

# Introduction

---

INFORMATICS AND ORGANIZATIONAL SECURITY

# Security objectives (1/3)

---

## Defense against catastrophic events

- Natural phenomena
  - Abnormal temperature, lightning, thunder, flooding, radiation, ...
- Degradation of computer hardware
  - bad sectors in disks, failure of power supplies, bit errors in RAM cells or SSD, etc.

## Solution:

- Realistic prevention: focus in the most probable events
- Information backup
- Replication
  - Information
  - Computational resources

# Security objectives (2/3)

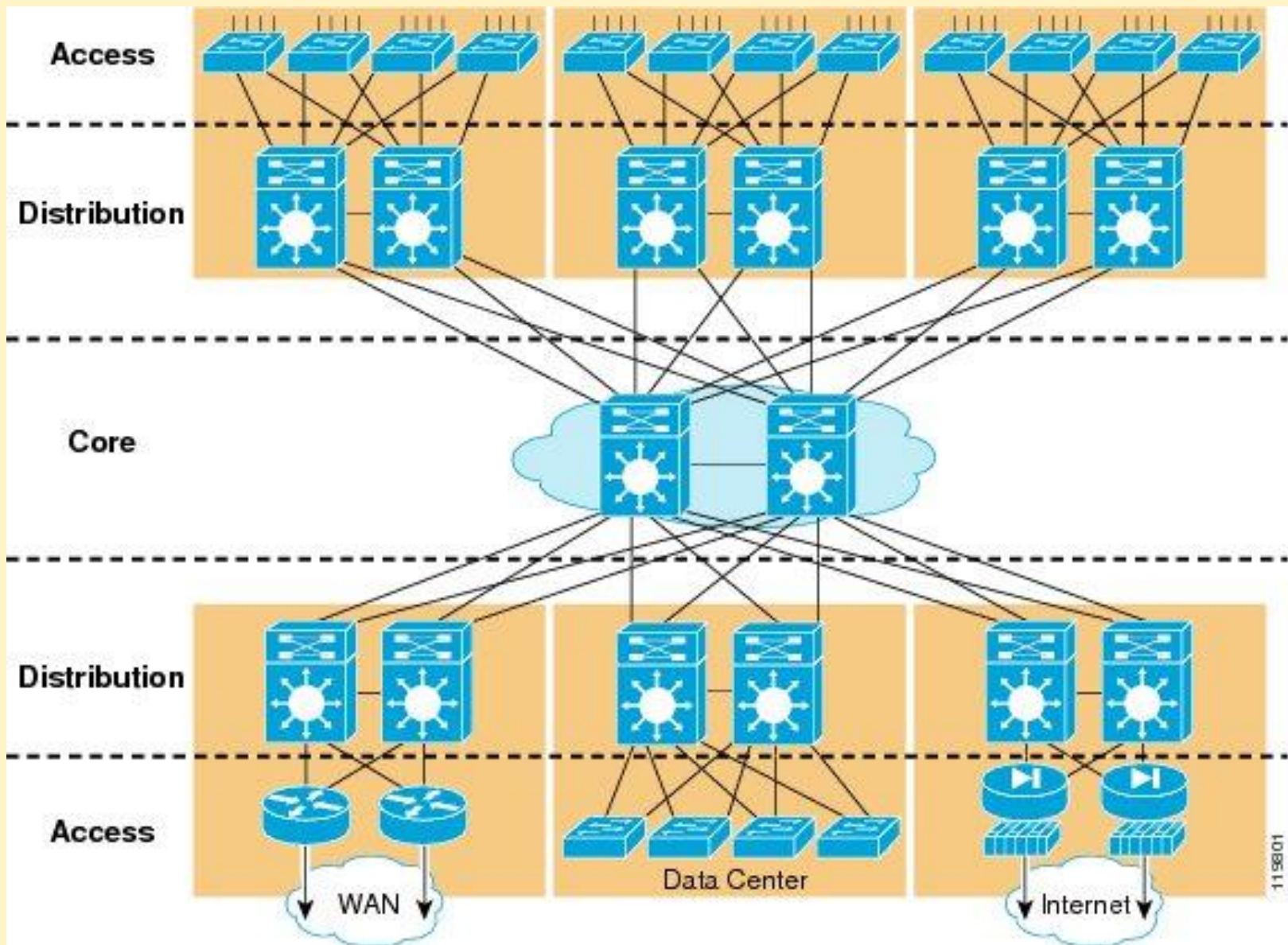
---

## Defense against ordinary faults / failures

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

## Solution:

- Redundant / alternative power supplies (and fan, disks, links...)
- Transactional systems
- Dynamic Routing, re-transmission



Source: CISCO



Source: DELL

# Security objectives (3/3)

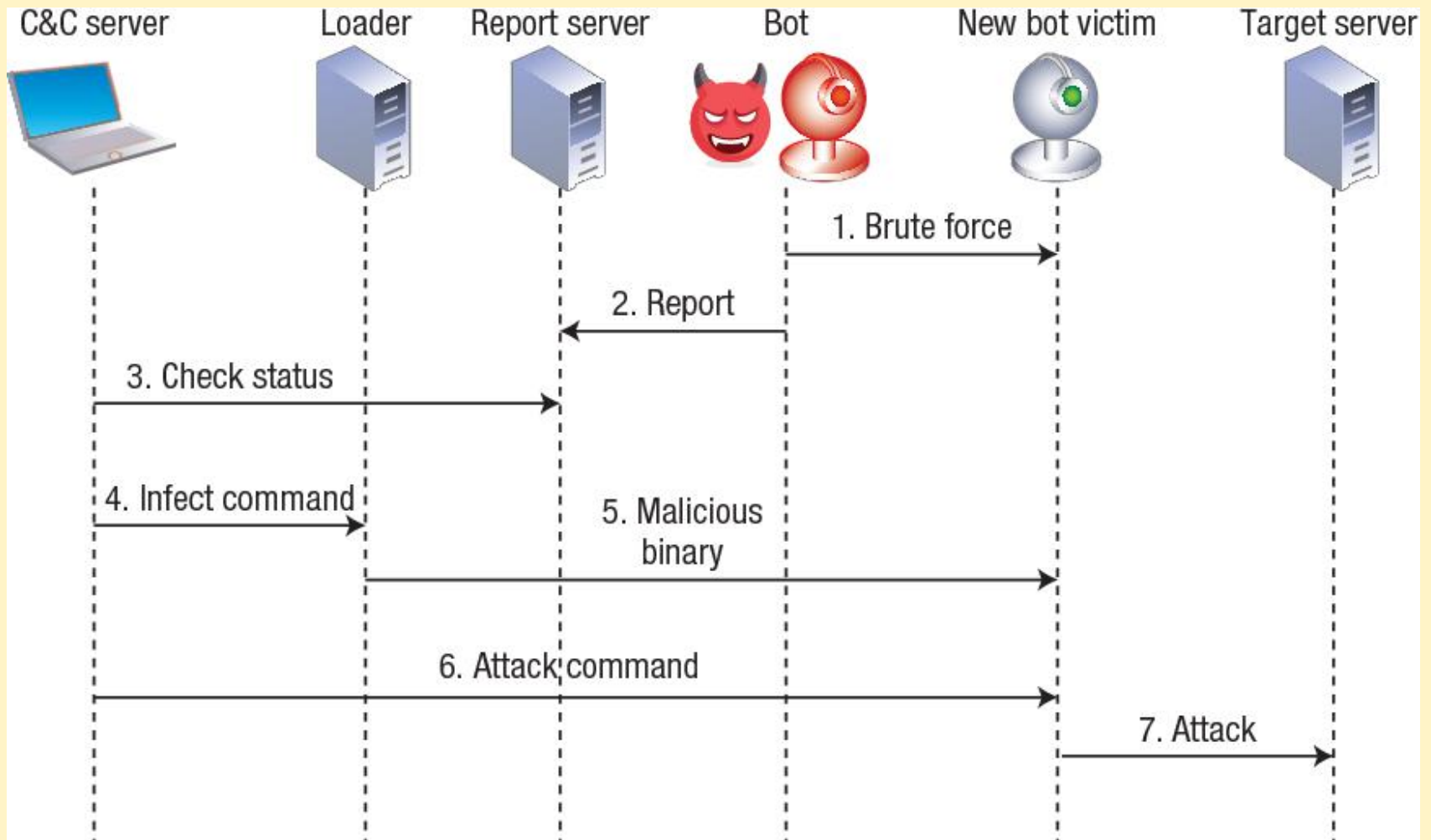
---

## **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



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



# Security in computing systems: Complex problems

---

## **Computers can do much damage in a short time frame**

- 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

## **Networks allow novel attack mechanisms**

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



# 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

# The pragmatic approach

---

## **Perfect protection of a system is impossible**

- Solution: Cost-efficiency balance
- Problem: determine cost and efficiency

## **Security is expensive**

- Dedicated technology, additional resources, skilled professionals, processes
- Solution: Deploy the minimum required

## **Protection, value and punishment**

- Protection that is “good enough” for the most frequent attacks
- Less interference with daily work than the damage caused by the attackers
- Use the police and courts to track and prosecute attackers
  - It is important in any society or organization to avoid the notion of total impunity

# Lexicon

---

## **Vulnerability**

- A system weakness that makes it sensible to attacks
- Can be present in every step of its life cycle

## **Attack**

- A set of actions that lead to the execution of illegal activities
  - Attacks usually explore vulnerabilities

## **Risk / threat**

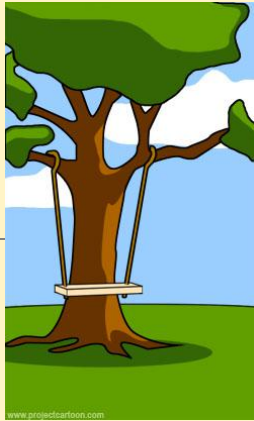
- The possible damage resulting from an attack

## **Defense**

- Set of policies, mechanisms and technologies aiming at
  - Reduce the amount of vulnerabilities
  - Detect as fast as possible actual and past attacks
  - Reduce the risks of the systems



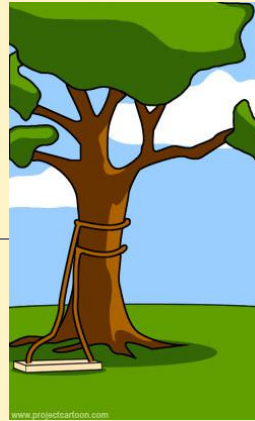
How the customer explained it



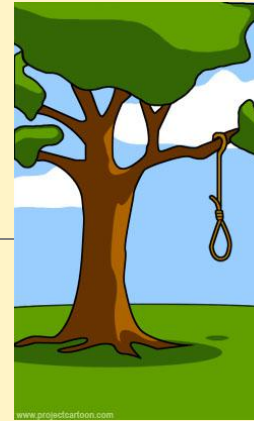
How the project leader understood it



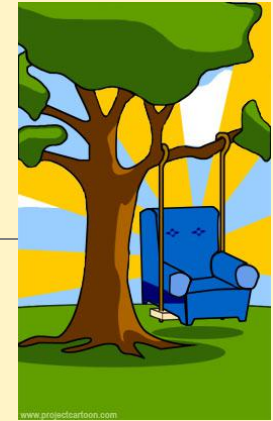
How the analyst designed it



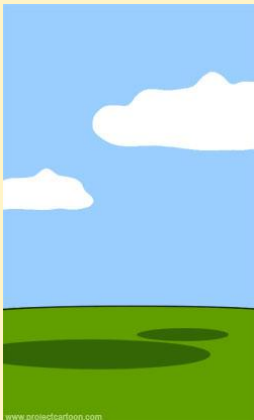
How the programmer wrote it



What the beta testers received



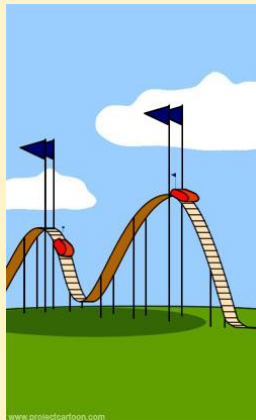
How the business consultant described it



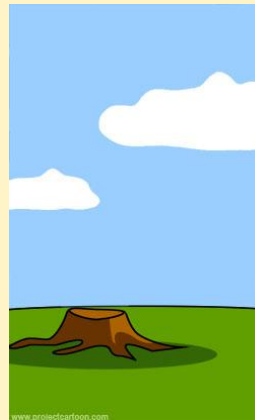
How the project was documented



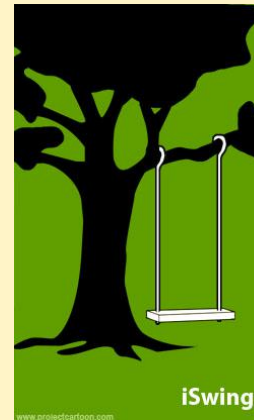
What operations installed



How the customer was billed



How it was supported



What marketing advertised



What the customer really needed

Project development is complex, crosses several expert domains, several teams, with potentially unaligned incentives or expertise. Vulnerabilities can be introduced in every point

# Security risks

---

## **Information, time e resources (money)**

- Destruction or tampering of information

## **Confidentiality**

- Non-authorized access to information

## **Privacy**

- Non-authorized gathering of personal information
- Data warehousing (or distribution) of personal information

## **Resources Availability**

- Disruption of systems, communications, processes

## **Impersonation**

- Non-authorized exploitation of personal accounts /profiles
  - accounts related to people, services, entities, etc...

# 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 (ex, macros)

## Users

- Ignorant or careless
  - telnet vs. ssh, IMAP vs. IMAPS, HTTP vs HTTPS
  - False sense of security (I have an anti-virus, so I'm protected!)
- Hostile

## Defective administration

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

## Communication over uncontrolled/unknown network links

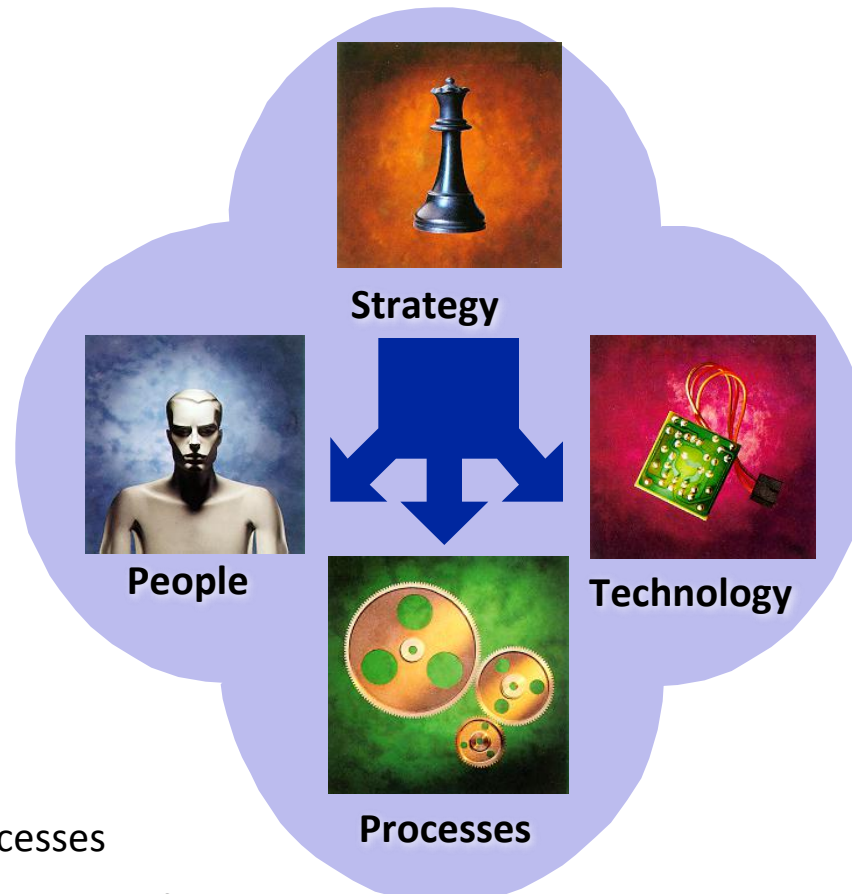
- Public hotspots, campus networks, hostile governments

# Dimensions to consider

---

- Selection
- Training
- Awareness
- Organization of security

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



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



# 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**

# Security Policies

---

## **Define the power of each subject**

- Least privilege principle; Hardening

## **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)

## **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**

- Black list model: All that is not forbidden is allowed
- White list model: All that is not allowed 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 level (of a computer)

---

## Defined by:

- Available security policies
- Correctness and effectiveness of their specification/implementation

## Evaluation criteria

- NCSC Trusted Computer System Evaluation Criteria (TCSEC, Orange Book)
  - Classes: **D**, **C** (1, 2), **B** (1, 2, 3) e **A** (1)
  - D: insecure (minimum protection level)
  - A1: most secure
    - very demanding and expensive protection policies
    - formal validation of the specification with highly supervised implementation
- EC Information Technology Security Evaluation Criteria (ITSEC)
  - Levels: **E1** to **E6**
    - Level of formal specification
    - Correctness of the implementation

# Case Study: NCSC TCSEC (C)

---

## **C1 – Discretionary Security Protection**

- Identification and authentication
- Separation of users and data
- Discretionary Access Control (DAC) capable of enforcing access limitations on an individual basis
- Required System Documentation and user manuals

## **C2 – Controlled Access Protection**

- More finely grained DAC
- Individual accountability through login procedures
- Audit trails
- Object reuse
- Resource isolation

# Case Study: NCSC TCSEC (C)

---

## Object Reuse Policy:

- All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation or reallocation to a subject from the TCB's pool of unused storage objects.
- No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system.

**Storage object:** An object that supports both read and write accesses.

# Security policies for distributed systems

---

**Must encompass several hosts and networks**

## **Security Domains**

- Definition of the set of hosts and networks of the domain
- Definition of the set of accepted/authorized users
- Definition of the set of accepted/not accepted activities

## **Security gateways**

- Definition of the set of allowed in-out interactions



# Security policies for distributed systems

---

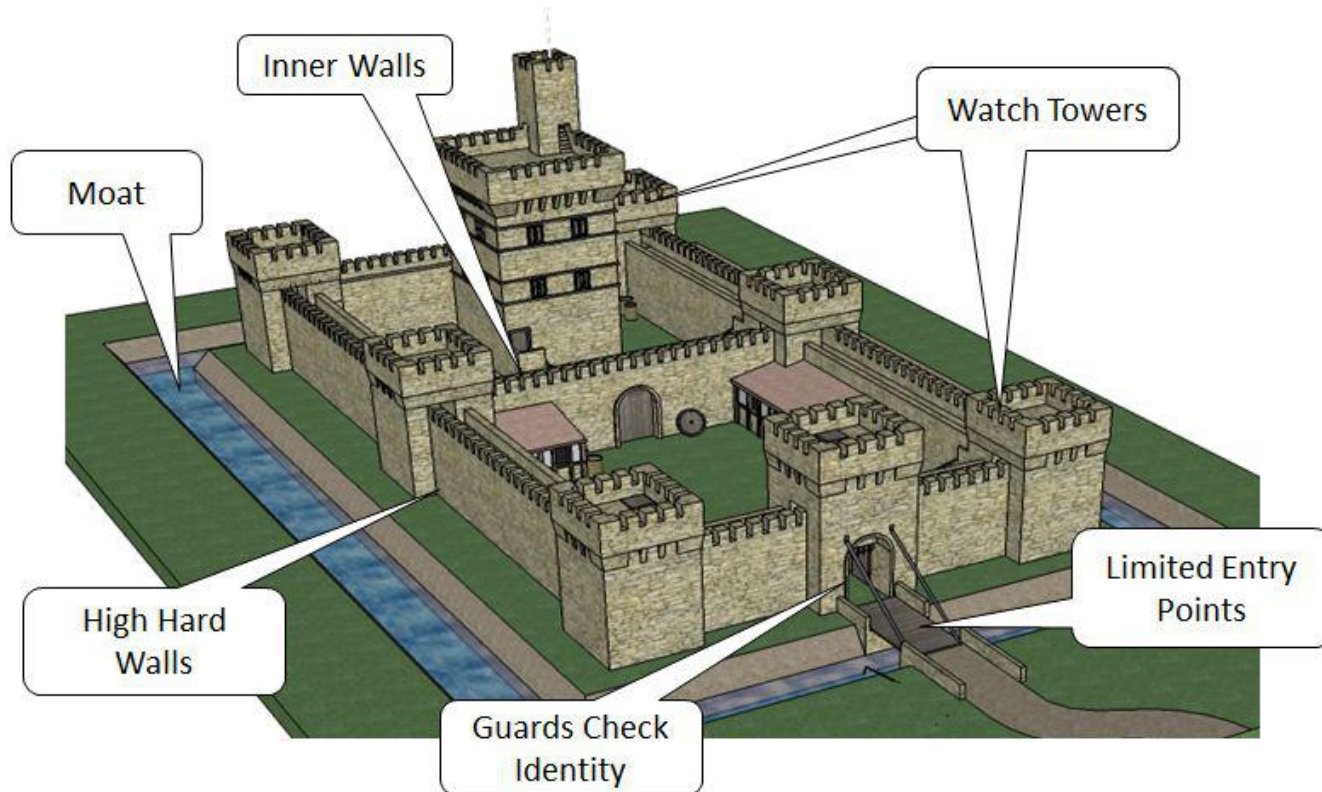
## Perimeter defense

*(minimal defense)*



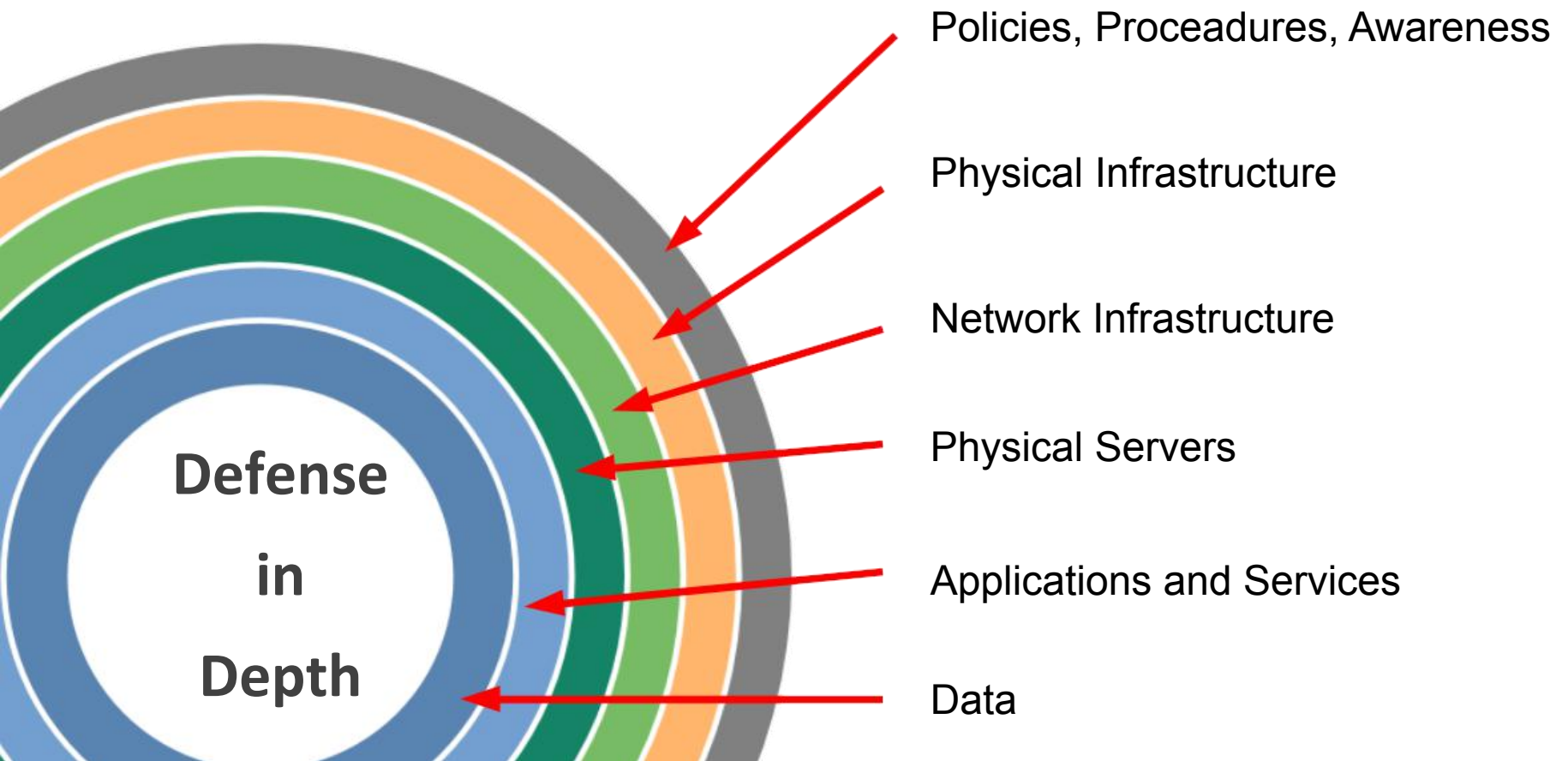
# Security policies for distributed systems

## Defense in depth



# Security policies for distributed systems

---



# Security policies for distributed systems

---

## Target-specific attacks

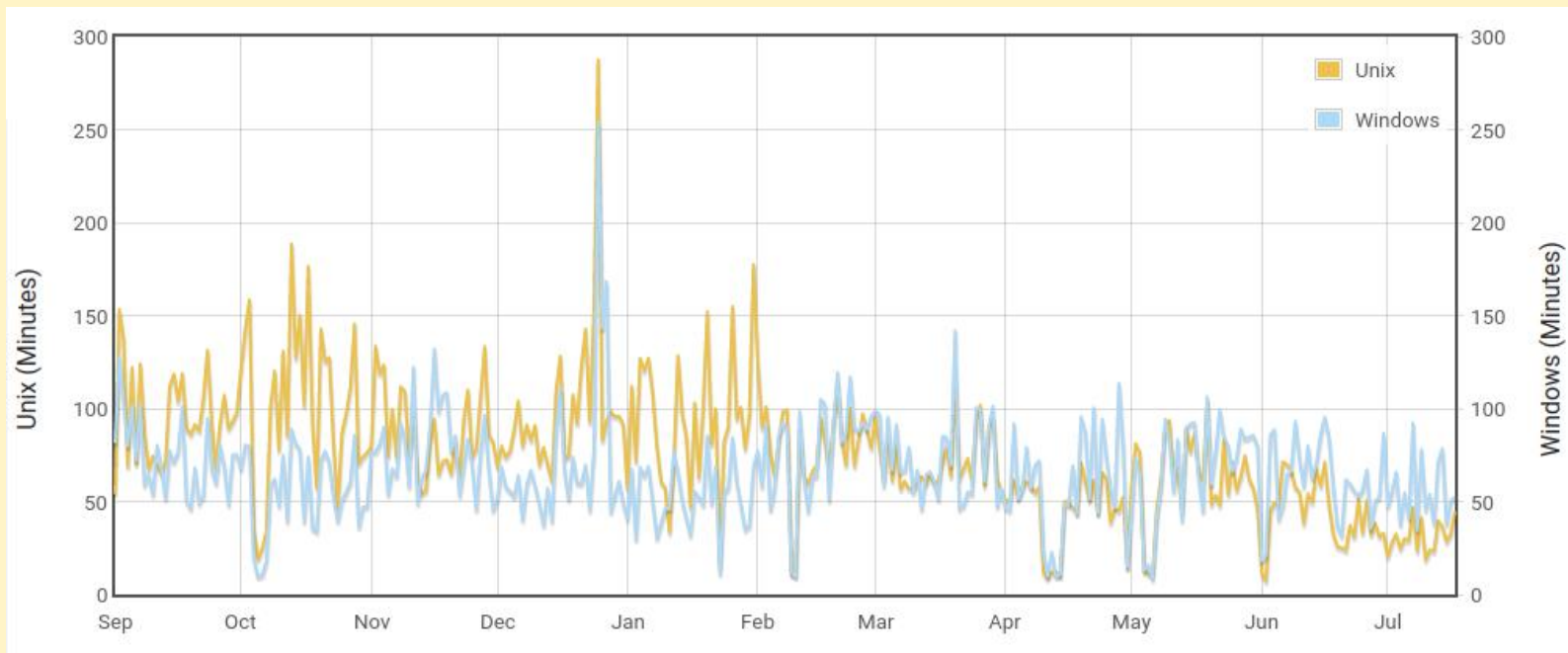
- Conceived for a particular host / network
- Idealized and conducted in real-time by specialists

## Generic, autonomous attacks

- Conceived for exploiting well-known, common vulnerabilities
- Coded for many scenarios and targets
- Mean survivability time :
  - Time between two consecutive automatic attacks
  - There are “network sensors” that help to compute it
- Executed by professionals, curious individuals, students

# Case Study: Mean Survival Time

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



# Mechanisms for distributed systems

---

## **Trusted Operating Systems**

- Security levels, certification
- Secure execution environments for servers
- Sand-boxing / virtual machines

## **Firewalls & Security Appliances**

- Traffic control between networks
- Monitoring (traffic load, etc.)

## **Secure communications / VPNs**

- Secure channels over insecure, public networks
- Secure extension of organizational networks

# Mechanisms for distributed systems

---

## **Authentication**

- Local
- Remote (network authentication)
- Single Sign-On
- Using secrets, token, bio-metrics, device, location

## **Certification Authorities / PKI**

- Management of public key certificates

## **Encryption of files and sessions**

- Privacy / confidentiality of network data
- Privacy / confidentiality of long-term stored data



# Mechanisms for distributed systems

---

## **Intrusion detection**

- Detection of forbidden / abnormal activities
- Network-Based / Host-based

## **Vulnerability scanners**

- Scanning for problem fixing or exploitation
- Network-based / Host-based

## **Penetration testing**

- Vulnerability assessment
- Demo penetration attempts
- Testing of installed security mechanisms
- Assessment of badly implemented security policies

# Mechanisms for distributed systems

---

## **Content monitoring**

- Detection of virus, worms or other cyber plagues

## **Security administration**

- Development of security policies
- Distributed enforcement of policies
- Co-administration / outsourcing of security services

## **Real-Time Security Awareness / Incident Response**

- Capacity to detect and react to security incidents in real-time
- Means for a rapid and effective incident reaction