Tufts Security & Privacy Lab

medcrypt

Present

Threat Modeling in 90 minutes

If you haven’t completed the anonymous intake survey, go to:
go.tufts.edu/cybermed

Bottom Line Up Front

What you will learn today:

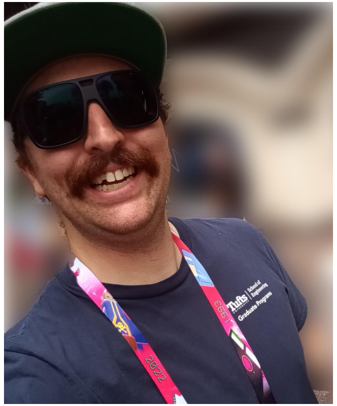
- A means to communicate about the security of a system with any audience
- How to take a no-frills approach to threat modeling
- Specific outcomes that you should expect from every threat modeling session

Why listen to us:

- We bring combined perspectives as practitioners, regulators, and researchers
- TSP Lab is actively researching threat modeling for medical devices
  - Previously worked with NYC Cyber Command
  - Currently studying MDMs usage of threat models

Insights are informed by our research, experience, and where noted external sources

Your Instructors



Ron Thompson (he/his)
CS PhD Student
Tufts Security & Privacy Lab

Prior: Data Engineer/Scientist, Sys Admin, Behavioral Scientist, MedTech Investment Analyst

Naomi Schwartz
Sr. Dir - Cybersecurity Quality & Safety
MedCrypt

Prior: FDA Software & Cybersecurity Reviewer, Radar/Electronic Warfare Engineer, Standards Liaison

What are we doing today?

Agenda

0900 - 0920	Introduction to Threat Modeling
0920 - 0930	Form Groups
0930 - 1115	90(ish) Minute Threat Model
1115 - 1120	Break
1120 - 1135	Selecting the right tools in your toolbox
1135 - 1145	Making a living threat model
1145 - 1150	5-minute reflection
1150 - 1200	Q&A

Introduction to Threat Modeling

How we define threat modeling

Threat Modeling is a formal and systematic approach to reviewing system and software architecture and design with the goal to identify exposure to cyber threats and to analyze and communicate cybersecurity risks as well as to implement architecture and design improvements to address these

- Ron Thompson & Axel Wirth

Why threat model?

!!!NOT JUST FOR COMPLIANCE!!!

Why threat model?

Discover potential security issues

Systematic approach to understand why you need your security requirements

Identify potential problems in design before you start building

Communicate security issues / risks and why mitigations are needed

Adapted from *Threat Modeling* by Adam Shostack &
Threat Modeling by Frank Swiderski and Window Snyder

When not to threat model?

When you need to…

- “quantify” security risk
- verify your implementation
- triage a vulnerability

Instead use as an input/guide for…

- performing risk assessment
- running static code analysis / fuzzer
- Stay ahead of security

Threat model early and often to get the most benefit

Adapted from *Threat Modeling* by Adam Shostack &
Threat Modeling by Frank Swiderski and Window Snyder

Guiding questions

Why are we here?

What are we building?

What can go wrong?

What are we going to do about it?

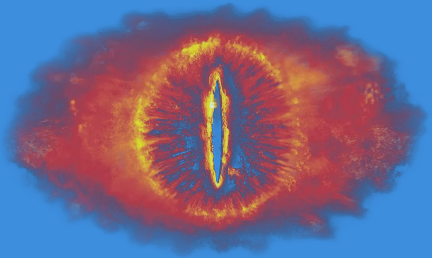
Did we do a good enough job?

Adapted from the *Threat Modeling Manifesto*

Defining terms with Lord of the Rings

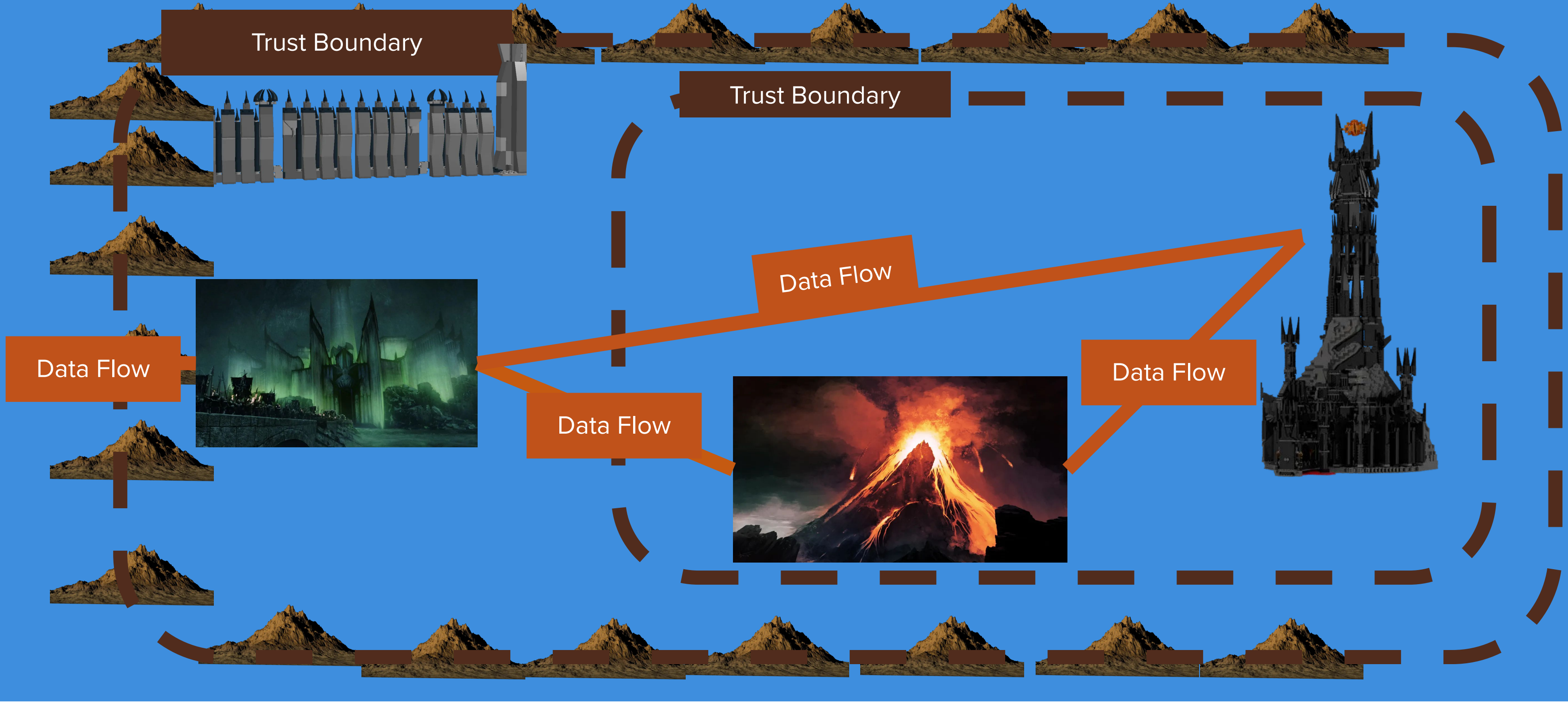
Nazgûl
(Security Team)

Barad-dûr
(Most Critical Asset)



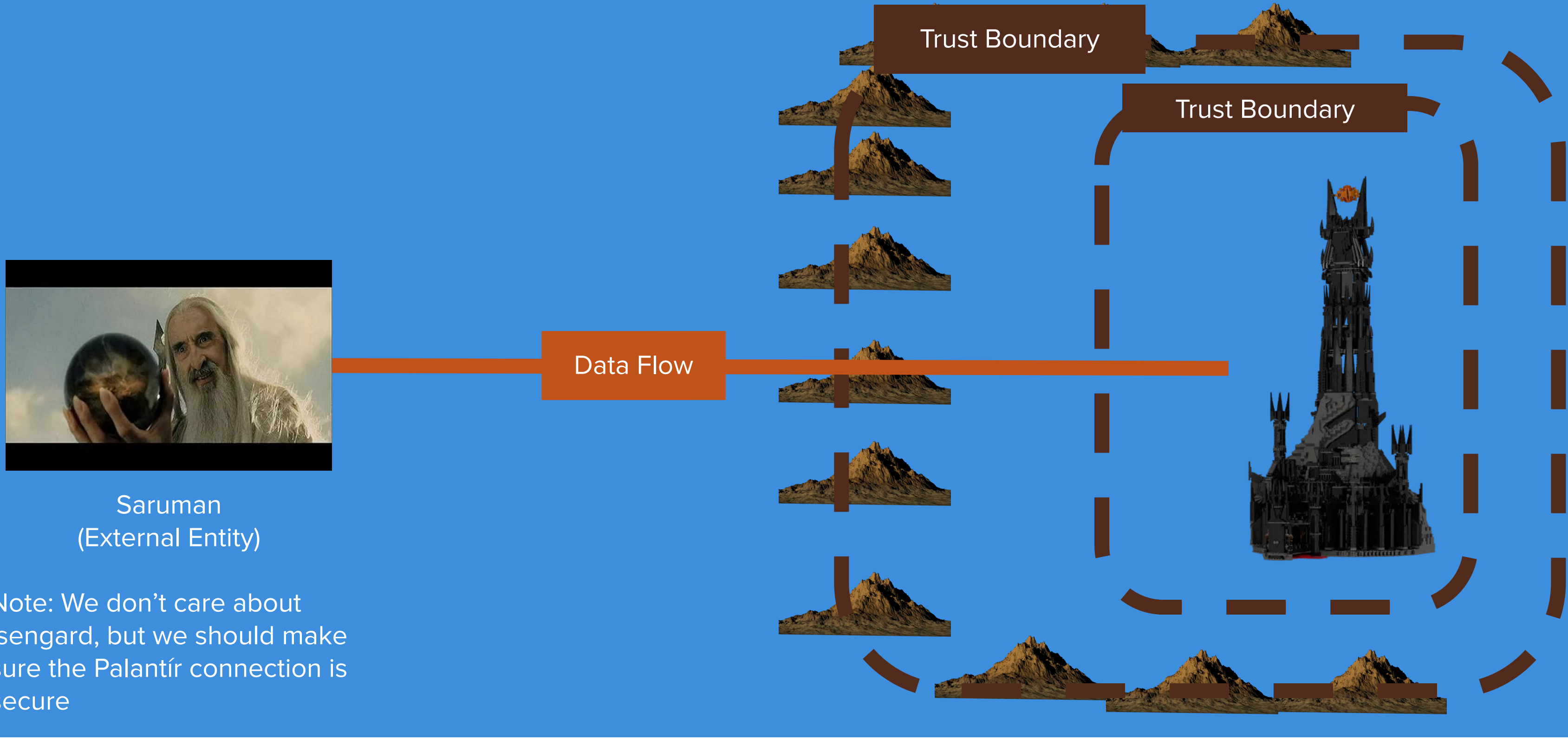
Eye of Sauron
(Most Critical Function)

Data Flow Diagram for Mordor



Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Data Flow Diagram for Mordor - External Entities



Note: We don't care about Isengard, but we should make sure the Palantír connection is secure

Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Attack Surfaces of Mordor

Black Gate
(Asset)

Minas Morgul
(Asset)

Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Threat Actors - The Fellowship of the Ring

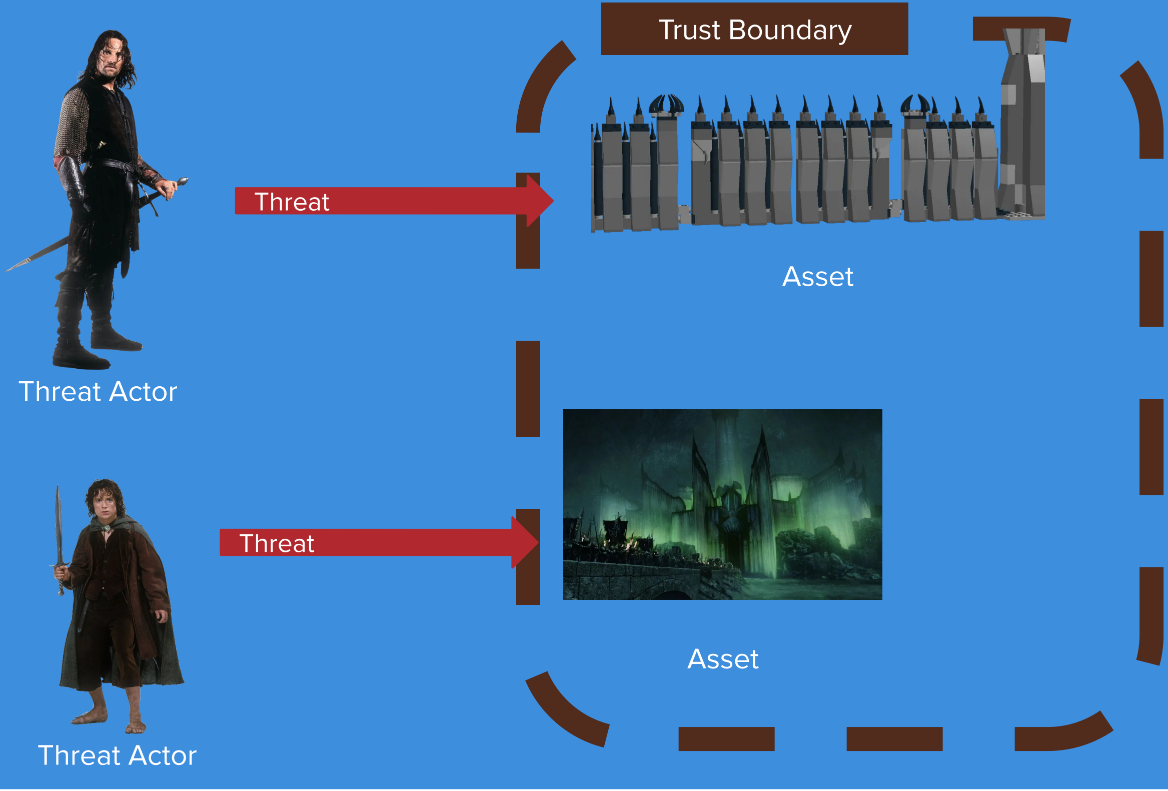
Frodo (Threat Actor)

Exploit to destroy
Barad-dûr

Aragorn (Threat Actor)

Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Threats to Mordor



Trust Boundary

Threat Actor

Threat

Asset

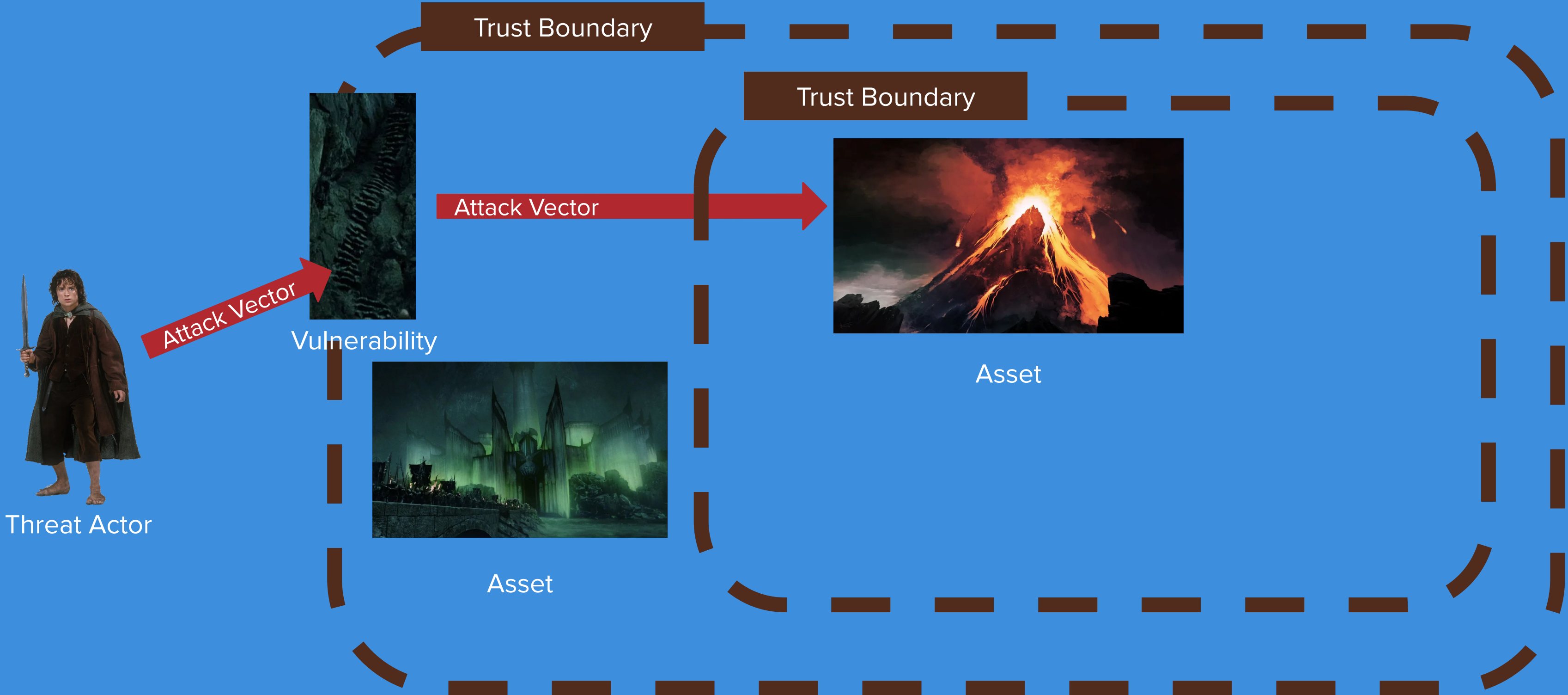
Threat Actor

Threat

Asset

Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Exploiting the Stairs of Cirith Ungol



Trust Boundary

Trust Boundary

Threat Actor

Attack Vector

Vulnerability

Attack Vector

Asset

Asset

Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Attacking the final target

Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Security Incident

Homage to Adam Shostack
(copying his awesome Star Wars metaphor felt uncool)

Some things to note

We are teaching a way of thinking not a checklist

We won’t “score” threats

Explain your thought process

Progress not perfection

Be respectful of your fellow students

Split into groups

Try to be with people that:

- You don’t know
- Don’t work with the same things (patients/devices/networks/security/other)

Roles:

- Easel admin to draw and write
- Scribe to take notes
- Reporter to summarize for the group

Introduce yourselves

Name

Pronouns

What do you primarily work with? [Patients/Devices/Networks/Security/Other]

Have you had experience with threat modeling? [Yes/No]

90-minute Threat Model

90 minutes?

What do we want to accomplish?

What are we building?

- Understand what the system is and how it works
- Discover the most critical assets and workflows
- Identify attack surfaces, and trust boundaries

What can go wrong?

- Analyze threats

What are we going to do about it?

- Propose mitigations

What does this look like?

Evenly split time between each set of questions

- 30 minutes on Architecture/Design
- 30 minutes on Threats
- 30 minutes on Mitigations

We'll use a slightly abbreviated version to accommodate training

Introducing MedCrypt’s Adrenal Gland Implant (AGI)

Developed originally as a permanent implant to treat Addison's Disease, we discovered it can be effectively used in both male and female patients to prevent or augment fertility. This universal adrenal gland regulator is permanently implanted in the patient's abdomen. We are working on connecting the implant controller so that it can be externally and remotely controlled based on the patient's current desired clinical outcomes (i.e., pregnancy prevention or achieving pregnancy).

AGI IS A FICTIONAL DEVICE
ANY RESEMBLANCE TO AN ACTUAL DEVICE IS PURELY COINCIDENTAL

AGI Today

Features

- Implantable device that periodically delivers a cocktail of drugs
- A physical controller sets these levels before the device is implanted into the patient
- Only way to change these levels is by physically removing the implant

Potential for Harm

- This drug cocktail if given in high enough dosages can lead to patient death

AGI IS A FICTIONAL DEVICE
ANY RESEMBLANCE TO AN ACTUAL DEVICE IS PURELY COINCIDENTAL

Updates to AGI

Required Features

- Implant needs to talk to a remote controller - our team is leaning towards using a mobile phone application
- Dosage amounts will need to change periodically based on the treatment plan designed by the clinician
- Clinician should be able to update the treatment plan remotely

AGI IS A FICTIONAL DEVICE
ANY RESEMBLANCE TO AN ACTUAL DEVICE IS PURELY COINCIDENTAL

Updates to AGI

Desired Features

- Clinician can send messages to the patient via the remote controller
- Patient is able to view their treatment plan via the remote controller
- Connect the treatment plan directly into the clinician’s EHR rather than them manually adding
- We have hired data scientists to see if we can provide recommendations on treatment plans to clinicians based off of EHR data and prior treatment plans
    - They need to have access to the data from AGI and the EHR
    - When they have built an AI engine this should be integrated into the system

In your groups

- Discuss AGI’s requirements
- Build a pictorial representation of what the system would look like
- Write down or draw the major clinical and technical workflows
- Think about what third-parties you might have to interact with and capture those

Some ways to represent & think about threats

Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win.

- John Lambert, Microsoft

Some ways to represent & think about threats

Map out a route an attacker might take, what conditions need to be met

Classify threats based on security principles, such as:

- Confidentiality, Integrity, Availability, Safety
- Authentication, Authorization, Auditing

What components need to trust one another? What permissions do they need?

What unilateral actions might an insider threat be able to make?

Adapted from *Fundamentals of Computer Security Technology* by Edward Amoroso, *Threat Modeling* by Adam Shostack & *Designing Secure Software* by Loren Kohnfelder

Mindsets

Try to take a varied mindset:

- What are my riskiest assets? Are these the same as my highest priority assets?
- What data flows/communication channels am I most worried about?
- Even if a feature is meant to do something, can someone manipulate it to do something else?
- If I were to attack this system what would I be going after? What’s the path of least resistance?
- Who can I trust? What can I do to promote trust?

In your groups

- Brainstorm potential threats to the system
  - What’s the worst thing that can happen?
  - What are some other potential security issues? Do they threaten the safety of the patient?
- Associate the threats with the workflows you came up with

General ways to address threats

Eliminate - remove asset or connection or feature if it’s not necessary

Mitigate - redesign or add defense to lower occurrence or degree of harm

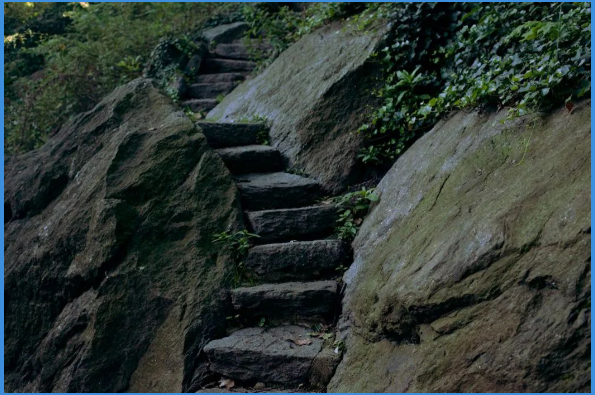
Transfer - offload responsibility to a third-party (ex insurance or specialized vendor)

Accept - determine risk is reasonable to occur

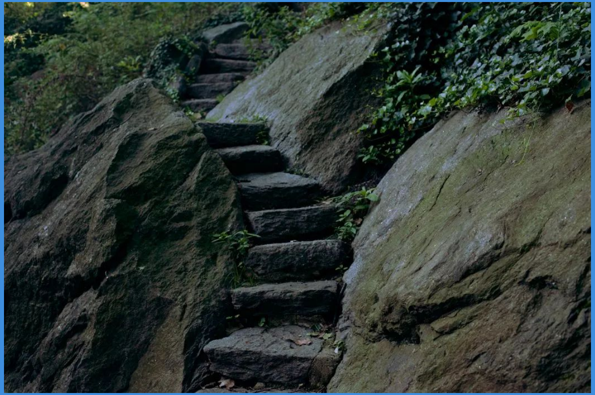
Eliminate

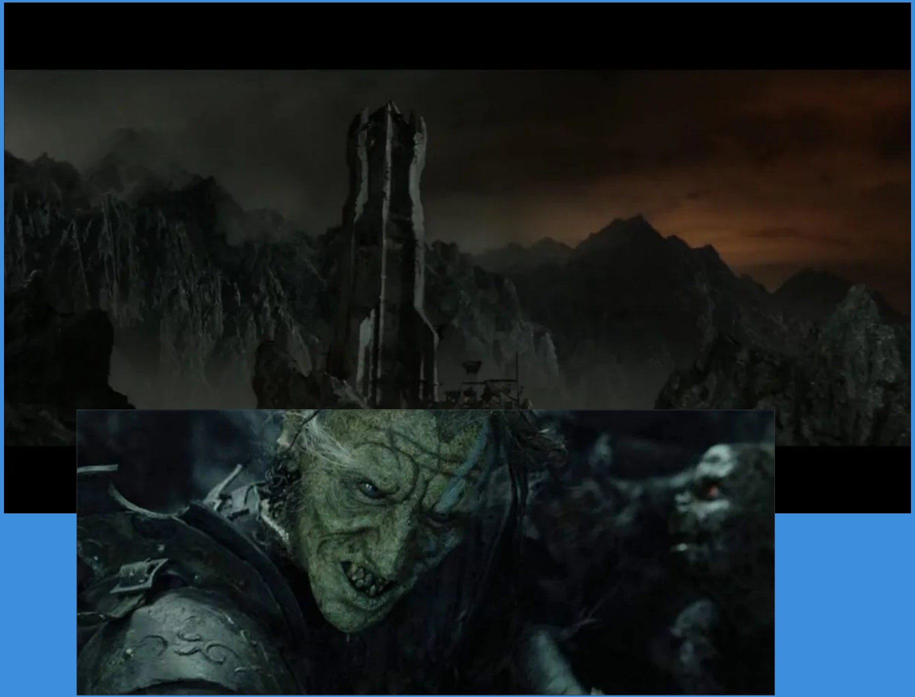
Remove asset or connection or feature if it’s not necessary

Zero Day?



Mitigate



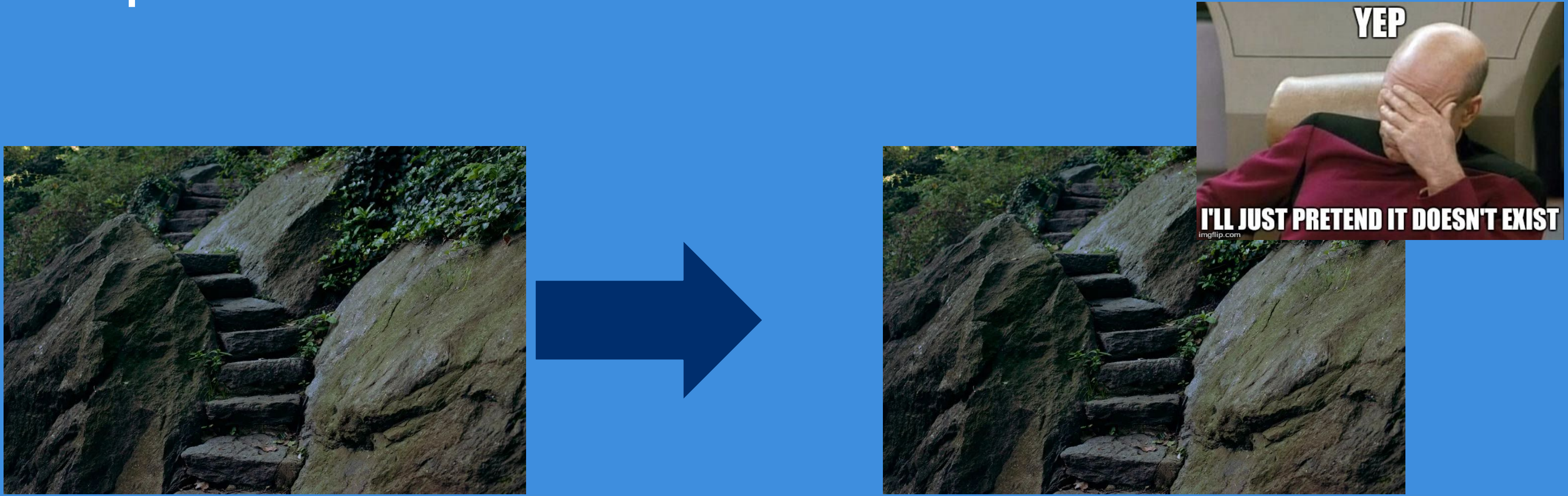


Redesign or add defense to lower occurrence or degree of harm

Transfer

Offload responsibility to a third-party (ex insurance or specialized vendor)

Accept



Determine risk is reasonable to occur

In your groups

- For each threat you identified
  - What approach would you take [Eliminate, Mitigate, Transfer, Accept]
  - Why you have chosen that approach
  - Explain how you might achieve that approach, such encryption or with Software License Agreements

5-Minute Break!

What are we doing today?

Agenda

0900 - 0920	Introduction to Threat Modeling
0920 - 0930	Form Groups
0930 - 1115	90(ish) Minute Threat Model
1115 - 1120	Break
1120 - 1135	Selecting the right tools in your toolbox
1135 - 1145	Making a living threat model
1145 - 1150	5-minute reflection
1150 - 1200	Q&A

Selecting the right tools from your toolbox

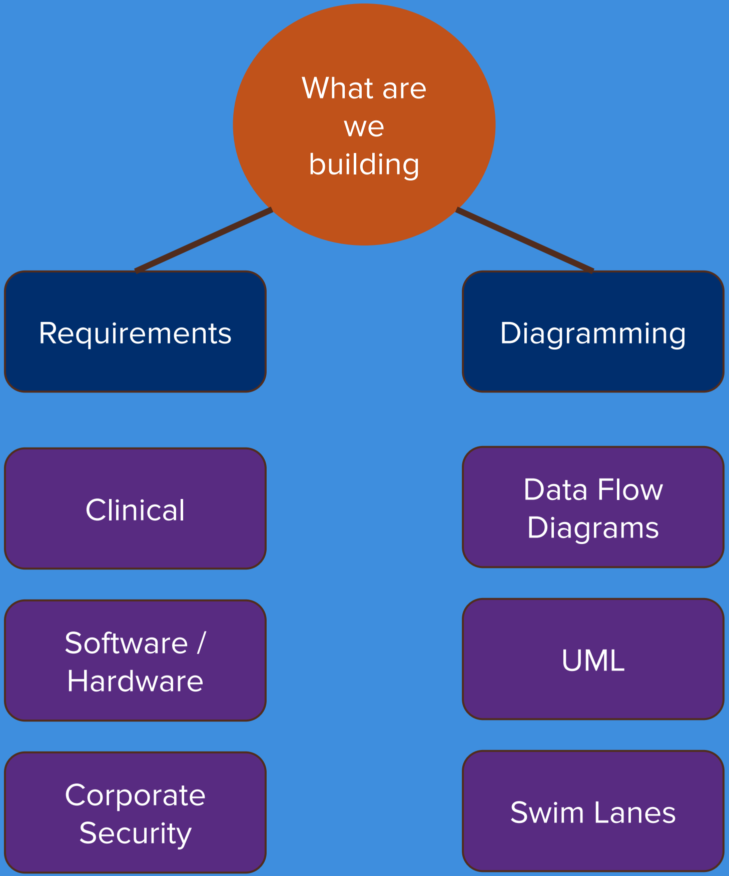
THIS IS
NOT THE ONLY
WAY

What are we building

What can go wrong

What are we going to do about it?

Did we do a good enough job



What are we building

Requirements

Diagramming

Clinical

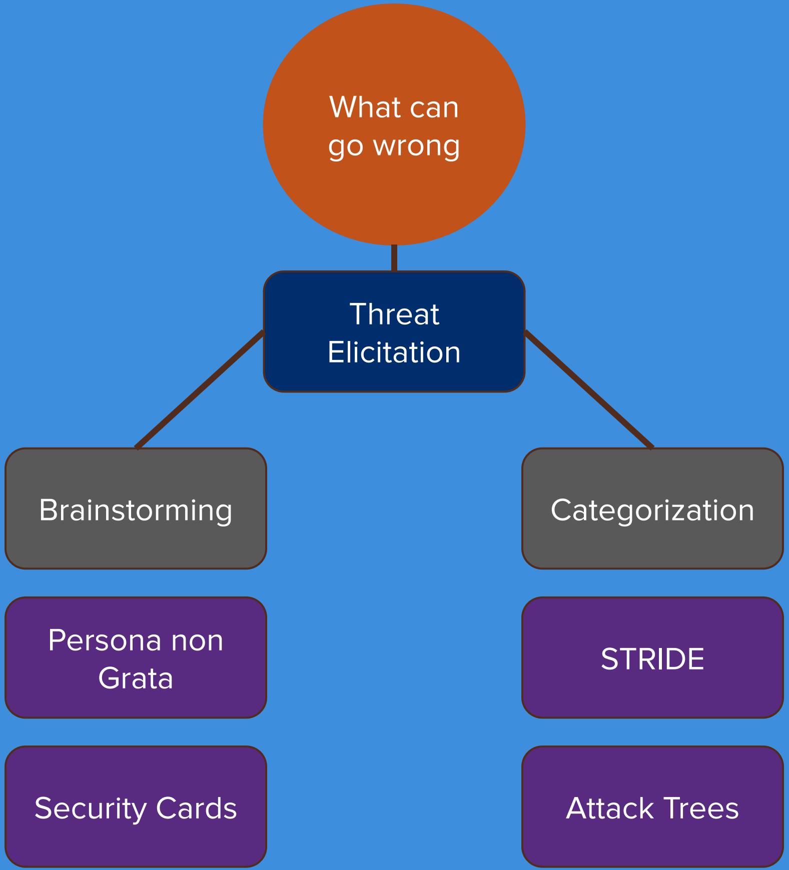
Software / Hardware

Corporate Security

Data Flow Diagrams

UML

Swim Lanes



What can go wrong

Threat Elicitation

Brainstorming

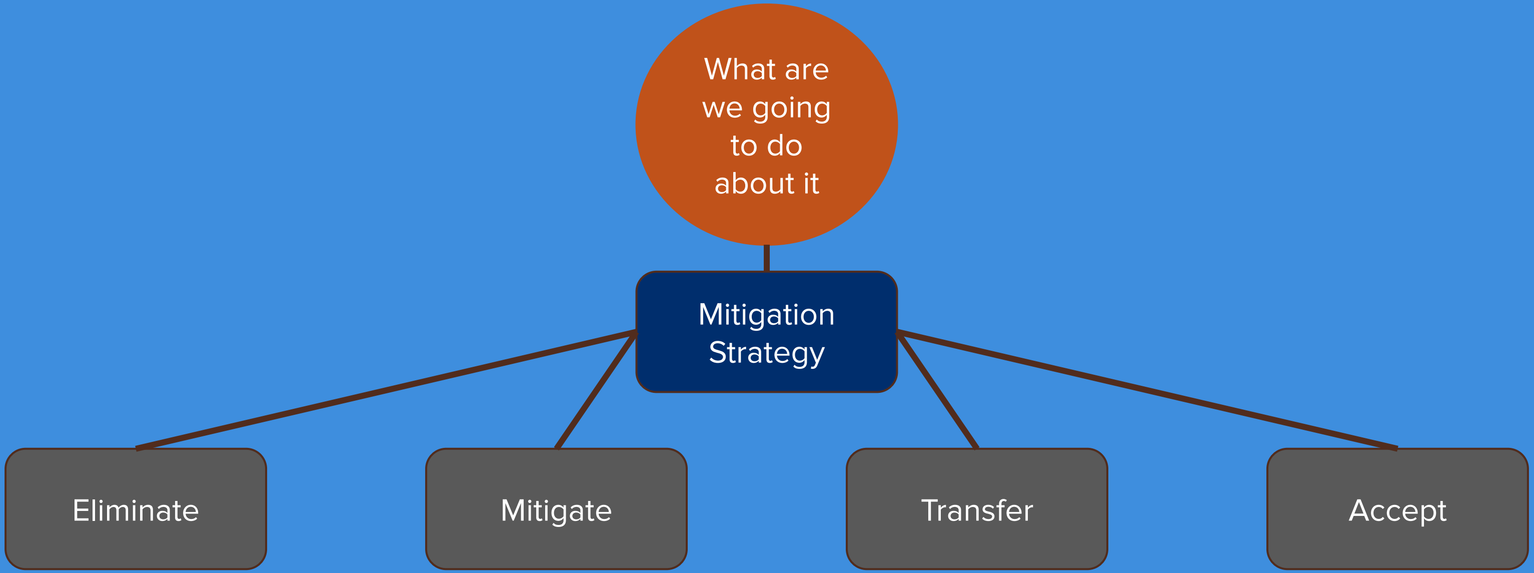
Persona non Grata

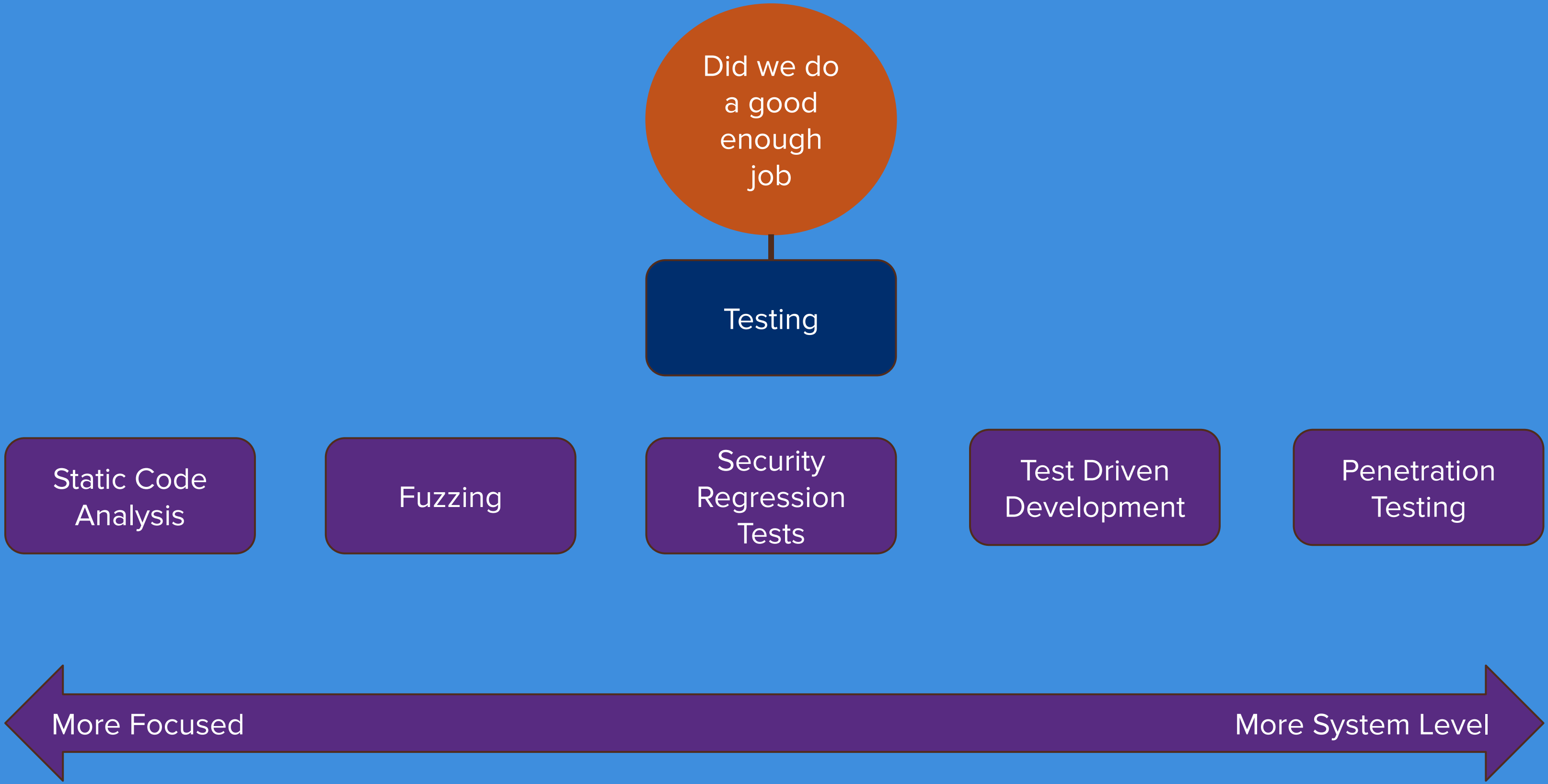
Security Cards

Categorization

STRIDE

Attack Trees


What are we going to do about it
Mitigation Strategy
Eliminate
Mitigate
Transfer
Accept


Did we do a good enough job
Testing
Static Code Analysis
Fuzzing
Security Regression Tests
Test Driven Development
Penetration Testing
More Focused
More System Level

Some “higher level” approaches

Process for Attack Simulation and Threat Analysis (PASTA)

- 7 stage process that guides how to go about answering the four questions
- Risk-centric approach

Center of Gravity (CoG)

- Military strategy to defend “primary entity that possesses inherent capability to achieve the objective”
- Protect central resource

From *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis* by Tony UcedaVelez and prior work by TSP Lab

What does this mean for you?

Your Threat Modeling Toolbox

What does this mean for you?

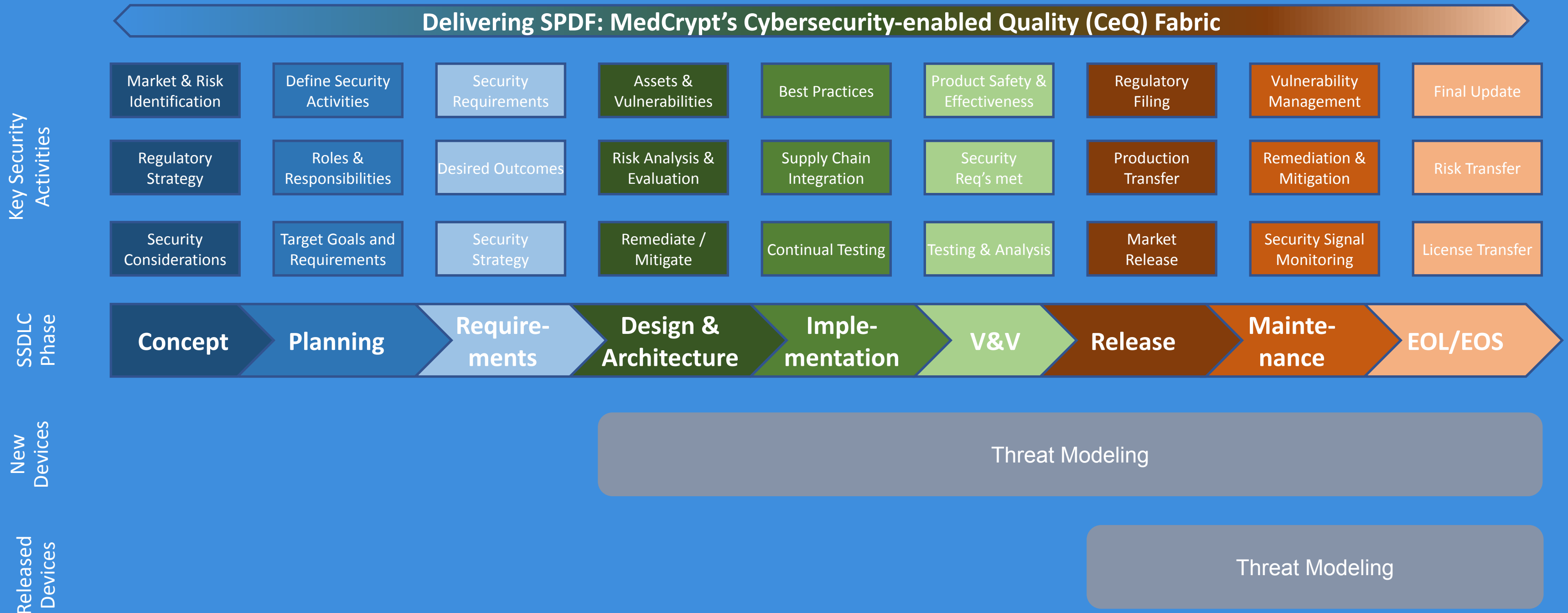
Don’t necessarily wed yourself to a specific framework, instead build process that can adapt to the situation

Ability to mix and match. Examples (**not the only approach for these scenarios**):

- Defending an enterprise network: use CoG as your guiding framework with Data Flow Diagrams and STRIDE
- An embedded system with no networking: use swimlanes with attack trees

Making a living threat model

Secure Software Development Lifecycle



Delivering SPDF: MedCrypt's Cybersecurity-enabled Quality (CeQ) Fabric

Key Security Activities	Market & Risk Identification	Define Security Activities	Security Requirements	Assets & Vulnerabilities	Best Practices	Product Safety & Effectiveness	Regulatory Filing	Vulnerability Management	Final Update
	Regulatory Strategy	Roles & Responsibilities	Desired Outcomes	Risk Analysis & Evaluation	Supply Chain Integration	Security Req's met	Production Transfer	Remediation & Mitigation	Risk Transfer
	Security Considerations	Target Goals and Requirements	Security Strategy	Remediate / Mitigate	Continual Testing	Testing & Analysis	Market Release	Security Signal Monitoring	License Transfer
SSDLC Phase	Concept	Planning	Require-ments	Design & Architecture	Imple-mentation	V&V	Release	Mainte-nance	EOL/EOS
New Devices				Threat Modeling					
Released Devices							Threat Modeling		

Tufts Security & Privacy Lab

Courtesy of Axel Wirth, MedCrypt

medcrypt

Documentation

Make sure your documentation meets your development process needs

- Documentation for developers about security requirements to be implemented
- Justifications for executives about why a security requirement is needed (*security isn’t meant to be for security’s sake*)
- Linking workflows to threats/controls to explain why a control is needed in IFU
- Connect your assets/connections to implementations so you can trace discovered vulnerabilities

Oh, and if you do all of this it will easily meet regulatory requirements

Spend 5 minutes reflecting on the workshop and what you learned.
Prompting questions:

- How would you define threat modeling?
- What concepts or aspects do you think are the hardest to understand?
- What questions do you have?

What are we doing today?

Agenda

0900 - 0920	Introduction to Threat Modeling
0920 - 0930	Form Groups
0930 - 1115	90(ish) Minute Threat Model
1115 - 1120	Break
1120 - 1135	Selecting the right tools in your toolbox
1135 - 1145	Making a living threat model
1145 - 1150	5-minute reflection
1150 - 1200	Q&A

Want to craft the future of threat modeling?

Sign-up for our current study

go.tufts.edu/mdm_threat_model

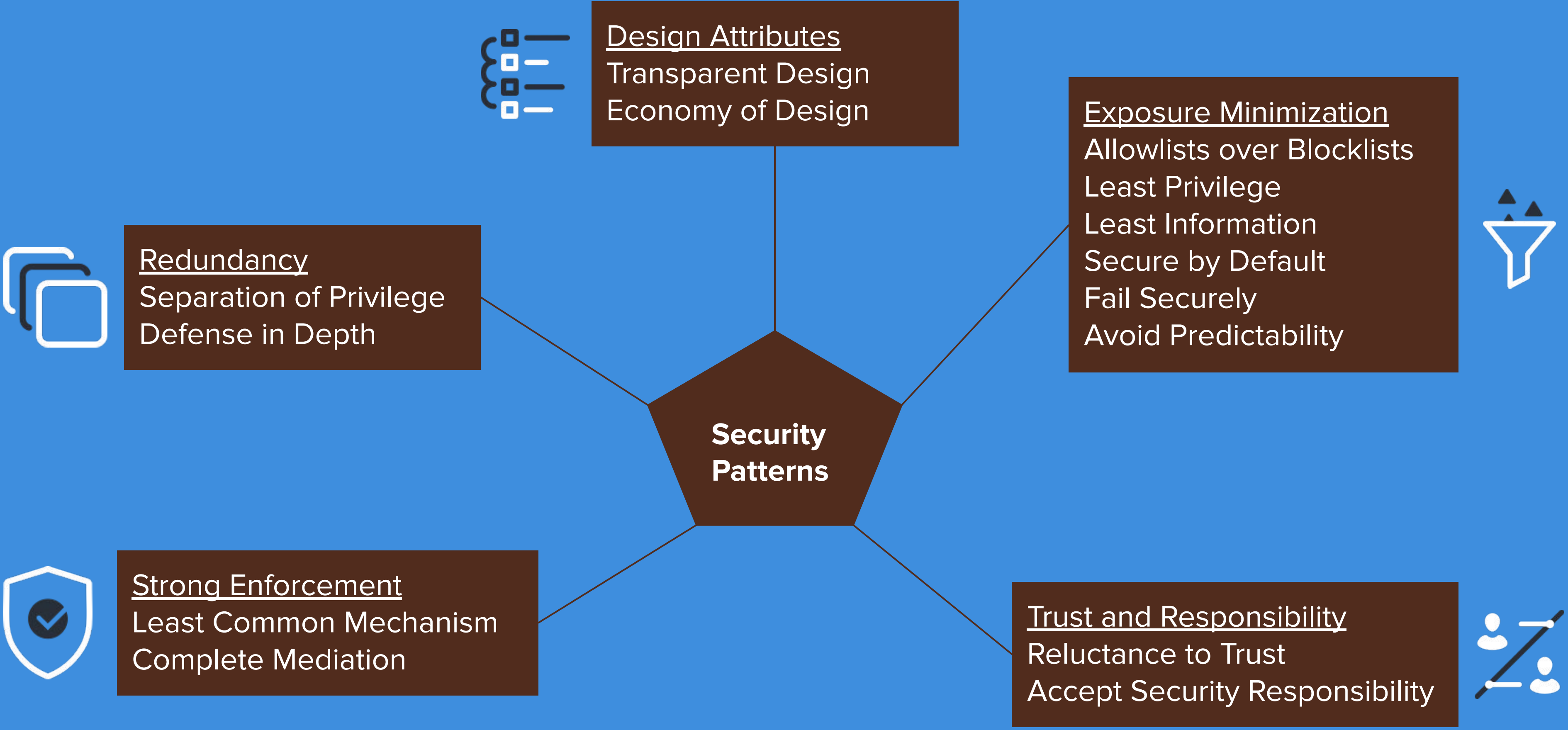
Learn more:

tsp.cs.tufts.edu/medical-devices

Contact:

medical-devices@cs.tufts.edu

Archive



Security Patterns

Design Attributes
Transparent Design
Economy of Design

Exposure Minimization
Allowlists over Blocklists
Least Privilege
Least Information
Secure by Default
Fail Securely
Avoid Predictability

Redundancy
Separation of Privilege
Defense in Depth

Strong Enforcement
Least Common Mechanism
Complete Mediation

Trust and Responsibility
Reluctance to Trust
Accept Security Responsibility

From *Designing Secure Software* by Loren Kohnfelder