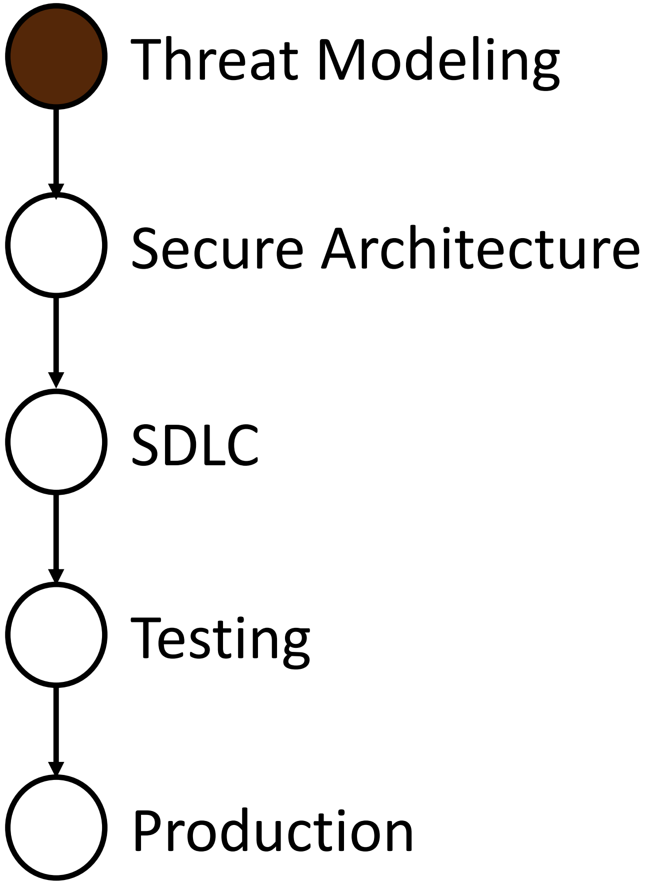
Secure Architecture Process



Threat Modeling

Secure Architecture

SDLC

Testing

Production

Threat Modeling

Be Aware / Recommend

Goal:

Identify potential threats for the system and discuss ways to mitigate them

Participants:

Project Manager
Architect
Dev Manager
System Analyst

CISO
IT (OP)
Developers (OP)
QA (OP)

(OP) = Optional

What is Threat Modeling?

- The process of identifying potential threats for the system
- Done once, but might be repeated later
- Should be very methodical
- Everyone’s input is welcome

What is Threat Modeling?

- Based on 4 core questions:

What do we build?

What can go wrong?

How can we mitigate that?

Did we succeed?

What Do We Build

- Describe functional and non-functional requirements
- If there are any known technical or architectural details – include them

What Do We Build

Example:

“We’re designing an HR system to manage the employees’ data, including salary, vacations, etc.”

What Can Go Wrong

- Describe what are the main threats the application might face, based on:
  - Sensitivity of the information the system stores
  - Its location
  - Competition
  - Any other factor that might be relevant…

What Can Go Wrong

Example:

> *“Since we store sensitive data in the system, we want to make sure it won’t leak”*

How Can We Mitigate That

- Discuss mitigations to the potential threats
- Research various mitigations methods if needed
- Make sure to include the mitigations in the work plan
- If the dev team does not know how to implement it – design a training plan

How Can We Mitigate That

Example:

“All the sensitive data is going to be encrypted. In addition, database access will be given on a least-privilege basis only”

Did We Succeed

- Design tests to validate the solution designed
- Usually will be carried out in the testing phase, but

  sometimes also during development

Did We Succeed

Example:

“We’re going to ask a security expert to extract and decrypt the encrypted data and see if she succeeds.”

Threat Modeling Example #1

What do we build?

HR system to manage the employees' data, including salary, vacations, etc.

What can go wrong?

Since we store sensitive data in the system, we want to make sure it won't leak

How can we mitigate that?

All the sensitive data is going to be encrypted. In addition, database access will be given on a least-privilege basis only

Did we succeed?

We're going to ask a security expert to extract and decrypt the encrypted data and see if she succeeds

Threat Modeling Example #2

What do we build?

Mobile game for kids, where they can learn English

What can go wrong?

Since it's a world-wide system, with a lot of competition, we're afraid of DDoS attacks

How can we mitigate that?

Using state-of-the-art firewalls and app gateways, and having DR sites for when the app is down

Did we succeed?

We're going to conduct massive, geo-distributed load tests, simulating some previously-executed DDoS attacks, and see what happen

Result of Threat Modeling

- Threat Modeling document
- Documents all the steps taken in the process
- Mainly documents the last 3 questions
    - We already know what we build…
- Project manager decides who creates the document
    - Usually not the Architect
- Sometimes might be loaded to a dedicated tool

Conducting the Threat Modeling

- We talked about the What…

- Let’s talk about the When and How

When

At the beginning of the project

(When there are functional and non-functional requirements)

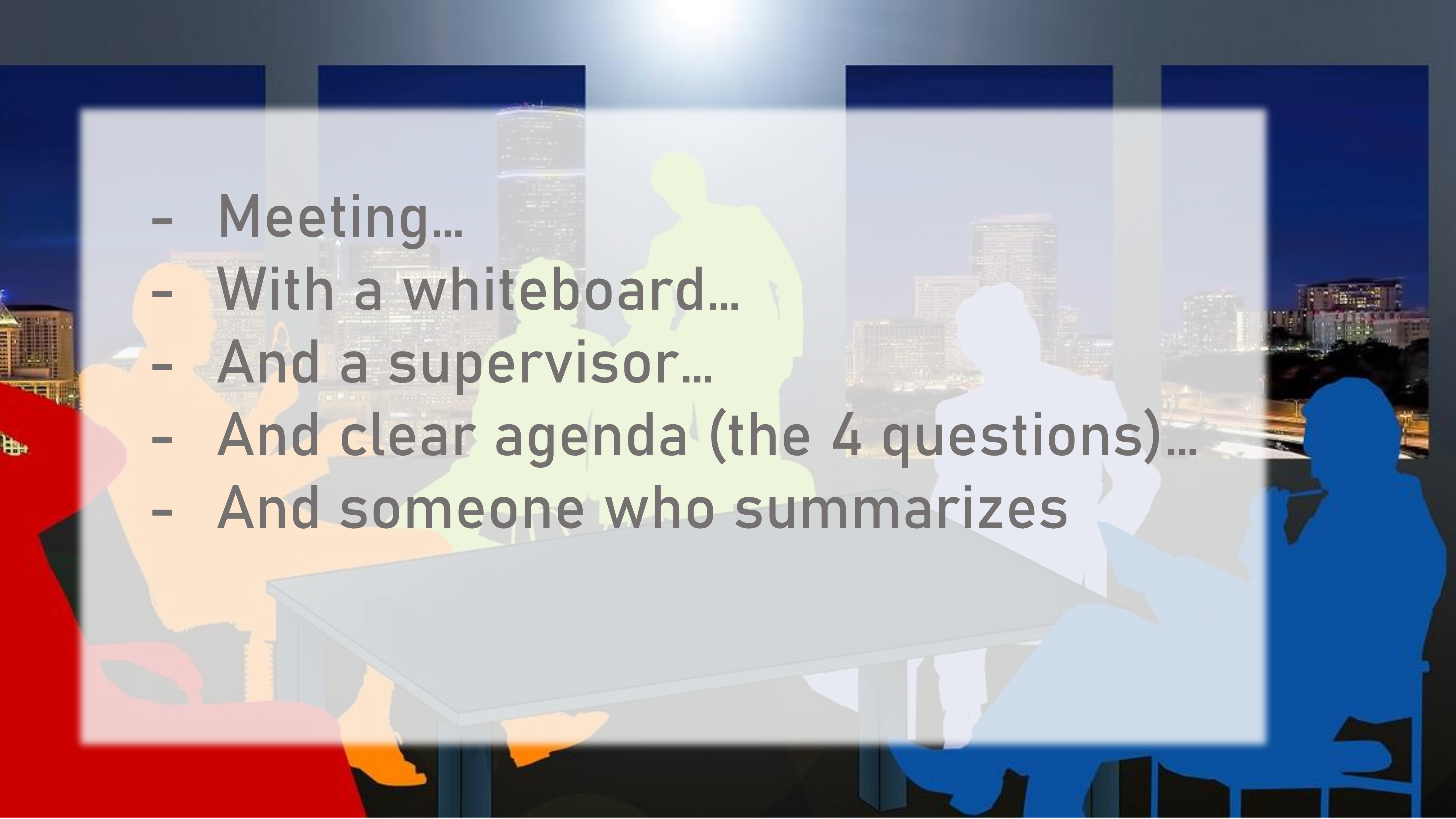
After major changes

(That might present new security risks)

Following security incident

(To find out what went wrong)

How



- Meeting…
- With a whiteboard…
- And a supervisor…
- And clear agenda (the 4 questions)…
- And someone who summarizes

Threat Modeling Methodologies

- Usually Threat Modeling is done using free-form discussions
- Some methodologies exist that formalize the process
- Most organizations don’t use them
- We’re going to have a quick look, not a comprehensive overview

STRIDE

- The most mature methodology
- Developed in 1999
- Adopted by Microsoft in 2002

STRIDE

- Stands for:
  - Spoofing
  - Tampering
  - Repudiation
  - Information Disclosure
  - Denial of Service
  - Elevation of Privilege

Each one represents a potential threat to the system

STRIDE

- In the modeling process the 6 threats are discussed against flow diagrams of the system
- When a potential threat is found, a mitigation plan is formed
- Used by some groups in Microsoft and Cyber-Security companies

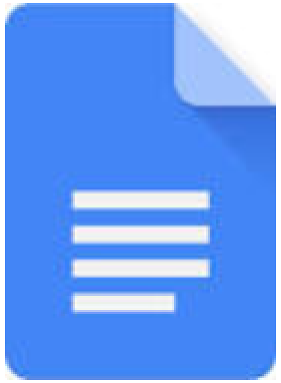
Other Methodologies

- PASTA
- DREAD
- Attack Tree
- CVSS
- And more…

None is widely used

Threat Modeling Tools

- Threat Modeling usually utilizes:

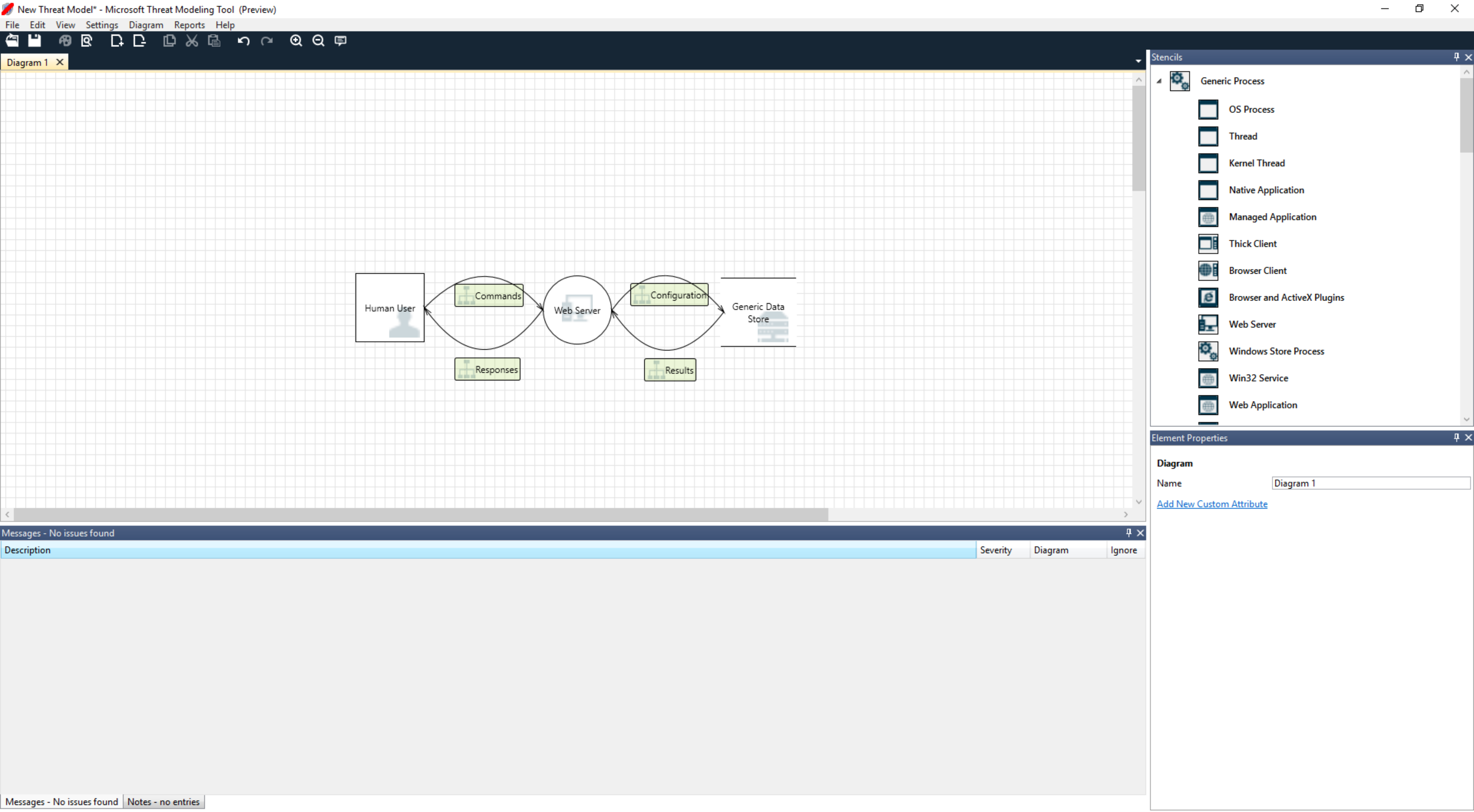


Threat Modeling Tools

- Some tools exist to help with the process
- None is widely used
- We’re going to take a quick look at two tools

Microsoft Threat Modeling Tool (TMT)

- Used for a complete Threat Modeling process
- Contains visual designer to build Data Flow Diagrams (DFD)
- Uses the STRIDE methodology
- Designed for developers
- Stand-Alone app, requires installation



Messages - No issues found

Description	Severity	Diagram	Ignore

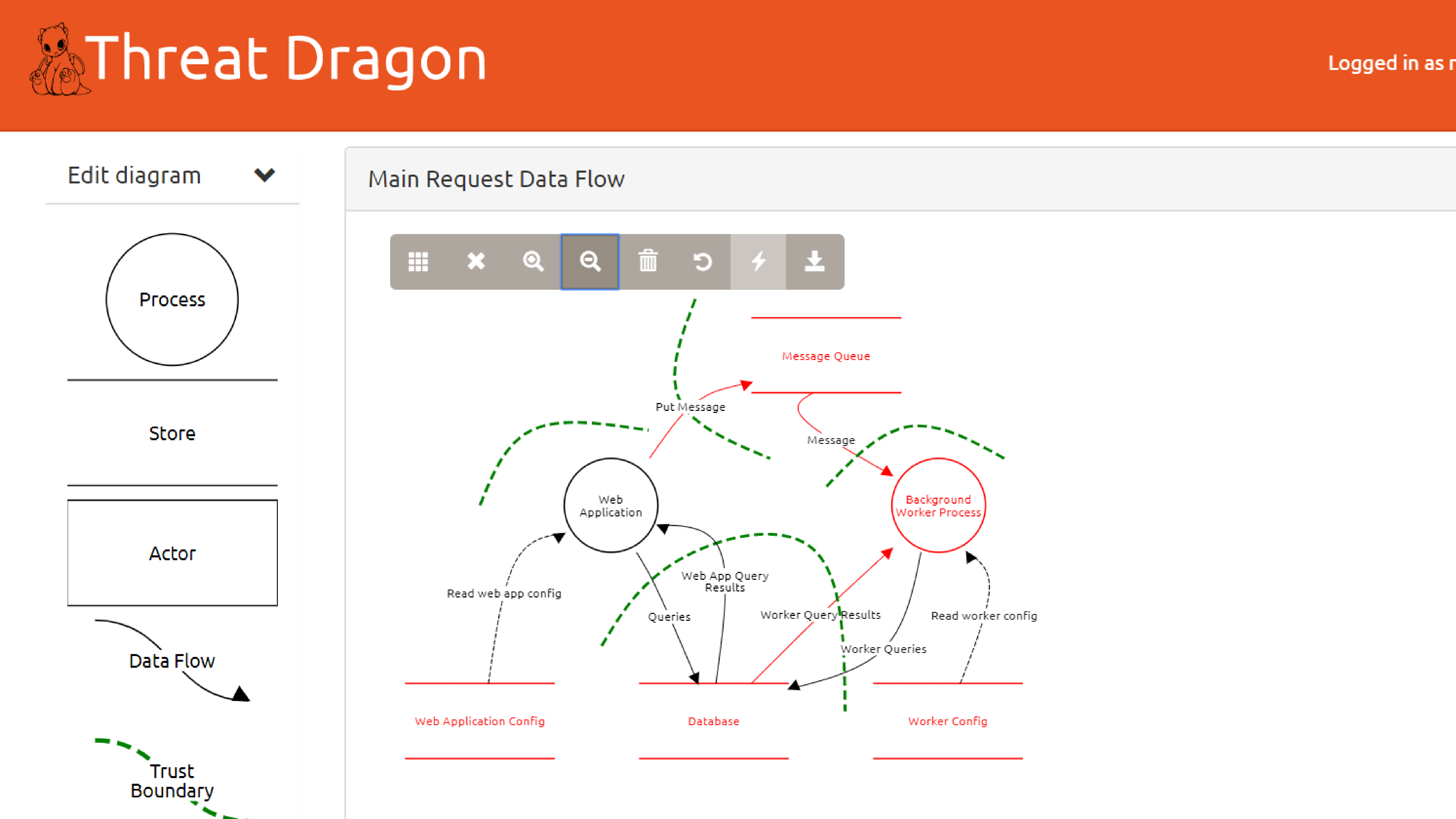
Diagram

Name: Diagram 1

Messages - No issues found | Notes - no entries

Threat Dragon

- New app, still in early stages
- Developed by OWASP
- Electron-based, can be installed locally or used as a web app
- Has visual designer for DFDs
- Based on the STRIDE methodology


Threat Dragon
Logged in as r
Edit diagram
Main Request Data Flow
Process
Store
Actor
Data Flow
Trust Boundary
Message Queue
Put Message
Message
Web Application
Background Worker Process
Read web app config
Web App Query Results
Queries
Worker Query Results
Read worker config
Worker Queries
Web Application Config
Database
Worker Config

Threat Modeling Summary

- A process for identifying potential threats for the system
- Should be conducted early in the project lifecycle
- Based on 4 core questions
- Involves almost everyone in the team
- Might utilize formal methodologies and tools