Windows Phone 7 OEMS – Microsoft BlueHat Executive Briefings

Alex Plaskett – November 2011

Main Objectives

- High level overview
- Demonstrate that OEM's negatively impact the security posture of phones
- Provide independent viewpoint on security
- Provide thoughts for future

Introduction

Who am I?

- Security Consultant @ MWR InfoSecurity
- Presented on WP7 at 44con, T2 etc..
- Breaking stuff for fun for a while ☺

What this talk will cover

- OEM Features and Risks
- OEM Vulnerabilities
- Future Thoughts

Platform OEM Comparisons

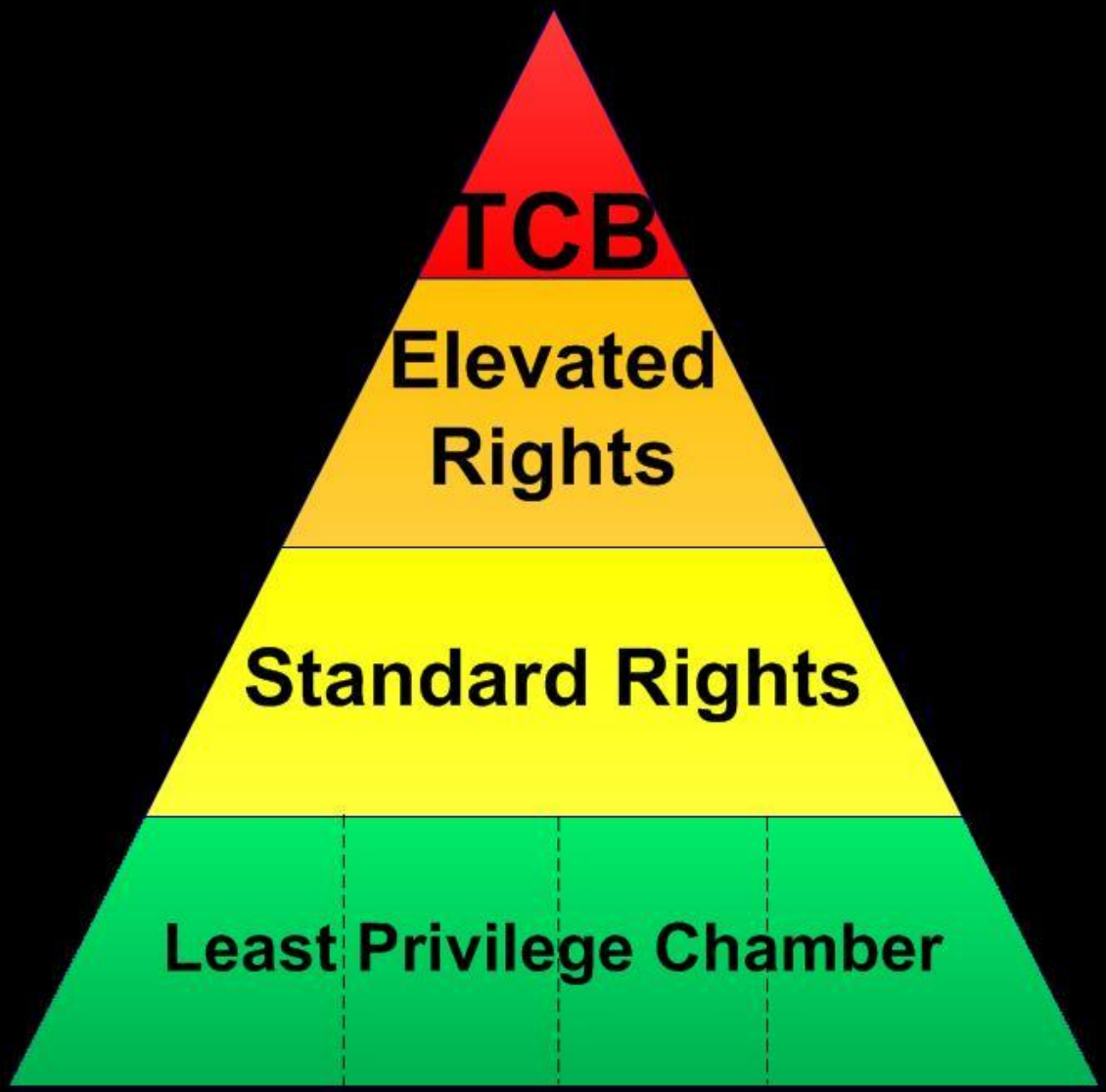
- WP7: HTC, Samsung, LG, Dell
- Android: Acer, HTC, LG, Motorola
- iOS: Apple
- BBOS: RIM

⇒ WP7 and Android: greater attack surface/more complex security ecosystems

WP7 Security Model

- Process Sandbox
- Code Signing
- Centralised Security Policy
- Exploit Mitigations

Chamber Based Security Model



WP7 OEM Features

Third Party Development	OEM/MO Development
Managed Code Only	Managed + Native Code
User space Only	User space and Kernel
LPC Sandbox Applications	Up to high privilege chambers
No accessible services	Globally Accessible Services

WP7 OEM Risks

- Vulnerabilities in OEM Apps or Drivers
- Privileged Application Functionality
- Extra Delay in Patching OEM Code
- Vulnerabilities in OEM code misattributed to MS vulnerabilities?

Vulnerabilities

Other Platform OEM Vulnerabilities

- Android

HTC Browser INSTALL Permissions

HTC Sound Recorder

HTC Logger

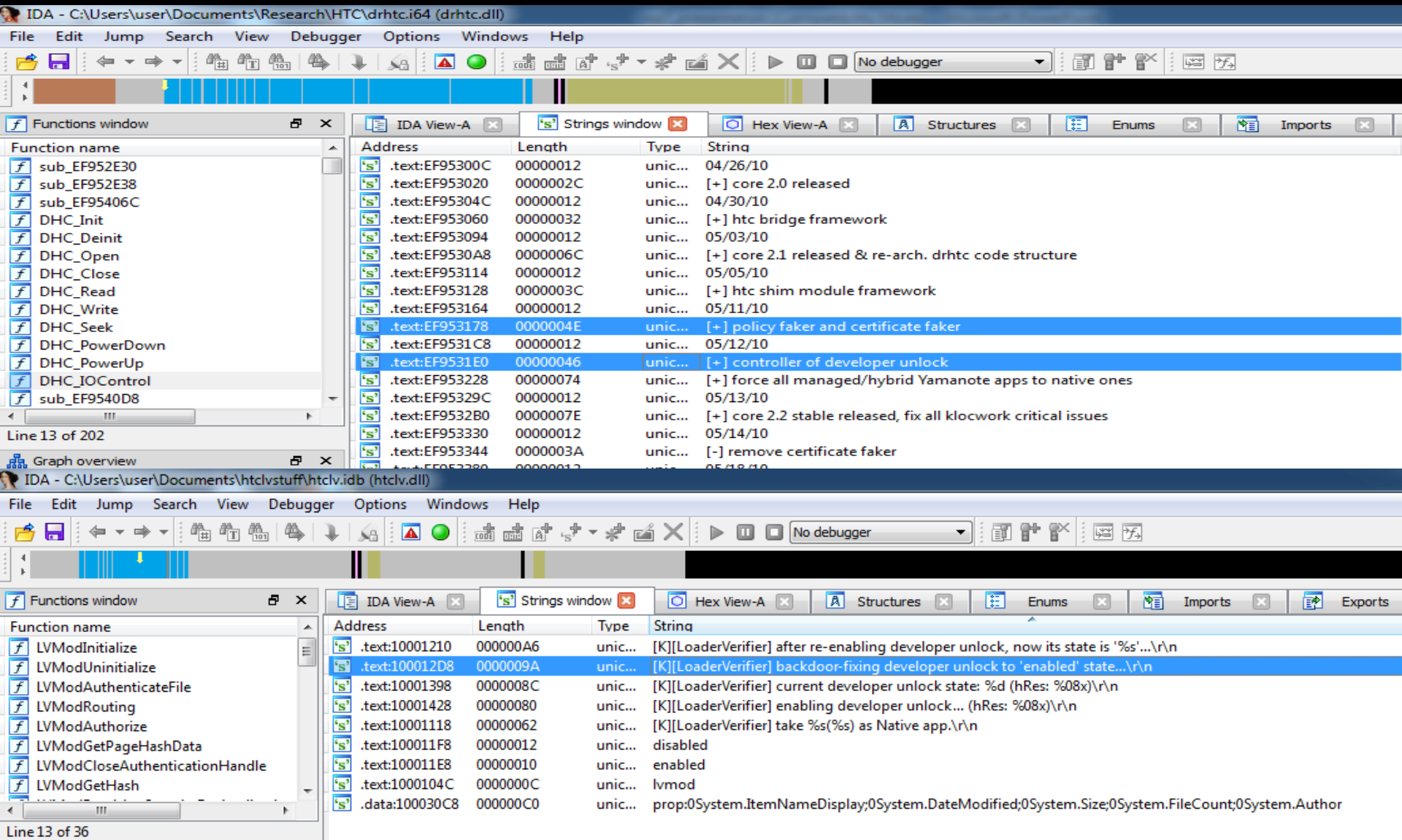
- iPhone / BlackBerry:

N/A

WP7 Potentially Dangerous OEM Functionality

- Samsung Diagnostic Application
- LG MFG Application
- HTC Debug Code

Concerning OEM Code



WP7 OEM Vulnerabilities

- HTC Kernel Arbitrary Read/Write
- Samsung PROVXML Privilege Escalation

Browser Exploitation

- Samsung Diagnostic Application For Debugging
- Samsung PROVXML Vulnerability For File System Access

=> Not Directly Using OEM Vulnerabilities

- Browser lacks ID_CAP_INTEROPSERVICES

Demo

Identified Problems

- Gap between MS and OEM code quality
- OEM’s introduce dangerous features to offer customers / internal developers extra functionality at the potential expense of security
- MSFT gets blamed for OEM mistakes?

Future Thoughts

Mango and onwards

- Restricts method I used to debug and develop exploits against the platform (ID_CAP_INTEROPSERVICES) and new web browser.
- However, design and policy still allows OEM applications to use driver functionality
- OEM code could still expose MS to an unnecessary level of risk

Discussion Points

- Better Integration between MS SDL with OEM's Development?
- More granular permissions for OEMs – Provide secure APIs for OEM requirements?
- Does MS have oversight in what the OEM's are shipping?
- More stringent controls on what OEM's ship?

Conclusions

- Strong Granular Security Model
- Attackers need multiple vulnerabilities
- MS needs to motivate OEM’s to deliver better code
- Attackers could use OEM vulnerabilities

Questions?