Intro to Threat Modelling

whoami

Matt Belvedere
Penetration tester for N years
Hacker for N+M years

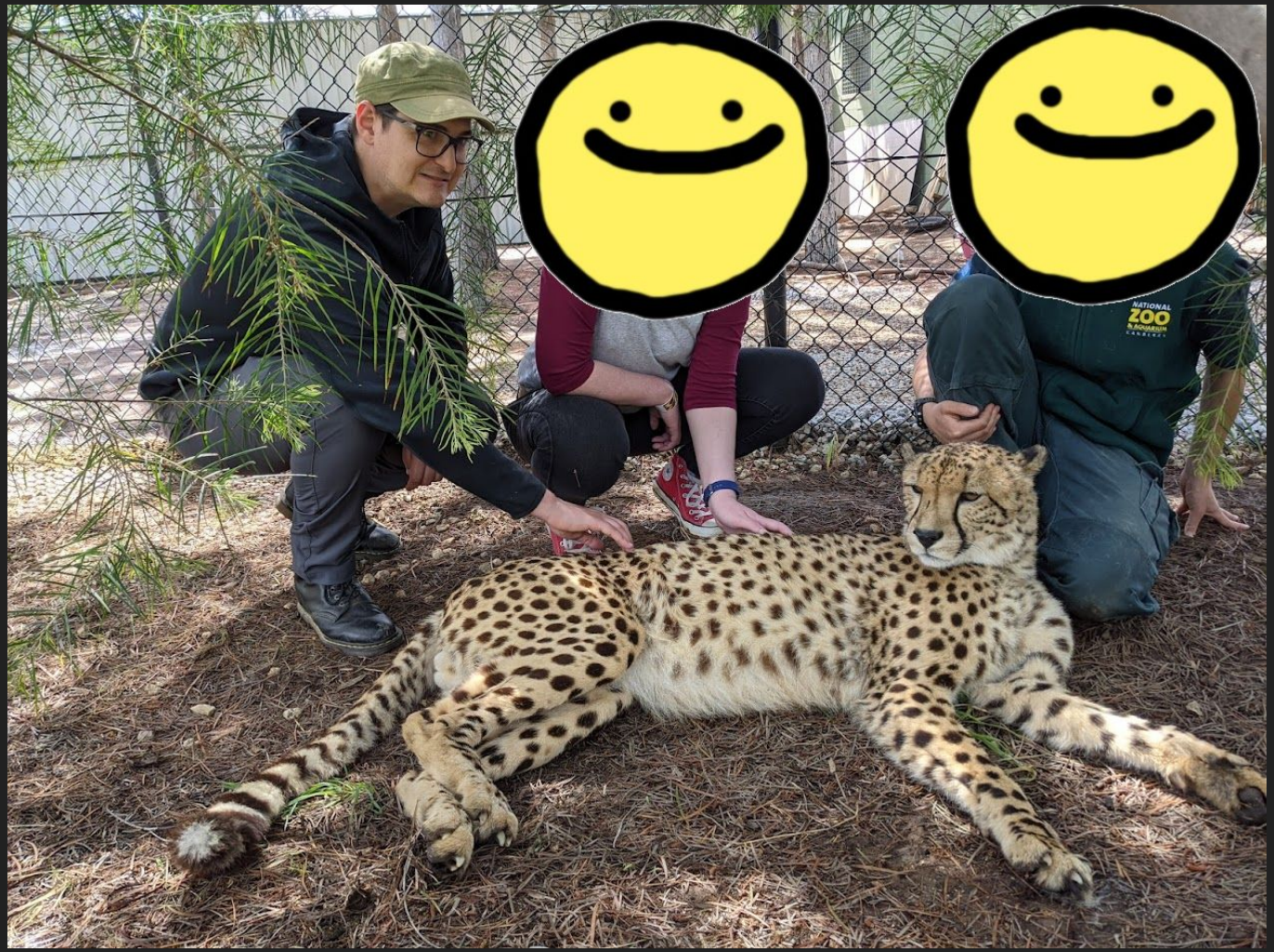
Worked at **$BIG_PENTEST_SHOP**

Worked at
$SHOP_THAT_SPONSORS_SPORTS

Now at my own shop

https://proactivelabs.com.au

Before we begin
A simple example


NATIONAL
ZOO
& AQUARIUM

Anyway

What this is

- An introduction to threat modelling
- What is a threat model
- How to do threat modelling
  - Assets
  - Threats
  - Likelihoods
  - (Mitigations?)
  - (Actors?)
- Applying to offensive contexts

What this is not

- A deep dive into the N different formal methods of threat modelling

What this is not

- If you know what PASTA / STRIDE / DREAD / PNG is, this talk is (probably) not for you
- Formal mechanisms have their strengths* and weaknesses*
- I’ll mention them as appropriate, but it’s not the focus
- This is aimed for an introduction
- I have 30 minutes

Threat Modelling

Threat modelling?

“Threat modeling is a process by which potential threats, such as structural vulnerabilities or the absence of appropriate safeguards, can be identified and enumerated, and countermeasures prioritized”

en.wikipedia.org/wiki/Threat_model
(sort of STRIDE-y)

Too wordy

Threat Modelling

- Think of your system
- Think of bad things that could happen
- How do we make those things not happen
- Order likelihood / prioritise
- Figure out what to spend time fixing
- Bonus: Figure out if the fixes / mitigations are effective
- Bonus: Think of the people/things involved that might do bad things

Threat Modelling (For offensive purposes)

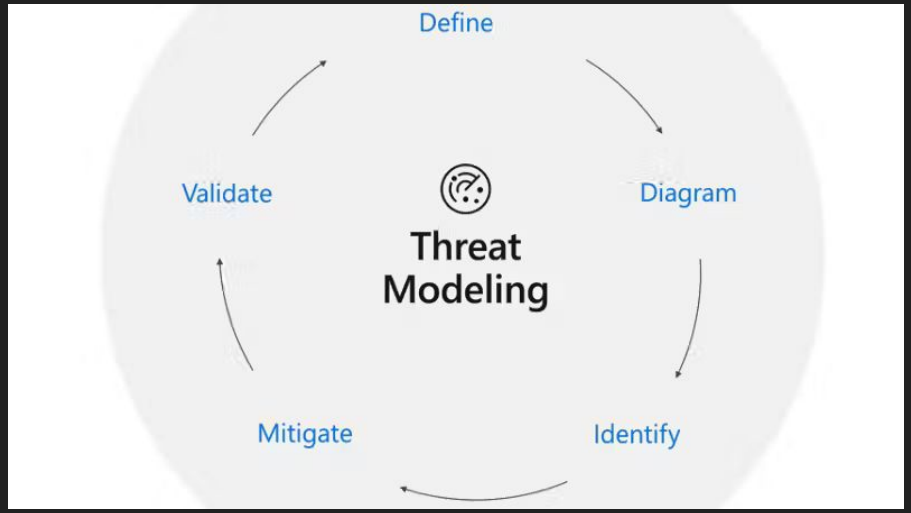
- Think of your system
- Think of bad things that could happen
- ~~How do we make those things not happen~~
- ~~Order likelihood / figure out what to spend time fixing~~
- Be the bad guy
- How do we **make** these bad things happen?
- Which bad things do we prioritise?

https://news.sky.com/story/emotet-police-raids-take-down-botnet-that-hacked-millions-of-computers-worldwide-12200460

Why

Why Threat Modelling (Defensive)

- Defensive:
  - Design* teams - Define security requirements ahead
  - Engineering* teams - “Secure by design” / part of Security Development Lifecycle
  - Pentesting / Red Teaming - Think of the likely threats of a system, then emulate them



https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling

Why Threat Modelling (Offensive)

- Offensive:
  - Ransomware actor - figure out if your tooling will get you raided
  - Figure out if your tradecraft is going to flag on every MDR/XDR/$DETECTION_SYSTEM
  - Figure out what tech / processes your surprise clients have or might have
  - Good red teams / pentest / threat emulation people are putting themselves into these shoes

https://therecord.media/alphv-black-cat-ransomware-takedown-fbi

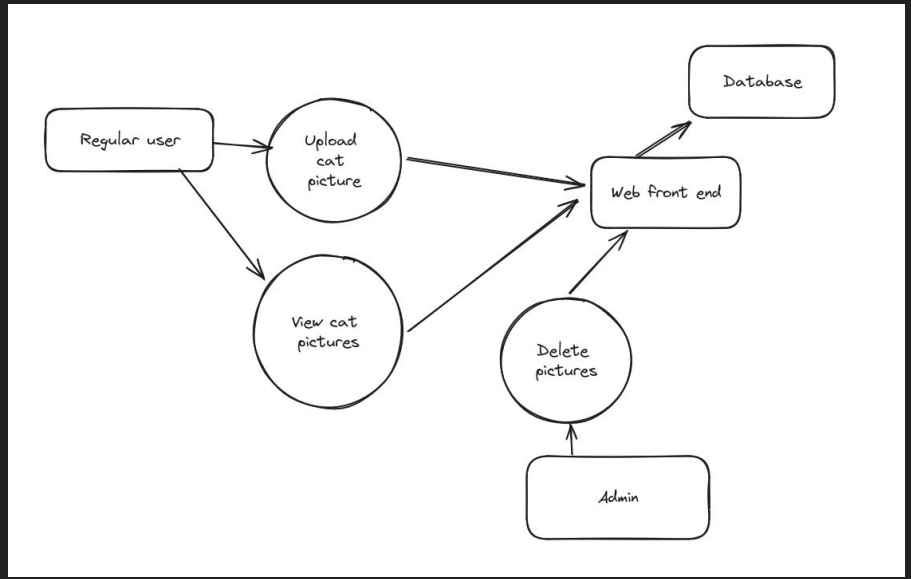
Your system

What is your system?

- I.e. What are you trying to protect?
- What are its crown jewels?
- What tech stack does it use?
- Where does it sit / How is it hosted?
- Who uses it?
- What happens if it doesn’t work properly?
- What happens if it doesn't work at all?

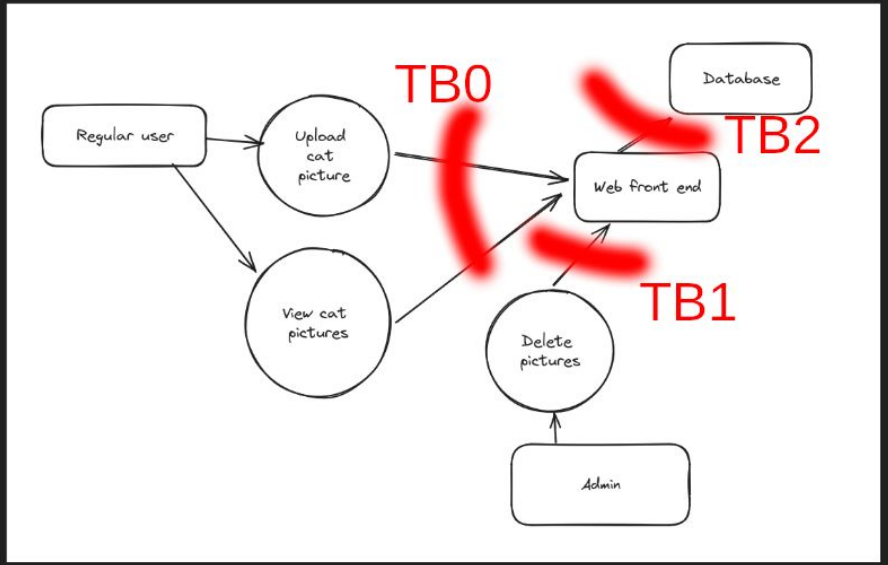
Entry / Exit / Data Flow

- What are the systems entry points?
- What are the systems exit points?
- How does data flow through the system?
- Data Flow Diagrams - Even napkin based ones, are very useful here!



Entry / Exit / Data Flow (Trust Boundaries)

- Think of boundaries of trust (i.e. trust boundaries)
- Consider:
  - Explicit or implicit
  - Enforceable / unenforceable



System Behaviours / Intended states

- Have you considered using the behaviours into unexpected ways?
- What are the intended states?
  - What is the intended finite state machine?
- Does the system have dangerous or “interesting” functions
- Think Weird States / “What is exploitation”
  - Weird machines, exploitability and unexploitability

http://www.dullien.net/thomas/weird-machines-exploitability.pdf

Threats

What is a threat?

- I.e. Bad things that could happen
- Formal models would say bad things in the following categories:
  - Spoofing
  - Tampering
  - Repudiation
  - Information Disclosure
  - Denial of Service
  - Elevation of Privilege
- (STRIDE)

https://en.wikipedia.org/wiki/STRIDE_(security)

Example threats

- Web Application
  - Cross Site Scripting (Stored / Reflected)
  - Deserialisation issues
  - Server Side Request Forgery
- Generic system issues
  - Authorisation bypasses
  - Authentication issues
  - Broken Logging (repudiation issues)
  - Compromised upstream compression library used on all of your servers

https://www.openwall.com/lists/oss-security/2024/03/29/4

Other bits

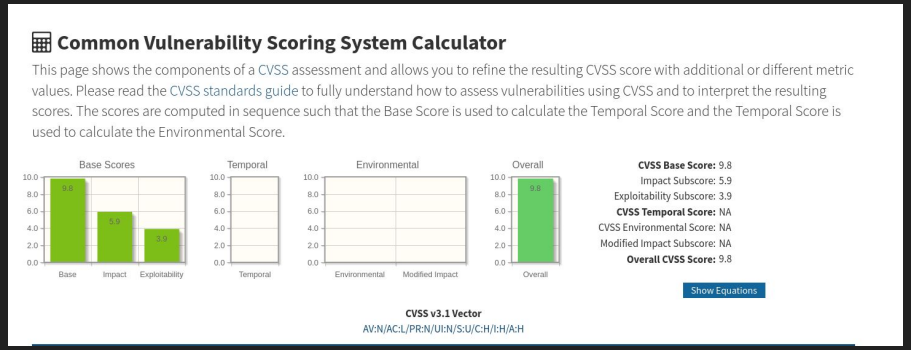
Other things to consider

- People / Actors / Groups that might want to break your stuff
  - Have you considered the types of groups your system might attract?
  - Have you considered they may operate with very different parameters?
    - I.e. impunity, state sponsored activities, or just live in a country without an extradition policy
- Prior work against similar systems
  - Any issues they typically have?
  - I.e. systemic XSS in every app framework used?
- Any prior work for your organisation?
  - Any systemic processes / issues / noteworthy things to consider?
  - This can also include weird operating requirements!

Sidenote - Prioritisation

Prioritising / Categorising issues

- You've likely already been doing this, or at least exposed to this
- Think CVSS
  - A score that considers impact to confidentiality / integrity / availability
  - Pros* and Cons exist
- Others may do a N x M risk* matrix



https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

Scenarios

QR Codes

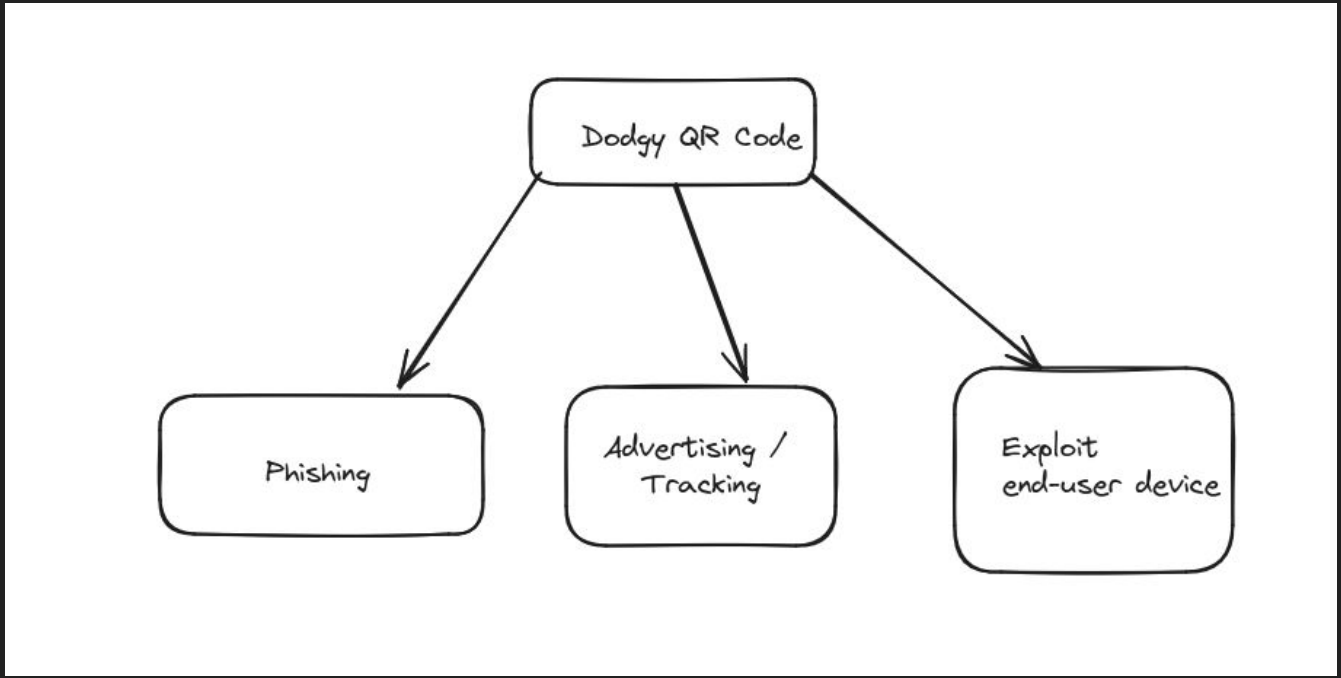
Quick-Response codes

Example threat model - QR codes

- What is it?
  - QR Codes
- What does it do?
  - Allows people to not have to type out long URLs to browse resources
- What can go wrong?
  - Sending you to a malicious or obnoxious website
  - Shell your box
- How can it do that?
  - Link to a phishing site
  - 0day in QR code parser

Is this QR code malicious?

Attack tree

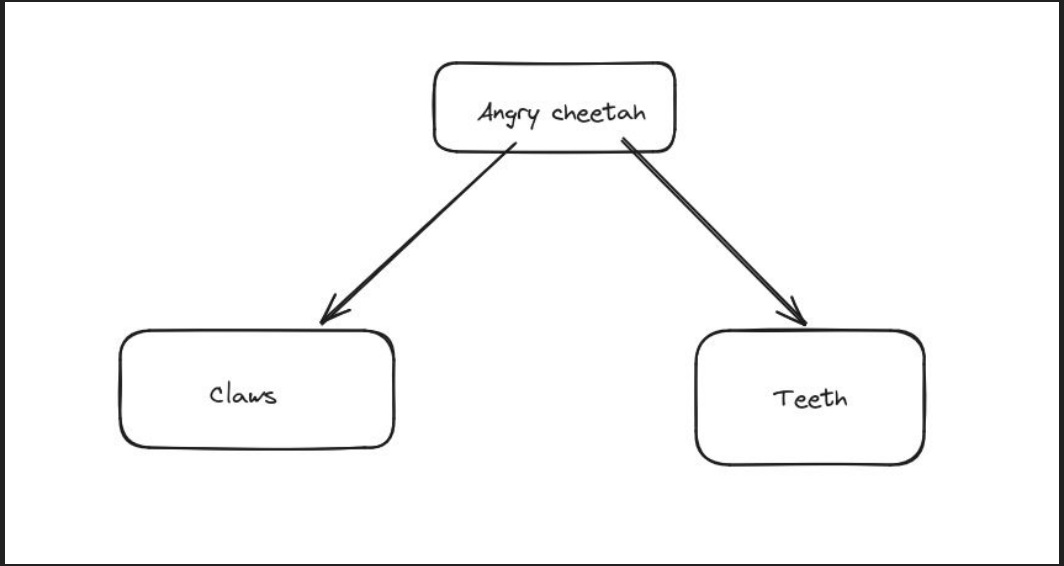

Dodgy QR Code
Phishing
Advertising / Tracking
Exploit end-user device

Big Cat Experience

Big Cat Experience(™)

- What is it?
  - Visit a very large wild animal at a local Zoo
- What does it do?
  - The Cat - Basically sits there (and hopefully doesn’t eat you)
  - You - Hang out with a very large cat
  - zookeeper - Take some photos of you and the cat
  - You - Pat the cat
- What can go wrong?
  - It decides to ruin your day
- How can it do that?
  - Big claws
  - Big Teeth

Attack tree



Example threat model - Cheetahs v human in the wild

- You should be considering threat intelligence or other data to help shape threats
- However; apply scrutiny to the data
- Bad threat intel / sample data / logs / sampling **will** misdirect your efforts significantly

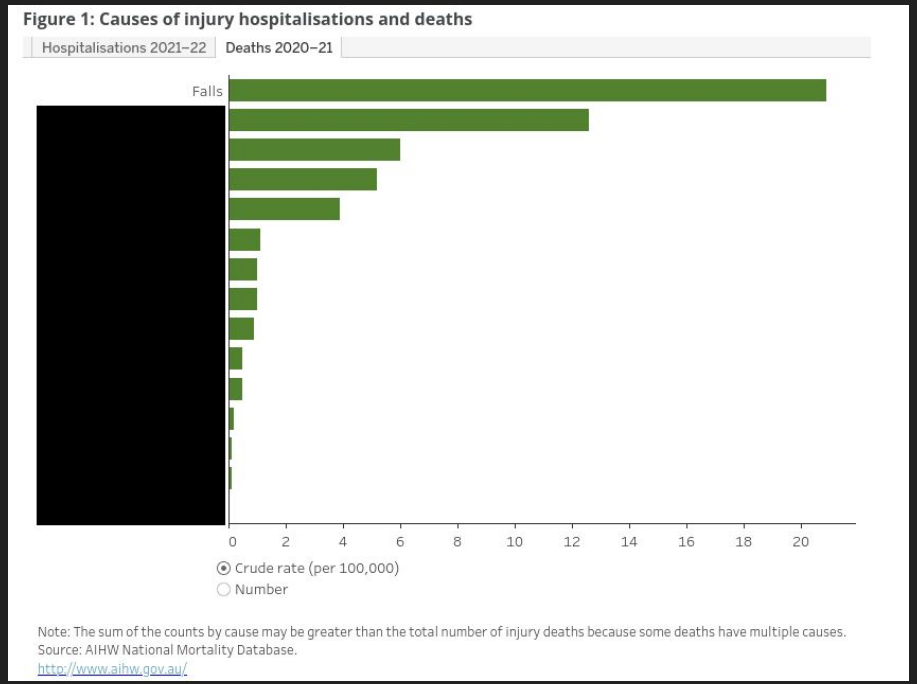
Overlooking things

Did you consider other problems?

- Other People - Zookeeper(s)
  - Has your unlocked phone, copies your photos off, installs malware, etc
  - Feeds you to the cheetah and makes it look like an accident
- The cheetahs friend (the dog)
  - 33 death-by-dog events in Australia since 1979
- Falling over in the dirt and being very injured
  - (Turns out this is way more likely to happen than the first two)

Sidenote - Falling over

No really - Falling over is fairly dangerous



https://www.aihw.gov.au/reports/injury/injury-in-australia/contents/introduction

Falling over threat intel

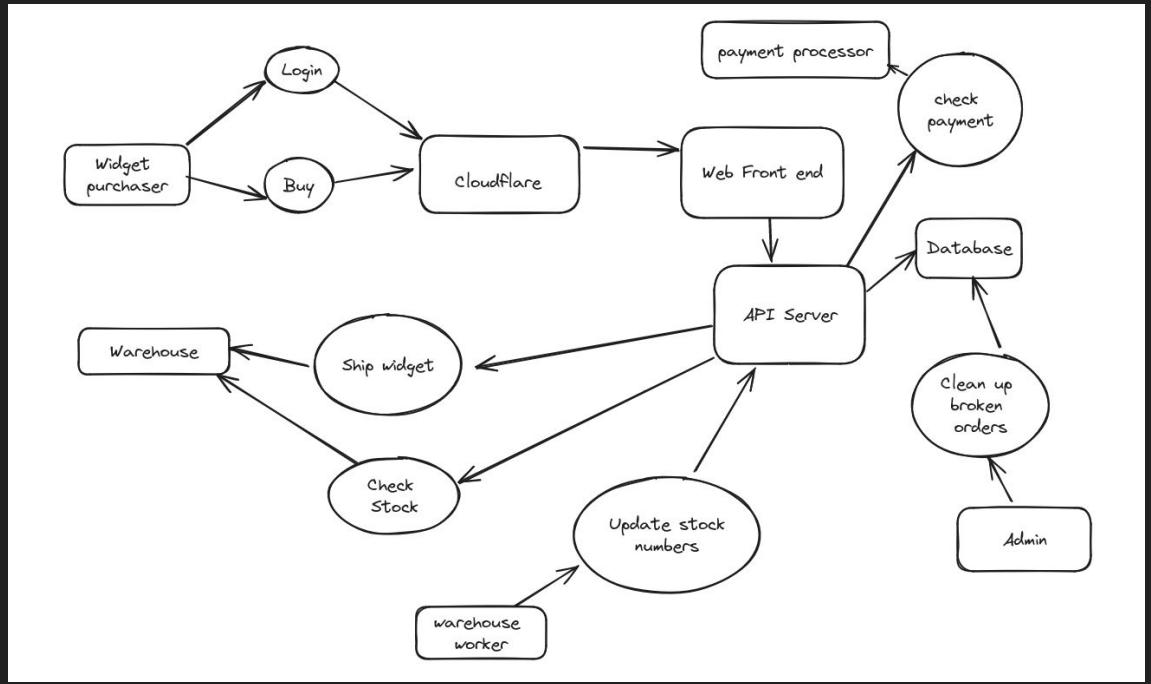
When modelling:

- Think of what attackers are *likely* to do
- This is intended to help prioritise what attacks to perform (or mitigate)
- But keep in mind that picking the wrong events means picking the wrong attacks (or mitigations)

Applying it to computers

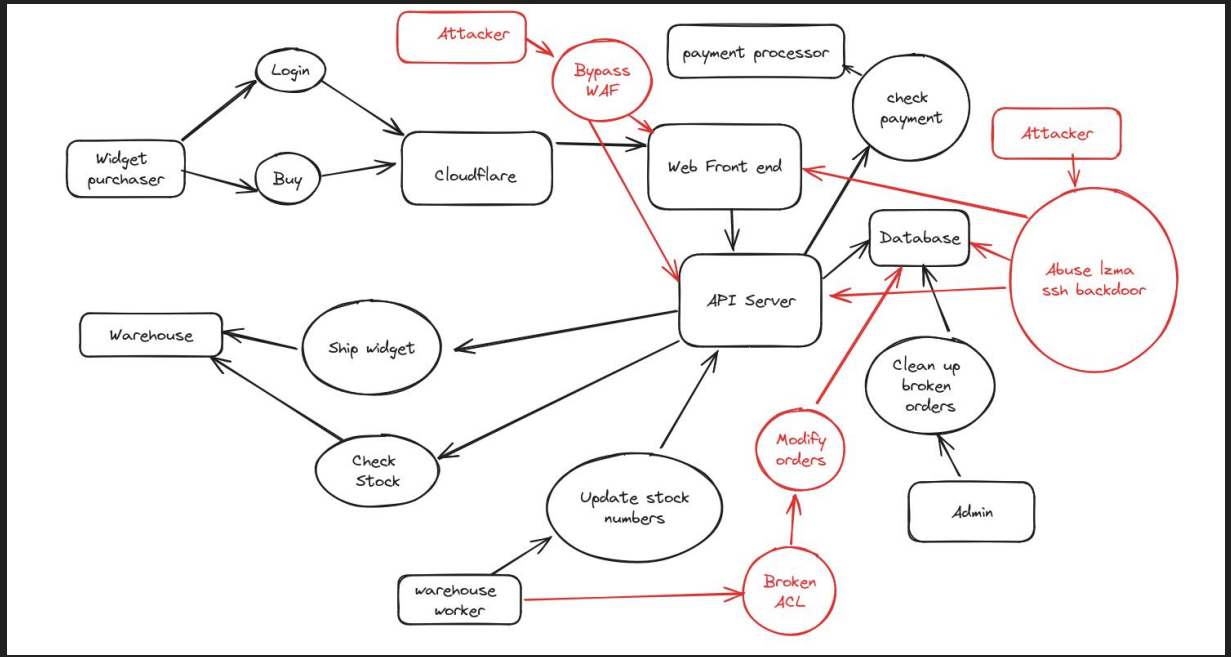
E-Commerce

- What are the trust boundaries?
- How are they enforced?
- Did we get all the processes?
- Did we get all the components / assets?



E-Commerce

- Did we consider the boundaries are probably not enforced?
  - I.e. bypass cloudflare, hit origin directly
  - No firewall to DB server, DB server is Debian, attacker abuses lzma backdoor
- Did we consider unintended states?
  - Warehouse workers can modify orders and steal things



Putting it together

Tools

- Data Flow Diagrams
  - Draw in users / components / flow
  - Draw in trust boundaries
- Attack trees
  - Draw up what bad things happen and how they chain together
- Adversarial thinking!
  - Really think of what the bad guy is doing
  - Put yourself into the proverbial ransomware den

Start small

- Get a whiteboard
- Get some other people
- Draw the system, even crudely
- Map some entry / exit points
- Ask what can go wrong
- Think adversarially

Formalising

- Techniques like Shostack's 4 question frame for threat modelling work well
- Look into STRIDE / PASTA / PNG, then know when / why to ignore them
- It's ok to stick with attack graphs or dot points of "what can go wrong"
- Beers, a whiteboard, and some other adversarial people around will work wonders

End

Slides up (shortly) at
https://github.com/proactivelabs
Sometimes the biggest threat is something stealing your dinner