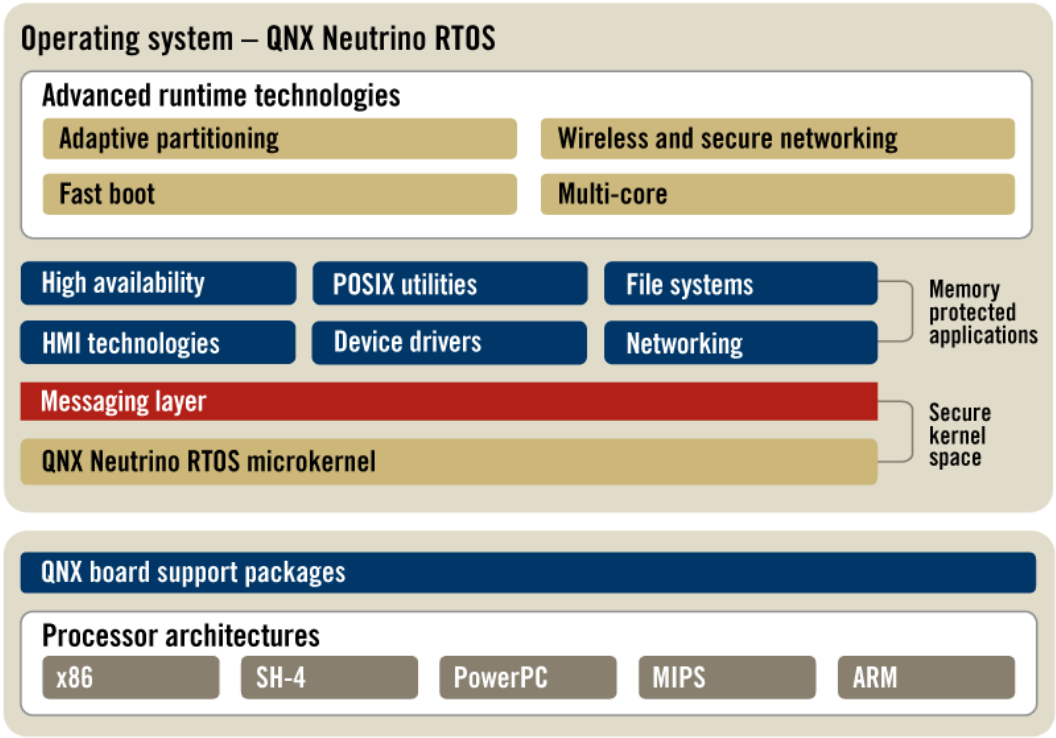
We have a long-standing security partnership with BlackBerry, and this gives us confidence that the BlackBerry 10 platform is likely to represent a viable solution for UK Government."

QNX Architecture

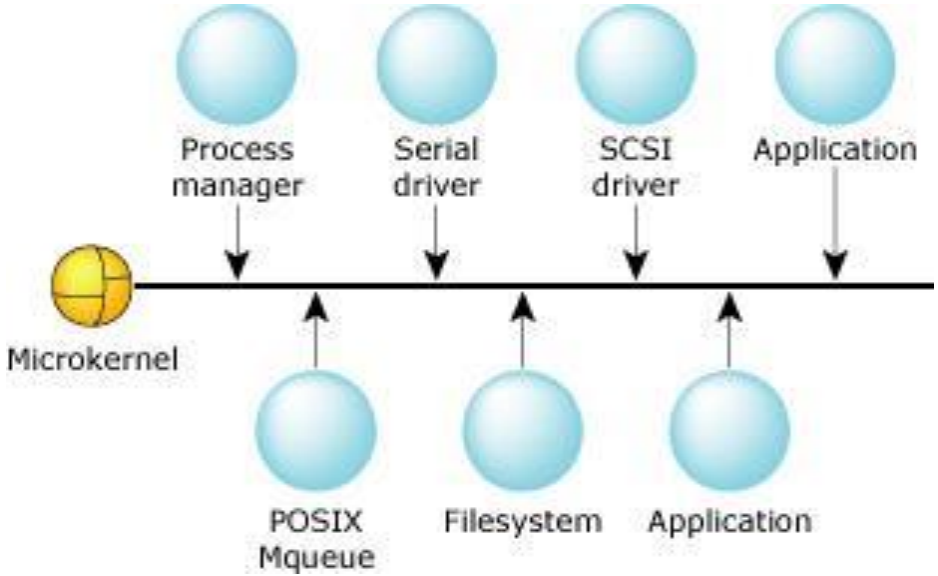
QNX History

- First version released in 1982
- Late 1980's largely rewritten (QNX 4.0)
- 2007 QNX released its source code (QNX 6.*)
- 2010 RIM acquired QNX Software Systems (Source Code access restricted)
- 2013 BB10 (QNX 8.0) released

QNX Architecture



QNX Message Passing



How this actually works

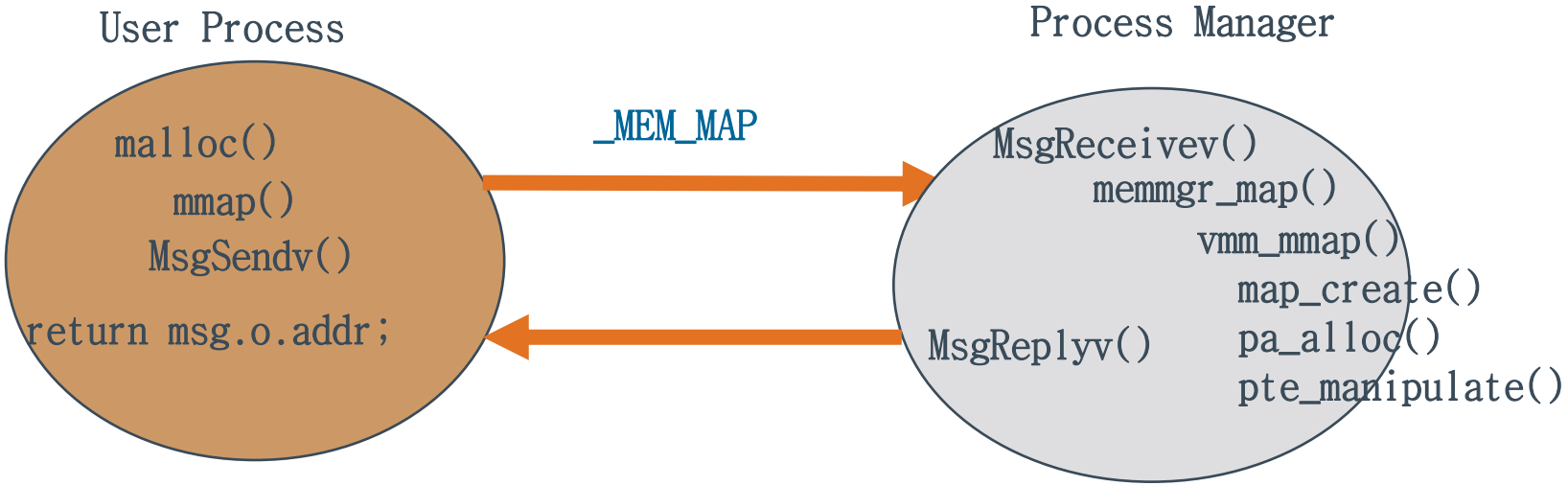
```
mmap(void *addr, size_t len, int prot, int flags, int fd,
off_t off)
{
..
        MsgSend(MEMMGR_COID,..);
}
```

Syscall Transition

```
LOAD:000483A0 MsgSend     proc near               ; DATA XREF: ...
LOAD:000483C3             jz      short loc_483D8
LOAD:000483C5             mov     ecx, esp
LOAD:000483C7             sysenter
```

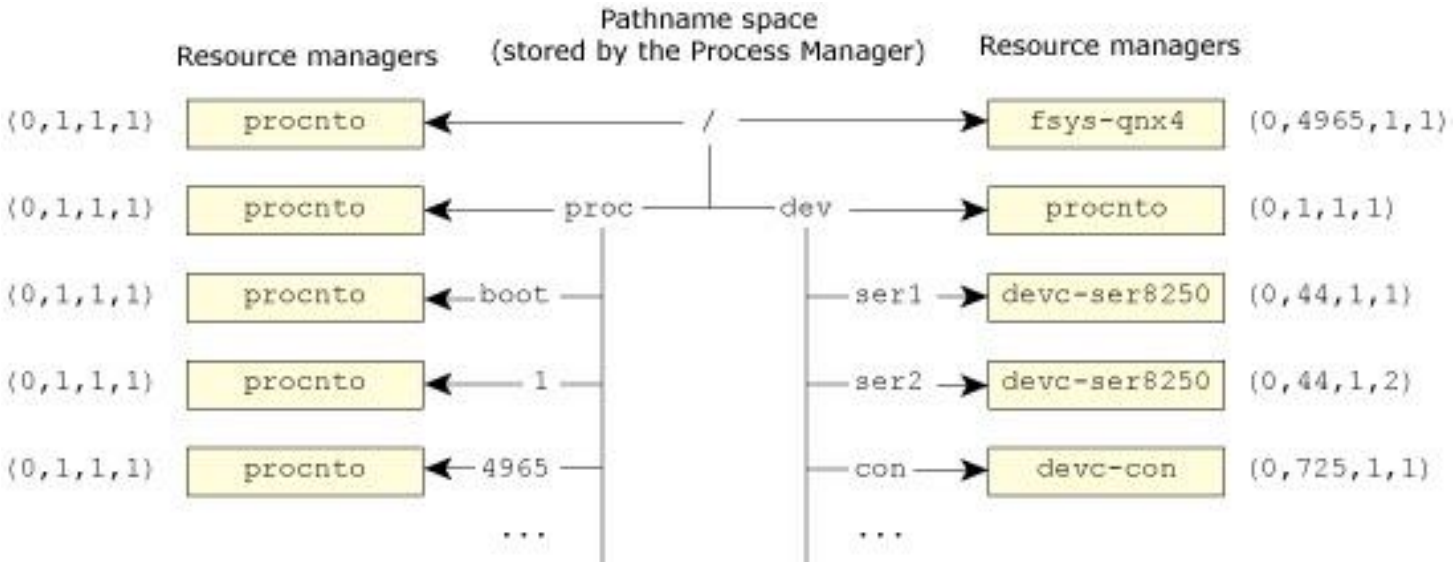
Resource Managers

\target_10_1_0_1020\qnx6\usr\include\sys\memmsg.h



community.**qnx**.com/...Micro**kernel**.../Webinar_**kernel**_oct07_final.ppt

Process Manager

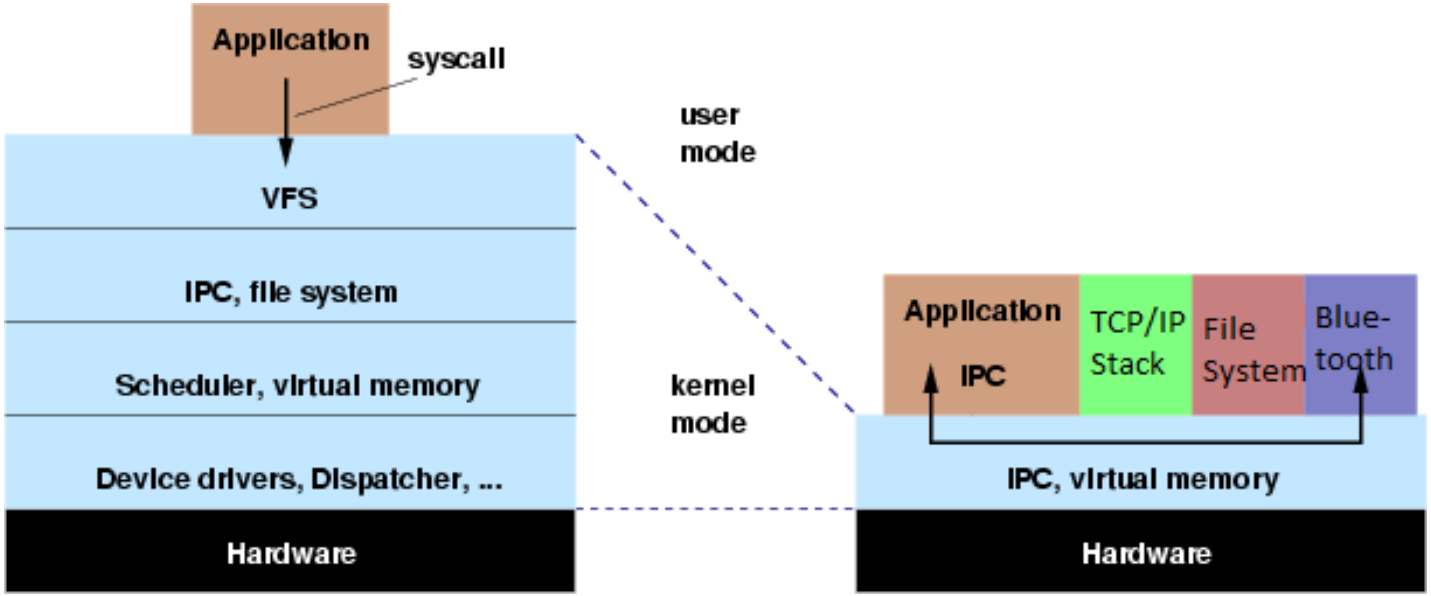


Resource Managers

- Resmgr_attach – Register for path in pathname space
- iofunc_func_init – Initialize the POSIX-layer function table
- Message_attach – Attaches a handler to a message range

Kernel Comparison

Monolith Kernel (Android) Microkernel (QNX)



Microkernel Security Advantages

- **Minimal Size of Trusted Computing Base (procnto)**
- **Principle of Least Privilege**
- **Division of Responsibilities**
- **Fault Tolerant**

Simulator Process Listing

procnto-smp-instr	0	0	0	0	0	0
devc-con	0	0	0	0	0	0
pci-bios	0	0	0	0	0	0
slogger2	26	25	26	25	26	25
slogger	25	25	25	25	25	25
pipe	36	36	36	36	36	36
devb-eidge	132	132	132	132	132	132
m_resource_manager	0	0	0	0	0	0
pps	339	86	339	86	339	86
perimeter_mgr	0	0	0	0	0	0
authman	0	0	505	0	0	0
ves-server	99	0	99	0	99	0
battmgr	0	0	0	0	0	0
io-usb	0	0	0	0	0	0
dumper	27	412	27	412	27	412
io-hid	0	0	0	0	0	0
devc-ser8250	0	0	0	0	0	0
screen	0	0	0	0	0	0
drmclock	0	0	0	0	0	0
random	0	0	0	0	0	0
coreServices2	0	32	31	32	0	32

But what does this actually mean?

- Microkernel attack surface minimised (77 syscalls vs Linux 338)
- However, 27+ processes running as EUID 0 (root!)
- Already found some issues (kernel panics)
- **Cross-Process messaging based attacks in future?**

Application Security

MWR InfoSecurity

BB10


C/C++ Native SDK


C++/Qt Cascades

HTML5/WebWorks


Adobe Air


Android Java Runtime

BB7

Java

BB10 Application Security

- Application Sandbox
- Application Code Signing
- Application Permissions

Application Sandbox

- Applications are installed into /apps/
- Apps cannot read another applications code (/apps) or data (/accounts/*/appdata).
- OS permissions and Authman enforce this

Application Sandbox

	BB10	iOS	Android	WP8
IPC	Yes	Disallowed	Yes	Disallowed
URL Handlers	No * (built in ones)	Yes	Yes	Yes
File Handlers	No	Yes	Yes	Yes

Application Code Signing

- Applications need signed before they can run on a BB10 device
- Developer devices can side load and run unsigned code using a debug token.
- Blackberry World used for distribution

Code Signing Comparison

	BB10	iOS	Android	WP8
Unsigned Code	Debug Token (Free)	Dev unlock (Non-Free)	Yes	Dev Unlock (Non-Free)
Mandatory Application Code Signing	Yes	Yes	Yes (but self-signed is allowed!)	Yes

Code Signing Differences

- **QNX executable binaries do not require code signing for devuser or a debug token**
- Possible to SCP these to BB10 device
- Anyone can dev unlock a device and do this (free).

- Useful for testing local exploits / jailbreaks! ☺

Blackberry World Communication

- Downloads applications in Plaintext HTTP
- Applications are integrity checked (code signed)
- However, applications are not encrypted / obfuscated (iOS and WP8 are)
- Reverse engineer!

Application Permissions

- Security and privacy critical functionality
- User prompted on installation
- Developers specify permissions in the bar manifest

<permission>read_device_identifying_information</permission>

Application Permissions (Unique Feature)

Application Permissions Implementation (MAC)

- /etc/authman/sys.res

use_camera:

prompt *

allow sys.*

Allow – means the identified apps can use the permission

Deny – means that capability cannot be used by the app

Prompt – means that the app must prompt the user first

Application Permissions Implementation (MAC)

- /etc/authman/sys.acl

```
use_camera
        MAC macro_access_camera_service

macro_access_camera_service
        ACL rw /dev/camera/front1
        ACL rw /dev/camera/rear1
```

Enterprise Security

BES 5

- Only supports BB7
- Granular device policies
- Blackberry attachment service (vulnerabilities)

BES 10

- Supports Android, iOS, BB10 MDM
- Not backwards compatible with BB7
- Can be installed on the same server as BB5 potentially..

Blackberry Balance

Personal

Work

Blackberry Balance

- Creates separate user accounts, groups and data stores for work data
- Separate ACL for Authman (/etc/authman)
- Even restricts copy / paste
- Classifies data based on source
- Remote wipe of only work data

Blackberry World for Work

- Enterprise Application Store
- Approved Work Applications / Company Applications

Data Protection

- Device Encryption (XTS-AES-256)
- SD Card Encryption
- Application Crypto APIs

Conclusions

- Early Days
- Large number of security controls implemented
- QNX architecture weaknesses?

Questions?