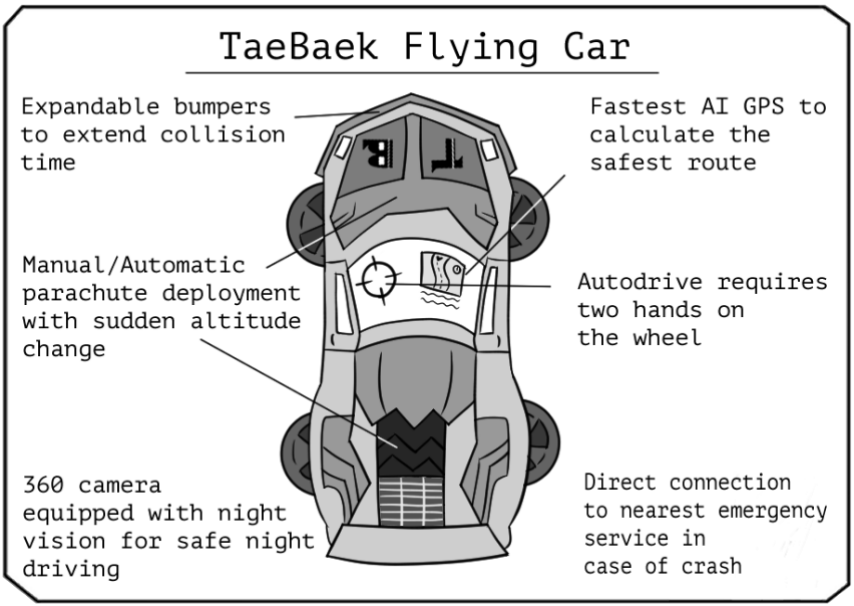
Threat Modeling and Attack Surface

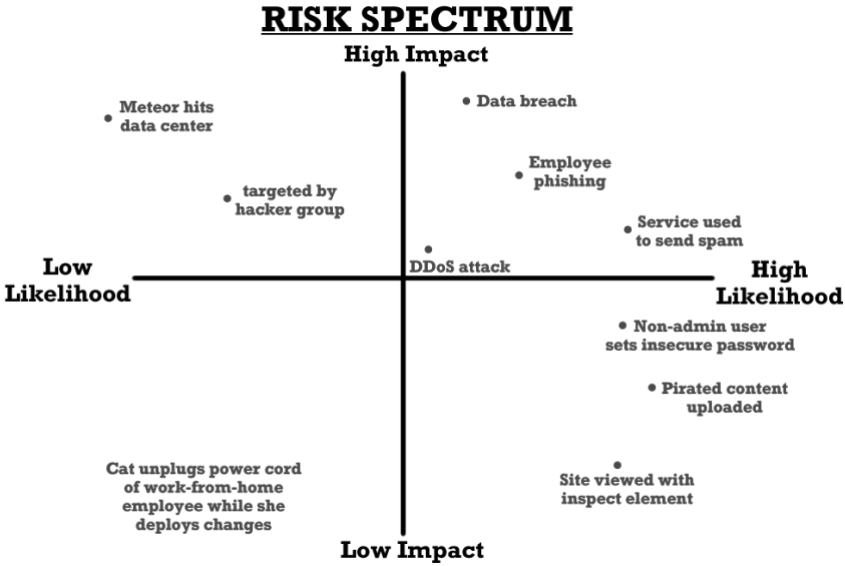
Security begins by determining what you are protecting and who you are guarding it against. Cars deploy a wide variety of safety features to protect their operators and passengers. Anti-lock brakes, crumple zones, seatbelts, and airbags are built into the physical structure of the car, while innovations like back-up cameras, adaptive cruise control, and blind-spot warnings augment the driver's senses. Each of these features was included to protect a specific area—the vehicle's cabin and its inhabitants—from specific dangers.



Securing an aging relative's desktop computer against email-delivered viruses requires different steps than securing the internal network of a critical infrastructure provider from foreign state-sponsored hackers. And

knowing the kinds of threats an application will face is essential for allocating security resources. The best anti-lock brakes in the world will do nothing to protect a driver from denting their bumper when parallel parking, while a back-up camera could save the driver an expensive repair.

Risk is a function of the likelihood of an attack multiplied by its impact. Protect against your biggest risks first and most ferociously. But what are your biggest risks? You can answer this for your own work with threat modeling.



Threat modeling is the practice of determining what types of threats to secure your application against. Threats could range from sabotage by your competitors to an attempted breach by ransomware hackers. It is impossible to defend every aspect of your systems against every conceivable

threat, so this process selects from the universe of possibilities to present likely scenarios and achievable goals.

A single threat model should answer:

- **Who** will attack your application?
- **When** will they attack?
- **Where** will they attack?
- **How** will they attack?
- **Why** will they attack?
- **What** resources will they have?

If you know the enemy and know yourself,
you need not fear the result of a hundred battles.

-Sun Tzu, The Art of War

Who keeps you up at night? Make general categories of threats to consider similar types of opponents in groups. A technology startup may be concerned about rivals trying to gain access to its proprietary code, while a financial services

company needs to consider the risk of internal employees embezzling funds from customer accounts. And while some attacks can strike seemingly at random, your industry might have a busy season, like accountants preparing for tax day or e-commerce sites handling holiday volume, making threats more pressing during those times.

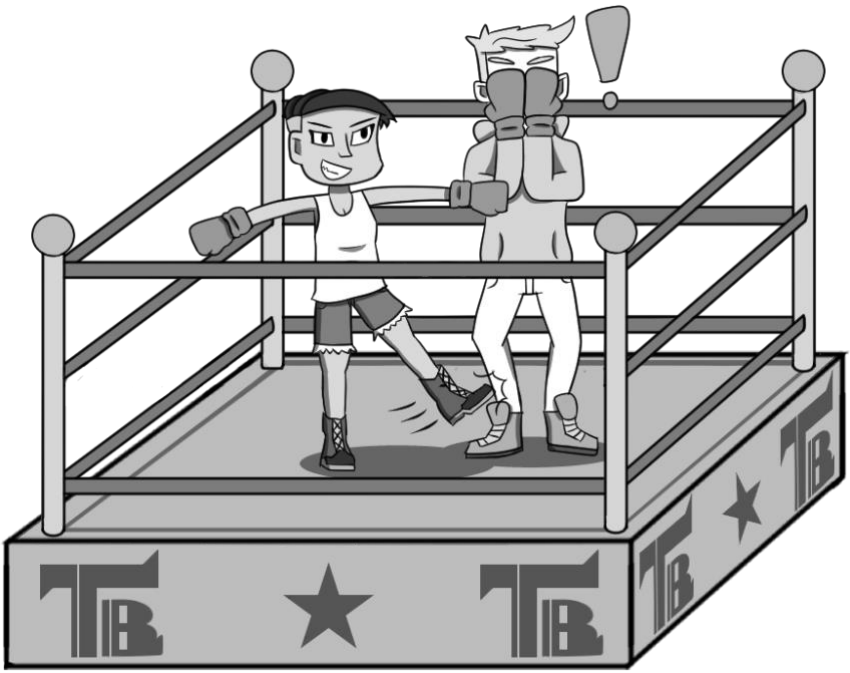
Then, define what you want to protect from these adversaries. This is the "attack surface" of your application. It could be physical (a stock exchange does not want just anyone wandering up to its servers) or digital (money should only be withdrawn from bank accounts by the account owner; a database of users' passwords should be kept private).

More abstract assets like reputation and trust may need defending as well. If a hacker infiltrates a system and does not remove any data but instead vandalizes a website, customers may lose trust in that company.

Understanding your attack surface is essential because your adversaries will attempt to exploit anything they can get their hands on. The customer-facing components of your application are clearly part of your attack surface, but it actually extends to everything exposed to the public internet. Components that are publicly exposed, but not intended for general use, like internal APIs, open SSH ports, and admin panels are all attractive targets on your attack surface.

As you review your attack surface, you may discover that some parts of your application do not need to be protected

against some threats. A boxer moving forward with his gloved hands guarding his face is defenseless against a kick to the shins. But such kicks are illegal in boxing and would be penalized by the referee. In the real world, there are no referees, but there may be aspects of your application that are structurally protected against certain threats. For example, a website that doesn't use a database doesn't need to worry about its database being compromised or corrupted.



A final step of threat modeling is thinking through your opponent's motivation and resources. No enemy has unlimited willpower or infinite resources.[7]

Your adversary's motivation limits what resources they are willing to deploy. A bored college student looking to score a quick buck from a bug bounty program might give up after basic attacks prove fruitless. But a properly incentivized opponent will employ their full resources against you.

Signal, an open-source and privacy-focused messenger app, counts the full code-breaking capabilities of every government in the world among its adversaries.[8] Their threat model for a situation might include:

- A technically advanced authoritarian government has seized a journalist's phone.
- The government wants to use messages sent over Signal as an excuse to imprison the journalist.
- The phone is unlocked.
- The government owns multiple supercomputers.

Even this adversary is not omnipotent; there are certain encryption schemes that are beyond the capabilities of all of the computing power in the world to break.[9] And while technical sophistication and financial means are important, resources extend to skill, patience, legal authority, and moral code. Anything from an internal saboteur's access credentials to a foreign nation's weakly-enforced copyright law can be a resource for your adversaries.

Enough To Be Dangerous

Security doesn't exist in a vacuum. Effective security combats the actual threats your application faces. Use these powerful concepts to focus your security mindset:

- A threat model is a description of who is attacking your application and why they are doing so.
- Your application's attack surface is anything the adversary can access and attempt to exploit.
- Attackers are not omnipotent and have finite resources. Don't underestimate your adversaries, but don't overestimate them either; they can be dissuaded and defeated.
- Risk equals likelihood times impact.