Advancing your Threat Modeling Approaches in an Agile/DevOps World

Robert Hurlbut

CodeMash 2025

Who am I?

X (*Twitter*): @RobertHurlbut
BlueSky: roberthurlbut.bsky.social
LinkedIn: 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

Threat Modeling

Introduction / Review

Threat Modeling + Agile / DevOps

User-Story / Agile / DevOps Approaches

What’s next?

Threat Modeling Introduction / Review

Threat Modeling – Getting lots of attention

"Threat Modeling Manifesto" (2020)

https://threatmodelingmanifesto.org/

- Definition
- Values
- Principles
- Anti-Patterns

OWASP®

A04:2021 – Insecure Design

added to OWASP Top 10 (**2021**)

https://owasp.org/www-project-top-ten/

Countermeasure: Threat Modeling!

NIST includes Threat Modeling in recommendation (2021)

NIST

Recommended Minimum Standard for Vendor or Developer Verification of Code (in response to EO 14028, Improving the Nation's Cyber Security)

Threat Modeling Connect (2022)

https://www.threatmodelingconnect.com/

- Support and insights from other TMers
- Monthly Community Meetups
- Threat Modeling Hackathons
- Threat Modeling Conferences

THREAT MODELING CONNECT

Threat Modeling – Getting lots of attention

“Shifting the Balance of Cybersecurity Risk:

Principles and Approaches to Security-by-Design and –Default”

(**April 13, 2023**) – Revised: (**October 25, 2023**)

Cybersecurity and Infrastructure Agency (CISA)

(NSA | FBI | ACSC | NCSC-UK | CCCS | BSI | NCSC-NL | CERT NZ | NCSC-NZ)

https://www.cisa.gov/sites/default/files/2023-10/SecureByDesign_1025_508c.pdf

> *"Use a tailored threat model during resource allocation and development to prioritize the most critical and high-impact features. Threat models consider a product’s use case, enabling development teams to fortify products. Finally, senior leadership should hold teams accountable for delivering secure products as a key element of product excellence and quality."*

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**"

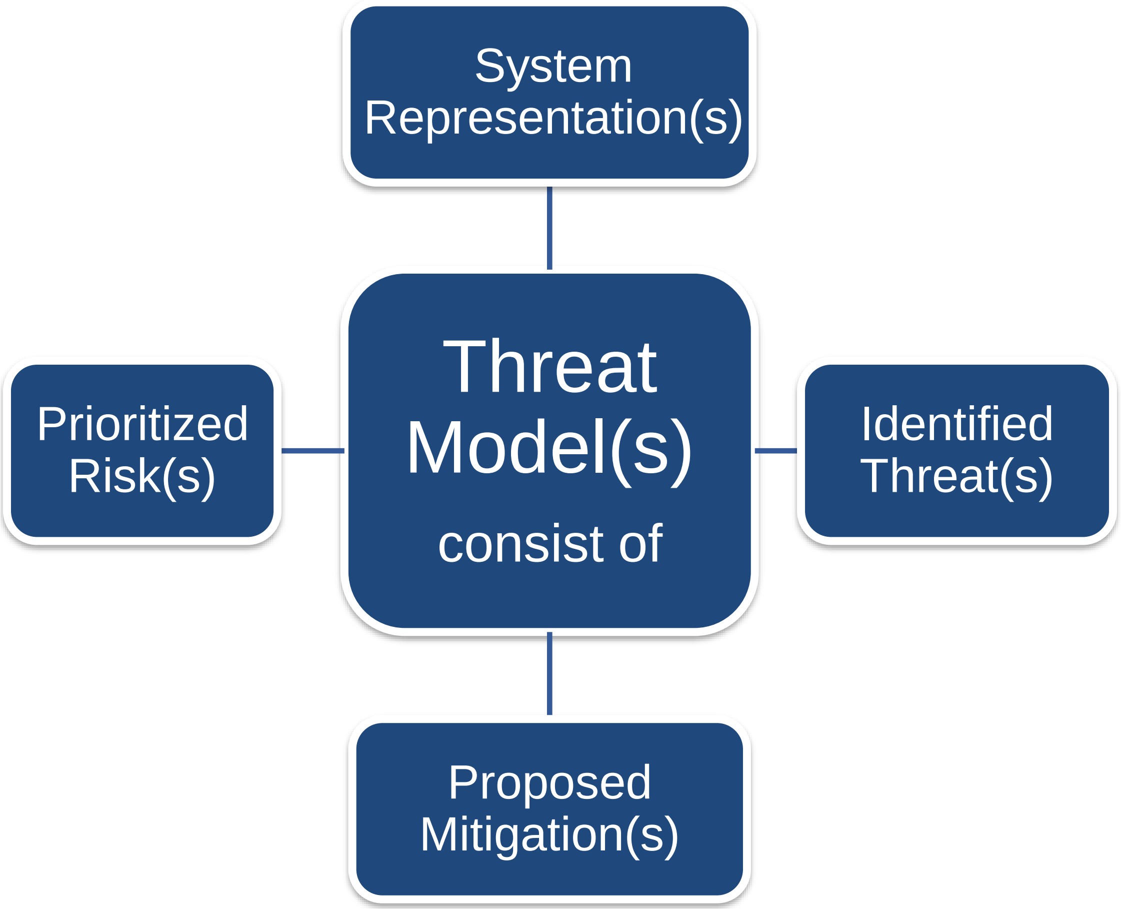
What is Threat Modeling? (cont.)

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.)



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

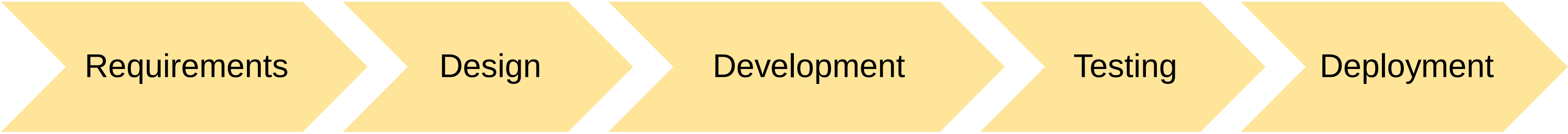
- Coding errors
- Requires developers to understand secure coding
- Can be automated
- Patching less costly in production

Security design flaws

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

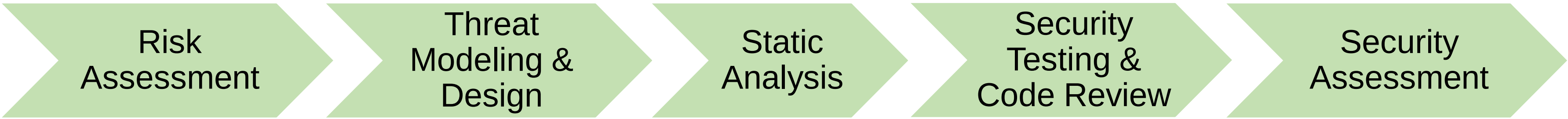
When do you do Threat Modeling? (continued)

SDLC* Process



Requirements → Design → Development → Testing → Deployment

Secure SDLC* Process



Risk Assessment → Threat Modeling & Design → Static Analysis → Security Testing & Code Review → Security Assessment

* SDLC = Software Development Life Cycle

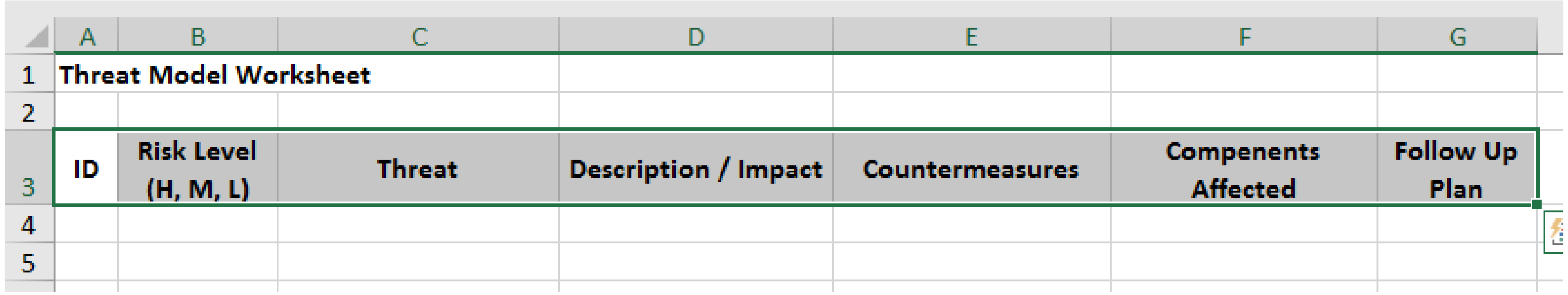
Threat Modeling: Getting Started

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							

Threat Modeling Process

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/

1a. What are we working on?

Gather Team

- Developers, Testers, Architects, Project Managers, Stakeholders, etc.

Domain Knowledge

Business / Technical Goals

Focused Sessions – *“What are we working on **now**?”*

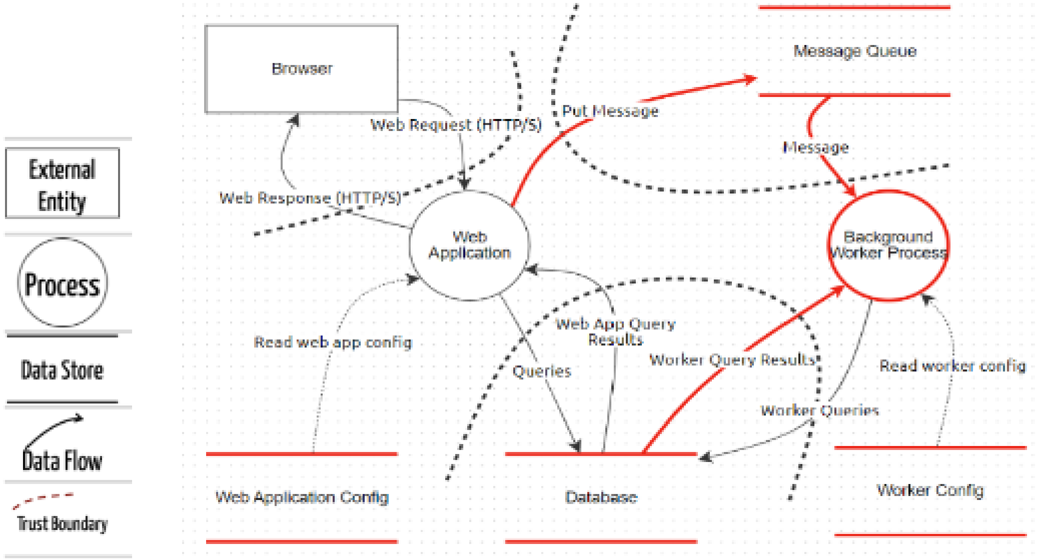
Important: Be honest, leave ego at the door, no blaming!

Be sure to document what you learn!

1b. What are we working on?

You can use an Architecture or Network diagram

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



(Sample DFD created with OWASP Threat Dragon 2.0)

2. What could go wrong? Looking at STRIDE to identify threats

Threat	Examples	Control we want
Spoofing	Pretending to be someone else	Identity Assurance
Tampering	Modifying data that should not be modifiable	Integrity
Repudiation (lack of proof)	Claiming someone didn’t do something	Non-repudiation (proof – Auditability)
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

2. What could go wrong? Identifying 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 or Attacker Cases

3. What are we going to do about it?

Mitigation Options:

Leave as-is

Remove from product

Remedy with technology countermeasure

Warn user

Make the mitigations/countermeasures part of your Security acceptance criteria.

Risk => Ease of exploitation x Business Impact

What is the risk associated with the identified threat <u>not</u> being mitigated?

4. Did we do a good enough job?

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

Threat Modeling Process
Tools

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
Tutamantic	Paid	API-based, Cloud
Devici (Beta – almost Prod)	Paid	Web-based, In-house or 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

Threat Modeling + Agile / DevOps

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.”* *

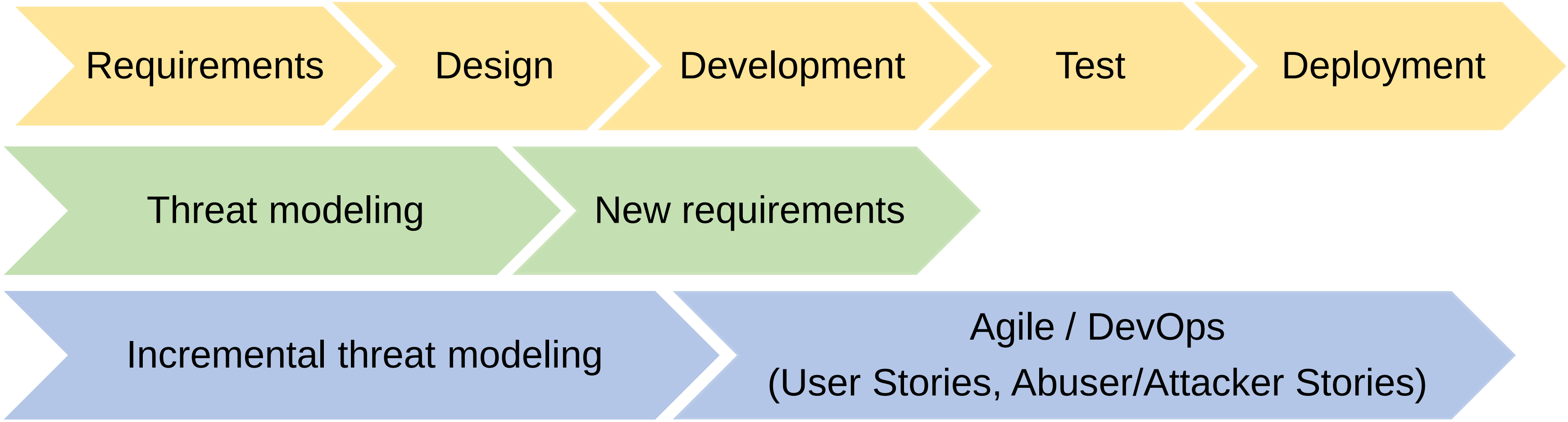
* (Quoted from an article by Ed Moyle: https://www.ecommercetimes.com/story/Invisible-Technologies-What-You-Cant-See-Can-Hurt-You-84852.html)

	Waterfall: Threat Model Documents	Agile: Bugs and conversations
System Model	• Big complex scope • System diagrams and essays • Gates, dependencies	• Scope tiny: this sprint's change • Big picture as security debt
Finding Threats	• Brainstorm • STRIDE • Kill Chain	• Same, aim at in-sprint code
Fixes	• Controls • Mitigations • Test Cases	• Spikes to understand • Security-focused stories in sprint, backlog, or epic • Security acceptance criteria
Quality	• Test plans	• 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):



Requirements

Design

Development

Test

Deployment

Threat modeling

New requirements

Incremental threat modeling

Agile / DevOps
(User Stories, Abuser/Attacker Stories)

* SDLC = Software
Development Life Cycle

Incremental Threat Modeling

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

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 (OK, but not great)

Security user stories are similar to regular user stories <u>but are sometimes more challenging to manage – *there may be too many*</u>.

Examples:

- As a user, I want to log into the application.
- As a user, I want 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 (Much better)

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

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

Value of DevOps in Threat Modeling

Not for Developers only

Operations can bring a great perspective

- Functional misuse
- Creative situations no one considered in design

Bring everyone to the table

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

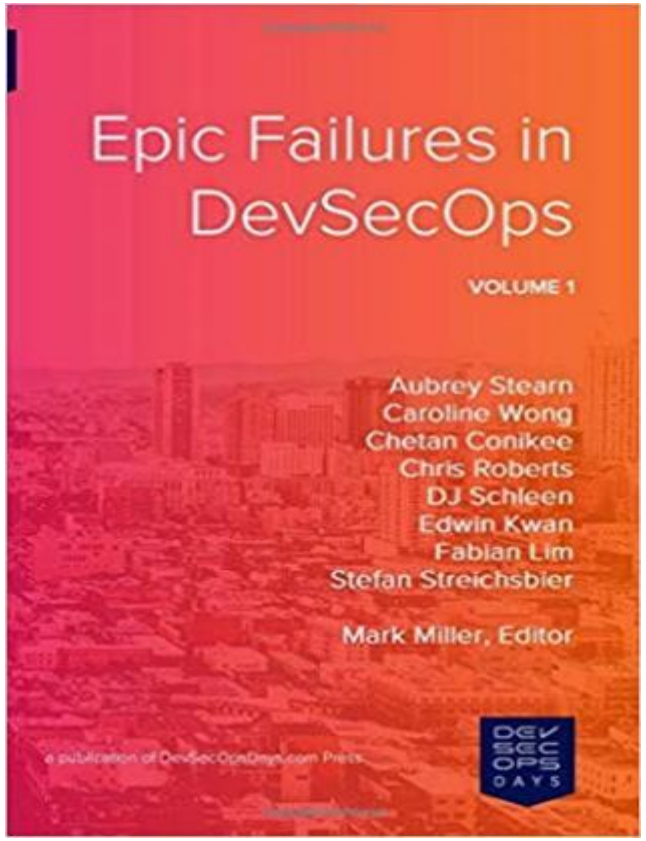
Goal: Threat Modeling for Devs

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***

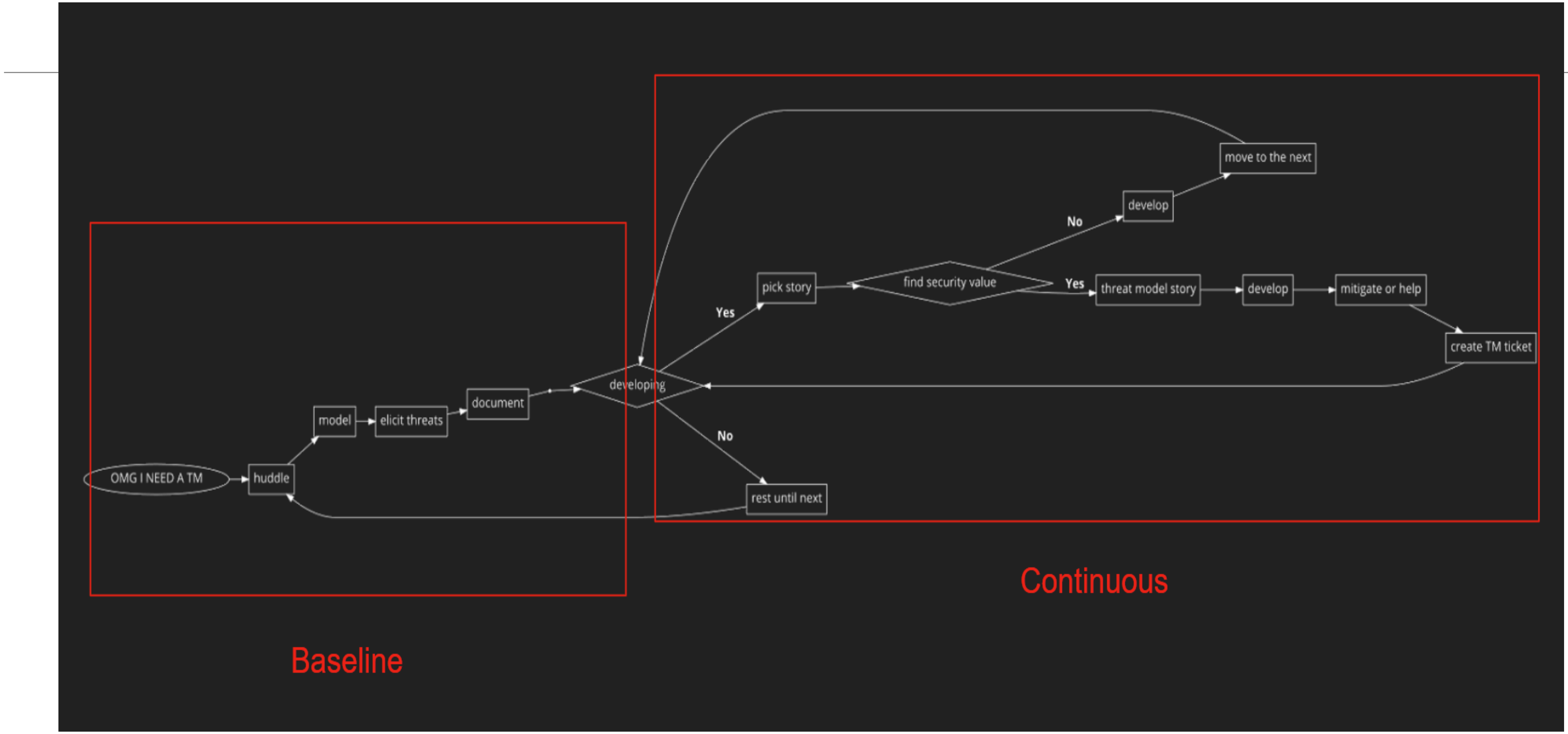
Focused on services and entry points:

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

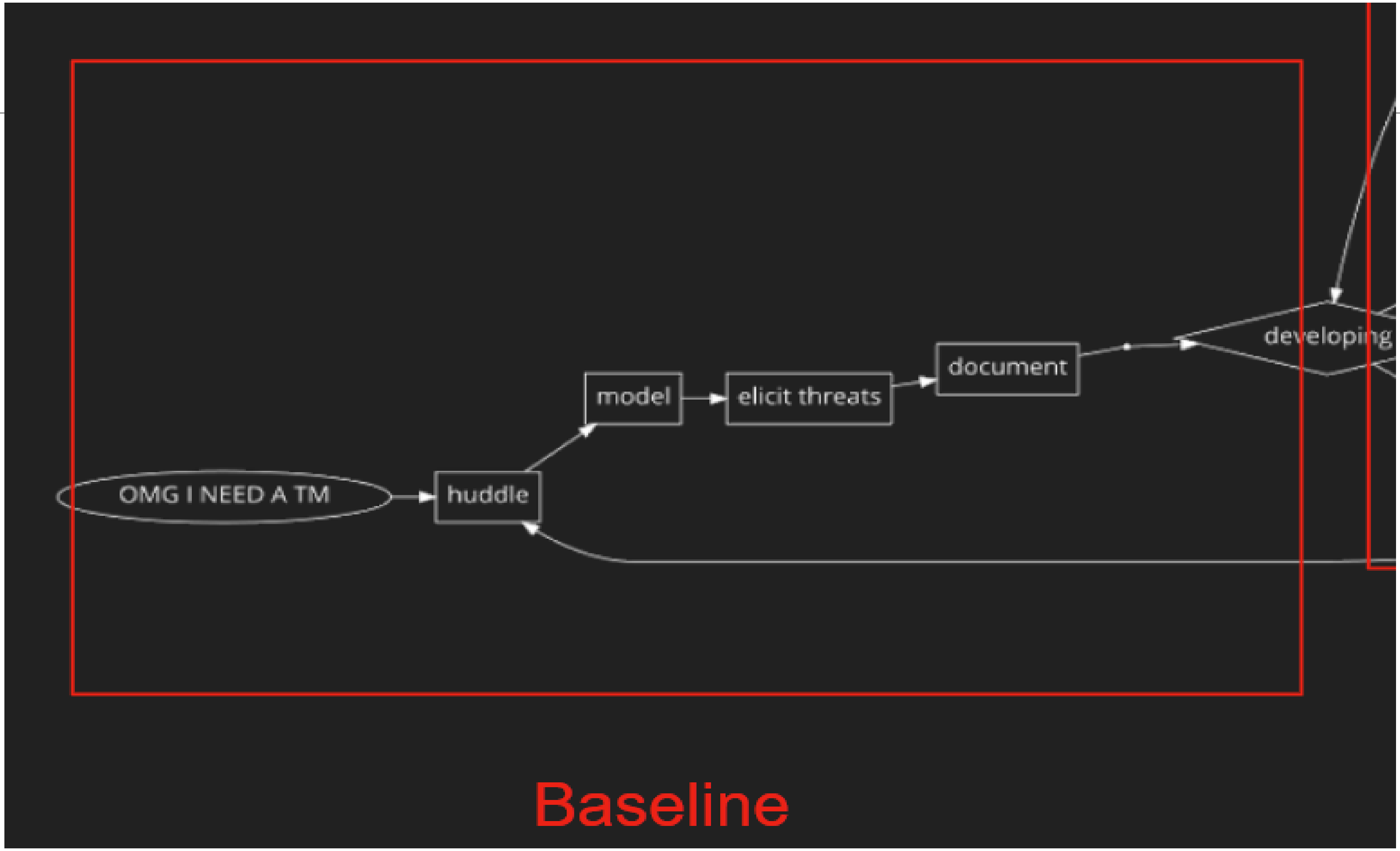
Blog post: https://home.edwinkwan.com/rapid-risk-assessments/

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*

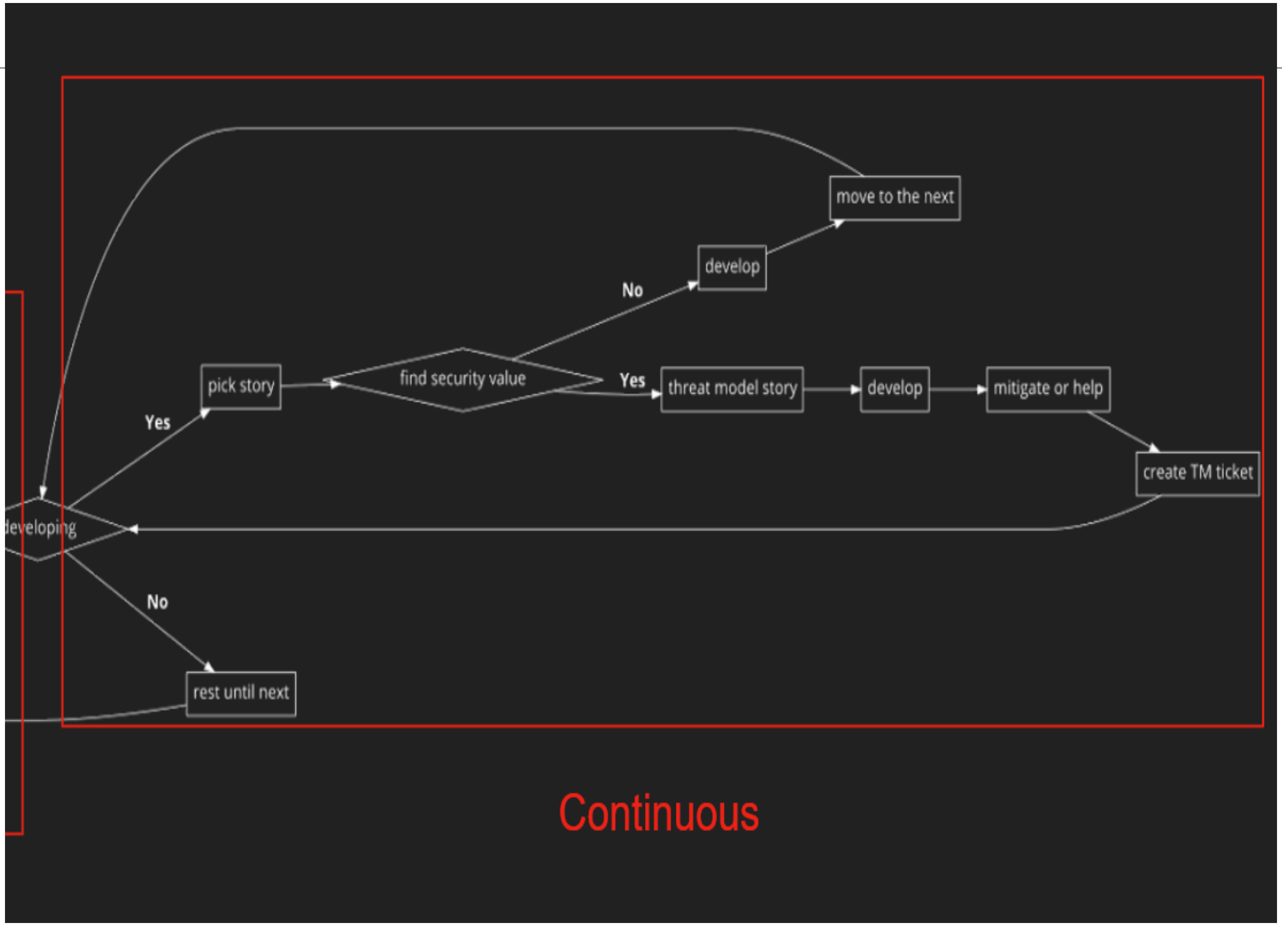


*Threat Modeling: A Practical Guide for Development Teams by *Izar Tarandach and Matthew J. Coles*

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*

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)

Identity threats - Games

Elevation of Privilege (EoP)

The EoP game focuses on the following threats (STRIDE):

- Spoofing
- Tampering
- Repudiation
- Information Disclosure
- Denial of Service
- Elevation of Privilege

EoP Mobile Deck
(Online play)

https://tabletopia.com/games/elevation-of-privilege-eop

Identity threats - Games

OWASP Cornucopia

Suits:

Data validation and encoding

Authentication

Session Management

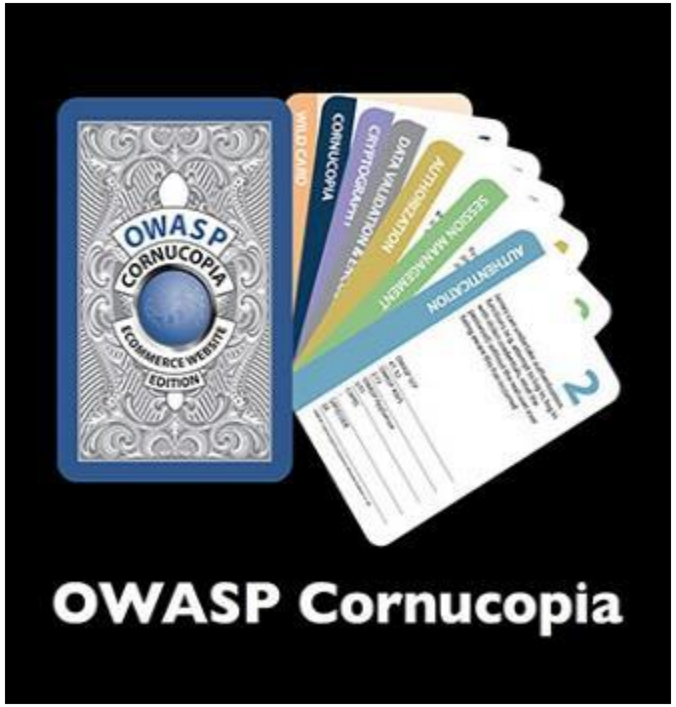
Authorization

Cryptography

Cornucopia

13 cards per suit, 2 Jokers

Play a round, highest value wins

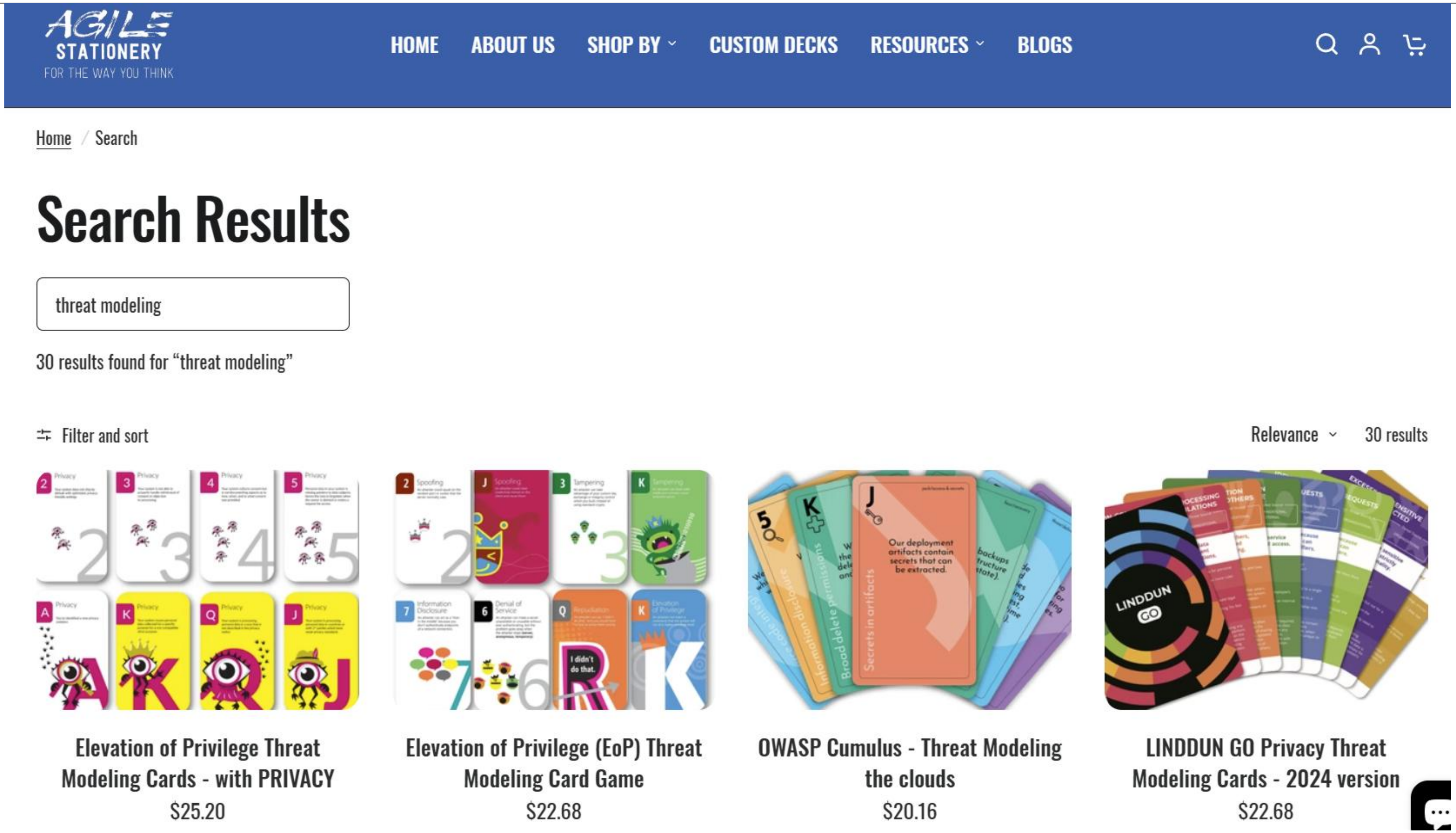


Cornucopia Mobile Deck
(Online play)

https://copi.securedelivery.io

Agile Stationary

Great Resource for Card Games – https://agilestationary.com



Home / Search

Search Results

threat modeling

30 results found for “threat modeling”

Elevation of Privilege Threat Modeling Cards - with PRIVACY
$25.20

Elevation of Privilege (EoP) Threat Modeling Card Game
$22.68

OWASP Cumulus - Threat Modeling the clouds
$20.16

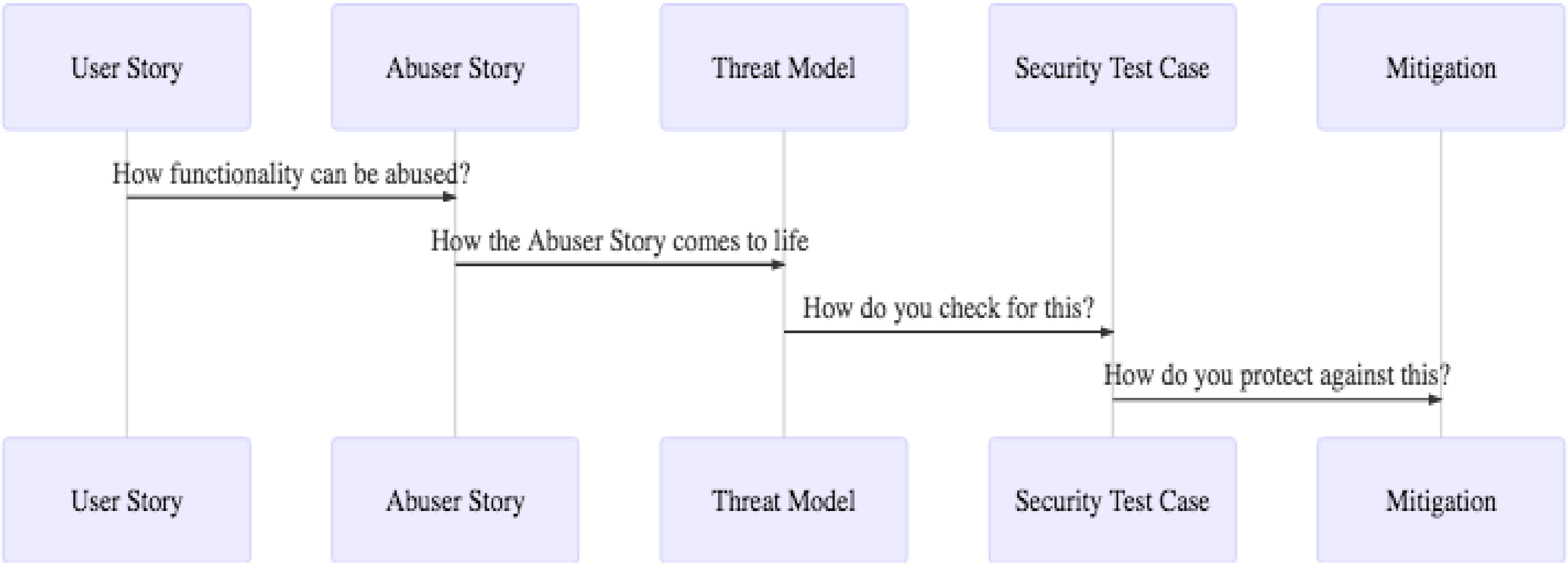
LINDDUN GO Privacy Threat Modeling Cards - 2024 version
$22.68

Threat Modeling AS Code

What	How
ThreatPlaybook, Threagile, hcltm	Threat modeling ***FROM*** code
ThreatSpec	Threat modeling ***IN*** code
PyTM	Threat modeling ***WITH*** code

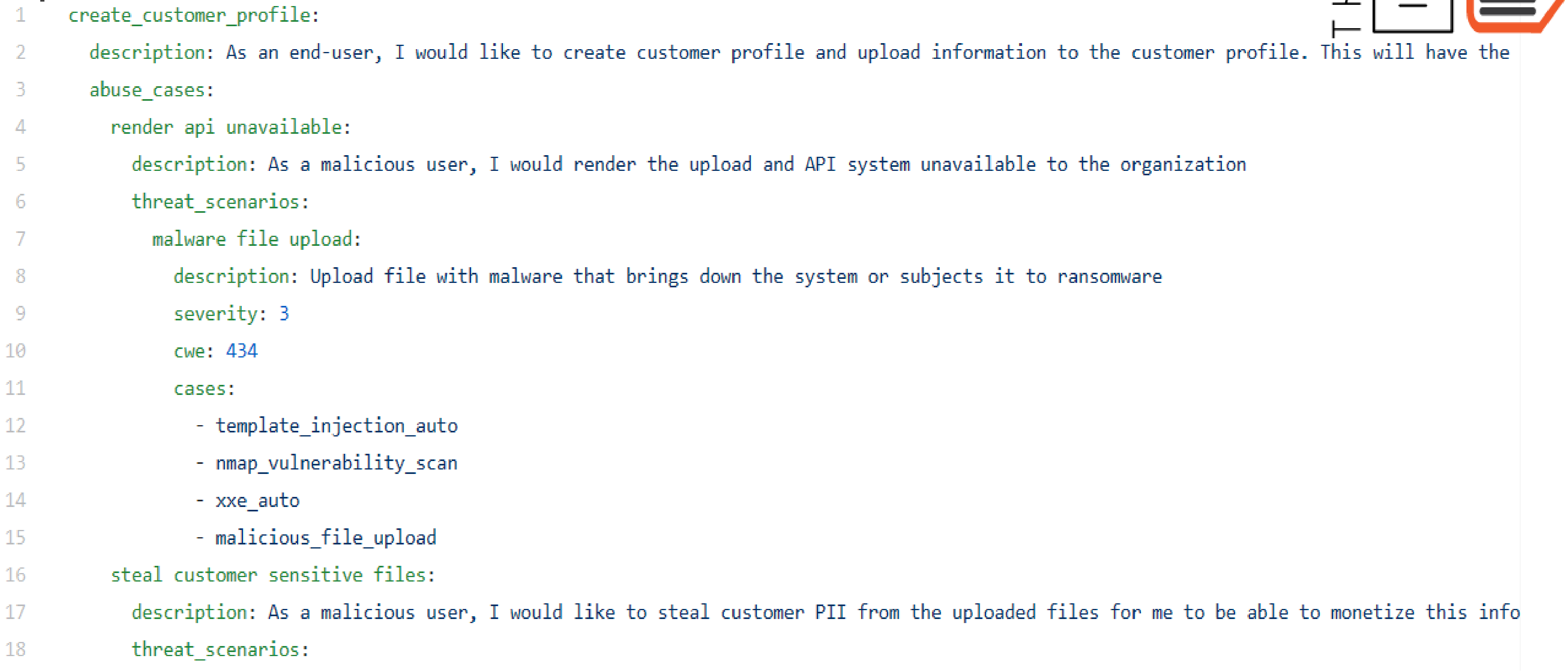
Threat Modeling FROM Code – ThreatPlaybook (2018)

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


User Story
Abuser Story
Threat Model
Security Test Case
Mitigation
How functionality can be abused?
How the Abuser Story comes to life
How do you check for this?
How do you protect against this?
User Story
Abuser Story
Threat Model
Security Test Case
Mitigation

Threat Modeling FROM Code – ThreatPlaybook (2018)

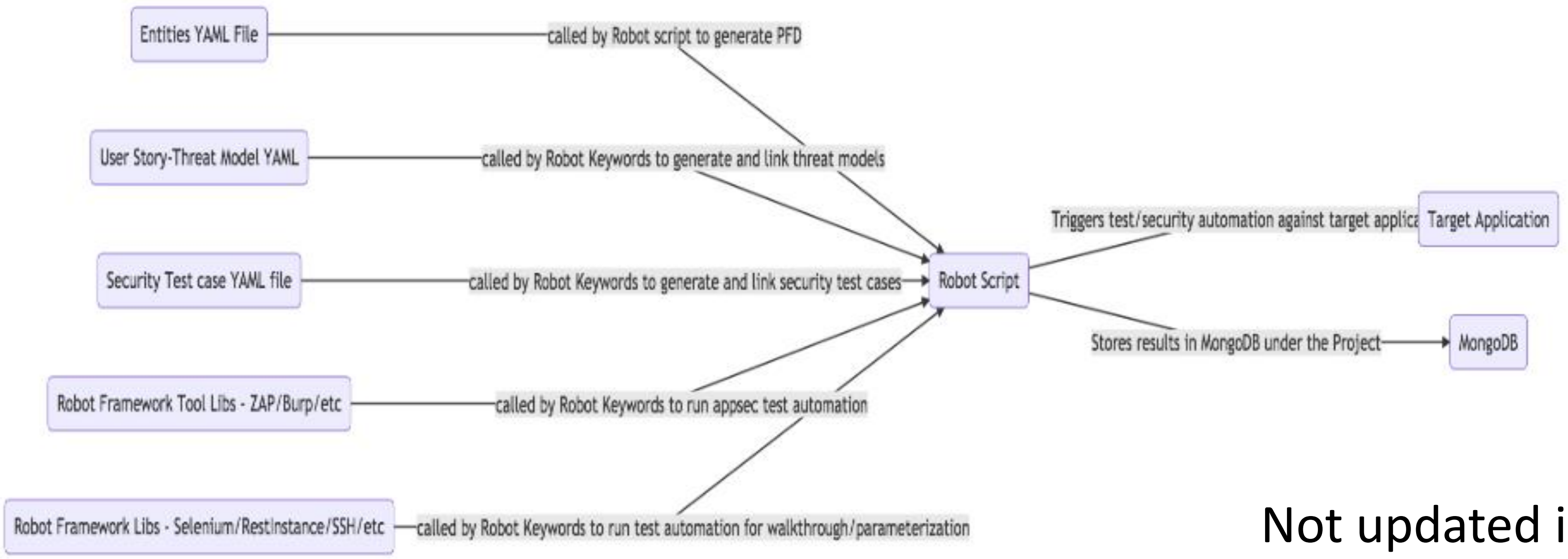
Sample YAML file:



```yaml
create_customer_profile:
  description: As an end-user, I would like to create customer profile and upload information to the customer profile. This will have the
  abuse_cases:
    render api unavailable:
      description: As a malicious user, I would render the upload and API system unavailable to the organization
      threat_scenarios:
        malware file upload:
          description: Upload file with malware that brings down the system or subjects it to ransomware
          severity: 3
          cwe: 434
          cases:
            - template_injection_auto
            - nmap_vulnerability_scan
            - xxe_auto
            - malicious_file_upload
    steal customer sensitive files:
      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
      threat_scenarios:
```

Threat Modeling FROM Code – ThreatPlaybook (2018)

Processing YAML files through RobotFramework:


Entities YAML File
called by Robot script to generate PFD
User Story-Threat Model YAML
called by Robot Keywords to generate and link threat models
Security Test case YAML file
called by Robot Keywords to generate and link security test cases
Robot Framework Tool Libs - ZAP/Burp/etc
called by Robot Keywords to run appsec test automation
Robot Framework Libs - Selenium/RestInstance/SSH/etc
called by Robot Keywords to run test automation for walkthrough/parameterization
Robot Script
Triggers test/security automation against target applica
Target Application
Stores results in MongoDB under the Project
MongoDB

Not updated in over 5 years …

Threat Modeling IN Code – ThreatSpec (2019)

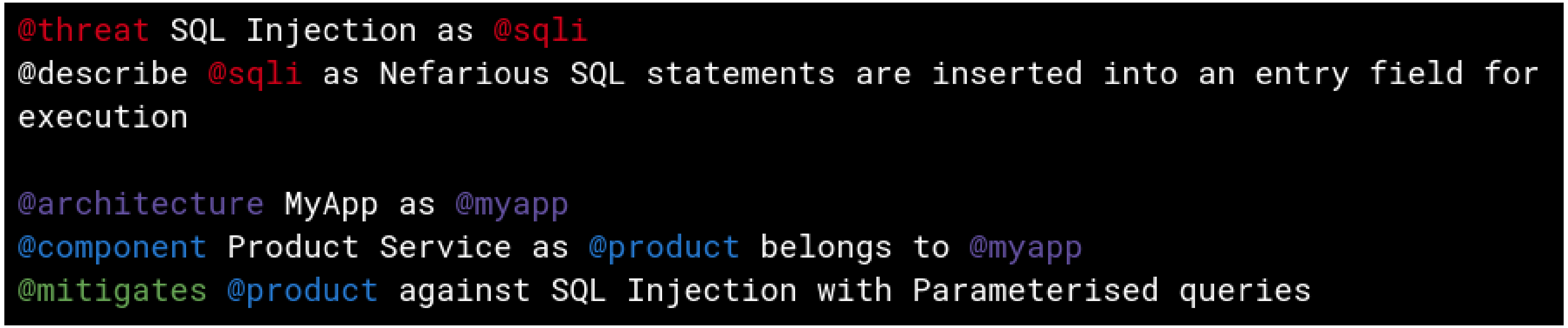
{♥} threatspec

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

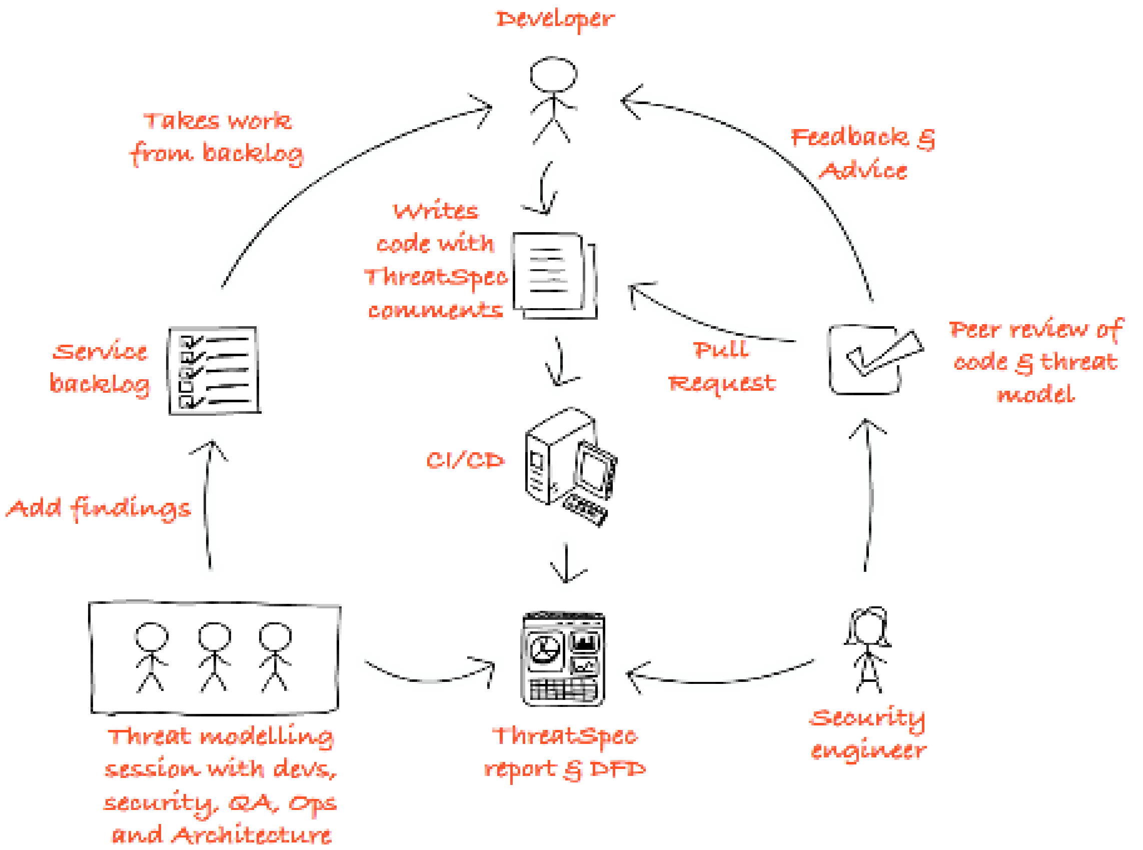
(Lots of examples for LAMP development.)



```
@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 (2019)

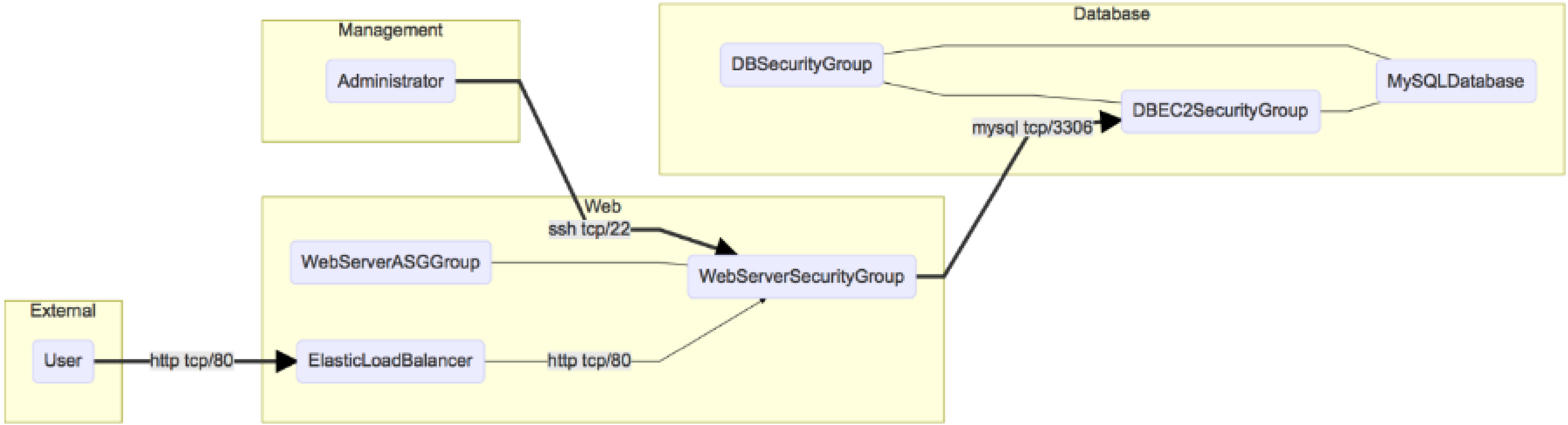
{♥} threatspec


Developer
Takes work from backlog
Feedback & Advice
Writes code with ThreatSpec comments
Service backlog
Pull Request
Peer review of code & threat model
CI/CD
Add findings
Threat modelling session with devs, security, QA, Ops and Architecture
ThreatSpec report & DFD
Security engineer

Threat Modeling IN Code – ThreatSpec (2019)

{♥} threatspec

Generated DFD:



Not updated in over 6 years …

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.

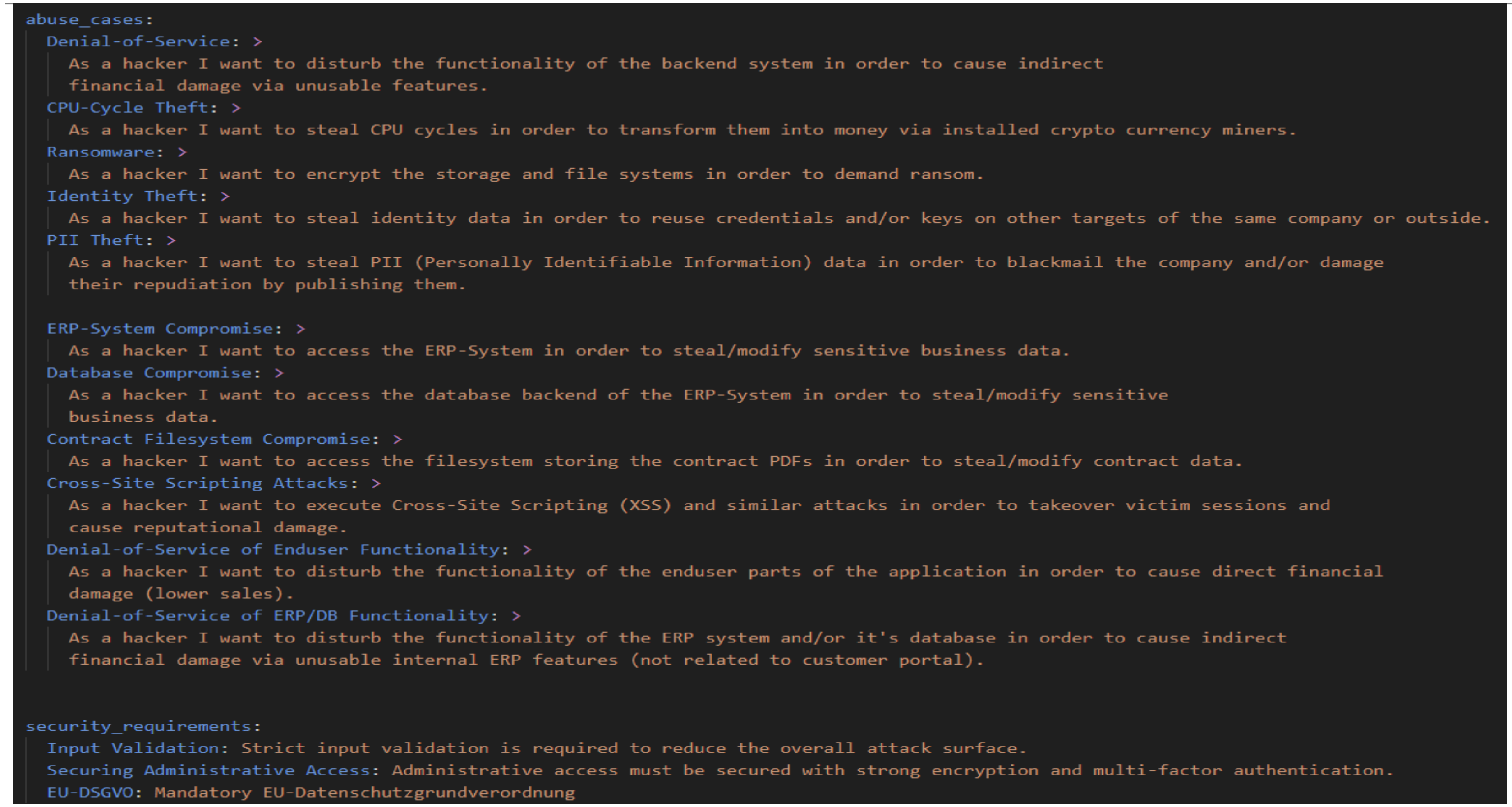
Threagile is an open-source toolkit that allows one to model architecture with its assets in an agile declarative fashion as a YAML file directly inside the IDE or any YAML editor.

Includes GitHub Workflow Integration and GitHub Actions.

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
```

Last update a year ago

* 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)

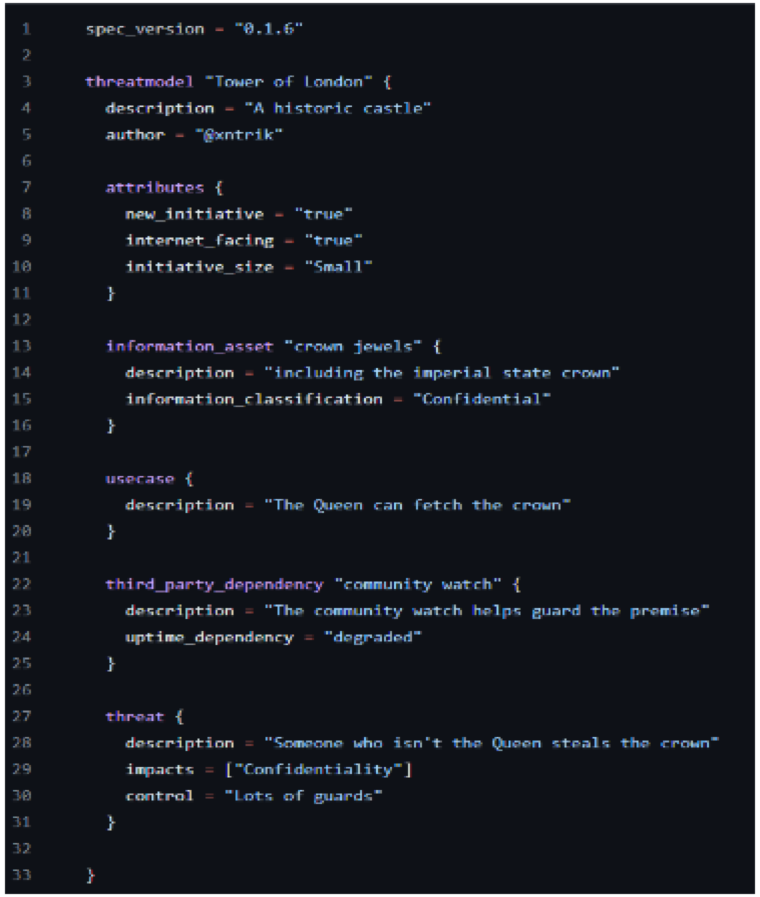
Last update – 5 months ago.
Actively updated.

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

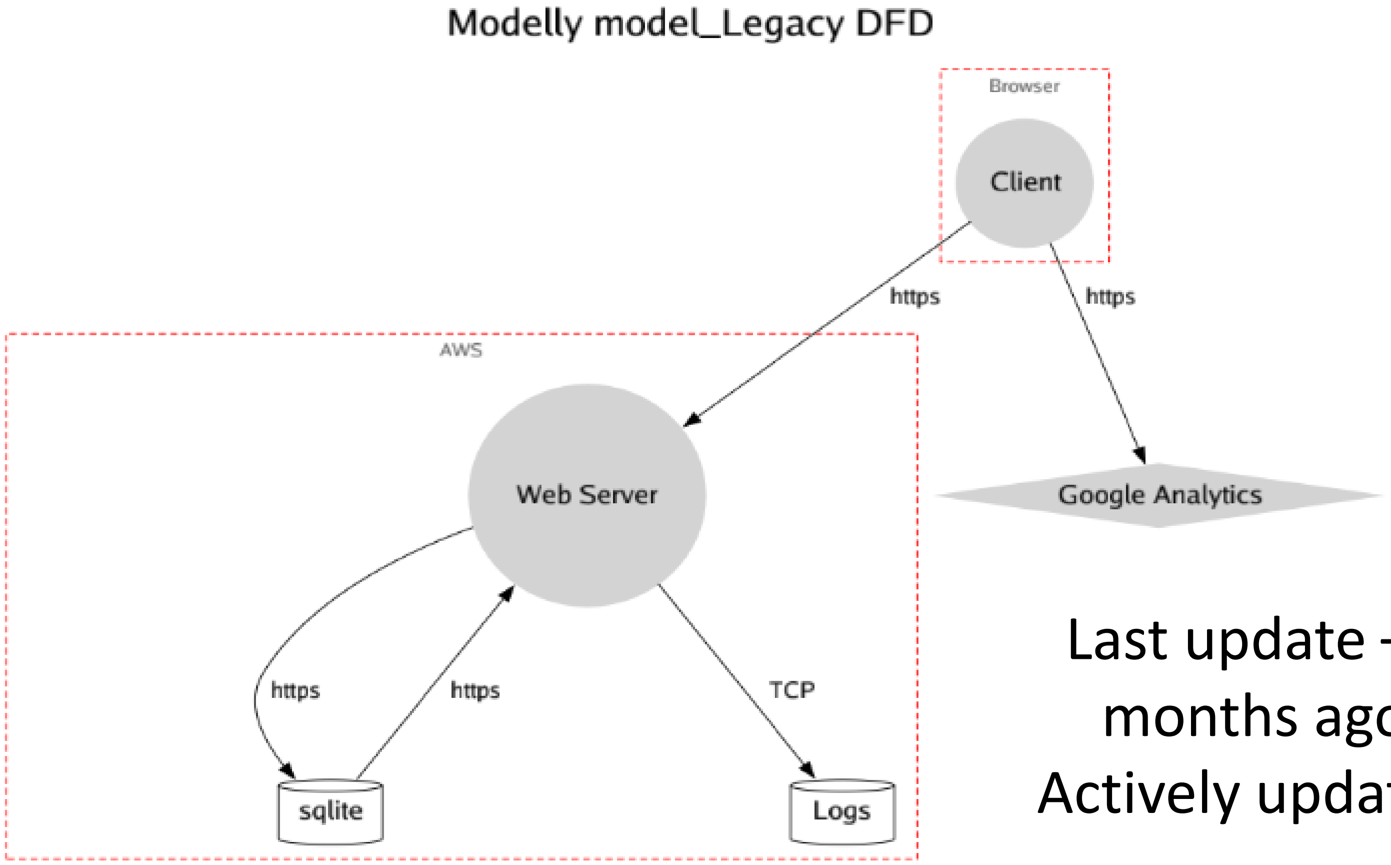
See: https://threatcl.github.io/

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

Threat Modeling FROM Code
threatcl (2024) – Examples (from https://github.com/threatcl/)



```
spec_version = "0.1.6"

threatmodel "Tower of London" {
  description = "A historic castle"
  author = "@xntrik"

  attributes {
    new_initiative = "true"
    internet_facing = "true"
    initiative_size = "Small"
  }

  information_asset "crown jewels" {
    description = "including the imperial state crown"
    information_classification = "Confidential"
  }

  usecase {
    description = "The Queen can fetch the crown"
  }

  third_party_dependency "community watch" {
    description = "The community watch helps guard the premise"
    uptime_dependency = "degraded"
  }

  threat {
    description = "Someone who isn't the Queen steals the crown"
    impacts = ["Confidentiality"]
    control = "Lots of guards"
  }

}
```

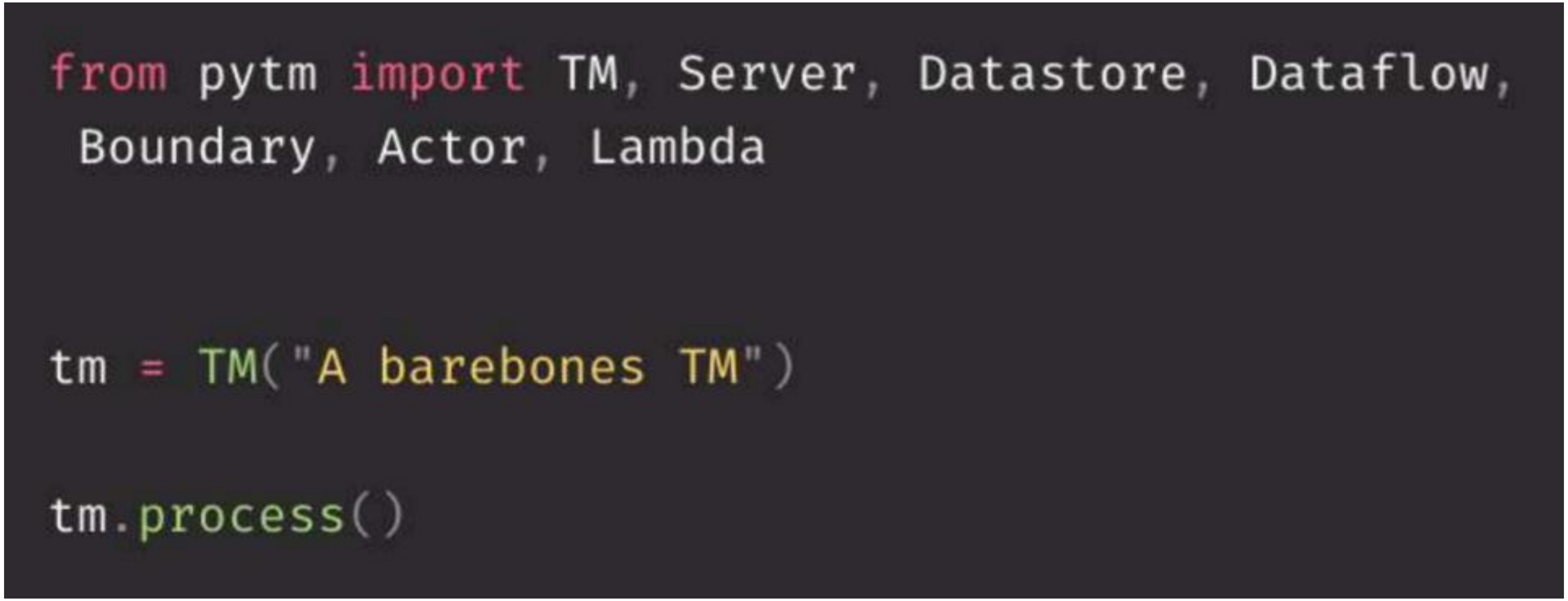


Last update – 5 months ago. Actively updated.

Threat Modeling WITH Code – **PyTM** (2018)

Izar Tarandach @izar_t

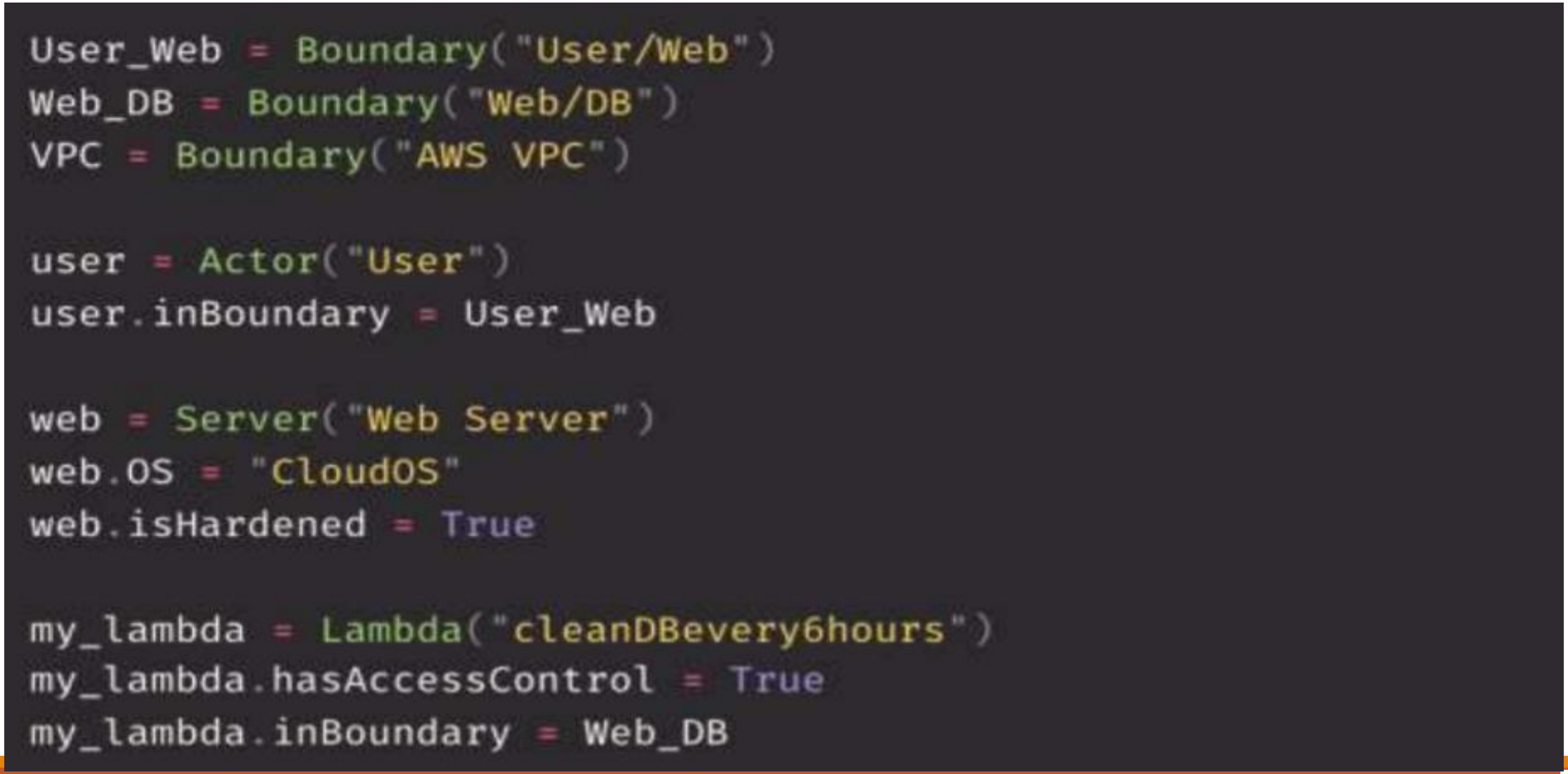
Pythonic way of TM'ing – Creating a Threat Model



```
from pytm import TM, Server, Datastore, Dataflow,
 Boundary, Actor, Lambda


tm = TM("A barebones TM")

tm.process()
```

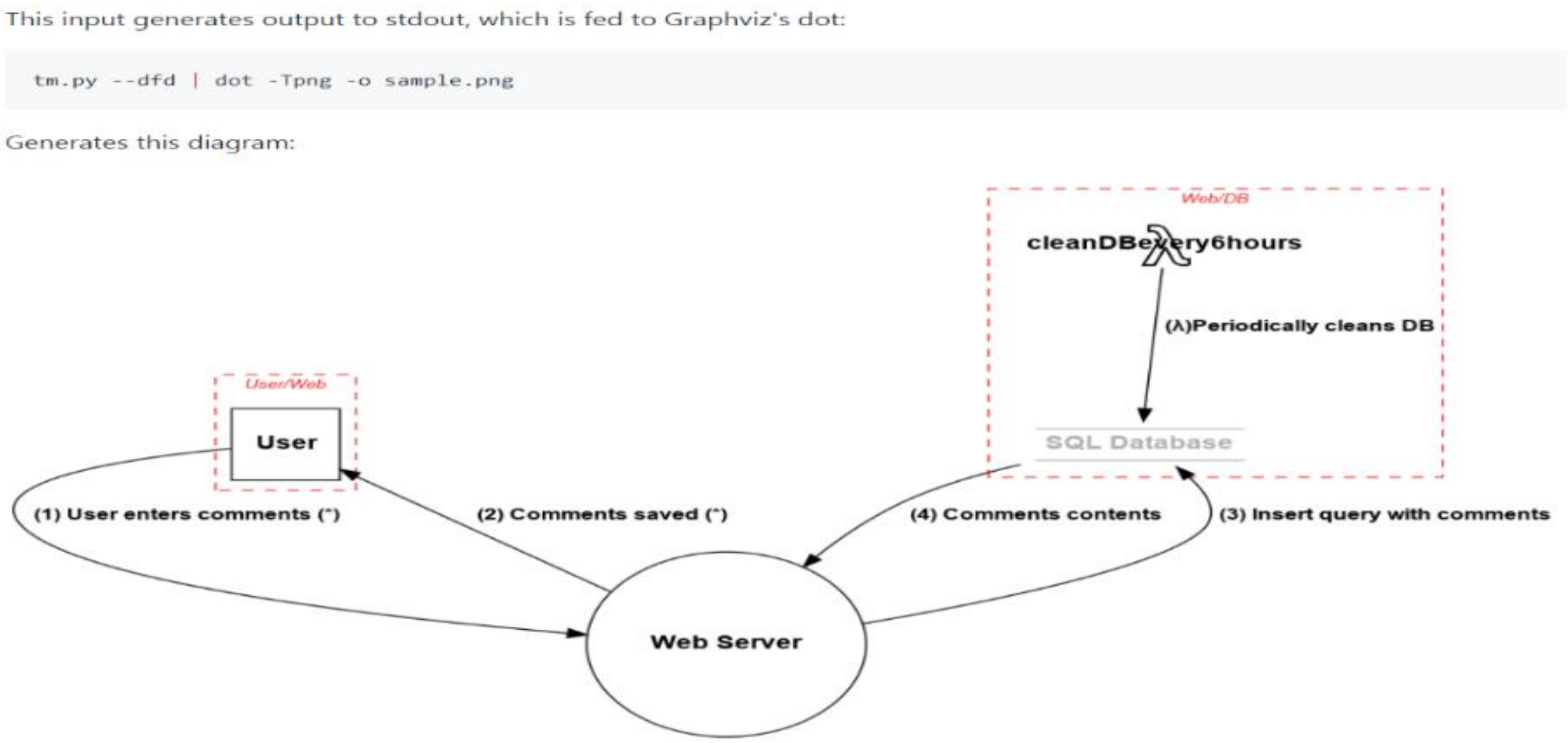
Threat Modeling WITH Code – **PyTM** (2018)

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)

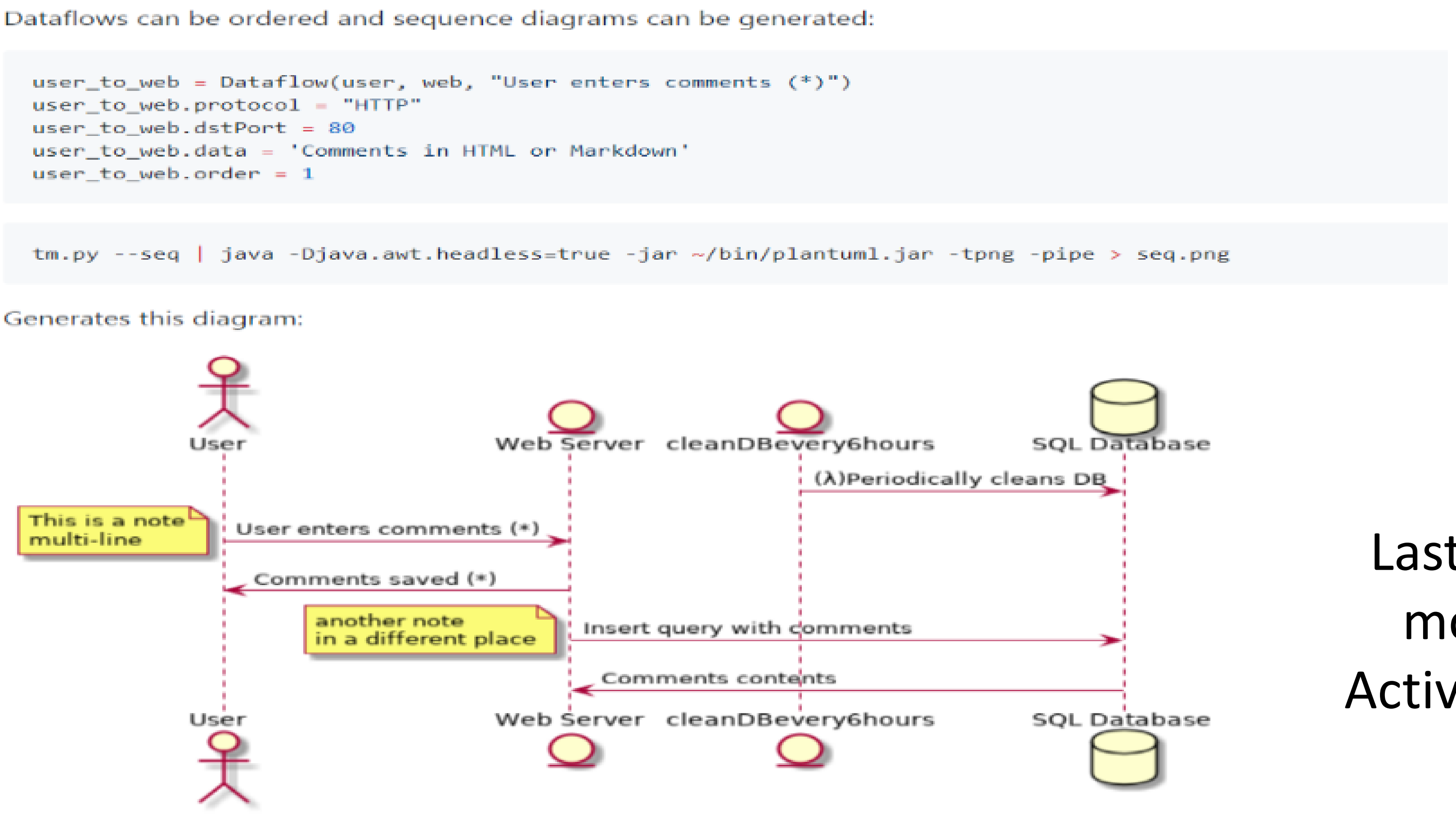


This input generates output to stdout, which is fed to Graphviz's dot:

```
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:

Last update – 2 months ago. Actively updated.

Open Threat Model (OTM) Standard (2023)

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 (JSON)

Read more here: https://www.iriusrisk.com/resources-blog/introduction-to-the-open-threat-model-standard

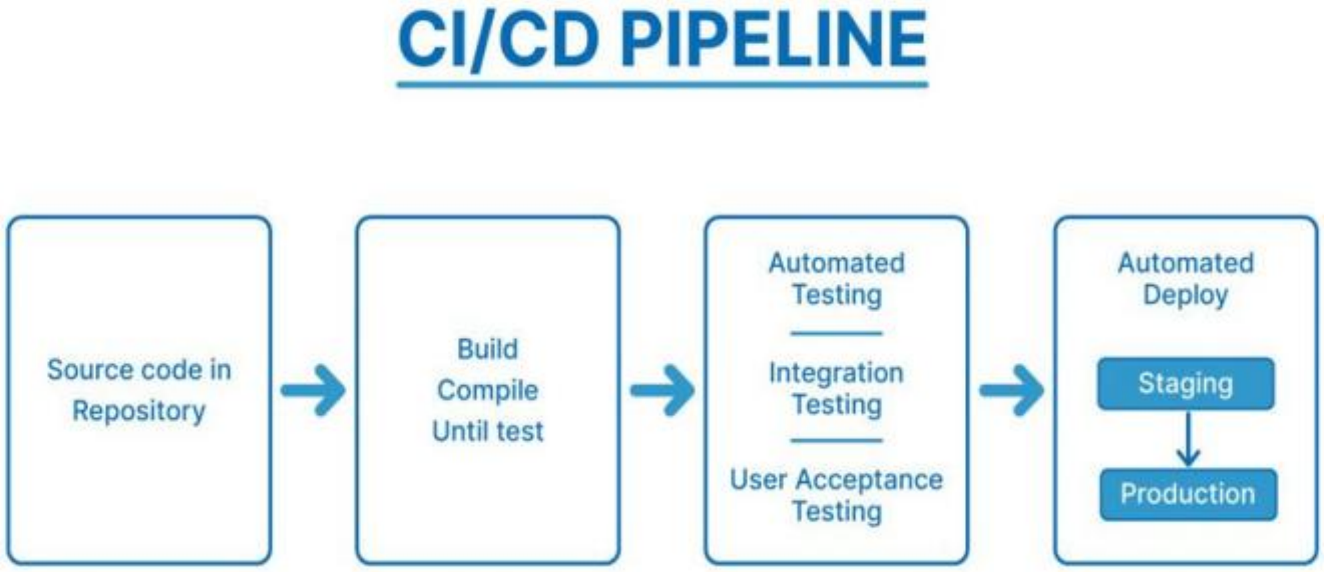
Code: https://github.com/iriusrisk/OpenThreatModel

```
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. …



* 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:

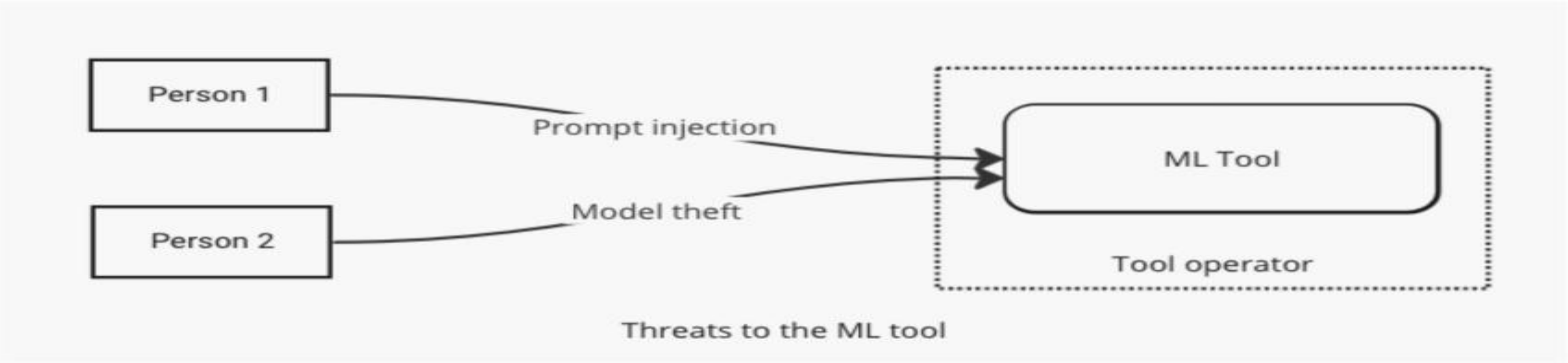
- Attack technique dataset
- Network log dataset
- Vulnerability dataset
- Threat intelligence dataset

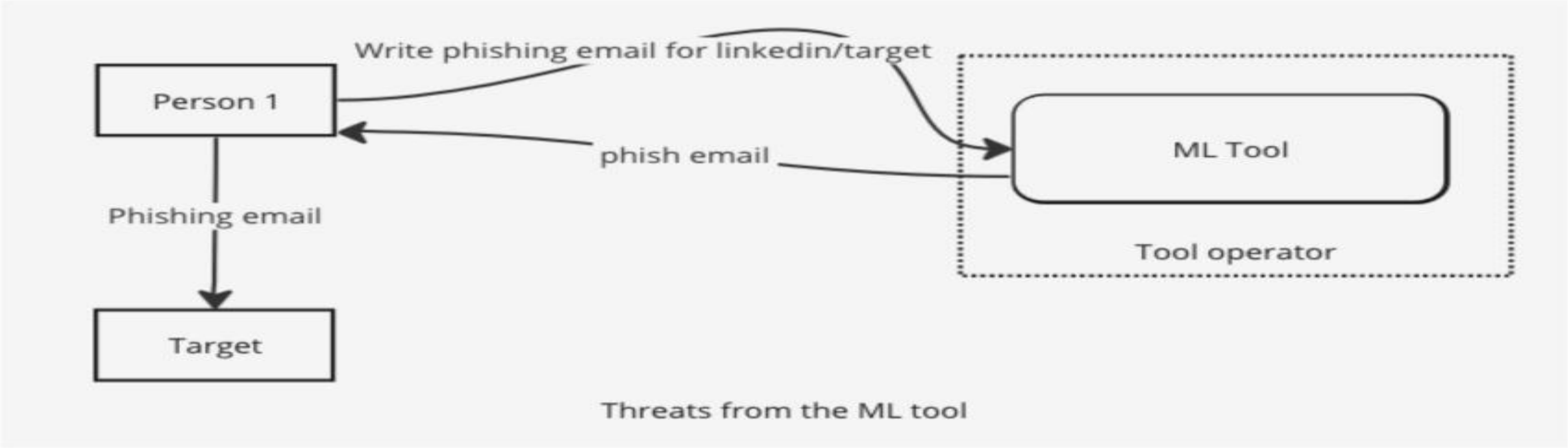
AI can help with automation and efficiency, continuous real-time monitoring, adaptive defense, scalability, threat detection / Intelligence

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.

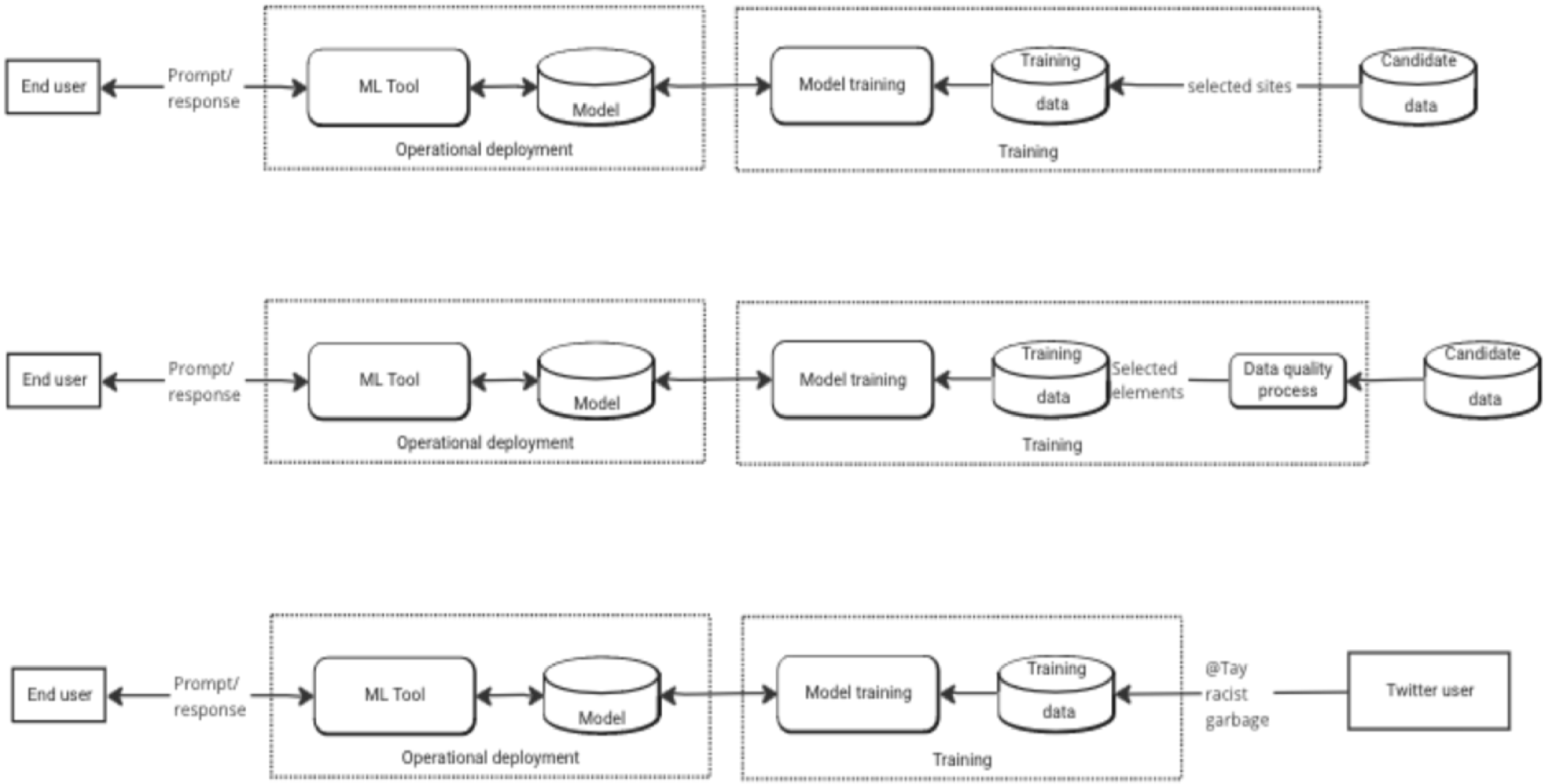
Threat model diagrams for ML





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)?

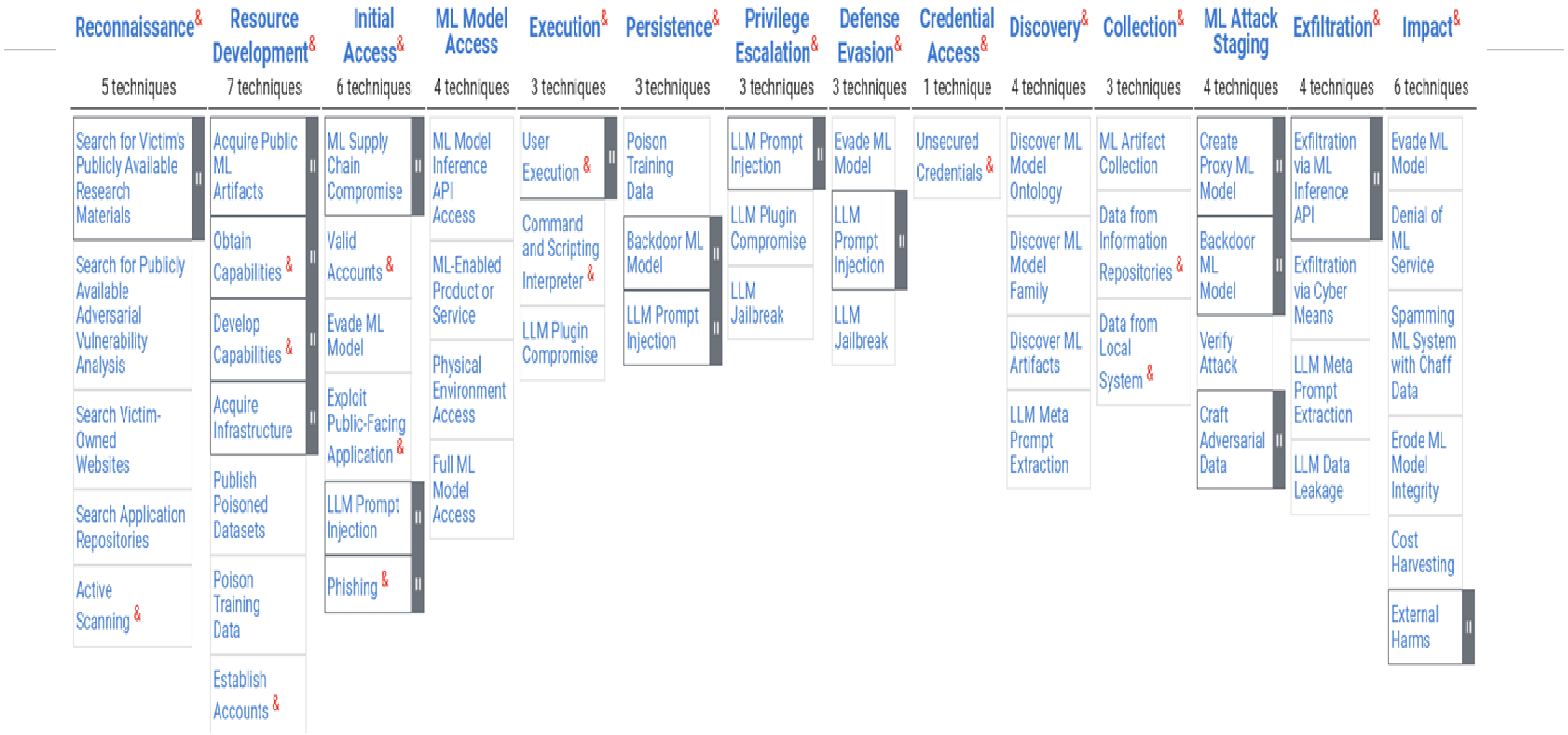
There are still issues around AI/ML which benefit from analysis using STRIDE:

- 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.

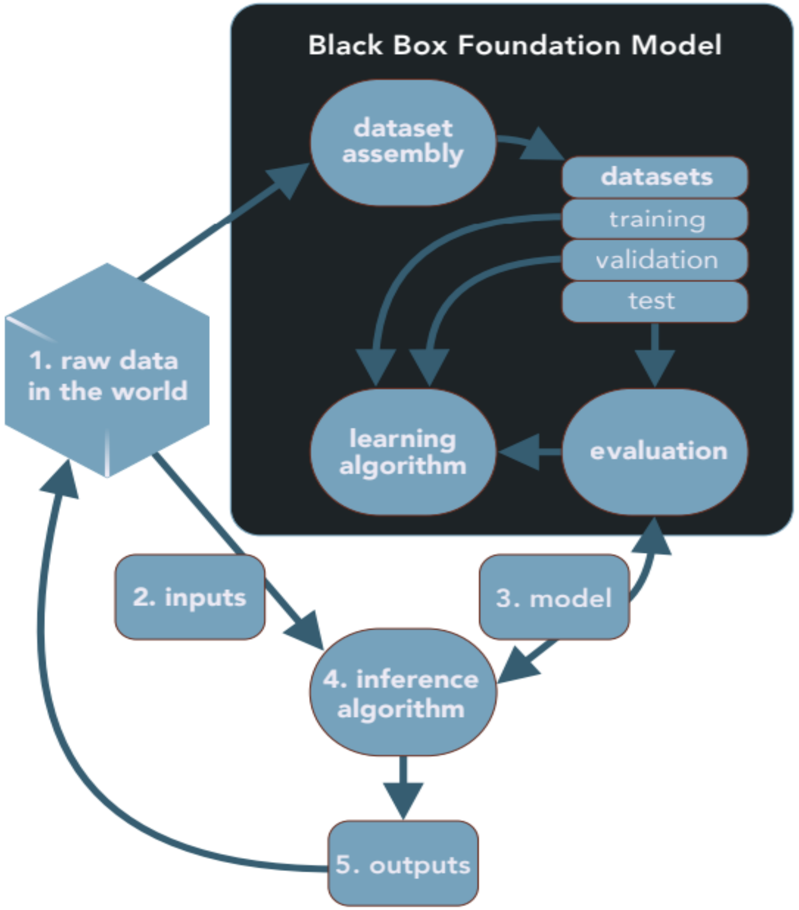
MITRE ATLAS Matrix



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

What next?

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/

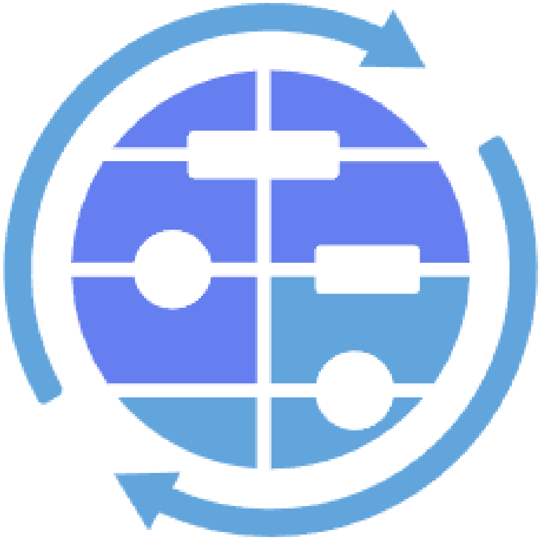
- Definition
- Values
- Principles
- Anti-Patterns

Resources – Threat Modeling Capabilities

“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

BlueSky: roberthurlbut.bsky.social

LinkedIn: roberthurlbut

Discord: robert.ct (robertct)

Thank you!