

Hands-On Threat Modeling Workshop

Tue, 1/14/2025 8:00 am – Sandusky, Ohio, United States



Robert Hurlbut



Who am I?



X (Twitter): [@RobertHurlbut](https://twitter.com/RobertHurlbut)

BlueSky: [roberthurlbut.bsky.social](https://bsky.social/roberthurlbut)

LinkedIn: [roberthurlbut](https://www.linkedin.com/in/roberthurlbut)

Discord: robert.ct (robertct)

Robert Hurlbut

Principal Application Security Architect /
Threat Modeling Lead

@ Aquia, Inc. (<https://aquia.us>)



- Microsoft MVP – Dev Sec / Dev Tech
- (ISC2) CSSLP
- Boston Code Camp – Co-Organizer
- Boston .NET Architecture Group – Founder / Leader
- Amherst Security Group – Leader
- Application Security Podcast – Co-Host
- “Threat Modeling Manifesto” – Co-Author
- “Threat Modeling Capabilities” – Co-Author
- Threat Modeling Connect – Co-Founding Member
- Expert Witness (Threat Modeling, Cybersecurity)
- Ph.D. Student – Space Cybersecurity



Agenda

Why, What, When, Who - Threat Modeling?

Threat Modeling: Getting Started

Threat Modeling Process

- Hands-on Exercises / Labs

Threat Modeling Tools

- Hands-on Exercises / Labs

Threat Modeling in an Agile/DevOps (and AI) World

What's next?

Need CPEs?
This workshop
is worth 4
CPEs. Contact
me afterward
for a form.



Pre-Compiler Lab Materials

<https://prereqs.codemash.org>

or

<https://github.com/rhurlbut/CodeMash2025>



Agenda

Why, What, When, Who - Threat Modeling?

Threat Modeling: Getting Started

Threat Modeling Process

- Hands-on Exercises / Labs

Threat Modeling Tools

- Hands-on Exercises / Labs

Threat Modeling in an Agile/DevOps (and AI) World

What's next?

**Need CPEs?
This workshop
is worth 4
CPEs. Contact
me afterward
for a form.**



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 anticipate things going wrong



Why Threat Modeling? (continued)

Find
Architecture / Design
Flaws Impossible
by other Methods

Identifies most
“at-risk”
components

Helps prioritize
security
remediation

Think about
Application
Attack Surfaces

Increases maturity in
Security Practices

Provides basis for
abuse cases –
“think like an attacker”



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
- When we look around to cross the street



What is Threat Modeling? (cont.)

When we think ahead on:

What could go wrong (*ask “what if” questions*)

Weigh risks

Act accordingly

... we are **“threat modeling”**



What is Threat Modeling? (cont.)

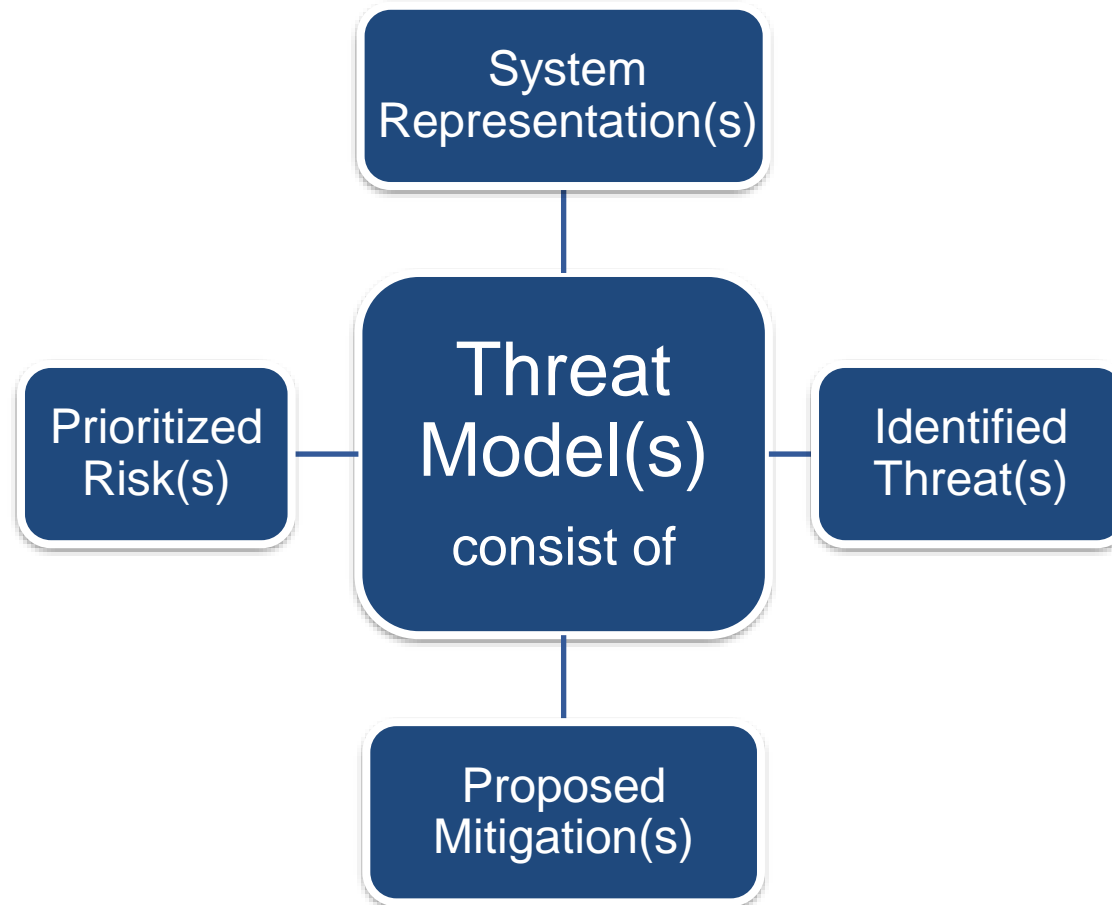
Threat Modeling

Analyzing representations of a system to highlight concerns about security and privacy characteristics*



* Threat Modeling Manifesto, 2020 – <https://threatmodelingmanifesto.org/>

What is Threat Modeling? (cont.)



Threat models all around us ...

System/situation:

Catching a flight



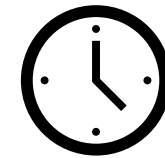
Mitigations?

Set alarm

Leave early

Bring a book

Reschedule



What could go wrong?

Miss the flight

Miss boarding

Delays

Cancelled



Anything else to help?

Ticket ready

Prepare luggage



Who should do Threat Modeling?

You

Everyone

Anyone who is concerned about the privacy, safety, and security of their system*



* Threat Modeling Manifesto, 2020 – <https://threatmodelingmanifesto.org/>

When do you do Threat Modeling?

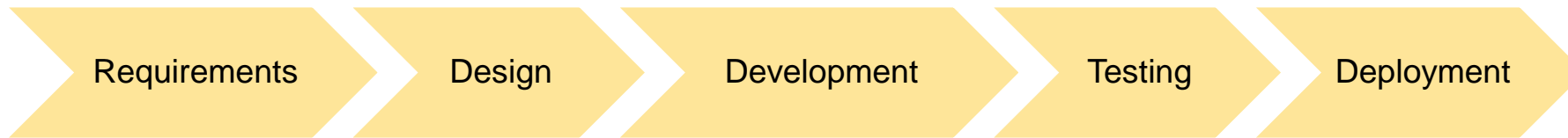


When do you do Threat Modeling? (continued)

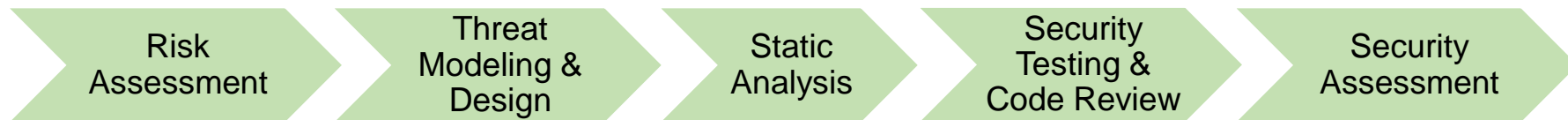


When do you do Threat Modeling? (continued)

SDLC* Process



Secure SDLC* Process

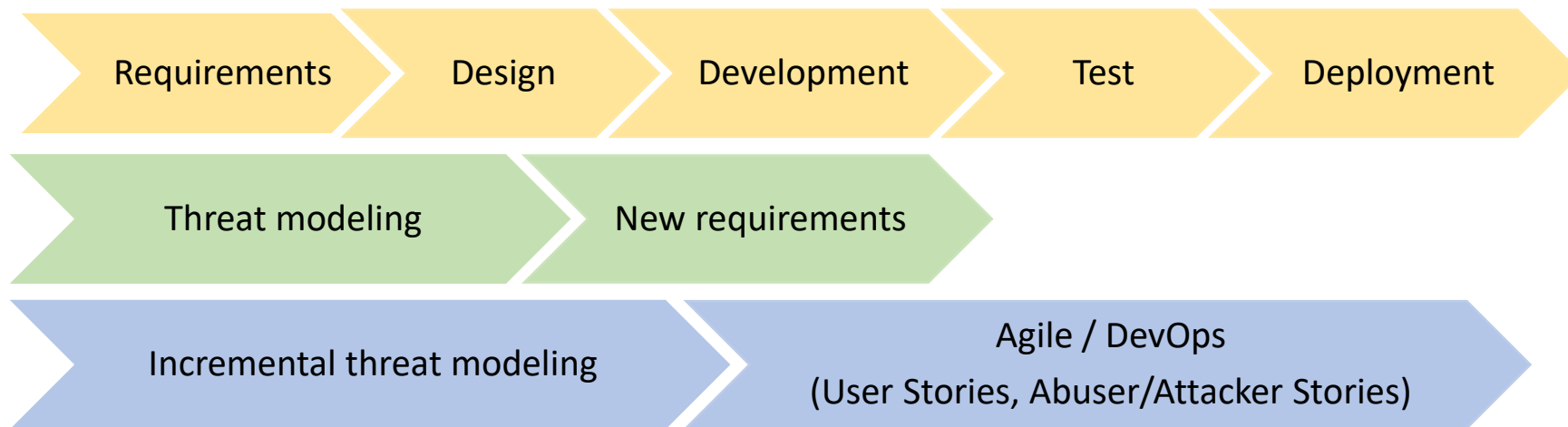


* SDLC = Software Development Life Cycle



When do you do Threat Modeling? (continued)

In SDLC* – Requirements and Design phase(s):



* SDLC = Software Development Life Cycle

Definitions

Term	Definition
Threat Actor	Someone motivated to do harm to an Asset
Asset	Something of value we want to protect
Vulnerability	Opening in a system that helps Threat Actor realize Threat
Threat	Exploits the Vulnerability (intentional or accidental) to control, damage, or destroy Asset or its Data
Attack	Successful exploitation of a Threat
Risk	Likelihood and Impact of Threat being realized
Control	Mitigation/Countermeasure to minimize Threat



Approaches to Threat Modeling

Asset-centric

Software-centric

Attacker-centric

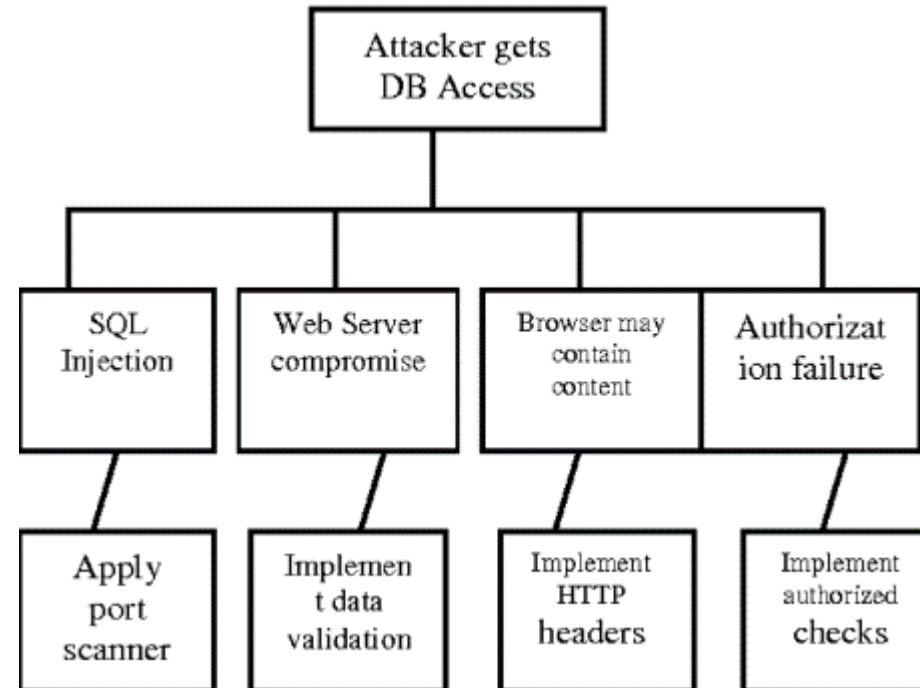


Approaches to Threat Modeling – Asset-centric

Assets

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

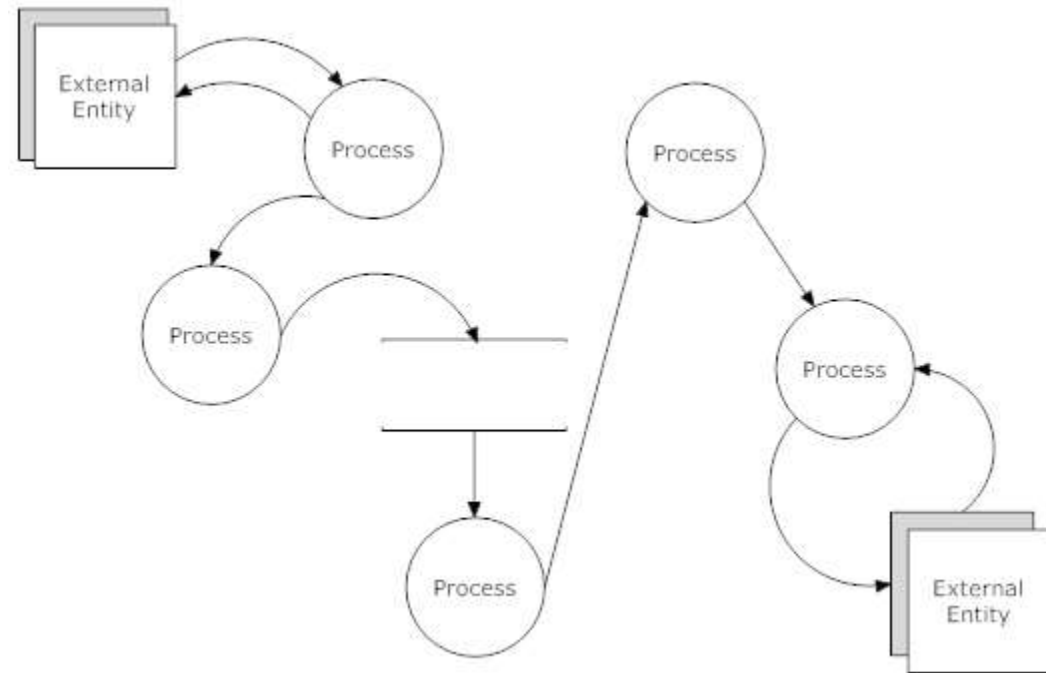
Attack trees



Secure Design

Understanding
secure activity
within an
architecture

Data Flow Diagram(DFD)



Approaches to Threat Modeling – Attacker-centric

Profiles

Script Kiddie

Hacktivist

Nation-state attacker

Patterns

Copies scripts – try anything

Political agenda – deface website

Money, intellectual property theft - phishing



Threat Modeling your House



Asset-centric

Family, irreplaceable photos, valuable artwork

Software-centric

Physical features/entry points (front and back door)

Attacker-centric

Who might break in, the current security system

Approaches to Threat Modeling

Asset-centric

Software-centric ←

*Today – we'll mainly focus on
Software-centric*

Attacker-centric

Agenda

Why, What, When, Who - Threat Modeling?

Threat Modeling: Getting Started

Threat Modeling Process

- Hands-on Exercises / Labs

Threat Modeling Tools

- Hands-on Exercises / Labs

Threat Modeling in an Agile/DevOps (and AI) World

What's next?

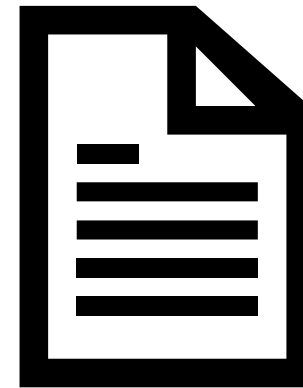
Need CPEs?
This workshop
is worth 4
CPEs. Contact
me afterward
for a form.



Getting Started - Simple Tools



Diagramming
(Whiteboard -
Real or Virtual)



Documenting
(Word / Excel)
(Confluence / Jira)

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							



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



<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 to understand 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



Agenda

Why, What, When, Who - Threat Modeling?

Threat Modeling: Getting Started

Threat Modeling Process

- **Hands-on Exercises / Labs**

Threat Modeling Tools

- Hands-on Exercises / Labs

Threat Modeling in an Agile/DevOps (and AI) World

What's next?

**Need CPEs?
This workshop
is worth 4
CPEs. Contact
me afterward
for a form.**



Threat Modeling Process

At the highest levels, when we threat model, we ask four key questions*:

1.

What are we working on?

2.

What can go wrong?

3.

What are we going to do about it?

4.

Did we do a good enough job?



* Threat Modeling Manifesto, 2020 – <https://threatmodelingmanifesto.org/>

Threat Modeling Process

1. *What are we working on?*

Understand/diagram your system

2. *What could go wrong?*

Identify threats through answers to questions

3. *What are we going to do about it?*

Determine mitigations and risks

4. *Did we do a good enough job?*

Review and follow through



Threat Modeling Process

1. *What are we working on?*

Understand/diagram your system

2. *What could go wrong?*

Identify threats through answers to questions

3. *What are we going to do about it?*

Determine mitigations and risks

4. *Did we do a good enough job?*

Review and follow through



1. Understand / diagram your system

Gather Team

Software Developers / Testers, Architects, Project Managers,

Automation Engineers / Code Release Manager, Security Champions,

Other Stakeholders

Domain Knowledge

Business / Technical Goals

Focused Sessions (1-2 hours max, or per Sprint / per Story)

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

Be sure to document what you learn!



1. Understand / diagram your system

You can use an Architecture or Network diagram

In many cases, a Data Flow Diagram (DFD) is very useful for Threat Modeling

Draw a Data Flow Diagram (DFD)

Notation element	Reference	Examples
	External entity	People (e.g., users), systems (e.g., other devices), cloud services, browsers
	Process	DDL, exe(D)COM, web service, virtual machine, threat
	Data store	File, database, registry, cache, cookie
	Data flow	http request or response, remote procedure call, UDP communication
	Trust boundary (inside you trust the processes and data stores, outside you don't)	Device boundary, process boundary

You can use the drawing tool of choice – however, try to stay with the basic shapes and meanings for consistency

1. Understand / diagram your system

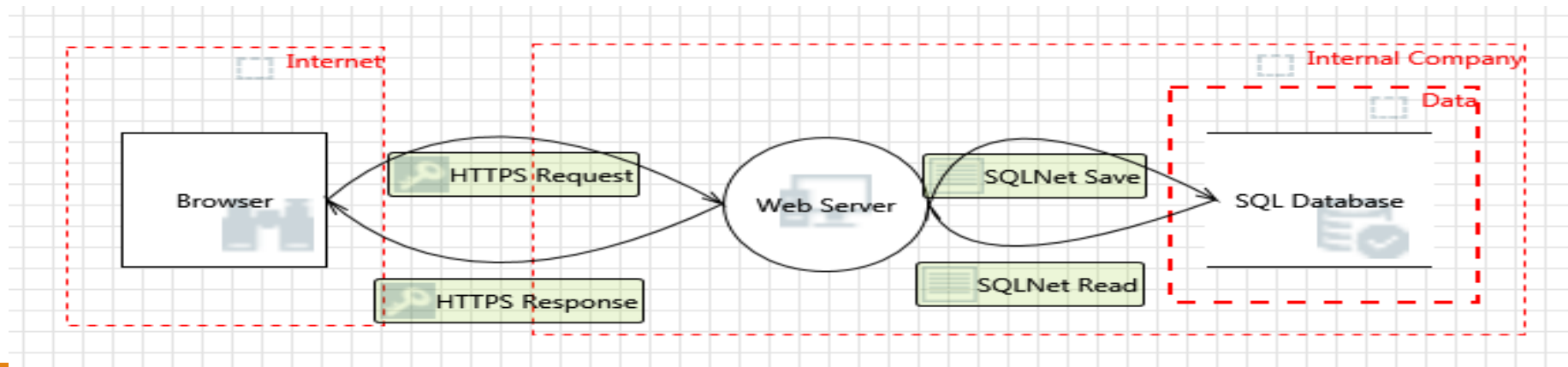
How do the External Entities, Processes, and Data Stores connect?

Connect the information points with the Data Flow arrows.

Where are the Trust Boundaries?

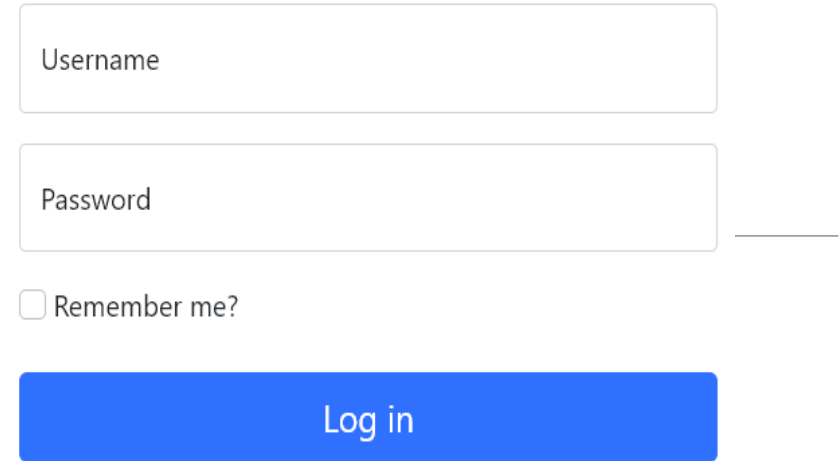
For example:

- Browser (external entity) sends/receives data (data flow) with a web server/app (process) which saves/reads data (data flow) using a SQL Database (data store)
- Trust Boundaries indicate where trust changes — Authenticate / Authorize / Validate



Sample Architecture

- User enters a web browser's login credentials (username and password).



Username

Password

☐ Remember me?

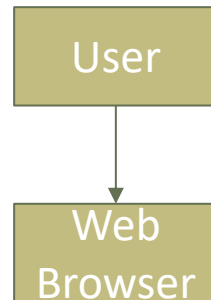
Log in

- The web browser sends them to the cloud provider's internet gateway.
- The internet gateway forwards the request to the load balancer, which receives the login credentials, creates a session token for the user, and sends it back to the web browser.
- The web browser stores the session token as a cookie or in local storage.
- The load balancer allows the user to access resources in the cloud environment.



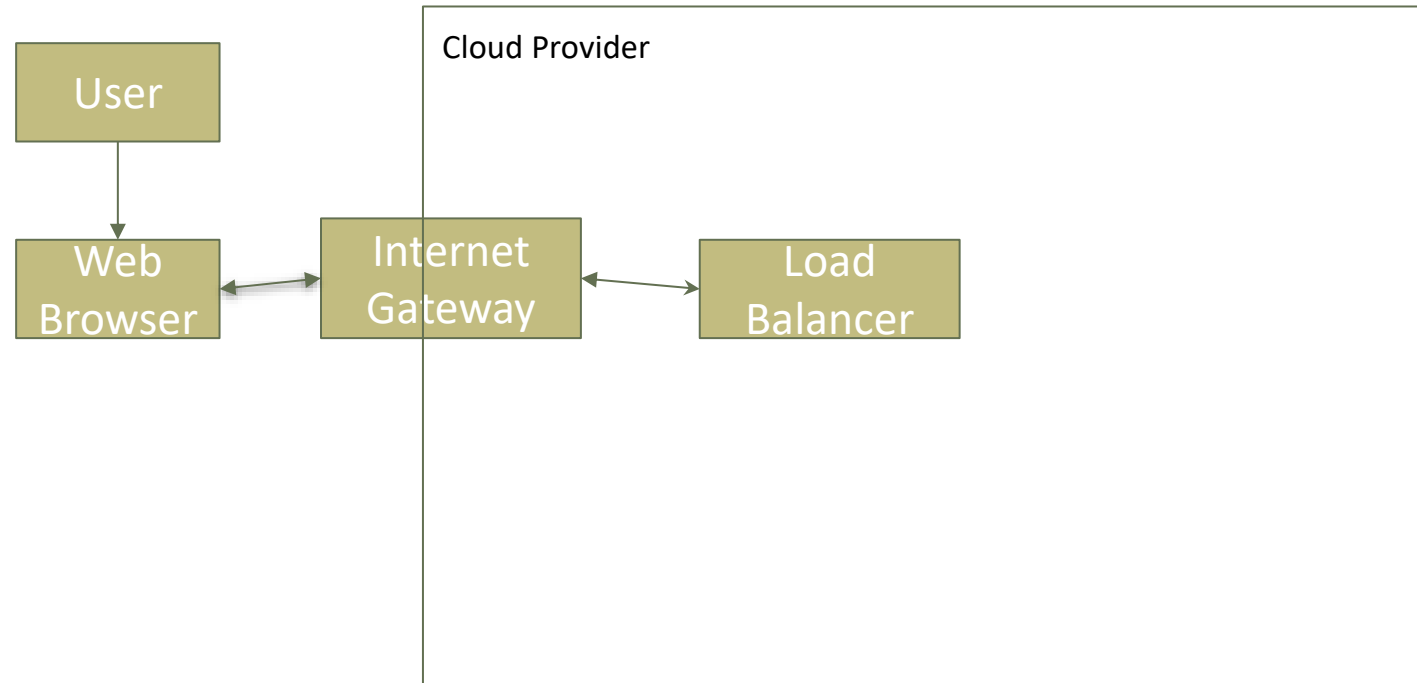
Building a Data Flow Diagram (DFD)

- User enters a web browser's login credentials (username and password).



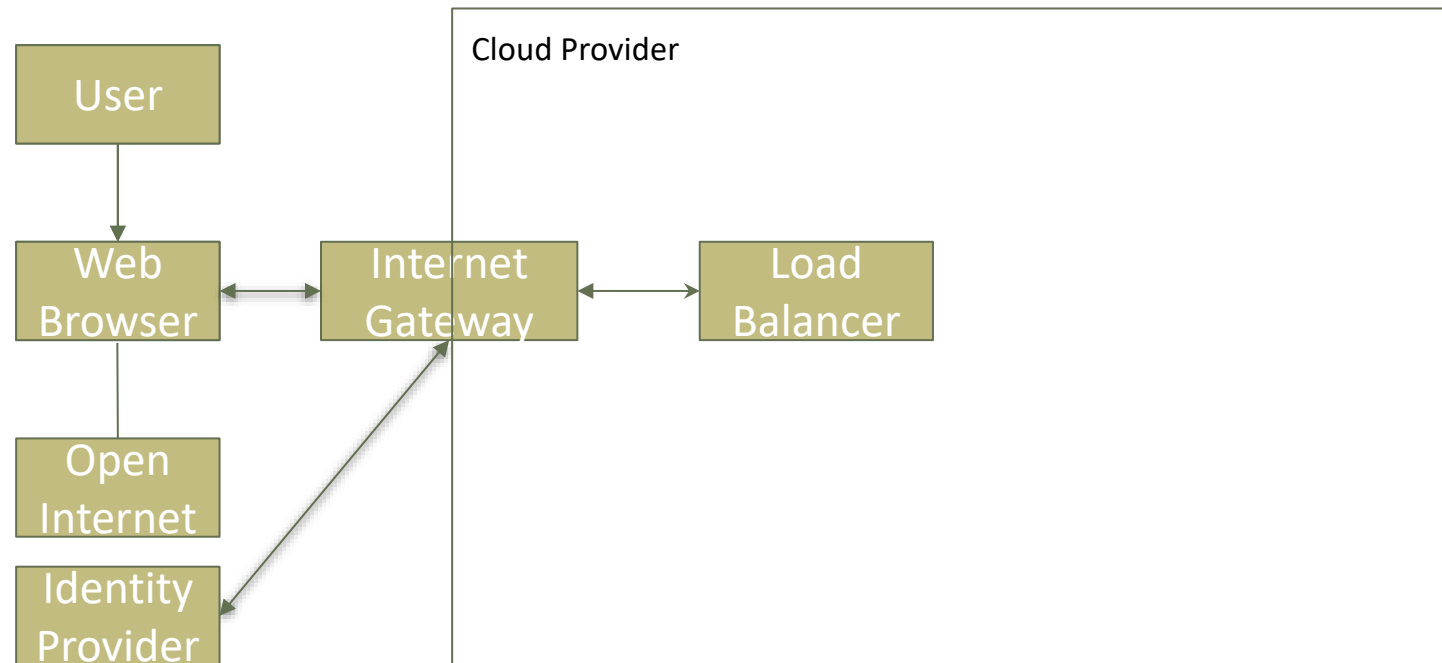
Building a Data Flow Diagram (DFD)

- User enters a web browser's login credentials (username and password).
- The web browser sends them to the internet gateway.
- The gateway sends the request to the load balancer to establish a session



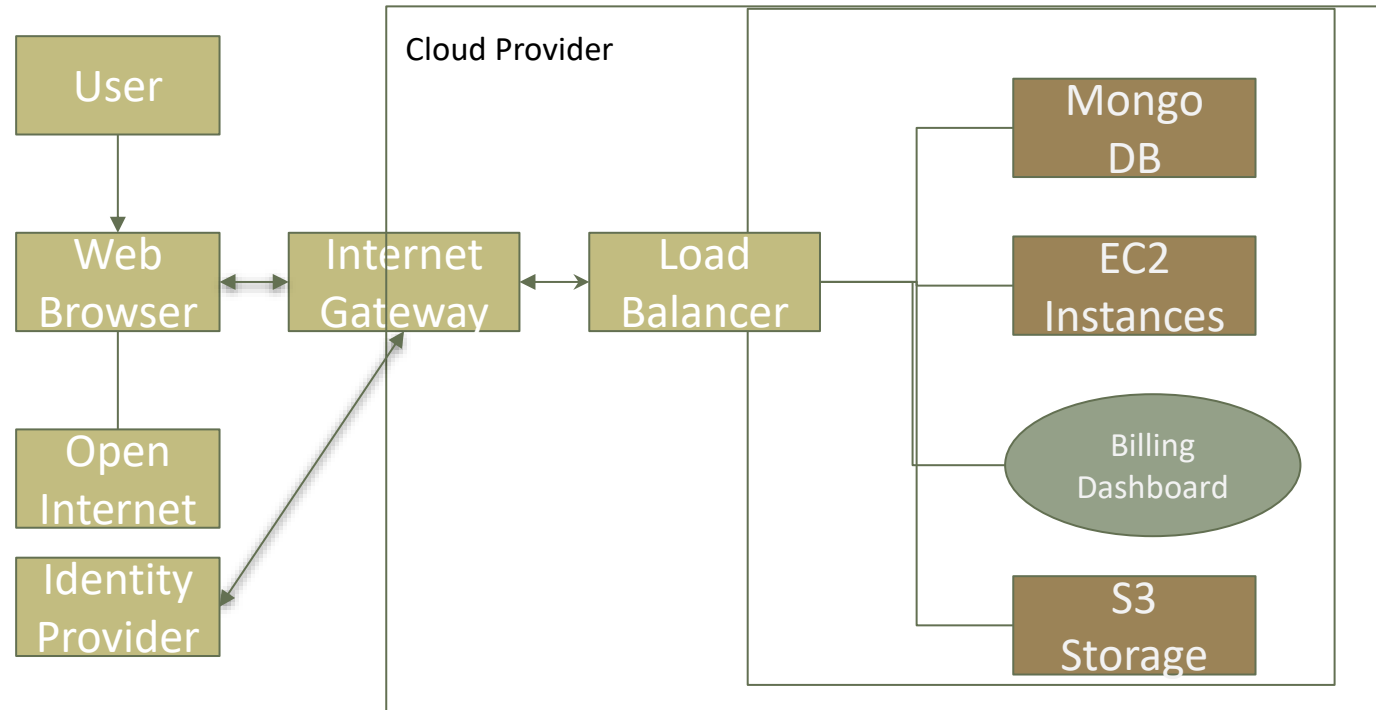
Building a Data Flow Diagram (DFD)

- User enters a web browser's login credentials (username and password).
- The web browser sends them to the cloud provider.
- The internet gateway validates against the identity provider, creates a session token for the user, and sends it back to the web browser.

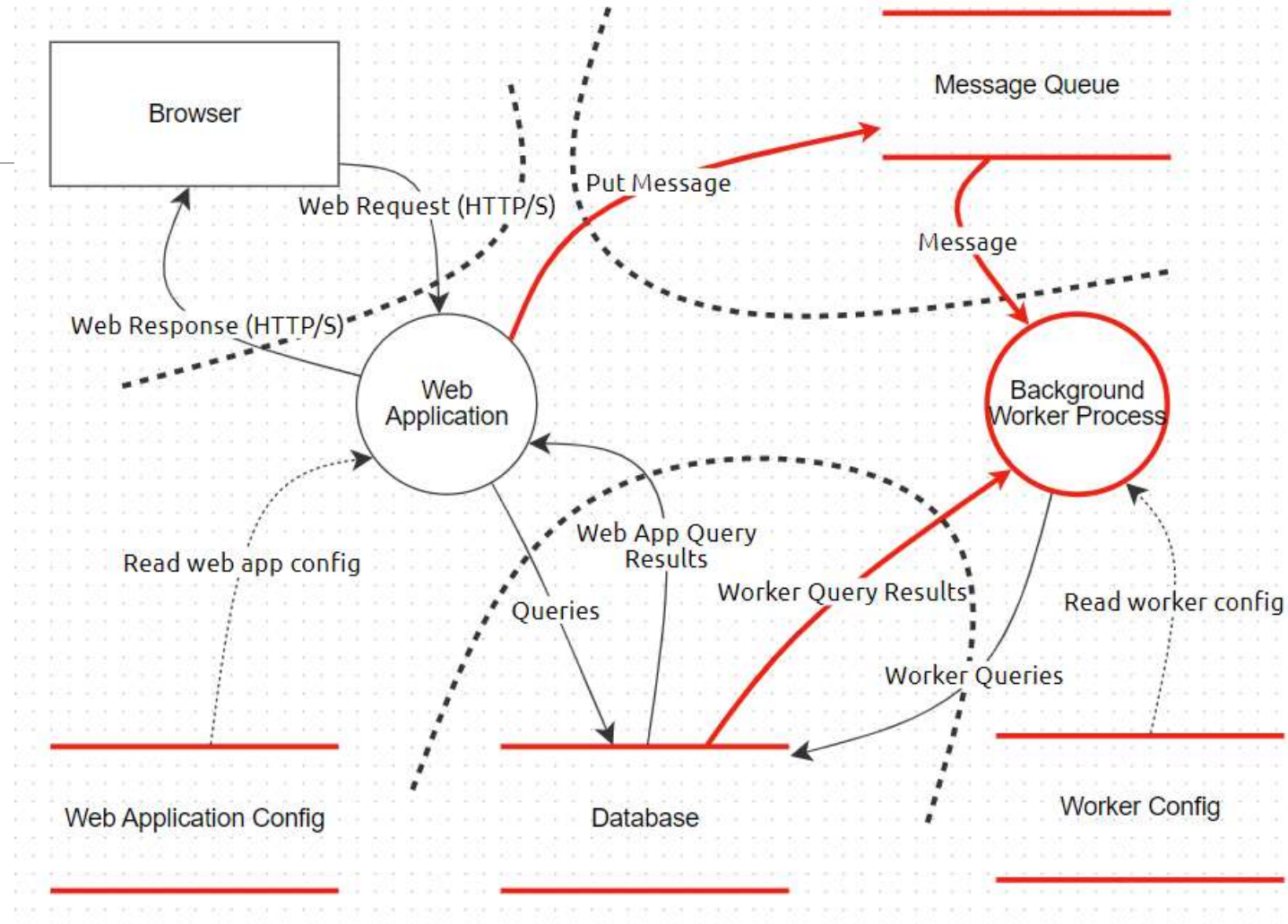


Building a Data Flow Diagram (DFD)

- User enters a web browser's login credentials (username and password).
- The web browser sends them to the cloud provider.
- The web browser can connect to the Billing Dashboard, which uses various data storage options (MongoDB, S3 Storage, etc.)



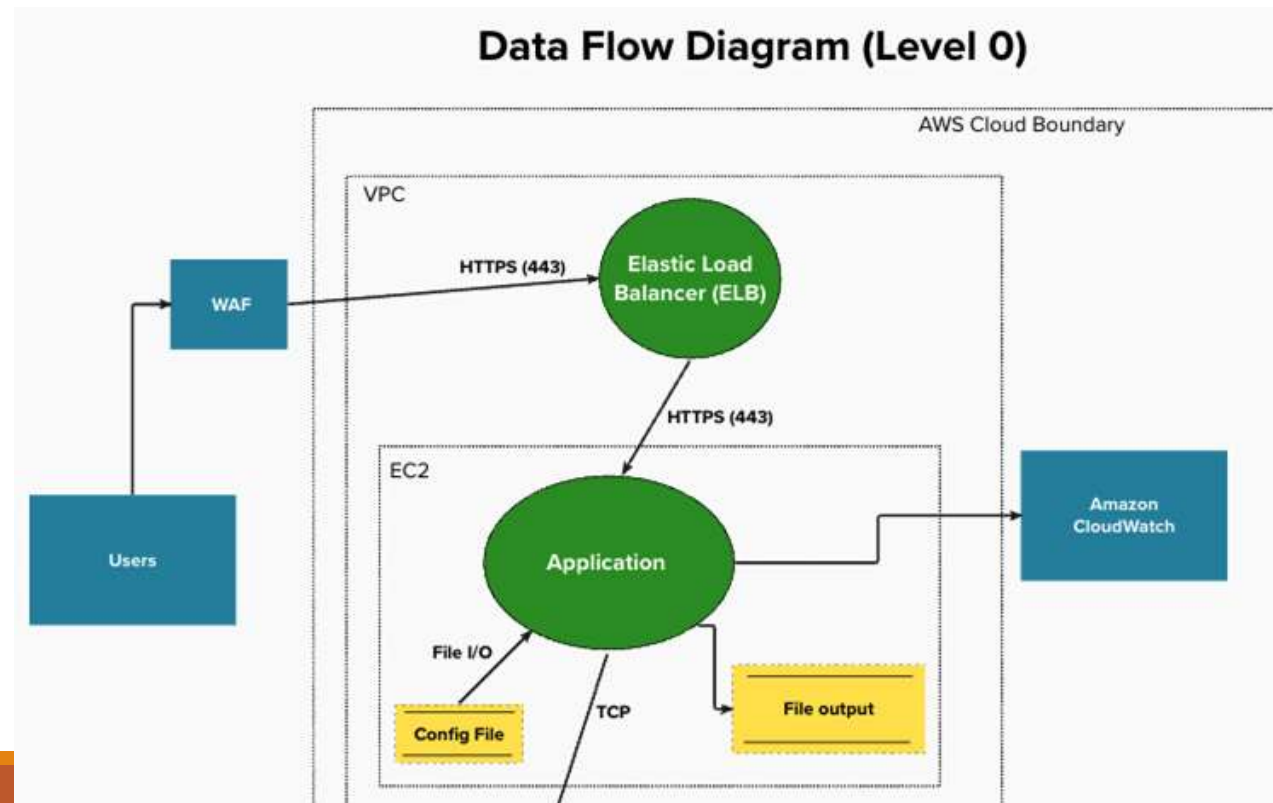
Example Data Flow Diagram (DFD)



(Sample DFD created with OWASP Threat Dragon 2.0)

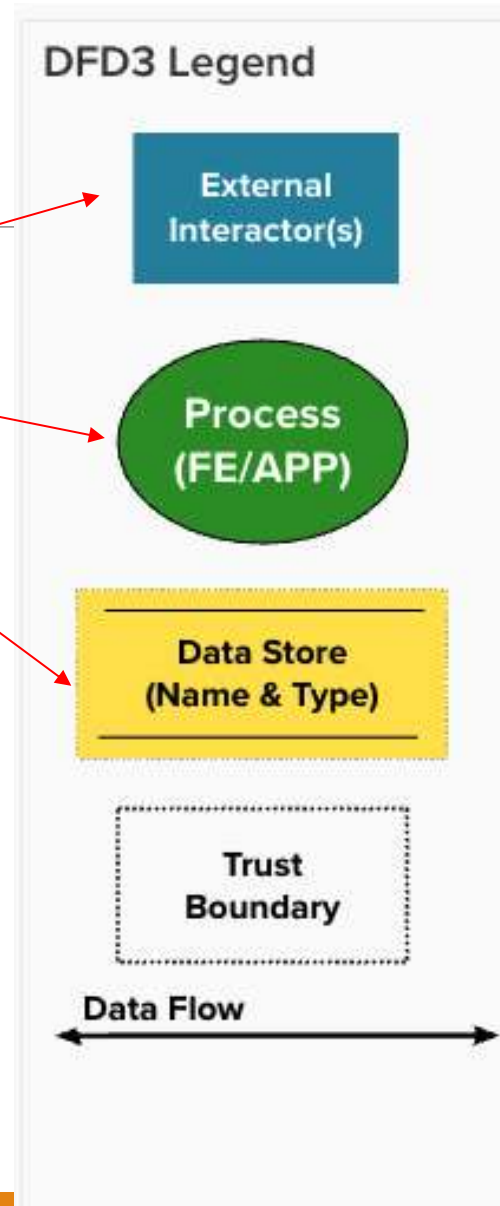
Important Considerations

1. Identify the system boundaries
2. Identify the data flows
3. Identify the assets
4. Consider multiple avenues of access
5. Consider privilege levels



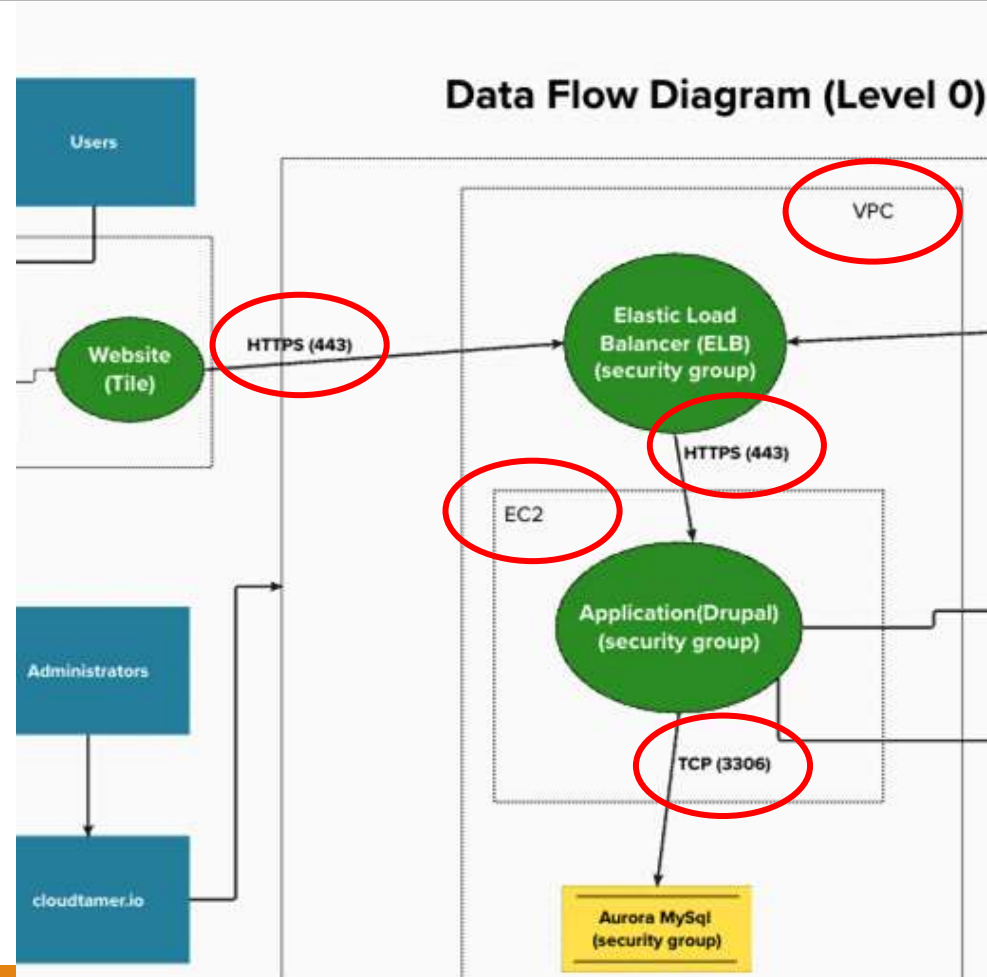
DFD Dos

- Do use standard symbols



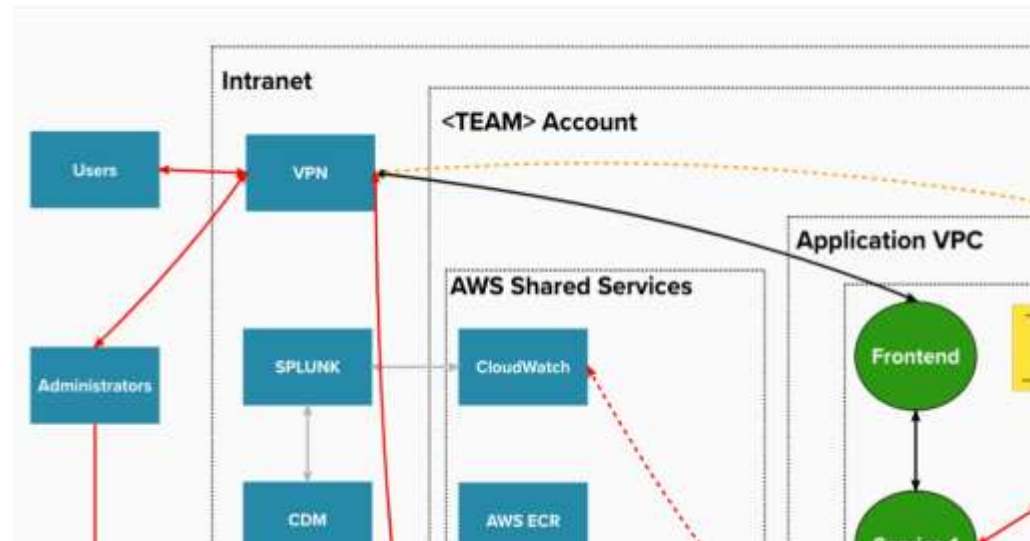
DFD Dos

- Do use standard symbols
- Do label everything



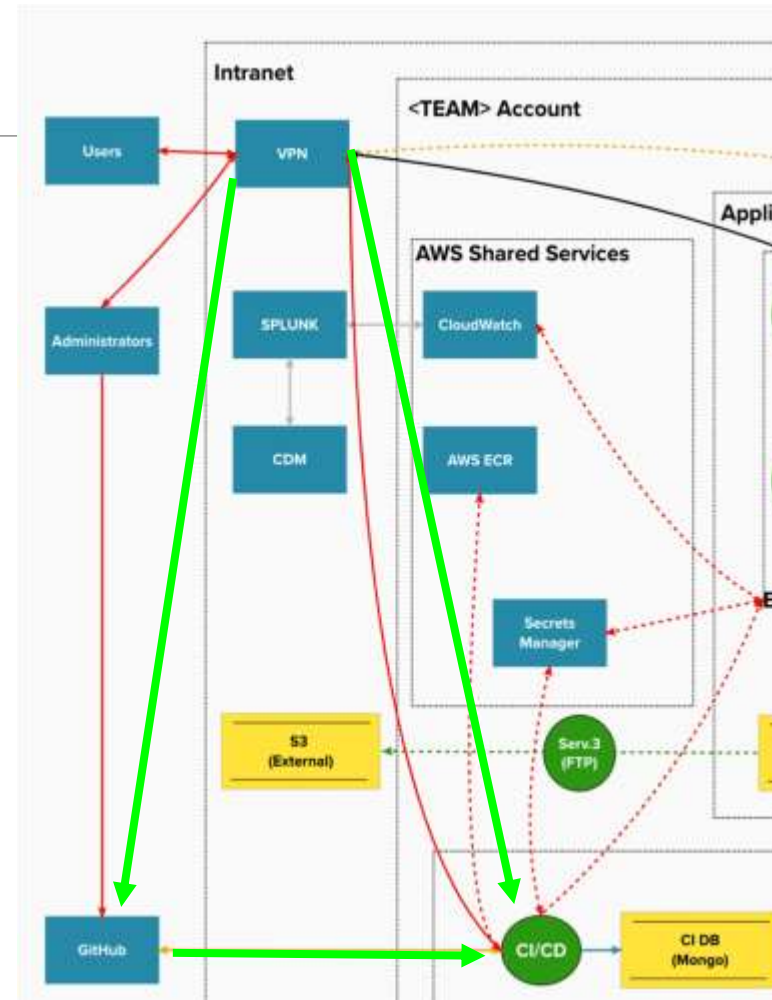
DFD Dos

- Do use standard symbols
- Do label everything
- Do use layers



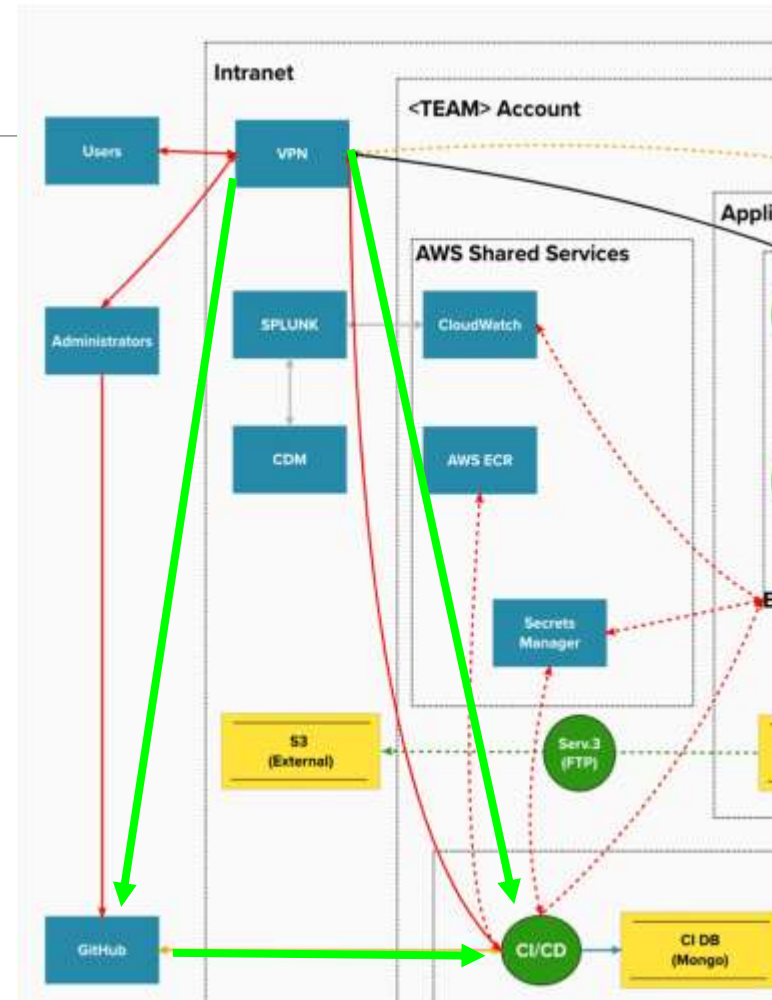
DFD Dos

- Do use standard symbols
- Do label everything
- Do use layers
- Do consider different perspectives



DFD Dos

- Do use standard symbols
- Do label everything
- Do use layers
- Do consider different perspectives
- Do validate the diagram



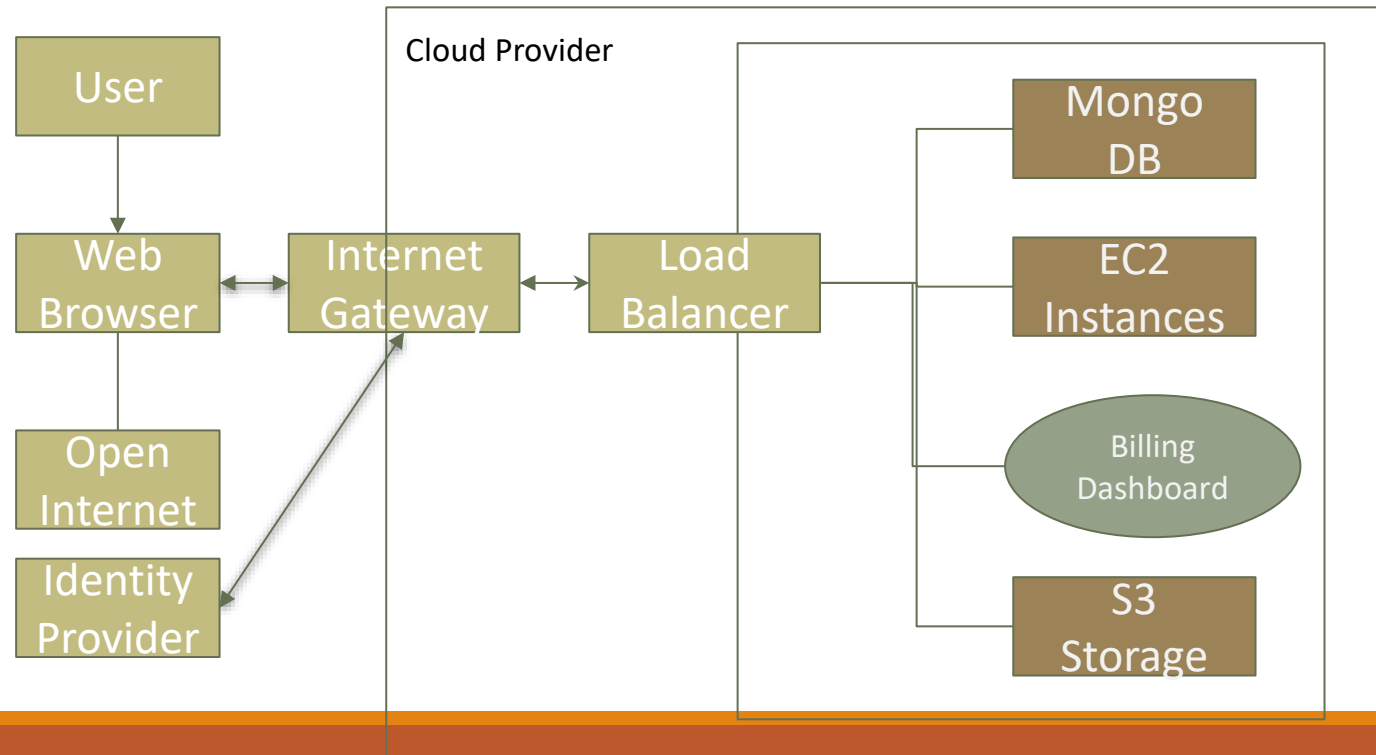
DFD Don'ts

- Don't make the diagram too complex
- Don't use too many data stores
- Don't include implementation details
- Don't omit essential elements
- Don't forget to update the diagram



Building a Data Flow Diagram (DFD)

- User enters a web browser's login credentials (username and password).
- The web browser sends them to the cloud provider.
- The web browser can connect to the Billing Dashboard, which uses various data storage options (MongoDB, S3 Storage, etc.)



Threat Modeling Lab 1:

Review case study

Draw a Data Flow Diagram (DFD)



Threat Modeling Process

1. *What are we working on?*

Understand/diagram your system

2. *What could go wrong?*

Identify threats through answers to questions

3. *What are we going to do about it?*

Determine mitigations and risks

4. *Did we do a good enough job?*

Review and follow through



2. Identify Threats – “What can go wrong?”

Conspicuously overloaded truck stopped by State Police - Springfield, MA



“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)

Threat	Examples	Control we want
S poofing	Pretending to be someone else	Identity Assurance
T ampering	Modifying data that should not be modifiable	Integrity
R epudiation (lack of proof)	Claiming someone didn't do something	Non-repudiation (proof – Auditability)
I nformation Disclosure	Exposing information	Confidentiality
D enial of Service	Preventing a system from providing service	Availability
E levation of Privilege	Doing things that one isn't suppose to do	Least Privilege



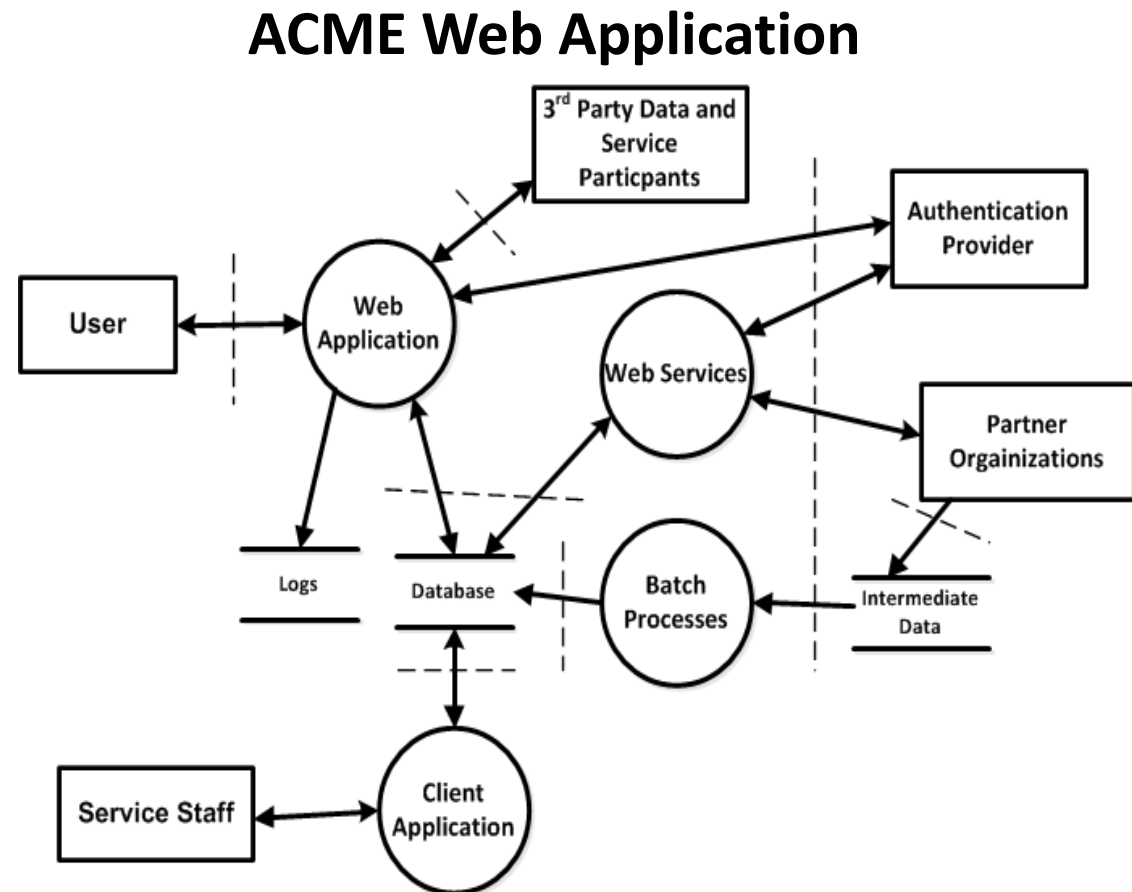
2. Identify Threats – Applying STRIDE to a DFD

Options:

Each part of STRIDE applies to specific elements or interactions.

and/or

You can look at STRIDE per interaction.



Using STRIDE to Identify Threats

Spoofing

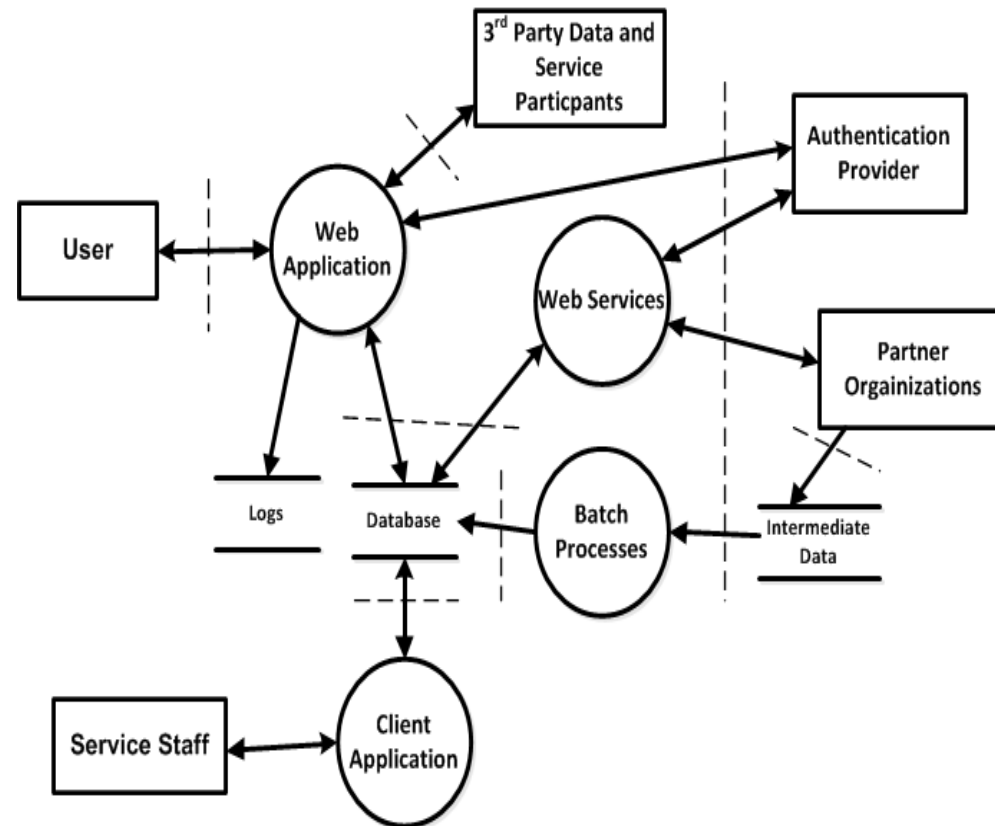
User could be spoofed by an attacker to connect to Web App

Tampering

Requests from User to Web App may be modified

Repudiation

How would we know actions performed by the Web App?



Using STRIDE to Identify Threats

Information

Disclosure

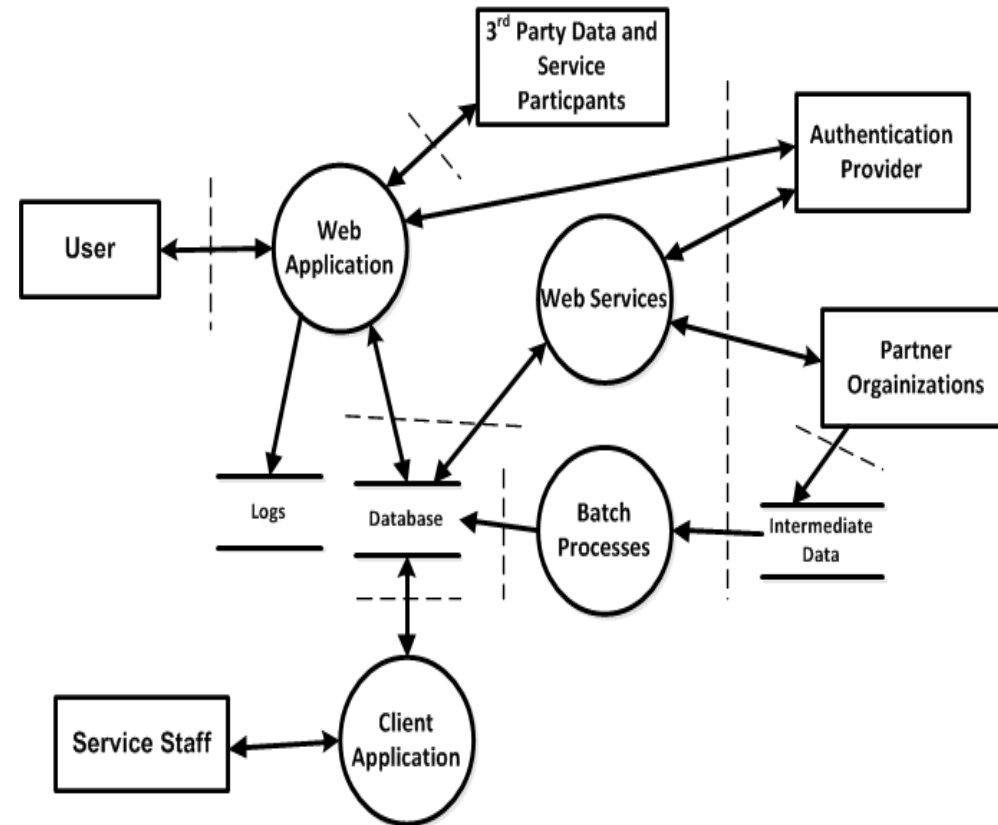
Setting and getting credentials could be exposed in transit

Denial of Service

What happens if Authentication Provider is not available?

Elevation of Privilege

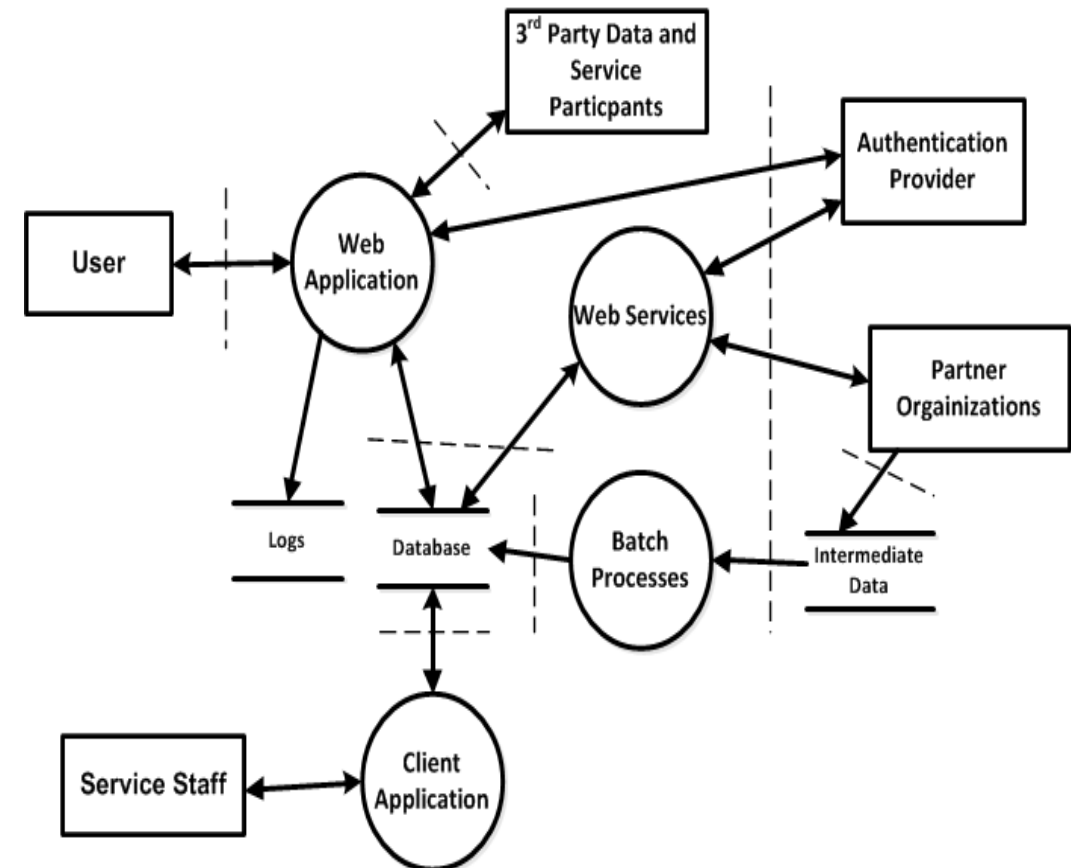
Does audit data have access control for reading?



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 – Many Ways

STRIDE (software-centric)

LINDDUN (privacy-focused)

Attack Trees (asset or attacker-centric)

PASTA (risk-centric)

MITRE ATT&CK (intrusion-centric knowledge base)

Other:

Card Games - OWASP Cornucopia, Elevation of Privilege

Use Cases / Abuse Cases



Threat Modeling Lab 2: Identify Threats



Threat Modeling Process

1. *What are we working on?*

Understand/diagram your system

2. *What could go wrong?*

Identify threats through answers to questions

3. *What are we going to do about it?*

Determine mitigations and risks

4. *Did we do a good enough job?*

Review and follow through



3. Determine mitigations and risks – STRIDE mapped to common controls

STRIDE/Properties	Common controls
(Spoofing) Authentication	<ul style="list-style-type: none">• Authentication based on key exchange• Decide on single factor, two-factor, multi-factor• Offload authentication to another provider• Restrict authentication to certain IP ranges/locations
(Tampering) Integrity	<ul style="list-style-type: none">• Data protected with cryptographic integrity mechanisms• Only enumerated authorized users can modify data
(Repudiation) Non-Repudiation	<ul style="list-style-type: none">• Maintain logs• Digital signatures
(Information Disclosure) Confidentiality	<ul style="list-style-type: none">• Data in files/databases available only to authorized users• Name/existence of database exposed only authorized users• Communication restricted to and between authorized users
(Denial of Service) Availability	<ul style="list-style-type: none">• Rate-limiting or throttling access to a service• Real-time monitoring of log files/other resources - sudden changes
(Elevation of Privilege) Authorization	<ul style="list-style-type: none">• System uses central authorization engine• Use ACLs, limits on writing to higher integrity levels, roles/accounts/permissions for managing access

3. Determine mitigations and risks

Mitigation Options:

Leave as-is

Remove from product

Remedy with technology countermeasure

Warn user

Make the mitigations/countermeasures part of your Security acceptance criteria



3. Determine mitigations and risks

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

Risk is a product of two factors:

Ease of Exploitation

Business Impact

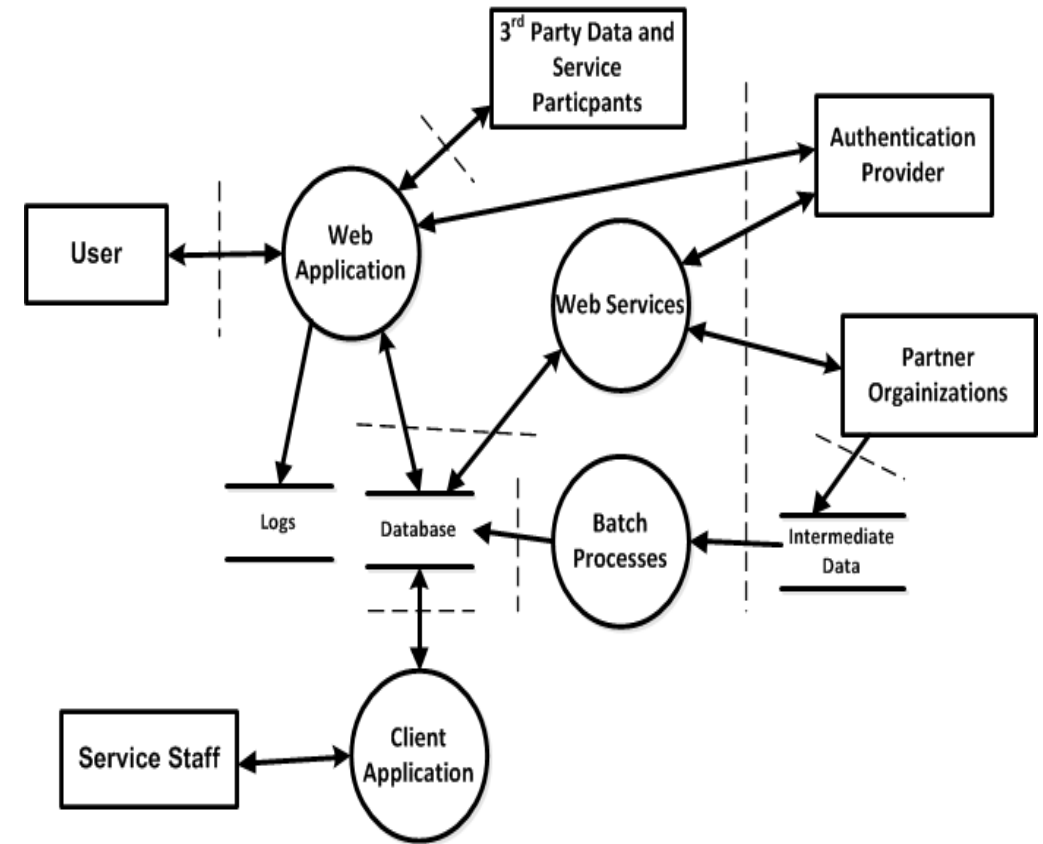
At a bare minimum, use a Risk Rating where the overall risk of a threat is expressed in High, Medium, or Low.



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. *What are we working on?*

Understand/diagram your system

2. *What could go wrong?*

Identify threats through answers to questions

3. *What are we going to do about it?*

Determine mitigations and risks

4. *Did we do a good enough job?*

Review and follow through



4. Review and follow through

Document findings and decisions

File bugs or new requirements (as stories)

Verify bugs fixed / new requirements (stories) implemented

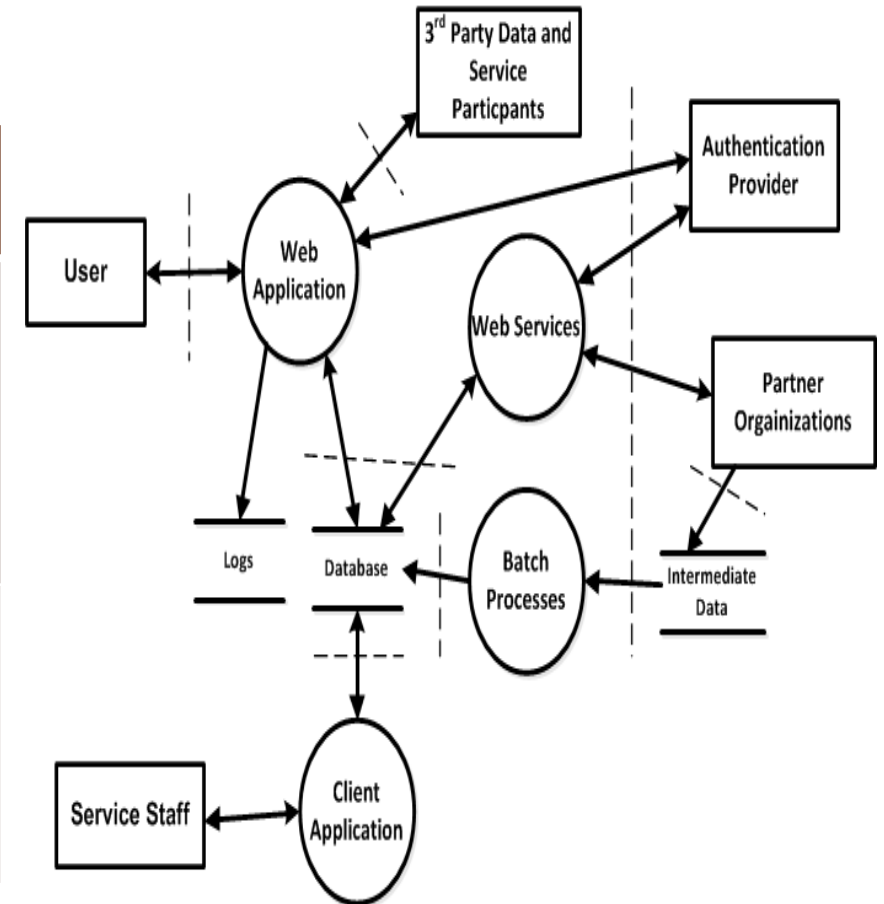
Did we miss anything? Review again

Anything new? Review again

4. Review and follow through

Threat Model for ACME Web Application

Threat	STRIDE	Mitigation / Risk	Review / Action Items
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 the issue in the 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



Repeat or iterate as needed

Consider a baseline threat model for your project if you have never, ever created a threat model before

Then, update and/or review your threat model as you continue to add new features

Agenda

Why, What, When, Who - Threat Modeling?

Threat Modeling: Getting Started

Threat Modeling Process

- Hands-on Exercises / Labs

Threat Modeling Tools

- **Hands-on Exercises / Labs**

Threat Modeling in an Agile/DevOps (and AI) World

What's next?

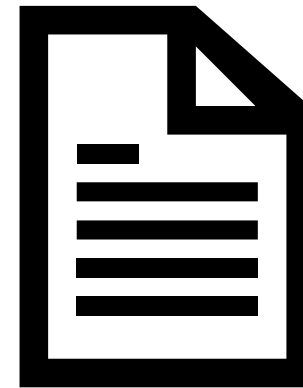
Need CPEs?
This workshop
is worth 4
CPEs. Contact
me afterward
for a form.



Threat Modeling Process - Simple Tools



Diagramming
(Whiteboard -
Real or Virtual)



Documenting
(Word / Excel)
(Confluence / Jira)



Tools	Cost	Platforms
Microsoft Threat Modeling Tool	Free	Desktop, Windows OS Install only
Threats Manager Studio	Free	Desktop, Windows OS Install only
ThreatModeler	Paid	Web-based, In-house or Cloud, CI/CD integration
IriusRisk	Paid	Web-based, In-house or Cloud, CI/CD integration
SD Elements	Paid	Web-based, In-house or Cloud
Devici	Free/Paid	Web-based, Cloud
OWASP Threat Dragon	Free	Web-based, Windows, Mac, Linux installs
Drawing tools – Draw.IO, Mural, Miro, etc.	Free-ish	Web-based, Windows, Mac, Linux installs



The screenshot shows the Threat Modeling Tool 2016 interface. The main workspace displays a diagram of a web application architecture. A Browser (represented by a computer icon) is connected to a Web Application (represented by a circle with a computer icon) via an HTTPS connection (represented by a green box with 'HTTPS'). The Web Application is connected to a SQL Database (represented by a cylinder icon) via an ODBC connection (represented by a green box with 'ODBC'). A dashed red line labeled 'Internet Boundary' separates the Browser from the rest of the system. A dashed red rectangle labeled 'Company' encloses the Web Application and the SQL Database. On the right side, a 'Stencils' panel lists various components. A tooltip for the 'SQL Database' icon is visible, stating 'A representation of a SQL Database.' The bottom status bar shows 'Messages - No issues found' and 'Notes - no entries'.

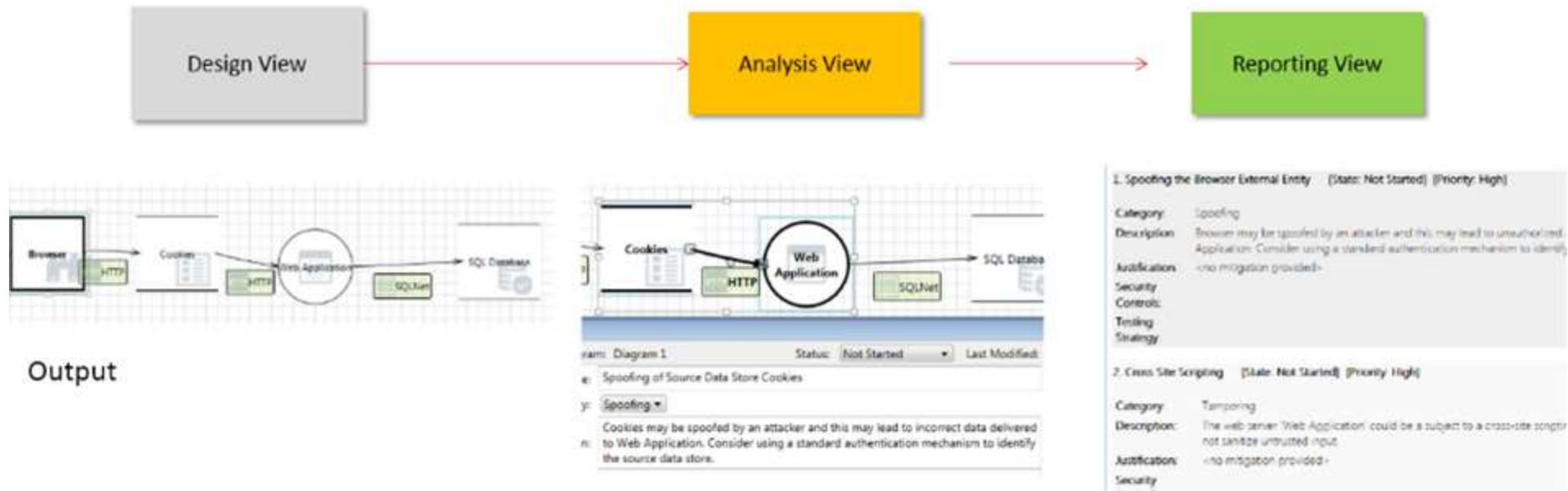
Microsoft Threat Modeling Tool

Input

Data Flows

Threat Vectors,
Mitigating Controls

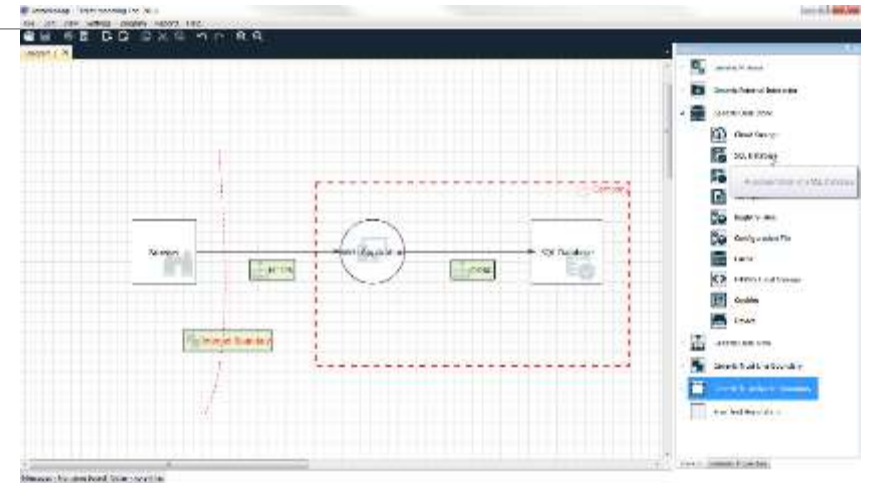
HTML reporting



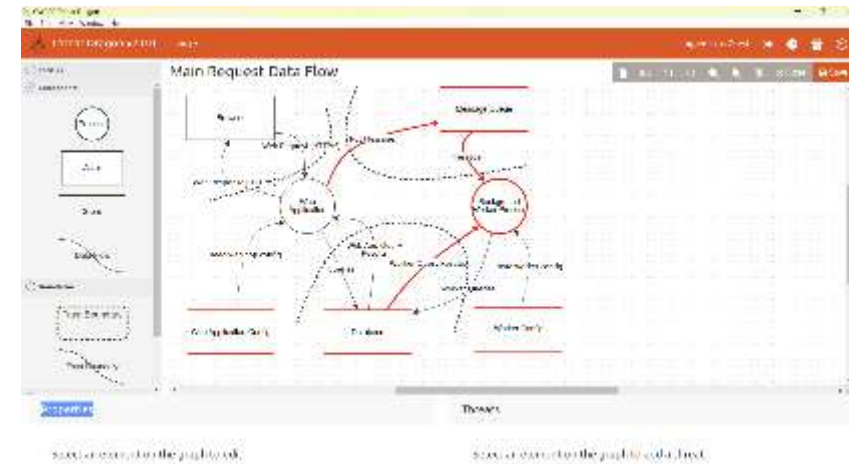
Threat Modeling Tools – MS TMT and OWAS Threat Dragon

Microsoft Threat Modeling Tool

<https://aka.ms/threatmodelingtool>



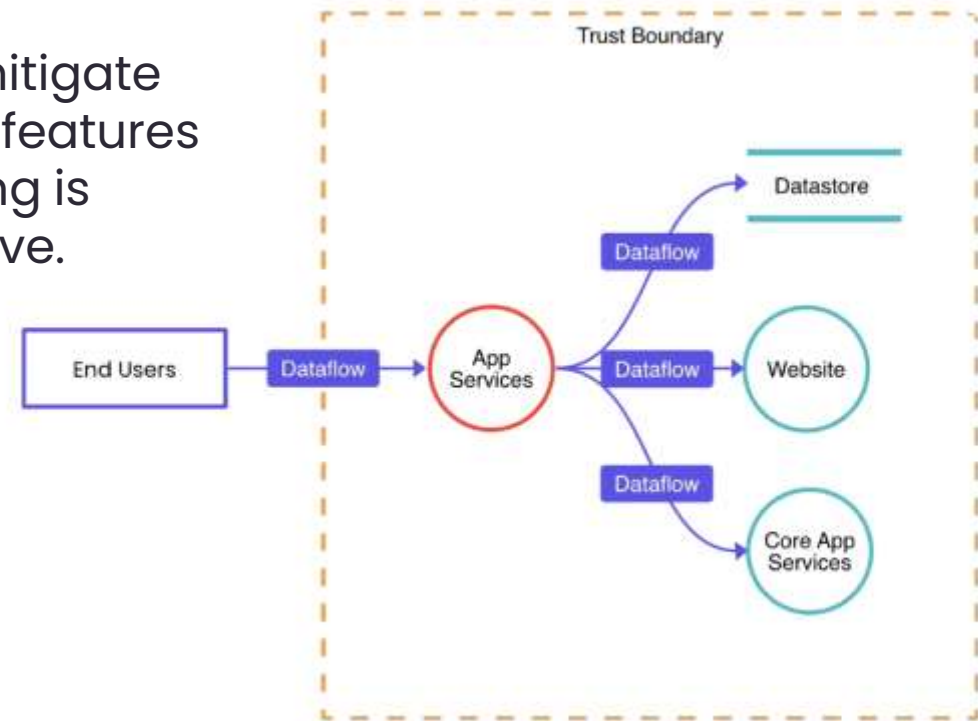
OWASP Threat Dragon



Threat Modeling Tools – Devici

The Devici platform empowers teams to mitigate threats in everything they build. From new features to legacy code, application threat modeling is about to be easy, efficient, and collaborative.

<https://devici.com>



Agenda

Why, What, When, Who - Threat Modeling?

Threat Modeling: Getting Started

Threat Modeling Process

- Hands-on Exercises / Labs

Threat Modeling Tools

- Hands-on Exercises / Labs

Threat Modeling in an Agile/DevOps (and AI) World

What's next?

Need CPEs?
This workshop
is worth 4
CPEs. Contact
me afterward
for a form.



Value of Threat Modeling

Ed Moyle (2017):

*“Very few organizations will have the time or resources to **threat model** their entire ecosystem. Assuming you do not have that luxury, you still can realize quite a bit of **value** just by adopting the mindset of looking for blind spots and questioning assumptions.” **

* (Quoted from an article by Ed Moyle: <https://www.ecommercetimes.com/story/Invisible-Technologies-What-You-Cant-See-Can-Hurt-You-84852.html>)



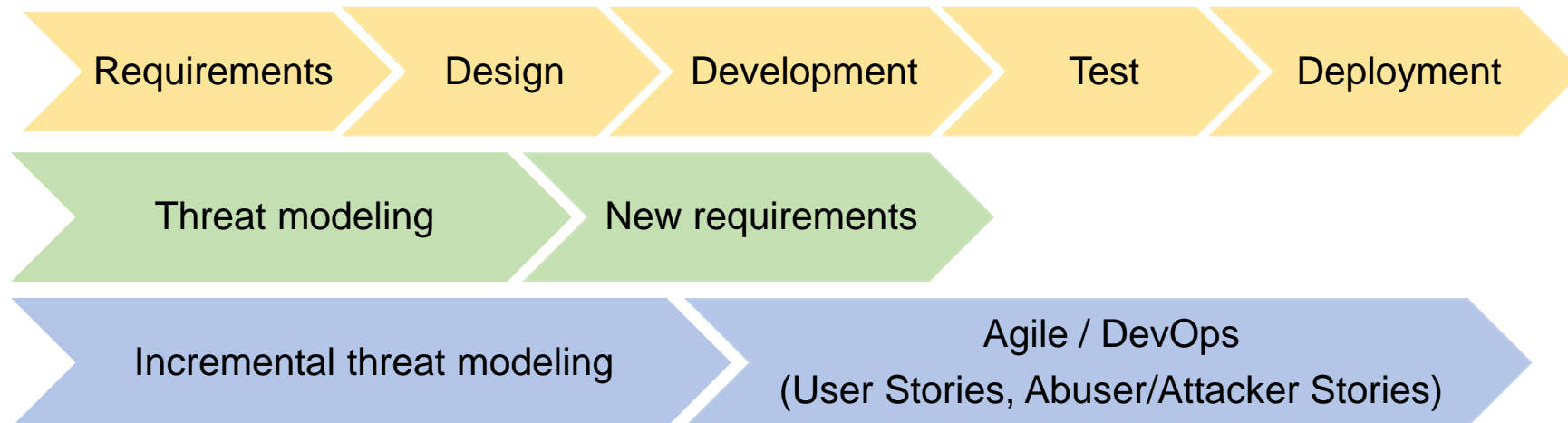
	Waterfall: Threat Model Documents	Agile: Bugs and conversations
System Model	<ul style="list-style-type: none"> • Big complex scope • System diagrams and essays • Gates, dependencies 	<ul style="list-style-type: none"> • Scope tiny: this sprint's change • Big picture as security debt
Finding Threats	<ul style="list-style-type: none"> • Brainstorm • STRIDE • Kill Chain 	<ul style="list-style-type: none"> • Same, aim at in-sprint code
Fixes	<ul style="list-style-type: none"> • Controls • Mitigations • Test Cases 	<ul style="list-style-type: none"> • Spikes to understand • Security-focused stories in sprint, backlog, or epic • Security acceptance criteria
Quality	<ul style="list-style-type: none"> • Test plans 	<ul style="list-style-type: none"> • Test automation

*Adapted from Adam Shostack's talks at BlackHat, other conferences



When do you do Threat Modeling + Agile / DevOps?

In SDLC* – Requirements and Design phase(s):



* SDLC = Software Development Life Cycle

When do you do Threat Modeling + Agile / DevOps?

There are many out-of-band activities (as opposed to inline activities such as coding, etc.)

- Sprint planning

- Spikes

Add Threat Modeling as another out-of-band activity

and/or

In addition to when you create User Stories (or Abuser Stories)

User stories

User stories are typically written like this:

As a <type of user>, I want <some goal> so that <some reason>

Examples:

- As a user, I can backup my entire hard drive.
- As a power user, I can specify files or folders to backup based on file size, date created, and date modified.
- As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saving.

Security User stories

Security user stories are similar to regular user stories, but are sometimes more challenging to manage – there may be too many of them.

Examples:

- As a user, I want to log into the application.
- As a user, I want to be able to see my account information and not other users' information.
- As an admin, I want access to the application's configuration settings.

Abuser / attacker stories

Abuser/attacker stories do this differently:

As <someone with malicious intent>, I want to <do some bad thing>

Examples:

- As a hacker, I want to read the application log files.
- As an insider, I want to access a customer's account information.
- As a disgruntled employee, I want to change product pricing.

See OWASP Abuse Case Cheat Sheet for help in creating these.
https://www.owasp.org/index.php/Abuse_Case_Cheat_Sheet

Abuser stories applied to OWASP Top 10 *

- **A7:2021-Identification and Authentication Failures**

- *Epic:*

Attackers can access hundreds of millions of valid username and password combinations for credential stuffing, default administrative account lists, automated brute force, and dictionary attack tools. Session management attacks are well understood, particularly in relation to unexpired session tokens.

- *Abuse Case:*

As an attacker, I can access hundreds of millions of valid username and password combinations for credential stuffing.

- *Abuse Case:*

As an attacker, I use default administrative account lists, automated brute force, and dictionary attack tools against the application's login areas and support systems.

- *Abuse Case:*

As an attacker, I manipulate session tokens using expired and fake tokens to gain access.

[*https://www.owasp.org/index.php/Abuse_Case_Cheat_Sheet](https://www.owasp.org/index.php/Abuse_Case_Cheat_Sheet)



Typical Threat Modeling Session (Agile / DevOps version)

In Sprint Planning:

- Team
- Focused scope to set of stories
- Understand requirements, and keep business / technical goals in mind

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



Prioritize issues in the backlog

Work through user stories/abuser stories – determine threats and mitigations as you go

As you find issues, write these to the backlog

Prioritize based on risk

Not for Developers only

Operations can bring a great perspective

- Functional misuse
- Creative situations no one considered in design

Bring everyone to the table

Irene Michlin* [@IreneMichlin](#)

“Threat modeling, because you do it by bringing the whole team together at the beginning of each iteration—helps to solve the problems. The people responsible for operations will gain visibility into the development.”

* <https://techbeacon.com/security/threat-modeling-devops-3-lessons-front-lines>



User-Story / Agile / DevOps Approaches



Threat Modeling in the Industry – DevSecOps Case Study

Chapter 5: ***Threat Modeling - A Disaster*** by Edwin Kwan

Edwin Kwan works at an Australian bank

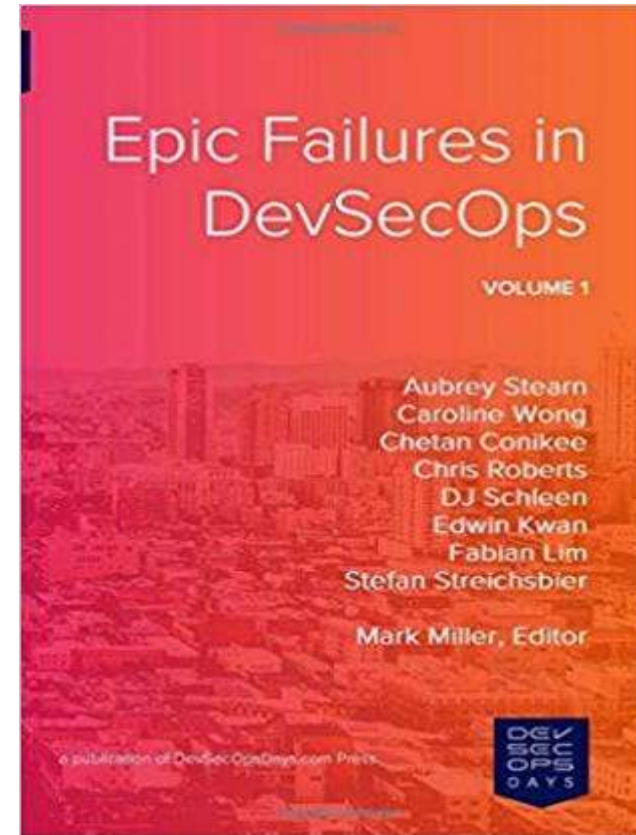
Goal: Threat Modeling for Devs

Looked at: MS Threat Modeling Tool – challenging to scale

Solution: Integrating TM in code/tests

Applying RRA from Mozilla (APIs especially)

Update since book: Dev teams using TM in tests, RRA, Attack Trees



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 risks?

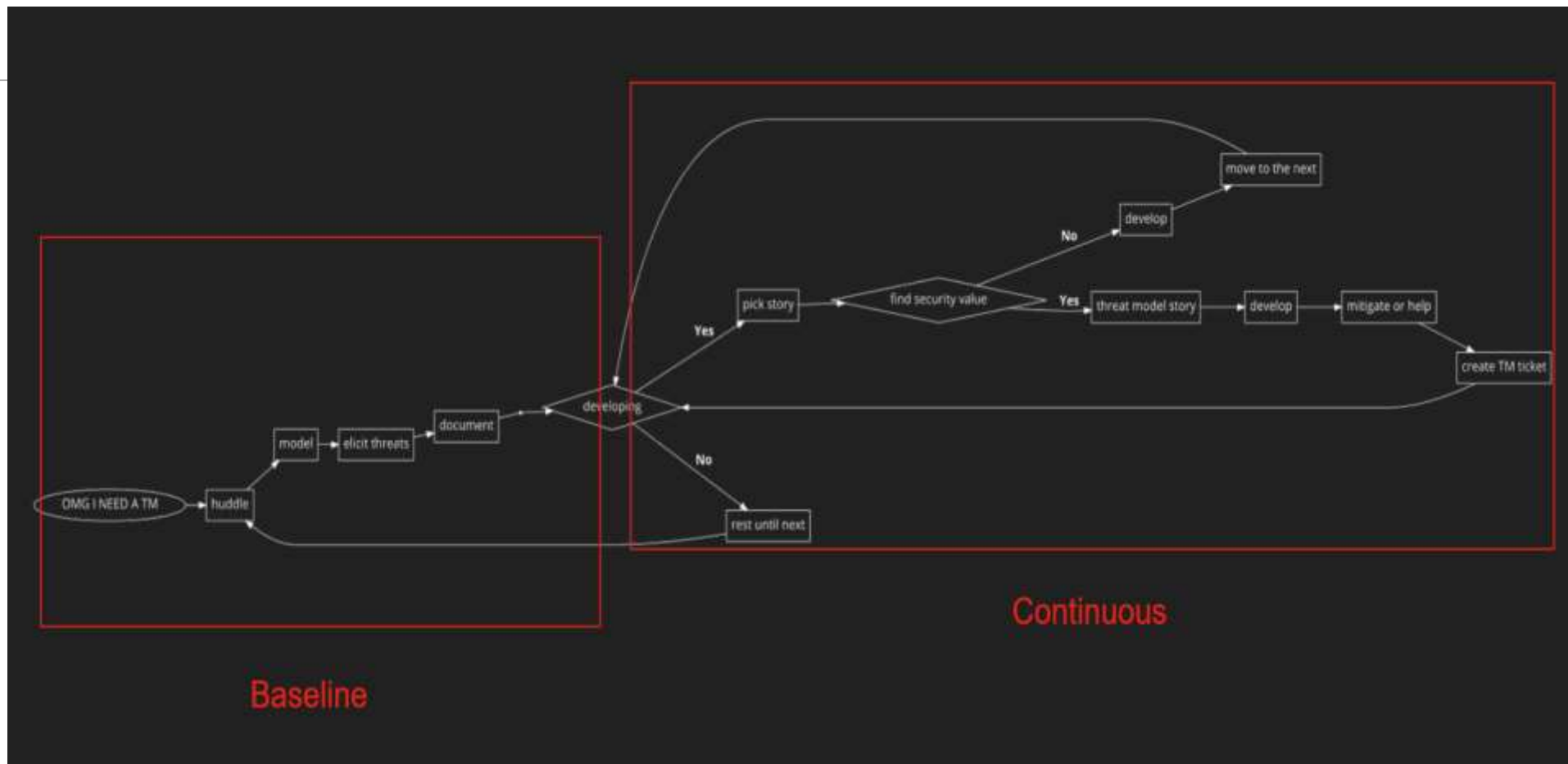
*https://infosec.mozilla.org/guidelines/risk/rapid_risk_assessment.html

Blog post: <https://home.edwinkwan.com/rapid-risk-assessments/>



Continuous Threat Modeling (CTM)*

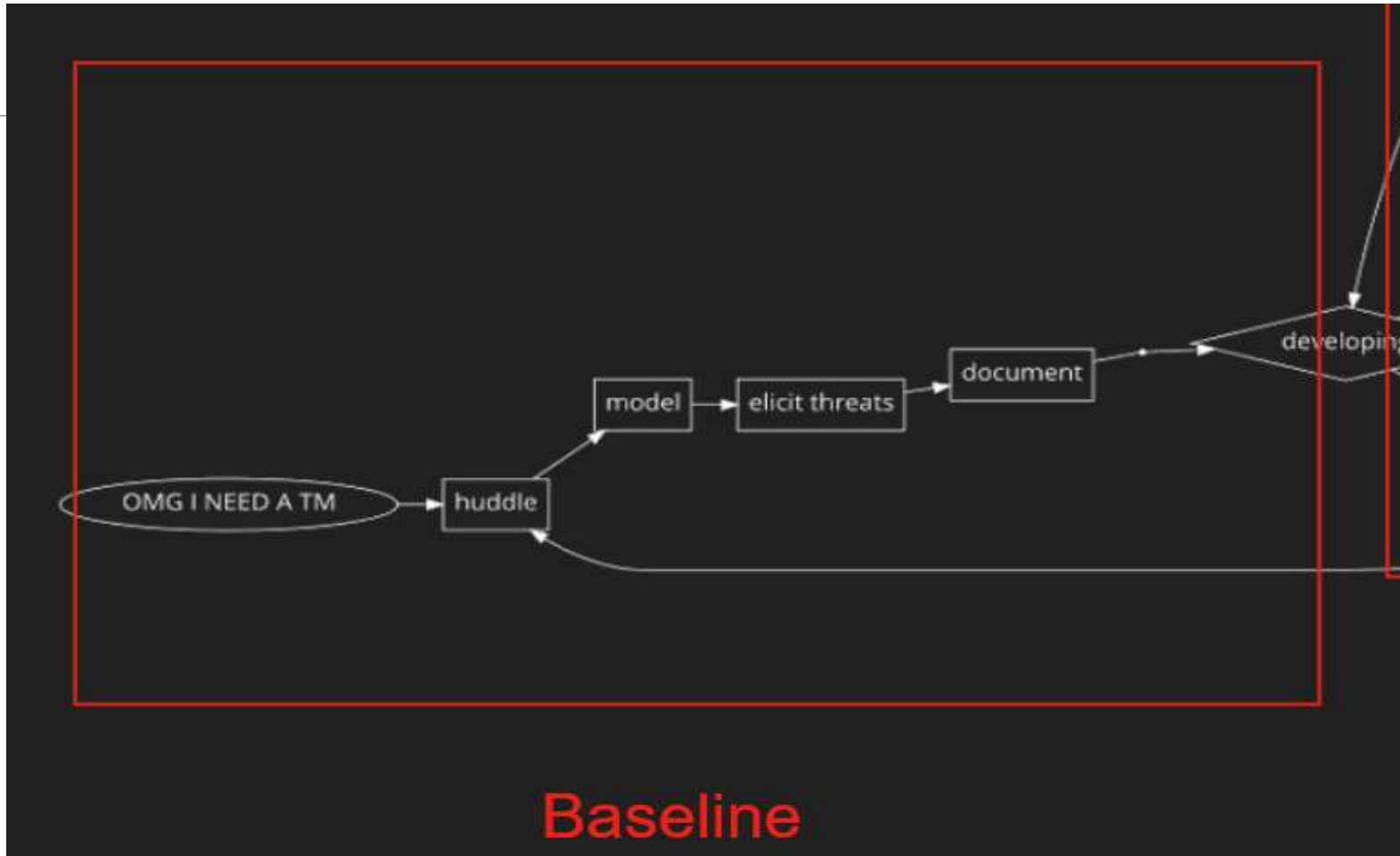
Threat Modeling Every Story - *Izar Tarandach, Matthew Coles*



*Threat Modeling: A Practical Guide for Development Teams by Izar Tarandach and Matthew J. Coles

Continuous Threat Modeling (CTM)*

Threat Modeling Every Story - *Izar Tarandach, Matthew Coles*



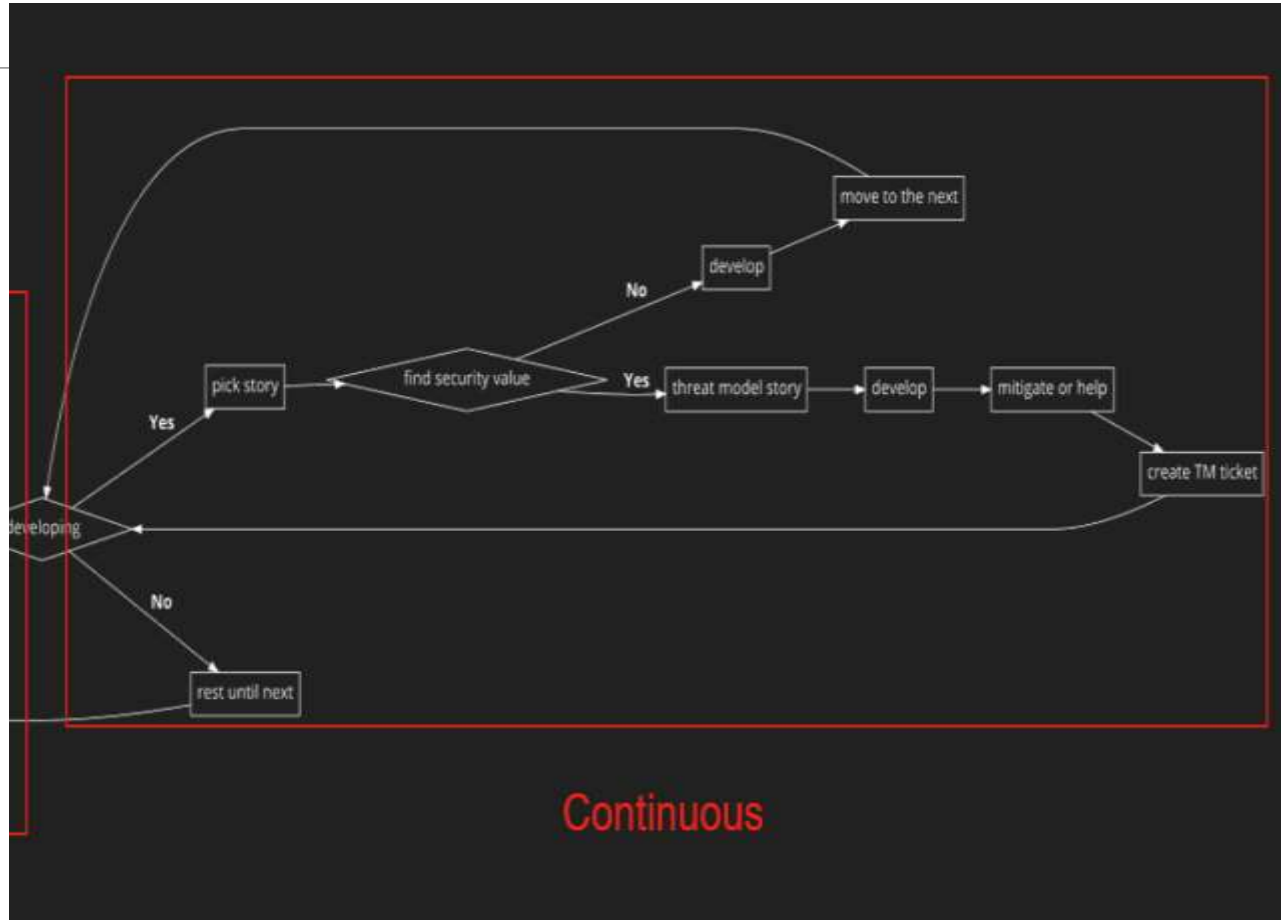
*Threat Modeling: A Practical Guide for Development Teams by *Izar Tarandach and Matthew J. Coles*

Continuous Threat Modeling (CTM)*

Threat Modeling Every Story - *Izar Tarandach, Matthew Coles*

As developing ...

1. Pick story
2. Find security value
 - a) Yes
 - i. Threat model story
 - ii. Develop
 - iii. Mitigate or help
 - iv. Create a TM ticket
 - b) No
 - i. Develop
 - ii. Move to next
3. Repeat



*Threat Modeling: A Practical Guide for Development Teams by *Izar Tarandach and Matthew J. Coles*

Threagile – Agile Threat Modeling (Christian Schneider)

Threagile enables teams to execute Agile Threat Modeling as seamlessly as possible, even highly integrated into DevSecOps environments.

Threagile is an open-source toolkit that allows you to model an architecture with its assets in an agile declarative fashion as a YAML file directly inside the IDE or any YAML editor.

<https://threagile.io/>

Use of Threat Libraries / Knowledge Bases

MITRE CAPEC - 500+ attack patterns

MITRE ATT&CK – threats / attack techniques + technical mitigation and detection approaches

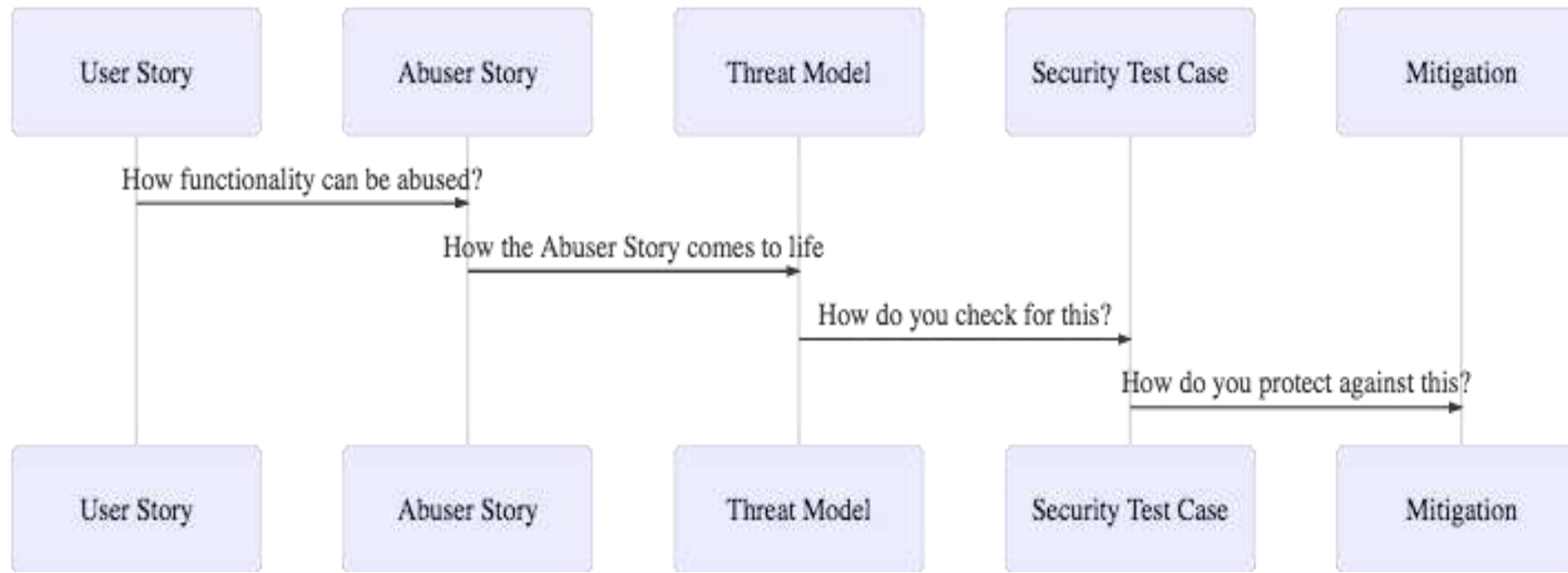
MITRE D3FEND – defensive cybersecurity countermeasures (complements ATT&CK)

Threat Modeling AS Code

What	Who	How
ThreatPlaybook	Abhay Bhargav @abhaybargav	Threat modeling FROM code
ThreatSpec	Fraser Scott @zeroXten	Threat modeling IN code
PyTM	Izar Tarandach @izar_t	Threat modeling WITH code

Threat Modeling FROM Code – ThreatPlaybook

Providing a way to combine User / Abuser stories, threat scenarios, and automated security testing.



Threat Modeling FROM Code – ThreatPlaybook



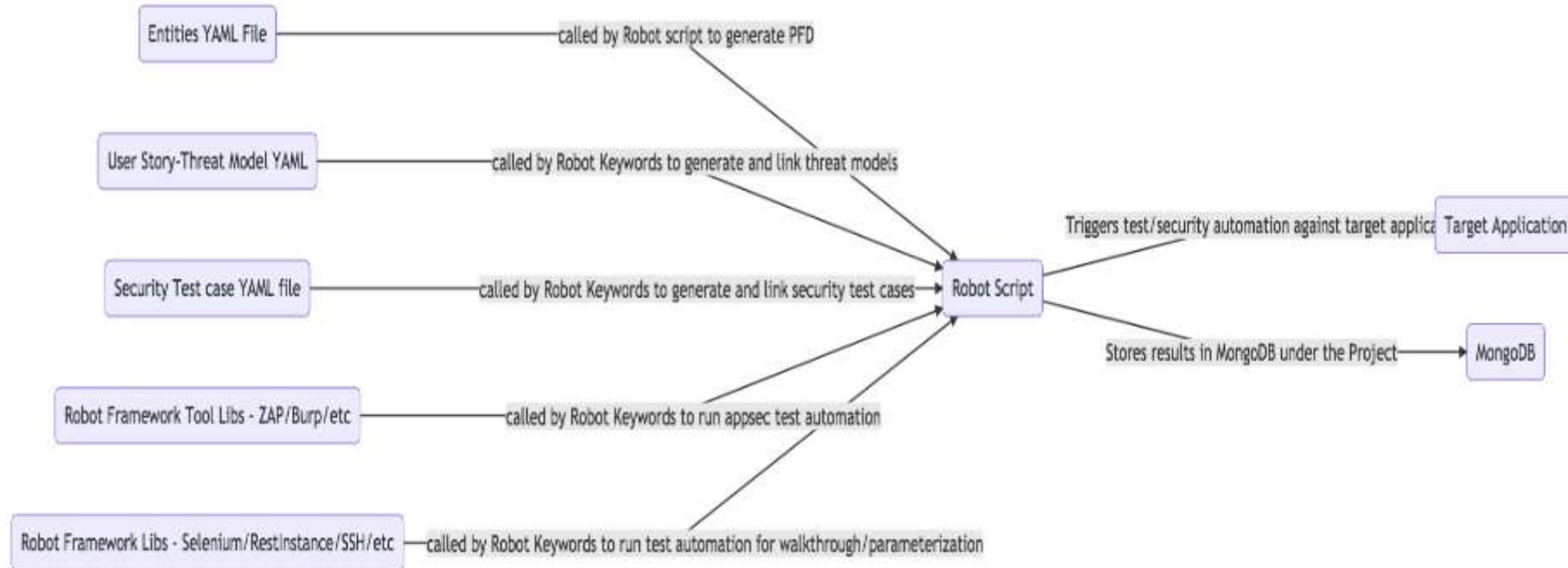
Sample YAML file:

```
1  create_customer_profile:
2    description: As an end-user, I would like to create customer profile and upload information to the customer profile. This will have the
3    abuse_cases:
4      render_api_unavailable:
5        description: As a malicious user, I would render the upload and API system unavailable to the organization
6        threat_scenarios:
7          malware_file_upload:
8            description: Upload file with malware that brings down the system or subjects it to ransomware
9            severity: 3
10           cwe: 434
11           cases:
12             - template_injection_auto
13             - nmap_vulnerability_scan
14             - xxe_auto
15             - malicious_file_upload
16       steal_customer_sensitive_files:
17         description: As a malicious user, I would like to steal customer PII from the uploaded files for me to be able to monetize this info
18         threat_scenarios:
```


Threat Modeling FROM Code – ThreatPlaybook



Processing YAML files through RobotFramework:



Threat Modeling IN Code – ThreatSpec



ThreatSpec [@ThreatSpec](#)

Fraser Scott [@zeroXten](#)

ThreatSpec - Have developers and security engineers write threat specifications alongside code, then dynamically generate reports and data-flow diagrams from the code

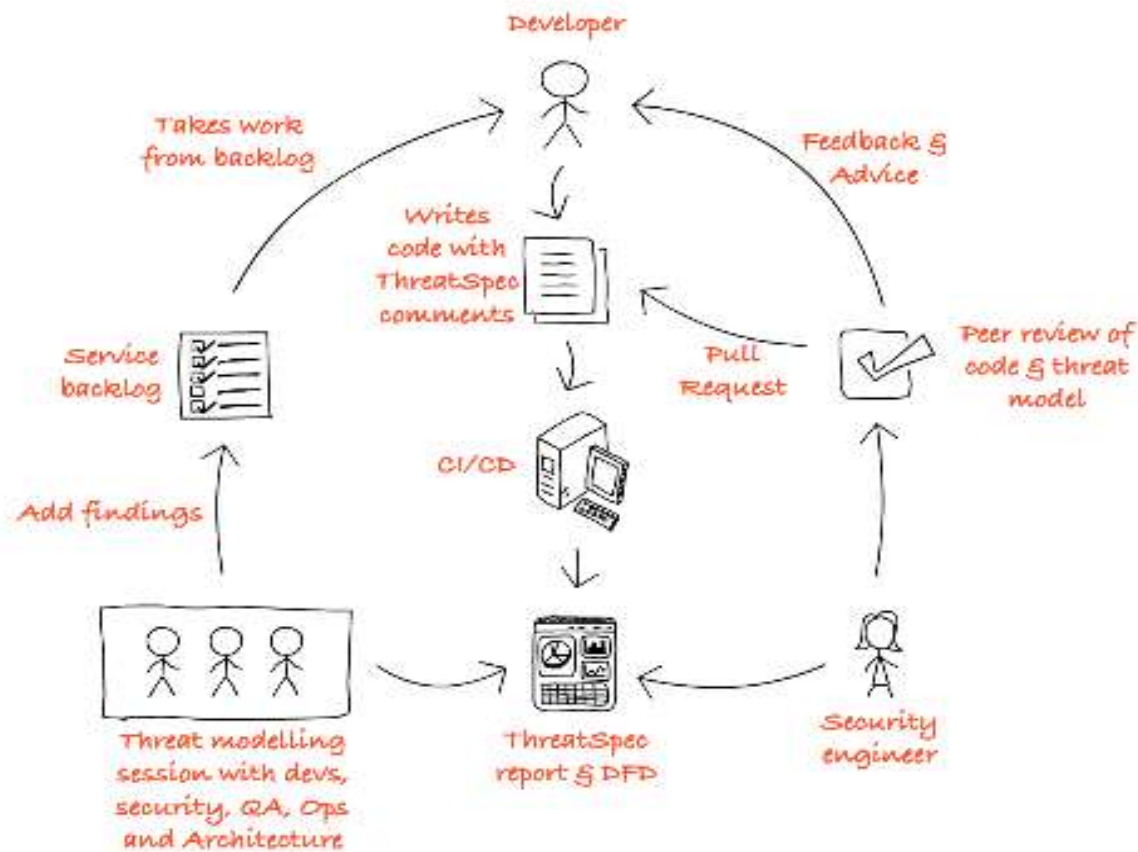
(Lots of examples for LAMP development.)

```
@threat SQL Injection as @sqli
@describe @sqli as Nefarious SQL statements are inserted into an entry field for
execution

@architecture MyApp as @myapp
@component Product Service as @product belongs to @myapp
@mitigates @product against SQL Injection with Parameterised queries
```

Threat Modeling IN Code – ThreatSpec

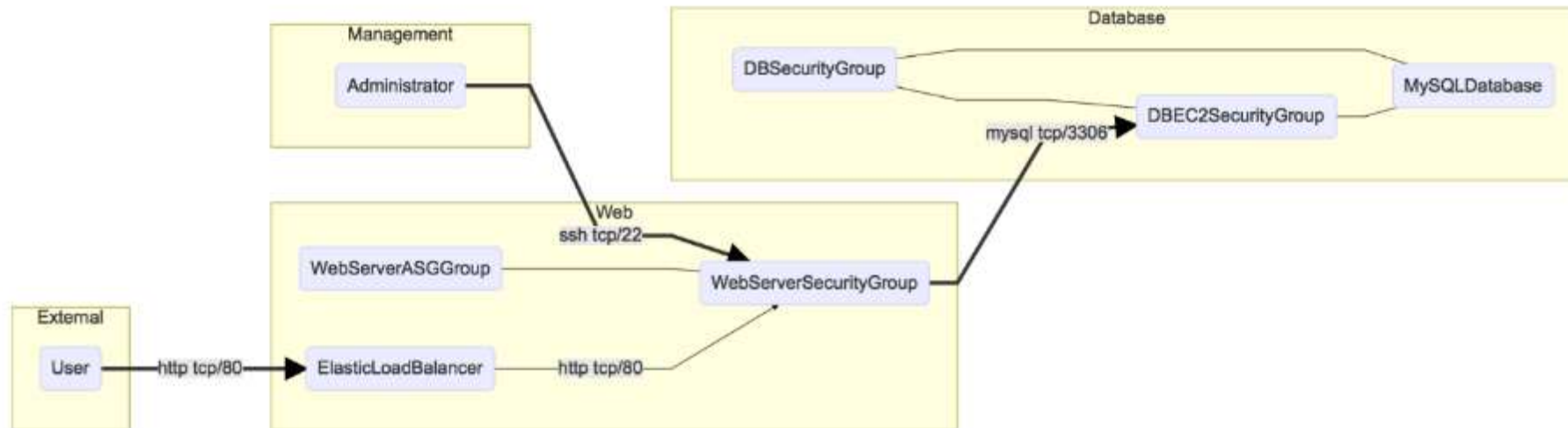
 threatspec



Threat Modeling IN Code – ThreatSpec



Generated DFD:



They would love feedback!

<https://threatspec.org/survey/>



Threat Modeling WITH Code – PyTM (2018)

Izar Tarandach [@izar_t](#)

Pythonic way of TM'ing – Creating a Threat Model

```
from pytm import TM, Server, Datastore, Dataflow,  
    Boundary, Actor, Lambda  
  
tm = TM("A barebones TM")  
  
tm.process()
```



Threat Modeling WITH Code – PyTM (2018)

Pythonic way of TM'ing – Elements and Attributes

```
User_Web = Boundary("User/Web")
Web_DB = Boundary("Web/DB")
VPC = Boundary("AWS VPC")

user = Actor("User")
user.inBoundary = User_Web

web = Server("Web Server")
web.OS = "CloudOS"
web.isHardened = True

my_lambda = Lambda("cleanDBevery6hours")
my_lambda.hasAccessControl = True
my_lambda.inBoundary = Web_DB
```

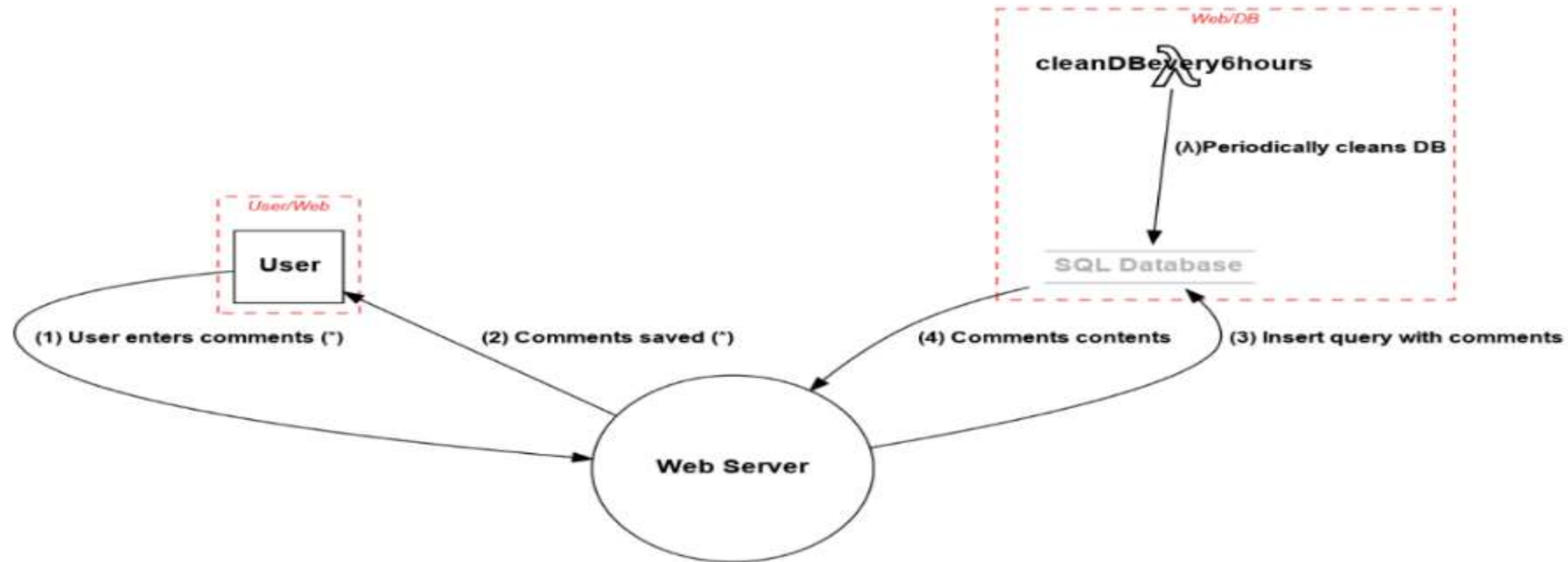


Threat Modeling WITH Code – PyTM (2018)

This input generates output to stdout, which is fed to Graphviz's dot:

```
tm.py --dfd | dot -Tpng -o sample.png
```

Generates this diagram:





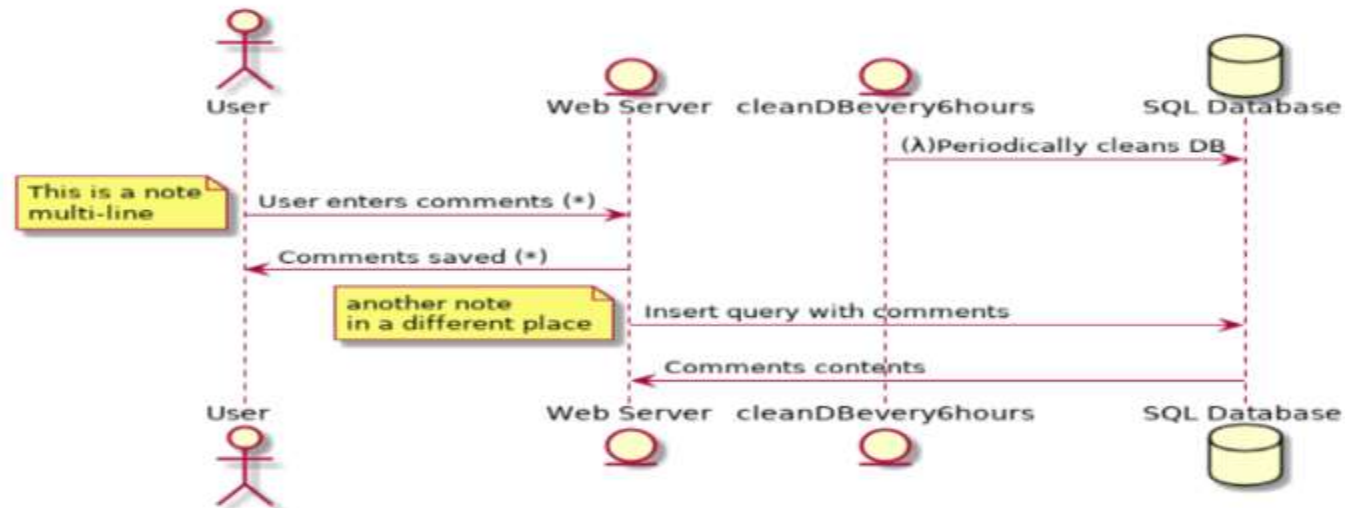
Threat Modeling WITH Code – PyTM (2018)

Dataflows can be ordered and sequence diagrams can be generated:

```
user_to_web = Dataflow(user, web, "User enters comments (*)")
user_to_web.protocol = "HTTP"
user_to_web.dstPort = 80
user_to_web.data = 'Comments in HTML or Markdown'
user_to_web.order = 1
```

```
tm.py --seq | java -Djava.awt.headless=true -jar ~/bin/plantuml.jar -tpng -pipe > seq.png
```

Generates this diagram:



Threat Modeling FROM Code:

Threagile (2020) – Agile Threat Modeling (Christian Schneider)

Threagile* enables teams to execute Agile Threat Modeling as seamlessly as possible, even highly integrated into DevSecOps environments.

Threagile is the open-source toolkit which allows to model an architecture with its assets in an agile declarative fashion as a YAML file directly inside the IDE or any YAML editor.

See: <https://threagile.io/>

* Threagile was introduced at **Black Hat Arsenal 2020** and **DEF CON 2020**

Threat Modeling FROM Code:

Threagile (2020) – Examples (from *threagile-example-model.yaml**)

```
abuse_cases:
  Denial-of-Service: >
    As a hacker I want to disturb the functionality of the backend system in order to cause indirect
    financial damage via unusable features.
  CPU-Cycle Theft: >
    As a hacker I want to steal CPU cycles in order to transform them into money via installed crypto currency miners.
  Ransomware: >
    As a hacker I want to encrypt the storage and file systems in order to demand ransom.
  Identity Theft: >
    As a hacker I want to steal identity data in order to reuse credentials and/or keys on other targets of the same company or outside.
  PII Theft: >
    As a hacker I want to steal PII (Personally Identifiable Information) data in order to blackmail the company and/or damage
    their repudiation by publishing them.

  ERP-System Compromise: >
    As a hacker I want to access the ERP-System in order to steal/modify sensitive business data.
  Database Compromise: >
    As a hacker I want to access the database backend of the ERP-System in order to steal/modify sensitive
    business data.
  Contract Filesystem Compromise: >
    As a hacker I want to access the filesystem storing the contract PDFs in order to steal/modify contract data.
  Cross-Site Scripting Attacks: >
    As a hacker I want to execute Cross-Site Scripting (XSS) and similar attacks in order to takeover victim sessions and
    cause reputational damage.
  Denial-of-Service of Enduser Functionality: >
    As a hacker I want to disturb the functionality of the enduser parts of the application in order to cause direct financial
    damage (lower sales).
  Denial-of-Service of ERP/DB Functionality: >
    As a hacker I want to disturb the functionality of the ERP system and/or it's database in order to cause indirect
    financial damage via unusable internal ERP features (not related to customer portal).

security_requirements:
  Input Validation: Strict input validation is required to reduce the overall attack surface.
  Securing Administrative Access: Administrative access must be secured with strong encryption and multi-factor authentication.
  EU-DSGVO: Mandatory EU-Datenschutzgrundverordnung
```

* From <https://run.threagile.io>

Threat Modeling FROM Code:

```
Customer Contract Database:
id: sql-database
#diagram_tweak_order: 0 # affects left to right positioning (only within a trust boundary)
description: The database behind the ERP system
type: datastore # values: external-entity, process, datastore
usage: business # values: business, devops
used_as_client_by_human: false
out_of_scope: false
justification_out_of_scope:
size: component # values: system, service, application, component
technology: database # values: see help
tags:
  - linux
  - mysql
internet: false
machine: virtual # values: physical, virtual, container, serverless
encryption: data-with-symmetric-shared-key # values: none, transparent, data-with-symmetric-shared-key, data-with-asymmetric-shared-key, data-with-enduser-individual-key
owner: Company ABC
confidentiality: strictly-confidential # values: public, internal, restricted, confidential, strictly-confidential
integrity: mission-critical # values: archive, operational, important, critical, mission-critical
availability: mission-critical # values: archive, operational, important, critical, mission-critical
justification_cia_rating: >
  The ERP system's database contains business-relevant sensitive data for the leasing processes and eventually also
  for other Company XYZ internal processes.
multi_tenant: false
redundant: false
custom_developed_parts: false
data_assets_processed: # sequence of IDs to reference
  - db-dumps
data_assets_stored: # sequence of IDs to reference
  - customer-accounts
  - customer-operational-data
  - internal-business-data
data_formats_accepted: # sequence of formats like: json, xml, serialization, file, csv
communication_links:
```

* From <https://run.threagile.io>



Threat Modeling FROM Code

~~hcltm~~ (2022) - renamed to **threatcl** (2024)

Documenting your threat models with HCL (Christian Frichot)

threatcl aims to provide a Git/DevOps-first approach to documenting a system threat model by focusing on the following goals:

- Simple text-file format
- Simple cli-driven user experience
- Integration into version control systems (VCS)

Side benefits:

- Ability to generate Data Flow Diagrams (DFDs)
- (New with threatcl update) Ability to export threat models to JSON or OTM (OpenThreatModel)

(NOTE: threatcl spec is based on HCL2, HashiCorp's Configuration Language, and with tools like Terraform)

Also, see AppSecPodcast episode on "Threat Modeling with hcltm": <https://www.youtube.com/watch?v=dfC-nMGL14k>

See: <https://threatcl.github.io/>

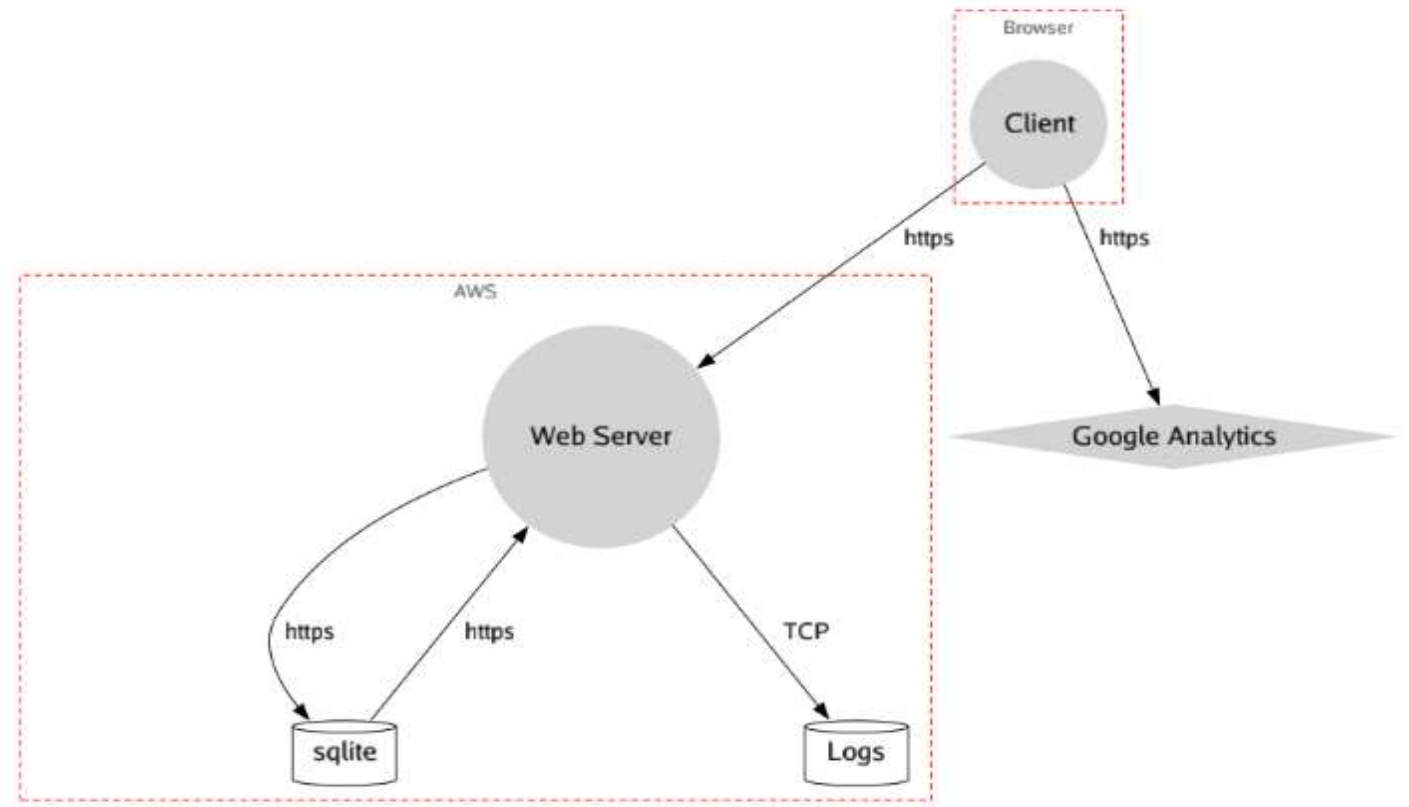


Threat Modeling FROM Code

threatcl (2024) – Examples (from <https://github.com/threatcl/>)

```
1 spec_version = "0.1.6"
2
3 threatmodel "Tower of London" {
4   description = "A historic castle"
5   author = "@xntrik"
6
7   attributes {
8     new_initiative = "true"
9     internet_facing = "true"
10    initiative_size = "Small"
11  }
12
13  information_asset "crown jewels" {
14    description = "including the imperial state crown"
15    information_classification = "Confidential"
16  }
17
18  usecase {
19    description = "The Queen can fetch the crown"
20  }
21
22  third_party_dependency "community watch" {
23    description = "The community watch helps guard the premise"
24    uptime_dependency = "degraded"
25  }
26
27  threat {
28    description = "Someone who isn't the Queen steals the crown"
29    impacts = ["Confidentiality"]
30    control = "Lots of guards"
31  }
32
33 }
```

Modelly model_Legacy DFD



Open Threat Model (OTM) Standard

Invented by Fraser Scott (ThreatSpec) – formerly at Capital One, now at IriusRisk

The Open Threat Model (OTM) standard is a generic and tool-agnostic way of describing a threat model in a simple-to-use and understand format

Read more here:

<https://www.iriusrisk.com/resources-blog/introduction-to-the-open-threat-model-standard>

Code: <https://github.com/iriusrisk/OpenThreatModel>

```
otmVersion: 0.1.0
project:
  id: helloworld
  name: Hello World
```

```
trustZones:
  - name: Public
    id: 6376d53e-6461-412b-8e04-7b3fe2b397de
    risk:
      trustRating: 10
  - name: Private Secured
    id: 2ab4effa-40b7-4cd2-ba81-8247d29a6f2d
    risk:
      trustRating: 90
```

```
components:
  - name: Client
    id: client
    type: generic-client
    parent:
      trustZone: 6376d53e-6461-412b-8e04-7b3fe2b397de
  - name: REST Service
    id: rest-service
    type: rest-full-web-service
    parent:
      trustZone: 2ab4effa-40b7-4cd2-ba81-8247d29a6f2d
```

```
dataflows:
  - name: Client to REST service
    id: client-to-rest
    source: client
    destination: rest-service
    tags:
      - HTTPS
```

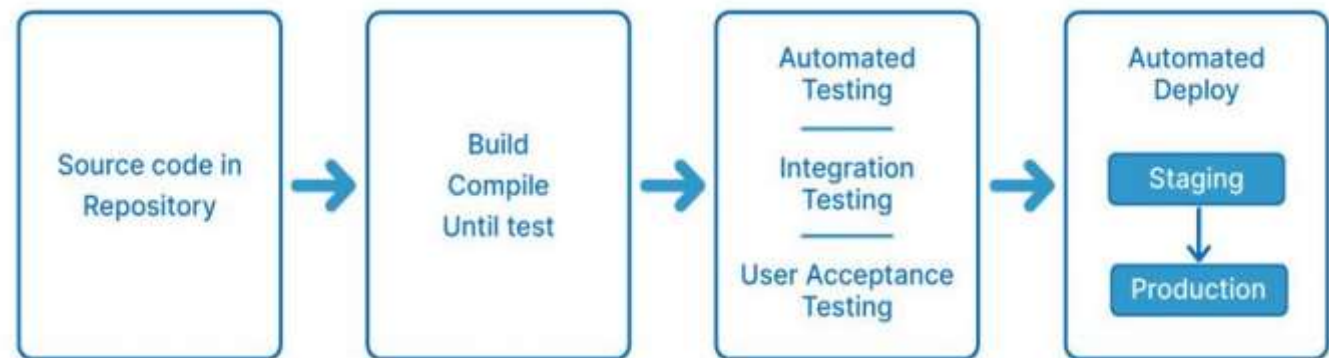


Threat Modeling your CI/CD Pipeline

Start with the Top 10 CI/CD security threats
(Cider Security*)

1. Insufficient Flow Control Mechanisms
2. Weak Identity and Access Management (IAM)
3. Dependency Chain Exploits
4. Poisoned Pipeline Execution (PPE)
5. ...

CI/CD PIPELINE



* <https://www.cidersecurity.io/top-10-cicd-security-risks/>

Threat Modeling and Generative AI

AI can help automate the mapping of components with potential threats and countermeasures, identify attack surfaces, rank threats, etc.

**Artificial
Intelligence**

AI can train on data:

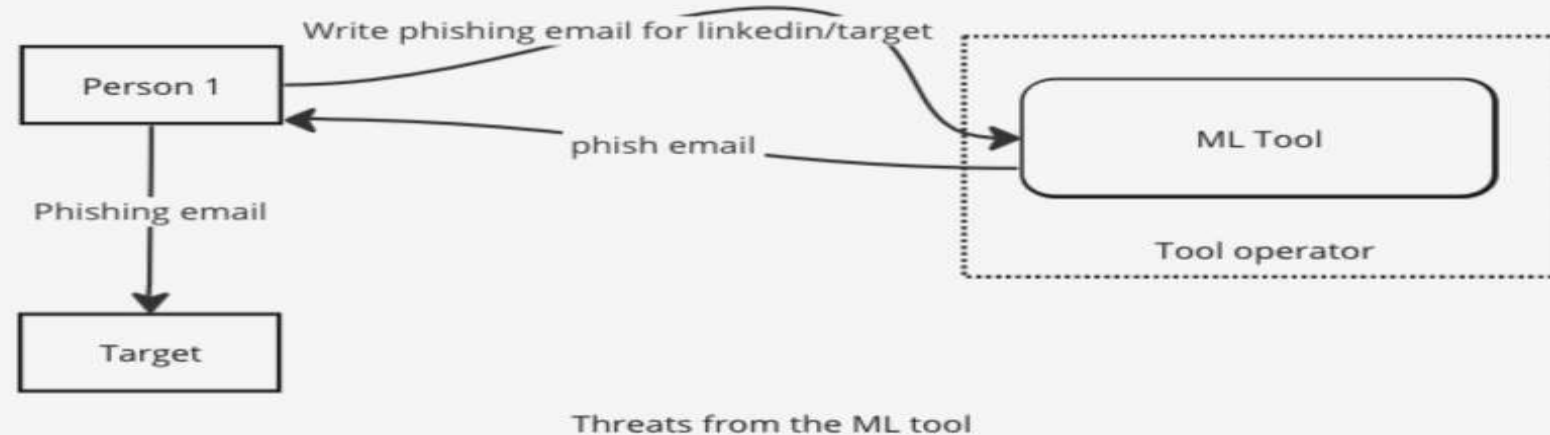
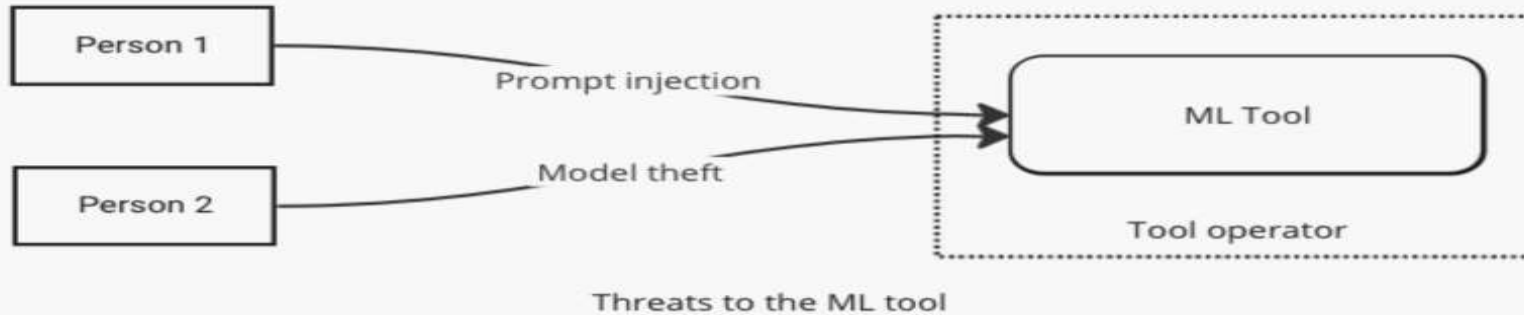
- Attack technique dataset
- Network log dataset
- Vulnerability dataset
- Threat intelligence dataset

AI can help with automation and efficiency, continuous real-time monitoring, adaptive defense, scalability, threat detection / Intelligence

However, still need human expertise to discern data quality, zero-day attacks

Many Threat Modeling tools are including AI / LLM features:
PyTM, IriusRisk, ThreatModeler, Devici, etc.

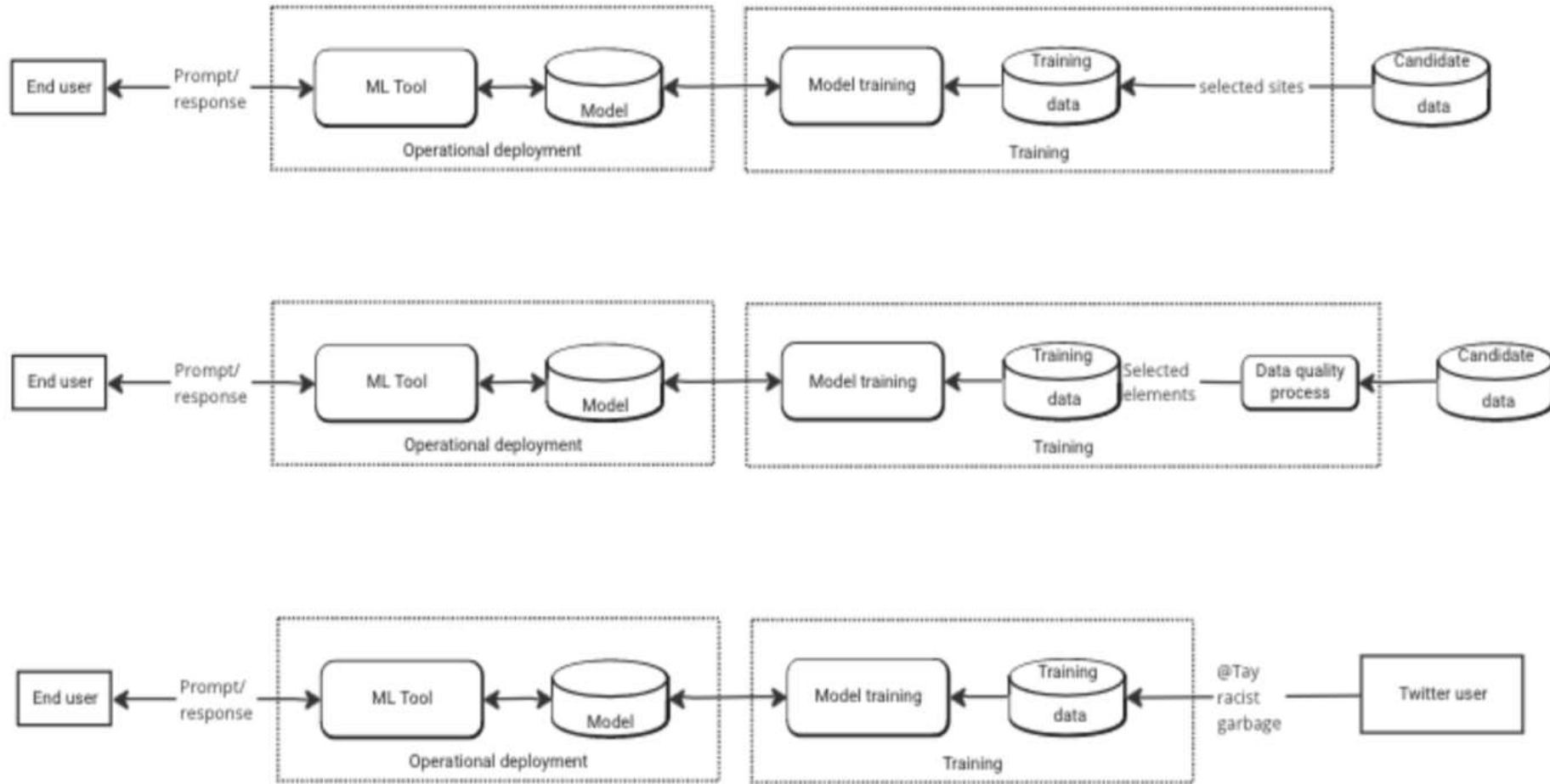
Threat model diagrams for ML



<https://shostack.org/blog/five-threat-model-diagrams-for-ml/>



Threat model diagrams for ML (continued)



<https://shostack.org/blog/five-threat-model-diagrams-for-ml/>



Does STRIDE still apply (to AI projects)?

There are still issues around AI/ML which benefit from analysis using STRIDE:

- Access (Authentication / Authorization) to Data and Models
- Integrity of Input Data / Training Data / Testing Data
- Tampering with Data and Models
- Sensitive information disclosed from Models



Threat Modeling Tools for AI Projects

Tools	Description
MITRE ATLAS	The ATLAS Matrix shows the progression of tactics used in attacks with ML techniques belonging to each tactic (this is an adaptation from MITRE ATT&CK).
Plot 4AI	PLOT4ai is a library (currently) containing 86 threats related to AI/ML. The library classifies threats into eight (8) different categories.
Berryville Institute of Machine Learning	The Berryville Institute of Machine Learning (BIML) lists general threats and risks against Machine Learning and Large Language Model systems.
OWASP AI Exchange	The AI Exchange provides comprehensive guidance and alignment on protecting AI against security threats - by professionals, for professionals.
Threat Modeling AI/ML Systems and Dependencies	(Microsoft Learn) Guidance on threat enumeration and mitigation specific to AI/ML security design review.



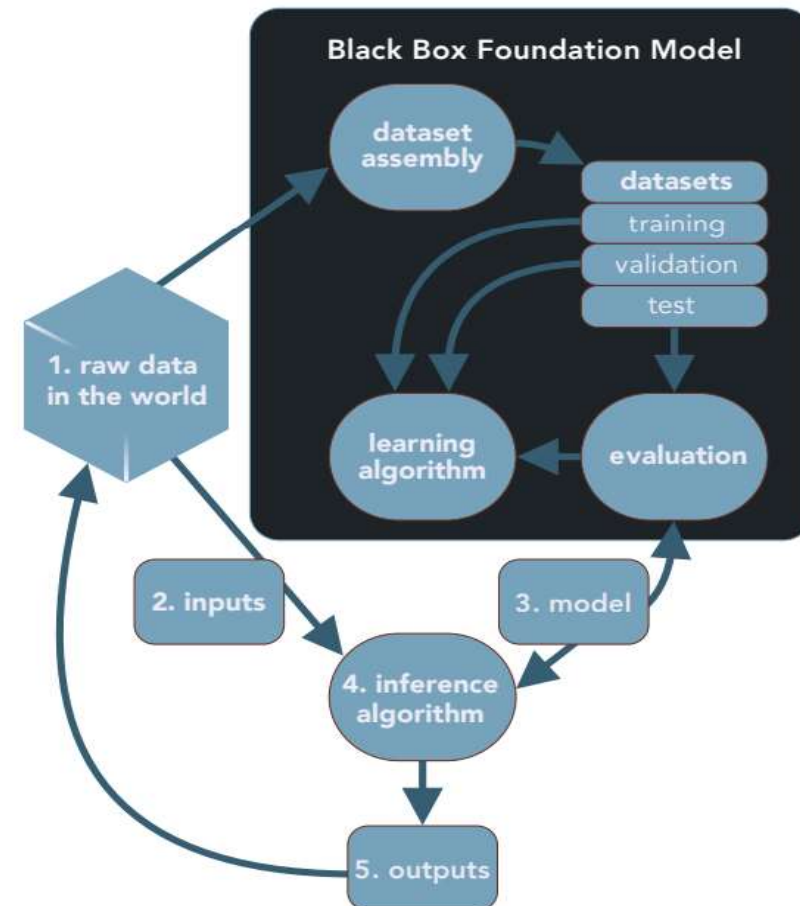
MITRE ATLAS Matrix

Reconnaissance ^{&}	Resource Development ^{&}	Initial Access ^{&}	ML Model Access	Execution ^{&}	Persistence ^{&}	Privilege Escalation ^{&}	Defense Evasion ^{&}	Credential Access ^{&}	Discovery ^{&}	Collection ^{&}	ML Attack Staging	Exfiltration ^{&}	Impact ^{&}
5 techniques	7 techniques	6 techniques	4 techniques	3 techniques	3 techniques	3 techniques	3 techniques	1 technique	4 techniques	3 techniques	4 techniques	4 techniques	6 techniques
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution ^{&}	Poison Training Data	LLM Prompt Injection	Evade ML Model	Unsecured Credentials ^{&}	Discover ML Model Ontology	ML Artifact Collection	Create Proxy ML Model	Exfiltration via ML Inference API	Evade ML Model
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities ^{&}	Valid Accounts ^{&}	ML-Enabled Product or Service	Command and Scripting Interpreter ^{&}	Backdoor ML Model	LLM Plugin Compromise	LLM Prompt Injection		Discover ML Model Family	Data from Information Repositories ^{&}	Backdoor ML Model	Exfiltration via Cyber Means	Denial of ML Service
Search Victim-Owned Websites	Develop Capabilities ^{&}	Evade ML Model	Physical Environment Access	LLM Plugin Compromise	LLM Prompt Injection	LLM Jailbreak	LLM Jailbreak		Discover ML Artifacts	Data from Local System ^{&}	Verify Attack	LLM Meta Prompt Extraction	Spamming ML System with Chaff Data
Search Application Repositories	Acquire Infrastructure	Exploit Public-Facing Application ^{&}	Full ML Model Access						LLM Meta Prompt Extraction		Craft Adversarial Data	LLM Data Leakage	Erode ML Model Integrity
Active Scanning ^{&}	Publish Poisoned Datasets	LLM Prompt Injection											Cost Harvesting
	Poison Training Data	Phishing ^{&}											External Harms
	Establish Accounts ^{&}												



Berryville List - Top 10 LLM Risks

1. Recursive pollution
2. Data debt
3. Improper use
4. Black box opacity
5. Prompt manipulation
6. Poison in the data
7. Reproducibility economics
8. Data Ownership
9. Model trustworthiness
10. Encoding integrity



Mitigating LLM threats

- Look into RAG (Retrieval Augmented Generation) (“data moat” approaches):
 - Choose the best LLM for specific needs
 - Protect data and IP through external, retrieve-only, authoritative data store
 - Always populate with the latest and greatest content to prevent stale data
 - Prevent hallucinations using factual data
 - Keep compliance and regulations in check through control of the data
 - Costs, though not insignificant, are less than trying to train an LLM on your own
- Other considerations include LLM Fine Tuning or a combination of RAG + LLM Fine Tuning



Mitigating AI threats in general

- Many threats/risks mentioned have recommended mitigations
- However, the reality is that some threats/risks don't have a full 100% mitigation
 - Prompt injection (for example)
- Ultimately, understand the threats/attacks and make risk acceptance decisions when introducing LLM into the enterprise



Agenda

Why, What, When, Who - Threat Modeling?

Threat Modeling: Getting Started

Threat Modeling Process

- Hands-on Exercises / Labs

Threat Modeling Tools

- Hands-on Exercises / Labs

Threat Modeling in an Agile/DevOps (and AI) World

What's next?

**Need CPEs?
This workshop
is worth 4
CPEs. Contact
me afterward
for a form.**



What next?

Learn more about:

- Privacy Threat Modeling
 - LINDDUN (<https://www.linddun.org/>)
- Attack Trees
 - Bruce Schneier's 1999 article
- Incremental Threat Modeling
 - Agile approaches – Irene Michlin ([@IreneMichlin](#))
- MITRE ATT&CK / D3FEND
 - <https://attack.mitre.org/>
 - <https://d3fend.mitre.org/>

Conclusion

Threat Modeling is too important not to do it

In an Agile / DevOps (and AI) world, we still need to think about Secure Design

Find ways to integrate Threat Modeling into Agile sprints and DevOps processes with Attacker / Abuser Stories, Quick Reviews, Automated Threat Analysis, etc.



Resources – Threat Modeling Manifesto

“Threat Modeling Manifesto” (2020)

<https://threatmodelingmanifesto.org/>

- Definition
- Values
- Principles
- Anti-Patterns



“Threat Modeling Capabilities”(2024)

<https://threatmodelingmanifesto.org/capabilities/>

- Strategy
- Education
- Creating Threat Models
- Acting on Threat Models
- Communications
- Measurement
- Program Management



Resources – Threat Modeling Connect

Threat Modeling Connect (started Fall, 2022)

<https://www.threatmodelingconnect.com/>

- Support and insights from other Threat Modelers
- Monthly Community Meetups
- Threat Modeling Hackathons
- Threat Modeling Open Forum
- Threat Modeling Conferences – Washington, D.C. (2023), Lisbon and San Francisco (2024)
- 2025 Conferences: Barcelona (May) and Washington, D.C. (Nov)
<https://threatmodcon.com>



Resources - Books

Threat Modeling as a Practice:

Threat Modeling: A Practical Guide for Development Teams (2020)

Izar Tarandach and Matthew Coles

Threat Modeling: Designing for Security (2014)

and

Threats: What Every Engineer Should Learn from Star Wars (2023)

Adam Shostack

Securing Systems: Applied Architecture and Threat Models (2015)

Brook S.E. Schoenfield

Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis (2015)

Marco Morana and Tony UcedaVelez

Threat Modeling Gameplay with EoP:
A reference manual for spotting
threats in software architecture
(2024)

Brett Crawley

Resources - Books

Applied Threat Modeling:

Hacking Kubernetes: Threat-Driven Analysis and Defense (2021)

Andrew Martin, Michael Hausenblas

Playbook for Threat Modeling Medical Devices (2021)

MITRE: <https://www.mitre.org/sites/default/files/2021-11/Playbook-for-Threat-Modeling-Medical-Devices.pdf>

Resources – Articles

“Integrating threat modeling with DevOps” (December, 2022)

<https://learn.microsoft.com/en-us/security/engineering/threat-modeling-with-dev-ops>



Resources - Tools

Microsoft Threat Modeling Tool

<https://aka.ms/threatmodelingtool>

ThreatModeler – Web-Based (in-house) Tool

<https://threatmodeler.com>

IriusRisk Software Risk Manager

<https://iriusrisk.com>

Devici Threat Modeling

<https://devici.com>

OWASP Threat Dragon

<https://owasp.org/www-project-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?

Slides:

<https://github.com/rhurlbut/CodeMash2025>



X (Twitter): [@RobertHurlbut](https://twitter.com/RobertHurlbut)

BlueSky: [roberthurlbut.bsky.social](https://bsky.social/roberthurlbut)

LinkedIn: [roberthurlbut](https://www.linkedin.com/in/roberthurlbut)

Discord: robert.ct (robertct)

Thank you!

