
expoQA®26
MADRID 26th, 27th & 28th May
expoqa.eu

TMT
Threat Modelling Testing
from modelling to integration

WHOAMI

Tester since 2014

Tech advocate

Cybersecurity passionate

Aerialist

World **Traveler**

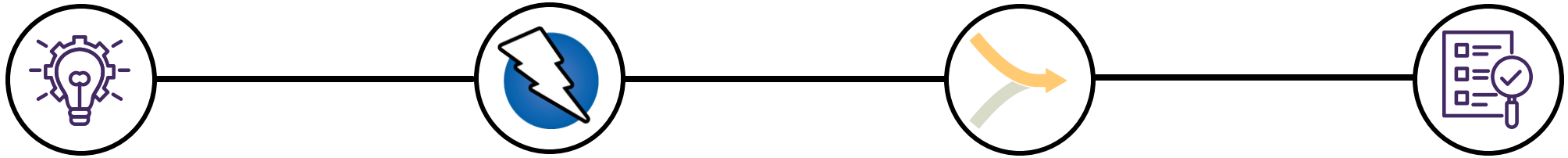
@swtestingsoul

saramartinezginer

sara@testingsoul.com

www.testingsoul.com

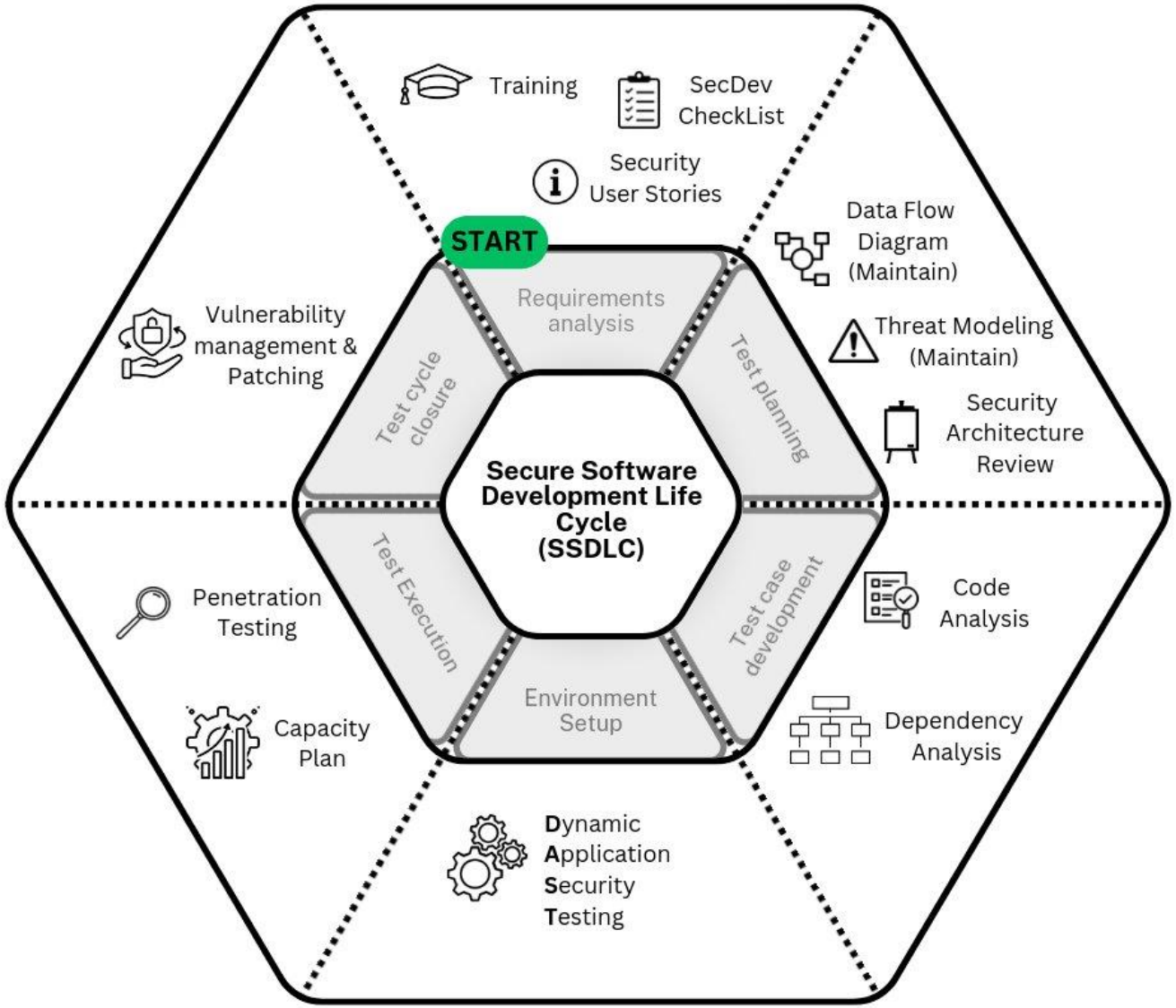
GOALS & TAKEAWAYS



SSDLC **Threat Models** **Hands-on!** **Prevention**

"Security is quality" – Security testing is not an afterthought, it's a set of feedback loops across the lifecycle.

SSDLC: SECURE SOFTWARE DEVELOPMENT LIFECYCLE



If you are part of the cycle, you will have responsibility for **testing** and **cybersecurity**.

CASE STUDY

140K Childcare Records Exposed in CRM Database Leak

140K
Records Exposed

Sensitive Data
Personal and contact information

ROOT CAUSES

- Public exposure of Elasticsearch
- No authentication or access controls
- Network segmentation gap
- Lack of monitoring and alerts

INCIDENT TIMELINE

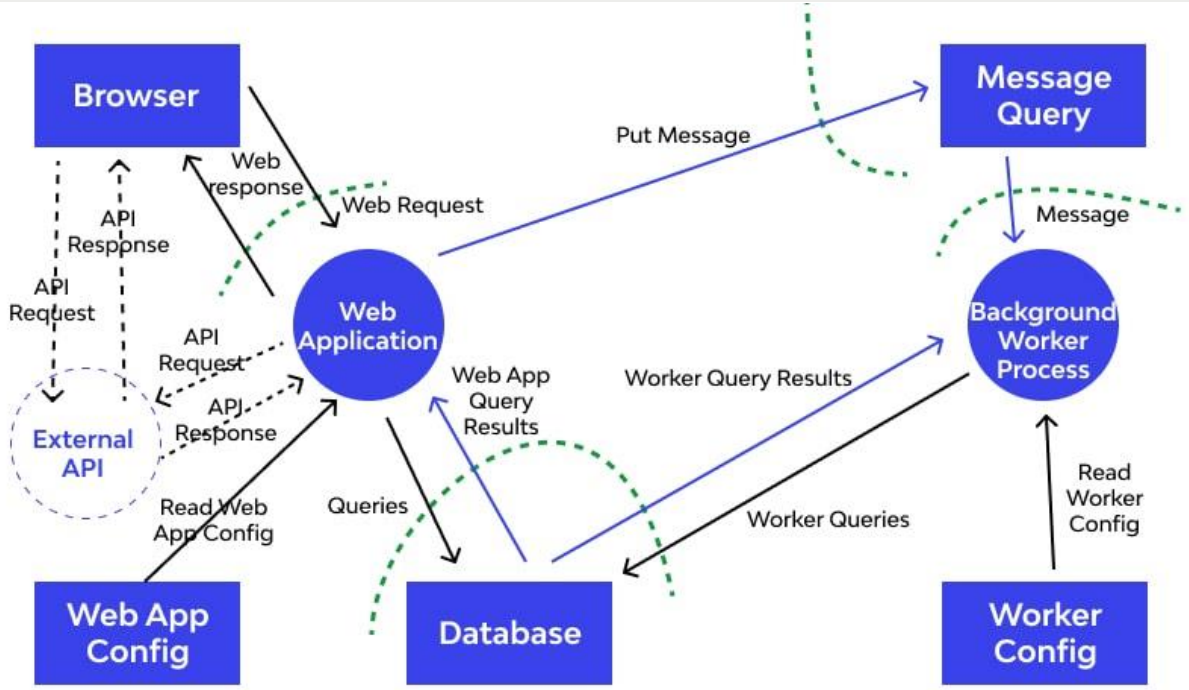
DEC 03, 2025 09:15 AM	DEC 03, 2025 09:45 AM	DEC 03, 2025 11:30 AM	DEC 03, 2025 01:20 PM	DEC 03, 2025 04:10 PM
Exposure Detected	Control Not Implemented	Incident Reported	Remediation Started	Containment Confirmed

Source: https://www.esecurityplanet.com/threats/140k-childcare-records-exposed-in-crm-database-leak
Source: https://lineleader.com/privacy-policy

THREAT MODELING

"Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics."

https://www.threatmodelingmanifesto.org/



THREAT MODELING

MANIFESTO VALUES

- **A culture of finding and fixing design issues**
  over checkbox compliance
- **People and collaboration**
  over processes, methodologies, and tools
- **A journey of understanding**
  over a security or privacy snapshot
- **Doing threat modeling**
  over talking about it
- **Continuous refinement**
  over a single delivery.

THREAT MODELING

PATTERNS

- **Systematic Approach**
  Structured and repeatable security analysis
- **Informed Creativity**
  Creative thinking with security context
- **Varied Viewpoints**
  Diverse expertise improves threat discovery
- **Useful Toolkit**
  Tools improve consistency and efficiency
- **Theory into Practice**
  Apply proven real-world security techniques

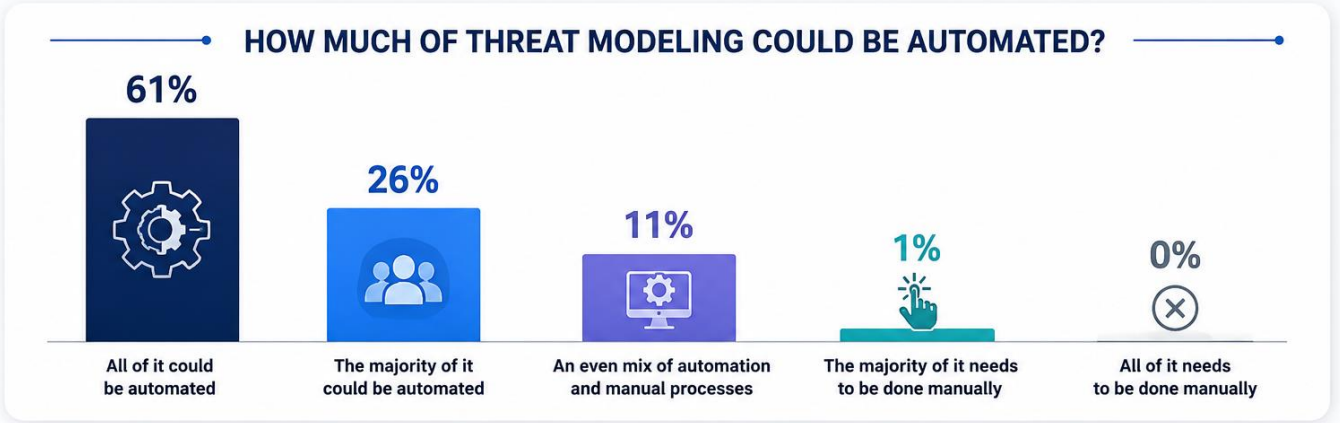
ANTI-PATTERNS

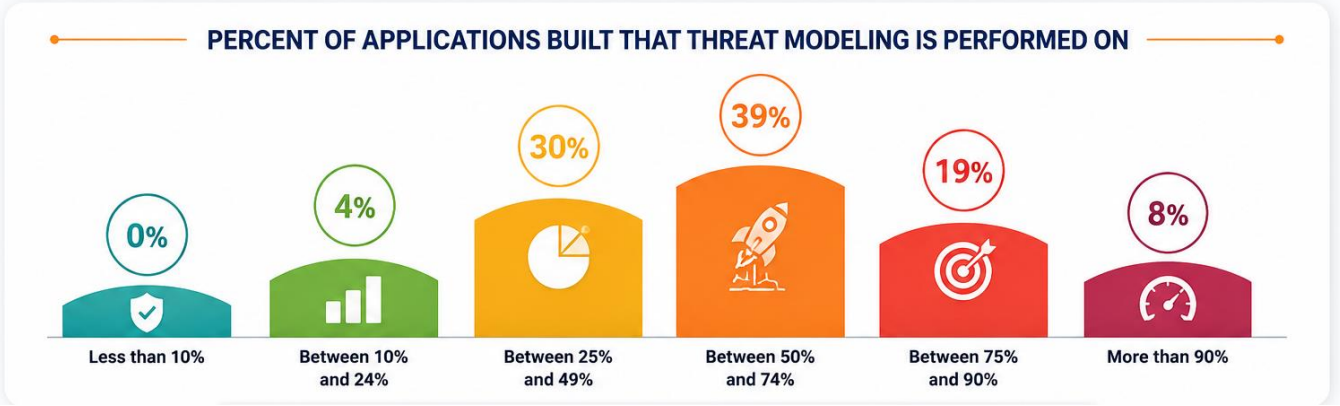
- **Hero Threat Modeler**
  Security should not depend on individuals
- **Admiration for the Problem**
  Focus on solutions, not complexity
- **Tendency to Overfocus**
  Keep the big security picture
- **Perfect Representation**
  Multiple models reveal different risks

THREAT MODELING

SOME DATA

- **Only 25%** perform threat modeling during early design and requirements phases
- **Most organizations** apply threat modeling to only **50–74% of applications**
- **61% believe** threat modeling can be **fully automated**
- **Automation is growing**, but expert validation remains essential





Source: https://www.securitycompass.com/reports/the-2023-state-of-threat-modeling/


THREAT MODELING
IS NOTHING WITHOUT CONTROL
www.pirelli.com
PIRELLI

CASE STUDY: Salesforce Experience Cloud Misconfiguration (2026)

Attackers exploited misconfigured public Salesforce portals to steal CRM data from hundreds of organizations.

WHAT HAPPENED?

- Exposed APIs
- Overly permissive guest access
- CRM data accessed without auth
- Impacted many major organizations

Source: TechRadar (Jan 2026) | ITPro (Jan 2026)

WHY THIS HAPPENED (BAD OR UNTESTED THREAT MODEL)

Broken Trust Boundary
Public users could access internal data

Excessive Permissions
Guest profiles had more access than needed

Missing Abuse Scenarios
API scraping, enumeration, token abuse not considered

Weak Defense in Depth
Single config issue exposed sensitive data

Failure of Zero Trust
Public-facing components overtrusted

KEY TAKEAWAY

"The breach was caused by insecure assumptions — not advanced hacking."

Threat modeling must challenge what public users should really be able to access.

ATTACK TIMELINE: How the Exposure Happened

1. Public Portal Deployed
2. Guest Permissions Enabled
3. APIs Exposed Publicly
4. Attackers Scanned & Enumerated
5. CRM Data Extracted
6. Incident Detected
7. Mitigations Applied

Root Cause: Insecure assumptions, excessive permissions and untested threat scenarios.

TMT
Threat Modelling Testing

THREAT MODELING TESTING

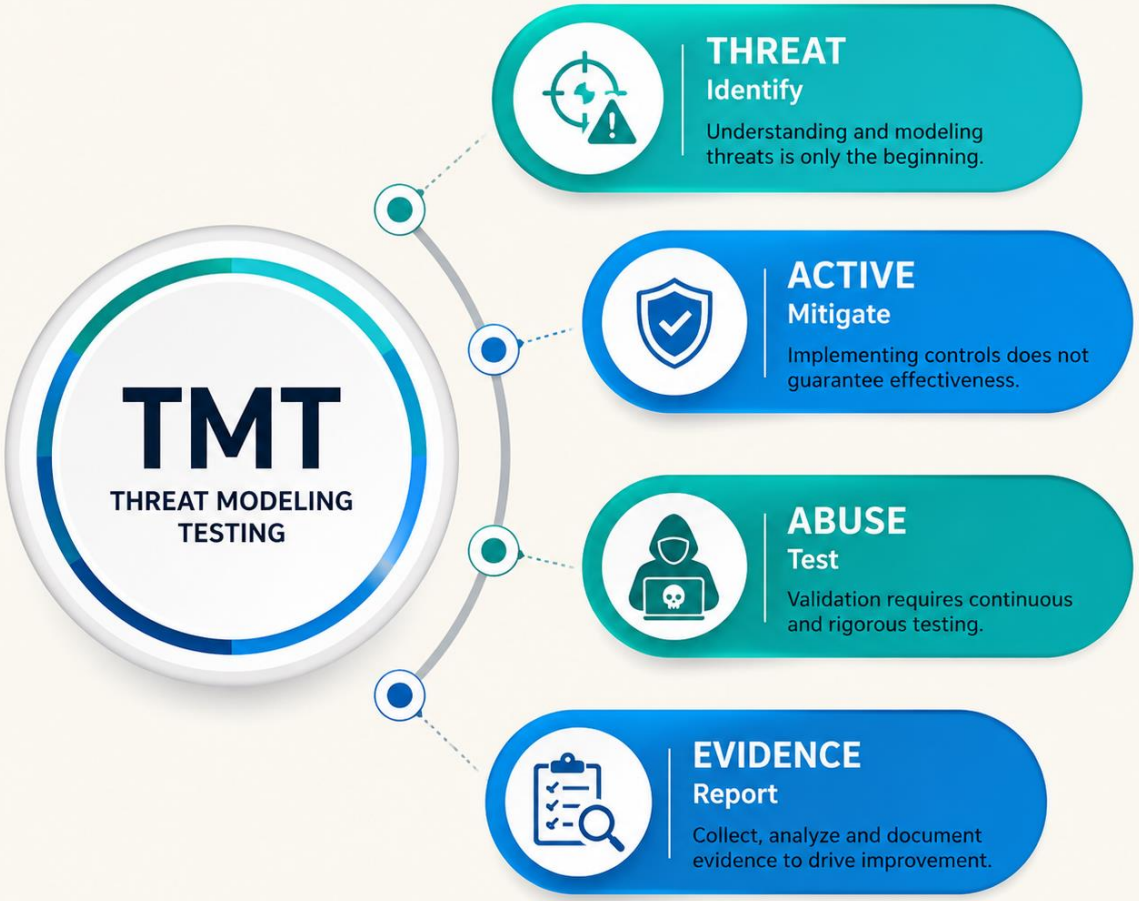
GOALS

1. **Identify**
   Understanding and modeling threats is only the beginning.
2. **Mitigate**
   Implementing controls does not guarantee effectiveness.
3. **Test**
   Validation requires continuous and rigorous testing.
4. **Report**
   Collect, analyze and document evidence to drive improvement.

THREAT MODELING TESTING

DEFINITION

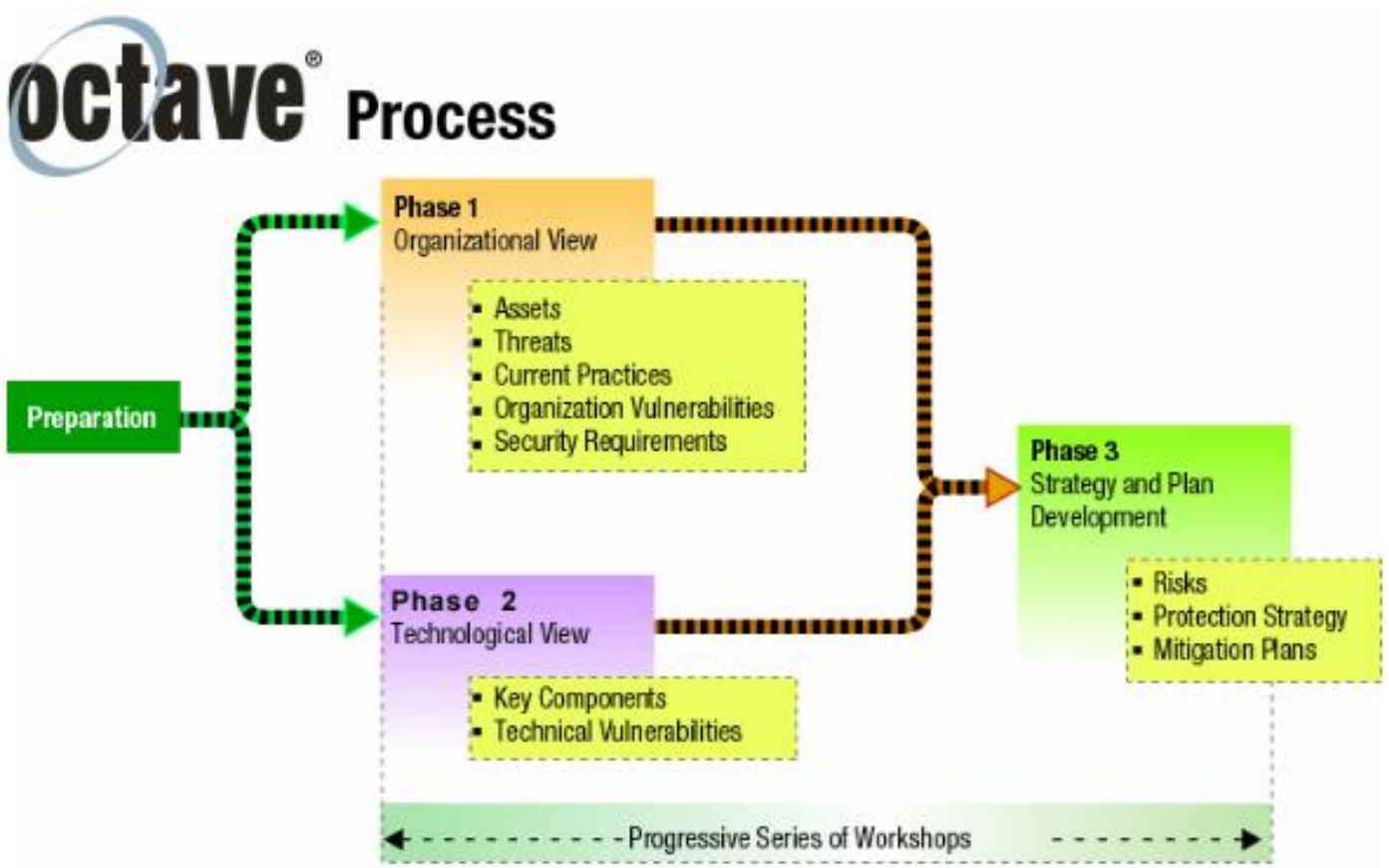
TMT combines **threat modeling** with **testing**, creating a **robust security** approach to **mitigate** risks.



THREAT MODELING FAMEWORKS

FRAMEWORKS	
STRIDE	categorize threats (Spoofing, Tampering, Repudiation...)
DREAD	prioritize risks by likelihood/impact
PASTA	risk analysis driven by business/process context
LINDDUN	privacy-focused threat modeling

THREAT MODELING FAMEWORKS



THREAT MODELING FAMEWORKS

PASTA THREAT MODELING RACI DIAGRAM

APPLICATION THREAT MODELING ACTIVITIES per STAGE	BU/Product Groups						Corporate Functions							3rd Party	
	MGT	PMO	BA	ARC	SWE	QA	SYS	SOC	RL	PC	SA	EA	CTO	VA	PT
STAGE 1 - DEFINE BUSINESS OBJECTIVES - Est. New TM = 2-4 hours | Est. Repeat TM = < 1 hour	A	R	R	A	I	I	I	-	I	R	I	I	R	-	-
Obtain business objectives for product or application	A	I	R	A	I	I	I	-	I	-	-	I	I	-	-
Identify regulatory compliance obligations	A	I	I	A	I	I	I	-	I	R	-	I	I	-	-
Define a risk profile or business criticality level for the application	A	I	I	A	I	I	I	-	I	C	I	I	R	-	-
Identify the key business use cases for the application/product	A	R	R	A	I	I	I	-	I	-	-	I	I	-	-
STAGE 2 - TECHNICAL SCOPE - Est. New TM = 3-4 hours | Est. Repeat TM = 1-3 hours	I	I	C	A	R/A	C	I	-	I	-	I	C	I	-	-
Enumerate software applications/database in support of product/application	I	I	C	A	R/A	C	I	-	-	-	-	C	I	-	-
Identify any client-side technologies (Flash, DHTML5, etc.)	I	I	C	A	R/A	C	I	-	-	-	I	C	I	-	-
Enumerate system platforms that support product/application	I	I	C	A	R/A	C	I	-	-	-	I	C	I	-	-
Identify all application/product actors	I	I	C	A	R/A	C	I	-	-	-	I	C	I	-	-
Enumerate services needed for application/product use & management	I	I	C	A	R/A	C	I	-	-	-	I	C	I	-	-
Enumerate 3rd party COTS needed for solution	I	I	C	A	R/A	C	I	-	-	-	I	C	I	-	-
Identify 3rd party infrastructures, cloud solutions, hosted networks, mobile devices	I	I	C	A	R/A	C	I	-	-	-	I	C	I	-	-
Obtain business objectives for product or application	I	I	C	A	R/A	C	I	-	I	-	I	C	I	-	-
STAGE 3 - APPLICATION DECOMPOSITION - Est. New TM = 8 hours | Est. Repeat TM = 4 hours	I	I	I	A	R	C	C	-	I	-	-	C	-	-	-
Perform data flow diagram of application environment	I	I	I	A	R	I	C	-	-	-	-	C	-	-	-
Define application trust boundaries/trust models	I	I	I	A	R	C	C	-	-	-	-	C	-	-	-
Enumerate application actors	I	I	I	A	R	C	C	-	-	-	-	C	-	-	-
Identify any stored procedures/batch processing	I	I	I	A	R	C	C	-	-	-	-	C	-	-	-
Enumerate all application use cases (ex: login, account update, delete users, etc.)	I	I	I	A	R	C	C	-	-	-	-	C	-	-	-
STAGE 4 - THREAT ANALYSIS - Est. New TM = 6 hours | Est. Repeat TM = 2 hours	I	I	R/A	A	R/A	R/A	C	C	-	-	-	I	-	-	-
Gather/correlate relevant threat intel from internal/external threat groups	I	I	R/A	A	C	I	C	C	-	-	-	C	-	-	-
Review recent log data around application environment for heightened security alerts	-	-	I	A	R	R/A	I	C	-	-	-	C	-	-	-
Gather audit reports around access control violations	-	I	I	A	R	C	I	C	-	-	-	C	-	-	-
Identify probable threat motives, attack vectors & misuse cases	I	I	I	A	R/A	C	I	C	-	-	-	C	-	-	-
STAGE 5 - VULNERABILITY ASSESSMENT - Est. New TM = 12 hours | Est. Repeat TM = 6 hours	I	I	I	A	R	C	I	C	I	-	-	I	-	R/A	R
Conduct targeted vulnerability scans based upon threat analysis	-	-	-	A	R	C	I	C	I	-	-	I	-	R	R
Identify weak design patterns in architecture	-	-	-	A	R	C	I	-	-	-	-	C	-	R	C
Review/correlate existing vulnerability data	I	I	I	A	R	I	I	C	-	-	-	I	-	R/A	I
Map vulnerabilities to attack tree	-	I	I	A	R	I	I	-	-	-	-	C	-	C	I
STAGE 6 - ATTACK ENUMERATION - Est. New TM = 10 hours | Est. Repeat TM = 5 hours	I	I	I	A	R	R	-	-	I	-	-	C	I	I	R/A
Enumerate all inherent and targeted attacks for product/application	I	I	I	A	R	C	-	-	I	-	-	C	I	I	R/A
Map attack patterns to attack tree vulnerability branches (attack tree finalization)	-	-	-	A	R	C	-	-	I	-	-	C	-	I	A
Conduct targeted attacks to determine probability level of attack patterns	-	-	-	A	C	R	-	-	I	-	-	C	-	I	R/A
Reform threat analysis based upon exploitation results	I	I	I	A	R	C	-	-	I	-	-	C	I	I	C
STAGE 7 - RESIDUAL RISK ANALYSIS - Est. New & Repeat TM = 5 days (inc. countermeasure dev.)	C	I	I	A	R	C	C	C	I	I	C	C	I	I	R
Review application/product risk analysis based upon completed threat analysis	I	I	I	A	R	C	I	C	I	I	C	C	I	I	R
List recommended countermeasures for residual risk reduction	I	I	I	A	R	C	C	C	I	I	C	C	I	I	R
Re-evaluate overall application risk profile and report.	C	I	I	A	R	C	I	I	I	C	C	C	I	I	I

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

Corporate Functions

Office of the CTO
Compliance
Security (ISRM)

RACI Legend

R	Responsible
A	Accountable
C	Consulted (2 way)
I	Informed (1 way)

VERSPRITE

VerSprite's Process for Attack, Simulation and Threat Analysis (PASTA) benefits stakeholders by assessing threats to your application environment by designing secure applications and deciding how to mitigate risks by applying risk mitigation strategies. Examples include Architects, Developers, Security Testers, Project Managers, Business Managers, and Information Risk Officers. Learn more >>

“What isn’t tested eventually gets exploited.”

STRIDE
As testing engine

THREAT MODELING: STRIDE

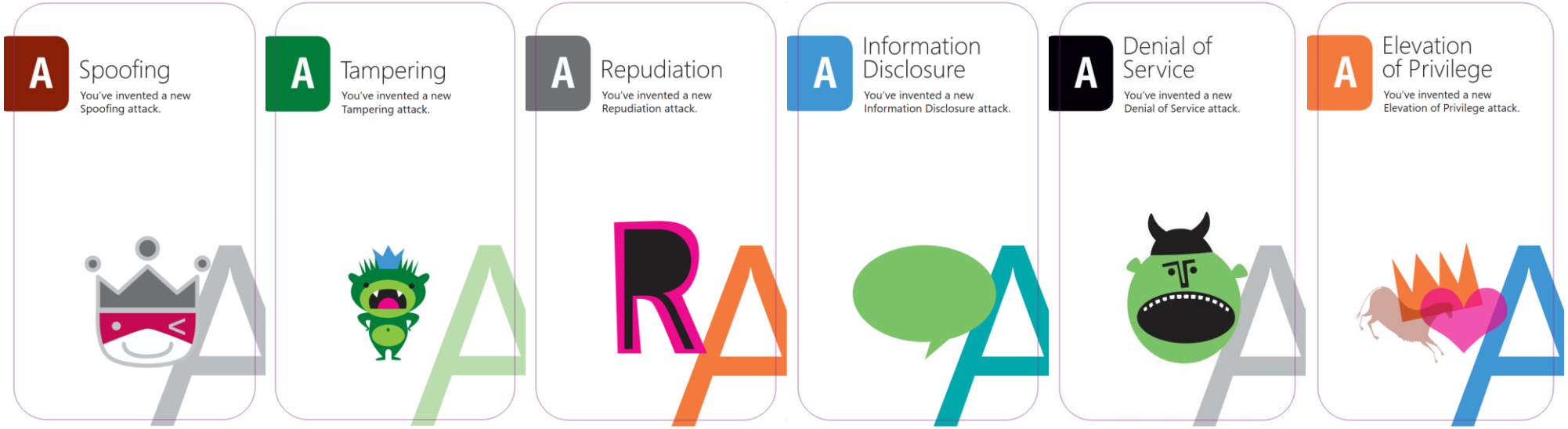
	Category	Violates	Definition	Examples
S	Spoofing	Authenticity	Pretending to be something or someone other than yourself	An attacker steals the authentication token of a legitimate user and uses it to impersonate the user.
T	Tampering	Integrity	Modifying something on disk, network, memory, or elsewhere	An attacker abuses the application to perform unintended updates to a database.
R	Repudiation	Accounting	Claiming that you did not do something or were not responsible; can be honest or false	An attacker manipulates logs to cover their actions.
I	Information Disclosure	Confidentiality	Someone obtaining information they are not authorized to access	An attacker extracts data from a database having user account info.
D	Denial of Service	Availability	Exhausting resources needed to provide service	An attacker locks a legitimate user out of their account by performing many failed authentication attempts.
E	Elevation of Privileges	Authorization	Allowing someone to do something they are not authorized to do	An attacker tampers with a JWT to change their role.

After Break!

Hands-on!
Threat Modeling

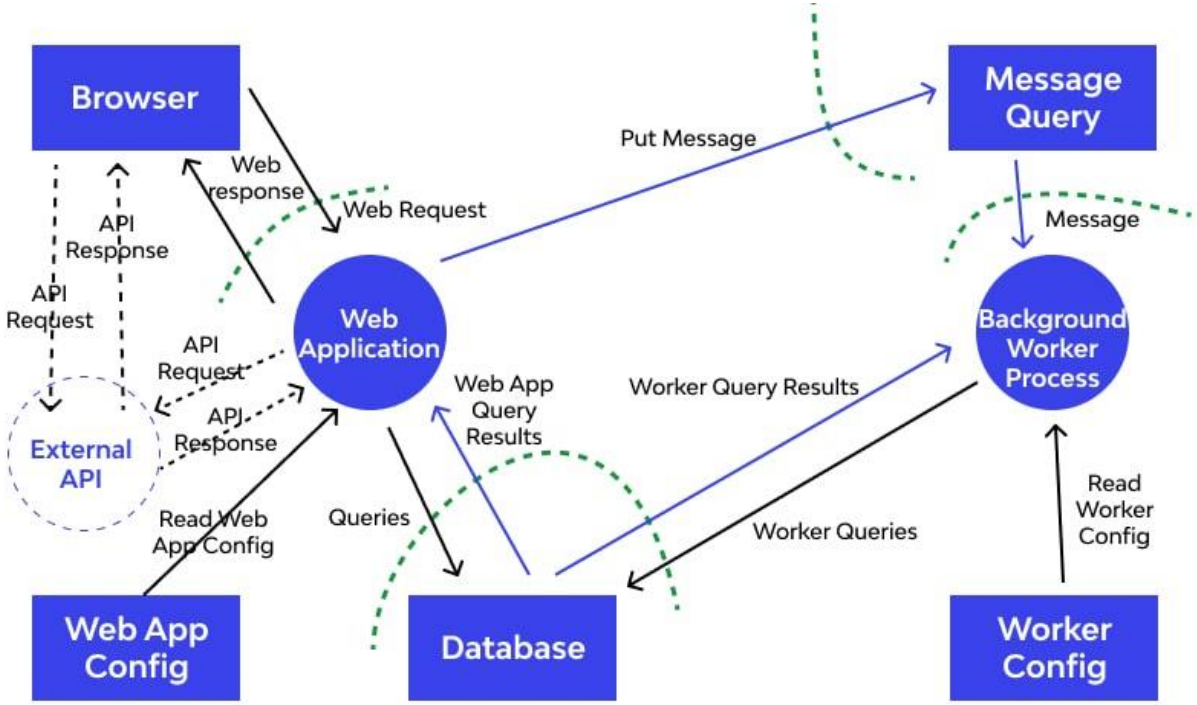
THREAT MODELING: LET'S PLAY!

S	T	R	I	D	E



GAME GUIDE

Category	Violates	Definition	Examples
Spoofing	Authenticity	Pretending to be something or someone other than yourself	An attacker steals the authentication token of a legitimate user and uses it to impersonate the user.
Tampering	Integrity	Modifying something on disk, network, memory, or elsewhere	An attacker abuses the application to perform unintended updates to a database.
Repudiation	Accounting	Claiming that you did not do something or were not responsible; can be honest or false	An attacker manipulates logs to cover their actions.
Information Disclosure	Confidentiality	Someone obtaining information they are not authorized to access	An attacker extracts data from a database having user account info.
Denial of Service	Availability	Exhausting resources needed to provide service	An attacker locks a legitimate user out of their account by performing many failed authentication attempts.
Elevation of Privileges	Authorization	Allowing someone to do something they are not authorized to do	An attacker tampers with a JWT to change their role.

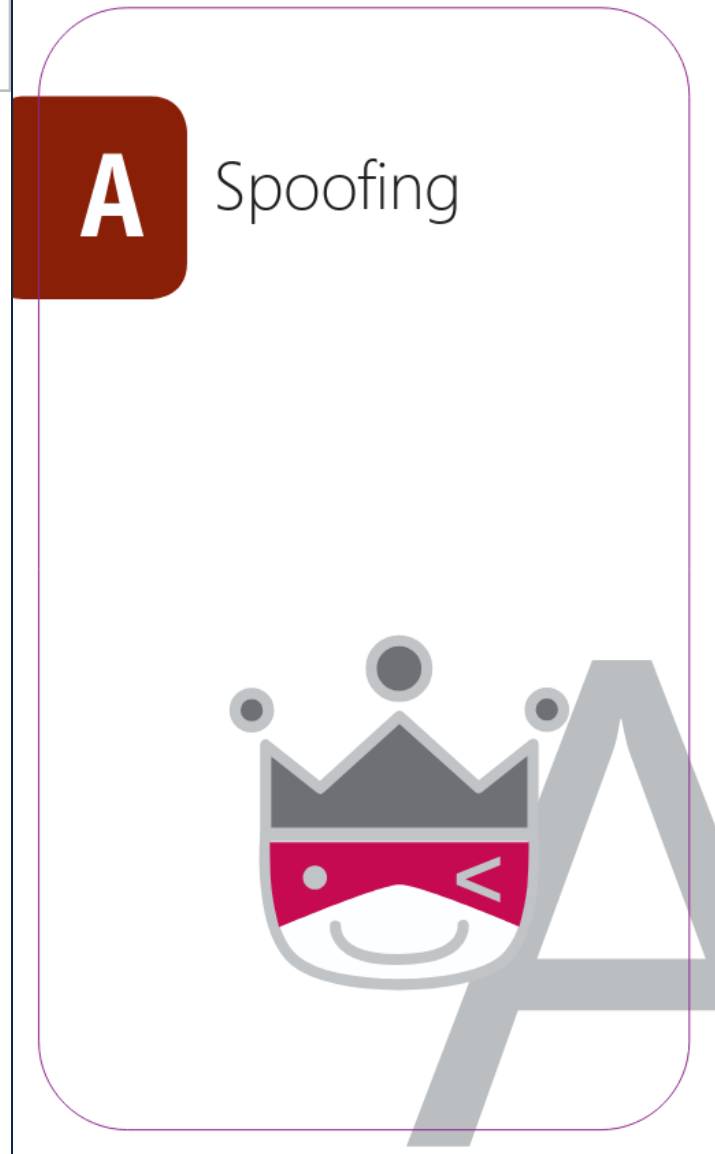


TMT CHEATSHEETS

https://github.com/testingsoul/TMT-CheatSheets

	Violates	Definition
SPOOFING	Authenticity	Pretending to be something or someone other than yourself

```
@TM:XXX
Scenario: Invalid access to endpoint
  Given I set an invalid token
  And I configure the target endpoint
  When I send the request
  Then the response code is 401 Unauthorized
  And a security event is registered
```



	Violates	Definition
TAMPERING	Integrity	Modifying something on disk, network, memory, or elsewhere

```
@TM:XXX
Scenario: Edit parameter with invalid data
  Given I set a valid token
  And I configure the target endpoint
  And I set a body to set invalid data
  When I send the request
  Then the response code is 400 Bad Request
  And a security event is registered
  And parameter is not modified in DB
```

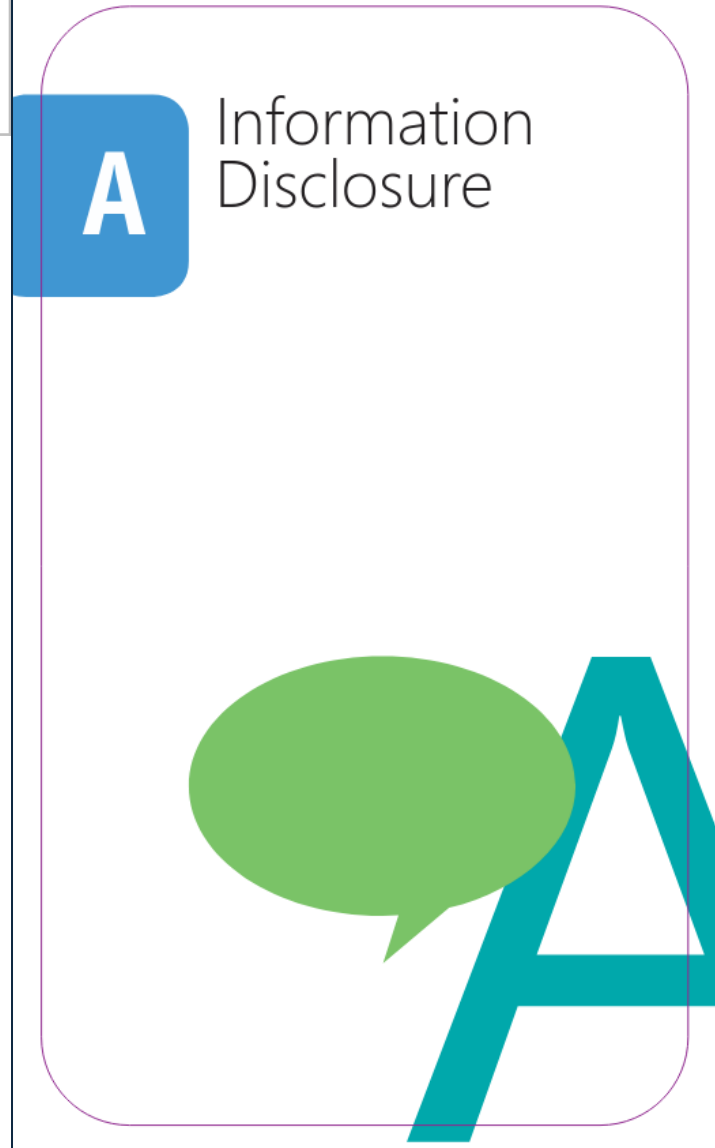
A Tampering

	Violates	Definition
REPUDIATION	Accounting	Claiming that you did not do something or were not responsible; can be honest or false

```
@TM:XXX
Scenario: Delete evidence in logs is forbidden
  Given I have a privileged account
  And I access to audit logs
  When I delete security audit logs
  Then the action is denied
  And a security event is registered
```

	Violates	Definition
INFORMATION DISCLOSURE	Confidentiality	Someone obtaining information they are not authorized to access

```
@TM:XXX
Scenario: Get values from encrypted parameter without permissions
  Given I set a valid token without decryption permissions
  And I configure the target endpoint
  And I set a body to get encrypted parameters
  When I send the request
  Then the response code is 200
  And the sensitive data is not decrypted
```



	Violates	Definition
DENIAL OF SERVICE	Availability	Exhausting resources needed to provide service

```
@TM:XXX
Scenario: The service remains available under forced load
  Given a ramp-up load is set over limits
  When I execute the load profiles
  Then the 95th percentile latency stays stable for legitimate traffic
  And the 5xx error rate stays below 1%
  And IPs exceeding the threshold receive a 429 response
  And no total service outage occurs
  And saturation metrics and alerts are generated in monitoring
```

A

Denial of Service

	Violates	Definition
ELEVATION OF PRIVILEGES	Authorization	Allowing someone to do something they are not authorized to do

```
@TM:XXX
Scenario: Edit parameter without permissions
  Given I set a valid token with limited permissions
  And I configure the target endpoint
  And I set a body to set unauthorized parameters
  When I send the request
  Then the response code is 403 Forbidden
  And a security event is registered
  And parameter is not modified in DB
```

CONCLUSIONS

Call to Action:

- Review tests scenarios
- Add at least 1 test per dimension (STRIDE)
- Integrate them into your CI

expoQA®26

MADRID 26th, 27th & 28th May

Thank you for attending

expoqa.eu