Issues for Common Criteria in Distributed Adaptive Service Based Architectures

Combinatore Chandrasekaren
Edward A. Schneider
William R. Simpson

Outline

- Common Criteria
- Distributed Adaptive Service Oriented Systems
- CC Issues
- Summary

Current Evaluation Process Characteristics

- CC almost exclusively used for products*
  - Components of many systems
- Certificate valid only for the version, patch level, configuration, and environment specified in ST
  - Consumers must make sure that these match what they purchase and their threats.
- CC formulation static by nature.

* Some notable exceptions (i.e., FAA) have not been overly successful.

Current Evaluation Process

Some attempts have been made to handle dynamic issues such as security patches, product enhancements

- ALC_FLR (Flaw Remediation)
  - Patching, however, alters version and make the product no longer the certificated product.
- AMA (Maintenance of Assurance)

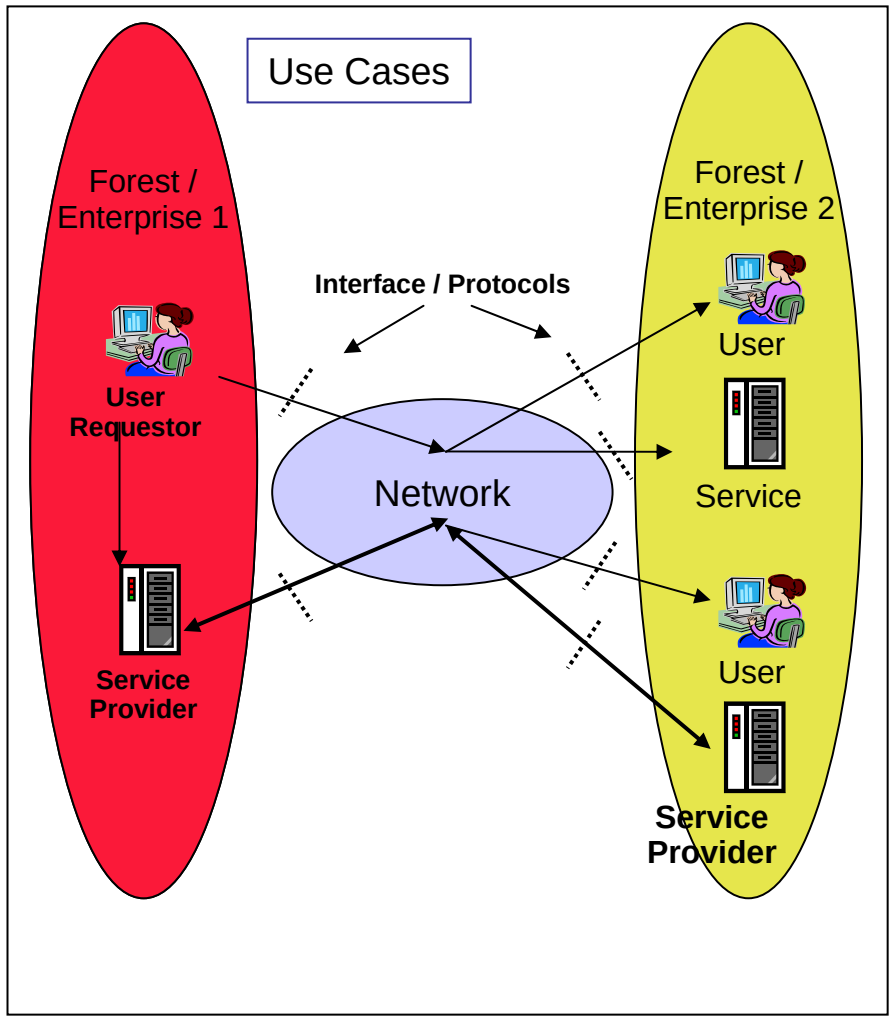
However, Flaw Remediation and Maintenance of Assurance are not part of any EAL Package and in fact AMA was removed from the 2.3 version of the CC.

Large scale Distributed Service oriented Systems

- Internet scale Distributed systems
  - Are growing in size
  - Are supporting richer functionality
  - Are growing in applications
  - Are growing in protocols supported
  - Are supporting more complex trust relationships
  - Are running in more hostile environments
  - Are being subjected to more systematic attacks
  - Conclusion ➔ Dramatically more complex
- Assurance
  - Technology has NOT kept pace
  - Is being done in a very ad hoc fashion
  - Approach used is modeled after stand alone systems
  - Inconsistency between sites for similar configurations
  - Very little change in practice in the last 15 years
  - Reduced confidence in results and not being used for component selection

Basics of SOA / Netcentricity Paradigm

- End-to-End Usage scenarios
  - Any Authorized Consumer to Any Authorized Provider
    - User – User or Service
    - Service – User or Service
- Netcentric Interactions [enterprise wide or cross enterprise]
  - But everything happens via a Service
- Basic interaction paradigm
  - Discover – Within Netcentric Domain
    - Available services, including security
  - Locate
    - Exchange Security Data?
  - Connect
    - Exchange Security Data?
  - Negotiate [No standard]
    - Exchange Security Data?
  - Authenticate
    - Exchange Security Data?
  - Authorize Access [No enterprise standard]
    - Exchange Security Data?



Enterprise Systems

- What consumers really care about
  - How can CC be applied to large distributed systems with hundreds of products from many different vendors?
- Certification used to assess risk
  - How do security properties of a set of products affect those of the total distributed system?
- Much of the product evaluation data is proprietary to lab and not available to consumer

Issues In Evaluation of Distributed Systems

- Service Architectures
- Dynamically Changing Communities Of Interest (COIs)
- Dynamic Security Policy
- Shared Security Data
- Hierarchy of TOEs
- Need for Standard Interfaces
- The Problem of Composition

Service Architecture

- Each Participant brings services to the table
- These are made available through discovery process
- Not all services will be available to all players (security policy issue)
- This makes all services, by necessity, security enabled. (all software in the service architecture is subject to evaluation).
- Configuration of all software modules are an issue.

CC ISSUE: Even non-security services software are security enabled. Items normally considered system certification are critical to evaluation.

Dynamically Changing Communities of Interest

- Each new community has its own players and community security policies.
  - Diversity in policy
  - Dynamically changing roles
- These details must be made available to all players and held secure and tamper-proof.
- Constantly changing roles and privileges.

CC ISSUE: Dynamically changing roles and privileges. Currently predefined in CC formulation.

Dynamically Changing Security Policy

- The distributed system must balance the “need to know” with the “need to share”.
- Security Policy not only changes with the players. But with the environment and the urgency.
  - Conflict resolution

CC ISSUE: Software services may behave differently under the differing policies. How to evaluate and how to test the range of events.

Shared Security Data

- Security data must be shared among services.
- The sharing itself must be secure.
- May happen in a separate service.

CC Issue: How do we evaluate the sharing of security data across many services and a varying operational security policy? What functionalities are required?

An example would be shared key exchanges, participant certifications and authenticated repositories for security policy and roles that are evolving and dynamic. The former two have been addressed in the literature extensively, the latter has been the subject of related research by Gershon Kedem Duke University, et. Al. - *Paranoid: A Global Secure File Access Control System*

Hierarchy of TOEs

- Request for services may induce a dialogue between security enable software sets.
- Such a dialogue may include Identification and Authorization, local encryption and key management and other security related events without user interface.

CC Issue: TOE to TOE interface across multiple TOEs, protection of security data, and Composition of such systems.

Standard Interfaces

- In many instances the security enabled software must invoke a layered protocol including security services.
- Theses layered protocols must be standardized to allow new services to be interoperable with existing services.

CC Issue: These standardized security enabled interfaces must(?) be separately evaluated.

Current research into application programming interfaces (API) present there own security issues – see Amerson Lim Thesis at MIT and Alloy system

The Problem of Composition

- Hierarchical TOEs and Dynamically Changing membership and security Policy complicated the problem of TOE interaction.
- The composition problem is a many on many problem and not a pair by pair communication evaluation.
  - The number of free variables in this space is enormous.
  - Standard APIs and shared security data may help reduce this space.
- Composition must be handled by requirements definition including standard interfaces.

CC Issue: The nature of the PP must be expanded to include requirements that are neither functional nor assurance.

The problem of composition has been approached from a rule-based composition which is similar to the requirements model – see Anupam Datta, et. Al. on Secure protocol Composition

Summary

- Distributed Systems with Dynamic membership, policy and behaviors present significant challenge to product evaluation.
- Such systems are being developed and individual software products used in these systems are all security enabled.
- CC must provide a method of accommodating these dynamic processes.
- Current research has offered some possible solutions in the areas of composition, secure data sharing, secure key exchanges and others.