Sistemi Operativi e Reti di Calcolatori (SOReCa)

Corso di Laurea in *Ingegneria Informatica e Automatica (BIAR)*

Terzo Anno | Primo Semestre

A.A. 2024/2025

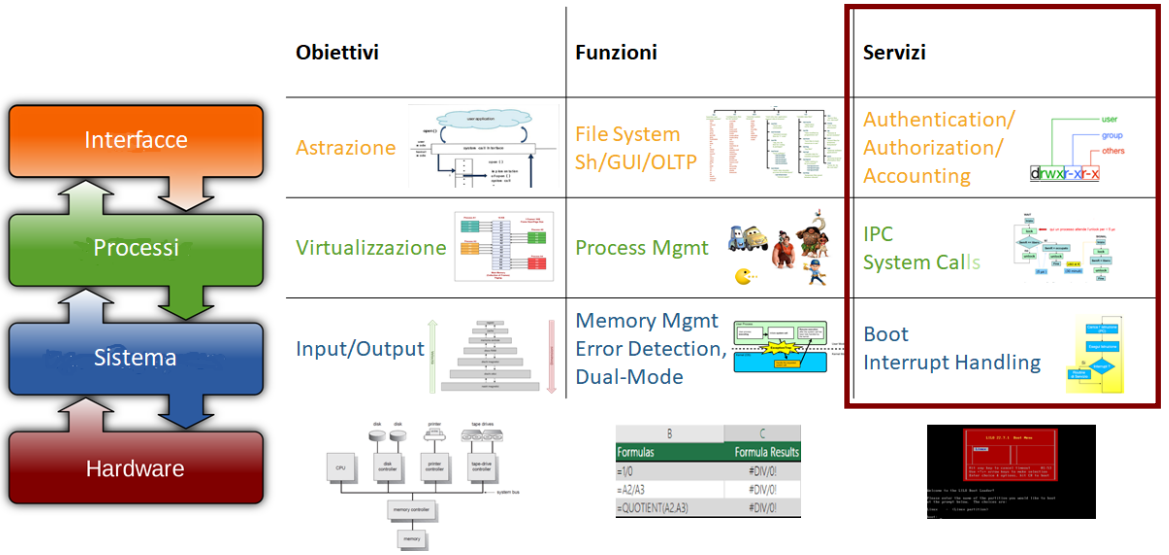
Sicurezza

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Sistemi operativi (3 CFU)

- Il sistema operativo
- Concorrenza e sincronizzazione
- Deadlock
- Inter-process communication (IPC)
- Scheduling
- Memoria centrale e virtuale
- Memoria di massa e File system
- Sicurezza informatica



Lezioni: Settembre - Ottobre

Dipartimento di Ingegneria informatica
automatica e gestionale Antonio Ruberti

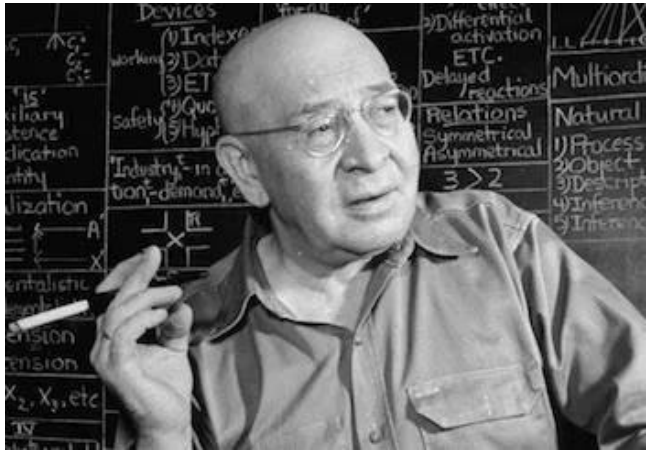
Sapienza
Università di Roma

Principi Fondamentali

Rischio, CIA Triad, PDCA (Shewart-Deming), NIST etc.

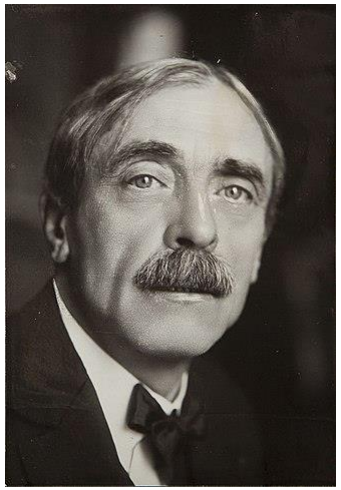
SOReCa - Sicurezza: Principi Fondamentali

The Main Law of Representation



- “la mappa non è il territorio”,
- “la parola non è la cosa”

[Alfred Korzybski]



- "Tutto quel che è semplice è falso."
- «Tutto ciò che è complesso è inutilizzabile"

[Paul Valery]

SOReCa - Sicurezza: Principi Fondamentali

The Main Law of Representation: sequential models for security

#	**Security**
1	Sicurezza
1	Measure: Risk
2	MOM: Vulns/Threats
3	Goal: CIA, People-Process-Technology
4	Shewart/Deming Cycle: PDCA (Plan Do Check Act) CObIT 4: PD, AI, DS, ME (Planning&Design, Acquire&Implement, Delivery&Support, Monitor&Evaluate)
5	ZTA: Pillars (Identity, EndPoint, Network, Workload, Data) COBIT 5: Processi (Govern,
6	function del NIST CSF 2.0 (Govern, Identify, Protect, Detect, Response, Recovery)
8	8 CyberSec Domains: Sec & Risk Management, Asset Sec, Sec Arch & Engineering, Comms/Net Sec, IAM-IAG, Sec Assessment & Testing, Sec Ops, SW Dev Sec
13	Fundamental Principles of IT Security (C I A + A A A + Least Privilege + Sec by Design + Defense in Depth + Incident Response + Education-Awareness + CMM + Compliance)
21	Design Shutters (modello per DevSecOps): PDCA x Biz-Arc-Compo-Code + DevOps + Govern

Sicurezza (1)

Obiettivo

SOReCa – Sicurezza (1)

Security: Definizione

Sicurezza: situazione in cui l'esistenza di una o più **misure elementari** rende minimo o nullo il **rischio** connesso al verificarsi di un danno temuto

[definizione militare]

SOReCa – Sicurezza (1)

Security: Risk

Rischio: eventualità di un danno comunque temuto e conseguente un qualsiasi evento negativo paventato

Rischio {e} = Danno {e} x Prob {e}

[definizione militare]

Risk Management

1. **Risk Avoidance**
2. **Risk Transfer**
3. **Risk Mitigation**
4. **Risk Acceptance**

[San Shi Liu Ji: 36° stratagemma]

Sicurezza (1b)

Analisi del Rischio

SOReCa – Sicurezza (1b): Analisi del Rischio

(why): risk of cyber-threats

Quantitative Risk == **ARO** x **SLE**

probability (ARO) of loosing money (SLE) from incidents or attacks (Threats) by exploiting 1+ vulnerability.

Usually, the security risk is calculated on an annual basis

The overall Risk is the combination of all the single impacts.

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Qualitative Risk (e.g. OWASP Risk Methodology)

ARO: **Annual Rate of Occurrence** → Likelihood (probability), external factor: **threat**
SLE: **Single Loss Expectancy** → Impact (money), internal factor: **vulnerability**

SOReCa – Sicurezza (1b): Analisi del Rischio

(who): risk of cyber-threats

Sun Tzu Ping Fa
Reduce Losses, Know Occurrences

"If you know the enemy (**ARO**) and know yourself (**SLE**), you need **not fear** the result of a hundred battles.

If you know yourself (**SLE**) but not the enemy (~~**ARO**~~), for every victory gained you will also **suffer a defeat**.

If you know neither the enemy (~~**ARO**~~) nor yourself (~~**SLE**~~), you will **succumb in every battle**."

(from ch. III "Attack by Stratagems", #18)

SLE ➔ Vulnerabilities: combination of Business and the 3 remaining layers ("Weaknesses", "Proactive Design" and "Defensive Coding".
ARO ➔ Threats: external factors

SOReCa – Sicurezza (1b): Official Birthday

(when): “Misure Elementari” - November 22°, 1988 (Morris Worm)

Floppy disk containing the source code for the Morris Worm, at the Computer History Museum

The **Morris worm** or **Internet worm of November 2, 1988**, is one of the oldest computer worms distributed via the Internet, and the first to gain significant mainstream media attention.

It resulted in the first felony conviction in the US under the 1986 Computer Fraud and Abuse Act.

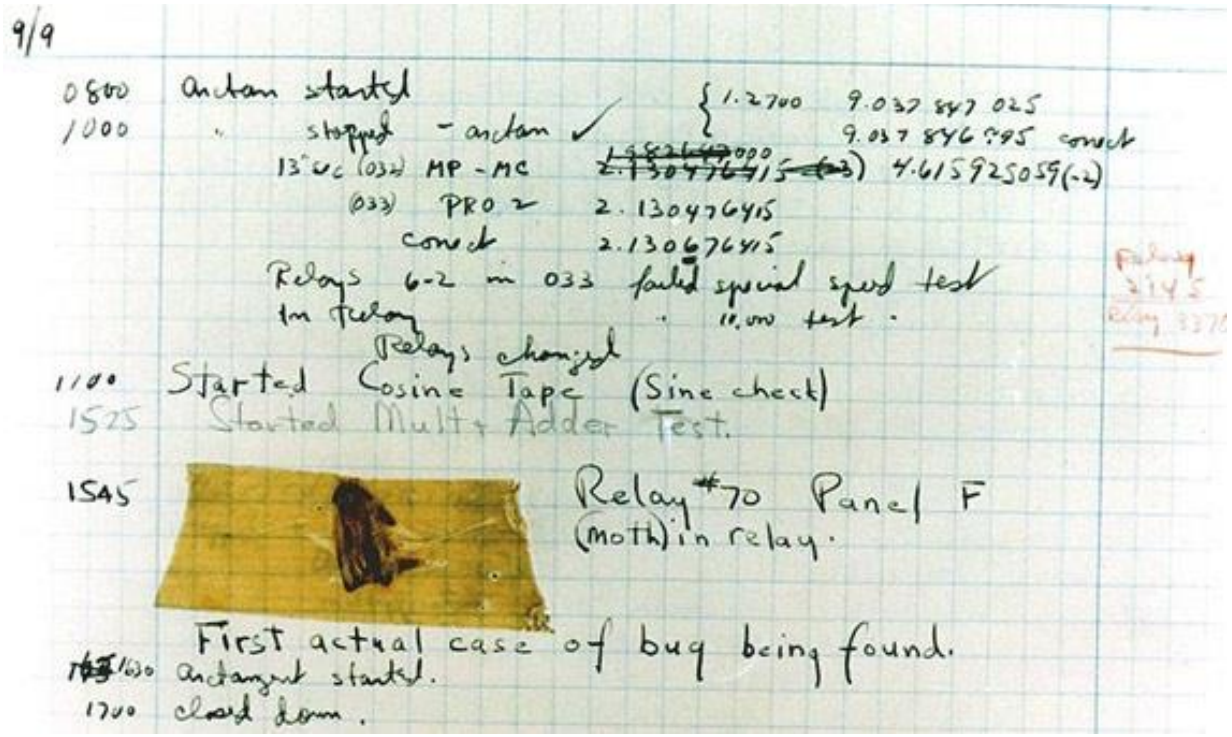
It was written by a graduate student at Cornell University, Robert Tappan Morris, and launched on 8:30 pm November 2, 1988, from the Massachusetts Institute of Technology network.

The worm exploited several vulnerabilities of targeted systems, including:

- A hole in the debug mode of the Unix sendmail program
- A buffer overflow or overrun hole in the finger network service
- The transitive trust enabled by people setting up network logins with no password requirements via remote execution (rexec) with Remote Shell (rsh), termed rexec/rsh

SOReCa – Sicurezza (1b): Official Birthday

(how): “Misure Elementari” - September 9th, 1947 “First actual case of bug”



The first bug (Source: Naval Historical Center Online Library Photograph)

The causes of security breaches are varied, but many of them owe to a defect (or **bug**) or design flaw in a targeted computer system's software.

After finding a moth inside the Harvard Mark II computer on September 9th, 1947 at 3:45 p.m., Grace Murray Hopper logged the first computer bug in her log book.

She wrote the time and the sentence: “First actual case of bug being found”.

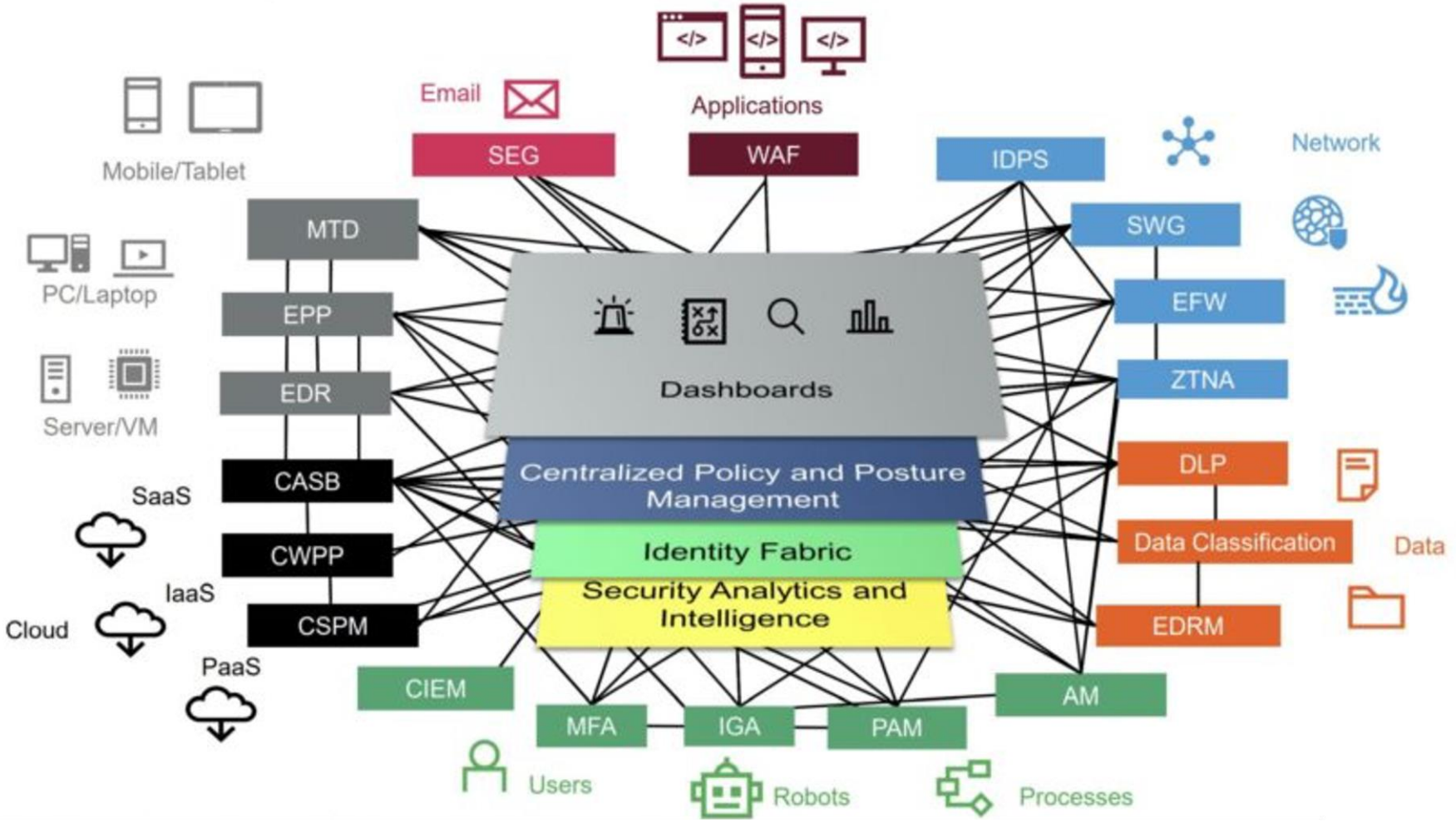
Nowadays, the term “bug” in computer science is not taken literally, of course. We use it to talk about a flaw or failure in a computer program that causes it to produce an unexpected result or crash.

SOReCa – Sicurezza (1b): Architetture

(where): “Misure Elementari” - designing safer integration

Nowadays application software should guarantee interoperability, that is the ability to communicate and share information about cybersecurity.

No more silos: every component is part of a bigger infrastructure, giving some service and obtaining some other back.



Gartner CSMA: Cyber Security Mesh Architecture

SOReCa – Sicurezza (1b): SW Composition

(what): “Misure Elementari” - removing exploitable defects in software and libraries

MITRE: from Cold-War era

“MITRE began in 1958, sponsored by the U.S. Air Force to bridge across the academic research community and industry to architect the Semi-Automatic Ground Environment, or SAGE, a key component of Cold War-era air defense. We were founded as a not-for-profit company to serve as objective advisers in systems engineering to government agencies, both military and civilian.
We are innovators—from advances in radar technology, cyber, GPS, cancer research, and aviation collision-avoidance systems to breakthroughs in evolving disciplines such as vehicle autonomy, artificial intelligence, and synthetic biology.
Moreover, as a company that doesn’t compete with industry, we’re uniquely positioned to convene government, industry, and academia to collaborate on big societal challenges, from pandemic response to highway safety to social justice.
At its core, MITRE’s story is about our people. We’re proud that more than 9,000 multi-talented and creative individuals choose to stand with us every day, dedicating themselves to our mission of solving problems for a safer world.”

SOReCa – Sicurezza (1b): SW Composition

(what): “Misure Elementari” - removing exploitable defects in software and libraries

MITRE: federal research – CVE, CWE, CPE

(Interestingly, MITRE is not an acronym, though some thought it stood for Massachusetts Institute of Technology Research and Engineering. The name is the creation of James McCormack, an early board member, who wanted a name that meant nothing, but sounded evocative.)

“We discover. We create. We lead.
MITRE is trusted to lead—by government, industry, and academia.
The bedrock of any trusted relationship is integrity. For more than 60 years, MITRE has proudly operated federally funded research and development centers, or FFRDCs. We now operate six of the 42 FFRDCs in existence—a high honor.
Since our inception, MITRE has consistently addressed the most complex whole-of-nation challenges that threaten our country’s safety, security, and prosperity. Our mission-driven teams bring technical expertise, objectivity, and an interdisciplinary approach to drive innovation and accelerate solutions in the public interest.
Above all, MITRE is trusted to deliver data-driven results and recommendations without any conflicts of interest.”

MOM (2): Methods, Opportunity, Means

Threats – Vulnerabilities

SOReCa - Sicurezza: MOM

Sicurezza: Rischio – Threats, Vulns, Risk, Countermeasures

Sicurezza

1. **Threats:** minaccia di reato ➔ **MOM**
   - **Motive**: movente
   - **Opportunity**: disponibilità di risorse/presenza di vulnerabilità
   - **Means**: capacità di commettere il crimine ➔ (vulnerability, ability)

2. **Vulnerability** (**Opportunity**)**:**
   - **Configuration ➔ Security Hygiene, Secure Architecture**
   - **Software Errors ➔ Patching** (avoiding **CVE**)

3. **Risk: Impatto (€) x Prob (Minaccia) / anno**
   - **Impatto**: perdita derivante dal danneggiamento
   - **Probabilità di Minaccia**: desunta, in generale, dai valori storici

4. **Countermeasures ➔ Protezione**
   - **Costo** di implementazione/Manutenzione (€)
   - **Riduzione** del Rischio

Threat: Something that can damage or destroy an asset

Vulnerability: A weakness or gap in your protection

Risk: Where assets, threats, and vulnerabilities intersect

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

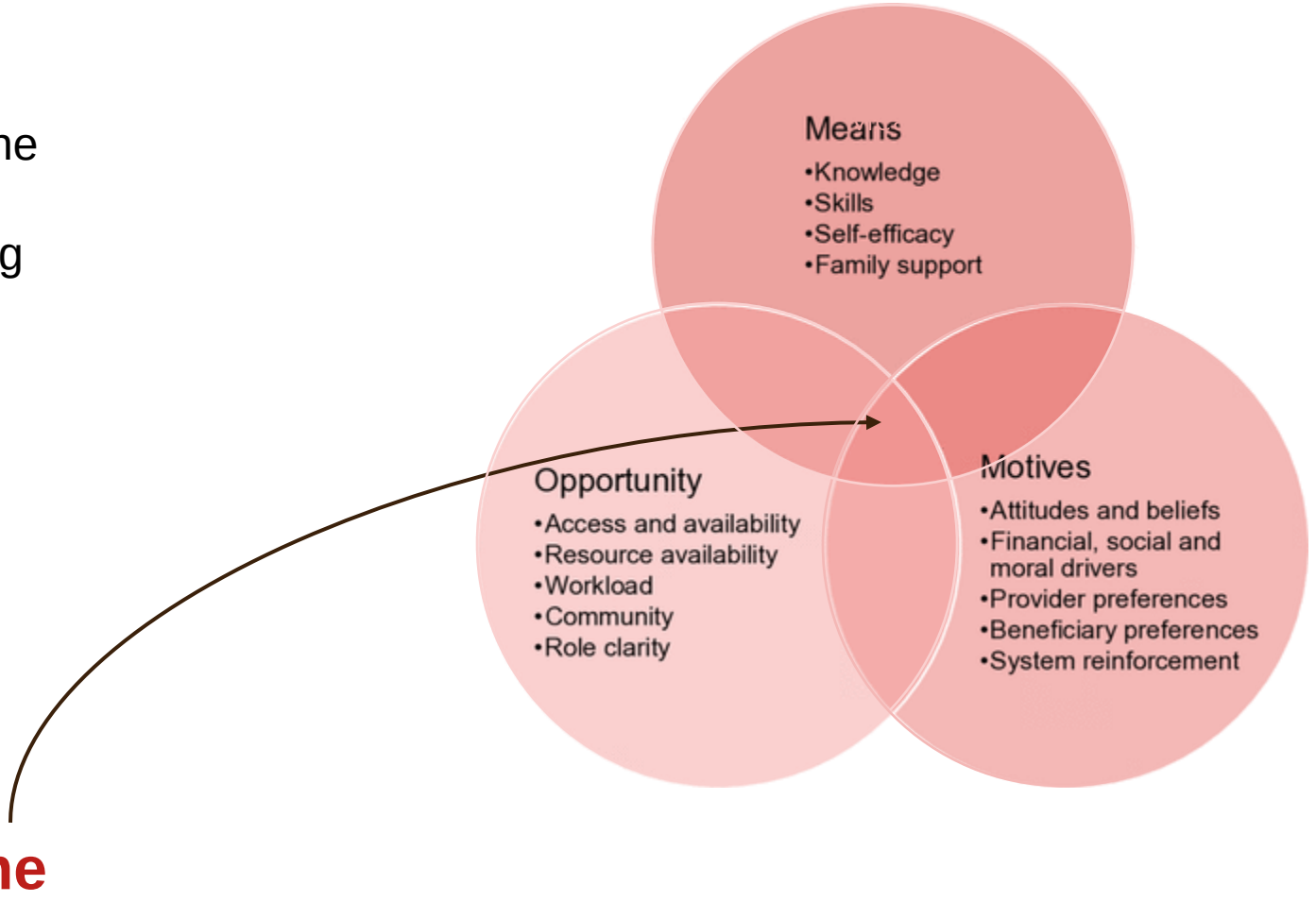
SOReCa - Sicurezza: MOM

Motive, Opportunity, Means 1/4

- **Motive:** a reason/ motivation to commit the crime. Reasons (Goals) that motivate the threat agents towards that option, including interests, values and so-on.

- **Opportunity:** adequate chance(s) to commit the crime.

- **Means:** the ability and tools necessary to commit the crime. The right skills for performing the violation.



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa - Sicurezza: MOM

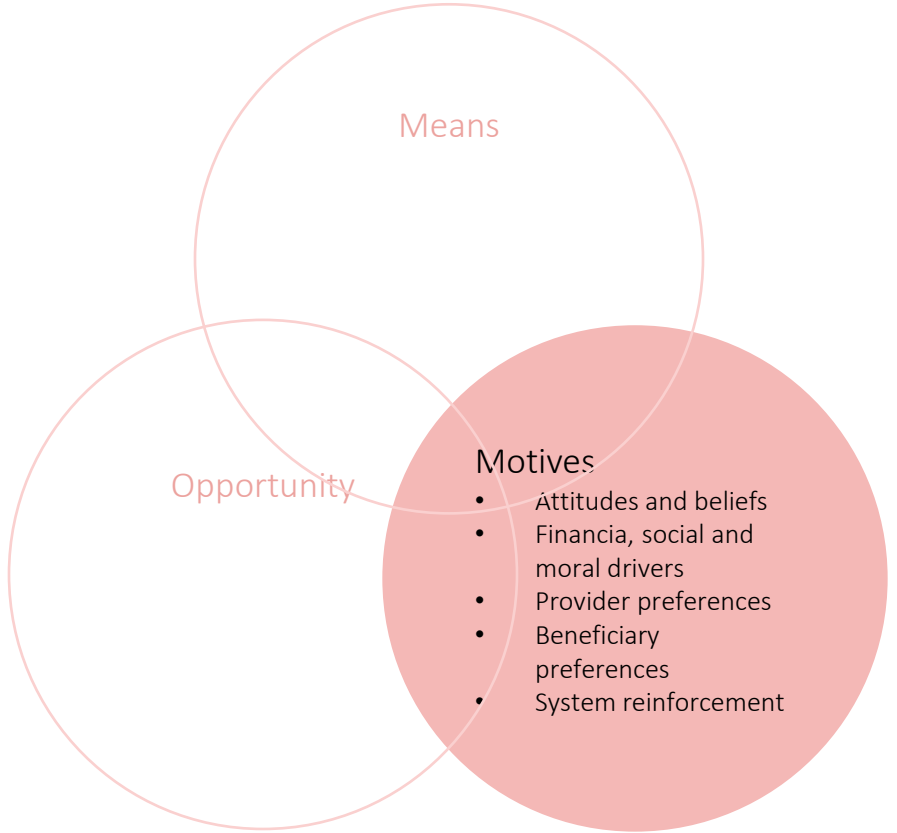
MOM: Motive, Opportunity, Means 2/4

•**Motive:**a reason/motivation to commit the IT crime

- **Intruding**: access system to steal.

- **Profiteering**: access to system, in order to hidely operate through.

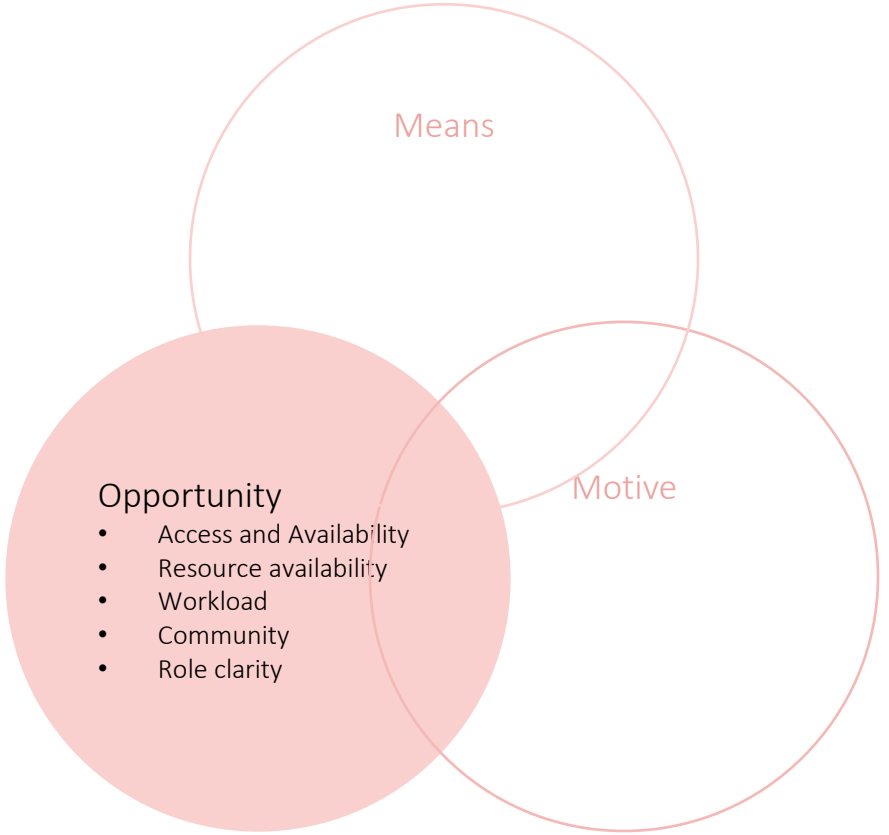
- **Damaging**: make the system unaccessible by anyone.



SOReCa - Sicurezza: MOM

MOM: Motive, Opportunity, Means 3/4

Opportunity: adequate chance(s) to commit the crime



Vulnerability: debolezza del sistema (ferita) → **Opportunity**

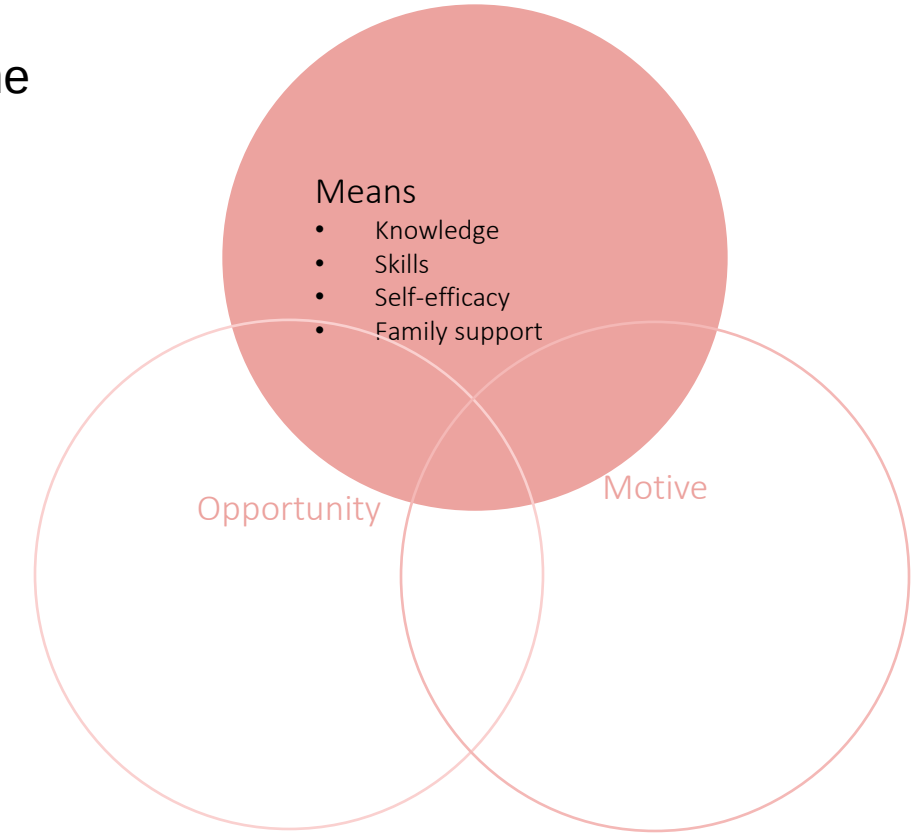
Threat: attore con obiettivi ben determinati (squalo) → **Motive**, **Means**

SOReCa - Sicurezza: MOM

MOM: Motive, Opportunity, Means 4/4

•**Means:** the ability and tools necessary to commit the crime. The right skills for performing the violation.

- **Wannabe Lamer (Script Kiddie)**
- **Cracker**
- **Ethical Hacker**
- **Industrial Spy**
- **Cyber Warrior (Quiet, Paranoid, Skilled Hacker)**
- **Government Agency (Military Hacker)**



DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

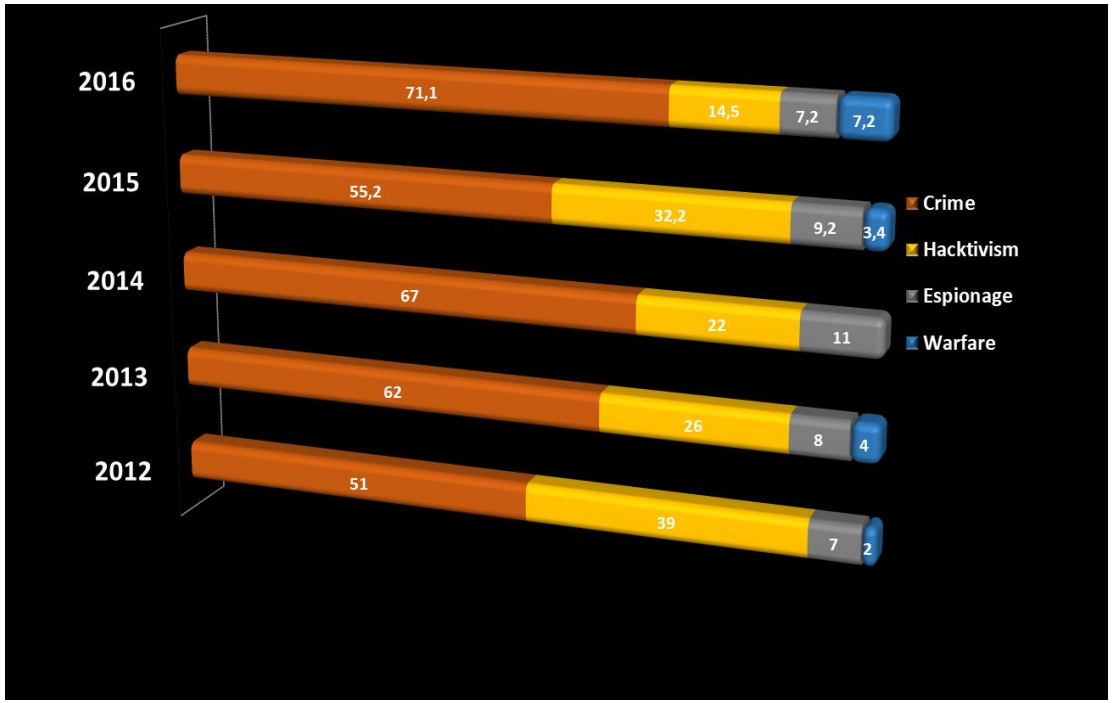
SAPIENZA
UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: MOM

Motive: Attaccanti e loro scopi – Composizione percentuale negli anni

Team		FBI name	Goal
Unstructured		Insider	Money
Structured		Crime	Money
		Espionage	Information
		Hactivism	Socio-Politics
National		Warfare	War
		Terrorism	War

Squali: Threat Actors



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa - Sicurezza: MOM

Motive: Attaccanti e loro scopi

Risk: Rischi tipici

- **Intruding**: access system in order to:
  - **Steal Money** (Banks and Finance)
  - **Read User Info** (e.g. Control the performed actions)
  - **Steal Company Info** (e.g. Patented Material, Royalties, Control the performed actions)

- **Profiteering**: access to system, in order to hidely operate:
  - **Spam**: perform Phishing and Advertising through Elaboration facilities
  - **DDoS**: network capacities (to 3° parties: for preparing Wasting activities)

- **Damaging**: make the system unaccessible from anyone
  DDoS: overwhelming the systems with requests
  Defacement: overwrite the siste pages
  Brake: Services and Systems

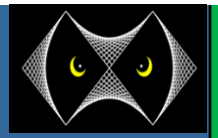
DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: MOM

Motive: Attaccanti e loro scopi

MOM-Risk: esemplificazione di associazione Threat-Risk generale

	 **Crime**	 **Hacktivism**	 **Warfare**	 **Espionage**
 **Intruding**	Steal Money Read User-Info	Steal Company Info	Steal Company Info	Steal Company Info
 **Profiteering**	Spam DDoS (3° party)			
 **Damaging**	DDoS (competitors)	Defacement	Break System	
	71%	15%	7%	7%

Cyber Threats are aimed by the **Main Purposes**:

- **Crime**: (in)direct immediate **economical advantages**. Habitual Victims are final users of web functionalities and services, sharing *payment* information or *customer profiling*
- **Hacktivism**: information theft, in order to pursue **social and political goals**. Habitual Victims are newspapers, government, liable of *image damage*
- **Espionage**: information theft, aimed at **industrial intelligence**. Less common issue but raising more perseverance and technical refinement. Habitual Victims are intended factories and firms. It could be a special case of Crime if aimed at *payment* or *customer profiling*
- **Warfare**: war events between countries, aimed at Critical Infrastructure. Niche phenomena, too complex to be treated here

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: MOM

Motive: Attaccanti e loro scopi

Attacchi agli Endpoint: esemplificazione delle finalità di infezione

Finalità	
BotNet	rete di computer compromessi da malware e comandati a distanza per scopi illegali. Si entra a far parte di una botnet inconsapevolmente quando il proprio computer non è adeguatamente protetto ed aggiornato. Le botnet costituiscono una minaccia insidiosa in quanto un'infezione può rimanere a lungo non rilevata e silente per essere sfruttata successivamente per produrre danni ingenti a sistemi di terze parti
Ramsonware	Limitazione dell'accesso al dispositivo infettato, richiedendo un riscatto (ransom) da pagare per rimuoverla
Tailored	Insieme di processi di hacking informatici furtivi e continui, appositamente orchestrati per mirare ad una specifica entità, danneggiando solo i sistemi dotati di particolari requisiti

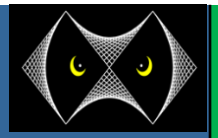
Infezione: processo di assoggettamento di un sistema, perpetrato in uno dei seguenti modi:

1. Phishing: apertura di email infette o di documenti ad essi allegati
2. Malware: nascosto in programmi scaricati dagli utenti (es. crack), volto a turbare il normale funzionamento di un sistema
3. Known Vulnerabilities: sfruttamento di vulnerabilità specifiche dei sistemi e delle applicazioni non aggiornati

SOReCa - Sicurezza: MOM

Motive: Attaccanti e loro scopi

Attacchi agli Endpoint: mappatura tra tecniche di infezione e tipologie di attori di minaccia

	 Crime	 Hacktivism	 Warfare	 Espionage
 **Intruding**	Ramsonware Tailored		Tailored	Tailored
 **Profiteering**	BotNet			
	86%		5%	9%

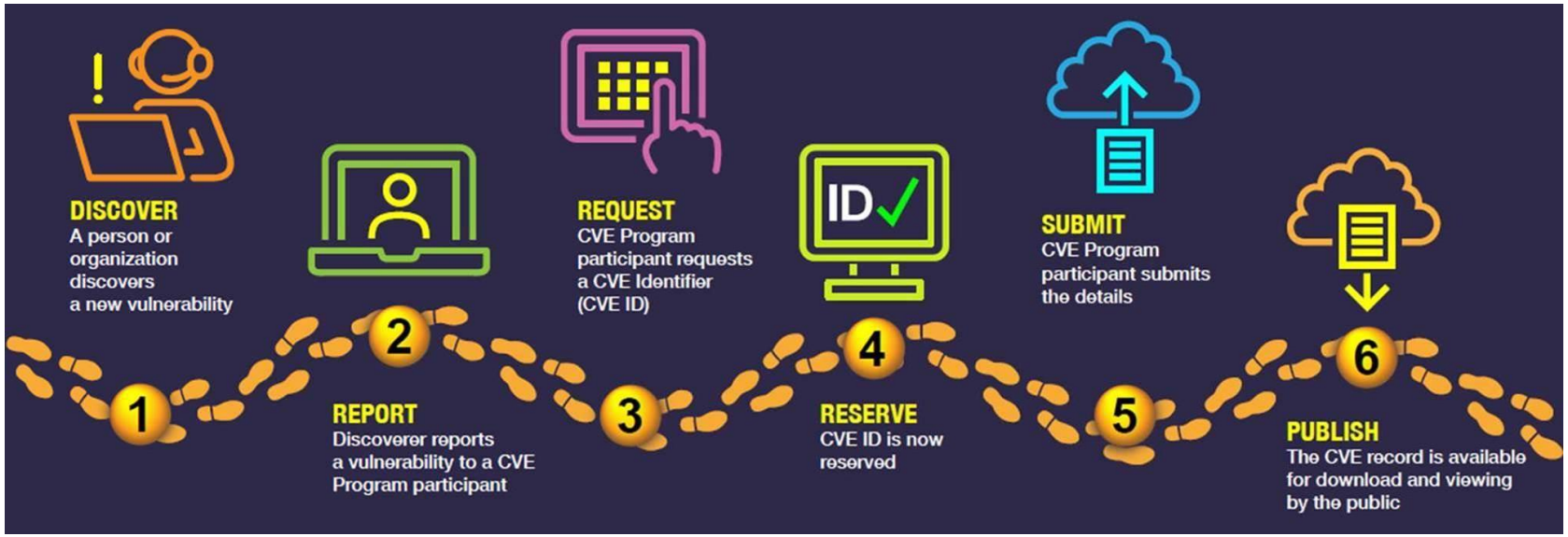
SOReCa - Sicurezza: MOM

Opportunity: Vulnerability

Vulnerability: flaws or glitches that weaken a system's overall security. Vulnerabilities can be weaknesses in either the hardware itself or the software that runs on it. Exploitable by threat actors.

1. **Discover**: A person or organization discovers a new vulnerability.
2. **Report**: Discoverer reports a vulnerability to a CVE Program participant.
3. **Request**: CVE Program participant requests a CVE Identifier (CVE ID)
4. **Reserve**: The ID is reserved, which is the initial state of a CVE Record. The Reserved state means that CVE stakeholder(s) are using the CVE ID for early-stage vulnerability coordination and management, but the CNA is not yet ready to publicly disclose the vulnerability.
5. **Submit**: CVE Program participant submits the details. Details include but are not limited to affected product(s); affected or fixed product versions; vulnerability type, root cause, or impact; and at least one public reference.
6. **Publish**: Once the minimum required data elements are included in the CVE Record (**usually 30 days**), it is published to the CVE List by the responsible CNA. The CVE Record is now available for download and viewing by the public.

CVE Record Lifecycle

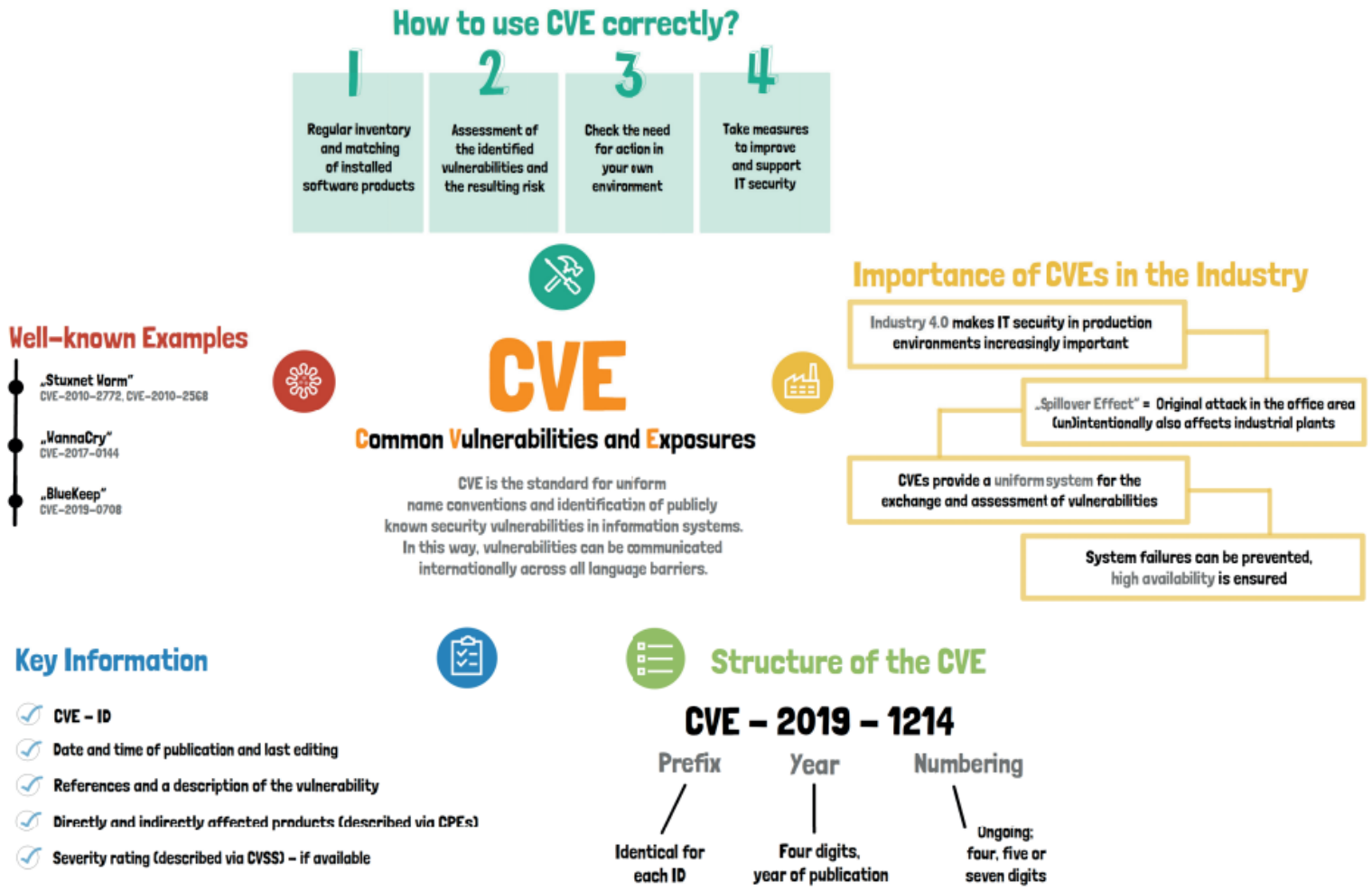


Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa - Sicurezza: MOM

Opportunity: Common Vulnerabilities and Exposures



Vulnerability Management: continuous, proactive, and often automated process that keeps computer systems, networks, and enterprise applications safe from cyberattacks and data breaches.

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa - Sicurezza: MOM

Means: Capabilities, Skills, Profiles

Hacker (Cracker) Generations: thinks change

- **First Generation** (70's): inspired by the **need of knowledge** (by accessing proprietary and hidden information for free). First attempts of **Phracking**, since the need of telephone for free (issuing digital communications).
- **Second Generation** (80's): driven by **curiosity** mixed with knowledge starving: the only way to learn **OSs** was **by hacking** them; **later**, hacking become a **trend**.
- **Third Generation** (90's): pushed by the anger of hacking (mix of **addiction**, **curiosity** for «*the new world coming*», **learning** «*the new stuff*», interaction with the **underground community**: demonstration of capability, information exchange). New concepts were coming like hacker e-zines (Phrack, 2600 Magazine) along BBS (Bullettin Board System).
- **Fourth Generation** (00's): driven by **angerness** and, mainly, **money** (Cracking is identified as «*cool*») . Web 2.0: many violations could be executed at only **Application Level**, interaction with the OS could not be needed. Many subjects with very low know-how, even if actively involved in the criminal workd (**cybercrime**), taking advantages from BotNet (roBot on Networks), using command and control (C&C) software paradigm.
- **Fifth Generation** (10's – today): met by politics (**cyber-hactivism**, **terrorism** , **warfare**). Strong use of automatic tools, enabling the Advanced Threat (**malware**), capable of self-spreading. Counteraction by ATP (**Advanced Threat Prevention**) or APT (Advanced Persisten Threat) if the malware is funded by a State (e.g. StuxNet).
- **Sixth Generation** (20's): emerging of **Fileless** as mainstream type of attack.

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: MOM

Means: Capabilities, Skills, Profiles

Profile	OFFENDER ID	LONE / GROUP HACKER	TARGET	MOTIVATIONS / PURPOSES
Wanna Be Lamer	9-16 years "I would like to be a hacker, but I can't"	GROUP	End-User	For fashion, It's "cool" => to boast and brag
Script Kiddie	10-18 years The script boy	GROUP: but they may act alone	SME / Specific security flaws	To give vent of their anger / attract mass-media attention
Cracker	17-30 years The destructor, burned ground	LONE	Business company	To demonstrate their power / attract mass-media attention
Ethical Hacker	15-50 years The "ethical" hacker's world	LONE / GROUP (only for fun)	Vendor / Technology	For curiosity (to learn) and altruistic purposes
Quiet, Paranoid, Skilled Hacker	16-40 years The very specialized and paranoid attacker	LONE	On necessity	For curiosity (to learn) => egoistic purposes
Cyber-Warrior	18-50 years The soldier, hacking for money	LONE	"Symbol" business company / End-User	For profit
Industrial Spy	22-45 years Industrial espionage	LONE	Business company / Corporation	For profit
Government Agent	25-45 years CIA, Mossad, FBI, etc.	LONE / GROUP	Government / Suspected Terrorist/ Strategic company/ Individual	Espionage/ Counter-espionage Vulnerability test Activity-monitoring
Military Hacker	25-45 years	LONE / GROUP	Government / Strategic company	Monitoring / controlling / crashing systems

Hacker Profiling Project: https://www.slideshare.net/slideshow/def-camp-2013-day-1-key-note-raoul-chiesa/29329573

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: MOM

Means: Usuals

Infection	process of subjecting a system, perpetrated in one of the following ways:
Phishing	opening infected emails or documents attached to them
Malware	hidden in programs downloaded by users (e.g. cracks), aimed at disturbing the normal functioning of a system
Known Vulnerabilities	exploit specific vulnerabilities of out-of-date systems and applications

Goals (3): CIA, PPT, TTP

CIA: Confidentiality-Integrity-Availability

PPT: People-Process-Technology

TTP: Tactiques, Techniques, Procedures

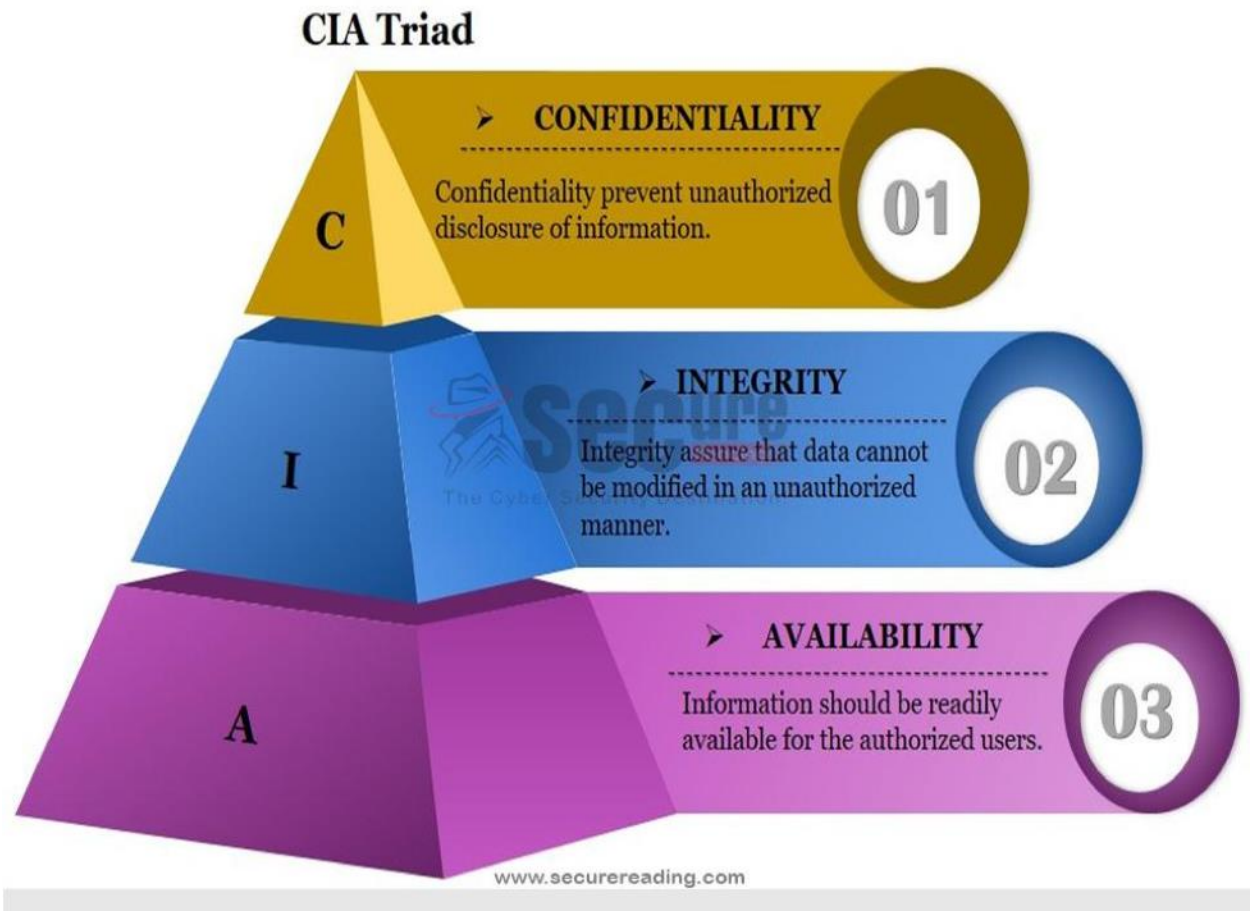
SOReCa - Sicurezza: Goals (3)

Principi Fondamentali: CIA (RID)

Riservatezza

Integrità

Disponibilità



SOReCa - Sicurezza: Goals (3)

Principi Fondamentali: CIA (RID)

➔ **Confidentiality** (Riservatezza): le informazioni possono essere accessibili in lettura solo ai corretti destinatari (➔ Data Protection, Privacy, Confidentiality)

➔ **Integrity** (Integrità): le informazioni possono essere accessibili in scrittura solo ai corretti operatori (➔ Data Quality: Prevention of Data Corruption)

➔ **Availability** (Disponibilità): le informazioni devono essere accessibile in lettura/scrittura a tutti i soggetti previsti (➔ Resilience: Data Duplication)

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

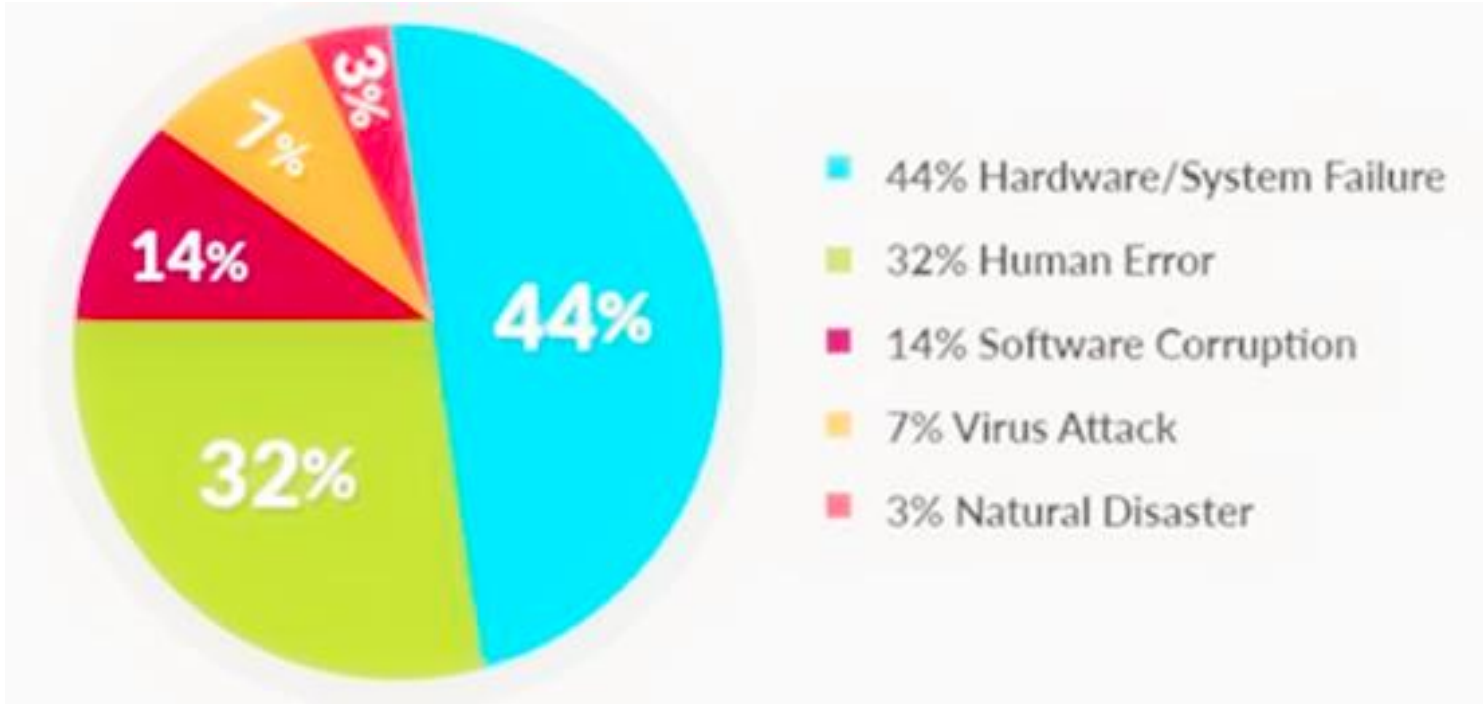
Sapienza Università di Roma

SOReCa - Sicurezza: Goals (3)

Availability: Perdita di Dati Accidentale

Cause comuni di perdita accidentale di dati:

- **HW/System Failure**: malfunzionamenti della CPU,dischi o nastri illeggibili, errori di telecomunicazione,
- **Human Error**: immissione dati errata, nastro o CD-ROM erroneamente montati, esecuzione errata del programma, disco o nastro perso, oppurequalche altro errore.
- **Software Corruption**: bug dei programmi.
- **Virus Attack**: attacco non finalizzato al riscatto
- **Natural Disaster**: incendi, inondazioni, terremoti, guerre, sommosse o topi che rosicchiano nastri di backup



SOReCa - Sicurezza: Goals (3)

Confidentiality ➔ Bell-La Padula Model 1/3

Confidentiality (Confidenzialità/Riservatezza): le informazioni possono essere accessibili in lettura solo ai corretti destinatari (➔ Data Protection, Privacy)

➔ Modello Bell-La Padula: 2 regole (proprietà)

1. **No Read Up** (Simple Security Property): Un processo in esecuzione a livello di sicurezza k può leggere solo oggetti al suo livello o inferiore

2. **No Write Down** (* Property): Un processo in esecuzione al livello di sicurezza k può scrivere solo oggetti al suo livello o superiore

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

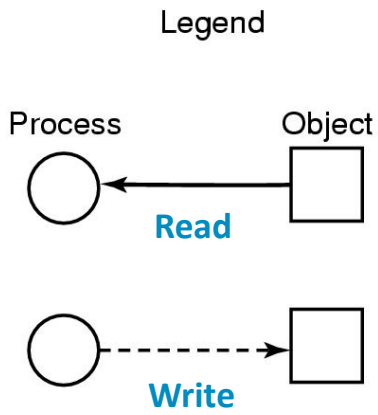
SOReCa - Sicurezza: Goals (3)

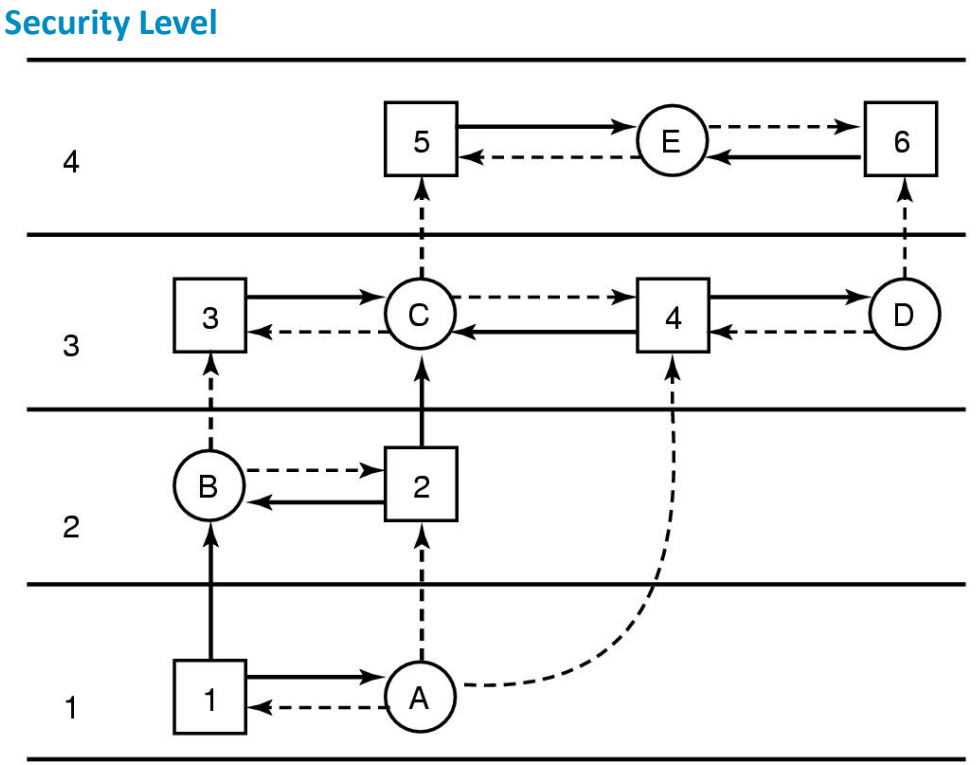
Confidentiality ➔ Bell-La Padula Model 2/3

Modello Bell-La Padula: 2 regole (proprietà)

1. **No Read Up** (Simple Security Property ➔ non leggere informazioni potenzialmente più confidenziali

2. **No Write Down** (* Property) ➔ non scrivere inavvertitamente informazioni più confidenziali





Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa - Sicurezza: Goals (3)

Confidentiality ➔ Bell-La Padula Model 3/3

- Cosa facciamo
- L'intelligence
- Collaborazione istituzionale
- Rapporti con l'Autorità giudiziaria
- Tutela delle informazioni
  - Autorità nazionale per la sicurezza
  - Il segreto di Stato
  - Classifiche di segretezza
  - Rilascio delle abilitazioni di sicurezza
- I controlli sul Sistema

Home » Cosa facciamo » Tutela delle informazioni » Classifiche di segretezza

Classifiche di segretezza

La classifica di segretezza è l'**indicatore del livello di segretezza** attribuito in ambito nazionale a una determinata informazione. Si configurano come **documenti classificati** qualsiasi supporto – materiale o immateriale, analogico o digitale – contenente informazioni classificate e, pertanto, sottoposto a misure di protezione fisica, logica e tecnica dal momento della sua origine fino a quello della sua distruzione o declassifica. Durante tale arco di vita, la sua trattazione e gestione sono disciplinate da modalità specifiche. Le singole parti di un documento possono richiedere classifiche differenti. In questo caso il livello generale di classifica dell'intero documento è pari almeno a quello della parte con classifica più elevata.

Le classifiche sono quattro:

- **segretissimo** (SS)
- **segreto** (S)
- **riservatissimo** (RR)
- **riservato** (R)

Rep. Italiana	NATO
Segretissimo (**SS**)	Top Secret
Segreto (**S**)	Secret
Riservatissimo (**RR**)	Confidential
Riservato (**R**)	Reserved

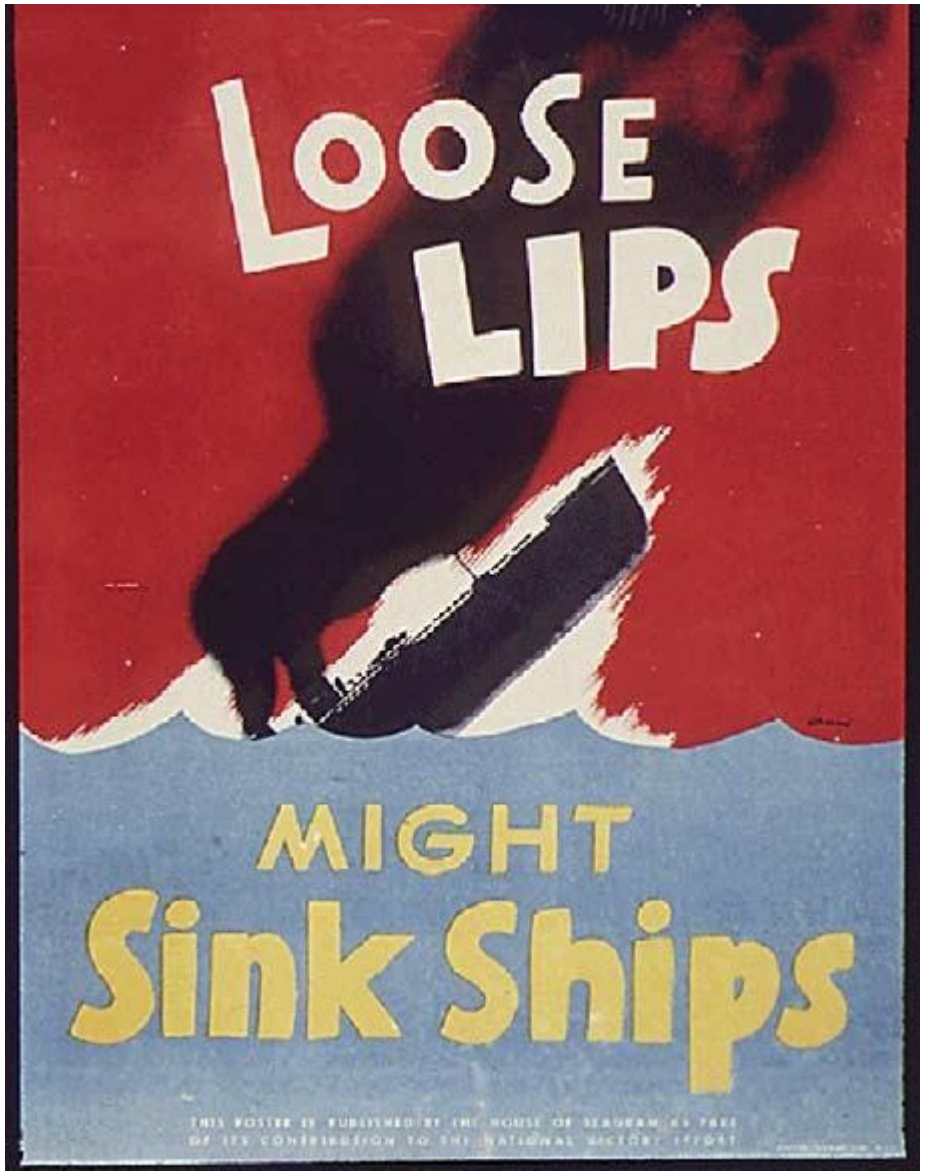
➔ Nulla Osta di Sicurezza (NOS) ➔ Livello (R, RR, S, SS)

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: Goals (3)

Integrity ➔ Biba Model 1/2



Integrity (Integrità): le informazioni possono essere accessibili in scrittura solo ai corretti operatori (➔Data Quality: Prevention of Data Corruption)

➔ Modello Biba: 2 regole (proprietà)

1. **No Write Up** (Simple Integrity Principle): Un processo in esecuzione al livello di integrità k può scrivere solo oggetti al suo livello o inferiore ➔ non inserire informazioni meno integre

2. **No Read Down** (Integrity * Property): Un processo in esecuzione a livello di integrità k può leggere solo oggetti al suo livello o superiore ➔ non utilizzare informazioni meno integre

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

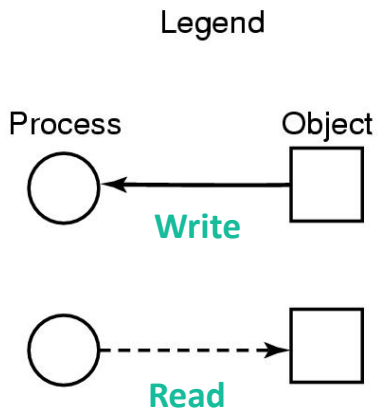
SOReCa - Sicurezza: Goals (3)

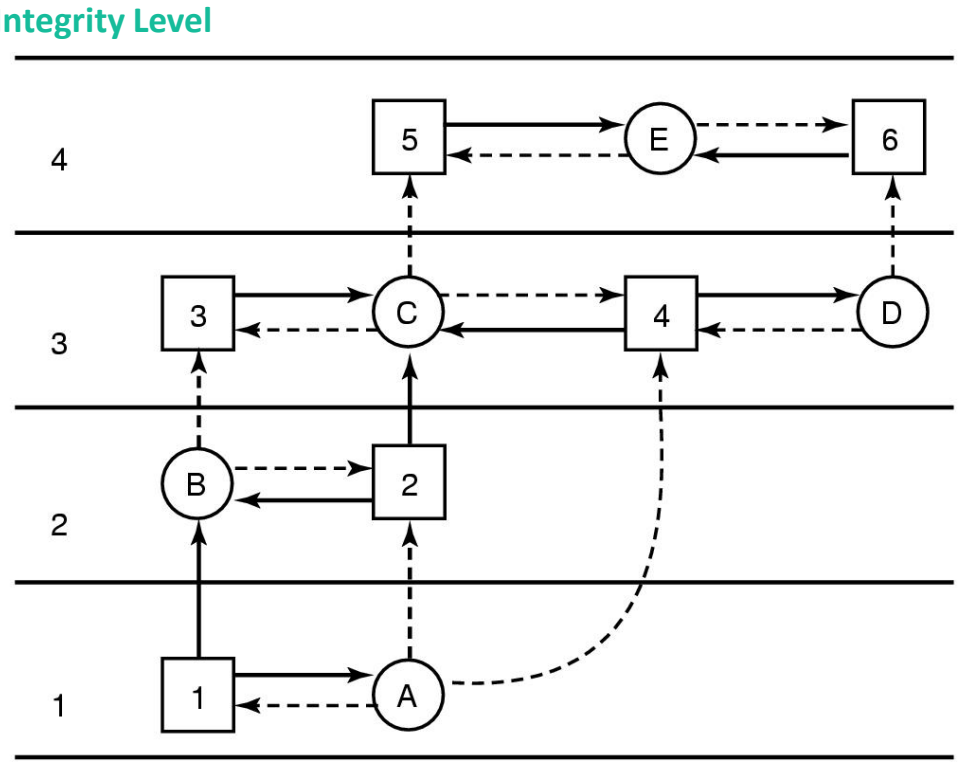
Integrity ➔ Biba Model 2/2

Modello Biba: 2 regole (proprietà)

1. **No Write Up** (Simple Integrity Principle) ➔ non inserire informazioni meno integre

2. **No Read Down** (Integrity * Property) ➔ non utilizzare informazioni meno integre



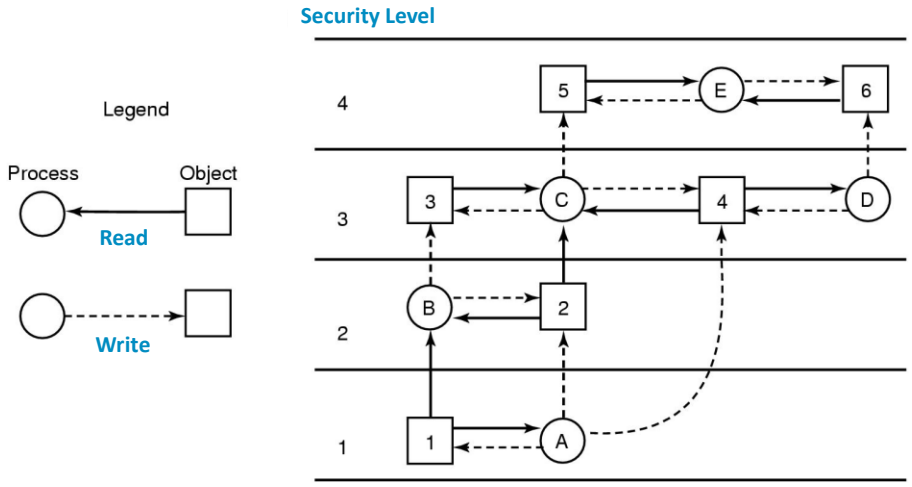


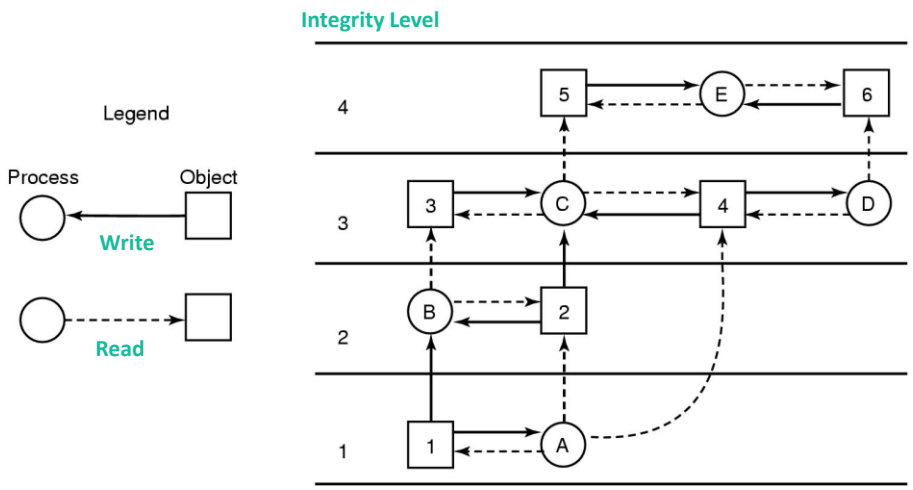
DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

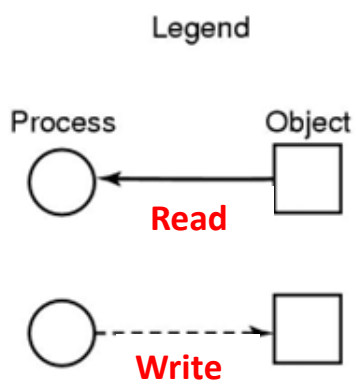
SAPIENZA
UNIVERSITÀ DI ROMA

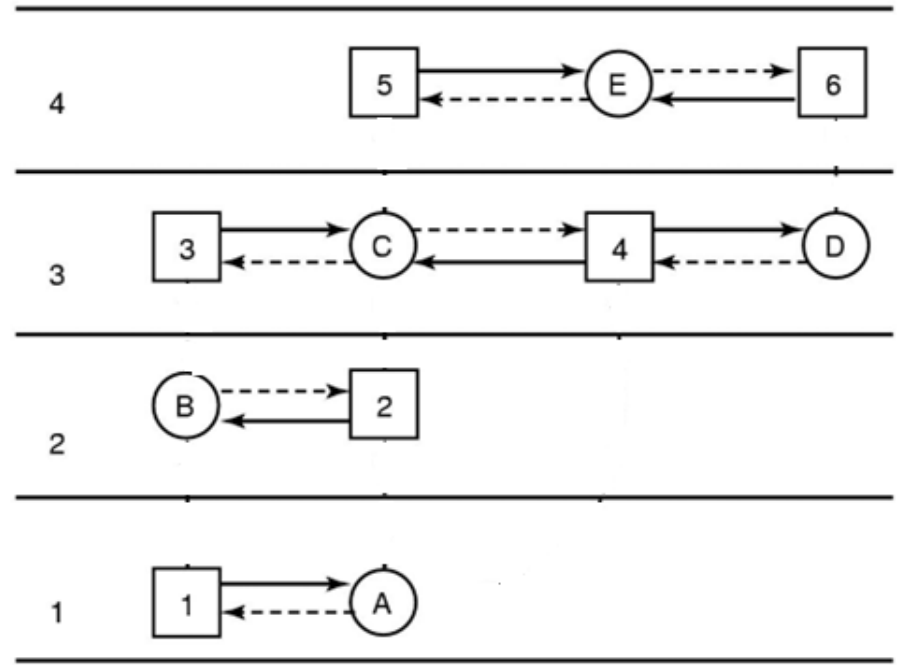
SOReCa - Sicurezza: Goals (3)

Confidentialiy + Integrity ➔ possible isolation









DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

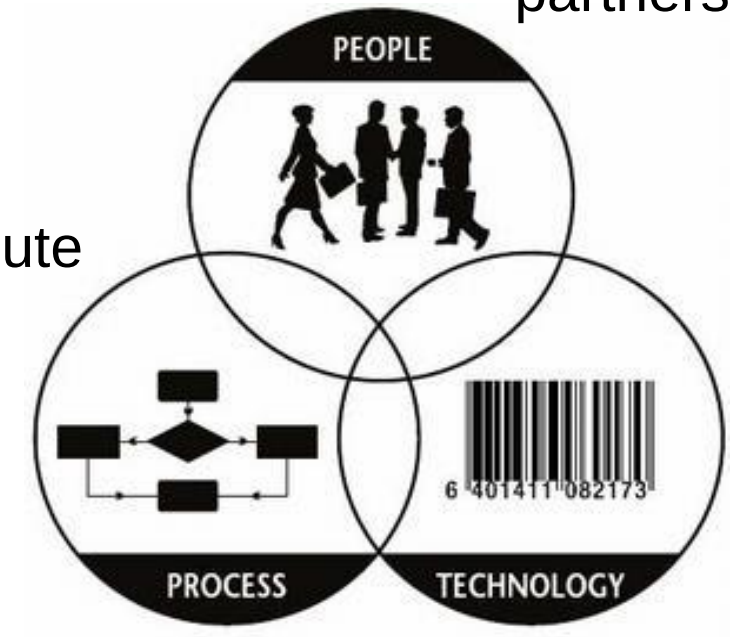
SAPIENZA
UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: Goals (3)

IT Components ➔ People, Process, Technology

People: stakeholders (employees, customers, partners, suppliers)

Process: action performed to execute business



Technology: tools used to perform processes by people

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

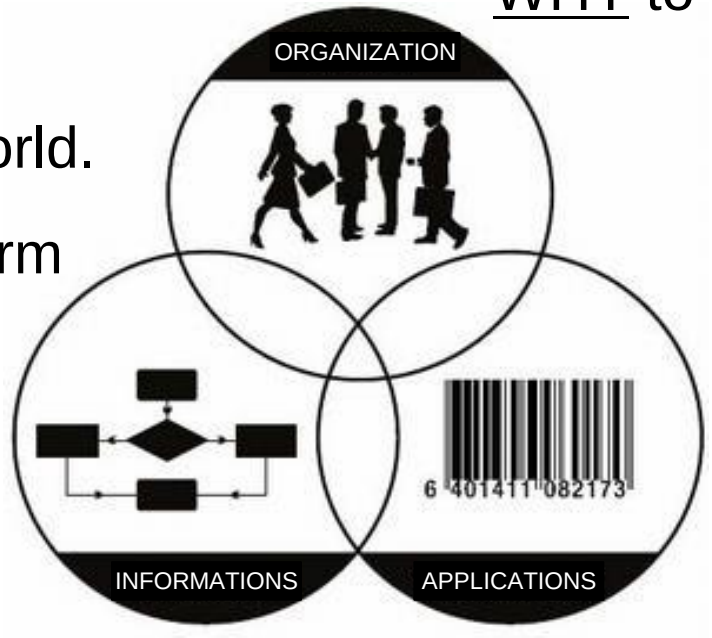
SAPIENZA UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: Goals (3)

IT Security ➔ Identity, Information, Applications

Identity: personification in the IT world. Instance of WHO and WHY to perform the actions

Information: objects surrounding the IT world. Instance of on WHAT and WHERE to perform actions



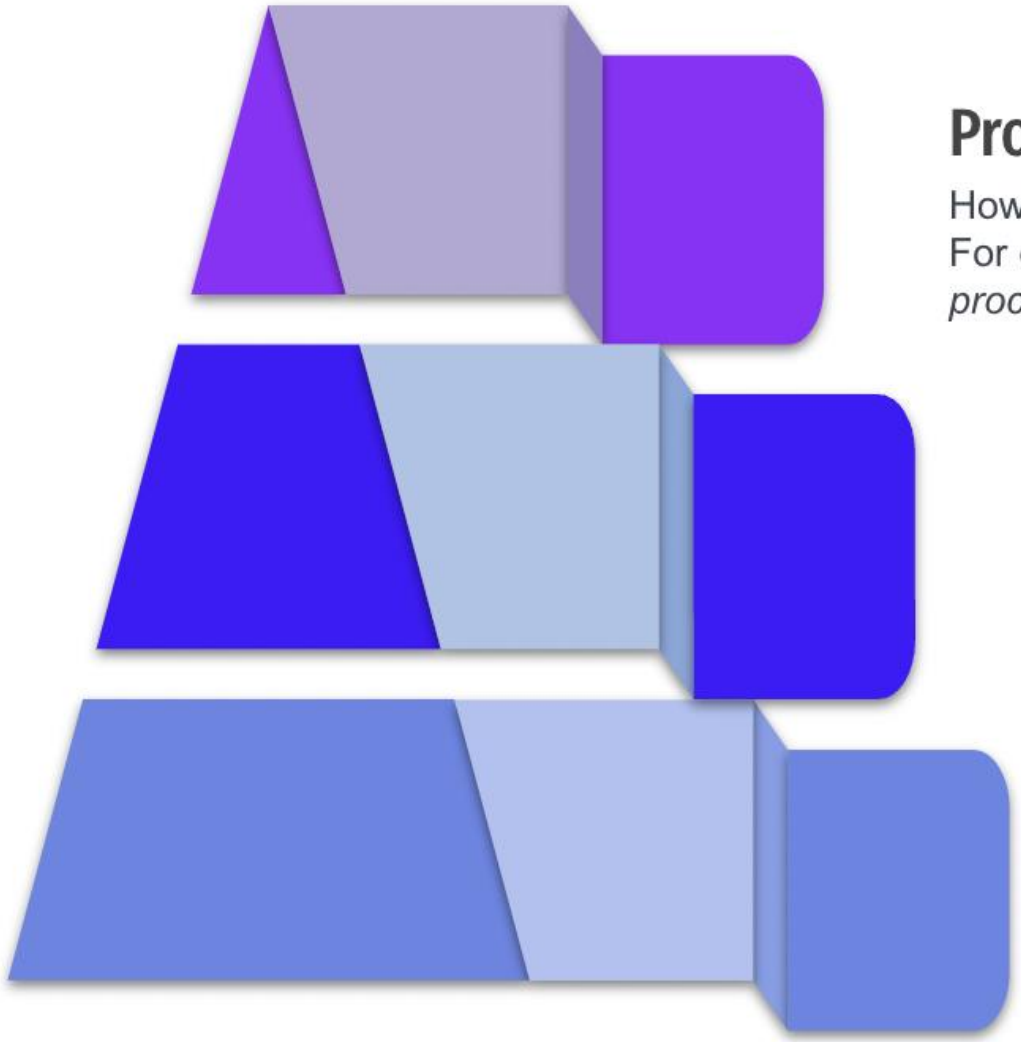
Application: objects composing the IT world. Instantiation of HOW and WHEN to perform the actions

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

SOReCa - Sicurezza: Goals (3)

IT Adversary ➔ TTP: Tactics, Techniques, Procedures



Procedures

How the technique was carried out. For example, the attacker used *procdump -ma lsass.exe lsass_dump*

Techniques

Techniques represent the tactical goal of the procedure. For example, T1003.001 - OS Credential Dumping: LSASS Memory.

Tactics

Tactics represent the strategic goal of the adversary. For example, TA006 - Credential Access

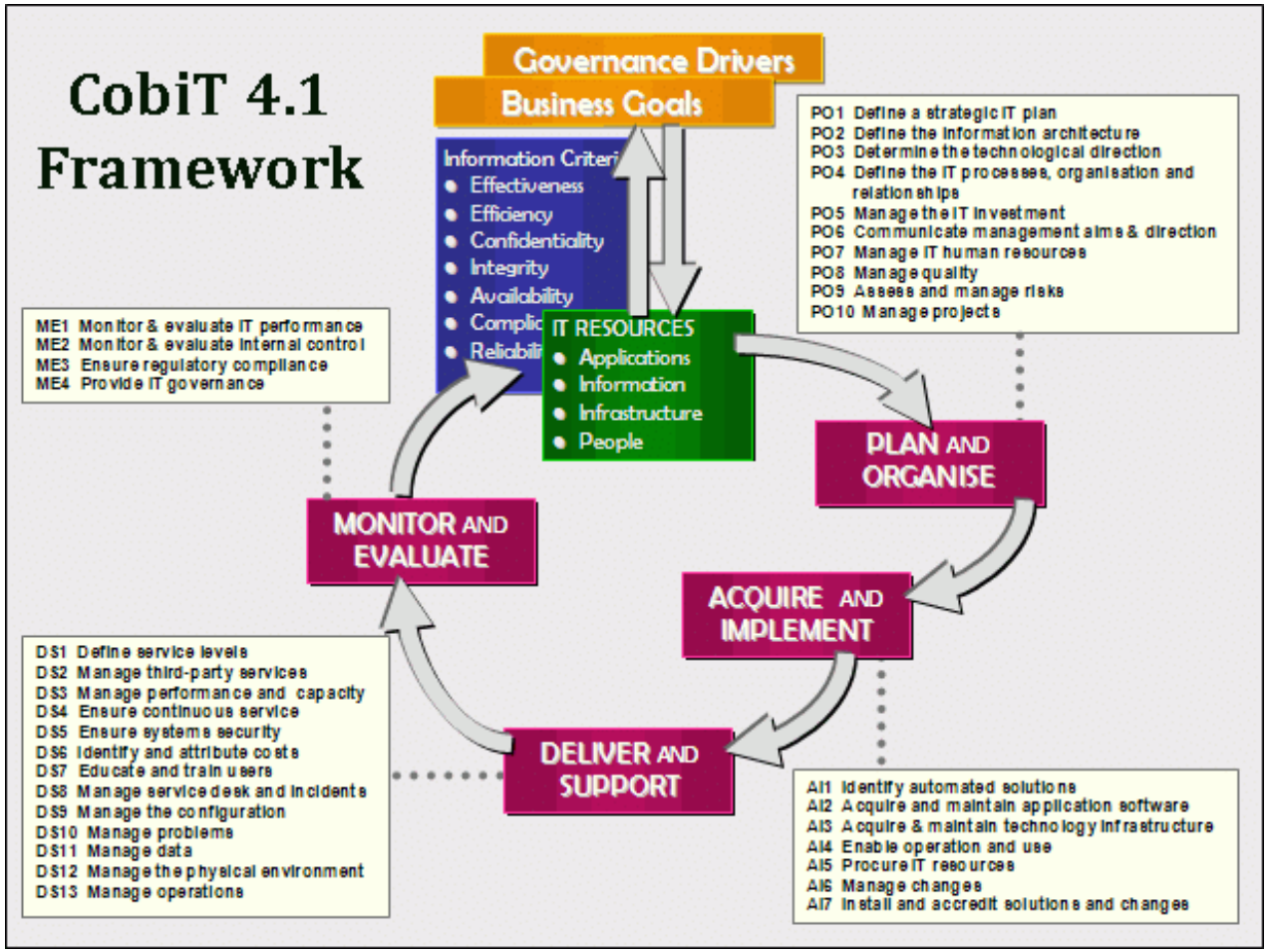
Security Management (4): Feedback

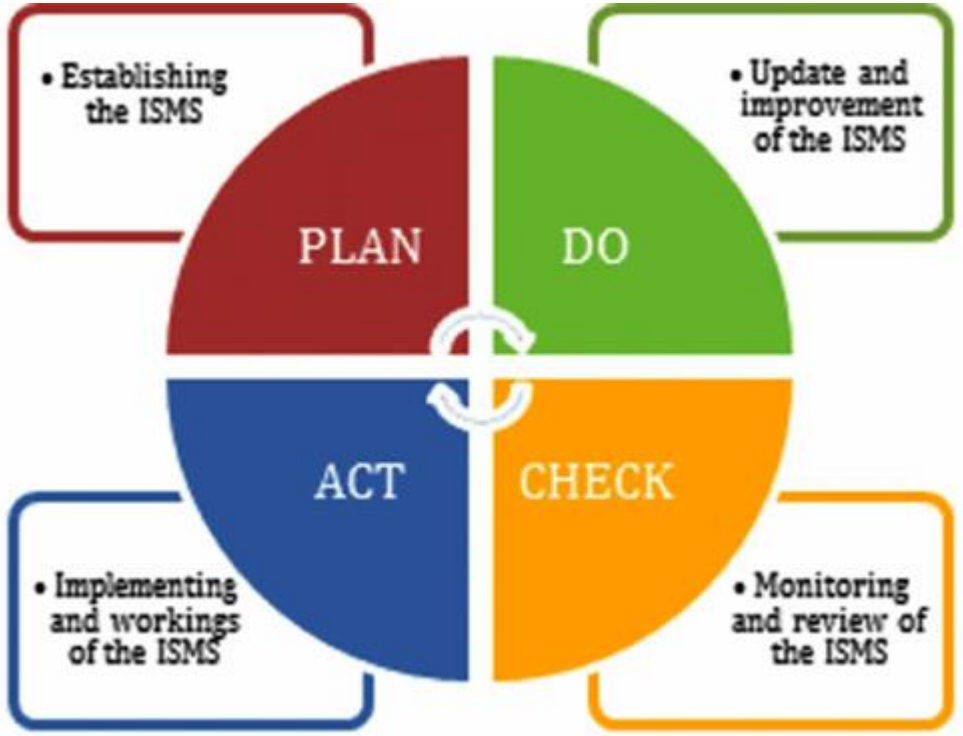
Process:

- ISO 27001: PDCA (Plan Do Check Act),
- CObIT: PO, AI, DS, ME
- NIST CSF (Cyber Security Framework)

SOReCa – Sicurezza: Security Management (4)

CObIT 4 & PDCA cycle:





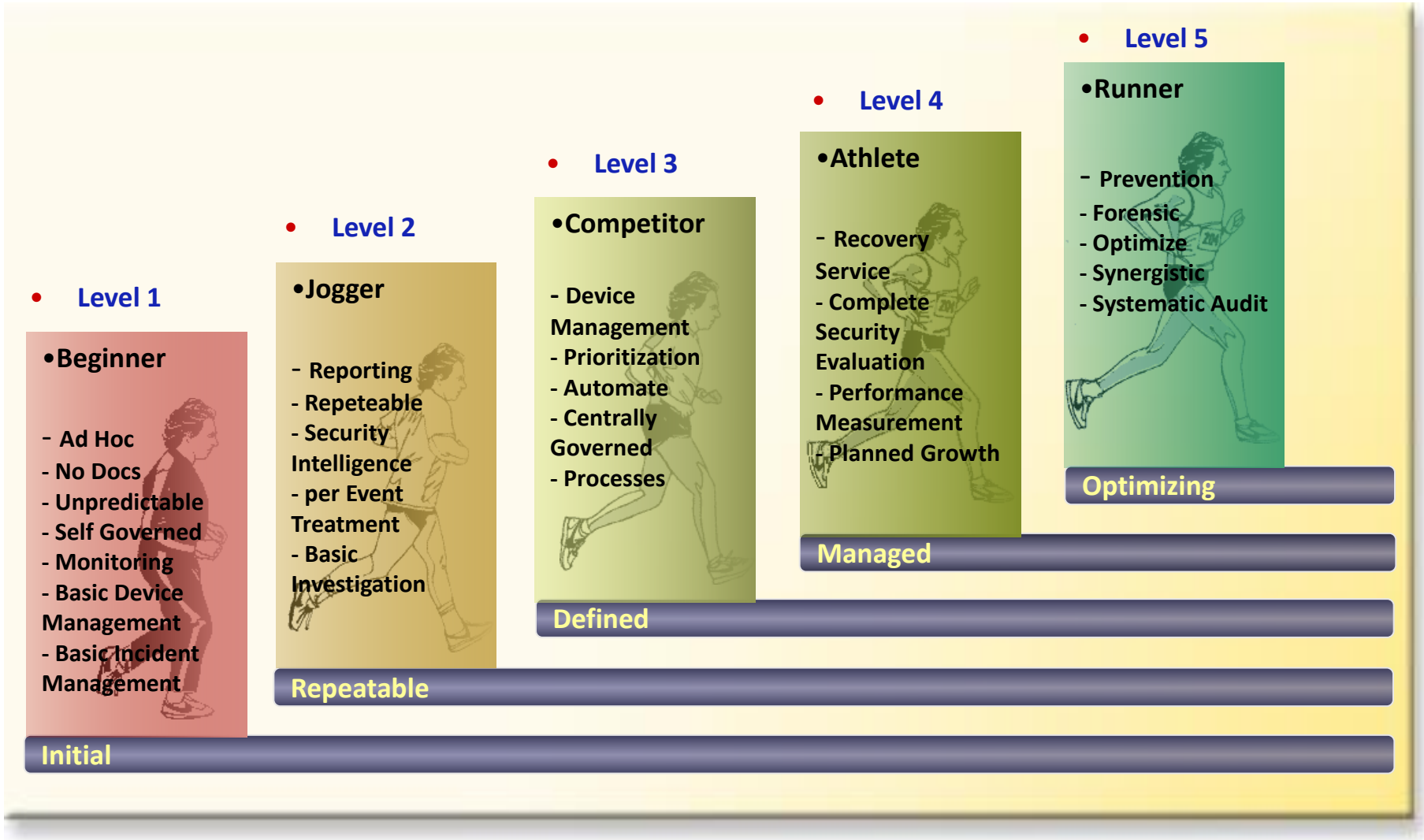
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

SOReCa – Sicurezza: Security Management (4)

4+1: CMMI (Capability Maturity Model Integrated)

Addressing "πάντα ῥεῖ" (Ἡράκλειτος, Ἐφέσιος VI - VII century B.C.)



5 Levels of **CMMI**

(Carnegie Mellon 1999)

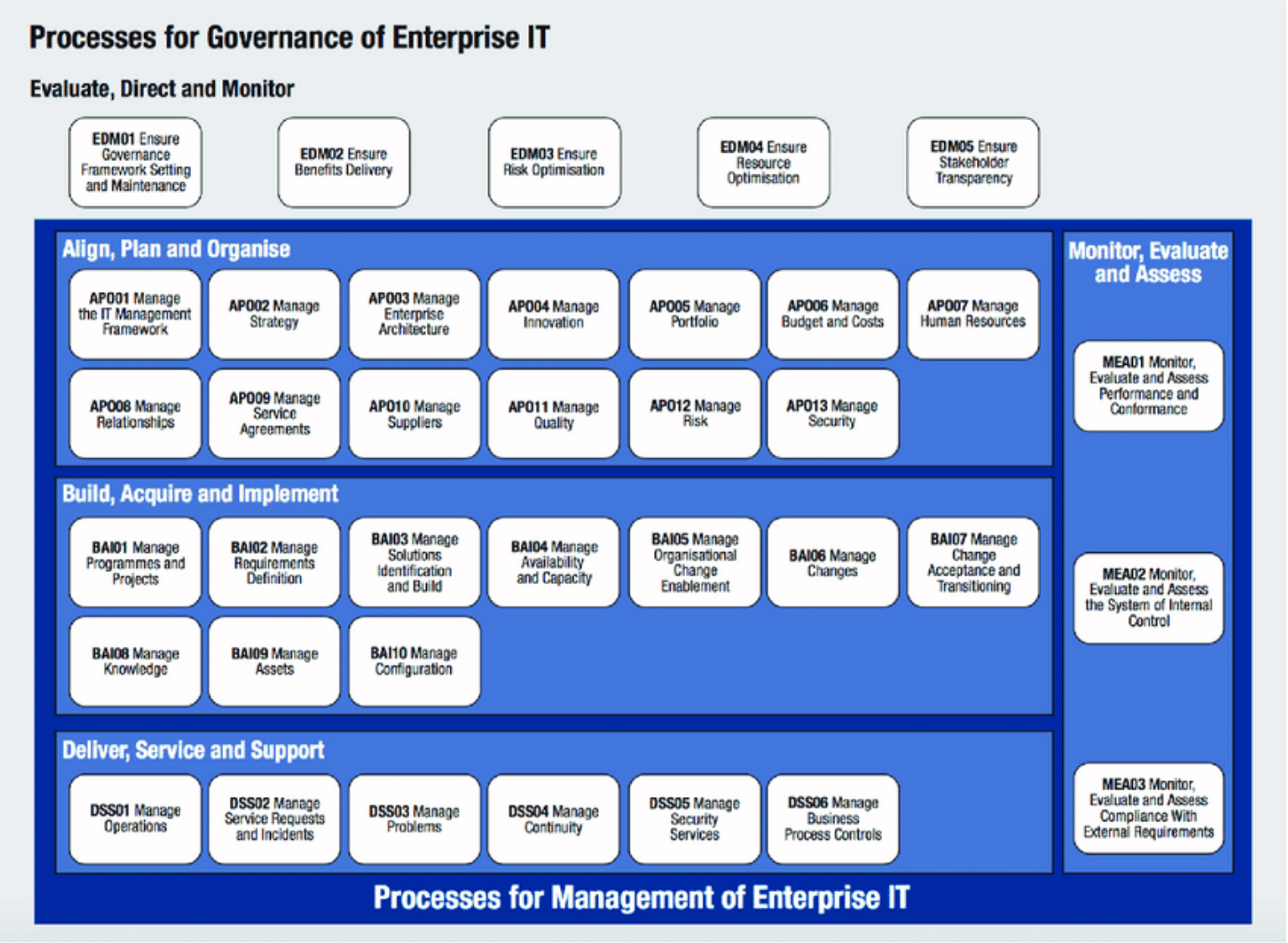
1. **Initial**: Heroic
2. **Repeatable**: (some) Procedures
3. **Defined**: Processes
4. **Managed**: Quality/Performance Measurement
5. **Optimizing**: Continous Improvement ➔ **Governance**

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa – Sicurezza: Security Management (4)

4+1: CObIT 5



1. Govern: EDM
2. Define: APO
3. Design: BAI
4. Operate: DSS
5. Control: MEA

Manage ≠ Govern

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

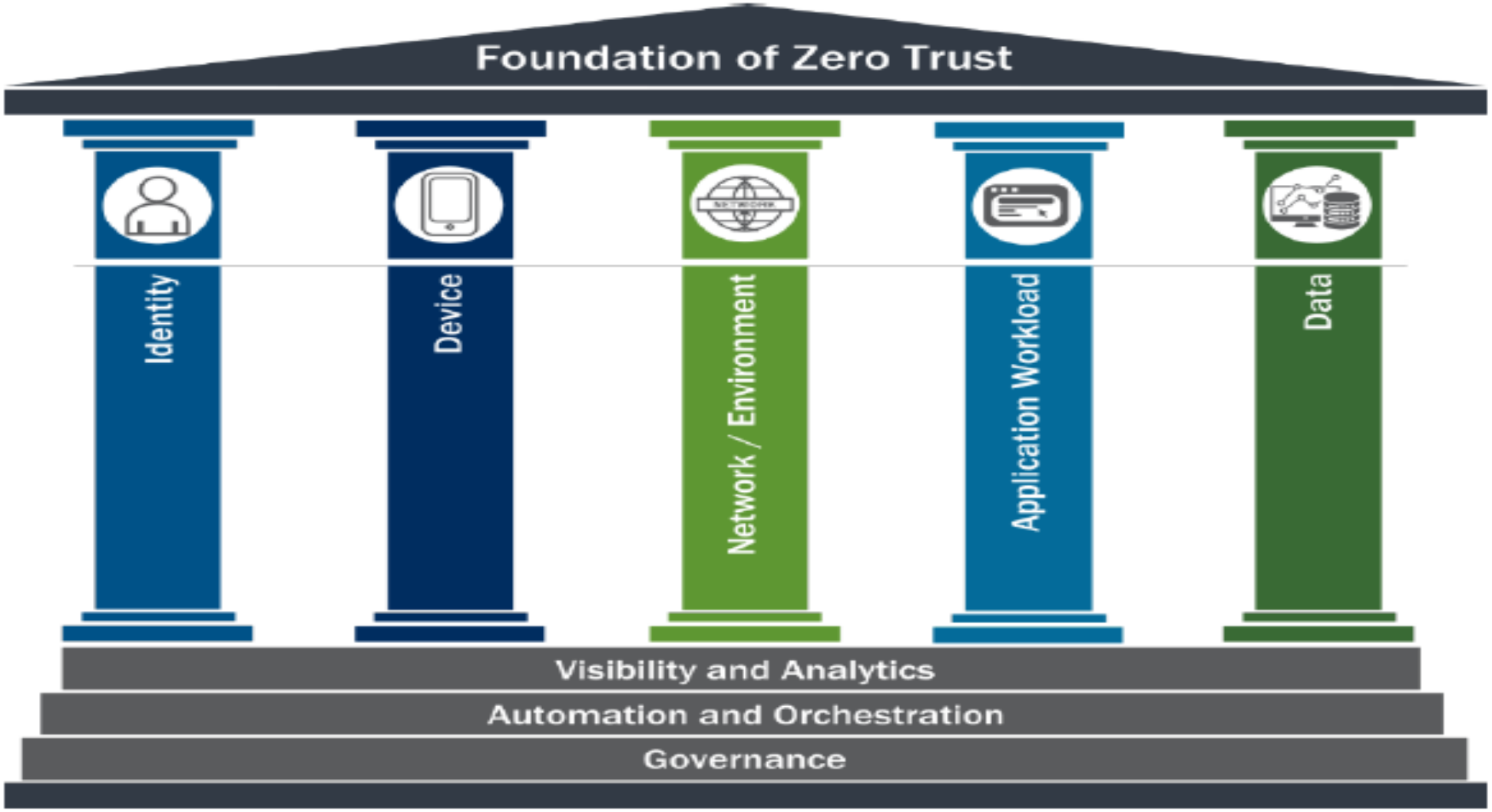
Sapienza Università di Roma

ZTA Pillars (5)

Environments: Identity, EndPoint, Network, Workload, Data

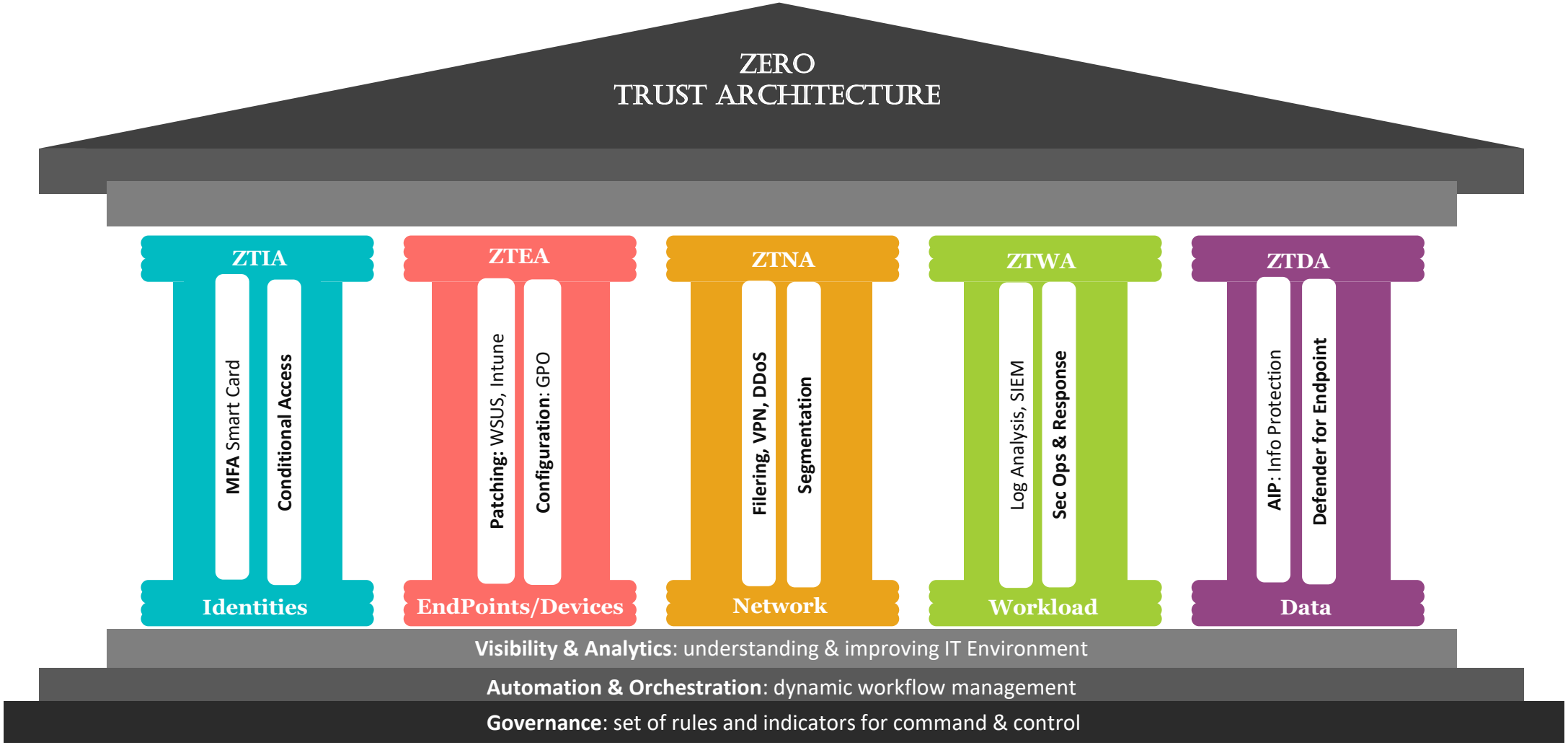
SOReCa – Sicurezza: ZTA Pillars (5)

NIST SP 800-207 (CISA ZT Maturity Model)



SOReCa – Sicurezza: ZTA Pillars (5)

NIST SP 800-207 (CISA ZT Maturity Model)

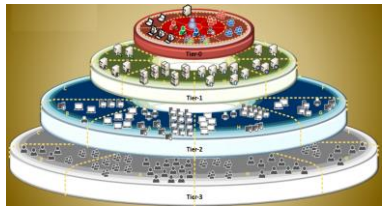


DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

SOReCa – Sicurezza: ZTA Pillars (5)

ZTA: Evolution of Trust Models & Topologies



Years	Name	Fashion	Remote	Description	Trust	Tools	Drawback
‘90s	**Tier Model** strict separation of assets	«Circles of Hell»	No / a Few	logical separation of assets by boundaries in the same physical location (old-fashioned **Perimeter-Centric**).	Inside Yes, Outside. No Delegation Model	FW IDS	No Remote
’00s	**Hub & Spoke** connect outlying points to a central "hub".	«Airline Routes»	Some	remote connections secured by VPN tunnels (strong pub-key cryptography) converging at one location (**Centralized Branch Office**)	Outside could get as Inside Central Visibility & Control	VPN SSL-VPN VDI RDP	Bottleneck and SPoF
‘20s	**Zero Trust** Authentication GW Distribution	«Never Trust, Always Verify»	Most	connections are granted after careful verification (Identity, Device, Time, Geolocation, Security Posture (**Default Deny**)	per-transaction basis. Pervasive Telemetry	PEP (CASB, ATP, DLP, …) ➔SASE	Distributed network of PoPs

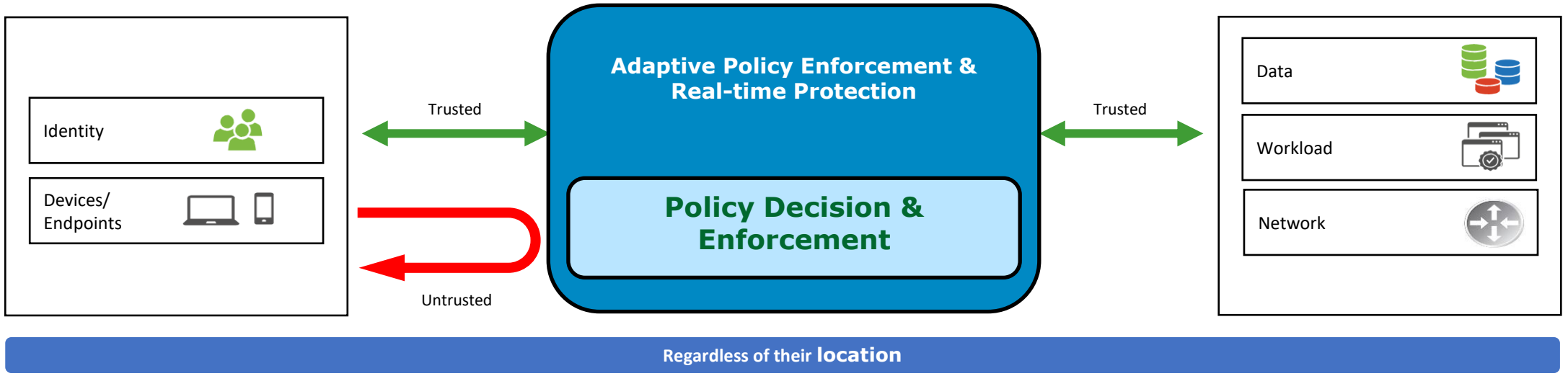
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa – Sicurezza: ZTA Pillars (5)

ZTA: Alternative Trust Model & Topology

Zero Trust is an **Alternative Cybersecurity model**, addressing the shortfalls of perimeter centric protection

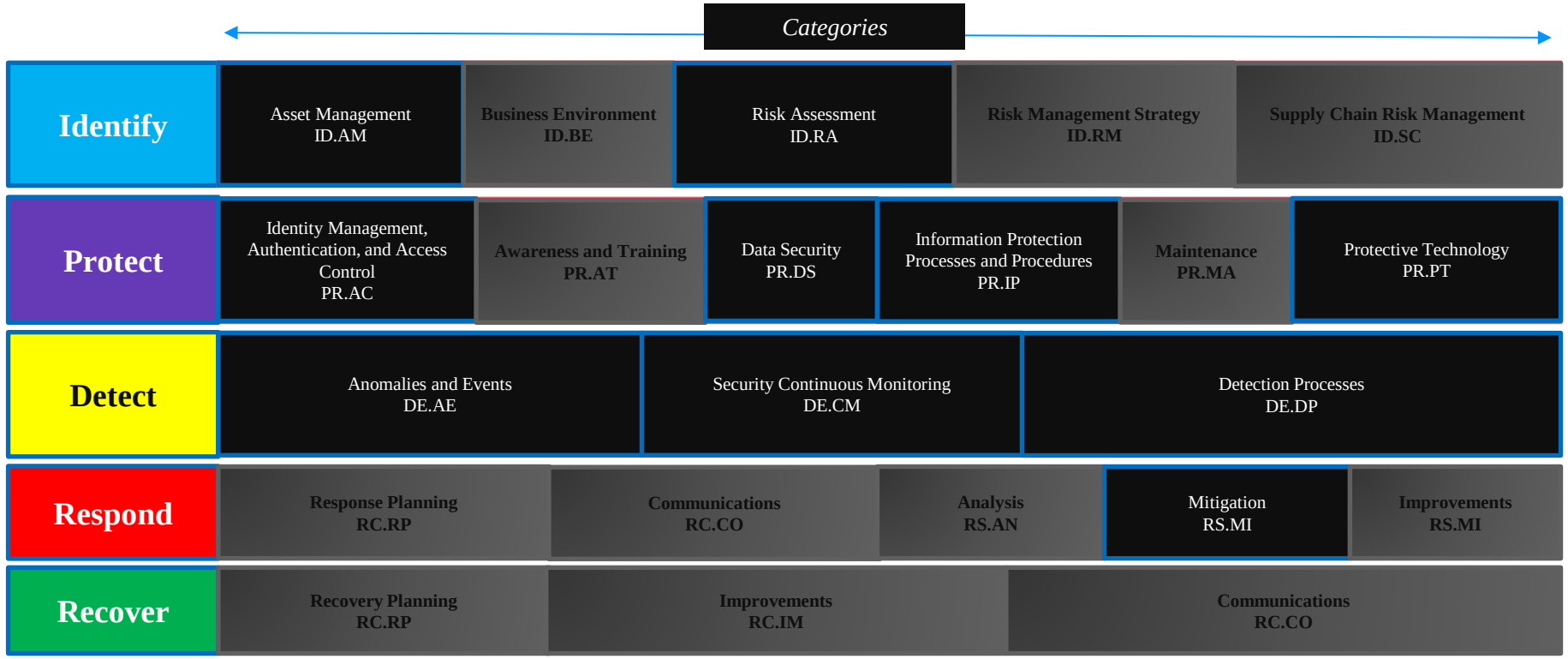


- Focusing on Protecting Data rather than access to devices, removing the assumption of perimeter trust.
- Enforcing Access Control by a Decision/Enforcement Point, based not more only on Network rules but on dynamic Policies calculated on continuous verification
- Assuming Identity as the new front line (together with accessing device), continuously assessing it and his behaviours.

SOReCa – Sicurezza: ZTA Pillars (5)

ZTA: NIST CSF subset

NIST Cyber Security Framework: Category mapping for Pervasive Telemetry



As mapped by NCCoE in the paper "Implementing a ZTA"

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

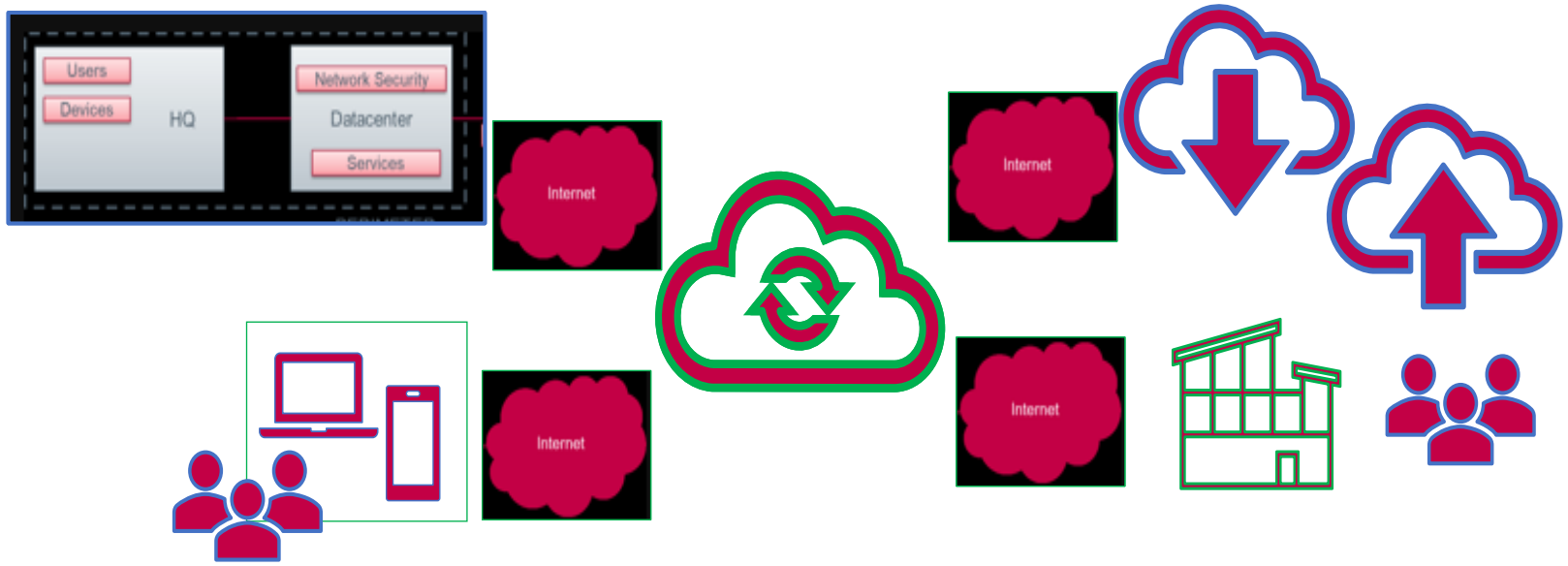
SOReCa – Sicurezza: ZTA Pillars (5)

ZTA: IT Functions

ZTA: IT Functions: security & protection

Several tools enabling ZTA for Hybrid Cloud.

Those could be classified on:



- **Infrastructure**: tools for security management of the Hybrid Cloud components, its usage readiness and configuration. That is, by *static* point of view, focused on the management of the service items and their status. Without direct relation to any specific connection, interaction, activity (about 2/3 of the tools).

- **Transaction**: tools for security & management of any specific connection, interaction, activity amidst the Hybrid Cloud. That is, by *dynamic* point of view, focused on access, about the usage of the configuration set by the infrastructure tool (about 1/3 of the tools). Often integrated in **SASE** platforms and **SD-WAN** as well.

SOReCa – Sicurezza: ZTA Pillars (5)

ZTA: Platform for protecting Infrastructure

ZTA: Platforms for protecting infrastructure

Pillar(s)	Function	Name	Enforce	Enabling
Identity	**IGA**	Identity Governance (SoD)	Authorizations: Permissions	Identity Lifecycle.
Identity	**CIEM**	Cloud Infrastructure Entitlement Management	Roles: Entitlements	Business & Application Lifecycle
Identity	**PAM**	Privileged Access Management	Authorizations: Privileged	Privilege Administration
EndPoint	**CMDB**	Asset Mgmt	Item identification	Item Configuration
EndPoint	**MDM**	Mobile Device Management	Patching	Vulnerability Management; Change & Configuration Mgmt
Network	**CNS**	Cloud Network Security	Segregation & Segmentation	Micro-Segmentation
Network Workload	**DDoS**	Anti-DDoS	Protect against obscuration	Application Availability
Workload	**SCM**	SW Configuration Mgmt	Config & Change	Approval Workflow
Workload	**CSPM**	Cloud Security Posture Mgmt	Secure Configuration	Compliance
Workload	**CWP**	Cloud Workload Protection	SW Mgmt	Configuration Management
Workload	**XDR**	eXtended Detection & Response	Threat Detection	Block advanced malware, exploits and fileless attacks
Workload	**IRM**	Integrated Risk Management	Security Dashboard	Security Governance by KPI
Data	**CKMS**	Cloud Key Mgmt Service	Secure Key Mgmt	Centralized key control in hybrid cloud

SOReCa – Sicurezza: ZTA Pillars (5)

ZTA: Platform for protecting Transationcs

ZTA: Platforms for Protecting Transaction ➔ SASE

Pillar(s)	Function	Name	Enforce	Enabling
Identity Workload	**CASB**	Cloud Access Security Broker	threats, and data leakage identification	Access to cloud applications and shadow IT
EndPoint	**SWG**	Secure Web Gateway	URL filtering	Access to Internet
EndPoint	**ATP**	Advanced Threat Prevention	Blocking threats	Spreading across endpoints and nets.
EndPoint Network	**DNS-Sec**	DNS Security	predicting, blocking, and tracking malicious activity	Access to Internet
Network	**VPN**	Virtual Private Network	threats, and data leakage	Access to shadow IT
Network	**SD-WAN**	SW Defined WAN	intelligent unified view and simplified mgmt	Traffic Prioritization, WAN Optimization, converged backbones)
Network	**FWaaS**	FW as a Service	Next Generation Rules	Net Filtering
Data	**DLP**	Data Loss Prevention	Detecting/Blocking Exfiltration	Access to Company Data

Not all SASE vendors do implement all the listed ZTA functions

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

CSF Function (6)

Process: Govern, Identify, Protect, Detect, Response, Recovery

SOReCa – Sicurezza: CSF Function (6)

6-1: NIST CSF 1.1 (Cyber Security Framework)

Core **Functions** form the "**operational culture**" that addresses cybersecurity risks.

Function Identifier	Function	Category Identifier	Category
ID	Identify	ID.AM	Asset Management
		ID.BE	Business Environment
		ID.GV	Governance
		ID.RA	Risk Assessment
		ID.RM	Risk Management Strategy
		ID.SC	Supply Chain Risk Management
PR	Protect	PR.AC	Identity Management and Access Control
		PR.AT	Awareness and Training
		PR.DS	Data Security
		PR.IP	Information Protection Processes and Procedures
		PR.MA	Maintenance
		PR.PT	Protective Technology
DE	Detect	DE.AE	Anomalies and Events
		DE.CM	Security Continuous Monitoring
		DE.DP	Detection Processes
RS	Respond	RS.RP	Response Planning
		RS.CO	Communications
		RS.AN	Analysis
		RS.MI	Mitigation
		RS.IM	Improvements
RC	Recover	RC.RP	Recovery Planning
		RC.IM	Improvements
		RC.CO	Communications

The NIST Cybersecurity Framework formally defines its Core as "a **set of cybersecurity activities**, desired outcomes, and applicable references across **critical infrastructure sectors**."
The Core consists of standard cybersecurity controls slotted into a taxonomy of:

- 5 Functions,
- 22 Categories,
- 98 Subcategories.

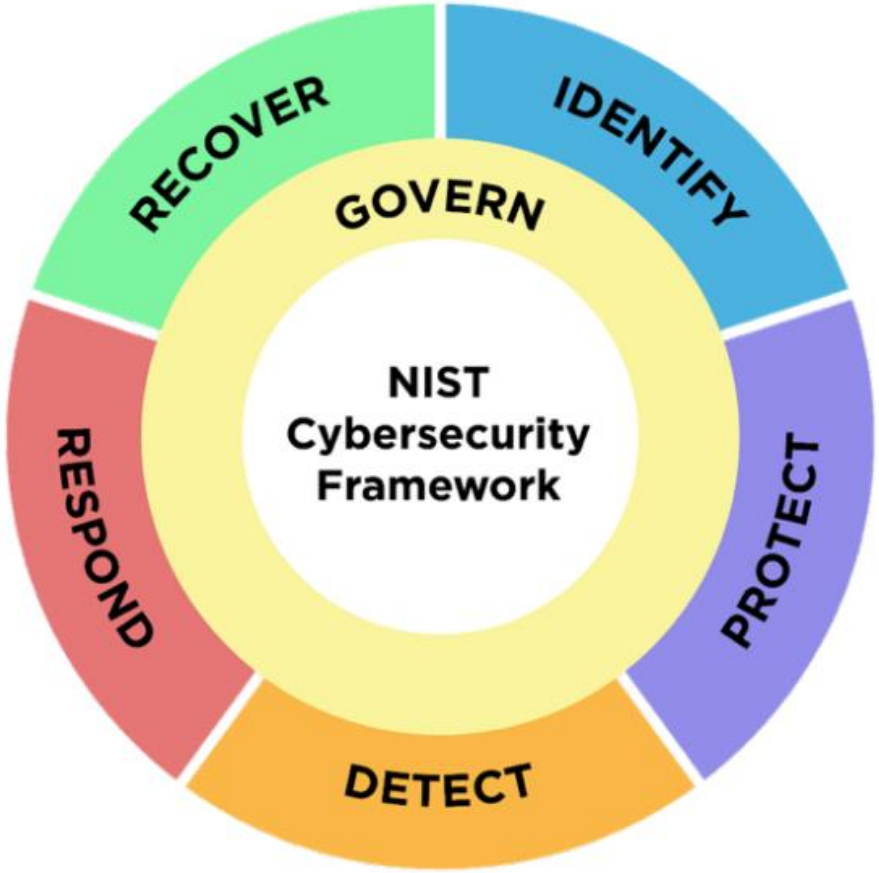
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa – Sicurezza: CSF Function (6)

NIST CSF from 1.1 to 2.0 (addressing Governance)

NIST Cybersecurity Framework 2.0		
CSF 2.0 Function	**CSF 2.0 Category**	**CSF 2.0 Category Identifier**
Govern (GV)	Organizational Context	GV.OC
	Risk Management Strategy	GV.RM
	Roles and Responsibilities	GV.RR
	Policies and Procedures	GV.PO
Identify (ID)	Asset Management	ID.AM
	Risk Assessment	ID.RA
	Supply Chain Risk Management	ID.SC
	Improvement	ID.IM
Protect (PR)	Identity Management, Authentication, and Access Control	PR.AA
	Awareness and Training	PR.AT
	Data Security	PR.DS
	Platform Security	PR.PS
	Technology Infrastructure Resilience	PR.IR
Detect (DE)	Adverse Event Analysis	DE.AE
	Continuous Monitoring	DE.CM
Respond (RS)	Incident Management	RS.MA
	Incident Analysis	RS.AN
	Incident Response Reporting and Communication	RS.CO
	Incident Mitigation	RS.MI
Recover (RC)	Incident Recovery Plan Execution	RC.RP
	Incident Recovery Communication	RC.CO



DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

SOReCa – Sicurezza: CSF Function (6)

4+1: NIST CSF (Cyber Security Framework)

Function	Category	Category Identifier
Govern (GV)	Organizational Context	GV.OC
	Risk Management Strategy	GV.RM
	Roles, Responsibilities, and Authorities	GV.RR
	Policy	GV.PO
	Oversight	GV.OV
	Cybersecurity Supply Chain Risk Management	GV.SC
Identify (ID)	Asset Management	ID.AM
	Risk Assessment	ID.RA
	Improvement	ID.IM
Protect (PR)	Identity Management, Authentication, and Access Control	PR.AA
	Awareness and Training	PR.AT
	Data Security	PR.DS
	Platform Security	PR.PS
	Technology Infrastructure Resilience	PR.IR
Detect (DE)	Continuous Monitoring	DE.CM
	Adverse Event Analysis	DE.AE
Respond (RS)	Incident Management	RS.MA
	Incident Analysis	RS.AN
	Incident Response Reporting and Communication	RS.CO
	Incident Mitigation	RS.MI
Recover (RC)	Incident Recovery Plan Execution	RC.RP
	Incident Recovery Communication	RC.CO

Function	Cybersecurity	Goals	Manage
GOVERN (GV)	Risks	Strategy, Expectations, and Policy	establish, communicate, monitor
IDENTIFY (ID)	Risks	Current Assessment	understand
PROTECT (PT)	Risks	Safeguards	use, manage
DETECT (DE)	Contingency	Attacks and Compromises	find, analyze
RESPOND (RS)	Contingency	Mitigation	undertake
RECOVER (RS)	Status	Asset and Operations	restore

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Security & Protection (8)

Sec & Risk Mgmt, Asset Sec, Sec Arch & Engineering, Comms/Net Sec, IAM-IAG, Sec Assessment & Testing, Sec Ops, SW Dev Sec

SOReCa – Sicurezza: Security & Protection (8)

Classificazione delle «Misure Elementari»

8 principali categorie:

1. **Sec & Risk Mgmt**: Risk Analysis (cfr. «Sicurezza (1b)») + Risk Management (countermeasure selection)

2. **Asset Security** (**Endpoint Security**): Hardening (Security Hygiene). [→ corso Sicurezza: Run-time Mitigations (DEP, ASLR, stack integrity, CFI, etc)]

3. **Sec Arch & Engineering**: Tier Model (cfr. «ZTA Pillars») + Clustering (cfr. «04-SOReCa-IPC»: Clusters)

4. **Comms/Net Sec**: Firewall, IDPS, WAF → corso SOReCa (componente Reti di Calcolatori)

5. **IAM-IAG**: Identity & Access Management/Governance (AAA)

6. **Sec Assessment & Testing**: VA, PT, WAPT, SAST, SCA, DAST → corso Sicurezza: Software Testing (bug vs vulnerability, static and dynamic analysis methodologies)

7. **Sec Ops**: SOC, Cyber Kill Chain, MITRE ATT&CK

8. **SW Dev Sec**: DevSecOps, BOF, UAF, Input Validation → corso Sicurezza: Software Life Cycle, Secure Coding, Testing Software for Security Bugs

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

SOReCa – Sicurezza: Security & Protection (8)

1. Sec & Risk Mgmt

Le informazioni devono poter essere elaborate:

- nelle modalità stabilite ➔ **Sicurezza**
- dal modello adottato ➔ **Protezione**

Sicurezza: Identificazione del Modello da adottare (**Statica**)

1. **Threats**
2. **Vulnerability**
3. **Risk Analysis**
4. **Countermeasures**

Protezione: Applicazione del Modello scelta della contromisura (**Dinamica**)

1. **Costo**
2. **Fattibilità Tecnica**
3. **Tempo di Implementazione**
4. **Valutazioni di Conformità**

SOReCa – Sicurezza (1b): Analisi del Rischio

(why): risk of cyber-threats

Quantitative Risk == **ARO** x **SLE**

probability (ARO) of loosing money (SLE) from incidents or attacks (Threats) by exploiting 1+ vulnerability.

Usually, the security risk is calculated on an annual basis

The overall Risk is the combination of all the single impacts.

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Qualitative Risk (e.g. OWASP Risk Methodology)

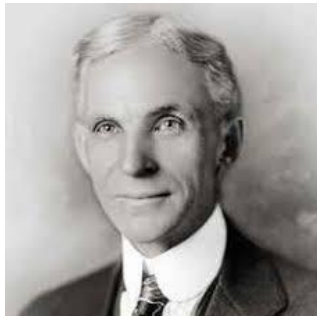
ARO: **Annual Rate of Occurrence** → Likelihood (probability), external factor: **threat**
SLE: **Single Loss Expectancy** → Impact (money), internal factor: **vulnerability**

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa – Sicurezza: Security & Protection (8)

2. Asset Security: Minimization - The Main Law of Engineering



- “Quello che non c’è non si rompe" [Henry Ford],

- "... non pesa" [Colin Chapman],

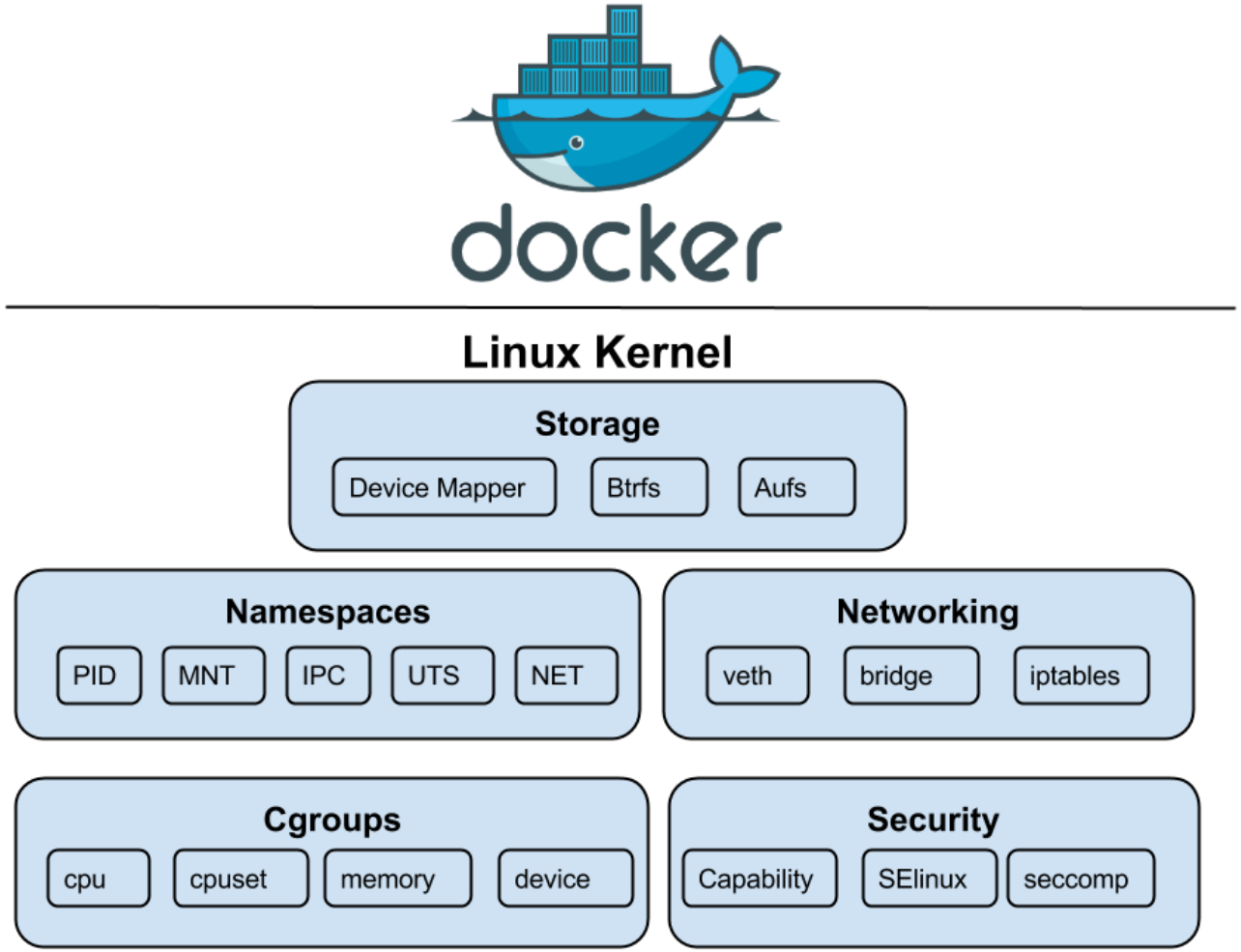
- "... non costa" [Ratan Tata]

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

SOReCa – Sicurezza: Security & Protection (8)

2. Asset Security: Minimization -



Namespace	Isolating
Mount	FS (like chroot)
PID	PID
IPC	IPC
UTS	Hostname, domainname
NET	Interfaces, routing, FW
CGROUP	Resources (CPU, RAM, dev)
TIME	clock
SECCOMP	Restrict set of SysCall
SELinux	access only to its own files
capability	Only necessary root permissions

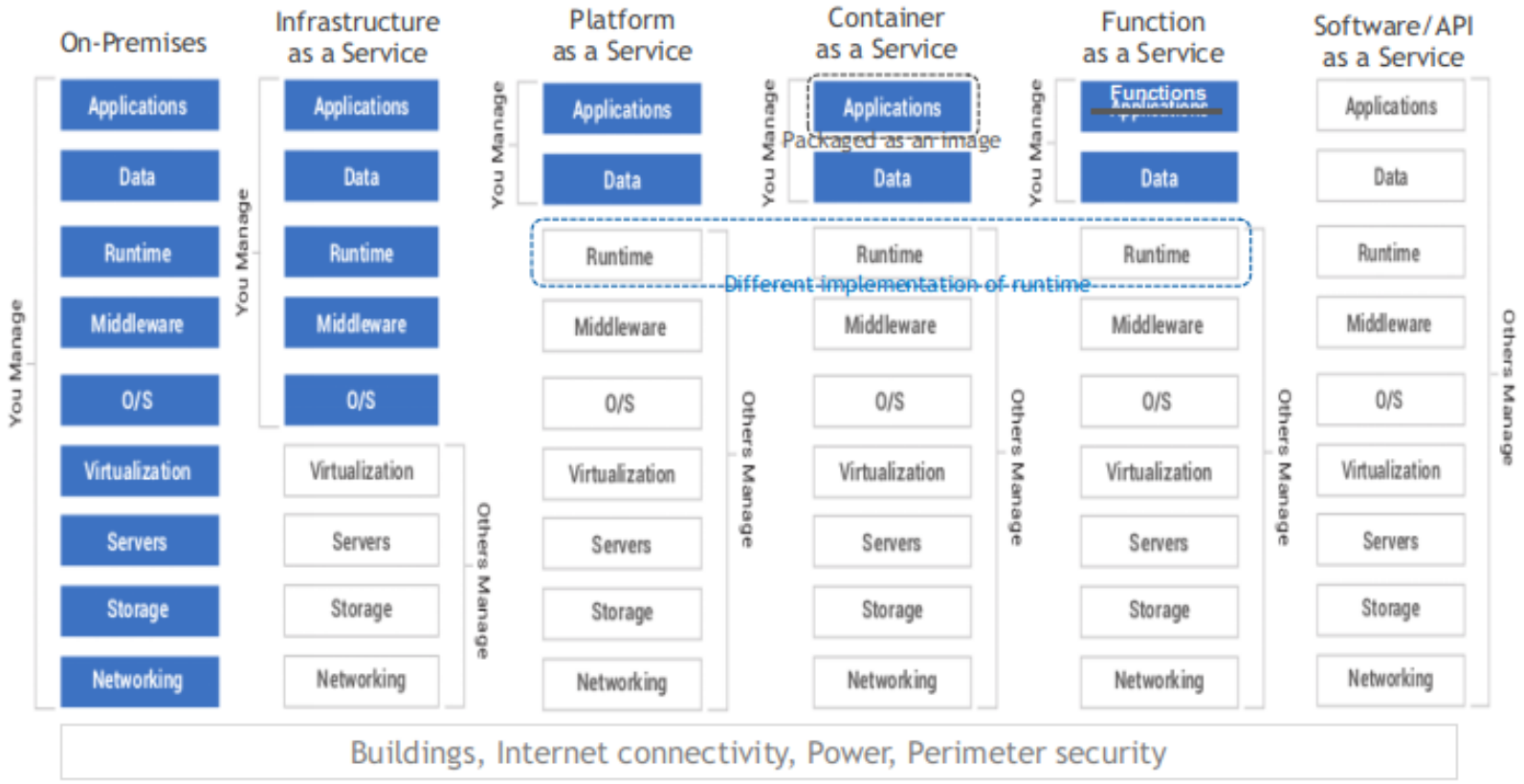
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa – Sicurezza: Security & Protection (8)

2. Asset Security: Minimization -

The initial shared responsibility model was further refined, splitting the PaaS model in 3 more detailed services, introducing as new interfaces the containers (CaaS) and the API (FaaS)



- **IaaS**: base infrastructure (Virtual machine, Software Define Network, Storage attached). End user have to configure and manage platform and environment, deploy applications on it
- **PaaS**: has morphed into two more services, container (CaaS) and function (FaaS).
- **CaaS**: bundle of application, usually shipped as Docker images, organized in cluster (usually managed by Kubernetes)
- **FaaS**: functions “on-demand software”. It allows developers to write code in small units and the service will manage everything else
- **SaaS**: “on-demand software”. Typically accessed by users using a thin client via a web browser. In SaaS everything can be managed by vendors: applications, runtime, data, middleware, OSes, virtualization, servers, storage and networking, End users have to use it.

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

SOReCa – Sicurezza: Security & Protection (8)

2. Asset Security: Minimization -

Containers **Benefits ➔ Reduction of:**

Time for (un)loading - **# Incidents** - **Transportation Costs**

1. **< 1960s**: Different Size, Shape , Consistence, Weights. Manual, specific work for (un)loading

2. **> 1960s**: Same Size, Shape, Consistence and Weight (of the Containers). Automatable work for (un)loading, stacking and transportation (without being opened!)

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

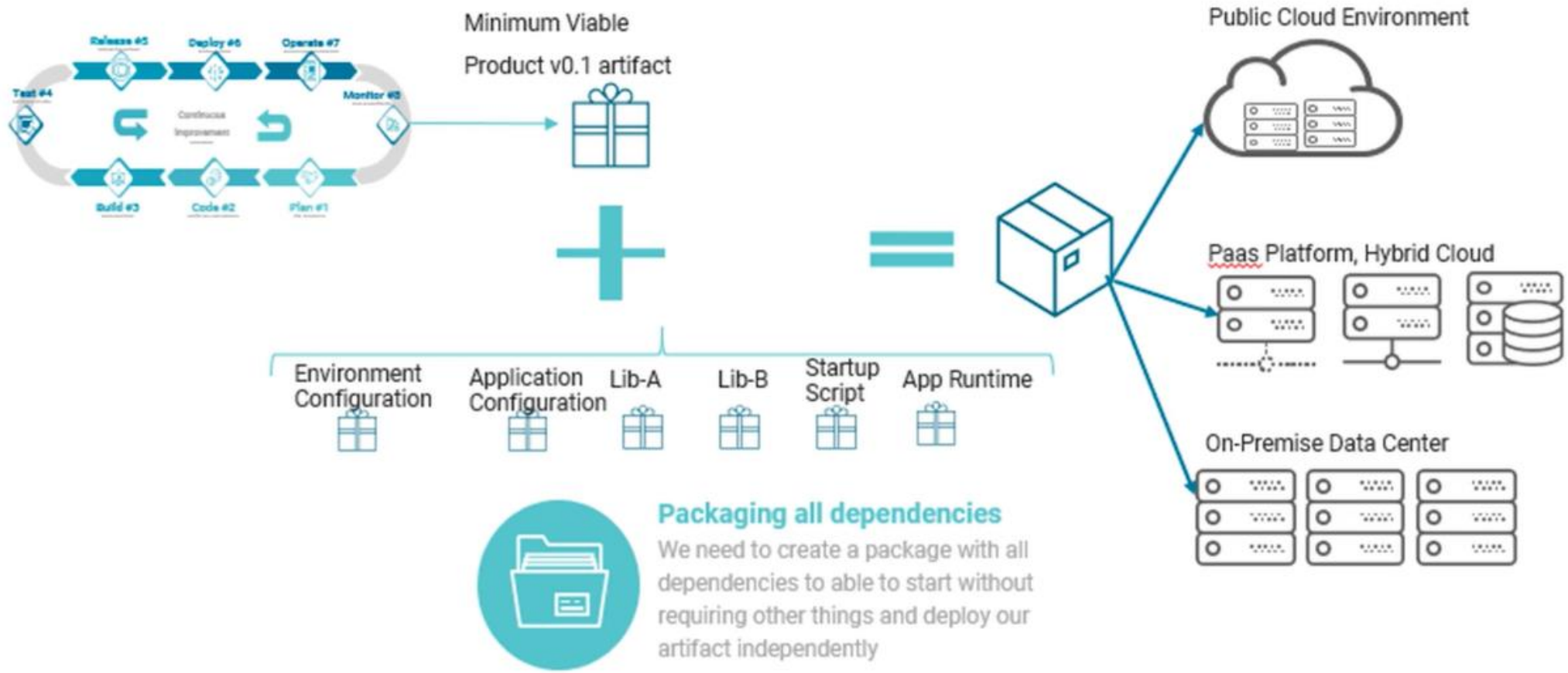
SAPIENZA UNIVERSITÀ DI ROMA

SOReCa – Sicurezza: Security & Protection (8)

2. Asset Security: Minimization -

Containers **Benefits ➔ Reduction of:**

Time for (un)loading - **# Incidents** - **Transportation Costs**

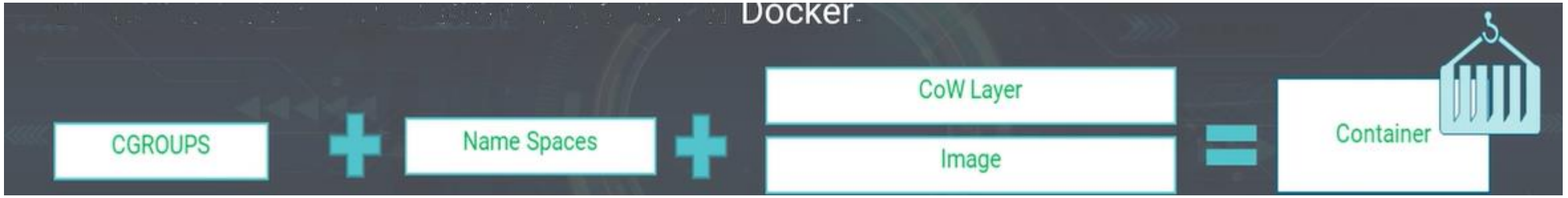


SOReCa – Sicurezza: Security & Protection (8)

2. Asset Security: Minimization -

Containers: like isolated processes

1. Bring all the items to run the application inside it
2. Status: run, started, stopped, moved, deleted
3. Run-time component of Docker



CGroups:

Kernal Features

Group processes

Control Resources Usage

Allocate HW Resources

1. CPU (CPU set)
2. Memory
3. Disk (Block I/O)

NameSpaces: isolation among Docker containers (introduced in Linux Kernel 2.6.x)

1. PID (process ID)
2. NET
3. IPC (Inter Proc Comms)
4. UTS (Unix Time Share)
5. User namespaces

CoW: Copy on Write. the copy operation is deferred until the first write.

Image: hierarchical tar ball containing:

1. O.S.
2. App
3. Lobs
4. Config

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protection();

Operating Systems nowadays provide useful countermeasures for protection:

- **Filter** (Network Flow, Input Control etc) ← Confidentiality | Integrity
  - **Access** (Mutual Exclusion)← Confidentiality | Integrity (es. DEP, ASLR)
  - **AAA** (AuthN, AuthZ, Accn) ← Confidentiality (es. FS)
  - **Cryptography** ← Confidentiality | Integrity (es. BitLocker, FDE)

- **Duplicate** (Back-Up, Cluster)← Availability (es. Replication, Failover, Load Balancing)

Qualora vi sia un incidente:

- **Log** (stderr, records) ← Response | Recovery | Post Analysis (Forensics)

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protection();

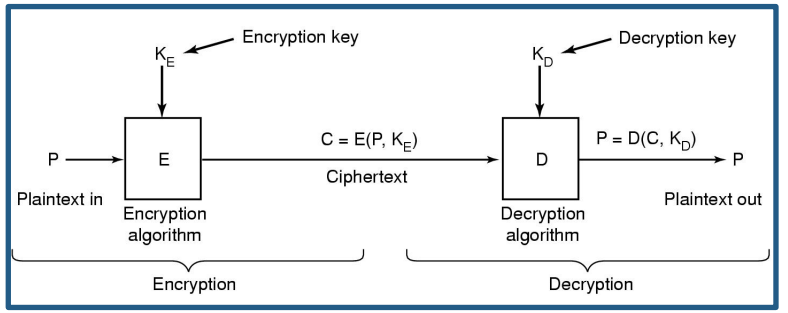
Meccanismi implementati nel SO, in dipendenza dalle funzionalità da fornire e dagli attacchi da cui difendersi.

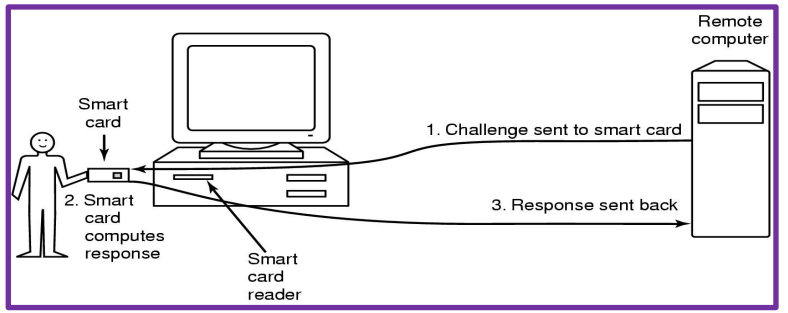
1. **Cryptography**: **Secret (Symmetric) Key**, **Public (Asymmetric) Key**, **Digital Signature**
   Codifica delle informazioni ➔ κρυπτός = nascosto, γραφω = scrivo

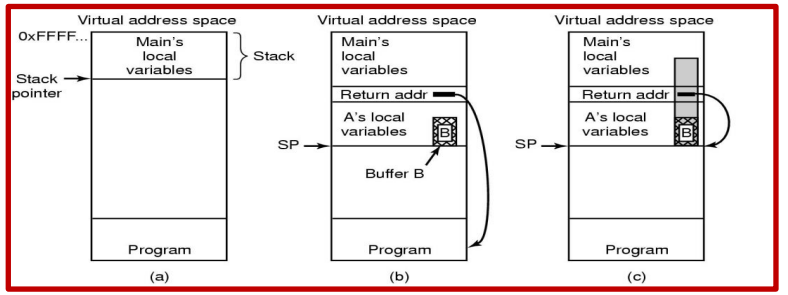
2. **AAA**: **Password**, **Physical Object**, **Biometrics**, **Permission**, **Log-Trails**
   Metodi per asseverare l'identità dichiarata:
   - Qualcosa che sai
   - Qualcosa che hai
   - Qualcosa che sei

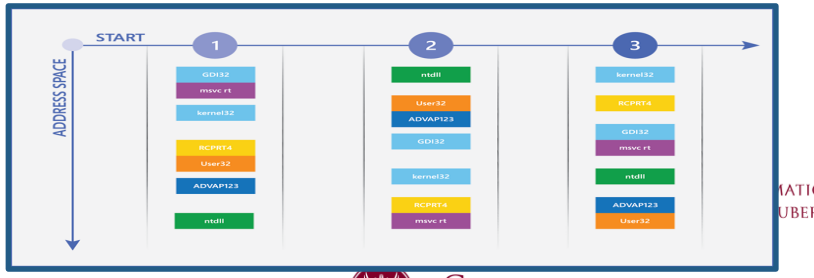
3. **Attacchi**: **Trap Doors**, **Buffer Overflow**, **Code Injection**, **Rootkits**
   Tipiche metodologie di violazione dei sistemi.

4. **Protection**: **ACL**, **NX**, **ASLR**, **Code Signing**, **Jailing**, **TCB**
   Meccanismi di difesa.







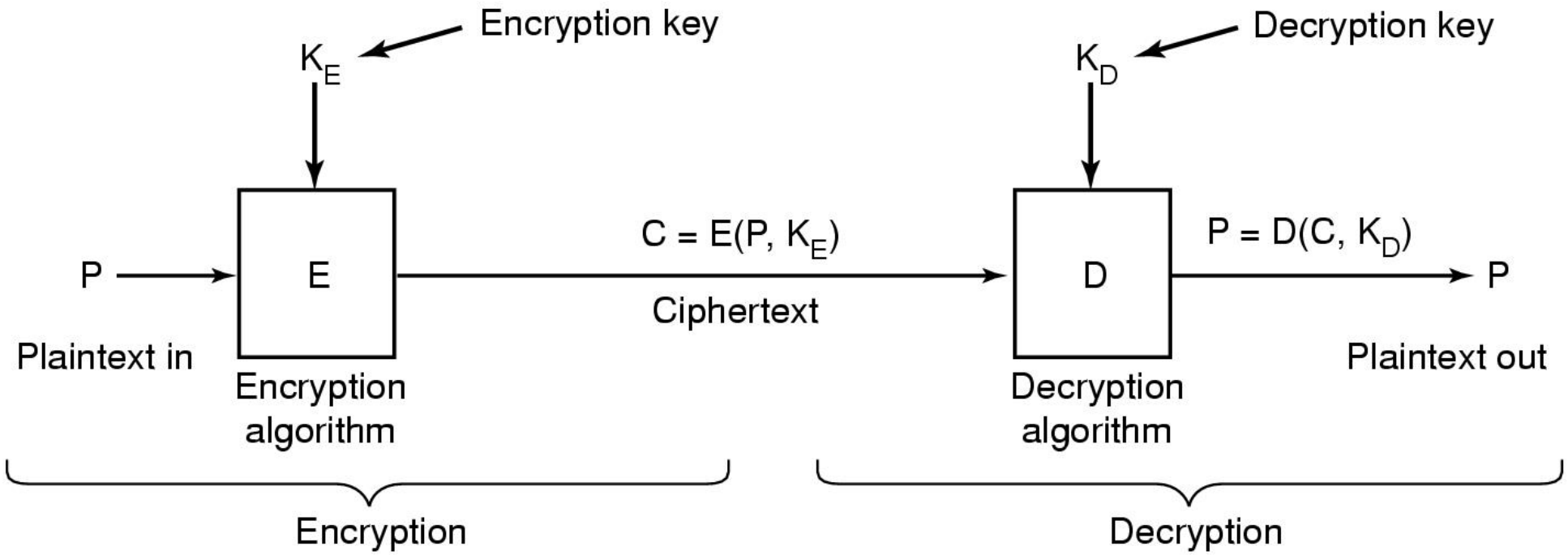


SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 1/8

Meccanismo generale di crittografia (cifratura)



1. Secret (Symmetric) Key: $K_E = K_D$ (la chiave deve essere segreta/trasmessa su altro canale)

2. Public (Asymmetric) Key: $K_E = K_D$ (la chiave K_D è pubblica; K_E è segreta: mai trasmessa)

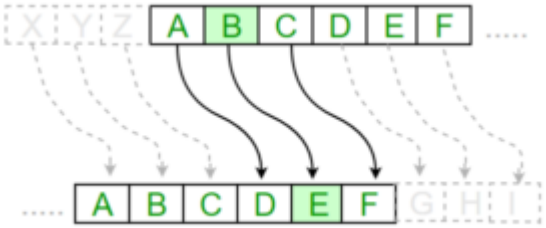
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 2/8

Secret (Symmetric) Encryption: crittografia simmetrica (a chiave privata)



Substitution cypher(Giulio Cesare): il cyphertext è ottenuto sostituendo ad ogni lettera P la lettera P + K. Giulio Cesare ha usato prima K = «C» (come in figura) e poi «D»

Utilizzo: messaggio e chiave devono essere scambiati su 2 canali diversi:

- **Chiave**: canale di scambio preliminare

- **Messaggio**: canale abituale

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

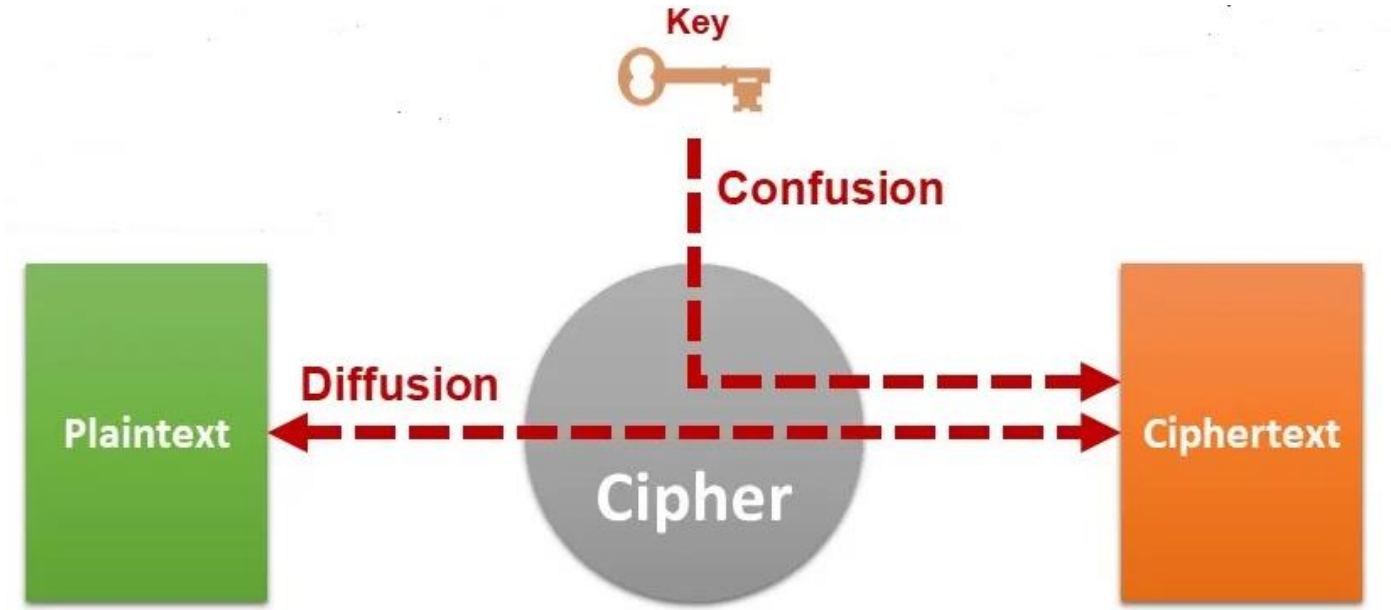
Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 3/8

Secret (Symmetric) Encryption: crittografia simmetrica (a chiave privata)

Confusion & Diffusion (Claude Shannon): in «*A Mathematical Theory of Cryptography*“ del 1945 sono riportate le 2 qualità di un cifratore sicuro:

- **Confusion**: ogni cifra binaria (bit) del testo cifrato deve dipendere da più parti della chiave, oscurando le connessioni tra le due. Nasconde la relazione tra il testo cifrato e la chiave ➔ difficile trovare la chiave dal testo cifrato. Fornita dalle **Substitution Boxes**
- **Diffusion**: il cambio di un singolo bit del testo in chiaro, comporta la modifica di circa la metà dei bit nel testo cifrato ➔ nascondere la relazione statistica tra il testo cifrato e il testo in chiaro. Fornata dalle **Permutation Boxes**



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 4/8

Public (Asymmetric) Encryption: crittografia simmetrica (a chiave privata). Algoritmo RSA del 1973, presso il Government Communications Headquarters, ad opera di Clifford Cocks, desecretato nel 1997.

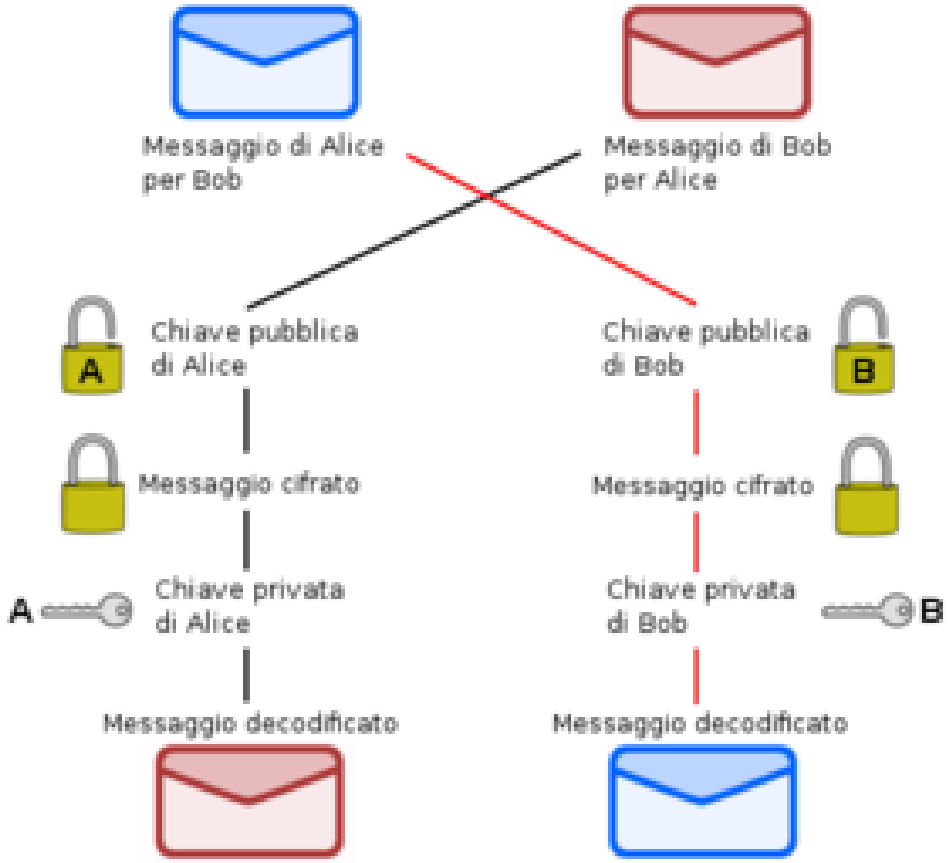
Encryption: $C = E(P, K_E)$
Es. $K_E = (17,23)$
RSA_{2048} ➔ (p_{1024}, q_{1024})

Decryption: $P = D(C, K_D)$
Impossibile desumere la chiave K_E (p, q) da K_D (p x q)
Es. $K_D = 17 \times 23 = 391$; $K_E = (17,23)$ per trovare KE occorre fattorizzare numeri primi

Utilizzo: possibile anche

- cifrare con K_D (chiave pubblica, conosciuta tra gli interlocutori)
- decifrare con K_E (chiave privata, conosciuta solo dal proprietario)

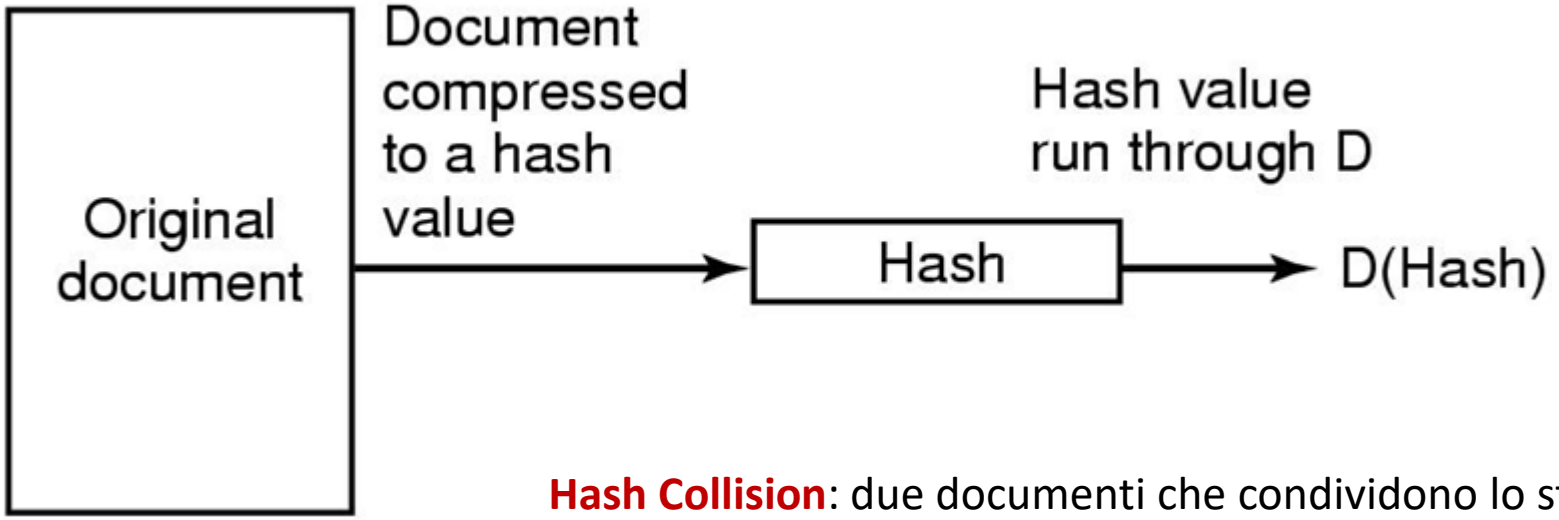
➔ diffusione delle chiavi pubbliche
➔ Cifratura molto più dispendiosa di quella simmetrica



Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 5/8

Hash: operazione irreversibile. Calcolo di un attributo del plaintext

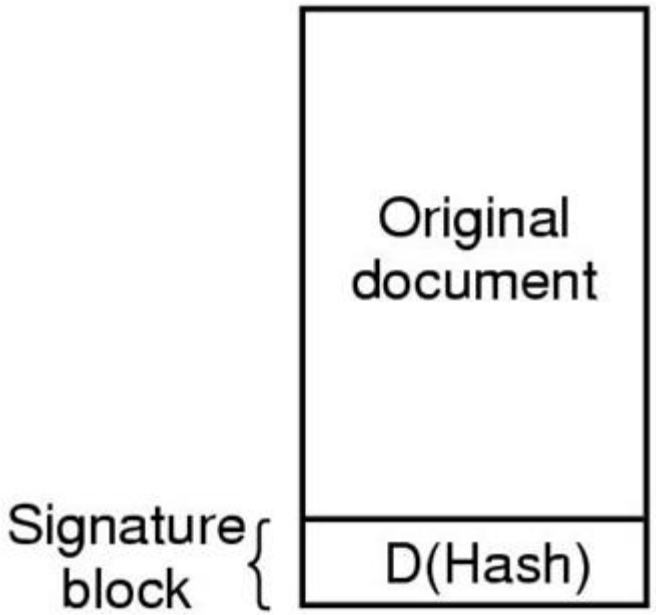


Hash Collision: due documenti che condividono lo stesso hash
Possibile poiché i documenti possono essere molto più lunghi dell'hash (ca. 256 bit)
es. MD5 and SHA- 1 sono algoritimi di uso diffuso ma vulnerabili alla hash-collision, basata su differential cryptanalysis

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 6/8

Digital Signature: «Firma» dell'hash effettuata tramite la chiave privata.



Signature: E(Hash, K_E). Per eseguire la verifica, è possibile:

- Ottenere Hash = D (E, K_D)
- Ricalcolare H (Original Document)
- Confrontarlo con Hash

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 7/8

Cypher Suite: insieme di algoritmi utilizzati per rendere sicuri i collegamenti di rete basati su Transport Layer Security (TLS).

TLS Specification: RFC 8446, Transport Layer Security 1.3

- Key Exchange
- Certificate Key ➔ Asymmetric Encryption
- Transport Cipher ➔ Symmetric + Block Chaining
- Integrity ➔ Hash


```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
                OU=Certification Services Division,
                CN=Thawte Server CA/Email=server-certs@thawte.com
        Validity
            Not Before: Aug  1 00:00:00 1996 GMT
            Not After : Dec 31 23:59:59 2020 GMT
        Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
                 OU=Certification Services Division,
                 CN=Thawte Server CA/Email=server-certs@thawte.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:d3:a4:50:6e:c8:ff:56:6b:e6:cf:5d:b6:ea:0c:
                    68:75:47:a2:aa:c2:da:84:25:fc:a8:f4:47:51:da:
                    85:b5:20:74:94:86:1e:0f:75:c9:e9:08:61:f5:06:
                    6d:30:6e:15:19:02:e9:52:c0:62:db:4d:99:9e:e2:
                    6a:0c:44:38:cd:fe:be:e3:64:09:70:c5:fe:b1:6b:
                    29:b6:2f:49:c8:3b:d4:27:04:25:10:97:2f:e7:90:
                    6d:c0:28:42:99:d7:4c:43:de:c3:f5:21:6d:54:9f:
                    5d:c3:58:e1:c0:e4:d9:5b:b0:b8:dc:b4:7b:df:36:
                    3a:c2:b5:66:22:12:d6:87:0d
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: md5WithRSAEncryption
        07:fa:4c:69:5c:fb:95:cc:46:ee:85:83:4d:21:30:8c:ca:d9:
        a8:6f:49:1a:e6:da:51:e3:60:70:6c:04:61:11:a1:1a:c0:40:
        3e:59:43:7d:4f:95:3d:a1:6b:b7:0b:62:98:7a:75:8a:dd:68:
        4e:4e:9e:40:db:a8:cc:32:74:b9:6f:0d:c6:e3:b3:44:0b:d9:
        8a:6f:9a:29:9b:99:18:28:3b:d1:e3:40:28:9a:5a:3c:d5:b5:
        e7:20:1b:8b:ca:a4:ab:8d:e9:51:d9:e2:4c:2c:59:a9:da:b9:
        b2:75:1b:f6:42:f2:cf:c7:f2:18:f9:89:bc:a3:ff:8a:23:2c:
        70:47
```

Digital Certificate x509v3

➔ Delega di responsabilità: identità

➔ PKI: Public Key Infrastructure

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Crittografia 8/8

PKCS (Public-Key Cryptography Standards): specifiche per la crittografia prodotte dai laboratori RSA Inc. dal 1991.

PKCS	Argomento	Standard Attuale
PKCS#1	Crittografia RSA	RFC 3447
PKCS#3	Scambio di chiavi Diffie-Hellman	RFC 2631
PKCS#5	Crittazione con Password	RFC 2898
PKCS#7	Firma e Cifratura di Messaggi in una PKI	RFC 2315
PKCS#8	Sintassi dell'informazione della chiave privata	RFC 5208
PKCS#9	Tipi di attributi selezionati usati nei PKCS	RFC 2985
PKCS#10	Sintassi delle richieste di certificazione: Certificate Sign Request (**CSR**)	RFC 2986
PKCS #11	Uniform Resource Identifier (URI)	RFC 7512
PKCS #12	Sintassi per il Trasferimento di Informazioni di Identità	RFC 7292
PKCS #15	Applicazioni Crittografiche nelle Smart Card	ISO/IEC 7816-15

Dipartimento di Ingegneria informatica
automatica e gestionale Antonio Ruberti

RFC: Request for Comment https://www.ietf.org/standards/rfcs/. IETF: Internet Engineering Task Force

Sapienza
Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 1/8

Identification: processo preliminare di assegnazione ad un utente di un identificativo univoco.

Operating Systems: Obiettivi Funzioni Servizi

User Authentication/Authorization

/etc/passwd ha tipicamente permessi di file system che gli permettono di essere leggibile da tutti gli utenti del sistema (world-readable), anche se può essere modificato solo dal superutente o utilizzando alcuni comandi privilegiati a scopo speciale

nish:x:501:501:Nishant Rastogi:/home/nish:/bin/bash

1 2 3 4 5 6 7

1. Username (login name)
2. Password («x» indica che è nel file /etc/shadow)
3. UID (User Identifier)
4. GID (Group Identifier)
5. General Electric Comprehensive Operating Supervisor
6. Home directory (Nome complete, Edificio, stanza, telefono d'ufficio, telefono di casa, indirizzo email, etc)
7. Shell di avvio ad ogni nuova connessione

Consentire l'accesso al sistema solo agli utenti che ne hanno diritto.

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 2/8

Authentication: procedimento atto ad attestare l'autenticità dell'utente che si appresta ad interagire col sistema.

3 Fattori

Knowledge: Qualcosa che sai (es Password, PIN)

Possession: Qualcosa che hai (es. Smart Card, SmartPhone)

Biometric: Qualcosa che sei (es. retina, impronte digitali)

Multifactor Authentication: uso di più tipi di fattori durante la stessa autenticazione

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 3/8

Password: codice stabilito dall'utente e memorizzato (generalmente in modo cifrato) sul sistema.

User Guessing: possibile stabilire i login-name nel caso (b)

Bobbie, 4238, e(Dog, 4238)
Tony, 2918, e(6%%TaeFF, 2918)
Laura, 6902, e(Shakespeare, 6902)
Mark, 1694, e(XaB#Bwcz, 1694)
Deborah, 1092, e(LordByron,1092)

LOGIN: mitch
PASSWORD: FooBar!-7
SUCCESSFUL LOGIN

(a)

LOGIN: carol
INVALID LOGIN NAME
LOGIN:

(b)

LOGIN: carol
PASSWORD: Idunno
INVALID LOGIN
LOGIN:

(c)

Encryption + Salt: in Unix le password sono conservate cifrate (encrypted) insieme ad un salt per impedirne il cracking.

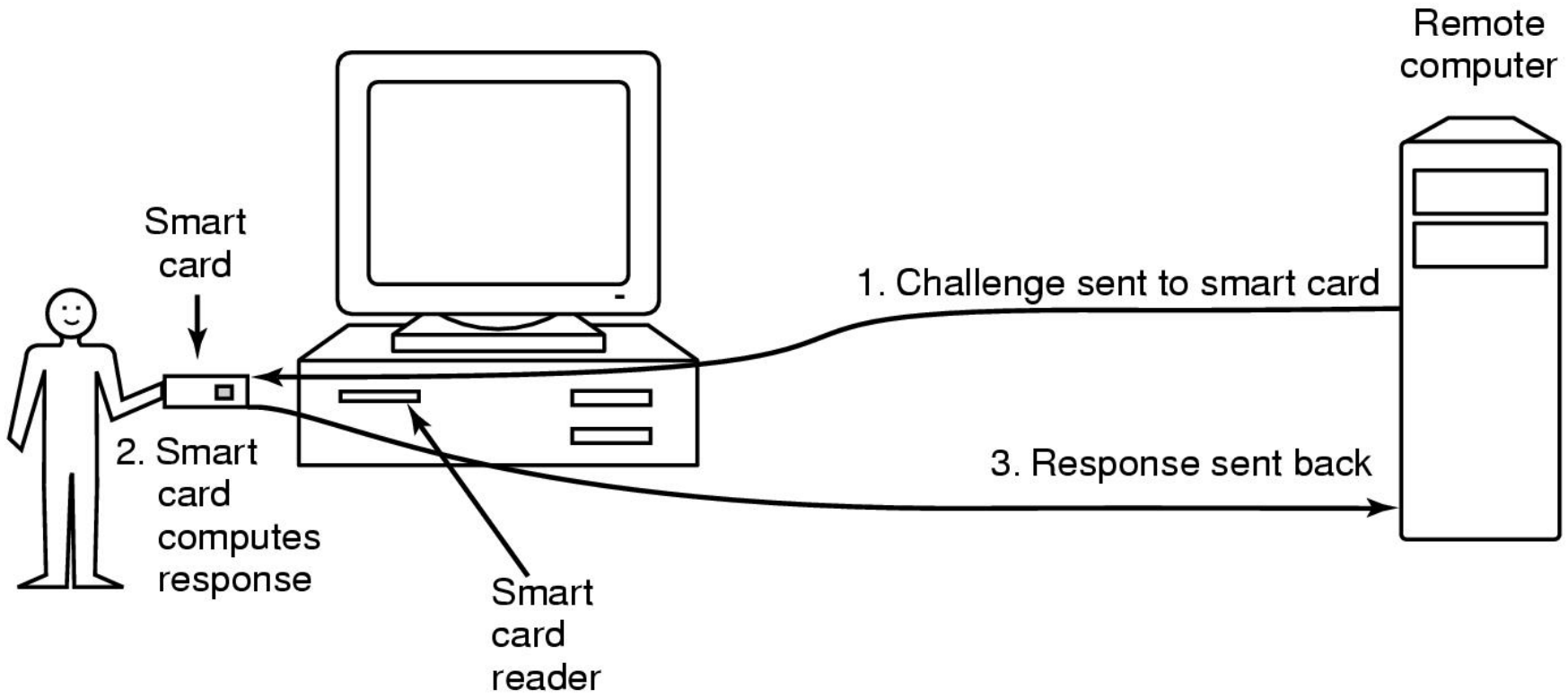
Password Quality (complexity): per rendere meno prevedili le password, queste devono avere alcune caratteristiche (cfr. `/etc/security/pwquality.conf`)

- **minlen**: lunghezza minima
- **minclass**: il numero minimo di tipologie di caratteri (maiuscole, minuscole, numeri, caratteri speciali)
- **dictcheck**: password non nel dizionario (cracklib)
- **gecoscheck**: password non contiene informazioni dell'utente presenti nei campi GECOS
- **usercheck**: password non contiene il login dell'utente in nessuna forma
- **remember**: numero di password contenute nella history (non riconfigurabili)
- **aging**: numero massimo di giorni in cui può essere usata una password prima di doverla cambiare

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 4/8

Physical Object: oggetto di cui si è in possesso. Prevede l'utilizzo di librerie (es. PAM: Pluggable Authentication Module)



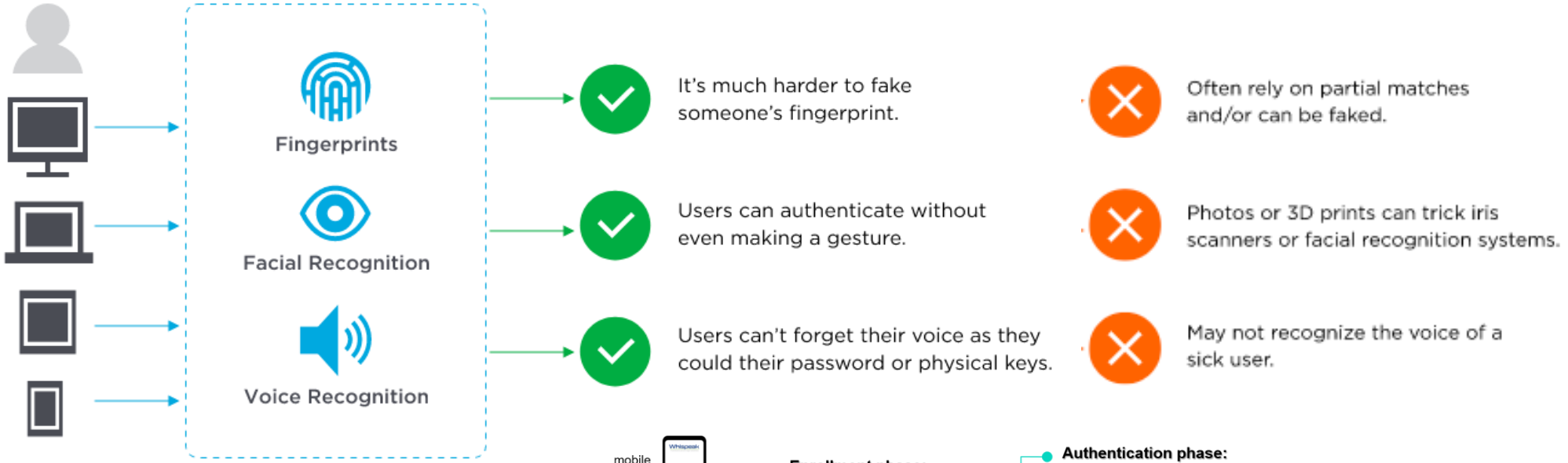
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

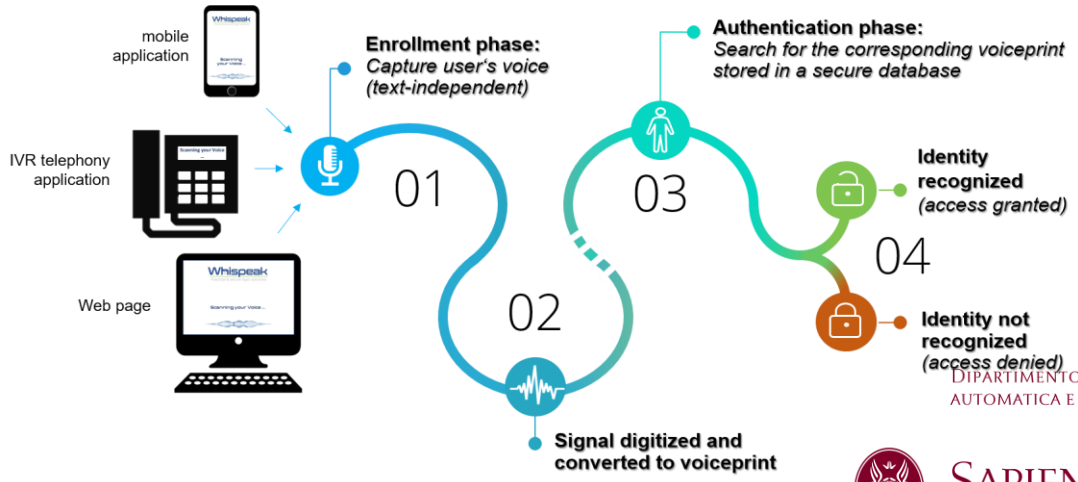
Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 5/8

Biometrics: proprietà dell'utente. Prevede l'utilizzo di librerie (es. PAM: Pluggable Authentication Module).



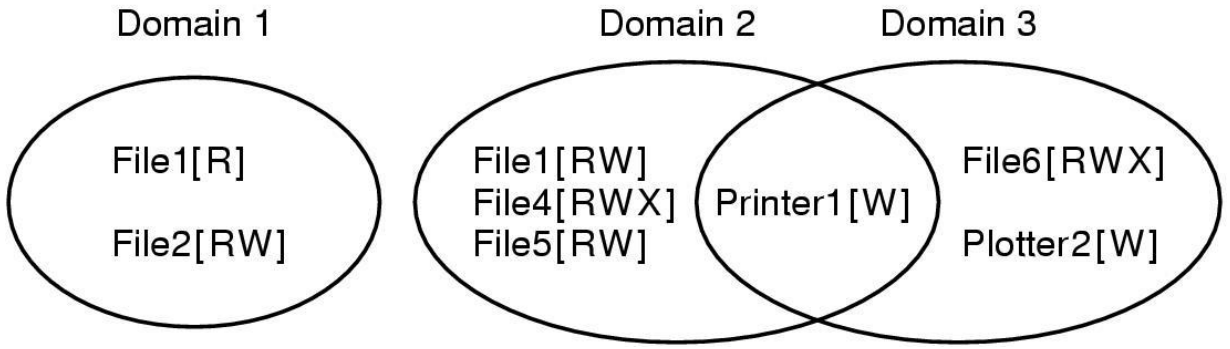
Enrollment: registrazione iniziale delle caratteristiche dell'utente



Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 6/8

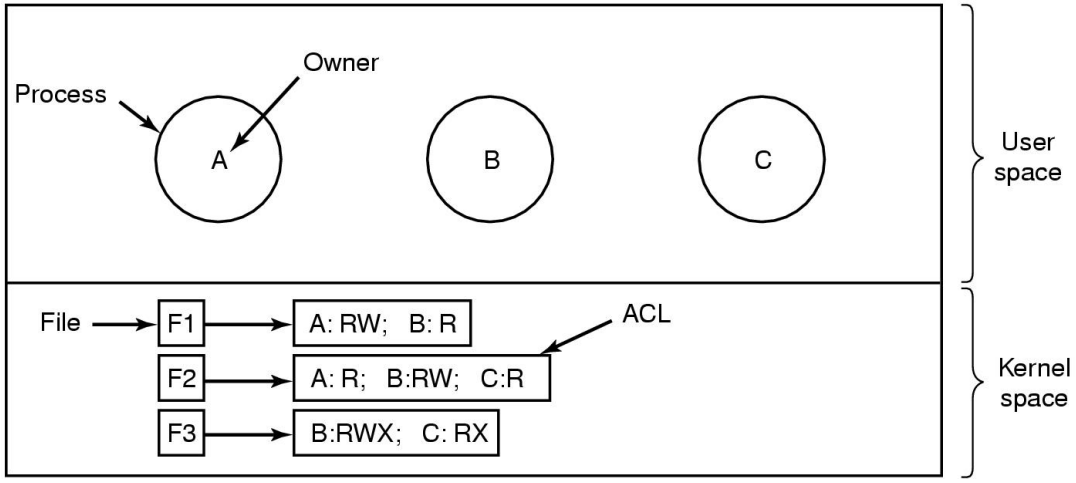
Permission: autorizzazione all'accesso mediante regole.



Domini: insiemi di risorse per organizzarne l'accesso

Domain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
	Object										
1	Read	Read Write								Enter	
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

Protection Domain: ...



ACL: Access Control List

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 7/8

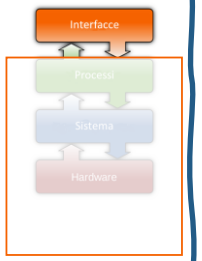
Log-Trails: registrazioni degli eventi che si sono succeduti sul sistema.

Log: registrazioni inerenti:

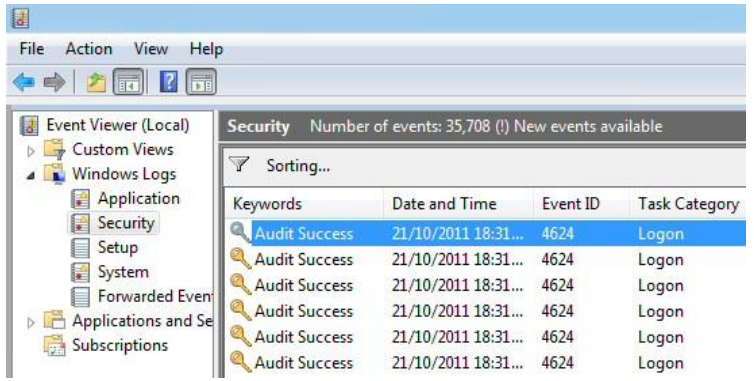
- Autenticazioni
- Accessi
- Operazioni effettuate
- Eventuali Errori verificatisi

Operating Systems: Obiettivi Funzioni Servizi

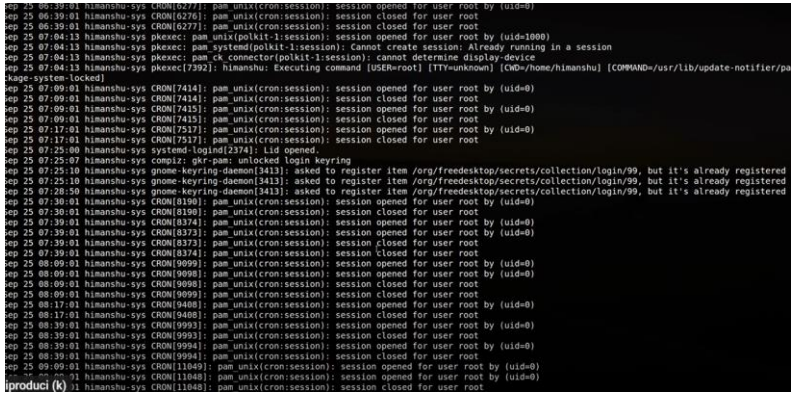
Accounting



Windows Security Log



Linux /var/log/auth.log



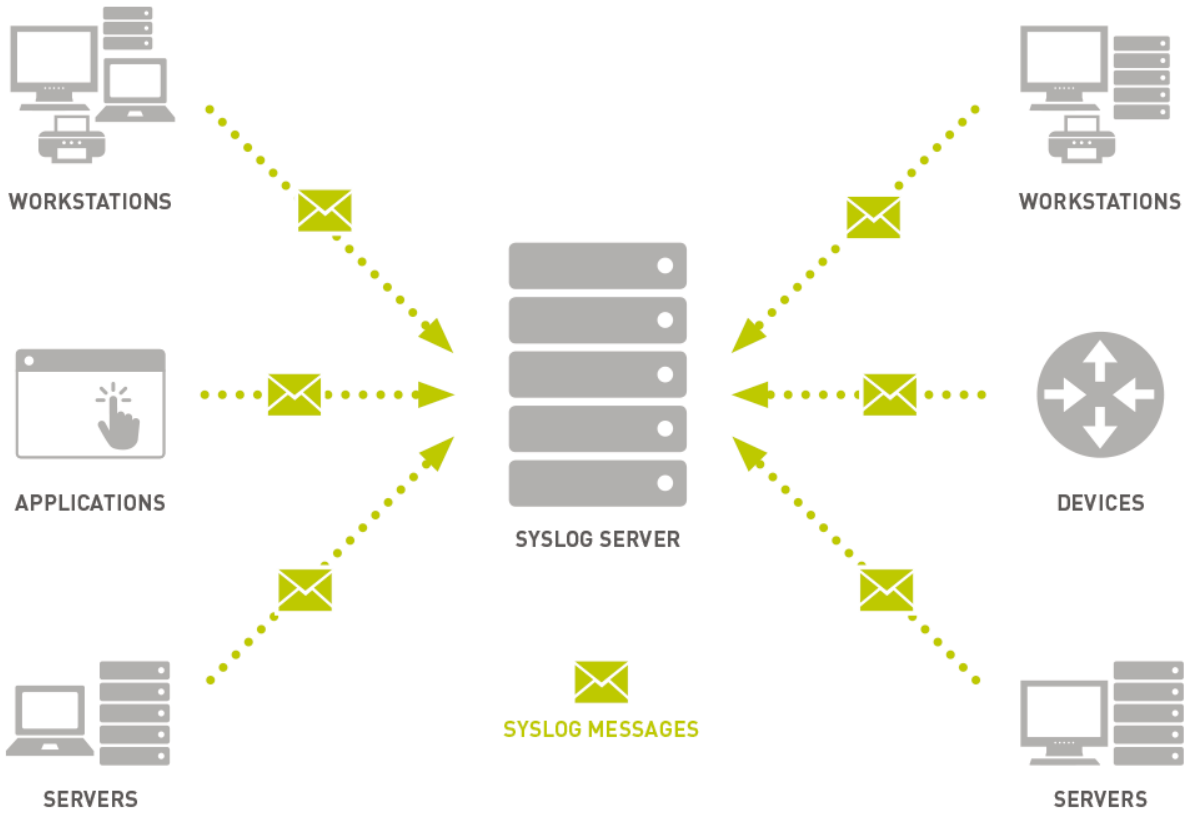
File di rendicontazione: /var/log/
Utmp, btmp, wtmp

Registrare gli accessi al sistema e poterli rendicontare (se necessario).

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) AAA 8/8

SysLog: protocollo di rete utilizzato per trasmettere attraverso una rete semplici informazioni di log.



Log: necessità di inviare i log al di fuori della macchina, in modo da preservare eventuali cancellazioni locali effettuate durante l'attacco.

Syslog è configurabile tramite apposita modifica del file /etc/syslog.conf, indicando ler ogni riga una attività di logging:

```
facility.loglevel /var/log/file.log
```

dove

- **Facility**: auth, auth-priv, cron, daemon, kern, local0-7, lpr, mail, news, user, syslog, uucp

- **LogLevel**: livello di criticità del messaggio: debug, info, notice, warning, error, crit, alert, emerg

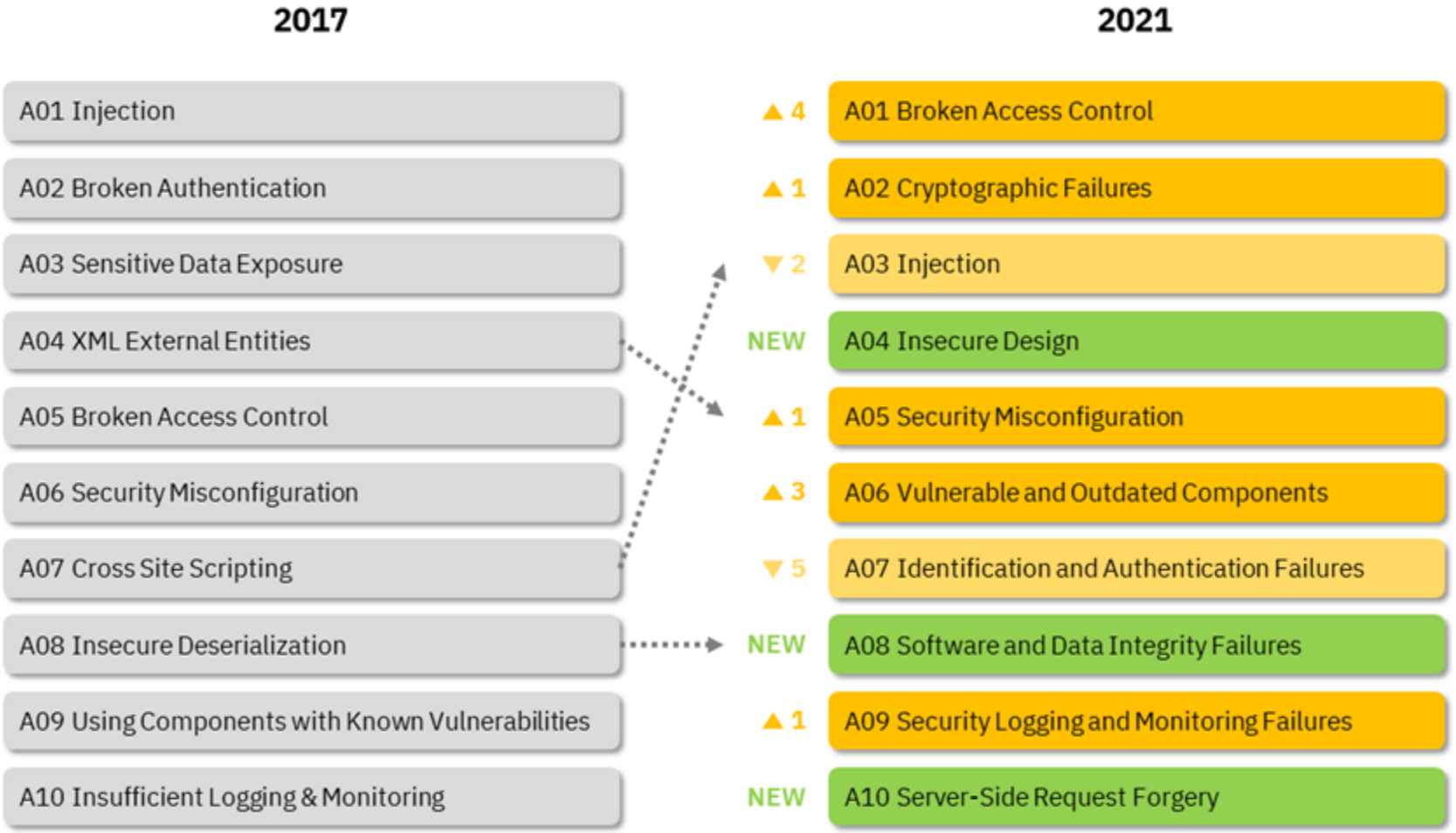
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Attacchi 0/4

OWASP: Open Web Application Security Project



1. **AAA**: **Permissions**
2. **Cryptography**: **SecretKey**, **Public Key**
3. **Attacchi**: **Code Injection**
4. **Protection**: **DEP**, **ASLR**
5. **Protection**: **ACL**, **Jailing**, **TCB**
6. **Protection**: **Updating**
7. **Cryptography**: **Digital Signature**
8. **Protection**: **Code Signing**
9. **AAA**: **Log-Trails, Syslog**
10. **Attacchi**: **Trap Doors/Bugs**

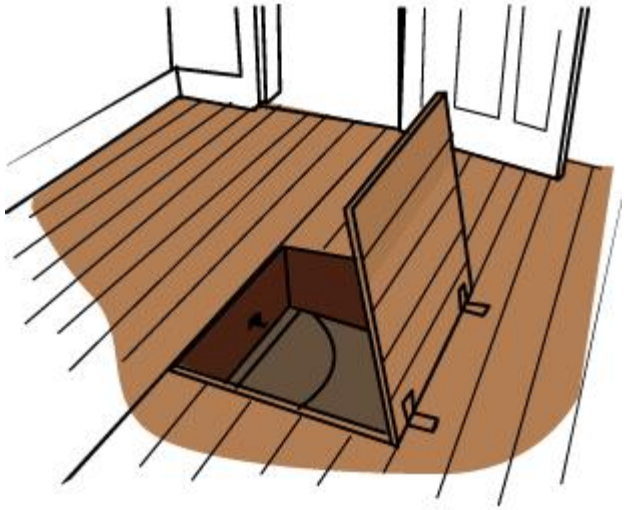
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Attacchi 1/4

Trap Doors (botola): punto di ingresso segreto che consente l'accesso senza le normali procedure di accesso in sicurezza



```
while (TRUE) {
        printf("login: ");
        get_string(name);
        disable_echoing( );
        printf("password: ");
        get_string(password);
        enable_echoing( );
        v = check_validity(name, password);
        if (v) break;
}
execute_shell(name);
```

(a)

```
while (TRUE) {
        printf("login: ");
        get_string(name);
        disable_echoing( );
        printf("password: ");
        get_string(password);
        enable_echoing( );
        v = check_validity(name, password);
        if (v || strcmp(name, "zzzzz") == 0) break;
}
execute_shell(name);
```

(b)

(a) Codice normale.
(b) Codice con trap door inserita

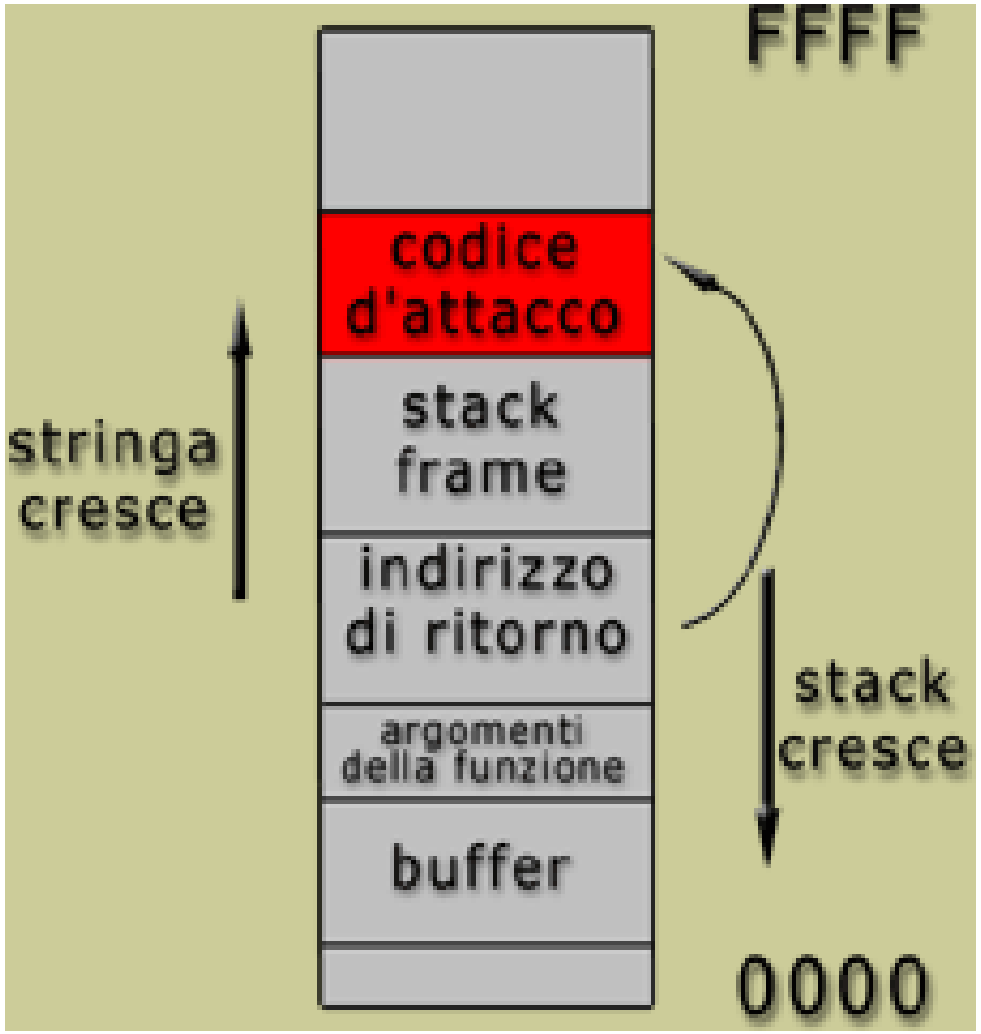
DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

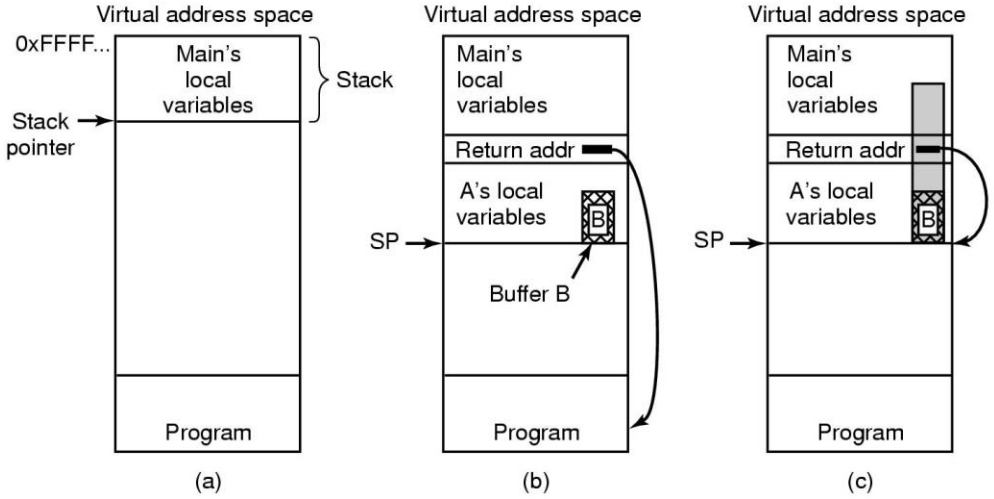
SAPIENZA
UNIVERSITÀ DI ROMA

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Attacchi 2/4

Buffer Overflow: traboccamento di un buffer, senza controllo sul limite dei suoi input, per via di troppi dati forniti.





sovvertire la normale progressione di un programma di modo che l'attaccante possa prenderne il controllo. Occorre:

1. Predisporre il codice adatto, da eseguire nello spazio d'indirizzamento del programma.

2. Permettere al programma di saltare a quel codice, con parametri esatti, caricati nei registri e nella memoria

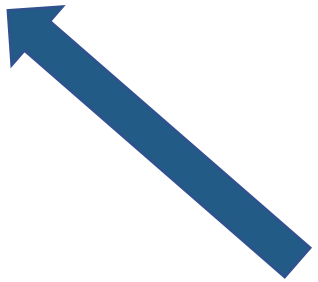
Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Attacchi 3/4

Code Injection: elaborazione di dati non validi, surrettiziamente inseriti tramite un bug del software.

```
int main(int argc, char *argv[])
{
  char src[100], dst[100], cmd[205] = "cp ";          /* declare 3 strings */
  printf("Please enter name of source file: ");       /* ask for source file */
  gets(src);                                          /* get input from the keyboard */
  strcat(cmd, src);                                   /* concatenate src after cp */
  strcat(cmd, " ");                                   /* add a space to the end of cmd */
  printf("Please enter name of destination file: ");  /* ask for output file name */
  gets(dst);                                          /* get input from the keyboard */
  strcat(cmd, dst);                                   /* complete the commands string */
  system(cmd);                                        /* execute the cp command */
}
```

No controllo su input

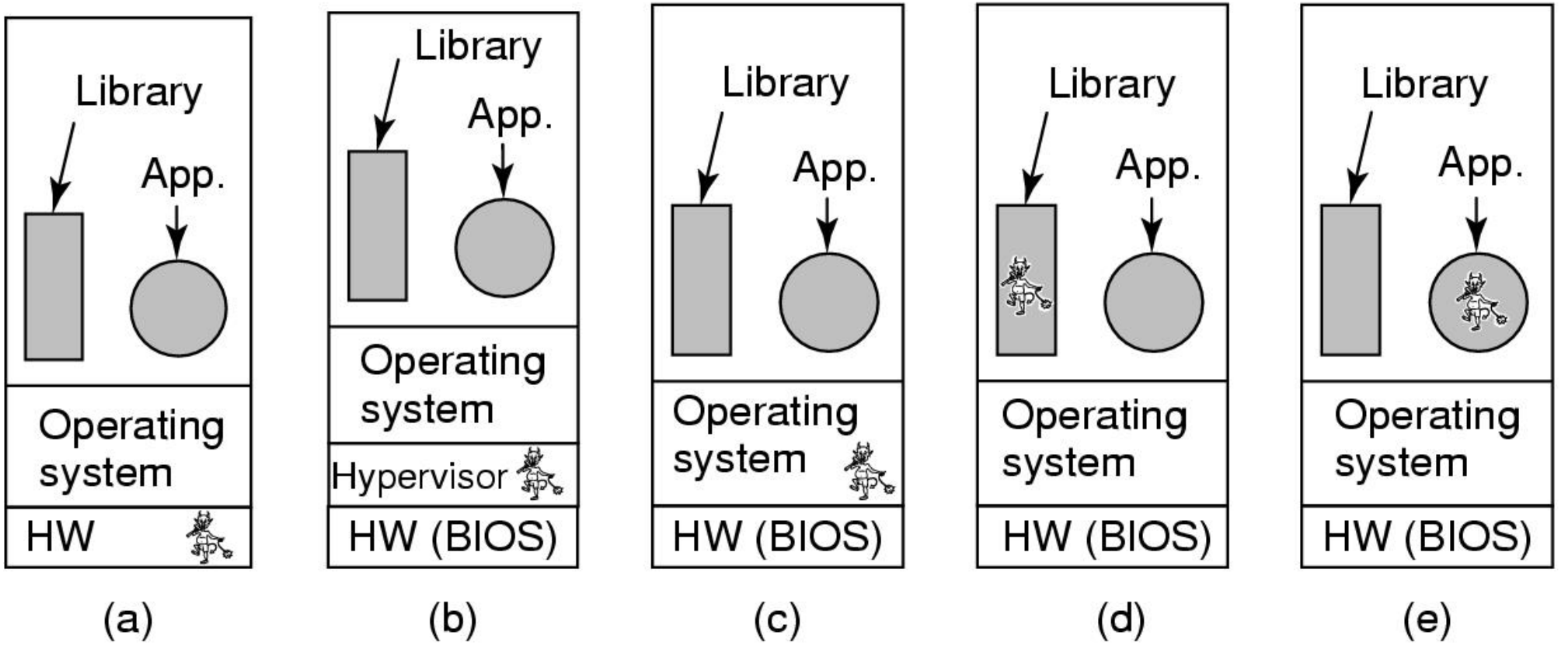


Utilizzo della stringa in input

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Attacchi 4/4

Rootkits: malware progettato per fornire a persone con scarsa capacità tecnica uno strumento di comando e controllo.



(a) Firmware rootkits

(b) Hypervisor rootkits

(c) Kernel rootkits

(d) Library rootkits

(e) Application rootkits

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

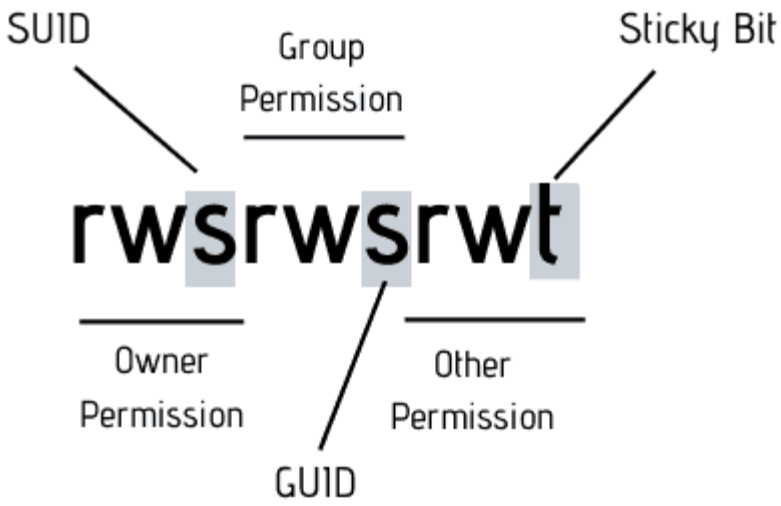
Operating Systems: Security & Protection

2. Asset Security: if (!minimizat

ACL ed altri permessi: Access Control List

ACL: permessi inclusi in una lista

File	Access control list
Password	tana, sysadm: RW
Pigeon_data	bill, pigfan: RW; tana, pigfan: RW; ...



Modifiche ai normali permessi di esecuzione (x):

SUID (s): Set User ID. Indica che il file va eseguito con i privilegi dell'utente proprietario del file anziché con quelli dell'utente che lo avvia. Non si applica alle directory.

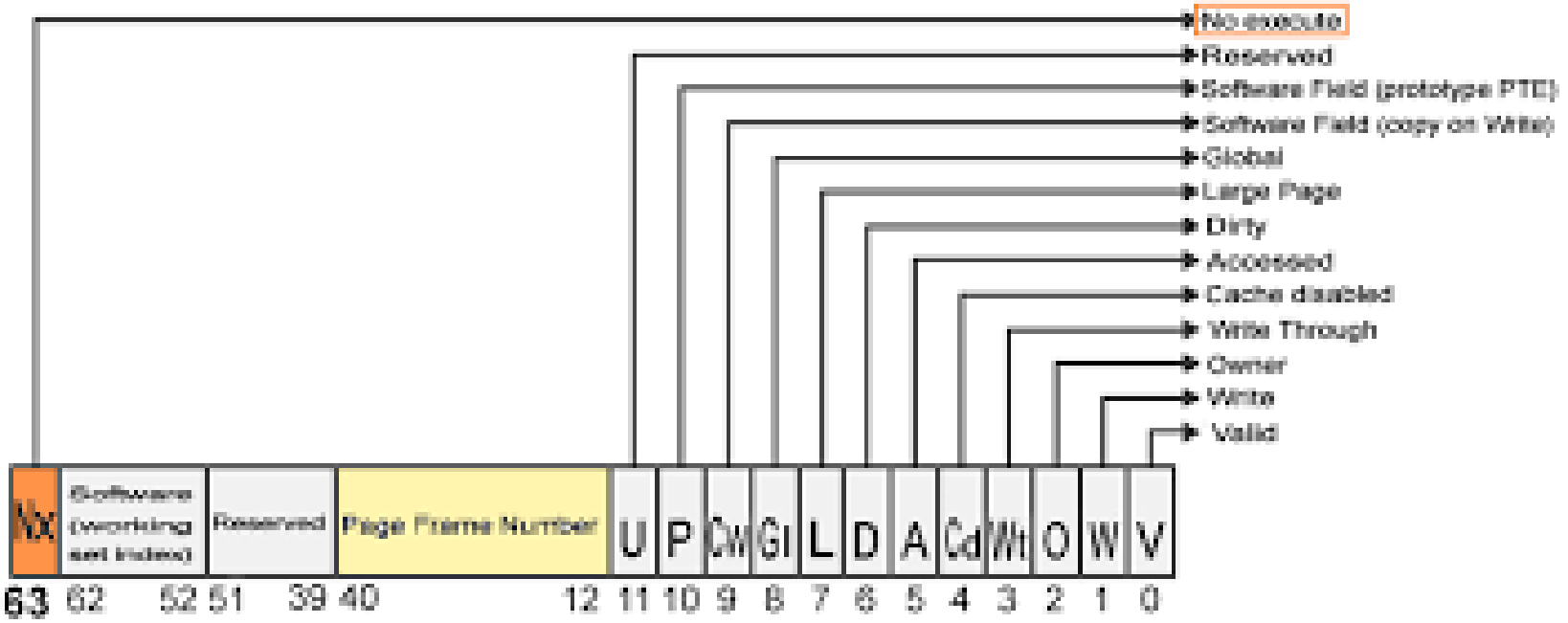
GUID (s): Set Group ID. indice che il file va eseguito con i permessi del gruppo assegnato al file anziché quelli del gruppo principale dell'utente che lo avvia. Non si applica alle directory.

Sticky (t): applicato ai file eseguibili, suggerisce al kernel di mantenere nel file di swap una copia del file eseguibile anche dopo che era terminato. Applicato alle directory, i file in essa contenuti possono essere cancellati e spostati solamente dagli utenti che ne sono proprietari, o dall'utente proprietario della directory che li contiene, o ancora dal superuser

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 2/6

NX: No eXecution. Inibizione della esecuzione di una certa frazione di dati. Detto anche Data Execution Prevention.



NX/DEP viene eseguito in due modalità:

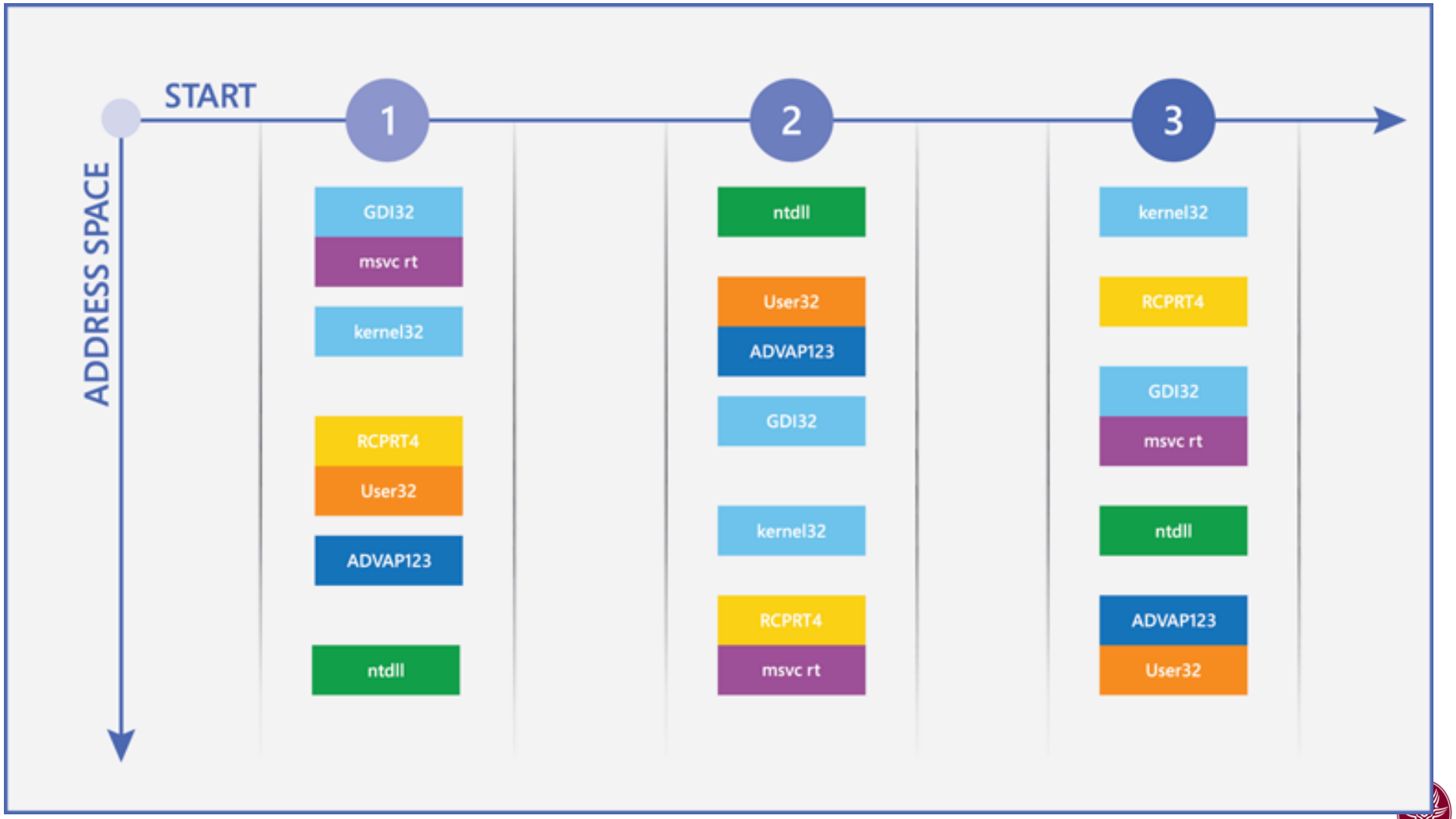
1. **NX bit**: DEP imposto dall'hardware per le CPU che possono contrassegnare le partizioni di memoria come non eseguibili e
2. **SW**: DEP imposto dal software con una prevenzione limitata per le CPU che non dispongono di supporto hardware. Il DEP applicato dal software protegge solo nella esecuzione di programmi compilati con Safe Structured Exception Handling (SafeSEH).

DEP è stato introdotto su Linux nel 2000, su Windows nel 2004 con Windows XP Service Pack 2, mentre Apple ha introdotto DEP nel 2006

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 3/6

ASLR: Address Space Layout Randomization. Posizionamento sempre diverso del processo in memoria, ad ogni esecuzione.



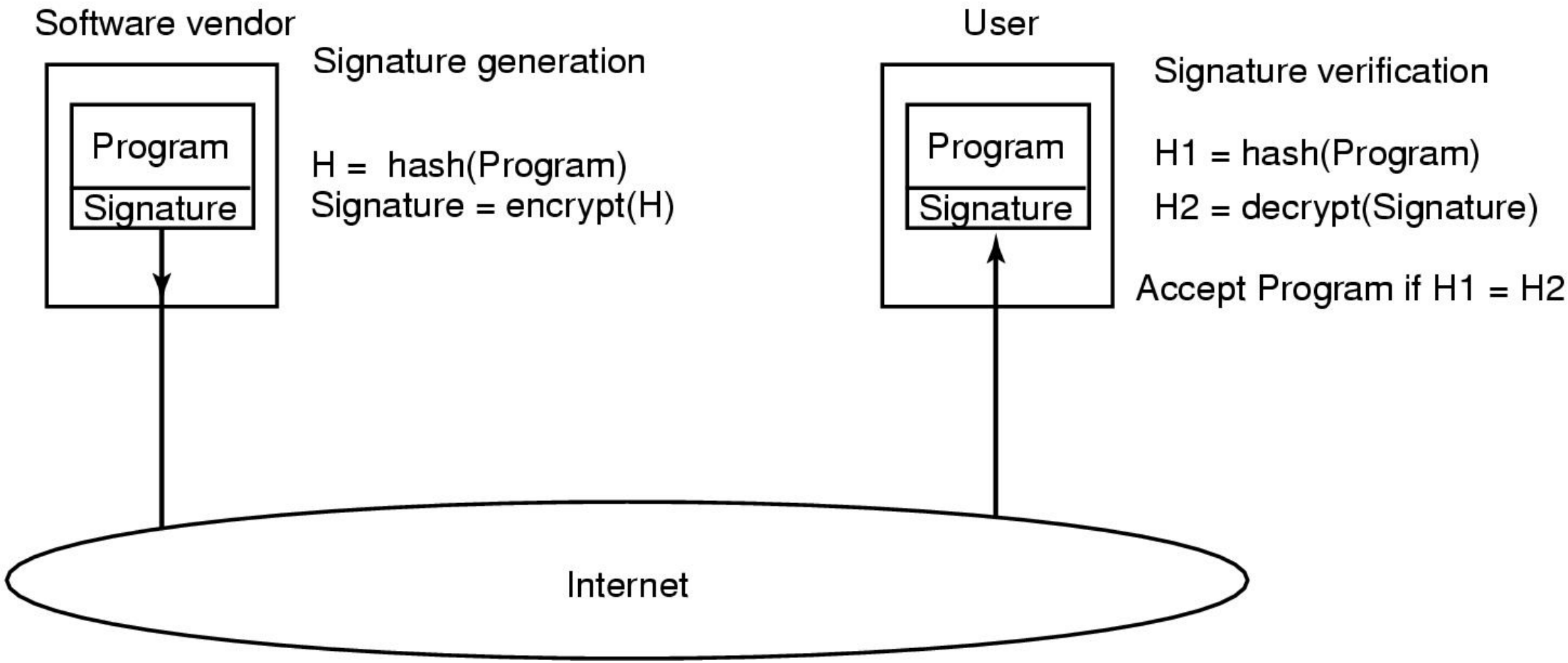
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 4/6

Code Signing: confermare l'autore del software, garantendo che il codice non sia stato alterato o corrotto dopo la firma.



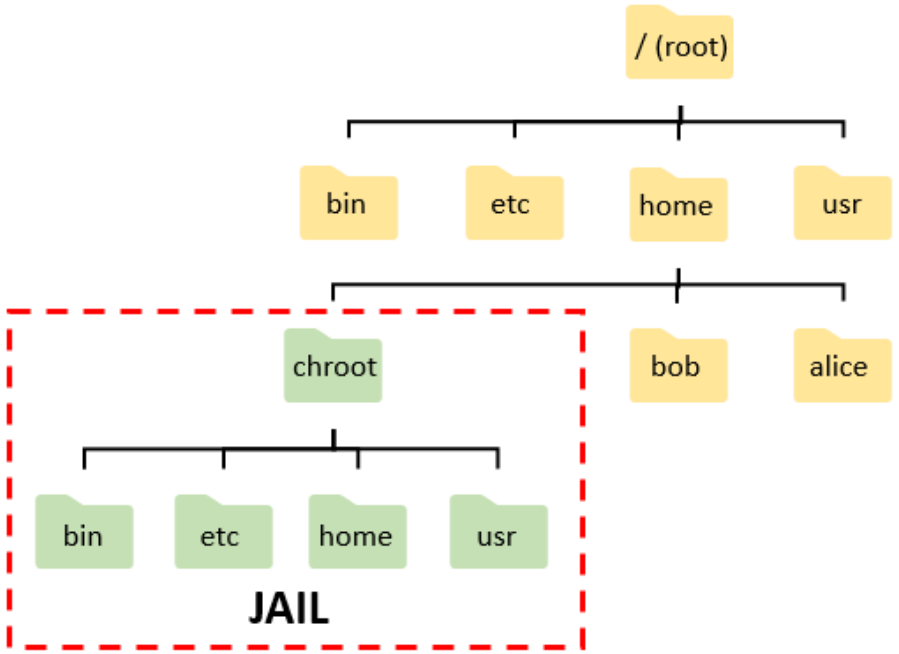
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 5/6

Jailing: modo per isolare un processo e i suoi figli dal resto del file system.
Sandboxing: modo per isolare un processo anche dal resto della memoria del sistema.



ChRoot: Change Root
Fare in modo che un processo creda che un sottoalbero sia l'intero file system.

Il file al di fuori di questo sottoalbero semplicemente non esiste.

Sandboxing: Change Root
Fare in modo che un processo sia impedito dall'accedere in frazioni di memoria non di suo stretto interesse (comprese chiamate di Sistema, che espongono dati di altri processi).

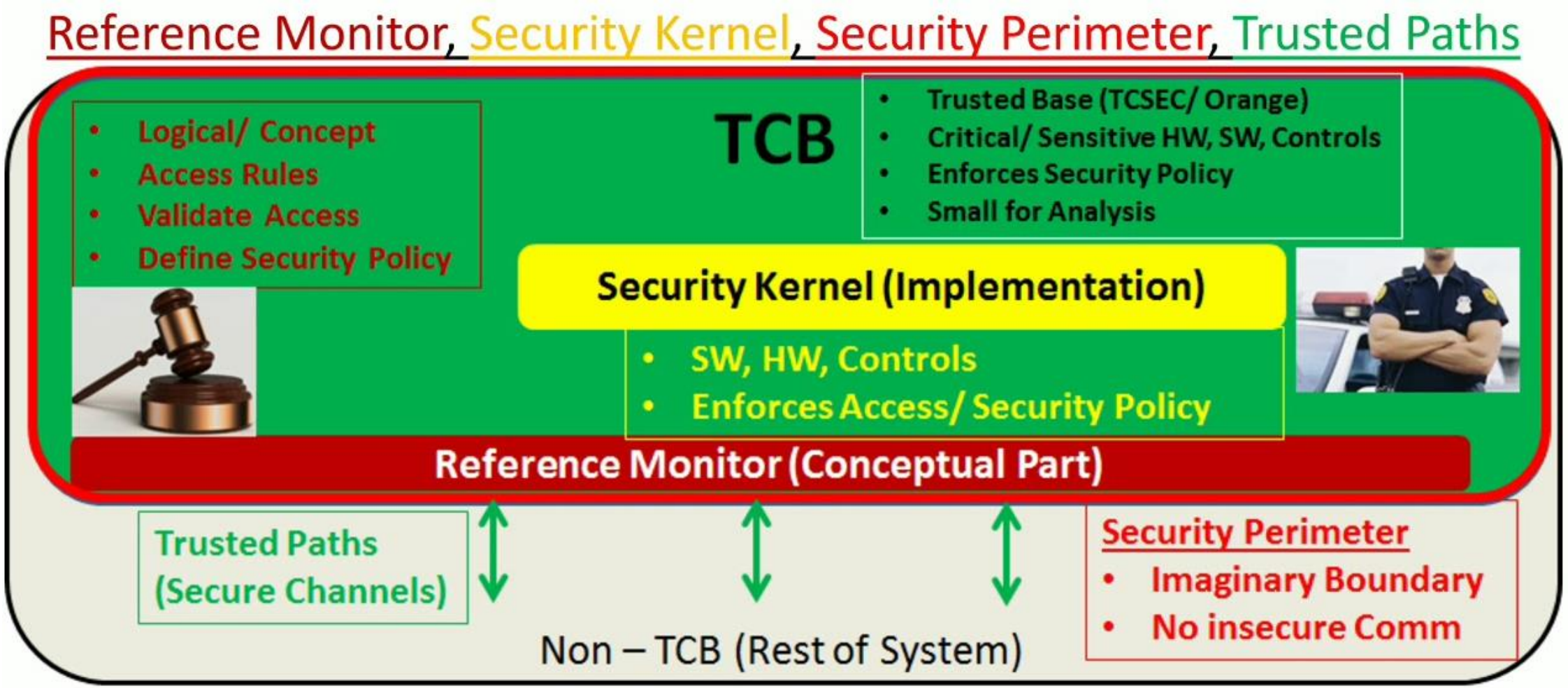
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 6/6

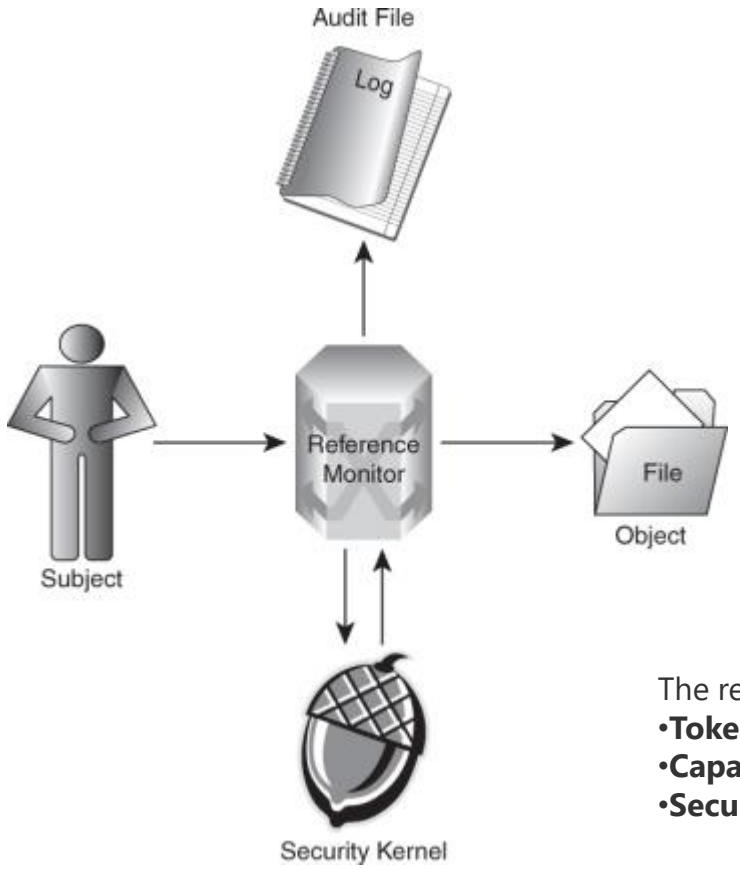
TCB: Trusted Computing Base.



Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 6/6

Reference Monitor: abstract machine that is used to implement security



The *reference monitor* properties:

- **Job**: to validate access to objects by authorized subjects.
- **Position**: the boundary between the trusted and untrusted realm (**Security Perimeter**).
- **Properties**:
  1. Cannot be bypassed and controls all access
  2. Cannot be altered and is protected from modification or change
  3. Can be verified and tested to be correct
- **Role**: to verify the subject meets the minimum requirements for access to an object

The reference monitor can be designed to use tokens, capability lists, or labels.
•**Tokens**—Communicate security attributes before requesting access.
•**Capability lists**—Offer faster lookup than security tokens but are not as flexible.
•**Security labels**—Used by high-security systems because labels offer permanence. This is provided only by security labels.

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

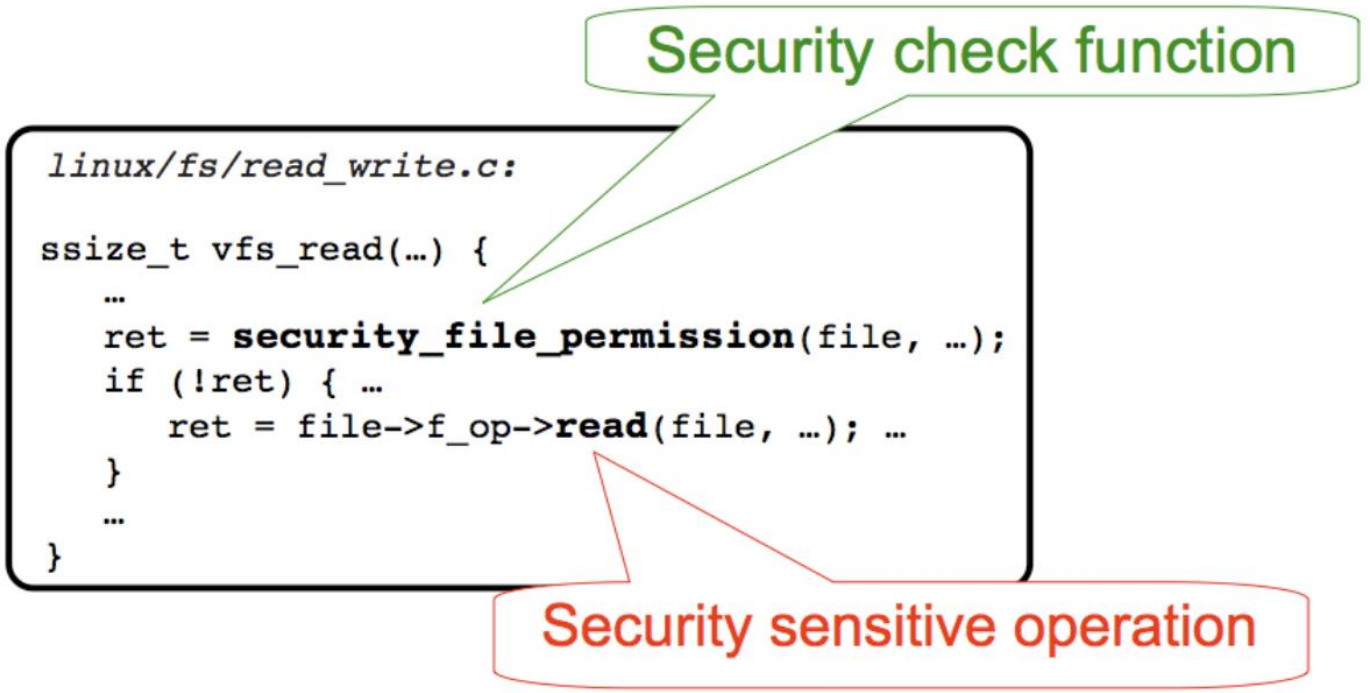
Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 6/6

Reference Monitor: → **LSM** (Linux Security Module). Framework for designing Reference Monitor

The *Linux Security Module* properties:

- **Implementation**: Linux Kernel module. To be loaded and explicitly referenced.
- **Referencing**: function to allow modules to register (and unregister) as security module.
- **Opaque Security Fields**: for associating security information to kernel objects. Implemented as `void*`.
- **Security Hooks**: function call inserted at various point in the kernel code. These override function calls to manage security fields and mediate access to kernel objects. Per packet, superblock, shared memory, processes. Called via function pointer stored in `security->ops`.

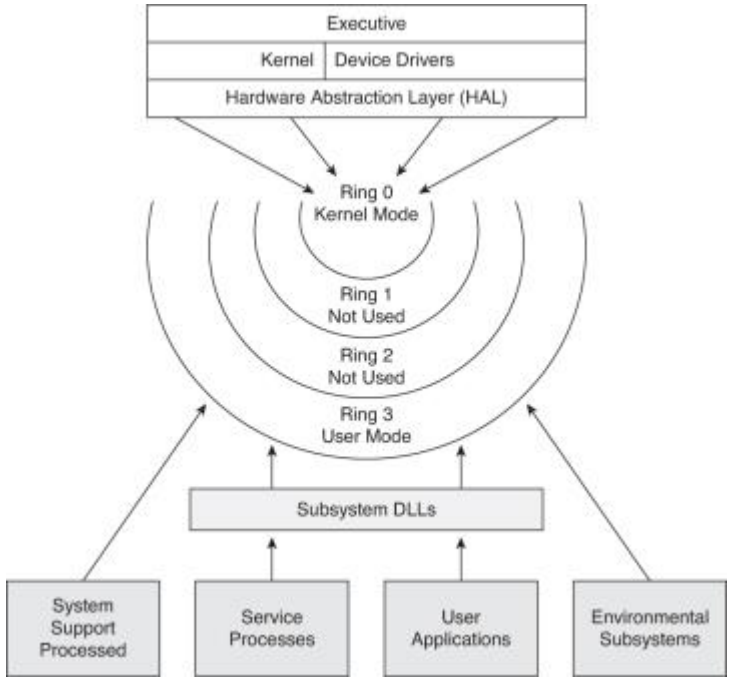


- **Trusted Path Execution**: denies users from executing programs that are not owned by root, or are writable.

Operating Systems: Security & Protection

2. Asset Security: if (!minimization()) Protezione 6/6

Security Kernel: hearth of the system



The *Security Kernel* properties:

- **Resources**: handling all user/application requests for access to system resources
- **Responsibility**: running the required controls used to enforce functionality and resist known attacks
- **Size**: of the security kernel:
  - Small: easy to verify, test, and validate as secure
  - Real-Life: processes located inside can function faster and have privileged access (performance gain as in Windows and Linux)
  - Can be verified and tested to be correct
- **Role**: to verify the subject meets the minimum requirements for access to an object

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

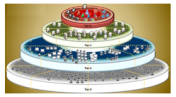
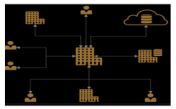
SOReCa – Sicurezza: Security & Protection (8)

3. Sec Arch & Engineering

Tier Model (cfr. «ZTA Pillars»)

SOReCa – Sicurezza: ZTA Pillars (5)

ZTA: Evolution of Trust Models & Topologies

	Years	Name	Fashion	Remote	Description	Trust	Tools	Drawback
	'90s	**Tier Model** strict separation of assets	«Circles of Hell»	No / a Few	logical separation of assets by boundaries in the same physical location (old-fashioned **Perimeter-Centric**).	Inside Yes, Outside. No Delegation Model	FW IDS	No Remote
	'00s	**Hub & Spoke** connect outlying points to a central "hub".	«Airline Routes»	Some	remote connections secured by VPN tunnels (strong pub-key cryptography) converging at one location (**Centralized Branch Office**)	Outside could get as Inside Central Visibility & Control	VPN SSL-VPN VDI RDP	Bottleneck and SPoF
	'20s	**Zero Trust** Authentication GW Distribution	«Never Trust, Always Verify»	Most	connections are granted after careful verification (Identity, Device, Time, Geolocation, Security Posture (**Default Deny**)	per-transaction basis. Pervasive Telemetry	PEP (CASB, ATP, DLP, …) ➔SASE	Distributed network of PoPs

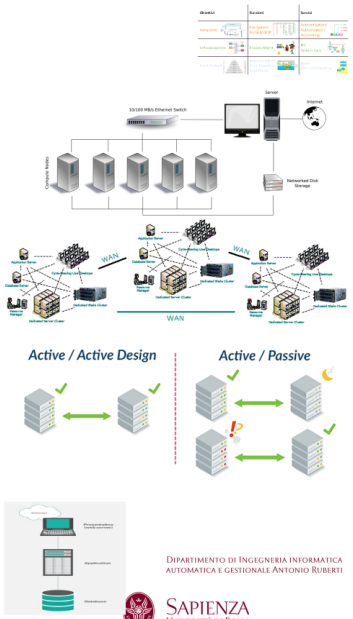
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti
Sapienza Università di Roma

Operating Systems: Organizzazione

Client/Server (Distribuito): es. Clusters

In generale, un cluster si riferisce a un insieme di computer collegati tra loro.

- **High Performance Computer** (**HPC**): fisicamente situati uno vicino all'altro, al fine di risolvere I problemi in modo più efficiente. Generalmente, eseguono la stessa imagine di Sistema Operativo.
- **Grid Computing**: uso di una griglia computazionale (stazioni di lavoro, server blade, ecc.) applicando le risorse della griglia, tramite rete, a un singolo problema allo stesso tempo, mentre si superano i confini politici e teorici.
- **High Availability** (**HA**): un sistema informatico funge da sistema di backup per uno o più sistemi primari, tutti situati uno vicino all'altro. Quando c'è un guasto in un sistema primario, le applicazioni critiche in esecuzione su quel sistema vengono trasferite al sistema di backup designato.
  - **Load Balancing** (**LB**): come HA ma tutti i sistemi funzionanti ed aderenti al cluster dividono il carico di lavoro (Attivo/Attivo).
  - **Cluster Geografico**: come HA ma i sistemi sono situati a distanza.
- **Three Tier**: architetture si sistema a 3 livelli: presentazione (web Server), elaborazione (App), dati (DB). Ognuno su server diversi, in HA/LB e SO distinti



Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti
Sapienza Università di Roma

Clustering (cfr. «04-SOReCa-IPC»: Clusters)

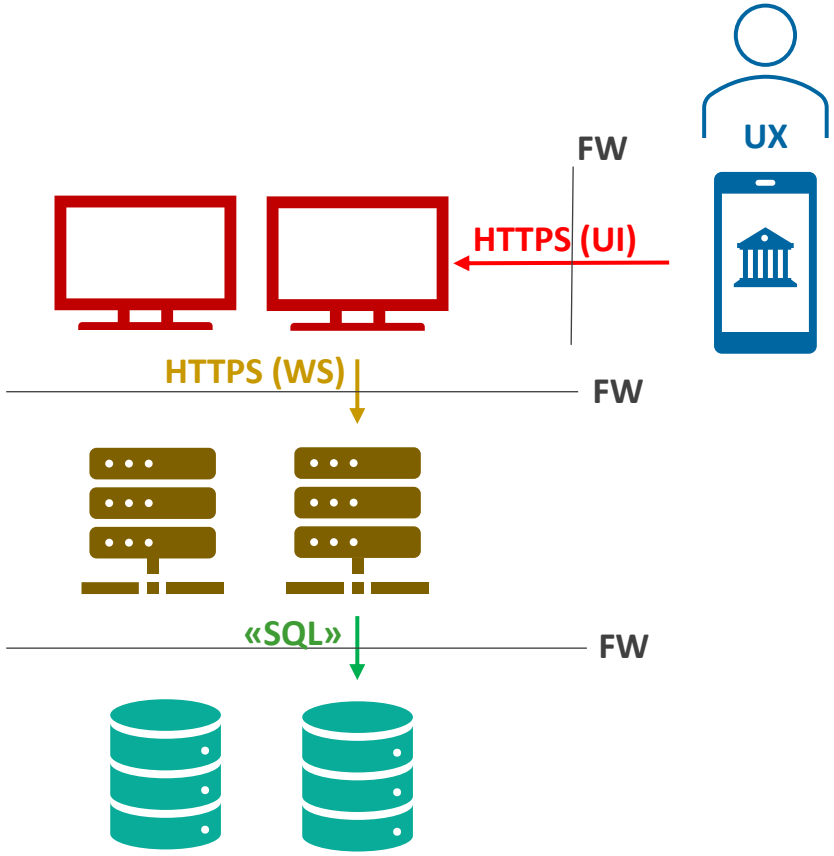
Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti
Sapienza Università di Roma

SOReCa – Sicurezza: Security & Protection (8)

3. Sec Arch & Engineering

Distributed elaboration: **Three Tier**: system architecture based on the segregation into 3 layers, each having its peculiarities:

1. Presentation (Web Server),
2. Elaboration (App Server),
3. Data (DB Server).



Each one on different item instance, hosts and networks (separated by Firewalls)

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Analysis of Cyberattacks(13)

MITRE ATT&CK

SOReCa – Sicurezza: Analysis of Cyberattacks (13)

MITRE ATT&CK

ATT&CK is a guideline for classifying and describing cyberattacks and intrusions, created by the Mitre Corporation (2013).

1.Reconnaissance: gathering information to plan future adversary operations, i.e., information about the target organization
2.Resource Development: establishing resources to support operations, i.e., setting up command and control infrastructure
3.Initial Access: trying to get into your network, i.e., spear phishing
4.Execution: trying the run malicious code, i.e., running a remote access tool
5.Persistence: trying to maintain their foothold, i.e., changing configurations
6.Privilege Escalation: trying to gain higher-level permissions, i.e., leveraging a vulnerability to elevate access
7.Defense Evasion: trying to avoid being detected, i.e., using trusted processes to hide malware
8.Credential Access: stealing accounts names and passwords, i.e., keylogging
9.Discovery: trying to figure out your environment, i.e., exploring what they can control
10.Lateral Movement: moving through your environment, i.e., using legitimate credentials to pivot through multiple systems
11.Collection: gathering data of interest to the adversary goal, i.e., accessing data in cloud storage
12.Command and Control: communicating with compromised systems to control them, i.e., mimicking normal web traffic to communicate with a victim network
13.Exfiltration: stealing data, i.e., transfer data to cloud account
14.Impact: manipulate, interrupt, or destroy systems and data, i.e., encrypting data with ransomware

ATT&CK identifies tactics that indicate an attack is in progress (→ to be used in SOC)

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

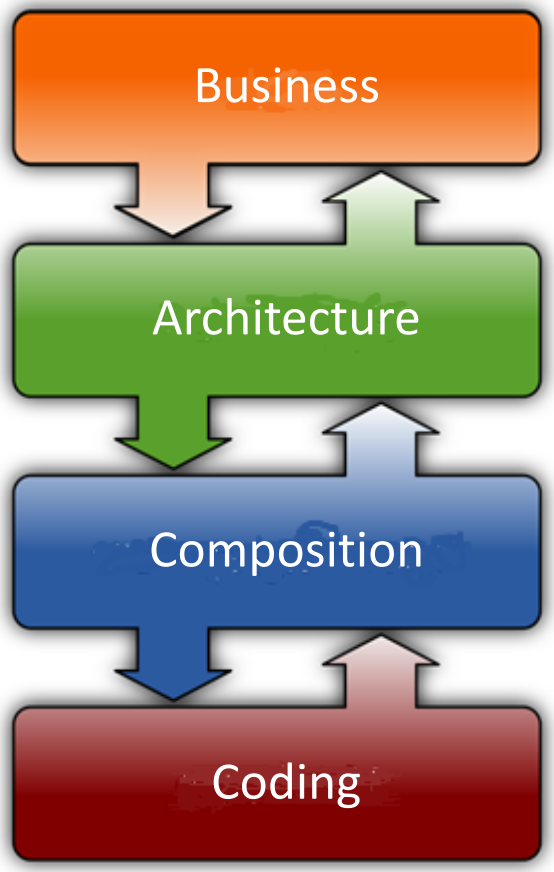
SAPIENZA UNIVERSITÀ DI ROMA

Design Shutters (21)

Modello per DevSecOps

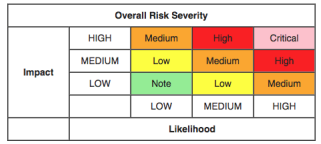
21 Framework: Design Shutters

Putting All Together

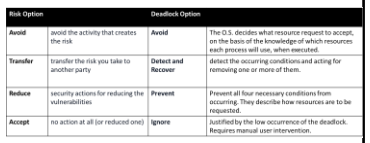


Key Indicators

Attackers



Risk Rating



Risk Mgmt

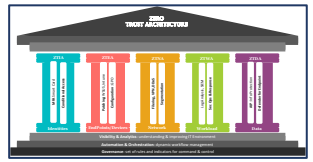
Enforce

Plan
Monitot

Users

DAST

Arch Mgmt



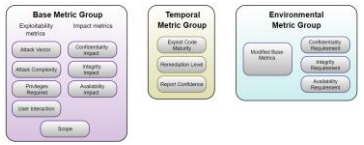
ZTA Pillars

Operate
Deploy

Hack Profiles

SCA

Vuln Mgmt



CVE/CVSS

Release
Test

(Audit) Log

SAST

Input Mgmt

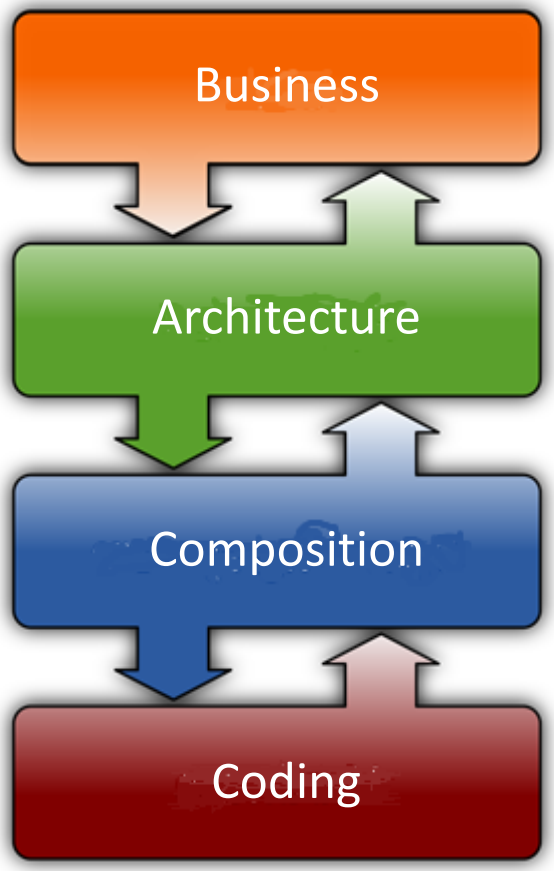
Access Control

Build
Code

Identify **Evaluate** **Decide** **Protect** **DevOps**

21 Framework: Design Shutters

Putting All Together



- KPI
- SLA
- KGI
- KRI

Key Indicators

Identify (6)	Evaluate (3)	Decide (4)	Act (5)	DevOps (2)
Attackers • Insider • Competitor • Crime • Hacktivist • Warfare • Terrorist	**Risk Rating** • Likelihood • Impact • Level	**Risk Mgmt** • Avoid • Transfer • Mitigate • Accepts	**Enforce** • AAA • Duplicate • Filter • Log • Encode	**Plan** **Monitor**
Users • Customer • Employee • Partner • App2App • SysAdmin • Developer	**DAST** • Explore • Test • Evaluate	**Arch Mgmt** • WAF • Supplier • Implement • Ignore/Postpone	**ZTA (Pillars)** • Identity • Endpoint • Network • Workload • Data	**Operate** **Deploy**
Hackers • Wannabe Lamer (Script Kiddie) • Cracker • Ethical Hacker • Cyber Warrior • Quiet, Paranoid, Skilled Hacker • Industrial Spy, Gvnmt Agency	**SCA** • Identify Dependecies • Vulns (OSInt, ClOSInt) • Speed	**Vuln Mgmt** • Substitute • Virtual Patch • Patch • Ignore/Postpone	**CVE/CVSS** • Description • Severity • References • Weaknesses • Configuration	**Release** **Test**
(Audit) Log • Date, Time • Identity • Device • Net Addr, Prot • Location • Event/Activity	**SAST** • Scan • Prioritize • Verify	**Input Mgmt** • Checking Whitelist • Sanitizing Escape • Checking Blacklist • Sanitizing Blacklist	**Access Control** • Identify • AuthN • AuthZ • Govern (Certify) • Monitor	**Build** **Cod**