OWASP

The Open Web Application Security Project

A Short Introduction to Threat Modeling

as part of a secure software development lifecycle

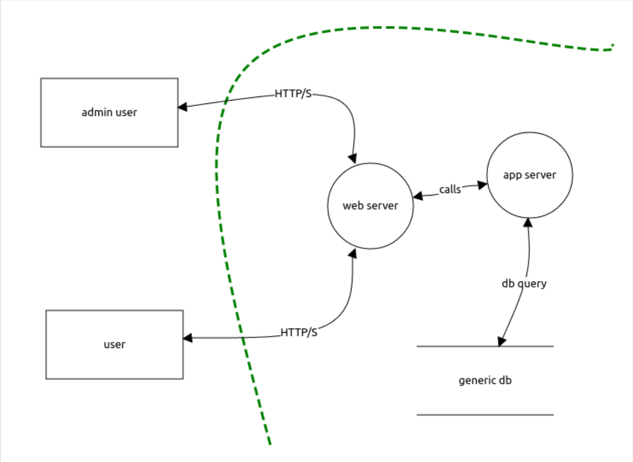
What will be covered

OWASP
The Open Web Application Security Project

- Threat models
- The tools
- Why they are useful
- Open source Threat Dragon
- Cup Cake!

What is a threat model?

Essentially a data flow diagram



- Protected assets
- Attack vectors
- Attack surfaces

... that lists possible threats

Who models threats?

- It can be diagrammatic
- It can be a spreadsheet
- It can be descriptive

… so, who already uses threat modeling?

Recap on vulnerability, asset, threat

- Vulnerability

  an exploitable weakness in a system or its design

- Asset

  anything that is valuable to an organization

- Threat

  potential danger to an asset

- Vector

  method to realise an exploit

- Trust boundary
- *change in level of trust for information or execution*

OWASP

- OWASP top ten threats
- OWASP top ten remediations
- OWASP threat modeling cheat sheet(s)

OWASP Top Ten (2013 Edition)

OWASP
The Open Web Application Security Project

- A1: Injection
- A2: Broken Authentication and Session Management
- A3: Cross-Site Scripting (XSS)
- A4: Insecure Direct Object References
- A5: Security Misconfiguration
- A6: Sensitive Data Exposure
- A7: Missing Function Level Access Control
- A8: Cross Site Request Forgery (CSRF)
- A9: Using Known Vulnerable Components
- A10: Unvalidated Redirects and Forwards

OWASP Top 10 Proactive Controls

OWASP
The Open Web Application Security Project

- C1 Verify for security often and early
- C2 Parameterize Queries
- C3 Encode Data Before Use
- C4 Validate all Inputs
- C5 Establish Authentication and Identity Controls
- C6 Implement Appropriate Access Controls
- C7 Protect Data
- C8 Implement Logging And Intrusion Detection
- C9 Leverage Security Frameworks and Libraries
- C10 Error and Exception Handlin

OWASP cheat sheets

- OWASP threat modeling cheat sheet(s)

  DRAFT CHEAT SHEET - WORK IN PROGRESS

  Introduction

  The objective of this cheat sheet is to provide guidance to developers, reviewers, designers and architects on conducting successful threat modeling. The main goal of threat modeling is to understand the controls needed for a software system. This is a complex endeavor that will involve investigations into:

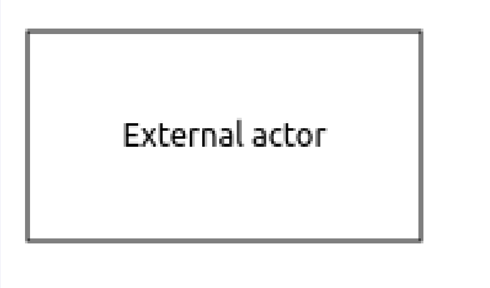
- *The trust boundaries to and within the solution that we build*
- *The actors that interact within and outside of the trust boundaries*
- *etc*

Actors, Processes and Trust Boundaries

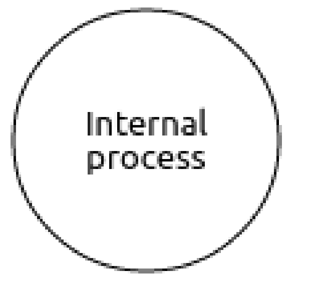
OWASP
The Open Web Application Security Project

Threat Model components

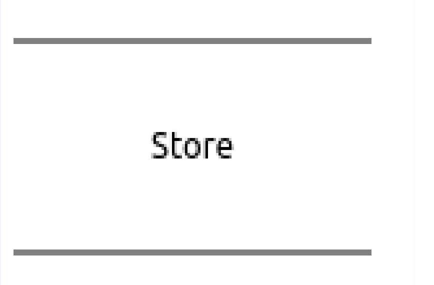
Actor



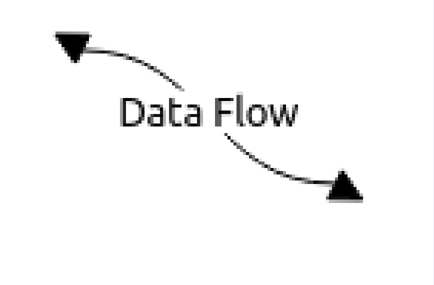
Process



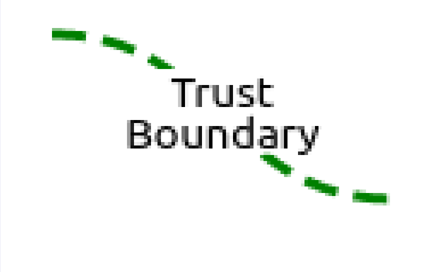
Storage



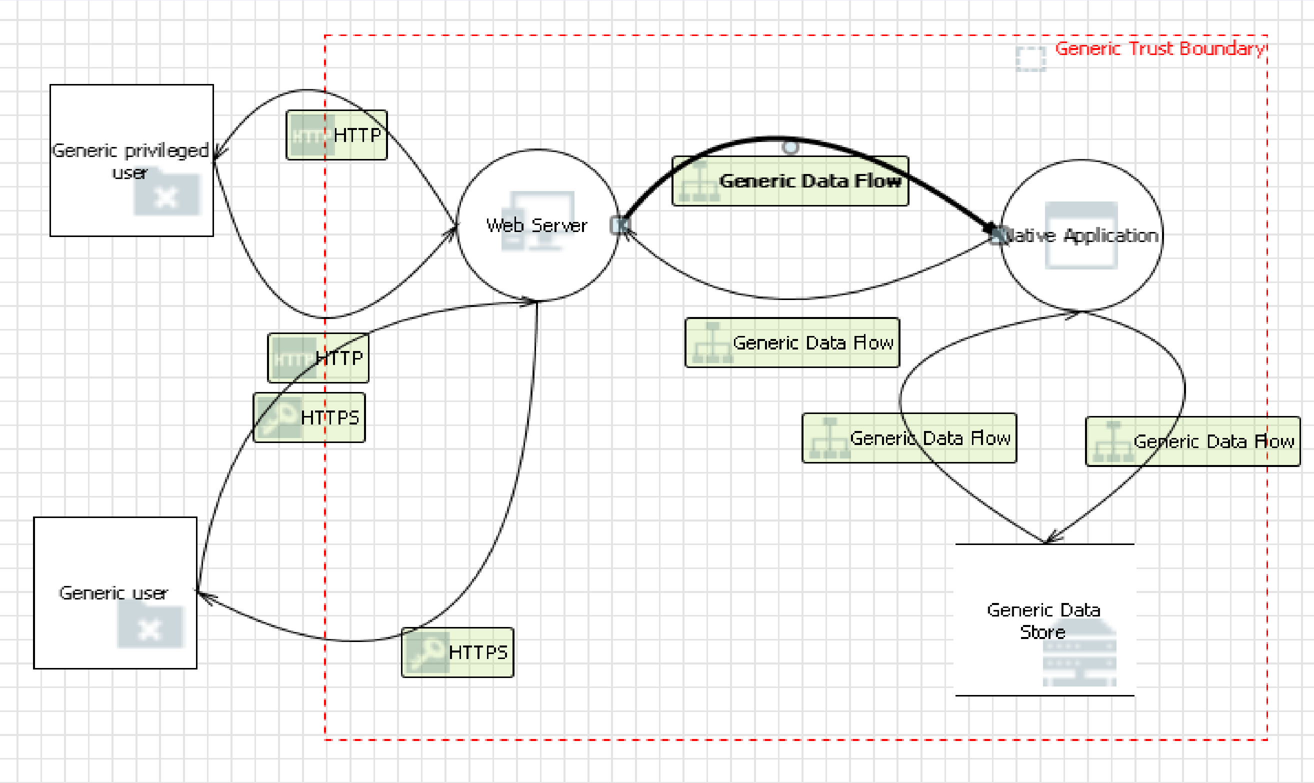
Data flow



Trust boundary



Microsoft Threat Model Tool



- 33 *possible* threats automatically identified

Something is better than nothing

- The days of 'the whip' are very last century
- More tact and carrot
- Think hard before modeling existing systems
- Incremental threat modeling

Some threats can never be modeled

- Government agencies
- Service provider
- Back doors
- The human (wet ware)

Questions (and answers)

Questions?

(and maybe some answers)

… before the Threat Dragon demo

Demonstration of Threat Dragon

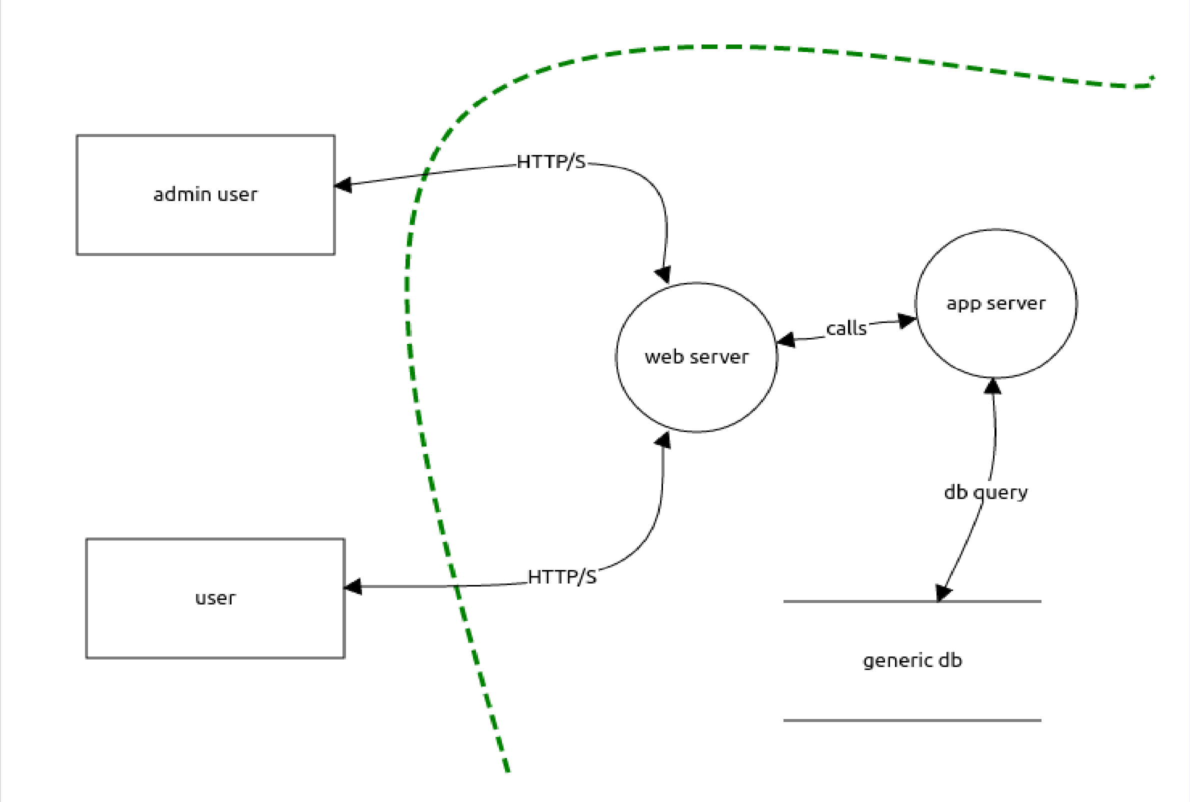
The Open Web Application Security Project

Welcome! (0.1.26)

- Free, open source threat modeling tool
- OWASP incubator project
- Web application
- Desktop application for local use
- Mike Goodwin

OWASP Threat Dragon

OWASP
The Open Web Application Security Project



- 0 threats identified (needs work)

Cupcake's call for contributors

- Contribute
- Have a github account?
- Node.js
- Angular
- Electron
- MongoDB (well, not yet)