Process for Attack Simulation & Threat Analysis

VERSPRITE

OVERVIEW

The purpose of PASTA (Process for Attack Simulation and Threat Analysis) is to provide a process for simulating attacks to applications, analyzing cyber threats that originate them and mitigate cybercrime risks that these attacks and threats pose to organizations. PASTA consists of a seven stage process for simulating attacks and analyzing threats to an application environment with the objective of minimizing risk and associated impact to the business. By following this process, businesses can determine the adequate level of countermeasures that can be deployed to mitigate the risk from cyberthreats and attacks to applications.

Solvency

The Process for Attack Simulation and Threat Analysis (PASTA) provides businesses a strategic process for mitigating cybercrime risks by looking first and foremost at cyber threat mitigation as a business problem. The process provides the tactical steps that can be followed to provide effective countermeasures for mitigating existing vulnerabilities by analyzing the attacks that can exploit these vulnerabilities and mapping these attacks to threat scenarios that specifically focus on the application as a business-asset target.

Comparative Analysis

Several formal methods and tools have been developed today for modeling threats (threat modeling methodologies) and for analyzing risks to information assets that these threats pose. These formal methods, known as threat modeling processes, and tools are either software centric, whose purpose is model threats to application software, or data asset centric, whose purpose is to analyze the risk that these threats pose to the data assets. These formal methods model threats independently of the business impact and business context. Moreover, threat modeling methods available today do not model the threat agent motives and their drivers for attacking businesses. Software centric threat modeling

methodologies are limited to determination of technical impact (e.g. security control compromise) of the threat. The inability to determine the business impact various threats present (e.g. revenue lost because of security control-data compromise-loss) is addressed by the PASTA methodology. The new process for attack simulation and threat analysis specifically takes into consideration business context to determine the business impact of the threats, and maps the technical risks to the business risks. Only threats specifically impacting the organization application environment are considered, allowing an in-depth and focused analysis of the attacks and the attack scenarios that are simulated to determine the possible exploits and devise countermeasures to eliminate them.

Audience

PASTA provides different benefits to the stakeholders involved in assessing threats to the application environment, design secure applications and decide how to mitigate risks by applying a risk mitigation strategies. Examples include: Architects, Developers, Security Testers, Project managers, Business managers, Information Risk Officers.

Benefits

PASTA allows architects to understand how vulnerabilities to the application affect threat mitigation, identify trust boundaries and classification of the data assets, and identify vulnerabilities and apply countermeasures via proper design. PASTA helps developers understand which components of the application are vulnerable and then learn how to mitigate vulnerabilities. Security testers can use security requirements derived through the methodology as well to create positive and negative test cases. Project managers can prioritize remediation of security defects according to risks. Business managers can determine which business objectives have impact on security while information risk officers can make strategic risk management decisions by mitigating technical risks while considering costs of countermeasures versus costs associated with business impact as risk mitigation factors.

DETAILED DESCRIPTION

The stages of the PASTA methodology are described herein:

Stage I – Define the Objectives

Identify business objectives and ensure an appropriate level of security requirements to support the business goals for the application yet meeting compliance with security standards. Identify preliminary security and compliance risks and their business impacts to the application.

Stage II

Define the Technical Scope

Define the technical scope/boundaries of threat modeling as they depend on the various technologies, software and hardware, components and services used by the application. Categorize any architectural and technologies/components whose function is to provide security controls (e.g. authentication, encryption) and security features (e.g. protection of CIA).

Stage III

Decompose the Application

Decompose the application into essential elements of the application architecture (e.g. users, servers, data assets) that can be further analyzed for attack simulation and threat analysis from both the attacker and the defender perspective.

Stage IV

Analyze the Threats

Enumerate the possible threats targeting the application as an asset. Identify the most probable attack scenarios based upon threat agent models, security event monitoring, fraud mapping and threat intelligence reports. The final goal is to analyze the threat and attack scenarios that are most probable and need to be prioritized later for attack simulation.

Stage V

Vulnerabilities & Weaknesses Analysis

The main goal of this stage of the methodology is to map vulnerabilities identified for different assets that include the application as well as the application infrastructure to the threats and the attack scenarios previously identified in the previous threat analysis stage. Formal methods that map threats to several generic types of vulnerabilities such as threat trees will be used to identify which ones can be used for attacking the application assets. Once these vulnerabilities are identified, they will be enumerated and scored using standard vulnerability enumeration (CVE, CWE) and scoring methods (CVSS, CWSS).

Stage VI

Analyze the Attacks

The goal of this stage is to analyze how the application and the application context including the users-agents, the application and the application environment, can be attacked by exploiting vulnerabilities and using different attack libraries and attack vectors. Formal methods for the attack analysis used at this stage include attack surface analysis, attack trees and attack libraries-patterns. The ultimate outcome of this stage is to map attacks to vulnerabilities and document how these vulnerabilities can be exploited by different attack vectors.

DESCRIPTION OF OVERALL STRUCTURE

PASTA is a framework for modeling threats to the application environment that begins with a phase for understanding key business objectives to be supported by the application threat modeling process and concludes with a risk mitigation phase that provides the opportunity to mitigate any business risk issues that have been identified and qualified as part of the threat modeling effort. The above stages provide a fundamental framework for an iterative methodology for threat modeling.

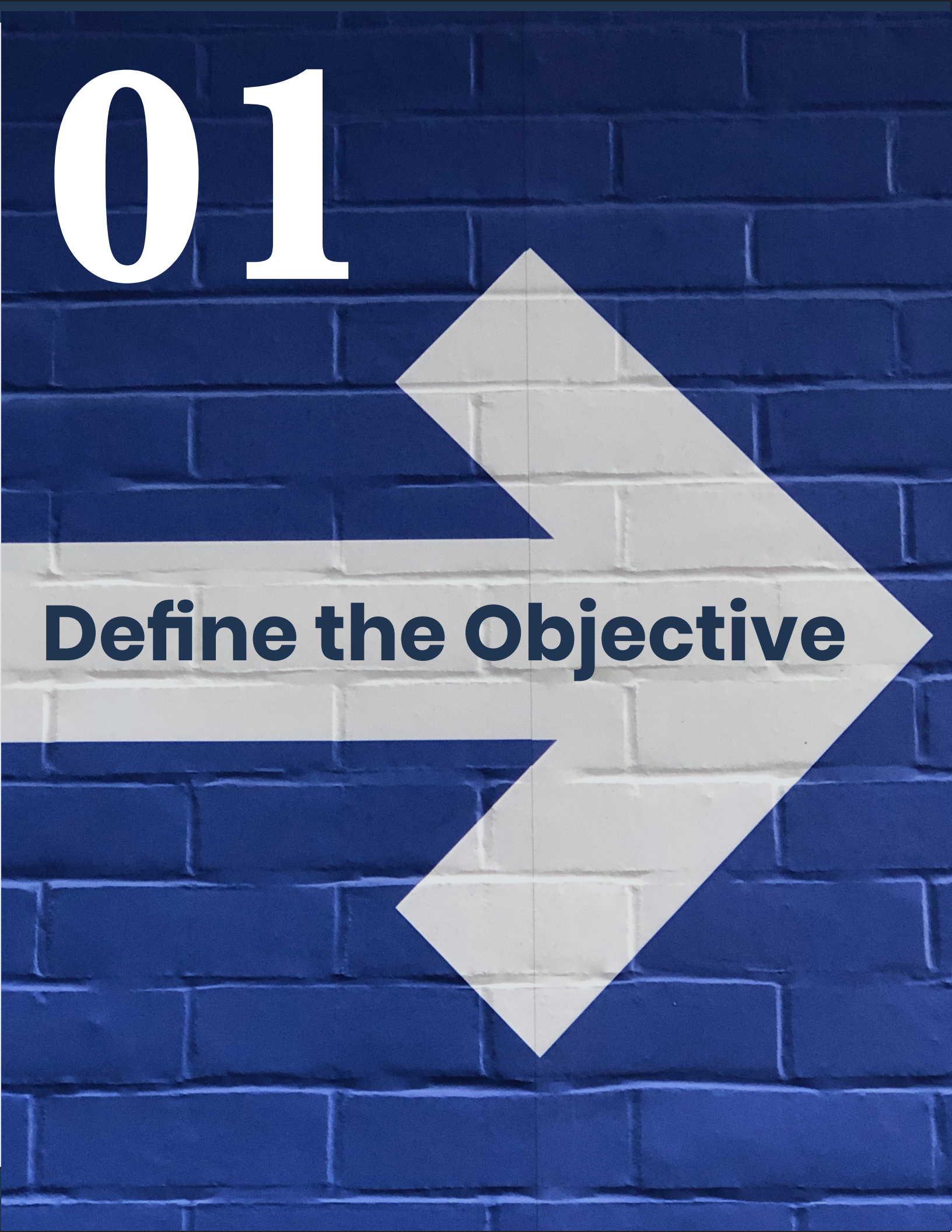
This iterative process can be applied to an application that is preferably under development within the boundaries of a relatively mature or quickly maturing SDLC lifecycle. Each of these seven phases are critical to the overall success of fulfilling the objectives of application threat modeling, whether they are related to risk mitigation, threat identification, or improved application design and software behavior (essentially regardless of threat modeling approach).

Stage By Stage Walkthrough

The following is a synopsis of the PASTA methodology across all of its seven stages. The following information recaps the overall goals, steps, participants for each stage with reference and in appreciation to the unique McKesson environment. Please reference the diagrams that start on page 10 in order to match with the numeric references mentioned within each recap.

In terms of roles, a RACI (Responsible-Accountable-Consulted-Informed) model is provided below to denote McKesson groups and generic roles that could be leveraged as part of each stage's activities. The generic roles found to be somewhat relevant across PLM groups as well as Corporate functions, and in some cases, inclusive of 3rd parties, where applicable. The following list includes symbols and definitions, which are appropriately referenced in each of the subsequent stages of this PASTA abstract.

MGT	Product Management	SWE	Software Engineer	RL	IT Risk Leader
PMO	Project Management	QA	Quality Assurance	PC	Product Compliance
BA	Business Analyst	SYS	SysAdmin	SA	Software Assurance
ARC	Architect	SOC	Security Operations	EA	Enterprise Architect
CTO	Administration	VA	Vuln Assessor	PT	Pen Tester


01
Define the Objective

Stage I

The first stage, **Define the Objectives (1.0)**, is central for fulfilling security, risk, and improved software security by the business boundaries that it creates.

After a generic mission is established, the next step is to **Define the Business Objectives (1.1)** to which the application needs to adhere. These business objectives are fulfilled by a description, at level of pieces of application functionality that constitute a description of preliminary functional-business requirements for the application. This stage is critical for defining the business scope boundaries of threat modeling that relies on the description of the essential business objectives of the application.

Upon defining a set of business objectives, it will be possible to **Define Security Requirements (1.2)** to sustain the business goals of the application. The security requirements that can be defined at this stage may include requirements for the security of the application, for protecting the data-assets, and for the security processes-assessments as well as for the security of technology used by the application.

Since application security requirements also need to address compliance with information security standards, policies and regulations, they also need to **Define Compliance Requirements (1.3)**.

At stage I, it also possible to identify the business critical components, services and data-assets whose loss or compromise can cause a tangible business impact to the business. This process is also referred as **Preliminary Business Impact Analysis (1.4)**. At this stage, BIA helps to identify critical, important and non-critical systems, components and services and estimate the financial, revenue and non-revenue impacts associated with each. As part of the BIA, a preliminary list of threats is also considered and the likelihood that these might occur is estimated to assess the potential impact and assess the need for countermeasures needed as part of the application design to deal with the identified threat.

Roles:

BU/Product Groups						Corporate Functions							3rd Party	
MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
A	R	R/A	I	I	I	I	-	-	C	-	A	R	-	-

02

Define the Technical Scope

Stage II

The second stage, **Define the Technical Scope (2.0)** is critical to define the scope of threat modeling – it essentially answers what should be in scope for the attack simulation and the threat analysis.

The technical scope first requires the analyst to **Identify Application Boundaries (2.1)** and the dependencies of the various technologies used. In order to define the technical scope, it is necessary to capture the architecture of the application at high level and to **Identify the Application Dependencies From Network Environment (2.2)** as well to **Identify Application Dependencies from Servers and Infrastructure (2.3**) including hosts and servers.

It is also important to **Identify the Application Dependencies from Software (2.4)** that includes the software aspects of the application: services, components, frameworks and programming languages as a dependency. The technical scope for application threat model can be captured using architecture security review questionnaires. Architectural reviews can be conducted holistically to cover all technical aspects of the application that can be relevant for the threat modeling exercise: the objective

is to develop a comprehensive view of the various layers of the application architecture and the technical environment that includes everything from the network, host platforms, third party sites, system/network services, and application interfaces.

The technical scope needs to characterize the architectural elements of the application from attacks, threats and risk mitigation perspective; for example, the various services, software and hardware components-technologies used need to be categorized in terms of exposure to attacks, security impact

(e.g. CIA) and threat mitigation (e.g. countermeasures).

At this stage, it is important to look at the previously identified security requirements and business objectives. For example, the application architecture needs to cover all basic functionality and be representative of the application components that can be potentially exposed to threats and attacks.

At this stage, identifying the technology dependencies helps determine potential exploits of technical vulnerabilities by the attack vectors identified as part of the attack analysis. Identifying third party hosted servers is critical in the determination of exposure to risks.

Roles:

BU/Product Groups						Corporate Functions							3rd Party	
MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
I	A	C	A	R/A	C	R/A	-	-	-	-	C/I	-	-	-

03

Decompose the Application

Stage III

The third stage consists of **Decomposing the Application (3.0)**. Decomposition is an essential step in every threat modeling methodology and provides the threat analyst with an understanding of the application from both the attacker and the defender perspectives. From the attacker perspective, this stage of the methodology allows the threat modeler to dissect an application into parts in order to identify the targets such as the assets, the application entry points and the communication channels.

The application decomposition is supported by documentation of **Data Flow Diagramming and Trust Boundaries (3.1)**. The DFD exercise provides a formal representation of the data flows of the application by visualizing the application tiered architecture including user interfaces, servers-processes and data-assets. By visualizing the application in a DFD, it is possible to identify the trust levels-boundaries required to access application assets such as servers-processes and data and determine their exposure to potential threats.

At this stage, all basic elements of the application architecture

relevant to the threat analysis are listed, also requiring the threat modeler to **Identify Users-actors and their Roles- permissions (3.2)** and to **Identify the Assets, Data, Services, Hardware and Software (3.3.)** and to **Identify the Data Entry Points and Trust Levels (3.4)** that consist of the application interfaces and the trust levels required to access these interfaces.

Roles:

BU/Product Groups						Corporate Functions							3rd Party	
MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
I	I	I	A	A	I	A	-	-	-	-	A	-	-	-

04

Analyze the Threats

Stage IV

The fourth stage, **Analyze the Threats (4.0)**, is an essential step of the methodology whose purpose is to identify the threats that affect the application and the application environment and to map these threats to the most probable attack scenarios. Threat frameworks that map threat agents to motives and targets can be used to **Analyze Probabilistic Attack Scenarios (4.1)** and to characterize the threat landscape and the likely attack scenarios.

The Analysis of Incidents and Fraud-Case management reports (4.2), such as reported security incidents resulting in data breaches and fraud, and the **Analysis of the Application Logs and Security Events (4.3)**, the occurrence of monitored security events, can be correlated to threats to determine which ones are most likely to affect the application and its environment. Another important factor for the threat analysis is to **Correlate Incident and Fraud with Threat Intelligence (4.4)** to refine the threat analysis of most probable threat scenarios.

Roles:

BU/Product Groups						Corporate Functions							3rd Party	
MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
-	-	-	I	I	I	I	-	-	-	-	R/A	-	-	-

05

Vulnerabilities & Weaknesses Analysis

Stage V

The fifth stage, **Vulnerabilities and Weaknesses Analysis (5.0)**, objective is to map threats to vulnerabilities by leveraging existing application vulnerability assessments from different sources that might include secure architecture design reviews, source code analysis/reviews and manual pen test- application vulnerability scanning. The assumption is that these vulnerability assessments all feed into a common repository- database for vulnerability management, analysis and reporting. To conduct this stage, it is also necessary that all reported vulnerabilities are categorized using a standard- common categorization for listing vulnerability types (e.g. CVE).

The first step of the vulnerabilities and weakness analysis is to **Correlate Vulnerabilities to the Application Assets (5.1)** such as data, data repositories and data sources, hardware and software as previously identified in step (3.3) of stage III, and decompose the application.

The next step of this stage is to **Map Threats to Vulnerabilities using Threat Trees (5.2)**. Threat trees analyze the threat affecting the asset by looking at different vulnerabilities that can be used to

exploit the threat. A known vulnerability can be mapped to a node of the threat tree to determine the likelihood of the threat to be realized. Another way to map threats to vulnerabilities that can be exploited is through use and abuse cases.

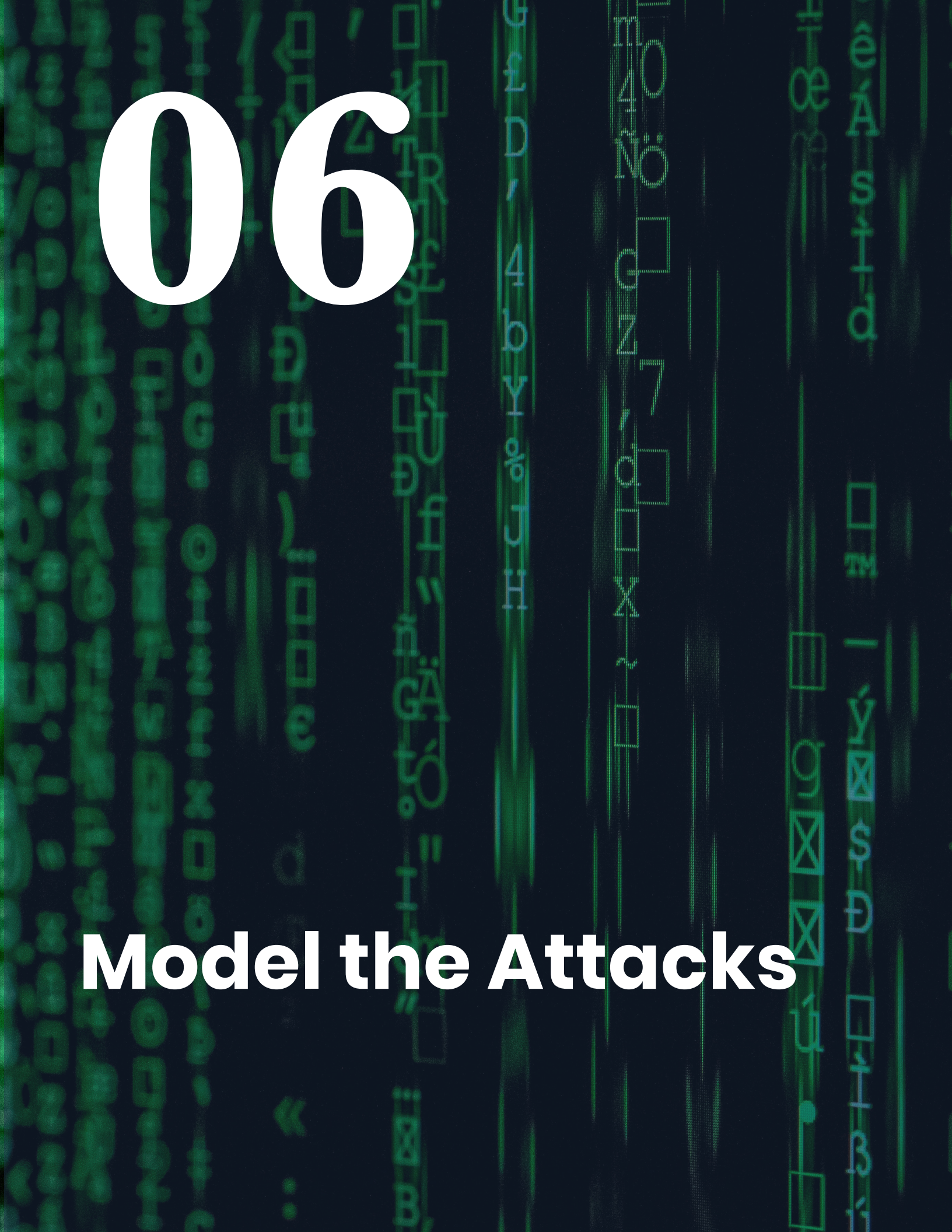
The step **Map Threats to Security Flaws Using Use and Abuse Cases (5.3)** is a methodology that can be used to determine how the application and the application transactions can be abused by an attacker to attack the data and the transactions by exploiting weaknesses in security controls due to design flaws. Weaknesses in security controls can be identified when the threat posed by the attacker is not mitigated by a security control. This

step is also particularly useful to identify security business logic flaws that can potentially be exploited by attacks directed at the business logic of the application.

After the vulnerabilities are mapped to assets and to the threats that could exploit them using threat trees and use and abuse cases, the next step is to **Enumerate and Score Vulnerabilities (5.4)**. The goal of this step is to categorize the vulnerabilities in light of their exposure to threats and the impact to assets using standard methods such as CWE (Common Weakness Enumeration) and score them using CVSS/CWSS methods.

Roles:

BU/Product Groups						Corporate Functions							3rd Party	
MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
-	-	-	I	I	C	I	-	-	-	-	R/A	-	-	-



06

Model the Attacks

Stage VI

The goal of stage six is **Attack Modeling (6.0)**. The first step of modeling the attacks to the application is to **Identify the Application Attack Surface (6.1)**. The attack surface identifies at high level how application assets and functions can be attacked via the different attack points, such as data entry points, hosts and systems-servers that are exposed to the attacker, data flows and communication channels. Identifying the type of interfaces and data entry points, for example, is critical for determining the type of attack vectors and attack libraries that can be used by an attacker.

Once the opportunities for attacking the application have been identified, the next step is to analyze, within these opportunities, how an attacker could attack the application assets from the perspective of the attacker goals the different means-techniques to pursue these goals as well as the attacker's options (OR nodes) and alternatives (AND nodes).

This attack analysis is supported using a methodology referred as attack trees and covered by the step **Derive Attack Trees For Threats and Assets (6.2)**. To complete these step, several attack

trees would need to be developed to cover the most likely critical threat agent-attack scenarios previously identified in step 4.1.

After the most likely attack scenarios have been fully represented with attack trees, the next step of this analysis is to enumerate a list of attack vectors that can be used by an attacker to pursue the attacker's goals. The list of applicable attack vectors to exploit potential vulnerabilities can be extracted from existing attack libraries.

The step **Map Attacks Vectors to Nodes of the Attack Trees (6.3)** is the essential step in determining the attacker's opportunities in terms of exploits of the potential vulnerabilities identified at each node of the attack tree.

The **Identification of Exploits and Attack Paths using Attack Trees (6.4)** is the ultimate outcome of modeling the attacks since it allows the threat analyst to identify the attack paths that lead to vulnerability exploit and impact. This information can later be used by a pen tester to conduct tests to exploit application vulnerabilities using different attack vectors.

A risk analyst can also use the attack modeling information pro-

vided at this stage to analyze the risk posed by different attack scenarios and determine the likelihood and impact of exploiting existing vulnerabilities.

Roles:

BU/Product Groups						Corporate Functions							3rd Party	
MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
-	-	-	-	-	C	-	-	-	-	-	R/A	-	I	R/A

07

Risk & Impact Analysis

Stage VII

The stage VII, the final stage, of the methodology consists of **Residual Risk & Impact Analysis (7.0)**. A prerequisite to conduct this stage is completing the previous stages (I to VI) of the methodology.

The goal of this stage is to **Quantify and Qualify Business Impacts (7.1)**, **Identify Gaps in Security Controls (7.2)**, **Calculate Residual Risk (7.3)** and **Identify Risk Mitigation Strategies (7.4)**. The business impact that was preliminary analyzed in stage I at step 1.4 can be revised to take into consideration the results of the analysis conducted in the previous stages of the methodology.

In particular, this includes the technical environment in scope for the assessment defined at stage II, the assets impacted identified at a level of granularity of the components of the application architecture and the application environment identified at stage III, the most probable threats to the assets based upon models of threat agents and other threat factors defined at stage IV, the vulnerabilities analysis that map threats to vulnerabilities and the impacted assets at stage V and the pattern of attacks that can exploit these vulnerabilities to cause impact at stage VI.

By factoring all this information, it is possible to assess qualitatively (e.g. likelihood times impact) and quantitatively (e.g. ALE) the impact derived by the realization of the threats through the simulated attacks, analyze the impact of the exploit of identified vulnerabilities, identify any gaps in countermeasures and calculate the residual risk to the business after new countermeasures are applied or existing mitigating controls are also considered.

With all this information in hand, it is finally possible to make informed risk management decisions based upon different risk mitigation strategies.

Roles:

BU/Product Groups						Corporate Functions							3rd Party	
MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
C	C	C	C	C	I	C	I	C	C	C	R/A	-	I	I

Uniqueness of PASTA

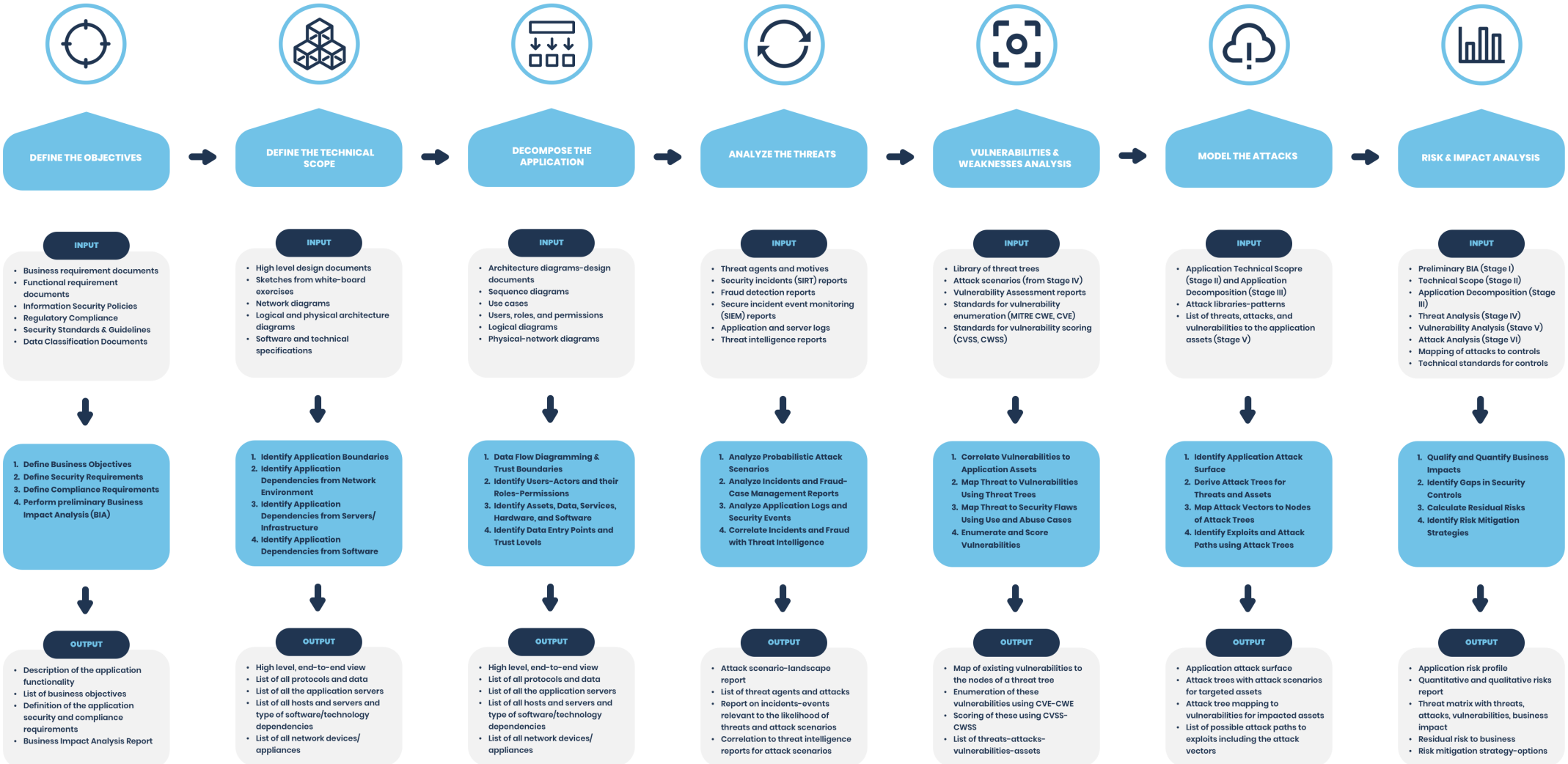
Compared with other threat modeling and risk assessment methodologies-processes, PASTA is unique because it is indifferent to the approach (security centric, risk based, asset centric, software centric) yet embodies all the basic elements of threat analysis both from the attacker and the defender perspectives.

PASTA's uniqueness stands on the holistic view of threats and attacks to the application environment by including first and foremost the business aspects of mitigating risks posed by cyber threats to a level that is manageable by the business.

PASTA fills the current gap between technical and business risk analysis disciplines when addressing cyber threats. PASTA provides a process that corporations can follow to address security issues from the inception of the software development lifecycle. PASTA allows corporations to evolve vulnerability assessments to threats and attack analysis as the drivers for determining the risk mitigation strategy.

Making PASTA

PASTA provides the framework for integrating existing information security, security engineering and risk management disciplines. Execution of PASTA can be driven by corporate information security strategy and incorporated as enterprise- wide initiative aimed to improve the organization efficiency in dealing with security defect management, vulnerability management, information security and risk control and management. Some of the stages of the process can be automated with the use of threat modeling tools; while critical to the execution of the process is the education and training of PASTA for the different stakeholders of the methodology.


DEFINE THE OBJECTIVES
INPUT
Business requirement documents
Functional requirement documents
Information Security Policies
Regulatory Compliance
Security Standards & Guidelines
Data Classification Documents
1. Define Business Objectives
2. Define Security Requirements
3. Define Compliance Requirements
4. Perform preliminary Business Impact Analysis (BIA)
OUTPUT
Description of the application functionality
List of business objectives
Definition of the application security and compliance requirements
Business Impact Analysis Report
DEFINE THE TECHNICAL SCOPE
INPUT
High level design documents
Sketches from white-board exercises
Network diagrams
Logical and physical architecture diagrams
Software and technical specifications
1. Identify Application Boundaries
2. Identify Application Dependencies from Network Environment
3. Identify Application Dependencies from Servers/ Infrastructure
4. Identify Application Dependencies from Software
OUTPUT
High level, end-to-end view
List of all protocols and data
List of all the application servers
List of all hosts and servers and type of software/technology dependencies
List of all network devices/ appliances
DECOMPOSE THE APPLICATION
INPUT
Architecture diagrams-design documents
Sequence diagrams
Use cases
Users, roles, and permissions
Logical diagrams
Physical-network diagrams
1. Data Flow Diagramming & Trust Boundaries
2. Identify Users-Actors and their Roles-Permissions
3. Identify Assets, Data, Services, Hardware, and Software
4. Identify Data Entry Points and Trust Levels
OUTPUT
High level, end-to-end view
List of all protocols and data
List of all the application servers
List of all hosts and servers and type of software/technology dependencies
List of all network devices/ appliances
ANALYZE THE THREATS
INPUT
Threat agents and motives
Security incidents (SIRT) reports
Fraud detection reports
Secure incident event monitoring (SIEM) reports
Application and server logs
Threat intelligence reports
1. Analyze Probabilistic Attack Scenarios
2. Analyze Incidents and Fraud-Case Management Reports
3. Analyze Application Logs and Security Events
4. Correlate Incidents and Fraud with Threat Intelligence
OUTPUT
Attack scenario-landscape report
List of threat agents and attacks
Report on incidents-events relevant to the likelihood of threats and attack scenarios
Correlation to threat intelligence reports for attack scenarios
VULNERABILITIES & WEAKNESSES ANALYSIS
INPUT
Library of threat trees
Attack scenarios (from Stage IV)
Vulnerability Assessment reports
Standards for vulnerability enumeration (MITRE CWE, CVE)
Standards for vulnerability scoring (CVSS, CWSS)
1. Correlate Vulnerabilities to Application Assets
2. Map Threat to Vulnerabilities Using Threat Trees
3. Map Threat to Security Flaws Using Use and Abuse Cases
4. Enumerate and Score Vulnerabilities
OUTPUT
Map of existing vulnerabilities to the nodes of a threat tree
Enumeration of these vulnerabilities using CVE-CWE
Scoring of these using CVSS-CWSS
List of threats-attacks-vulnerabilities-assets
MODEL THE ATTACKS
INPUT
Application Technical Scopre (Stage II) and Application Decomposition (Stage III)
Attack libraries-patterns
List of threats, attacks, and vulnerabilities to the application assets (Stage V)
1. Identify Application Attack Surface
2. Derive Attack Trees for Threats and Assets
3. Map Attack Vectors to Nodes of Attack Trees
4. Identify Exploits and Attack Paths using Attack Trees
OUTPUT
Application attack surface
Attack trees with attack scenarios for targeted assets
Attack tree mapping to vulnerabilities for impacted assets
List of possible attack paths to exploits including the attack vectors
RISK & IMPACT ANALYSIS
INPUT
Preliminary BIA (Stage I)
Technical Scope (Stage II)
Application Decomposition (Stage III)
Threat Analysis (Stage IV)
Vulnerability Analysis (Stave V)
Attack Analysis (Stage VI)
Mapping of attacks to controls
Technical standards for controls
1. Qualify and Quantify Business Impacts
2. Identify Gaps in Security Controls
3. Calculate Residual Risks
4. Identify Risk Mitigation Strategies
OUTPUT
Application risk profile
Quantitative and qualitative risks report
Threat matrix with threats, attacks, vulnerabilities, business impact
Residual risk to business
Risk mitigation strategy-options