T-401-ICYB
Principles of Defense

Stephan Schiffel
stephans@ru.is
Reykjavik University, Iceland

08.12.2025

Outline

Economics of Defense

The Defender's Dilemma

The Asymmetry of Cyber Warfare

Defenders must be right **100%** of the time.
Attackers only need to be right **once**.

The Goal: The CIA Triad

- **Confidentiality:** Only authorized access.
- **Integrity:** Data is trustworthy and unaltered.
- **Availability:** Systems work when needed.

Security vs. Usability Trade-off

- The most secure computer is unplugged, buried in concrete, and guarded by sharks.
- *Problem:* It is useless.

Principle

Security measures must be **Psychologically Acceptable**. If security is too hard, users will bypass it.

Understanding Risk

We do not *eliminate* risk; we *manage* it.

$$Risk = Threat \times Vulnerability \times Cost$$

- **Threat:** Who attacks? (Script Kiddie vs. Nation State)
- **Vulnerability:** Where are we weak? (Unpatched bug)
- **Cost/Impact:** What do we lose? (Data, Fines, Reputation)
- $(Threat \times Vulnerability = Likelihood)$

Strategies for Handling Risk

Mitigation is not the only option when facing a risk:

- **Mitigate/Reduce:** Reduce likelihood or impact to an acceptable level.
- **Avoid:** Discontinue the risky activity (e.g., "We will not store credit card numbers").
- **Accept:** The cost of the fix > impact. Management signs off.
- **Transfer:** Move risk to a 3rd party (Cyber Insurance, Cloud Provider).

Cost-Benefit Analysis

Example

You do not buy a \$5,000 titanium lock to secure a \$50 bicycle.

Application in CS:

- Defense costs must not exceed the value of the asset.
- Defense in Depth is a form of Risk Management.

Design Principles & Defense in Depth

Core Design Principles

Saltzer & Schroeder, The Protection of Information in Computer Systems, 1975

Least Privilege

Every program and user should operate using the least amount of privilege necessary to complete the job.
Example: Don't run your web server as root.

Fail-Safe Defaults

Access decisions should be based on permission rather than exclusion. Default to "Deny."
Example: Firewalls should block all traffic by default.

Core Design Principles (Continued)

Economy of Mechanism

Keep the design simple. Complexity hides vulnerabilities.

Complete Mediation

Every access must be checked against the access control database (even in the internal network). No caching permissions blindly.

Open Design

Security should not depend on the secrecy of the implementation.
Avoid: Security by Obscurity

Defense in Depth

The Layered Approach (The Onion)

No single control is infallible. If one layer fails, the next must catch the threat. For example:

1. **Physical:** Locks, cameras, guards.
2. **Technical:**
   - **Perimeter/Network:** Firewalls, DMZ, VPN, Intrusion Detection
   - **Host/Endpoint:** Antivirus, Monitoring
   - **Application:** Input validation, secure code.
   - **Data:** Authorization, Encryption (at rest/transit), Hashing, Backups.
3. **Administrative:**
   - **People:** MFA, Password policies, Training
   - **Technology:** Patch Management, Risk Assessment
   - **Operations:** Principle of least privilege

Scenario: If a laptop is stolen (Physical fail), the hard drive encryption (Data layer) protects the information.

Classifying Controls

By Type:

- **Physical:** Fences, Locks, Fire Suppression.
- **Technical:** Firewalls, Encryption, ACLs.
- **Administrative:** Policies, Training, Background Checks.

By Function:

- **Preventive:** Stop the attack (Firewall).
- **Detective:** Notice the attack (IDS, Logs).
- **Corrective:** Fix the damage (Backups).

Standards, Frameworks, Checklists

On the Importance of Checklists

- Checklists fossilize procedures
  - Set expected procedure and outcome
  - Can/Should be Audited
  - Reviewed when mistakes happen and improved.
  - Think of them as a synchronisation point
- Allow training and practice
  - Critical when dealing with ”real-time” situations
  - When the incident is started need to react quickly
- Widely adopted by critical industries - Aviation, medicine, etc.

NO—Complexity - Simple Security Checklists

Key Frameworks & Standards

Don't reinvent the wheel. Use established blueprints.

Management & Process (Organization)

- **ISO/IEC 27000 Series:** International standard for Information Security Management Systems (ISMS). Focus on governance.
- **NIST CSF:** The US Standard. Identify, Protect, Detect, Respond, Recover.

Technical Implementation (Developers/Ops)

- **CIS Controls:** Prioritized set of actions (e.g., Inventory, Data Protection, Log Management).
- **OWASP:** The standard for web application security (e.g., OWASP Top 10).

EU Regulations

Apply in Iceland because of EEA membership.

- **GDPR (General Data Protection Regulation):** Stringent rules on privacy, consent, and heavy fines for data breaches.
- **NIS2 Directive:** Mandatory security measures for "Essential Entities" (Energy, Transport, Health, Digital Infrastructure).
- **Cyber Resilience Act (CRA):** *Critical for Developers:* Mandatory security requirements for products with digital elements (Software/IoT) placed on the EU market.
- **Digital Operational Resilience Act (DORA):** IT security/resilience regulation for financial entities and their suppliers.

Other Compliance Standards

- **PCI-DSS:** Payment Card Industry (Handling Credit Cards).
- **HIPAA:** Health Insurance Portability and Accountability Act (US Medical Data).
- **SOC 2:** Audit procedure for service providers (SaaS/Cloud).

Incident Response & Recovery

Response: The Assumption of Breach

Assume you will be (or already are) breached.

- **Incident Response Plan (IRP):** The "Fire Drill." Who do you call? What do you turn off?
- **Chain of Custody:** Preserving evidence for legal action. Don't just reboot the server!

Recovery: RPO and RTO

RPO (Recovery Point Objective)

- "How much data can we afford to lose?"
- Determined by backup frequency.

RTO (Recovery Time Objective)

- "How long can the system be down?"
- Determined by redundancy and failover speed.

Resilience: The 3-2-1 Backup Rule

Ransomware targets backups first.

The 3-2-1 Rule

- **3** Copies of data.
- **2** Different media types (e.g., Disk + Tape/Cloud).
- **1** Copy ~~offsite~~ offline (physically separated).

Immutable Backups: Backups that cannot be altered or deleted, even by an administrator, for a set period.

CTFs and Bug Bounties

What is a CTF?

Definition

Capture The Flag (CTF) competitions are cybersecurity exercises where participants solve challenges to find a hidden string of text, known as the "Flag".

The Format:

- **Jeopardy Style:** Challenges are categorized by topic and difficulty. Most common for beginners.
- **Attack-Defense:** Teams defend their own server while attacking others. (Advanced).

CTFs are gamified learning.
Platforms: picoCTF, TryHackMe, OverTheWire, Hakkaraskóli GGFÍ, HackTheBox, ...

What is a Bug Bounty?

The "Gig Economy" of Security

Crowdsourced security testing where companies pay independent researchers to find and report vulnerabilities.

How it works:

1. Company launches a program.
2. Researcher finds a bug (e.g., Remote Code Execution).
3. Researcher writes a report explaining **how** to reproduce it.
4. Company validates the bug and pays a "Bounty".

Major Platforms: HackerOne, Bugcrowd, Intigriti, **Defend Iceland**.

CRITICAL: The Rules of Engagement

The Difference between Hacking and Crime is SCOPE.

- **In a CTF:** You have implied permission to break specific assets.
- **In Bug Bounties:** You must read the **Policy Page**.
- **Scope Constraints:** Examples:
  - "Only test `*.dev.example.com`"
  - "Do not perform DDoS"
  - "Do not access user data"

If it's not in scope, don't touch it.

Legal Context: Europe & Iceland

International: The Budapest Convention

The foundational treaty on cybercrime. Signed by US, Iceland, and EU members. Harmonizes definitions of "Illegal Access."

Iceland (Almenn hegningarlög 19/1940)

- **Article 228:** Criminalizes unauthorized access to data.
- **Article 257:** Criminalizes damaging data or systems.

GDPR

Hacking often involves **Personal Data**.

- Exfiltrating a database of users is not just hacking; it is a GDPR violation.
- **Rule:** Prove the bug exists, but **never** download user data.

Summary

Modern Evolution: Zero Trust

The "Castle and Moat" model is outdated due to Cloud and Remote Work.

Zero Trust Architecture

"Never Trust, Always Verify."

- No "inside" vs. "outside" the network.
- Every request is authenticated, authorized, and encrypted.
- Continuous verification.

Stay Informed

- Exploit DB https://www.exploit-db.com/google-hacking-database/
- Microsoft Secure https://cloudblogs.microsoft.com/microsoftsecure/
- National Vulnerability Database (USA) https://nvd.nist.gov/
- Internet Storm Center https://isc.sans.edu/
- Common Vulnerablities and Exposures (CVE) https://cve.mitre.org/
- Full Disclosure Mailing Lists: fulldisclosure@seclists.org seclists.org

Summary

1. **Mindset:** Defense is difficult; attackers only need one win.
2. **Risk:** Drive decisions based on Cost vs. Benefit.
3. **Design:** Least Privilege, Simplicity, Open Design.
4. **Architecture:** Layered defense (Defense in Depth).
5. **Resilience:** Plan for failure (Response & Recovery).

Up Next ..

Further Studies

Zero Trust sounds great in theory ("Never Trust, Always Verify"), but how do you actually implement that?

- Read the Google Research Paper "BeyondCorp: A New Approach to Enterprise Security".
- Find out how Google replaced VPN and Firewalls as a defense mechanism by relying on "Zero Trust".
- What replaces the Firewall as a primary enforcement point?
- How does a system decide if a device is trusted?
- What are advantages and disadvantages of this approach regarding security (Hint: look at the Core Design Principles)?

Lab today

- Lab 10: LLM Jailbreak