APPLICATION THREAT MODELING

APPENDIX – PROCESS FOR ATTACK SIMULATION AND THREAT ANALYSIS

Marco M. Morana

WILEY

Contents

Appendix – process for attack simulation and threat analysis 0

Application Threat Modeling Process 4

Introduction to PASTA™ Process for Attack Simulation and Threat Analysis 4

Stage I - Definition the Objectives (DO) for the Analysis of Risks 11

DO 1.1 - Obtain the business requirements 13

DO 1.2 – Define the data protection requirements 13

DO 1.3 - Identify standards and regulatory compliance obligations 13

DO 1.4 - Identify the privacy laws 13

DO 1.5 - Determine the initial risk profile 13

DO 1.6 – Define risk management objectives 14

Stage II – Definition of the Technical Scope (DTS) 16

DTS 2.1 - Capture the technical details 17

DTS 2.2 - Assert completeness of technical documentation 18

Stage III - Application Decomposition and Analysis (ADA) 19

ADA 3.1- Decomposition of the application 20

ADA 3.2 – Security controls assessment 21

ADA 3.4 – Functional analysis 21

Stage IV - Threat Analysis (TA) 22

TA 4.1 - Document threat scenarios 24

TA 4.2 - Update the threat library 24

T4 4.3 - Assign probability to each threat 24

TA 4.3 - Map threats to security controls 25

Stage V - Weakness and Vulnerability Analysis (WVA) 26

WVA 5.1 – Query existing vulnerabilities of security controls 28

WVA 5.2 - Map threats to security control vulnerabilities 28

WVA 5.3 - Calculate risk severity to vulnerabilities 28

WVA 5.4 - Prioritize security controls for vulnerability testing 29

Stage VI - Attack Modeling & Simulation (AMS) 30

AMS 6.1 - Model the attack scenarios 34
AMS 6.2 - Update the attack library 34
AMS 6.3 - Identify the attack surface and enumerate the attack vectors 34
AMS 6.4 - Assess the probability and impact of each attack scenario. 35
AMS 6.5 - Derive a set of cases to test existing countermeasures. 35
AMS 6.6 - Conduct attack driven security tests and simulations 35
STAGE VII - Risk Analysis & Management (RAM) 37
RAM 7.1 – Calculate the risk of each threat. 38
RAM 7.2 – Identify the countermeasures. 39
RAM 7.3 – Calculate the residual risks. 39
RAM 7.4 – Recommend strategy to manage risks 39

List of Tables and Figures

No table of figures entries found.

APPLICATION THREAT MODELING PROCESS

"Risk is about uncertainty or, more importantly, the effect of uncertainty on the achievement of objectives. The really successful organizations, work on understanding the uncertainty involved in achieving their objectives and ensuring they manage their risks so as to ensure a successful outcome"

Kevin Knight, Chairman of ISO Working Group that developed ISO 31000

Introduction to PASTA™ Process for Attack Simulation and Threat Analysis

The goal of this chapter is to demonstrate the Process for Attack Simulation and Threat Analysis (PASTA) by walking through each stage of the process and by providing guidance on how to conduct the various activities that the process entitles to. "P" in the PASTA acronym stands for Process. As a process, PASTA describes a set of process events, or stages, the recommended set of process inputs and the expected process outputs that are produced as outcome of the execution of the activities of each stage. The emphasis of "process" and not on "framework" is important as well as the emphasis on the main activities that can be followed for the execution of this process.

As a process, PASTA is executed in a specific context that is the context of application security and risk management but PASTA per definition is not risk assessment methodology it is rather a process that can assist organizations in managing the various technical and non-technical risks caused by threats seeking to exploit vulnerabilities in applications. The focus of this process is the simulation of attacks and the analysis of threats.

Before we analyze more in depth the different stages of PASTA it is important to explain the basic terminology that is used for threats, attacks and vulnerabilities. First of all the notion of attack implies the understanding of the notion of threat. For threats, threat agents, attacks and vulnerabilities we will use standard definitions from the National Institute of Standards and Technologies (NIST)

According to NIST SP 800-37 a threat is *”Any circumstance or event with the potential to adversely impact agency operations (including mission, functions, image, or reputation), agency assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. (CNSS Inst. 4009, Adapted)”.* A threat is often associated to a threat source that is according to NIST SP 800-37*:“Either: (i) intent and method targeted at the intentional exploitation of vulnerabilities; or (ii) a situation and method that may accidentally trigger vulnerabilities.* The other essential element of PASTA is the threat analysis, Threat analysis as for NIST SP 800-30 consist on ”the examination of threat-sources against system vulnerabilities to determine the threats for a particular system in a particular operational environment”. In the NIST definition of threats there is a notion of vulnerability. A vulnerability is defined in several of NIST special publications, the one that is closer to the concept of vulnerability in PASTA is found in NIST SP 800-30: as” *a flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised (accidentally triggered or intentionally exploited) and results in a security breach or a violation of the system's security policy.”*

In the definition of threats there is a mention of vulnerabilities: it is important not to confuse the two: threats represent a potential for damage while vulnerabilities represents a condition for realizing that damage. Threats represent a potential for a bad event occurring in the presence of a threat environment and of threat agents. For a web application that process credit card data for example, a threat environment is represented by the internet, since it is potentially exposed by different threat agents that characterize that threat environment. These threats are typically referred as cyber-threats and might include threat agents with different capabilities, methods and objectives. Examples of threat agents that characterize the threat environment today are cyber-criminals, cyber-spies, cyber-hacker, fraudsters and hacktivists.

Vulnerabilities are weaknesses in security controls that these threat agents might seek to exploit such as when targeting an application.
Examples of vulnerabilities include weaknesses in the application controls caused by how these controls have been designed, implemented or configured as well as in vulnerabilities in the software components that are used by the application, the operative systems software that the application uses and the vulnerabilities of the underlying network and data infrastructure in which these applications operate.

Threats and vulnerabilities have also a different meaning from a perspective of an attacker or a defender. From a perspective of an attacker to be a threat agent represents a capability to conduct an attack against a target, to be part of a group such as a cyber-gang and have access to resources such as money to invest in the development of cyber-attack tools and malware to attack applications and systems. For the defender perspective, a threat represents a risk of a potential negative impact for the organization/business. For a threat agent vulnerabilities are opportunities to attack an application to achieve specific goals such as stealing confidential information. For a defender, vulnerabilities represent security issues that need to be fixed before an application is put into production so these cannot be exploited by a threat agent.

In the definition of threats besides the notion of vulnerabilities there is also a notion of attack. Attacks should not be confused with threats: attacks describe how threats can be realized to cause an impact. A standard definition of attacks is the one covered in NISP 800-28v2: *“Attacks are the realization of some specific threat impacting the confidentiality, integrity, accountability, or availability of a computational resource.”* For example the realization of an information disclosure threat can be described by a SQL injection attack where a threat agent will seek to the exploit SQL injection vulnerabilities in a payment processing application. The attacker will first probe the application web pages with a vulnerability scanner and then manually inject SQL commands in the application web pages in the attempt to alter the SQL query statement. The attacker goal is to gain unauthorized access to confidential information such as credit and debit card data stored in the database. From the defender perspective the modeling of an attack helps to identify which type of security measures should be deployed and where should be deployed to detect the various attack events and to identify countermeasures that are effective in protecting from the various attack vectors targeting the application.

Threats, vulnerabilities and attacks are the basic factors for the assessment of risk. Risk is often associated with factors of probability (or occurrence) and impacts (e.g. technical and business). NIST 800-33 provides a standard definition of risk “*The probability of a particular threat-source will exercise (accidentally trigger or intentionally exploit) a particular information system vulnerability and the resulting impact if this should occur.*” In this NIST standard definition of risk there is emphasis on the probability of a threat source to exercise a vulnerability to cause an impact.

Threat analysis and attack simulation are critical activities to determine the factors of probability of the occurrence of a threat to successfully conduct attacks seeking to exploit vulnerabilities. For example a threat probability can be associated with the occurrence of threat events (based upon past events) such as previously observed security incidents and vulnerability exploits. Threat probabilities can also be associated with the inherent characteristics of the threat agent such as his motivations in attacking certain targets and the capabilities and opportunities to conduct these attacks. The attack probability can be associated with .the modeling of different paths of attacks within an attack tree: some attack paths are subject to possible choices of attacks that can be pursued by an attacker when these have higher probability of success of the attacker objectives by minimizing the costs of the tools and computer resources that are required to conduct them.

The determination of the factors of likelihood and impact is the essence of risk assessment. A standard definition for risk assessment can be found in NIST SP 800-30, "*risk assessment is the process of identifying the risks to system security and determining the probability of occurrence, the resulting impact, and additional safeguards that would mitigate this impact".* The assessment of the impacts depends on the different factors given to the impact such as technical and business. A technical impact is what is assessed as a risk of an exploit of vulnerabilities in a security measure/control that exposes the assets (e.g. data, functionality) to partial or total loss of confidentiality, integrity and availability of these assets. A business impact consists on either tangible (e.g. monetary losses) or intangible (e.g. loss of reputation/image) impact.

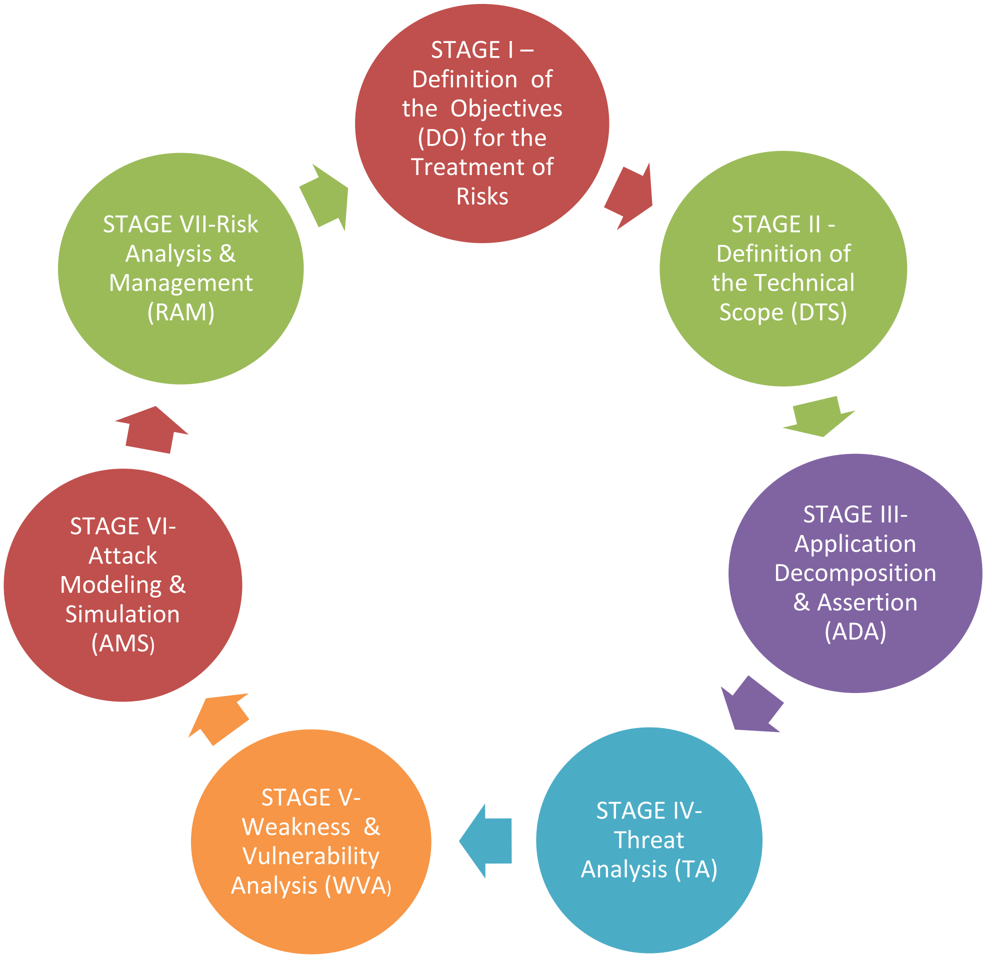
The assessment of risk includes the determination of the level of risks as qualitative level (e.g. HIGH, MEDIUM, LOW) as well as quantitative levels (e.g. $ 100, $ 100,000, $ 1 ML etc). Typically qualitative risk formulas that factor different levels and averages for likelihood and impact are used to determine the overall technical impacts while qualitative formulas that factor the occurrence of a security incident event and the monetary loss caused by that event are used for the calculations of business risks. Once technical risks and business risks are assessed using the various risk calculation formulas that are part of the organization risk assessment process, the next step is to determine how to manage risks.

The standard definition of rusk management is NIST 800-30" *The total process of identifying, controlling, and mitigating information system-related risks. It includes risk assessment; cost-benefit analysis; and the selection, implementation, test, and security evaluation of safeguards. This overall system security review considers both effectiveness and efficiency, including impact on the mission and constraints due to policy, regulations, and laws.*

A more compressive standard definition of risk management includes the implementation of a risk mitigation strategy as defined in NIST SP 800-53r2*: "The process of managing risks to organizational operations (including mission, functions, image, or reputation), organizational* *assets, or individuals resulting from the operation of an information system, and includes: (i) the conduct of a risk assessment; (ii) the implementation of a risk mitigation strategy; and (iii) employment of techniques and procedures for the continuous monitoring of the security state of the information system. (FIPS 200)"*

In alignment of the industry accepted standard definitions for threats, threat agents, threat analysis, vulnerabilities, attacks and risks assessment and management the process of PASTA can be defined as the following: ***"A risk management process aimed at considering possible threats scenarios, attacks and vulnerabilities within a proposed or existing application environment for the purpose of assessing the risks and managing the impacts (technical and business impacts)"***

As a risk management process (PASTA)™ consists on seven stages. Each of the stages has specific objectives and expected deliverables. For each stage there is a description of the set of activities that can be followed to accomplish the goals of teach stage.


STAGE I – Definition of the Objectives (DO) for the Treatment of Risks
STAGE II - Definition of the Technical Scope (DTS)
STAGE III- Application Decomposition & Assertion (ADA)
STAGE IV- Threat Analysis (TA)
STAGE V- Weakness & Vulnerability Analysis (WVA)
STAGE VI- Attack Modeling & Simulation (AMS)
STAGE VII-Risk Analysis & Management (RAM)

Stage I – Definition the Objectives (DO) for the Analysis of Risks

Goal: Identification of the business objectives and information security policies, applicable compliance and data privacy requirements in scope for the application/product. Definition of technical and business objectives for risks analysis (includes analysis of the likelihood of threats and technical and business impacts assessments)”

Guidance: the stage I of PASTA™ “Defining the Objectives (DO) of the Analysis” consists on set of activities whose main goal is to define the main objectives of the analysis of cyber-threats, modeling of attacks and assessments of risks that these threats pose to the application in scope. Initially these objectives are defined in the context of the organization from the defensive perspective and based upon information security policy, regulatory compliance, privacy and requirements for the managing of business risks (e.g. business continuity and financial impact) . The objective of the stage is to define the objectives for the treatment of risk that also include information security requirements, compliance requirements, data privacy requirements and risk management requirements in the scope of the web application. Prerequisite for this stage is the reference to information security policies and standards, regulatory compliance, privacy laws and business requirements. Since information security policies and regulatory compliance apply also to web application assets that are owned and managed by the organization, it is important to review also application domain specific security standards if these are available. The outcomes of this stage are:

1) The initial security and risk profile of the web application in scope for the analysis;
2) A set of documented high level objectives for managing application risks including non-compliance with information security, unlawful regulatory non-compliance, business continuity risks, financial risks and security incident risks of the web application.

Specific information security, compliance and risk mitigation requirements can be documented during the design phase of the SDLC for both new applications of for application changes. Business risks are derived from a preliminary risk analysis of potential business impacts based upon the initial classification of the confidentiality of the data and the value given to the application as an asset for the organization. In order to be ready to execute this stage, it is mandatory that an organization has adopted information security standards and policies that for the protection of the data assets that are processed by software and applications. For compliance perspective this includes applicable requirements from high level information security standards (e.g. ISO 27001) and industry specific requirements (e.g. PCI-DSS). For the security risk management perspective, the definition of risk mitigation objectives includes the definition of the inherent risks based upon the initial business-functional requirements and exposure of the application assets and the inherent value of the asset. At high level, a risk statement objective can be "the web application might expose high financial risk assets to external threat agents for $$ millions and this is considered an HIGH financial risk ought to be prioritized for risk management" helps to translate the initial risk ratings to high level risk management objectives. These high level information security, compliance and risk management objectives constitute the initial set of high level requirements that can be followed both in the design document of a web application as well as to assess the risks by walking through the different stages of PASTA. During this stage it is also important to consider at high level who the severity level of cyber threat as received from threat advisories (e.g. in the case of the financial sector this will be FS ISAC) . Since threats will be later analyzed in STAGE IV of PASTA™, the initial risk management objectives for these threats are mostly derived what is known initially about the existing application/product or the new features that will be introduced in the application/product. This information would be challenging to collect when the application or product is new therefore this is based upon an estimate that would need to be revisited later when all the details of the application/product can be fleshed out. The objective of the initial risk profile is to identify, at high level, the inherent technical and business risks where the technical risks depend upon the risk of losing confidentiality, integrity and availability of the data assets while the business risks depend on the value given by the business to these assets whose exposure for the business in the case these assets are lost/compromised include a business impact.

Activities: the objectives of this stage of the process can be achieved through the following six activities:

DO 1.1 - Obtain the business requirements for the new product or application. This objective of this step is to derive a list of high level requirements that describe the functionality of the application.

DO 1.2 – Define the data protection requirements. At high level these are asset protection requirements for the protection of confidentiality, integrity, availability and accountability of the data and functions that are accessed, processed and stored by the application.

DO 1.3 - Identify standards and regulatory compliance obligations in scope. Compliance can be determined by the type of data and functionality provided by the application. For example, if the application handles credit card payment data, the application is in scope for PCI-DSS and compliance requirements for masking credit card data and encrypting during transmission and storage.

DO 1.4 - Identify the privacy laws based upon country and type of business and derive specific requirements for the protection of personal identifiable information.

DO 1.5 - Determine the initial risk profile for the application/product based upon the known information about the application operational environment and inherent risks such as information security risks (based upon exposure, confidentiality of the data and volumes), non-compliance risks, loss of privacy risks and business impact risks (based upon the level of estimate of the loss of revenue) for the application/product. Include in the risk profile also the severity level of the cyber-threat as probability of the application product to be attacked. Estimate and assign risk impact levels for threat severity-probability of attacks, technical impacts and business impacts (H, M, and L)

DO 1.6 – Define risk management objectives for the analysis and management of risk that includes the analysis of the risk of cyber-threats and their probability (use initial severity levels of threat as estimate) and the determination of the technical impacts (loss of confidentiality, integrity and availability of the data). The analysis also includes the analysis of the impact to the business in the case of a security incident ought to occur (based upon cyber-threat probability) to cause tangible (e.g. revenue-monetary loss, fraud loss, operational losses etc.) and intangible losses (e.g. image reputation, brand) to the business also in consideration of losses also because of compliance fines, data privacy loss fines, license liabilities and legal costs lawsuits.

Inputs: Information to assess information security, compliance and business and technical profile for the web application in scope and set the objectives (high level requirements) for the analysis such as:

1. Information Security Policies and technical information security standards
2. Application information security standards
3. Industry applicable compliance regulations
4. Consumers and users applicable data privacy laws
5. Business impact analysis calculation methods
6. Risk assessment processes
7. Technical and business risk impact calculation methods
8. Security incident response procedures
9. Business continuity plans
10. Asset IT inventory

Outputs: the following outputs are generated by the execution of this stage:

Documented Application Profile including the following information:

a) Application/product type and description
b) Application/product operational exposure (internal vs. external)

c) Applicable Information security publicity and technical standards scope
d) Applicable regulatory compliance laws and standards
e) Applicable data privacy laws (list of applicable privacy laws, country and business specific)
f) Threat severity risk level (HIGH, MEDIUM, LOW) as probability of an attack (based upon ISAC information and threat advisories. Can be considered as cyber-Threat Risk Level (e.g. this is the risk of potential cyber-attacks, depends on threat advisory for the organization, security incident reports, threat intelligence information) and the security incident response plans (applicable SIRT plans)
g) Security risk impact level (HIGH, MEDIUM, and LOW): This is an estimate based upon a high level assignment of technical risk factors for the impacts due to loss of data and violation of information security policies and standards (e.g. depends on volumes of handling of confidential data, financial accounts)
h) Business risk impact level (HIGH, MEDIUM, LOW): This is an estimate based upon an estimated impacts of a security incident causing fines for noncompliance (depends on type of application and industry, the sensitivity of data processed/stored, the risk of transactions) to determine the scope (list of compliance regulations, industry-organization specific technical standards). Privacy risks impact such as loss of confidentiality of private data of clients/consumers (e.g. personal identifiable information) and the volume of data stored (e.g. hundredths of users, thousands of users, millions of users). For business impacts also need to factor the loss of revenue in the case of loss of availability of service based upon continuity plans in scope (e.g. business continuity plans).

Documented Risk Analysis Objectives: this is a set of high level objectives for the analysis and management of risks of the application whose risk profile was previously captured in the application risk profile. These objectives are defined in support of the business objectives in scope for the application/product:

a. Threat analysis objectives. These are objectives for the analysis of the threat agents such as capabilities, motives and attack vectors used and the determination of how these threats might be realized in attacks to cause a negative impact to the application assets
b. Technical risk impact analysis objectives. These based upon applicable information security policies and standards and impact risk levels estimated in the risk profile assuming the type of threats and the inherent information security risks
c. Business risk impact analysis objectives. These are objectives for the management of the risk levels estimated in the risk profile assuming the type of threats and the inherent business risks due to legal and regulatory risks for non-compliance, loss of data privacy and business impacts due to loss of availability and compromise of critical functionality at risk of security incidents caused by cyber-threats.

Stage II – Definition of the Technical Scope (DTS)

Goal: assert that the technical details of the application architecture are documented and contain all the necessary technical details for secure design of the application and for conducting the risk analysis of the application architecture.

Guidance: this stage objective is to capture the technical details of the application/product so we can assess later the risks and the technical impacts. The technical details need to be captured to define what should be in scope for the threat analysis. For the definition of the technical scope it is important to be able to collect these technical details from the engineering teams. Ideally these details should be captured in standard technical design documentation such as architecture design documents inclusive of architecture components, servers and network infrastructure dependencies. The technical design documentation should document the details of security controls such as authentication, authorizations, data protection in storage and in transit, data filtering, session management as well as details of the design patterns used and the architecture components, data interfaces and the trust boundaries of the application with other application software and software with server components, third party software libraries and frameworks, servers O.S. and the supporting data center, network infrastructure in support of the application and product in scope. Since threat modeling focuses on identifying how threats are realized in attacks against the various technical components of the application/product as well as the various components of the technology stack, it is important that this technical information is completed and documented with all the necessary details. These are the details necessary for conducting the threat modeling exercise.

At the project level, it is important to being able to capture the technical details of the design and featured changes introduced with new applications and products so it is possible to assert the technical impact of these changes. These are design changes are ideally documented in the technical documentation. As the technical details are captured through interviews, questionnaires and other methods (e.g. analysis of network topology, reverse engineering of code to determine libraries used etc.) they also need to be documented. The next step is to assert if any details are missing and need to be added to the design. Specifically, the security controls to protect the data assets need to be asserted and documented and any gaps in documentation of the security controls need to be identified. Any gaps in the technical documentation of security controls identified are typically addressed by documenting them during the design phase so these can be tracked for implementation and for security testing in the subsequent phases of the SDLC.

Activities: this stage consists on the following activities:

> **DTS 2.1 - Capture the technical details** of the application architecture and the details and of the design of security controls. The details of what should be documented can be defined in document templates that can be given to the application architecture teams so can be filled with all the details necessary for the architectural risk analysis. To determine which security controls should be documented and how, it is important to rely upon application security standards and guidelines and document templates that include the

mandatory sections of the document that need to be documented for the initial technical scope assessment from the information security team. These document templates can also include pointed guidance of which architecture diagrams should be documented and how and include an information security section with mandatory documentation of security controls (e.g. authentication, authorization, data validation, encryption, session management, audit and logging). It should responsibility of the application design architects to document all the mandatory sections of the architecture design and functional design document.

DTS 2.2 - Assert completeness of technical documentation including the technical details for the design of the security controls and the architecture. Assert technical details of security controls are documented in design documents (e.g. functional detail design, high level architecture design). Each design document should include a set of Information Security (IS) requirements as mandated in the template document. The type of classification of the data and the risk levels assigned to this data in the risk profile should be used to define IS requirements for protecting these assets (e.g. confidential data should be encrypted). An inventory of software libraries, type of servers and network devices used by the application should contain technical details for the management of vulnerabilities. The technical documentation should provide enough details for identifying and analyzing the presence of gaps in security controls such as design flaws.

The definition of what should be in scope for technical design depends on the risk analysis objectives that were previously set as high level objectives. For example if the web application exposes high risk functionality would require additional security controls such as Multi Factor Authentication (MFA). The requirement for MFA should be defined in the organization information security standards and the scope for the design of this control depends on the previously defined security requirements and the application risk profile (e.g. the web application is internet facing and either allows authenticated users the access to customer confidential data or to high risk transactions such as money movements and payments).

The technical scope assessment of a web application for example requires the documentation of logical and network architecture as well as of the application functionality that is exposed through the web and mobile channels. The enforcement of the user's roles and permissions drives the requirement for the design of authorization controls (e.g. role base access controls). The technical documentation should include specific security requirements for the web application in compliance with applicable information security standards and regulations.

Inputs: information to identify, capture and assert the technical scope of the risk analysis:

a. Information Security, Compliance and Risk Profile of the Web Application (from previous STAGE I)
b. Architecture design documents
c. Functional specification documents
d. Technology stack (from IT asset repository)
e. Network components (from IT asset repository)

Outputs: complete technical details for the secure design of the application architecture as well for the subsequent threat modeling and architectural risk analysis of the application.

Stage III - Application Decomposition and Analysis (ADA)

Goal: decomposition of the application in basic elements such as users, roles, data storages, and data flows, functions, security controls and trust boundaries. Decomposition in functional components and analysis of security controls to protect the functionality provided by each component"

Guidance: the purpose of the application decomposition is to decompose the application in simple components so each one can be analyzed for his exposure to threats and for specific design flaws and vulnerabilities that these threats might seek to exploit. These basic application components consists on the users of the application, their roles, the data assets in storage and transit, the application use cases and the type of functionality that can be assessed by these users, the security controls to protect data and functions, the entry and the exit points for the data, the trust boundaries to access data and processes and the data flows. Once the application is dissected in the various application components using architecture analysis techniques such as data flow diagrams, it will be possible to identify any potential design flaws and later on analyze the impact that various threats might have by exploiting these design flaws.

Without considering specific threats and the library of threats for each component that will be done during the threat analysis per component in stage IV, by decomposing the application in this stage, it is possible to assert that security requirements for confidentiality, integrity and availability of data and functional assets are enforced by design. Another important aspect of the analysis that is covered during this stage is the analysis of the functionality of the application by breaking it down to the user's functions; the data processed by these functions and the security controls to protect this data as well as to enforce authentication and authorization, secure session management and audit and logging.

Activities: this stage consists on the following activities:

> **ADA 3.1- Decomposition of the application** into basic data and functional components. Decompose the application in basic components of the application architecture by extracting the information from the previously captured technical documentation during the definition of the technical scope. The basic components that the application architecture can be decomposed using data flow diagrams to identify user interfaces, data interfaces and data processes, data storages, data entry points and data exit points and trust boundaries. The decomposition is also done at the functional level by identifying each architectural component by the specific functionality provided by that component.

ADA 3.2 – Security controls assessment. After the design of the application has been decomposed in basic architecture elements including the type of data that these components process as data in transit through other components and as data stored it is possible to analyze the scope for the security control and to assert that security requirements are followed for the design of security controls and application components. If there are any gaps in the design of security controls or design flaws in the implementation, these can be identified in this stage and the exposure of threats can be evaluated during the threat analysis stage. If a design flaw is identified, this should be reported as finding and an initial risk level should be assigned. This risk will be revised later during the threat analysis by analyzing the likelihood of a threat exploiting such design flaw.

ADA 3.4 – Functional analysis. The goal of this activity is to identify security control gaps in the protection of the application functions. This activity is also referred as transactional analysis and in essence consists in the analysis of application functions to assert the presence of security controls that protect the confidentiality, integrity, availability and accountability of these functions. Examples of controls that protect functionality are authentication and authorization, encryption, input and output validation, session management and audit and logging. At functional level each of the application functionality exposed by the application to a user can be analyzed to determine the protection of the functionality by the security controls. If a gap in a security control is identified, a finding should be reported and the design flaw should later considered for exposure to threat to identify the countermeasure that would mitigate the impact of such threat. Note that functionality can be abused because of lack of security control but also bypassed by exploiting specific flaws in the implementation of the application business logic. A possible way to identify business logic flaws is to analyze all uses and abused of the application functionality using use and abuse cases.

Stage inputs: information for validating high level risk objectives:

a. Initial objectives defined in stage I
b. Technical documentation in scope as output from stage II

Stage Outputs: decomposition of application architecture and analysis of the data flows processed by each architectural component and of the security controls that protect each component. Decomposition of the application in basic components of functionality and assertion of the security controls protecting such functionality associated with each component. Analysis of security controls and identification of any control gaps.

Stage IV - Threat Analysis (TA)

Goal: "Definition of the threat landscape and identification of the specific threat agents targeting the application in scope for the analysis. Analysis of internal and external threat agents. Analysis of the threat agent's capabilities motivations and opportunities. Estimation of the threat agent probability to be realized in attacks against the application/product in scope. Mapping of the threats to the assets (functions and data) of each the data and functional components of the application previously analyzed.

Guidance: The objective of the Threat Analysis (TA) stage of PASTA(TM) is to conduct the threat analysis in the context of the application/product in scope. The goal of this threat analysis is analyze the type of cyber-threat agents, human (e.g. script kiddies, hacktivists, cyber-criminals, fraudsters, cyber-spies) and non-human (e.g. malware), that might intentionally or opportunistically target the application in scope. Initially it is important to capture the threat environment that depends on the business and technical environment in which the application operates. This initial analysis is necessary to characterize the threat landscape. Every threat environment can be characterized by specific threat agent factors and events. The categorization for the threat agents might include threats that are either human or automated, their capabilities such as the type of skills, the attack techniques and tools at their disposal, their motivations (political, monetary, espionage), their opportunities such as the type of targets that can be explored and attacked and the application vulnerabilities that can be discovered and exploited. The characterization of the threat landscape largely depends on the information at the disposal of the threat analyst such as threat reports and threat information gathered from different sources of threat intelligence such as the ISACs (information Sharing and Analysis Centers) and sources of security incidents caused by cyber-attacks such as Web Hacking Incident Database (WHID). The purpose of the analysis of threat intelligence is to determine the level of severity of the threats environment and decide the course of action such as issuing alerts and advisories for customers and clients, monitor specific events for detect incoming attacks and be prepared to respond in the case the application will be attacked. From the security incident response perspective, the monitoring of specific cyber threats events might rely upon the logging of specific suspicious events such as abnormal access control violations and triggers (e.g. velocity checks on application functions, web page errors, etc.). For the correlation of these events, the threat analyst might utilize kill-chain techniques (e.g. chain of events) to determine if these events might indicate the potential course of action of a possible attack. From the application threat modeling perspective the analysis of threats and threat events is instrumental to build a repository of threat knowledge and threat libraries that can be used to determine the probability of these threats and correlate them with the security controls and the vulnerabilities that are affected by these as well as the technical impacts that are sought. The assignment of a probability value to each threat of the threat library can be based upon the analysis and assignment of factors of threat probability such as threat agent capabilities (e.g. level of skills required to perform the attack), motive (e.g. level of possible gain/reward) and opportunity (e.g. the level of sophistication of the resources required to conduct the attack). Standards methods for scoring the severity of threats include threat agent factors to determine probability of threats used in the OWASP Risk Rating Methodology (MOSP) Motive, Opportunity, Skill Required, Population Size and threat risk scorings such as (DREAD) Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability. These factors can be assigned score levels and averaged to determine the overall threat agent likelihood factor (OWASP-MOSP) and threat risk severity (MS-DREAD) (Note in the case of MS-DREAD threats are classified as technical threats using STRIDE Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privileges). Once the threat library is update with the level of risk of each threat it would be possible to apply these threats to each of the security controls to determine to whether these controls are present to either reduce or increase the level of risk that is factored as probability of the threat to be realized in an attack as well as the technical and business impact of that threat.

Activities: this stage consists on the following activities:

TA 4.1 - Document threat scenarios based upon sources of cyber-threat intelligence (internal/ external) and categorize these threats by the type of threat agents (e.g. human, automated), skills, group capabilities, motivations, opportunities, type of vulnerabilities exploited, targets and cyber-threat severity reported;

TA 4.2 - Update the threat library with the threats analyzed from real time data feed from source of internal (internal honeypots, recent log data and SIEM events, suspicious activity events of user actions, velocity checks on user's functions, security controls failure events) and external threat intelligence (external honeypots and incident data from WHID, datalossDB, ISACs). Include information about the threat agents the assets (e.g. data and functions) and security controls impacted and the vulnerabilities/gaps that are exploited in the attacks/incidents.

T4 4.3 - Assign probability to each threat of the threat library based upon factors of threat probability used (e.g. skills, motivation, opportunity, group size) (NOTE you can use standard threat risk rating methodologies such as OWASP, DREAD).

TA 4.3 - Map threats to security controls identified by asserting can meant to protect functional and data assets as these were identified in stage III.

Note: the impact of the threats analyzed during this stage can be considered as unmitigated till a further validation of the presence of design flaws and vulnerabilities (Stage V) in these controls as well as possible exploits of these controls by specific attack vectors identified during the attack modeling and simulation stage (stage VI).

Inputs: Information for the analysis of cyber-threats:

a. Information about threats from available internal and external sources of threat information (Threat Intelligence Reports, Advisories, SIRT, SIEM. logs
b. Application components decomposed in stage III that can be targeted by threats

Outputs: the output is a threat analysis report for the application that includes the following information:

a. Threat scenario and general threat statement assessment for the main threats that should be considered based upon cyber-threat advisories (e.g. threat severity level from threat advisory to know how close these threats are to be realized in attacks)
b. Updated threat library that include information of each of the threats that are part of the threat scenarios each characterized by threat agents factors (use normalized threat characterization) based upon the information learned from threat intelligence
c. Enumeration of threats that are part of the threat scenario targeting (as potential attack or as detected threat event) specific application controls. These are the components in scope for the risk analysis (use DTS for data assets and function asset in scope) that have been decomposed in stage III
d. Assignment of risk severity as this is the probability for these specific threats targeting application security controls and application components (using threat agent probability factors of skill level, motivation, opportunity). (NOTE Possible threat

probability assignment methods are OWASP and DREAD (but considered too subjective) to assign risk to each threat)

Tools: the execution of this stage can be facilitated and standardized by relying on formal methods for the threat analysis such as:

a. Threat classification for categorizing threats (e.g. STRIDE, OWASP, MITRE-STIX, Events-Incidents, Agents, Motives, Damage Potentials, Tools, Exploits, Impacts (application-users)
b. Threat and controls frameworks that map threats to assets (application data and functions) and security controls affected by each threat vector (documenting attacks and vulnerabilities used by each threat in a threat library)
c. Threat risk severity calculation formulas (e.g. OWASP, FFIEC, DREAD) to calculate severity for each threat agent-source by considering the probability of the event.

Stage V - Weakness and Vulnerability Analysis (WVA)

Goals: Analysis of previously identified implementation vulnerabilities and an application security controls weaknesses such as design flaws and security control gaps identified during stage III that might expose the application assets, data and function to potential threats. Correlation between threats and vulnerabilities with previously identified threats in stage IV. Risk calculation of the severity of vulnerabilities and weaknesses by re-factoring the exposure of assets to potential threats. Prioritization of security tests for specific type of vulnerabilities and security control weaknesses based upon the re-assessed risk severity.

Guidance: The goal of the vulnerabilities and weaknesses analysis stage is to assess existing application vulnerabilities and weaknesses in the design of security controls, correlate them with the threats previously analyzed and re-assess the vulnerability risks by considering the factors of probability and technical impact. The vulnerability reports from different tests such as application dynamic security tests and static application security tests constitute the knowledge base of vulnerabilities that is considered for this assessment as well as any design flaws that were previously identified during the security control assessments in stage III.

Any previously identified application vulnerability should be stored in a vulnerability management repository that is available for querying based upon the specific assets in scope for the risk analysis identified during stage II and decomposed during stage III. These application and system vulnerability assessments are typically performed in compliance with standards and processes such as security architecture reviews, secure code reviews, manual application security tests and automatic dynamic and static code analysis scans. The assumption of this stage is that application vulnerabilities can be mapped for impact to each of the web application assets that include source code, architectural components and data potentially targeted by the threats previously analyzed. Any security control gaps and weaknesses in security controls (e.g. design flaws) identified during the application architectural analysis in stage III are also correlated to threats during this stage. The goal of this stage is to map threats to vulnerabilities and control weaknesses to ascertain whether:

1) Which security control vulnerabilities expose assets (data and functions) to threats;
2) The risk severity of the vulnerabilities as factor of probability and impact of specific threats

Among the factors that can be used for scoring of risk for vulnerabilities a threat analyst can use the ease of discovery of the vulnerability, the exposure of the vulnerability to specific threats and the fact that events to exploit the vulnerability are logged and audited. The factors that can be assessed at this stage also include technical impact factors such as impact on data confidentiality, integrity and availability in the case these vulnerabilities are exploited by a specific threat. The calculation of the risk severity of vulnerabilities in presence of threats can use standard vulnerability severity risk scoring methods such as CVSS. Finally the vulnerabilities and threat that have the highest probability and technical impact that can be assessed during this stage can be put in scope for additional security tests to validate the likelihood of exploit and determine the technical impact.

Activities: this stage consists on the following activities:

WVA 5.1 – Query existing vulnerabilities of security controls. This activity consists on the query of existing application vulnerabilities affecting security controls that protect assets such as data and functions from threats to these assets. Typically each organization should have a vulnerability management repository that is the repository that stores vulnerability data from different security tests previously executed on the application such as architectural risk analysis, source code analysis and penetration testing. During this activity, the threat analysis will be querying vulnerabilities previously identified for each web application and select vulnerabilities in security controls that expose specific data and functional assets such as authentication data, confidential data and various type of functionality such as user authentication, change of authentication credentials, read access to confidential data, change of confidential data and others.

WVA 5.2 - Map threats to security control vulnerabilities and to design flaws in security controls. This activity consists on mapping threats to vulnerabilities and architectural design flaws (e.g. control gaps/ weak design of security controls). Design flaws that are previously identified during the application decomposition analysis and assertion should also be mapped. This activity will consider each threat of the threat library that is already mapped to a security control identified in stage IV to determine if these controls have vulnerabilities or design flaws that might increase the exposure to each threat.

WVA 5.3 - Calculate risk severity to vulnerabilities. With this activity we will assign the risk severity to vulnerabilities by considering the risk posed by exposure to threats. Factor for risk are probability and impact. For probability factors that can be considered are ease of discovery, easy of exploit, awareness, detection of exploit of vulnerabilities. For impact factors we will have technical impact such as impact on CIA of data because of the vulnerability. The risk calculation should use standard grouping for vulnerabilities such as CWE and standard risk scoring methods such as CVSS.

WVA 5.4 - Prioritize security controls for vulnerability testing. The goal of this activity is to prioritize a new suite of security tests for specific vulnerabilities based upon their re-assessment of their severity in the presence of specific threats. These security tests are no longer blind of threats but consider the type of threat agents and attack tools used by these threat agents. The purpose of these tests is to assert the exposure and the risk as technical impact on the assets. For example this can be a case of testing a known SQL injection vulnerability with specific attack vectors used by the threat agents.

Note: *the goal of these tests is not to identify if these vulnerabilities can be exploited but to determine a more accurate value of risk based upon their known exposure to threats. The test or simulation of vulnerability exploits as well as the test of abuses of functionality will be done with specific tests that simulate these attacks after analyzing the attacks in stage V*

Inputs: Documentation to identify assets with inherent risk potentially affected by vulnerabilities, security control gaps:

a. Documented application security risk profile for the assignment of risk to vulnerabilities from stage I
b. Technical documentation in scope from stage II
c. Application architecture documents detailing users, components, functions, data assets analyzed in stage III

Documentation on previously identified threats targeting assets and vulnerabilities for possible exploitation:

a. Threat analysis profile with threat list for web application assets (data and functions) from stage IV

b. Vulnerability reports from previous vulnerability assessments (ethical hacking, source code analysis, secure code review
c. Vulnerability reports/metrics for web application assets (data and function) from security tests, pen testing, source code analysis, architecture risk analysis
d. Design flaws identified in stage III during the application decomposition and assertion

Tools: Vulnerability risk severity calculation tools such as standard vulnerability risk scoring methods (CVSS, CWE) to score risks of vulnerabilities in presence of threats targeting them

Outputs: Weakness and Vulnerability Analysis including the following Information:

a. List of vulnerabilities by correlation of threats to assets and assets to vulnerabilities analyzed (using a threat tree or threat library)
b. List of control gaps/design flaws exposing data assets/functions to threats previously analyzed
c. Calculation of risk scoring for vulnerabilities (using CVSS) and for control gaps/weaknesses based upon their “threat illumination" (exposure to threats as opportunity and target)
d. Updated vulnerability list with prioritization of vulnerabilities and control weaknesses/gaps by the risk severity in consideration of threat and vulnerability likelihood
e. Updated test cases for testing vulnerabilities to validated the potential impact based upon correlation of vulnerabilities to threats whose probability of both threat agent and vulnerability factors are critical

Stage VI - Attack Modeling & Simulation (AMS)

Goal: The goal of this stage is to analyze how the various threats analyzed in Stage IV can be realized in attacks that will produce a negative impact to the organization. The analysis of the attacks relies on the analysis of the chain of events leading to an observed security incident whose root cases are analyzed “post mortem. This analysis leads to the identification of the attack tools and techniques used by the attackers and the description of the various events that characterize the course of action of the attack to that these attacks can be simulated using security tests. The objective of these security tests is to determine the likelihood of exploits and to identify countermeasures to prevent and detect these attacks. These test cases will factor the specific threat agents, the attacking tools and attack techniques analyzed during stage IV and consider the presence of vulnerabilities and weaknesses that were previously identified in (Stage V). The goal of these test cases is to simulate realistic attack scenarios and determine if exploits are possible and identify countermeasures to prevent and detect them.

Commentary: The scope of the Attack Modeling and Simulation (AMS) stage is to model and simulate the attacks against the application to determine whether these attacks lead to a possible impact. By analyzing the various attack scenarios, it is possible to determine the effectiveness of existing security controls in mitigating these attacks. The modeling of these attacks starts first by considering the threat environment, the various types of threat agents and the attack vectors used by these threat agents. Some attacks can be attributed to specific threat agent’s capabilities and motivations.

When modeling the attacks is important to enumerate both opportunistic and targeted attacks against product/ application and include all known attacks that are part of the attack scenario. A comprehensive attack modeling needs to enumerate all type of possible attacks for specific types of security controls (e.g. authentication) using an attack library.

Some of these attacks are also opportunistically target all possible vulnerabilities and data interfaces including vulnerabilities that should be considered inherent of the technology used by the application. For example, mobile applications have inherent vulnerabilities that might be exploited by specific threat agents using different attack vectors. These attacks might follow a specific chain of events such as will seek to compromise the mobile device first and then bypass authentication controls such as multi factor authentication with (MiTMo) Man in the Mobile Attacks.

A model of attacks also allows the visualization of how the attacker’s main goals can be achieved. This can be done by modeling the sequence of attack events (e.g. course of action), the fulfillment of sub-steps goals by modeling the attack in an attack tree and the consideration of all possible user interactions with the target to identify the possible abuses of functionality (e.g. use and abuse cases and visualization of attack paths).

When modeling these attacks, it is important to be able to identify the various vulnerabilities and design flaws of security controls that are exploited by the attacks as well as the business processes/functions that can be by-passed and abused in the pursuit of the attacker's goals. The next step in the attack modeling is to analyze how these attacks lead to an exploit of vulnerabilities and weakness in security controls, determine the possible attack paths, the attacker’s course of actions and attack events that have higher probability to succeed in achieving the attacker’s goal. Since an attack describes how a threat can be realized, the probability of a threat to be realized in an attack depends on several factors and among them the costs for an attacker to conduct the attack based upon the attacking tools at his disposal and the opportunities to exploit known vulnerabilities. To assess the probability of a threat to be realized in an attack to cause an impact, it is important to assign the values to the costs of the attacks and the benefit for the attacker as gain and choose to simulate the attacks that maximize the gain and minimize the costs for the attackers.

A possible way to conduct this type of analysis is by modeling the attacks using attack trees. Attack trees allow the assignment of a probability to each node of the attack tree to decide whether a threat has a probability to succeed and move to the next step of the attack till the attack produces the desired gain for the attacker. The realization of an attack depends on conditions that can be assigned to each node of the attack tree as either “OR” or “AND” conditions. For example a threat agent seeking to compromise the data in a database for the attack scenario of a banking Trojan, need to own the client “AND” attack the application “OR” attack the application directly. Given that the probability of attacking the client is lower than attacking the application directly it will choose that attack path. The probability of success is given by the consideration of all single probabilities of each node whose conditions need to be fulfilled in order for the attacker to reach his goal. Once the most probable attack paths have been analyzed including the application components, processes and security controls that a threat agent need to transverse on its way to the target, it is possible to create specific test cases for simulating the attacks.

Once each of the attacks is simulated with tests it is possible to determine for each threat agent the probability to achieve his desired impact such as exfiltration of data, stealing money and fraud. The analysis of impacts include both technical impacts such as loss of data confidentiality, integrity, accountability, availability as well as business impact as monetized loss associated with a data asset. The determination of the probability of threats to be realized in attacks and the likelihood of the attack to produce either a technical or business impact allows determining the risk of each attack. The identification of the various events of an attack leading to an exploit of vulnerabilities to cause an impact is critical to determine the detective controls that can be deployed to detect these events as indication of an attack and to decide the appropriate response to mitigate the impact.

Once the attack has been analyzed and simulated, it is important to update the attack library with the mapping of threats, controls and vulnerabilities that these attacks seek to exploit. This attack library is inclusive of all known attack vectors and these can be used for simulation of the attacks for testing security controls and countermeasures.

Activities: this stage consists on the following activities:

AMS 6.1 - Model the attack scenarios. An attack scenario can be created for each threat agent based upon the information captured from various sources such as threat intelligence, logged security events incidents and the analysis post mortem of security incidents. Threat agents can be classified based upon their motives and capabilities. Examples include hacktivists, script-kiddies, cyber-criminals, fraudsters and government/state and corporate-sponsored spies. A threat agent might rely on different type of manual attack techniques and automated attacking tool. The attacks against the targets can follow a chain of events such as the attack course of action. The analysis of the sequence of events of a cyber-attack helps to model the attack scenario. The model of the attack scenario also include the attacking vectors used and the specific vulnerabilities that are exploited by the threat agents to realize their goals such as data compromise, data theft, online fraud, abuse of functionality, denial of service and abuse of privileges as example. The outcome of this activity is to create attack stories that describe how the various threat agents might attack the application.

AMS 6.2 - Update the attack library. After the attack scenario has been analyzed and modeled it would be possible that some of the attack vectors attributed to the threat agent's attack techniques and tools used are not part of the threat library and therefore need to be updated including the type of countermeasures that have been found effective in either detecting or preventing the attacks. The new countermeasures that should be considered in the threat library might include new emerging threat attack vectors analyzed in the attack modeling exercise as well as updated vulnerabilities and weaknesses and gaps in preventive and detective security controls (NOTE it is important to revisit the control assessment risk framework to add new preventive and detective controls as options to mitigate the attacks)

AMS 6.3 - Identify the attack surface and enumerate the attack vectors against the data entry points of the application. Use the up to date threat library to identify the type of attack vectors that can be used, the type of vulnerabilities of the architecture components that can be exploited and the security controls that should be in place to detect and prevent the attacks.

AMS 6.4 - Assess the probability and impact of each attack scenario. Determine the probability of each attack of the threat scenario using attack trees (probability of exploit producing the desired technical and business impact that maximize gain for the attacker). Identify the various security controls that can be bypassed and functionality that can be abused leading to the exploit and impact. Each attack scenario can be associated with the probability of the attack and the impact so can be prioritized for the analysis and mitigation.

AMS 6.5 - Derive a set of cases to test existing countermeasures. These test cases are both threat driven using threat classification (e.g. STRIDE) and threats, vulnerabilities identified in each architectural components. Attack driven test cases can be documented based upon the previous analysis of the attack chain of events and the attack vectors used by the attackers. Additional tests can be created using use and abuse cases. Each test case can be prioritized based upon the risk that was previously calculated for each of the attack scenarios using attack trees.

AMS 6.6 - Conduct attack driven security tests and simulations. The scope of these tests is to identify vulnerabilities and design flaws that are exploited by the attack and determine the risks as factor of probability and the impact. The risk ought to factor the technical impact (loss of confidentiality, integrity, accountability, availability) of these exploits using the results of the simulated attack exercise. After each test case is executed, the risk values that were previously assigned to each threat should be revised higher or lower depending on the results of the tests.

Inputs: Documentation on previously identified threats targeting assets and vulnerabilities whose attacks are to be modeled and simulated/tested:

a) Threat analysis artifacts from Stage IV (threat profile and threat model) to identify the specific threats should be modeled in the attack scenarios;
b) The design flaws and vulnerabilities whose attacks might exploit
c) The application data and functional assets that are likely (e.g. whose threat likelihood is high) targeted by threat agents;
d) The application data and functional assets that are exposed to threats because of vulnerabilities whose attacks might exploit;

Tools: threat modeling tools and formal methods for the analysis of attacks and simulation of exploits such as attack trees, use and abuse cases and attack libraries.

Outputs: the outcome of this stage includes the following:

a) The model of the attack scenarios including the course of action of the attacks that is followed by the different threat agents;
b) Attack tree with the determination of the most probable attack paths leading to exploits and identification of various vulnerabilities and security controls that might be bypassed;
c) Use and abuse cases covering bypass of security controls and abuse of functionality for fraud and data compromise;
d) The attack surface that is targeted by threat agents previously identified in the attack modeling scenarios (analyzed for threats and data entry points previously identified stage IV Threat Analysis);
e) Updated attack library with new attacks that can be enumerated for simulating/testing the various attack scenarios;
f) Documented test cases that can be used to test the resilience of the application in presence of specific conditions of exposure of vulnerabilities as well as by considering the real attack scenarios, automated and not, single or group based, capabilities, motivations, cyber-attack tools and techniques used in real attack case scenarios;
g) Attack testing/simulation report that includes the results of the test cases for testing the various attack scenarios for vulnerabilities and design flaws. The report includes the assessment of risks of each security issue (design flaws, vulnerabilities) that

is identified such as probability and technical impact can be determined based upon the results of the test/simulation.

STAGE VII - Risk Analysis & Management (RAM)

Goal: The goal of this stage is to analyze the risk of each attack scenario that was previously simulated and tested and identify both the technical and the business impacts. After risks have been analyzed they need to be managed to reduce the impact to acceptable levels by following a risk management strategy that is alignment with the business objectives and the risk mitigation objectives defined in stage I.

Commentary: The objective of the risk analysis & management (RAM) stage is to analyze the various risks of the threats, the attacks that are used to realize these threats and the vulnerabilities that can be exploited and produce an impact on the application assets that include both data and functionality. The analysis of risk includes the calculation of the risk levels of the analysis of the probability and factors of technical impact such as loss of confidentiality, integrity, availability, accountability as well as business impact such as the various monetary legal impacts to the business that include business continuity loss, fraud monetary loss, non-compliance violations, data privacy law violation impacts.

For the assessment of technical impact factors and business factors it is possible to use technical risk and business risk calculation formulas used by each business. Examples of generic risk assessments for technical risks include the standard scoring methods for vulnerabilities such as First CVSS and the categorization of vulnerabilities such as Mitre's CVE. The analysis of technical and business impacts can consider the specific factors for assigning the probability and impact of threats. Once the risks are assigned to each one of the attack scenarios and vulnerabilities identified in the attack simulation analysis, the next step is to recommend the adoption of new countermeasures to reduce the impacts of threats and recommendations for fixing any design flaws and vulnerabilities that were previously identified during the attack simulation tests.

Recommendations for the adoption of countermeasures need to follow a risk mitigation strategy and a risk management process. The risk management process might dictate whether a risk should be mitigated by implementing countermeasures of the risk can be either transferred to a third party with cyber-insurance or accepted when compensating controls could be identified. Any recommendations of new countermeasure need to consider both the costs and the benefits such as the effectiveness of these countermeasures in reducing the various impacts of threats targeting the application. The cost of countermeasures can be calculated based on their total cost of ownership that includes the cost of acquiring, deploying and maintaining such countermeasure. The costs of countermeasures need to commensurate with the potential impacts. The selection of countermeasures should also factor the residual risks after these countermeasures are deployed to determine their effectiveness as a whole. After the countermeasures are identified, the decision of whether to implement these countermeasures need to align with the organization risk mitigation strategies and the objectives for the treatment of risk defined in STAGE I.

Activities: this stage consists on the following activities:

RAM 7.1 – Calculate the risk of each threat. The goal of this activity is to analyze the technical impact caused by the attacks that were previously analyzed and factor the probability to determine the risk of each threat to be realized in an attack. Examples of technical impacts include the loss of confidentiality, integrity, availability and accountability of data and functionality. By associating the probability of each threat occurring based upon the capabilities and motivation of each threat agent, it is possible to assign the risk to each threat attack. Once the technical impact have been calculated it is also possible to associate the monetary value of the asset that is at risk to be compromised and quantify the business impact caused the various types of threats and exploit;

RAM 7.2 – Identify the countermeasures. The goal of this activity is to recommend security measures to detect and protect from the various attack scenarios and to remediate security issues whose root causes are found to be either in the design flaws or implementation type of vulnerabilities; The types of countermeasures includes preventive and detective security controls that are found effective in mitigating the impact of the attacks that were previously simulated. These countermeasures can be deployed in addition to fixing of any design flaws and vulnerabilities identified in the test.

RAM 7.3 – Calculate the residual risks. The goal of this activity is to calculate the levels of residual risk both qualitatively and quantitatively after the different risk mitigation options (countermeasures) are considered for deployment;

RAM 7.4 – Recommend strategy to manage risks The goal of this activity is to recommend the actions that should be taken to treat the risk to reduce the business impacts to the organization: these include accepting the risk when is deemed acceptable by the organization based upon the risk level and the presence of compensating controls/measures, apply the countermeasures to reduce the risk to acceptable levels, transfer the risk to a third party such as by signing cyber-insurance and avoid the risk such as in the case of deciding of not implementing the application feature or storing valued assets that might put the application at risk.

Inputs: the various activities of take inputs from the following activities:

a) Threat probabilities;
b) Severity of the vulnerabilities exploited by each threat;
c) Exposure of vulnerabilities to each threats;
d) Attack simulation and tests results;
e) Countermeasures that mitigate the vulnerabilities and design flaws identified in the attack simulation;
f) Cost and effectiveness of the countermeasures

Tools: risk calculators (quantitative and qualitative). Methods to calculate the residual risks after countermeasures are applied. Risk frameworks/templates that map threats to countermeasures Note: Quantitative risk analysis need to consider the asset value as input that was preliminary evaluated as well as the risks management objectives. Control frameworks and threat tree analysis can be used for identification of countermeasures.

Outputs: the outcome of this stage includes the following:

a. Analysis of risks each threats including technical and business impacts;
b. List of countermeasures and recommended risk mitigation options for reduce technical and business impacts (security measures present and non) such as to reduce probability, security and business impact, liability, noncompliance risks;
c. Analysis of the residual risk after recommended risk mitigation options are selected/considered based upon their risk mitigation effectiveness and their costs;
d. Recommended risk mitigation strategy for threats targeting the application/product in scope. This strategy includes both technical and business recommendations for reducing the risks to acceptable levels in alignment with risk objectives stated in stage I.