Using security patterns to combine security metrics

Thomas Heyman, Riccardo Scandariato, Christophe Huygens and Wouter Joosen

DistriNet, dept. of Computer Science
K.U.Leuven - Belgium

thomas.heyman@cs.kuleuven.be
Some metrics to get started...

- Technical security examples
  - Perimeter defense
    - Nb. of firewall rule changes by business unit/...
    - Virus incident cleanup cost
  - Coverage and control
    - Nb. of patches applied per period
    - Percentage of servers protected by antivirus software

Examples from A. Jaquith, Security Metrics, 2007
...and a bit of motivation

- Problems with metrics
  - Which metrics are **relevant** for me?
  - Is the metric **useful** (interesting) vs. is it **measurable**?
- Problems with the “big picture”
  - Given these measurements, is my system “secure”?
  - How can I remedy bad scores?
Elements of the solution

- Need a structured **repository** of metrics
  - Collect measurable time-tested metrics
- Need a framework for measuring
  - Facilitate **selection** during development
  - **Aggregate/interpret** measurements to useful (qualitative) indicators
Outline

- Problem statement
- Our solution

I. Associating metrics to security patterns
   II. Instantiate metrics through security patterns
   III. Interpret measurements during production (or development)
- Conclusion and future work
Security patterns

- Package **domain-independent** security knowledge in a **reusable** format

- Help realize certain **security objectives**
  (as observed in [SESS07])

- E.g. the Secure Logger, Audit Interceptor and Authentication Enforcer [j2ee]

Secure engineering w/ patterns

- Part of larger research track (e.g., [SESS07,Inventory])
- Iterative, quality-based
- Realizes security trough patterns
  - Result is **coherent set** of patterns
  - Instantiate during appropriate design phase

[Inventory] K. Yskout, T. Heyman, R. Scandariato, and W. Joosen, A system of security patterns
I. Security patterns and metrics

- Possible to attach *implementable* metrics to patterns
  - Measure *positive* security contribution

- Select a coherent set of patterns (implied by underlying methodology)
  - Implies selection of *relevant* metrics

[MetriCon1] T. Heyman and C. Huygens, Software Security Patterns and Risk
Outline

- Problem statement
- Our solution

I. Associating metrics to security patterns

II. Instantiate metrics through security patterns

III. Interpret measurements during production (or development)

- Conclusion and future work
II. Integrating metrics during Security requirements engineering

- Security requirements
  - Security objectives
    - Security patterns + metrics!
  - Metrics selection
- Measurement interpretation
- Architecture, design, implementation

Secure application
Metrics implementation - illustration

Security objectives (and sub-objectives)
- Accountability
  - Auditing
  - Non-repudiation
    - Principal authentication
    - Authentication Enforcer

Security patterns (and metrics)
- Audit Interceptor
  - Metric
- Secure Logger
  - Metric
- Authentication Enforcer
  - Metric

Metrics selection

AND
OR
Outline

- Problem statement
- Our solution
  I. Associating metrics to security patterns
  II. Instantiate metrics through security patterns
  III. Interpret measurements during production (or development)
- Conclusion and future work
III. Interpreting measurements

- Aggregate measurements
  - Score of pattern contributes to achievement of parent requirement
- Take interrelationships of patterns into account
  - Patterns might depend on other patterns/objectives
    - E.g. Audit Interceptor depends on Secure Logger
- Combine measurements through dependency graph
Interpreting measurements -

Security objectives (and sub-objectives)

Auditing

Non-repudiation

Accountability

Principal authentication

Authentication Enforcer

Audit Interceptor

Secure Logger

Measurement interpretation

Security patterns (and metrics)

Metric

Metric

 Metric

 Metric

 Metric

 Metric

 Metric

 AND

 OR
Outline

- Problem statement
- Our solution
  1. Associating metrics to security patterns
  2. Instantiate metrics through security patterns
  3. Interpret measurements during production (or development)
- **Conclusion and future work**
Conclusion

- A gap exists between:
  - high-level security requirements
  - production-level measurements
- Security patterns help to bridge this gap
  - Facilitate metric elicitation
  - Support selection and instantiation throughout development
  - Enable aggregation and interpretation
Problem statement revisited

- Only implement metrics for instantiated patterns
  - Solves relevance issue
  - But are those metrics sufficient?
- Problems with the “big picture”
  - Solves interpretation and diagnostic issues
  - But do final scores indicate acceptable risk?
- Will be handled in future work...
Thank you!

Questions or remarks?

thomas.heyman@cs.kuleuven.be