SONY

Software component evaluation

A developer's perspective

Sony Corporation's presentation for the
6th International Common Criteria Conference

Contents

SONY

- The discussion of the modularity of systems versus the modularity of software
- The discussion of disparity between the hardware and software evaluations
- The discussion of the complexity of software and components
- A preview of what is coming

System security

SONY

- Current approach is to evaluate “finished products”
- Those products are later used as components to build secure systems
- Security of the system depends more on the overall security policies and system design than on the product security
- Products specify how they should be used to remain secure even inside a system
- Most of the time, systems are not externally evaluated

Product security

SONY

- Products are seen as almost completely independent units
- Products are rigorously evaluated for their security
- Product is seen as a big lump of matter, thus contradicting the design principles of the product
- The consequence is that the effort is wasted on reinventing the wheel
- Product security should be looked at from the point of view of a system and components that go into this system

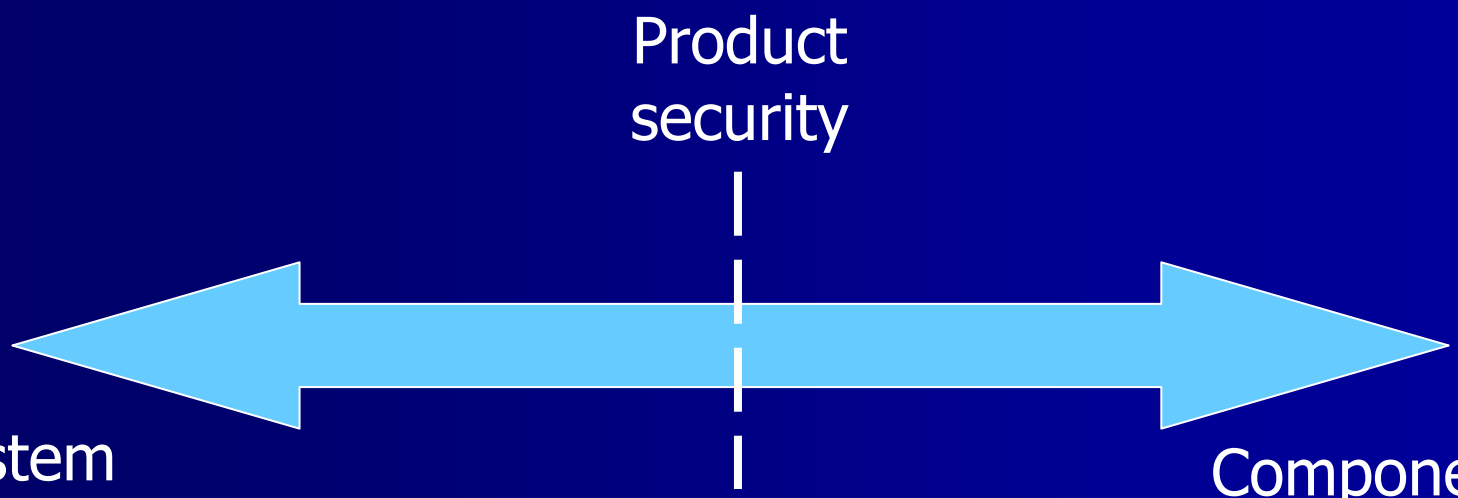
Component security

SONY

- Products consist of components
- At the system level, we can evaluate individual components, known as “products”, independently
- At the product level, we may not evaluate individual components?
- There is an obvious disparity between the approaches to the security at different levels
- This disparity is greatest when we look at the software evaluations

The spectrum of security

SONY



Overall security

SONY

- The components of a product must be evaluated the way we evaluate the products
- The context of a product provides the environment for the component just as the system provides the context for the product
- The approach must be systematically consistent from evaluating software and hardware components all the way to building the systems

Current state

SONY

- Lots of experience in evaluating hardware and hardware-based products
- Even complex composite products are evaluated without much trouble
- Evaluation of software components is far behind
- Lagging of software component evaluations drags down the natural process of product composition from certified components

What is so peculiar about software?

SONY

- Much higher complexity
- Much easier to develop using logically separated components
- Quick development of the functionality but long time to get the details right
- Infinite stability

Complexity

SONY

- Software is much easier to develop compared to hardware
- Therefore we make very complex things in software
- The way we deal with the increasing complexity is to split the software into components
- Components, in turn, get increasingly complex

Development cycle

SONY

- There are different models for managing the complexity of development
- In the end, there are always two phases:
  - Development of the functionality
  - Getting the functionality exactly right
- The second stage may take longer
- A fully understood, tested and verified software can be easily reused

Software stability

SONY

- As opposed to the hardware, the software is not subject to wear and tear
- Software does not need the maintenance or protection required for the hardware
- Software will keep performing the required function indefinitely

Software vs. hardware: implementation

SONY

- Hardware is built from real-world matter while software is built of ideal mathematical objects with behaviour defined precisely with abstract rules
- Hardware can fail, software cannot
- Hardware can have dependencies that would be absurd for the software
- The dependencies in the software are much easier to identify and analyze.

What is the point?

SONY

- Software, once written, stays functional forever
- Software can be evaluated once and for all
- The total sum of software in a product is usually split into building blocks – components
- A product may be created by using infinitely stable, evaluated components

More interestingly

SONY

- Hardware does not know how it will be used, software knows exactly what it needs to do and how it will use the hardware.
- Hardware operates in accordance to the way the software is written.
- Software is primary, software is the “master plan”

Software summary

SONY

- Software is the master plan of function
- High level of complexity
- Precisely defined behaviour
- Infinite stability
- No possibility of failure
- Dependencies are easy to define

What all this leads to?

SONY

- The software should be the basis of the evaluation
- We are used to evaluating the hardware first and then seeing how it is used by the software
- We should reverse the process, evaluate the software first and evaluate the hardware in accordance with its use by the software

Compare with the evaluation flow

SONY

- In a CC evaluation, we start from ST to see what has to be done and we proceed downwards to see how it is supported
- The current state is like starting from the code to see what it can do and proceeding upwards to check that ST does, indeed, need some of the functions implemented in the code
- Let's start from the logical beginning – the software that rules the functionality!

Certified components

SONY

- The purpose is to make components self-contained
- The functionality of a component is not affected by the functionality of other components
- A component can be fully tested and relied on to keep the set functionality
- Certified components become the basis for building secure systems

What is in it for us?

SONY

- Assembly from certified components: lower cost
- Independent component support: lower effort
- Clear “separation of duty”: higher security

What is the problem then?

- Software requires some hardware to be able to run for testing.
- Hardware introduces dependencies.
- Software is much more complex and big, it takes a lot of time to analyze, especially when it has many dependencies.

And what is the solution?

SONY

- The solution is to use the component evaluation.
- The solution is two-fold:
  - Make sure the components are self-contained for the most part and contain a clearly defined and stable functionality
  - Make sure the component describes clearly what it will expect from the environment and how that environment will be used.

What do we need?

SONY

- An agreed way to provide the description of the software and hardware dependencies
  - Policies
  - Security Functional Requirements
  - Developer documentation
  - … ?
- An agreed evaluation methodology
- A product to test all of these on

What will we do then?

SONY

- We started a project for software component evaluation that will allow us to test the methodology and gain some experience
- The product is a smart card with a complicated structure of software
- The purpose is to certify the software components separately and then reassemble the product from those certified components

Who is at the forefront?

SONY

- Certification body: CESG of UK
- Evaluation labs: LogicaCMG and SiVenture
- Developer: Sony Corporation

- The “guinea pig” product: Sony FeliCa smart card

SONY

Thank you!

Albert Dorofeev
General Manager
Sony Secure Communications Europe
albert.dorofeev@eu.sony.com