



Security Management

By:

Joseph Ronald Canedo



It is a Risky World



Vulnerabilities

- n Security objectives:
 - n Prevent attacks
 - n Detect attacks
 - n Recover from attacks
- n Attacks: against weaknesses in the information systems
- n Need: find weaknesses



Identifying and Eliminating Weaknesses

- I. Vulnerability monitoring
- II. Secure system development
- III. User training and awareness
- IV. Avoiding single point of failure



I. Vulnerability Monitoring

- n Identify potential weaknesses in existing information systems
- n Reveal wide-range of vulnerabilities



I. Security Flaws

- n Secure software installation
 - n Correct installation of software
 - n Change default settings
 - n Validate upgrades/changes
 - n Patch new security flaws



I. Vulnerability Detection Tools

- n Computer Oracle and Password System (COPS) – FREE
 - n Checks vulnerabilities of UNIX systems
- n Secure Analysis Tool for Auditing Networks (SATAN) – FREE
- n SAFEsuite (Internet Security Systems, Inc.)
 - n Family of network security assessment tools (web security scanner, firewall scanner, intranet scanner, system security scanner)
 - n Keyed to the IP address of the customer



I. Keeping up with Security Publications

- n Legal publications: how to remove vulnerabilities
 - n CERT advisories
 - n SANS Security Digest
- n Hacker publications: “how to” exploit known vulnerabilities
- n Security mailing lists



II. Building Secure Systems

- n 1960s: US Department of Defense (DoD) risk of unsecured information systems
- n 1981: National Computer Security Center (NCSC) at the NSA
 - n DoD Trusted Computer System Evaluation Criteria (TCSEC) == Orange Book



II. Orange Book

- n Orange Book objectives:
 - n Guidance of what security features to build into new products
 - n Provide measurement to evaluate security of systems
 - n Basis for specifying security requirements
- n Security features and Assurances
- n Trusted Computing Base (TCB) security components of the system



II. Orange Book Levels

Highest Security



- n A1 Verified protection
- n B3 Security Domains
- n B2 Structured Protection
- n B1 labeled Security Protections
- n C2 Controlled Access Protection
- n C1 Discretionary Security Protection
- n D Minimal Protection

No Security

II. Security Policy

	C1	C2	B1	B2	B3	A1
DAC	+	+	nc	nc	+	nc
Object Reuse	0	+	nc	nc	nc	nc
Labels	0	0	+	+	nc	nc
Label integrity	0	0	+	nc	nc	nc
Exploration of labeled info	0	0	+	nc	nc	nc
Labeling human-readable output	0	0	+	nc	nc	nc
MAC	0	0	+	+	nc	nc
Subject sensitive labels	0	0	0	+	nc	nc
Device Labels	0	0	0	+	nc	nc

0 no requirements

+

added requirement

nc no change

Systems Security and
Control

MAC

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)

II. Accountability

	C1	C2	B1	B2	B3	A1
Identification and Authentication	+	+	+	nc	nc	nc
Audit	0	+	+	+	+	nc
Trusted Path	0	0	0	+	+	nc



Assurance changes

0 no requirements
 + added requirement
 nc no change

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)

II. Assurance

	C1	C2	B1	B2	B3	A1
System Architecture	+	+	+	+	+	nc
System Integrity	+	nc	nc	nc	nc	nc
Security Testing	+	+	+	+	+	+
Design Specification and Verification	0	0	+	+	+	+
Covert Channel Analysis	0	0	0	+	+	+
Trusted Facility Management	0	0	0	+	+	nc
Configuration Management	0	0	0	+	nc	+
Trusted Recovery	0	0	0	0	+	nc
Trusted Distribution	0	0	0	← 0	0 →	+ →

0 no requirements

+ added requirement

nc no change

Systems Security and Control

No covert channel

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)



II. Documentation

	C1	C2	B1	B2	B3	A1
Security Features User's Guide	+	nc	nc	nc	nc	nc
Trusted Facility Manual	+	+	+	+	+	nc
Test Documentation	+	nc	nc	+	nc	+
Design Documentation	+	nc	+	+	+	+

0 no requirements
+ added requirement
nc no change

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)



II. Covert Channel Analysis

- n B1: no requirements
- n B2: covert storage channels
- n B3: covert channels (timing and storage channels)
- n A1: formal methods (proof of covert channel analysis)

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)



II. Design Specifications and Verification

- n C2: no requirement
- n B1: informal or formal model of the security policy
- n B2: formal model of the security policy that is proven consistent with its axioms, DTLS (descriptive top-level specification) of the TCB)
- n B3: convincing argument shall be given that the DTLS is consistent with the model
- n A1: FTLS (formal top-level specification) of the TCB
 - n Formal and informal techniques to show that FTLS is consistent with the model
 - n Convincing argument the DTLS is consistent with the model

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)



II. Orange Book Classes

- n C1, C2: simple enhancement of existing systems. Does not break applications.
- n B1: relatively simple enhancement of existing system. May break some of the applications.

- n B2: major enhancement of existing systems. Will break many applications.
- n B3: failed A1
- n A1: top-down design and implementation of a new system from scratch.

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)



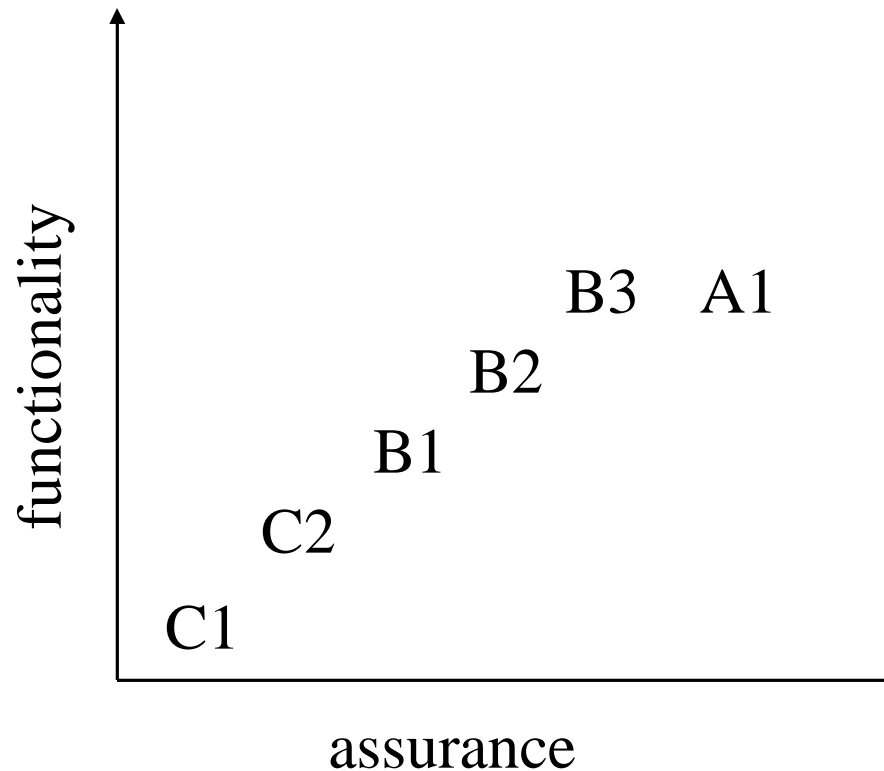
II. Orange Book Criticisms

- n Mixes various levels of abstraction in a single document
- n Does not address integrity of data
- n Combines functionality and assurance in a single linear rating scale

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)

II. Functionality vs Assurance

- functionality is multidimensional
- assurance has a linear progression



(from lecture notes of Jajodia <http://www.ise.gmu.edu>)



II. NCSC Rainbow Series

- n **Orange:** Trusted Computer System Evaluation Criteria
- n **Yellow:** Guidance for applying the Orange Book
- n **Red:** Trusted Network Interpretation
- n **Lavender:** Trusted Database Interpretation

(from lecture notes of Jajodia <http://www.ise.gmu.edu>)



II. European Criteria

- n German Information Security Agency: German Green Book (1988)
- n British Department of Trade and Industry and Ministry of Defense: several volumes of criteria
- n Canada, Australia, France: works on evaluation criteria
- n 1991: Information Technology Security Evaluation Criteria (ITSEC)
 - n For European community
 - n Decoupled features from assurance
 - n Introduced new functionality requirement classes
 - n Accommodated commercial security requirements



II. United State

- n January 1996: Common Criteria
 - n Joint work with Canada and Europe
 - n Separates functionality from assurance
 - n Nine classes of functionality: audit, communications, user data protection, identification and authentication, privacy, protection of trusted functions, resource utilization, establishing user sessions, and trusted path.
 - n Seven classes of assurance: configuration management, delivery and operation, development, guidance documents, life cycle support, tests, and vulnerability assessment.



II. Common Criteria

- n Evaluation Assurance Levels (EAL)
 - n EAL1: functionally tested
 - n EAL2: structurally tested
 - n EAL3: methodologically tested and checked
 - n EAL4: methodologically designed, tested and reviewed
 - n EAL5: semi-formally designed and tested
 - n EAL6: semi-formally verified and tested
 - n EAL7: formally verified design and tested



II. National Information Assurance Partnership (NIAP)

- n 1997: National Institute of Standards and Technology (NIST), National Security Agency (NSA), and Industry
- n Aims to improve the efficiency of evaluation
- n Transfer methodologies and techniques to private sector laboratories
- n Functions: developing tests, test methods, tools for evaluating and improving security products, developing protection profiles and associated tests, establish formal and international schema for CC.



III. Security Awareness and Training

- n Major weakness: users unawareness
- n Organizational effort
- n Educational effort
- n Customer training
- n Federal Trade Commission: program to educate customers about web scams



IV. Avoid Single Point of Failure

- n Critical information resources
 - n Identification
 - n Backup
 - n Hiding
- n Separation of duties
 - n Multi-person requirements
 - n Limit temptations

