SECURITY-BY-DESIGN METHODOLOGIES AND SECURITY METRICS

Prof. Valentina Casola
University of Napoli Federico II, Italy

SWForum.eu - The Way Forward: Workshop on Future Challenges in Software Engineering – Milan, June 2023

Rationale

- **Objectives**: provide an overview of the main challenges behind quantitative security evaluation and a secure system development process and present ongoing research activities to reduce the complexity of security management

- **Outline:**
  - Security-as-a-Service opportunities and challenges
  - Security Service Level Agreements and Security metrics
  - A SLA-based security-by-design development process
  - Final remarks

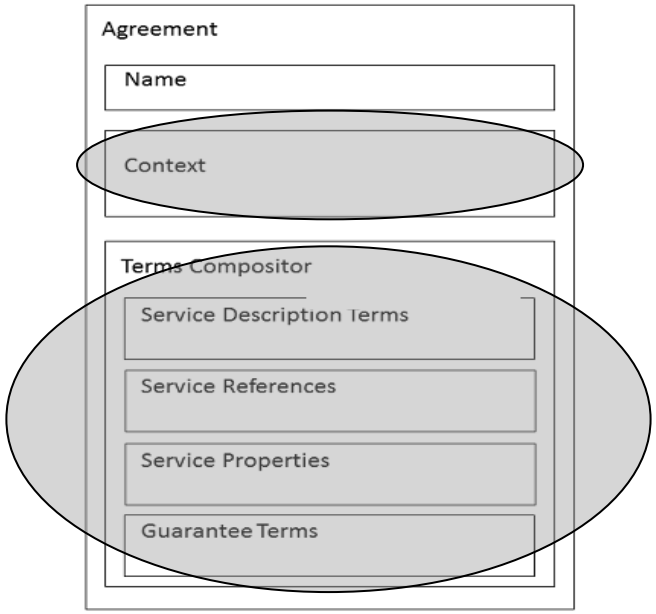
Security-as-a-Service: what is missing

- Security-as-a-Service
  - We need to deliver security capabilities as-a-service (offered by third parties CSP, in a multi-cloud environment, too)
- Security Services and SLAs
  - Security services need to be guaranteed under the control of security SLAs (todate, providers offer not-negotiable SLA with few security controls)
  - How to represent Security SLAs and measurable guarantees?
  - How to enforce and continuously monitor?
  - How to measure security?

What is a SLA

- A **Service Level Agreement** is a "Contract" which describes the Service, the associated quality levels and specifies the responsibilities (typically 'soft' formal obligations!) of both the Provider and the Customer.

- **Security SLAs** are contracts among CSP and CSCs regulating the security level granted over provisioned services



Requirements

- Define Security terms according to standards and known best practices, understandable by both CSC and CSP
- Security terms must be measurable and verifiable for both CSC and CSP

- ***Is security measurable?***

- Security SLA puts the question from a different perspective:
  - What is possible to measure in security?
  - What is possible to grant on such measured value?

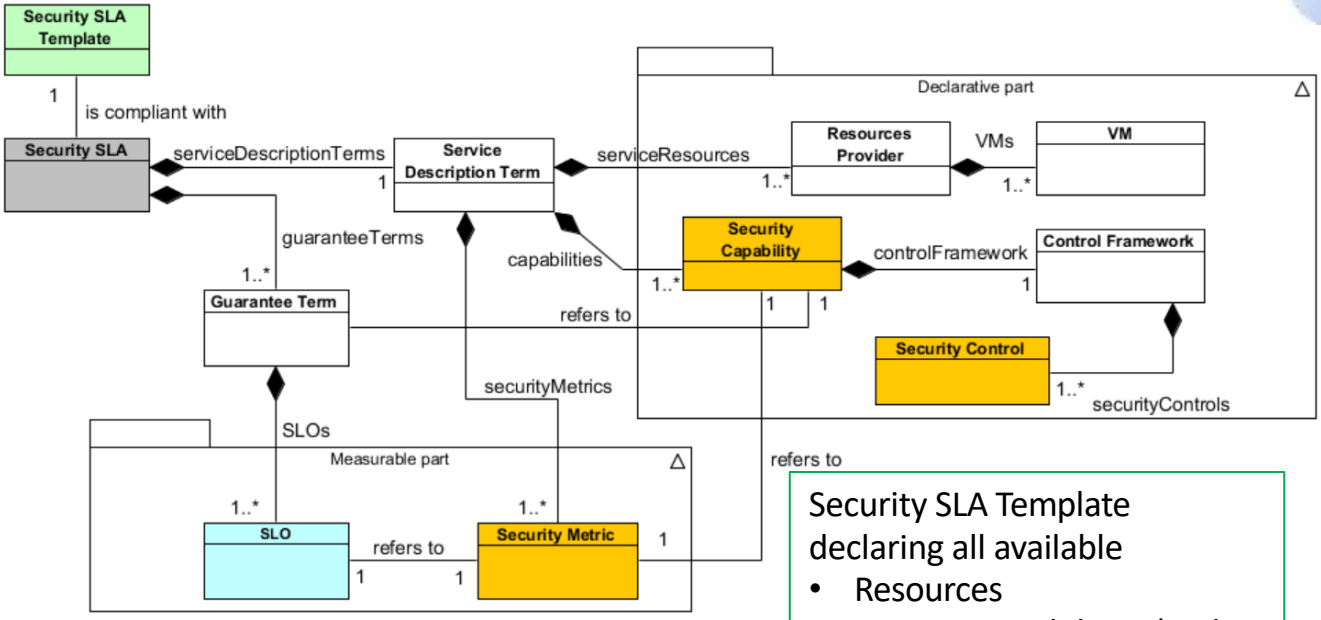
The idea is to define Security Metrics that give evidence of right application of security controls and so, YES, we can measure and grant security.

The SPECS proposed approach

- ***Security Terms*** expressed trough a declaration of ***Security Controls*** implemented by CSPs (derived from standard frameworks ISO 27001, NIST SP-800-53, CSA CCM,…)

- ***Service Level Objectives*** defined through ***Security Metrics*** associated to the declared Security Controls (derived from CIS Metrics, NIST metrics, SLALOM metrics, SPECS&MUSA Metric catalogue)

The SPECS Security SLA model

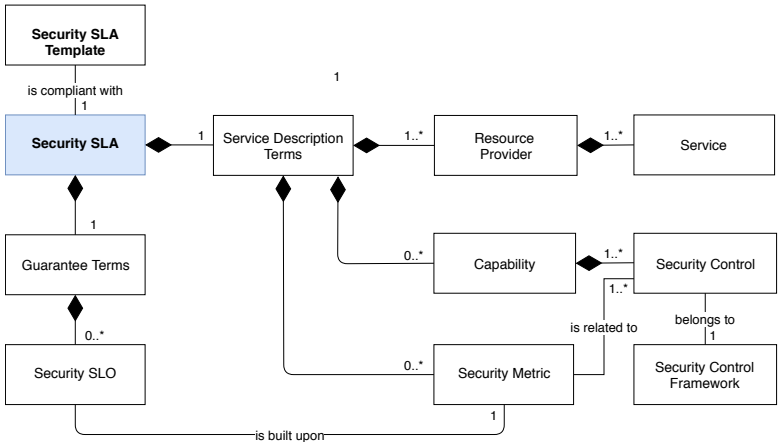


Supported Control Frameworks: NIST 800-53; CCM-CSA; (ISO e CC ongoing)

Security SLA Template declaring all available

- Resources
- Security capabilities (and related security controls)
- Security metrics

The proposed Security Metric catalogue - example



- ***Security requirements*** expressed in terms of standard ***security controls*** (from international frameworks).
- ***Security SLOs*** expressed in terms of ***security metrics***.

#	Metric	Values	Description
1	USR_AUTH_BEHAV_CHG	bool	User Authentication Behavior Change
2	ACC_CONTR_CORRECT	bool	This metric ensures that all access control rules are respected.
3	ACC_CONTR_LOGGING	bool	This metric ensures that all tentative of access are logged

#Metric	#SC
1	AC-7, AC-9, AC-9-4
2	AC-9-4,AC-9,AC-7
3	AU-1,AU-2,AC-7

SLA-BASED SECURITY-BY-DESIGN DEVELOPMENT PROCESS

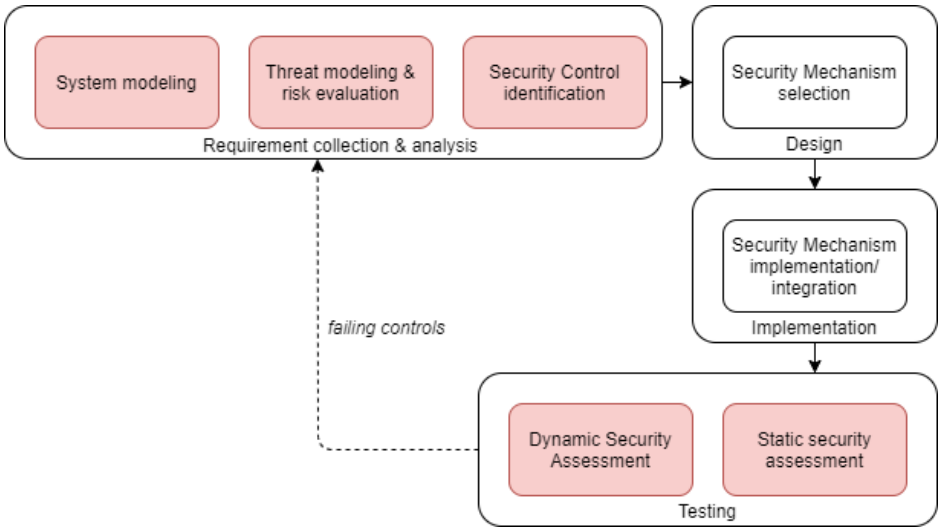
*"**Secure by design**, in software engineering, means that the software has been designed from the foundation to be secure…. the alternate security tactics and patterns are first thought; among them, the best are selected and enforced by the architecture design, and then used as guiding principles for developers"*

A Security-by-Design development methodology – Required features

- Needs of ***models and quantitative metrics*** to enable the security-by-design approach and take secure-informed choices
- Needs of ***automated mechanisms*** to support developers and tester in the development life cycle (security design, implementation, security assessment/testing)
- Can be integrated with common ***agile methodologies*** (e.g. SCRUM)
- Easy to be ***adopted by devops teams*** (mainly developers and tester, not security experts)

A Security-by-Design development methodology

- The methodology devises a guided risk analysis process and a partially automated (static and dynamic) security assessment

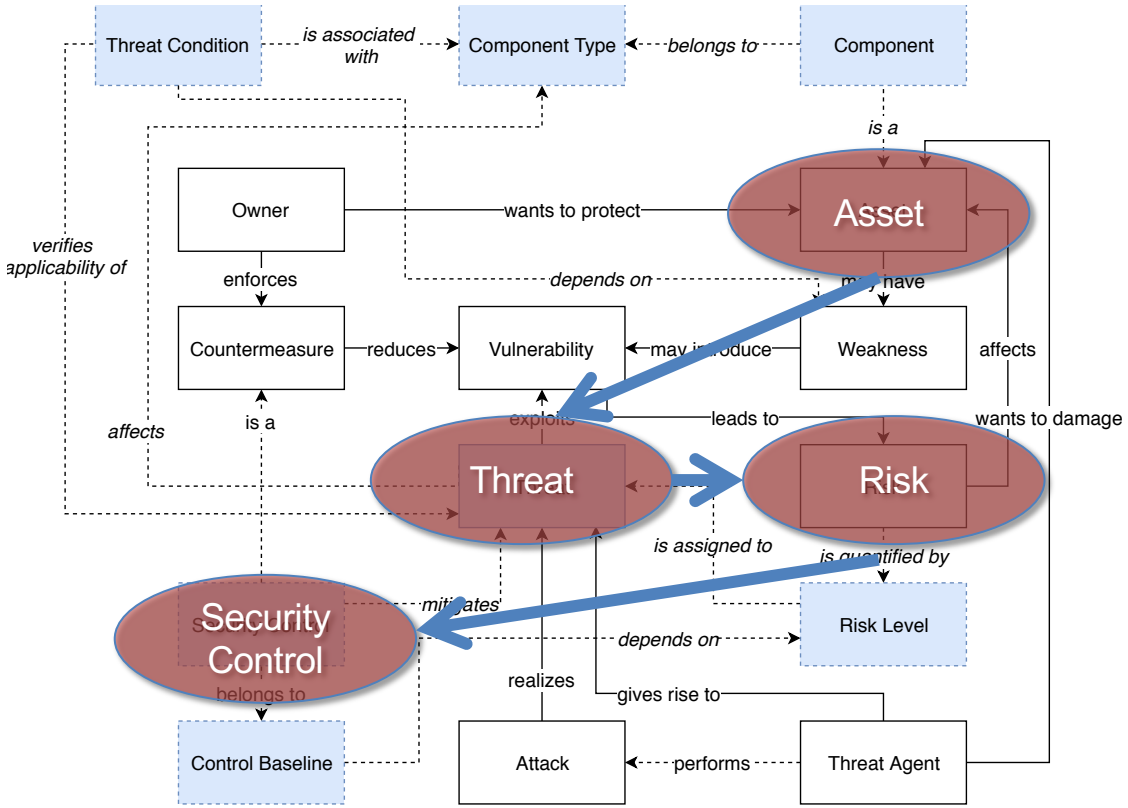


V. Casola, A. De Benedictis, M. Rak, U. Villano (2020). A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach . Journal of Systems and Software
V. Casola, A. De Benedictis, M. Rak, G. Salzillo (2020) A Cloud SecDevOps Methodology: From Design to Testing. In: Proceedings of the Int. Conf. On Quality of Information and Communications Technology. QUATIC 2020.

Supporting models

- **An enriched security system model**
  - to support the risk analysis and the continuous security assessment
  - based on open catalogues of component types, threats, security metrics,….
- **The security SLA model**
  - Including security controls and security levels to guarantee (metrics)
- **The MACM:**
  - Modelling a cloud application as a set of interacting components, hosted by different cloud service providers,
  - Modelling the security controls and their configurations with Security service level agreements,
  - evaluate how the composition of different services and their deployment in different environments may affect the security granted by the application.

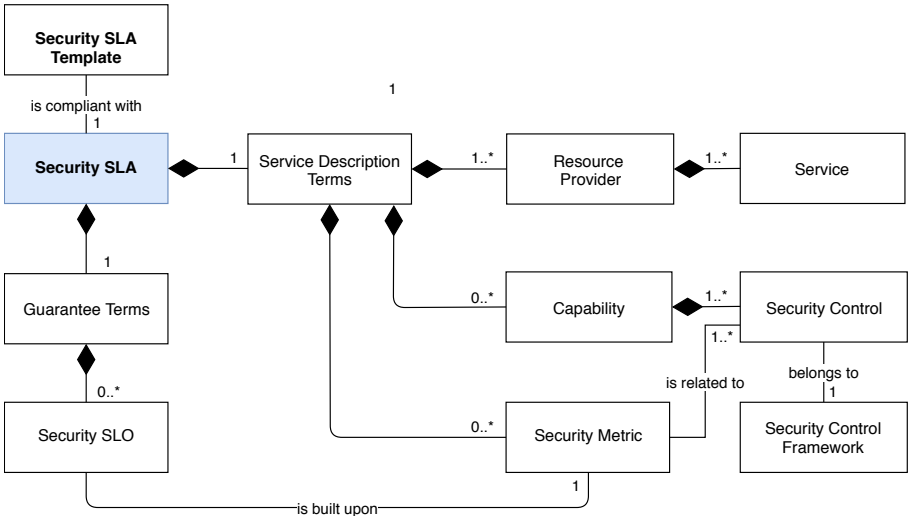
The Reference Security Model



- The methodology leverages a complex security data model that includes and correlates the main concepts involved in the design and assessment phases
- The data model is implemented by an open knowledge base (**Threat Catalogue**)[5] that currently includes more than 150 well-known threats against different component types belonging to web-based, cloud, IoT, edge applications mainly gathered from standards, open repositories and scientific papers

[5] http://bitbucket.org/cerict/sla-model

The security SLA model

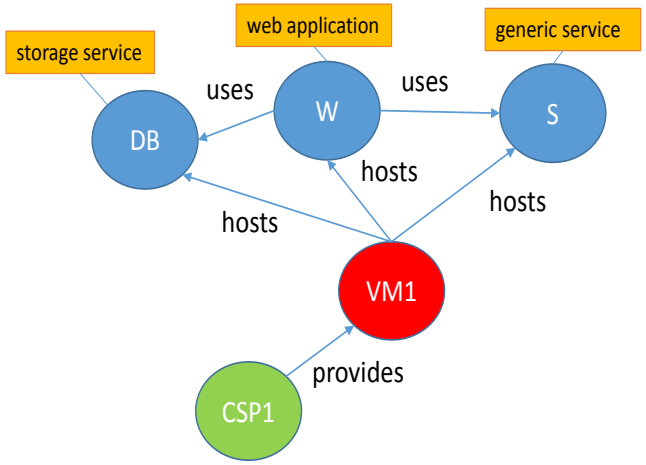


- ***Security requirements*** expressed in terms of standard ***security controls*** (from international frameworks).
- ***Security SLOs*** expressed in terms of ***security metrics***.

#	Metric	Values	Description
1	USR_AUTH_BEHAV_CHG	bool	User Authentication Behavior Change
2	ACC_CONTR_CORRECT	bool	This metric ensures that all access control rules are respected.
3	ACC_CONTR_LOGGING	bool	This metric ensures that all tentative of access are logged

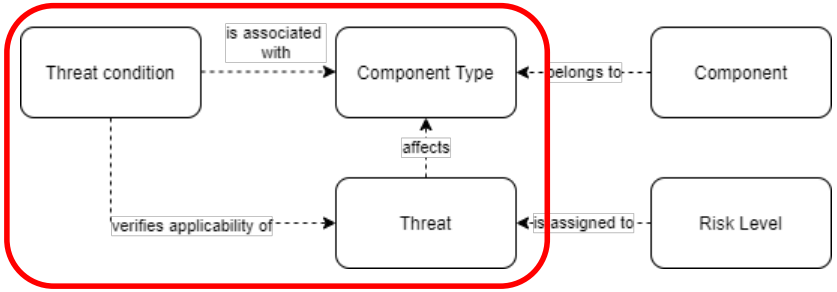
#Metric	#SC
1	AC-7, AC-9, AC-9-4
2	AC-9-4,AC-9,AC-7
3	AU-1,AU-2,AC-7

System modeling



- System modeling leverages a graph-based formalism named **MACM** (Multi-Cloud Application Composition Model), which enables to describe the high-level architecture of a system in terms of its components and their interconnections
- System components include both logical software modules implementing the business logic of the system and deployment resources (such as physical or virtual machines), where the logical modules execute

Threat modeling

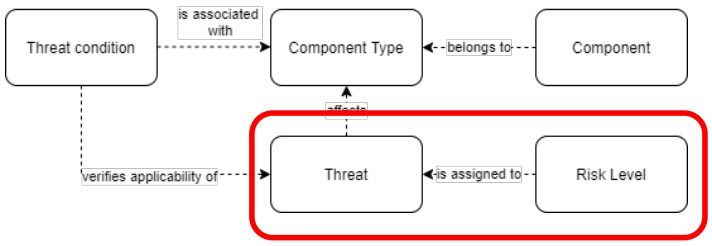


- Threats are automatically retrieved by a Threat Catalogue, based on the assets involved in the system
  - Threats are grouped based on the popular STRIDE classification
  - A refinement step can take place leveraging an ad-hoc questionnaire

Threat	STRIDE cat.	Condition
Account Hijacking	SPOOFING	Does the component maintain account information?
Cross-Site Request Forgery (CSRF)	TAMPERING	Does the component accept requests without checking their origin and trustworthiness?
Cross-Site Scripting (XSS)	TAMPERING	Are the inputs from users (forms, http requests) directly used without validation?
Man in the Middle attack	SPOOFING	Does the component communicate with other components without ensuring the authenticity and integrity of the communications?
Buffer overflow exploitation	INFORMATION DISCLOSURE, DOS, TAMPERING	Does the component allow to write on memory without restrictions?
Missing Function Level Access Control	ELEVATION OF PRIVILEGES	Does the component expose different functions to different users based on their access rights by only differentiating the user interface (without performing an actual authorization at each request)? Does the component manage functions/data with different access rights?

Risk evaluation

- The risk evaluation step enables to rate the level of risk associated with each threat identified by the previous task by automating the OWASP Risk Rating Methodology, which takes into account 16 different parameters related to the **likelihood** and the **impact** of threats
  - Likelihood factors and technical impact factors are pre-set with default values
  - Business impact factors are evaluated for groups of threats

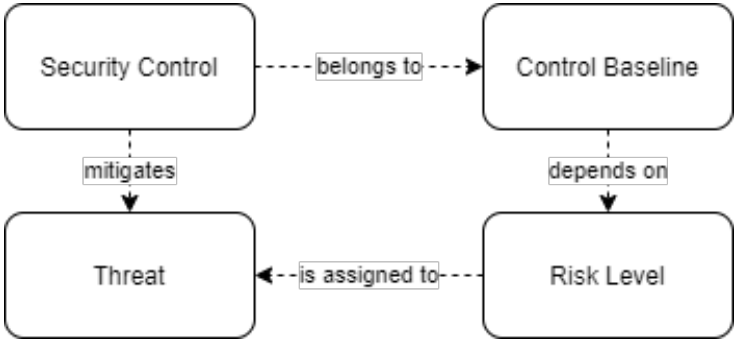


OVERALL RISK SEVERITY = HIGH

LIKELIHOOD = 4.875				IMPACT = 7			
Threat Agent Factors		**Vulnerability Factors**		**Technical Impact Factors**		**Business Impact Factors**	
Skill level	9	Ease of discovery	3	Loss of confidentiality	9	Financial damage	0
Motive	4	Ease of exploit	3	Loss of integrity	9	Reputation damage	0
Opportunity	4	Awareness	4	Loss of availability	1	Non-compliance	0
Size	9	Intrusion detection	3	Loss of accountability	9	Privacy violation	0

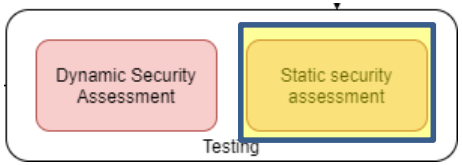
Security Controls identification

- Security control identification is devoted to identifying the **countermeasures** to adopt, in terms of the security controls to enforce (belonging to the NIST Security Control Framework), in order to mitigate existing threats against considered assets, based on the actual risk level.
  - First, all the security controls associated with the threats affecting a component are retrieved
  - Then, for each threat, the risk level is considered: only controls belonging to the baseline that matches (i.e., is equal or lower than) such level of risk are kept



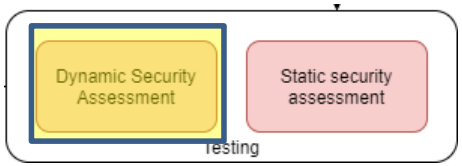
NIST. SP 800-53 Rev. 5. Security and Privacy Controls for Information Systems and Organizations. 2020 (last version)

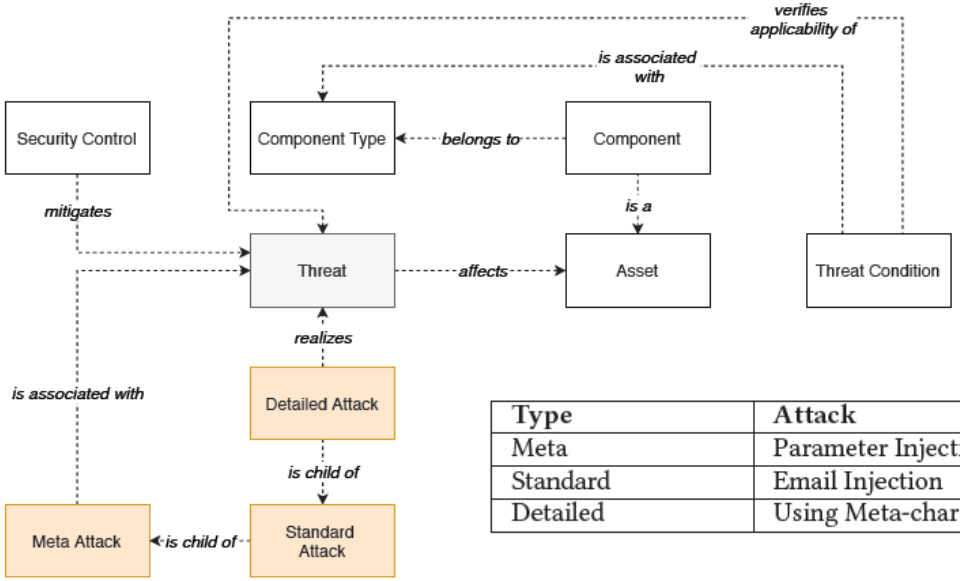
Static assessment



- Static assessment verifies that components are properly implemented and configured to correctly enforce the security controls identified during the analysis stage
  - **Per-component assessment**: performed by means of a **code review approach** based on ad-hoc questionnaires
    - *questionnaires* list the checks to perform on the code of the application to verify that each needed security control is in place
    - the output of the process is the list of the controls that result correctly implemented when considering each component in isolation
  - **Per-application assessment:** suitably **combines the security policies** implemented by each component by taking into account the existing component dependencies and the impact of deployment choices, in order to identify the set of security controls that can be declared as correctly implemented and that can be actually granted by the application as a whole
    - the assessment leverages a **reasoning** process that takes into account the architecture of the application, the components' security policies (after static assessment) and simple logic rules built ad-hoc for each control or control family
      - ✓ deployment components' security policies can be retrieved from respective security SLAs, when available (done for cloud services by leveraging existing initiatives from CSA)

Dynamic assessment

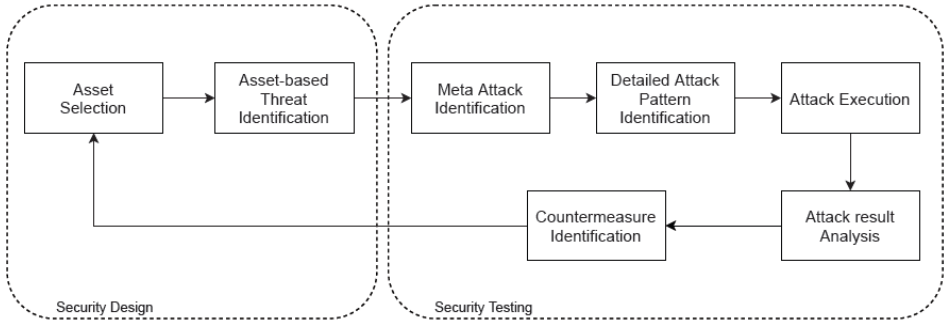




- Dynamic security assessment consists of a guided risk-driven **penetration testing** activity, planned according to the results of the analysis phase
- An attack plan is built considering the mapping between threats and attacks according to the CAPEC

Type	Attack
Meta	Parameter Injection
Standard	Email Injection
Detailed	Using Meta-characters in E-mail Headers to Inject Malicious Payloads

https://capec.mitre.org/



Security Catalogues

- Metric Catalogue
  - Catalogue of security Metrics
  - Mapping security metrics / security controls
- Threat Catalogue
  - Catalogue of Threats (Cloud/IoT)
  - Mapping Threats /Asset Types
  - Mapping Threats / Security Controls (countermeasures)
- Planning Catalogue for implementation and testing
  - Catalogue of automated planned actions
  - Mapping Attack Actions/Threats and Asset Types
- Work-In-progress
  - Mapping with MITRE CWE, CVE, CAPEC and ATT&CK

Conclusions –
Security metrics are the keys towards a comphrehensive approach to security-by-design and fully effective secdevops

- **Guarantee security** is possible trough the adoption of security SLA.
- The security-by-design approach is leveraged by fostering **automation** in all the secure development phases
- Thanks to security models and knowledge based, the **security skills** of developers to perform the process can be **limited**
- We discussed a possible vision of a comprehensive **risk-based approach to security development** and saw that some tasks can be automated only in part and there is still a lot of work for the research community
- Building secure software is a never-ending challenge!
  - The watchwords of the day are **EASE&SPEED**

References

1. Casola, V., De Benedictis, A., Rak, J. Modic, M. Erascu (2016) “**Automatically Enforcing Security SLAs in the Cloud**”. In IEEE Transactions on Services Computing, Vol. 10 (5), 741-755.
2. Valentina Casola, Alessandra De Benedictis, Massimiliano Rak, Umberto Villano (2020). **A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach**. In Journal of System and Software. Volume 163.
3. Casola V., Mazzeo A., Mazzocca N., Vittorini V. (2007). “**A policy-based methodology for security evaluation: A Security Metric for Public Key Infrastructures**”. In Journal of Computer Security, vol. 15, pp. 197-229.
4. Casola V., Fasolino A.R., Mazzocca N., Tramontano P. (2009)" **An AHP-based Framework for Quality and Security Evaluation** " in IEEE Proceedings of CSE 09, August 2009, Vancouver, Canada.
5. Valentina Casola, Alessandra De Benedictis, Massimiliano Rak, Umberto Villano (2019). **Toward the automation of threat modeling and risk assessment in IoT systems**. In Internet of Things Vol.7.

Contacts

Prof. Valentina Casola

University of Napoli Federico II, Napoli, Italy

e-mail: casolav@unina.it