Code Mash 2020

Hands-On Threat Modeling Workshop

January 7, 2020
Robert Hurlbut
@RobertHurlbut

BANK OF AMERICA

Agenda

Overview (30 minutes)

- Introductions
- What / Why Threat Modeling?

Getting Started (1 hour 30 minutes)

- Threat Modeling Process
- Hands-On Exercises using Whiteboards

Break (5 minutes)

Using Tools (1 hour 25 minutes)

- Threat Modeling Tools and Card Games
- Hands-on Exercises / Labs

What’s next? (30 minutes)

Who am I?

Robert Hurlbut

SVP, Threat Modeling Architect / Lead

Cyber Security Technology

Global Information Security

Bank of America

Legal Disclaimer

- Neither Bank of America nor its affiliates provide information security or information technology (IT) consulting services.
- This material is provided "as is,“ with no guarantee of completeness, accuracy, timeliness or of the results obtained from the use of this material, and without warranty of any kind, expressed or implied, including, but not limited to warranties of performance, quality and fitness for a particular purpose.
- This material should be regarded as general information on information security and IT considerations and is not intended to provide specific information security or IT advice nor is it any substitute for your own independent investigations.
- If you have questions regarding your particular IT system or information security concerns, please contact your IT or information security advisor.

Pre-Compiler Materials

https://github.com/rhurlbut/CodeMash2020

What is Threat Modeling?

What is threat modeling?

Something we all do in our personal lives …
… when we lock our doors to our house
… when we lock the windows

… when we lock the doors to our car

What is threat modeling?, continued

When we ...

think ahead on what could go wrong

(i.e. the " what if" questions),

weigh the risks,

and act accordingly ...

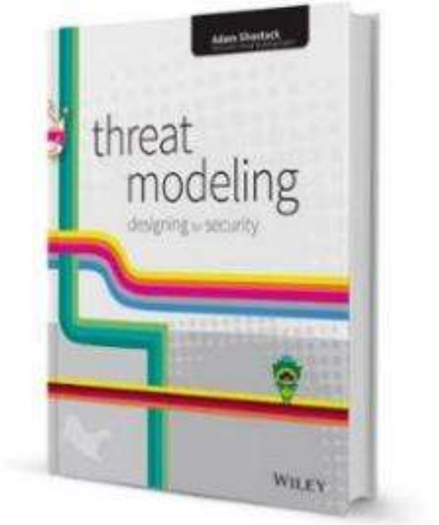
... we are "**threat modeling**"

What is threat modeling?, continued

Threat Modeling: Designing for Security
by Adam Shostack
https://threatmodelingbook.com/

Asks four questions:

1. What are you working on?
2. What could go wrong?
3. What are you going to do about it?
4. Did you do a good job of analysis?



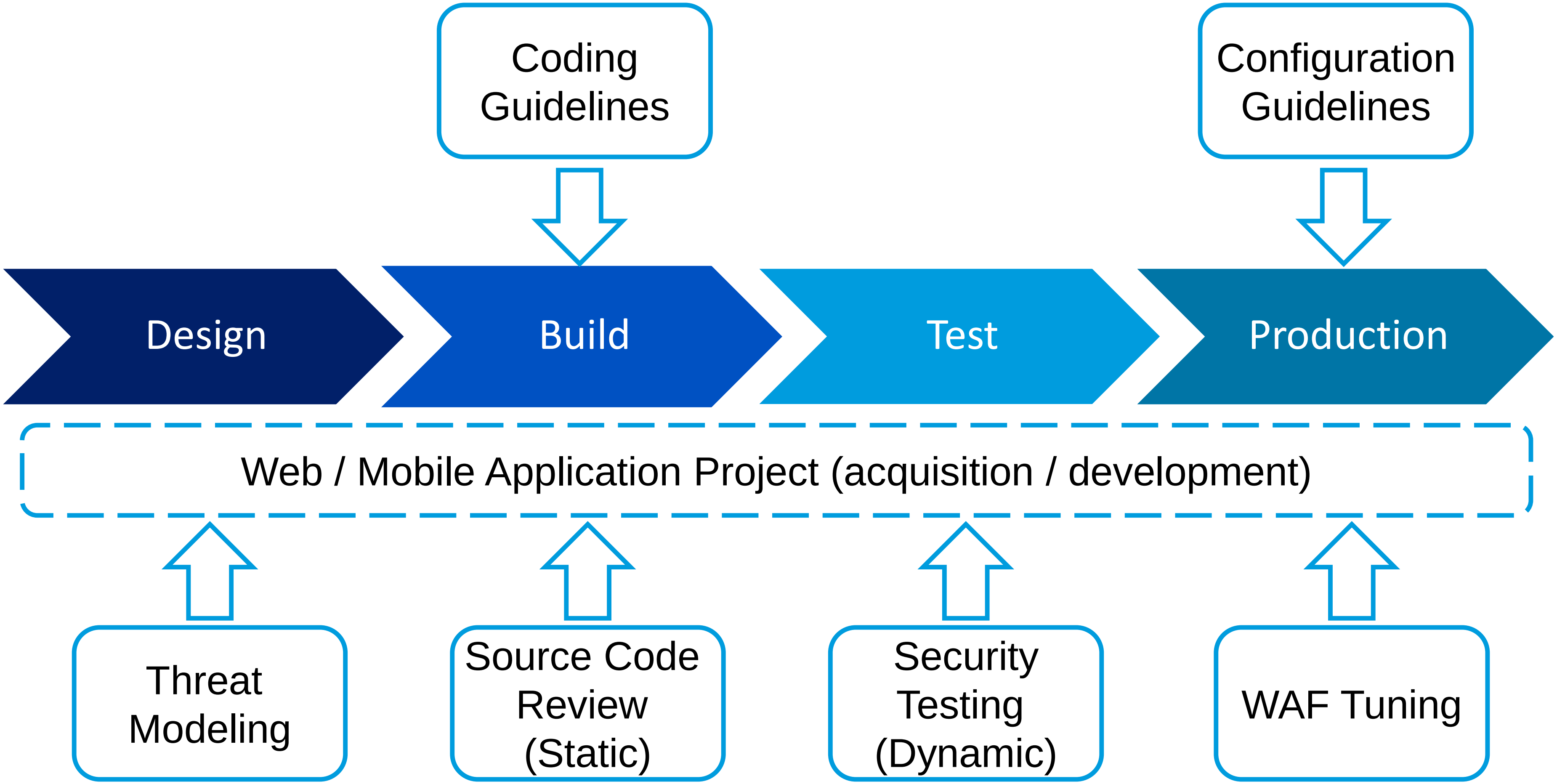
What is threat modeling?, continued

Threat modeling is:

Process of understanding
your system and potential
threats against your system
and related countermeasures

i.e. ***Critical Thinking*** about Security

Secure development lifecycle



Coding Guidelines

Configuration Guidelines

Design

Build

Test

Production

Web / Mobile Application Project (acquisition / development)

Threat Modeling

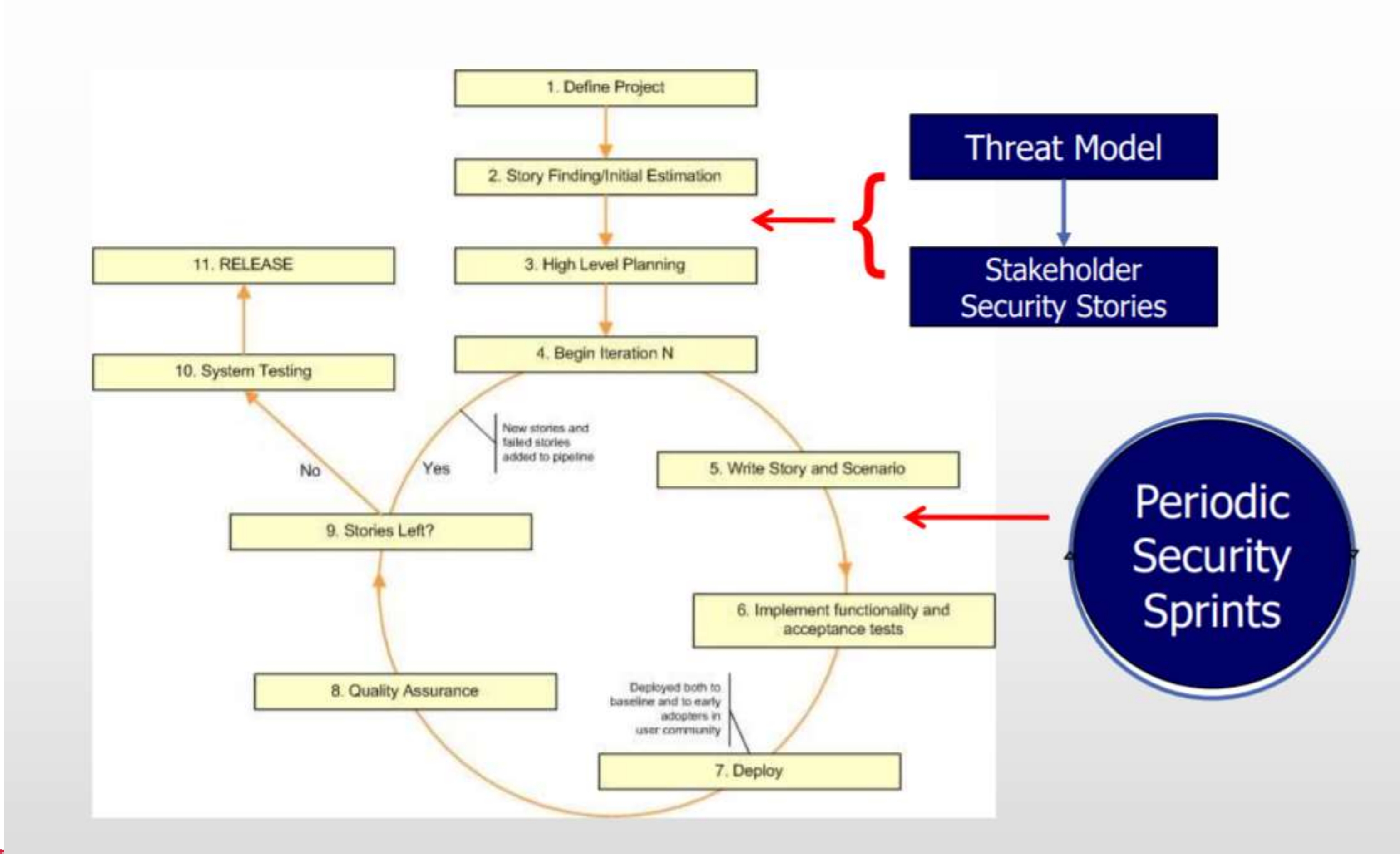
Source Code Review (Static)

Security Testing (Dynamic)

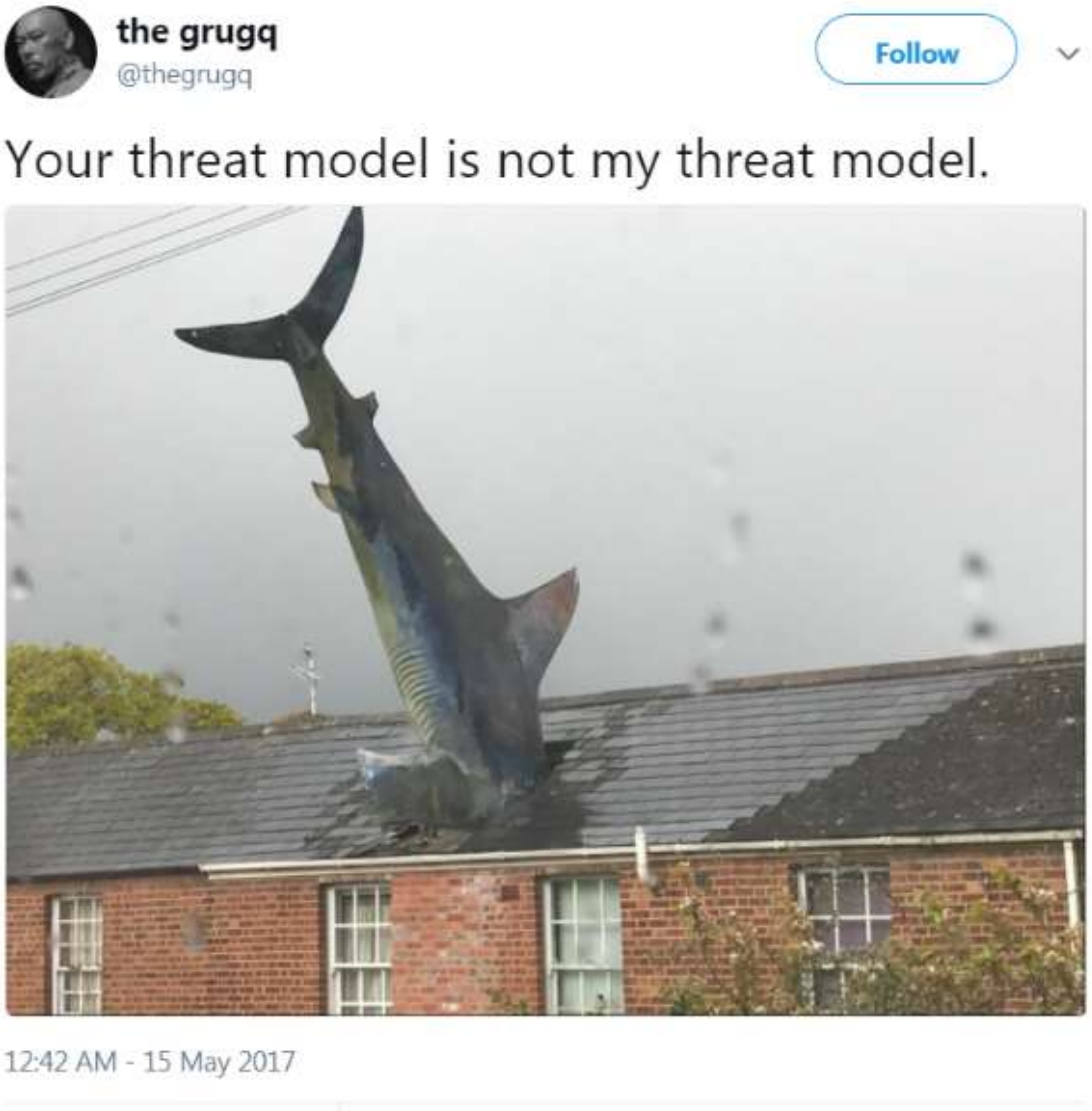
WAF Tuning

Agile / DevOps – Incremental Threat Modeling

User Stories, Attacker Stories


1. Define Project
2. Story Finding/Initial Estimation
3. High Level Planning
4. Begin Iteration N
5. Write Story and Scenario
6. Implement functionality and acceptance tests
7. Deploy
8. Quality Assurance
9. Stories Left?
10. System Testing
11. RELEASE
Yes
No
New stories and failed stories added to pipeline
Deployed both to baseline and to early adopters in user community
Threat Model
Stakeholder Security Stories
Periodic Security Sprints

Threat models can vary – and that's ok



the grugq
@thegrugq

Your threat model is not my threat model.

12:42 AM - 15 May 2017

Why Threat Modeling?

Why perform threat modeling?

Get team on same page with shared vision on security

Prevent security design flaws

Identify and address greatest risks

Prioritize development efforts based on risk weighting

Increased risk awareness and understanding

Cost justification and support for needed controls

Example Secure Design Issue:
How to secure data in the cloud?

Storage?
Accessed?
Monitored?
Configured properly?

Threat Modeling helps us focus on these questions and answers to lead to secure design

Common data breach problem

Misconfigured AWS S3 Buckets

Impacted in 2017-2018 *:

- FedEx
- GoDaddy
- Accenture
- Verizon
- American voter data (198 million American voters)
- National Credit Federation
- Booz Allen Hampton
- Dow Jones
- Keeper and Blur (password managers)

* https://www.zdnet.com/article/security-lapse-exposes-198-million-united-states-voter-records/

Approaches to Threat Modeling

Asset-centric

Software-centric

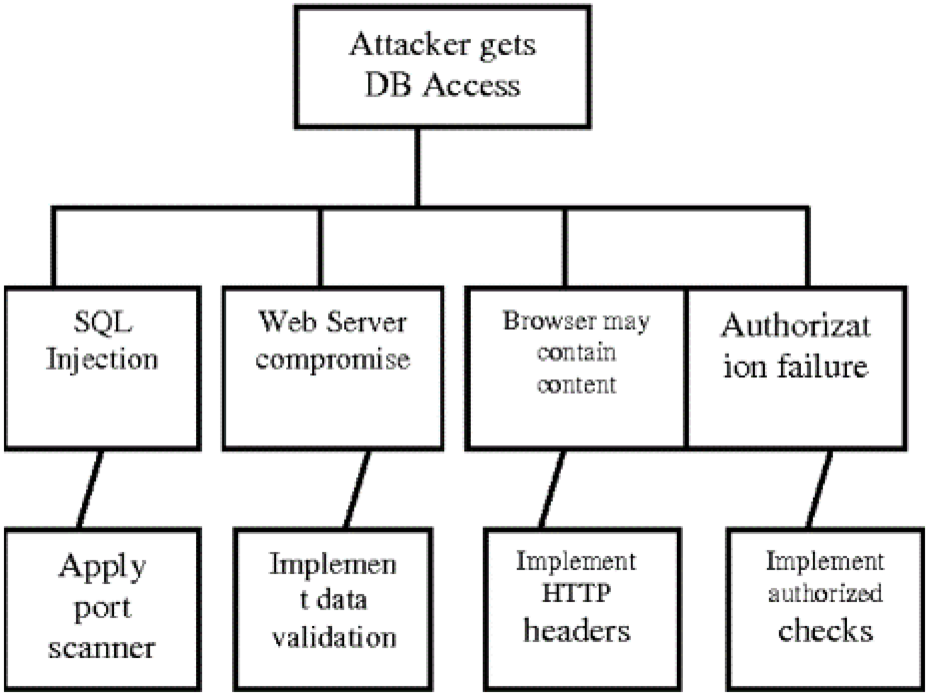
Attacker-centric

Approaches to Threat Modeling – Asset-centric

Assets

Things of value. For example: Databases which may contain credit card data, personal Identifiable Information (PII), etc.

Attack trees

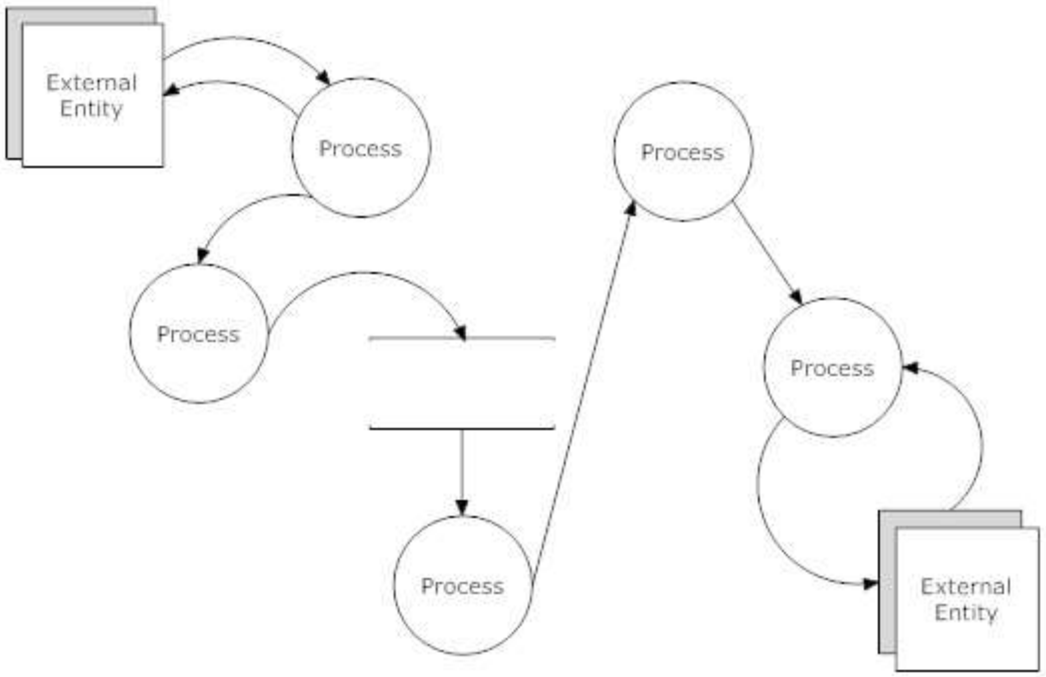


Approaches to Threat Modeling – Software-centric

Secure Design

Understanding secure activity within an architecture

Data Flow Diagrams (DFDs)



Approaches to Threat Modeling – Attacker-centric

Profiles	Patterns
Script Kiddie	Copies scripts – tries anything
Hacktivist	Political agenda – deface website
Nation-state attacker	Money, intellectual property theft - phishing

Threat Modeling your House

Asset-centric

Family, irreplaceable photos, valuable artwork

Software-centric

Physical features (basement door, porch)

Attacker-centric

Who might break in, current security system

What is threat modeling?

Threat model includes:

understanding of system,
identified threat(s),
proposed mitigation(s),
priorities by risk

Threat Modeling:
Getting Started

Typical Threat Modeling Session

Domain Knowledge

Team

Business / Technical Goals

Focused

Important: Be honest, leave ego at the door,
no blaming!

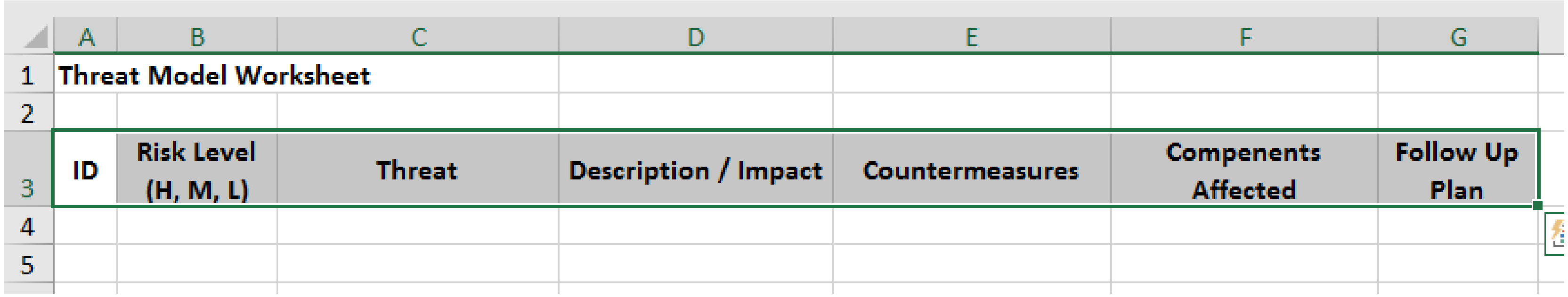
Simple Tools

Whiteboard

Visio (or equivalent) – diagraming

Word (or equivalent) / Excel (or equivalent) - documenting threats / mitigations

Threat Model Sample Worksheet



	A	B	C	D	E	F	G
1	Threat Model Worksheet						
2							
3	**ID**	**Risk Level (H, M, L)**	**Threat**	**Description / Impact**	**Countermeasures**	**Compenents Affected**	**Follow Up Plan**
4							
5							

Other Tools

Tool	Cost	Platforms
MS Threat Modeling Tool	Free	Windows OS Install only
ThreatModeler	Paid	Web Based
IriusRisk	Paid	Web Based
OWASP Threat Dragon	Free	Web Based / Windows, Mac, Linux installs
Draw.IO	Free	Web Based / Windows, Mac, Linux installs

IEEE Computer Society's Center for Secure Design (2015) *

AVOIDING THE TOP 10 SOFTWARE SECURITY DESIGN FLAWS

* http://www.computer.org/cms/CYBSI/docs/Top-10-Flaws.pdf

Avoiding the Top 10 Software Security Design Flaws: Bugs vs Flaws

Bug – an implementation-level software problem

Flaw – deeper level problem - result of mistake or oversight at design level

In Threat Modeling, we try to identify design flaws to improve secure design

Avoiding the Top 10 Software Security Design Flaws: Bugs vs Flaws

Security coding bugs

- Coding errors
- Requires developers understanding secure coding
- Can be automated
- Patching less costly in production

Security design flaws

- Errors in design, security requirements, architecture
- Need contextual knowledge
- No automation
- Costly to change in production

Threat Modeling Process

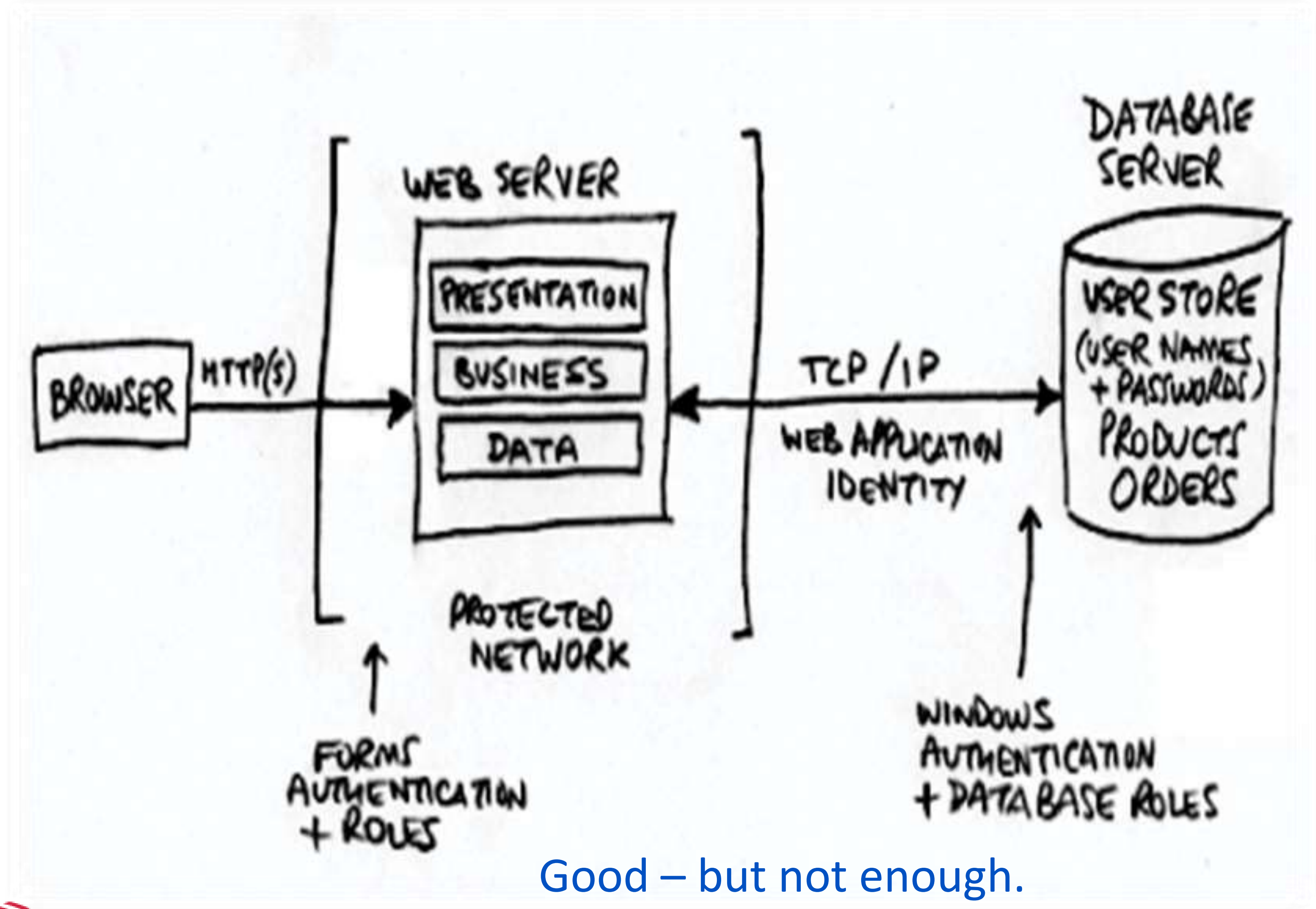
Threat Modeling Process

1. Diagram / understand your system and data flows
2. Identify threats through answers to questions
3. Determine mitigations and risks
4. Follow through

Threat Modeling Process

1. **Diagram / understand your system and data flows**
2. Identify threats through answers to questions
3. Determine mitigations and risks
4. Follow through

Draw a picture

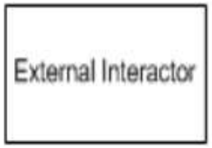
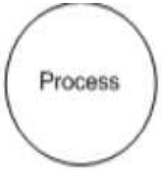
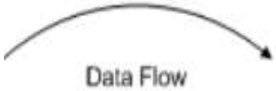
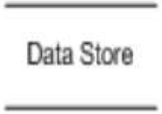
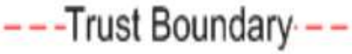


Good – but not enough.
Let's explore further.

1. Understand the system - create a Data Flow Diagram (DFD)

Decomposes the system into a series of external interactors, processes, data stores, and data flows.

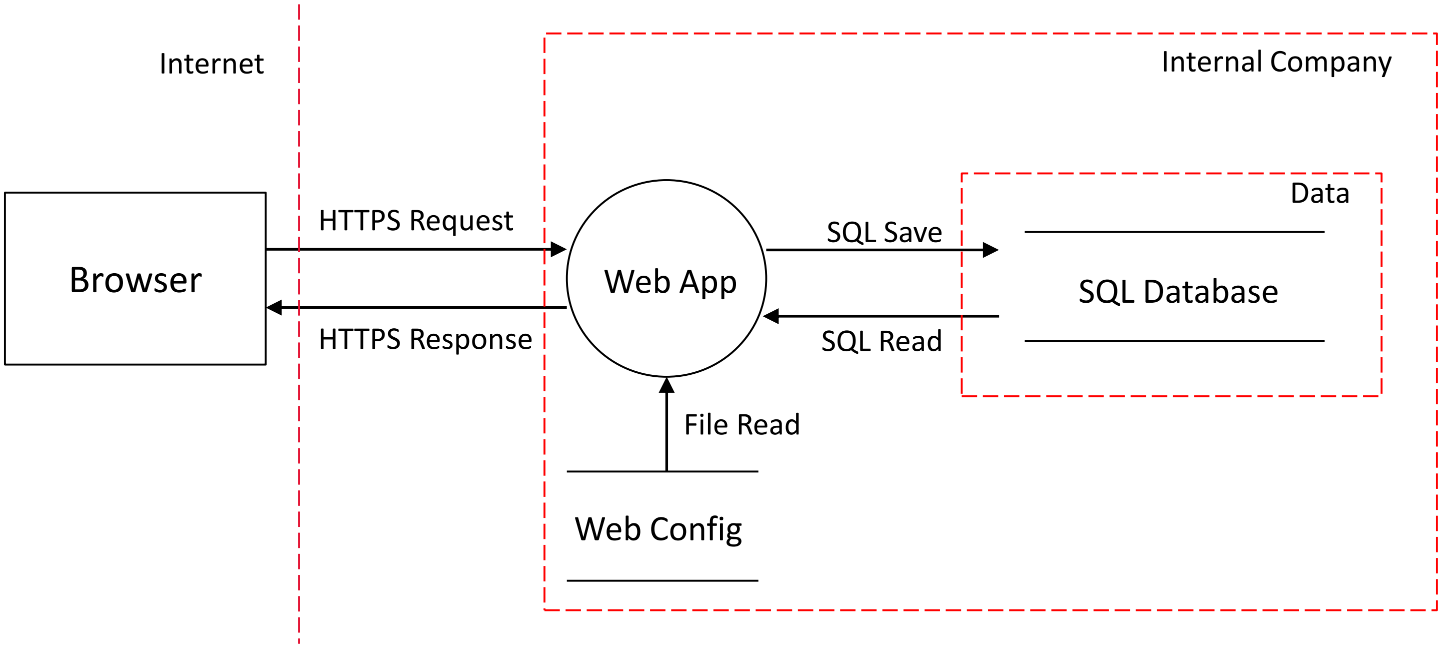
Explicitly identify trust boundaries.

	External Interactor / Entity – systems, users, - “static” elements (“We don’t own or control”)
	Process (single circle) or Complex Process (double circle) – handles/processes/moves data (“We own or control”)
	Data Flow – direction data flows
	Data Store – data storage such as databases, file systems, caches
	Trust Boundary – change of trust levels

1. Understand the system - create a Data Flow Diagram (DFD)

How do the entities, processes and data stores connect? Connect the info points with the data flow arrows.
Where are the trust boundaries?
For example:

- Browser (entity) sends / receives data (data flow) with a web application (process) which saves / reads data (data flow) using a SQL Database (data store)
- Web application (process) reads (data flow) web configuration file (data store)
- Trust boundaries indicate where trust changes — authenticate / authorize / validate



Threat Modeling Lab 1:
Review case study
Build data flow diagram (DFD)

Threat Modeling Process: Identify threats

2. Identify Threats – "What can go wrong?"

Conspicuously overloaded truck stopped by State Police

"Please remember, when traveling with a load in a vehicle, take a look at it and before taking to the roads, ask yourself, '**What could go wrong?**' " (Boston Globe, June 21, 2018)

2. Identify threats – Many Ways

STRIDE

Attack Trees

- Bruce Schneier - Slide deck

Threat Libraries

- CAPEC, ATT&CK, OWASP Top 10, SANS Top 25

Checklists

- OWASP ASVS, OWASP Proactive Controls

Card Games

- OWASP Cornucopia, Elevation of Privilege

Use Cases / Abuse Cases

Misuse Cases help with …

No one would ever do that!

Why / who would ever do that?!

STRIDE* Framework – Data Flow

Threat	Examples	Property we want
Spoofing	Pretending to be someone else	Identity Assurance
Tampering	Modifying data that should not be modifiable	Integrity
Repudiation	Claiming someone didn’t do something	Non-repudiation
Information Disclosure	Exposing information	Confidentiality
Denial of Service	Preventing a system from providing service	Availability
Elevation of Privilege	Doing things that one isn’t suppose to do	Least Privilege

* STRIDE was invented by Loren Kohnfelder and Praerit Garg (1999)

2. Identify Threats – Applying STRIDE to a DFD

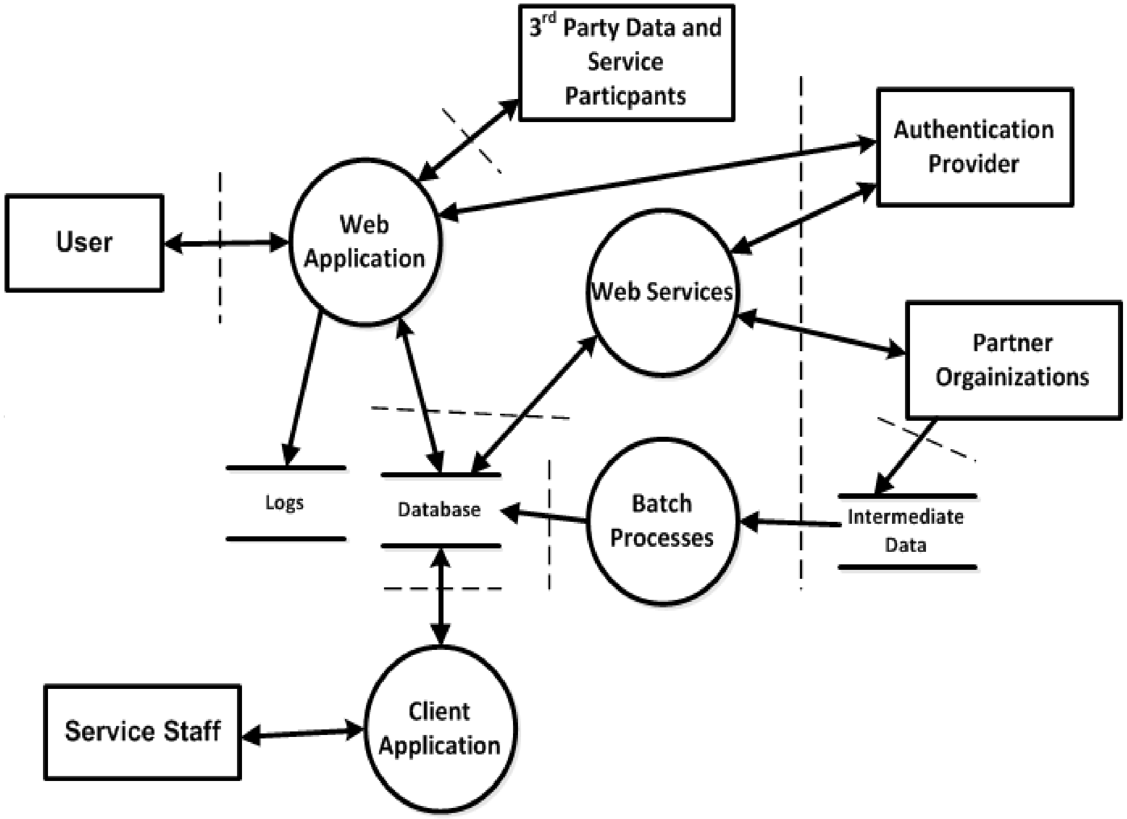
Options:

Each part of STRIDE applies to specific elements or interactions

and/or

You can look at STRIDE per interaction.

ACME Web Application



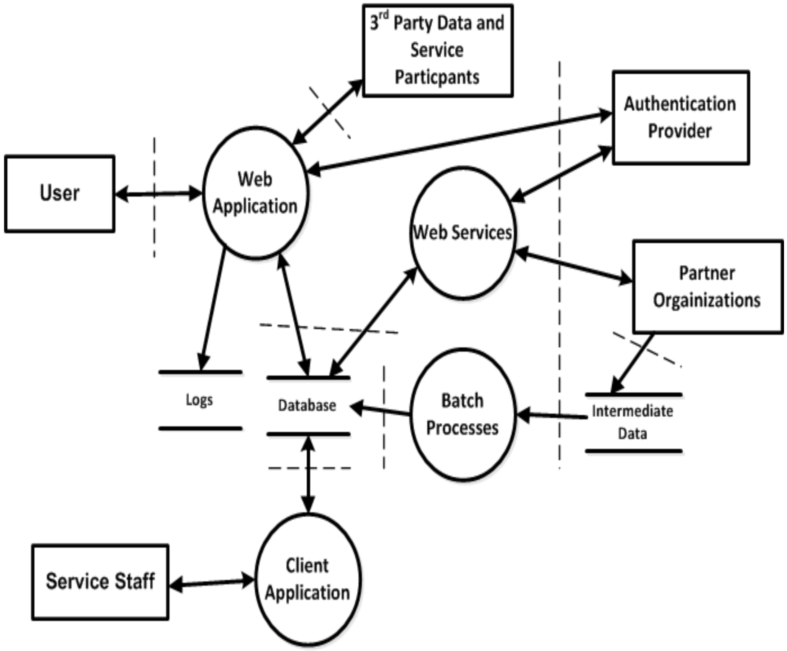
2. Identify threats — Mapping STRIDE to DFD

Threats	Data Flows	Data Stores	Processes	Entities
Spoofing			X	X
Tampering	X	X	X	
Repudiation		X	X	X
Information Disclosure	X	X	X	
Denial of Service	X	X	X	
Elevation of Privilege			X	

2. Identify Threats – Applying STRIDE to a DFD

Threat Model for ACME Web Application

Threat	STRIDE	
Partner Organization communication to Web Services may be compromised	Tampering, Information Disclosure	
Logs for Web Application may be tampered with	Tampering, Repudiation	



2. Identify Threats – Functional

Input and data validation

Authentication

Authorization

Configuration management

Data Classification

- Public, Proprietary, Confidential

2. Identify Threats – Functional

Session management

Cryptography

Parameter manipulation

Exception management

Auditing, logging, and monitoring

2. Identity Threats – Ask Questions

Who’s interested in app and data (threat agents)?

What goals (assets)?

What attack methods (how)?

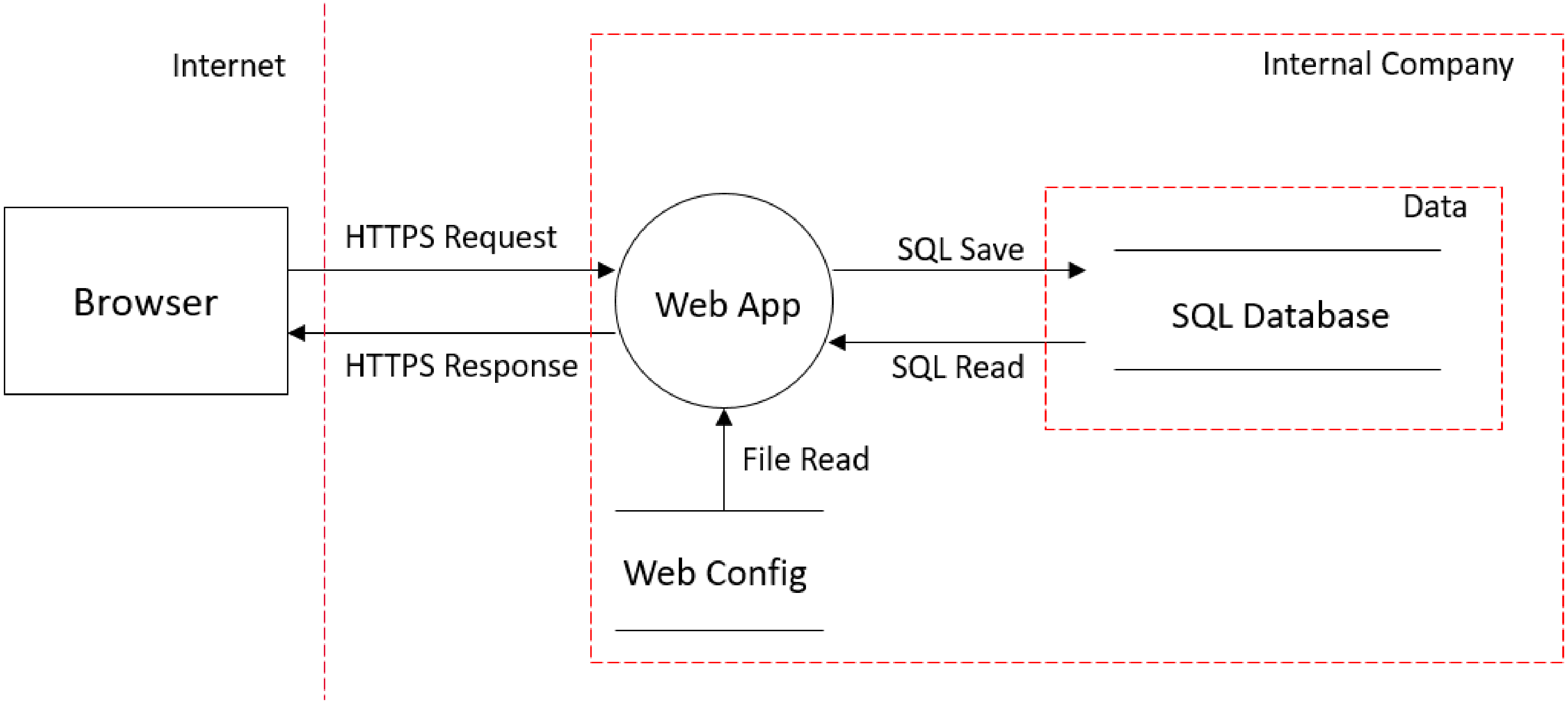
Any attack surfaces (trust boundaries) exposed?

Any input/output (data flows) missing?

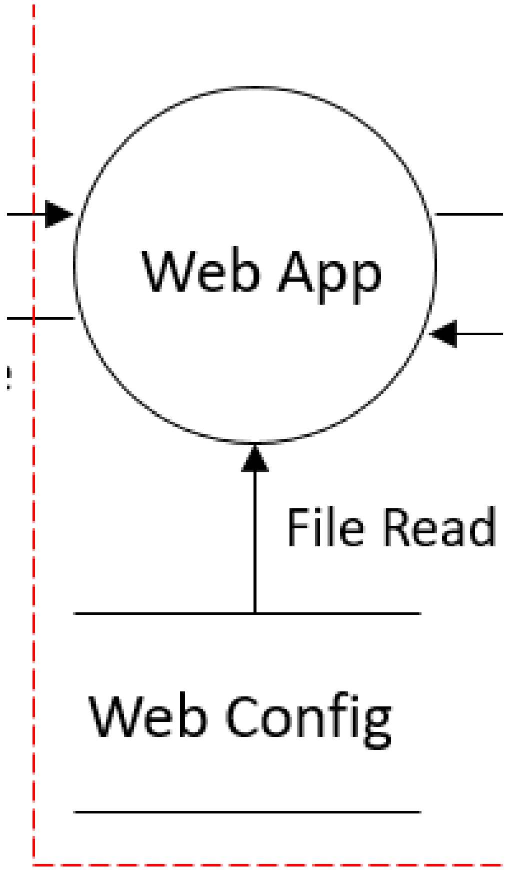
One of the best questions …

Is there anything keeping you up at night worrying about this system?

Scenario – Configuration Management



Scenario – Configuration Management



Data Files such as configuration files

Scenario – Configuration Management

System: Web application uses configuration files

Security principles:

Be reluctant to trust, Assume secrets not safe

Questions to identify threats:

How does the app use the configuration files?

What validation is applied?

Implied trust?

Can anyone update / change the files?

Threat Modeling Lab 2: Identify threats

Threat Modeling Process

1. Diagram / understand your system and data flows
2. Identify threats through answers to questions
3. **Determine mitigations and risks**
4. Follow through

Addressing each threat

Mitigation patterns:

Authentication / Identity Assurance -> mitigating spoofing

Integrity -> mitigating tampering

Non-repudiation -> mitigating repudiation

Confidentiality -> mitigating information disclosure

Availability -> mitigating denial of service

Authorization / Least Privilege -> mitigating elevation of privilege

Determine mitigations – Mitigations mapped to STRIDE

STRIDE	Example mitigations
Identity Assurance (**S**poofing)	• Authentication based on key exchange • Decide on single-factor, two-factor, or multi-factor authentication • Offload authentication to another provider • Restrict authentication to certain IP ranges or locations
Integrity (**T**ampering)	• Data protected from tampering with cryptographic integrity mechanisms • Only enumerated authorized users may modify data
Non-Repudiation (**R**epudiation)	• Maintain logs • Digital signature
Confidentiality (**I**nformation Disclosure)	• Data in files / database will only be available to authorized users • Name / existence of database will only be exposed to authorized users • Content and existence of communication between Alice and Bob will only be exposed to these authorized users
Availability (**D**enial of Service)	• Rate limiting or throttling access to a service • Real-time monitoring of log files and other resources to note sudden changes
Least Privilege (**E**levation of Privilege)	• System has a central authorization engine • Authorization controls stored with item being controlled using ACLs • System limits who can write data to higher integrity level • System uses roles / accounts or permissions to manage access

Mitigation patterns

Apply appropriate secure design patterns

Leverage proven best practices

Reuse organization security services e.g.
Single-Sign-On, Log Server, etc.

Do not reinvent the wheel

For threats not (completely) covered

Redesign to eliminate

Apply standard mitigations

Create new mitigations

Accept vulnerability in design

Mitigation Options:

- Leave as-is
- Remove from product
- Remedy with technology countermeasure
- Warn user

Determine risks

What is the risk associated with the vulnerability and threat identified?

Determine mitigations and risks

Risk Management

FAIR (Factor Analysis of Information Risk) – Jack Freund, Jack Jones

Risk Rating (High, Medium, Low)

Risk Rating

Overall risk of the threat expressed in High, Medium, or Low.

Risk is product of two factors:

- Ease of exploitation
- Business impact

Risk Rating – Ease of Exploitation

Risk Rating	Description
High	• Tools and exploits are readily available on the Internet or other locations • Exploitation requires no specialized knowledge of the system and little or no programming skills • Anonymous users can exploit the issue
Medium	• Tools and exploits are available but need to be modified to work successfully • Exploitation requires basic knowledge of the system and may require some programming skills • User-level access may be a pre-condition
Low	• Working tools or exploits are not readily available • Exploitation requires in-depth knowledge of the system and/or may require strong programming skills • User-level (or perhaps higher privilege) access may be one of a number of pre-conditions

Risk Rating – Business Impact

Risk Rating	Description
High	- Administrator-level access (for arbitrary code execution through privilege escalation for instance) or disclosure of sensitive information - Depending on the criticality of the system, some denial-of-service issues are considered high impact - All or significant number of users affected - Impact to brand or reputation
Medium	- User-level access with no disclosure of sensitive information - Depending on the criticality of the system, some denial-of-service issues are considered medium impact
Low	- Disclosure of non-sensitive information, such as configuration details that may assist an attacker - Failure to adhere to recommended best practices (which does not result in an immediately visible exploit) also falls into this bracket - Low number of user affected

Example – Medium Risk Threat

ID - Risk	3 - Medium
Threat	Lack of CSRF protection allows attackers to submit commands on behalf of users
Description/Impact	Client applications could be subject to a CSRF attack where the attacker embeds commands in the client applications and uses it to submit commands to the server on behalf of the users
Countermeasures	Per transaction codes (nonce), thresholds, event visibility
Components Affected	CO-3

Scenario – Configuration Management

System: Web application uses configuration files

Security principles:

Be reluctant to trust, Assume secrets not safe

Questions to identify threats:

How does the app use the configuration files?

What validation is applied?

Implied trust?

Can anyone change / update the files?

Possible controls / mitigations:

Set permissions on configuration files.

Validate all data input from files.

Use fuzz testing to insure input validation.

Scenario – Configuration Management

System: Web application uses configuration files

Security principles:

Be reluctant to trust, Assume secrets not safe

Questions to identify threats:

How does the app use the configuration files?

What validation is applied?

Implied trust?

Can anyone change / update the files?

Possible controls / mitigations:

Set permissions on configuration files.

Validate all data input from files.

Use fuzz testing to insure input validation.

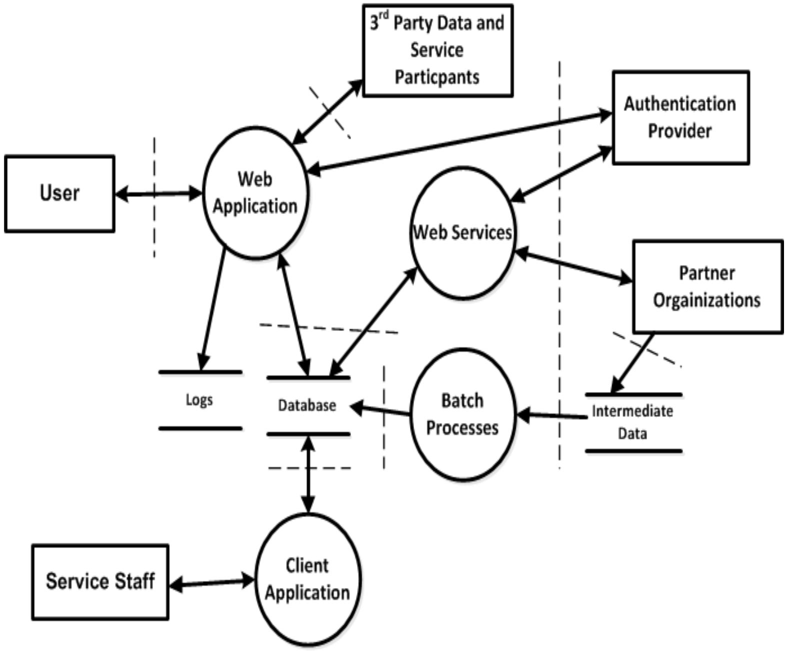
Risk Rating:

On-Premises (Medium/Low) vs. Cloud (High)

3. Determine mitigations and risks

Threat Model for ACME Web Application:

Threat	STRIDE	Mitigation / Risk
Partner Organization communication to Web Services may be compromised	Tampering, Information Disclosure	Implement encryption (HTTPS TLS 1.2) and validation of message integrity (High)
Logs for Web Application may be tampered with	Tampering, Repudiation	Apply access control on logs, send logs to centralized server (Medium)



Threat Modeling Lab 3: Determine mitigations

Threat Modeling Process

1. Diagram / understand your system and data flows
2. Identify threats through answers to questions
3. Determine mitigations and risks
4. **Follow through**

4. Follow through

Document findings and decisions

File bugs or new requirements

Verify bugs fixed / new requirements implemented

Did we miss anything? Review again

Anything new? Review again

4. Follow through - Communicate Your Threat Model

Present results – in person, ideally

Discuss countermeasures – cost vs. impact

Complete threat model with proposed action list you know is acceptable

4. Follow through - Communicate Your Threat Model

Architects – Integrate proposition to update design

Developers – Benefit from the threat model transparently through updated specification

Security testing team – Now know what to test!

Software editor – If acquiring software, add threat model to software acceptance tasks

4. Follow through - Update Your Threat Model

First Threat Model during design

Update Threat Model during technology decisions

Review Threat Model before implementation

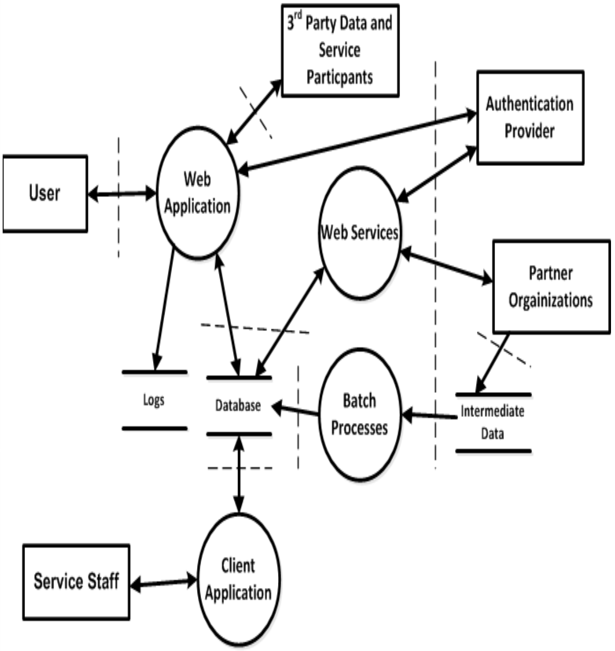
Refine and verify Threat Model during security review

Iterate

4. Follow through

Threat Model for ACME Web Application

Threat	STRIDE	Mitigation / Risk	Follow through
Partner Organization communication to Web Services may be compromised	Tampering, Information Disclosure	Implement encryption (HTTPS TLS 1.2) and validation of message integrity (High)	Address issue in next Sprint
Logs for Web Application may be tampered with	Tampering, Repudiation	Apply access control on logs, send logs to centralized server (Medium)	Evaluate if will fix in next Sprint or future Sprint



Your threat model now consists of …

1. Diagram / understand your system and data flows
2. Identify threats through answers to questions
3. Determine mitigations and risks
4. Follow through

A living threat model!

Threat Modeling:
Using Tools

Threat Modeling Lab 4: Threat Modeling Tools and Card Games / Decks

What next?

What next?

Look at tools that can help take you further (DFDs):

- MS Threat Modeling Tool
- OWASP Threat Dragon
- Draw.IO – see Michael Enriksen’s article: https://michenricksen.com/blog/drawio-for-threat-modeling

What next?, continued

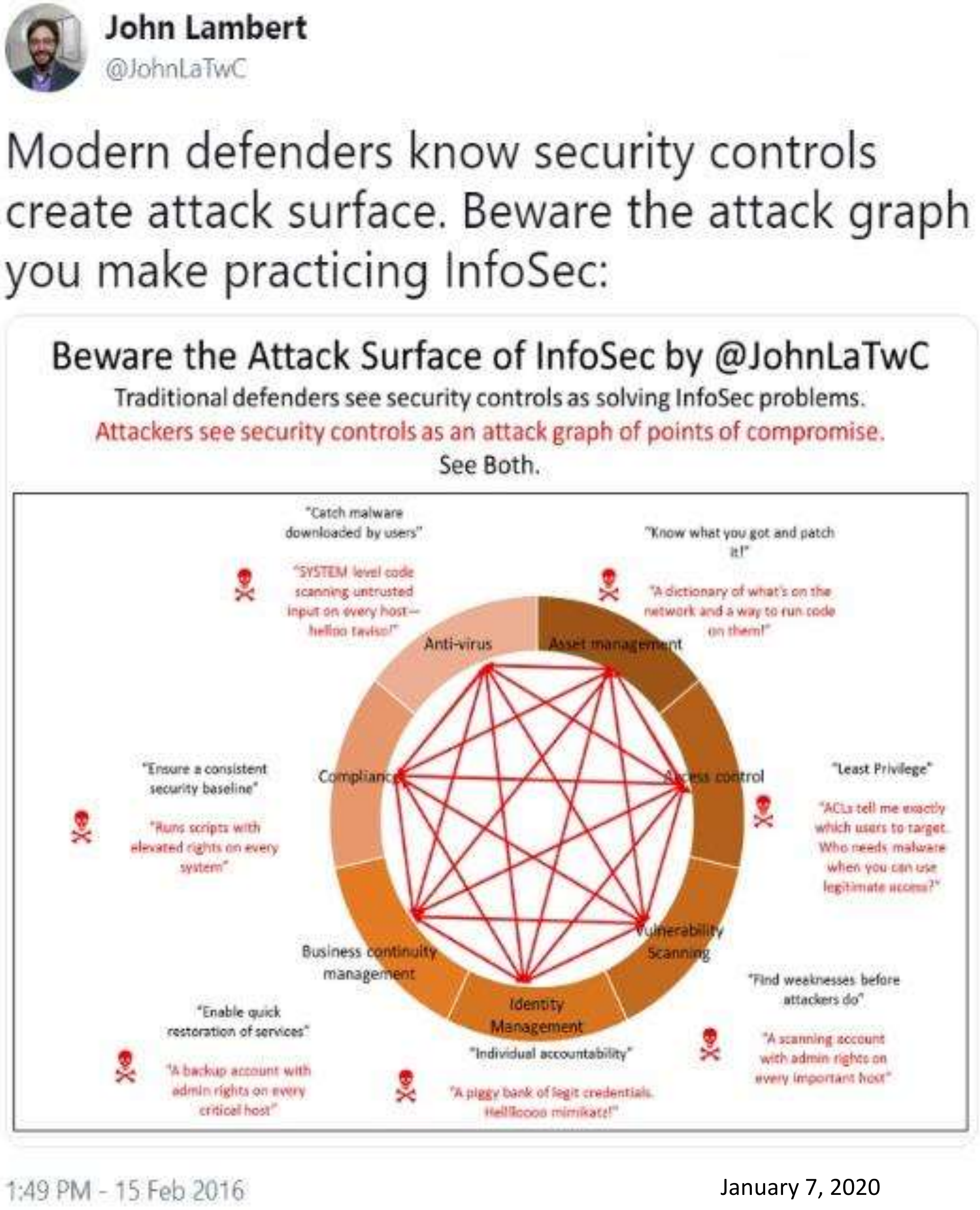
Learn more about:

- Attack Trees
  - Bruce Schneier’s 1999 article
- Incremental Threat Modeling
  - Agile approaches – Irene Michlin (@IreneMichlin)
- Lateral Movement
  - “The Industrial Revolution for Lateral Movement” BlackHat 2017
- Using MITRE ATT&CK for Threat Modeling
  - Brook Schoenfield “Secrets Of A Cyber Security Architect”, due Fall 2019 or Winter 2020

What next?, continued

Learn more about:

- List vs Graph Thinking
- Recursive Threat Modeling
  - John Lambert (@JohnLaTwC) at Microsoft



John Lambert
@JohnLaTwC

Modern defenders know security controls create attack surface. Beware the attack graph you make practicing InfoSec:

Beware the Attack Surface of InfoSec by @JohnLaTwC

Traditional defenders see security controls as solving InfoSec problems.

Attackers see security controls as an attack graph of points of compromise.

See Both.

"Catch malware downloaded by users"
"SYSTEM level code scanning untrusted input on every host— helloo taviso!"
"Know what you got and patch it!"
"A dictionary of what's on the network and a way to run code on them!"
Anti-virus
Asset management
"Ensure a consistent security baseline"
"Runs scripts with elevated rights on every system"
Compliance
Access control
"Least Privilege"
"ACLs tell me exactly which users to target. Who needs malware when you can use legitimate access?"
Business continuity management
Vulnerability Scanning
Identity Management
"Find weaknesses before attackers do"
"Enable quick restoration of services"
"Individual accountability"
"A scanning account with admin rights on every important host"
"A backup account with admin rights on every critical host"
"A piggy bank of legit credentials. Hellloooo mimikatz!"

1:49 PM - 15 Feb 2016

Mozilla's Rapid Risk Assessment (RRA) *

No time for a full threat model? ***RRA in 30 minutes***

Focused on services and entry points:

1. Are you making changes to the attack surface? (i.e new entry points)
2. Are you changing the application stack or application security controls?
3. Are you adding confidential/sensitive data?
4. Have threat agents changed? Are we facing new risk?

* https://infosec.mozilla.org/guidelines/risk/rapid_risk_assessment.html
Blog post: https://home.edwinkwan.com/rapid-risk-assessments/

What next?, continued

Learn more about:

- Threat Modeling as Code
  - ThreatPlaybook (@abhaybhargav)
  - ThreatSpec (@ThreatSpec, @zeroXten)
  - PyTM, CTM (@izar_t)

Conclusion

Get started with Threat Modeling today:

Start with secure design as goal

Ask the “what if” questions

Understand bigger picture

Resources - Books

Threat Modeling: Designing for Security

Adam Shostack

Securing Systems: Applied Architecture and Threat Models

Brook S.E. Schoenfield

Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis

Marco Morana and Tony UcedaVelez

Measuring and Managing Information Risk: A FAIR Approach

Jack Jones and Jack Freund

Resources - Books

Agile Application Security

Laura Bell, Michael Brunton-Spall, Rich Smith, Jim Bird

Secrets of a Cyber Security Architect

Brook S.E. Schoenfield

Upcoming books:

Threat Modeling (April, 2020)

Izar Tarandach, Matthew J. Coles

Resources - Tools

Microsoft Threat Modeling Tool

https://aka.ms/threatmodelingtool

ThreatModeler

https://threatmodeler.com

IriusRisk Software Risk Manager

https://iriusrisk.com/threat-modeling-tool/

OWASP Threat Dragon

https://www.owasp.org/index.php/OWASP_Threat_Dragon

Resources - Tools

Attack Trees – Bruce Schneier on Security

https://www.schneier.com/attacktrees.pdf

Elevation of Privilege (EoP) Game

http://www.microsoft.com/en-us/download/details.aspx?id=20303

OWASP Cornucopia

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

OWASP Application Security Verification Standard (ASVS)

https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

OWASP Top 10 Proactive Controls 2018

https://www.owasp.org/index.php/OWASP_Proactive_Controls

Questions?

@RobertHurlbut

Thank you!