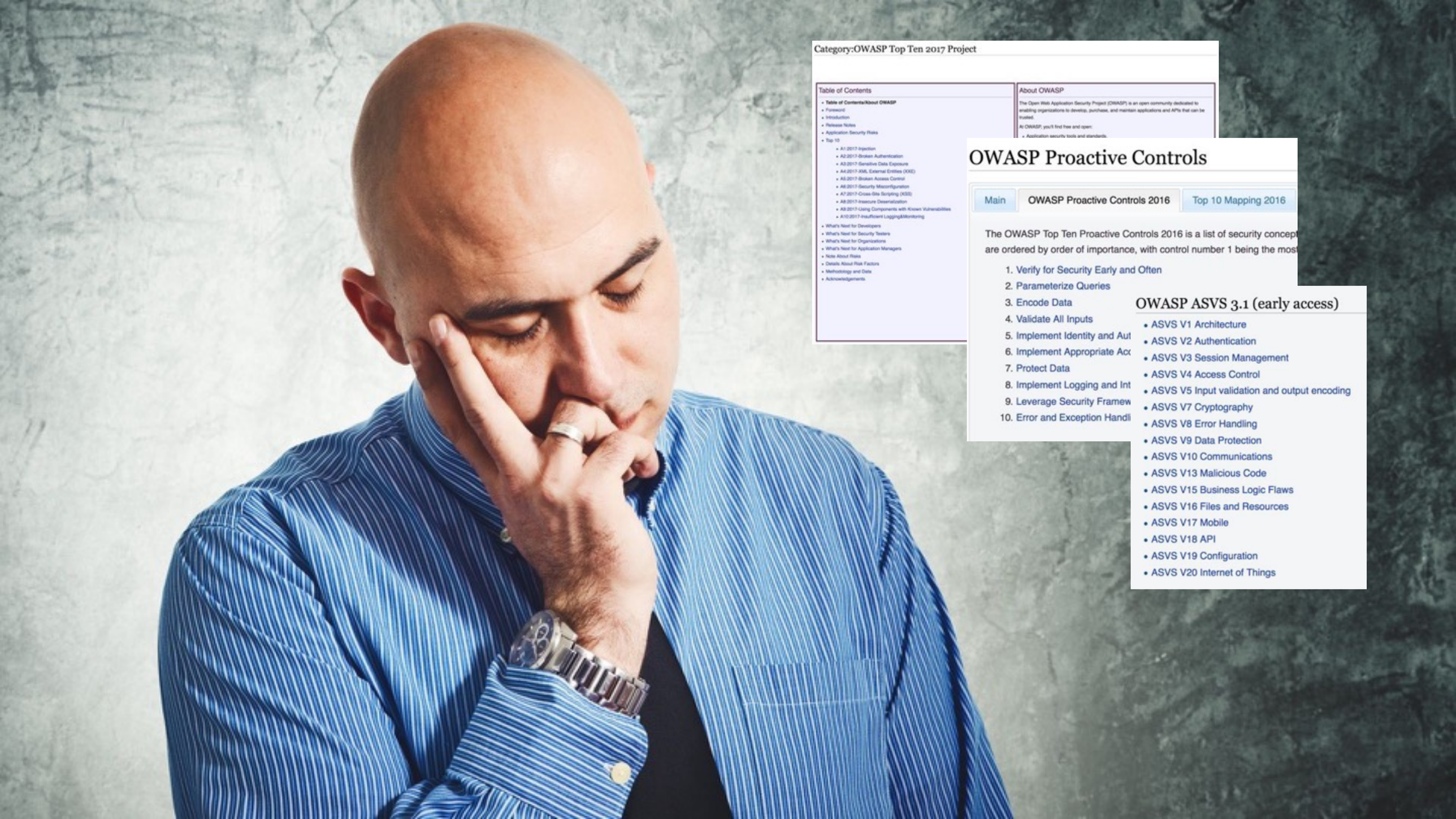
Let’s play security

JAX 2018

Dominik Schadow | bridgingIT


Category:OWASP Top Ten 2017 Project
Table of Contents
Table of Contents/About OWASP
Foreword
Introduction
Release Notes
Application Security Risks
Top 10
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging&Monitoring
What's Next for Developers
What's Next for Security Testers
What's Next for Organizations
What's Next for Application Managers
Note About Risks
Details About Risk Factors
Methodology and Data
Acknowledgements
About OWASP
The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications and APIs that can be trusted.
At OWASP, you'll find free and open:
Application security tools and standards.
OWASP Proactive Controls
Main
OWASP Proactive Controls 2016
Top 10 Mapping 2016
The OWASP Top Ten Proactive Controls 2016 is a list of security concept
are ordered by order of importance, with control number 1 being the most
1. Verify for Security Early and Often
2. Parameterize Queries
3. Encode Data
4. Validate All Inputs
5. Implement Identity and Aut
6. Implement Appropriate Acc
7. Protect Data
8. Implement Logging and Int
9. Leverage Security Framew
10. Error and Exception Handli
OWASP ASVS 3.1 (early access)
ASVS V1 Architecture
ASVS V2 Authentication
ASVS V3 Session Management
ASVS V4 Access Control
ASVS V5 Input validation and output encoding
ASVS V7 Cryptography
ASVS V8 Error Handling
ASVS V9 Data Protection
ASVS V10 Communications
ASVS V13 Malicious Code
ASVS V15 Business Logic Flaws
ASVS V16 Files and Resources
ASVS V17 Mobile
ASVS V18 API
ASVS V19 Configuration
ASVS V20 Internet of Things

Security awareness

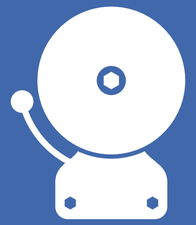

Add (some) fun
to security


The earlier you start,
the more time to
plan and fix

Playing serious games at work

Preparation

for a threat modeling game before integrating the team



Playing

a threat modeling game with the development team

Extensions

to deepen the security knowledge of the developers

Preparation

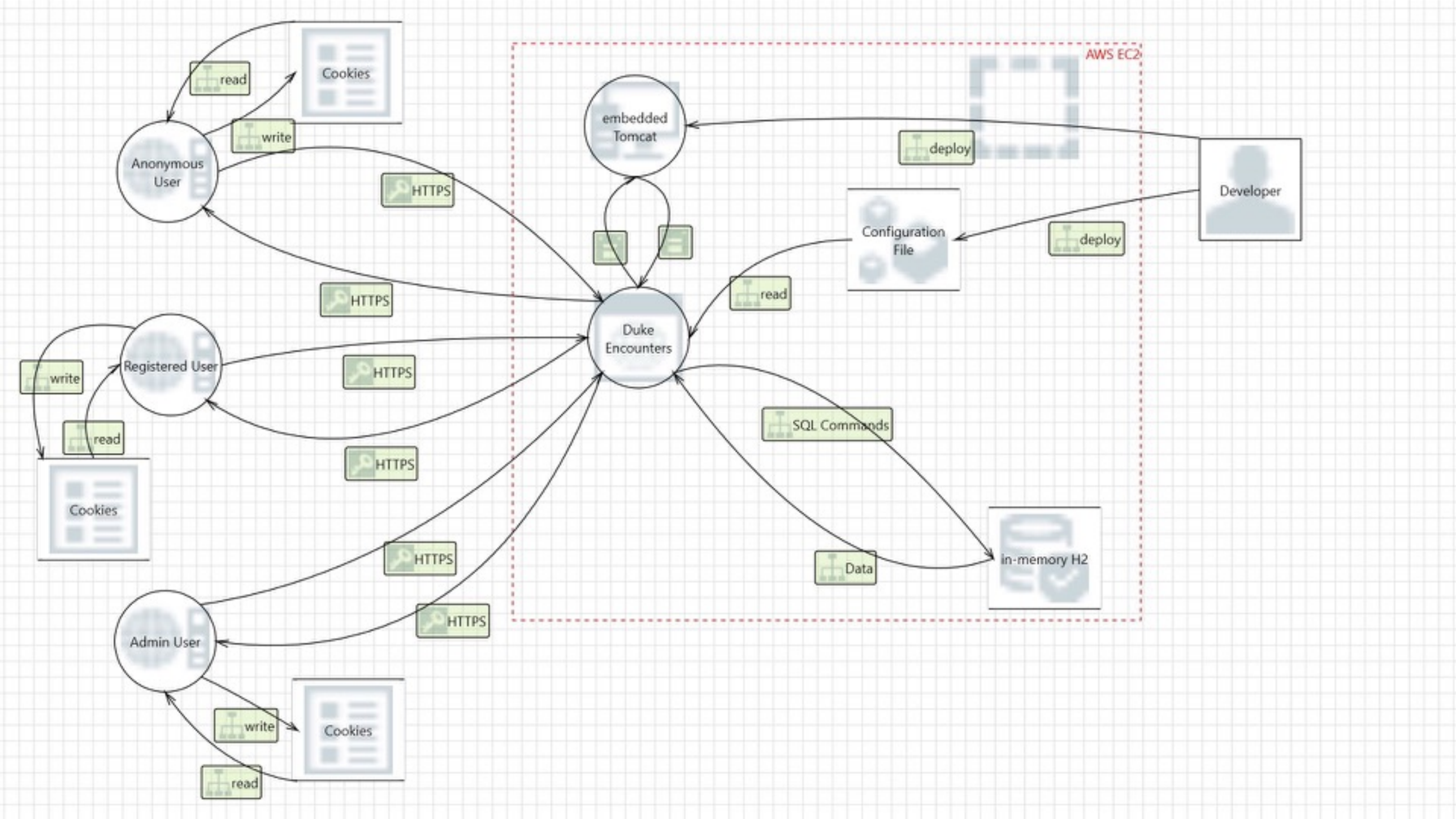
Playing a threat modeling game requires preparation

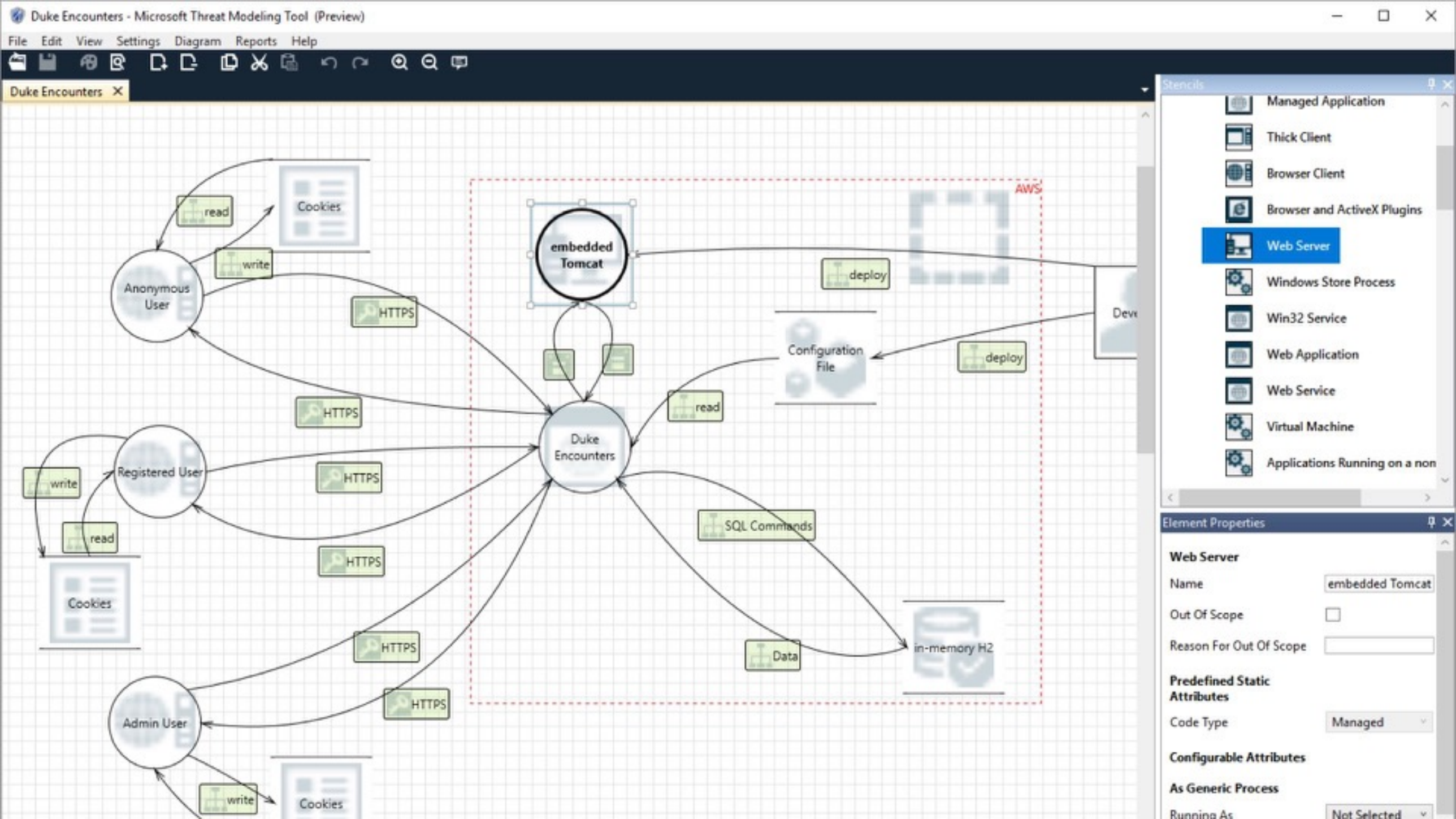
Identifying threats in applications

What are you building?

Using Data Flow Diagrams to show connections

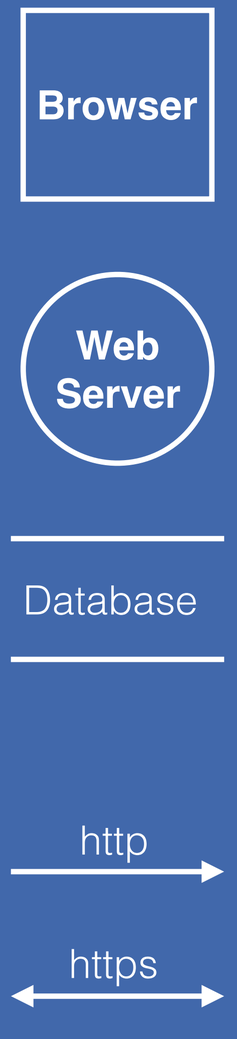
Not a complete architecture diagram
Focus on details that are relevant for security


AWS EC2
Cookies
read
write
Anonymous User
HTTPS
HTTPS
embedded Tomcat
deploy
Developer
Configuration File
deploy
read
Duke Encounters
write
Registered User
HTTPS
read
HTTPS
Cookies
SQL Commands
HTTPS
Data
in-memory H2
HTTPS
Admin User
write
Cookies
read


Duke Encounters - Microsoft Threat Modeling Tool (Preview)
File Edit View Settings Diagram Reports Help
Duke Encounters
Stencils
Managed Application
Thick Client
Browser Client
Browser and ActiveX Plugins
Web Server
Windows Store Process
Win32 Service
Web Application
Web Service
Virtual Machine
Applications Running on a non
Element Properties
Web Server
Name
embedded Tomcat
Out Of Scope
Reason For Out Of Scope
Predefined Static Attributes
Code Type
Managed
Configurable Attributes
As Generic Process
Running As
Not Selected
AWS
embedded Tomcat
deploy
Configuration File
deploy
Dev
read
Duke Encounters
SQL Commands
Data
in-memory H2
Anonymous User
read
write
Cookies
HTTPS
HTTPS
Registered User
write
read
Cookies
HTTPS
HTTPS
Admin User
write
Cookies
HTTPS
HTTPS

Data Flow Diagrams

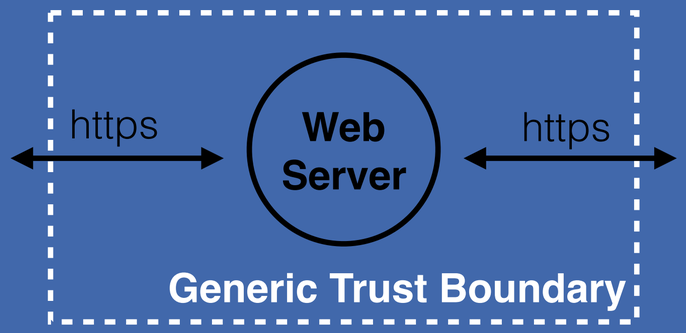
External Entity	People or code outside your control that interact with the application
Process	Code and components that handle data and the dev team controls
Data Store	Anything that stores data and does not modify it
Data Flow	Represents data movement within the application (including direction)

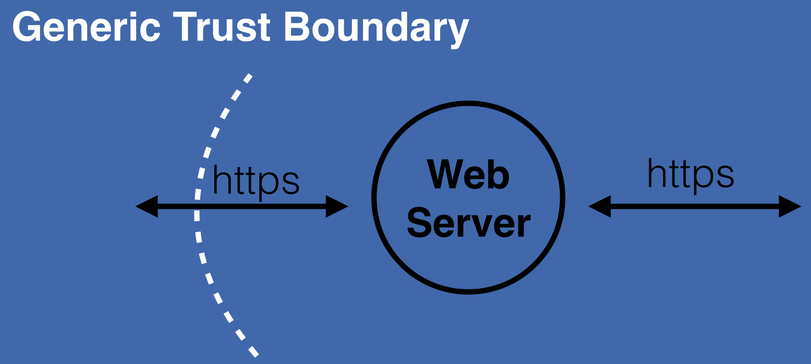


Trust Boundaries

Trust Boundary

Represents the change of privilege levels as the data flows through the application (change in level of trust)





Typical boundary locations

Follow the data, add boundary for new principal

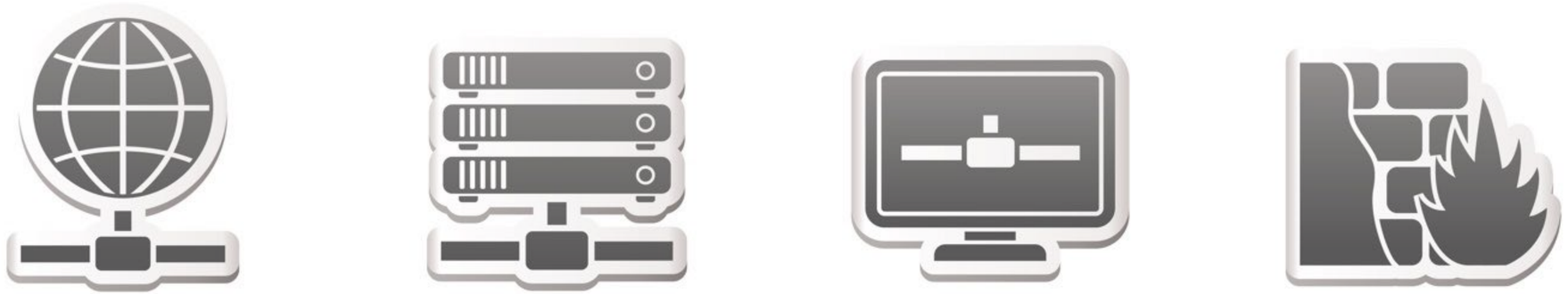
`Anonymous user`

`Tomcat user`

`MySQL user`

Typical boundaries

Can be technical or organizational



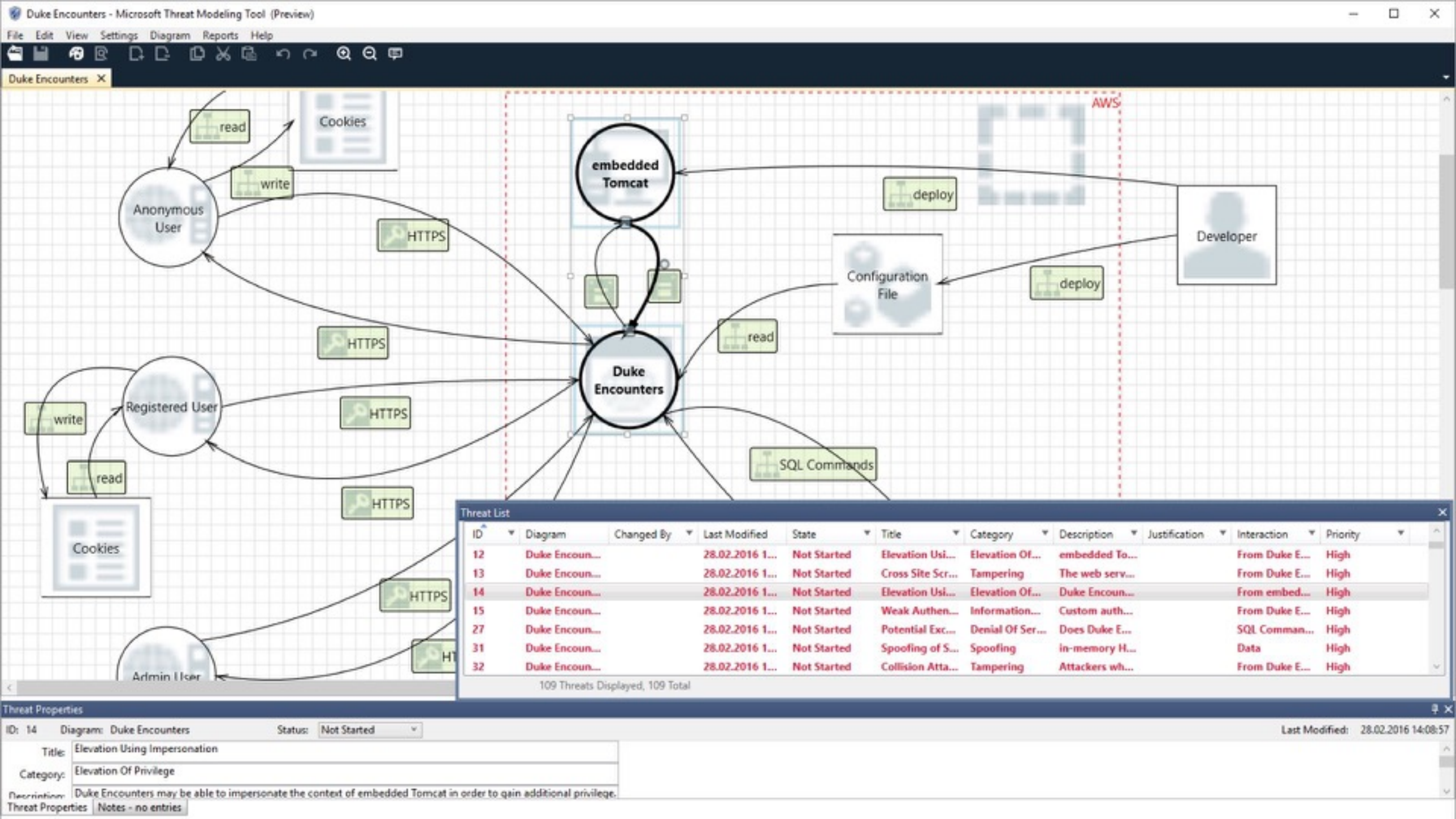
A data flow diagram is a living document

Version every diagram in a repo and keep it in sync with reality

Check and update them every time the application changes and regularly from time to time

Execution

Track down and identify threats


Duke Encounters - Microsoft Threat Modeling Tool (Preview)
File Edit View Settings Diagram Reports Help
Duke Encounters
AWS
read
Cookies
write
Anonymous User
HTTPS
embedded Tomcat
deploy
Developer
Configuration File
deploy
read
HTTPS
Duke Encounters
HTTPS
write
Registered User
read
Cookies
HTTPS
SQL Commands
HTTPS
Admin User
Threat List
ID Diagram Changed By Last Modified State Title Category Description Justification Interaction Priority
12 Duke Encoun... 28.02.2016 1... Not Started Elevation Usi... Elevation Of... embedded To... From Duke E... High
13 Duke Encoun... 28.02.2016 1... Not Started Cross Site Scr... Tampering The web serv... From Duke E... High
14 Duke Encoun... 28.02.2016 1... Not Started Elevation Usi... Elevation Of... Duke Encoun... From embed... High
15 Duke Encoun... 28.02.2016 1... Not Started Weak Authen... Information... Custom auth... From Duke E... High
27 Duke Encoun... 28.02.2016 1... Not Started Potential Exc... Denial Of Ser... Does Duke E... SQL Comman... High
31 Duke Encoun... 28.02.2016 1... Not Started Spoofing of S... Spoofing in-memory H... Data High
32 Duke Encoun... 28.02.2016 1... Not Started Collision Atta... Tampering Attackers wh... From Duke E... High
109 Threats Displayed, 109 Total
Threat Properties
ID: 14 Diagram: Duke Encounters Status: Not Started Last Modified: 28.02.2016 14:08:57
Title: Elevation Using Impersonation
Category: Elevation Of Privilege
Description: Duke Encounters may be able to impersonate the context of embedded Tomcat in order to gain additional privilege.
Threat Properties Notes - no entries

103
163
112
136

Start with a Data Flow Diagram walkthrough

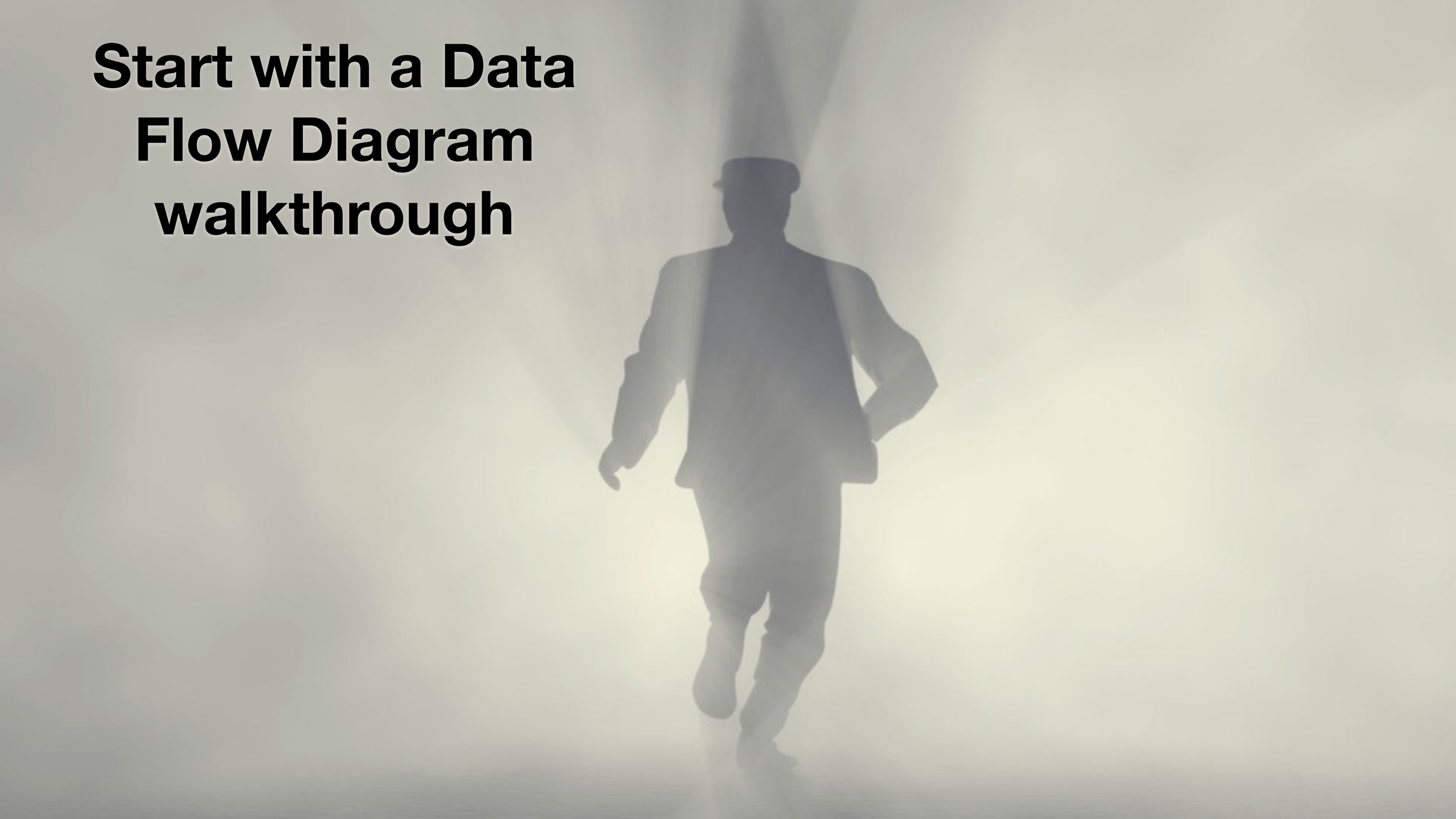


Table of Contents

The Elevation of Privilege game

Helps non-security-experts to find security flaws

Card game for 3 - 5 players (78 cards)
Requires STRIDE knowledge
Not all are Java web application relevant

Suits

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 authN

Tampering Modifying something on disk, network or memory
Violated property: Integrity
Standard defenses: Digital signatures, hashes

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

STRIDE

Information Disclosure

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

Denial of Service

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

Elevation of Privilege

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

STRIDE per element

Not all threats are likely to all elements

	S	T	R	I	D	E
External Entity	X		X			
Process	X	X	X	X	X	X
Data Store		X		X	X	
Data Flow		X		X	X	

4

Tampering

Your code makes access control decisions all over the place, rather than with a security kernel.

4

Category

Description

Score

Follow the data

Identify security flaws in your web application

Start with external entities
Follow the data flow through your application in a structured way

Focus on threats

Developers tend to think in solutions

It’s important to find the threats first
Play a threat even if you think it is already mitigated (*really, everywhere?*)
Be specific in naming threats

Score Card

Microsoft® Security Development Lifecycle

Name	Points	Card	Component(s)	Notes on Threat

Score +1 for each card with a valid threat to the application under consideration

Score +1 for the winner of a round

The OWASP Cornucopia game

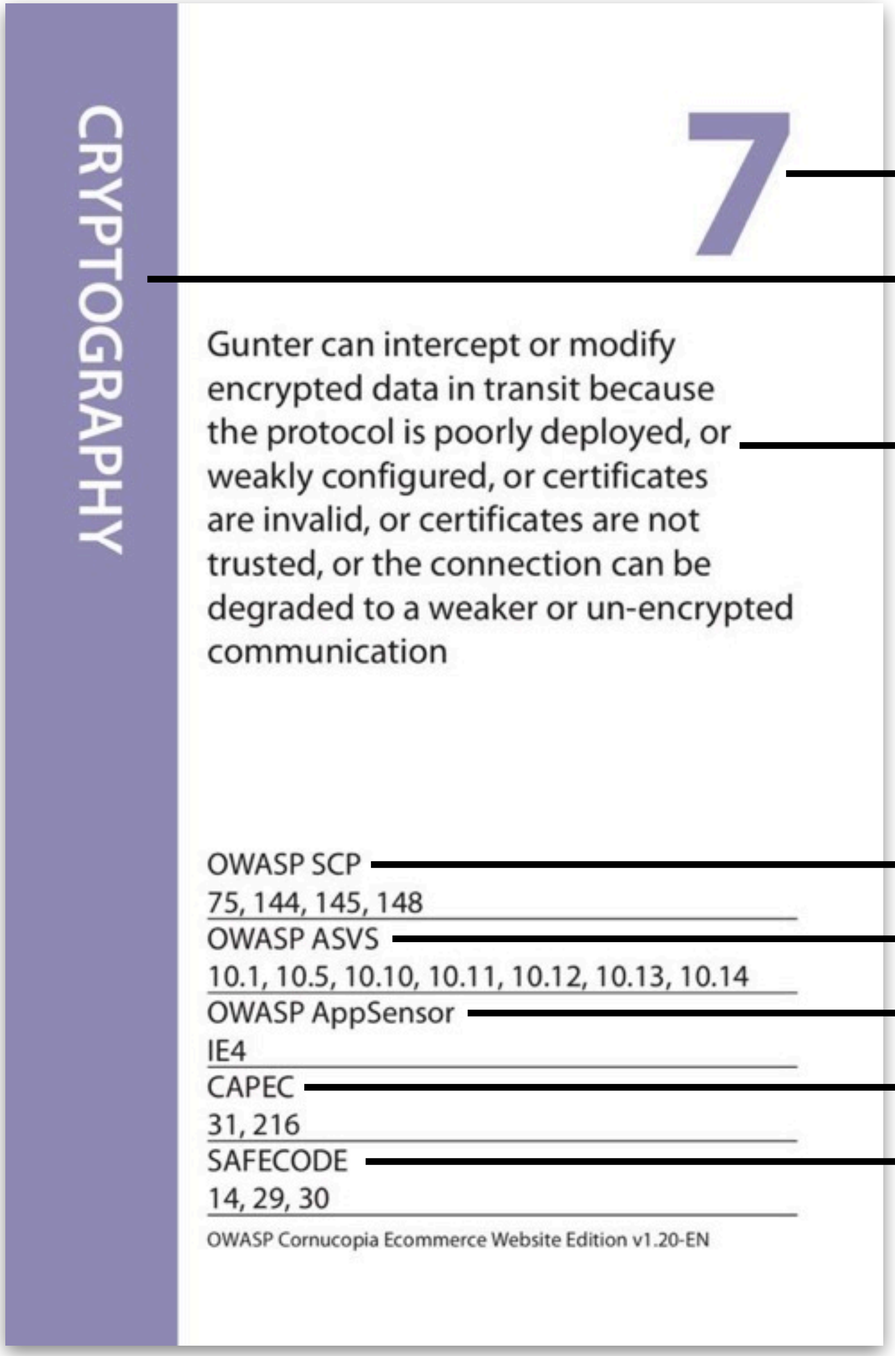
Identify application security requirements

Card game for 4 - 6 players (80 cards)
Same ideas and rules as EoP
Web application centric

Suits

Structure based on different OWASP guides

Data Validation and Encoding, Authentication, Session Management, Authorization, Cryptography, Cornucopia („*Füllhorn*“, everything else, acts as trump)



Score

Category

Description

(OWASP Secure Coding Practices)

OWASP Application Security Verification Standard

OWASP Application Intrusion Detection

Common Attack Pattern Enumeration and Classification

(Practical Security Stories and Security Tasks for Agile Development Environments)

Data Validation and Encoding (VE)
2
3
4
5
6
7
8
9
10
J
Q
K
A

Session Management (SM)
2
3
4
5
6
7
8
9
10
J
Q
K
A

Cryptography (CR)
2
3
4
5
6
7
8
9
10
J
Q
K
A

Player Name	Tally			
	Requirements	Rounds/hands	Total	Rank
Example	*IIII*	*1*	*5*	

Authentication (AT)
2
3
4
5
6
7
8
9
10

Authorization (AZ)
2
3
4
5
6
7

Cornucopia (C)
2
3
4
5
6
7

Application

Score +1 for each card with a valid threat to the application under consideration

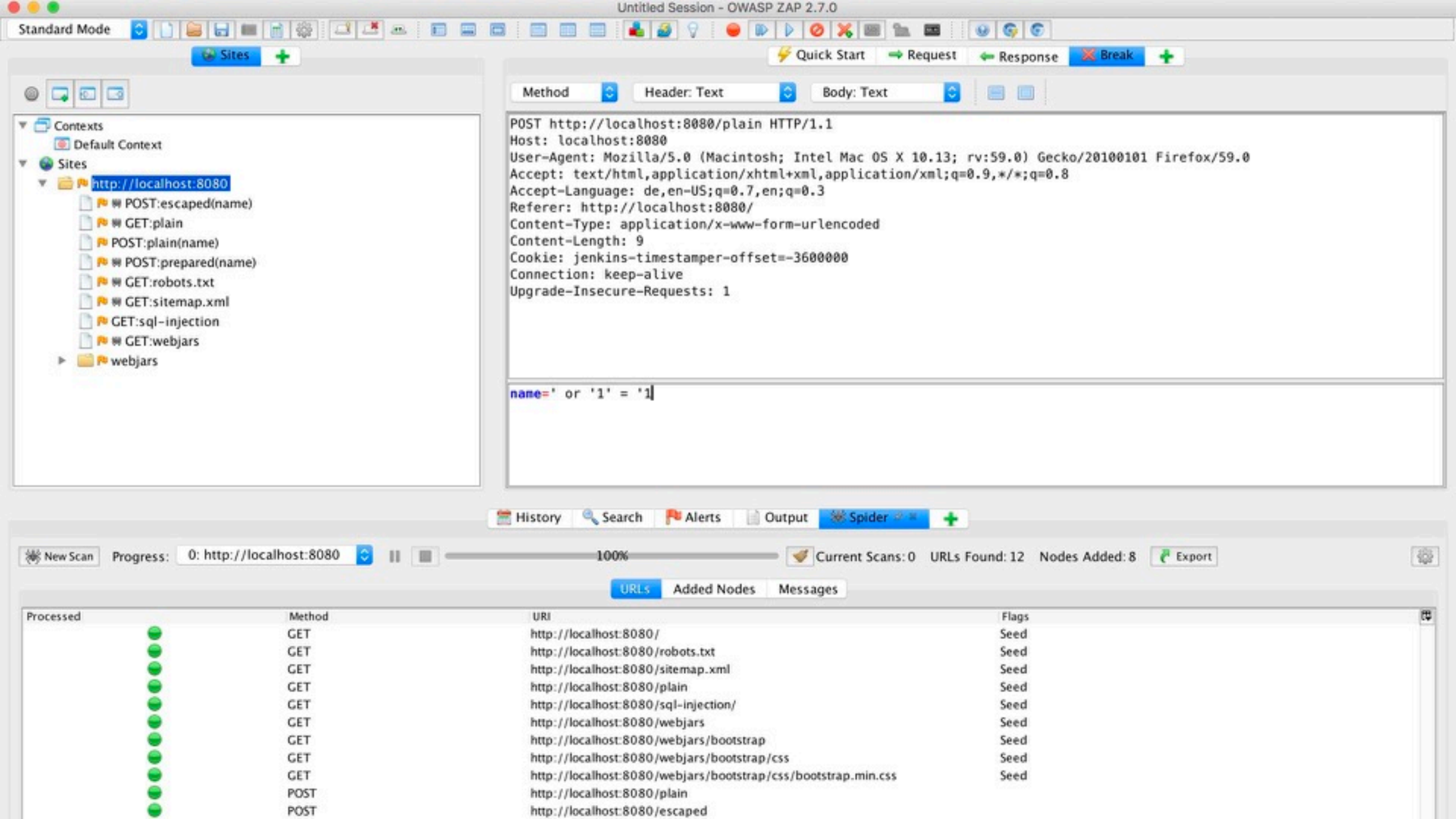
Score +1 for the winner of a round


Hack yourself
(first)

Threat verification

A lot of threats are easily discoverable

Attack your web application on a **TEST** system or **localhost**
Try to prove presence or absence of a threat


Untitled Session - OWASP ZAP 2.7.0
Standard Mode
Sites
Quick Start
Request
Response
Break
Method
Header: Text
Body: Text
Contexts
Default Context
Sites
http://localhost:8080
POST:escaped(name)
GET:plain
POST:plain(name)
POST:prepared(name)
GET:robots.txt
GET:sitemap.xml
GET:sql-injection
GET:webjars
webjars
POST http://localhost:8080/plain HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:59.0) Gecko/20100101 Firefox/59.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: de,en-US;q=0.7,en;q=0.3
Referer: http://localhost:8080/
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Cookie: jenkins-timestamper-offset=-3600000
Connection: keep-alive
Upgrade-Insecure-Requests: 1
name=' or '1' = '1
History
Search
Alerts
Output
Spider
New Scan
Progress:
0: http://localhost:8080
100%
Current Scans: 0
URLs Found: 12
Nodes Added: 8
Export
URLs
Added Nodes
Messages
Processed
Method
URI
Flags
GET http://localhost:8080/ Seed
GET http://localhost:8080/robots.txt Seed
GET http://localhost:8080/sitemap.xml Seed
GET http://localhost:8080/plain Seed
GET http://localhost:8080/sql-injection/ Seed
GET http://localhost:8080/webjars Seed
GET http://localhost:8080/webjars/bootstrap Seed
GET http://localhost:8080/webjars/bootstrap/css Seed
GET http://localhost:8080/webjars/bootstrap/css/bootstrap.min.css Seed
POST http://localhost:8080/plain
POST http://localhost:8080/escaped

Wrap-up every game

Issues Boards Create Search

Duke Encounters / DUKE-1

29. Cross Site Request Forgery

Edit Comment Assign to 'In Progress' Admin

Details

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

Description

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

Attachments

Drop files to attach, or browse.

Activity

All | Comments | Work log | History | Activity

People

Assignee: Dominik
Reporter: Dominik

Votes: 0
Watchers: 1 Stop watching

Dates

Created: 7 minutes ago
Updated: 7 minutes ago

Agile

View on Board

Address 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

Add Spring Security AND configure it

Eliminate it

Most secure solution

Results in feature elimination most of the time

Remove admin functionality from Internet facing application

Transfer it

Team solution

Someone/ something else handles the risk,
depending who can easily fix the problem

Operations adds a web application firewall

Accept it

Last resort solution

Stop worrying about it and live with the risk

Someone steals your servers' hard disk

Someday your web applications will be too secure for you to hack them easily

Continue hacking with vulnerable web apps

Use intentionally vulnerable web applications and hack them for educational purposes

Extensions

Integrate intentionally vulnerable web applications

All Products

Image	Product	Description	Price	
	Apple Juice (1000ml)	The all-time classic.	1.99	
	Apple Pomace	Finest pressings of apples. Allergy disclaimer: Might contain traces of worms. Can be **sent back to us** for recycling.	0.89	
	Banana Juice (1000ml)	Monkeys love it the most.	1.99	
	Carrot Juice (1000ml)	As the old German saying goes: "Carrots are good for the eyes. Or has anyone ever seen a rabbit with glasses?"	2.99	
	Eggfruit Juice (500ml)	Now with even more exotic flavour.	8.99	
	Fruit Press	Fruits go in. Juice comes out. Pomace you can send back to us for recycling purposes.		
	Green Smoothie	Looks poisonous but is actually very good for your health! Made from green cabbage, spinach, kiwi and grass.		

Punkteübersicht

27%

Schwierigkeit

1: 7/7 | 2: 8/8 | 3: 1/13 | 4: 0/15 | 5: 0/10 | 6: 0/6

★ ★

Name	Beschreibung	Status
Basket Access	Access someone else's basket.	gelöst
Christmas Special	Order the Christmas special offer of 2014.	gelöst
Deprecated Interface	Use a deprecated B2B interface that was not properly shut down.	gelöst
Five-Star Feedback	Get rid of all 5-star customer feedback.	gelöst
Login Admin	Log in with the administrator's user account.	gelöst
Login MC SafeSearch	Log in with MC SafeSearch's original user credentials without applying SQL Injection or any other bypass.	gelöst
Password Strength	Log in with the administrator's user credentials without previously changing them or applying SQL Injection.	gelöst
Weird Crypto	**Inform the shop** about an algorithm or library it should definitely not use the way it does.	gelöst

★ ★ ★

Name	Beschreibung	Status

Reset lesson

1

What is WebGoat?

WebGoat is a deliberately insecure application that allows interested developers just like you to *test vulnerabilities* commonly found in Java-based applications that use common and popular open source components.

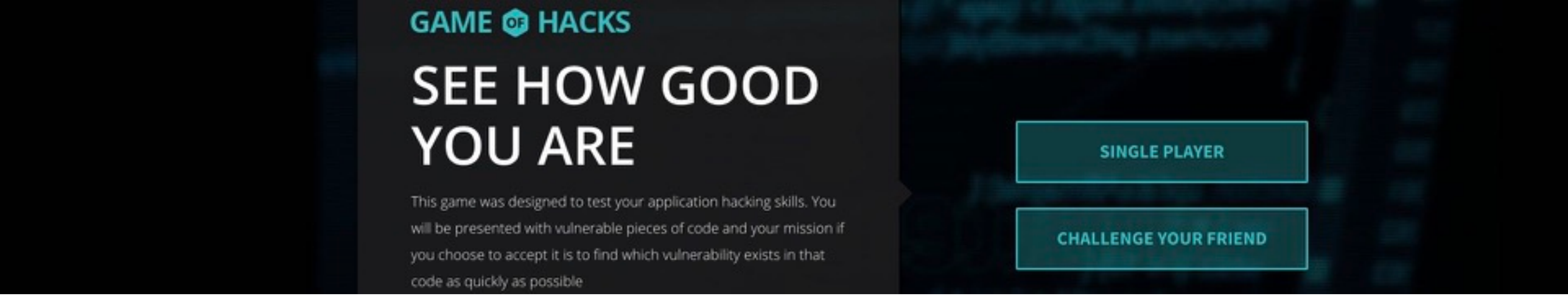
Now, while we in no way condone causing intentional harm to any animal, goat or otherwise, we think learning everything you can about security vulnerabilities is essential to understanding just what happens when even a small bit of unintended code gets into your applications.

What better way to do that than with your very own scapegoat?

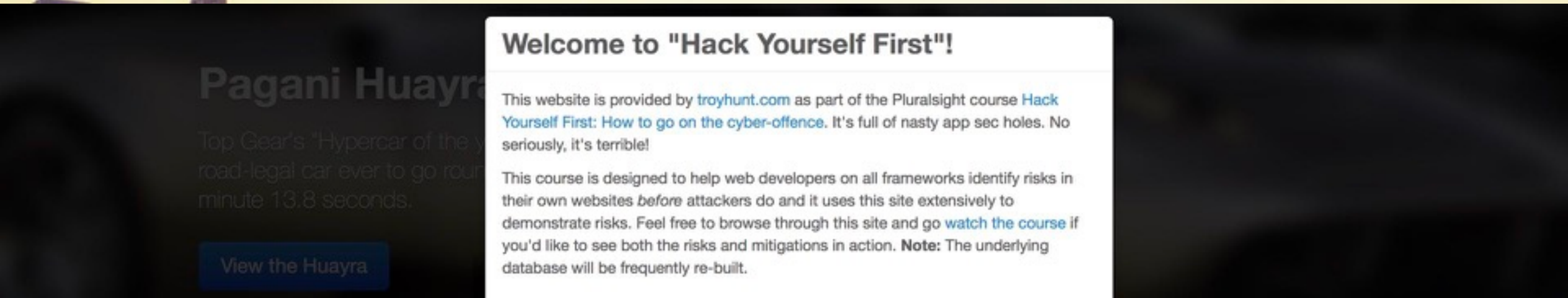
Feel free to do what you will with Hack. Poke, prod and if it makes you feel better, scare him until your heart's content. Go ahead, and Hack the goat. We promise he likes it.

Thanks for your interest!

The WebGoat Team


GAME OF HACKS
SEE HOW GOOD YOU ARE
This game was designed to test your application hacking skills. You will be presented with vulnerable pieces of code and your mission if you choose to accept it is to find which vulnerability exists in that code as quickly as possible
SINGLE PLAYER
CHALLENGE YOUR FRIEND


HACK-ME
THE HOUSE OF THE RISING SANDBOX
BUILD, HOST, AND SHARE FREELY VULNERABLE WEB APPLICATIONS
EXPLORE THE PROJECT
Start a hackme
RUN A VULNERABLE WEB APP ON-THE-FLY
Upload/create your hackme
SHARE YOUR SKILLS WITH THE COMMUNITY

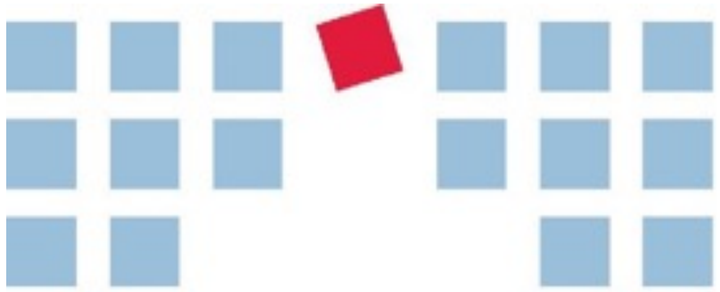

Pagani Huayra
Top Gear's "Hypercar of the
road-legal car ever to go rou
minute 13.8 seconds.
View the Huayra
Welcome to "Hack Yourself First"!
This website is provided by troyhunt.com as part of the Pluralsight course Hack Yourself First: How to go on the cyber-offence. It's full of nasty app sec holes. No seriously, it's terrible!
This course is designed to help web developers on all frameworks identify risks in their own websites *before* attackers do and it uses this site extensively to demonstrate risks. Feel free to browse through this site and go watch the course if you'd like to see both the risks and mitigations in action. **Note:** The underlying database will be frequently re-built.

Security must be an important part of every development project

Use **Threat Modeling** to discover serious security flaws

EoP and **Cornucopia** help developers to locate security problems

Vulnerable web applications deepen security knowledge



bridging IT

Marienstr. 17
70178 Stuttgart

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

Blog blog.dominikschadow.de
Twitter @dschadow

Common Attack Pattern Enumeration and Classification
https://capec.mitre.org

Elevation of Privilege Card Game
https://www.microsoft.com/en-us/SDL/adopt/eop.aspx

Game of Hacks
http://www.gameofhacks.com

hack.me
https://hack.me

Hack Yourself First (Troy Hunt)
http://hackyourselffirst.troyhunt.com

Microsoft Threat Modeling Tool
https://docs.microsoft.com/de-de/azure/security/azure-security-threat-modeling-tool

OWASP Application Security Verification Standard
https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

OWASP AppSensor
https://www.owasp.org/index.php/OWASP_AppSensor_Project

OWASP Cornucopia
https://www.owasp.org/index.php/OWASP_Cornucopia

OWASP Juice Shop
https://github.com/bkimminich/juice-shop

OWASP Secure Coding Practices
https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide

OWASP WebGoat
https://github.com/WebGoat/WebGoat

OWASP Zed Attack Proxy
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Practical Security Stories and Security Tasks for Agile Development Environments
http://www.safecode.org/publications/SAFECode_Agile_Dev_Security0712.pdf

Pictures
http://www.dreamstime.com