

Threat Model



TCP: No Secrecy

No Secrecy



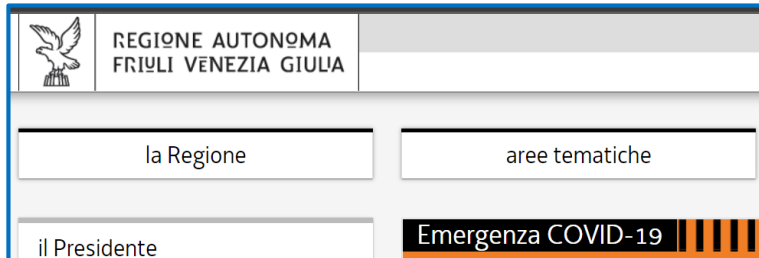
Can do "bad things" in the network

TCP: No Authentication

DNS

```
...  
regione.fvg.it      A      IP-s  
...
```

<http://regione.fvg.it>



TCP



IP-a

IP-s

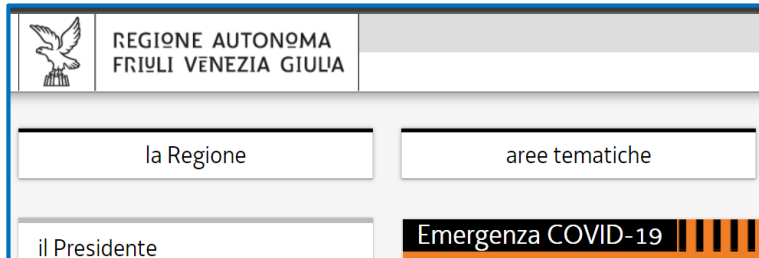
TCP

TCP: No Integrity

DNS

```
...  
regione.fvg.it      A      IP-s  
...
```

<http://regione.fvg.it>



TCP



IP-a



IP-s

TCP

TLS: Security Properties

Secrecy

Server Authentication

Integrity



❑ **Cryptographic** techniques for “strengthening” TCP connection

❑ HTTPS : HTTP over TLS

Let's change scenario



❑ **Scenario 1:** Network Attacker

⇒ TLS **guarantees** Secrecy, Integrity, Authentication

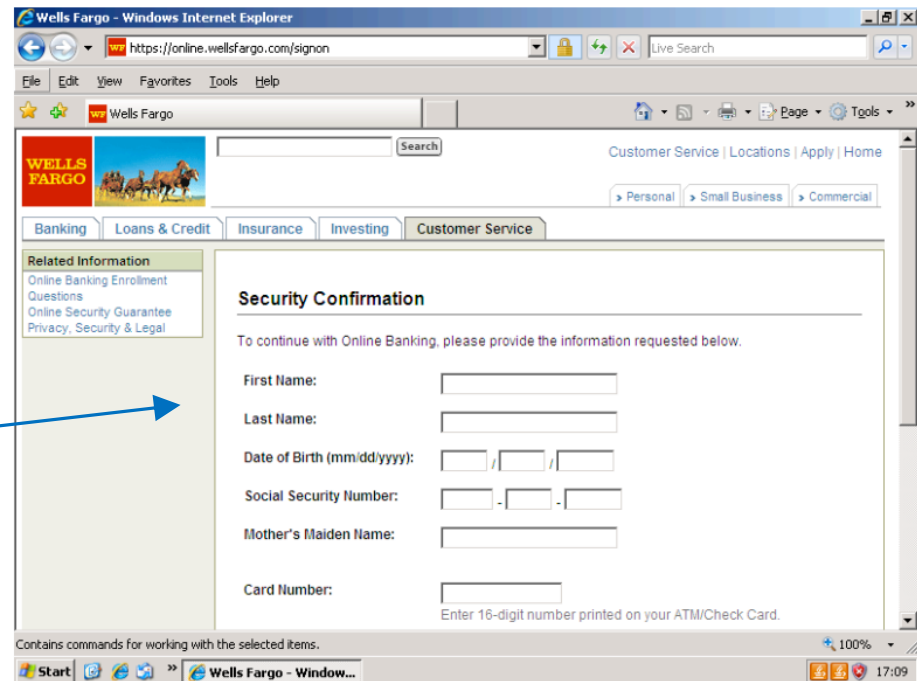
❑ **Scenario 2:** Attacker has installed **malware** in Client

...

Example

- ❑ **Malware** controllable and configurable from remote
- ❑ Can **modify** all web pages (HTTP or HTTPS)
- ❑ When on a configured banking site:
 - ❑ Fetches an HTML form from an attacker-controlled web site
 - ❑ Replaces the original form

Visually identical to the
page sent by the
banking site



Very important Question

❑ **Scenario 1:** Network Attacker

⇒ TLS **guarantees** Secrecy, Integrity, Authentication

❑ **Scenario 2:** Attacker has installed **malware** in Client

⇒ TLS **does not guarantee** Secrecy, Integrity, Authentication

So, does TLS give me security guarantees or not????



It **DEPENDS** on the Threat Model




- ❑ **Threat Model:** Set of Attacker capabilities ("what the Attacker can do")
- ❑ FUNDAMENTAL Concept in cybersecurity

❑ **Threat Model:** Network Attacker
⇒ TLS **guarantees** Secrecy, Integrity, Authentication

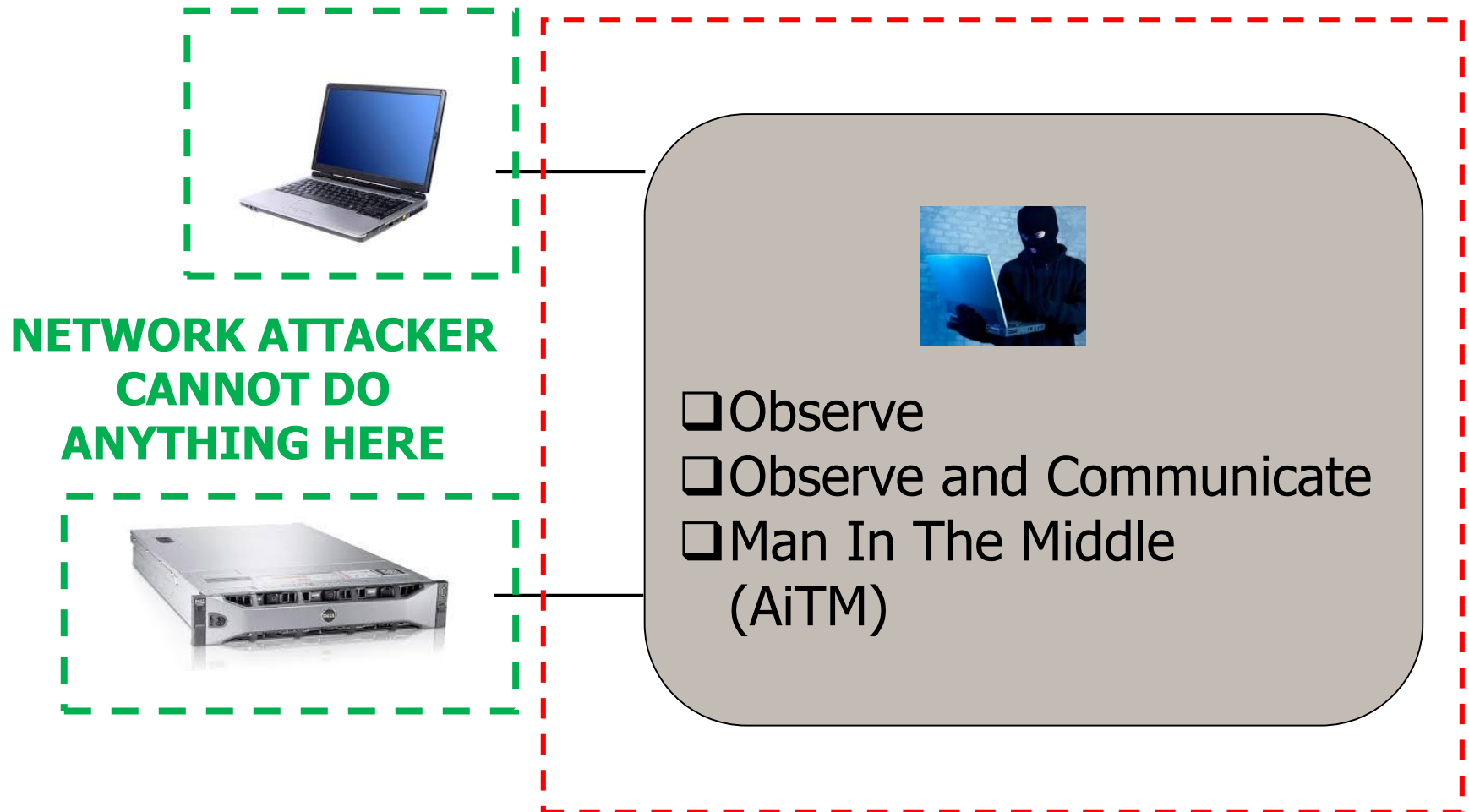
❑ **Threat Model:** Attacker has installed **malware** in Client
⇒ TLS **does not guarantee** Secrecy, Integrity, Authentication

ALWAYS specify the Threat Model!



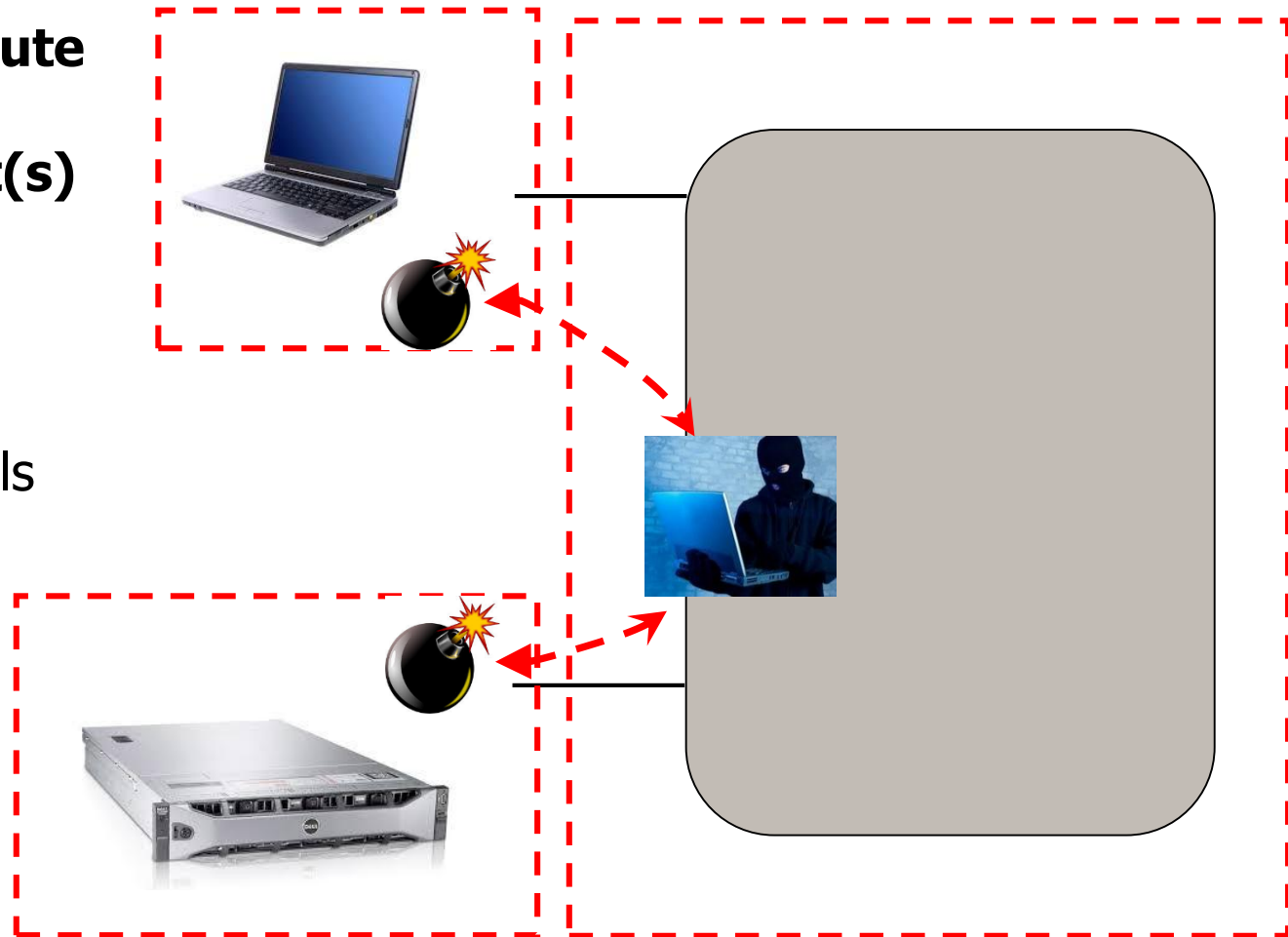
- ❑ Reasoning about "security of a system" **does not make any sense**
- ❑ You must **always** reason in terms of "security of a system with a **specified** threat model"

Threat Model: Network Attacker



Threat Model: Compromised endpoint (I)

- ❑ Attacker can execute some actions on some endpoint(s)
- ❑ Realistic?
 - ❑ Vulnerabilities
 - ❑ Stolen credentials
 - ❑ ...



Threat Model:

Compromised endpoint (II)

- ❑ Attacker can:
 - ❑ Read **some** information
 - ❑ Read **every** information
 - ❑ **Execute** some existing procedure
 - ❑ **Execute arbitrary code**



Pessimism

Other Relevant Threat models



☐ Physical access

- ☐ "If a bad guy has physical access to your computer, it is not your computer anymore"

☐ Insider

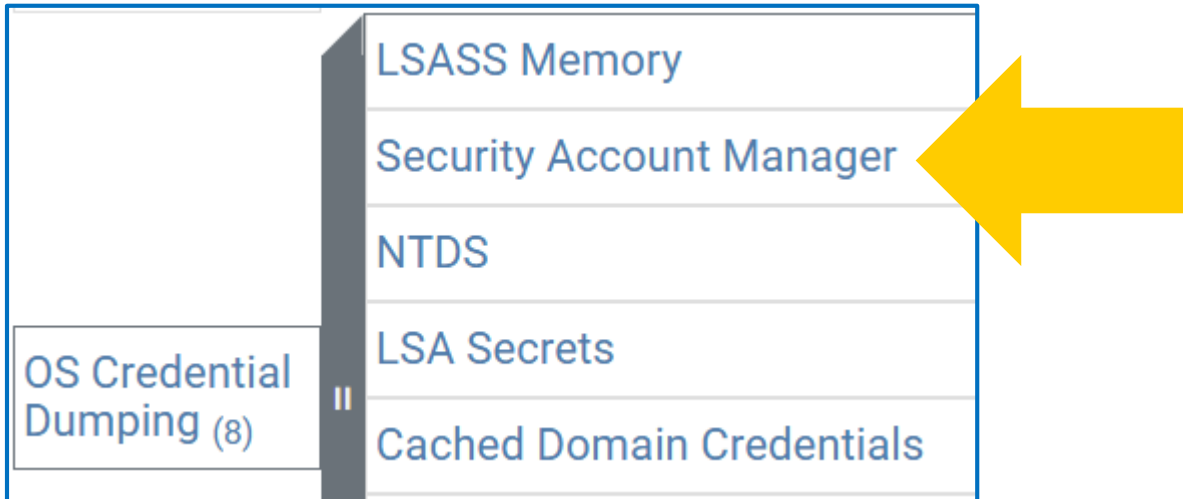
☐ Supply chain compromise

Every attack technique has a Threat Model



- ❑ Whenever you have an **attack technique**, **understand its threat model**
- ❑ To use this technique, which capabilities does the Adversary need to have?

Example



Adversaries may attempt to extract credential material from the Security Account Manager (SAM) database either through in-memory techniques or through the Windows Registry where the SAM database is stored.

The SAM is a database file that contains local accounts for the host

Every defensive tool has a Threat Model



- ❑ Whenever you have a **defensive tool**, **understand its threat model**
- ❑ What attack techniques does this tool prevent?
- ❑ What attack techniques does this tool **not** prevent?

Example: HTTPS

(as most crypto defenses)

☐ Network attacker

- ☐ Observe
- ☐ Observe and Communicate
- ☐ Man In The Middle

Secrecy
Integrity
Authentication

☐ Compromised endpoint

- ☐ Malware

~~Secrecy
Integrity
Authentication~~

☐ Physical access

☐ Supply chain compromise

- ☐ Software libraries (or a lot of other things)

~~Secrecy
Integrity
Authentication
Secrecy
Integrity
Authentication~~

Exams:

Important suggestion



- ❑ *Discuss attack X*
- ❑ *Discuss defense Y*
- ❑ **ALWAYS describe the assumed threat model!**

❑ Phishing

- ❑ Ability to send an email to the target +
Control a website reachable by the target

❑ Kerberoasting

- ❑ Valid credentials +
Ability to contact the domain controller

Case Study



Passkeys

About Passkey

Passkeys are a convenient and secure way to sign in to your Amazon account without using a password.

With passkeys, you can sign in to your Amazon account by simply using your face, fingerprint, or the PIN that you use to unlock your device. You will not need to provide your Amazon password to sign in.

Passkeys are secure and convenient sign in options as they:

- Work on most major platforms and browsers. For example, iPhones, Android phones, Apple and Windows desktops.
- Are end-to-end encrypted. Your passkeys and biometric information are never shared with Amazon, making your account safe from phishing attacks or data breaches.

- ❑ Private key stored on user device
- ❑ Public key stored at matching Web application
- ❑ Very powerful innovation in account authentication

Hmmm...

- ❑ Private key stored on user device

What if an Attacker steals the private key?



- ❑ The Attacker can impersonate the user, of course
- ❑ Exactly like any other crypto application (e.g., server authentication in TLS)

That's why



6. FIDO Security Assumptions

In this section, we enumerate the assumptions we are making regarding the security characteristics of the operating environment components on which a FIDO implementation depends.

- ❑ [SA-4] **The computing environment on the user device and the applications involved** act as **trustworthy agents of the user**

Major vulnerability



- ❑ SquareX researchers disclosed a **major passkey vulnerability**
- ❑ **This discovery breaks the myth that passkeys cannot be stolen**, demonstrating that “passkey stealing” is not only possible, but as trivial as traditional credential stealing.

Major vulnerability (?)



- ❑ SquareX researchers disclosed a **major passkey vulnerability** that uses **malicious extensions/scripts to fake passkey registration and logins**, allowing attackers to access enterprise SaaS apps **without the user's device or biometrics**.
- ❑ **This discovery breaks the myth that passkeys cannot be stolen**, demonstrating that “passkey stealing” is not only possible, but as trivial as traditional credential stealing.



Misunderstanding of "Threat model"



DEBUNKING THE BUNK

Unpacking Passkeys Pwned: Possibly the most specious research in decades

Researchers take note: When the endpoint is compromised, all bets are off.

DAN GOODIN – AUG 28, 2025 3:00 PM | 133

A fundamental misunderstanding of security

SquareX is now claiming all of that has changed because it found a way to hijack the passkey registration process. Those claims are based on a lack of familiarity with the FIDO spec, flawed logic, and a **fundamental misunderstanding of security in general.**

Understanding Threat Models



No predefined list to choose from



- ❑ Some threats are **general**
 - ❑ Modifying / Forging network messages
 - ❑ Stolen password
- ❑ Some others may depend on a **specific** environment
 - ❑ Frequent usage of external personnel on networking devices
 - ❑ Wide freedom in physical access
 - ❑ Low skilled staff are overprivileged
 - ❑ ...
- ❑ **No list** (sort of "partial order")

Threat model for organizations

❑ "Assume breach"

- ❑ Some credentials(s) compromised
- ❑ Can run software on many internal computers
- ❑ Full control of some internal computer(s)

❑ **Only** realistic model for organizations today

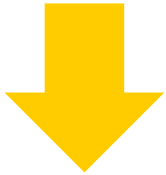
❑ Lots of (bad)implications

- ❑ In most cases, it can download **a description of everything** (except for credentials): accounts, groups, resources, ACLs
- ❑ In most cases, it can steal credentials on computers it controls

Suggestion: Forget "how"



- ❑ You are assuming a certain threat model



- ❑ **Forget** about how the Attacker can arrive there
 - ❑ There are usually **a lot** of **complex** ways
 - ❑ You would get **confused** and **miss the focus**
 - ❑ Just **take it for granted**

Example:

MITM Network Attacker



1. Vulnerabilities in network devices
2. Dishonest administrators (access point, router, DNS server)
3. Judicial authorities / Intelligence agencies
4. ARP spoofing
(open WiFi, "single password" WiFi in promiscuous places)
5. DNS spoofing
(open WiFi, "single password" WiFi in promiscuous places)
(many Windows networks)
6. BGP spoofing
7. ...


Threat Escalation: Examples



1. MITM without any credential
 2. Steal credentials
 3. MITM + One valid credential
-
1. Can communicate
 2. Inject exploit for RCE vulnerability
 3. Can communicate + Compromised endpoint

Suggestion:

Think in modular steps



1. You assume a certain threat model
(and forget about how the Adversary arrives there)
2. You realize that the Adversary can increase his capabilities
3. You assume **another and more powerful** threat model
(and **forget** about how the Adversary arrives there)



Naive question 1

- *How can I tell what Attackers can do?*
- *Maybe my threat model is too optimistic!*



Threat MODEL



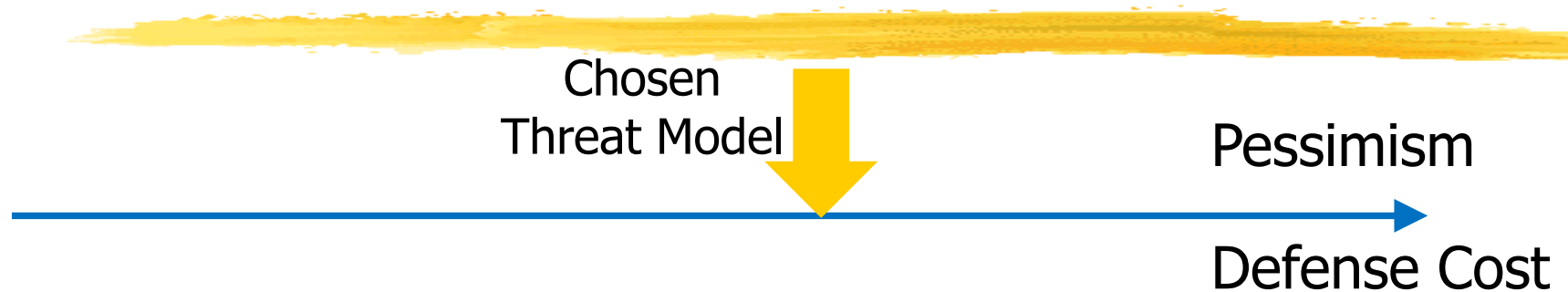
- ❑ It is a **model**
- ❑ You make some **hypotheses** about Attackers and then reason accordingly
- ❑ If the Attackers are more powerful than you assumed then your defenses will not work
- ❑ It is **impossible** to guarantee that **real** Attackers will adhere to your **model**

Naive question 2

- *Why not choose the most pessimistic threat model?*



More pessimism implies More costs



- ❑ Who can afford to build everything with threat model Supply chain compromise?
- ❑ In practice:
 1. Choose a "reasonable" working point
 2. Cross your fingers

REMIND...



- ❑ To understand cybersecurity **never** think only in **technical** terms
- ❑ **Always** think in **economical** terms
- ❑ What is the cost?
 - ❑ Attack, Defense, Incident
- ❑ Who pays?
- ❑ **Money is what drives the world**
 - ❑ It may sound cynical...but thinking in these terms is very helpful