Secure Firmware Considerations
Problem

- Firmware updates secured by keys
- Tools use these keys to prevent unauthorized updates
Solution

• **Use** strong RSA keys for firmware updates
• **Safeguard** keys and firmware source code
• **Consider** using more secure methods of updates (e.g. UpdateCapsule)
• **Enable** customers to verify compliance
Background

• Developments in the ecosystem require PC system designers to review system security
• Attacks more likely with increased scrutiny into Secure Boot
• Products, processes, & even factory require thorough security
• Security vulnerability impacts all OS’s (Windows, Linux, etc…)
• Trusted Boot impacted by vulnerability in early phase of boot (UEFI SEC/PEI phase).
• Vulnerability is not UEFI-based
Windows Hardware Certification Requirement

• Systems shipping with non-production keys in firmware are in violation of the WHCR for Windows 8

• “The firmware update process must also protect against rolling back to insecure versions, or non-production versions that may disable secure boot or include non-production keys.”
Firmware Design Principles

• Secure Boot requires secure flash
• System Management Mode (SMM) tools use System Management Interrupt to manage NV-flash
• NV-flash is traditionally secured by System Management Mode which verifies update keys (OEM, ODM, IBV)
• Not the same as Secure Boot Keys (e.g. PK, KEK, DB/DBX)
• Proprietary tools use keys to enter into SMM, granting unrestricted access to NV-flash
Infecting Firmware

Normal Power-on procedures:
Code from NV-flash clean, at factory settings, secured with strong keys

Infected Power-on procedures:
Malware injects and infects NV-flash with malicious firmware & possibly new keys
Methods of Injection

• A customer is lured into clicking on a link to a system firmware update, upon downloading the link the desktop app executes SMI calls with known keys that enables it to modify NV-flash

• A physically present attack would be possible by inserting removable media with a rogue boot loader initiating a code injection of malicious code into NV-flash
Mitigation & Remediation

• No firmware ‘backdoors’ for any purpose
• Use PKI-based keys for platform support tools
• Update tools on the factory floor ASAP
• Notify customers of the need to update firmware
• Publish test key discovery tools
• Use HSMs or contact your BIOS partner for keys
Software bugs are expensive for everyone...

Code fixes performed *after release* can cost up to 30 times more than fixes performed during the design phase.

**Relative cost to fix, based on time of detection**

- **Requirements / Coding**
- **Integration / Component**
- **System / Acceptance**
- **Production / Post-Release**

Source: National Institute of Standards and Technology

*Recent Aberdeen Group Research Brief (Dec 2010) estimates $300k avg. cost per vulnerability*
Common Misconceptions about SDL

“...only for Windows”
  – Appropriate for non-Microsoft platforms
    • Microsoft is a huge Macintosh ISV...
  – Based on proven, generally accepted security practices

“...for shrink-wrapped products”
  – Also covers Line of Business (LOB) and online services (Cloud) development

“...for waterfall or spiral development”
  – Agile methods are also supported

“...requires Microsoft tools”
  – Use the appropriate tools for the job – no “rip & replace” required.

“...requires Microsoft-level resources to implement”
  – SDL as its applied at Microsoft != SDL for other development orgs.
Simplified Implementation of the Microsoft SDL

• 170+ pages of Microsoft SDL guidance reduced to 17 pages and 16 practices
  – Non-proprietary
    • Creative Commons License
  – Suitable for organizations of any size
  – Platform agnostic
  – Mapped to well known compliance regs (PCI, HIPAA, PRINCE2)
  – Core elements based off the SDL process used at Microsoft
  – Holistic – Not the typical “list of lists” approach common to other methodologies

• Since April 2008
  – SDL Guidance: Over a quarter million downloads
  – SDL Tooling & Automation: Over a half million downloads
Resources

SDL Portal
http://www.microsoft.com/sdl

SDL Blog
http://blogs.msdn.com/sdl/

SDL Process on MSDN (Web)

Simplified Implementation of the Microsoft SDL
http://go.microsoft.com/?linkid=9708425
Further Reading

• NIST 800-147, 800-147b

• FIPS 140-2

• CANSEC Presentation

• National Vulnerability Database
  http://nvd.nist.gov/