Threat Modeling

JavaLand 2016

Dominik Schadow | bridgingIT

Java Web Application

Java 8
Spring Boot 1.3 (Spring 4.2, Spring Security 4)
Thymeleaf 2.1
Tomcat 8
MySQL 5 database (users and application data)

localhost
Encounters
Search
Search Encounter
My Account
Log out
Edit Userdata
Change Password
Add Encounter
Add Confirmation

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?

We developers tend to focus on programming errors and ignore the underlying flaws.

Agenda

Threat Modeling **Basics**

Identifying Threats in Applications

Threat Modeling in **Action**

Threat Modeling Basics

Security flaws are introduced early in the development lifecycle, with no code developed yet

- Threat modeling is all about finding security problems
- Threat modeling starts early

Different ways to threat model

Which one is working out for you?

Focus on attackers: Can you really think like an attacker?
Focus on assets: What is an asset in your application? How do you link assets to threats?

Problems tend to follow the data flow

We are developers

Focus on the application you are developing

Start with external entities - events which drive activity like a click in the browser

Movie Plot Threats

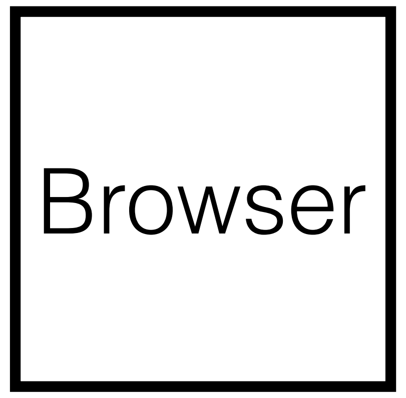
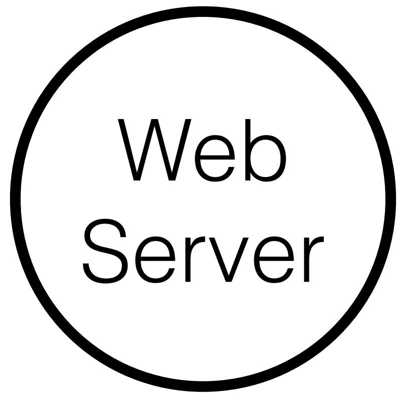
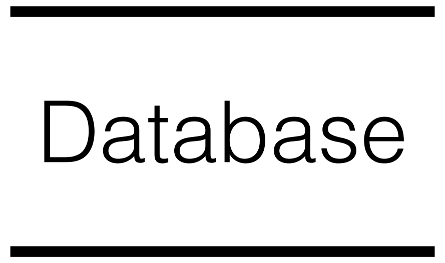
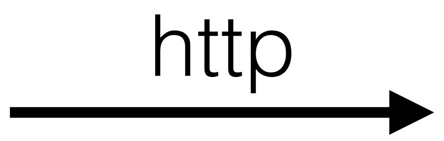
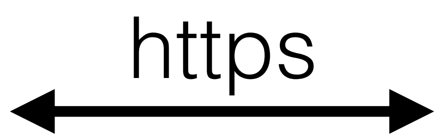
- Fun to discuss
- But not really helpful
- Focus on realistic threats

Creative process

Integrate with bug tracking

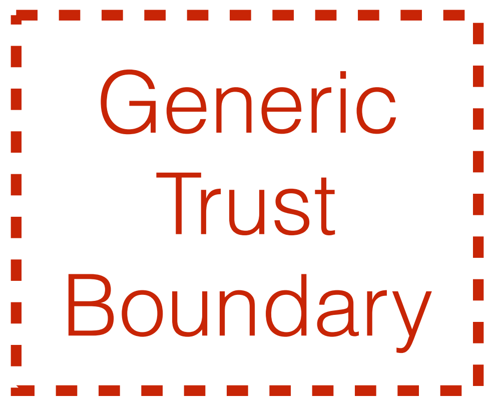
Add any discovered threat, even if you are looking for something else
Tag as security bug in your bug tracker

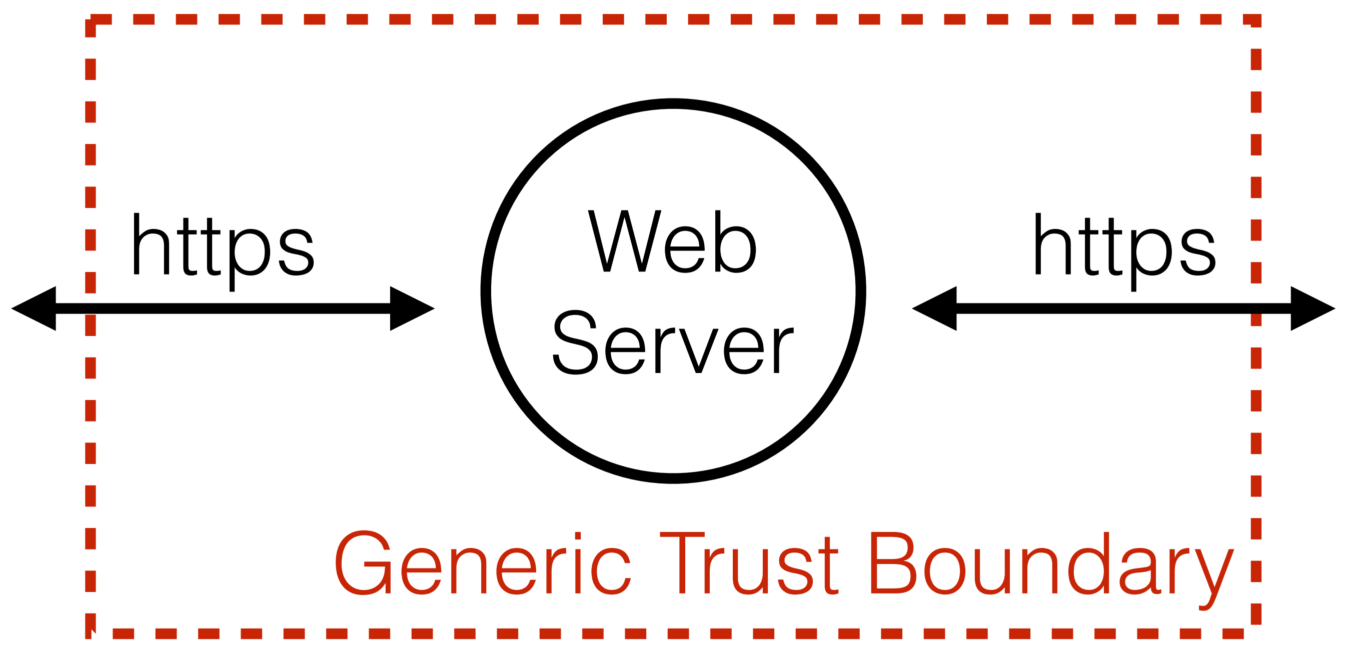
Data Flow Diagrams

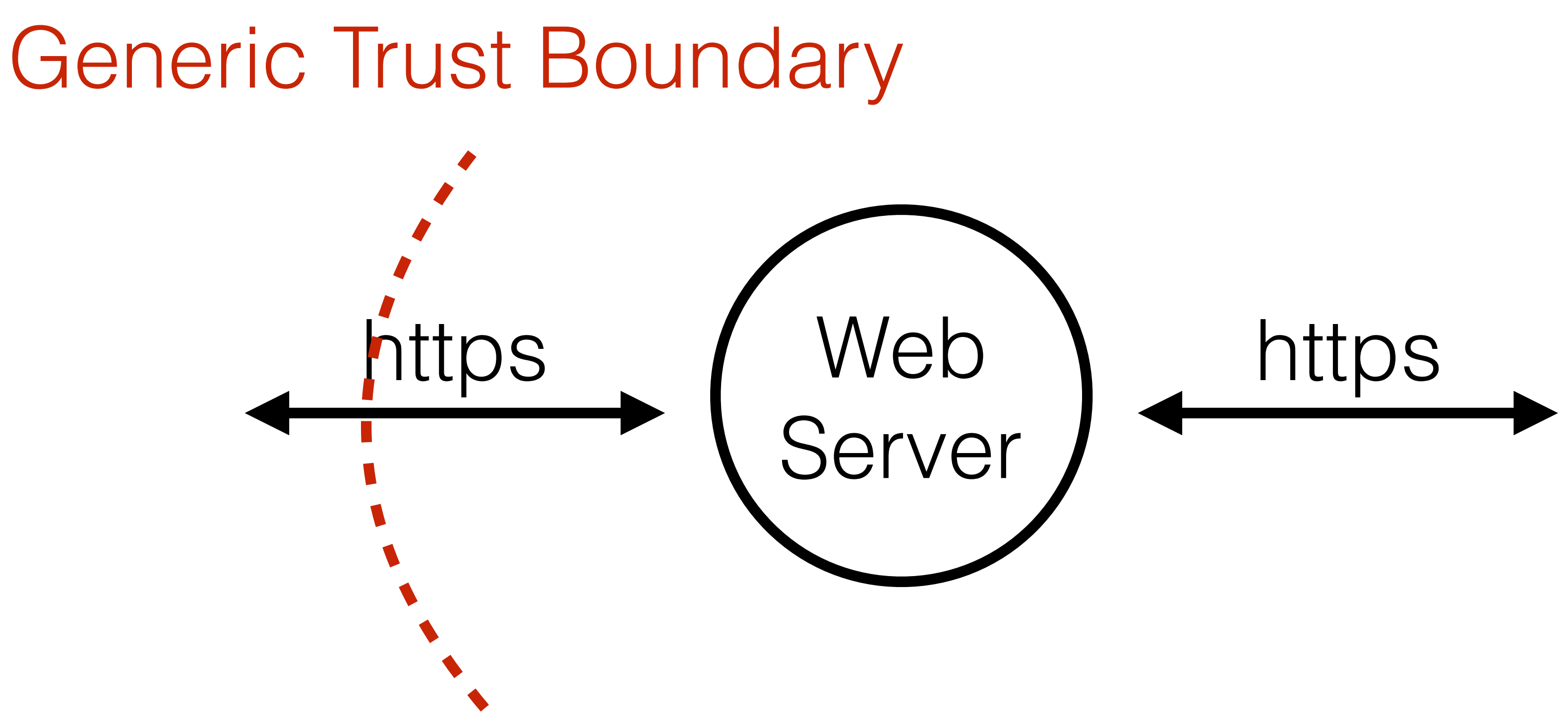
External Entity	People or code outside your control	
Process	Any running code	
Data Store	Things that store data	
Data Flow	Communication between processes or processes and data stores	 

Trust Boundaries

Trust Boundary Where entities with different privileges interact







What are typical boundaries?

Can be technical or organizational

Networks **Servers** **VMs** **Firewalls**

Where are the boundaries?

Start on one side, add a boundary every time the principal changes

1. Browser - anonymous Internet user
2. Web Server - Tomcat user
3. Database - MySQL user

Identifying Threats in Applications

1. What are you building?
2. What can go wrong?
3. What should you do about those things that can go wrong?
4. Did you do a decent job of analysis?

What are you building?

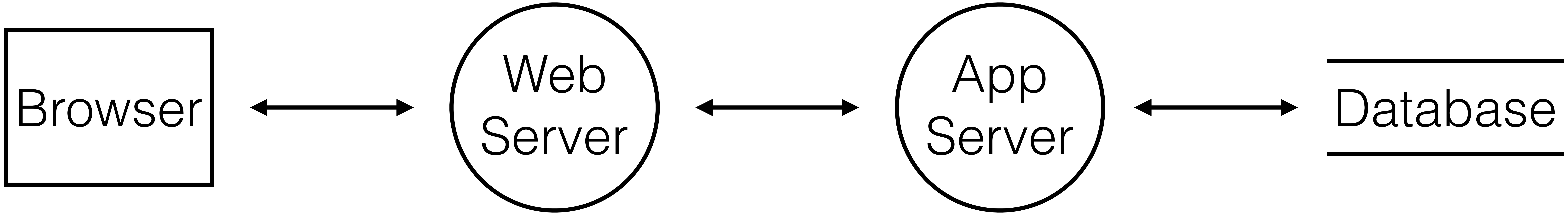
Focus on data flow

„*Sometimes*“ indicates alternatives: model all

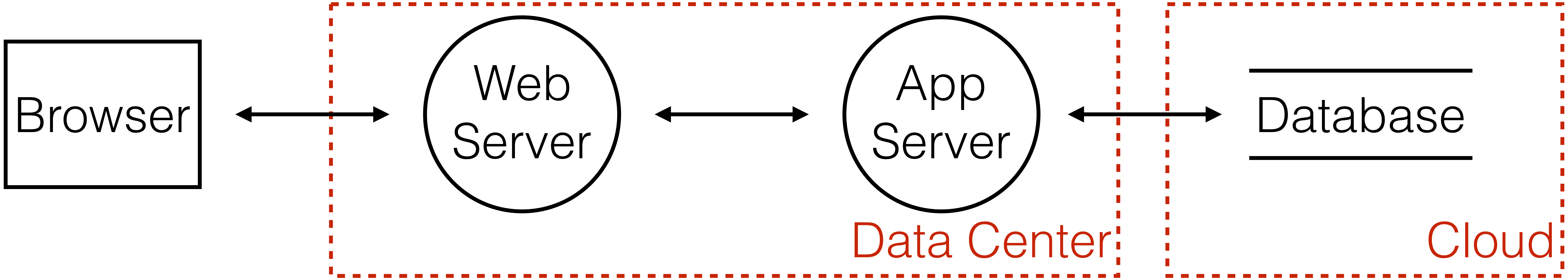
No data sinks: show the consumers

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

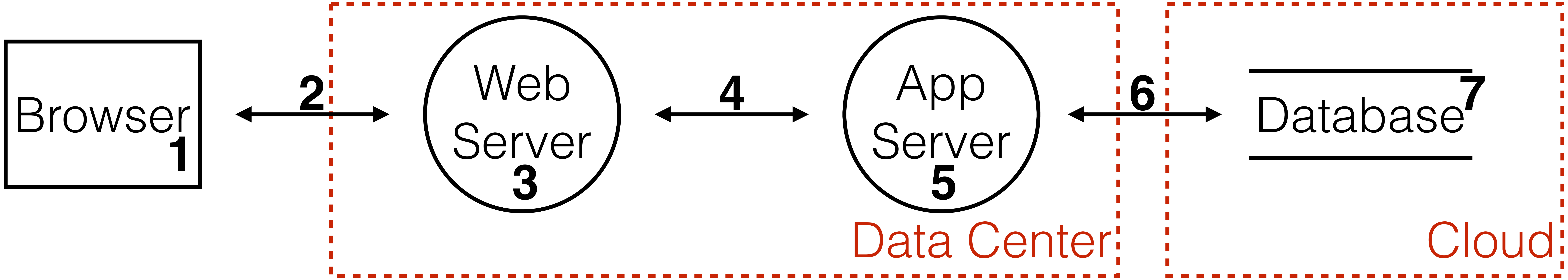
Follow the data



Add trust boundaries



Identify each element



What can go wrong?

Start with the data crossing trust boundaries

Brainstorm meetings with technology experts

Elevation of Privilege game

STRIDE

Focus on threat, not on category

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**

Tampering

Modifying something on disk, network or memory
Violated property: **Integrity**

Repudiation

Claiming that someone didn’t do something
Violated property: **Non-Repudiation**

STRIDE

Information Disclosure

Providing information to someone not authorized
Violated property: **Confidentiality**

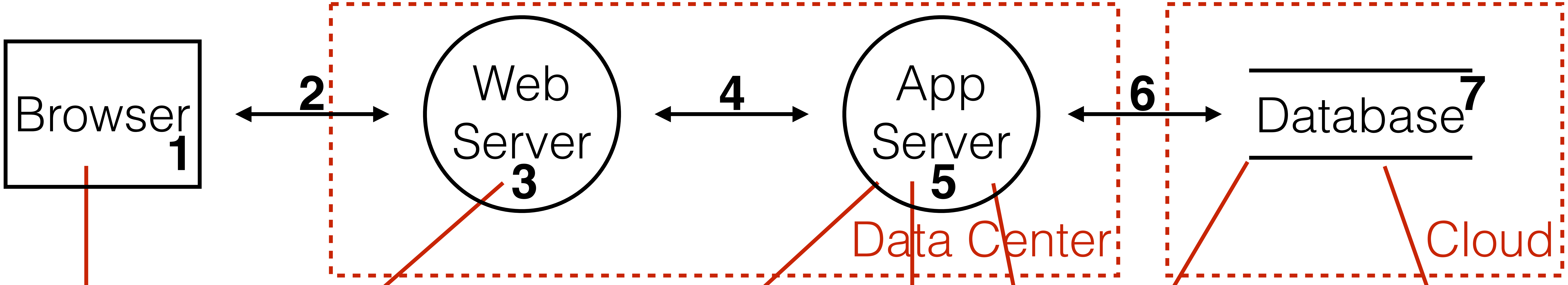
Denial of Service

Absorbing resources needed to provide service
Violated property: **Availability**

Elevation of Privilege

Doing something someone is not authorized to do
Violated property: **Authorization**

Add threats



Denial of Service

Repudiation
- Protect logs against tampering

Tampering
- Data manipulation

Information Disclosure
- Read/ write access to DB

Spoofing
- CSRF
- Identify user (authentication)
- Identify website (certificate)

Elevation of Privilege
- Can someone access backend logic directly?

Addressing each threat

Decide for each threat how to handle it

Mitigate   Eliminate   Transfer   Accept

Mitigate it

Preferred solution

Do something 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)

Transfer it

Team solution

Someone/ something else handles the risk - make sure they do (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 server hard disk)

Threat Target	Mitigation Strategy	Mitigation Technique	Priority	Issue 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 (servers and databases) to provide service	3	1006
Tampering network packets	Cryptography	HTTPS/TLS	1	1007

Is it complete?

Let someone 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 way: Have a threat of each type in STRIDE
Harder way: Have one threat per element of the diagram

Threat Modeling in **Action**

Use one tool to threat model, version your models in a repo and check/ update them every time the application changes.

localhost
Encounters
Search
Search Encounter
My Account
Log out
Edit Userdata
Change Password
Add Encounter
Add Confirmation

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

Duke Encounters - Threat Modeling Tool 2016
File
Edit
View
Settings
Diagram
Reports
Help
Duke Encounters
Data Center
Browser Client
request (HTTPS)
Web Server
response (HTTPS)
request
response
Web Application
write
SQL Database
Cloud
read

Threat List

ID	Title	Category	Description	Justification	Interaction	Diagram	Changed By	Last Modified	State	Priority
1	Spoofing the Browser Client Proc...	Spoofing	Browser Client...		request (HTTPS)	Duke Encount...		28.02.2016 14:0...	Not Started	High
2	Cross Site Scripting	Tampering	The web server...		request (HTTPS)	Duke Encount...		28.02.2016 14:0...	Not Started	High
3	Potential Data Repudiation by We...	Repudiation	Web Server cla...		request (HTTPS)	Duke Encount...		28.02.2016 14:0...	Not Started	High
4	Potential Process Crash or Stop fo...	Denial Of Servi...	Web Server cra...		request (HTTPS)	Duke Encount...		28.02.2016 14:0...	Not Started	High

44 Threats Displayed, 44 Total

Threat Properties

ID: 2 Diagram: Duke Encounters Status: Not Started Last Modified: 28.02.2016 14:07:53

Title: Cross Site Scripting

Category: Tampering

Description: The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.

Threat Properties | Notes - no entries

Demo

Spoofing

Threat Target	Mitigation Strategy	Mitigation Technique
Spoofing a user	Identification and authentication	Password policy, token, password reset process
Fake users	Registration form protection and email verification	Captcha in registration form, pending account unless verified by clicking on email link

localhost
Diagram Information
New Threat
Threat Information
Stencils
Stencil Properties - Web Application Process
Delete Element
https
Data Center <Boundary>
https
https
https
Web Application <Process>
JDBC
JDBC
Web Browser <Process>
Cloud <Boundary>
JDBC
JDBC
Database <Store>

Title

Web Application

Category

Process

Tags

Comma separated tags

Icon

images/icons/website22.svg

Code Type

Managed

Running As

Local Service

Accepts Input From

Kernel, System, or Local Admin

Has Authentication Scheme

□

Has Communication Protocol

□

Has Authorization Scheme

Summary

Threat model before you start to code

Make sure you have addressed every threat

Update your threat model frequently

bridging IT

Königstraße 42
70173 Stuttgart

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

Blog blog.dominikschadow.de
Twitter @dschadow

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

Mozilla SeaSponge
air.mozilla.org/mozilla-winter-of-security-seasponge-a-tool-for-easy-threat-modeling

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

Pictures
www.dreamstime.com