

A Consumer's Perspective on the Application of the Common Criteria

Nir Naaman, CISSP
Metatron, Ltd.
nir.naaman@metasec.com

Presentation Agenda

- How do I define “consumer”?
- What are the consumer’s IT Security needs?
- Where does the CC fit in?
- What could it do better?

Consumer – a Definition

- A medium-to-large organization with business-critical IT infrastructure
- Typically large IT workforce with hundreds or thousands of developers and integrators
- IT Security is not a main business focus



CC – Who Needs It?

- Evaluation Labs need it!
- Vendors need it!
- Not enough to sustain the CC industry long term



What does a Consumer Need?

- Security (a subjective quality)
- Limited liability (standard of due care)

**Both can be better (cheaper)
served by firewalls and
penetration testing**



•

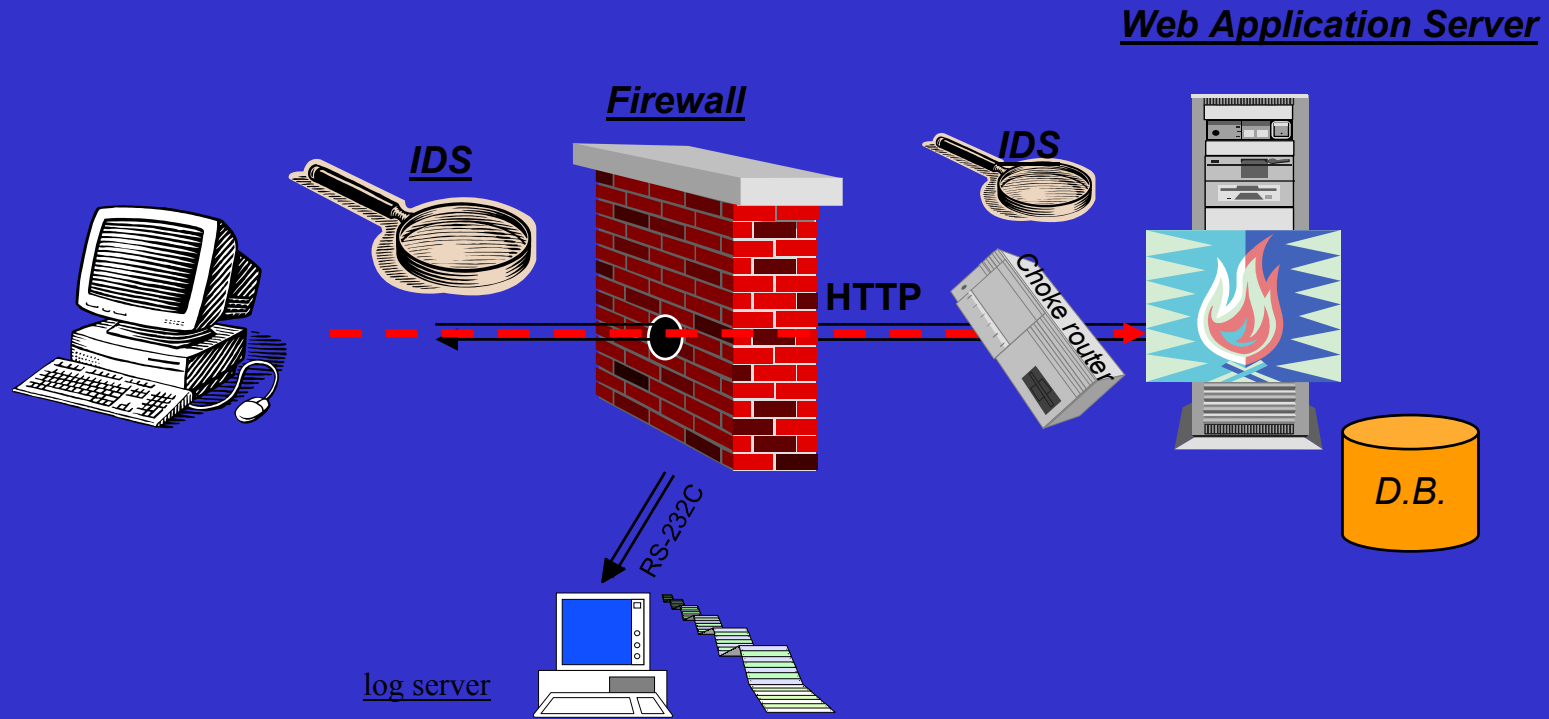
What the Consumer doesn't Need

- Vendor writes ST (or ST-Lite) for existing product
- Generic description of TOE environment
- Threat model is abstracted out
- Security concept is abstracted out
- Security requirements match marketing brochures
- Evaluation lab generates secret ETR that explains why the TOE was passed even though it has residual vulnerabilities
- Scheme issues certificate and certification report

Building Secure “Systems”

- What the consumer really needs is to know that his “systems” are secure
- COTS components
 - Need to be clear on capabilities and services
 - Must integrate securely with other components
- Custom development
 - Security engineering
- System Integration and Security Composition
 - Hard problem

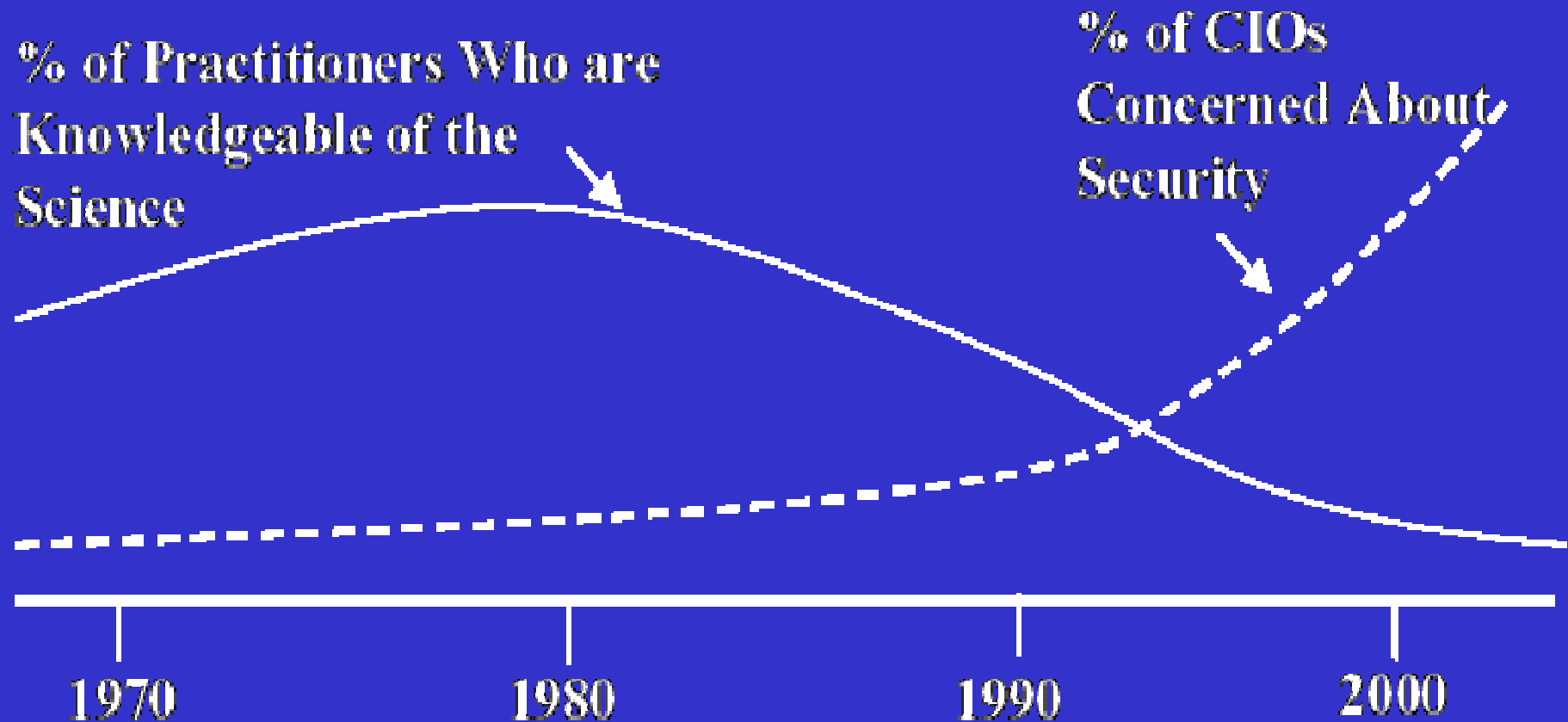
Example "System": Web Access



Appropriateness to Consumer of PP/ST

- The evaluator shall confirm that the TOE description is coherent and internally consistent, and is consistent with the other parts of the PP.
 - The evaluator determines in particular that the TOE description does not describe threats, security features or configurations of the TOE that are not considered elsewhere in the PP.
- The evaluator shall confirm that the statement of TOE security environment is coherent and internally consistent.
 - The evaluator determines that each assumption about the intended usage of the TOE is explained in sufficient detail to enable consumers to determine that their intended usage matches the assumption. If the assumptions are not clearly understood, the end result may be that consumers will use the TOE in an environment for which it is not intended.
- The evaluator shall confirm that the statements of security objectives, IT security requirements, TOE summary specification are **complete**, coherent, and internally consistent.
- The evaluator shall confirm that the PP claims are a correct instantiation of the PP.

CC as an Encoding of the Science of IA



Information Security: Science, Pseudoscience, and Flying Pigs
(Dr. Roger R. Schell)

CC Value Proposition

- “Cookbook” for secure systems
 - **Lemma I:** A system that’s been designed and developed to pass evaluation is more secure than one that has not;
 - **Lemma II:** The Common Criteria are the culmination of hundreds of man-years of the best security professionals on the planet
- CC Provides methodology for:
 - Security requirements specification
 - Security Engineering (ADV)
 - Development security (ACM, ALC_DVS, etc.)
 - Operational security (ADO, AGD, AVA_MSU, etc.)
 - ...

Alternative Approach (to CC): C&A

- Post-development C&A is too late. It is after-the-fact validation. Political pressures are such that a major re-engineering is impossible at that stage.
- Major re-engineering is often sorely needed ☹️
- Pre-development C&A (“Developmental ST&E”) – no better existing framework than the CC
- ➔ Systems must be engineered to pass CC evaluation

CC As a Common IT Security Language

- Common language between stakeholders and IT developers
 - Development organizations are willing to pay for clearer security requirements
 - Cuts analysis, negotiation, and review costs
 - Developers hate: “The system shall be secure...”
 - Reduction in C&A costs
- Common language between the consumer and COTS component providers
 - “Everybody” speaks the CC language
 - Useful for RFPs and security lifecycle management

Promoting CC as a Brand

- CC measures both process and result
 - Process evaluation can be partially reused
 - ACM, ALC, AMA, ADO, ...
 - → “CC Certified” development organization
- CC Deliverables are well-defined
 - → “CC Certified” development tools
 - CASE tools for PP/ST authoring
 - Configuration Management Tools (ACM)
 - Specification Tools (FSP, HLD, LLD)
 - Flaw Tracking Tools (ALC_FLR, AMA)
 - Test Management (ATE, AVA)
- “CC Certified” Security Professional

Tailoring the CC to the Organization

- Organizations typically classify systems/data
 - Increased security functional/operational requirements for higher classifications
 - Balanced Assurance
- Create classification-parameterized PP templates that encode organizational security policy and guidelines
- Examples
 - FAA NAS Template
 - NIST SP800-37 Guidelines for the Security Certification and Accreditation of Federal Information Technology Systems
 - IATF 3.1

Role of Schemes and Evaluation Labs

- Consumers prefer national-language documentation
- Evaluation is just another tool in the IA arsenal
- Evaluation labs are a third-party outsourcing function, similar to other QA services
- Validation will usually be performed by or on behalf of the consumer
- Certification is only useful when consumer needs to prove system security to third party

The CC Community as Trusted Third Party

- Evaluation labs/Schemes as trusted repository
 - Source code escrow
 - Code signing
 - Secure delivery
- Reuse of evaluation results
 - Critical for evaluation-in-parts
 - Evidence must be available across labs/Schemes
- CC Community must expand beyond PP/ST
 - Common interest dictates sharing of “good” documentation practices
 - ICCC prizes for best in each assurance category
 - Security Design Patterns

Hard Questions

- How can we trust COTS products for which we do not have access to evaluation evidence?
- How can we trust non-evaluated products?
- How can we trust evaluated products?
- How can we trust custom products for which we do have access to evaluation evidence?

Hard Answers

- We can't
- We don't have a choice

- Solution
 - ➔ Don't trust products – they are irrelevant
 - ➔ Only trust trustworthy solutions
 - ➔ Use CC to determine what is trustworthy
 - ➔ Everything else **MUST** be outside of the TSF



Is a vendor-written ST any good?

- It is a preliminary input to the acquisition process
- It can be partially reused to comply with a consumer-written PP
- It can be (mis)used for C&A
- It helps identify what the product DOESN'T do
- C&A and/or evaluation-in-parts and/or re-validation require additional evidence (ETR-Lite? Not necessarily enough)
 - Favors large consumers/hungry vendors
 - Open Source advantage

Some More Ideas

- New AMA family:
 - AMA_PHB (Put your Head on the Block)
 - Dependencies: ALC_FLR.1
 - For every security flaw discovered in the ‘product’ **after** certification:
 - Developer shall provide an analysis of whether the flaw is part of the evaluated configuration, why there are no similar flaws, why it wasn’t caught, etc.
 - Evaluator will perform a process improvement analysis
- Outsourcing of the C&A function

CC Modularity

- Multiple, incompatible, sometimes mutually-hostile stakeholders
- One answer: evaluation-in-parts (horizontally)
 - Confidentiality TSP
 - Integrity TSP
 - Availability TSP
- Interestingly enough – TSF is not necessarily the same for all TSPs! Neither is SPD!

Example: two access control SFPs

```
FDP_ACC.1(1) ([SFP for object1], [subject, object1, operations]*)
FDP_ACF.1(1) ([SFP for object1], [security attributes], [rules governing object1 access])
FMT_MSA.3(1) ([SFP for object1], (restrictive/responsive), [roles])
FMT_MSA.1(1) ([SFP for object1], (operation selection), [security attributes],
              [roles allowed to perform operation on specified security attributes])
FAU_GEN.1(1) ([auditable events], [other audit relevant information])
FMT_SMR.1(1) ([authorised identified roles])
FMT_SMF.1(1) ([security management functions])
FIA_UID.1(1) ([actions allowed before management user is identified])
```

```
FPT_STM.1(None)
```

```
// and now for the second SFP:
```

```
FDP_ACC.1(2) ([SFP for object2], [subject, object2, operations]*)
FDP_ACF.1(2) ([SFP for object2], [security attributes], [rules governing object1 access])
FMT_MSA.3(2) ([SFP for object2], (restrictive/responsive), [roles])
FMT_MSA.1(2) ([SFP for object2], (operation selection), [security attributes],
              [roles allowed to perform operation on specified security attributes])
FAU_GEN.1(2) ([auditable events], [other audit relevant information])
FMT_SMR.1(2) ([authorised identified roles])
FMT_SMF.1(2) ([security management functions])
FIA_UID.1(2) ([actions allowed before management user is identified])
```

Example (cont.)

- What we really want to say:
 Apply accessControlFunction(
 Subjects=...,
 Objects=object1,
 Attributes=...,
 Rules=...);
 Apply accessControlFunction(
 Subjects=...,
 Objects=object2,
 Attributes=...,
 Rules=...);

Example (cont.)

Package Interface accessControlFunction

Parameters:

```
Define objects are [assignment: objects protected by the access control SFP];  
Define subjects are [assignment: subjects controlled by the SFP] default to "all  
authorized subjects";  
Define attributes are [assignment: security attrs.];  
Define rules are [assignment: access control rules based on "attributes"];
```

Assets:

```
"objects" describes the set of objects protected by the access control function  
SFP;
```

Threats:

```
Complies with NSA_THREAT_TAXONOMY;  
T.UNAUTHORIZED_ACCESS unauthorized threat agent perform(s) unauthorized access on  
"objects" resulting in [selection: advantage gained by threat agent, loss of  
confidentiality, loss of integrity, loss of availability, [assignment: arbitrary  
effect]];
```

Example (cont.)

Services:

O.ACCESS The TSF will ensure that users gain only authorized access to controlled resources;

Capabilities:

O.PROTECT The TSF will ensure that access control functions are invoked and succeed before each function within the PSC (Package Scope of Control) is allowed to proceed;

Objectives Rationale:

T.UNAUTHORIZED_ACCESS unauthorized user performs unauthorized access to "objects"

O.ACCESS controls access to controlled objects. O.PROTECT ensures that functions are invoked and operate correctly.

Dependencies:

O.PROTECT The TSF shall provide protection against unauthorized interference from external users;

O.SECURE_TIME The TSF shall provide secure time stamps;

O.AUDIT_TRAIL The TSF shall provide an audit facility;

Assurance Claims:

Complies with package EAL4;

End Package Interface;

Example (cont.)

```
Package Body accessControlFunction
  Apply ReferenceMonitor(Objects = "objects", Subjects = "subjects", Functions =
    "rules");

  // Now for the SFRs themselves:
  FDP_ACC.1 (accessControl_SFP, "subjects", "objects", read and write);
  FDP_ACF.1 (accessControl_SFP, "attributes", "rules");

  FMT_MSA.3 (accessControl_SFP, restrictive, sysAdmin);
  FMT_MSA.1 (accessControl_SFP, {query, modify, and delete}, "attributes", sysAdmin);

  FAU_GEN.1 (all requests to perform an operation on an object covered by the SFP,
    None);

  FMT_SMR.1 (sysAdmin);

  FMT_SMF.1 (management of "attributes" on "objects");

  FIA_UID.1 (None);
End Package Body;
```

Summary

- Consumer buy-in is critical to the success of the CC
- Consumers have different needs than evaluators and vendors do
- Consumers will pay for real value
- Top Priorities:
 - Framework for system composition
 - Integration with Security Engineering methodologies
- What else?
 - Risk analysis needs to feed into CC framework
 - C&A needs to build on top of CC framework

Questions?

Nir Naaman, CISSP

V.P., Security Services

Metatron, Ltd.

nir.naaman@metasec.com

