Iterative
Threat Modeling

/thoughtworks

Jagdsh Chand (Jags)

About Me

Mobile Application Developer
AI Enthusiast
Penetration Tester
DevOps
Frontend Developer
Extreme Programmer
Backend Developer
DevSecOps

jagdshlk@thoughtworks.com
LinkedIn: https://www.linkedin.com/in/jagdshlk/

Expectations

Threat Modelling (TM)

Agile Threat Modelling

Steps in Threat Modelling explained with an Juice Shop example

Iterative Threat Modelling

TM in Security Development Lifecycle (SDL)

Threat Modelling

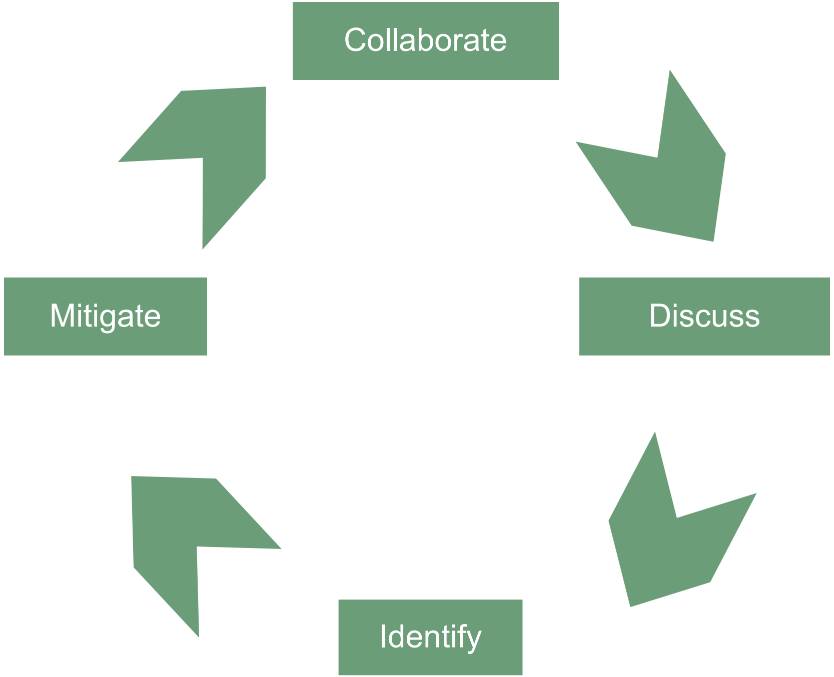
thoughtworks

/thoughtworks | Threat Modelling

Discussing about things that can go wrong

so we can do something about them

before they go wrong.



Misconceptions about Threat Modelling

As a security process, threat modeling is subject to several misconceptions. Some people believe threat modeling is only a design-stage activity

Penetration testing and code reviews can't substitute for threat modeling

Penetration testing and secure code review are two activities that are effective for finding bugs in code. However, security assessments are better at uncovering design flaws.

There's a good reason to conduct a threat model after deployment.

Understanding the issues in the current deployment influences future security architecture strategy, and monitoring weaknesses allows for faster and more effective remediation.

Threat modeling isn't that complicated and it takes time

Many developers are intimidated by the idea of threat modeling. At first glance, it can seem daunting. The key is to start with basic best practices.

Agile Threat Modelling

What do we need ....?

1. Need to bring defensive mindset into the development team
2. Need to create a collaborative, tailored approach with the development team for capturing threats proactively
3. Need to overcome the human errors because of unawareness
4. Need a simple exercise which can be introduced at any time of the project delivery and can be repeated iteratively

Agile Threat Modelling

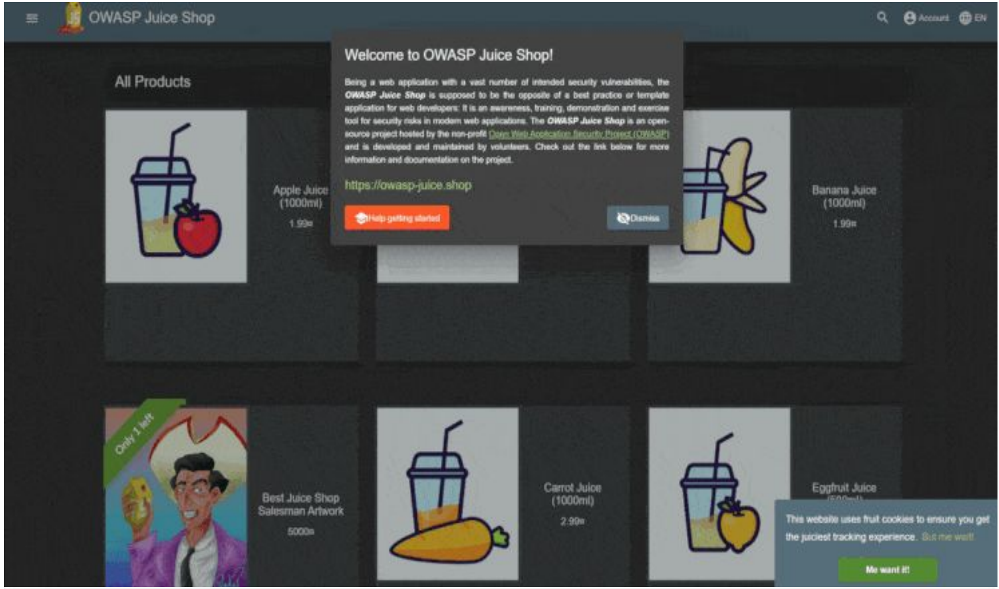
Steps

1. What do we want to accomplish?
2. What are we building?
3. What can go wrong?
4. What are we going to do about it?
5. Did we do a good job?

OWASP Juice Shop

Guinea pig

OWASP Juice Shop is probably the most modern and sophisticated <u>insecure</u> web application!

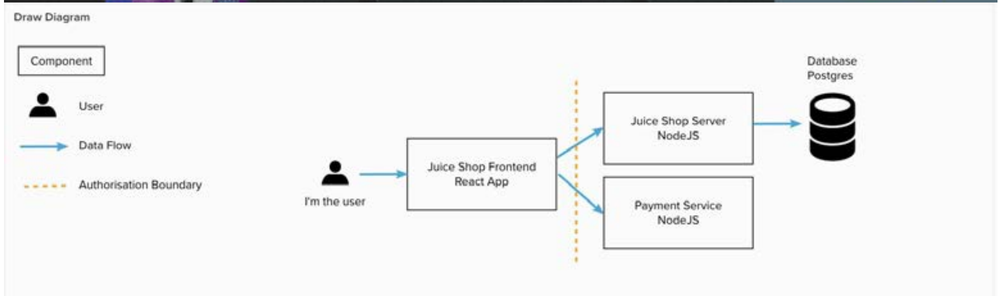


Welcome to OWASP Juice Shop!

Being a web application with a vast number of intended security vulnerabilities, the **OWASP Juice Shop** is supposed to be the opposite of a best practice or template application for web developers: It is an awareness, training, demonstration and exercise tool for security risks in modern web applications. The **OWASP Juice Shop** is an open-source project hosted by the non-profit Open Web Application Security Project (OWASP) and is developed and maintained by volunteers. Check out the link below for more information and documentation on the project.

https://owasp-juice.shop

This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!

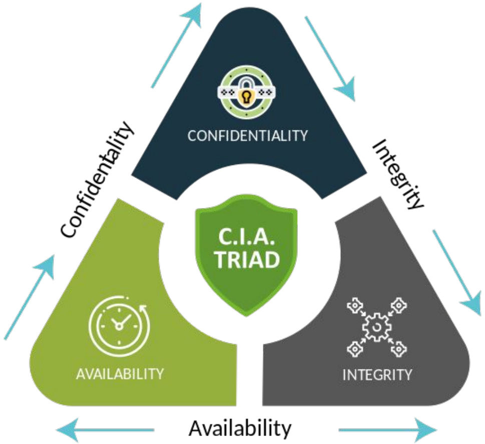


Before Starting....

Define security objective

Answer the following questions

- What kind of losses puts the organisation's objective in jeopardy? Is it having the customer database stolen or payments?
- Are we worried about fraud? Malicious insiders? Particularly capable hackers?



Confidentiality

It is the process of keeping an organization or individual's data private and ensuring only authorized people can access it.

Integrity

It refers to data that hasn't been tampered with. Data that has been tampered with or compromised has lost its integrity. Integrity ensures the protection of data in transit, use, and storage.

Availability

Applications, systems, or data are of no use to an organization or its customers if they are not accessible as and when required – as in the case of a denial-of-service attack..

Example: Security Objective

Juice shop

- Reduce the chance of any event which affects the reputation of the juice shop negatively, particularly leading to reduction in sales
- Reduce the chance of a breach of personally identifiable information of customers
- Reduce the risk of malicious alteration leading to financial loss
- The chance of a malicious denial of availability of the shop to customers should be reduced.

Agile Threat Modelling

Steps

1. What do we want to accomplish?
2. What are we building?
3. What can go wrong?
4. What are we going to do about it?
5. Did we do a good job?

Scoping

Little and Often

Too much scope at once will make no findings in the time available or you will overrun dramatically. It is much better to timebox threat modelling into manageable chunks, performing the activity.

New or a upcoming security sensitive feature, such as login, checkout flow

A particular microservice and its collaborating services

A high level overview of a system to identify security tech debt.

The continuous delivery pipeline and delivery infrastructure.

Example: Scoping

One feature

<u>Customer Login</u>

DevSecOps-2023

As a customer,
I need a page where I can enter login credentials
So that I can access the application as logged in user.

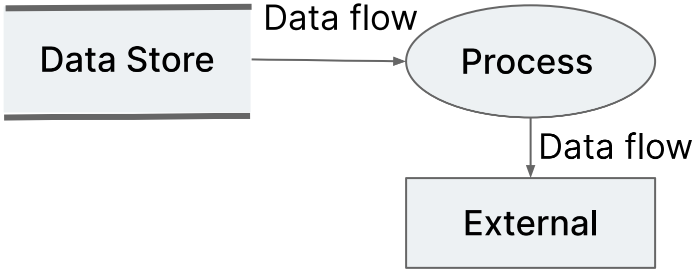
Agile Threat Modelling

Steps

1. What do we want to accomplish?
2. What are we building?
3. What can go wrong?
4. What are we going to do about it?
5. Did we do a good job?

Software-Centric approach

In a software-centric model we our represent our systems in holistic view with software layers, highlighting how data flows from one system to another.



Data flow diagram (DFD)

Stakeholders:

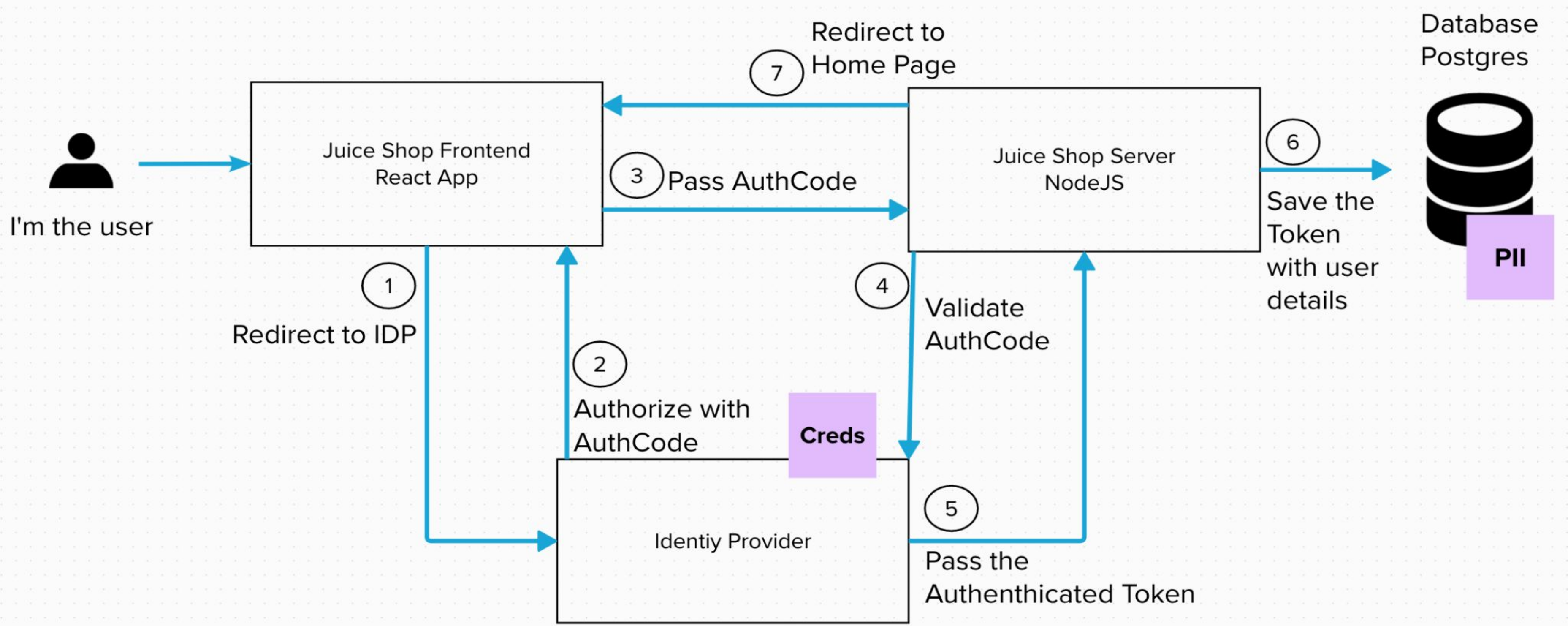
- **Engineers**

Principal:

- **Identify entry points, assets & trust levels that represent the access rights**
- **Identify system interactions**
- **Capture the End-to-End flow including external entity where data goes for every use case**

Example: Data flow diagram

Juice shop



Agile Threat Modelling

Steps

1. What do we want to accomplish?
2. What are we building?
3. What can go wrong?
4. What are we going to do about it?
5. Did we do a good job?

Evil Brainstorming

This is where we wear attackers hat in coming up with ways to attack, break or frustrate a particular bit of software from attackers mindset.

Stakeholders:

- Security team (Optional)
- Product Owners & BA
- Engineers

Principal:

- Always be aware of the time, it is common to go into rabbit hole discussion
- Focus on quantity over quality

Methodology. No ‘Best’ way

	Focused	Scope / System Model	Output	Good for
PASTA (Process for Attack Simulation and Threat Analysis)	Attacker Focused	Exhaustive system model with multiple perspectives	Detailed risk assessments. countermeasures, system diagrams and essays	Comprehensive assurance exercises
Attack Trees	Attacker Focused	Exhaustive for a single attacker motivation or goal	Graph like attack tree, overlaid with countermeasures. Can become quite complex	Focussing on a critical component in the context of a high risk attacker goal
VAST (Visual, Agile, and Simple Threat modeling)	Enterprise Focused	Exhaustive for automation, integration, and collaboration with enterprise focus with automated tools	Threat models with mitigation	Focussing on development and infrastructure teams for large enterprises
😍 Timeboxed STRIDE	Developer Focused	Small: This sprint’s changes Or big picture as security debt	Additional Acceptance Criteria, Tech Debt Stories, Additions to Definition of Done	**Agile teams**

Extended from Adam Shostack’s talk: https://i.blackhat.com/us-18/Wed-August-8/us-18-Shostack-Threat-Modeling-in-2018.pdf

Spoofed Identity

Can someone spoof an identity
and then abuse its authority?

Spoofing identity allows attackers
to do things they are not supposed to do.

Key concepts

- **Identity**
- **Authentication**

Tampering with input

How hard is it for an attacker to modify the data they submit to your system?

Can they break a trust boundary and modify the code which runs as part of your system?

Key concepts

- Validation
- Integrity
- Injection

Repudiation of action

How hard is it for users to deny performing an action? What evidence does the system collect
to help you to prove otherwise?

Non-repudiation refers to the ability of a system to ensure people are accountable for their actions.

Key concepts

- Non repudiation
- Logging
- Audit

Information disclosure

Can someone view information
they are not supposed to have access to?

Information disclosure threats involve the exposure or interception of information to unauthorised individuals.

Key concepts

- Confidentiality
- Encryption
- Leakage
- Man in the Middle

Denial of service

Can someone break a system so valid users are unable to use it?

Denial of service attacks work by flooding, wiping or otherwise breaking a particular service or system.

Key concepts

- Availability
- Botnets
- DDoS

Elevation of privilege

Can an unprivileged user gain more access to the system than they should have?

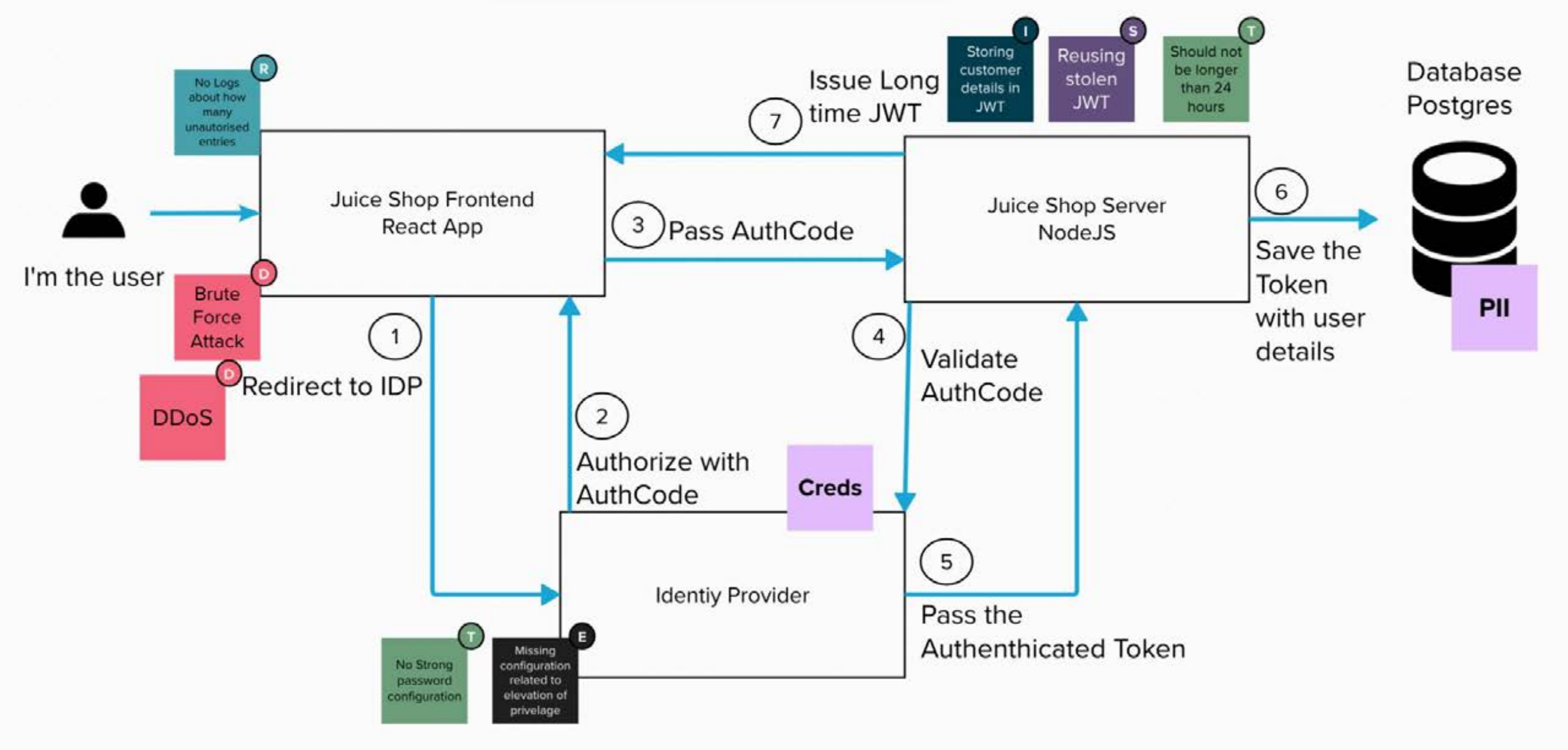
Elevation of privilege attacks are possible because authorisation boundaries are missing or inadequate.

Key concepts

- **Authorisation**
- **Isolation**
- **Blast radius**
- **Remote Code Execution**

Example: Applying STRIDE

Juice shop



I'm the user
Juice Shop Frontend React App
Juice Shop Server NodeJS
Identiy Provider
Database Postgres
PII
Creds
1
Redirect to IDP
2
Authorize with AuthCode
3
Pass AuthCode
4
Validate AuthCode
5
Pass the Authenthicated Token
6
Save the Token with user details
7
Issue Long time JWT
R
No Logs about how many unautorised entries
D
Brute Force Attack
D
DDoS
I
Storing customer details in JWT
S
Reusing stolen JWT
T
Should not be longer than 24 hours
T
No Strong password configuration
E
Missing configuration related to elevation of privelage

Agile Threat Modelling

Steps

1. What do we want to accomplish?
2. What are we building?
3. What can go wrong?
4. What are we going to do about it?
5. Did we do a good job?

Prioritize

This is where people vote the riskiest threats keeping our Security Objective in mind.

Stakeholders:

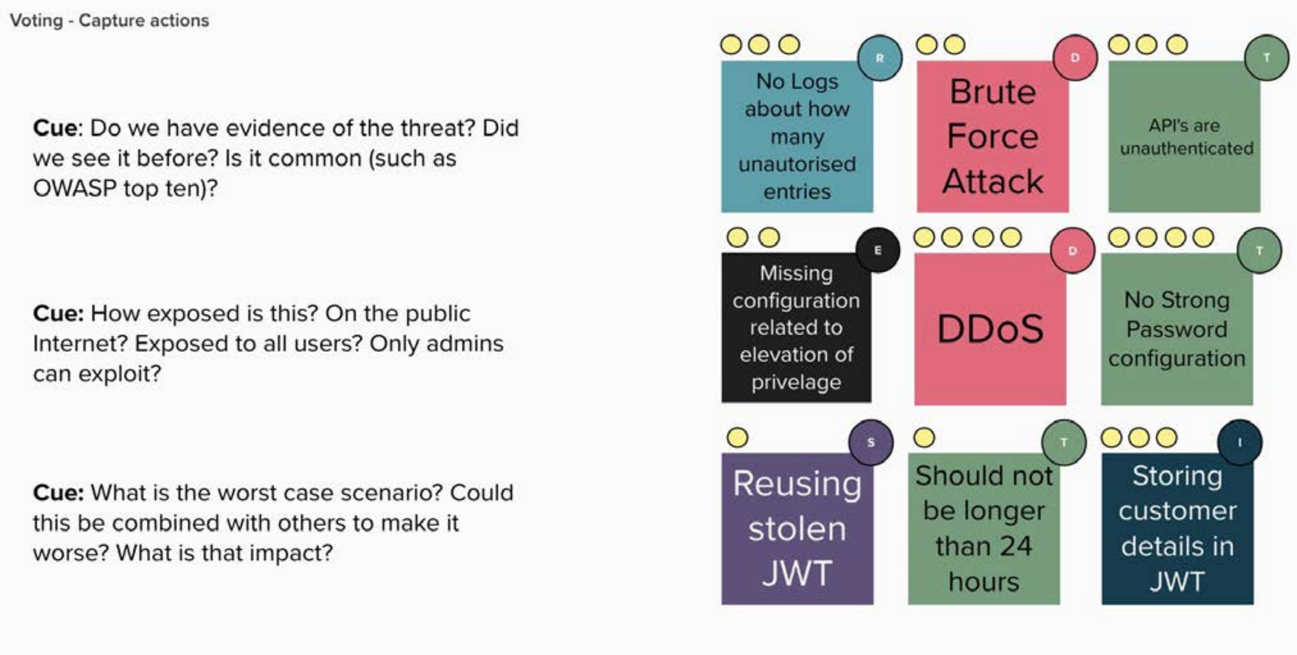
- **Security team (Optional)**
- **Product Owners & BA**
- **Engineers**

Principal: Use DREAD model & Security Objective

- **D**amage – how bad would an attack be?
- **R**eproducibility – how easy is it to reproduce the attack?
- **E**xploitability – how much work is it to launch the attack?
- **A**ffected users – how many people will be impacted?
- **D**iscoverability – how easy is it to discover the threat?

Example: Prioritize

Juice shop



Voting - Capture actions

Cue: Do we have evidence of the threat? Did we see it before? Is it common (such as OWASP top ten)?

Cue: How exposed is this? On the public Internet? Exposed to all users? Only admins can exploit?

Cue: What is the worst case scenario? Could this be combined with others to make it worse? What is that impact?

○○○ R No Logs about how many unautorised entries	○○ D Brute Force Attack	○○○ T API's are unauthenticated
○○ E Missing configuration related to elevation of privelage	○○○○ D DDoS	○○○○ T No Strong Password configuration
○ S Reusing stolen JWT	○ T Should not be longer than 24 hours	○○○ I Storing customer details in JWT

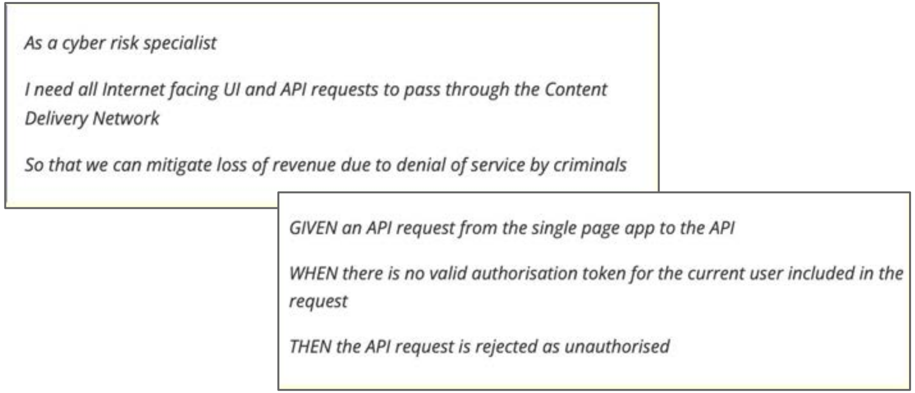
Security Objective:

- Reduce the chance of any event which affects the reputation of the juice shop negatively, particularly leading to reduction in sales
- Reduce the chance of a breach of personally identifiable information of customers.
- Reduce the risk of malicious alteration of data and lead to financial loss.
- Reduce the chance of a malicious denial of availability of the shop and checkout service to customers

Mitigation

Discuss the possible mitigations and capture it as one of the following action item:

* Tech debt
* User Story / Evil Story
* Acceptance Criteria
* Epics
* Spike
* Changes to definition of done
* Cross Functional Requirement



As a cyber risk specialist

I need all Internet facing UI and API requests to pass through the Content Delivery Network

So that we can mitigate loss of revenue due to denial of service by criminals

GIVEN an API request from the single page app to the API

WHEN there is no valid authorisation token for the current user included in the request

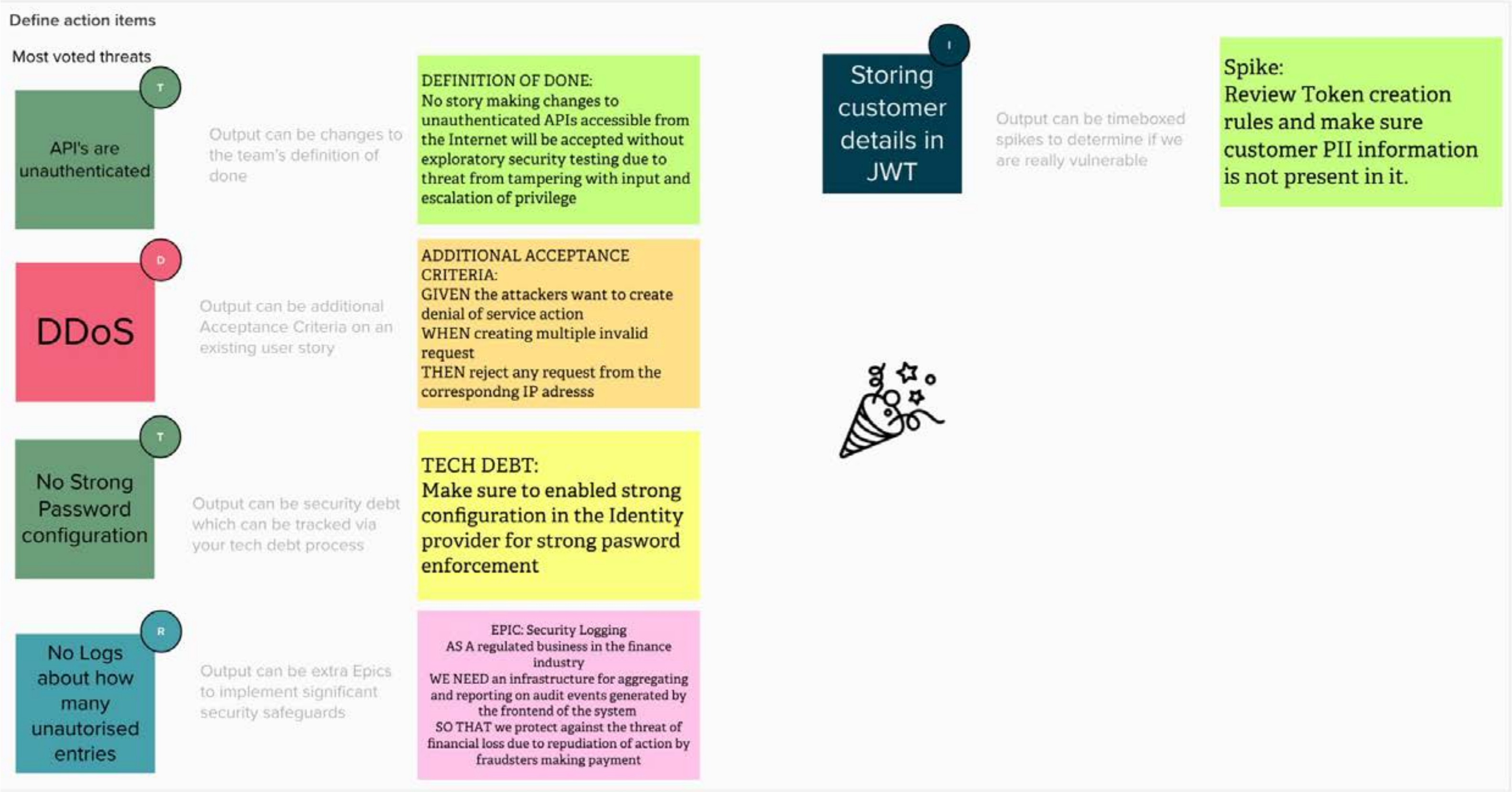
THEN the API request is rejected as unauthorised

Common anti-patterns

- **Capturing the mitigations in non project management platforms like spreadsheets, emails**
- **Prioritizing the mitigation over threat**

Example: Mitigation

Juice shop



Define action items

Most voted threats

T

API's are unauthenticated

Output can be changes to the team's definition of done

DEFINITION OF DONE:
No story making changes to unauthenticated APIs accessible from the Internet will be accepted without exploratory security testing due to threat from tampering with input and escalation of privilege

D

DDoS

Output can be additional Acceptance Criteria on an existing user story

ADDITIONAL ACCEPTANCE CRITERIA:
GIVEN the attackers want to create denial of service action
WHEN creating multiple invalid request
THEN reject any request from the correspondng IP adresss

T

No Strong Password configuration

Output can be security debt which can be tracked via your tech debt process

TECH DEBT:
Make sure to enabled strong configuration in the Identity provider for strong pasword enforcement

R

No Logs about how many unautorised entries

Output can be extra Epics to implement significant security safeguards

EPIC: Security Logging
AS A regulated business in the finance industry
WE NEED an infrastructure for aggregating and reporting on audit events generated by the frontend of the system
SO THAT we protect against the threat of financial loss due to repudiation of action by fraudsters making payment

I

Storing customer details in JWT

Output can be timeboxed spikes to determine if we are really vulnerable

Spike:
Review Token creation rules and make sure customer PII information is not present in it.

Agile Threat Modelling

Steps

1. What do we want to accomplish?
2. What are we building?
3. What can go wrong?
4. What are we going to do about it?
5. Did we do a good job?

Reflect....

Feedback and continuous improvement is central to managing risk.

Stakeholders:

- **Security team (Optional)**
- **Engineers**

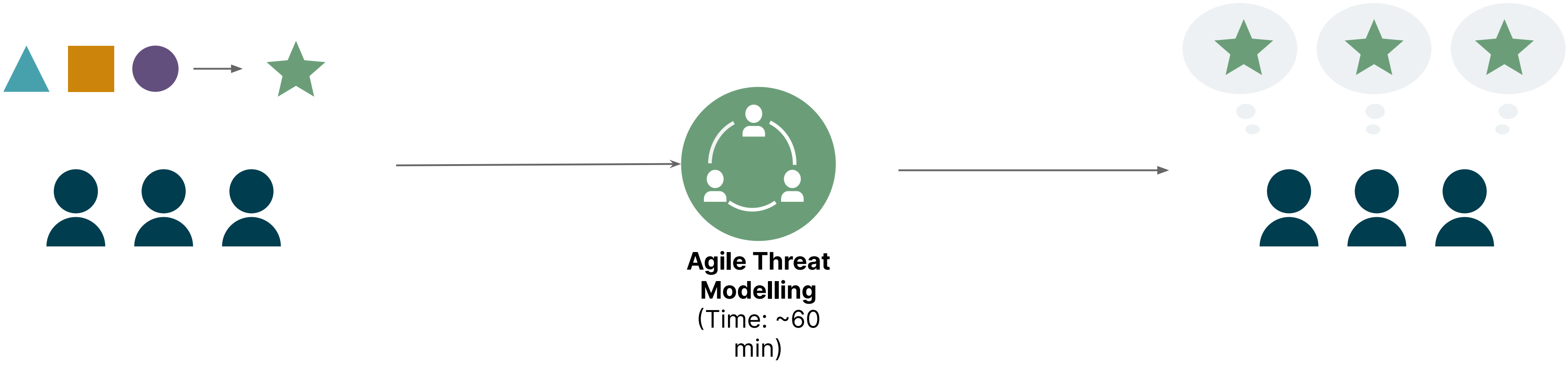
Principal Analysis:

- **Scope**
- **Tools.**
- **Outcome of the exercise**

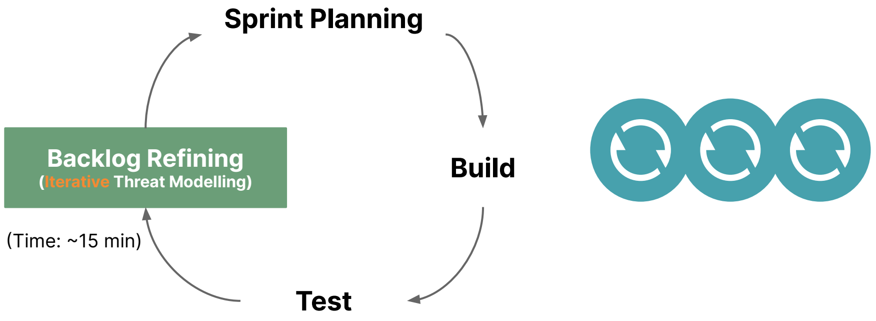
Iterative Threat Modelling

thoughtworks

…. And Repeat



1. What do we want to accomplish?
2. What are we building?
3. What can go wrong?
4. What are we going to do about it?
5. Did we do a good job?



Ways of running the workshop

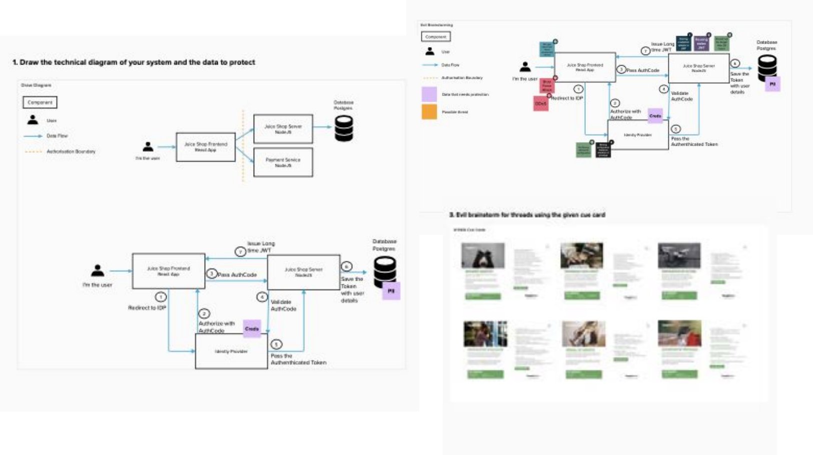
Face to Face

- Print cue cards from this presentation.
- Gather everyone against a white board and draw high level DFD in the board.
- Use stickers and sharpies to capture the threats and mitigation
- Save the artifacts digitally

Be time conscious!!

Hybrid & Remote

- Take a inspiration from the PDF attached and use any white board presentation in your organization to create the flow
- Capture the DFD / architecture diagram in the template.
- Use stickies to capture the threat and mitigation



Learn more

As a security process, threat modeling is subject to several misconceptions. Some people believe threat modeling is only a design-stage activity

threat-modeling

Join 500 other threat modellers on #threat-modeling on OWASP's Slack

r/threatmodeling

All things to do with threat and security modeling - from public examples to talks, tools and techniques

/tw

Security in Thoughtworks

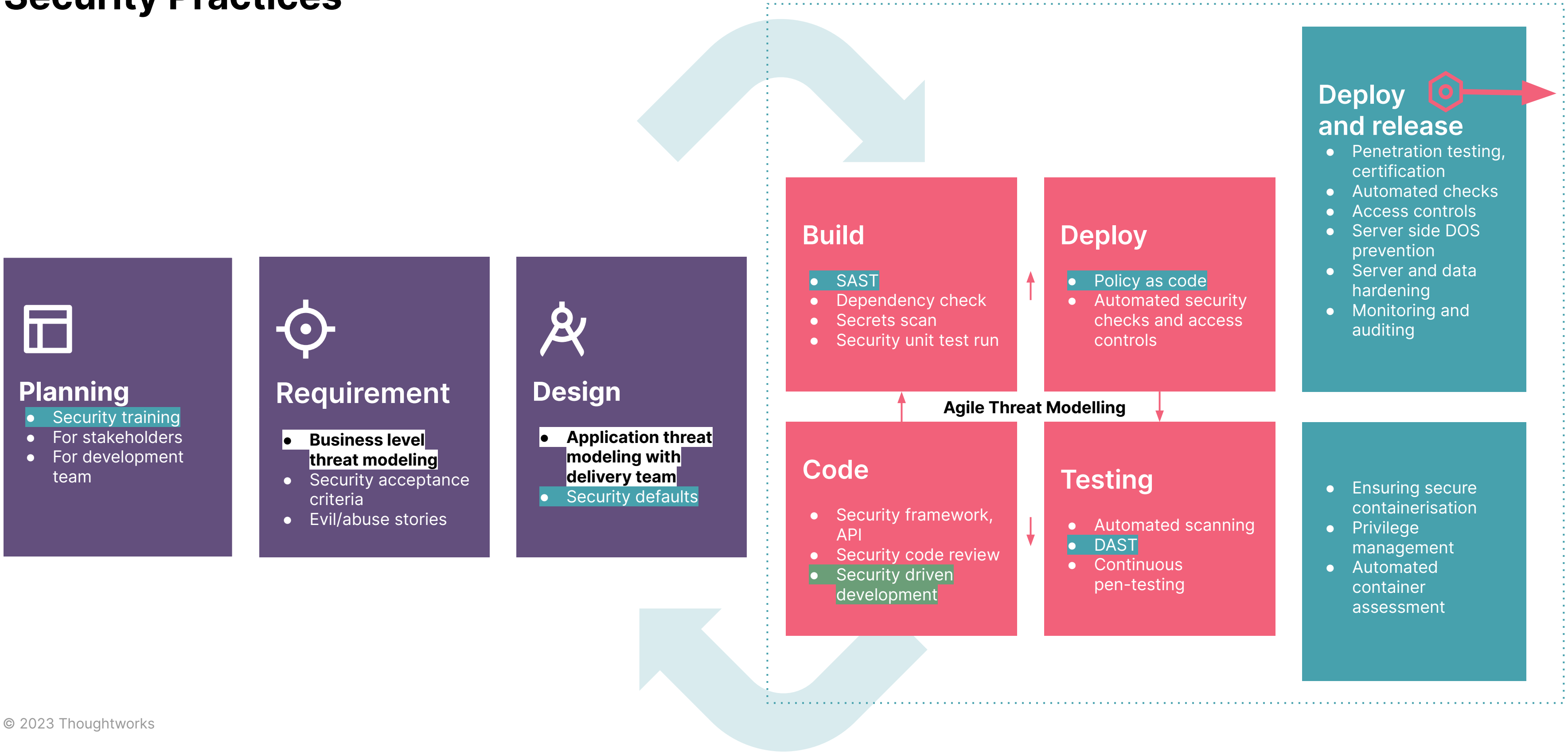
Look out for more Threat modelling and DevSecOps information in Thoughtworks *Security blog post*

Martin Fowler Blog

More about Threat Modelling for developers in the Martin Fowler Blog

Threat Modelling in Software Development Lifecycle

Security Practices



Planning

- Security training
- For stakeholders
- For development team

Requirement

- **Business level threat modeling**
- Security acceptance criteria
- Evil/abuse stories

Design

- **Application threat modeling with delivery team**
- Security defaults

Build

- SAST
- Dependency check
- Secrets scan
- Security unit test run

Deploy

- Policy as code
- Automated security checks and access controls

Agile Threat Modelling

Code

- Security framework, API
- Security code review
- Security driven development

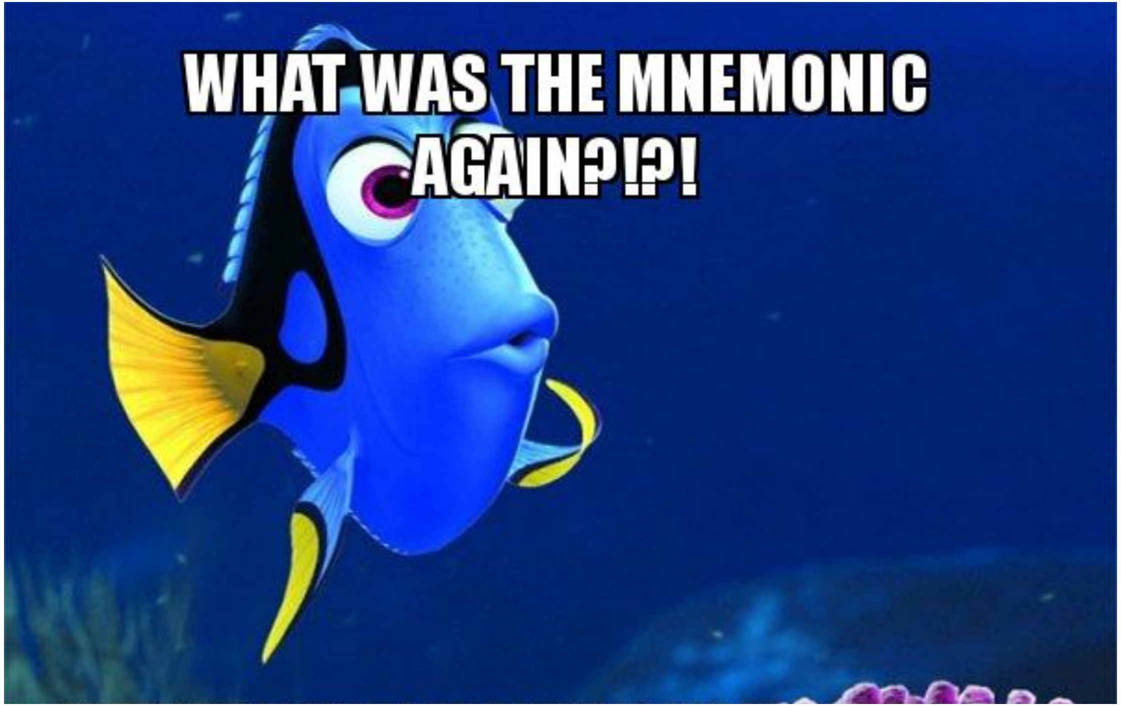
Testing

- Automated scanning
- DAST
- Continuous pen-testing

Deploy and release

- Penetration testing, certification
- Automated checks
- Access controls
- Server side DOS prevention
- Server and data hardening
- Monitoring and auditing

- Ensuring secure containerisation
- Privilege management
- Automated container assessment


WHAT WAS THE MNEMONIC AGAIN?!?!

Takeaways

- You don’t have to be a security engineer or expert to threat model!
- You will identify threats that you’ll never find with automation
- You can do threat modelling at any point in the delivery lifecycle
- Extend your existing ways of working and ask ‘what can go wrong?’
- There are lots of ways, but brainstorming with STRIDE is quick & flexible
- Actions might be stories, tasks, acceptance criteria or definition of done using DREAD
- There’s a whole community out there to support with resources

WHO IS INVOLVED?

ENGINEERS

- Learn security
- Create a deeper understanding
- Guide secure design & testing
- Find threats missed by automation
- Shift security “left”

SECURITY TEAM

- Provide input in collaborative way
- Perspective of threat landscape
- Give context of controls
- Meet compliance needs
  - For example NIST 800-53

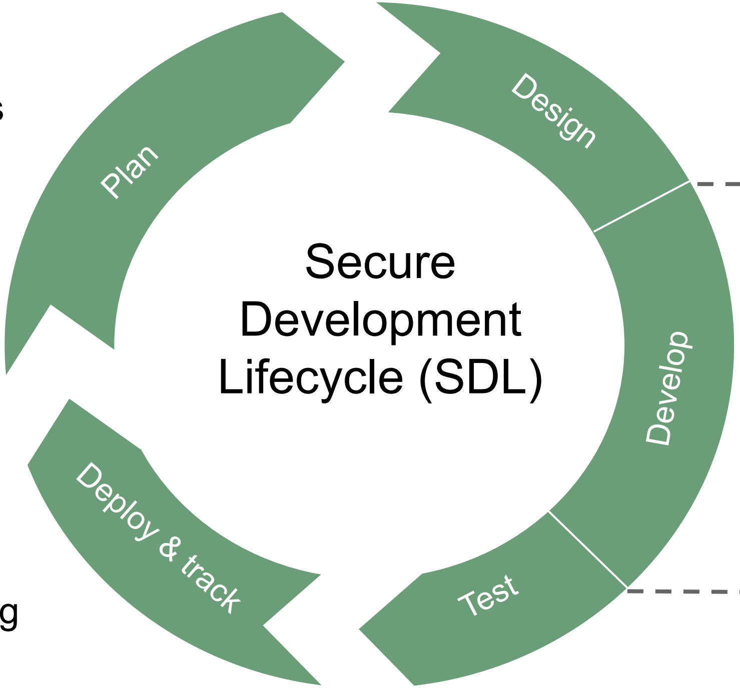
PRODUCT OWNER & BAs

- Prevent bad things from happening
- Save time doing security right
- Create a deeper understanding
- Prioritize according to risk
- Deliver on time

EVERYONE

- Reduce risk
- Greater confidence
- Breaks down silos

Secure Development Lifecycle (SDL)



Training

Security User Stories

Data Flow Diagram

Threat Modelling

Secure Code Analysis

Dependency Analysis

Static Analysis Security Tool (SAST)

Dynamic Analysis Security Testing Tool (DAST)

Policy as Code

Penetration Testing