Trust and threat evaluation

TkL Markus Peuhkuri

2008-05-06

Lecture topics

- Formal system evaluation
- Information security standards
- Threat trees
- Penetration testing
- Malicious logic
- After this lecture, you should
  - be able to name evaluation methods
  - know something about ISO 27000 standards
  - have an idea on threat estimation

Formal evaluation of systems

- Trust based on assurance *evidence*
  - a basis for confidence
  - not a perfect security
- Helps creating secure systems
- Evaluation methodology features
  - set of requirements for a security functionality
  - set of assurance requirements to establish that system meets functional requirements
  - methodology for determining that system meets functional requirements based on analysis of the assurance evidence
  - measure for evaluation (*level of trust*)

What evaluation guarantees?

- The system is perfectly secure – *NOT!*
- Lot of effort put on securing system
  - system evaluation
  - security documentation
  - development methodology

  ⇒An expensive trill
  - cost of external evaluators
  - cost of own work
- Evaluated system is *less likely* to have problems

Evaluation history

- Originated from military and government
- Need for assurance in commerce
  - trustworthy contractors
  - information systems
- Legal requirements, liability
  - cover your back: it *was* certified. Many laws, like Sarbanes-Oxley and HIPAA in U.S., require organisations to act very carefully and keep audit trails.

Trusted computer system evaluation criteria (TCSEC)

- Orange Book (1983–1999)
- U.S. Government
- Evaluation classes
  - C1, C2, B1, B2, B3, A1 (D for failed)
  - mainly information confidentiality
  - no requirements for availability
- Functional requirements
  - discretionary access control (DAC)
  - object reuse
  - mandatory access control (MAC) (B1)
  - label requirements (for MAC) (B1)
  - identification and authentication
  - trusted path (B2)
  - audit mechanism
  - architecture

TCSEC

- Operational requirements
  - separation of roles (B2)
  - secure recovery (A1)
  - system integrity validation (A1)

TCSEC

- Assurance requirements
  - configuration management (B2)
  - trusted distribution (A1)
  - system architecture (C1)
  - design specification and verification

    C1,C2 no requirements
    B1 informal
    B2 descriptive top level specification (DTLS, formal)
    B3 DTLS consistent with security policy

A1 formal top level specification (FTLS) with formal methods and mapping to source code

- testing
- product documentation
- internal documentation

TCSEC evaluation classes

D: failed systems that have failed evaluation

C1: discretionary protection identification and authentication

C2: controlled access protection used for commercial products

B1: labelled security protection MAC for set of objects

B2: structured protection MAC, trusted path, least privilege, covert channel analysis

B3: security domains reference validation, requirements for development methodology

A1: verified protection formal methods to evaluate B3 requirements

TCSEC evaluation process

1. Application
2. Preliminary technical review (PTR)
   - readiness review
3. Evaluation
   (a) design analysis
   - based on documentation
     $\Rightarrow$ requirements for complete and correct

   (b) test analysis
   - coverage assessment
   - vendor-supplied tests

   (c) final review

- Government-sponsored evaluators
- Ratings maintenance program

How good TCSEC is?

- Sets a baseline for evaluations
- Limited scope
  - only for operating systems
  - U.S. military and government needs (no integrity, availability or other business needs)
- Problems with the process
  - additional criteria
  - a slow process

Common Criteria (1998–)

- International followup to TCSEC, etc.
  - Common Criteria Recognition Agreement (CCRA)
  - ISO 15408
- Three parts
  - CC documents
  - CC evaluation methodology
  - national scheme
- Two types of evaluation

  PP: protection profile implementation independent set of requirements for set of products or systems

  ST: security targets evaluation of single product or system
- Requirements
  - security functional
    - 11 classes with 2–16 families
  - assurance
    - 10 classes with 2–8 families

CC Levels build on assurance

1. Functionally tested
2. Structurally tested
3. Methodically tested and checked
4. Methodically designed, tested and reviewed
5. Semi-formally designed and tested
6. Semi-formally verified design and tested
7. Formally verified design and tested

Approximate correspondence of TCSEC and CC; number of certified equipment for EALn.

TCSEC	CC	count	other
D	-		
-	EAL1	11	
C1	EAL2	62	need for FIPS 140-2 L2
C2	EAL3	43	need for FIPS 140-2 L3
B1	EAL4	63	need for FIPS 140-2 L4
B2	EAL5	1	
B3	EAL6		
A1	EAL7		

FIPS 140 (1994–)

- Evaluation of cryptographic modules
- Based on levels
  1. FIPS-approved algorithm; software or hardware
  2. physical security, role-based authentication, EAL2
  3. enhanced physical security, EAL3
  4. detecting and responding to physical access

- Areas of requirements (11 total)
  - cryptographic module specification, parts, interfaces
  - roles, authentication
  - logical model, design
  - physical security (EMI/EMC)
  - operating environment
  - mitigating attacks

level	4	3	2	1
FIPS 140-1	9	66	143	86
FIPS 140-2	-	40	89	88

ISO 27000 family of standards

- Originally BS7799 (British standard)
- Builds on other quality standards
  - Plan – Do – Check – Act *
  - continuous improvement
  - ISO 9000 (quality) and ISO 14000 (environment)

Suunnittele
Toteuta
Arvioi Toimi

- Starts from *business risk*
  - identifies
  - analyses
  - addresses
- Based on US business idea:
  - if it not on paper, it does not exists
    ⇒ lots of new paperwork for Finnish companies

Future of ISO 27000

27000 Principles and vocaulary

27001 Requirements (BS 7799-2)

27002 Methods (BS 7799-1
⇒ ISO 17799)

27003 Implementation guidance

27004 Measurement

27005 Risk management

ISO 27002 / 17799

- Has 11 security control clauses
  1. security policy
  2. organising information security
  3. asset management
  4. human resources security
  5. physical and environmental security
  6. communications and operations management
  7. access control

8. information systems acquisition, development and maintenance
9. information security incident management
10. business continuity management
11. compliance

- Total 39 main security categories
  - control objective: what
  - controls that can be applied to archive objective
  - implementation guidance

Example: Organisation of information security

- Has two categories, like:
- Internal organisation of information security
  - objective: to manage information security within the organisation
  - 8 controls, like:
  - management commitment to information security by clear direction, demonstrated commitment, explicit assignment and acknowledgement security responsibilities
    $\Rightarrow$ there must be enough resources for security and responsibilites are well defined

Example: Access control clause

- Has seven categories, like:
- Network access control
  - objective: to prevent unauthorised access to networked services
  - 7 controls, like:
  - groups of information services, users, and information systems should be segragated to networks
    $\Rightarrow$ use of routing, VLANs, firewalls. gateway, network classification

SSE-CMM

- Systems Security Engineering Capability Maturity Model
- ISO/IEC 21827
- Describes organisation maturity, build on SE-CMM used in software development

  0 not propriate; measure cannot be applied

  1 basic: work & work; unorganised, random, ad-hoc, case-by-case

  2 iterative: plan the work; planned, repeatable and procedures can be followed, local chaos controlled

  3 defined: work the plan; well defined, coordinated, documented procedures, organisational learning

  4 controlled, quality: measure the work; quantitative goals defined, process metrics captured, processes managed based on data

  5 optimising, developing: work the measure; strategic goals are quantitative, continuously improves processes based on data

- Each process area is scrored from 0 to 5
  - determine security vulnerabilities
  - manage configurations
  - provide ongoing skills and knowledge
- Finnish goverment is planned to have level 2 as basic requirement for information security by 2011

Other models

- ITIL: Information Technology Infrastucture Library
  - developing and deploying IT service management
  - library of best practises
- COBIT: Control Objectives for Information and related Technology
  - developed by auditors
  - thorough framework for IT control
    - defining
    - implementing
    - auditing
  - security one of 34 practises

Threat modelling

- Target: understand and document security threats
- Large number of possible threats
  ⇒ Ad-hoc treat searching incomplete
  ⇒ Must be methodological
- System threat profile described
- Characterisation of system security
- Threat is *not* a vulnerability
  - vulnerability is unmitigated threat
  - attack classification important

Collecting information

- How system will be used
- What system depends on (environment)
- What assumptions are made on implementation
- How system interacts to environment
- Basics of internal design decisions

Modelling system

- What are entry points
  - network, services, user interface, files, disk system
- Assets to protect
- Trust levels
  - user groups, unidentified
  - access to assets
- Data flow
  - how data flows from entry to processes

Determining threats

- Threat identification
  - shadow corners of valid-but-malicious data
  - invalid data
- Analyse threat
  - does it result a vulnerability
  - how it can be mitigated

Threat effect classification: STRIDE

Spoofing allows obtaining false identity

Tampering modifies data for goal

Repudiation not providing evidence to point guilty

Information disclosure for unauthorised user

Denial of service for legitimate users

Elevation of privilege for higher trust level

Risk of vulnerability: DREAD

Damage potential if exploited

Reproducibility of vulnerability

Exploitability how easily vulnerability can be exploited

Affected users if exploit is widely available

Discoverability is likehood that vulnerability will be found

Assign weight for each category, calculate average.

Threat tree [1]

- Goal as tree root
- An attack is decomposed to sub-goals

  AND all sub-goals must be meet
  OR any of subgoals is sufficient
- Attack costs or pre-requirements can be assigned
  - helps to determine seriousness
- Reuse of attack patterns

Survivability Compromise: Disclosure of ACME proprietary secrets

```
 OR 1. Physically scavenge discarded items from ACME
        OR 1. Inspect dumpster content on-site
           2. Inspect refuse after removal from site
    2. Monitor emanations from ACME machines
        AND 1. Survey physical perimeter to determine optimal monitor-
ing position
            2. Acquire necessary monitoring equipment
            3. Setup monitoring site
            4. Monitor emanations from site
    3. Recruit help of trusted ACME insider
        OR 1. Plant spy as trusted insider
           2. Use existing trusted insider
    4. Physically access ACME networks or machines
        OR 1. Get physical, on-site access to Intranet
           2. Get physical access to external machines
    5. Attack ACME intranet using its connections with Internet
        OR 1. Monitor communications over Internet for leakage
           2. Get trusted process to send sensitive information to at-
tacker over Internet
           3. Gain privileged access to Web server
    6. Attack ACME intranet using its connections with public telephone net-
work (PTN)
        OR 1. Monitor communications over PTN for leakage of sensi-
tive information
           2. Gain privileged access to machines on intranet con-
nected via Internet
```

Penetration tests

- Experimental evaluation of system security
- Layered models: different threats
  1. external attacker without knowledge about system
  2. external attacker with an access to system
  3. internal attacker with an access to system
- Flaw hypothesis methodology
  1. information gathering
  2. flaw hypothesis
  3. flaw testing
  4. flaw generalisation
  5. flaw elimination
- Unsuccessful penetration does not prove system secure
  - cracking contests mostly useless publicity stunts

Malicious logic

Trojan horse user unintentionally executes program

- documented effect (what user expects)
- covert effect (malicious)

- Trojan in compiler [2]
- "free" software add-ons (spyware, adware)
- may replicate itself

Virus inserts itself to file

- may have malicious actions
  - corrupts files
  - destroys equipment
- loss of performance
- several subtypes by infection, implementation method

Malicious logic

Worm propagates between systems

- may have an impact on network
- most of current malware
  - massmailers
  - chat
  - p2p networks

Rabbits/bacteria exhaust resources quickly
`main(){for(;;)fork();}` (*DO NOT* run code on public systems...)

Logic bombs event triggers malicious action

- disgruntled employee

Protection against malicious code

- Problem: too coarse access control
  - any program has access to all my data
  - root/administrator omnipotent
- Enforcing principle of least privilege
  - sandboxing
  - capacity model
- Code signing not a solution
- Virus scanners too slow, do not work for targeted attacks: recently many CEOs have received court orders with link to malicious document

Summary

- Formal evaluation supports system development
  - for higher levels must be integral
  - overall quality assurance
- Security standards help developing proper procedures
  - security must start from top management
- Threat evaluation needed in systems development
- Threat trees help in large system evaluation
  - even if components are certified, network maybe insecure
- Penetration testing is a practical evaluation
  - needs high level of skill and experience

References

[1] Bruce Schneier. *Secrets and Lies: digital security in a networked world.* Wiley Computer Publishing, 2000.

[2] Ken Thompson. Reflections on trusting trust. *Commun. ACM*, 27(8):761–763, 1984.