

New Protection Profile Efforts

USB PP and OS PP

22 September 2010

USB Protection Profile

USB PP Goals

- Start with a baseline set of requirements that are objective and achievable
- Address Two Use Cases:
 - Transfers
 - Storage

PP Goals

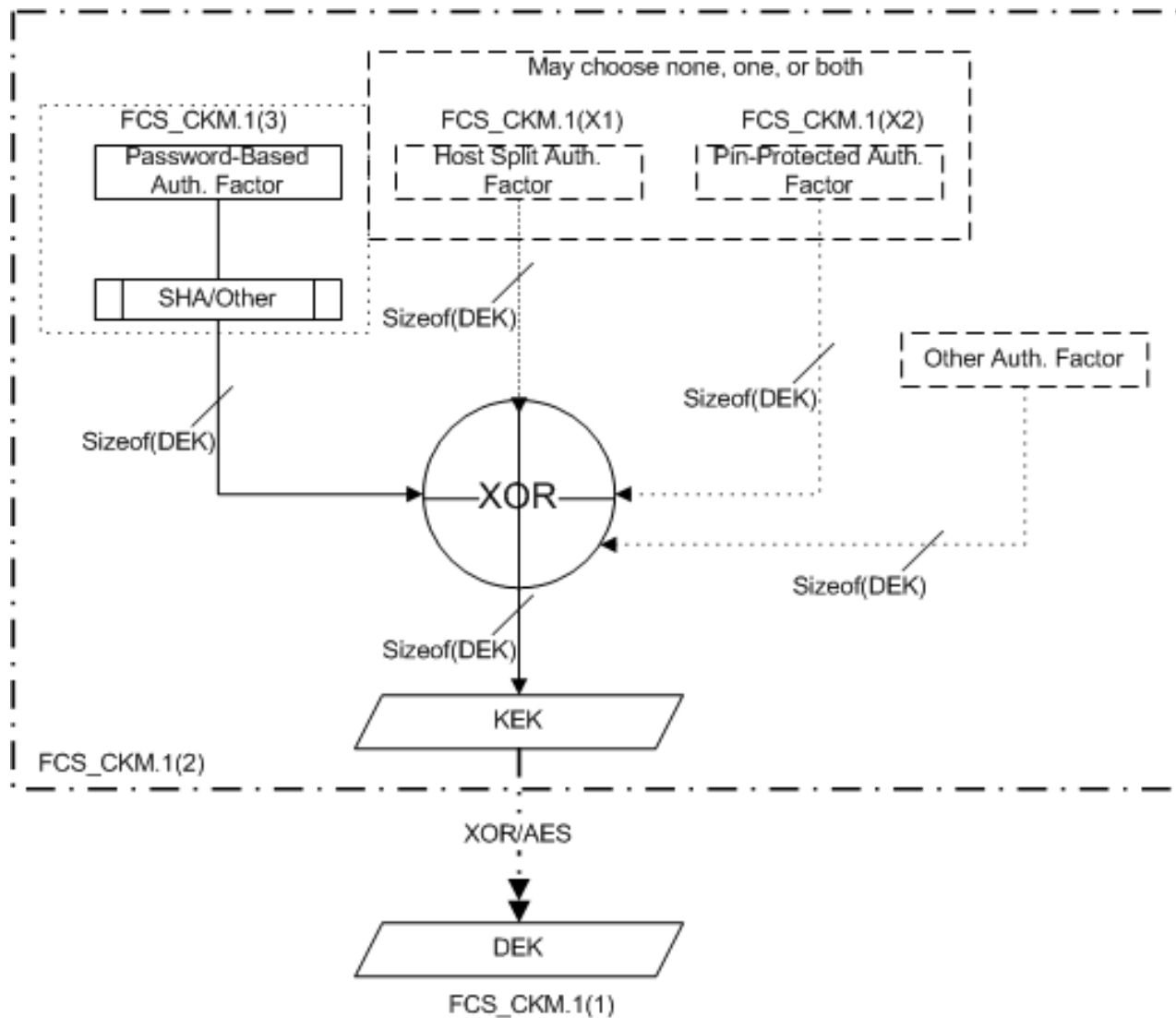
- Threat Scenarios
 - “Lost” USB
 - Autoplay attack
 - Malicious update
- Incorporate “New” Features
 - Assurance Activities
 - Appendix C

Requirements

- Cryptographic
 - Key (DEK, KEK) creation,
 - Signing (update function)
- Protection of data on device
 - Encryption for user data
 - Firmware protects “system files”

Requirements

- “Trusted” Updates
- Authorization before use
- Self-Test



“New” Features

- Assurance Activities
 - Set a minimum bar and tailor the CEM
 - Per-SFR activities
 - Cryptographic activities explicitly specified
 - Transparent objective activities are the first step towards mutual recognition

“New” Features

- Assurance Activities
 - Requires information in TSS
 - More information than we typically see
 - Focused on SFRs
 - Per-SAR activities
 - Testing
 - Guidance Documentation
 - Eventual notion of tying SARs to Threats

“New” Features

- Appendix C
 - Functions to “augment” baseline in PP
 - Used by ST writers
 - Constrains allowed functionality
 - Additions must be approved

“New” Features

Appendix C

– Additional Cryptographic requirements

- **Certain functions must be implemented by TOE**
 - Generation of DEK
 - Checking signatures for updates
- **Others are optional**
 - Creation of KEK
 - Wrapping/Unwrapping DEK
 - Implemented by TOE...move to body of ST
 - Implemented by OE...Create “Requirements on OE” section

Next Steps

- Already been through one review by vendors
 - Comments were incorporated, ready for second review
- Has been distributed to the CCDB for comment
- Additional functionality, assurances, tailored vulnerability assessment



OSPP: A New Approach

Security vs. Assurance

- Traditional high EALs can no longer be achieved on a common general purpose OS
- EAL 4 has not been achievable, repeatable, or comparable for some time
 - EAL escalation, everyone expects a 4!
- Actual exploits and vulnerabilities are either due to third party applications and drivers or in an obscure area of the OS

Security vs. Assurance

- Design review, code review, and pen testing are very costly and time consuming and yield minimal effect
- Finding a handful of vulnerabilities is doing very little for the overall security posture of the OS, or the OS technology landscape

Security vs. Assurance

- The major Operating Systems are some of the most exposed and attacked applications in the wild
- To expect a couple evaluators to make a greater impact than the application vendor space, hacker community, and OS vendor endorsed testing efforts is unreasonable and ineffective

Security vs. Assurance

- Thinking more in terms of security and available OS security services can allow us to devote attention to providing a more “disciplined” application environment

Security vs. Assurance

- Published APIs and available OS security services
- Visual Studio, GCC linker and compiler options

OS Services

- Cryptographic support
 - CAPI
- Network authentication and identification
 - Schannel/TLS.
- Remote Procedure Call (RPC)
- ASLR, DEP, Integrity Check
 - core feature rather than traditional service

OS Security guidance/evaluation

Where can we add value:

- Secure Configuration (SCAP)
- Attestation against Top 25 CWEs
 - Development activities
- DEP, ASLR, Canaries
 - Data Execution Prevention (HW, SW)
 - Exception handling
 - Address Space Layout Randomization
- Basic functional testing
- Detailed service module specification

Security vs. Assurance

- This approach begins recognition and mitigations of the real problem:
Composition
- This has always been ignored in the CC and real assurance is becoming impossible without some notion of composition assurance

Application Security

- DEP, ASLR, Canaries
- Attestation and minimal testing that core OS security services were utilized
- Where deviations occur, the application vendor will be required to provide a robust rationale

Questions/Guidelines

- Describe the threats the service is intended to counter
- Describe what the OS developer should do to design and implement the service
- Describe what the application should do to leverage the service
- Describe anything IT administrator would need to do to configure the security service
- Describe what a hardware developer may need to do to support the security service
- Describe what an acquisitions official may need to consider when making a decision to purchase the operating system or application
- Consider what evidence and activities an evaluator would need to validate the OS service
- Consider what evidence and activities an evaluator would need to validate the application

New Security

- NIAP evaluation provides the incentive and hammer to require applications to utilize the robust feature set of an OS
- Needs to be specific to a particular OS and a particular application
 - Eventually can require all driver signing
 - Application ‘logo’ testing
 - Full SCAP
 - CWE mitigations

Thank You

Questions