Threat Modeling Fun

With Cornucopia (Speed Mode)

Notes

Goal:

Taste of threat modeling

Taste of how to do threat modeling easily with cornucopia

Learn and have fun

Whoami

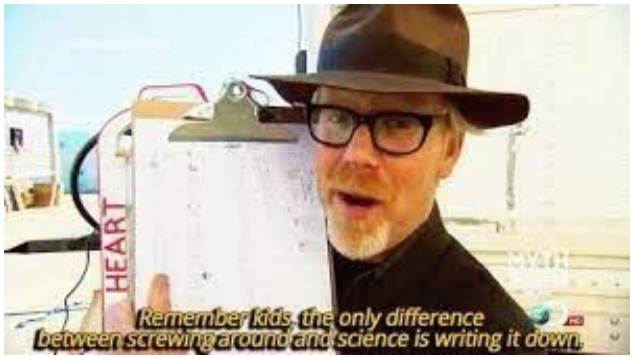
- Spyros Gasteratos
- Security Engineer @  OCURITY
- Open Source Developer (dracon.io, OpenCRE.org)

Itinerary

- Threat Modeling basics
- Cornucopia crash course
- Play 2 rounds of Cornucopia in 20 Minutes

Threat Modeling Basics!

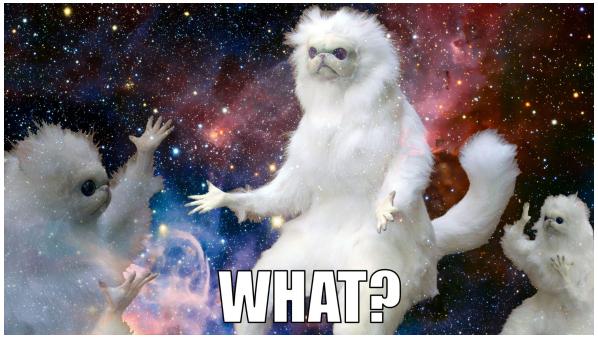
A threat model is a **structured representation** of all the **information** that **affects the security of an application**. In essence, it is a view of the application and its environment through the lens of security.



Threat Modeling Basics!

Write down everything related to the security aspects of a system.

- Development
- Operational
- Failure Mode



Threat Modeling Basics!

- Many approaches
  - STRIDE
  - PASTA
  - DREAD
  - LINDDUN
  - Attack Trees
  - ...
- Many Tools
  - Games (EoP, Cornucopia)
  - Threat Modeling as Code (PyTM, Threagile, HclTM)
  - Threat Modeling as Diagrams/Graphs (ThreatDragon)
  - ...

The 4 Questions Threat Modeling Framework

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

What are we working on?

- Diagram describing a dummy anti-Fraud application on your tables
- Suggested Trust boundaries drawn
- Data Flows labeled
- If a security control is not described, assume missing

Sequence Diagram

Data Flow Diagram

What are we working on? (10’ version)

- Diagram describing a dummy anti-Fraud application on your tables
- Suggested Trust boundaries drawn
- Data Flows labeled
- If a security control is not described, assume missing

Sequence Diagram

Data Flow Diagram

What are we working on?

F-Corp just finished coding their brand new multi tenant "Anti-Fraud 2.0" application to be used by their customers in the Fintech space.

Customers and internal policy require a Threat Model before the application is allowed into production.

F-Corp arranged a meeting of your threat modeling team with the lead developer of the application so you can ask them questions and they can answer them. Your job is to document as many security considerations(good and bad) around the project as possible and maybe find a few high-impact issues along the way.

Due to the criticality of the application, deployments are done personally by the lead developer.

To deploy a new instance the developer personally builds a container locally and pushes it to the remote registry.

What can go wrong? (How to play Speed-Cornucopia)

- Each player gets the **4** top **cards** from the deck
- On every round **play** the card that is most likely to be the **most serious vulnerability**
- **Explain** to your table **WHY** this is **important** and needs fixing, you need to convince them.
- At the end of the round the person with most fixes gets 1 point

- We have **20 minutes for two rounds!**

Recap

- Top 4 cards each
- Play the **highest vulnerability**
- Convince your table that you should fix as a priority
- Vote for highest impact : Most votes +1 points

Sequence Diagram

Data Flow Diagram

What are we going to do about it?

- In this scenario: nothing
- In reality, you would create tickets to fix the issues

Did we do a good job?

- Questions?