The Cryptography and Information Security Profile

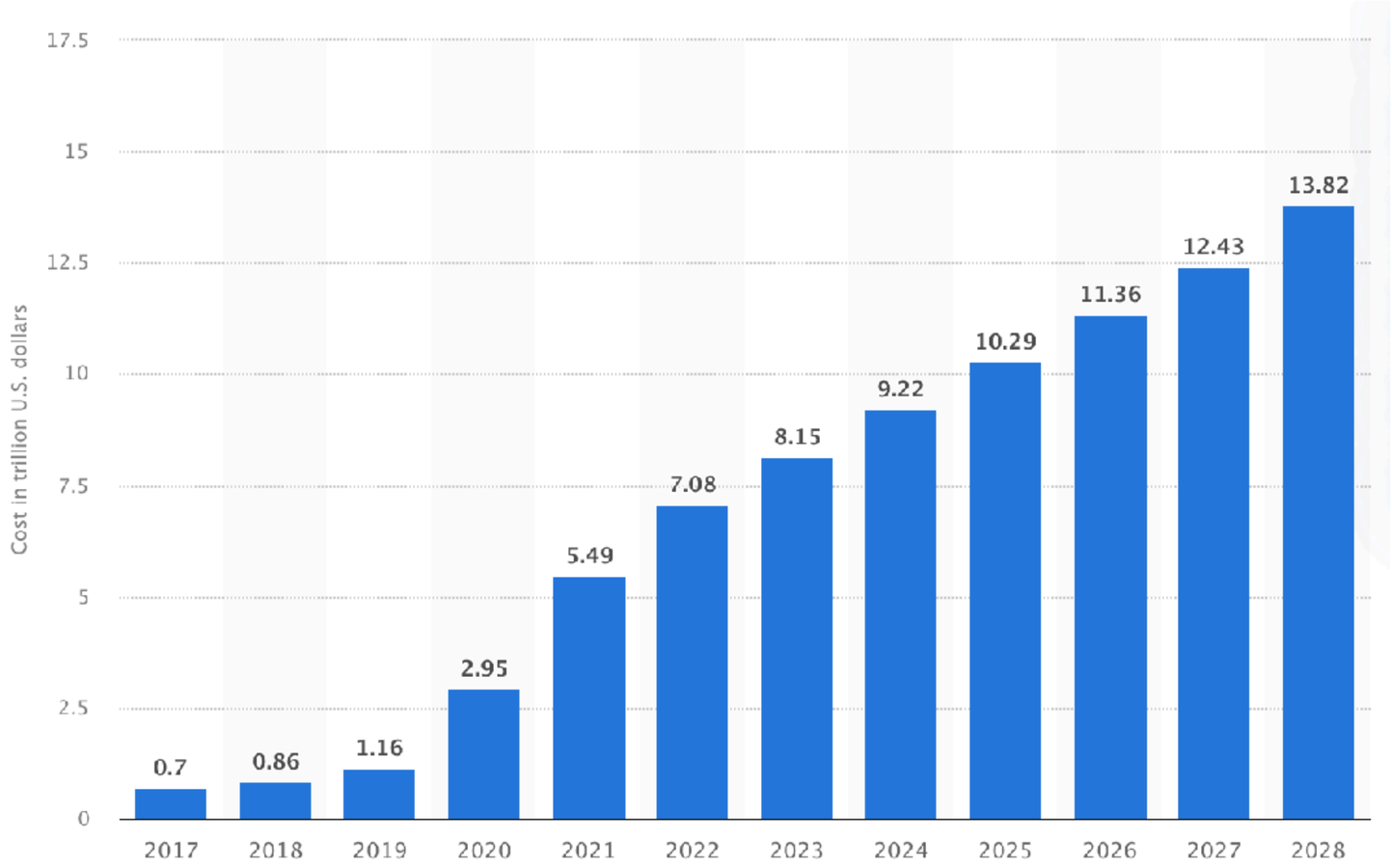
Master in Informatics Engineering
University of Minho
2024-25

Vítor Francisco Fonte, vff@di.uminho.pt, University of Minho, Dec 2024

The Cybersecurity Problem

Problem?
What problem?

Estimated of Cybercrime Worldwide 2017-2028



Source: https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide

Challenges of Cybersecurity

Adequate abstractions, frameworks, tools

Requirement gathering and specification

Product development and project management

Interdependencies and legacy components

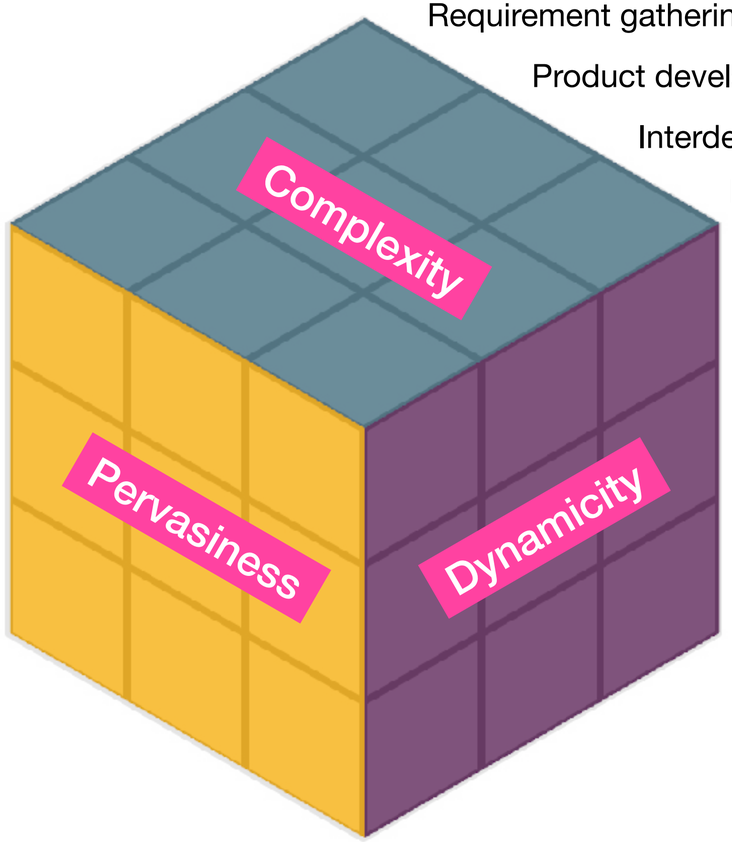
Legal, administrative and regulatory frameworks

Holistic approach

Hardware, software, communications

People, organisations

Cross-organisation, cross-domain



Continuous process

Ever-changing, unpredictable threat landscape

Race between attackers and defenders

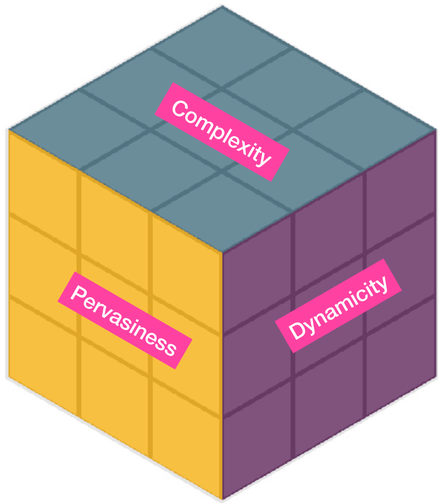
An uneven playing field

Defender

- Holistic, continuous, complex process
- Cost of preparedness can by high
- Impact of attack can be very damaging

Impact of attack vs. cost of being prepared

- Limited resources & budget
- Adequate risk analysis, prioritisation and planing of investment



R&D in Cybersecurity

Attacker

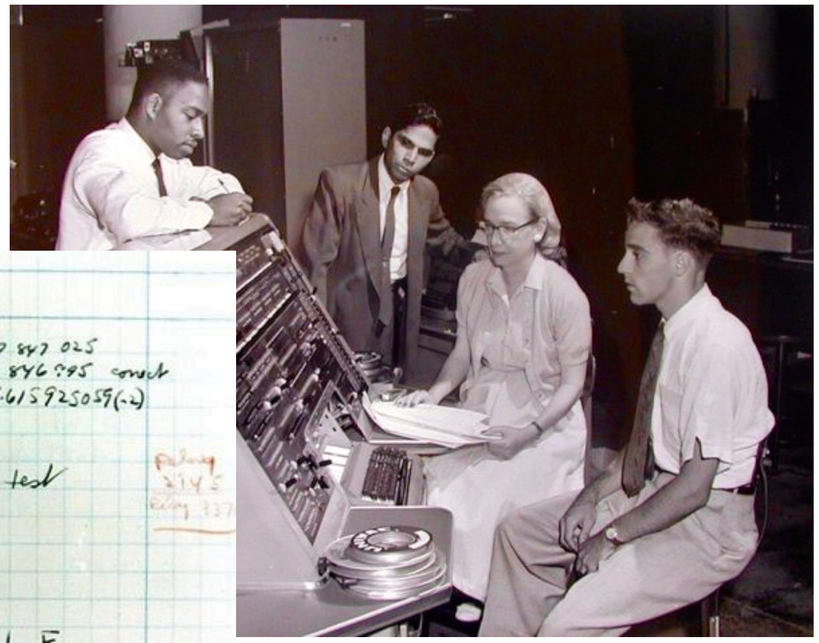
- May only need a single, exploitable vulnerability in any point of the system
- Cost of attack is often low
- Return of investment can be extremely high

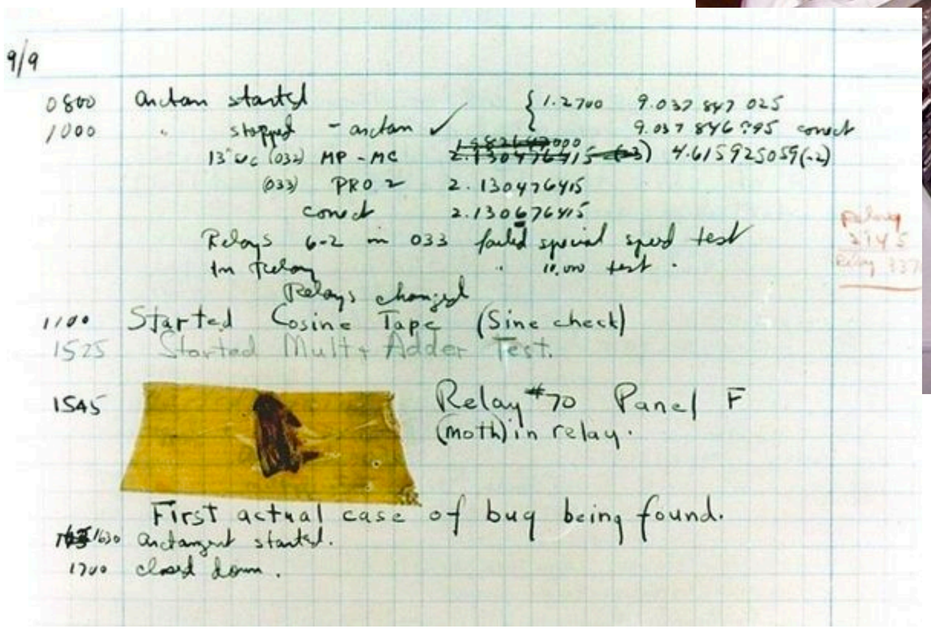
Levelling the field

- Lower the cost of preparedness
- Lower the impact of attack
- Rising the cost of attack
- Lower the RoI of attack

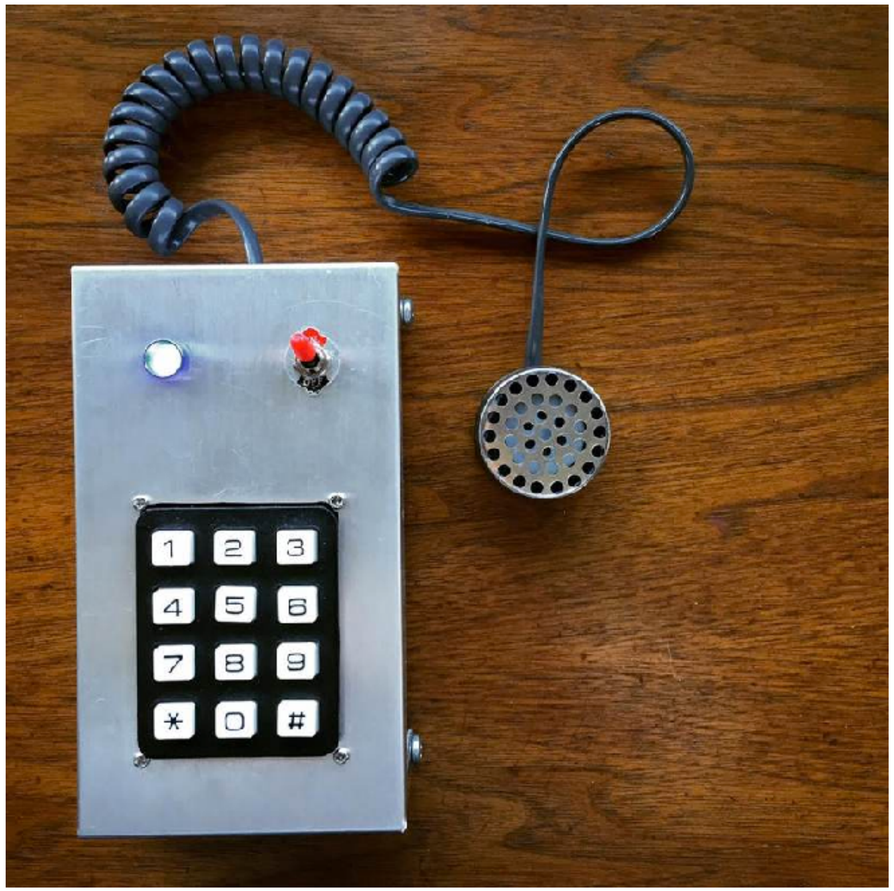
The first bug (1945)

- Grace Murray Hopper records a moth being found stuck between relay contacts of a Harvard Mark II computer. Hence the terms “bug” and “debugging”.

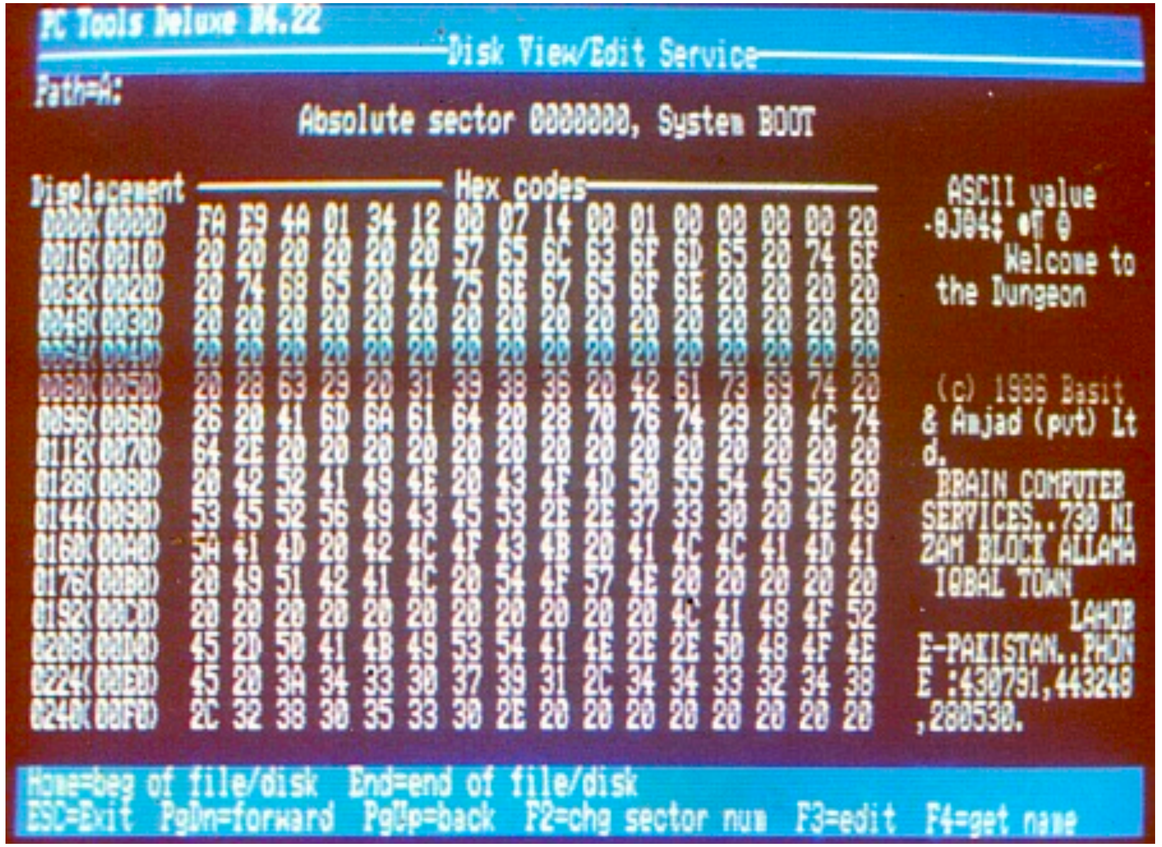




The “Phreaking” Era (1964 and 1972)



The first “worm” and “virus” (1979 and 1986)



The first network worm

USA, 1986

The first ransomware

USA, 1989

The first cyberwarfare

Estonia, 2007

The first highly complex attack

Stuxnet, 2012

Knowledge of global surveillance programs

Edward Snowden, 2013

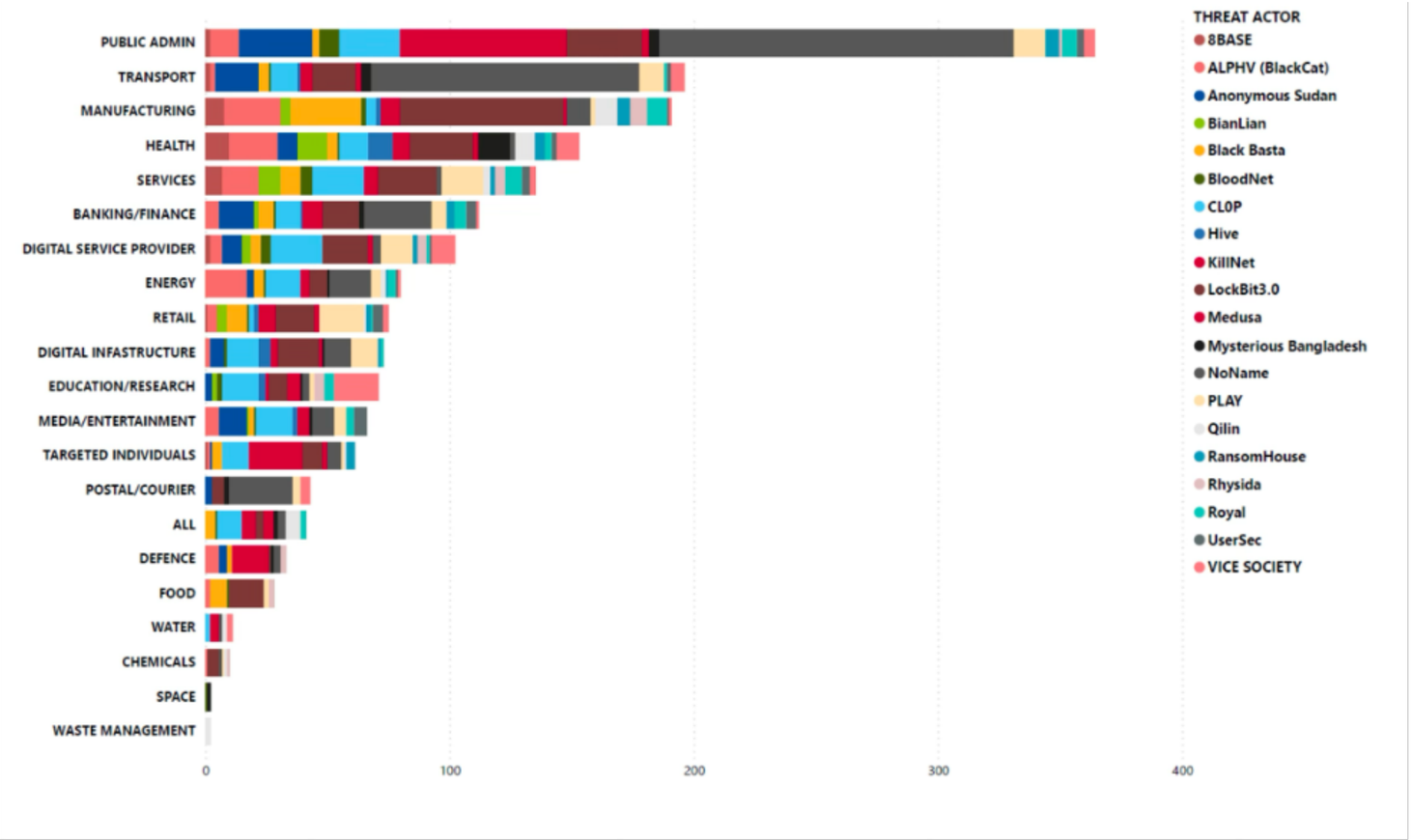
Social Engineering

Exploiting the Human Nature: Kevin Mitnick (1963-2023)

- Gained notoriety in the 1980s and 1990s for various high-profile hacking activities, including breaches of major corporations like Nokia, IBM, and Motorola.
- Became a fugitive in 1993 after being indicted on multiple charges related to computer crimes.
- Well-known for masterful exploitation of social engineering techniques (impersonation and rapport building) to manipulate individuals and gain access to secure systems and sensitive information.
- Captured by the FBI in 1995, leading to a highly publicized trial.
- Served five years in prison, including time in solitary confinement.
- After release in 2000, became a security consultant, author, and public speaker.

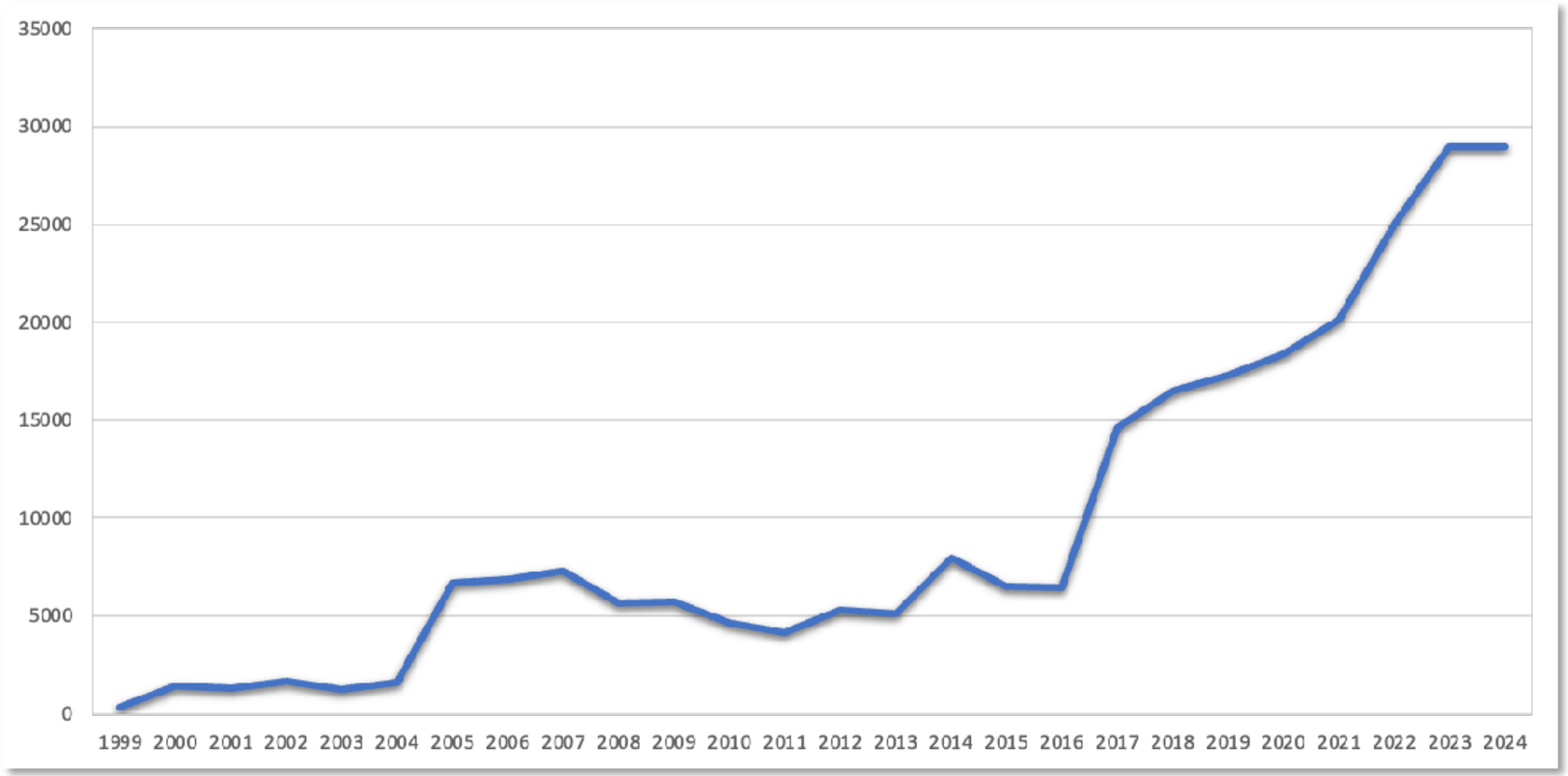
Advanced Persistent Threat (APT) Groups

Threat Actors per Sector (ENISA, 2023)



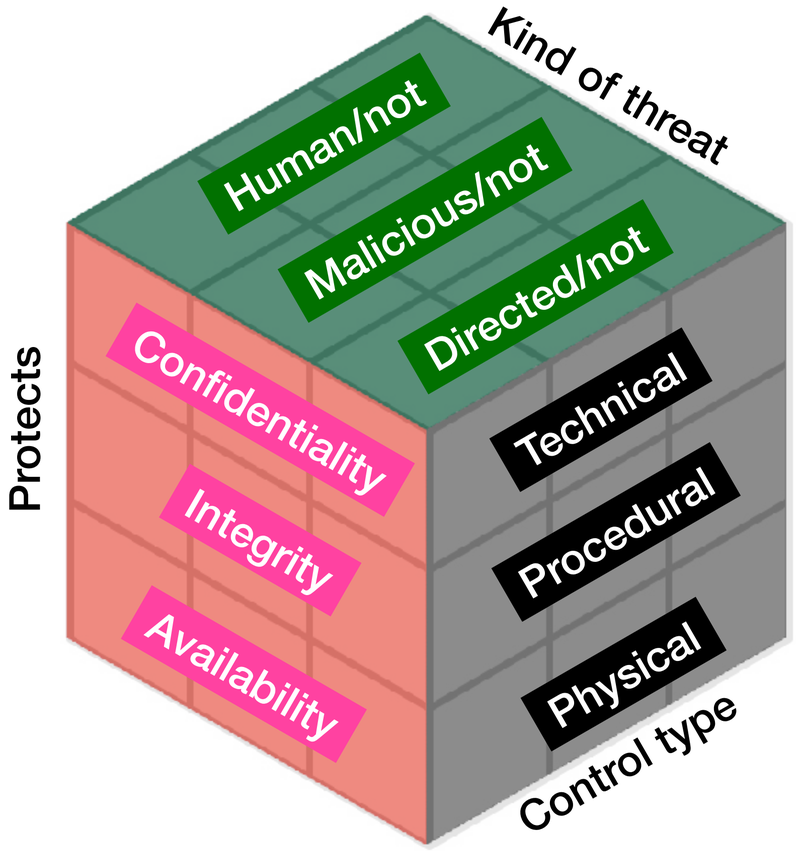
Source: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

Vulnerabilities per Year (CVE, Oct 2024)

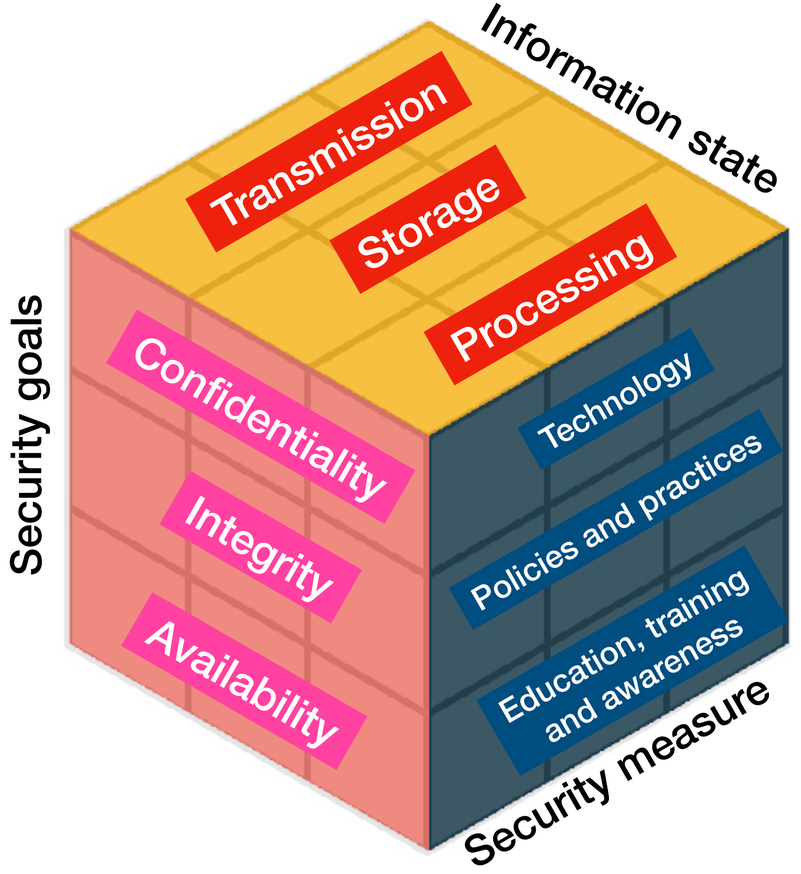


Source: https://www.cve.org/about/Metrics

Perspectives on Cybersecurity



Types of Countermeasures [Pfleeger et al 2015]



The Cybersecurity Cube [McCumber 1991]

A Glance at the most Common Software Weaknesses

Top 25 CWE 2021 (extract)

1. Out-of-bounds Write
2. Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
3. Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
4. Use After Free
5. Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
6. Improper Input Validation
7. Out-of-bounds Read
8. Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
9. Cross-Site Request Forgery (CSRF)
10. Unrestricted Upload of File with Dangerous Type

Source: https://cwe.mitre.org/top25/

Top 10 OWASP 2021 (focus on web-based software)

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

Source: https://owasp.org/www-project-top-ten/

Estimated Source Lines of Code (2016)

Software	#	Unit
Unix v1.0	10	thousand
Photoshop v1.0	100	thousand
Space Shuttle	400	thousand
F-22 fighter jet	1,7	million
Linux Kernel v2.2.0	2,0	million
Windows v3.1	2,5	million
Photoshop CS v6	4,5	million
DVD player on XBOX	4,7	million
Boeing 787	6,5	million
Windows NT v3.5	7,6	million
Windows NT v4.0	11,0	million

Software	#	Unit
Mozilla Core	12	million
Linux Kernel v3.1	15	million
F-35 fighter jet	24	million
Microsoft Office 2001	25	million
Microsoft Office 2013	45	million
Microsoft Windows Vista 2007	50	million
Facebook	62	million
MacOS v10.4 (Tiger)	86	million
Car software	100	million
Google (all services)	2	billion

Source: https://www.informationisbeautiful.net/vizualizations/million-lines-of-code/

Estimated Defects per Thousand Lines of Code

Can vary widely based on several factors, including the programming language, the complexity of the project, the experience of the developers, and the testing processes in place. However, a general guideline is as follows:

- Low defect density:
  - 0 to 1 defects/KLOC (highly stable or critical systems)
- Average defect density:
  - 1 to 10 defects/KLOC (common in many applications)
- High defect density:
  - 10+ defects/KLOC (often in rapidly developed or less rigorously tested systems)

From a technical standpoint, there is no satisfactory solution on the horizon.

“The world is never going to be perfect, either on- or offline; so, let’s not set impossibly high standards for online.”

— Esther Dyson

The Cryptography and Information Security Profile

Profile Overview

Objectives:

Master concepts, methodologies, processes and tools that support the development and operation of secure computer systems.

Curricular units:

Security Engineering

Security Technologies

Cryptographic Structures

Teaching methodology:

Lectures to introduce concepts

Theoretical & practical exercises to fix concepts & tools

Practical assignments aimed at solving real problems

Teaching team:

João Marco Cardoso Silva

José Carlos Bacelar Almeida

Vítor Francisco Gomes Fonte

Curricular Units

Security Engineering

Application vulnerabilities

Buffer overflow, input validation, race conditions

Software testing

Blackbox and whitebox testing, static and dynamic nalysis

Quality software development

Version control, source code quality, continuous integration

Secure software development life-cycle

Risk analysis, development standards and methodologies

Security Technologies

Security concepts

Properties, vulnerabilities, exploits, attacks, controls, layered security

Access control

Models, identification, authentication, local and distributed

Operating system and network security

Protection of resources, detection and reaction to intrusions

Security testing and information management

Security assessment methodologies, tools and practices, monitoring and protection of IT infrastructures

Cryptographic Structures

Creating and using environments with cryptographic operations

Groups, rings and finite bodies defined over integers

Rings and Bodies of polynomials

Finite bodies and elliptic curves

Reticulates and post-quantum cryptography

Symmetric ciphers

Application of the learning outcomes

Examples of topics that can be addressed in a dissertation:

- Security of digital identity systems
- Detection and prevention of intrusions
- Security models applied to complex systems

Research projects:

- IDINA — Non-Authoritative & Inclusive Digital Identity
- ISO/IEC 18013-5 mDoc-based research and development projects

Academy and industry:

- National and international cybersecurity market and eco-system

That’s it, thanks.

Hope to see you soon on the *Cryptography and Information Security* profile.