Introduction

INFORMATICS AND ORGANIZATIONAL SECURITY

Security

Subject focused on the predictability of systems, processes, environments…

Across all aspects of the life cycle:

- Planning
- Development
- Execution
- Processes
- People
- Clients and Supply Chain
- Mechanisms
- Standards and Laws
- Intellectual Property

Security: Planning

Design of a solution complying with some requirements under a normative context

Without flaws

- All operation states are the ones predicted
- There are no additional states escaping the expected logic
  - Even if forced transitions are used

Under the scope of a normative context

- Specific for each activity or sector
- Ex: ISO 27001, ISO 27007, ISO 37001

Security: Development

Implement a solution complying with the design, without other operation modes

Without bugs compromising the correct execution

- No crashes
- Without invalid or unexpected results
- With the correct execution times
- With adequate resource consumption
- Without information leaks

Software:

- Requires careful implementation
- Requires tests to obtain an implementation with the expected… and only the expected behavior

Security: Execution

Code executes as it was written, with all predicted processes

Environment is controlled, cannot be manipulated or observed

Without the existence of anomalous behavior, introduced by environmental aspects

- Such as: storage speed, RAM amount, trusted communications

Source: https://www.verificient.com/

ISO 27001 – Clean Desk Policy

Security: people and partners

Staff behavior cannot have a negative impact to the solution

Norms are in place to regulate what actions are expected

Staff is trained to distinguish correct from incorrect behavior

Staff has the correct incentives to behave adequately

When staff is compromised, or deviate, actions have limited impact

Security: Analysis and Auditing

What is the actual behavior of the solution?

Identify deviations from the expected atributes

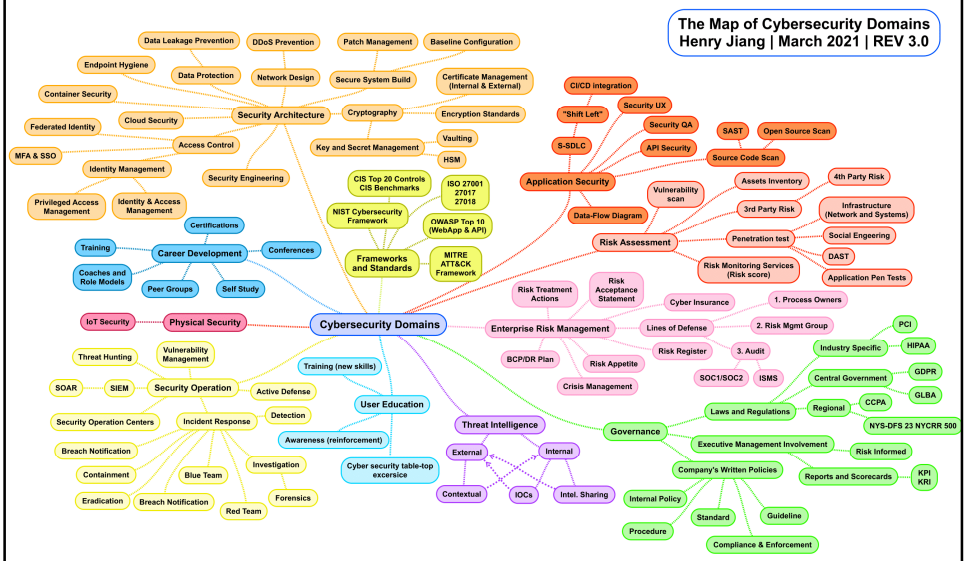
- Faults, errors, behavior

Identify the risk for the solution to be modified

- Exposition to possible attackers
- Incentives one may have to modify it
- Identify potential actors (threats)

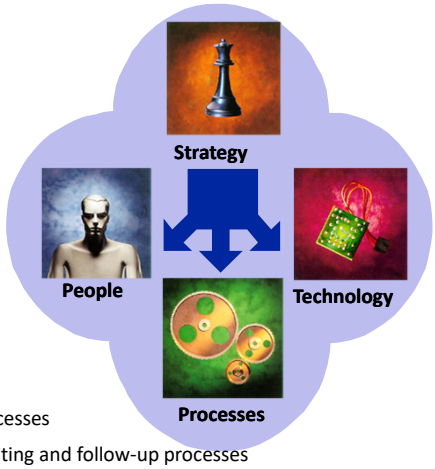
Identify the impact of the deviations

- Total loss of data? Denial of Service? Increase Operation Cost?



Dimensions to consider

- Selection
- Training
- Awareness
- Organization of security



- Vulnerability scanning
- Firewalls
- Authentication
- Access Control
- Cryptography
- Digital Signatures
- Certification authorities
- Certification hierarchies
- etc...

- Security policies
- Security administration processes
- Continued evolution of auditing and follow-up processes

Perspectives

Security has multiple intertwined perspectives

Defensive: focus on maintaining predictability

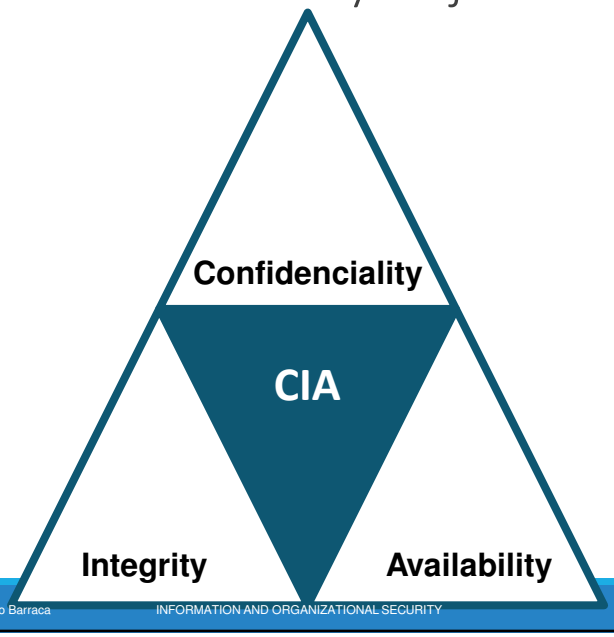
Offensive: focus on exploiting predictability

- May have malicious/criminal intent
- May have the purpose of validating the solution (Red Teams)

Other:

- Reverse Engineering: Recovery of design from built products
- Forensics: extract information and reconstruct previous events
- Disaster Recovery: minimize the impact of attacks
- Auditing: validate the solution complies with some set of requirements

Information Security Objectives



Information Security Objectives

Confidentiality: Information may only be accessed by a restricted group of entities

Measures:

- Encrypt information
- Use access passwords (strong)
- Use Identity Management and Authentication systems
- Doors, Strong walls
- Security personel
- Training

Information Security

Integrity: Information remains unchanged

- Can be applied to behavior of devices and services

Measures:

- Identity control (hashes)
- Backups
- Access Controls
- Robust Storage Devices
- Data verification processes

Information Security

Availability: Information is available to target entities

- Can be applied to services and devices

Measures:

- Backups
- Disaster recovery plans
- Redundancy
- Virtualization
- Monitoring

Information Security - Others

Privacy: how personal information is handled

- Acquired
- Processed
- Stored
- Shared
- Deleted

Measures:

- Access control
- Transparent processes
- Ciphers
- Integrity and authenticity controls
- Logs

Security objectives (1/3)

Defense against catastrophic events

- Natural phenomena
- Abnormal temperature, lightning, thunder, flooding, radiation, …

Degradation of computer hardware

- Failure of power supplies
- Bad sectors in disks
- Bit errors in RAM cells or SSD, etc.

Security objectives (2/3)

Defense against ordinary faults / failures

- Power outages
- Systems' internal failures
  - Linux Kernel panic, Windows blue screen, OS X panic
  - Deadlocks
  - Abnormal resource usage
- Software faults / Communication faults...

Security objectives (3/3)

Defense against non-authorized activities (adversaries)

- Initiated by someone “from outside” or “from inside”

Types of non-authorized activities:

- Information access
- Information alteration
- Resource usage
  - CPU, memory, print, network, etc.
- Denial of Service
- Vandalism
  - Interference with the normal system behavior without any benefit for the attacker

Core Concepts

1. **Domains**
2. **Policies**
3. **Mechanisms**
4. **Controls**

Security Domains

A set of entities sharing similar security attributes

Allow managing security in an aggregated manner

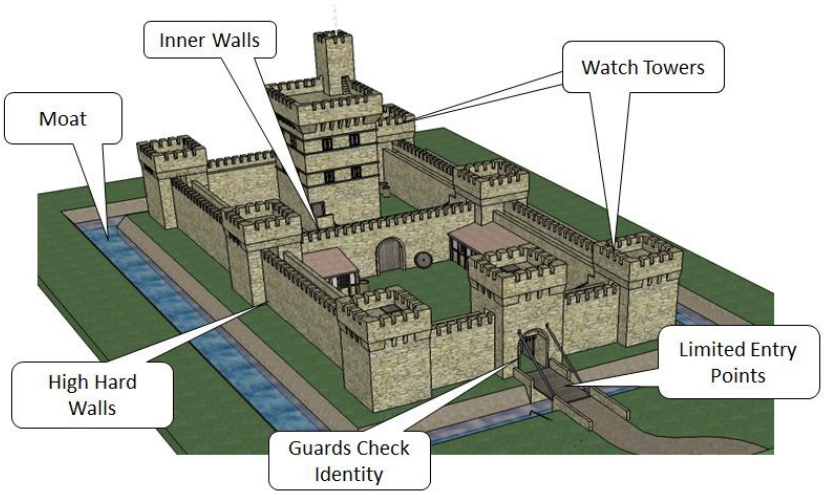
- Management will set the attributes of the domain
- Entities are added do the domain and will get the "group" attributes

Behavior and interactions are homogenous inside the domain

Domains can be organized in a flat of hierarchical manner

Interactions between domains are usually controlled

Security Domains



Security Policies

Set of guidelines related to security, that rule over a domain

Organization will contain multiple policies
- Applicable to each specific domain
- They may overlap and have different scopes/abstraction levels

The multiple policies must be coherent

Examples
- Users can only access web services
- Subjects must be authenticated in order to enter the domain
- Walls must be made of concrete
- Communications must be encrypted

Security Policies

Define the power of each subject

- Least privilege principle: each subject should only have the privileges required for the fulfillment of his duties

Define security procedures

- Who does what in which circumstances

Define the minimum security requirements of a domain

- Security levels, Security Groups
- Required authorization
  - And the related minimum authentication requirements (Strong/weak, single/multifactor, remote/face-to-face)

Security Policies

Define defense strategies and fight back tactics

- Defensive architecture
- Monitoring of critical activities or attack signs
- Reaction against attacks or other abnormal scenarios

Define what are legal and illegal activities

- Forbid list model: Some activities are denied, the rest are allowed
- Permit list model: Some activities are allowed, the rest is forbidden

Security mechanisms

Mechanisms implement policies

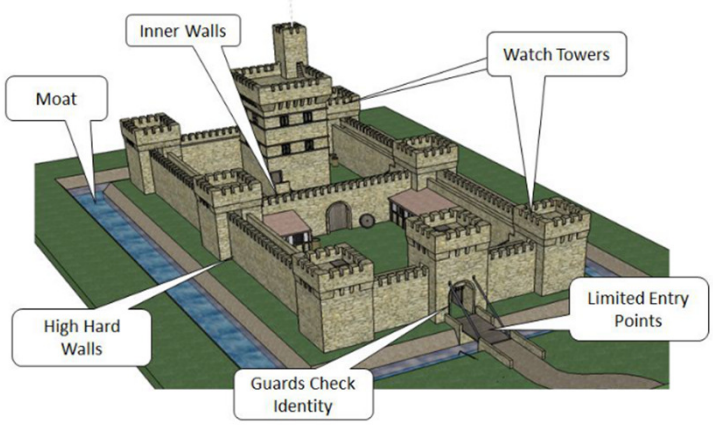
- Policies define, at a higher level, what needs to be done or exist
- Mechanisms are used to deploy policies

Generic security mechanisms

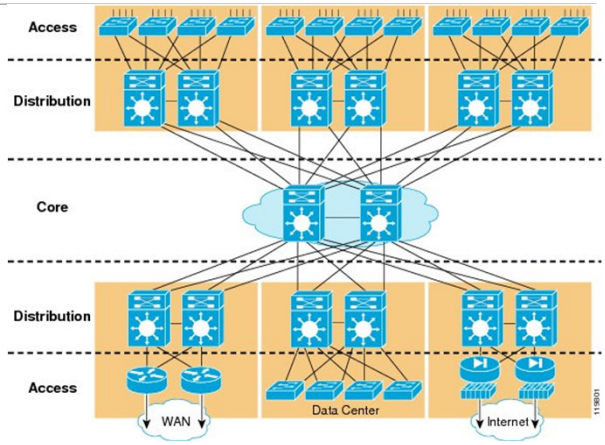
- Confinement (sandboxing)
- Authentication
- Access control
- Privileged Execution
- Filtering
- Logging
- Auditing
- Cryptographic algorithms
- Cryptographic protocols

Security mechanisms

Policy: Movement between domains is restricted
Mechanisms: Doors, guards, passwords, objects/documents



Policy: systems must be resilient
Mechanisms: equipments and links are doubled, arquitecture



Source: CISCO

Security Controls

Controls are any aspect allowing to minimize risk (protect the CIA properties)

Controls include policies & mechanisms, but also:

- Standards and Laws
- Processes
- Techniques

Controls are explicitly stated and can be auditable

- E.g.: ISO 27001 defines 114 controls in 14 groups
  - … asset management, physical security, incidente management…

Types of Security Controls

	Prevention	Detection	Correction
Physical	- Fences - Gates - Locks	- CCTV	- Repair Locks - Repair Windows - Redeploy access cards
Technical	- Firewall - Authentication - Antivirus	- Intrusion Detection Systems - Alarms - Honeypots	- Vulnerability patching - Reboot Systems - Redeploy VMs - Remove Virus
Administrative	- Contractual clauses - Separation of Duties - Information Classification	- Review Access Matrixes - Audits	- Implement a business continuity plan - Implement an incident response plan

Types of Security Controls

	Prevention	Detection	Correction
Physical	- Fences - Gates - Locks	- CCTV	- Repair Locks - Repair Windows
Technical	- Firewall - Authentication - Antivirus		
Administrative	- Contractual clau - S - Infor tion Cla		

Green: in relation to an event

Red: in relation to its nature

Practical Security

Realistic Prevention

Consider that perfect security is impossible!

Focus on the most probable events
- May depend on physical location, legal framework, …

Consider cost and profit
- A great number of controls has a low cost
- However, there is no upper limit on the cost of a security strategy

Consider all domains and entities
- A single breach can be escalated to a more serious situation

Practical Security

Realistic Prevention

Consider Impact
- Under the light of CIA and other potential impact areas (e.g., brand)

Consider the cost and recover time
- Monetary cost, reputation, market access

Characterize attackers
- Define controls specific for those attackers
- There will always exist more resourceful attackers

Consider that the system will be compromised
- Have recovery plans

Security in computing systems: Complex problems

Computers can do much damage in short time frames

- Computers manage huge amounts of information
- Process and communicate with very high speed

The number of weaknesses is always growing

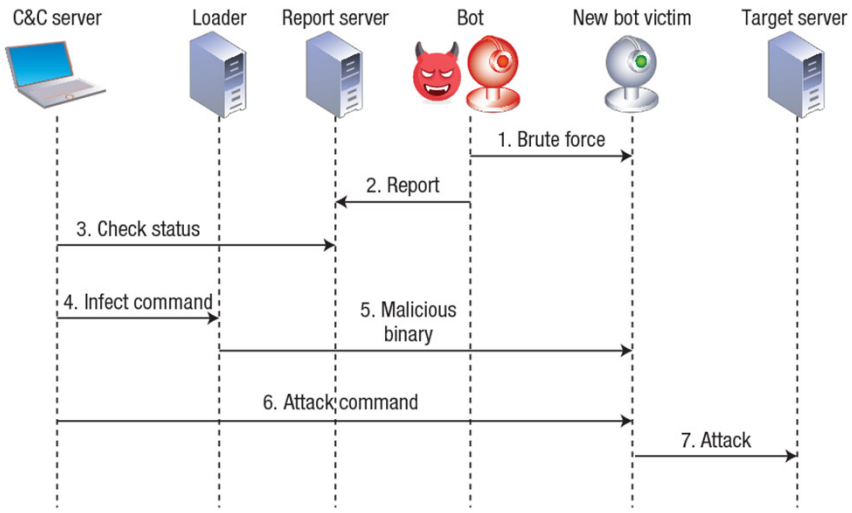
- Due to the increased complexity
- Due to every reducing time-to-market, or cost

Security in computing systems: Complex problems

Networks allow novel attack mechanisms

- "Anonymous" attacks from any place in the planet
- Fast spread across geographical boundaries
- Exploitation of insecure hosts and applications

- **Attackers can build complex attack chains**
  - First exploration
  - Lateral movement
  - Exfiltration
  - Check: https://attack.mitre.org/matrices/enterprise/



Mirai botnet operation and communication.
Mirai causes a distributed denial of service (DDoS) to a set of target servers by constantly propagating to weakly configured Internet of Things (IoT) devices.

source: Kolias, Constantinos et al. "DDoS in the IoT: Mirai and Other Botnets." Computer 50 (2017): 80-84.

Security in computing systems: Complex problems

Users are mostly unaware of the risks

- They do not know the problems,
- ... the impact
- ... the good practices
- ... nor the solutions

Users are mostly careless

- Because they take risks
- Do not care (do not have/identify any responsibility)
- Do not estimate the risk correctly

Main vulnerability sources

Hostile applications or bugs in applications

- Rootkits: Insert elements in the operating system
- Worms: Software programs controlled by an attacker
- Virus: Pieces of code that infect other files (e.g., macros)

Users

- Ignorant, careless or reckless
  - Use insecure alternatives instead of secure ones
  - Trust on security tools to solve all problems
  - Search and download illegal stuff
- Hostile

Main vulnerability sources

Defective administration

- Default configuration is seldom the most secure
- Security restriction vs flexible operation
- Exceptions to individuals

Communication over uncontrolled/unknown network links

- Public hotspots, campus networks, hostile governments

Perimeter Defense

(minimal defense, frequently not sufficient)

Perimeter Defense

Protection against external attackers
- Internet
- Foreign users
- Other organizations

Assumes that internal users are trusted and share the same policies
- Friends, family, collaborators

Used in domestic scenarios or small offices

Limitations
- Too simple
- Doesn't protect against internal attackers
  - Previously trusted users
  - Attackers that acquired internal access

Defense in Depth

Protection against internal and external attackers
- From the Internet
- Users
- Other organizations

Assumes well-defined domains across the organization
- Walls, doors, authentication, security personell, ciphers, secure networks

Limitations
- Needs coordination between the diferent controls
  - May end with overlapping controls, but also with holes in the security perimeters
- Cost
- Requires training, changes to processes and frequent audits

Zero Trust

Defense model without specific perimeters
- There is no inherent trust in entities just because they are internal
  - Actually, there may be no notion of internal and external

Model recommended for new systems
- Traditional systems should migrate to it
- Implies the design of systems/services specific for this model
- Legacy systems will need additional protection layers
  - Firewalls, filters, adapters, plugins

Zero Trust – Principles (NCSC)

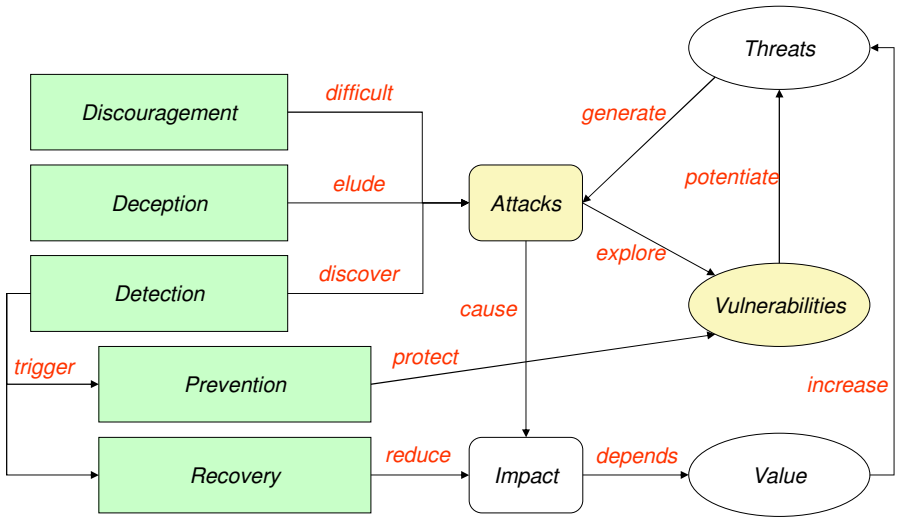
1. **Know your architecture**
   - Users, devices, services and data
2. **Know your identities**
   - Users, devices, services and data
3. **Assess user behaviour, service and device health**
4. **Use policies to authorize requests**

Zero Trust – Principles (NCSC)

5. **Authenticate and Authorize everywhere**
   - No open APIs, or IP address-based access
6. **Focus your monitoring on users, devices and services**
7. **Don't trust any network, including your own**
   - Internal attackers should not have more rights than external attackers
8. **Choose services designed for Zero Trust**
   - Legacy services to be avoided, but can be integrated

Vulnerabilities

Information Security Vulnerabilities and Attacks



Measures (and some tools)

Discouragement
- Punishment
  - Legal restrictions
  - Forensic evidences
- Security barriers
  - Firewalls
  - Autentication
  - Secure communication
  - Sandboxing

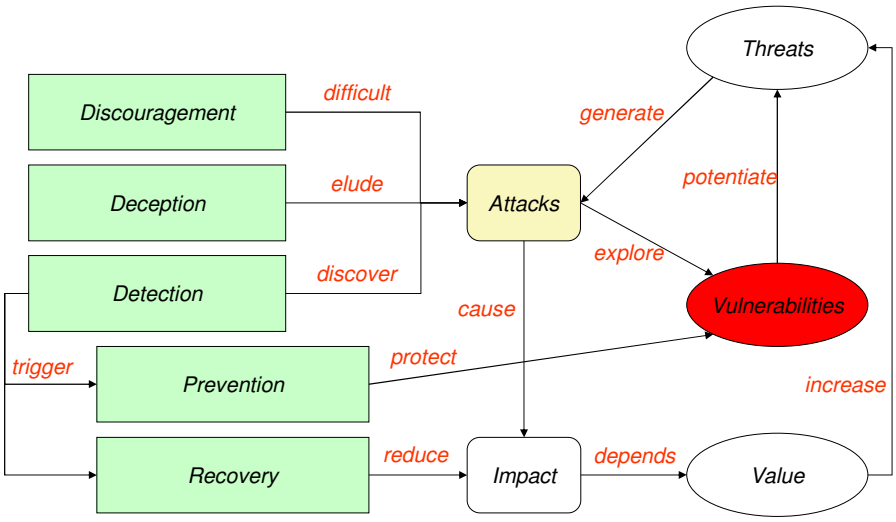
Detection
- Intrusion detection system
  - e.g. Seek, Bro, Suricata
- Auditing
- Forensic break-in analysis

Deception
- Honeypots / honeynets
- Forensic follow-up

Prevention
- Restrictive policies
  - e.g. least privilege principle
- Vulnerability scanning
  - e.g. OpenVAS, metasploit
- Vulnerability patching
  - e.g. regular updates

Recovery
- Backups
- Redundant systems
- Forensic recovery

Information Security Vulnerabilities and Attacks



Vulnerability

A mistake in software that can be directly used by an attacker to gain access to a system or network

A mistake is a vulnerability if it allows an attacker to use it to violate a reasonable security policy for that system

- This excludes entirely "open" security policies in which all users are trusted, or where there is no consideration of risk to the system

A CVE vulnerability is a state in a computing system (or set of systems) that either:

- Allows an attacker to execute commands as another user
- Allows an attacker to access data that is contrary to the specified access restrictions for that data
- Allows an attacker to pose as another entity
- Allows an attacker to conduct a denial of service

Exposure

A configuration issue or a mistake in software allowing access to information or capabilities used as a stepping-stone into a system or network

Is the state of being susceptible or vulnerable to potential security risks or threats.

A configuration issue or a mistake is an exposure if it does not directly allow compromise

- But could be an important component of a successful attack, and is a violation of a reasonable security policy

An exposure describes a state in a computing system (or set of systems) that is not a vulnerability, but either:

- Allows an attacker to conduct information gathering activities
- Allows an attacker to hide activities
- Includes a capability that behaves as expected, but can be easily compromised
- Is a primary point of entry that an attacker may attempt to use to gain access to the system or data
- Is considered a problem by some reasonable security policy

CVE
Common Vulnerabilities and Exposures

Dictionary of publicly known information security vulnerabilities and exposures

- For vulnerability management
- For patch management
- For vulnerability alerting
- For intrusion detection

Uses common identifiers for the same CVEs

- Enable data exchange between security products
- Provide a baseline index point for evaluating coverage of tools and services.

Details about a vulnerability can be kept private

- Part of responsible disclosure: until owner provides a fix

CVE-ID

CVE-2015-1538 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Integer overflow in the SampleTable::setSampleToChunkParams function in SampleTable.cpp in libstagefright in Android before 5.1.1 LMY48I allows remote attackers to execute arbitrary code via crafted atoms in MP4 data that trigger an unchecked multiplication, aka internal bug 20139950, a related issue to CVE-2015-4496.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- BID:76052
- URL:http://www.securityfocus.com/bid/76052
- CONFIRM:http://www.huawei.com/en/psirt/security-advisories/hw-448928
- CONFIRM:http://www1.huawei.com/en/security/psirt/security-bulletins/security-advisories/hw-448928.htm
- CONFIRM:https://android.googlesource.com/platform/frameworks/av/+/2434839bbd168469f80dd9a22f1328bc81046398
- EXPLOIT-DB:38124
- URL:https://www.exploit-db.com/exploits/38124/
- MISC:http://packetstormsecurity.com/files/134131/Libstagefright-Integer-Overflow-Check-Bypass.html
- MLIST:[android-security-updates] 20150812 Nexus Security Bulletin (August 2015)
- URL:https://groups.google.com/forum/message/raw?msg=android-security-updates/Ugvu3fi6RQM/yzJvoTVrIQAJ
- SECTRACK:1033094
- URL:http://www.securitytracker.com/id/1033094

Assigning CNA

MITRE Corporation

Date Entry Created

20150206 Disclaimer: The entry creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Phase (Legacy)

Assigned (20150206)

Votes (Legacy)

Comments (Legacy)

Proposed (Legacy)

N/A

This is an entry on the CVE List, which provides common identifiers for publicly known cybersecurity vulnerabilities.

SEARCH CVE USING KEYWORDS: Submit
You can also search by reference using the CVE Reference Maps.

For More Information: CVE Request Web Form (select "Other" from dropdown)

CVE identifiers

Aka CVE names, CVE numbers, CVE-IDs, CVEs

Unique, common identifiers for publicly known information security vulnerabilities
- Have "candidate" or "entry" status
- Candidate: under review for inclusion in the list
- Entry: accepted to the CVE List

Format
- CVE identifier number (CVE-Year-Order)
- Status (Candidate or Entry)
- Brief description of the vulnerability or exposure
- References to extra information

CVE benefits

Provides common language for referring to problems
- Facilitates data sharing among
- Intrusion detection systems
- Assessment tools
- Vulnerability databases
- Researchers
- Incident response teams

Will lead to improved security tools
- More comprehensive, better comparisons, interoperable
- Indications and warning systems

Will spark further innovations
- Focal point for discussing critical database content issues

CVE and Attacks

Attacks can be made possible through multiple vulnerabilities
- One CVE for each vulnerability

Example: Stagefright (Android, video in MMS messages)
- CVE-2015-1538, P0006, Google Stagefright 'stsc' MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1538, P0004, Google Stagefright 'ctts' MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1538, P0004, Google Stagefright 'stts' MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1538, P0004, Google Stagefright 'stss' MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1539, P0007, Google Stagefright 'esds' MP4 Atom Integer Underflow Remote Code Execution
- CVE-2015-3827, P0008, Google Stagefright 'covr' MP4 Atom Integer Underflow Remote Code Execution
- CVE-2015-3826, P0009, Google Stagefright 3GPP Metadata Buffer Overread
- CVE-2015-3828, P0010, Google Stagefright 3GPP Integer Underflow Remote Code Execution
- CVE-2015-3824, P0011, Google Stagefright 'tx3g' MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-3829, P0012, Google Stagefright 'covr' MP4 Atom Integer Overflow Remote Code Execution

Vulnerability detection

Specific tools can detect vulnerabilities
- Exploiting known vulnerabilities
- Testing known vulnerability patterns
  - e.g., buffer overflow, SQL injection, XSS, etc.

Specific tools can replicate known attacks
- Use known exploits for known vulnerabilities
  - e.g.: MS Samba v1 exploit used by WannaCry
- Can be used to implement countermeasures

Vital to assert the robustness of production systems and applications
- Service often provided by third-party companies

Vulnerability detection

Can be applied to:

- Source code (static analysis)
  - OWASP LAPSE+, RIPS, Veracode, …
- Running application (dynamic analysis)
  - Valgrind, Rational, AppScan, GCC, …
- Externally as a remote client:
  - OpenVAS, Metasploit, …

Should not be blindly applied to production systems!

- Potential data loss/corruption
- Potential DoS
- Potential illegal activity

CWE
Common Weakness Enumeration

Is a catalog of common software and hardware weaknesses, vulnerabilities, and issues that can lead to security problems in computer systems and applications

Common language of discourse for discussing, finding and dealing with the causes of software security vulnerabilities

- Found in code, design, or system architecture
- Each individual CWE represents a single vulnerability type
- Currently maintained by the MITRE Corporation
  - A detailed CWE list is currently available at the MITRE website
- The list provides a detailed definition for each individual CWE

Individual CWEs are held within a hierarchical structure

- CWEs at higher levels provide a broad overview of a vulnerability type
  - Can have many children CWEs associated with them
- CWEs at deeper levels provide a finer granularity
  - Usually have fewer or no children CWEs

CWE ≠ CVE

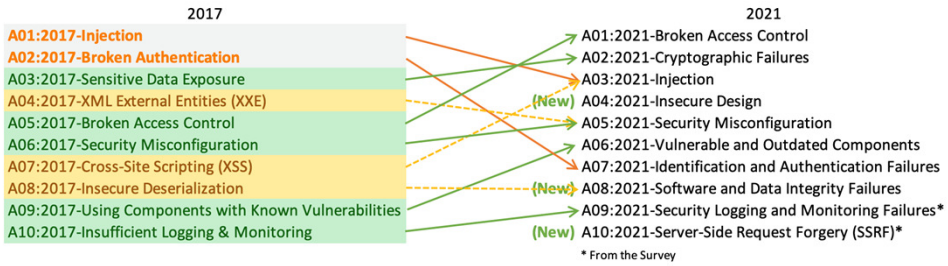
CVE is specific and assigns unique identifiers to individual vulnerabilities, making it easier to reference and track known security issues.

CWE is more general and provides a taxonomy of common weakness categories, helping to improve awareness and understanding of various security issues.

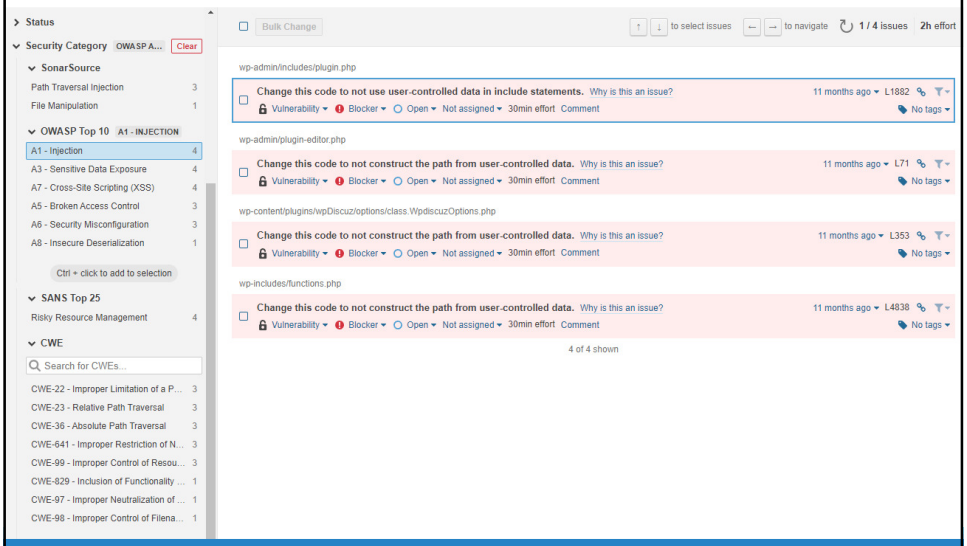
Vulnerability types
OWASP Top 10 (Web, 2021)

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

Vulnerability types
OWASP Top 10 (Web)



Static Analysis (with Sonarcloud)



Vulnerability Tracking by vendors

During the development cycle, vulnerabilities are handled as bugs

- May have a dedicated security team or not

When software is available, vulnerabilities are also tracked globally

- For every system and software publicly available

Public tracking helps…

- focusing the discussion around the same issue
  - Ex: a library that is used in multiple applications, distributions
- defenders to easily test their systems, enhancing the security
- attackers to easily know what vulnerability can be used

Vulnerability Tracking

Vulnerabilities are privately tracked

- Constitute an arsenal for future attacks against targets
- Exploits are weapons

Knowledge about vulnerabilities and exploits is publicly traded

- From 0 to 2-3M€ (more?) through direct markets, or acquisition programs
- Up to 2.5M€ for bug hunting programs or direct acquisition (Google, Zerodium)
  - 2.5M€: 1 click Android exploit
  - 2M€: 1 click iPhone exploit
  - 1.5M€: WhatsApp or iMessage exploit
  - ~2K for a XSS at HackerOne (although there are records of $1M payouts)

…and privately traded at unknown prices

- Private Companies, Organized Crime, APTs

CVE-2020-1472

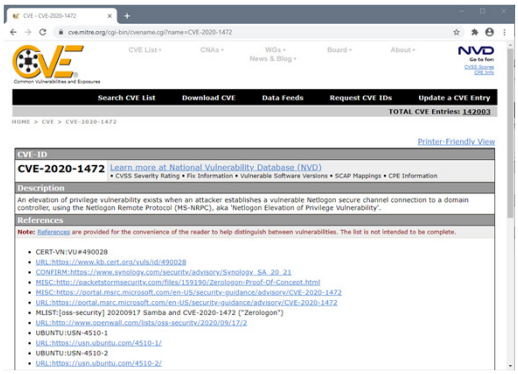
@MITRE

Basic information about the CVE

References to other trackers (provided for convenience)

Vendor pages

Mailing lists



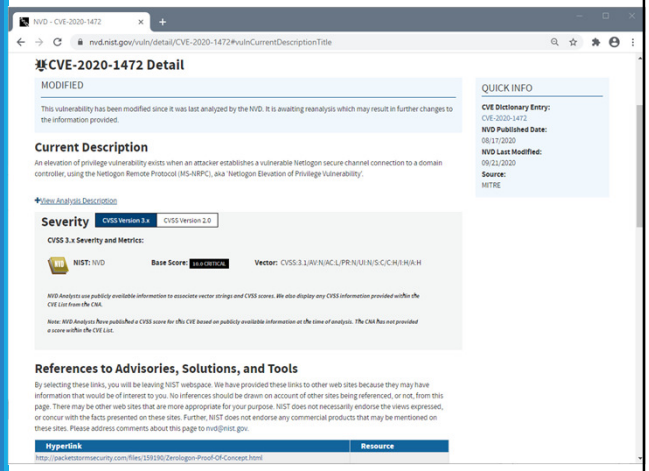
CVE-2020-1472

@NVD

Basic information about the CVE and a small analysis of it

The CVE Severity Score

Links to advisories, solutions



CVE-2020-1472

@Product Owner

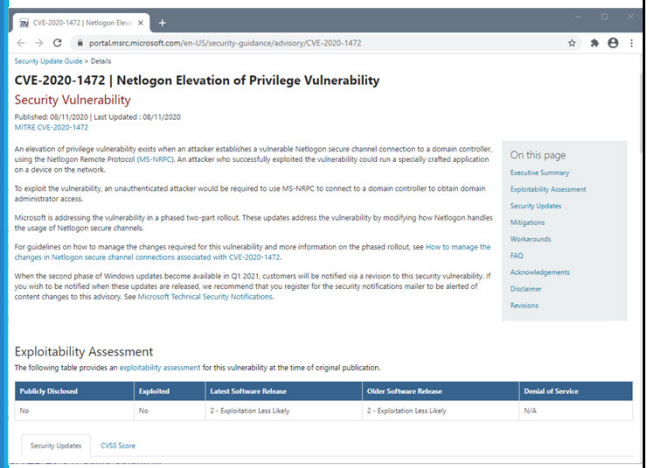
More detail, why it happens, and how it can be mitigated

Information about patches/updates available to help IT staff and users

Information about it's exploitability

Format is vendor dependent

Each vendor defines what/how to show information



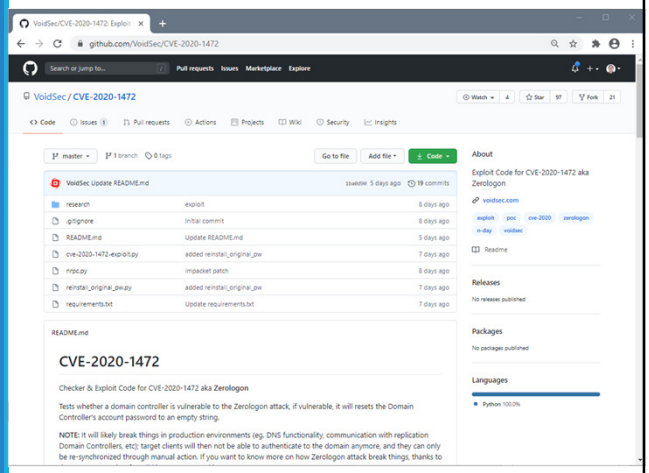
CVE-2020-1472

@Other places

Independent researchers may publish Proofs of concept (PoC)

Very dynamic community with public and private facets

PoC may help both defenders and attackers
- **Defenders can test**
- **Attackers have code to use**

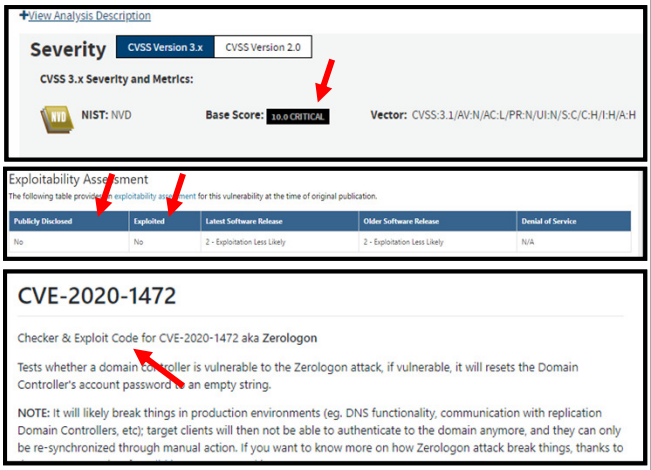


Vulnerability tracking

Not an easy task
- Exploits are not always known
- Impact and Value may be underestimated

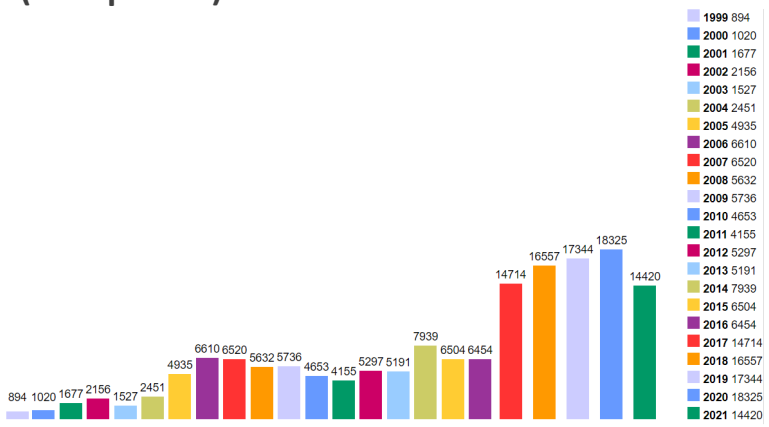
Old feeds may create a false sense of security

A highly dynamic community is great…
- To defenders as they can test and implement defenses
- To attackers as they can incorporate exploits



CVE per year – cvedetails.com

(at Sep 2021)



Zero Day (or Zero Hour) Attack/Threat

Attack using vulnerabilities which are:

- Unknown to others
- Undisclosed to the software vendor

Occurs at the day zero of the knowledge about those vulnerabilities

- For which no security fix is available

A single "day zero" may exist for months/years

- Known to attackers, unknown to others
- Frequently part of attack arsenal
- Traded around in specific markets

Survivability

How can we survive a zero-day attack?

How can we react to a massive zero-day attack?

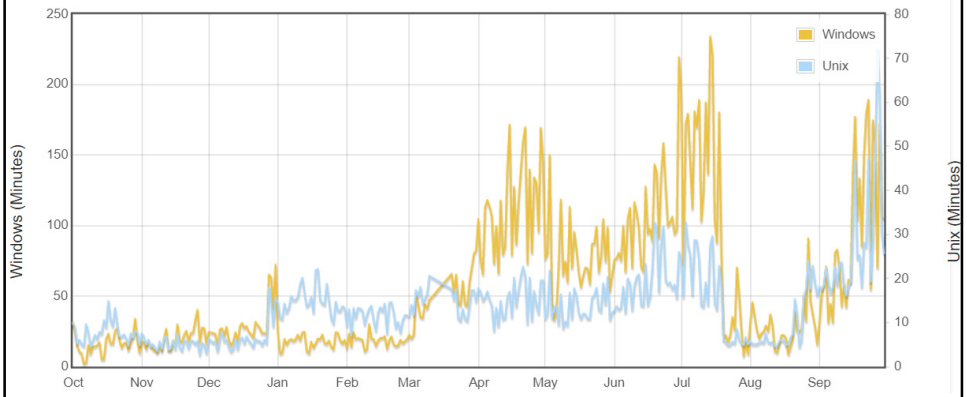
Diversity is one answer (as a policy) …

- but software production, distribution and update goes on the opposite direction!
  - And the same happens with hardware architectures
- Why is MS Windows such an interesting target?
  - And Apple macOS not so much?
- Are you using an Android cell phone?
  - What are the odds of being in the battlefront? (you are)
  - iOS landscape may be worst as it is more homogeneous

Mean Survival Time

Oct 2020 – Oct 2021

(http://isc.sans.org/survivaltime.html)



Defender will constantly spend resources in security

Attacker only needs to be successful once

- Attackers can screen for victims with low effort and in an automated manner

CERT

Computer Emergency Readiness Team

Organization ensuring that appropriate technology and systems' management practices are used to

- Resist attacks on networked systems
- Limit damage, ensure continuity of critical services
  - In spite of successful attacks, accidents, or failures

CERT/CC (Coordination Center) @ CMU

- One component of the larger CERT Program
- A major center for internet security problems
  - Established in November 1988, after the "Morris Worm"
  - It demonstrated the growing Internet exposure to attacks

CSIRT

Computer Security Incident Response Team

A service organization responsible for receiving, reviewing, and responding to computer security incident reports and activity

- Provides 24x7 Computer Security Incident Response Services to users, companies, government agencies or organizations
- Provides a reliable and trusted single point of contact for reporting computer security incidents worldwide
- CSIRT provides the means for reporting incidents and for disseminating important incident-related information

Portuguese CSIRTs

- CERT.PT: https://www.facebook.com/CentroNacionalCiberseguran caPT
- National CSIRT Network : https://www.redecsirt.pt/
- CSIRT @ UA: https://csirt.ua.pt

Security alerts & activity trends

Vital to the fast dissemination of knowledge about new vulnerabilities

- US-CERT Technical Cyber Security Alerts
- US-CERT (non-technical) Cyber Security Alerts
- SANS Internet Storm Center
  - Aka DShield (Defense Shield)
- Microsoft Security Response Center
- Cisco Security Center

- And many others …

Other sources of information

Reddit r/netsec

Twitter #infosec #cybersec

Discord, Slack and other private and public sources

- https://en.0day.today
- https://www.exploit-db.com/
- https://vuldb.com/

Access control models

Access types

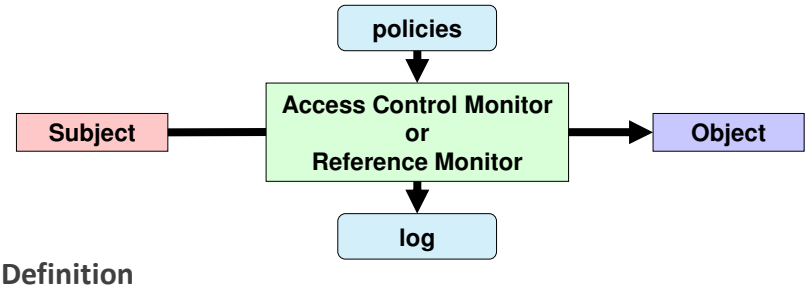
Physical access

- Physical contact between a subject and the object of interest
  - Facility, room, network, computer, storage device, authentication token, etc.
- Out of scope of this course …

Informatic or electronic access

- Information-oriented contact between a subject and the object of interest
  - Contact through request-response dialogs
- Contact is mediated by
  - Computers and networks
  - Operating systems, applications, middleware, devices, etc.

Access control



Definition

- Policies and mechanisms that mediate the access of a subject to an object

Normal requirements

- Authentication
  - With some Level of Assurance (LoA)
- Authorization policies
- Accountability ➔ logging

AAA

Access control

Subjects and objects: Both digital entities

Subjects are something exhibiting activity:

- Processes
- Computers
- Networks

Objects are the target of an action:

- Stored data
- CPU time
- Memory
- Processes
- Computers
- Networks

An entity can be both subject and object

Least privilege principle

Every program and every user of the system should operate using the least set of privileges necessary to complete the job

J. H. Saltzer, M. D. Schroeder,
The protection of information in computer systems, Proc. of the IEEE, 63(9) 1975

Privilege:

- Authorization to perform a given task
- Similar to access control clearance

Each subject should have, at any given time, the exact privileges required to the assigned tasks

- Less privileges than the required create unsurpassable barriers
- More privileges than the required create vulnerabilities
  - Damage resulting from accidents or errors
  - Potential interactions among privileged programs
  - Misuse of a privileges
  - Unwanted information flows
    - "need-to-know" military restrictions

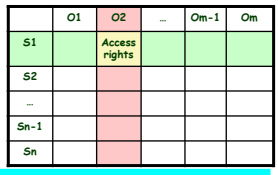
Access control models

	O1	O2	…	Om-1	Om
S1		Access rights			
S2					
…					
Sn-1					
Sn					

Access control matrix

- Matrix with all access rights for subjects relatively to objects
- Conceptual organization

Access control models



ACL-based mechanisms

- ACL: Access Control List
- Matrix column

Is a list of permissions or rules that specifies which users or system entities are granted access to specific resources or objects and what actions they can perform on those resources.

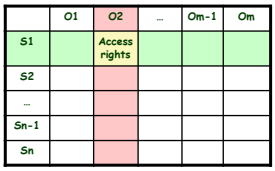
List of access rights for specific subjects

- Access rights can be positive or negative
- Default subjects may often be used

Usually, ACLs are stored along with objects

- e.g., for file system objects

Access control models



Capability-based mechanisms

- Capability: unforgeable authorization token
- Matrix row
- Contains object references and access rights

Access granting

- Transmission of capabilities between subjects
- Mediated / non-mediated

Usually, capabilities are kept by subjects

- e.g., OAuth 2.0 access tokens

Access control kinds: MAC and DAC

O controlo de acesso é rigidamente definido por políticas de segurança que são impostas pelo sistema operativo ou pela organização.

Mandatory access control (MAC)

- Fixed access control policy implemented by the access control monitor
- Access control rights cannot be tailored by subjects or object owners

O controlo de acesso é baseado na discricionariedade do proprietário do recurso

Discretionary access control (DAC)

- Some subjects can update rights granted or denied to other subjects for a given object
- Usually this is granted to object owners and system administrators

Access control kinds: Role-Based Access Control (RBAC)

D.F. Ferraiolo and D.R. Kuhn, "**Role Based Access Control**",
15th National Computer Security Conference, Baltimore, October 1992

É um modelo de controlo de acesso que se baseia na atribuição de papéis aos utilizadores e, em seguida, controla o acesso a recursos com base nesses papéis.

Not DAC or MAC

- Roles are dynamically assigned to subjects
- For access control it matters the role played by the subject and not the subject's identity
  - Identity is mostly relevant for role access and logging

Access control binds roles to (meaningful) operations

- Operations are complex, meaningful system transactions
  - Not the ordinary, low-level read/write/execute actions on individual objects
- Operations can involve many individual lower-level objects

Access control kinds: RBAC rules (1/2)

Role assignment:

- All subject activity on the system is conducted through transactions
  - And transactions are allowed to specific roles
  - Thus, all active subjects are required to have some active role
- A subject can execute a transaction **iff** it has selected or been assigned a role which can use the transaction

Access control kinds: RBAC rules (2/2)

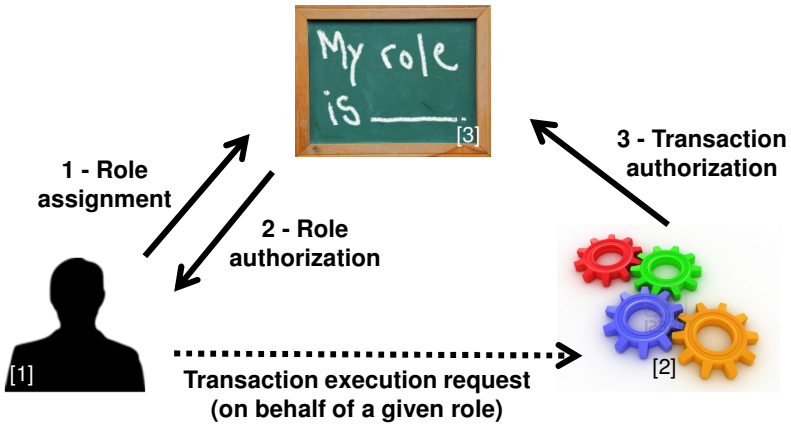
Role authorization:

- A subject's active role must be authorized for the subject

Transaction authorization:

- A subject can execute a transaction **iff**
  - the transaction is authorized through the subject's role memberships
- and
  - there are no other constraints that may be applied across subjects, roles, and permissions

RBAC rules



[1] From http://www.clker.com/clipart-24011.html
[2] From http://www.123rf.com/photo_12115593_three-dimensional-colored-toothed-wheels.html
[3] From http://www1.yorksolutions.net/Portals/115255/images/MyRoleIs.jpg

RBAC: Roles vs groups

Roles are a collection of permissions

- The permissions are granted to the subjects that, at a given instant, play the role
- A subject can (should) only play a role at a given time

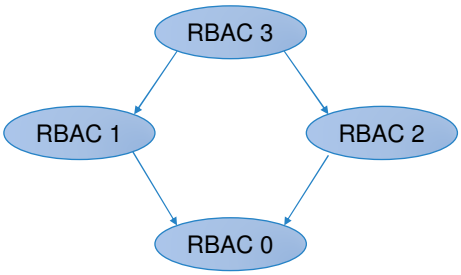
Groups are a collection of users

- And permissions can be granted both to users and groups
- A subject can belong to many groups at a given time

The session concept

- Role assignment is like a session activation
- Group membership is ordinarily a static attribute

RBAC variants



RBAC 0
- No role hierarchies
- No role constraints

RBAC 1
- RBAC 0 w/ role hierarchies (privilege inheritance)

RBAC 2
- RBAC 0 w/ role constraints (separation of duties)

RBAC 3
- RBAC 1 + RBAC 2

NIST RBAC model

Flat RBAC
- Simple RBAC model w/ **user-role review**

User-role review
Which users can have a role?
Role $\xrightarrow{?}$ users
Which roles can a user have?
User $\xrightarrow{?}$ roles

Hierarchical RBAC
- Flat RBAC w/ role hierarchies (DAG or tree)
- General and restricted hierarchies

Constraint RBAC
- RBAC w/ role constraints for separation of duty

Symmetric RBAC
- RBAC w/ **permission-role review**

Permission-role review
Which permissions has a role?
Role $\xrightarrow{?}$ permissions
Which roles have a permission?
Permission $\xrightarrow{?}$ roles

Access control kinds:

Context-Based Access Control (CBAC)

Access rights have an historical context

- The access rights cannot be determined without reasoning about past access operations
- Example:
  - Stateful packet filter firewall

Chinese Wall policy

D.F.C. Brewer and M.J. Nash,
"**The Chinese Wall Security Policy** ",
IEEE Symposium on Security and Privacy, 1989

- Conflict groups
- Access control policies need to address past accesses to objects in different members of conflict groups

Access control kinds:

Attribute-Based Access Control (ABAC)

Access control decisions are made based on attributes associated with relevant entities

OASIS XACML architecture

- Policy Administration Point (PAP)
  - Where policies are managed
- Policy Decision Point (PDP)
  - Where authorization decisions are evaluated and issued
- Policy Enforcement Point (PEP)
  - Where resource access requests are intercepted and confronted with PDP's decisions
- Policy Information Point (PIP)
  - Where the PDP gets external information

É um modelo de controlo de acesso que se baseia na avaliação de atributos e características dos utilizadores dos recursos e do ambiente para determinar se o acesso a um recurso é permitido.

XACML:
Access control with PEP and PDP

A subject sends a request

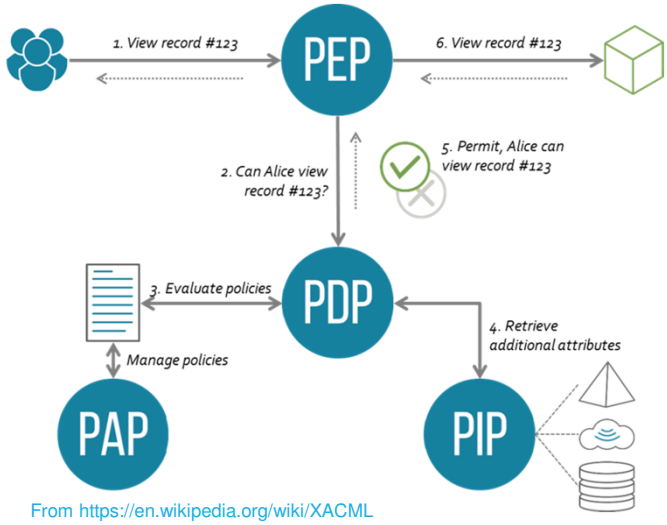
- Which is intercepted by the Policy Enforcement Point (PEP)
- The PEP sends an authorization request to the Policy Decision Point (PDP)

The PDP evaluates the authorization request against its policies and reaches a decision

- Which is returned to the PEP

- Policies are retrieved from a Policy Retrieval Point (PRP)
- Useful attributes are fetched from Policy Information Points (PIP)
- Policies are managed by the Policy Administration Point (PAP)

XACML big picture



From https://en.wikipedia.org/wiki/XACML

Break-the-glass access control model

In some scenarios it may be required to overcome the established access limitations

- e.g., in a life-threatening situation

In those cases, the subject may be presented with a break-the-glass decision upon a deny

- Can overcome the deny at their own responsibility
- Logging is fundamental to prevent abuses

Separation of duties

R.A. Botha, J.H.P. Eloff, "Separation of duties for access control enforcement in workflow environments", IBM Systems Journal, 2001

Fundamental security requirement for fraud and error prevention

- Dissemination of tasks and associated privileges for a specific business process among multiple subjects
- Often implemented with RBAC

Damage control

- Segregation of duties helps reducing the potential damage from the actions of one person
- Some duties should not be combined into one position

Segregation of duties:
ISACA (Inf. Systems Audit and Control Ass.) matrix guideline

Exhibit 2.9—Segregation of Duties Control Matrix

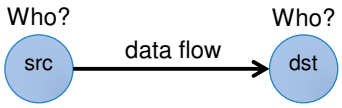
	Control Group	Systems Analyst	Application Programmer	Help Desk and Support Manager	End User	Data Entry	Computer Operator	Database Administrator	Network Administrator	Systems Administrator	Security Administrator	Systems Programmer	Quality Assurance
Control Group		X	X	X		X	X	X	X	X		X	
Systems Analyst	X			X	X		X				X		X
Application Programmer	X			X	X	X	X	X	X	X	X	X	X
Help Desk and Support Manager	X	X	X		X	X		X	X	X		X	
End User		X	X	X			X	X	X			X	X
Data Entry	X		X	X			X	X	X	X	X	X	
Computer Operator	X	X	X		X	X		X	X	X	X	X	
Database Administrator	X		X	X	X	X	X		X	X		X	
Network Administrator	X		X	X	X	X	X	X					
System Administrator	X		X	X		X	X	X				X	
Security Administrator		X	X			X	X					X	
Systems Programmer	X		X	X	X	X	X	X		X	X		X
Quality Assurance		X	X		X							X	

X—Combination of these functions may create a potential control weakness.

Information flow models

Authorization is applied to data flows

- Considering the data flow source and destination
- Goal: avoid unwanted/dangerous information flows



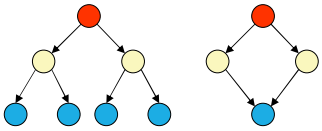
Src and Dst security-level attributes

- Information flows should occur only between entities with given security level (SL) attributes
- Authorization is given based on the SL attributes

Multilevel security

Subjects (or roles) act on different security levels

- Levels do not intersect themselves
- Levels have some partial order
  - Hierarchy
  - Lattice



Levels are used as attributes of subjects and objects

- Subjects: security level clearance
- Objects: security classification

Information flows & security levels

- Same security level → authorized
  - Still, subject to a "need to know"
- Different security levels → controlled

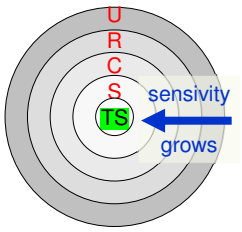
Multilevel security levels: Military / Intelligence organizations

Typical levels

- Top secret
- Secret
- Confidential
- Restricted
- Unclassified

Portugal (NTE01, NTE04)

- Muito Secreto
- Secreto
- Confidencial
- Reservado



EU example

- EU TOP SECRET
- EU SECRET
- EU CONFIDENTIAL
- EU RESTRICTED
- EU COUNCIL / COMMISSION

NATO example

- COSMIC TOP SECRET (CTS)
- NATO SECRET (NS)
- NATO CONFIDENTIAL (NC)
- NATO RESTRICTED (NR)

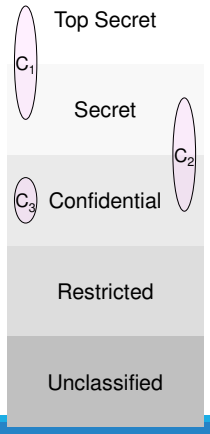
Security categories (or compartments)

Self-contained information environments
- May span several security levels

Military environments
- Military branches, military units

Civil environments
- Departments, organizational units

An object can belong to different compartments and have a different security classification in each of them
- (top-secret, crypto), (secret, weapon)

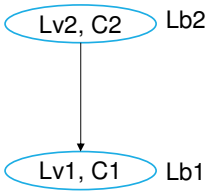


Security labels

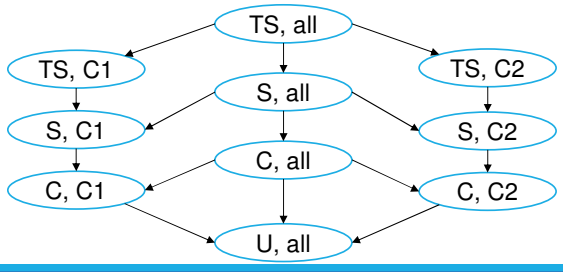
Label = Category + Level

Relative order between labels

$Lb1 \leq Lb2 \Rightarrow C1 \subseteq C2 \wedge Lv1 \leq Lv2$



Labels form a lattice



Bell-La Padula MLS Model

D. Elliott Bell, Leonard J. La Padula, "**Secure Computer Systems: Mathematical Foundations**", MITRE Tech. Report 2547, Volume I, 1973

Access control policy for controlling information flows
- Addresses data confidentiality and access to classified information
- Addresses disclosure of classified information
  - Object access control is not enough
  - One needs to restrict the flow of information from a source to authorized destinations

Uses a state-transition model
- In each state there are subjects, objects, an access matrix and the current access information
- State transition rules
- Security levels and clearances
  - Objects have a security labels
  - Subjects have security clearances
  - Both refer to security levels (e.g., CONFIDENTIAL)

Is primarily used to enforce confidentiality and access control in systems where information security is critical, particularly in government and military settings.

Baseado em MAC

Mandatory Access Control

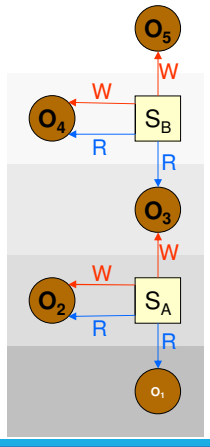
o controlo de acesso é rigidamente definido por políticas de segurança que são impostas pelo sistema operacional ou pela organização

Bell-La Padula MLS Model: Secure state-transition model

Simple security condition (no read up)
- S can read O iff L(S) ≥ L(O)

***-property (no write down)**
- S can write O iff L(S) ≤ L(O)
- aka confinement property

Discretionary Security Property
- DAC-based access control



dac ou mac?

Biba Integrity Model

K. J. Biba, "**Integrity Considerations for Secure Computer Systems**", MITRE Technical Report 3153, The Mitre Corporation, April 1977

Access control policy for controlling information flows

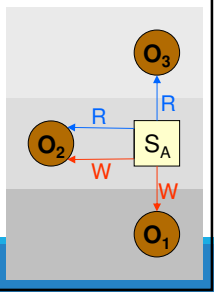
- For enforcing data integrity control
- Uses integrity levels, not security levels
- Similar to Bell-La Padula, with inverse rules

Simple Integrity Property (no read down)

- S can read O iff I(S) ≤ I(O)

Integrity *-Property (no write up)

- S can write O iff I(S) ≥ I(O)



Windows mandatory integrity control

Allows mandatory (priority and critical) access control enforcement prior to evaluate DACLs

- If access is denied, DACLs are not evaluated
- If access is allowed, DACLs are evaluated

Integrity labels

- Untrusted
- Low (or AppContainer)
- Medium
- Medium Plus
- High
- System
- Protected Process

Windows mandatory integrity control

Users

- Medium: standard users
- High: elevated users

Process integrity level

- The minimum associated to the owner and the executable file
- User processes usually are Medium or High
  - Except if executing Low-labeled executables
- Service processes: High

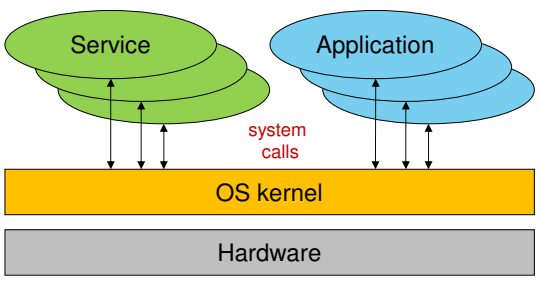
Windows mandatory integrity control

Securable objects mandatory label

- NO_WRITE_UP (default)
- NO_READ_UP
- NO_EXECUTE_UP

Operating systems

Operating Systems



User mode:
Execute in normal CPU mode,
No acess to privileged instructions

Kernel mode:
Execute in privileged CPU mode;
Has access to privileged instructions

Kernel Objectives

Initialize devices (boot time)

Virtualize the hardware

- Computational model

Enforce protection policies and provide protection mechanisms

- Against involuntary mistakes
- Against non-authorized activities

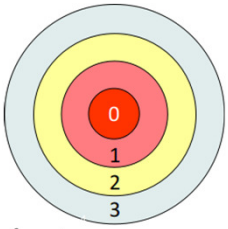
Provide a file system

- Agnostic of the actual storage devices used

Execution Rings

Different levels of privilege

- Forming a set of concentric rings
- Used by CPUs to prevent non-privileged code from running privileged opcodes
  - e.g., IN/OUT, TLB manipulation



Nowadays processors have 4 rings

- But OS's usually use only 2
  - 0 (supervisor/kernel mode)
  - 3 (user-mode)

Transfer of control between rings requires special gates

- The ones that are used by system calls (aka syscalls)
  - Traps
- Interrupt gates

Executing Virtual Machines

Common approach

- Software-based virtualization
- Direct execution of guest user-mode code (ring 3)
- Binary translation of privileged code (ring 0)
  - Guest OS kernels remain unchanged, but do not run directly on the host machine

Hardware-assisted virtualization

- Full virtualization
  - There is a ring -1 below ring 0
    - Hypervisor and kernel extensions such as Intel VT-x and AMD-V
- It can virtualize hardware for many ring 0 kernels
  - No need of binary translation
  - Guest OS's run faster (almost native performance)

Execution of Virtual Machines

Virtual machines implemente an essential security mechanism: confinement

- Implement a security domain constrained for use of a small set of applications
- Also provide a common abstraction with common hardware
  - Even if the host hardware is modified

Provide additional mechanisms

- Control resources
- Prioritize access to resources
- Creation of images for analysis
- Fast recovery to a known state

As vms isolam se confinam o software e as aplicações prevenindo estas de interferirem com o host system

Computational Model

Set of entities (objects) managed by the OS kernel

- Define how applications interact with the kernel

Examples

- User identifiers
- Processes
- Virtual memory
- Files and file systems
- Communication channels
- Physical devices
  - Storage
    - Magnetic disks, optical disks, silicon disks, tapes
  - Network interfaces
    - Wired, wireless
  - Human-computer interfaces
    - Keyboards, graphical screens, text consoles, mice
  - Serial/parallel I/O interfaces
    - USB, Bluetooth
    - Serial ports, parallel ports, infrared

User Identifiers (UID)

For the OS kernel a user is a number

- Established during a login operation
- User ID (UID)

All activities are executed on a computer on behalf of a UID

- UID allows the kernel to assert what is allowed/denied to them
- Linux: UID 0 is omnipotent (root)
  - Administration activities are usually executed with UID 0
- macOS: UID 0 is omnipotent for management
  - Some binaries and activities are restricted, even for root
- Windows: concept of privileges
  - For administration, system configuration, etc.
  - There is no unique, well-known administrator identifier
  - Administration privileges can be bound to several UIDs
    - Usually through administration groups
    - Administrators, Power Users, Backup Operators

Group Identifiers (GID)

OS also address group identifiers

- A group is composed by zero or more users
- A group may be composed by other groups
- Group ID: Integer value (Linux, Android, macOS) or UUID (Windows)

User may belong to multiple groups

- User rights = rights of its UID + rights of its GIDs

In Linux, activities always execute under the scope of a set of groups

- 1 primary group: user to define the ownership of created files
- Multiple secondary groups: used to condition access to resources

Processes

A process defines the context of an activity

- For taking security-related decisions
- For other purposes (e.g., scheduling)

Security-related context

- Effective Identity (eUID and eGIDs)
  - Fundamental for enforcing access control
  - May be the same as the identity of the user launching the process
- Resources being used
  - Open files
    - Including communication channels
  - Reserved virtual memory areas
  - CPU time used, priority, affinity, namespace

Virtual Memory

The address space where activities take place

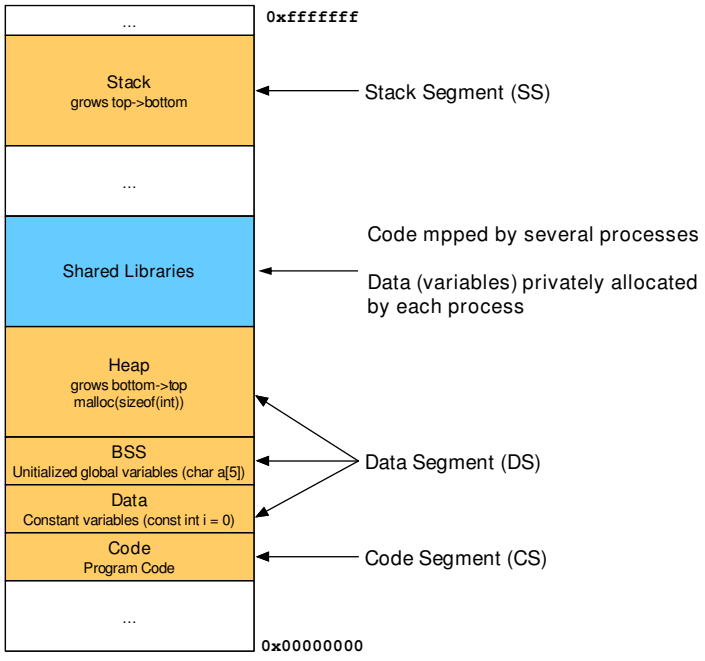
- Have the maximum size defined by the hardware architecture
  - 32 bits -> 2^{32} Bytes
  - 64 bits -> 2^{64} Bytes
- Managed in small chunks named pages (4 KiB)

Virtual Memory can be sparse

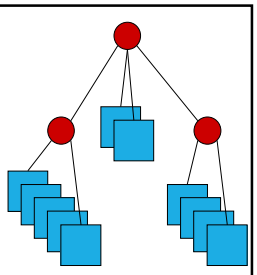
- Only the pages used must be allocated
- Although processes always see a contiguous memory space

Virtual Memory is mapped to RAM when actually used

- At a given moment, the RAM has pages from multiple address spaces
- The choice of how to manage those spaces is very important
  - Avoid fragmentation, management memory according to their freshness



File System: objects



Hierarchical structure for storing content

- Provide a method for representing mount points, directories, files and links

Mount Point

- An access to the root of a specific FS
- Windows uses letters (A:, .. C:..)
- Linux, macOs, Android use any directory

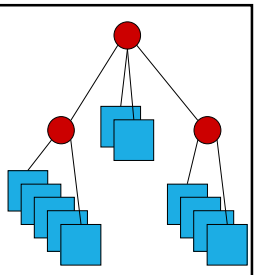
Directory (or folder)

- A hierarchical organization method
  - Similar to a container
- Can contain other directories, files, mount points, links
- The first (or top-most) is called by root

Links

- Indirection mechanisms in FS
- Soft Links: point to another feature in any FS
  - Windows: Shortcuts are similar to Soft Links, but handled at the application level
- Hard Links: provide multiple identifiers (names) for the same content (data) in the same FS
  - Usually allowed only for files

File System: files



Serve to store data on a persistent way

- But longevity is given by physical support and not by the file concept ...
- Erasing often means marked as deleted

Ordered sequences of bytes associated with a name

- The name allows you to retrieve/reuse these bytes later
- Its contents can be changed, removed, or added
  - As well as the name
- They have a protection that controls their use
  - Read, write, run, remove, lock, etc. permissions
  - The protection model depends on the file system

File System: security mechanisms

Mandatory protection mechanisms
- Owner
- Users and Groups allowed
- Permissions: Read, Write, Run
  - Different meanings for Files and Directories

Discretionary protection mechanisms
- User-defined specific rules

Additional mechanisms
- Implicit compression
- Indirection to remote resources (e.g., for OneDrive)
- Signature
- Encryption

Communication Channels

Allow the exchange of data between distinct but cooperative activities

Essential in any current system
- All applications use these mechanisms

Processes of the same SO/machine
- Pipes, UNIX Sockets, streams, etc.
- Communication between processes and kernel: syscalls, sockets

Processes on different machines
- TCP/IP and UDP/IP sockets

Canais de comunicação que permitem a troca de informações entre processos, threads, ou componentes de um sistema

Access Control

An OS kernel is an access control monitor

- Controls all interactions with the hardware
- Applications NEVER directly access resources
- Controls all interactions between computational model entities

Subjects

- Typically, local processes
  - Through the system calls API
  - A syscall is not an ordinary call to a function
- But also messages from other machines


```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main(int argc, char** argv){
     FILE *fp = fopen("hello.txt", "wb");
     char* str = "hello world";
     fwrite(str, strlen(str), 1, fp);
     fclose(fp);
}
```

```
$ gcc -o main ./main


$ strace ./main

....

openat(AT_FDCWD, "hello.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0

write(3, "hello world", 11)             = 11

close(3)                                = 0

...
```

File interactions are mediated by the kernel
Applications do not directly access resources

Mandatory Access Control

There are numerous cases of mandatory access control on an operating system

- They are part of the logic of the computational model
- They are not moldable by users and administrators
  - Unless they change the behavior of the kernel

Examples on Linux

- root can do everything
- Signals to processes can only be sent by root or the owner
- Sockets AF_PACKET (RAW) can only be created by root or by processes with the CAP_NET_RAW

Examples on macOS

- root can do almost anything
- root cannot change binaries and directories signed by Apple

Discritionary Access Control

Users can set rules for access control

- May be definable only by the owner/user
  - This limitation is itself a Mandatory Access rule

Examples

- Discretionary Access Control Lists (ACL)
  - Expressive lists that limit access to resources Linux
- Linux Apparmor
  - Stores settings in /etc/apparmor.d with application limitations
  - Rules applied automatically to applications regardless of user
- macOS sandboxd
  - Applications are launched within isolated contexts (sandbox)
  - The sandbox contains a definition of the information that enters/exits

Protection with ACLs

Each object has an Access Control List (ACL)

- Tell me who can do what

The ACL may be discretionary or mandatory

- When it is mandatory you cannot change
- When it is discretionary it can be changed

It is checked when an activity intends to manipulate the object

- If the manipulation request is not authorized it is denied
- The SO kernel makes the ACL validations
  - Acts as a Reference Monitor

Unix file protection ACLs:
Fixed-structure, discretionary ACL

Each file system object has an ACL

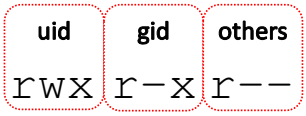
- Binding 3 rights to 3 subjects
- Only the owner can update the ACL

Rights: R W X

- Read right / Listing right
- Write right / create or remove files or subdirectories
- Execution right / use as process' current working directory

Subjects

- An UID (owner)
- A GID
- Others




```
[nobody@host ~]$ ls -la
total 12
drwxr-xr-x   2 root root 100 dez  7 21:39 .
drwxrwxrwt  25 root root 980 dez  7 21:39 ..
-rw-r-----   1 root root   6 dez  7 21:42 a
-rw-r--r--   1 root root   6 dez  7 21:42 b
-rw-r-x---+  1 root root   6 dez  7 21:42 c

[nobody@host ~]$ cat a
cat: a: Permission denied

[nobody@host ~]$ cat b
SIO_B
[nobody@host ~]$ cat c
SIO_C

[nobody@host ~]$ getfacl c
# file: c
# owner: root
# group: root
user::rw-
user:nobody:r-x
group::r--
mask::r-x
other::---
```

Windows file protection ACLs: Flexible-structure, discretionary ACL

Each file system object has an ACL and an owner

- The ACL grants 14 types of access rights to a variable-size list of subjects
- Owner can be an UID or a GID
- Owner has no special rights over the ACL

Subjects:

- Users (UIDs)
- Groups (GIDs)
  - The group "Everyone" stands for anybody

Rights:

- Traverse Folder / Execute File
- List Folder / Read Data
- Read Attributes
- Read Extended Attributes
- Create Files /Write Data
- Create Folders / Append Data
- Write Attributes
- Write Extended Attributes
- Delete Subfolders and Files
- Delete
- Read Permissions
- Change Permissions
- Take Ownership

Privilege Elevation: Set-UID

It is used to change the UID of a process running a program stored on a Set-UID file

- If the program file is owned by UID X and the set-UID ACL bit is set, then it will be executed in a process with UID X, independently of the UID of the subject that executed the program

It is used to provide privileged programs for running administration task invoked by normal, untrusted users

- Change the user's password (passwd)
- Change to super-user mode (su, sudo)
- Mount devices (mount)

Privilege Elevation: Set-UID

Effective UID / Real UID

- Real UID is the UID of the process creator
  - App launcher
- Effective UID is the UID of the process
  - The one that really matters for defining the rights of the process

UID change

- Ordinary application
  - eUID = rUID = UID of process that executed exec
  - eUID cannot be changed (unless = 0)
- Set-UID application
  - eUID = UID of exec'd application file, rUID = initial process UID
  - eUID can revert to rUID
- rUID cannot change

Set-UID, o EUID é temporariamente definido como o UID do proprietário do programa

O Real UID é o identificador de utilizador real. Ele representa o utilizador que iniciou o processo

Privilege Elevation: Set-UID

Administration by root is not advised

- One "identity", many people
- Who did what?

Preferable approach

- Administration role (uid = 0), many users assume it
  - Sudoers
  - Defined by a configuration file used by sudo

sudo is a Set-UID application with UID = 0

- Appropriate logging can take place on each command run with sudo

```
[user@linux ~]$ ls -la /usr/sbin/sudo
-rwsr-xr-x 1 root root 140576 nov 23 15:04 /usr/sbin/sudo

[user@linux ~]$ id
uid=1000(user) gid=1000(user) groups=1000(user),998(sudoers)

[user@linux ~]$ sudo -s
[sudo] password for user:

[root@linux ~]# id
uid=0(root) gid=0(root) groups=0(root)

[root@linux ~]# exit

[user@linux ~]$ sudo id
uid=0(root) gid=0(root) groups=0(root)
```


Linux login: Not an OS kernel operation

A privileged login application presents a login interface for getting users' credentials

- A username/password pair
- Biometric data
- Smartcard and activation PIN

The login application validates the credentials and fetches the appropriate UID and GIDs for the user

- And starts an initial user application on a process with those identifiers
  - In a Linux console this application is a shell
- When this process ends the login application reappears

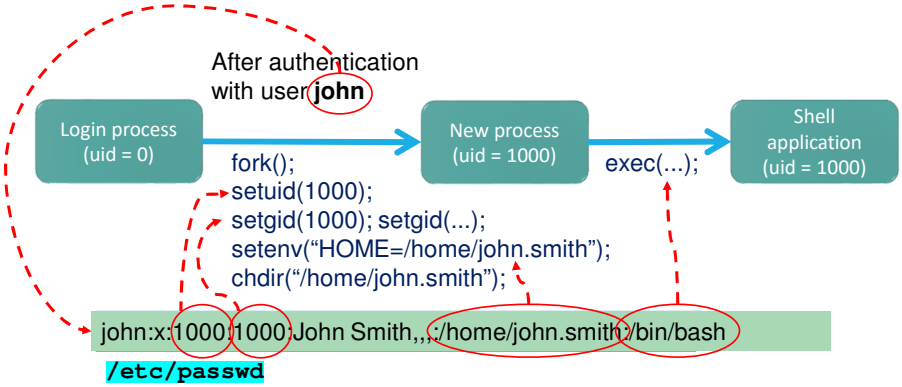
Thereafter all processes created by the user have its identifiers

- Inherited through forks

Linux: from login to session processes

The login process must be a privileged process

- Has to create processes with arbitrary UID and GIDs
  - The ones of the entity logging in



Login in Linux: Password validation process

Username is used to fetch a UID/GID pair from /etc/passwd

- And a set of additional GIDs in the /etc/group file

Supplied password is transformed using a digest function

- Currently configurable, for creating a new user (/etc/login.conf)
- Its identification is stored along with the transformed password

The result is checked against a value stored in /etc/shadow

- Indexed again by the username
- If they match, the user was correctly authenticated

File protections

- /etc/passwd and /etc/group can be read by anyone
  - Required for UID/GID → user name / group name translations
- /etc/shadow can only be read by root
  - Protection against dictionary attacks

Chroot mechanism

Used to reduce the visibility of a file system

- Each process descriptor has a root i-node number
  - From which absolute pathname resolution takes place
- chroot changes it to an arbitrary directory
  - The process' file system view gets reduced

Used to protect the file system from potentially problematic applications

- e.g., public servers, downloaded applications
- But it is not bullet proof!

Confinement: AppArmor

Mechanism for restricting applications based on a behavior model

- Requires kernel support
  - Linux Security Modules
- Focus on syscalls and their arguments
- Can work in complain and enforcement modes
- Generates entries in the system registry to audit the behavior

Configuration files define allowed activities

- Whitelisting
- By application, uploaded from a file
- Applications can never have more accesses than defined
  - Even if executed by root

Projetado para reforçar a segurança do sistema, limitando as ações que processos de aplicações podem executar.

Tem o objetivo de impedir que aplicações maliciosas ou comprometidos causem danos ao sistema ou a outras aplicações.

Confinement: Namespaces

Allows partitioning of resources in views (namespaces)

- Processes in a namespace have a restricted view of the system
- Activated through syscalls by a simple process:
  - clone: Defines a namespace to migrate the process to
  - unshare: disassociates the process from its current context
  - setns: puts the process in a Namespace

Types of Namespaces

- Mount: Applied to mount points
- process id: first process has id 1
- network: "independent" network stack (routes, interfaces...)
- IPC: methods of communication between processes
- uts: name independence (DNS)
- user id: segregation of permissions
- cgroup: limitation of resources used (memory, cpu...)

Confinement: Containers

Explores namespaces to provide a virtual view of the system

- Network isolation, user ids, mounts, cgroups, etc...

Processes are executed under a container

- A container is an applicational construction and not a kernel object
- Consists of an environment by composition of namespaces and cgroups
- Requires building bridges with the real system network interfaces, proxy processes

Relevant approaches

- LinuX Containers: focus on a complete virtualized environment
  - evolution of OpenVZ
- Docker: focus on running isolated applications based on a portable packet between systems
  - uses LXC
- Singularity: similar to docker, focus on HPC and multi-user sharing

São uma tecnologia de virtualização de nível de sistema operativo que permite empacotar, distribuir e executar aplicações e os seus ambientes de forma isolada e eficiente.

Defending an Organization

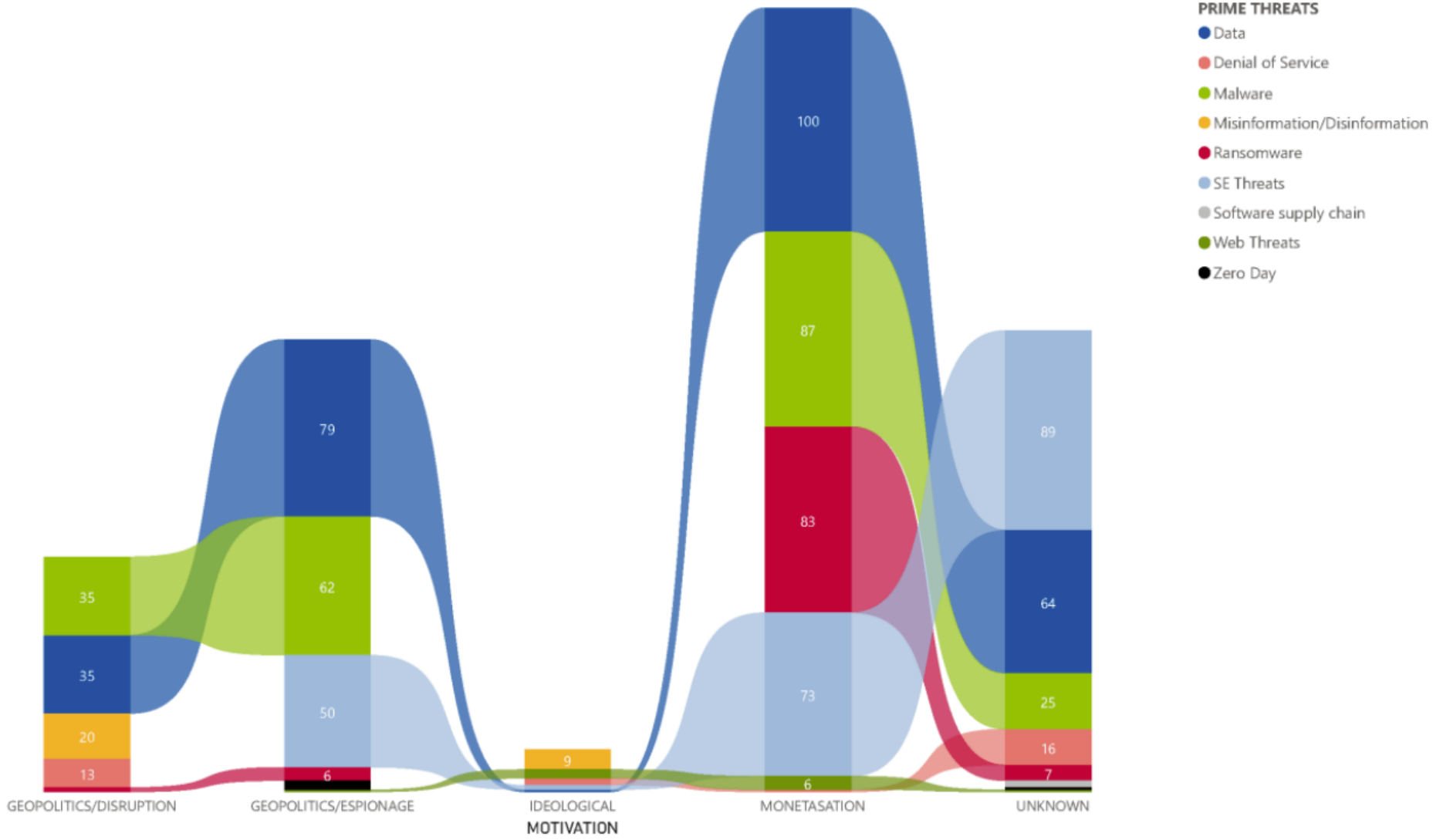
The current organizational landscape

- **Organizations are complex and must reach everyone**

- **Physical space: where we live since >10000y BC**
  - We know it, it's slow, it involves moving matter around
  - Laws are plentiful and cover most interactions

- **Cyberspace: to which organizations just tapped into**
  - We do not know it, it's fast, there are no barriers
  - Everything can be hidden, laws are limited

Malicious actors are highly motivated and organized



The current organizational landscape

- **Must comply with new regulatory frameworks**
  - 2016: NIS – Defines basic cybersec requirements
  - 2018: GDPR – Defines requirements for private data
    - Introduces fines for lack of proper data management
  - 2021: DL65 – Defines processes for inventory, reporting, formalize strategy
  - 2024: NIS 2 – Defines cyber teams and processes
    - Introduces fines for lack of security

- **Strategies are based on risk and maturity**
  - Risk: identify assets and determine their risk
  - Maturity: determine organization maturity over multiple áreas
    - Evolve all as adequate

Current requirements

1. **Identify security accountable individual**
   - Responsible for the Security Strategy
   - Typically called CISO: Chief Information Security Officer
   - Will be personally held accountable!
2. **Identify contact points for the organization**
3. **Identify and track the critical assets**
   - Crown Jewels
4. **Have a security plan**
5. **Report relevant incidents and cooperate**

Assets: Crown Jewels Approach

- **Focused on identifying and protecting the most critical assets**
  - To the organization mission!
- **What is a crown jewel?**
  - Sensitive Data
  - Servers
  - Software Systems
  - Any other equipment (HVAC, Generators…)
- **Disruption to the crown jewels will pose a serious impact to the organization mission**
- **Objective: Protect the crown jewels**
  - and grow from there to the rest of the organization
  - based on a risk assessment

Security Plan

- **Live document describing the security posture**
  - Allows organizations to know where they are and where they want to go
  - Considers authentication, backups, risk, access control, policies, etc.
- **Accepted by the organization, signed by Security Principal**
  - Periodically reviewed and improved
- **Written and accepted policies implies higher maturity**
  - Organizations frequently only have word of mouth or informal frequent practices

Incident Response and Coordination

- **Incident response coordinated by**
  - Relevant incidents must be reported

- **National CSIRT Network facilitates collaboration between entities**
  - https://www.redecsirt.pt

- **Fraud/Crime incidents are reported to authorities**
  - **Unidade Nacional de Combate ao Cibercrime e à Criminalidade Tecnológica (UNC3T):** unc3t@pj.pt

Cybersecurity at UA

- **Data: DPO (Data Protection Officer)**
  - https://www.ua.pt/pt/rgpd/page/24346
- **Strategy and Support: Cybersecurity Office**
  - https://www.ua.pt/ciberseguranca
- **Incident Response: the CSIRT@UA part of RNCSIRT**
  - https://www.ua.pt/pt/ciberseguranca/rfc2350
- **Research: several projects and thesis**
- **Education and Training:**
  - Cybersecurity in Teaching Units (DETI)
  - Masters in Cybersecurity (DETI)
  - Higher Professional Training Course in Cybersecurity (ESTGA)
  - Courses with CNCS C-Academy and at UNAVE

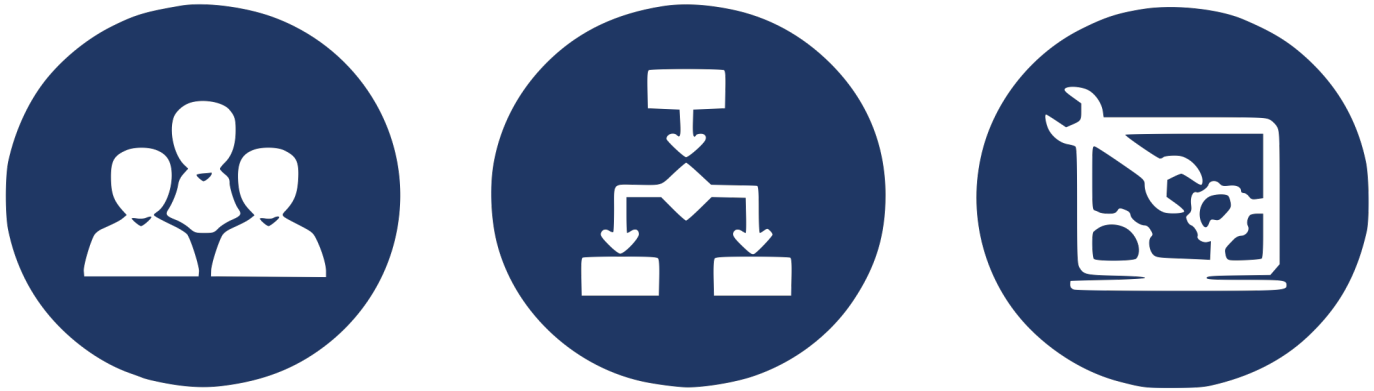
Security Teaming

- **Security operations are frequently organized in teams**
  - **Blue Team:** Defends an organization from malicious actors
  - **Red Team:** Attacks an organization to help finding weak spots
  - **Purple Team**: Mixed attack defence role
- **Each team uses specific tools and methods**

Blue Teams

- **Defend organizations from malicious actors**
  - Abusing and Careless actors, and general failures also

- **Typical fundamental tasks to address:**
  - People: training, awareness, culture
  - Processes: analysis, investigation, data, reporting
  - Technology: monitoring, detection, scripting, automation



Blue Teams

- **Mandatory for all organizations!**
  - Good amount of job opportunities
    - extreme shortage of professionals

- **Very demanding due to high asymmetry**
  - Attackers must succeed **once**, using their preferred TTPs
  - Defenders must defend **continuously**, from all attacks
    - To the entire organization attack surface, using any TTP

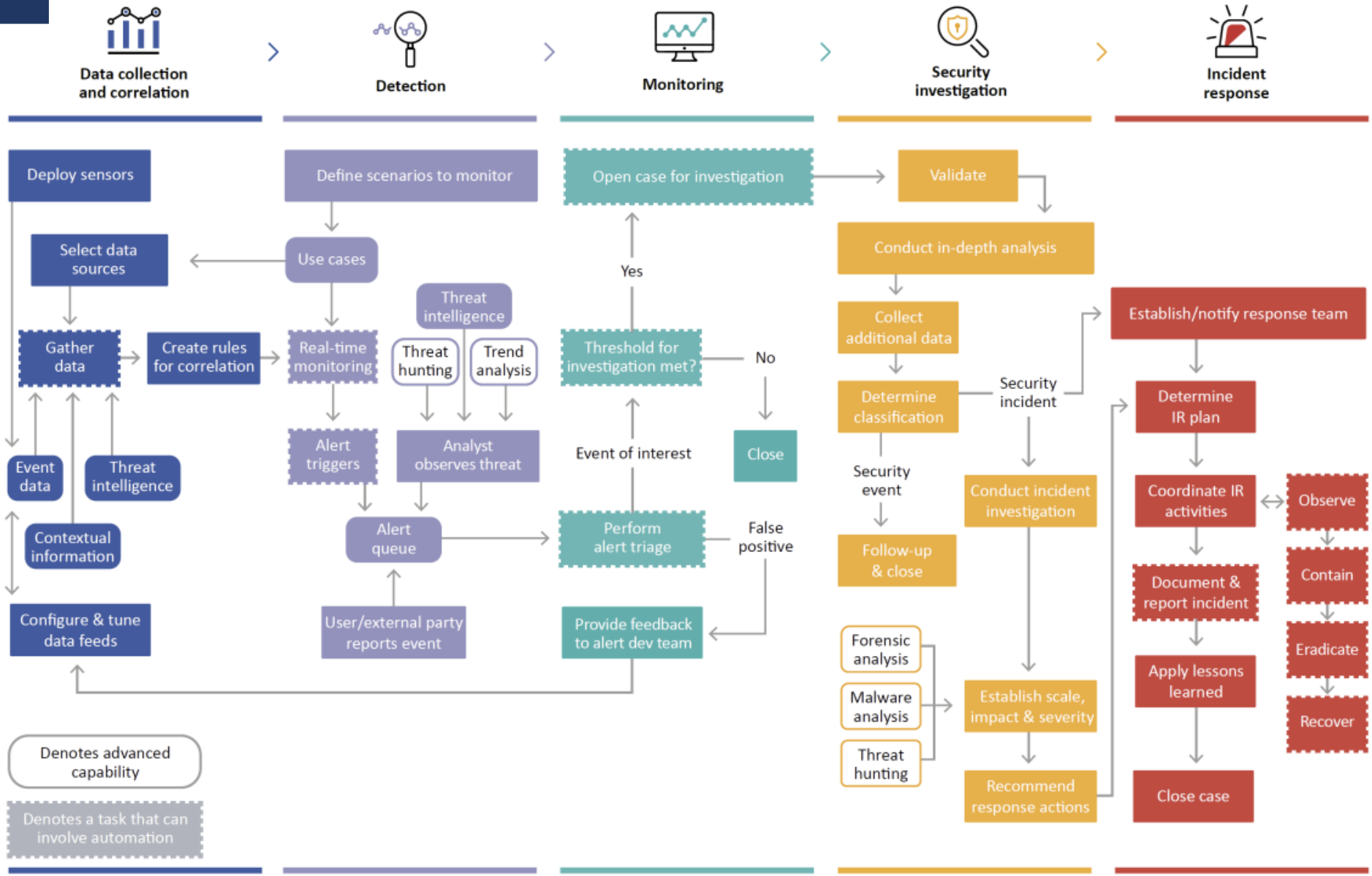
- **Challenging and interesting**
  - Many topics to address: prog, forensics, AI/ML, training…
  - Continuously evolving with new techniques and tools

Blue Team Defence Techniques

- **Everything Everywhere All at Once?**
  - No! Prioritize according to the organization mission
- **Current approaches focus on:**
  - the CIA triad
  - the crown jewels
    - Risk assessment
  - with the least pain
  - security plan

SOC – Security Operations Center

- **Responsible for continuously monitoring**
  - Organization's digital infrastructure
- **Monitor, detect and respond**
  - To cybersecurity threats
- **Empowered with skilled analysts and technology**
  - Security assessments
  - Data protection
  - Incident response


Data collection and correlation
Detection
Monitoring
Security investigation
Incident response
Deploy sensors
Select data sources
Gather data
Create rules for correlation
Event data
Threat intelligence
Contextual information
Configure & tune data feeds
Define scenarios to monitor
Use cases
Real-time monitoring
Threat intelligence
Threat hunting
Trend analysis
Alert triggers
Analyst observes threat
Alert queue
User/external party reports event
Open case for investigation
Yes
Threshold for investigation met?
No
Close
Event of interest
Perform alert triage
False positive
Provide feedback to alert dev team
Validate
Conduct in-depth analysis
Collect additional data
Determine classification
Security incident
Security event
Follow-up & close
Conduct incident investigation
Forensic analysis
Malware analysis
Threat hunting
Establish scale, impact & severity
Recommend response actions
Establish/notify response team
Determine IR plan
Coordinate IR activities
Observe
Contain
Eradicate
Recover
Document & report incident
Apply lessons learned
Close case
Denotes advanced capability
Denotes a task that can involve automation

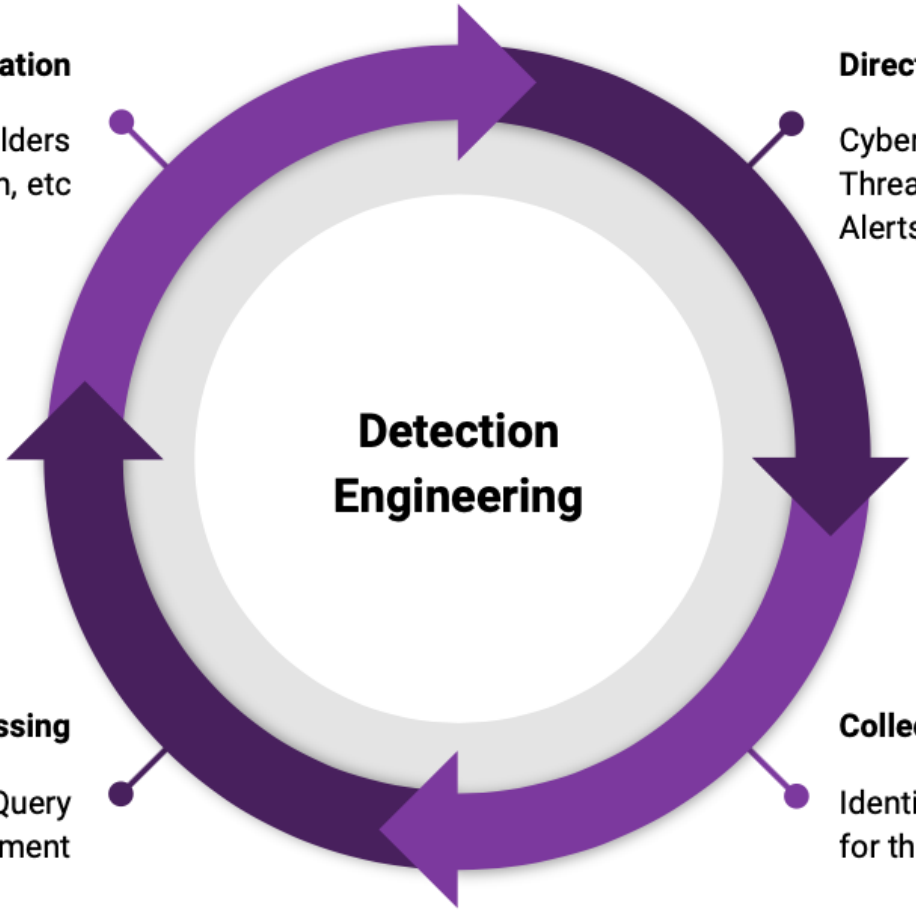
Main concepts

- **Defensive Security Engineering**
  - Firewalls, backups, logs
  - Secure Software Development Lifecycle
  - Security related requirements (e.g., OWASP ASVS)

- **Incident Response**
  - Have processes to handle incidents
  - Involve stakeholders and communicate

- **Detection Engineering**
  - designing, developing, testing, and maintaining threat detection logic

Detection Engineering

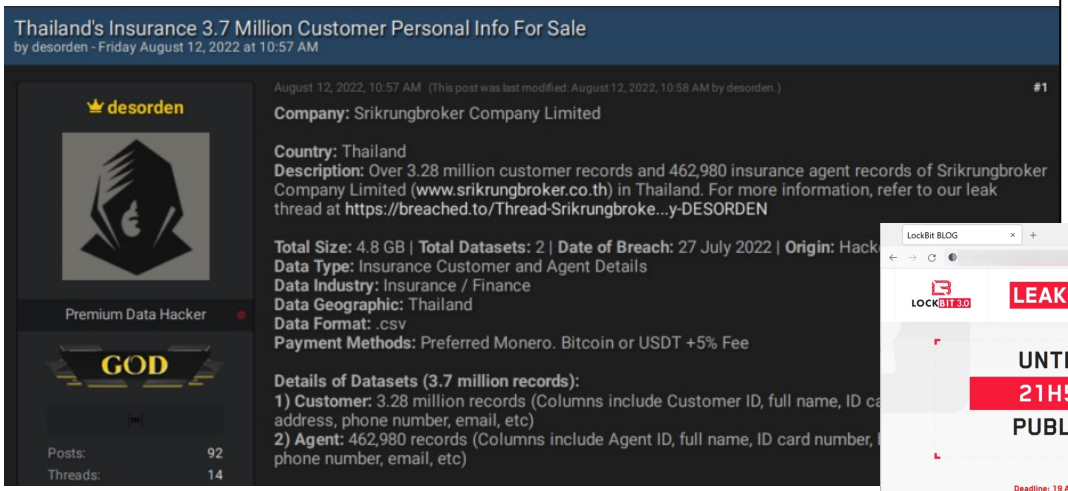


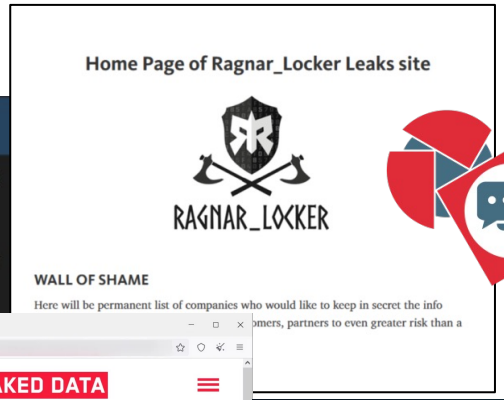
Source: SANS

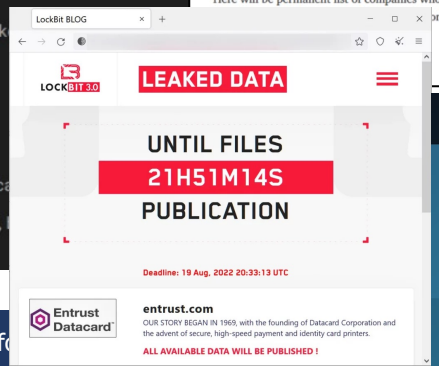
Direction: CTI

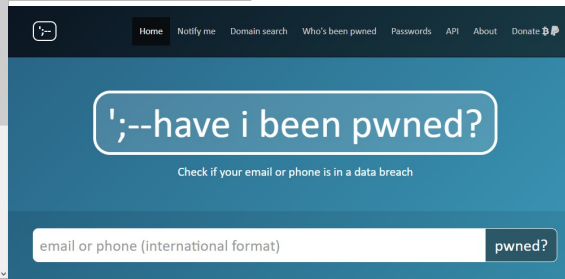
Assess the current threats from CTI

- **Cyber Threat Intelligence helps understanding the dynamics**
  - The “Dark web”: Tor forums, discords, telegrams, IRC, twitter, pastebins
  - Official reports: Security Researchers (Reversing, analysis)
  - How actors position themselves (hacktivists, crime)
  - Attacks to similar organizations





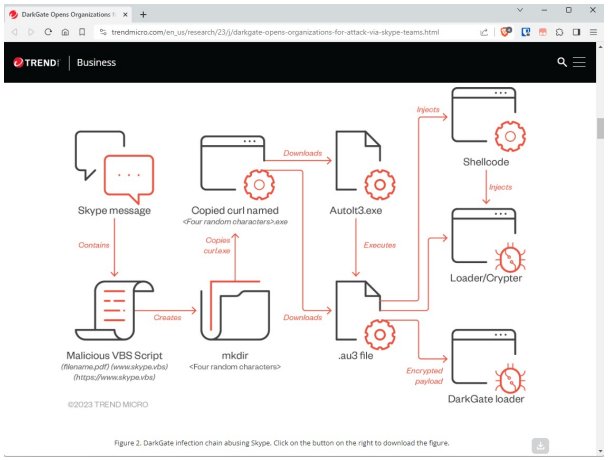


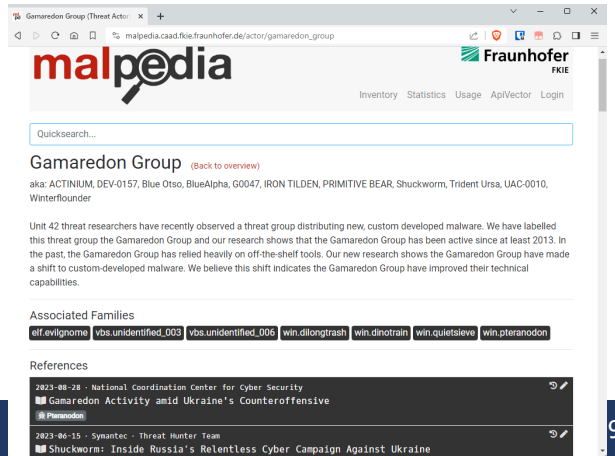


Direction: CTI

Assess the current threats from CTI

- **Threat Intelligence from researchers provide analysis and forecasts**
  - Official entities, private orgs





Direction: Alerts and Incidents

Alerts and Incidents

- **Current alerts will tailor future rules**
  - Identify popular threat actions
  - Reduce false positives
  - Keep the capability to detect new threats
    - Includes conducting controlled attacks to validate rules

- **Incident resolution impact resolution playbooks**
  - One a threat is found, what can the organization do?
  - Deficiencies in incident response define future improvements
    - Includes simulated incidents to test processes

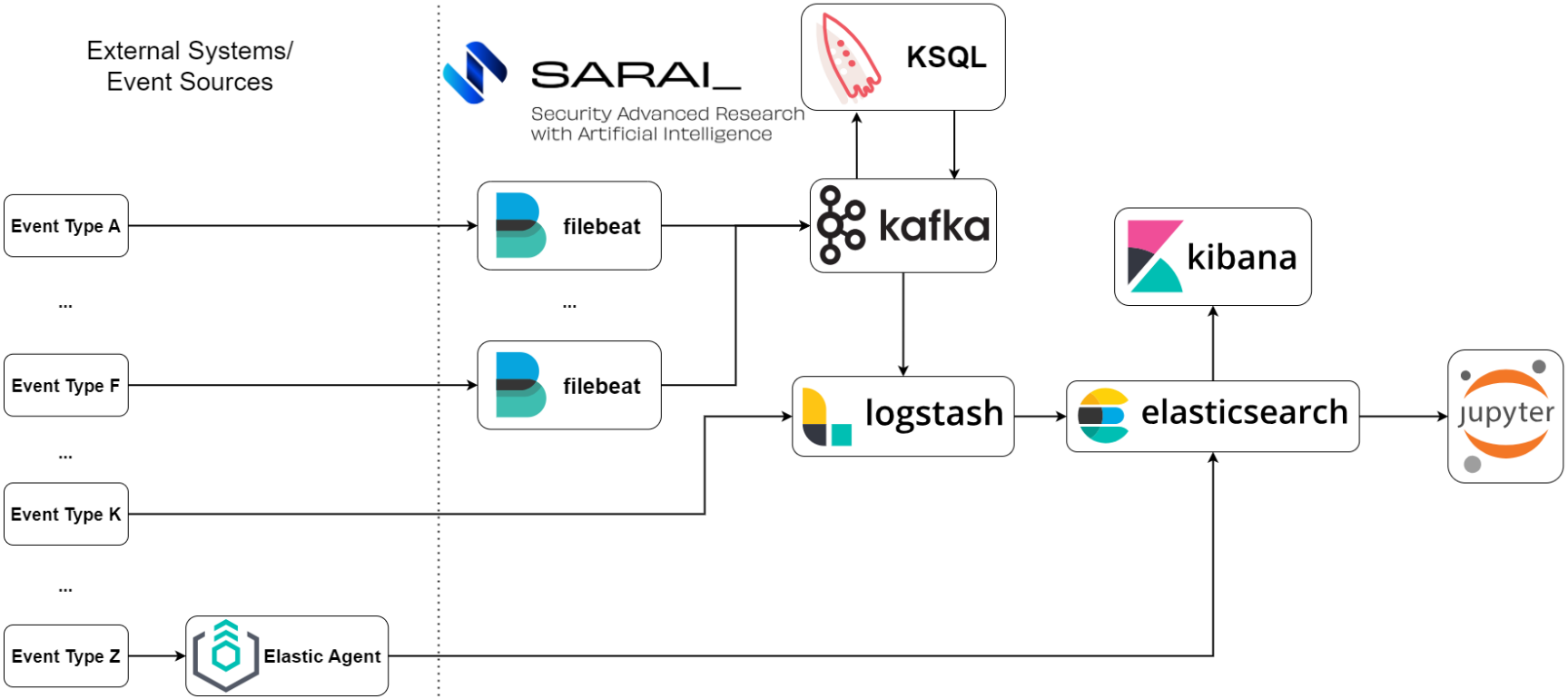
Collection: Data Harvesting

Engineer Data Collection

- **Focus on relevant data sources to address threats**
  - Cannot get all data
  - Visiblity will be limited
- **Potential targets**
  - Servers: AD, email, HTTP, Databases
  - Wireless Controllers
  - VPN access
  - Firewalls
  - Endpoints: Laptops, VMs, IoT devices

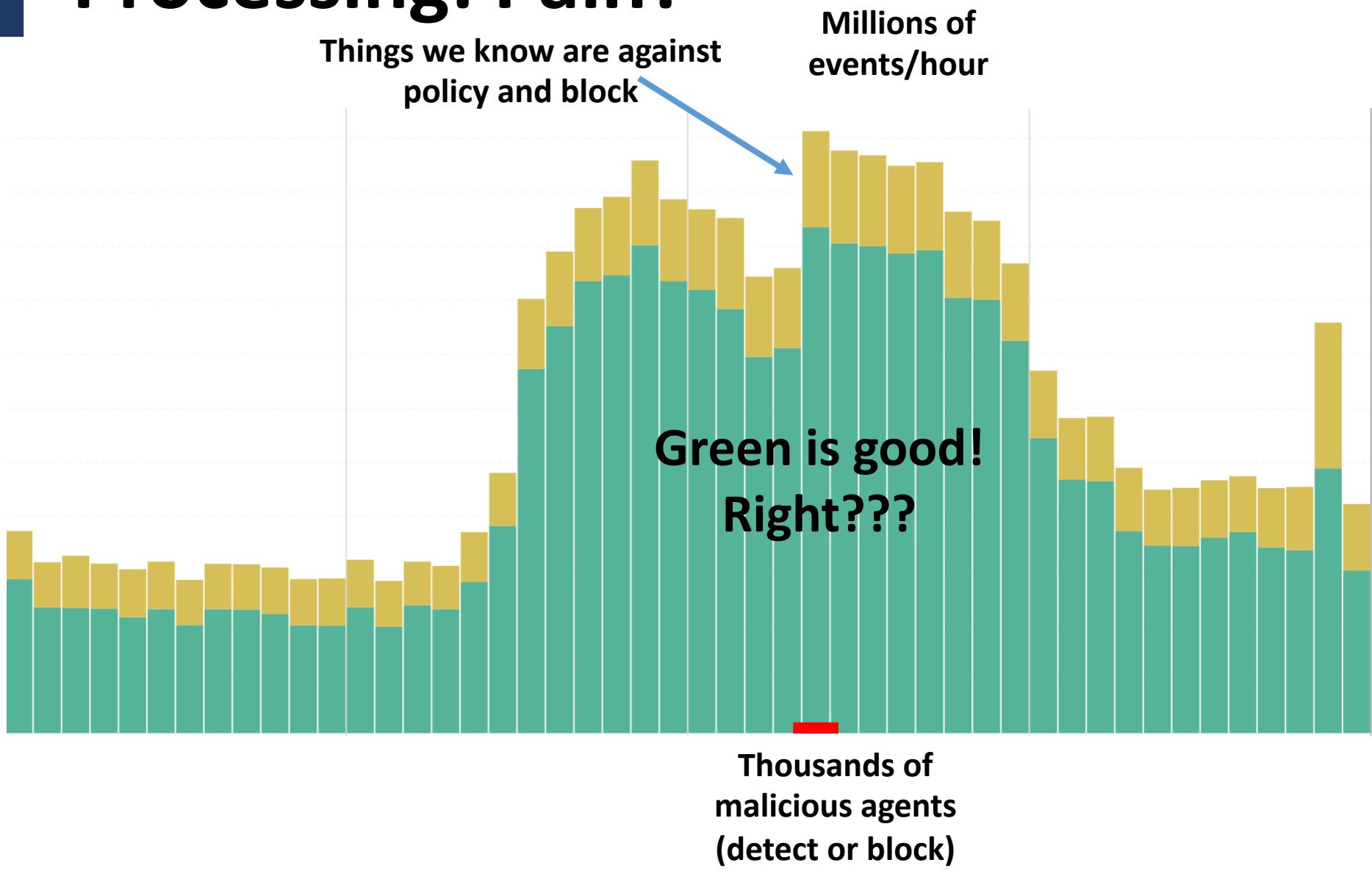
Collection: Data Harvesting

- **Current approaches focus on a large data lake**
  - Algorithms match rules, ML models, signatures, behavior

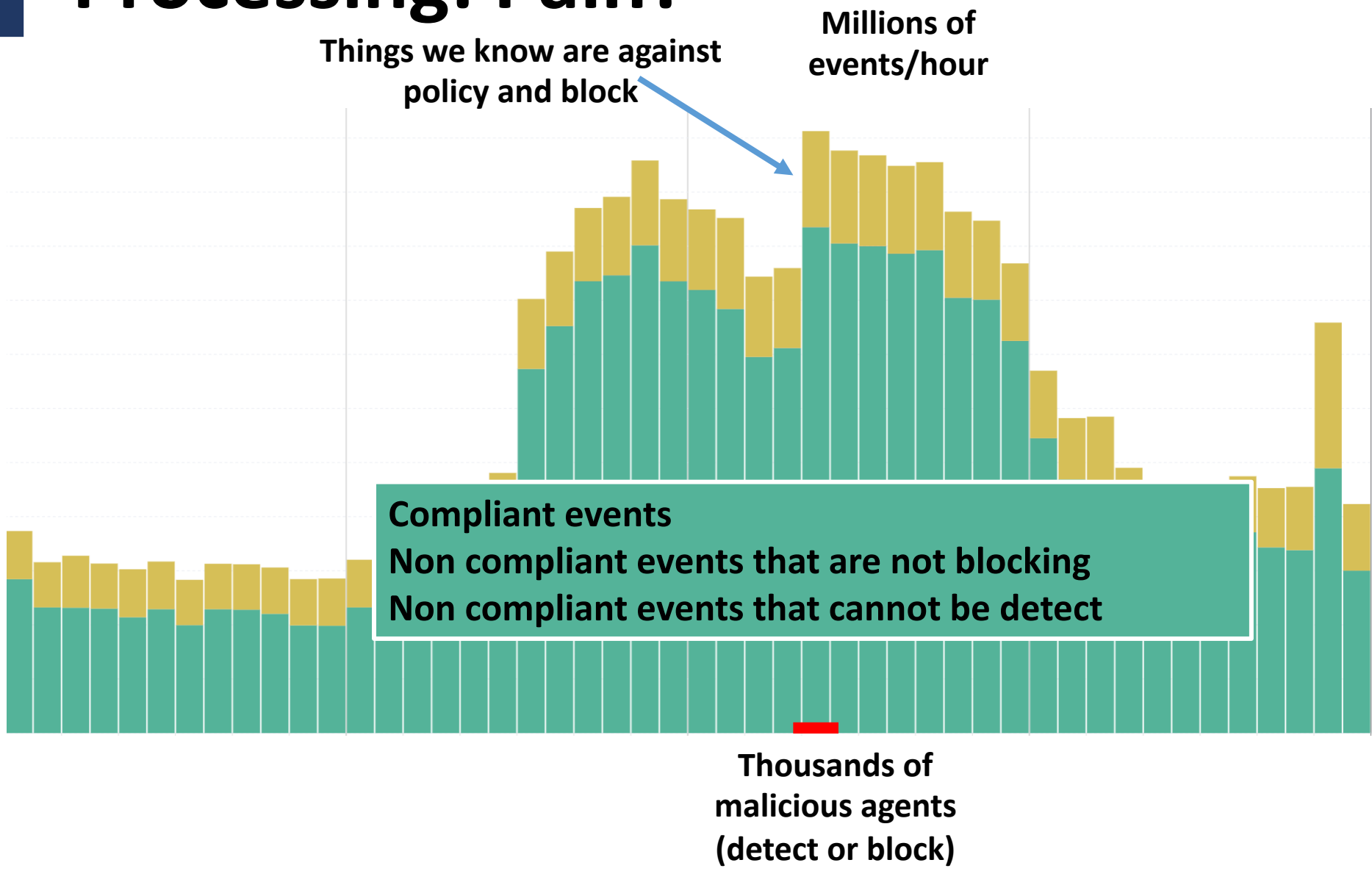


https://github.com/gcsuaveiro/gcs-sarai

Processing: Pain?



Processing: Pain?



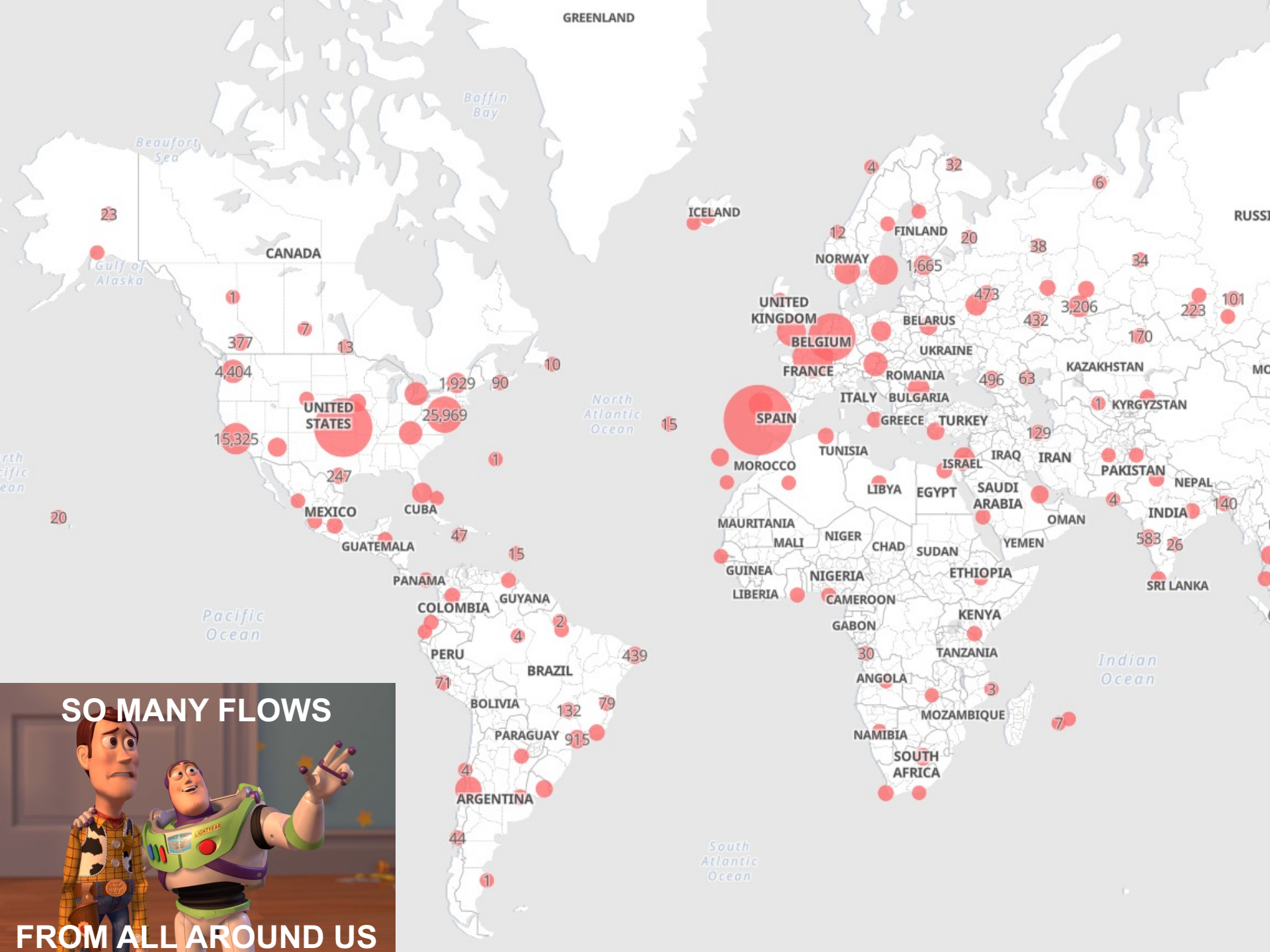
Processing: Pain?

Things we know are against policy and block

Millions of events/hour

Things we know
Things we know we do not know
Things we do not know we do not know

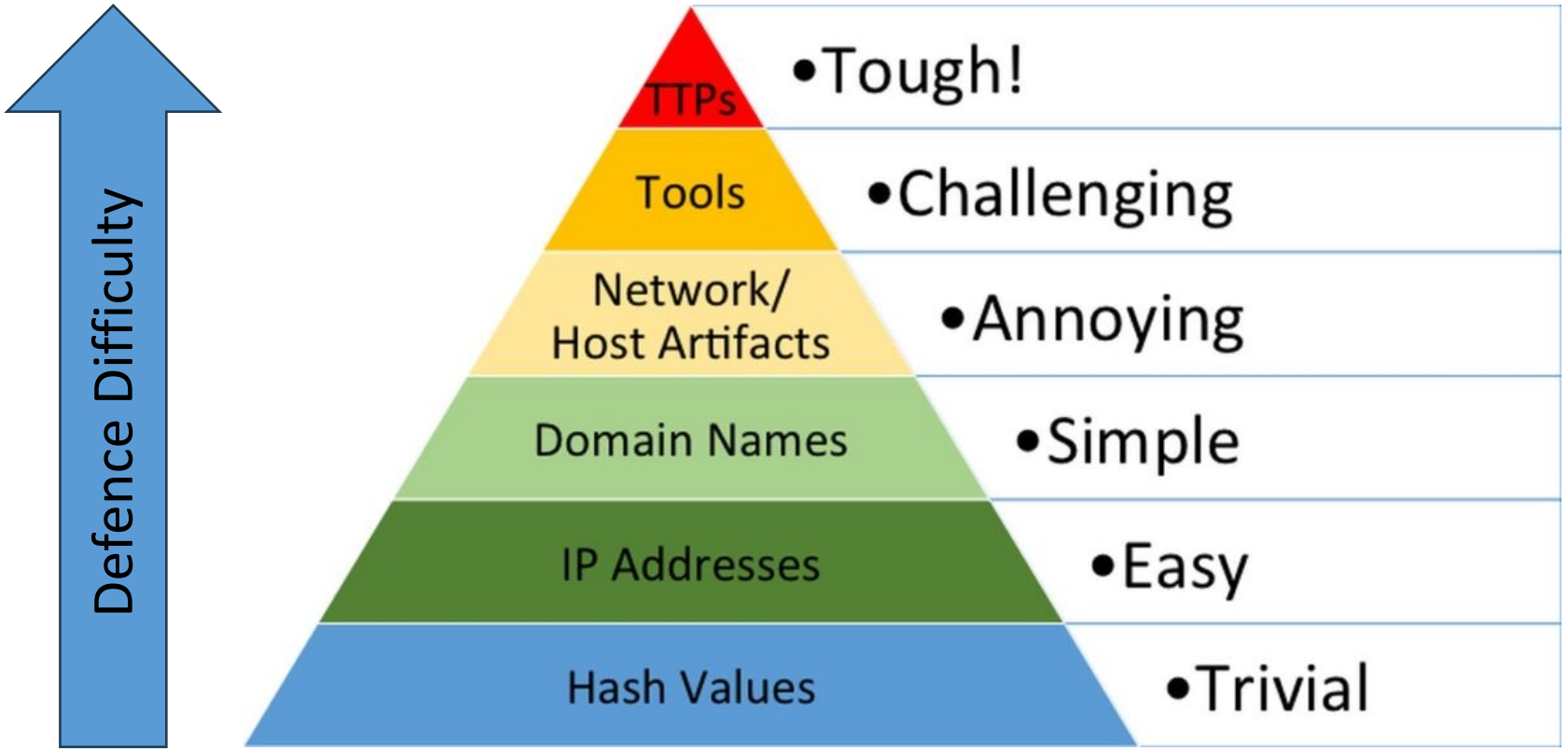
Thousands of malicious agents (detect or block)



SO MANY FLOWS

FROM ALL AROUND US

The Pyramid of Pain



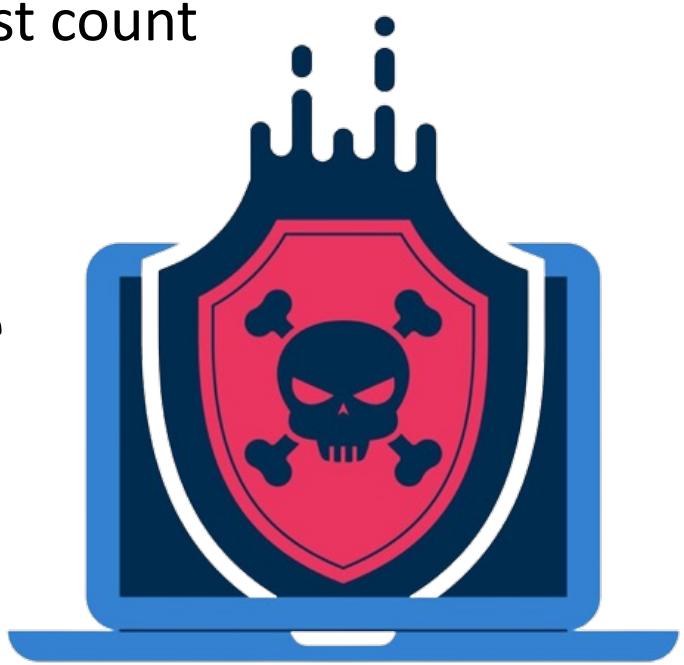
- Increase defence capabilities from the bottom to the top
- Why?
  - Detecting files/emails by comparing hashes is trivial
  - Understanding how actors behave is very difficult

Triage

- **One of an analyst's most important tasks**
  - You will likely never have only one option
- **Think like an ER doctor / emergency dispatcher**
  - Limited resources
  - Multiple problems
- **Where do you start?**
- **Pick the most dangerous alert**
  - Main goal, although it is difficult to chose

Definition of Dangerous

- **Could be one of several definitions**
  - Attack near completion
  - Targeting / affecting high-value items
    - Critical hosts, business processes, users, data
  - Advanced or targeted attackers
  - Unique, never fired before or lowest count
- **Will depend on the organization**
- **Anything that will cause damage**
  - It have a high cost
  - Or it is difficult to remedy
    - if it succeeds



How to find threats?

- **Behavior matching: mostly ML**
  - Known patterns
  - Anomally detection
- **Signature matching: YARA**
  - Signatures for malware are created and disseminated
- **Reputation evaluation: IP addresses /domains**
  - Low reputation addresses may generate alert or block
- **Known threats are identified be vendor software**
  - Challenge: Unknown/Tailored threats

How to find threats?

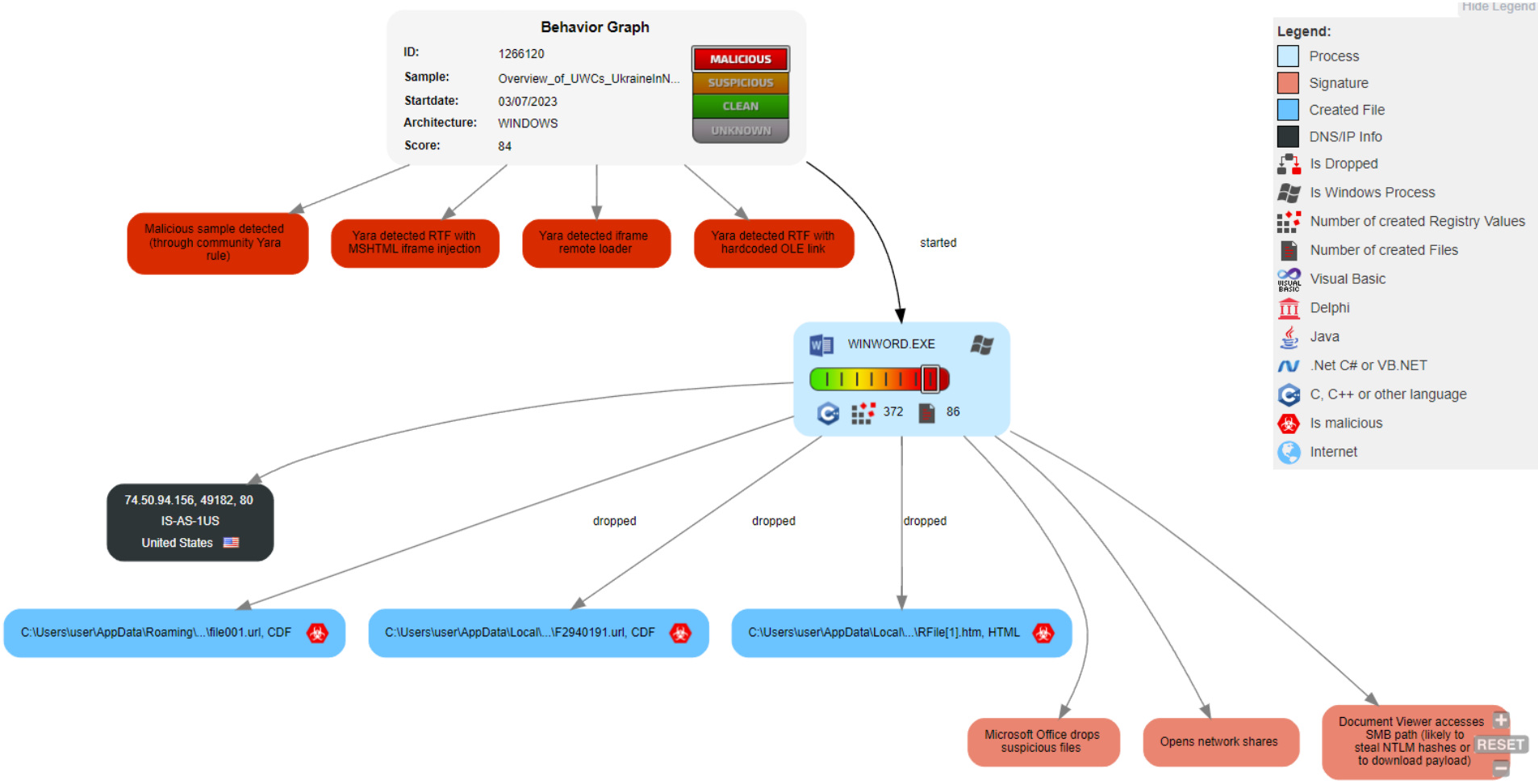
- **What if we do not know if something is malicicous?**
  - What is a malicious website or file?
- **New malware potentially has high impact**
  - It is not detected by Anti-virus
  - Explores unpatched vulnerabilities or flaws (0 day)
- **A new malicious asset is just a new program/website**
  - May be a variation of a existing malware
    - Different language/obfuscated/encrypted/packed
  - May simply bypass existing signatures
  - There is an robust market selling malware

Threat Research

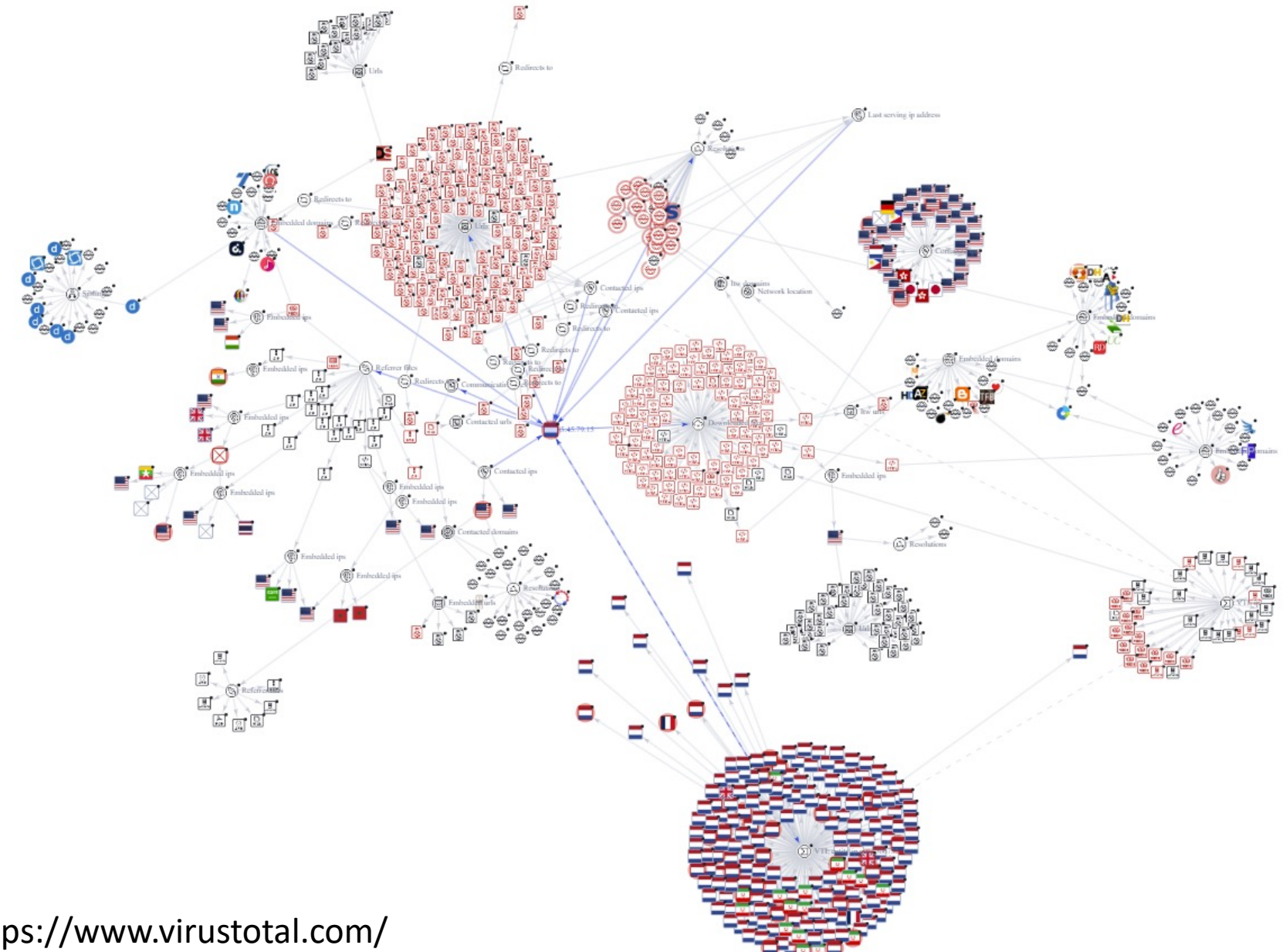
- **Threat Research allows detection of new offenses**
  - Takes a Indicator and determines its risk

- **Includes several knowledge areas**
  - Open Source Intelligence
    - Social Networks, DNS/TLS Records, Dark Web
  - Reverse Engineering
  - Networking concepts
  - Network traffic analysis
  - Cryptography
  - Machine Learning

Threat Research: Execution Graphs



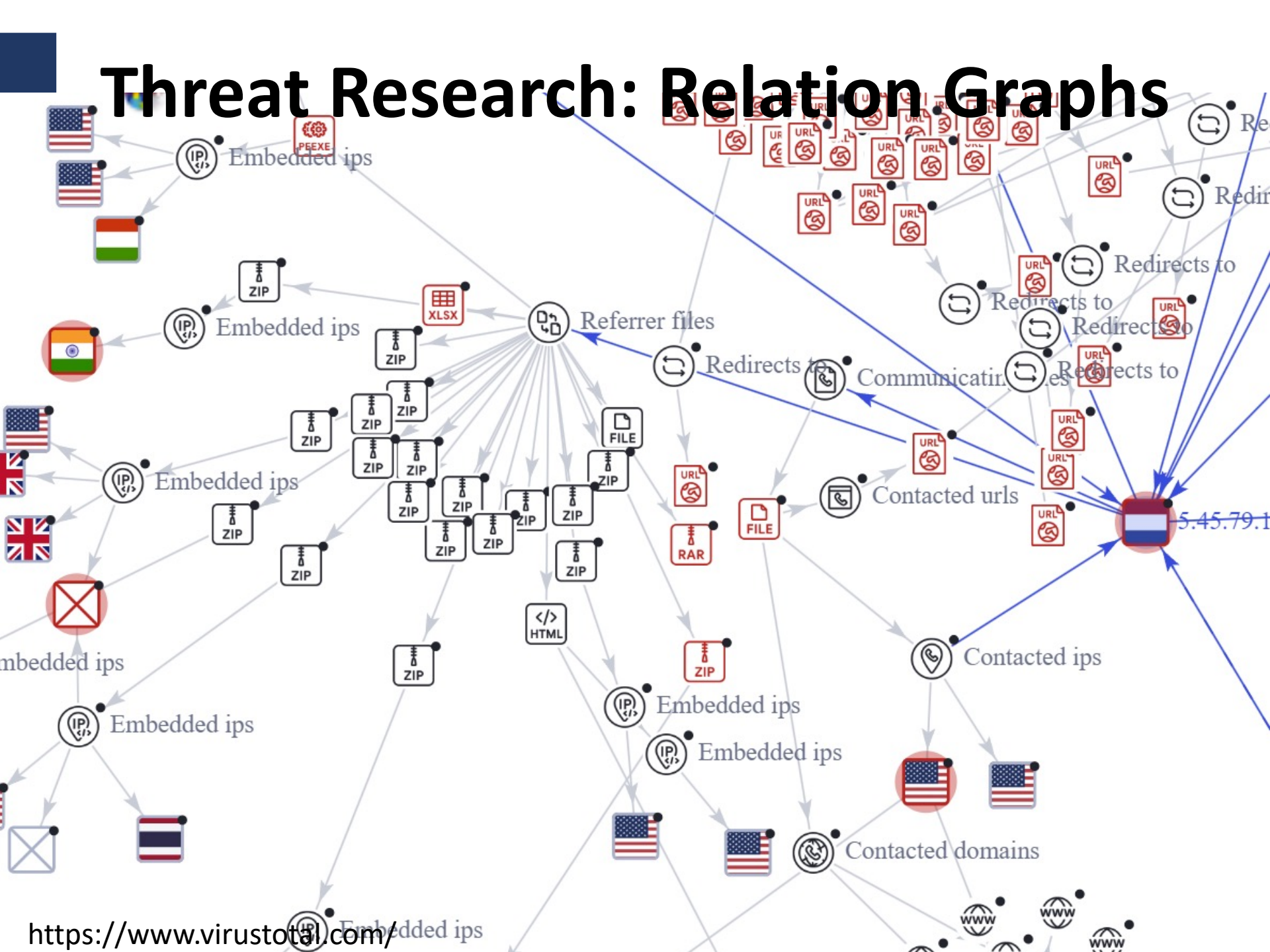
Threat Research: Relation Graphs



https://www.virustotal.com/

© Daab

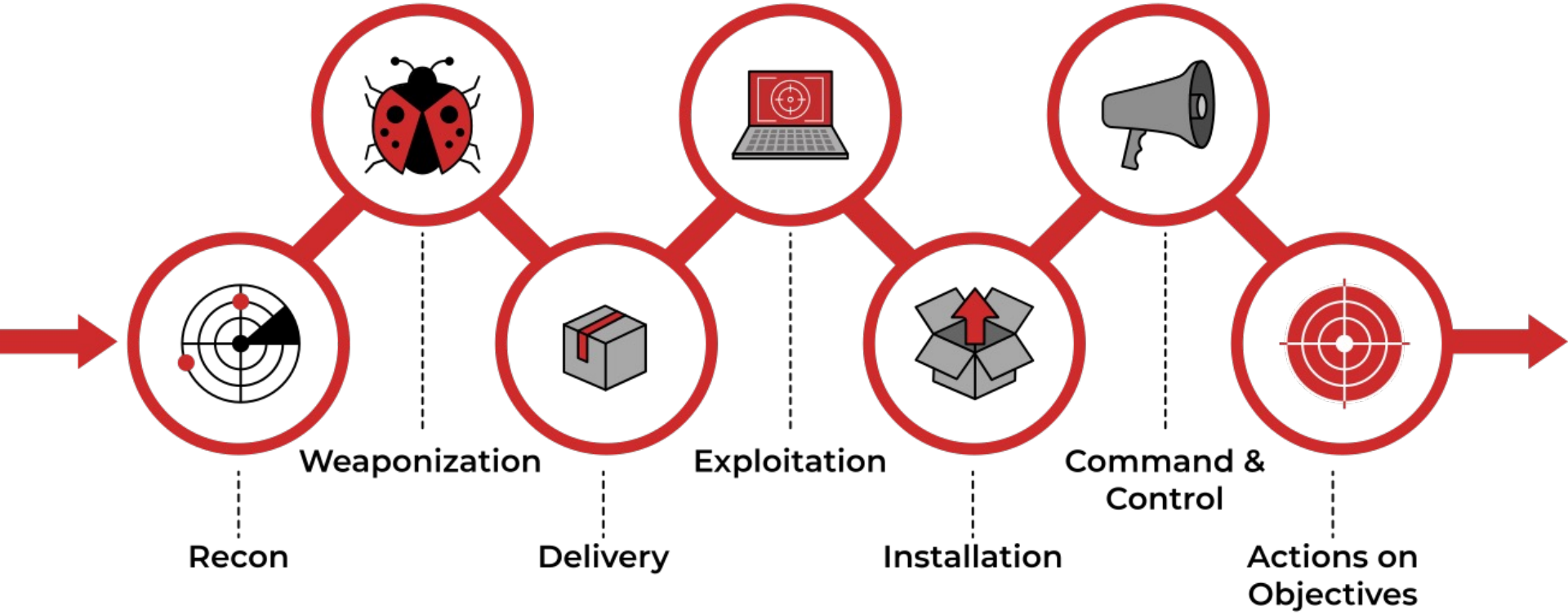
Threat Research: Relation Graphs



https://www.virustotal.com/

Think like a hacker

- **Lockheed Martin Cyber Kill Chain**
  - Helps understand and combat ransomware, security breaches, and advanced persistent attacks (APTs)

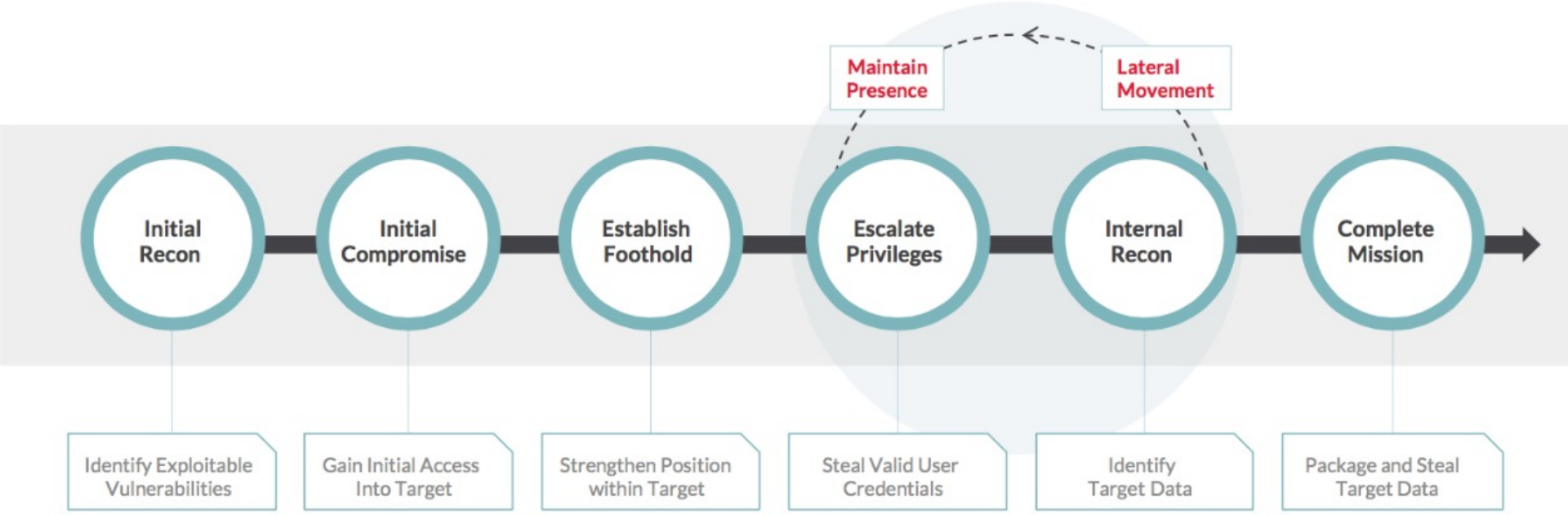


Think like a hacker

1. Enumerating employees, services, among others
2. Creating file with exploit included
3. Transferring exploit to victim
4. Asset exploited, unauthorized code run
5. Malicious code executes/installs
6. Remote control of asset
7. Exfil/destroy data, disrupt process, etc.

Think like a hacker

- Similar to the Lockheed Martin Cyber Kill Chain
- Emphasizes the iterative nature of compromise
- More literal steps

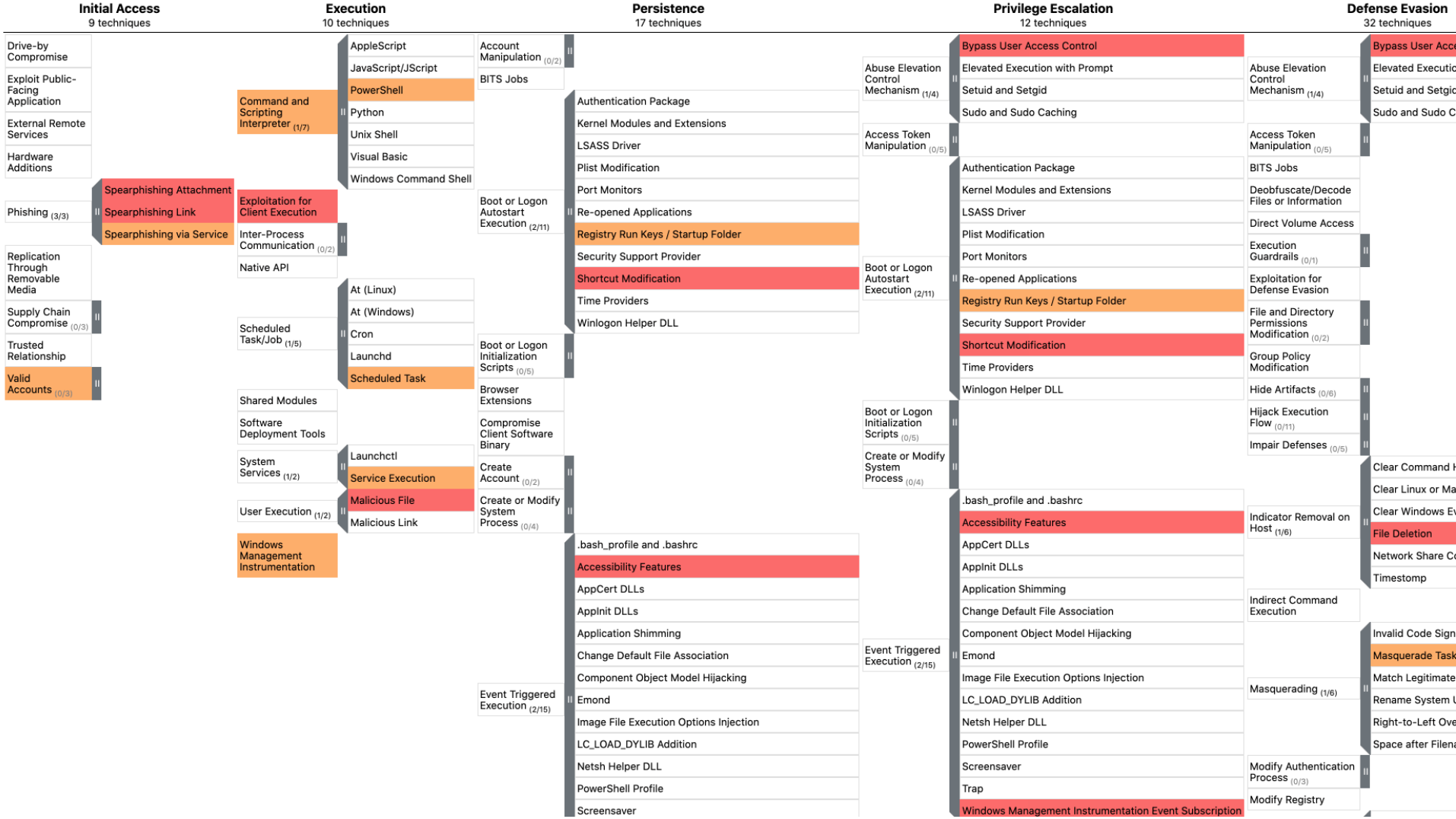


MITRE Att&ck Matrix

- **A globally-accessible knowledge base of adversary tactics and techniques**
  - based on real-world observations.

- **Allows organizations to map actions to a kill chain**
  - Also facilitates tracking the Actor or how it evolves
  - Actors will reuse tools, tactics and techniques

https://attack.mitre.org


Initial Access
9 techniques
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (3/3)
Spearphishing Attachment
Spearphishing Link
Spearphishing via Service
Replication Through Removable Media
Supply Chain Compromise (0/3)
Trusted Relationship
Valid Accounts (0/3)
Execution
10 techniques
Command and Scripting Interpreter (1/7)
AppleScript
JavaScript/JScript
PowerShell
Python
Unix Shell
Visual Basic
Windows Command Shell
Exploitation for Client Execution
Inter-Process Communication (0/2)
Native API
Scheduled Task/Job (1/5)
At (Linux)
At (Windows)
Cron
Launchd
Scheduled Task
Shared Modules
Software Deployment Tools
System Services (1/2)
Launchctl
Service Execution
User Execution (1/2)
Malicious File
Malicious Link
Windows Management Instrumentation
Persistence
17 techniques
Account Manipulation (0/2)
BITS Jobs
Boot or Logon Autostart Execution (2/11)
Authentication Package
Kernel Modules and Extensions
LSASS Driver
Plist Modification
Port Monitors
Re-opened Applications
Registry Run Keys / Startup Folder
Security Support Provider
Shortcut Modification
Time Providers
Winlogon Helper DLL
Boot or Logon Initialization Scripts (0/5)
Browser Extensions
Compromise Client Software Binary
Create Account (0/2)
Create or Modify System Process (0/4)
.bash_profile and .bashrc
Accessibility Features
AppCert DLLs
AppInit DLLs
Application Shimming
Change Default File Association
Component Object Model Hijacking
Event Triggered Execution (2/15)
Emond
Image File Execution Options Injection
LC_LOAD_DYLIB Addition
Netsh Helper DLL
PowerShell Profile
Screensaver
Privilege Escalation
12 techniques
Abuse Elevation Control Mechanism (1/4)
Bypass User Access Control
Elevated Execution with Prompt
Setuid and Setgid
Sudo and Sudo Caching
Access Token Manipulation (0/5)
Boot or Logon Autostart Execution (2/11)
Authentication Package
Kernel Modules and Extensions
LSASS Driver
Plist Modification
Port Monitors
Re-opened Applications
Registry Run Keys / Startup Folder
Security Support Provider
Shortcut Modification
Time Providers
Winlogon Helper DLL
Boot or Logon Initialization Scripts (0/5)
Create or Modify System Process (0/4)
.bash_profile and .bashrc
Accessibility Features
AppCert DLLs
AppInit DLLs
Application Shimming
Change Default File Association
Component Object Model Hijacking
Event Triggered Execution (2/15)
Emond
Image File Execution Options Injection
LC_LOAD_DYLIB Addition
Netsh Helper DLL
PowerShell Profile
Screensaver
Trap
Windows Management Instrumentation Event Subscription
Defense Evasion
32 techniques
Abuse Elevation Control Mechanism (1/4)
Bypass User Acce
Elevated Executio
Setuid and Setgic
Sudo and Sudo C
Access Token Manipulation (0/5)
BITS Jobs
Deobfuscate/Decode Files or Information
Direct Volume Access
Execution Guardrails (0/1)
Exploitation for Defense Evasion
File and Directory Permissions Modification (0/2)
Group Policy Modification
Hide Artifacts (0/6)
Hijack Execution Flow (0/11)
Impair Defenses (0/5)
Indicator Removal on Host (1/6)
Clear Command H
Clear Linux or Ma
Clear Windows Ev
File Deletion
Network Share Co
Timestomp
Indirect Command Execution
Masquerading (1/6)
Invalid Code Sign
Masquerade Task
Match Legitimate
Rename System U
Right-to-Left Ove
Space after Filena
Modify Authentication Process (0/3)
Modify Registry

Data exfiltration clues

- **High-volume of traffic**
  - DNS tunnelling
  - From unusual source
  - Long connection time to odd destination
- **Questionable compressed archive creation**
- **Multiple port firewall denies outbound from a single source**
- **URLs with long unexplained parameters**
- **DLP alerts**
  - Data Loss Prevention tools
- **UEBA alerts**
  - User Entity and Behaviour Analytics

Data destruction clues

- Compromise of patching servers
- Unusual file deletions
- Sudden system slowness or crashes
- Abnormal system and network behaviour

Attack identification

- **Domains, IP addresses or URLs**
  - Matching to APT
  - Domains unknown
    - To all OSINT sources
- **Email tailored to a specific person**
- **Suspicious information about business**
- **New executables**
  - Never been seen before anywhere
  - Customized attack files

Exploit alert triage

- **Prioritization**
  - How to do it? Which alerts we should prioritize?

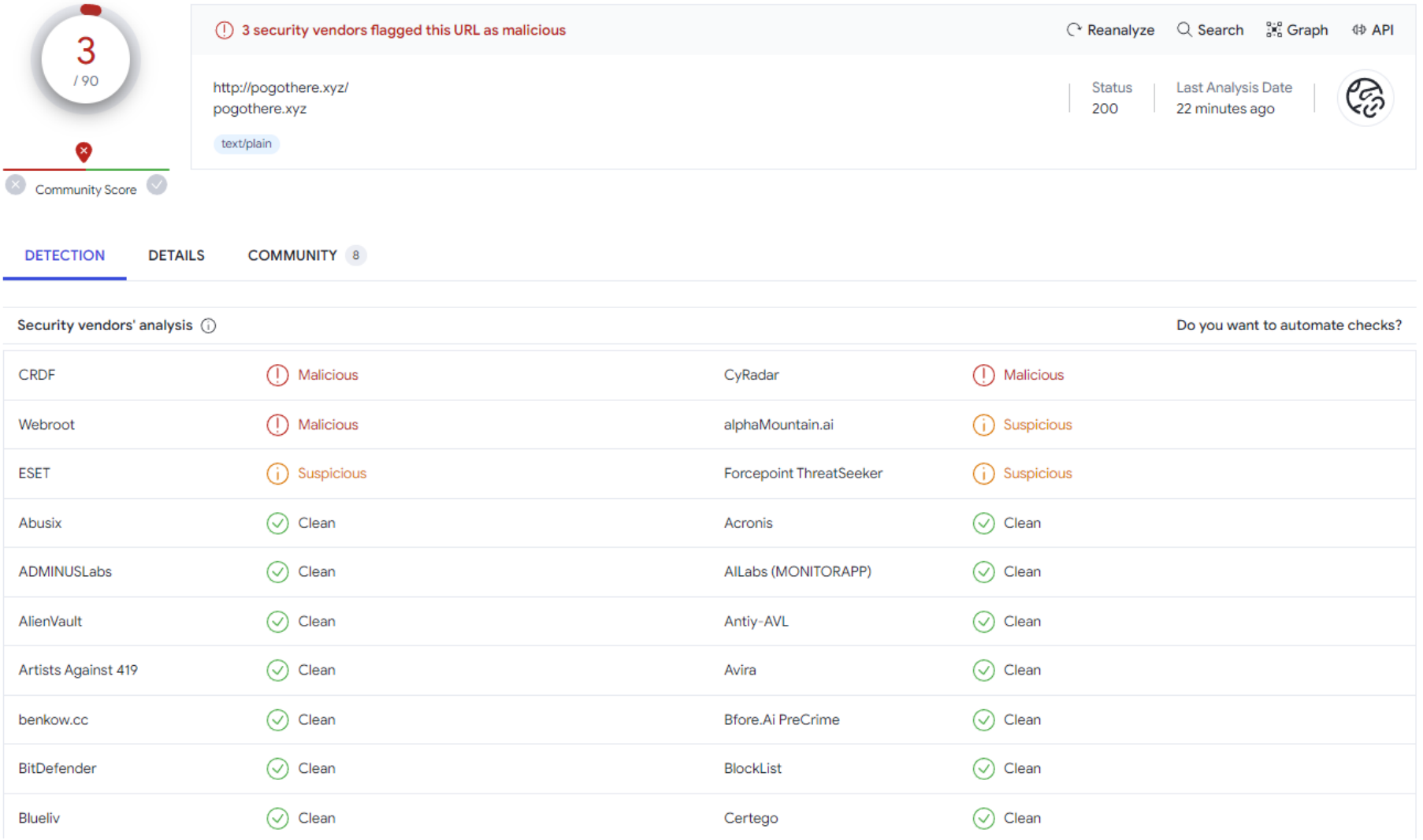
- **Ask yourself…**
  - What does the exploit do?
    - Give admin or user access? DoS?
  - Did the exploit work?
    - Is there evidence of install afterwards? Command and control?
  - What type of asset?
    - Internal? External? Desktop? Server?
    - Where is the asset located? DMZ? Sensitive server subnet?
    - Who is the user? Do they have admin access, critical data access?

Dissemination

- **When a threat is found information is disseminated**
  - Inside closed communities: MISP
  - To the public: Virustotal, AbuseCH, OTX, MISP…
- **Security software will include this information to protect organizations**
  - Current systems update signatures/rules dynamically
    - Several times per day
- **Golden rule: update!**

MISP: Global platform to share indicators of compromise

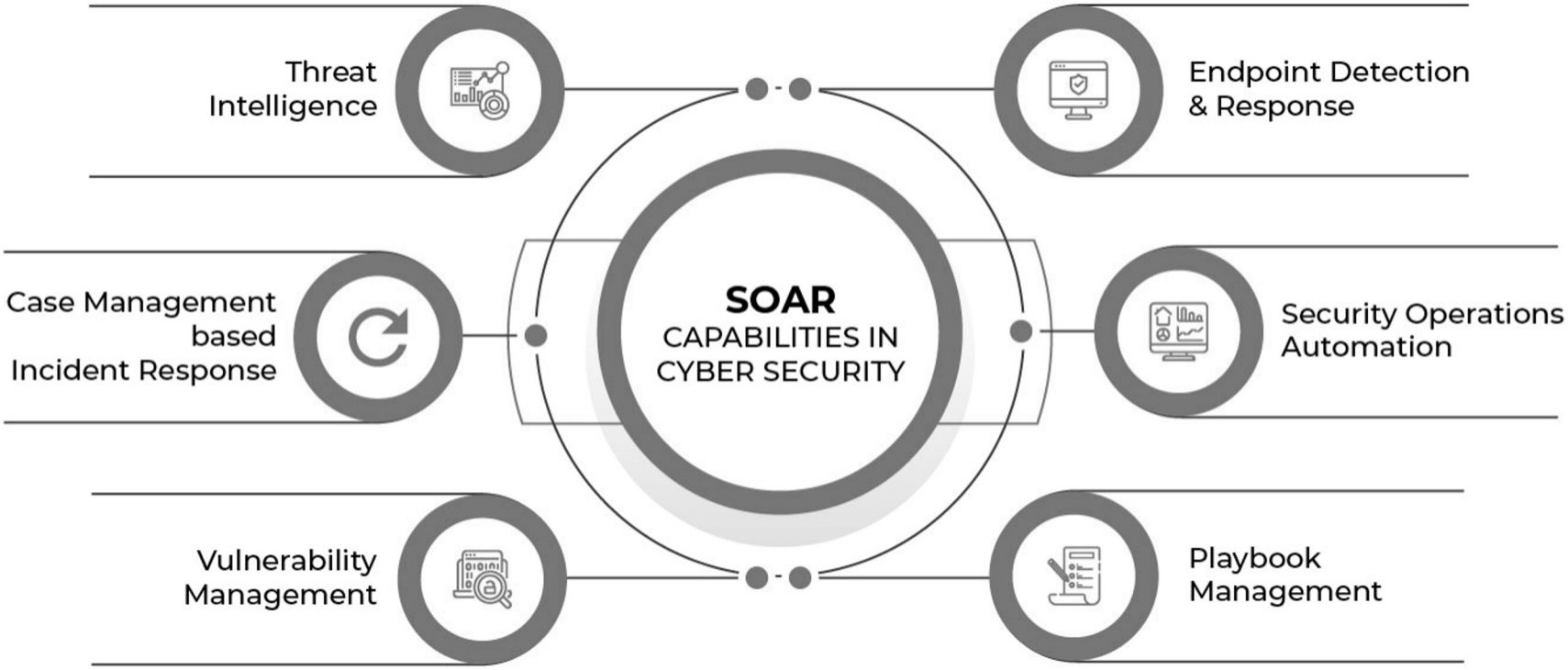
Is this site malicious?



https://www.virustotal.com/gui/url/9bfdba5d503dc46919a917a80885ebc442b1b88eb29d46898793c995abae5530

SOAR

- **Security Orchestration, Automation and Response**
  - Software that enables security teams to integrate and coordinate separated tools into streamlined threat response workflows



Firewalls

Objectives

Indispensable element in connecting a network domain

- Access control
- Flow control
- Content control

Centralized implementation of security policies

- Minimizes the impact of local vulnerabilities
  - Known or unknown
- Makes it easier to take more drastic positions
- Centralizes problem detection
  - and its treatment

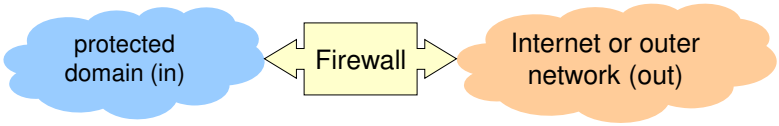
Definition (Cheswick & Bellovin)

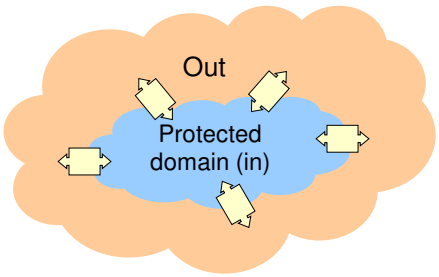
Link between networks
- of a protected perimeter (set of networks and machines)
- to an insecure network (Internet)

Component set
- Hardware and software

Properties
- In the path of all in ⇔ out traffic
- Controls the traffic passing through it
- Immune to penetration (by definition)

Definition (Cheswick & Bellovin)





Functionalities

Supervision of all in ⇔ out communication

- Control
  - The use of internal resources by external hosts/requests
  - The use of external resources by internal host/requests
- Defense from attacks
  - from outside the protected domain towards its resources
  - from the protected domain against external resources

Activation of gateway mechanisms

- To hide the structure from the protected perimeter
  - NAT (Network Address Translation)
  - Masquerading and Port Forwarding
- To extend the security perimeter
  - Secure tunneling (VPN)

Importance of Firewalls

Extreme!

Attacks on public systems are constant

- By specialized attackers
- By standalone applications

Systems do not always have adequate security mechanisms

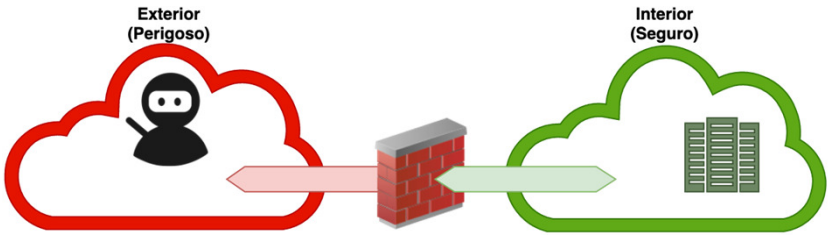
- Blocking after too many incorrect attempts
- Validation of communications
- Access control

Necessary to apply mechanisms defined by the administrator, in accordance with domain policies

- An application programmer is not aware of these

Firewalls servem como

Estrutura Genérica



Perimeter defense (of the domain)

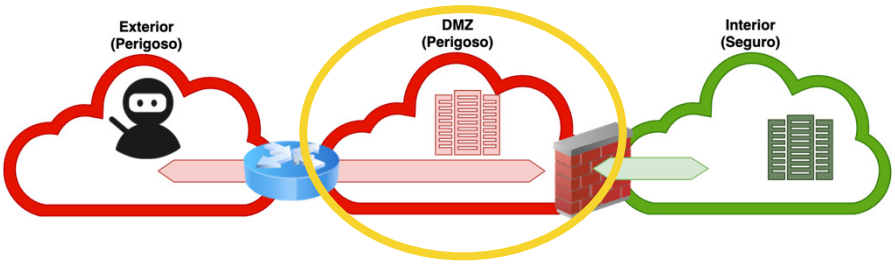
- Can be part of a defense in depth strategy

Consider an unsafe environment and a safe one

- Out: other domains or the Internet
- Inside: internal network

A single server: Bastion

Estrutura Genérica



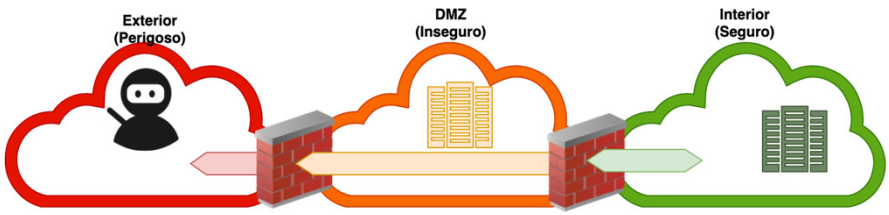
DMZ: DeMilitarized Network or Perimeter Network

- Insecure network
- Contains servers exposed to the world
- Sometimes necessary to use specific services/applications

É uma rede configurada de forma a permitir que alguns serviços sejam acessados a partir da Internet, enquanto mantém outros serviços e recursos protegidos na rede interna.

Fornecee uma camada adicional de proteção contra ameaças externas, como ataques cibernéticos. Isso é feito, em parte, por meio da implementação de firewalls

Estrutura Genérica



DMZ may have some protection

- System of two Firewalls with different rules

External firewall: quite permissive

- Control access to all networks

Internal firewall: more restricted

- Control access to the internal network

Types: packet filters

Reject unauthorized interactions based on the content of IP datagrams

- IP addresses (source and/or destination)
- IP/transport header options
- Transport protocols and ports (origin and/or destination)
- Directions for creating virtual circuits
- Data sent via transport protocol
- Datagram size

Can analyze flow behavior

- Example: detect port scans (with nmap)

Typically supported by core OS components

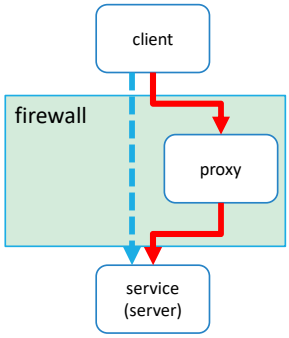
- Example: iptables, ipfw, pf

Types: applicational gateways

Control interactions at the application level
- But transparent to interacting applications
- There is usually a different firewall per protocol
  - proxy protocol

Client -> Proxy -> service (server)
- Proxies are servers

Aspects of operating a proxy
- User access control
- Analysis and modification of content
- Detailed logging
- Impersonation (proxying)
  - Transparent replacement of one of the interlocutors



Proxies podem controlar o acesso dos usuários, analisar e modificar o conteúdo das comunicações, manter registos detalhados e até mesmo se fazer passar por um dos interlocutores.

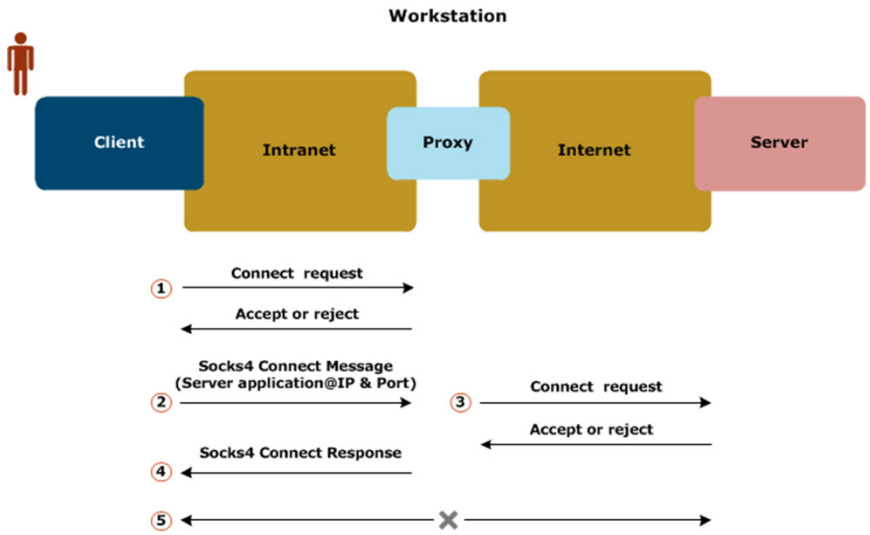
Types: circuit gateways

Kind of application gateway
- Contacted directly by customers

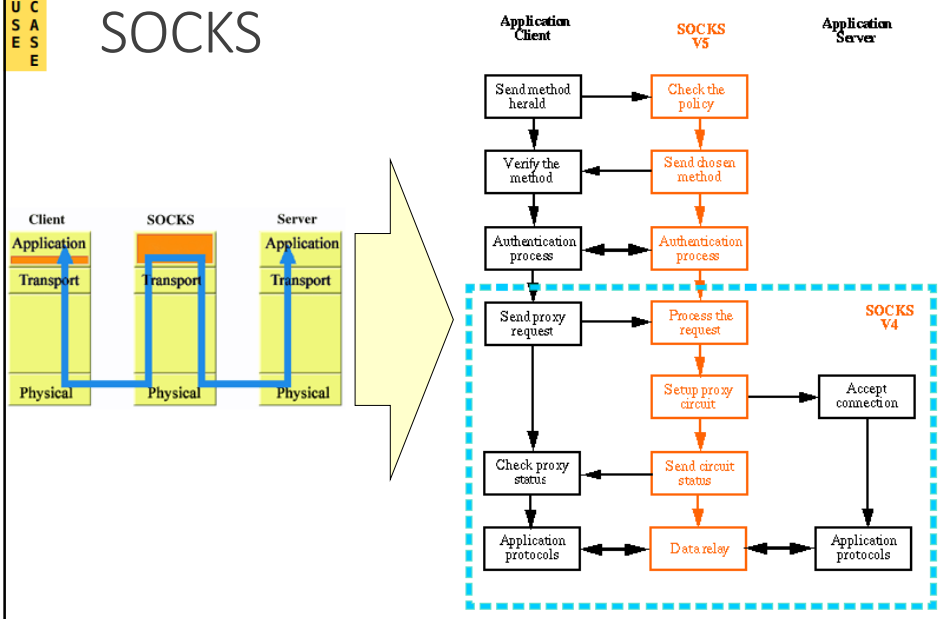
Non-transparent interposition
- For deploying specific authentication and authorization policies and mechanisms

Typically requires changing client applications
- Examples: SOCKS and HTTP Proxy

Types: SOCKS4 circuit gateways



SOCKS



Types: stateful packet filters

Dynamic (or context-sensitive) packet filter

- Sort of packet filter with historical context
- Context is key to certain decisions
- Common term: Stateful Packet Filter/Inspection (SPI)

Context examples:

- Decisions made for IP packet fragments
  - Defragmentation before filtering
- Established TCP virtual circuits
  - Circuit establishment requests are controlled
  - Established virtual circuits are allowed

São um tipo de firewall que vai além da filtragem simples de pacotes de rede.

Eles são capazes de tomar decisões com base em informações contextuais e históricas sobre o tráfego de rede.

Types: stateful packet filters

Context examples (cont.):

- Dynamic NAT tables
  - Creation of entries depending on observed traffic
- Request/response interactions over UDP
  - Dynamic authorization of responses to authorized requests
  - Example: DNS name resolution
- ICMP error messages
  - Related to previously sent TCP/UDP packets
- Identification of application protocols from data flows
  - To handle flows that use dynamic or “stolen” ports
  - Examples: FTP, RPC protocols, P2P protocols
  - Utility: filtering, transparent proxying, QoS

Bastion

Must run secure versions of operating systems

- With a secure configuration
- Only essential services are installed
- Telnet, DNS, FTP, SMTP and authentication proxies

Public servers should not perform in a bastion

- Examples: DNS, SMTP, HTTP, FTP, SSH, RAS, etc.
- Must run on isolated machines within DMZs
  - Preferably one per service
- Bastion only forwards traffic to the appropriate machines on a DMZ
  - And allows limited traffic from the DMZ

Bastion

It is often a platform for application gateways

- But the more proxies there are in the bastion, the lower its performance will be
- Proxies can run on specific machines
  - Security appliances
- Bastion only forwards traffic to and from the appliances

Secure execution of application gateways

- Independence
  - The compromise of one does not affect the rest
- No special privileges
  - Their compromise does not allow to affect the host

Servem como pontos de entrada seguros e protege a rede interna contra ameaças externas.

Eles são executados em sistemas operativos seguros e são cuidadosamente configurados para garantir que apenas o tráfego autorizado passe por eles.

Além disso, eles podem atuar como plataformas para aplicação de gateways, desde que se mantenha um equilíbrio entre segurança e desempenho.

Topology: Dual-homed (w/ or w/o DMZ)

DeMilitarized Network

Architecture

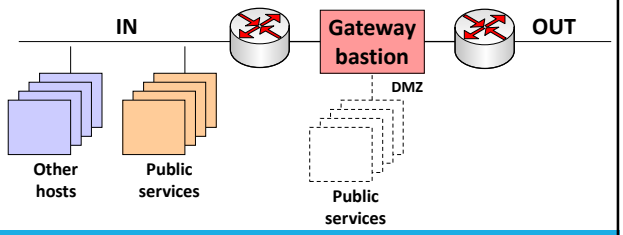
- A single machine
- Gateway bastion
- A pair of additional routers
  - To isolate the bastion of direct addressing
- Internal and public servers

Problems

- Bastion compromise disables firewall
- The firewall processing load is all on the bastion
- Public services are within the protected network

Benefits

- Simplicity
- Resource savings



Security services

Authorization

- From data streams (packet filters)
  - Transport or network level
- Users (application gateways / circuits)

Traffic Redirection

- For dedicated hosts
  - Local services (e.g., mail, www, ftp, etc.)
  - Proxies in security appliances
- Proxying
  - Explicit (e.g., circuit gateways)
  - Transparent (e.g., NAT address translations)

Security services

Application content processing

- Content analysis
  - Example: virus detection
- Changing high-level protocols
  - Example: virus removal

Secure communication

- Virtual Private Networks (VPNs)
  - Encryption and integrity control of data flows over public (insecure)
- Tunneling
  - IP domain extension to distant nodes
  - ex., PPTP, L2TP, IPSec

Security services

Defense against DoS attempts

- Attack detection
  - Abnormal traffic volumes, high volume, etc...
  - Filtering dangerous or malformed datagrams
    - ex. Land attack, Ping-of-Death
  - Activation of palliative measures
    - ex. SYN flooding relay/semi-gateway

Defense against information leaks

- Abnormal traffic detection
- Controlling behavior against known models

Limitations

They do not solve the problem of attackers within the internal network

- Unless the internal network is segmented into multiple subnets
- Switches typically do not support firewall operations
- VLANs provide minimal segregation (DMZ type)

Efficiency of control of all external connections

- Which can be done in parallel in countless ways:
  - PSTN & modems
  - Unregistered WLANs & Aps

Lack of control over camouflaged/hidden interactions

- Camouflaged interactions multiplexed by VPNs
- IP tunnels over HTTP, ICMP, DNS, etc.

Difficult to manage in environments with heterogeneous interests

- Universities, ISPs

Personal Firewalls

Adopted for the protection of individual / personal hosts

- Defense in depth vs. perimeter defense

Owners can set additional control policies

- Applications authorized to access the network
- The protocols that applications can use
- The hosts/networks that protocols/applications can interact with

Reduce the risk of compromise between hosts on a network

- Allows a machine to protect itself independently of the protection provided by its network
  - Do not make assumptions regarding other network protections
- Useful for machines that migrate between networks

Personal firewalls: issues

Normal users are not network security experts

- They don't normally understand how IP networks work
  - IP addresses, transport ports, transport protocols, etc.
- They do not know how to assess whether a given interaction is normal, acceptable, etc.
- They don't know the basic security policies they should apply

Blocking suspicious interactions may nullify functionality

- Network communication is currently commonplace
- Applications do not inform users of their communication needs

Personal firewalls: issues

Operational complexity

- Different operating environments → different policies
- Different network interfaces → different policies

The combination of operational scenarios, network interfaces and acceptable interactions for each case leads to a huge number of rules

- Confusion, incoherence → difficult to detect vulnerabilities

iptables

É uma ferramenta de filtragem de pacotes integrada com o kernel TCP/IP do Linux. Ele é amplamente usado para controlar o tráfego de rede, aplicar regras de segurança e encaminhar pacotes em sistemas Linux

Packet filter (with context, or stateful)

- Integrated with Linux kernel TCP/IP
- Can be extended in several ways
  - New core modules
  - User mode applications

5 chains

- INPUT, OUTPUT, FORWARD
- PREROUTING, POSTROUTING

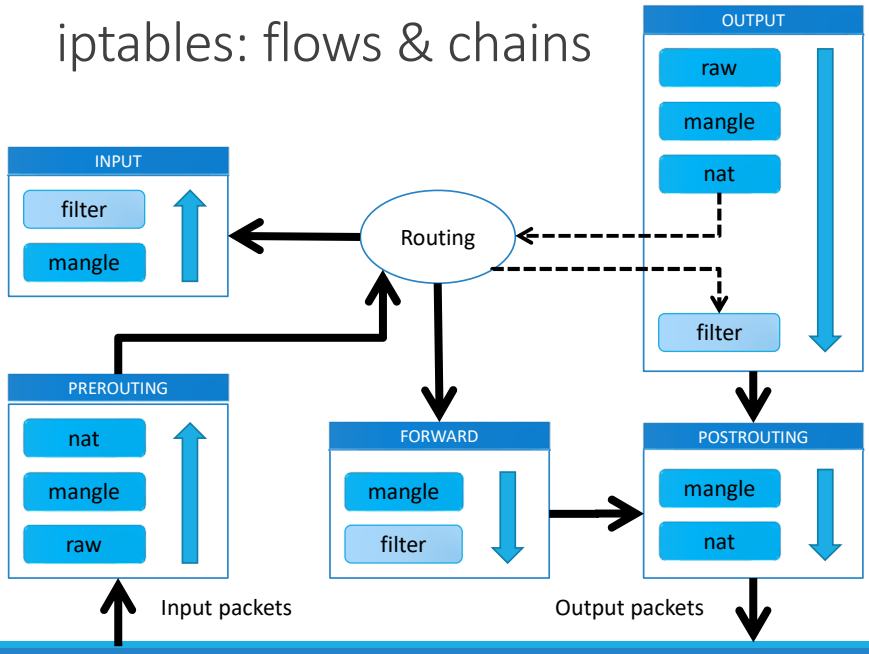
4 tables (per chain, but not for all)

- raw, mangle, nat, filter

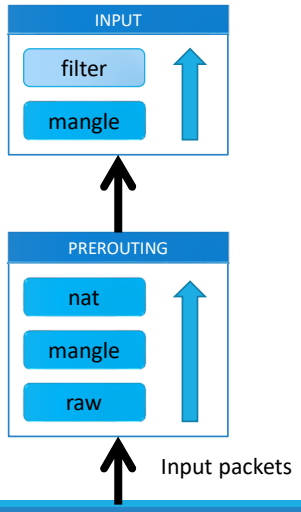
Various extra modules

- e.g., CONNTRACK (connection tracker, or flow follower)

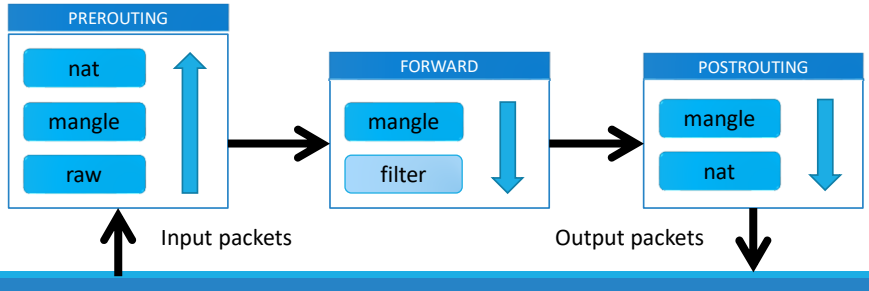
iptables: flows & chains



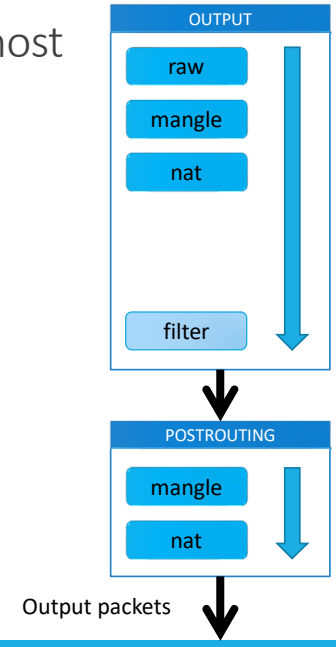
iptables: traffic for the host



iptables: routed traffic



iptables: traffic from the host



iptables: decisions

Basic decisions
- ACCEPT
  - Let the package continue
- DROP
  - Discard the package
- CONTINUES
  - Use decisions from other rules

Reusable Decisions
- New chains
- Jump to a new chain
  - The name of the chain is the decision
- RETURN
  - Leave the current chain

Other decisions
- LOG
- MARK
  - With internal label
  - Useful for making coherent decisions across different chains
- REJECT
  - Rejection with error message
- SNAT, MASQUERADE
  - Source NAT (masquerading)
- DNAT, REDIRECT
  - Destination NAT (port forwarding)
- Actions by applications
  - QUEUE

Iptables exploitation: fail2ban

Agent that observes records, comparing them with patterns

- Can prevent some DoS, brute force attacks (SSH), scans
- Reactive: Does not prevent the attack from starting
  - May not prevent attacks with few interactions
- Can be used with any service that creates records

Jail: a context composed of several rules

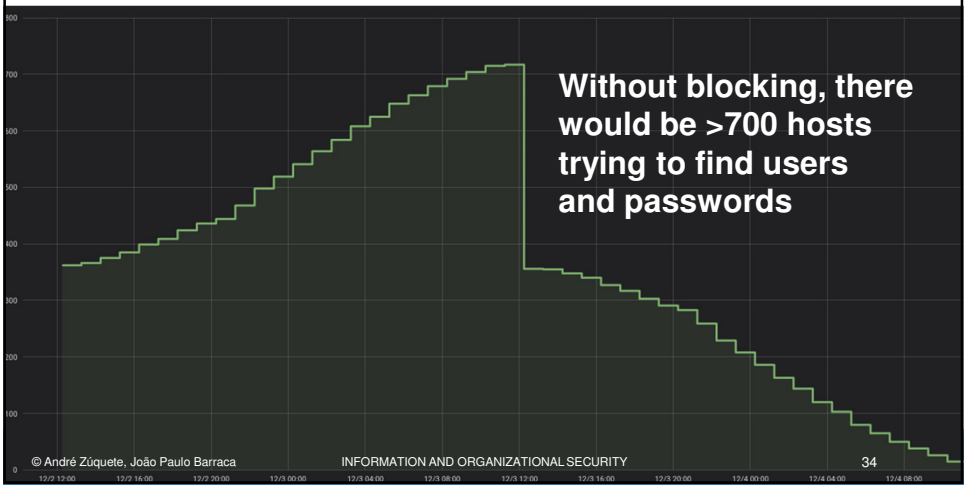
- Defines what to observe and what action to take
- Action: Implement a specific response
  - example: block communications on the firewall
  - Can use a local or remote firewall

Filter: a set of regexps that signal anomalous behavior

- Composed of expressions to consider and ignore (white list)

Iptables exploitation: fail2ban

"Anonymous" server, without content
# of IPs blocked due to SSH access attempt



Modern Cryptography

Symmetric ciphers

Cryptography: terminology (1/2)

Cryptography

- Art or science of hidden writing (confidential writing)
  - from Gr. kryptós, hidden + graph, r. de graphein, to write
- Initially used to maintain confidentiality of information

- **Steganography**: art of concealing data
  - from Gr. steganós, hidden + graph, r. de graphein, to write

Cryptanalysis

- Art or science of breaking cryptographic systems or encrypted information

Cryptology

- Cryptography + cryptanalysis

Cryptography: terminology (2/2)

Cipher

- Specific cryptographic technique

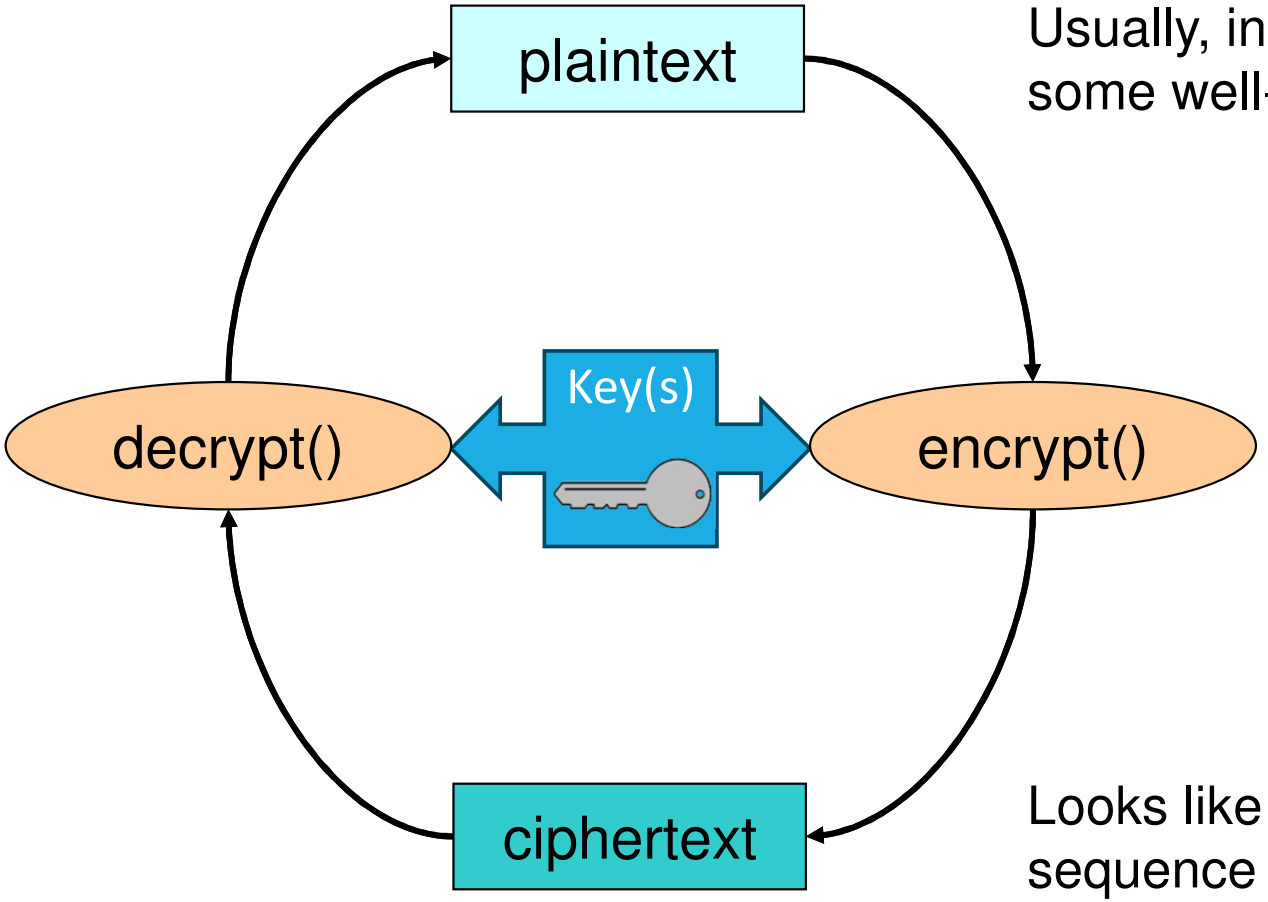
Cipher operation

- **Encryption**: original information → cryptogram
- **Decryption**: cryptogram → original information

- Original information aka plaintext or cleartext
- Cryptogram aka ciphertext

- **Algorithm**: way of transforming data
- **Key**: algorithm parameter
  - Influences algorithm execution

Operations of a Cipher



Usually, information that follows some well-know format

Looks like a random sequence of symbols

Use cases (symmetric ciphers)

Self protection with key K

- Alice encrypts plaintext **P** with key **K** → Alice: $C = \{P\}_k$
- Alice decrypts ciphertext **C** with key **K** → Alice: $P' = \{C\}_k$
- **P'** should be equal to **P** (requires checking)
- Only Alice needs to know **K**

Secure communication with key K

- Alice encrypts plaintext **P** with key **K** → Alice: $C = \{P\}_k$
- Bob decrypts cyphertext **C** with key **K** → Bob: $P' = \{C\}_k$
- **P'** should be equal to **P** (requires checking)
- **K** needs to be known by Alice & Bob

Cryptanalysis: goals

Discover original plaintext

- Which originated a given ciphertext

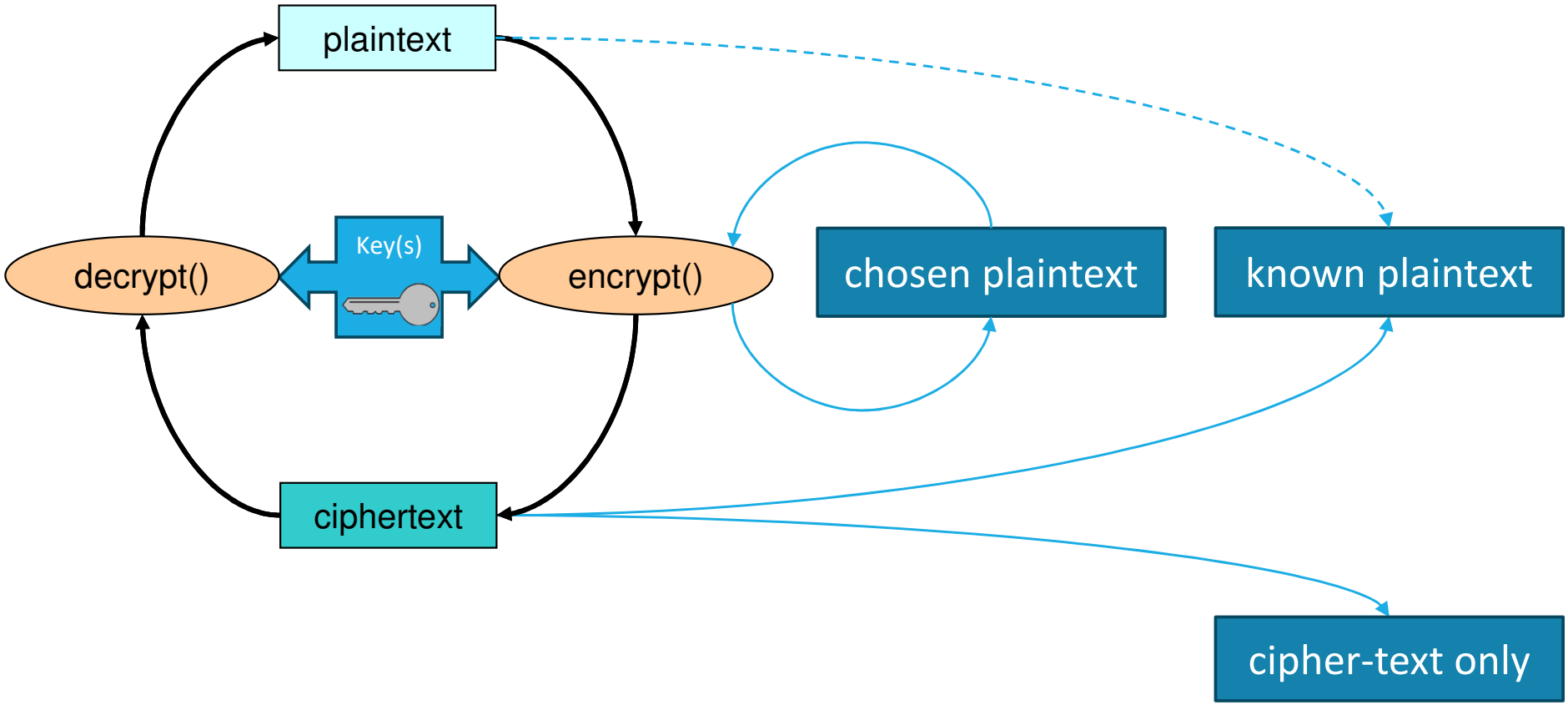
Discover a cipher key

- Allows the decryption of ciphertexts created with the same key

Discover the cipher algorithm

- Or an equivalent algorithm
- Usually algorithms are not secret, but there are exceptions
  - Lorenz, A5 (GSM), RC4 (WEP) , Crypto-1 (Mifare)
  - Algorithms for DRM (Digital Rights Management)
- Using reverse engineering

Cryptanalysis attacks: Approaches



Cryptanalysis attacks: Approaches

Brute force

- Exhaustive search of the key space until finding a match
- Usually unfeasible for a large key space
  - e.g., 128 bits keys have a search space of 2^{128} values
- Randomness is fundamental!

Clever attacks

- Reduce the search space to a smaller set of potential candidates: words, numbers, restricted size or alphabet
- Identify patterns in different operations, etc.

Computer ciphers

Operate by making substitutions

- Original information is a sequence of **symbols**
- Each symbol is replaced by a **substitution symbol**
  - Usually with the same size
  - Polyphonic substitution: several, larger substitution symbols for each original symbol
- Substitution symbols are picked from a **substitution alphabet**

Usual symbols

- Bit
- Block of bits

Strategies

- Monoalphabetic substitution: key → one substitution alphabet
- Polyalphabetic substitution: key → several substitution alphabets

Computer ciphers: stream ciphers

Encrypt/decrypt by mixing streams

- They consider the data to cipher or decipher as a bit stream
- Each plaintext/ciphertext bit is XORed (⊕) with each keystream bit

  plaintext ⊕ keystream → ciphertext

  ciphertext ⊕ keystream → plaintext

Are polyalphabetic ciphers

- Usually explored in low-level communication protocols

Keystream

- Randomly produced, as long as the processed data
  - Vernam cipher (or one-time pad)
  - The only perfect cipher
  - Rarely used
- Pseudo-randomly produced from a limited key
  - Ordinary stream ciphers

Computer ciphers: block ciphers

Encrypt/decrypt sequences of blocks

- Symbols <=> fixed-length blocks of bits
- Usually use byte blocks as symbols

Are monoalphabetic ciphers

- Some may be polyphonic ciphers

No caso da CBC

Computer ciphers: symmetric

Encrypt/decrypt with the same key

- The oldest strategy

Computer ciphers: asymmetric

Encrypt/decrypt with the different, related keys

- Key pair
  - Private component, public component
- An approach that was first proposed in 1978

Computer ciphers: combinations

(Symmetric) stream ciphers

- Polyalphabetic ciphers
- Keystream defined by the key
- Keystream and XOR implement a polyalphabetic transformation

Symmetric block ciphers

- Monoalphabetic ciphers
- Substitution alphabet is defined by the key

Asymmetric (block) ciphers

- Polyphonic ciphers
  - Not by nature, but for security reasons
- The functionalities of these ciphers are not homogeneous

Techniques used by ciphers

Confusion

- Complex relationship between the key, plaintext and the ciphertext
  - Output bits (ciphertext) should depend on the input bits (plaintext + key) in a very complex way

Diffusion

- Plaintext statistics are dissipated in the ciphertext
  - If one plaintext bit toggles, then the ciphertext changes substantially, in an unpredictable or pseudorandom manner
- Avalanche effect

(Symmetric) stream ciphers: Examples

A5/1, A5/2

- Cellular communications
- Initially secret, reverse engineered
- Explored in a weak fashion (64-bit keys w/ 10 bits stuck at zero)

E0

- Bluetooth communications
- Keys up to 128 bits

RC4

- Wi-Fi communications (WEP, deprecated)
- Initially secret, reverse engineered, never officially published
- Keys with 40 to 2048 bits

Other

- Salsa20, Chacha20, etc.

(Symmetric) stream ciphers: Approach

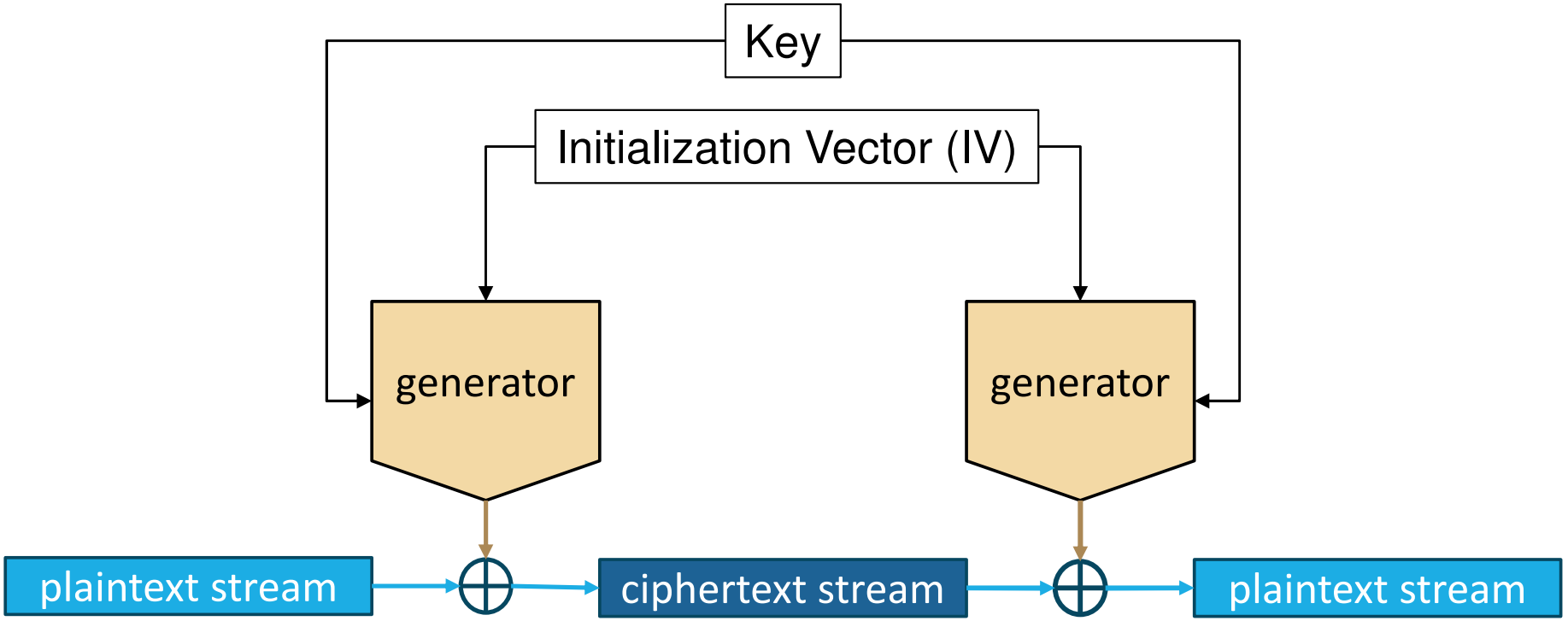
Use a cryptographically secure, pseudo-random bit generator

- This generator produces the keystream
- The generator implements a state machine
- The generator is controlled by two values:
  - **Initialization Vector** (defines the initial state of the state machine)
  - **Key** (defines how one state moves to the next to produce the keystream)

Cryptographically secure, pseudo-random means:

- Statistically, the keystream looks like a totally random sequence of zeros and ones
- If an attacker learns a part of the keystream, it cannot infer:
  - Past keystream values
  - Future keystream values

(Symmetric) stream ciphers: Approach



(Symmetric) stream ciphers: Exploitation considerations

No two messages should be encrypted with the same key and IV

- Because they will be encrypted with the same keystream
- The knowledge about um message reveals the other

$C1 = P1 \oplus KS$

$C2 = P2 \oplus KS$ $\rightarrow$ $P2 = C2 \oplus KS = C2 \oplus C1 \oplus P1$

- Knowledge about P1 => immediate knowledge about P2
- Known/chosen –plaintext attacks become very effective!

Keystreams may be periodic (have a cycle)

- Depends on the generator
- Same problem as the one above
- Plaintext should be shorter than the period length

(Symmetric) stream ciphers: Exploitation considerations

Ciphertexts can be deterministically manipulated

- Each cipher bit depends only on one plaintext bit

  $C' = C \oplus \Delta \quad \rightarrow \quad P' = P \oplus \Delta$

- It is fundamental to have integrity control elements
  - In the ciphertext
  - In the plaintext

Symmetric Block ciphers: Examples

DES (Data Encryption Standard)

- Proposed in 1974, standard in 1977
- Input/output: 64-bit blocks
- Key: 56 bits

AES (Advanced Encryption Standard)

- Proposed in 1998 (Rijndael), standard in 2001
- Input/output: 128-bit blocks
- Key: 128, 192 or 256 bits

Other

- IDEA, CAST, Twofish, Blowfish, RC5, RC6, Kasumi, etc.

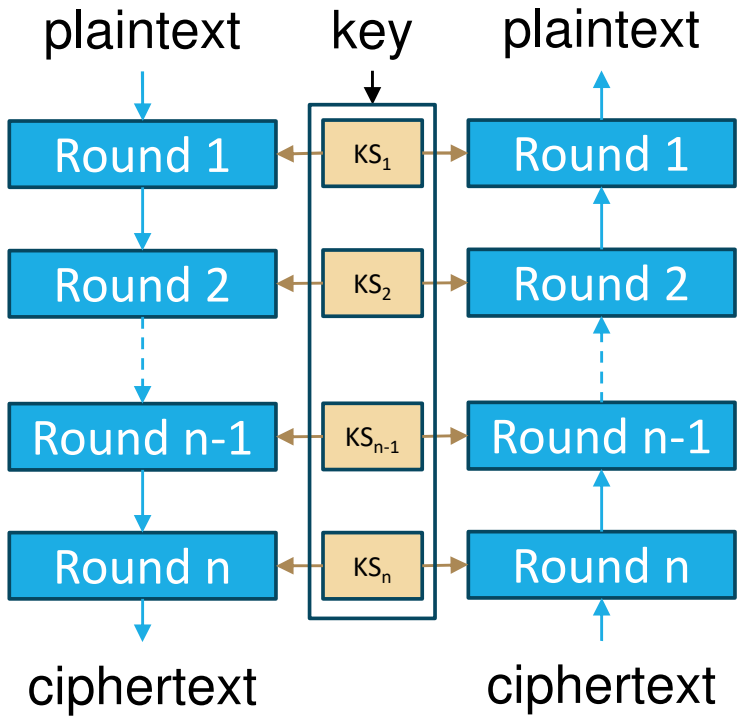
Symmetric block ciphers: Approach

Use a pipeline of transformation rounds

- Each round adds confusion and diffusion
- Each round is usually controlled by a subkey
  - Key schedule
  - A key derived from the key used provided for encryption/decryption

Rounds need to be reversible

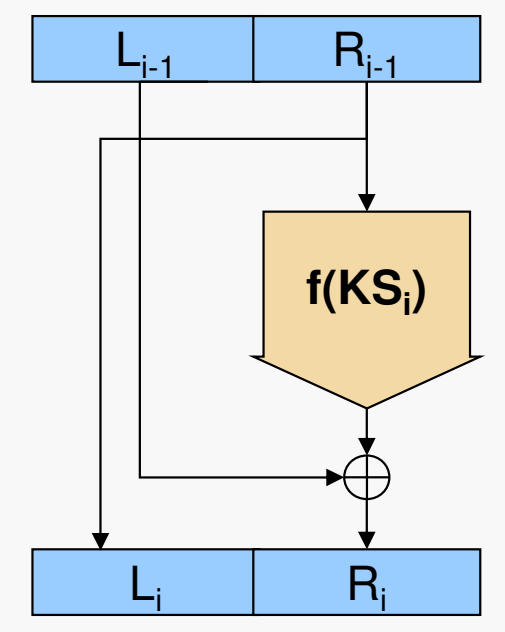
- To allow decrypting what was encrypted
- Feistel networks
- Substitution-permutation networks



Feistel networks

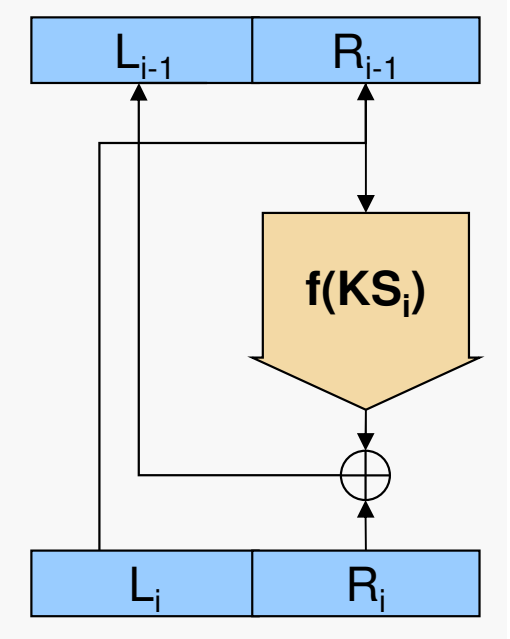
$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$



$R_{i-1} = L_i$

$L_{i-1} = R_i \oplus f(L_i, K_i)$



The function $f(KS_i)$ doesn't need to be reversible!

Substitution-Permutation Network

SBox

- Table with an output for an input (index)

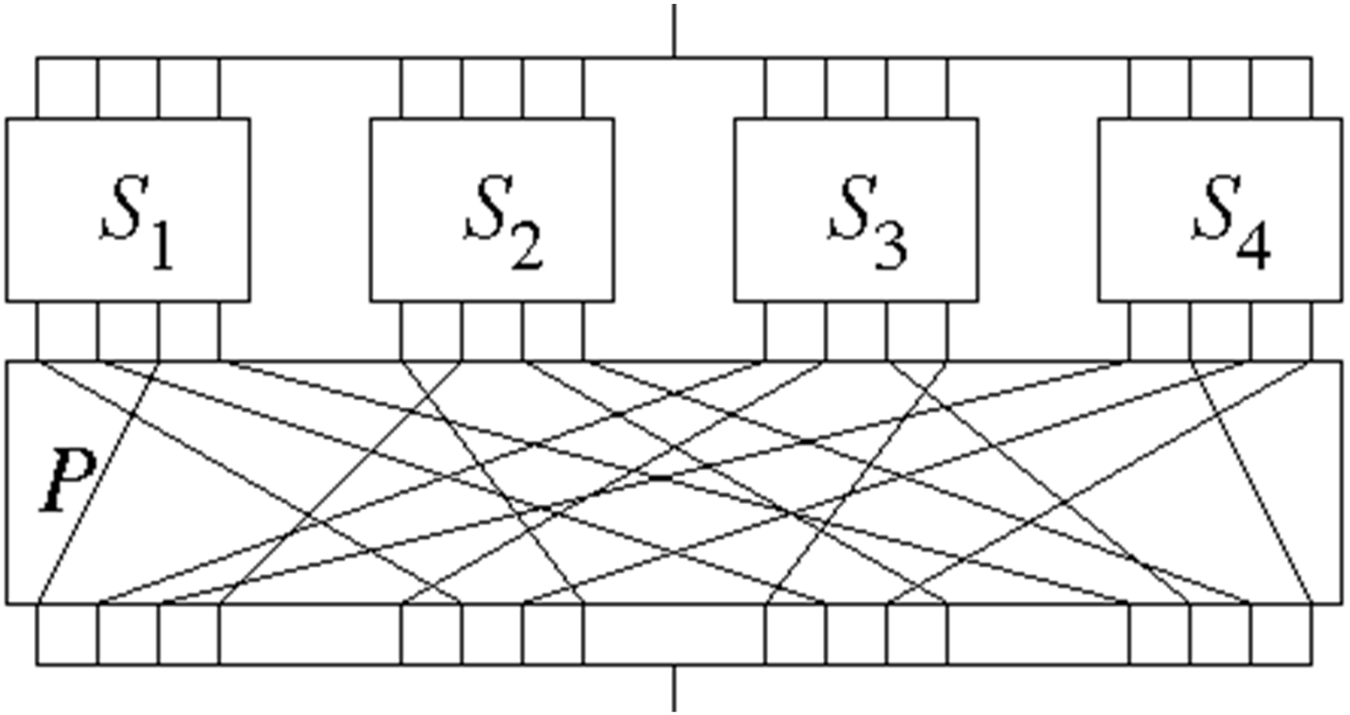
  output = SBox[input]

- SBoxes may be constant or key-dependent
  - DES and AES use constant Sboxes
  - Blowfish and Twofish use variable, key-dependent SBoxes
- In SP networks, SBoxes must be reversible
  - Bijective transformations
  - $y = SBox[x]$ $x = SBox^{-1}[y]$

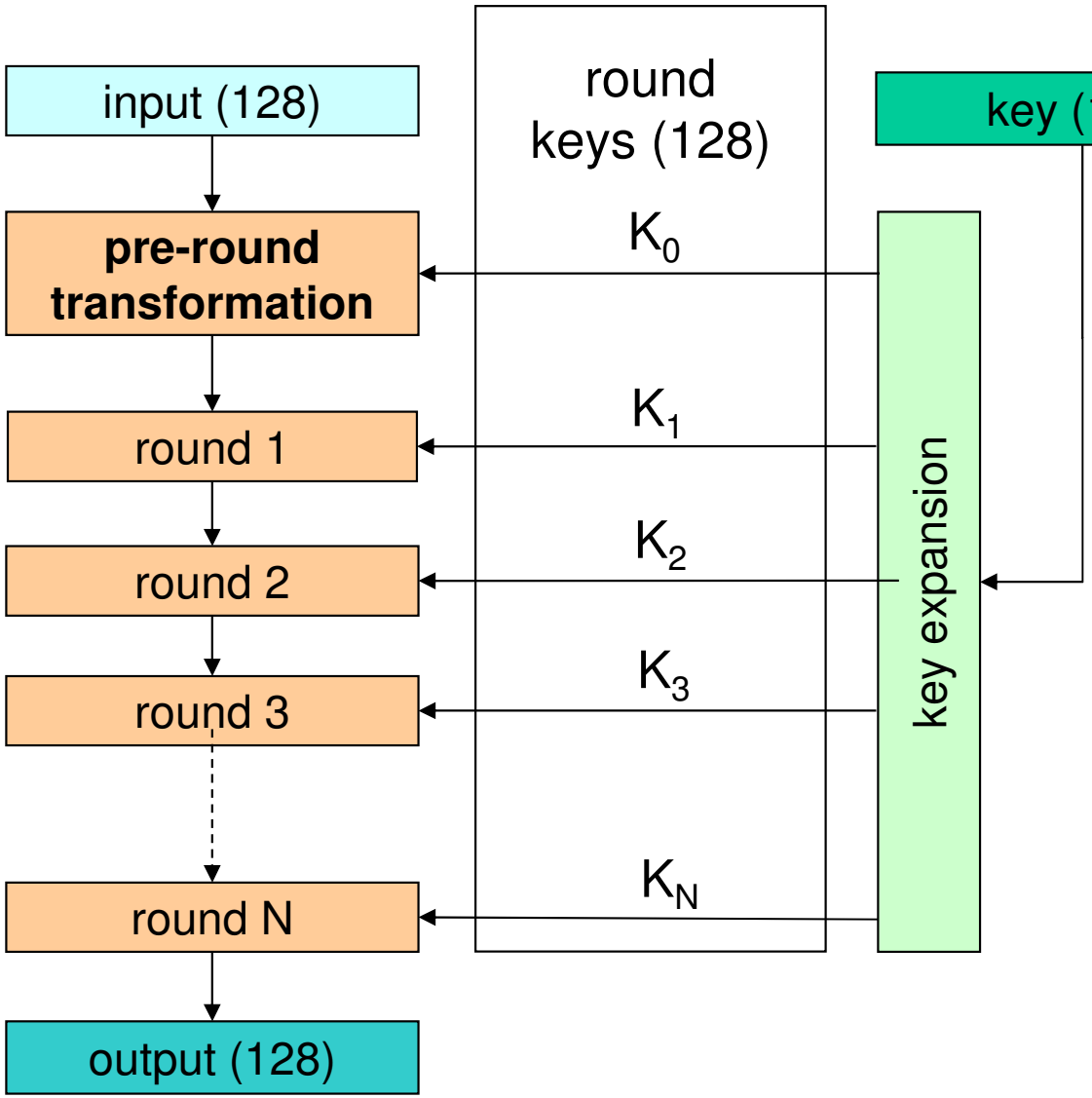
PBox

- Changes the positions of the input bits

Substitution-Permutation Network

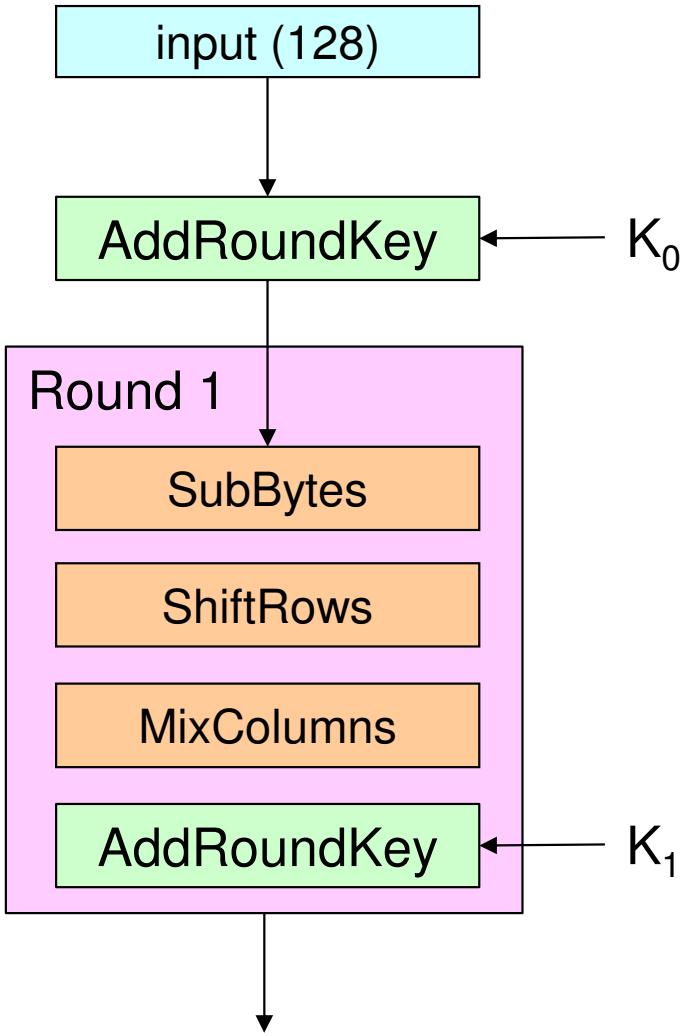


AES architecture



key size	rounds (N)
128	10
192	12
256	14

AES (encryption) round



AddRoundKey

- 128-bit XOR
- Output is a 4x4 byte matrix

SubBytes

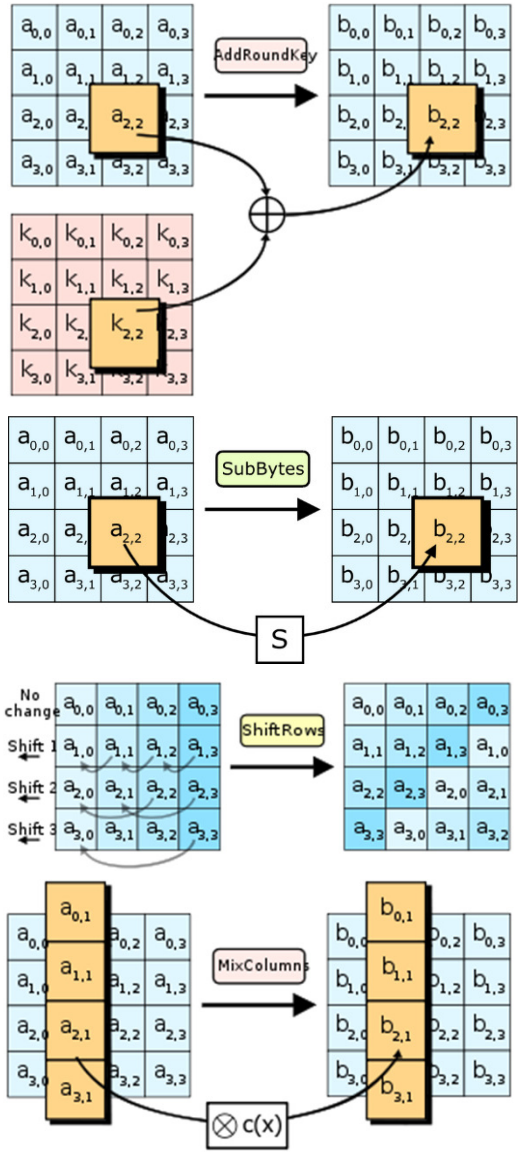
- 256-element S-box
- Each matrix bytes is substituted

ShiftRows

- Rows are rotated left
- Byte shifts vary (0, 1, 2 & 3)

MixColumns

- Each column is transformed
- Not performed in the last round



https://aescryptography.blogspot.com

AES in CPU instruction sets

Intel AES New Instructions (AES-NI)

AESENC	Perform one round of an AES encryption flow
AESENCLAST	Perform the last round of an AES encryption flow
AESDEC	Perform one round of an AES decryption flow
AESDECLAST	Perform the last round of an AES decryption flow
AESKEYGENASSIST	Assist in AES round key generation
AESIMC	Assist in AES Inverse Mix Columns

ARMv8 Cryptographic Extension

… and other

Cipher Modes: Electronic Code Book (ECB)

Direct encryption of each block: $C_i = E_K(P_i)$

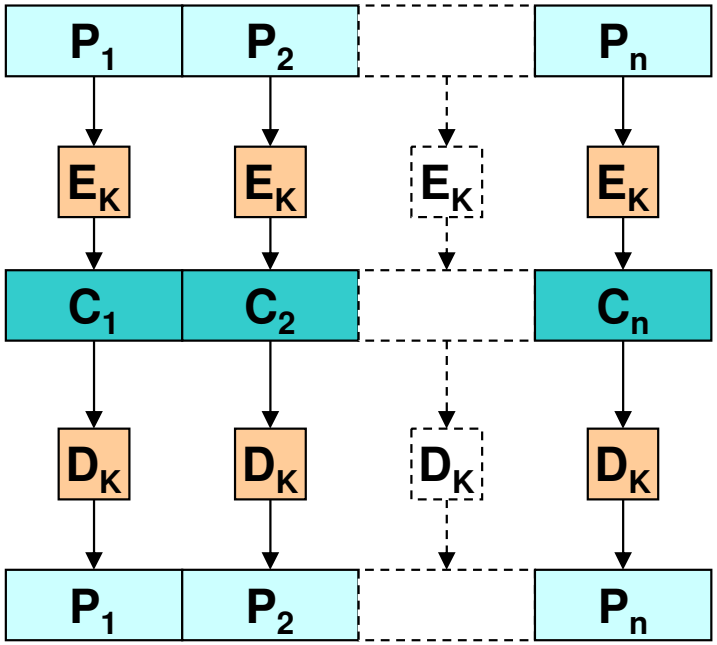
Direct decryption of each block: $P_i = D_K(C_i)$

Blocks are processed independently

- Parallelism is possible
- Uniform random access exists

Problem:

- Pattern exposure
- If $P_1 = P_2$ then $C_1 = C_2$



Cipher Modes: Cipher Block Chaining (CBC)

Encrypt each block T_i with feedback from C_{i-1}

- $C_i = E_K(P_i \oplus C_{i-1})$

Decrypt each block C_i with feedback from C_{i-1}

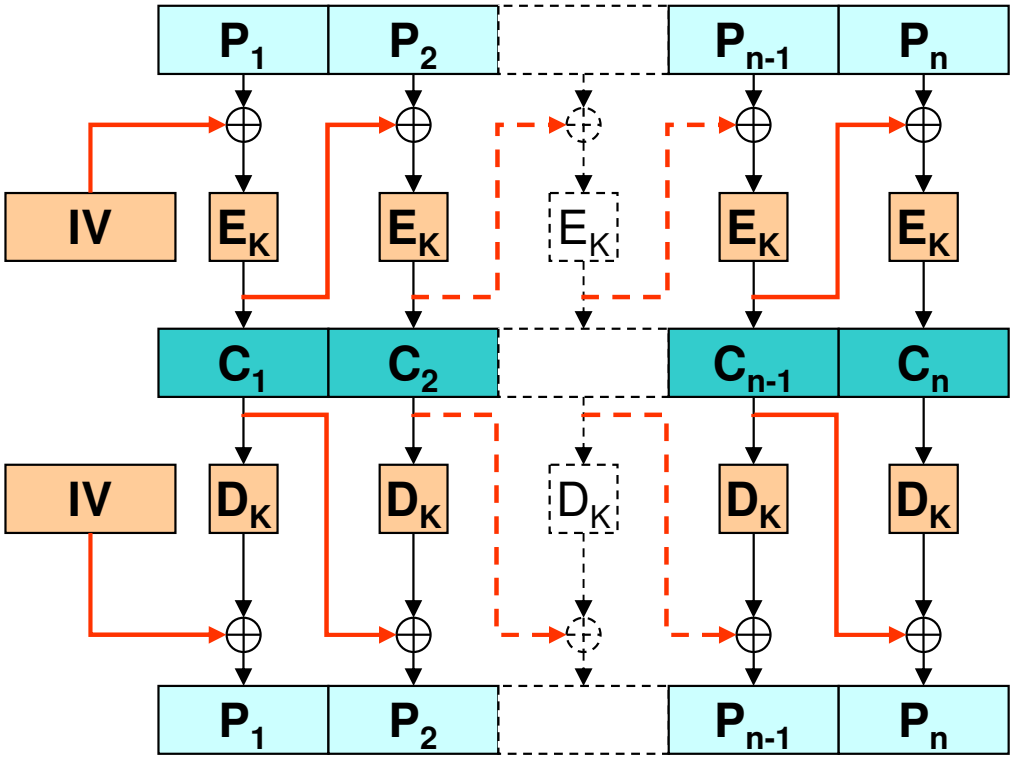
- $P_i = D_K(C_i) \oplus C_{i-1}$
- Parallelism and uniform random access is possible

First block uses an IV

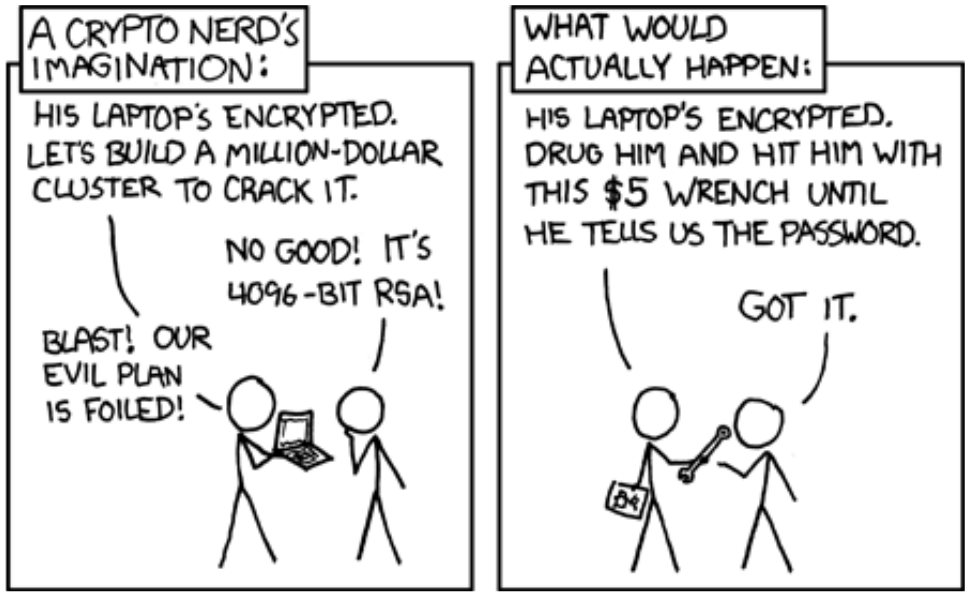
- IV: Initialization Vector
- Better not reuse for the same key
  - Random value, sequence value
- May be sent in clear

Polyalphabetic transformation

- The feedback prevents equal blocks from being equally processed
- Seems like we have a different key per block

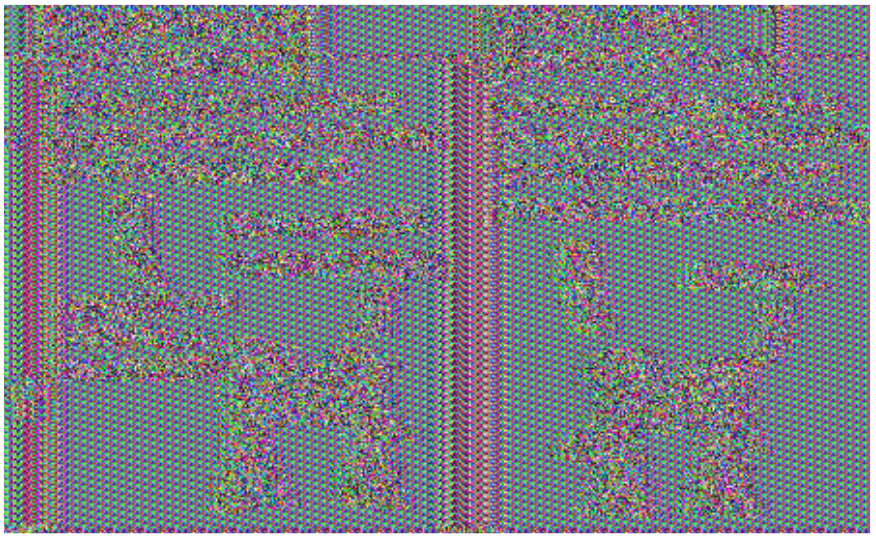


ECB vs CBC: pattern exposure



https://xkcd.com/538/

ECB



CBC

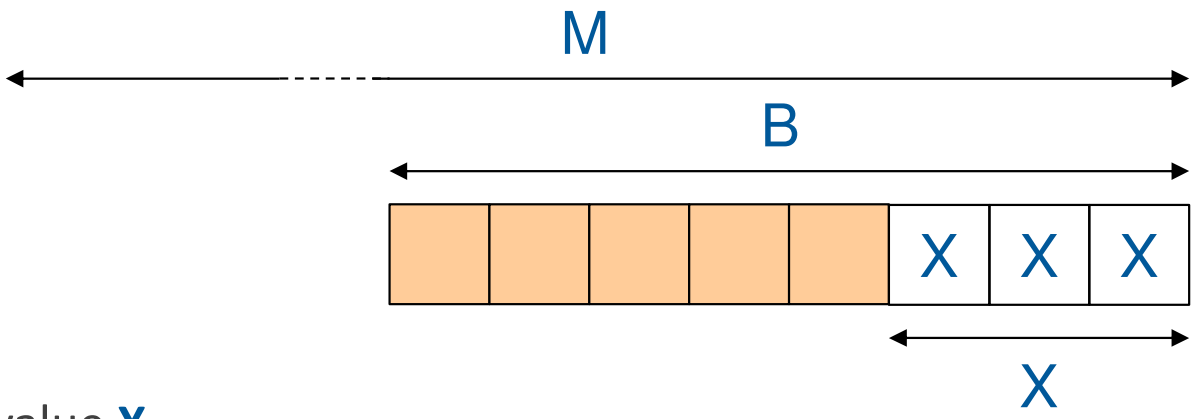
ECB/CBC cipher modes: Contents not block-aligned

ECB and CBC require block-aligned inputs

- Final sub-blocks need special treatment

Alternatives

- Padding
  - Of last block, identifiable
  - **PKCS #7**
    - **X = B – (M mod B)**
    - **X** extra bytes, with the value **X**
    - PKCS #5: Equal to PKCS #7 with B = 8
  - Perfectly aligned inputs get an extra padding block!
- Different processing for the sub-block
  - Adds complexity, rarely used



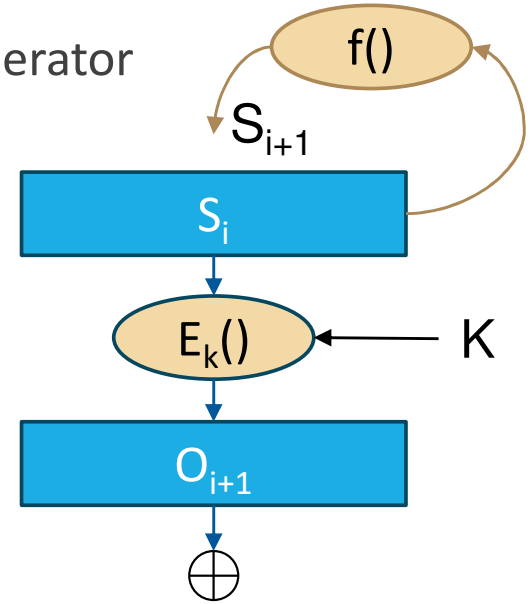
Stream cipher modes

Stream ciphers use a pseudorandom generator

- There are multiple techniques to implement them
- Some techniques are specially suited for hardware implementations
  - Typically used in mobile, battery-powered devices
- Other techniques are more suitable for CPU-based implementations

Stream cipher modes

- They use a block cipher to implement a stream cipher generator
- The fundamental idea is:
  - The generator e a state machine with state S_i on iteration i
  - The output of the generator for state S_i is $O_{i+1} = E_K(S_i)$
  - The state S_i is updated to S_{i+1} using some transformation function f
  - S_0 is defined by an IV
- The generator only uses block cipher encryptions
  - Or decryptions, it is irrelevant



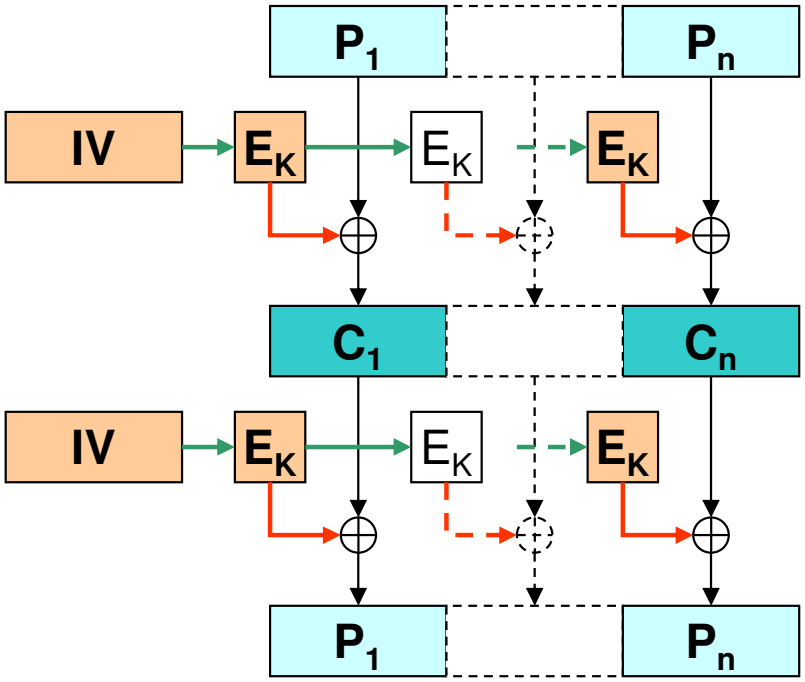
Stream cipher modes: n-bit OFB (Output Feedback)

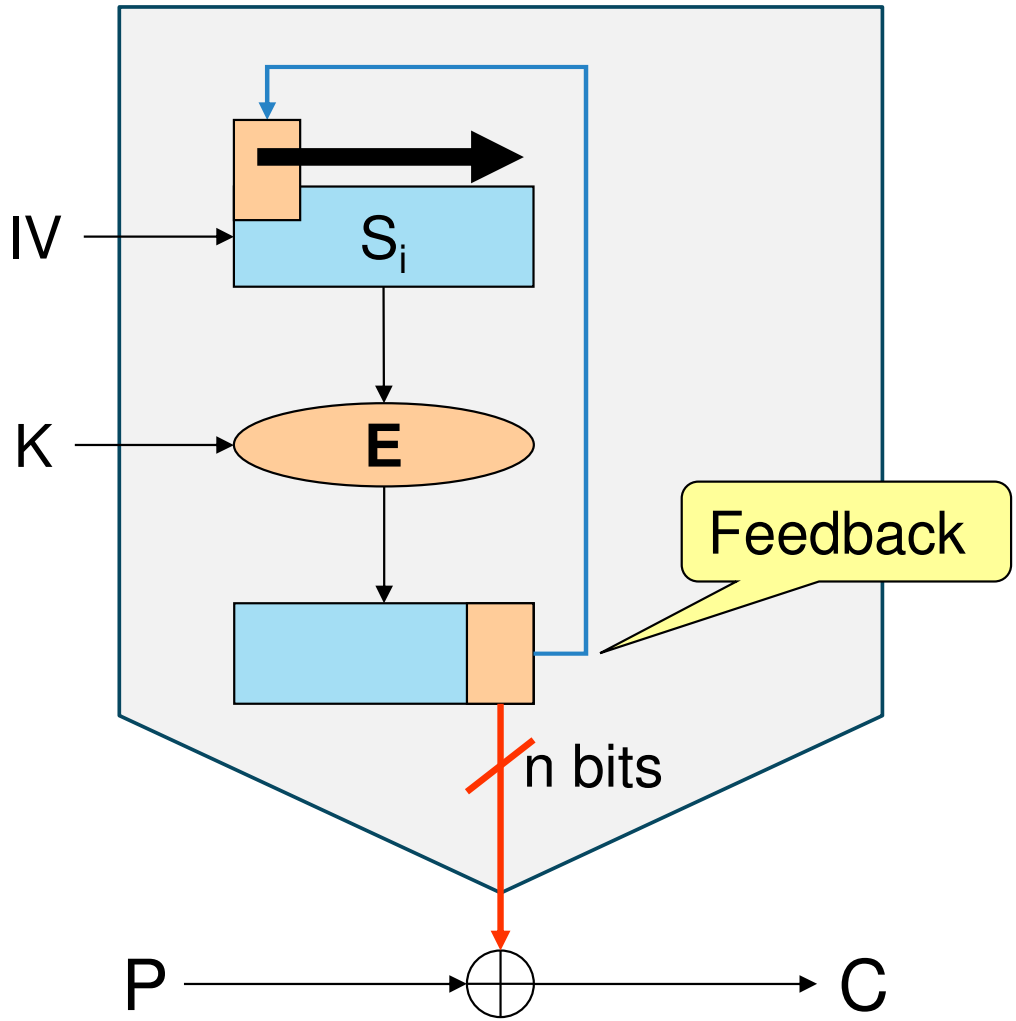
$C_i = T_i \oplus E_K(S_{i-1})$

$T_i = C_i \oplus E_K(S_{i-1})$

$S_{i+1} = f(S_i, E_K(S_i))$

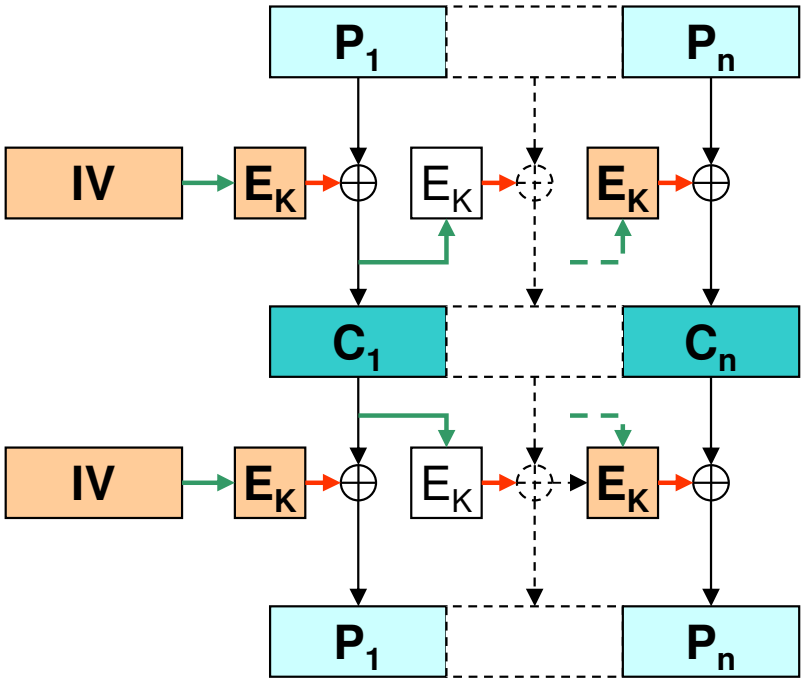
$S_0 = IV$

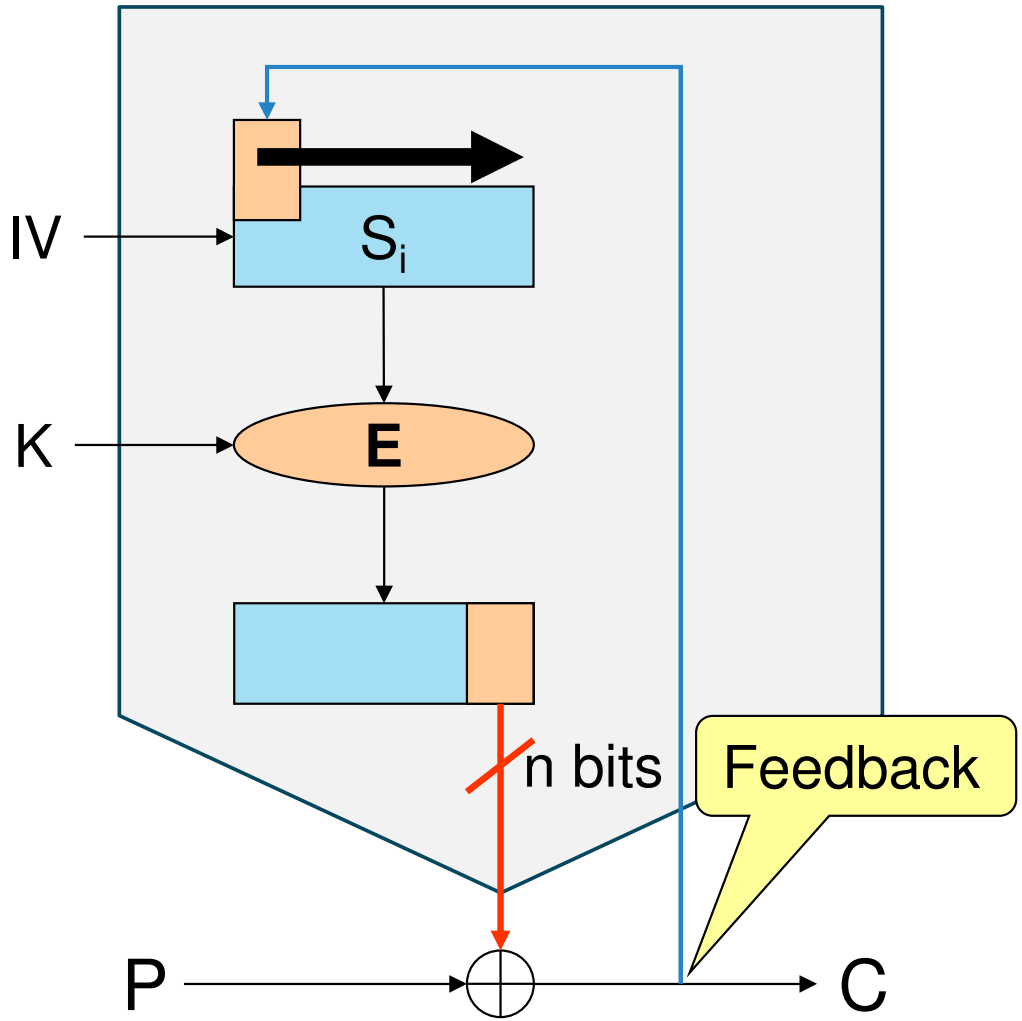




Stream cipher modes:
n-bit CFB (Ciphertext Feedback)

$C_i = T_i \oplus E_K(S_{i-1})$
$T_i = C_i \oplus E_K(S_{i-1})$
$S_{i+1} = f(S_i, C_i)$
$S_0 = IV$





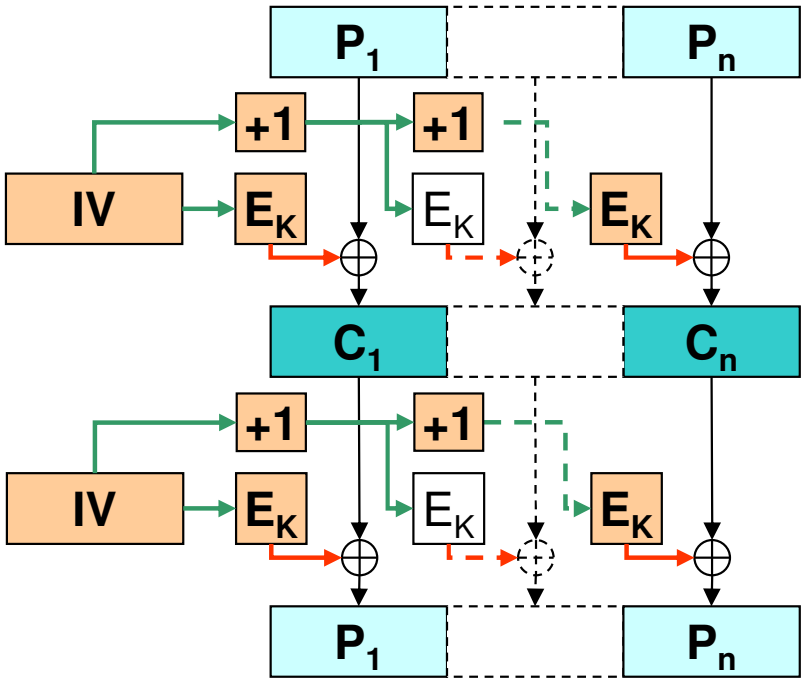
Cipher modes: n-bit CTR (Counter)

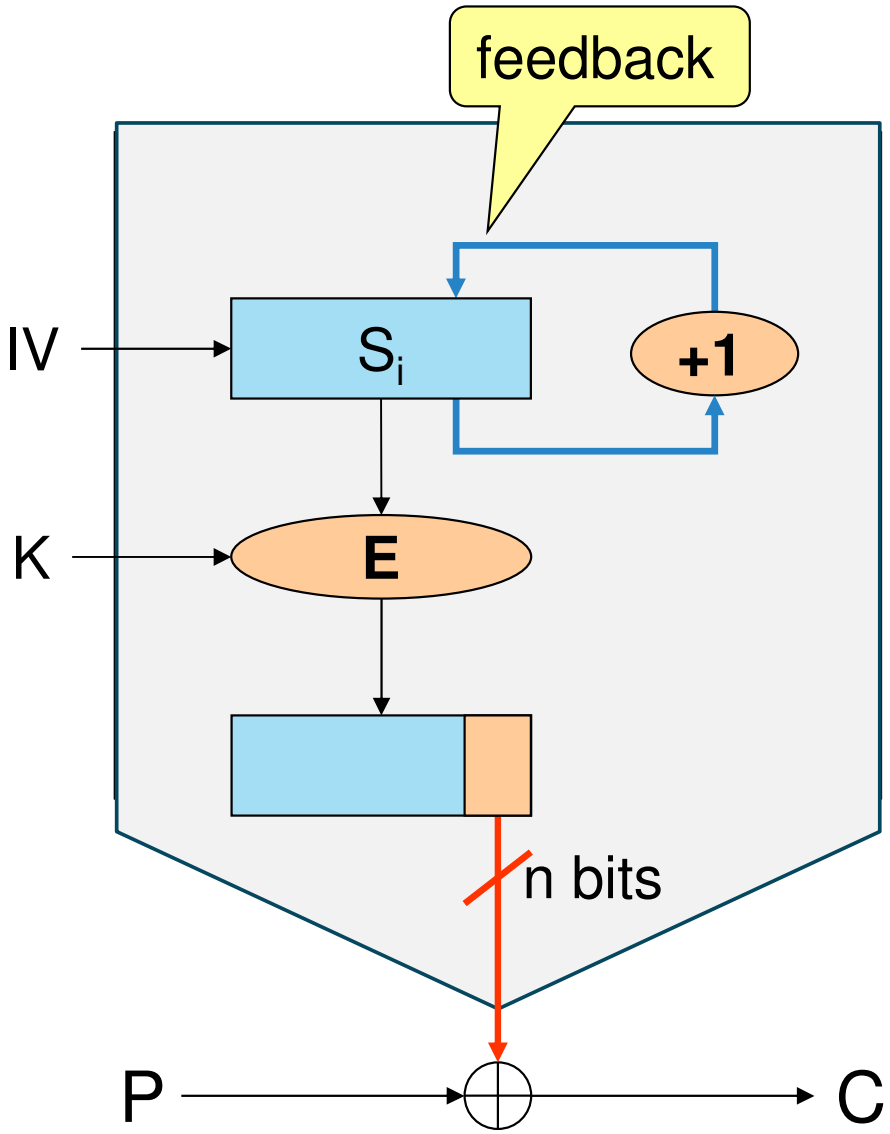
$C_i = T_i \oplus E_K(S_i)$

$T_i = C_i \oplus E_K(S_i)$

$S_i = S_{i-1}+1$

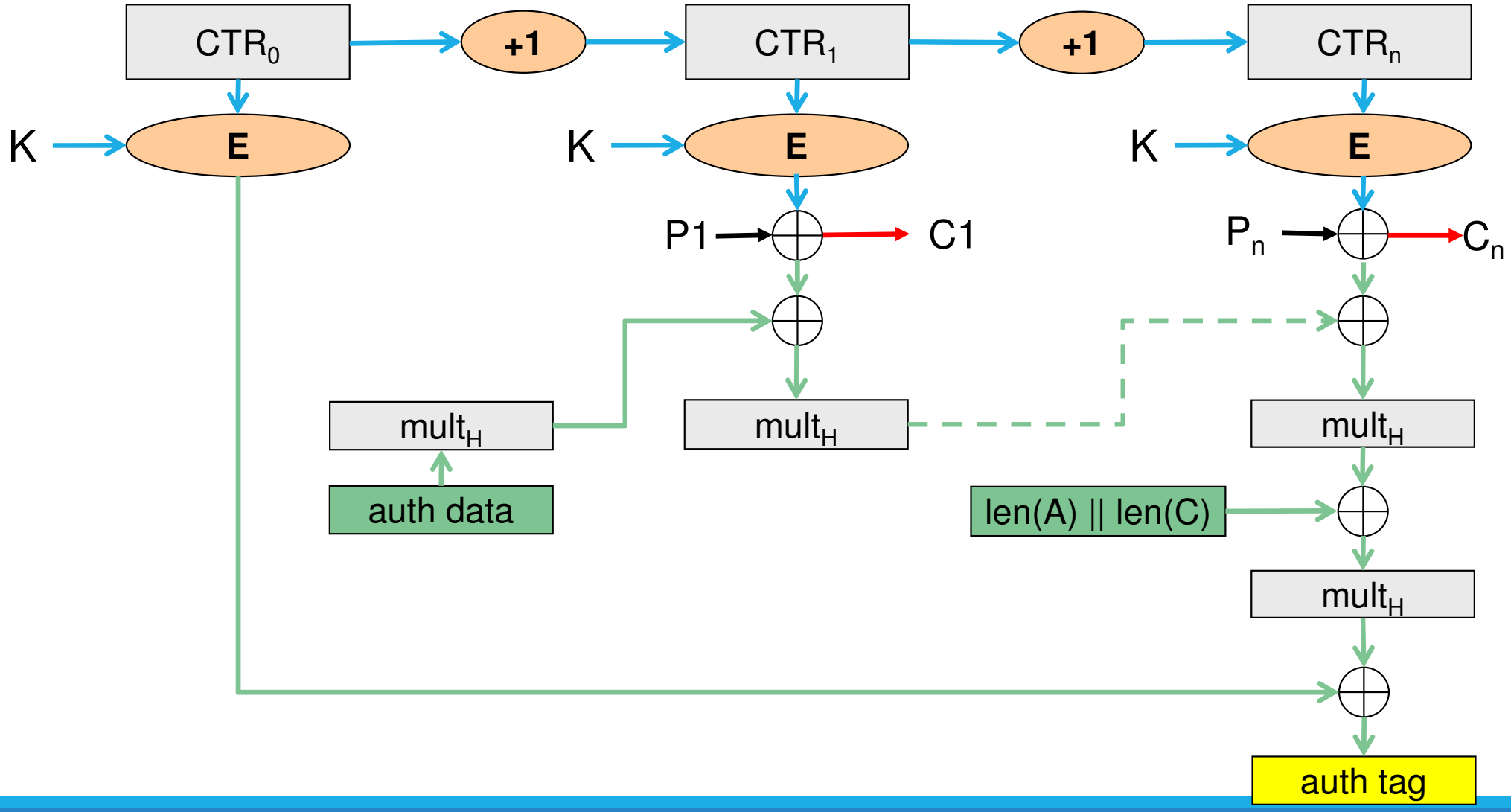
$S_0 = IV$





Stream cipher modes:

Galois with Counter Mode (GCM)



Cipher Modes: Comparison

	Block		Stream			
	ECB	**CBC**	**OFB**	**CFB**	**CTR**	**GCM**
Input pattern hiding		✓	✓	✓	✓	✓
Same key for different messages	✓	✓	other IV	other IV	other IV	other IV
Tampering difficulty	✓	✓ (…)		(…)		✓
Pre-processing			✓		✓	✓
Parallel processing	✓	decrypt	With pre-proc	decrypt	✓	✓
Uniform random access						
Cryptogram single bit error propagation on decryption	same block	same & next block		a few next bits		detected
Capacity to recover from losses	some	some		some		detected

Deteta mas propaga o erro

Cipher modes: multiple encryption

Invented for extending the lifetime of DES

- DES was never cryptanalysed
- But its key was too short (56 bits only)
- Its key could be discovered by brute force

Triple encryption EDE, or 3DES-EDE

- $C_i = E_{K3}(D_{K2}(E_{K1}(P_i)))$
- $P_i = D_{K1}(E_{K2}(D_{K3}(C_i))$
- With K1 ≠ K2 ≠ K3, it uses a 168-bit key
- With K1 = K3 ≠ K2, it uses a 112-bit key
- If K1 = K2 = K3 , then we get simple encryption

- In all cases, 3 times slower than DES

Cipher modes: DESX

Another solution for extending the lifetime of DES

- Much faster than 3DES
- Two extra keys are used to add confusion
  - Before the cipher input
  - After the cipher output
- $C_i = E_K(K_1 \oplus P_i) \oplus K_2$
- $P_i = K_1 \oplus D_K(K_2 \oplus C_i)$

- The length of the equivalent key is 184 bits
  - 64 + 64 + 56 bits
  - More than with 3DES