From Dependability to Security

Yves Deswarte
LAAS-CNRS, Toulouse, France
Yves.Deswarte@laas.fr

Dagstuhl
11 September 2006

2 examples of…

…Application of Dependability Techniques to Security

- Intrusion Tolerance
  (with help from David Powell)
- Quantitative Evaluation
  (with help from Mohamed Kaâniche)

1. Intrusion Tolerance

Case: Internet Security

Conventional Security Techniques

User Authentication
- Identify user
- User responsibility and liability

User Authorization
- Prevent illegitimate actions
- Least privilege principle: legitimate ⇔ needed

- Inefficient in Internet context:
  - Strong authentication infeasible on publicly-accessible sites
  - COTS OS and application SW
    - many flaws
    - patches not applied due to lack of time or competency, or for fear of losing needed functionality
  - Internet protocols are vulnerable (Arpanet heritage)
  - Economic pressures do not (yet) favor known defenses
    - ingress filtering,
    - trace-back facilities, …

Case: Safety-Critical Systems

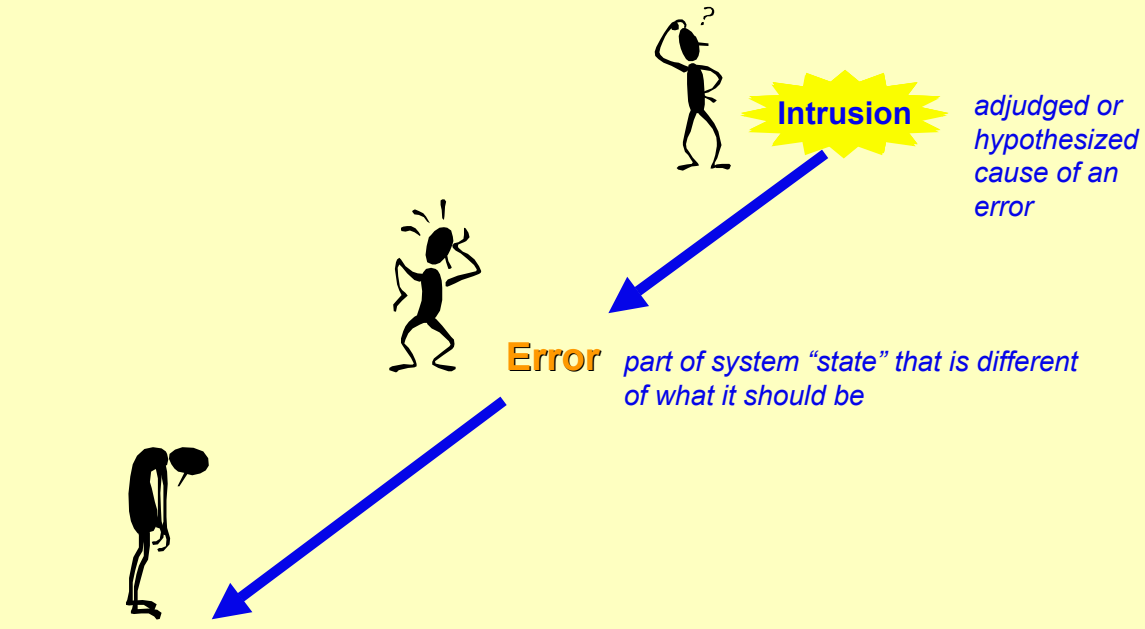
❖Conventional Safety Techniques:

- **Identify plausible faults:**
  - Accidental: physical, design, configuration/interaction
- **Assess their frequency and their consequences**
- **Avoidance:**
  - Quality, Simplicity, Formal methods (high assurance)
- **Tolerance means: fault coverage, assumption coverage**

❖New threats become preponderant

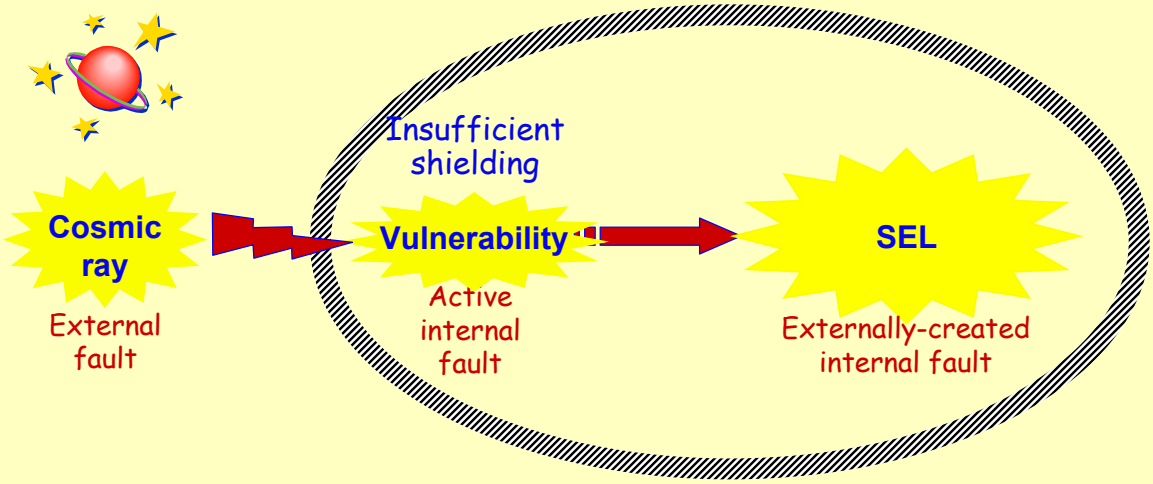
- **Malicious faults: deliberate attacks**
  - Difficult to avoid

"Dependability" Approach



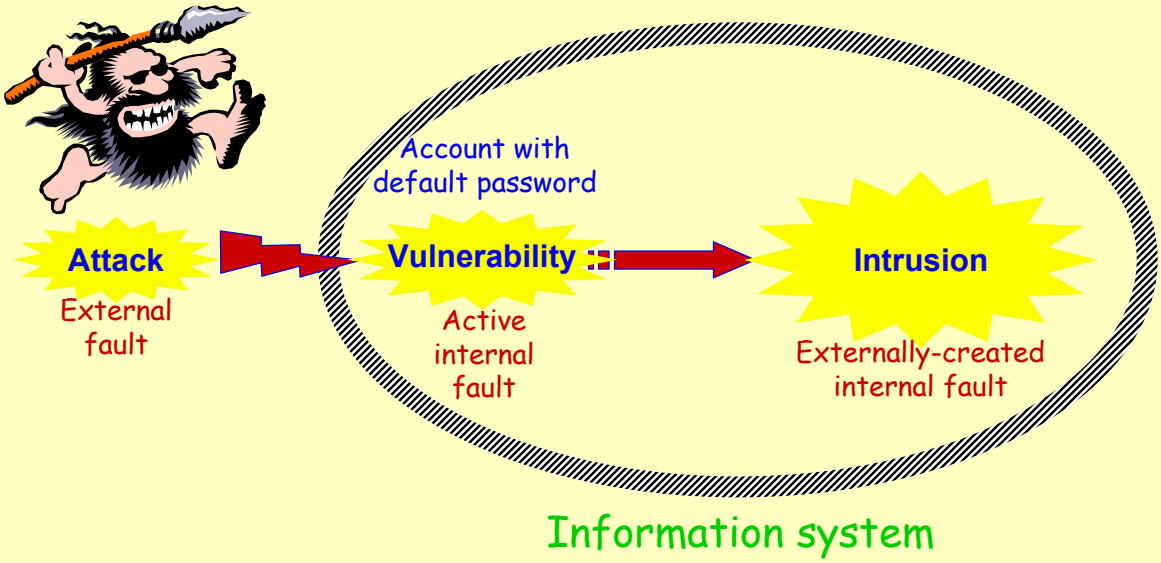
Example: Single Event Latch-up

SELs are reversible stuck-at faults
(ex. cosmic rays, heavy ions)

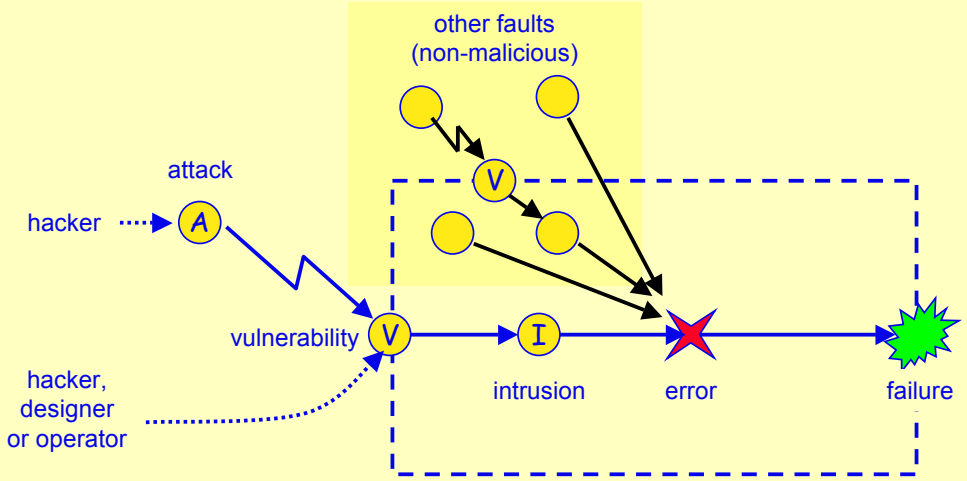


Intrusions

Intrusions are resulting from
(at least partially) successful attacks

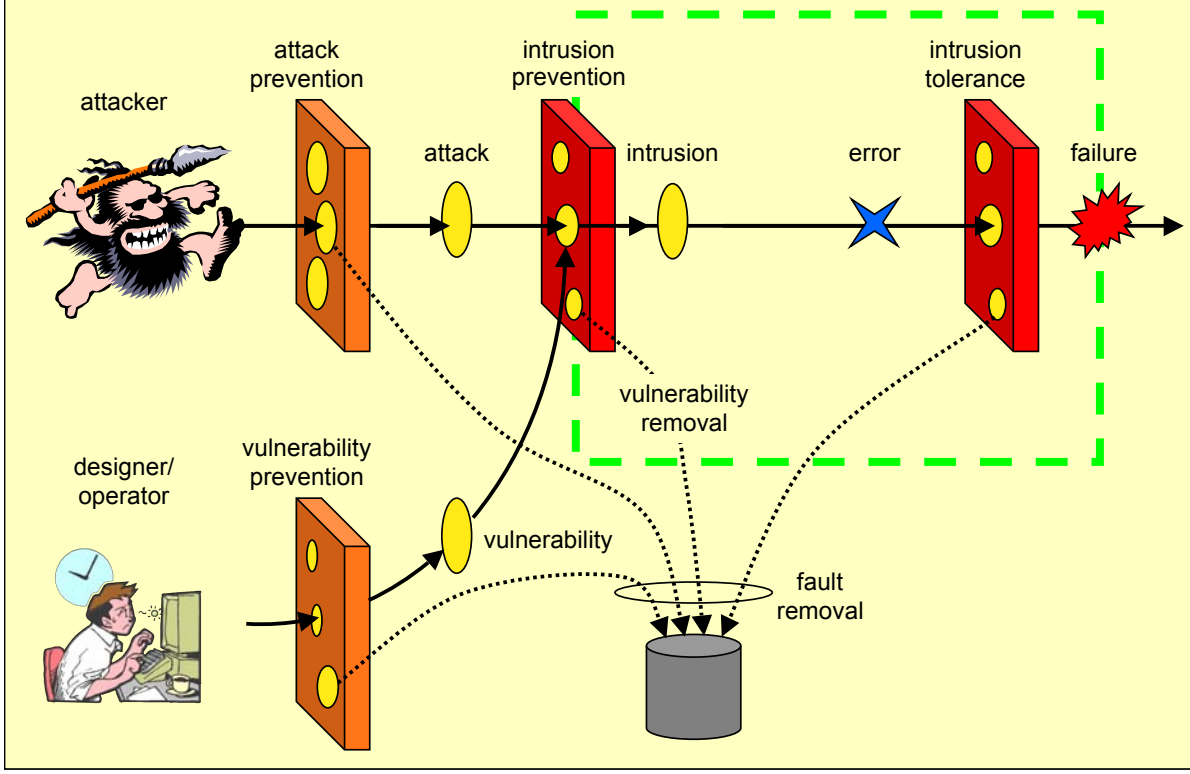


Fault Model



- **attack** - malicious external activity aiming to intentionally violate one or more security properties; an *intrusion* attempt
- **vulnerability** - a malicious or non-malicious fault, in the requirements, the specification, the design or the configuration of the system, or in the way it is used, that could be exploited to create an *intrusion*
- **intrusion** - a malicious fault resulting from an *attack* that has been successful in exploiting a *vulnerability*

Prevention, Tolerance and Removal



Intrusion Tolerance (IT)

- Intrusions are faults
- Faults can be tolerated

- But:
  - cannot rely on low likelihood of near-coincident attacks on different parts of system
- So, need to ensure that:
  - each part is sufficiently protected (no trivial attacks)
  - intrusion into one part does not facilitate intrusion into other parts
    - ↪ intrusion should not allow access to confidential data

DIT Project

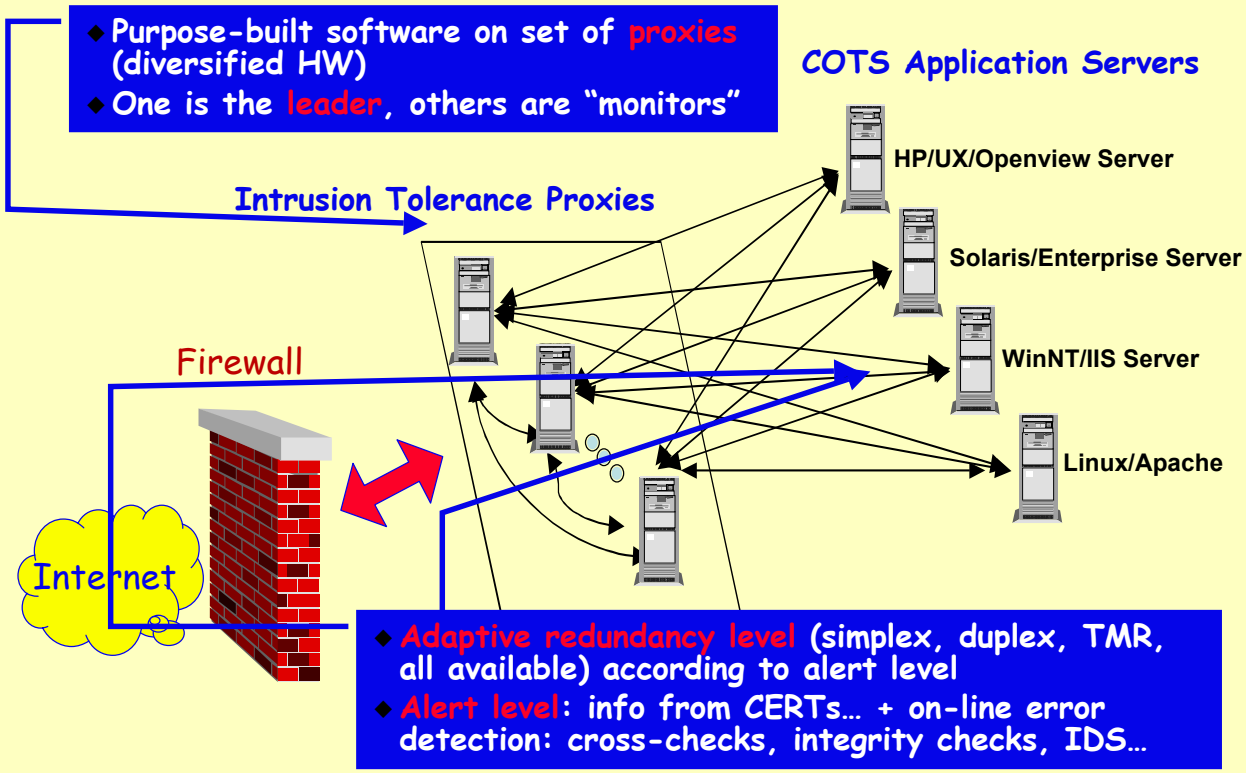
- DIT = Dependable Intrusion Tolerance

- DARPA OASIS (Organically Assured and Survivable Information Systems) program

- Partly sub-contracted to LAAS by SRI-International

- Design and implementation of a prototype intrusion-tolerant web server

DIT Architecture



2. Quantitative Evaluation of Security

Security evaluation

- Traditional methods
  - Evaluation criteria (TCSEC, ITSEC, CC, ...):
    ~ qualitative evaluation
  - Risk assessment: subjective evaluation of
    vulnerabilities, threats, consequences
  - Not well suited to take into account the dynamic evolution of systems and their environment during operation:
    "*How the system has been built?*" rather than
    "*How it is operated?*"

Quantitative security evaluation

- Probabilistic modeling framework
- Measure = effort needed for a potential attacker to defeat the security policy
- Objectives
  - Take into account security/usability trade-offs
  - Monitor security evolutions according to configuration and use changes
  - Identify the best security improvement for the least usability change

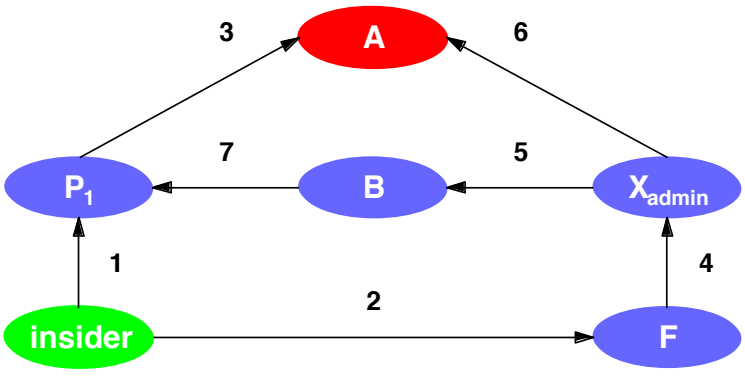
Proposed approach

- Identify security objectives: security policy
- Model system vulnerabilities
- Model the attack processes
- Compute significant measures

R. Ortalo, Y. Deswarte, M. Kaâniche, "Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security", *IEEE Transactions on Software Engineering*, Vol.25, N°5, pp.633-650, Sept./Oct. 1999.

Vulnerability modeling

Privilege graph



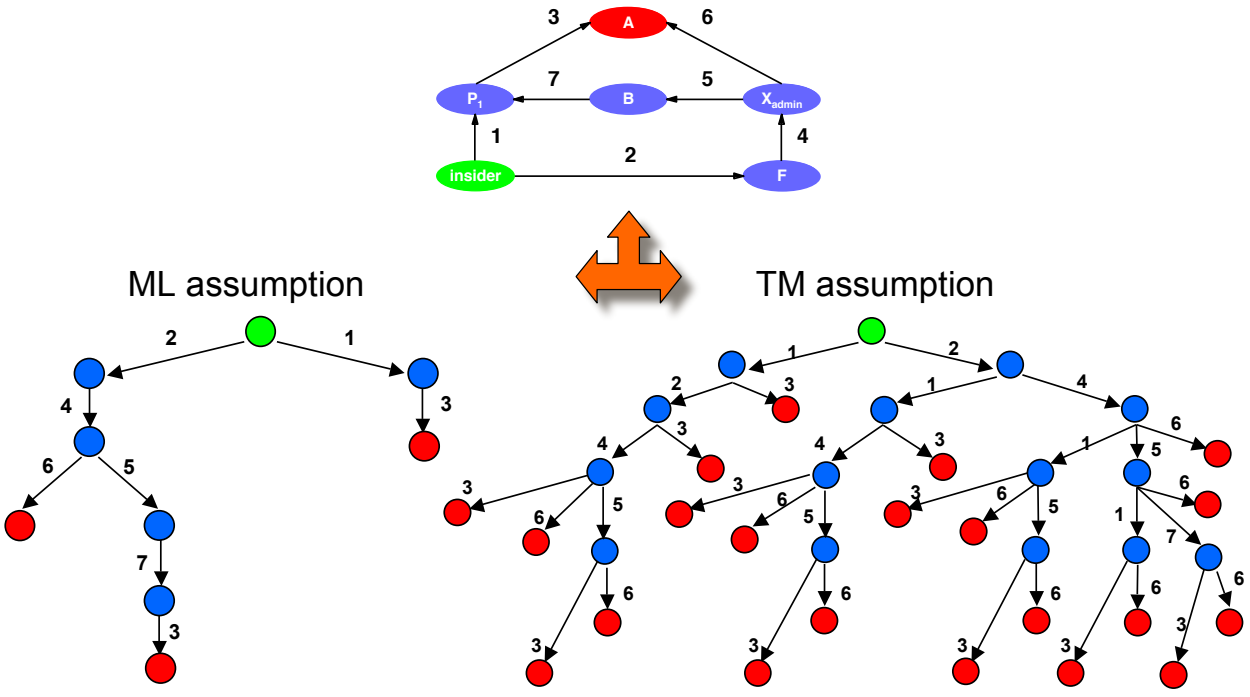
1) X can guess Y's password
2) X can install a Trojan horse that Y can activate
3) X can exploit a flaw in Y's mailer
4) Y is a subset of X
5) Y uses a program that X can modify
6) X can modify a "setuid" program owned by Y
7) X is in Y's `.rhosts`

- Node = a set of privileges (user, group, role, …)
- Arc = a method to transfer privileges = vulnerability
- Path = a set of vulnerabilities usable by a possible attacker to defeat a security objective
- Weight = for each arc, effort to exploit the arc's vulnerability

Attack process: Assumptions

- Attack processes = all possible successful attack scenarios
- General assumptions
  - The attacker knows only the vulnerabilities that can be exploited with the privileges he already owns
  - The attacker will not exploit vulnerabilities that would give privileges he already owns
- and, one of the following assumptions:
  - *Total Memory (MT)*: the attacker remembers all the vulnerabilities he did not exploit in the previous steps, and he can “backtrack”.
  - *Local Memory (ML)*: the attacker considers only the vulnerabilities that can be exploited with the new privileges he just acquired.

Attack process: Examples



Measure computation

① Identify the attacker-target couples

② For each couple, compute:

METF-ML: Mean Effort To security Failure (i.e. to reach the target) with ML assumption.

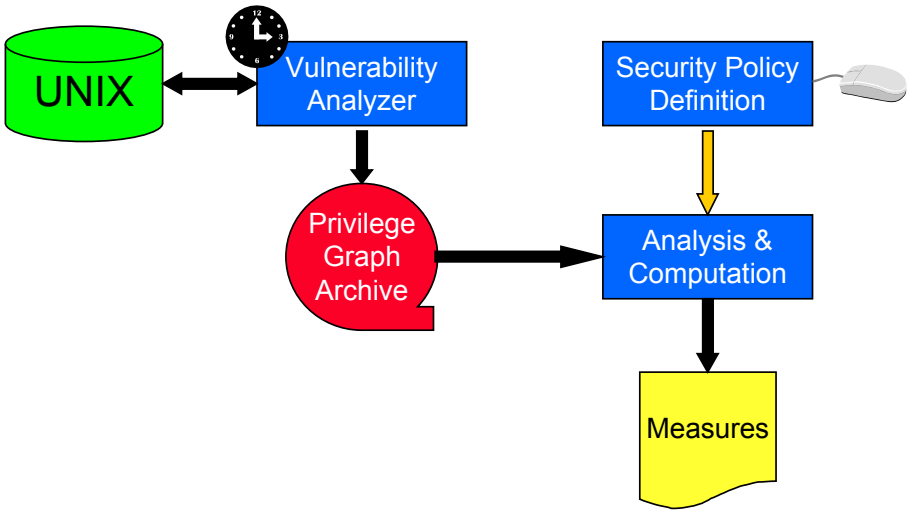
METF-TM: Mean Effort To security Failure with MT assumption.

Shortest Path: Mean effort to go through the shortest path.

Number of Paths: Number of possible paths from the attacker to the target nodes.

ESOPE tool set

(Évaluation de la Sécurité OPErationnelle)



Experimental assessment

- Objectives
  - Validate the approach
    - Assess the relevance of the measures wrt. system changes (configuration, users, …)
    - Demonstrate the feasibility of the approach considering an operational networked environment
  - The aim was not to:
    - correct the identified vulnerabilities

Experiment context

Target System:

- Unix
- 700 users
  300 machines - LAN
- 13 months
  (June 1995 - July 1996)

13 types of vulnerabilities
(fichiers `.rhosts`, `.*rc`, passwords, etc.)

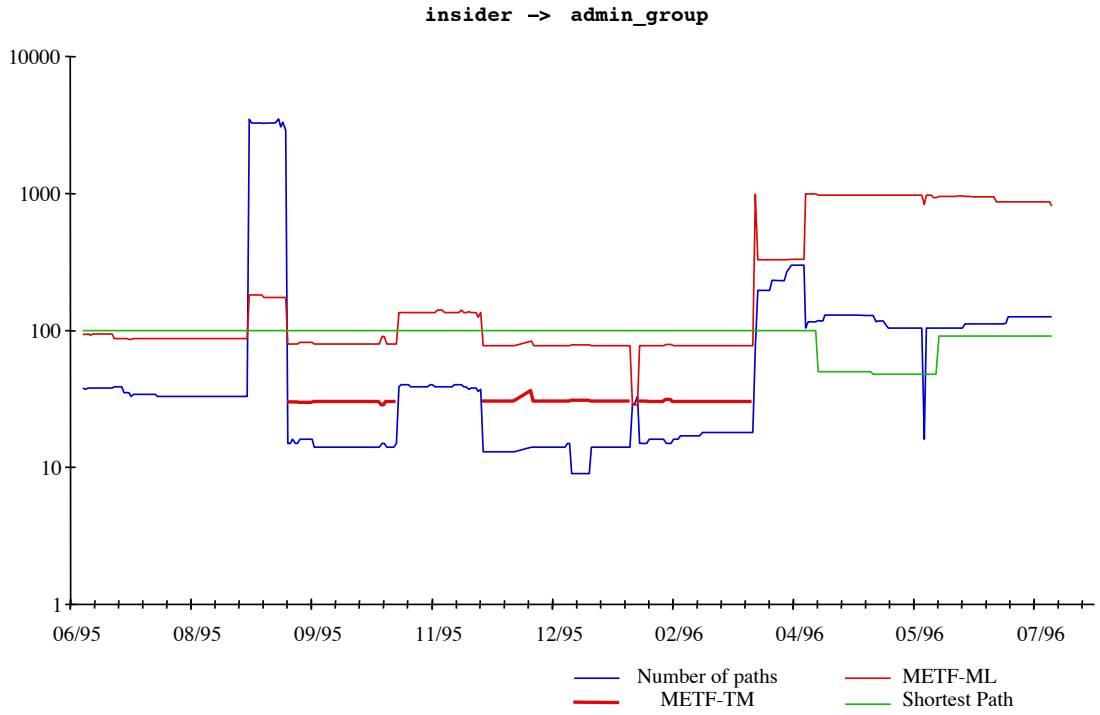
Security objectives

	Attacker	Target
Objective 1	insider	root
Objective 2	insider	admin_group

4 difficulty levels

Type	Weight
immediate	10
easy	10^2
difficult	10^3
very difficult	10^4

Result example



Problems

- Is the model valid in the real world?
- TM and ML are two extreme behaviors, but what would be a “real” attacker behavior?
- Weight parameters are assessed arbitrarily (subjective?)
  - Tenacity? Collusion? Attack rates?
- We need real data !!

Perspectives

- Data collection
  - Several honeynets (different domains, locations, etc.)
  - Need to analyze if data collected from different locations (e.g., .com vs. .edu) exhibit similar or different statistical patterns
- Data Analysis
  - Identify attacks and characterize their distribution in space & time
    - Known and new vulnerabilities
    - attack scenarios
    - trend analysis
- Security modeling and evaluation
  - Validate the proposed approach based on the privilege graph using high-interaction honeypots
  - Analyze how results are useful for designers/administrators