Hacking the Code, not the Business

by Nadir Jabiyev
CISO at Xelon Cloud

The following document is a copy of the one presented by **Nadir Jabiyev** during the **#GoHack25 Hackathon** on **November 7th, 2025**.

WHOAMI

Nadir Jabiyev

With around 20 years of background in IT, including Cloud CSPs and FinTech, building IaaS, PaaS infrastructures, implementing processes and controls of PCI-DSS, ISO 27001, NIST CSF and etc.

Certified as
CISSP, CCSP, CGEIT, CISA, CCNP Security/Enterprise/R&S,
VMWare VCP, ITIL4F, PMP, TOGAF Enterprise Architecture

CISO at XELON.CH

Secure Design

What's the purpose?

Stop Bolting Security On. Build It In.

✅ *"Secure design isn't a new, magical security tool. It's a change in philosophy. We must stop treating security as a final checkbox before deployment. We must 'shift left,' embedding security into the earliest stages of design.*

The data is clear: a flaw that costs $1 to fix in the design phase can cost over $1000 to fix once it's live. This isn't just better security; it's better business."

Security-by-Design

Known Principles

Minimize Attack Surface

Defense-in-Depth

Keep it simple
&
Human Usable
&
Auditable Design

Fail-Safe Security
&
Security by Default

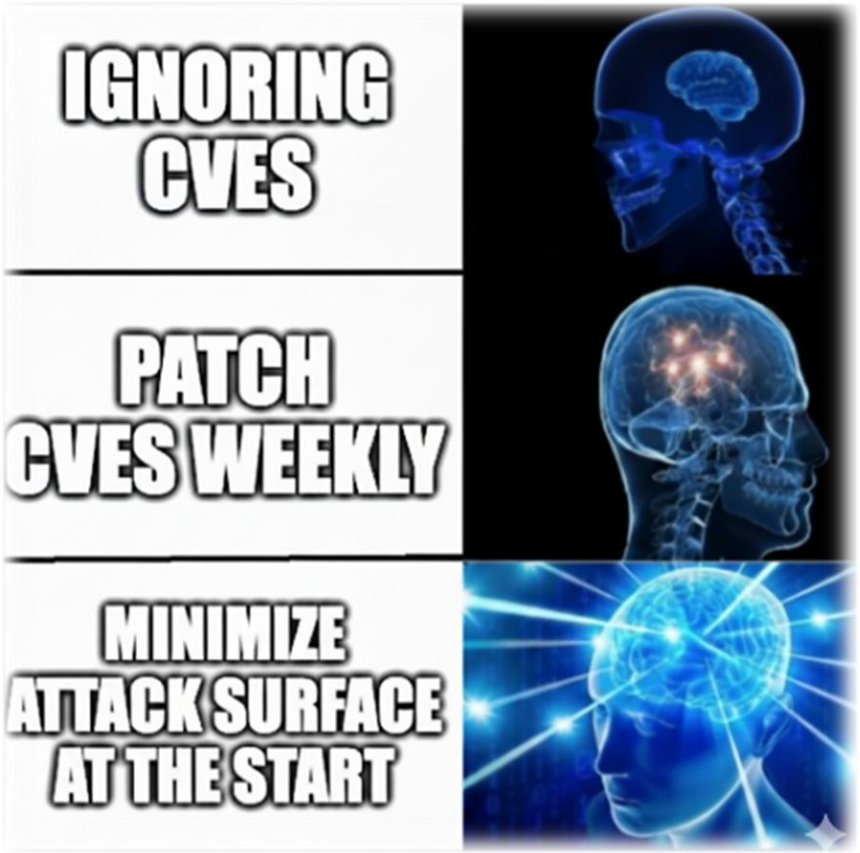
Least Privilege
&
Segregation of Duties

Segment / Isolate Systems

- Let's go through known points, and try to understand why Secure-by-Design should be threated much broader

Attack Surface ?

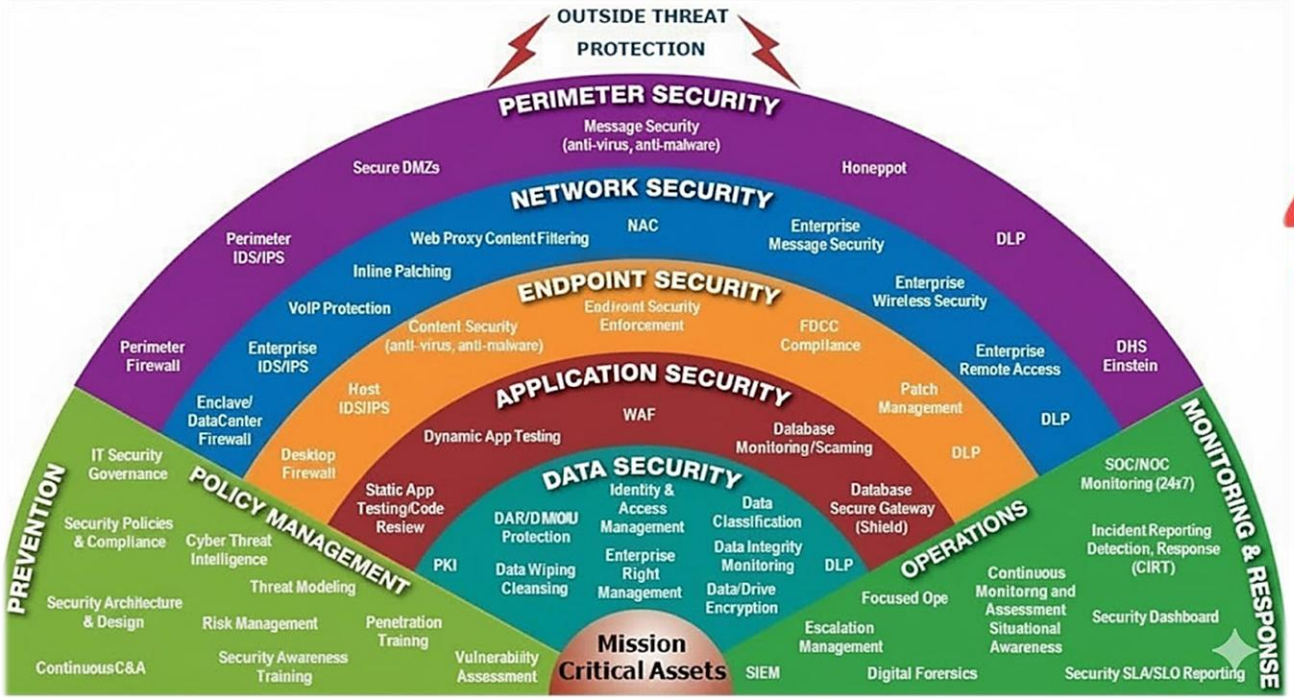
Is that the main thing ?



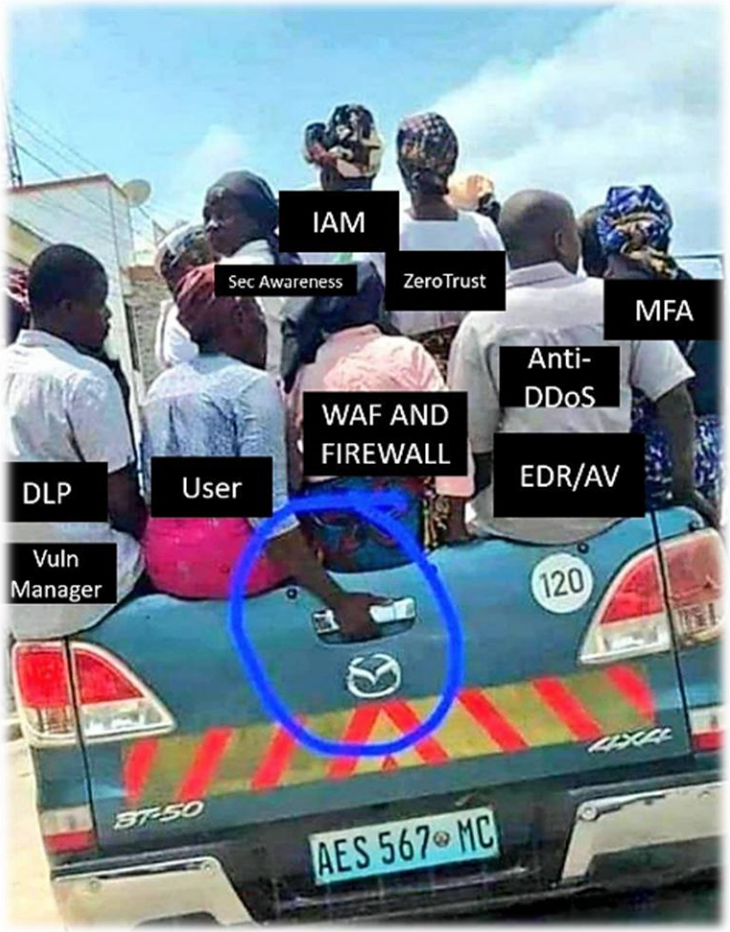
- Does “no attack surface” means minimal security risk ?
- Just remember how SCADA systems in Nuclear Facilities are deployed (Air-Gap Technology), and what happed during famous “Stuxnet Cyberattack”
- How Air-Gap help during famous “Stuxnet attack” on “Natanz Uranium Enrichment Facility” in Iran? Human factors were not well covered by the design, and some prohibited USB devices were used in protected environment
  (trojan horse scenario)

Zero Trust ?
and Defense-in-Depth ?

- Take a **Defense-in-Depth** quick-look and try to guess what is missing.



Defense-in-Depth
Focus Shift necessity



- You can obviously see your business logic vulnerability could be your core risk, and trying to mitigate with standard **DiD** approach could not work

- A lot of frauds are possible due to insecure design of high-level architectures and processes

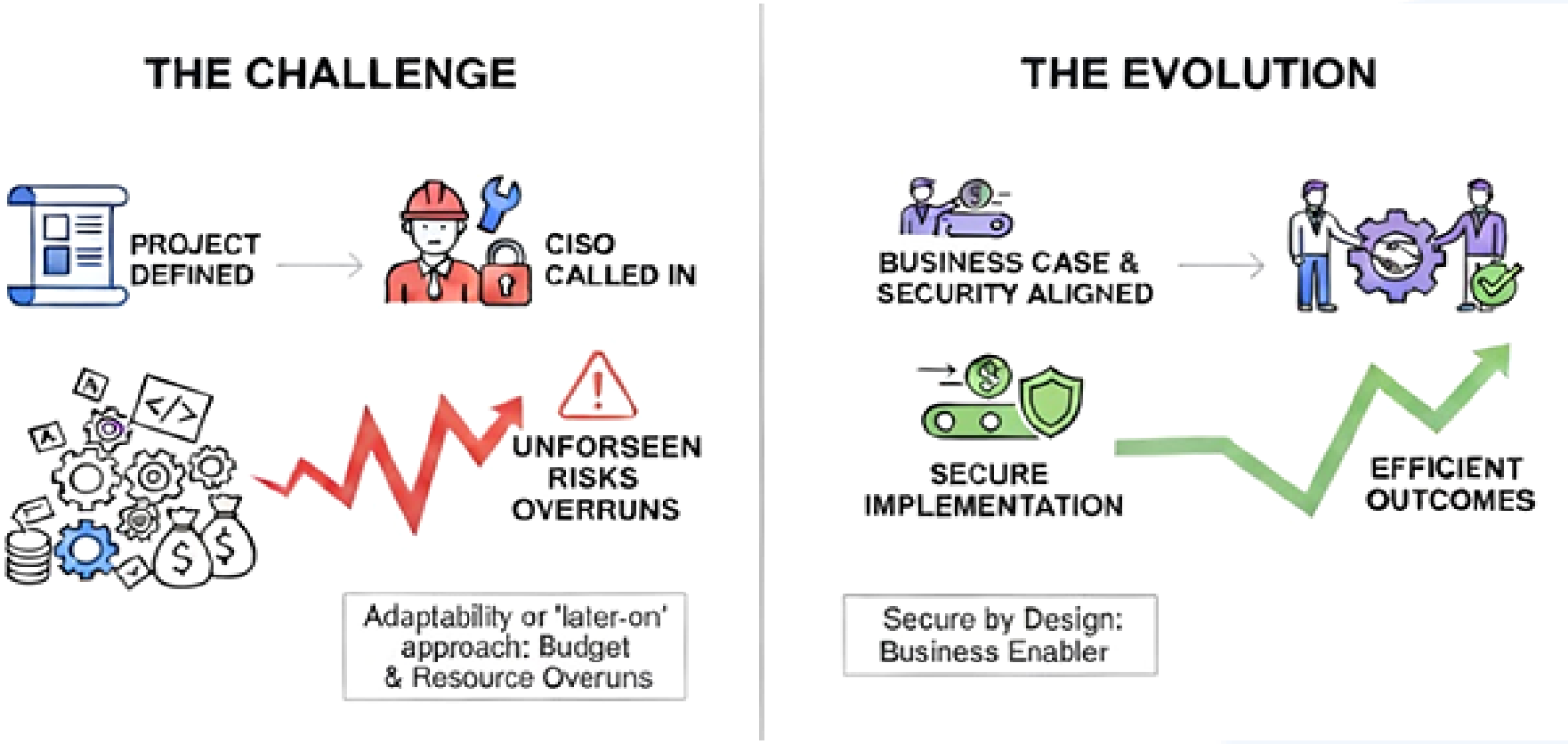
Why Business Logic First ?

- Security starts with business, but what Business Case Security actually is ?

- So where should Security-by-Design start?

The Business Logic

Security



- CISO is often called when project is already defined, and now he should secure the implementation process.

Business Logic

Security: Architectural View

- While Security-By-Design itself is a synonym for Early-Security let's try to think of abstract layers which facilitate the earliest securing actions to be taken:

Layer	Securing Activity
Business Case	Enterprise Architecture -> High Level Risk Management
Business Model / Logic	Enterprise Security Architecture
App: Logical Flaws	Threat Modelling, Data modelling
App: Bugs, Technical Flaws	Secure SDLC Framework
Infrastructure Security & Vulnerabilities	Security Compliance Frameworks Requirements (ISO, PCI, NIST, SOC2, CIS ...)
Security at the End	Security Operations or SOC

- **Security-By-Design**
  Inclusive Stages

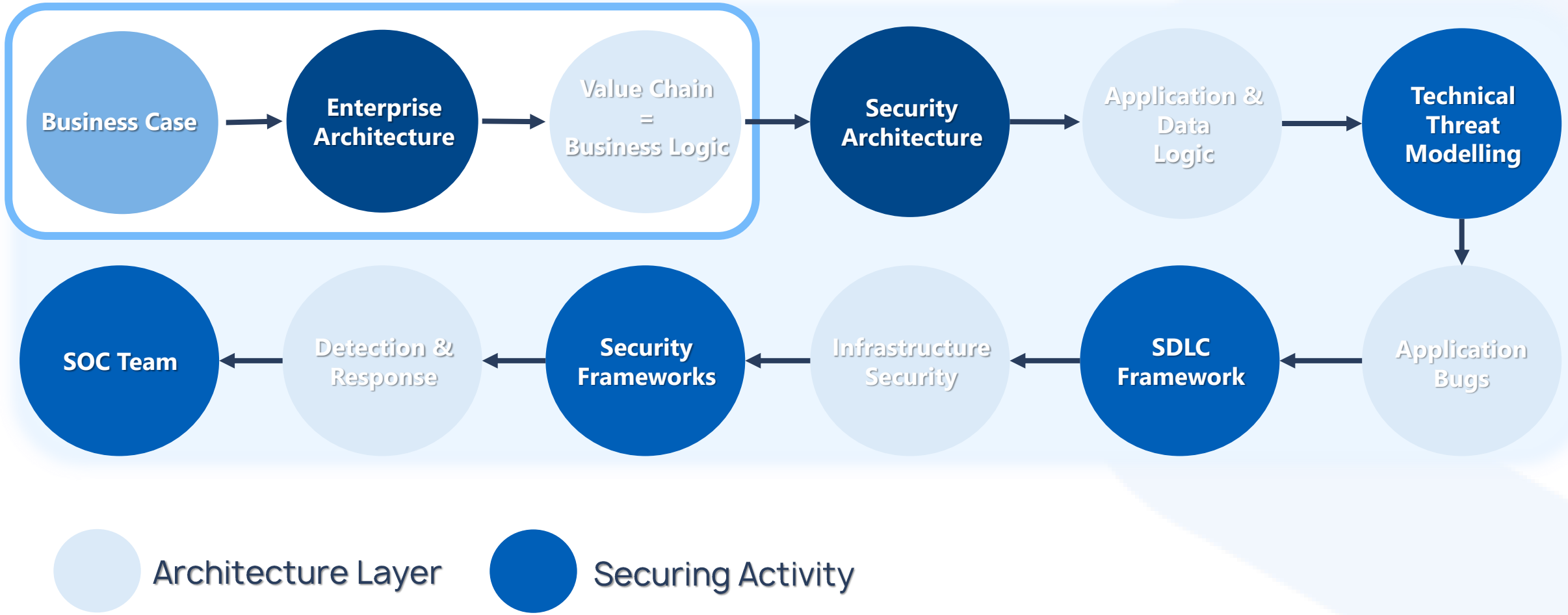
Mostly Passive Security → Act after incident

Mostly Proactive Security → Act before incident

Mostly Reactive Security → Act now

Business Logic

Security: Architectural Roadmap



Business Logic

Vulnerability?

- A business logic vulnerability occurs when flawed assumptions in an application's design affect its intended functionalities.

- Risk Factors:

  - Likelihood of exploitation
  - Vulnerability discovery
  - Business logic process flaws

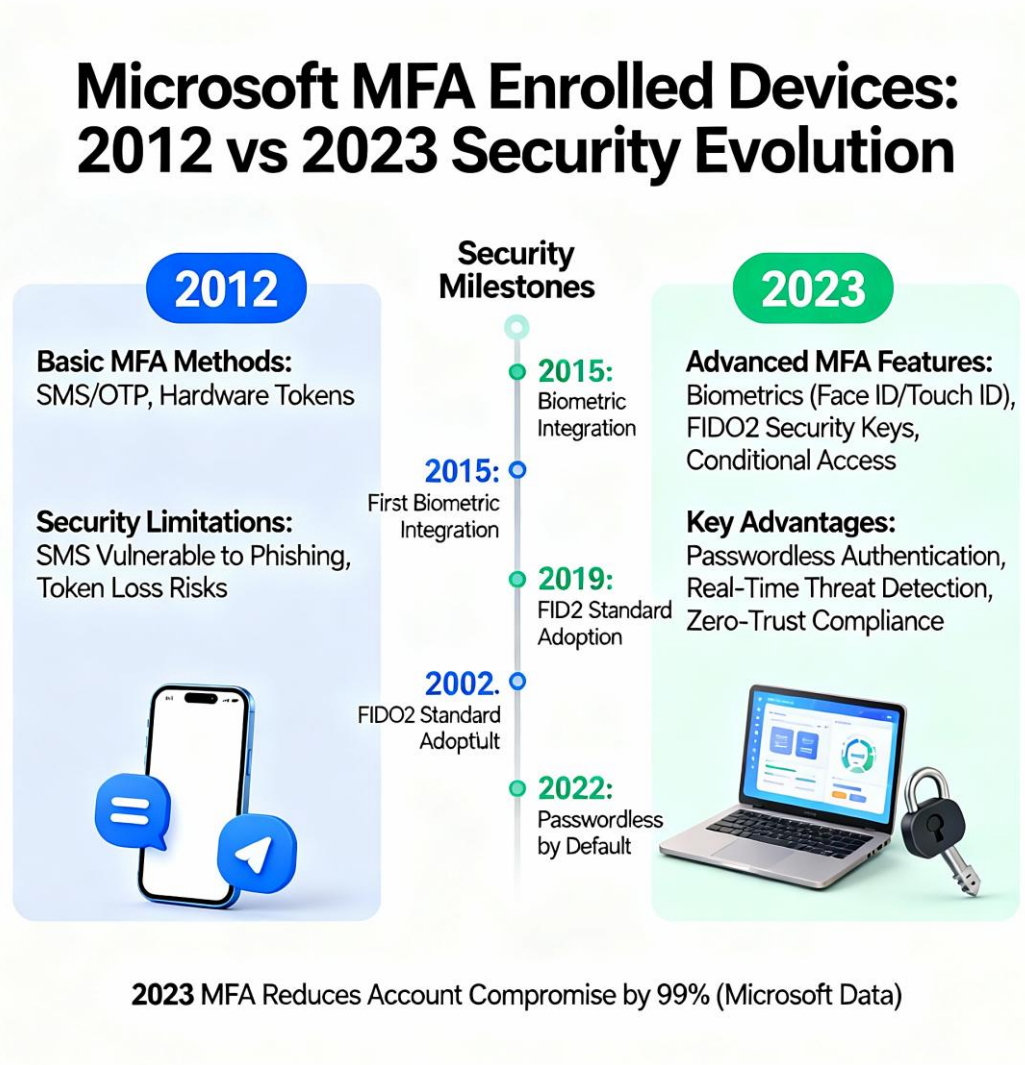
Case # 1

MFA Secure Design

Objectives: Authenticate user with additional factor into MS Services securely:

✅ *It took Microsoft 11 Years to realize that just MFA with QR enrollment and TOTP generation bound to clock is not secure in multiple ways.*

	MFA by Microsoft in 2012 with Insecure Design.	MFA since 2023 with Secure Design.
Enrollment	Could be enrolled anywhere multiple times by static QR. TOTP Could be generated at any device	Can only be enrolled with a specific device using a single QR once
Fact of Authority	Time-bound TOTP check	Push Notification Approval
Fact of Presence	None	Number matching on screen, followed-up by biometry



Case # 2

Enrollment Fraud – Risk A

- Objectives: Banking App should provide secure services to prevent frauds:

First registration / Device enrollment fraud: Often the weakest part of most applications including the banking ones.

1. Social Engineering to trick customer into telling the SMS codes (Numbers from Marketplaces get calls with further victim manipulation)
2. Threat actor initialize SMS OTP and Victim Phone Number to login to Banking App.
3. Getting access to banking app, cards, accounts → Transferring all the funds.

Case # 2

Enrollment Fraud – Risk B

- Objectives: Banking App should provide secure services to prevent frauds:

Sample: Fake "lottery" that uses Phone number as authentication factor:

1. Creation of fake domain website like targeted Bank and offering a lottery with emulation of banking app login interface (with TOTP and even pre-saved app password).
2. Starting social advertisement referring to fake lottery web portal.
3. Victim customers visit the lottery website, seeing interface like banking's app try to login specifying number, TOTP and even app password.
4. Threat actors receive the logon data in real-time, gaining access to Banking App → Transferring all the funds.

Case # 2

Enrollment Fraud – Flaw in the Design

Objectives: Redesigning the Device enrollment method would overcome the above-mentioned risks.

Not ideal

- Make manual confirmations of new devices → Resource constrains on business model
- Block transfers of newly registered devices for 3 days
- Authorize every transaction or setting with additional SMS
- Complicate device management with advanced controls and verifications
- Block app to a single device only, locking previous sessions after new device login with confirmation
- Implementing real-time analytics with SOC team to monitor all logins for anomalies.

Ideal

- Add government issued biometry verification requirement (Has to be integrated with authorized services to keep biometry as known factor)
- Allow previously authorized device to allow new device registration. (This is how it is implemented in Telegram messenger)
- Do not phone numbers as first login factor, try to map to a pre-known customer secret code (pre-known factor) requesting additional confirmation from existing devices.
- **Your Solution ?**

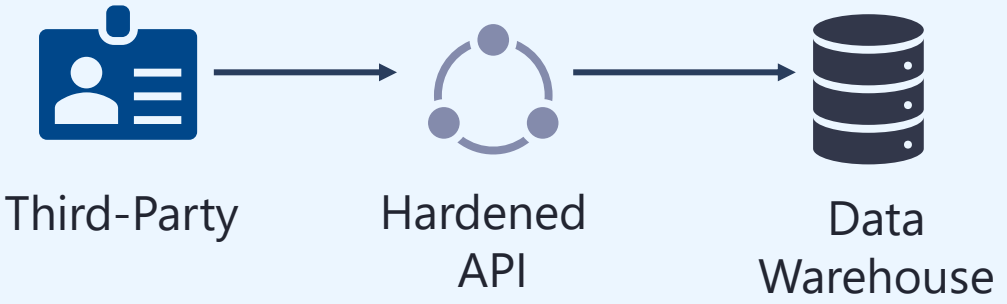
Case #3

Secure API vs Safe API

Objectives: Integrate third-party access into your system

Imagine an external system which should be integrated into your warehouse data with access to a filtered information.

- Typical approach?
- Risks?



Case #3

Secure API vs Safe API

Objectives: Integrate third-party access into your system

- Secure Designs steps in (as one of the options):

- Risks ?

Data and Surface Minimization, Least Privilege, Reduce Risk of leakage.

The Solution:
Build Security In

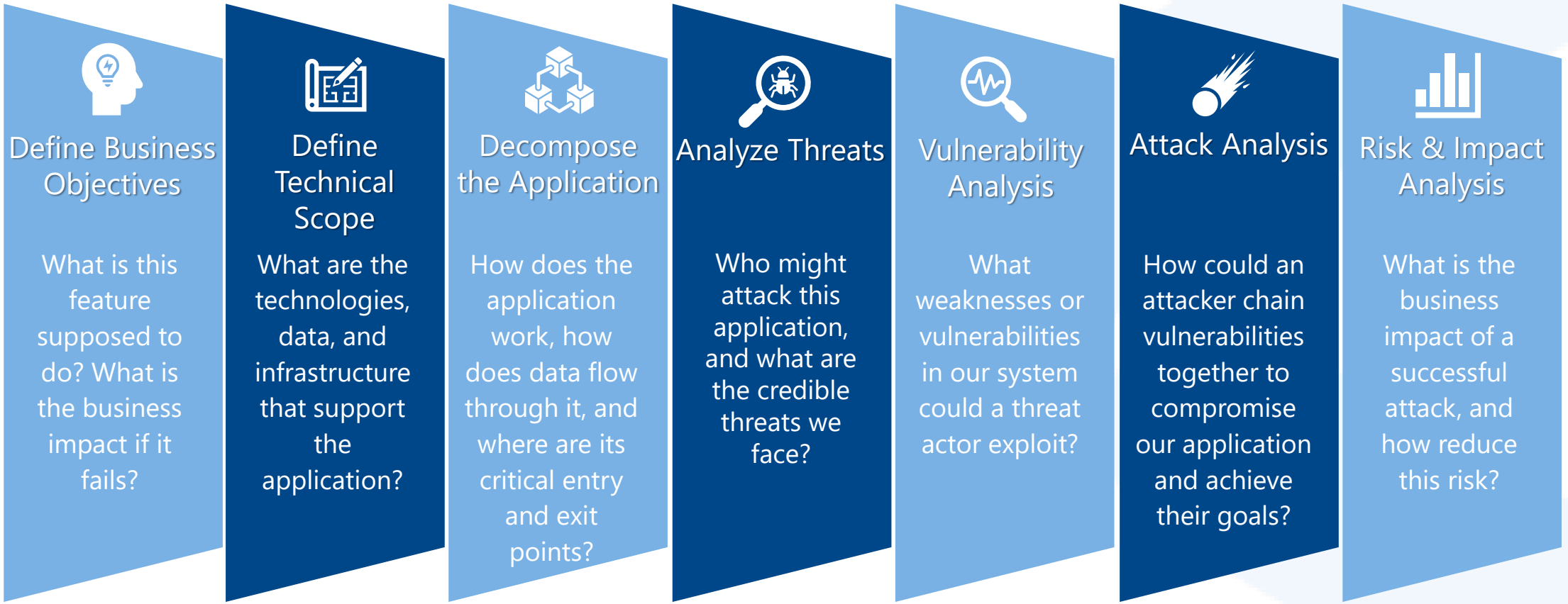
- **Shift Left / Secure by Design:** Move security from the end of the process to the very beginning.

- "A philosophy is great, but we need a practical tool. That's where a threat modeling methodology like PASTA comes in. Unlike purely technical models, PASTA is risk-centric. It forces us to start by asking business questions

** Note that PASTA threat modeling was created by Tony UcedaVélez and Marco M. Morana from VerSprite.*

PASTA Framework

- Practical Solution: Process for Attack Simulation and Threat Analysis is a risk-centric methodology that starts with the business objectives.



PASTA Framework

Business case: A "one-time use" promo code feature sample.

Define Business Objectives

What is this feature supposed to do? What is the business impact if it fails?

- ✓ Business goal: increase new user acquisition through limited-use codes.
- ✓ Critical asset: promo code logic, database, discount rules, campaign analytics.
- ✓ Risk appetite: moderate — fraud or code abuse affects profit margins.

Define Technical Scope

What are the technologies, data, and infrastructure that support the application?

Scope includes:

- ✓ Web / Mobile App front-end
- ✓ API endpoint /applyPromo
- ✓ Promo code database
- ✓ CRM / Marketing backend
- ✓ Payment / Checkout system

PASTA Framework

Business case: A "one-time use" promo code feature.

Decompose the Application

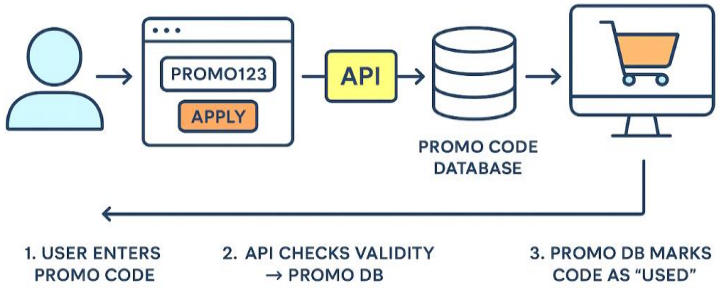
How does the application work, how does data flow through it, and where are its critical entry and exit points?

For One-Time Promo Code:

- User → enters promo code.
- API checks validity → promo DB.
- Promo DB → marks code as "used".
- Discount applied → checkout.

Application Decomposition & Data Flow

Break down how the feature works step-by-step



Analyze Threats

Who might attack this application, and what are the credible threats we face?

Examples of threats:

- **Replay attack** – user reuses same promo multiple times.
- **Code enumeration** – brute-forcing valid promo codes.
- **Tampering** – intercepting API traffic to modify code status.
- **Privilege abuse** – insider creates or resets codes.

PASTA Framework

Business case: A "one-time use" promo code feature.

Vulnerability Analysis

What weaknesses or vulnerabilities in our system could a threat actor exploit?

Threat	Possible Weakness
Replay Attack	No server-side state validation
Enumeration	Predictable promo code patterns
Tampering	Missing request signing / TLS
Insider abuse	Lack of audit logging / segregation of duties

✅ *Thought:* "Let's see which weaknesses exist in the current implementation."

Attack Analysis

How could an attacker chain vulnerabilities together to compromise our application and achieve their goals?

Attack Path Example:

- Attacker intercepts /applyPromo.
- Modifies request or replays same promo multiple times.
- System applies discounts repeatedly due to missing state lock.

✅ "We simulate the attack — yes, promo reapplication works. Financial loss confirmed."

PASTA in Action

Business case: A "one-time use" promo code feature.

Risk & Impact Analysis

What is the business impact of a successful attack, and how reduce this risk?

Threat	Likelihood	Impact	Risk	Mitigation
Replay attack	High	Medium	High	One-time use token with DB check
Enumeration	Medium	Medium	Medium	Rate limiting, code complexity
Tampering	Low	High	Medium	Strict TLS, signed requests
Insider abuse	Low	High	Medium	Logging, RBAC, approvals

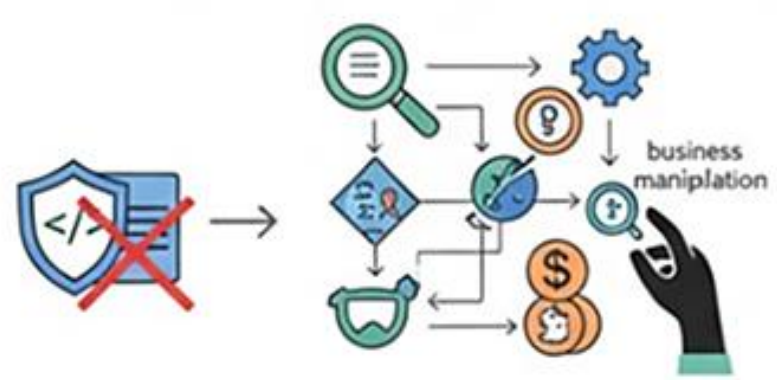
Business vs Technical

Approach

	PASTA (Higher-Layer & Business-Centric)	STRIDE (Technical-Layer & Model-Centric)
Primary Lens	**Risk-Centric & Business-Centric** * Analyzes threats based on their potential impact on **business objectives**.	**Model-Centric & Component-Centric** * Analyzes threats as they apply to **technical components** and data flows.
Analytical Scope	**Top-Down Approach** * Starts from **business goals** and drills down into technical vulnerabilities and weaknesses.	**Bottom-Up / Technical Approach** * Applies a set of known threat categories directly to the **system's design** (e.g., DFDs).
Typical Use Case	* Complex, **business-centric systems** where risk must be quantified and prioritized.	* During the **system design phase** to build security in from a technical standpoint.
Core Output	* A comprehensive, prioritized risk report that includes attack simulation and weakness analysis.	* A classification of potential threats against system components (Spoofing, Tampering, etc.).

Conclusion

- Secure the Business, Not Just the Code



The threat has evolved from code exploits to business process manuplation:

- A "Secure by Design" approach is more effective and cost cost-efficient:
- Use business-centric threat modeling (like PASTA) to find flaws early

Call to Action:

For Leaders: Champion a security-first culture. Security is business enabler, not cost center.

For Developers:
Learn the business domain you're coding for. Ask "How can this be abused?"

For Security Pros: Engage with business analysts.
Facilitate threat modeling. Become the great partner in secure design.

Thank you