Threat Modeling

Field guide to staying ahead of the bad guys

whoami

Application Security Architect at Genetec

10+ years in tech (QA, Dev, Security)

OSCP, OSWE

Web, cloud and offensive security

Love to make/break things, and teach people how to do it

https://www.linkedin.com/in/samuel-dussault-797489126/

5 simple questions

What

Who

Why

How

When

What

01

AKA the part where we define threat modeling

Threat Modeling Manifesto

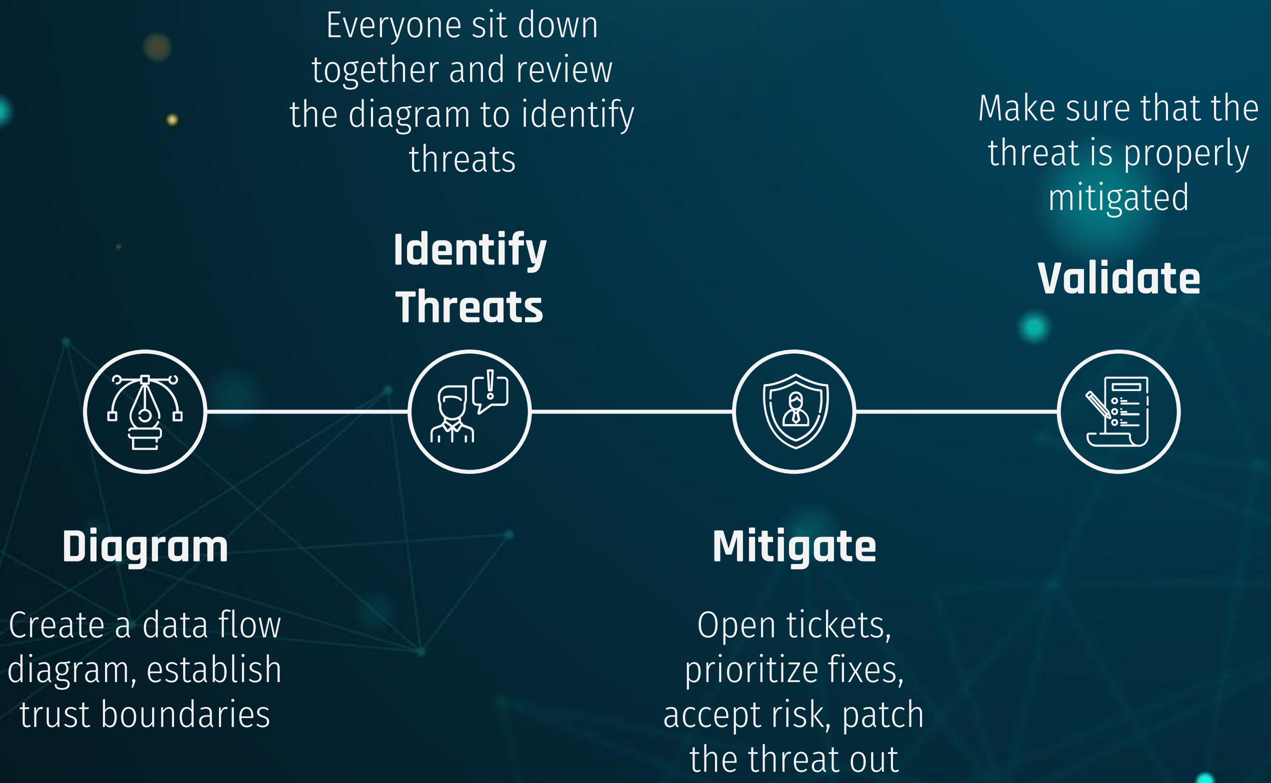
- Defines key values and principles
- Working group formed by leading experts in the field (Adam Shostack, Chris Romeo, Alyssa Miller)

- https://www.threatmodelingmanifesto.org/

Threat Modeling

Repeatable process to **establish security requirements**, **identify and quantify threat and vulnerabilities**, and **determine and prioritize remediation methods**

Stages



Diagram

Create a data flow diagram, establish trust boundaries

Identify Threats

Everyone sit down together and review the diagram to identify threats

Mitigate

Open tickets, prioritize fixes, accept risk, patch the threat out

Validate

Make sure that the threat is properly mitigated

Handling Threats

Mitigate (patch the feature)

Eliminate (remove the feature)

Transfer (firewall, WAF, to the user...)

Accept

What Threat Modeling is

A culture of finding and fixing security issues early and often

A chance for people with different skillsets and backgrounds to contribute to the security of a product

A great opportunity for developers, QA and security to sit down together and have a discussion

A systematic yet creative approach to shifting security left and identifying risk and issues

What threat modeling isn't

A single-person job or responsibility; the security team cannot, and should not, be the sole owner of the threat models

An absolute science; find what works best for you and your team

A formal assessment of the security of a product, a threat model is not a penetration test

Meant to be used as a way to judge or criticize anyone's work; the focus should always be on improving the product

Who

02

AKA the part where we define everyone's role

Everyone

Security is everyone's job

The worst mistake you can make when threat modeling is to delegate everything to a single person or team

Everyone in the company has a different skillset, knowledge of the product(s), and interests

Devs

Developers have a great understanding of the **technical implementation** and **requirements** of the product

They should generate the **data flow diagram, identify potential weaknesses** and ensure that the issues identified are **properly mitigated**

They should also evaluate when a **new threat model** should be produced, or an **existing one should be updated**

QA

Testers and QA engineers usually spend the **most time with the product, break it often** and are familiar with its **quirks and edge cases**

They should be involved in the process of **identifying threats** and **validating mitigation controls/fixes**

Security

Security folks bring the **attacker mindset** and the **compliance aspect** to the table

They should help **review the data flow diagram, identify threats** and **design mitigation controls**

Stakeholders

This includes **management, product owners, marketing, customer support specialists**

Stakeholders typically have a **more user-focused** and **less technical vision of the product**

This can be leveraged to help **identify additional threats** and **use cases**

Why

03

AKA “I already have a firewall and an anti-virus, so what’s the point in doing all of that?”

79% of all businesses push vulnerable code to their production environment

The most common reasons for doing so are developers trying to meet a tight deadline, the vulnerabilities not being considered as being severe enough or simply being found too late in the development process to properly remediate them

Source: https://www.csoonline.com/article/3571268/the-state-of-application-security-what-the-statistics-tell-us.html

Incidents in 2021

Average cost of a data breach in 2021 according to IBM

6.5 Million $ USD

30,000

Websites hacked daily according to techjury

Of companies worldwide have experienced at least one major attack

64%

Devs

Often struggle to meet very tight deadlines and have little to no time to invest in improving the security of their products

Already have to spend a lot of time on staying up to date with the technology and frameworks used to build their products and often cannot realistically stay up to date with modern vulnerabilities and attacks

Security

Often represent a very small fraction of the entire development/engineering team (typically 1-5%) and struggle to scale up their efforts

Cannot realistically be familiar with the technical implementation and latest features of each product (internal and third party)

Do not have the authority to block a release or enforce the proper usage of the tooling and processes that they implement

Stakeholders

Do not always have all the information required to properly identify and quantify the risk related to a given threat

Tend to have a fundamental misunderstanding of the value that the cyber security team can add to their products and see it as a blocker

Design Flaw | Code Issue

Compliance should be a byproduct of security

- Jean-Francois Beauchemin

How 04

AKA how to get started and get the ball rolling

Diagram

Microsoft Threat Modeling tool (TM7)

OWASP Threat Dragon

PYTM (Threat modeling as code)

Bring your own tool!

Identify Threats

STRIDE

Owasp ASVS or Security Testing Guides

NIST

Frameworks or Standard (GDPR, ISO…)

Existing security requirements, policies and tests/checklists

STRIDE

S = Spoofing

T = Tampering

R = Repudiation

I = Information Disclosure

D = Denial Of Service

E = Elevation of privileges

Tips

Break more complex systems down into smaller pieces

Make separate threat models to illustrate specific use cases and workflows

No threat model is ever perfect

There is no such thing as a stupid question; Never assume that everything will only ever go right

Mitigate

Open and assign tickets (Jira, Helpdesk, TFS...)

Prioritize based on company/security policies and estimated risk (CVSS)

Ideally, make sure to store/track the issues alongside the diagram for reference

Security should be involved as necessary (suggest fixes and help identify/design mitigation controls and methods)

Validate

As development teams gain more maturity, it should become easier for the to validate security fixes with confidence

QA should also help validating fixes through manual and automated testing if possible (security test cases)

When

05

AKA “just do it”

Measure twice, cut once

Threat Model

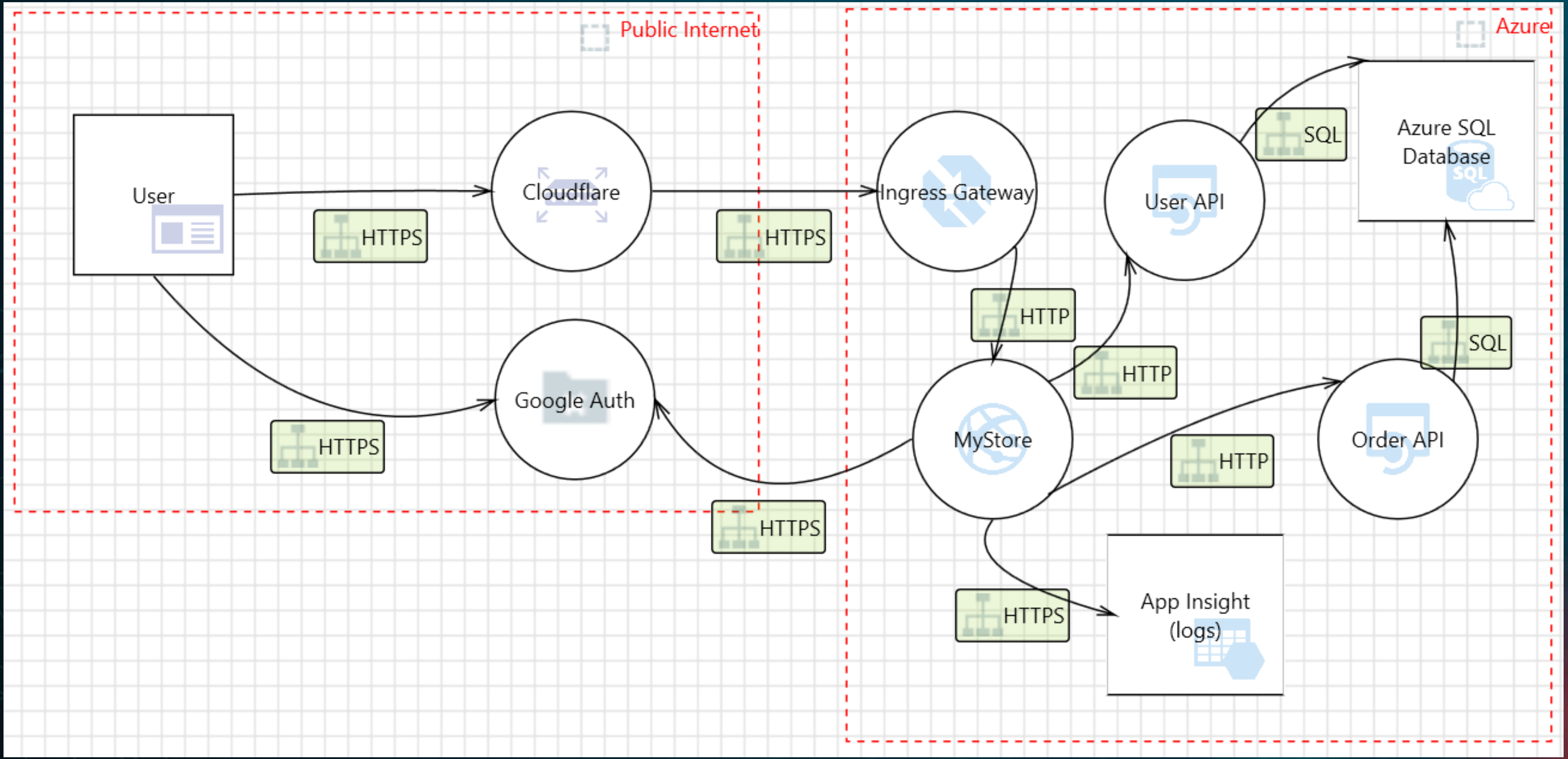
When starting a new project, as soon as the high-level architecture is finished

When designing a new feature, if it is security-relevant

When there is significant change in scope or thrust boundaries

NOW, hopefully

MyStore


Public Internet
Azure
User
HTTPS
Cloudflare
HTTPS
Ingress Gateway
HTTP
User API
SQL
Azure SQL Database
Google Auth
HTTPS
MyStore
HTTP
HTTP
Order API
SQL
HTTPS
HTTPS
App Insight (logs)

Key Takeaways

01 Threat modeling is an awesome culture that will save you time and let you solve problems before they happen

02 Shifting security left is the only way we will ever be able to face today’s cyber security challenges; security is everyone’s job!

03 Start small and work your way up from there, analysis paralysis is real and some threat models > no threat model. Threat model early and often.

04 Your threat models are only ever as good as the actions taken as the result of the risks and vulnerabilities identified

Questions?

Thank You!