Enter the Threat Dragon

OWASP Threat Dragon workshop

- Walk through the Threat Dragon features
- Showcase a simple example model
- Run through of a modeling session
- No prior experience neccessary

Introduction

OWASP Threat Dragon project and documentation

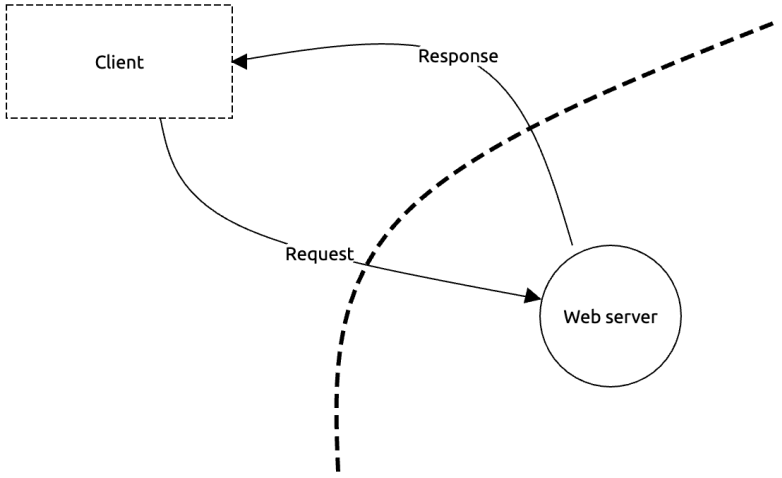
Project leaders

- Jon Gadsden
- Leo Reading
- Mike Goodwin – original author

Cupcake, making threat modeling less threatening

What to expect

- Installing
- Creating projects
- Creating diagrams
- Adding threats
- Putting it all together
- How it works in practice
- Call for help



Workshop

How it should work:

- 6 sections
  - Talk for no more than 15 minutes
  - Do the practical
  - A short discussion to allow catch-up

Context

Threat modeling as part of a Secure Development Lifecycle

- Security & crypto requirements
- **Threat modeling bitesize #1**
- Secure coding
- Third Party Software
- Static application security-testing
- **Threat modeling bitesize #2**
- Dynamic application security-testing

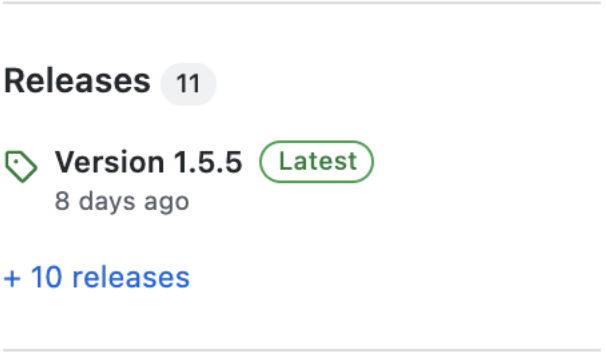
Context

- Required by various standards bodies
- Mitigation for OWASP *A04:2021 - Insecure Design*
- Incremental – make it bitesize
- Collaborative involving the whole team

Refer to the OWASP Threat Modeling project

Installing

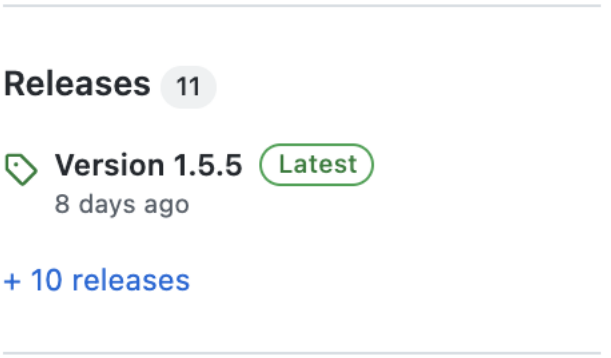
- Desktop version
  - Linux AppImage, Snap, deb and rpm
  - MacOS Apple Disk Image
  - Windows NSIS installer
- Web Application version
  - Docker container
  - From source



Installing

Desktop for MacOS or Windows

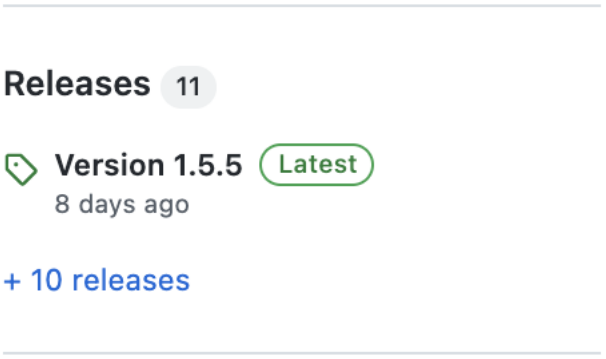
- Download from github site
- .dmg MacOS Apple Disk Image (also .zip)
- .exe Windows NSIS installer



Installing

Desktop for Linux systems

- Snap from the snapcraft site
- Download from github site:
  - AppImage
  - .deb or .rpm installers



Installing

As a web application #1

- Either container using dockerhub image
- Or direct from source

- Storage on github only (for now)
- Requires environment variables

Installing

As a web application #2

Environment variables – consider using .env

- GITHUB_CLIENT_ID
- GITHUB_CLIENT_SECRET
- NODE_ENV
- SESSION_STORE
- SESSION_SIGNING_KEY
- SESSION_ENCRYPTION_KEYS

Practical #1

Install the desktop version:

- Either Linux
- Or Windows
- Or MacOS

Alternatively the web application can be used

Discussion

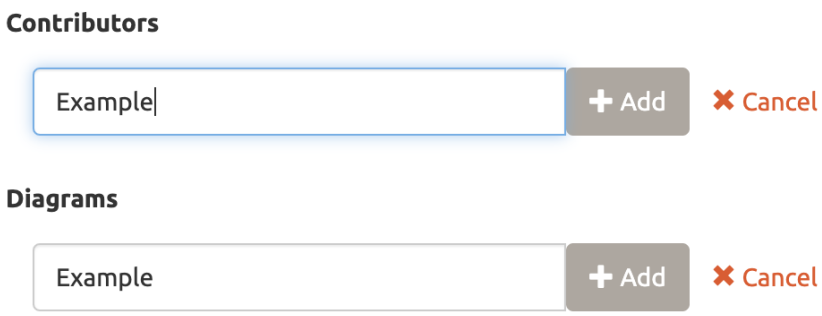
Of course there are alternatives

- Microsoft Threat Modeling Tool
- Text based threat modeling: eg OWASP pytm
- Whiteboards are widely used

New Model

Contextual information

- Title - the threat model title cannot be empty.
- Owner – there is only one owner, can be a team
- Reviewer – there is only one reviewer, can be a team
- High level system description
- Contributor(s) – remember the ‘Add’ button
- Diagram(s) – remember the ‘Add’ button
- Diagrams are not (yet) hierarchical



Practical #2

Create a new model and add :

- Title
- Owner and Reviewer
- High level system description
- Add multiple Contributors
- Diagram + duplicate diagram

Cheat: download ‘step 1’ from docs.threatdragon.org/downloads/

Practical #2

Example threat model

Owner:
Threat Dragon workshop team

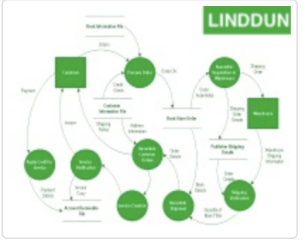
Reviewer:
Threat Dragon workshop attendees

Contributors:
Workshop attendee #1; Workshop attendee #1

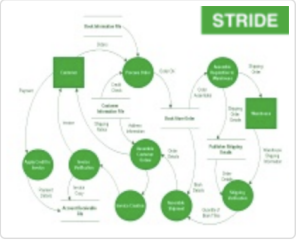
High level system description

This is an example model used for the PDX OWASP Training Day 2021
It is a threat model of Threat Dragon itself

Example



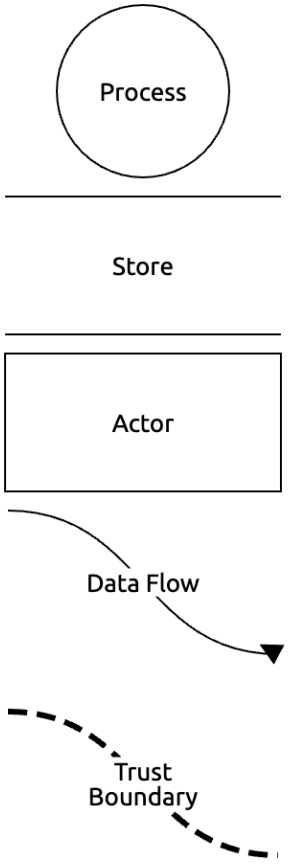
Copy of Example



Diagrams

Threat, not system, perspective

- Process
- Store
- Actor
- Data flow
- Trust boundary



Process

Usually a component under our control

- Name
- Description
- Out of scope? Reasoning

Context properties

- Privilege level

Store

Data at rest, almost always within the system but can be external

- The usual Name, Description, Out of scope? & Reasoning

Context properties

- Is a log?
- Stores credentials?
- Is encrypted?
- Is signed?

This could be regarded as an asset

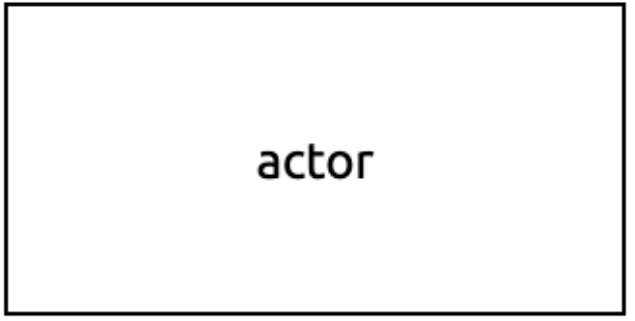
Actor

Commonly a component outside of our system

- The usual Name, Description, Out of scope? & Reasoning

Properties

- Provides authentication?



Data Flow

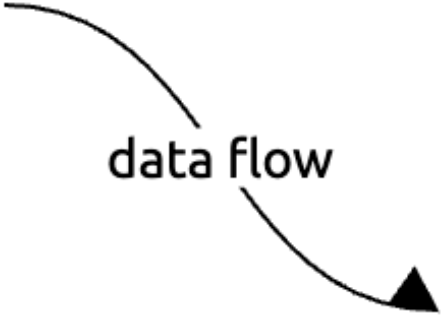
Data in transit, often cross trust boundaries

- The usual Name, Description, Out of scope? & Reasoning

Properties

- Protocol
- Is encrypted?
- Is over a public network?

Two ways to create data flow



Trust Boundary

- Name is optional in this case
- No other properties
- It is not a box (yet)
- *The most important of components*

Scope

Scope for diagram components

- Components can be declared out of scope
- Useful for focussing on important components
- Boundaries never out of scope
- Try and give a reasoning
- *Helps incremental*

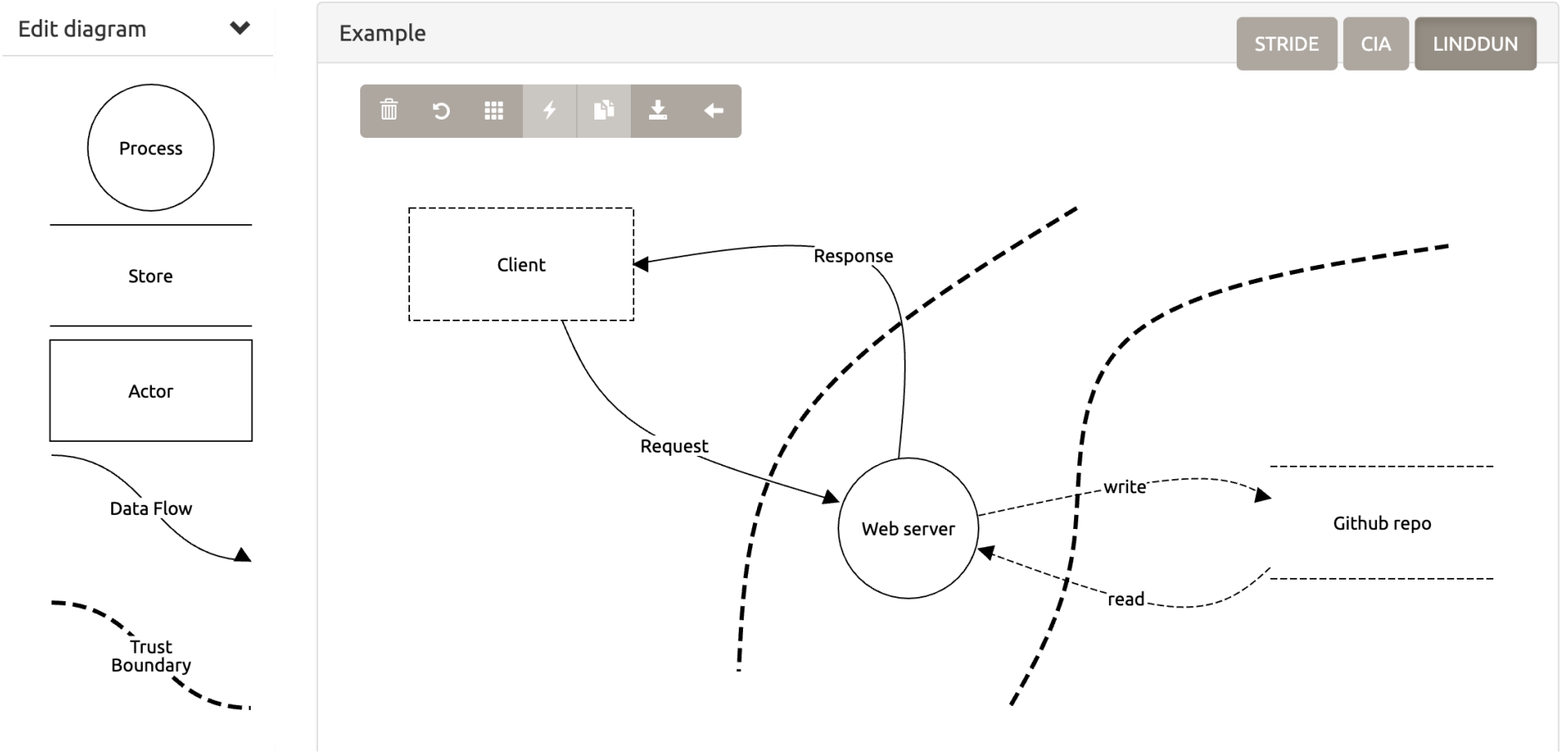
Practical #3

Add elements to the new diagram

- Processes, Stores, Actors, Trust boundaries
- Add data flows
- Add data flows using components
- Delete some diagram elements
- Take some elements in and out of scope

Cheat: download ‘step 2’ from docs.threatdragon.org/downloads/

Practical #3


Edit diagram
Process
Store
Actor
Data Flow
Trust Boundary
Example
STRIDE
CIA
LINDDUN
Client
Response
Request
Web server
write
read
Github repo

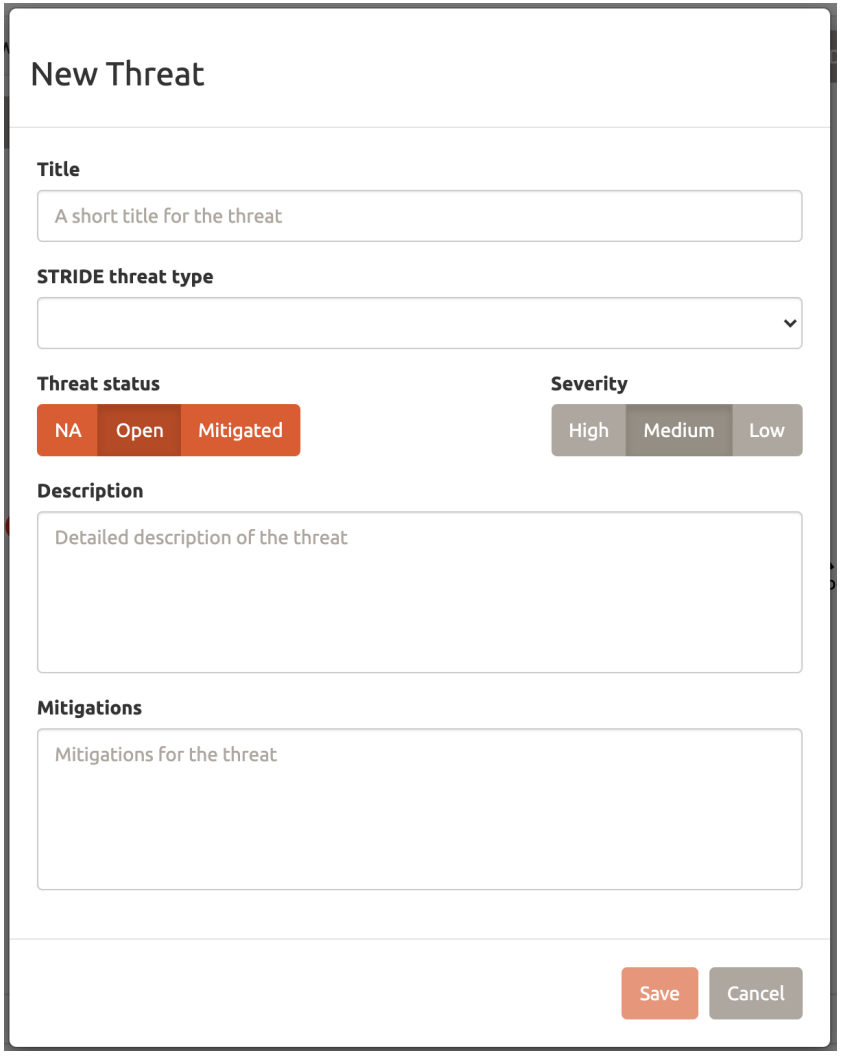
Discussion

- It is not a system diagram
- It is a threat model using a different perspective
- More like requirements “what can go wrong”?
- It comes before design and implementation

Threats

The reason for the threat model

- STRIDE / CIA / LINDDUN
- You can mix and match
- Status: NA / Open / Mitigated
- Priority: Low / Medium / High
- Description of threat
- Mitigation or even prevention



Threat Engine

STRIDE per Element

	Spoofing	Tampering	Repudiation	Information disclosure	Denial of service	Elevation of privileges
Process	X	X	X	X	X	X
Store		X	X	X	X	
Actor	X		X			
Data flow		X		X	X	

Threat Engine

LINDDUN per Element

	Linkability	Identifiability	Non-repudiation	Detectability	Disclosure of information	Unawareness	Non-compliance
Process	X	X	X	X	X		X
Store	X	X	X	X	X		X
Actor	X	X				X	
Data flow	X	X	X	X	X		X

Threat Engine

CIA

- Confidentiality
- Integrity
- Availability

For all elements

Threat Engine

Threats by Context

- Uses the properties of the diagram components
- Very incomplete, area of future work

So far only one threat suggestion:

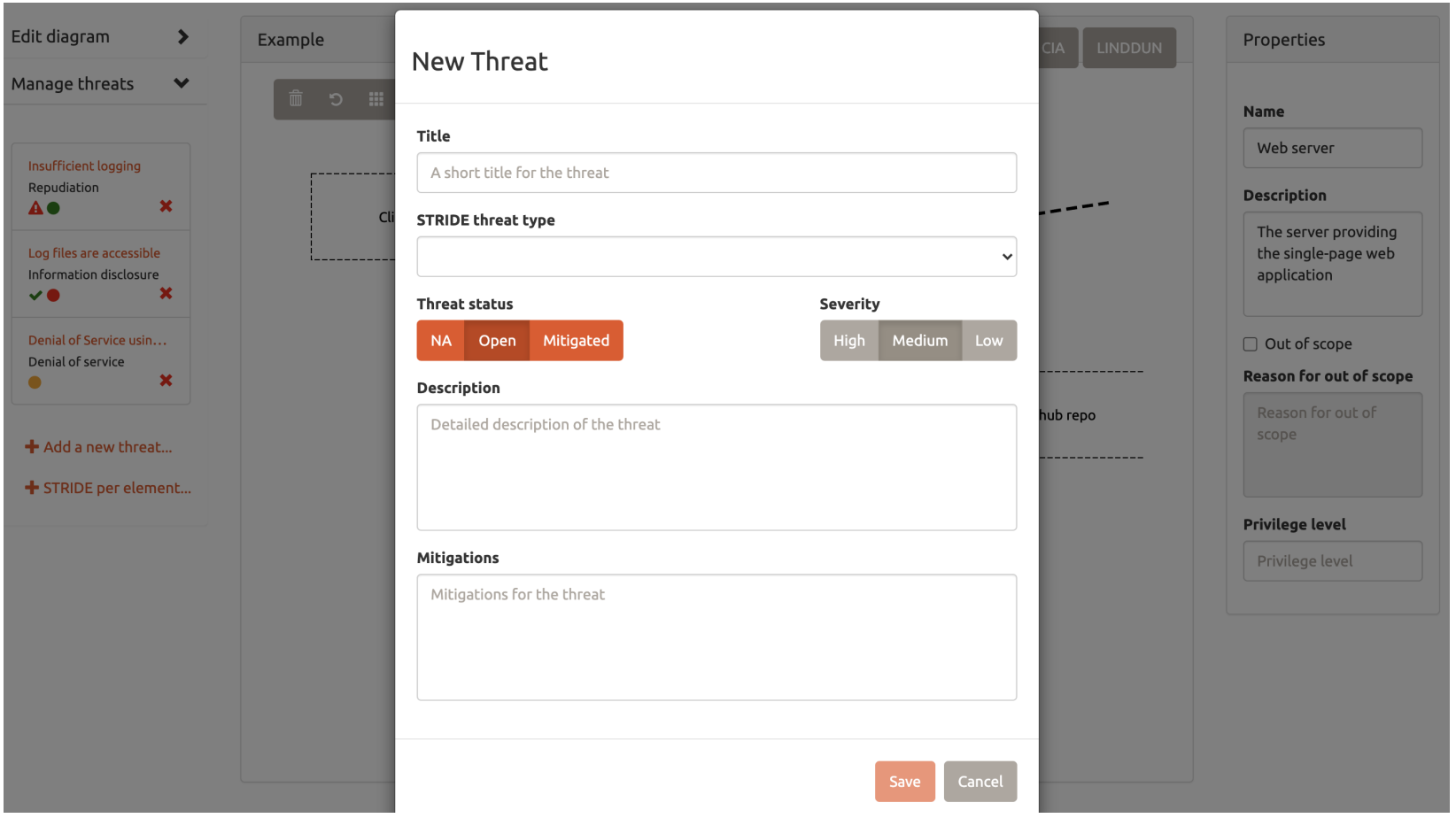
- If public data flow & not encrypted
- Suggest data flow is encrypted

Practical #4

- Add threats to the diagram
- Choose LINDDUN or CIA or STRIDE
- Add a specific threat
- Add threats per element
- Choose a different categorisation, mix and match
- Try the threat by context

Cheat: download ‘step 3’ from docs.threatdragon.org/downloads/

Practical #4


Edit diagram
Manage threats
Insufficient logging
Repudiation
Log files are accessible
Information disclosure
Denial of Service usin…
Denial of service
Add a new threat...
STRIDE per element...
Example
CIA
LINDDUN
hub repo
Properties
Name
Web server
Description
The server providing the single-page web application
Out of scope
Reason for out of scope
Reason for out of scope
Privilege level
Privilege level
New Threat
Title
A short title for the threat
STRIDE threat type
Threat status
NA
Open
Mitigated
Severity
High
Medium
Low
Description
Detailed description of the threat
Mitigations
Mitigations for the threat
Save
Cancel

Discussion

Save it, prove it, update it

- Output as PDF
- Hardcopy output
- Threat model as code

Reporting

Select your threats:

- Show out of scope elements
- Show mitigated threats
- Include threat model diagrams
- Landscape / Portrait (but not yet)

Practical #5

Putting it all together – model Threat Dragon itself

- Client
- Server
- Backend
- Boundary
- Reports

Cheat: download ‘step 4’ from docs.threatdragon.org/downloads/

Discussion

The 4 Questions

- What are we working on?
- **What can go wrong?**
- **What are we going to do about it?**
- Did we do a good job?

In Practice

- Incremental – make it bitesize
- Collaborative involving the whole team
- As valuable as you make it
- Threat Model as code
- Revisit the model
- No Security Heroes

Practical #6

Feature requirements: Cupcake’s Status

- Request: GET threatdragon.org/status
- Response: one of Awesome/Good/Fair/Asleep
- Set status: PUT threatdragon.org/super-secret-api
- Default status: Awesome

Call for Help

- Ask any question on the github project space
- Always looking for suggestions
- Always looking for help as well

Thankyou for joining, any last questions?