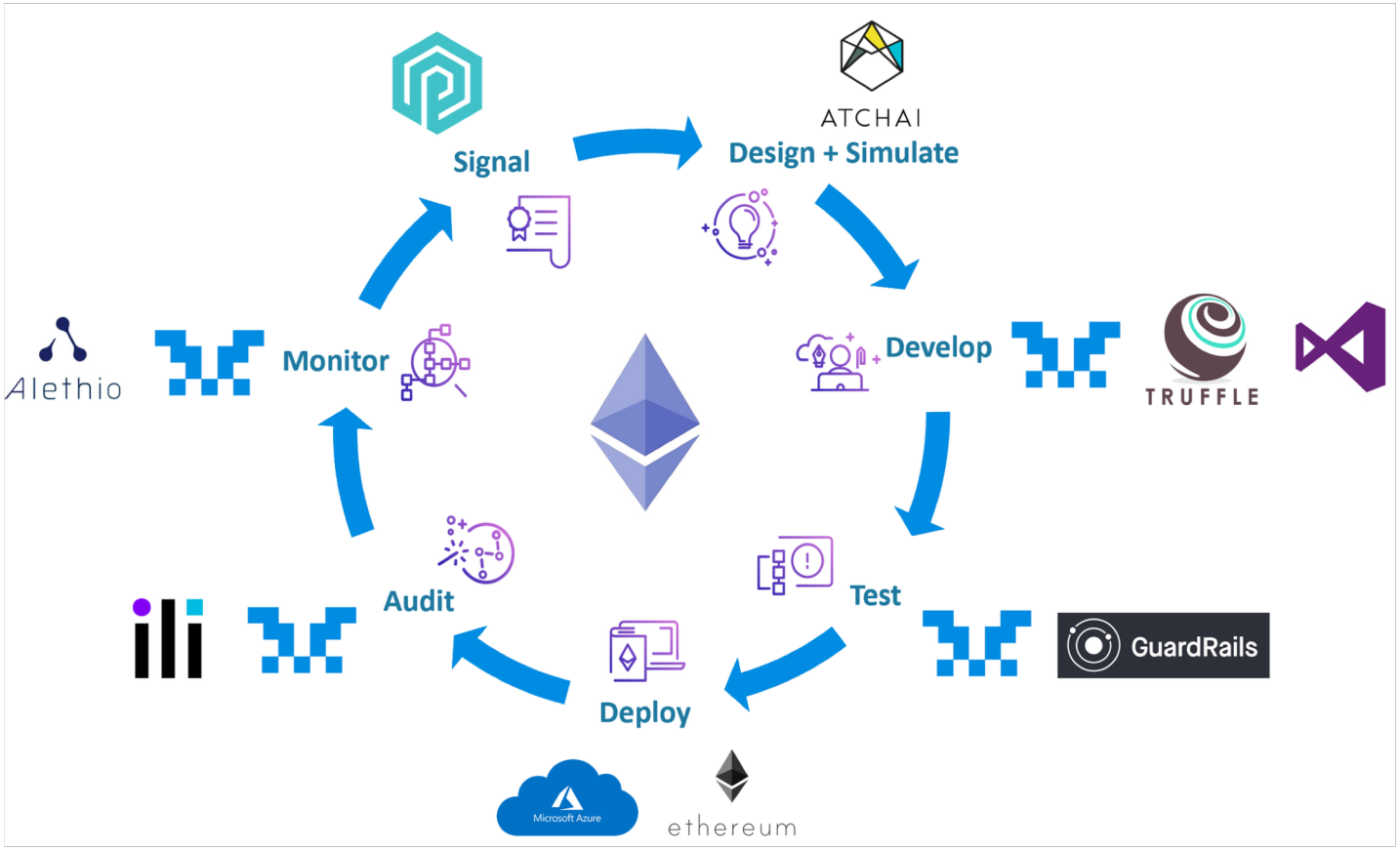
Learn Threat Modeling Smart Contracts in <60 minutes

November 2, 2018

CONSENSYS
Diligence

@g3rh4rdw4rn3r

Security Activities in the SDLC



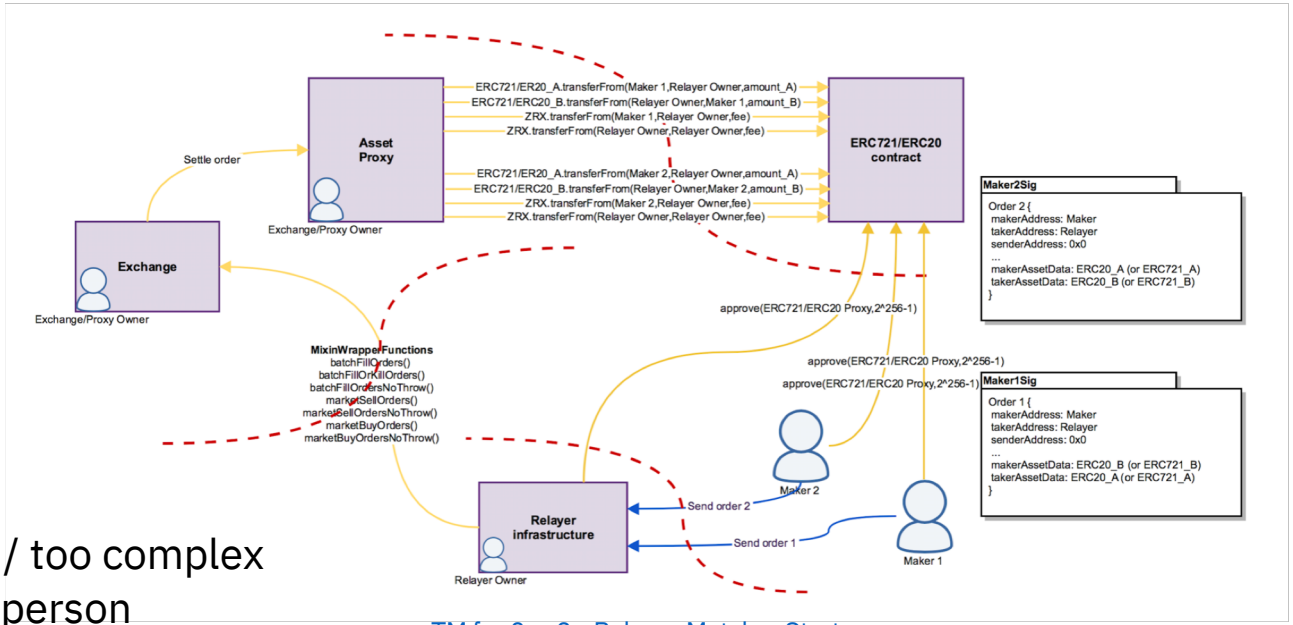
Formal Threat Modeling

Approaches

- Software-Centric
- Asset-Centric
- Attacker-Centric

Challenges

- Time Consuming
- For engineers too abstract / too complex
- Usually done by a security person
- Create report that nobody knows what to with



TM for 0x v2 - Relayer Matcher Strategy

Informal Threat Modeling

Everybody does Threat Modeling in their heads

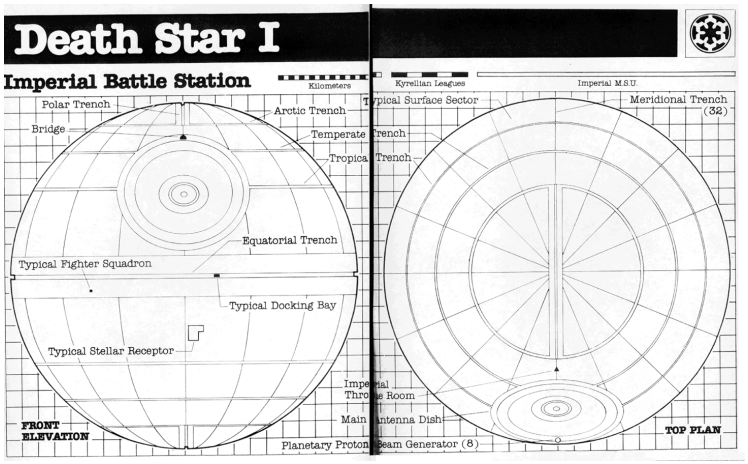
- Spawn an evil twin of yourself
- Get a list of threat scenarios from the evil twin
- Use the list to come up with counter strategies to something about it

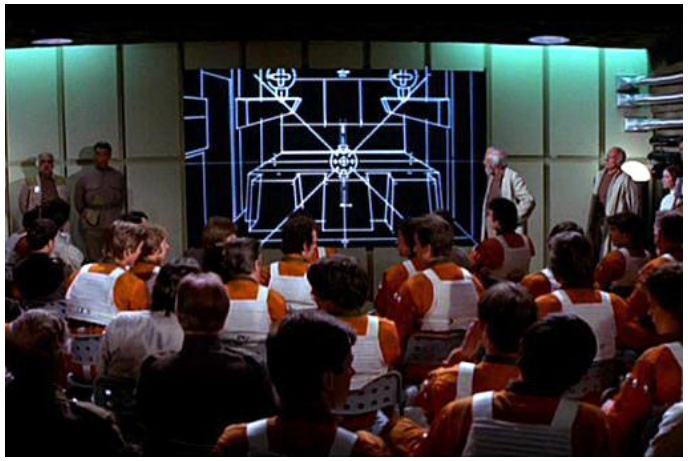
Problem: The strategy is only as good as your evil twin

Threat Modeling is about

Answering the following 3 questions

- What are you building?
- What can go wrong?
- What are you going to do about it?





Finding Relevant Threats

TM Approach needs to:

- Do better than our evil twin
- Safe time
- Not have to learn formal TM methods

Use data from past blockchain security incidents

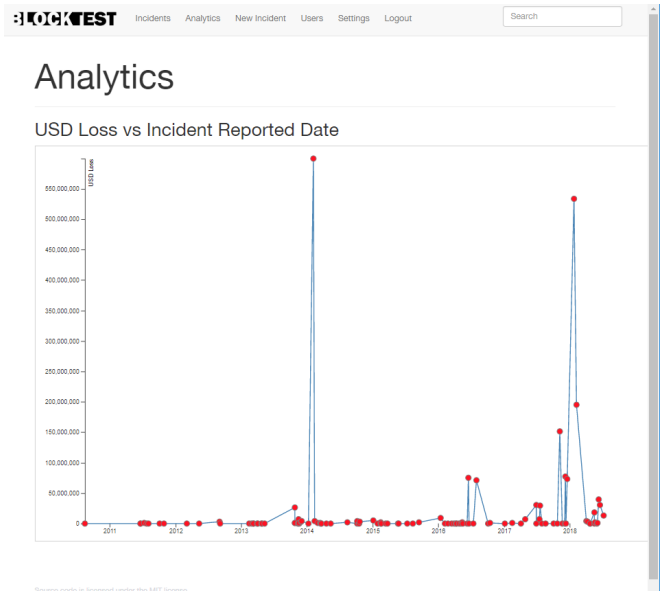
- Extract the most frequent attacks
- Categorize them based on attack patterns
- Use them to verify if they apply to your system

Blockchain Incident Database

- University Initiative
- 110 incidents in the database
- Nature: technical root cause, publicly reported
- Sources: Blockchain Graveyard, Reddit and other public places
- Updating incident db not straight forward
- Incident format STIX

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

BLOCKTEST



Blockchain Incident Categories

1. OPSEC

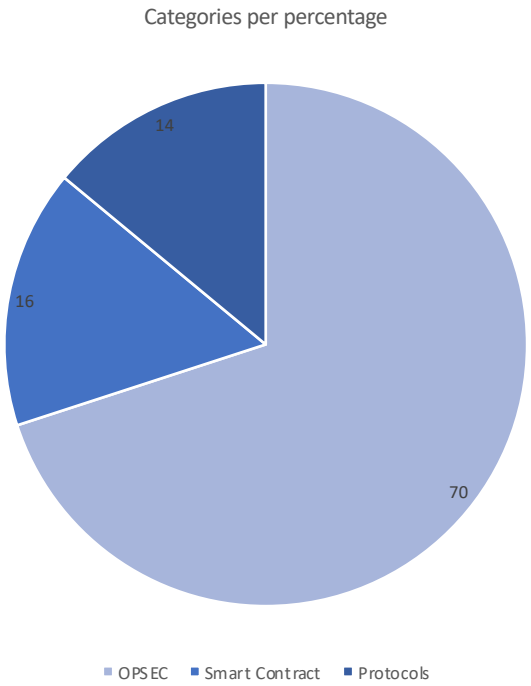
Incidents compromising an organization or individual's control of information and assets

2. Smart Contract

Incidents related to bugs in smart contract systems

3. Protocol

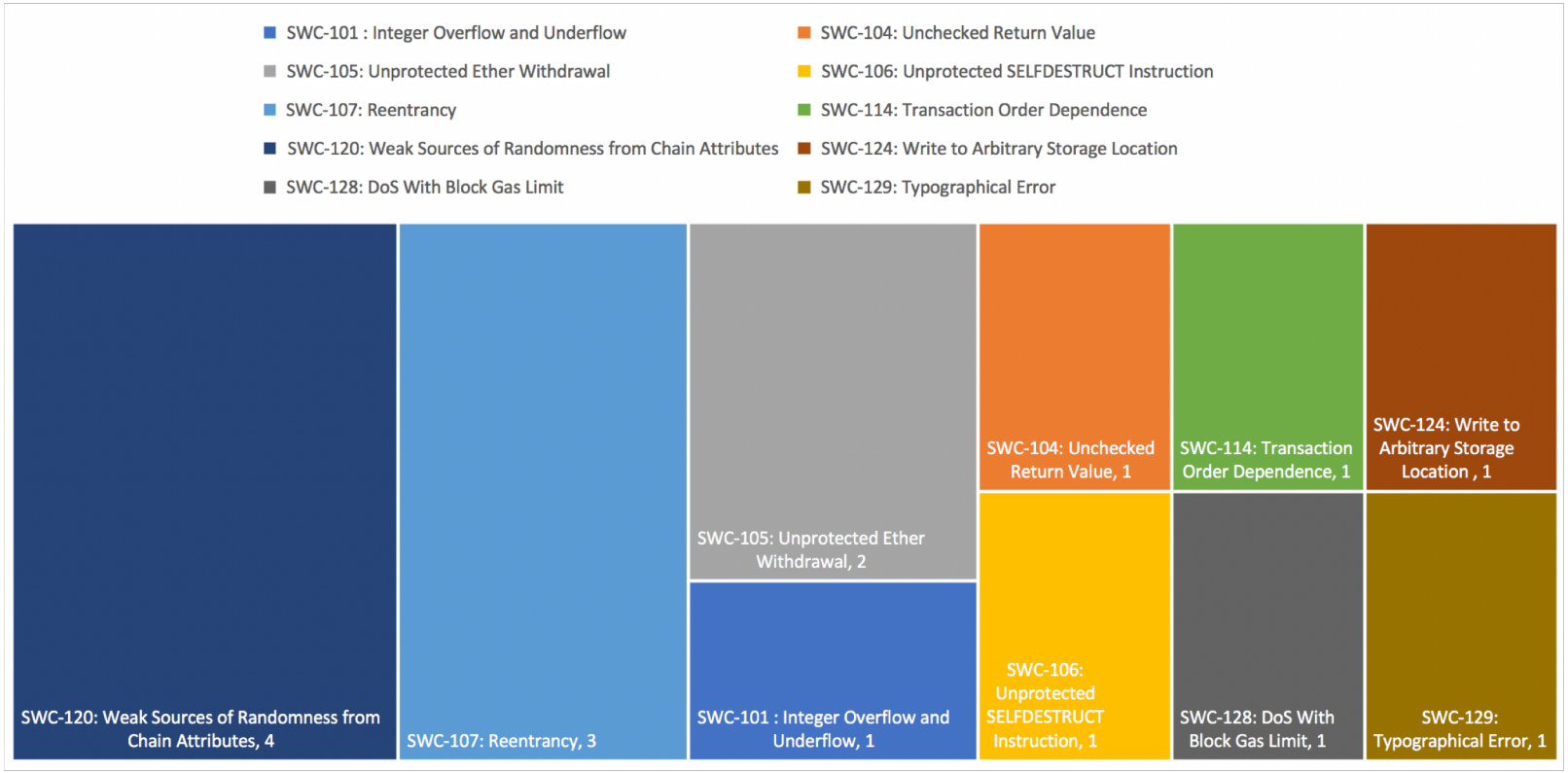
Incidents arising from malicious exploitation of protocol implementations and designs



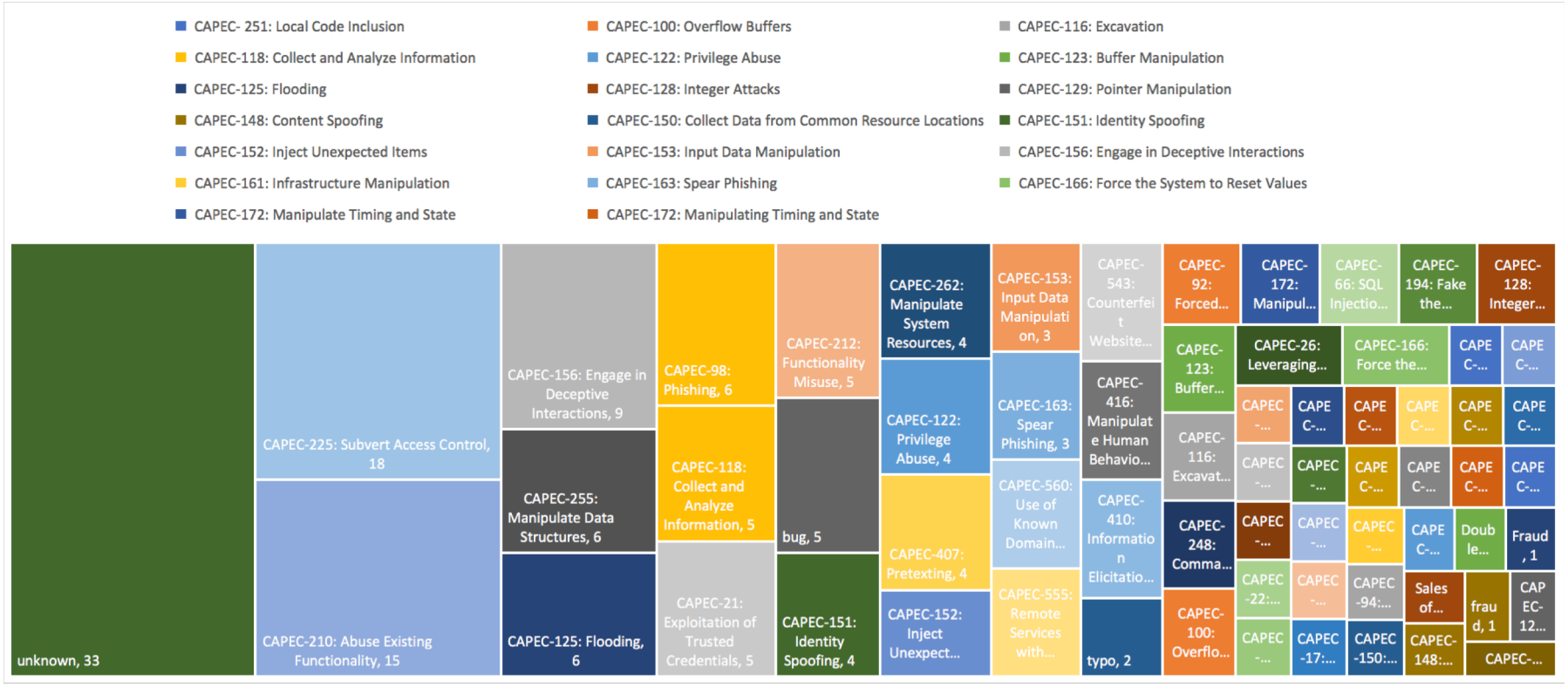
Smart Contract Weakness Classification (SWC)

- Scheme to classify weaknesses in smart contracts proposed in EIP-1470
- Currently in Draft
- Overlays on top of CWE
- Definitions in the SWC-registry (Github/Pages)
- Contains crafted as well as real world vulnerable contracts
- Currently 60 test samples and 30 weakness variants
- Working on Python and JS packages to access data

Smart Contract specific Incidents



Incidents categorized based on CAPEC



Mitigation – Types of strategies

Acceptance: does not reduce effects

e.g. Mitigation outweighs the cost of potential damage
e.g. It’s really unlikely that the threat occurs

Limitation: you have one or multiple mitigation strategies to reduce risk

e.g. Use a MultiSig wallet
e.g. Use hardware wallets to manage private keys
e.g. Conduct security awareness workshops

Transference: hand risk off to a 3rd party

e.g. Get an insurance

Threat Modeling Exercise

Threat Modeling Exercise

Overview

- Extracted the top attack patterns from the incident db
- Find at least 3 threats and mitigation strategies
- Time: 15 minutes

Options

1. Choose your own system
2. Choose the sample system

Material

- Devcon4 Github repo

Threat Modeling Exercise Cheat Sheet 1/2

Take a threat extracted from the [Blockchain Incident table]

Clone the [devcon4 repo]

Is the threat relevant to your system?

If no, move on to the next

If yes, think about mitigation strategies and add them to your threat model

Try to modify the threat example to make it more relevant
Tip: Use your evil twin

Threat Modeling Exercise Cheat Sheet 2/2

Threat - what can go wrong?
A good threat description both describes cause and effect e.g. A hacker successfully infiltrates an employee's computer and retrieves the file containing the private keys for an account.

Mitigation - you know already what can go wrong so what are you doing about it?
E.g. Employees use hardware wallets to make transactions and they always double check public address on the hardware device before they perform a transaction.

CAPEC / Attack Pattern – how adversary
Helps users to understand how adversaries exploit weaknesses by describing similar ways of attacking that frequently occur across a wide range of systems and technologies.

Threat Modeling Exercise Cheatsheet 2/2

Threat - what can go wrong?
A good threat description both describes cause and effect e.g. A hacker successfully infiltrates an employee's computer and retrieves the file containing the private keys for an account.

Mitigation - you know already what can go wrong so what are you doing about it?
E.g. Employees use hardware wallets to make transactions and they always double check public address on the hardware device before they perform a transaction.

CAPEC / Attack Pattern – how adversary
Helps users to understand how adversaries exploit weaknesses by describing similar ways of attacking that frequently occur across a wide range of systems and technologies.

Go, go, go
You have 15 minutes

Wrap Up

Takeaways

- OPSEC is not rocket science
- People don’t pay enough attention to OPSEC
- Centralization in systems heightens the need for good OPSEC

Where to go from here

- Open it up to the team
- Keep your threat model up to date
- Track mitigation efforts in defect tracker

Questions

CONSENSYS
Diligence