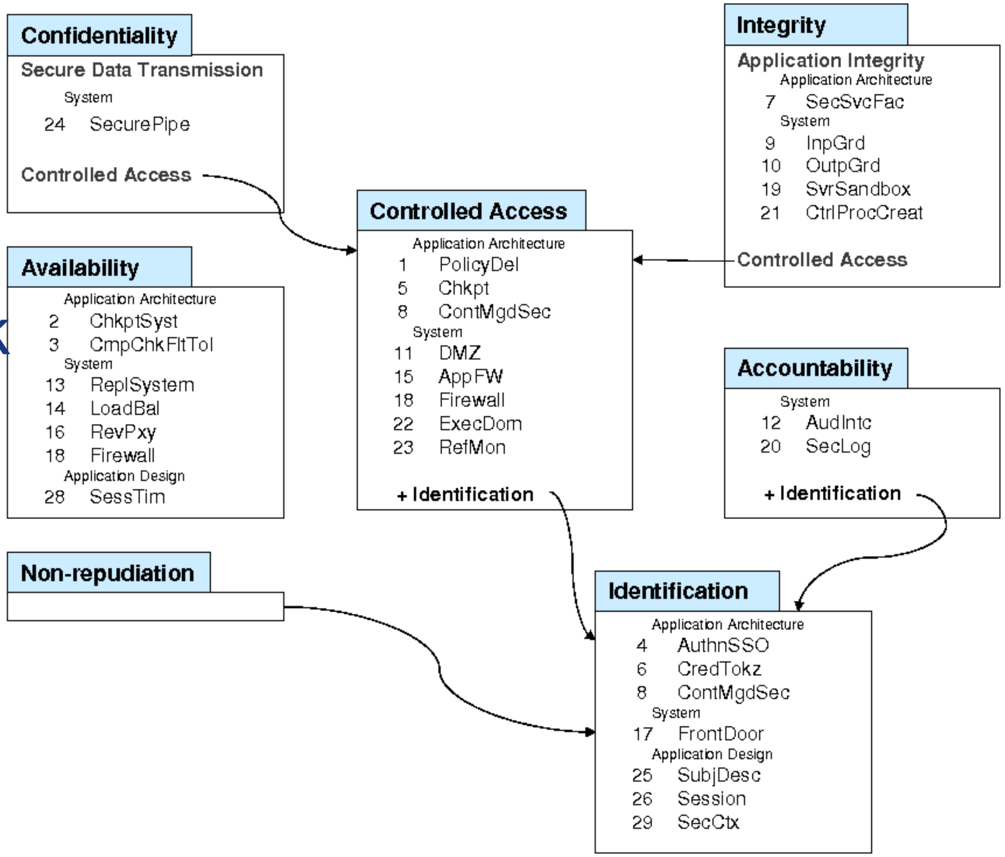
DistriNet
Research Group

Software security patterns and security metrics

Thomas Heyman, Christophe Huygens
DistriNet Research Group
K.U.Leuven

Security patterns

- A pattern
  - a solution to a problem
  - within a specific context
- Example: single access point pattern
- Why security patterns?
  - Capture expert knowledge
  - Domain-independent
  - Reusability!
- Related to our other work
  - Pattern taxonomy
  - Integration of patterns in design process



Confidentiality
Secure Data Transmission
System
24 SecurePipe
Controlled Access

Availability
Application Architecture
2 ChkptSyst
3 CmpChkFltTol
System
13 ReplSystem
14 LoadBal
16 RevPxy
18 Firewall
Application Design
28 SessTim

Non-repudiation

Controlled Access
Application Architecture
1 PolicyDel
5 Chkpt
8 ContMgdSec
System
11 DMZ
15 AppFW
18 Firewall
22 ExecDom
23 RefMon
+ Identification

Integrity
Application Integrity
Application Architecture
7 SecSvcFac
System
9 InpGrd
10 OutpGrd
19 SvrSandbox
21 CtrlProcCreat
Controlled Access

Accountability
System
12 AudIntc
20 SecLog
+ Identification

Identification
Application Architecture
4 AuthnSSO
6 CredTokz
8 ContMgdSec
System
17 FrontDoor
Application Design
25 SubjDesc
26 Session
29 SecCtx

Associating metrics to patterns

- Patterns: right granularity (?)
- Bringing measuring process closer to application semantics
  - Better integration in development cycle
  - Application state can be monitored more closely
- Core versus ecosystem
  - Some metrics provide feedback on *core* system security
  - ...others on the *hostility* of your environment (ecosystem)
- Proactive (state) versus reactive (event) measuring
  - Metrics can be associated to architecture or design as well, similar to code analysis approaches
- Enable aggregation and correlation
  - Combine metrics to form indicators for each objective
  - Flexibility through correlation algorithm (risk posture)

Examples of patterns and metrics

- Input guard, output guard
  - #guards vs. #access points for each component
  - #filtering incidents per invocation
  - discrepancies between output guard and input guard results
- Audit interceptor
  - #service invocations vs. #audit events
- Application-level firewall
  - #firewall invocations vs. #service invocations
  - #denied connections
- Secure object creator
  - #illegal access errors (incorrect privilege matching)

Preliminary results

- Possible to attach at least one metric to each pattern.
- Different types of metrics (see image):
  - depending on the security goal (CIA...)
  - core versus ecosystem
- Valuable contextual info can readily be obtained

	core		ecosystem
confidentiality	firewall/service invoc.		denied fw connections
integrity	i-/o-guard discrepancies	o-guard filtering incidents nb.guards/ access points firewall/service invoc.	i-guard filtering incidents denied fw connections
availability			
anonymity			
accountability	nb. of audit events/invocation		
...			

Open issues and questions

- Next – first validation
  - Prototype / PoC
  - Need for loss databases, reference tests
- Does this approach make sense?
- Are all applications suited for this approach?
- Aggregation/correlation
  - Possibility to combine metric values into indicators – how?
  - Similarity to IDS problems