Thoughts on Threat Modelling

John DiLeo (@gr4ybeard)

IriusRisk/OWASP New Zealand

July 2023

Thank You to Our Sponsors and Hosts!

Without them, this Conference couldn't happen.

About Me

- Past lives
  - Simulation developer and system analyst
  - University lecturer - Math, Comp Sci, IT, *et al.*
  - J2EE developer and architect
- Full-time AppSec Architect/Consultant - since 2014
- Moved from US to Auckland - late 2017

About My Day Job

IriusRisk – Solution Architect, Asia/Pacific

- Commercial Threat Modelling Tool
  - Customer Proof-of-Value Engagements
  - Some post-sales Customer Success support
- Practice and Brand Ambassador
  - Conferences
  - Training

About My *Other* 'Job'

"Dr. OWASP"

Chapter Leader, OWASP New Zealand

Chair, OWASP New Zealand Day – since 2019

Education and Training Committee

Co-Author, OWASP Software Assurance Maturity Model (SAMM)

Leader, OWASP State of AppSec Survey Project

A Software Assurance Program

Purpose

To provide confidence to all stakeholders that software products are free from vulnerabilities – intentional or unintentional – and that those products reliably function as intended

Goals

- Foster “Secure by Design” culture
- Improve code-level security of delivered software
- Focus on threats and risks in defining requirements
- Increase development efficiency
- Educate developers in best practices
- Assess and improve program maturity

How can we find security issues in our applications and systems?

Some Approaches

- Static analysis of code
- Dynamic testing
- Penetration testing
- Production bug reports
- Incident response

“Wouldn’t it be better to find security issues before you write or deploy a line of code?”

--Adam Shostack

The Five W's of Threat Modelling

WHY Threat Model?

- Improve efficiency
  - Think about security issues early
  - Invest effort more wisely
- Understand requirements better
  - Bring security and development together
  - Shared, maintainable, understanding of risks
- Avoid writing security issues into our code
  - Avoid costs of rework
- Improve stakeholder confidence
- And increasingly…because the regulator said so

Terms of Reference

- **Asset –** Anything we need to protect
- **Threat –** Anything that could let someone or something obtain, damage, or destroy an asset, if we fail to protect against it
- **Vulnerability –** A weakness or gap in our protection efforts
- **Risk –** The potential for loss, damage, or destruction of an ***asset***, due to a ***threat***'s having successfully exploited a ***vulnerability***

WHAT Is a Threat Model?

A **Threat Model** is a conceptual representation of a *system*, the *threats* to it that have been identified, and the *controls* that will be implemented to protect it.

Key considerations:

- To be useful ***to more than one person***, the model must be captured in a persistent, shareable form
- To ***remain*** useful, the model must be kept up-to-date and aligned with the real system

WHAT Should Be in a Threat Model?

- Description of the system
- List of assumptions
- List of threats
- Decision on how to address each threat
- Verification Approach
- Validation approach

WHO Should Create the Threat Model?

- All system stakeholders should take part
  - Security ”experts” play advisory role *only*

- Assign lifecycle roles:
  - Owner (Accountable)
  - Maintainer (Responsible)

WHEN to Create the Threat Model?

“The best time to plant a tree was 20 years ago. The second-best time is now.”

- Start as early as possible
- Existing system, without a Threat Model?
  - Start NOW
  - Use Incremental Threat Modelling approach (Irene Michlin)

WHEN to Update the Threat Model?

My recommendation:

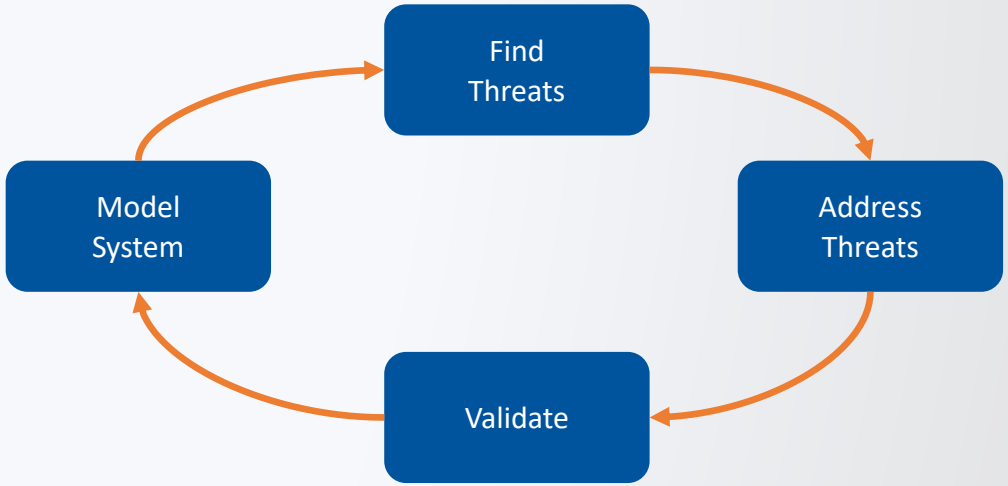
- Review Threat Model every update cycle
  - Do the proposed changes affect the model?
  - If ‘yes,’ include model update efforts *in the cycle*
- OK…but what’s an “update cycle”?
  - Agile/iterative: Each Sprint, or each Release
  - Waterfall: Each change order

WHERE Should the Threat Model Live?

- With other project/product documentation
  - Well-known location, with reliable backups
  - Ideally, place under revision control
  - Align model versions with product versions

HOW Do I Build a Threat Model?

DiLeo's Seven Questions
Expanding on Shostack's *Four*



1. What are we building? (Create DFD)
2. What can go wrong? (Identify threats)

3a. What *could* we do about it? (Identify *possible* mitigations)

3b. What *will* we do about it? (Select *planned* mitigations)

3c. Have all ***residual risks*** been accepted?

If not, repeat #3b, selecting additional/better mitigations, until "Yes"

4a. How will we know the mitigations work? (Verification)

4b. Is our model correct? (Validation)

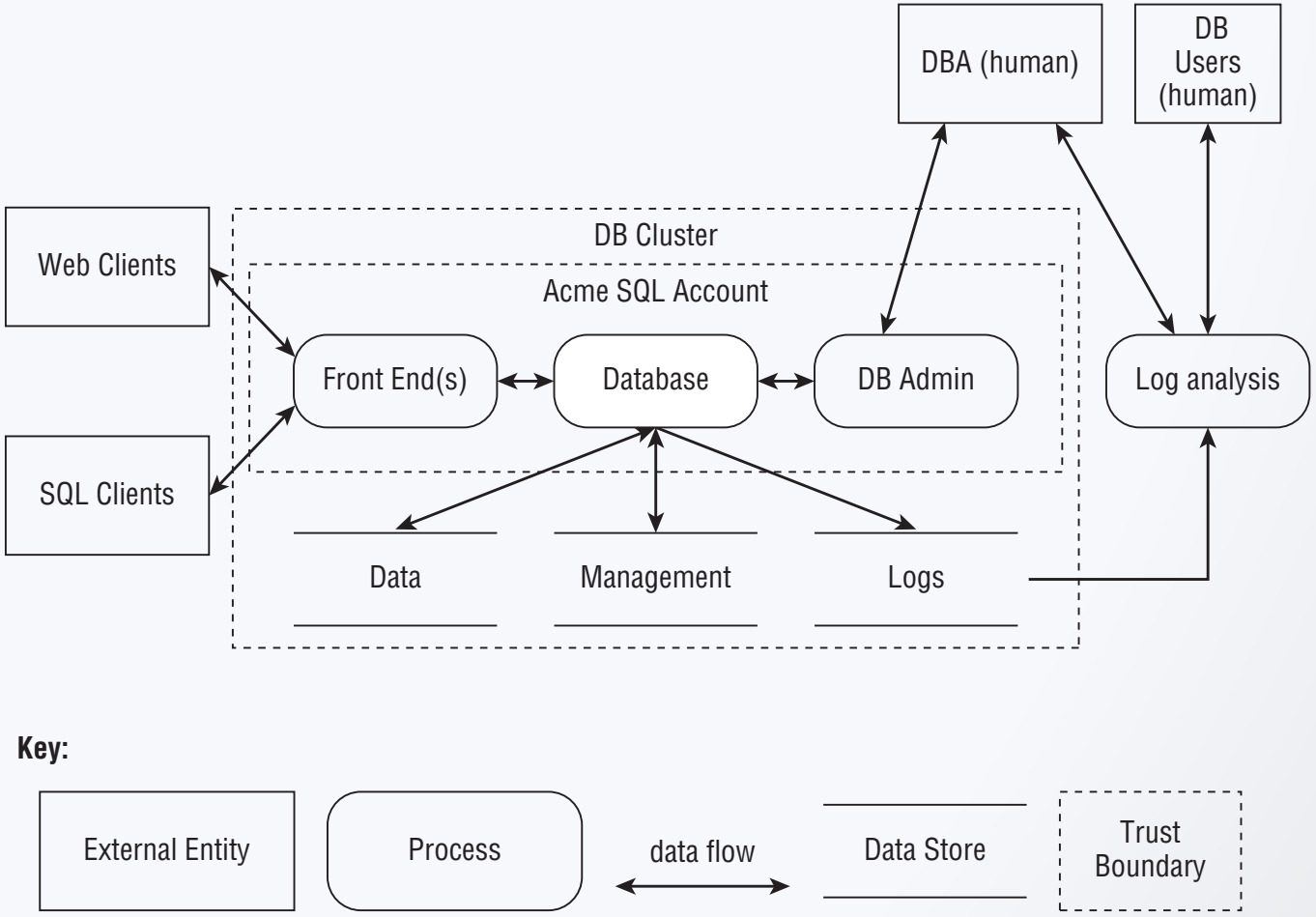
What Are We Building?

- Create a model of the system
  - Technology used
  - Data stored and processed
  - Software created or used
- A model abstracts away the details so you can look at the whole
  - Diagramming is a key approach
  - Whiteboard diagrams are a great way to start

DFD (Data Flow Diagram)

- Around since the early 70s
  - Simple: easy to learn, easy to draw
  - Threats often follow data
- Abstracts programs into:
  - Processes: Your code
  - Data Stores: Files, databases, shared memory
  - Data Flows: Connect processes to other elements
  - External Entities: Everything but your code & data
    Includes people and cloud software
  - Trust Boundaries

Data Flow Diagram (Example)



What Can Go Wrong?
Identifying Threats – Option 1

When Threat Modelling 'Manually'

- **STRIDE** mnemonic
  - Spoofing
  - Tampering
  - Repudiation
  - Information Disclosure
  - Denial of Service
  - Elevation of Privilege

What Can Go Wrong?
Identifying Threats – Option 2

When Using 'Automated' Threat Modelling Tools

- Built-in Component Libraries
- Pre-identified Threats, associated with each Component
- Review identified threats, confirm applicability

What *Could* We Do about It?
Identifying Possible Mitigations

For each identified threat, we could:

- Remove it (Avoid the risk)
- Implement controls (Mitigate the risk)
  - Technical
    - Preferred: Well-known commercial/open-source solutions
    - If you must, Custom mitigations – “roll your own” security
  - Non-technical
    - Physical protections
    - Administrative processes
- Do nothing (Accept the risk)
- Make it someone else’s problem (Transfer the risk)

Custom Mitigations
Proceed with Caution!

- Sometimes standard approaches don't work for your situation
- Custom (home-grown) mitigation is an option
- Easy to get custom mitigation wrong
  - No broader community supporting it
- Testing is difficult and expensive
- *And*…it's yours to maintain "forever"

What *Will* We Do about It?
Selecting Mitigations

Two-stage process:

1. For all mitigations that are easy, mandatory, and/or standard, *just do them*
   - Mark all relevant threats as mitigated
2. For all remaining threats:
   - Assess risk to system if *not* mitigated
   - Review candidate mitigations – cost vs. benefit
   - Select mitigation(s) to apply…or accept risk

Verifying Mitigations

- Each selected mitigation *can* be tested or verified
  - Functional security features: Positive and negative test cases, Regression tests
  - Security Specifications: Verification checklists
- In a test-driven development (TDD) methodology, use threat model as a source for tests
- Automate when possible – Test manually, only if you must

Validating Our Modelling Work

- Check software model/reality conformance
- Have all selected mitigations been implemented and tested/verified?
- Are all assumptions still valid?

Threat Modelling Tools

- You don't *necessarily* need a tool, when starting out
  - Whiteboards and sticky notes
  - Visio, Lucid Charts, Draw.io
- Free tools (e.g., OWASP Threat Dragon) – often enough for small portfolios, and for pilots
- Commercial tools provide economic *benefits* for larger portfolios (15 or more systems modelled)
  - Forrester Total Economic Impact Study (IriusRisk-sponsored)

Getting Started with Threat Modelling

Staged process:

- Define process, select and acquire tools
- Awareness and Education
- Add to AppSec policy / standards (not mandatory *yet*)
- *Carefully*-chosen pilot projects
- Just-in-Time training
- Success Managers (Security Champions)
- Celebrate successes, publish lessons learned
- Phased roll-out
- **THEN**…Make Threat Modelling mandatory

Questions?

Connect / Reach out

- Email:
  - Day job: *jdileo@iriusrisk.com*
  - "Other job": *john.dileo@owasp.org*
- Twitter: *@gr4ybeard*
- LinkedIn: *john-dileo*
- OWASP Slack *https://owasp.org/slack/invite*

OWASP

TM