TOREON

Whiteboard Hacking / Hands-on Threat Modeling

Introduction

Sebastien Deleersnyder

- 5 years developer experience
- 15+ years information security experience
- Application security consultant Toreon

- Belgian OWASP chapter founder
- OWASP volunteer
- www.owasp.org

- Co-founder www.BruCON.org

Threat modeling introduction

- Threat modeling in a secure development lifecycle
- What is threat modelling?
- Why threat modeling?
- Threat modeling stages
- Diagrams
- Identify threats
- Addressing threats
- Document a threat model

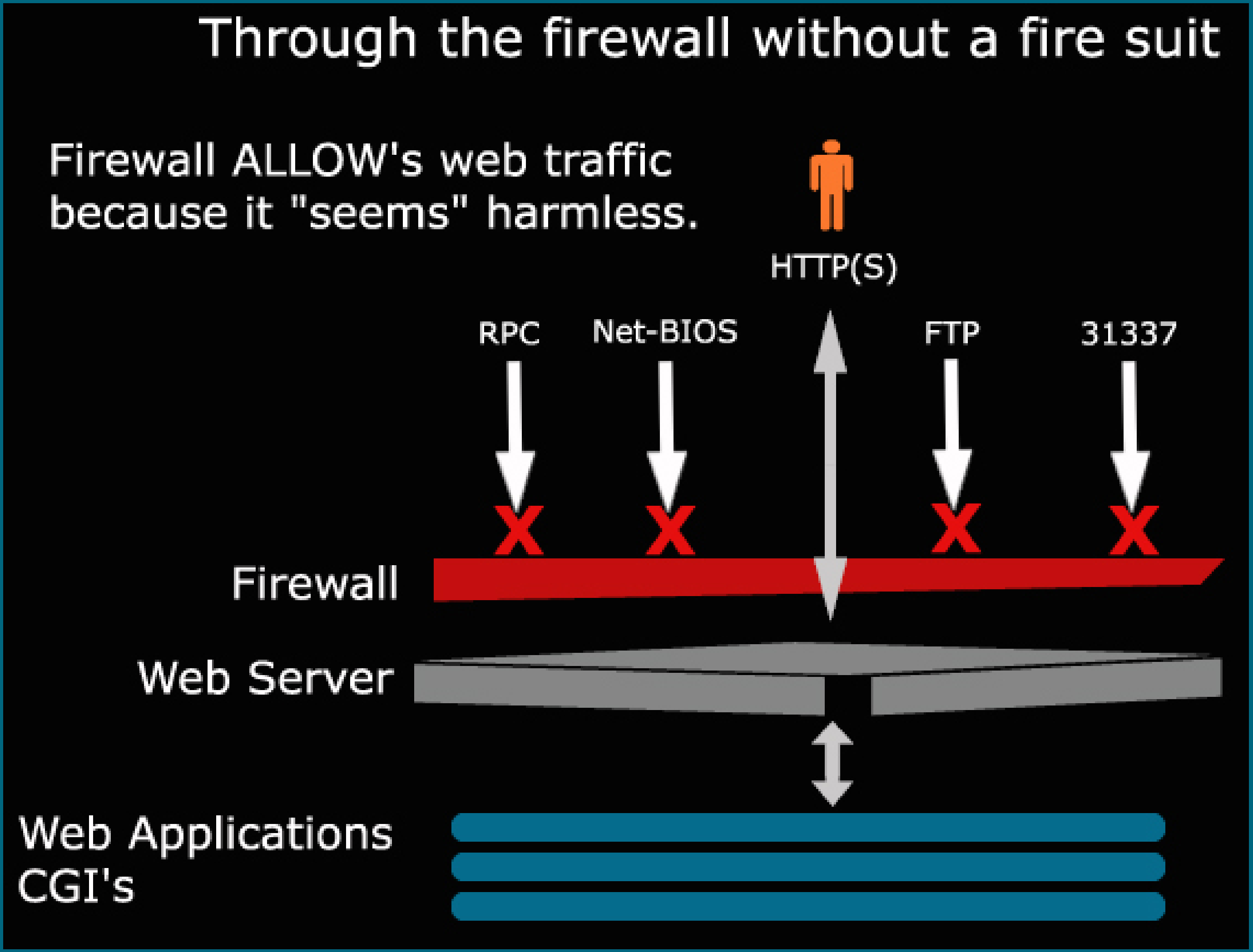
TOREON

Myth

Myth: we are secure because we have a firewall

75% of Internet Vulnerabilities are at Web Application Layer *

*Gartner Group (2002 report)



Through the firewall without a fire suit

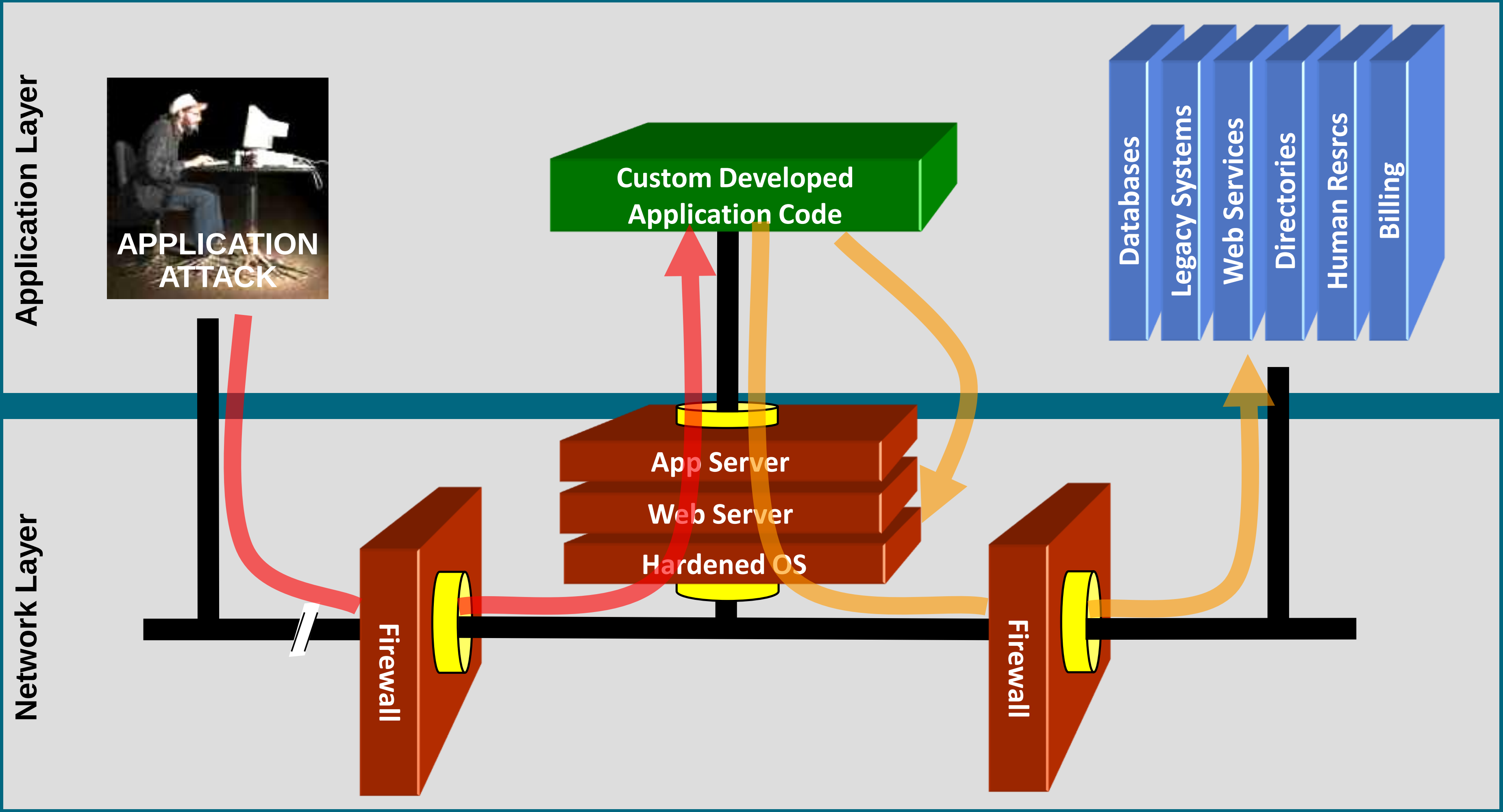
Firewall ALLOW's web traffic
because it "seems" harmless.

Source: Jeremiah Grossman, Black Hat 2001

OWASP Top Ten (2017 Edition)

- A1: Injection
- A2: Broken Authentication
- A3: Sensitive Data Exposure
- A4: XML External Entities (XXE)
- A5: Broken Access Control
- A6: Security Misconfiguration
- A7: Cross Site Scripting (XSS)
- A8: Insecure Deserialization
- A9: Using Known Vulnerable Components
- A10: Insufficient Logging & Monitoring

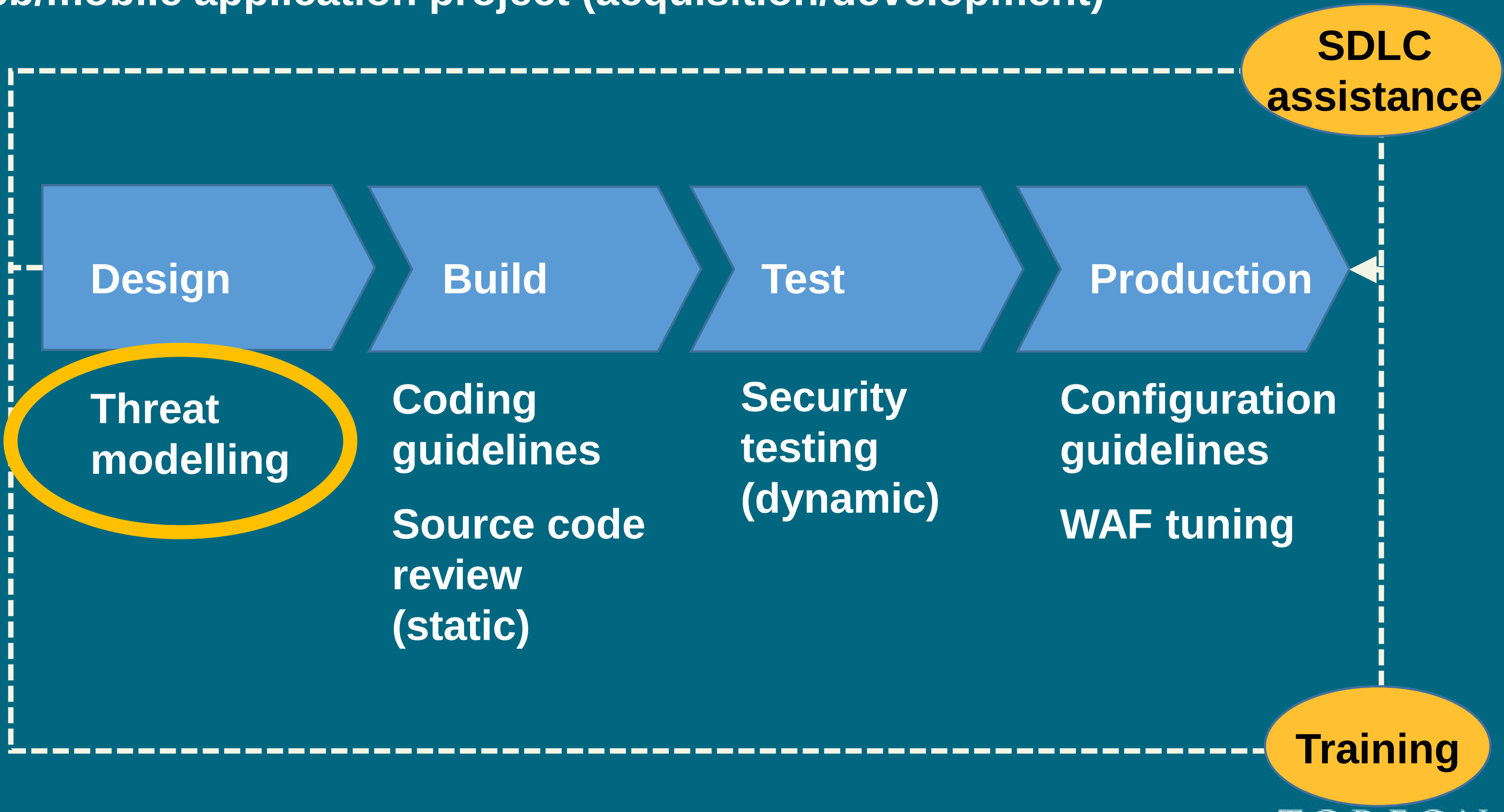
Your security "perimeter" has huge holes at the application layer



You can't use network layer protection (firewall, SSL, IDS, hardening) to stop or detect application layer attacks

Secure development lifecycle

Web/mobile application project (acquisition/development)



SDLC assistance

Design

Threat modelling

Build

Coding guidelines

Source code review (static)

Test

Security testing (dynamic)

Production

Configuration guidelines

WAF tuning

Training

TOREON

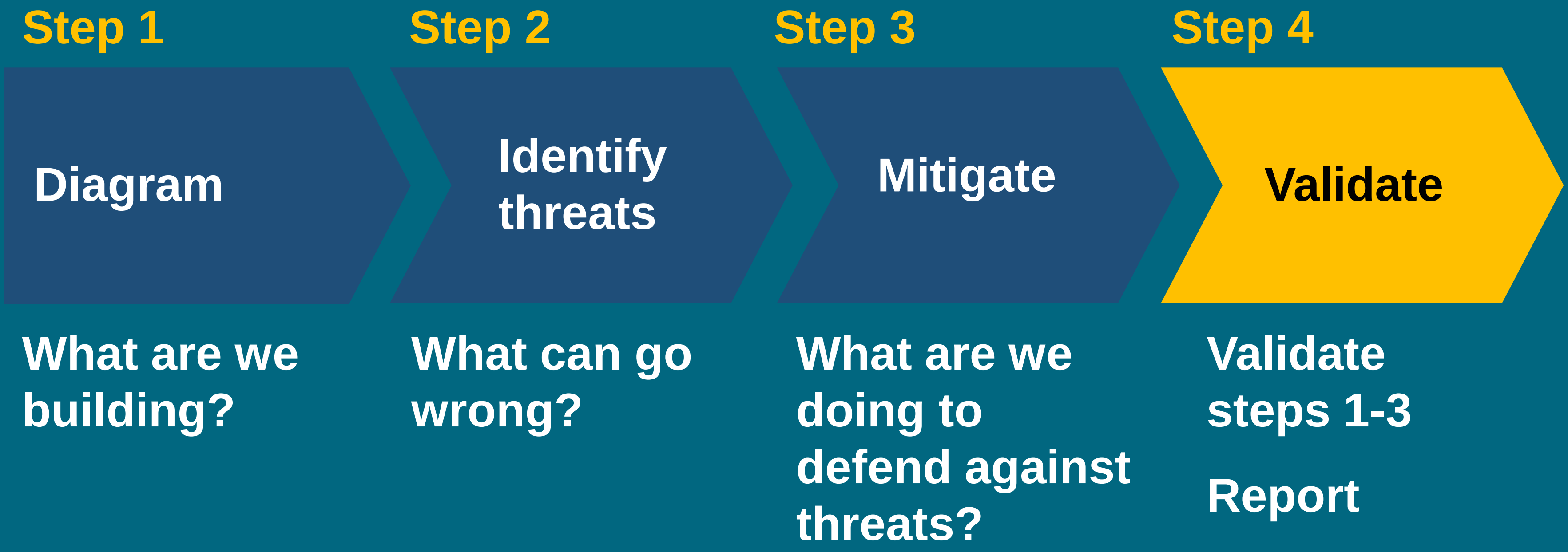
Threat modeling

- Threat modelling is the activity of identifying and managing application risks
- Threat modelling is also known as Architectural Risk Analysis

Why threat modeling?

- Prevent security design flaws when there’s time to fix them
- Select mitigation strategy and techniques based on identified, documented and rated threats.
- Identify & address greatest risks
- Ability to prioritize development efforts within a project team based on risk weighting
- Increased risk awareness and understanding
- Mechanism for reaching consensus and better trade-off decisions
- Means for communicating results
- Cost justification and support for needed controls
- Artifacts to document due diligence for each software project

Threat modelling stages



Step 1

Diagram

What are we building?

Step 2

Identify threats

What can go wrong?

Step 3

Mitigate

What are we doing to defend against threats?

Step 4

Validate

Validate steps 1-3

Report

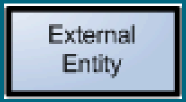
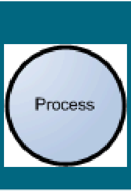
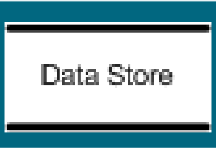
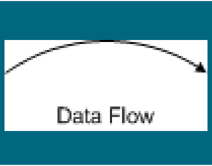
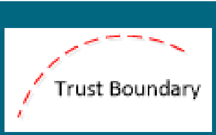
Diagrams

- Define scope
- Good understanding context / objectives
- Understand how the software works
- Who interacts with the software?
- With Data Flow Diagrams, Sequence Diagrams, State diagrams ...
- Identify attack surfaces
- Foundation for threat analysis

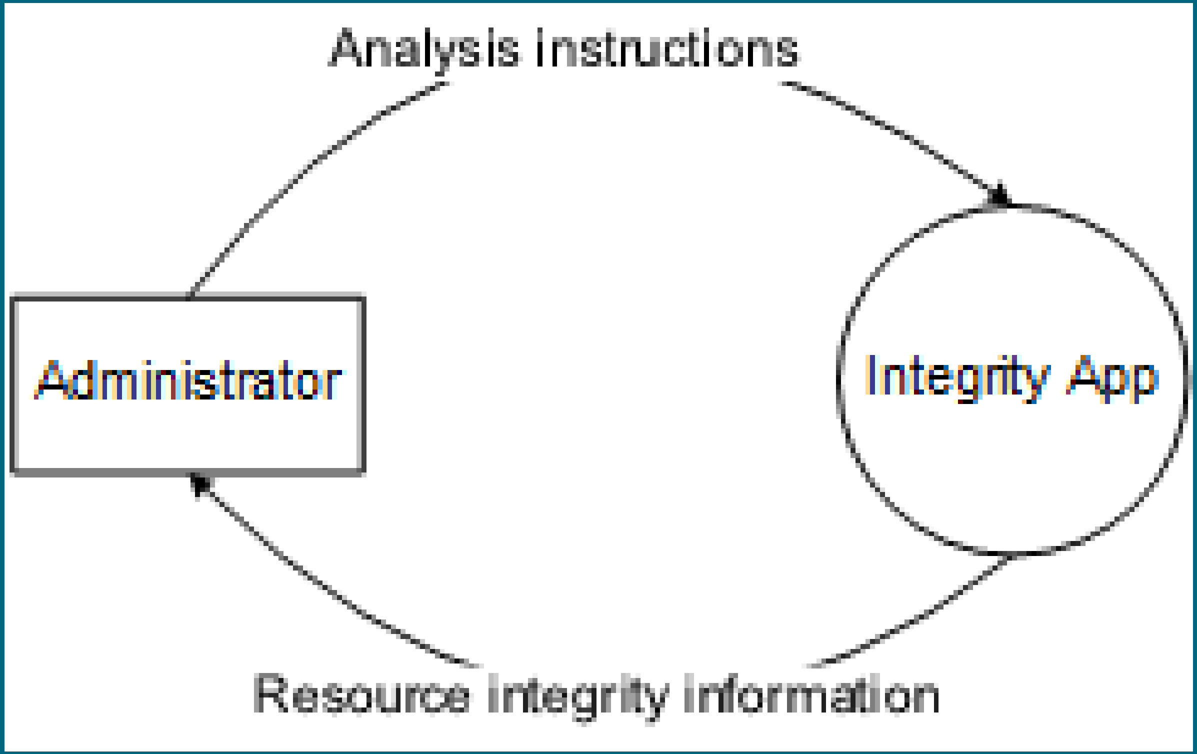
Diagramming

- Use DFDs (Data Flow Diagrams)
  - Include processes, data stores, data flows
  - Include trust boundaries
  - Diagrams per scenario may be helpful
- Update diagrams as web application changes
- Enumerate assumptions, dependencies
- Number everything (if manual)

DFD Basics

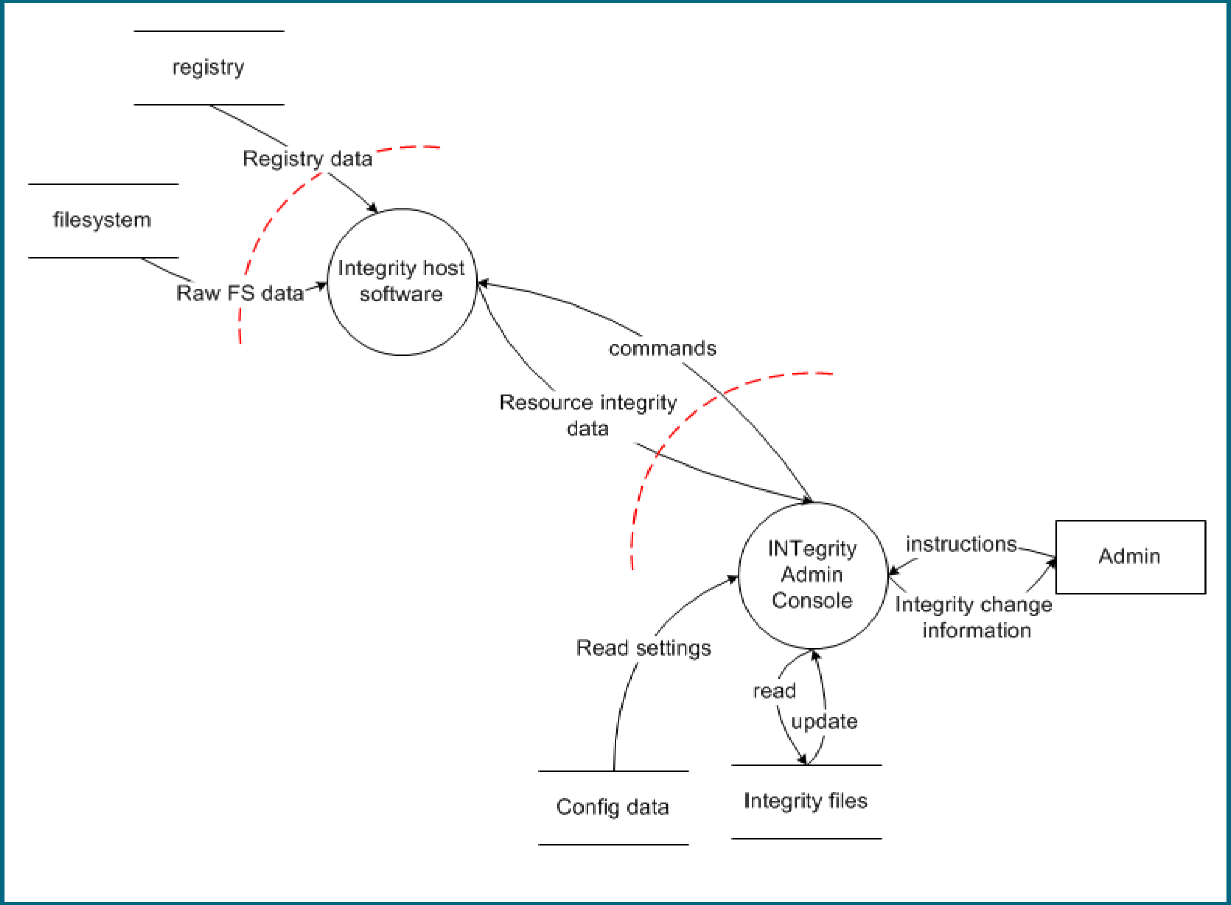
Symbol		Description
External Entity		• Represents entities outside the application that interact with the application via an entry point
Process		• Represents tasks that handle data within the application; tasks may process data or perform actions based on the data
Data Store		• Represents locations where data is stored; data stores do not modify data, they only store it.
Data Flow		• Represents data movement within applications; the arrow tells the direction of data movement
Trust Boundary		• Represents the change of trust levels as data flows through the application

Context diagram



TOREON

Level 1 Diagram



Identify threats

- Based on diagrams
- STRIDE analysis
- Focus on identifying threats

STRIDE

Spoofing	Can an attacker gain access using a false identity?
Tampering	Can an attacker modify data as it flows through the application?
Repudiation	If an attacker denies doing something, can we prove he did it?
Information Disclosure	Can an attacker gain access to private or potentially injurious data?
Denial of Service	Can an attacker crash or reduce the availability of the system?
Elevation of Privilege	Can an attacker assume the identity of a privileged user?

Apply STRIDE Threats to Each Element

Apply the relevant parts of STRIDE to each item on the diagram

- External Entity – S, T
- Process – S, T, R, I, D, E
- Data store, data flow – T, I, D
  - Data stores that are logs – T, I, D, and R

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

This is why you number things

Example

	Admin		>		Admin Console	
	Mitigations	Vulnerabilities	Mitigations	Vulnerabilities	Mitigations	Vulnerabilities
S	User/PW				SSL Cert	
T			SSL			
R		No audit log				No Audit log
I			SSL			
D						
E						No Access Control

Addressing threats

- Cover all threats
- Identify controls already in place
- Handle threats not (completely) covered

Addressing each threat

Mitigation patterns	
Authentication	Mitigating spoofing
Integrity	Mitigating tampering
Non-repudiation	Mitigating repudiation
Confidentiality	Mitigating information disclosure
Availability	Mitigating denial of service
Authorisation	Mitigating elevation of privilege
Hands-on	Threat mitigation OAuth scenarios for web and mobile applications

Mitigation patterns

- Apply appropriate secure design strategies
- Leverage proven best practices
- Reuse organisation security services, e.g.,
  - Single-Sign-On, Log Server
- Do not reinvent the wheel

For threats not (completely) covered

- Redesign to eliminate
- Apply standard mitigations
- Create new mitigations
- Accept vulnerability in design

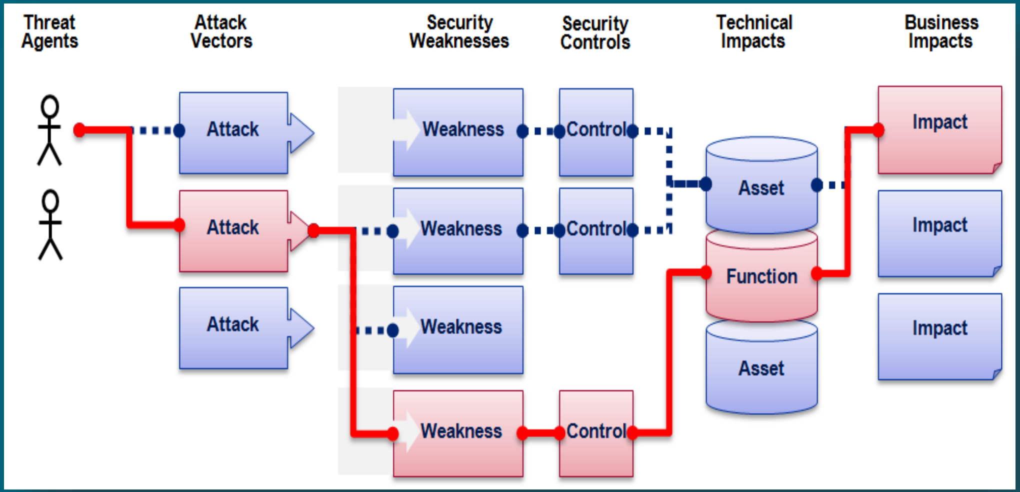
Risk-based Threat Management

“The only truly secure system is one that is powered off, cast in a block of concrete, and sealed in a lead-lined room with armed guards - and even then I have my doubts. ”

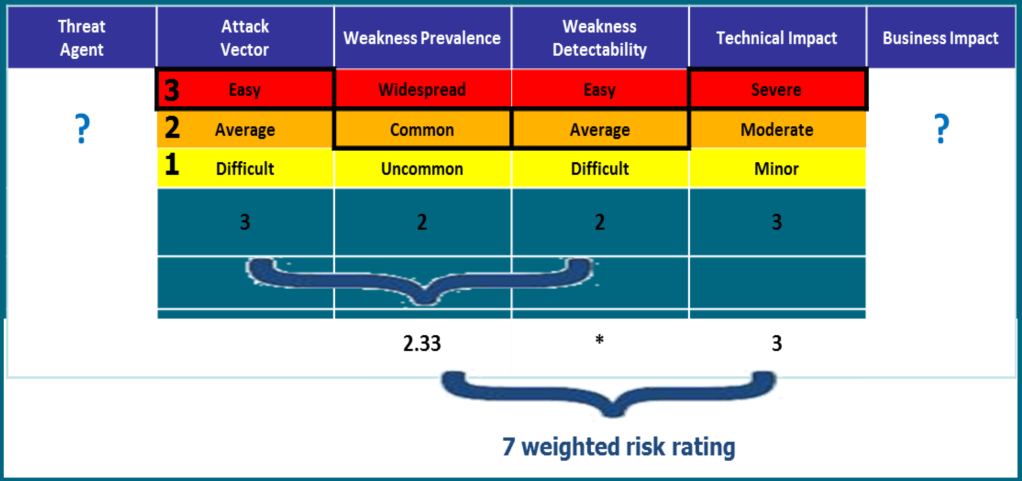
Prof Gene Spafford

Source: http://spaf.cerias.purdue.edu/quotes.html

OWASP risk rating



Injection Example



Threat Agent		Attack Vector	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact
?	3	Easy	Widespread	Easy	Severe	?
	2	Average	Common	Average	Moderate	
	1	Difficult	Uncommon	Difficult	Minor	
		3	2	2	3	

2.33 * 3

7 weighted risk rating

Example

Threat	Description	Vector	Prevalence	Detectability	Impact	Rating	Risk
TH – 01	• Credentials can be brute forced	2	2	3	3	7.00	High
TH – 02	• No security rules on password	2	2	2	3	6.00	Medium
TH – 03	• No SSL for Android App	2	3	2	2	4.67	Medium
TH – 04	• No SSL active for admin module	1	2	3	2	4.00	Medium
TH – 05	• No accountability of Drupal updates	3	2	2	1	2.33	Low
TH – 06	• API calls can be tampered with	1	1	1	2	2.00	Low
TH – 07	• Fake IDs can be used	1	1	1	2	2.00	Low

Low: 1-3, Medium: 4-6, High: 7-9

Communicate Your Threat Model

You cannot just “write and throw out” a security document

- Recipients often won’t understand it

Communicate Your Threat Model

To increase adoption

- Present the results to the audience, in person
- Discuss the countermeasures – cost vs. impact
- Complete the threat model with a proposed action list that you know is acceptable

Typical audience

Architects

- Should integrate the proposition to update the design

Developers

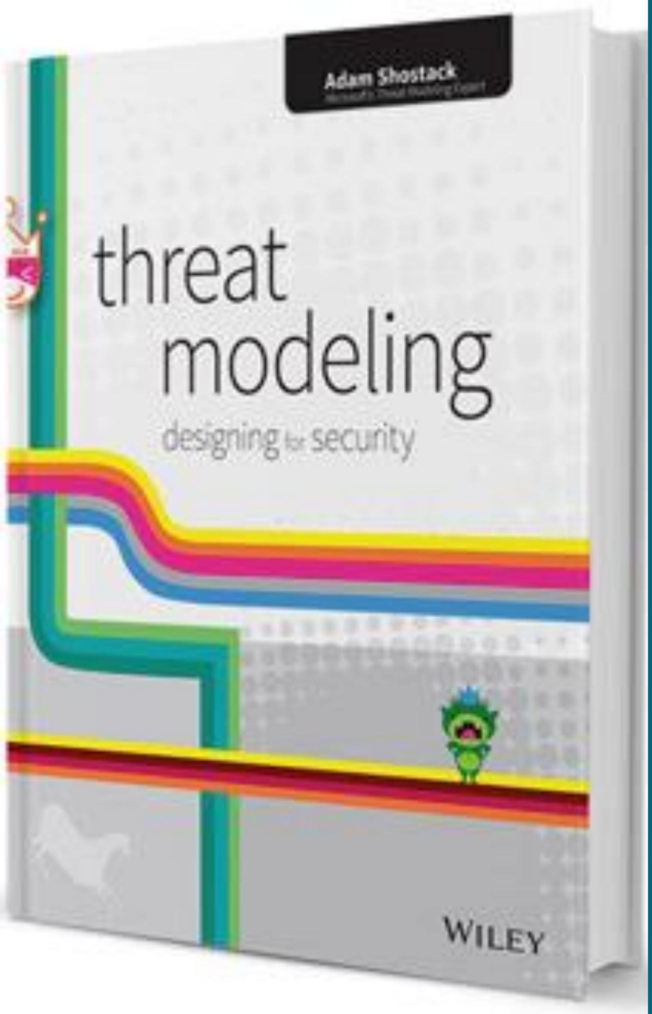
- Should benefit from the model transparently, through updated specification

Security testing team

- Now know precisely what to test!

Software editor

- If you are acquiring software, you can add the threat model to the software acceptance procedure


Adam Shostack
threat modeling
designing for security
WILEY

That’s All Folks

You can contact me through

- Toreon seba@toreon.com
- OWASP seba@owasp.org
- Twitter @SebaDele
- OWASP TM Slack channel

Hands-on Diagramming

Review the B2B case “ACME Hotel Bookings (AHB)” - diagram B2B web and mobile applications, sharing the same REST backend

- Create the data flow diagram with trust boundaries of the AHB Booking system (30 min)
- Perform one STRIDE analysis on the “customer to website” trust boundary, assume the following mitigations:
  - Customers login with Facebook or user name and password
  - The website uses SSL/TLS
  - No other protections are foreseen in the design. (30 min)

Trainer represents “the AHB customer” for your questions