Threat Modeling

HLMC Day Application Security 2016

Dominik Schadow | bridgingIT

Threat

A source of damage or danger

Anything that can act against an asset resulting in a potential loss

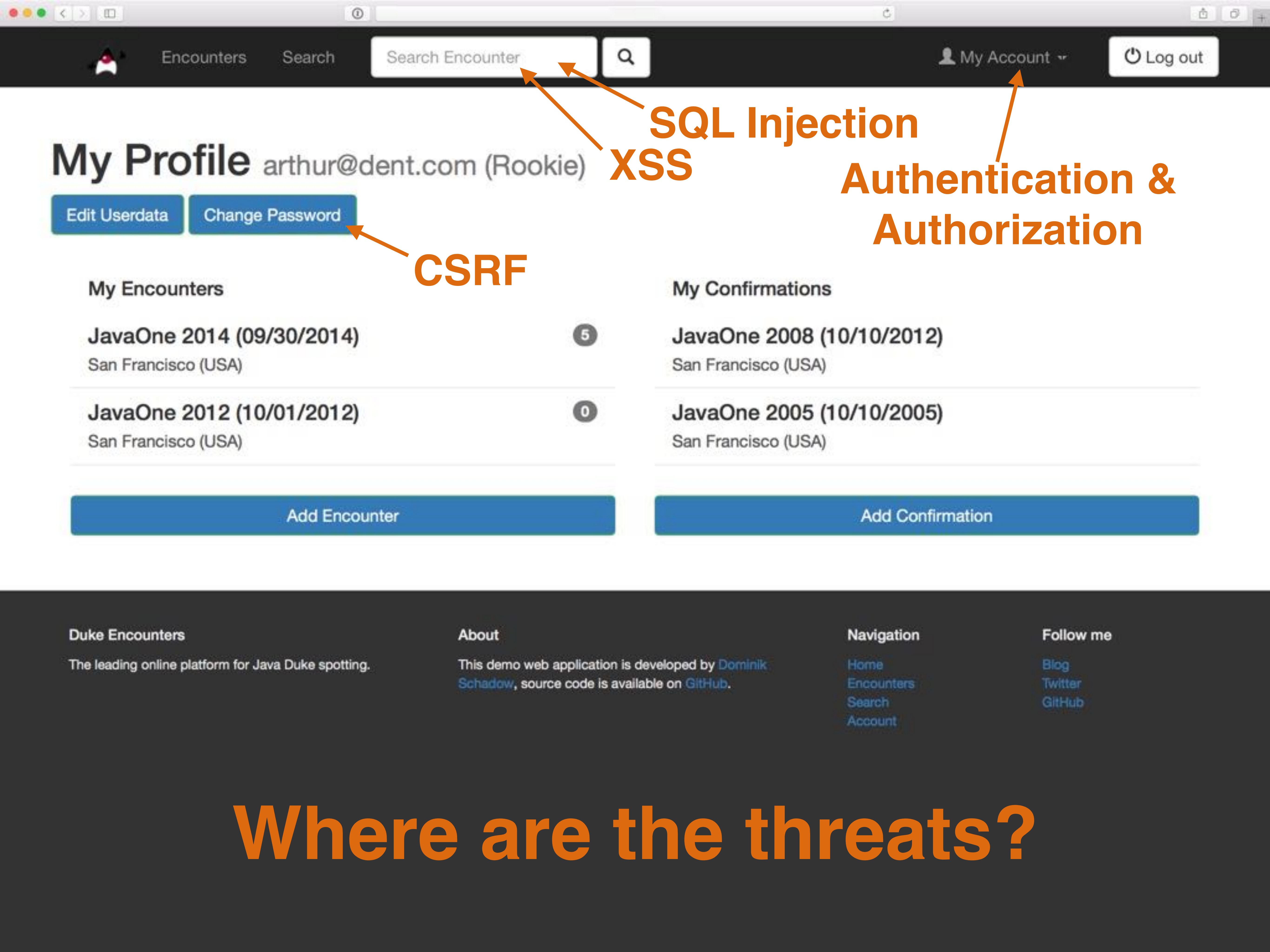
Where are the threats?

spring boot

Apache Tomcat

Thymeleaf

MySQL®



SQL Injection

XSS

Authentication & Authorization

CSRF

My Profile arthur@dent.com (Rookie)

My Encounters

JavaOne 2014 (09/30/2014) 5
San Francisco (USA)

JavaOne 2012 (10/01/2012) 0
San Francisco (USA)

My Confirmations

JavaOne 2008 (10/10/2012)
San Francisco (USA)

JavaOne 2005 (10/10/2005)
San Francisco (USA)

Duke Encounters

The leading online platform for Java Duke spotting.

About

This demo web application is developed by Dominik Schadow, source code is available on GitHub.

Navigation

Home
Encounters
Search
Account

Follow me

Blog
Twitter
GitHub

Where are the threats?

Both variants are **too late**

Agenda

Threat Modeling **Basics**

Identifying Threats in Applications

Threat Modeling in **Action**

Threat Modeling Basics

Threat Modeling

Analyze security incidents and scenarios

Used by IT (security) professionals

And developers?

Security flaws exist before code

Know and reduce attack surface with threat modeling

Forget to authenticate a user

Incomplete central user management system usage

Broken authorization

Missing auditing functionality

Think about…

Who might attack your system?

What is their goal?

Which vulnerabilities might they exploit?

Different ways to threat model

There is no single perfect way

Focus on attackers: Can developers really think like an attacker?

Focus on assets: Did the client name the assets that (may) need protection? How do you link assets to threats?

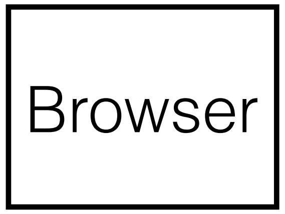
Follow the data

Threats tend to follow the data flow

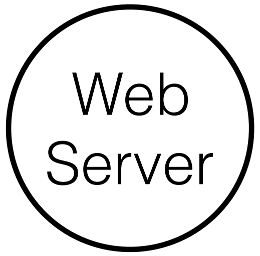
Start with external entities and follow the data flow through your application in a structured way and identify the real problems

Data Flow Diagrams

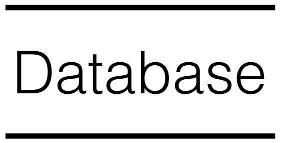
External Entity People or code outside your control that interact with the application



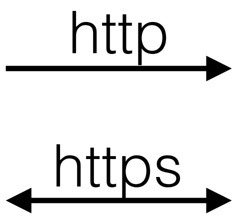
Process Code and components that handle data and the dev team controls



Data Store Anything that stores data and does not modify it

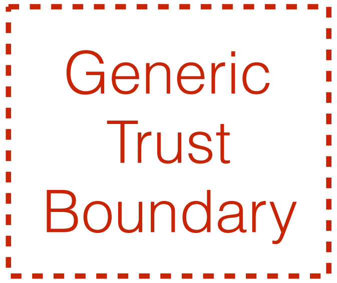


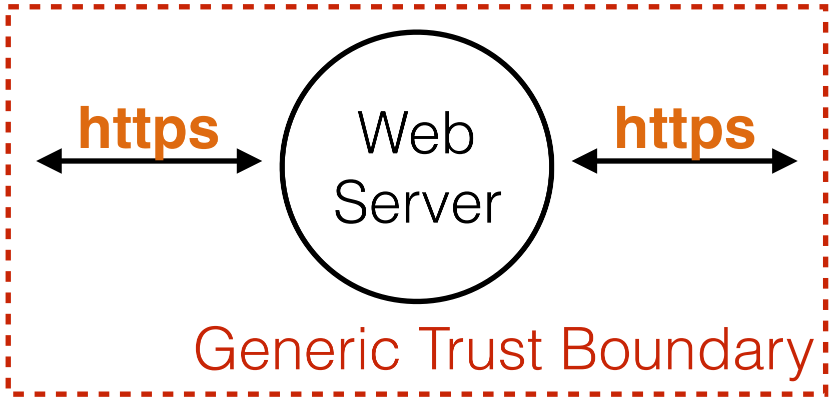
Data Flow Directed data movement within the application

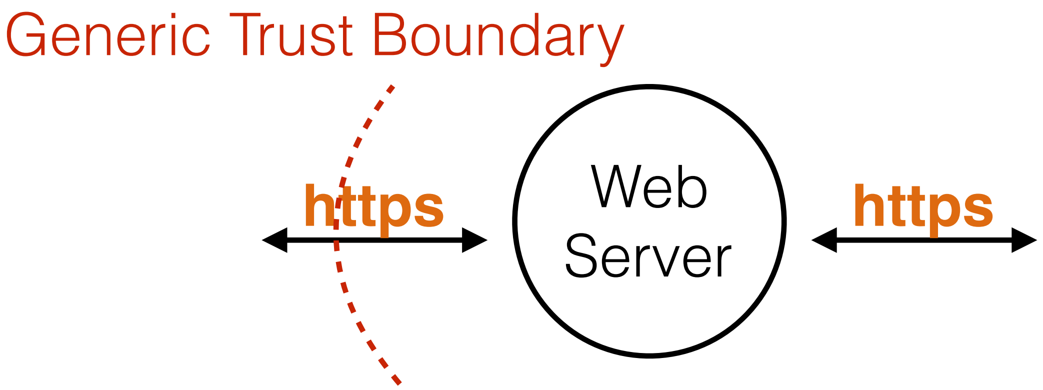


Trust Boundaries

Trust Boundary Change of privilege or trust levels as the data flows through the application





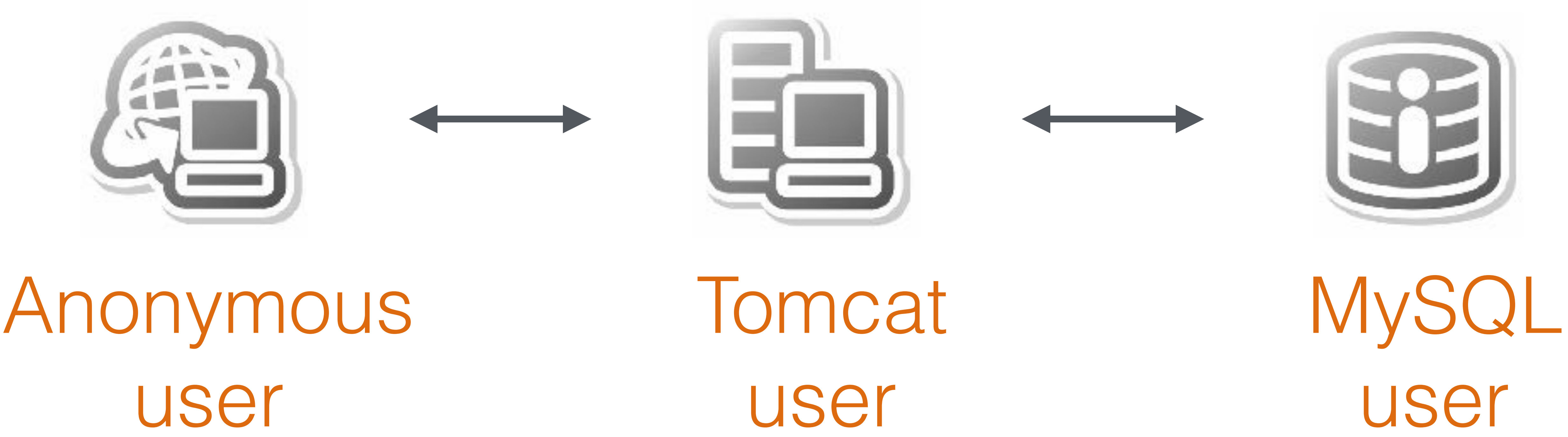


Typical boundaries

Can be technical or organizational

Typical boundary locations

Follow the data, add boundary for new principal



Anonymous user

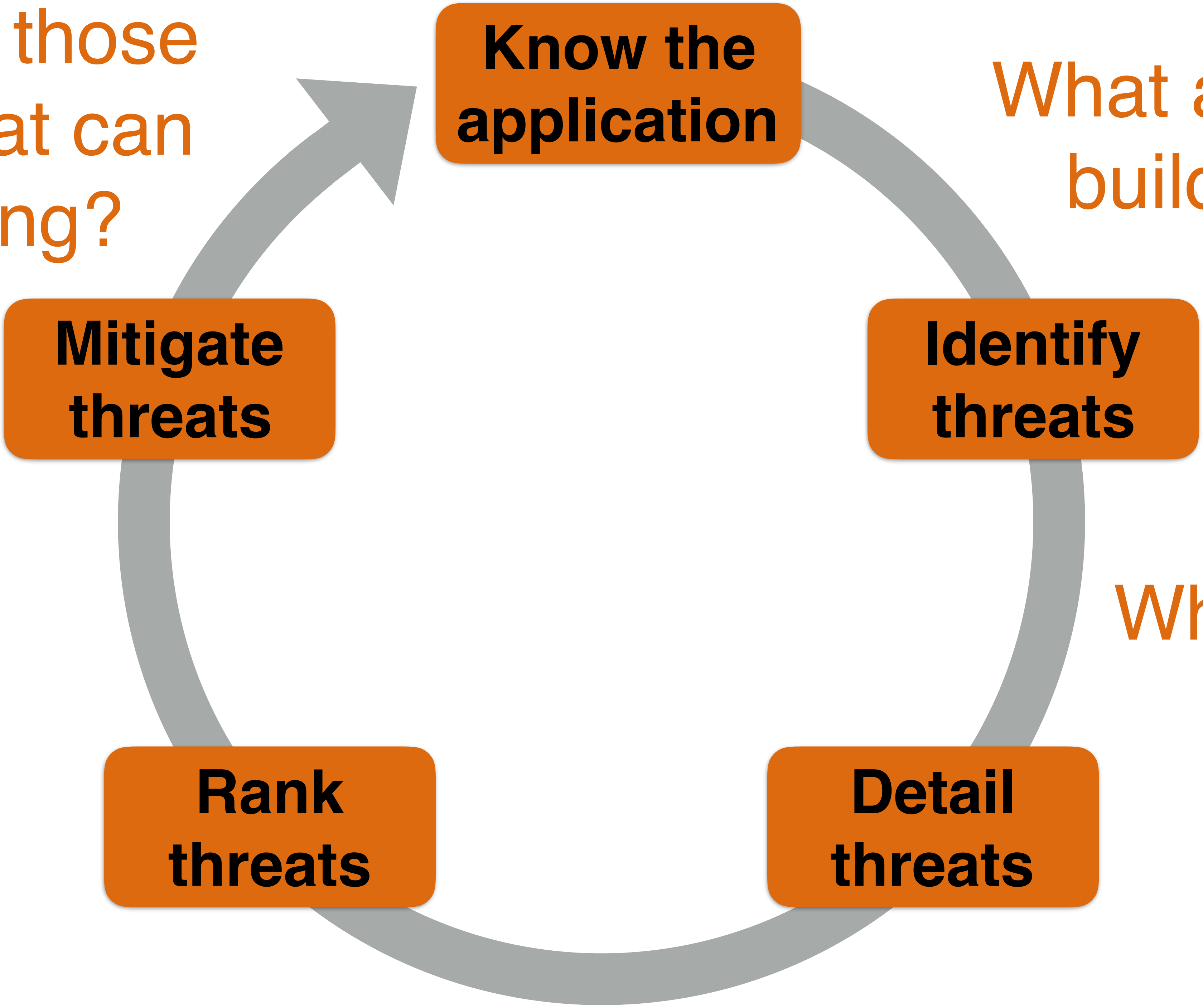
Tomcat user

MySQL user

Identifying Threats in Applications

Identifying threats in applications

What should you do about those things that can go wrong?



What are you building?

What can go wrong?

What are you building?

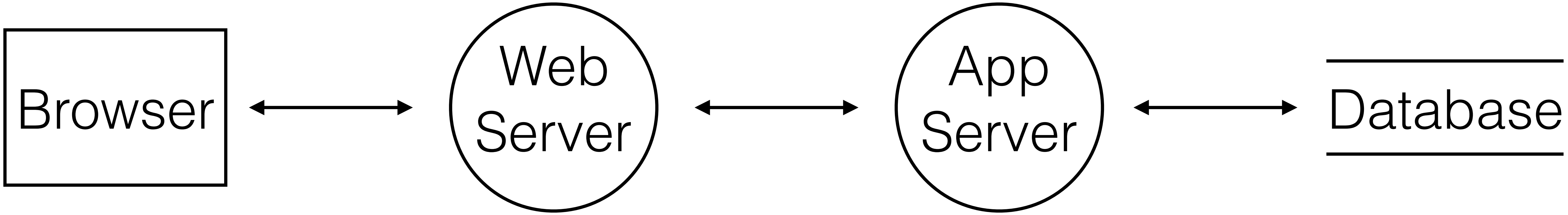
Focus on data flow

„*Sometimes…*“ : indicates alternatives, model them all

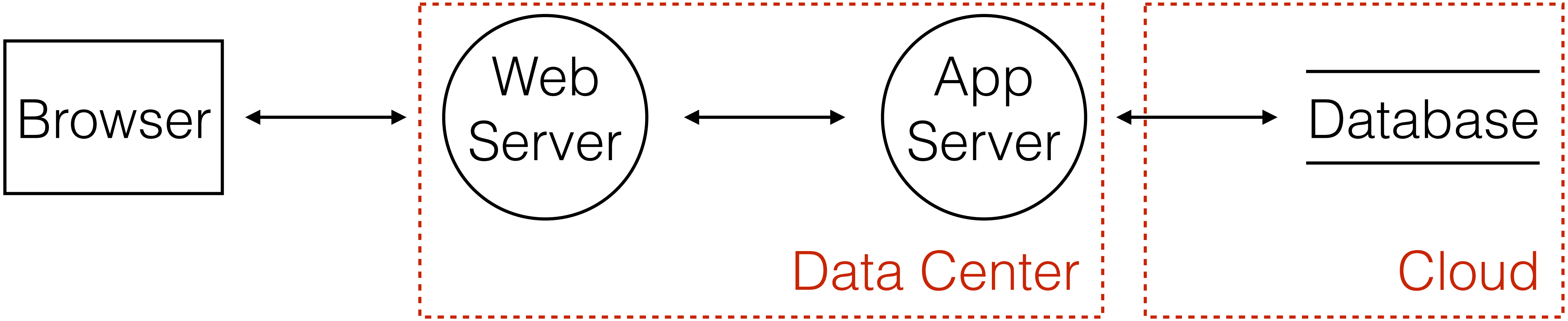
No data sinks: show the consumers

Data does not move by itself: draw the process moving it

Follow the data



Add trust boundaries



What can go wrong?

Start with data crossing trust boundaries

Brainstorm meetings with technology experts
Play the Elevation of Privilege game
Use STRIDE

STRIDE

STRIDE is the opposite of a property you want

Spoofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, **E**levation of Privilege

STRIDE

Spoofing	Pretending to be something or somebody else **Violated property:** Authentication **Standard defenses:** Passwords, multi-factor authentication
Tampering	Modifying something on disk, network or memory **Violated property:** Integrity **Standard defenses:** Digital signatures, hashes

STRIDE

Repudiation

Claiming that someone didn't do something
Violated property: Non-Repudiation
Standard defenses: Logging, auditing, timestamps

Information Disclosure

Providing information to someone not authorized
Violated property: Confidentiality
Standard defenses: Encryption, authorization

STRIDE

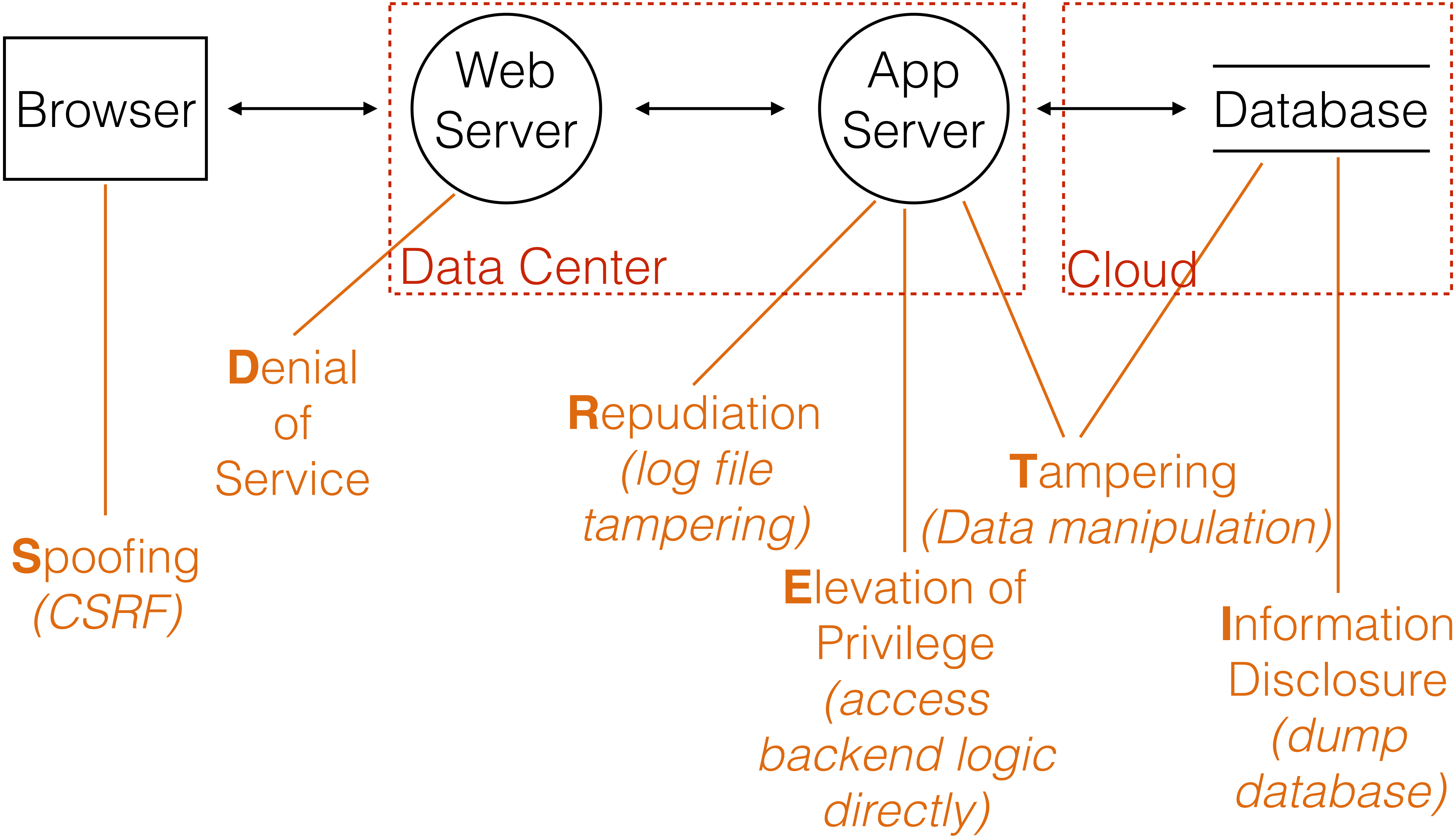
Denial of Service

Absorbing resources needed to provide service
Violated property: Availability
Standard defenses: Filtering, quotas

Elevation of Privilege

Doing something someone is not authorized to do
Violated property: Authorization
Standard defenses: Input validation, least privilege

Add threats



Browser

Web Server

App Server

Database

Data Center

Cloud

Spoofing *(CSRF)*

Denial of Service

Repudiation *(log file tampering)*

Elevation of Privilege *(access backend logic directly)*

Tampering *(Data manipulation)*

Information Disclosure *(dump database)*

Address each threat

Decide for each threat how to handle it

Mitigate

Eliminate

Transfer

Accept

Mitigate it

Preferred (and most common) solution

Reducing the attack surface to make it harder to take advantage of a threat (like introducing a password policy)

Eliminate it

Most secure solution

Results in feature elimination most of the time (like removing admin functionality from the Internet facing application)

Transfer it

Team solution

Someone/ something else handles the risk, depending who can easily fix the problem (like operations adding a web application firewall)

Accept it

Last resort solution

Stop worrying about it and live with the risk (like someone stealing your servers' hard disk)

Threat Target	Mitigation Strategy	Mitigation Technique	Prio	ID
Repudiating actions	Log	Logging all security relevant actions in an audit log	2	1001
Spoofing a user	Identification and authentication	Password policy, token, password reset process	1	1002
Network flooding	Elastic cloud	Dynamic cloud resources to provide service	3	1006
Tampering network packets	Cryptography	HTTPS/TLS	1	1007

Is it complete?

Let a developer introduce the application by following the data flow

Watch out for phrases like „*Sometimes we have to do … instead of … here*“ or „*A lot of things are happening here which are not completely listed…*“

Breadth before depth

Criteria exist to show you are NOT done, but none to show you are done

Easy

One threat of each STRIDE type

Harder

One threat per diagram element

Threat Modeling in Action

Name a security champion

A developer who knows and drives security

Should know more than security basics and challenge existing threat models and mitigations from time to time

Create the first threat model

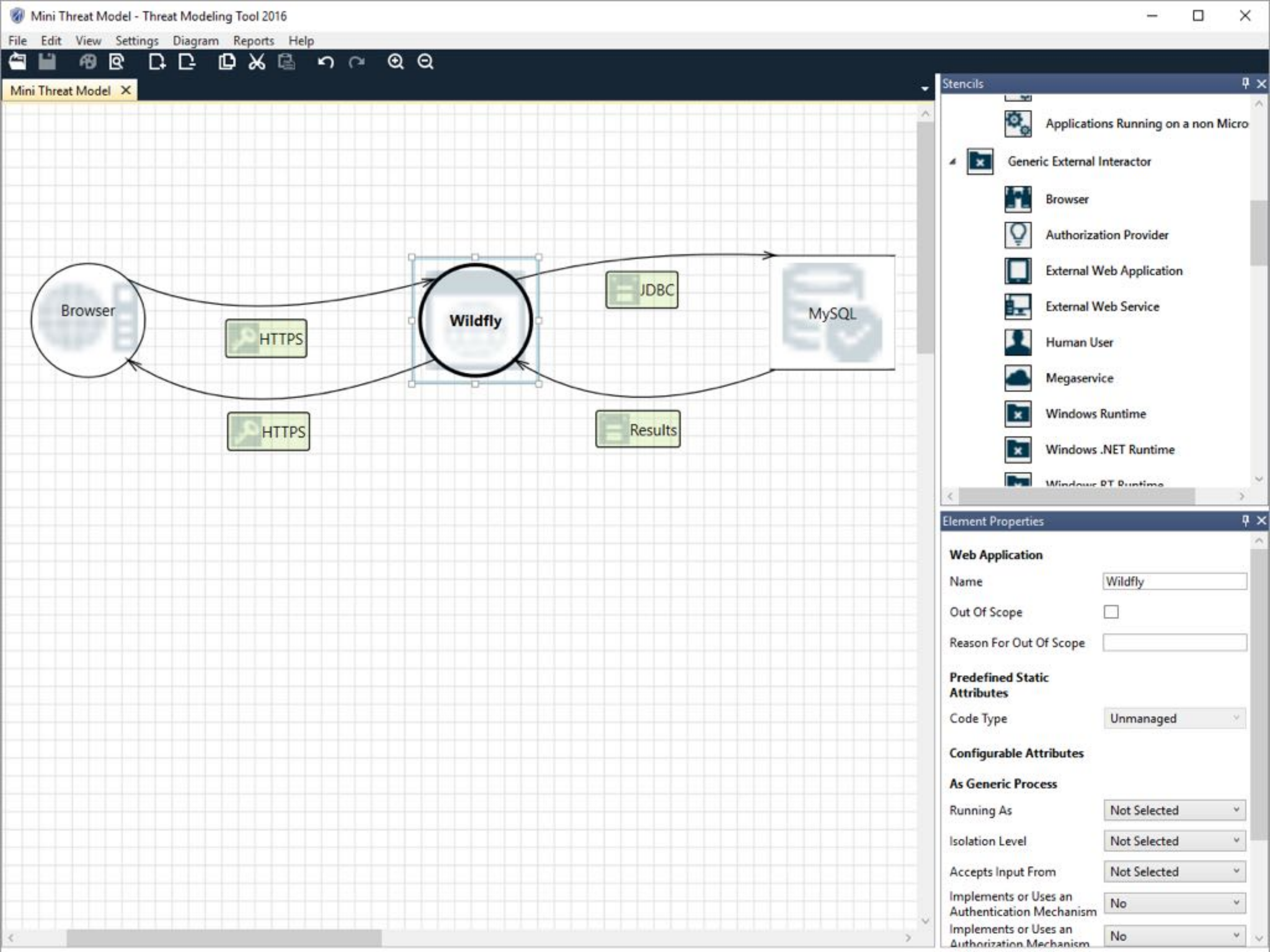
Will require some time, even for small applications

Let an architect and a developer create the initial data flow diagram and introduce it to the team afterwards

Discuss the threats

Use Microsoft Threat Modeling tool to get started

First take care of all recommended „Elevation of Privilege“ threats and make sure to involve the product owner into any threat mitigation discussions



Web Application

Name: Wildfly

Out Of Scope: □

Reason For Out Of Scope:

Predefined Static Attributes

Code Type: Unmanaged

Configurable Attributes

As Generic Process

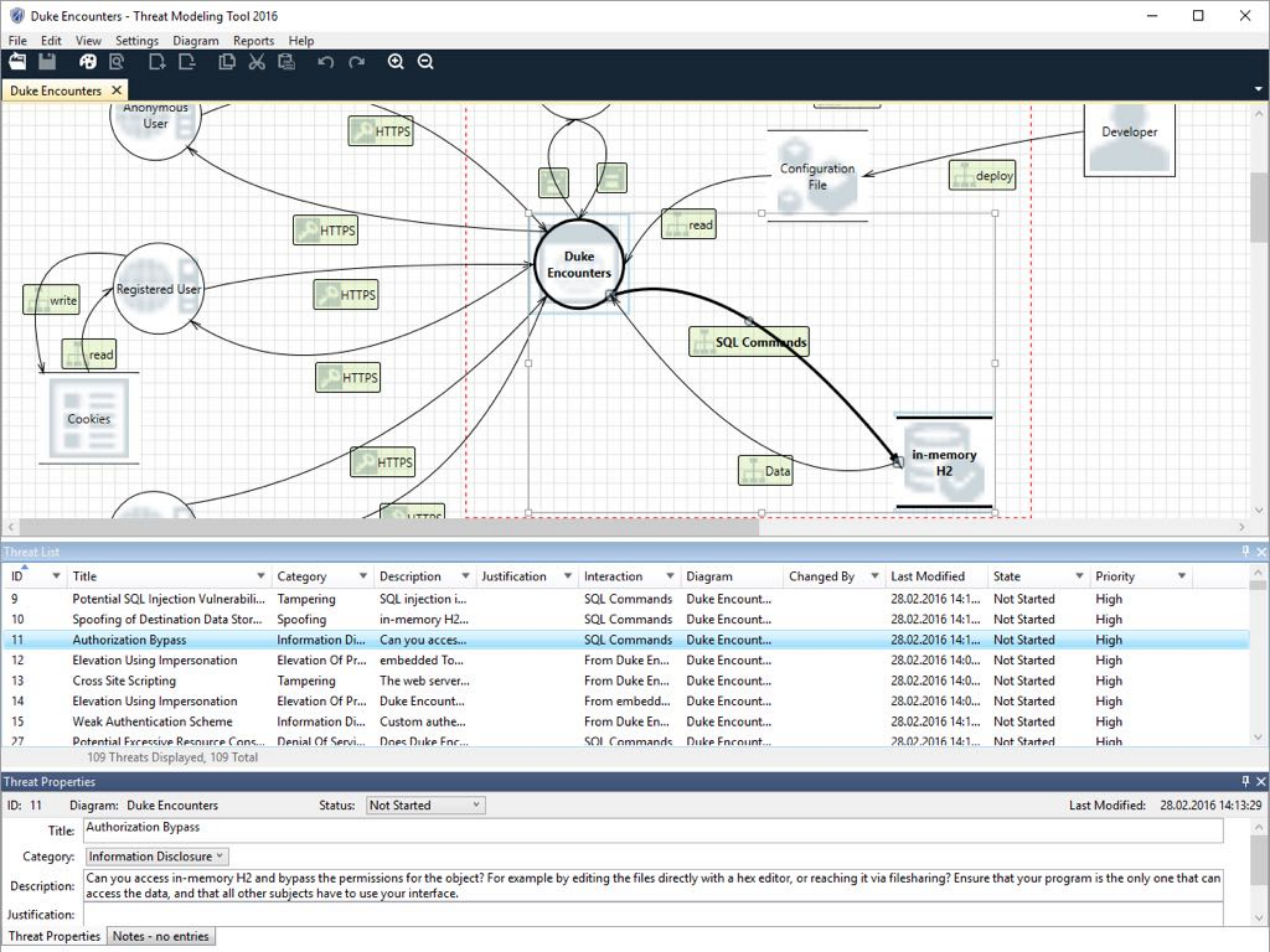
Running As: Not Selected

Isolation Level: Not Selected

Accepts Input From: Not Selected

Implements or Uses an Authentication Mechanism: No

Implements or Uses an Authorization Mechanism: No



ID	Title	Category	Description	Justification	Interaction	Diagram	Changed By	Last Modified	State	Priority
9	Potential SQL Injection Vulnerabili...	Tampering	SQL injection i...		SQL Commands	Duke Encount...		28.02.2016 14:1...	Not Started	High
10	Spoofing of Destination Data Stor...	Spoofing	in-memory H2...		SQL Commands	Duke Encount...		28.02.2016 14:1...	Not Started	High
11	Authorization Bypass	Information Di...	Can you acces...		SQL Commands	Duke Encount...		28.02.2016 14:1...	Not Started	High
12	Elevation Using Impersonation	Elevation Of Pr...	embedded To...		From Duke En...	Duke Encount...		28.02.2016 14:0...	Not Started	High
13	Cross Site Scripting	Tampering	The web server...		From Duke En...	Duke Encount...		28.02.2016 14:0...	Not Started	High
14	Elevation Using Impersonation	Elevation Of Pr...	Duke Encount...		From embedd...	Duke Encount...		28.02.2016 14:0...	Not Started	High
15	Weak Authentication Scheme	Information Di...	Custom authe...		From Duke En...	Duke Encount...		28.02.2016 14:1...	Not Started	High
27	Potential Excessive Resource Cons...	Denial Of Servi...	Does Duke Enc...		SQL Commands	Duke Encount...		28.02.2016 14:1...	Not Started	High

109 Threats Displayed, 109 Total

Threat Properties

ID: 11 Diagram: Duke Encounters Status: Not Started Last Modified: 28.02.2016 14:13:29

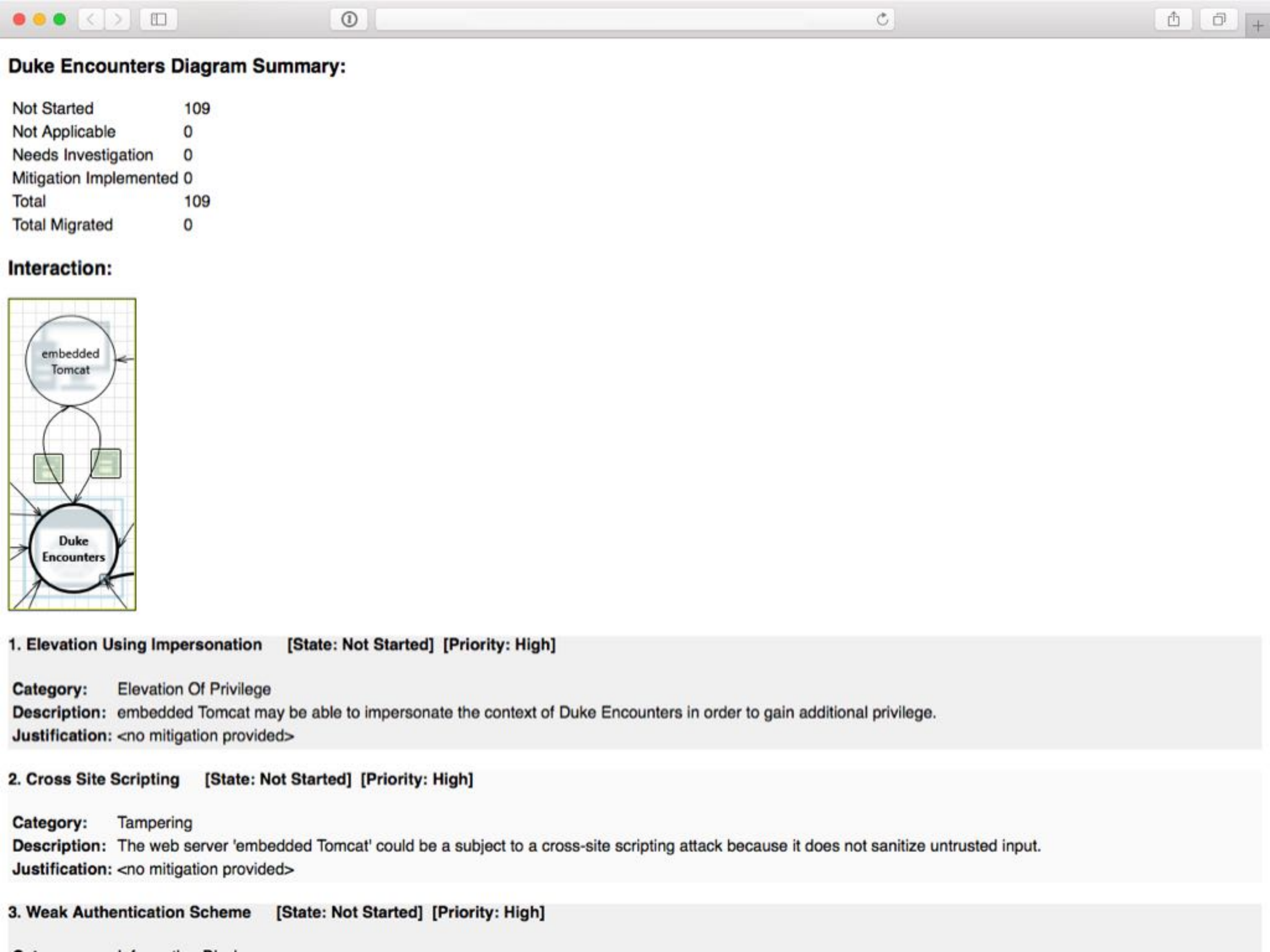
Title: Authorization Bypass

Category: Information Disclosure

Description: Can you access in-memory H2 and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.

Justification:

Threat Properties | Notes - no entries



Duke Encounters Diagram Summary:

Not Started	109
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	109
Total Migrated	0

Interaction:

1. Elevation Using Impersonation [State: Not Started] [Priority: High]

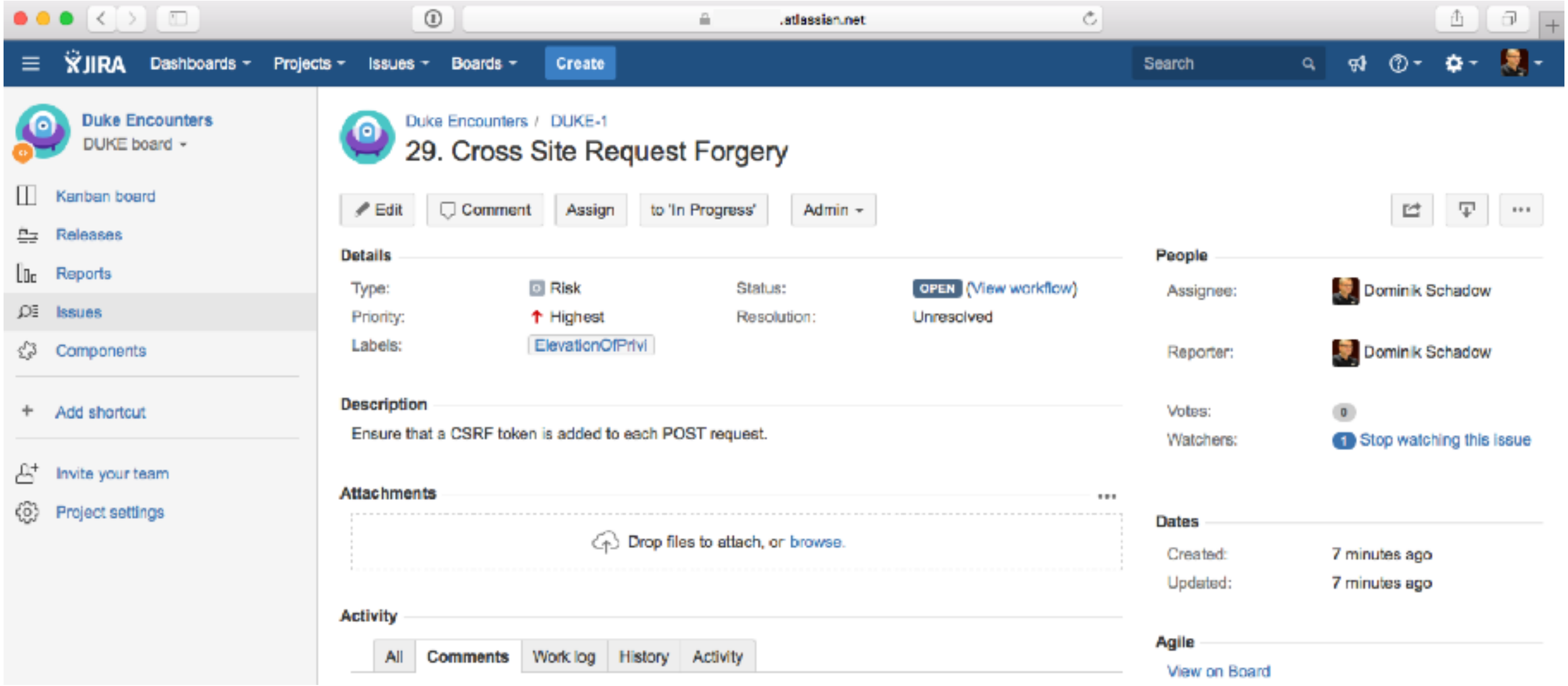
Category: Elevation Of Privilege
Description: embedded Tomcat may be able to impersonate the context of Duke Encounters in order to gain additional privilege.
Justification: <no mitigation provided>

2. Cross Site Scripting [State: Not Started] [Priority: High]

Category: Tampering
Description: The web server 'embedded Tomcat' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification: <no mitigation provided>

3. Weak Authentication Scheme [State: Not Started] [Priority: High]

Add all risks to bug tracking



Duke Encounters / DUKE-1

29. Cross Site Request Forgery

Details

Type:	Risk	Status:	OPEN (View workflow)
Priority:	Highest	Resolution:	Unresolved
Labels:	ElevationOfPrivl		

Description

Ensure that a CSRF token is added to each POST request.

Attachments

Drop files to attach, or browse.

Activity

People

Assignee:	Dominik Schadow
Reporter:	Dominik Schadow
Votes:	0
Watchers:	1 Stop watching this issue

Dates

Created:	7 minutes ago
Updated:	7 minutes ago

Agile

View on Board

Version every model

A threat model is a living document

After the initial version, discuss and update your threat models in every sprint (at least once a month)

Threat modeling has to feel as normal as creating a UML diagram

Summary

Threat model early

Threat model often

Document and address every threat

bridging IT

Marienstraße 17
70178 Stuttgart

dominik.schadow@bridging-it.de
www.bridging-it.de

Application Threat Modeling
www.owasp.org/index.php/Application_Threat_Modeling

Microsoft Threat Modeling Tool
www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx

SecDevOps Risk Workflow
leanpub.com/secdevops

Threat Modeling: Designing for Security (Adam Shostack)
eu.wiley.com/WileyCDA/WileyTitle/productCd-1118809998.html

Pictures
www.dreamstime.com

Blog blog.dominikschadow.de | Twitter @dschadow