Threat Modeling For Secure Software Design

Code Mash

January 12, 2017

Robert Hurlbut

RobertHurlbut.com • @RobertHurlbut

Robert Hurlbut

Software Security Consultant, Architect, and Trainer

- Owner / President of Robert Hurlbut Consulting Services
- Microsoft MVP – Developer Security 2005-2009, 2015, 2016
- (ISC)2 CSSLP 2014-2017
- Co-host with Chris Romeo – Application Security Podcast

Contacts

Web Site: https://roberthurlbut.com

Twitter: @RobertHurlbut, @AppSecPodcast

ROBERT HURLBUT CONSULTING SERVICES

securedevelop.NET

Software Design

Determine requirements

Determine features

Build software people will use

Secure Software Design

Determine secure requirements
Determine secure features
Build software people will use
… and will anticipate mis-use

How? A security mindset!

Teaching Security Mindset

Schneier - on teaching others (2012):

> *"Teach yourself and your students to cheat. We've always been taught to color inside the lines, stick to the rules, and never, ever, cheat. In seeking cyber security, we must drop that mindset."**

(* Quoted from a paper by Gregory Conti and James Caroland. See: https://www.schneier.com/blog/archives/2012/06/teaching_the_se.html and http://www.rumint.org/gregconti/publications/KobayashiMaru_PrePub.pdf)

What is threat modeling?

Threat modeling helps you think strategically about your software design, in particular your secure software design.

A “way of thinking” tool – not automated security tool

What is threat modeling?

Threat modeling is:

> Process of understanding your system and potential threats against your system

i.e. Critical Thinking about Security

What is threat modeling?

Threat model includes:

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

Definitions

Threat Agent

Someone (or a process) who could do harm to a system (also adversary or attacker)

Definitions

Threat

An adversary's goal

Definitions

Vulnerability

A flaw in the system that could help a threat agent realize a threat

Definitions

Attack

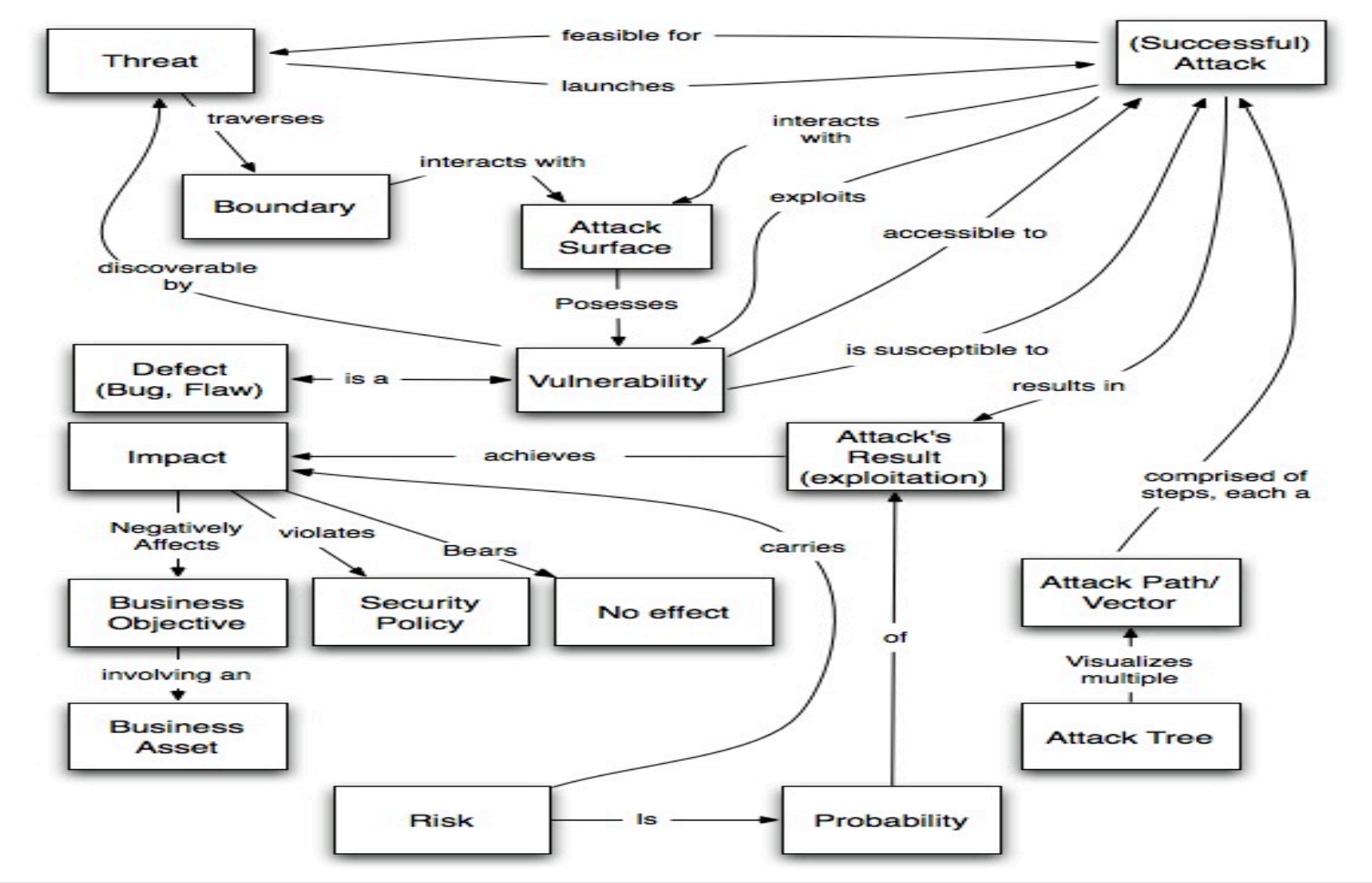
When a motivated and sufficiently skilled threat agent takes advantage of a vulnerability

Definitions

Asset

Something of value to valid users and adversaries alike

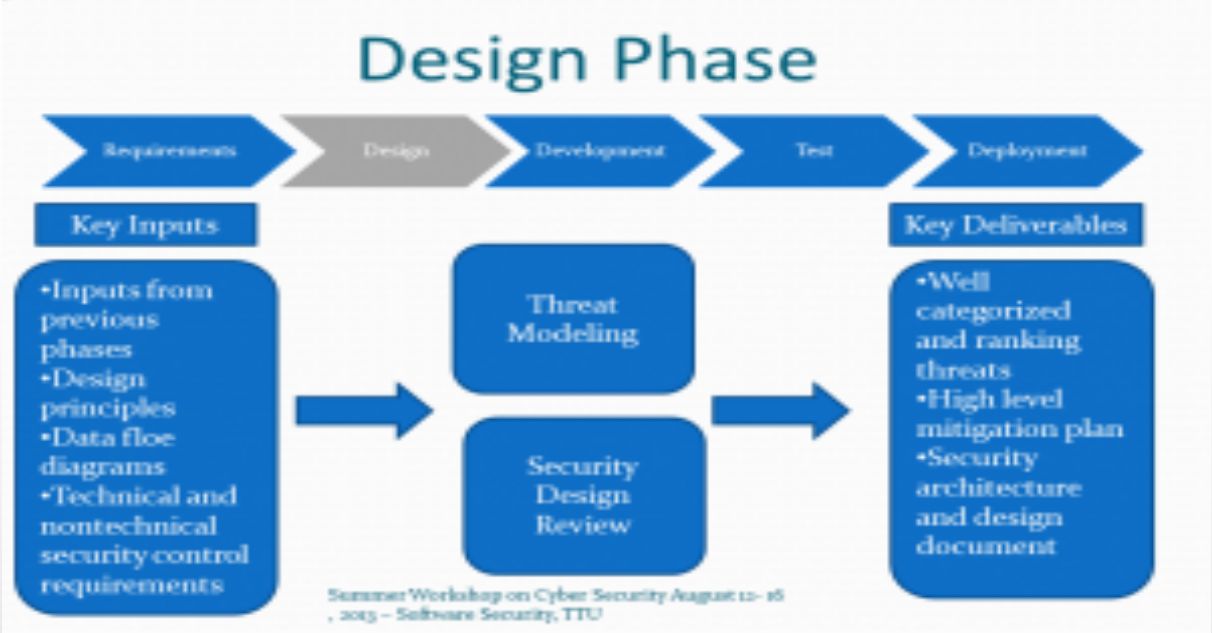
Threat Modeling Vocabulary*



* https://www.cigital.com/blog/threat-modeling-vocabulary/ (John Steven, Cigital)

When? Make threat modeling first priority

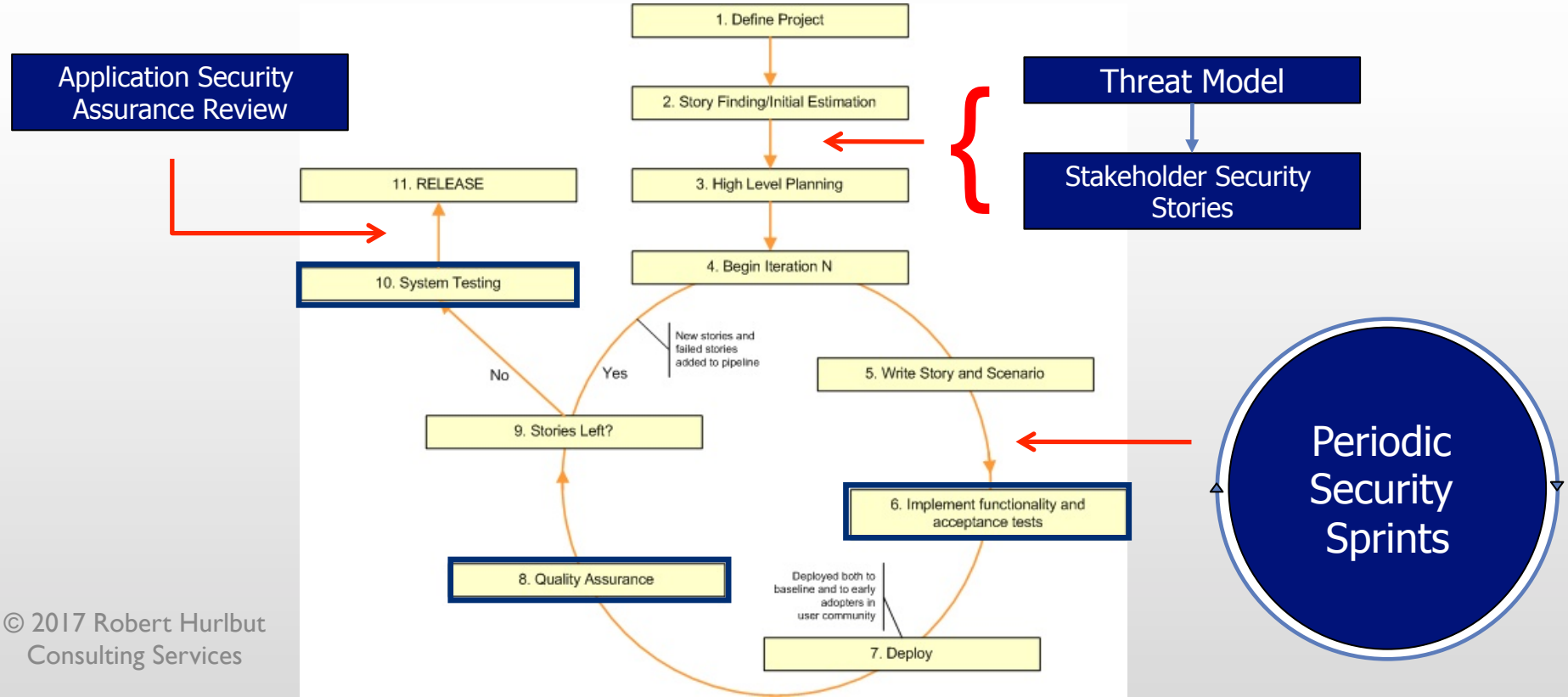
In SDLC – Requirements and Design phase



Threat modeling uncovers new requirements

When? Make threat modeling first priority

Agile Sprint Planning - User Stories, Attacker Stories



When?

What if we didn't?

- It's not too late to start threat modeling (generally)
- It will be more difficult to change major design decisions
- Do it anyway!

Simple Tools

Whiteboard

Visio (or equivalent) for diagraming

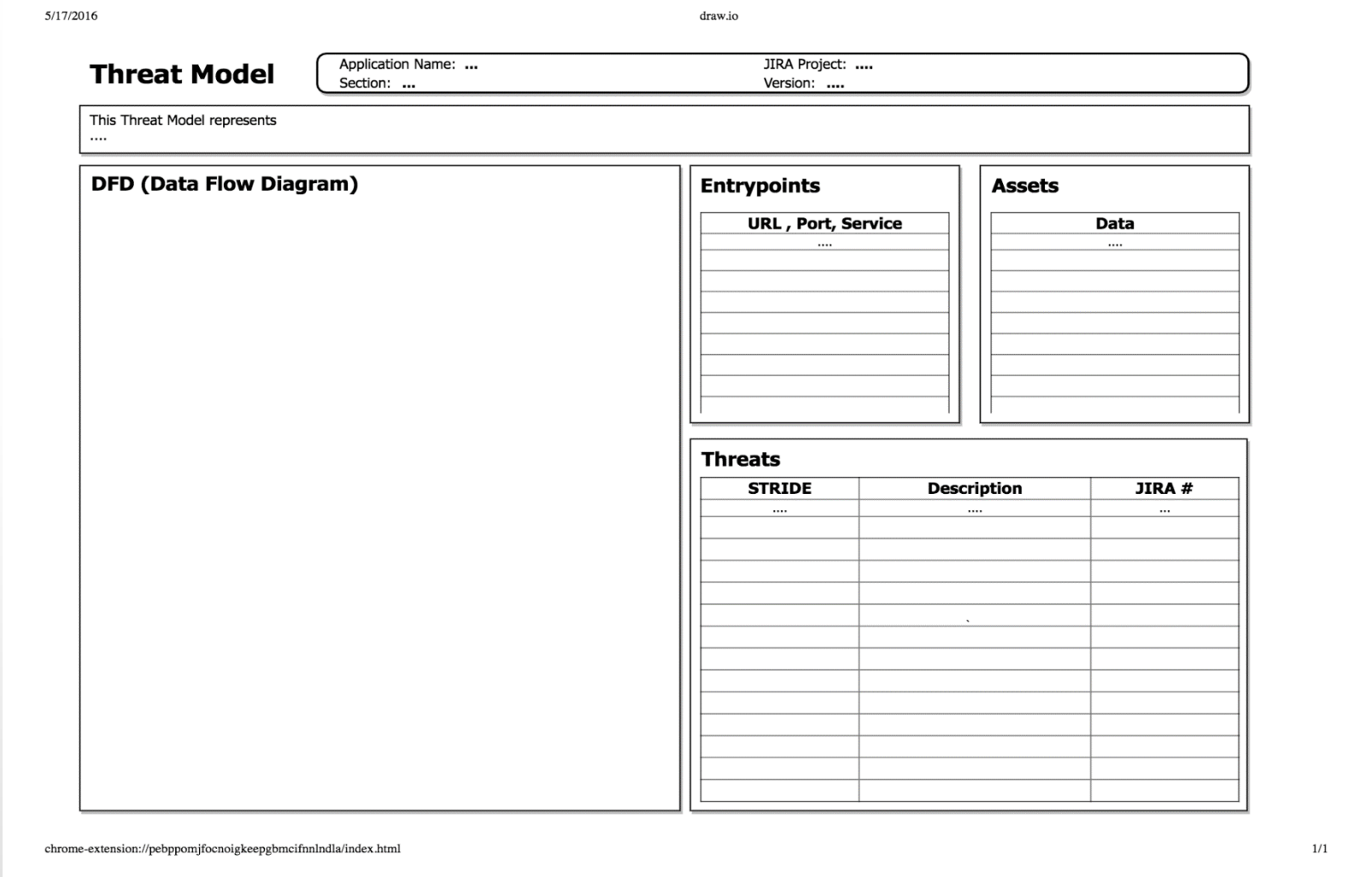
Word (or equivalent) or Excel (or equivalent) for documenting

Simple Threat Model – One Page

Look at Dinis Cruz' Simple Threat Model One Page Template and Concepts

http://blog.diniscruz.com/2016/05/threat-modeling-template-and-concepts.html

Simple Threat Model – One Page*



5/17/2016 draw.io

Threat Model

Application Name: ...
Section: ...
JIRA Project: ....
Version: ....

This Threat Model represents
....

DFD (Data Flow Diagram)

Entrypoints

URL , Port, Service
....

Assets

Data
....

Threats

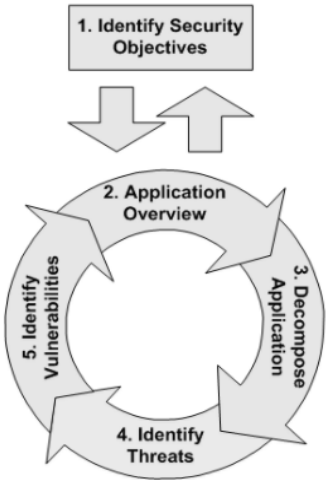
STRIDE	Description	JIRA #
....	....	...

chrome-extension://pebppomjfocnoigkeepgbmcifnnlndla/index.html 1/1

* https://github.com/DinisCruz/Security-Research/blob/master/pdfs/Threat-Modeling/Template/Threat-Model-Template-v0.1.pdf

Simple Threat Model – Concepts*

Threat Model Concepts



Vocabulary

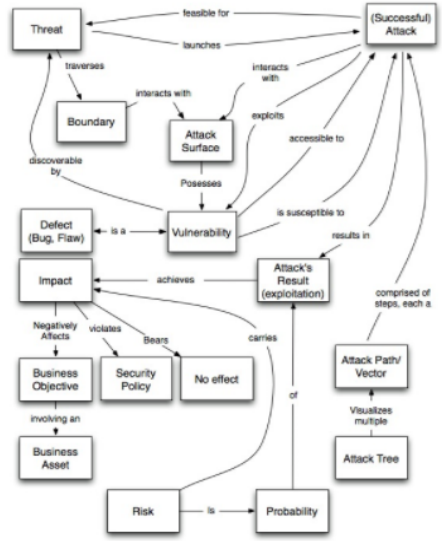


Image from https://www.cigital.com/blog/threat-modeling-vocabulary/

DFD Elements

External Entity	The external entity shape is used to represent any entity outside the application that interacts with the application via an entry point
Process	Represents a task that handles data within the application. The task may process the data or perform an action based on the
Multiple Process	Used to present a collection of subprocesses. The multiple process can be broken down into its subprocesses in another DFD.
Data Store	Represents locations where data is stored
Data Flow	Represents data movement within the application. The direction of the data movement is represented by the arrow.
Privilege Boundary	Represent the change of privilege levels as the data flows through the application.

Data Classification

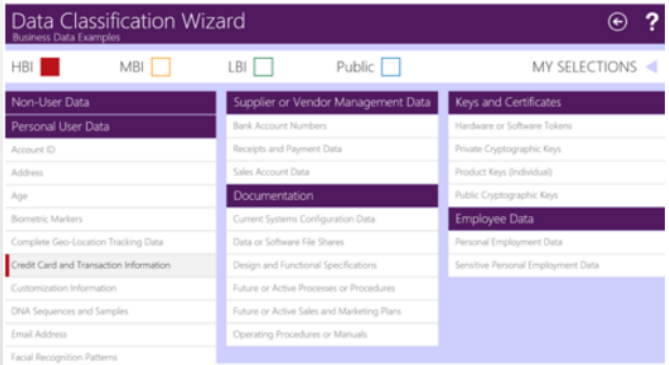


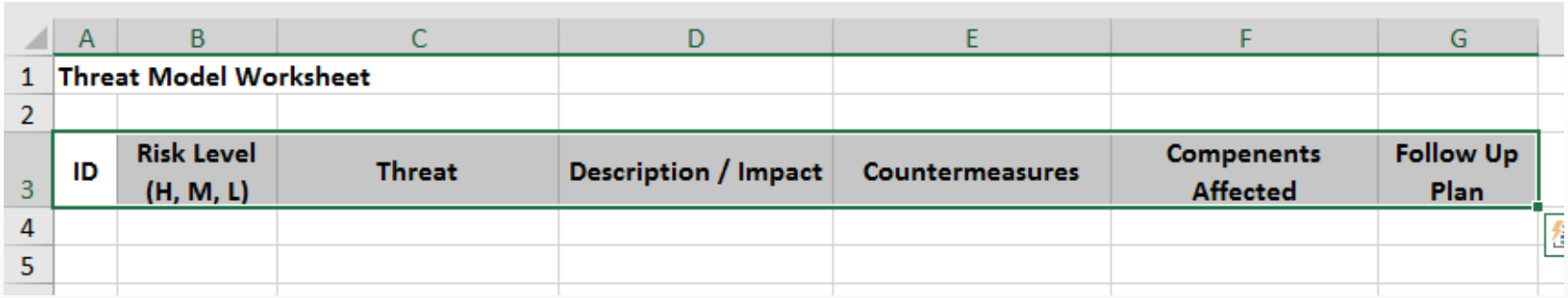
image from https://www.microsoft.com/security/data/

STRIDE

Threat	Description	Breaks
Spoofing	Pretending to be somebody else	Authentication
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 supposed to do	Authorization

* https://github.com/DinisCruz/Security-Research/blob/master/pdfs/Threat-Modeling/Concepts/Threat%20Model%20Concepts-v0.2.pdf

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

Microsoft Threat Modeling Tool 2016

ThreatModeler – Web Based (in-house) Tool

ThreadFix

IriusRisk Software Risk Manager

Review Security Principles

1. Secure the weakest link
2. Defend in depth
3. Fail securely
4. Grant least privilege
5. Separate privileges
6. Economize mechanisms

http://searchsecurity.techtarget.com/opinion/Thirteen-principles-to-ensure-enterprise-system-security Gary McGraw, 2013

Review Security Principles

7. Do not share mechanisms
8. Be reluctant to trust
9. Assume your secrets are not safe
10. Mediate completely
11. Make security usable
12. Promote privacy
13. Use your resources

http://searchsecurity.techtarget.com/opinion/Thirteen-principles-to-ensure-enterprise-system-security Gary McGraw, 2013

IEEE Computer Society's Center for Secure Design

Take a look at:

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

Bugs vs Flaws

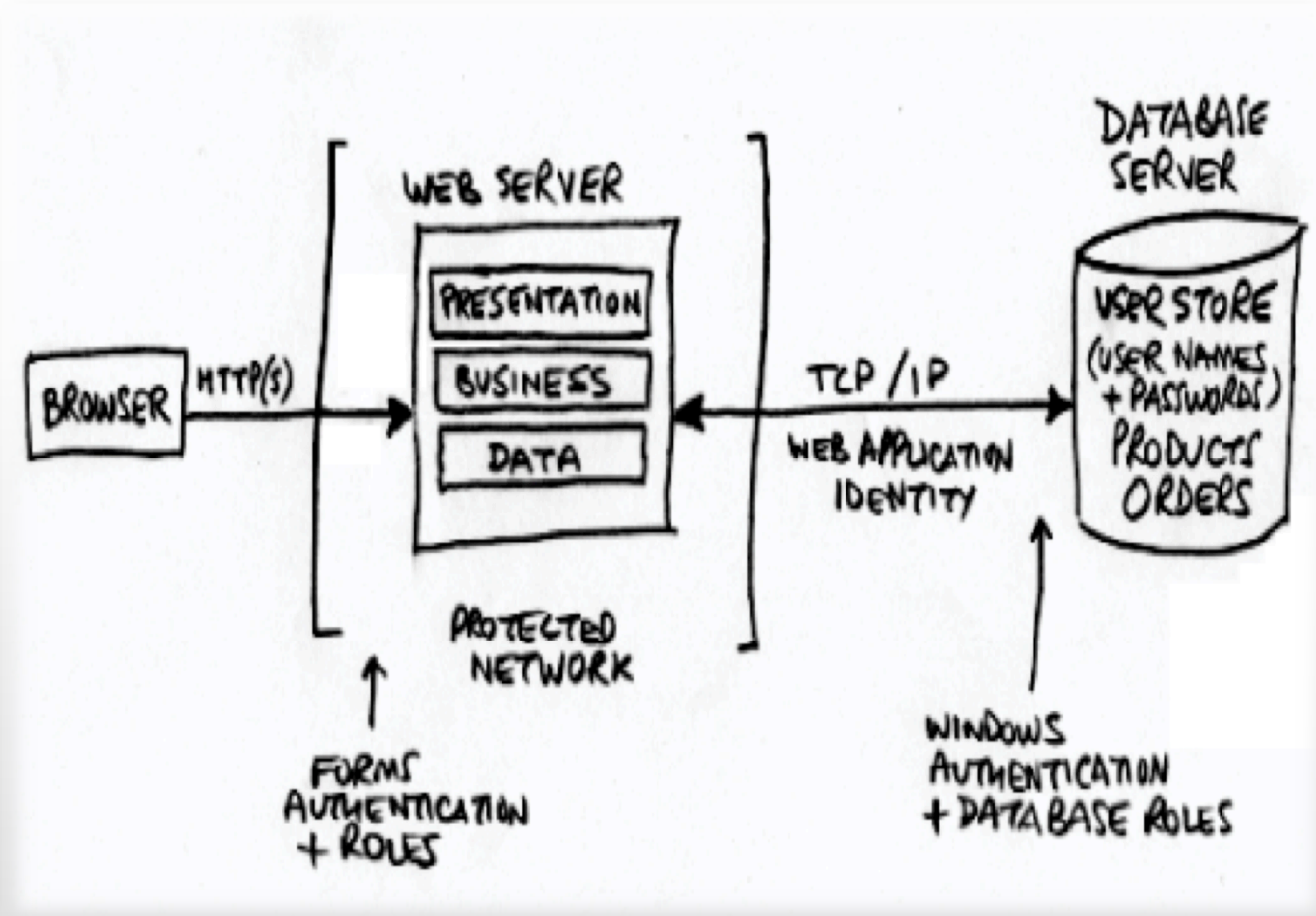
Bug – an implementation-level software problem

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

Threat Modeling Process

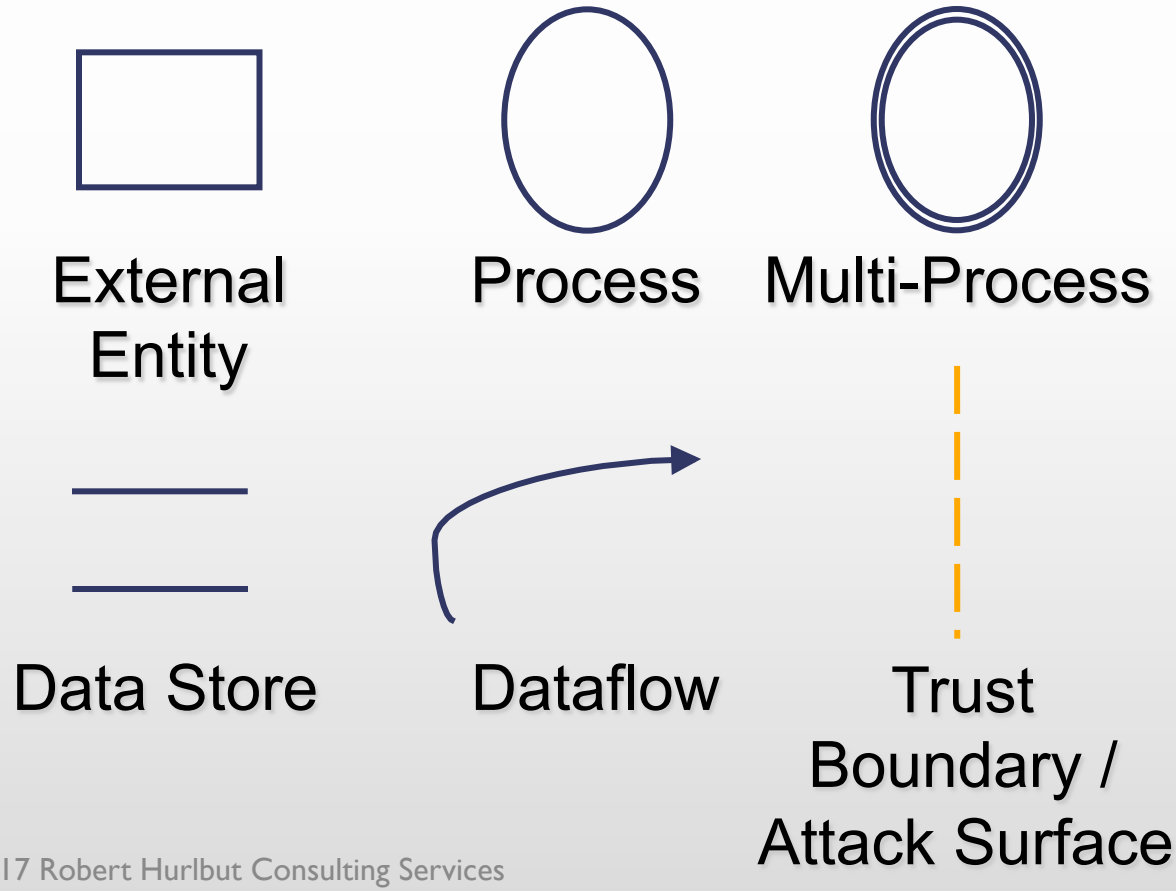
1. Draw your picture – understand the system and the data flows
2. Identify threats through answers to questions
3. Determine mitigations and risks
4. Follow through

Draw your picture



Understand the system

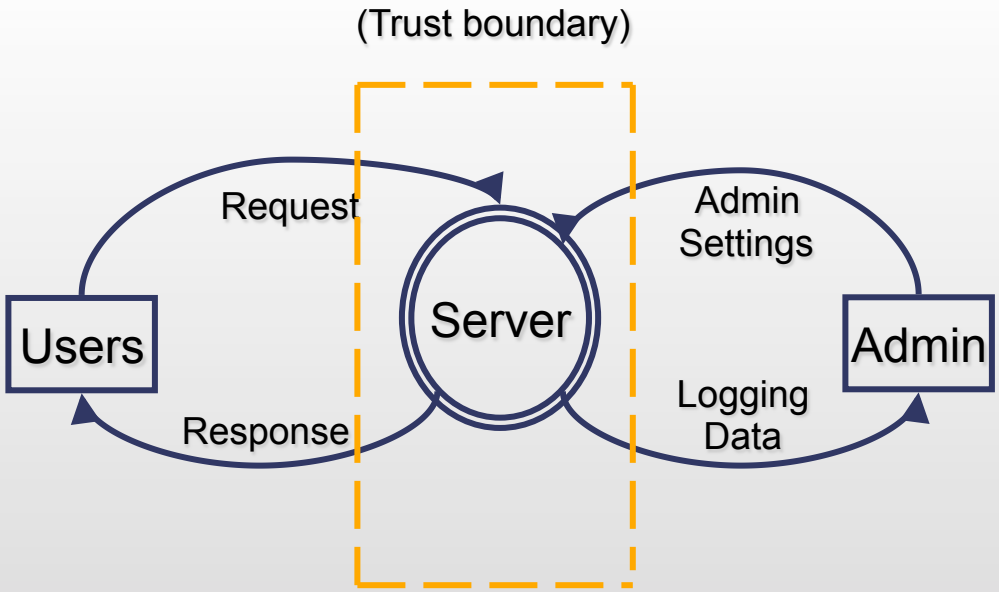
DFD – Data Flow Diagrams (MS SDL)



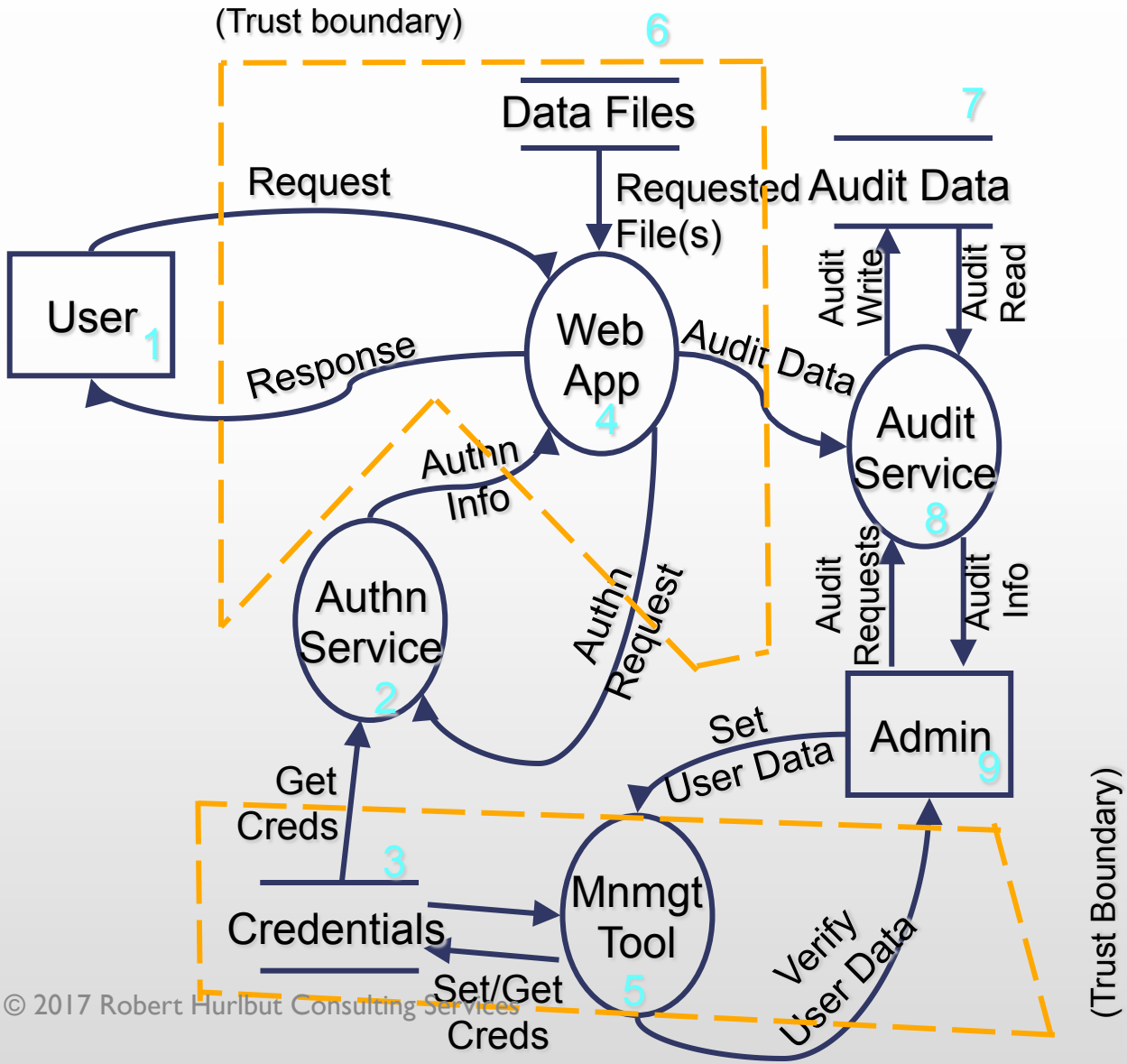
Understand the System

Understand logical and component architecture of system

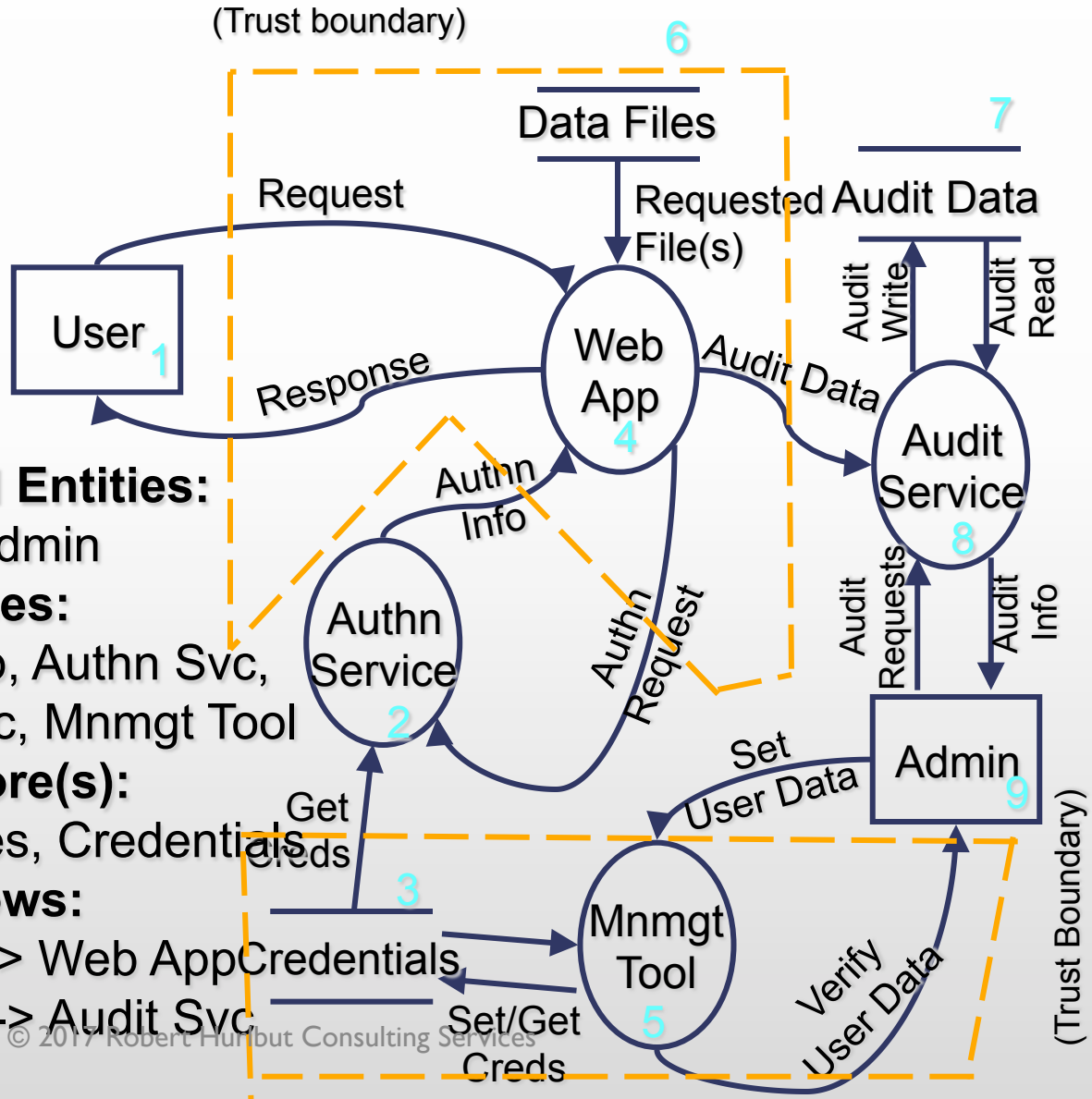
Understand every communication flow and valuable data moved and stored



Understand the system



Understand the system



External Entities:
Users, Admin
Processes:
Web App, Authn Svc, Audit Svc, Mnmgt Tool
Data Store(s):
Data Files, Credentials
Data Flows:
Users <-> Web AppCredentials
Admin <-> Audit Svc

Your threat model now consists of …

1. Diagram / understanding of your system and the data flows

Identify threats

Most important part of threat modeling (and most difficult)

Many ways – determine what works best for your team

Identify threats

Attack Trees

- Bruce Schneier - Slide deck

Threat Libraries

- CAPEC, OWASP Top 10, SANS Top 25

Checklists

- OWASP ASVS, OWASP Proactive Controls

Use Cases / Misuse Cases

Identity threats - Games

OWASP Cornucopia

Suits:

Data validation and encoding

Authentication

Session Management

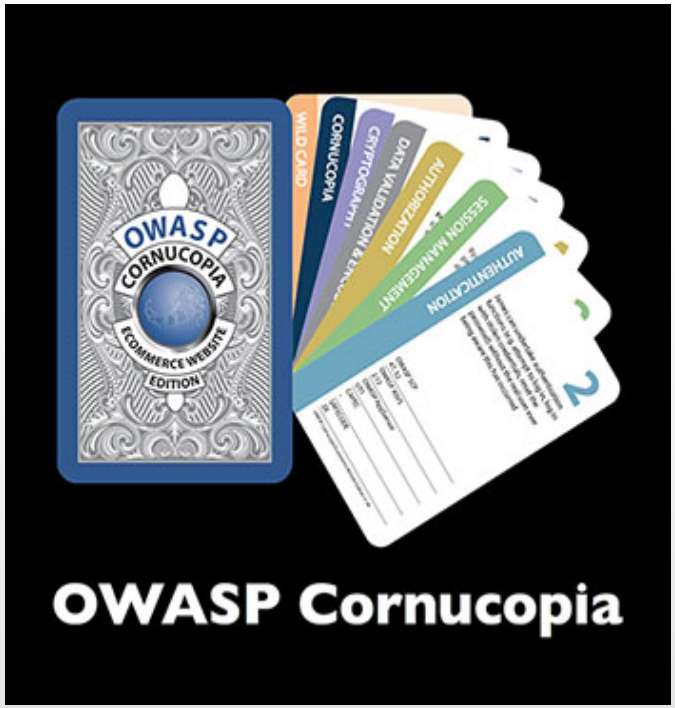
Authorization

Cryptography

Cornucopia

13 cards per suit, 2 Jokers

Play a round, highest value wins



STRIDE Framework – Data Flow

Threat	Property we want
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Identify Threats – Functional

Input and data validation

Authentication

Authorization

Configuration management

Sensitive data

Identify Threats – Functional

Session management

Cryptography

Parameter manipulation

Exception management

Auditing and logging

Identity Threats - Ask Questions

Who would be interested in the application and its data (threat agents)?

What are the goals (assets)?

What are attack methods for the system we are building?

Are there any attack surfaces exposed - data flows (input/output) we are missing?

Identity Threats – Ask Questions

How is authentication handled between callers and services?

What about authorization?

Are we sending data in the open?

Are we using cryptography properly?

Is there logging? What is stored?

Etc.

One of the best questions …

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

Identify Threats – Example

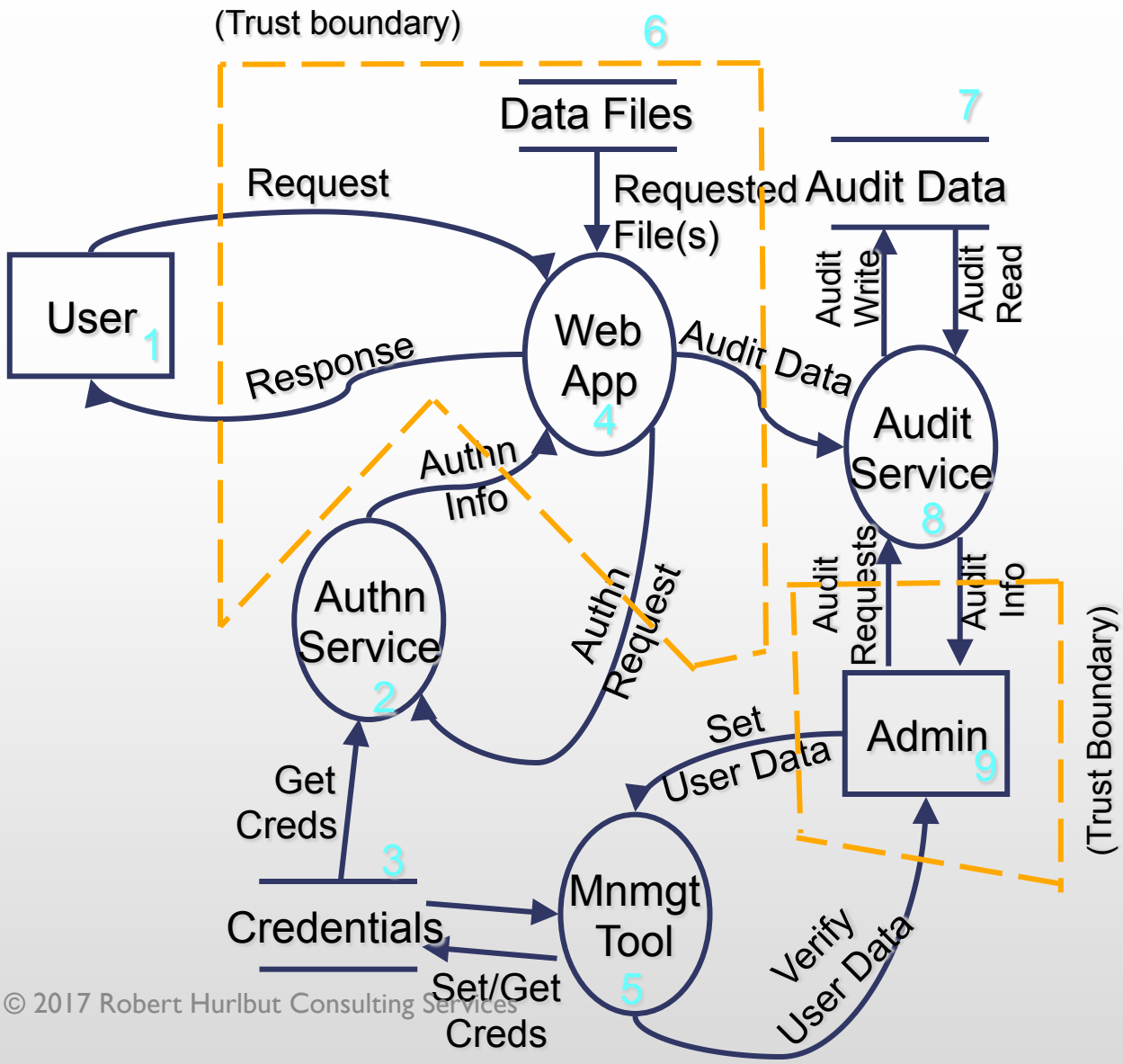
Confused Deputy Problem

Implied trust transferred to other services (usually seen with RBAC, CSRF, Clickjacking, etc.)

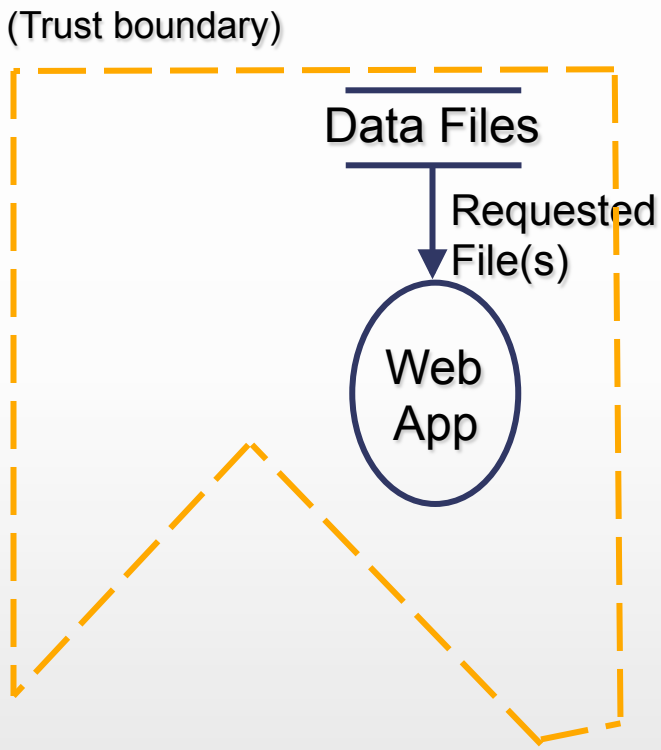
Action + Permission

Solve by capabilities (or claims)

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:

How does the app use the configuration files?

What validation is applied? Implied trust?

Possible controls/mitigation:

Set permissions on configuration files.

Validate all data input from files. Use fuzz testing to insure input validation.

Your threat model now consists of …

1. Diagram / understanding of your system and the data flows
2. Identify threats through answers to questions

Determine mitigations and risks

Mitigation Options:

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

What is the risk associated with the vulnerability?

Determine mitigations and risks

Risk Management

- FAIR (Factor Analysis of Information Risk) – Jack Jones, Jack Freund
- CVSS (Common Vulnerability Scoring System)
- Generic 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

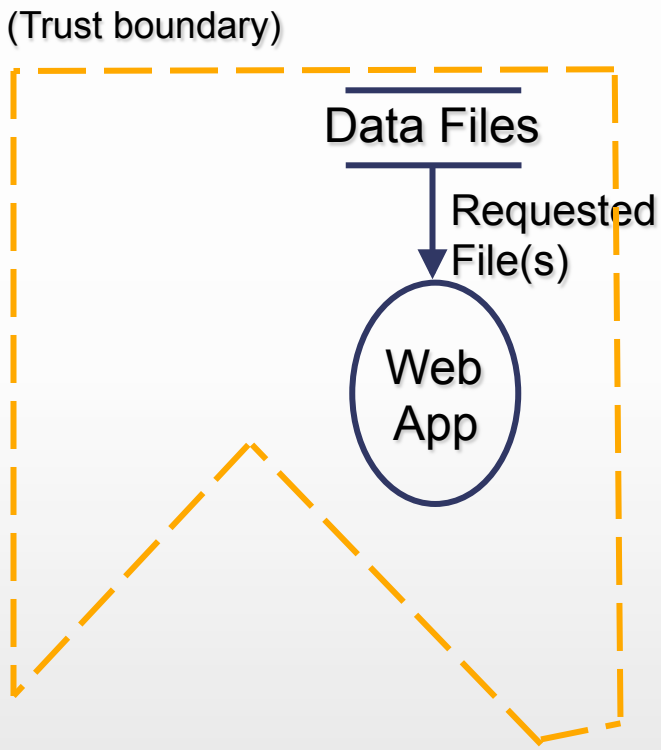
Risk Rating – Business Impact

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	RT-3
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



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:

How does the app use the configuration files?

What validation is applied? Implied trust?

Possible controls/mitigation:

Set permissions on configuration files.

Validate all data input from files. Use fuzz testing to insure input validation.

Risk Rating:

We own the box (Medium/Low), Hosted on cloud (High)

Your threat model now consists of …

1. Diagram / understanding of your system and the data flows
2. Identify threats through answers to questions
3. Mitigations and risks identified to deal with the threats

Follow through

Document what you found and decisions you make

File bugs or new requirements

Verify bugs fixed and new requirements implemented

Did we miss anything? Review again

Anything new? Review again

Your threat model now consists of …

1. Diagram / understanding of your system and the data flows
2. Identify threats through answers to questions
3. Mitigations and risks identified to deal with the threats
4. Follow through

A living threat model!

Recursive Threat Modeling

See John Lambert's article:

How Infosec Security Controls Create Vulnerability

https://blogs.technet.microsoft.com/johnla/2016/02/20/how-infosec-security-controls-create-vulnerability/

The selection of controls must be recursively and holistically threat modeled for completeness. *This difficulty in doing this can be exacerbated if the subject matter expertise to do the threat modeling is different at every layer. For example, an InfoSec practitioner using a Data Loss Prevention solution to mitigate sensitive data leaving the network may be an expert on SOX, PCI, and categories of customer PII, but they may not be an expert on the security implementation requirements of a Linux based appliance they procured.* ***Controls come with risks and must be treated accordingly.***

Your challenge

Use threat modeling for:

- secure design before new features
- driving your testing and other review activities
- understanding 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 - Tools

Microsoft Threat Modeling Tool 2016

http://www.microsoft.com/en-us/download/details.aspx?id=49168

ThreatModeler – Web Based (in-house) Tool

http://myappsecurity.com

ThreadFix

http://www.denimgroup.com/blog/denim_group/2016/03/threadfix-in-action-tracking-threats-and-threat-models.html

IriusRisk Software Risk Manager

https://iriusrisk.continuumsecurity.net

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 Proactive Controls 2016
https://www.owasp.org/index.php/OWASP_Proactive_Controls

Questions?

Contacts

Web Site: https://roberthurlbut.com

Twitter: @RobertHurlbut, @AppSecPodcast

Email: robert at roberthurlbut.com

ROBERT HURLBUT CONSULTING SERVICES

securedevelop.NET