Co-funded by the Erasmus+ Programme of the European Union

PENS
Pathway in Enterprise Systems Engineering

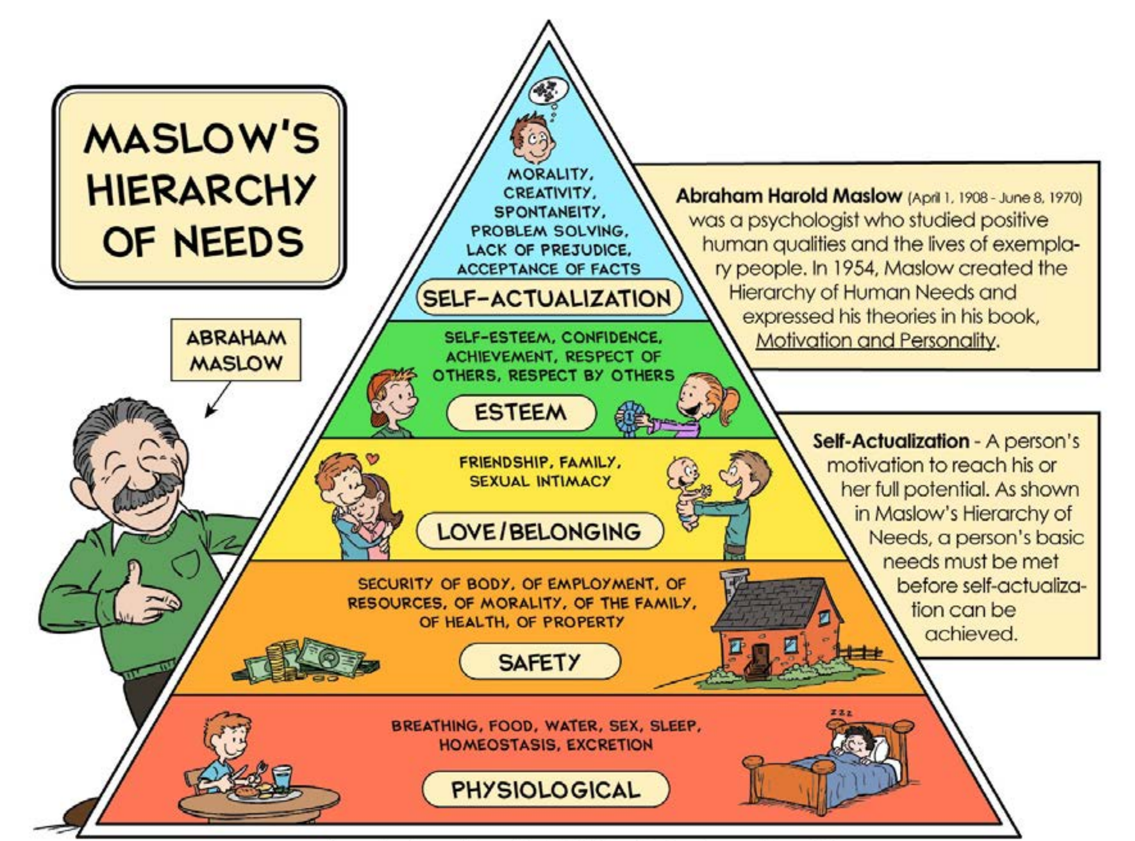
Pathway in Enterprise Systems Engineering (PENS)

INTRODUCTION TO INFORMATION SYSTEMS SECURITY

Giorgio Giacinto

18 July 2022
Universidad de Alcala

What is security?


MASLOW'S HIERARCHY OF NEEDS
ABRAHAM MASLOW
MORALITY, CREATIVITY, SPONTANEITY, PROBLEM SOLVING, LACK OF PREJUDICE, ACCEPTANCE OF FACTS
SELF-ACTUALIZATION
SELF-ESTEEM, CONFIDENCE, ACHIEVEMENT, RESPECT OF OTHERS, RESPECT BY OTHERS
ESTEEM
FRIENDSHIP, FAMILY, SEXUAL INTIMACY
LOVE/BELONGING
SECURITY OF BODY, OF EMPLOYMENT, OF RESOURCES, OF MORALITY, OF THE FAMILY, OF HEALTH, OF PROPERTY
SAFETY
BREATHING, FOOD, WATER, SEX, SLEEP, HOMEOSTASIS, EXCRETION
PHYSIOLOGICAL
zzz

Abraham Harold Maslow (April 1, 1908 - June 8, 1970) was a psychologist who studied positive human qualities and the lives of exemplary people. In 1954, Maslow created the Hierarchy of Human Needs and expressed his theories in his book, Motivation and Personality.

Self-Actualization - A person's motivation to reach his or her full potential. As shown in Maslow's Hierarchy of Needs, a person's basic needs must be met before self-actualization can be achieved.

www.timvandevall.com |Copyright © 2013 Dutch Renaissance Press LLC.

Co-funded by the Erasmus+ Programme of the European Union

The circle of trust

Meet the parents, 2000

Meet the Fockers, 2004

https://youtu.be/QHJGoZpFeM8

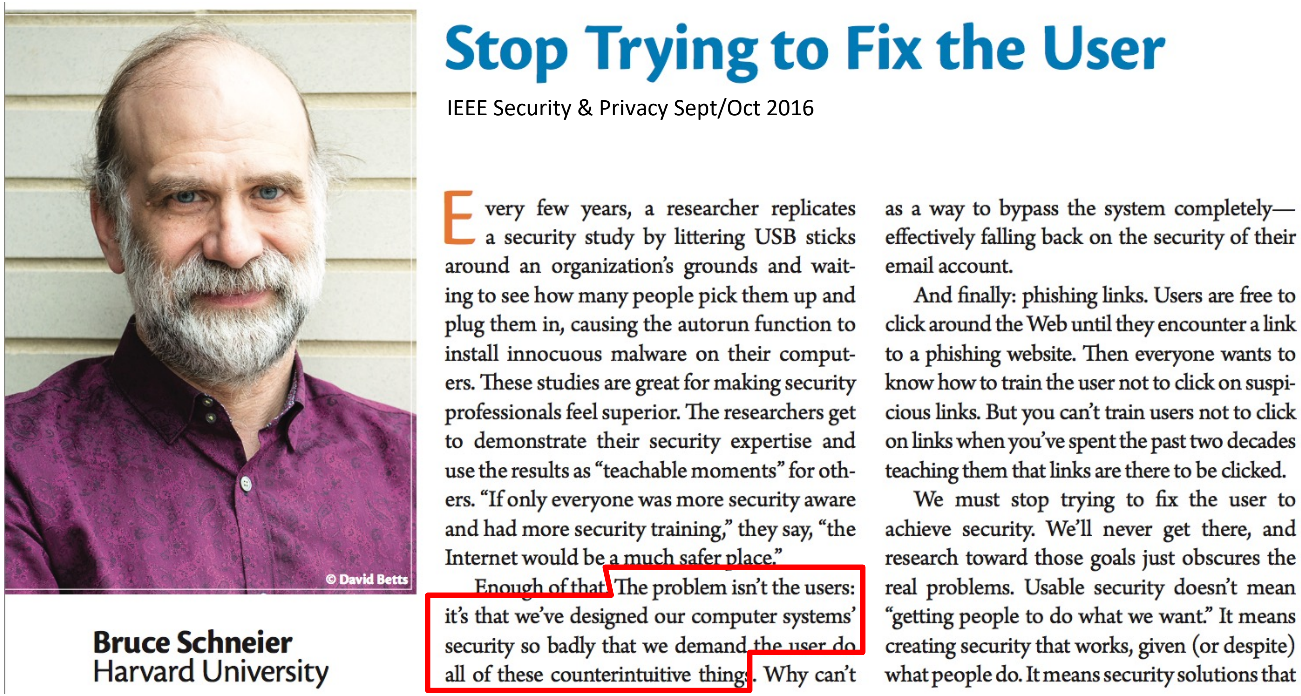
Co-funded by the Erasmus+ Programme of the European Union

Trust

Only amateurs attack machines
professionals target people.

Bruce Schneier

The "human factor"



Stop Trying to Fix the User

IEEE Security & Privacy Sept/Oct 2016

© David Betts

Bruce Schneier
Harvard University

Every few years, a researcher replicates a security study by littering USB sticks around an organization's grounds and waiting to see how many people pick them up and plug them in, causing the autorun function to install innocuous malware on their computers. These studies are great for making security professionals feel superior. The researchers get to demonstrate their security expertise and use the results as "teachable moments" for others. "If only everyone was more security aware and had more security training," they say, "the Internet would be a much safer place."

Enough of that. The problem isn't the users: it's that we've designed our computer systems' security so badly that we demand the user do all of these counterintuitive things. Why can't

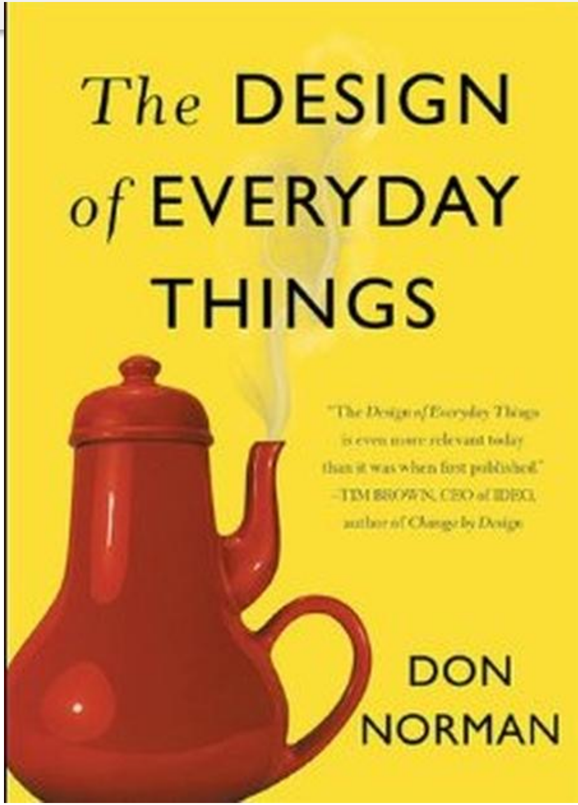
as a way to bypass the system completely—effectively falling back on the security of their email account.

And finally: phishing links. Users are free to click around the Web until they encounter a link to a phishing website. Then everyone wants to know how to train the user not to click on suspicious links. But you can't train users not to click on links when you've spent the past two decades teaching them that links are there to be clicked.

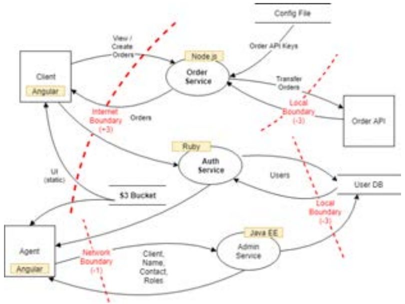
We must stop trying to fix the user to achieve security. We'll never get there, and research toward those goals just obscures the real problems. Usable security doesn't mean "getting people to do what we want." It means creating security that works, given (or despite) what people do. It means security solutions that

Co-funded by the
Erasmus+ Programme
of the European Union

Human-Centered Design


The DESIGN
of EVERYDAY
THINGS
"The Design of Everyday Things
is even more relevant today
than it was when first published."
–TIM BROWN, CEO of IDEO,
author of Change by Design
DON
NORMAN

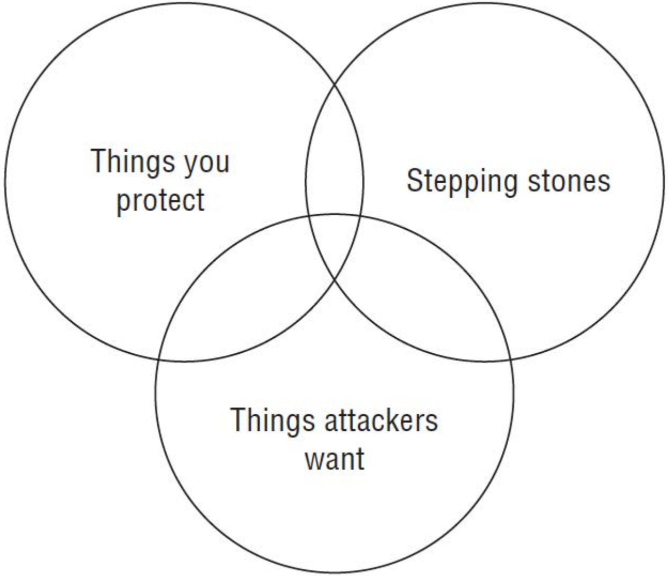
- Five psychological concepts
- AFFORDANCES
- SIGNIFIERS
- CONSTRAINTS
- MAPPINGS
- FEEDBACK
- Objects (and software) designed according to these concepts exhibit discoverability
  - what it does
  - how it works
  - what operations are possible



Threat Modeling

Assets To Protect

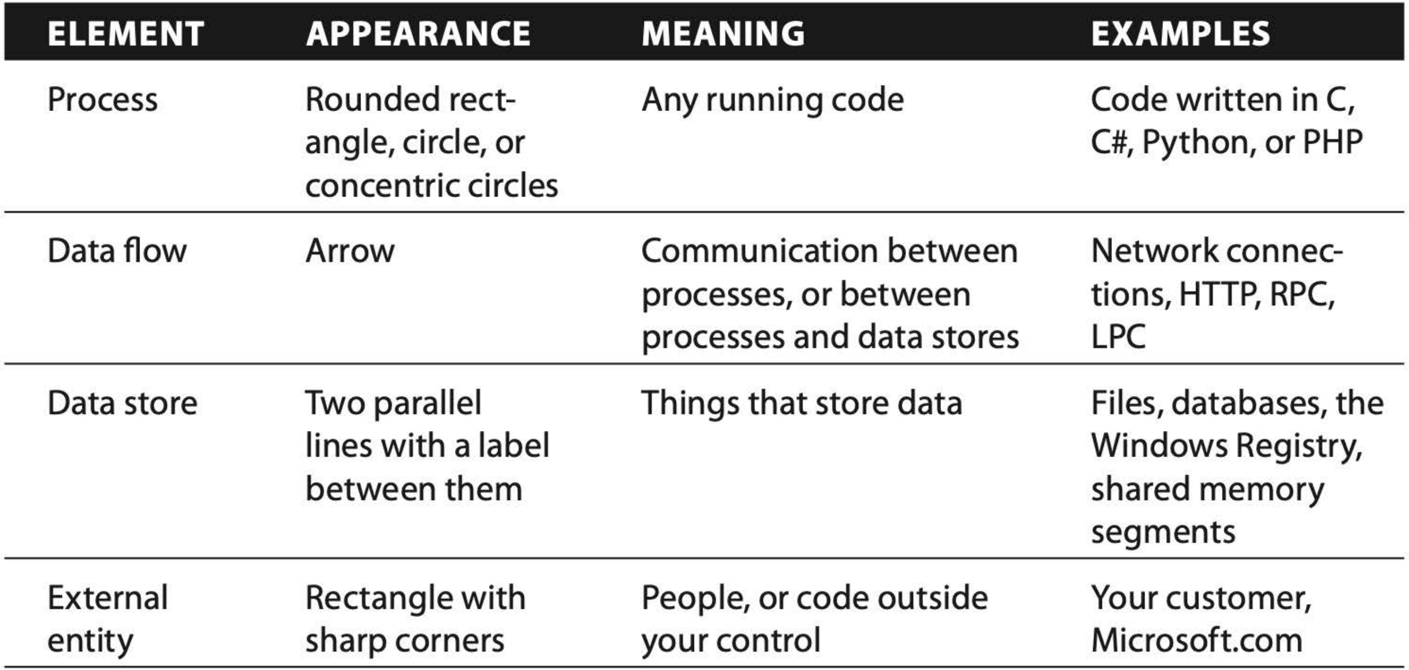
- **Things Attackers Want**
  - User passwords
  - SSN, identifiers
  - Credit card numbers
  - Confidential business data
- **Intangible Assets You Want to Protect**
  - Reputation
  - Goodwill
  - Unused assets
- **Stepping Stones**
  - Everything that can be used to attack other assets



Software THREAT MODELING

- Security-centric approach to threat modeling
- Based on software models described by diagrams
  - Data flow diagrams
  - UML
  - Swin Lane Diagrams
  - State diagrams
- Based on the definition of Trust Boundaries

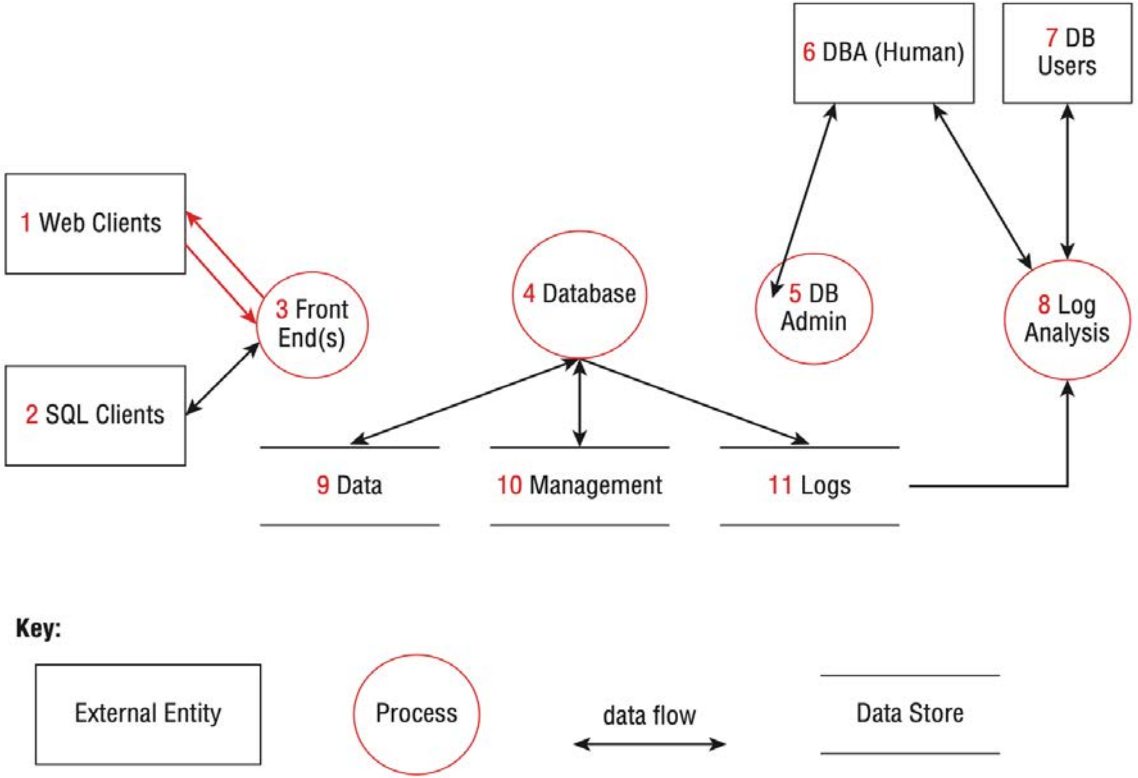
Data Flow Diagrams (DFD)



ELEMENT	APPEARANCE	MEANING	EXAMPLES
Process	Rounded rectangle, circle, or concentric circles	Any running code	Code written in C, C#, Python, or PHP
Data flow	Arrow	Communication between processes, or between processes and data stores	Network connections, HTTP, RPC, LPC
Data store	Two parallel lines with a label between them	Things that store data	Files, databases, the Windows Registry, shared memory segments
External entity	Rectangle with sharp corners	People, or code outside your control	Your customer, Microsoft.com

Co-funded by the Erasmus+ Programme of the European Union

Data Flow Diagram Example



Trust Boundaries

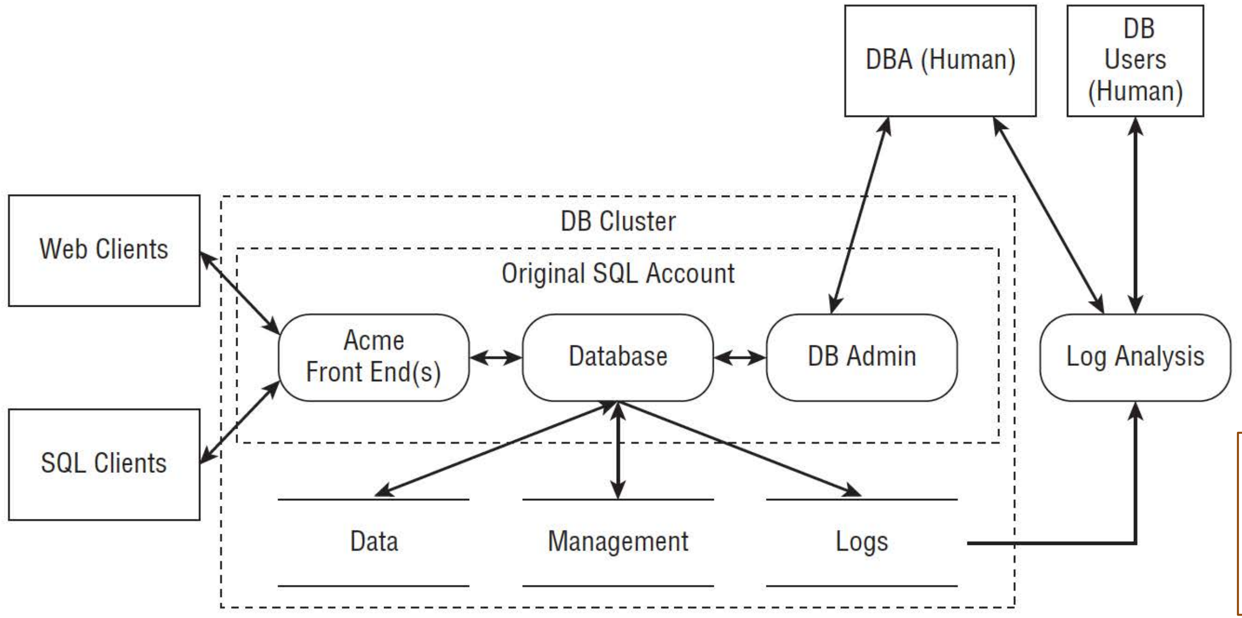
- Trust Boundaries are placed **where entities with different privileges interact**

- Two **questions** are useful to draw Trust Boundaries.
  - **First**: does everything in the system have the same level of privilege and access to everything else on the system?
  - **Second**: is everything your software communicates with inside that same boundary?

- If **either** of these **answers** are a **NO**, then you should now have clarified either a **missing boundary** or a **missing element** in the diagram, or both.

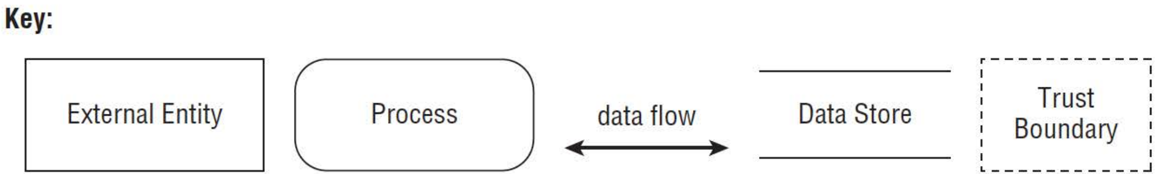
- If **both answers** are **YES**, then you should draw **a single trust boundary around everything**, and move on to other development activities

Trust Boundaries



Trust Boundaries typically **cross data flows**

The ACME Corporation is a fictional corporation featured in the Looney Tunes animated shorts



What can go wrong?

- **STRIDE** taxonomy (orginally proposed by Microsoft)
- **S**poofing
- **T**ampering
- **R**epudiation
- **I**nformation Disclosure
- **D**enial of Service
- **E**levation of Privilege

Co-funded by the Erasmus+ Programme of the European Union

Spoofing Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Spoofing a process on the same machine	Creates a file before the real process	
	Renaming / linking	Creating a Trojan “su” and altering the path
	Renaming	Naming your process “sshd”
Spoofing a file	Creates a file in the local directory	A library, executable or config file
	Creates a link and changes it	The change should happen between the link being checked and the link being accessed
	Creates many files in the expected directory	e.g., automatic creation of 10,000 files in the `/tmp` directory to fill all the available space

Co-funded by the
Erasmus+ Programme
of the European Union

Spoofing Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Spoofing a machine	ARP spoofing	
	IP spoofing	
	DNS spoofing	Forward or reverse
	DNS compromise	Compromise TLD, registrar or DNS operator
	IP redirection	At the switch or router level
Spoofing a person	Sets e-mail display name	
	Take over a real account	
Spoofing a role	Declares themselves to be that role	Sometimes opening a special account with a relevant name

Co-funded by the Erasmus+ Programme of the European Union

Tampering Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Tampering with a file	Modifies a file they own and which you rely on	
	Modify a file you own	
	Modifies a file on a file server that you own	
	Modifies a file on their file server	Effective when you include files from remote domains
	Modifies links or redirects	
Tampering with memory	Modifies your code	Hard to defend against once the attacker is running code as the same user
	Modifies data they've supplied to your API	Pass by values, not by reference when crossing a trust boundary

Tampering Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Tampering with a network	Redirects the flow of data to their machine	Often stage 1 of tampering
	Modifies data flowing over the network	Even easier when the network is wireless (e.g., WiFi, 4G, etc.)
	Enhance spoofing attacks	

Co-funded by the Erasmus+ Programme of the European Union

Repudiation Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Repudiating an action	Claims to have not clicked	
	Claims to have not received	How reliable are receipts of delivery / download?
	Claims to have been a fraud victim	
	Uses someone else’s account	
	Uses someone else’s payment instrument without authorization	
Attacking the logs	Notices you have no logs	
	Puts attacks in the logs to confuse logs, log-reading code, or persons reading the log	

Co-funded by the Erasmus+ Programme of the European Union

Information Disclosure Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Information disclosure against a process	Extracts secrets from error messages	
	Reads the error messages from username/passwords to entire database tables	
	Extracts machine secretes from error cases	Can make defense against memory corruption such as ASLR far less useful
	Extracts business/personal secrets from error cases	

Co-funded by the
Erasmus+ Programme
of the European Union

Information Disclosure Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Information disclosure against data stores	Takes advantage of inappropriate or missing ACLs	
	Takes advantage of bad database permissions	
	Finds file protected by obscurity	
	Finds crypto keys on disk (or in memory)	
	Sees interesting information in filenames	
	Reads files as they traverse the network	
	Gets data from logs or temp files	
	Gets data from swap or other temp storage	
	Extracts data by obtaining device, changing OS	

Co-funded by the
Erasmus+ Programme
of the European Union

Information Disclosure Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Information disclosure against a data flow	Reads data on the network	
	Redirects traffic to enable reading data on the network	
	Learns secretes by analyzing traffic	
	Learns who’s talking to whom by watching the DNS	
	Learns who’s talking to whom by social network info disclosure	

Co-funded by the
Erasmus+ Programme
of the European Union

Denial of Service Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Denial of service against a process	Absorbs memory (RAM or disk)	
	Absorbs CPU	
	Uses process as an amplifier	
Denial of service against a data store	Fills data store up	
	Makes enough requests to slow down the system	
Denial of service against a data flow	Consumes network resources	

Co-funded by the
Erasmus+ Programme
of the European Union

Elevation of Privilege Threats

THREAT EXAMPLES	WHAT THE ATTACKER DOES	NOTES
Elevation of privilege against a process by corrupting the process	Sends inputs that the code doesn't handle properly	These errors are very common, and have high impact
	Gains access to read or write memory inappropriately	Reading memory can enable further attacks
Elevation through missed authorization checks		
Elevation through buggy authorization checks		Centralizing such checks make bugs easier to manage
Elevation through data tampering	Modifies bits on disk to do things other than what the authorized user intends	

Co-funded by the
Erasmus+ Programme
of the European Union

Security

- *The state of being free from danger or threat*

- *The state of feeling safe, stable, and free from fear or anxiety*

Co-funded by the
Erasmus+ Programme
of the European Union

Enforcing security

Prevention

Detection/Deterrence

Reaction

These measures introduce constraints

Co-funded by the
Erasmus+ Programme
of the European Union

Security and constraints

- The **tradeoff** between the limitations and security
  - is subjective
  - depends on the context
- The evaluation of the tradeoff needs the evaluation of
  - **Threats**
  - **Risks**

    the *probability* of a given threat

    the *impact* of the threat

Security is the issue of the weakest link

- All systems have weak links

  …and the weakest link will be the target!

- Strategies to mitigate the *weakest link* risks
  - **Defense in depth**
    threat analysis on any part of the system
  - **Compartmentalization**
    exploiting one vulnerability should not affect the all system
  - **Choke points**
    a few known weak links where controls and defenses must be deployed

"The Prince of Egypt", 1998
https://youtu.be/PiJcKAXISLk?t=31

Co-funded by the Erasmus+ Programme of the European Union

Security is a complex system

- Security policies and mechanisms form a system that interacts with
  - itself
  - the protected assets
  - the context

- These interactions can cause **failures**
  - the system can *fail to prevent* / detect / respond to a threat
  - the system can *fail by reacting* in absence of a threat

All the causes of failure of the security system need to be carefully analysed

Co-funded by the Erasmus+ Programme of the European Union

Types of failure of security systems

- **Active Failures**
  The system performs some activities in absence of threats

- **Passive Failures**
  The systems does not manage the threat properly

- Threats are rare events
  - False alarms cannot be avoided
  - The behavior of the system in the absence of threats must be carefully analysed
  - Active failures can be simply annoying, but they could also be leveraged to hide the true threat
  - Active failures could produce severe consequences if the alarm triggers some reaction mechanisms

Co-funded by the
Erasmus+ Programme
of the European Union

Active Failures

”Il Mostro”, 1994 - https://youtu.be/0adl6T6nV1w

Co-funded by the
Erasmus+ Programme
of the European Union

Passive failures

Difficulties in attributing the threat correctly

“Baby Driver”, 2017 - https://youtu.be/6XMuUVw7TOM?t=241

Co-funded by the Erasmus+ Programme of the European Union

Security and Computers

Co-funded by the
Erasmus+ Programme
of the European Union

The Value of Things

Co-funded by the
Erasmus+ Programme
of the European Union

Cyber Crime

High gain/cost ratio

Goods and Risks are transformed into intangible assets

Low material costs

Life is rarely at risk

Cyber Crime is
not perceived as a Crime

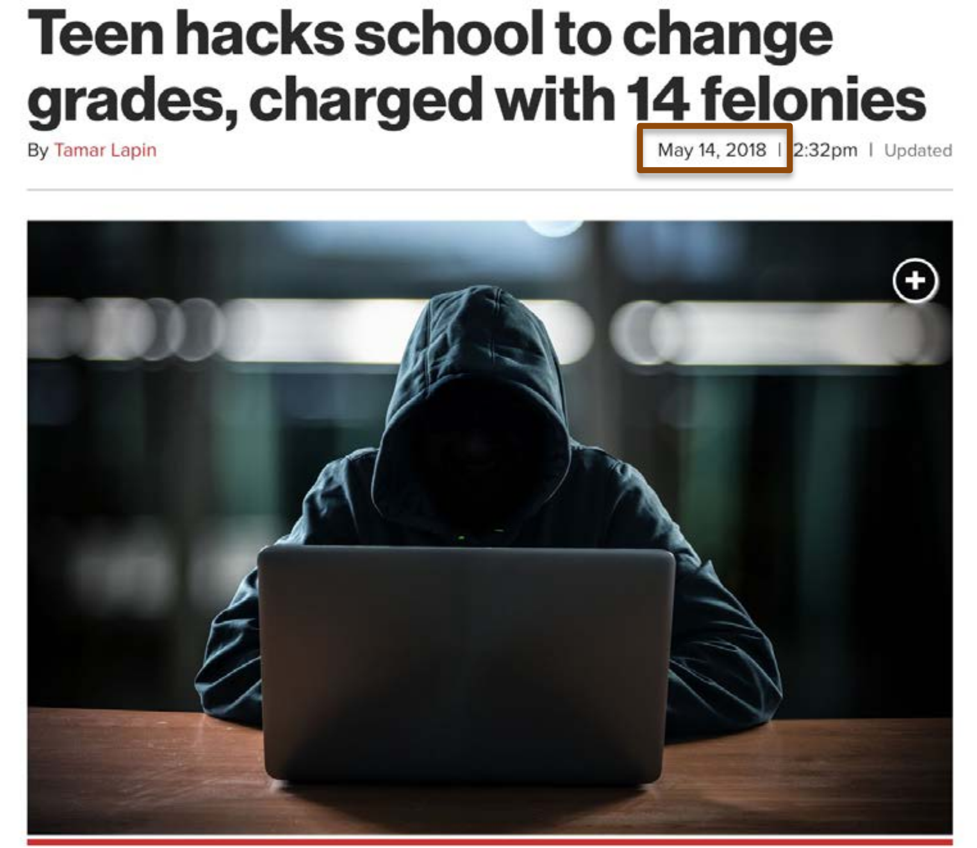
Co-funded by the
Erasmus+ Programme
of the European Union

The ’80…

“Wargames”, 1983 - https://youtu.be/U2_h-EFlztY

Co-funded by the Erasmus+ Programme of the European Union

Decades Later

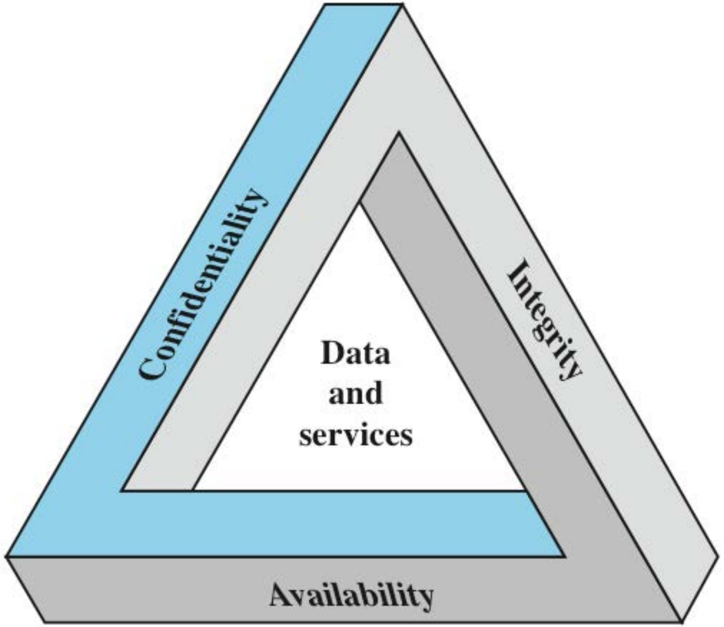


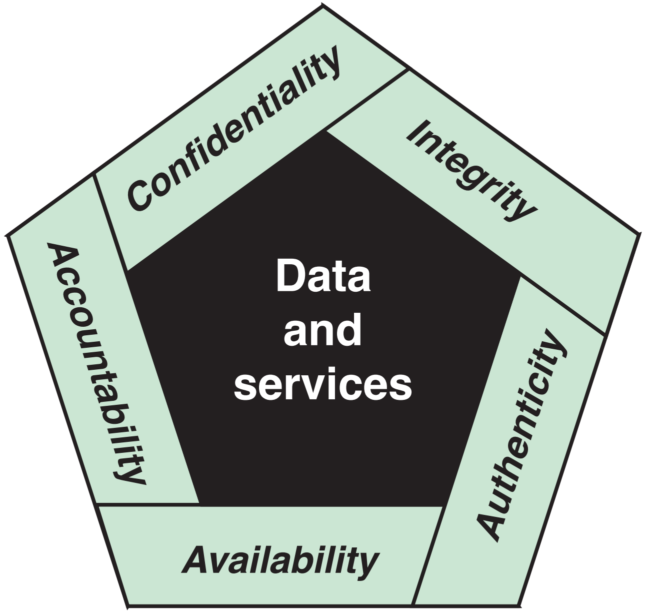
Teen hacks school to change grades, charged with 14 felonies

By Tamar Lapin May 14, 2018 | 2:32pm | Updated

Computer Threats

The CIA Triad





Stallings

Levels of Impact

on organizational **operations**, organizational **assets**, or **individuals**

LOW

The loss could be expected to have a **limited adverse effect**

MODERATE

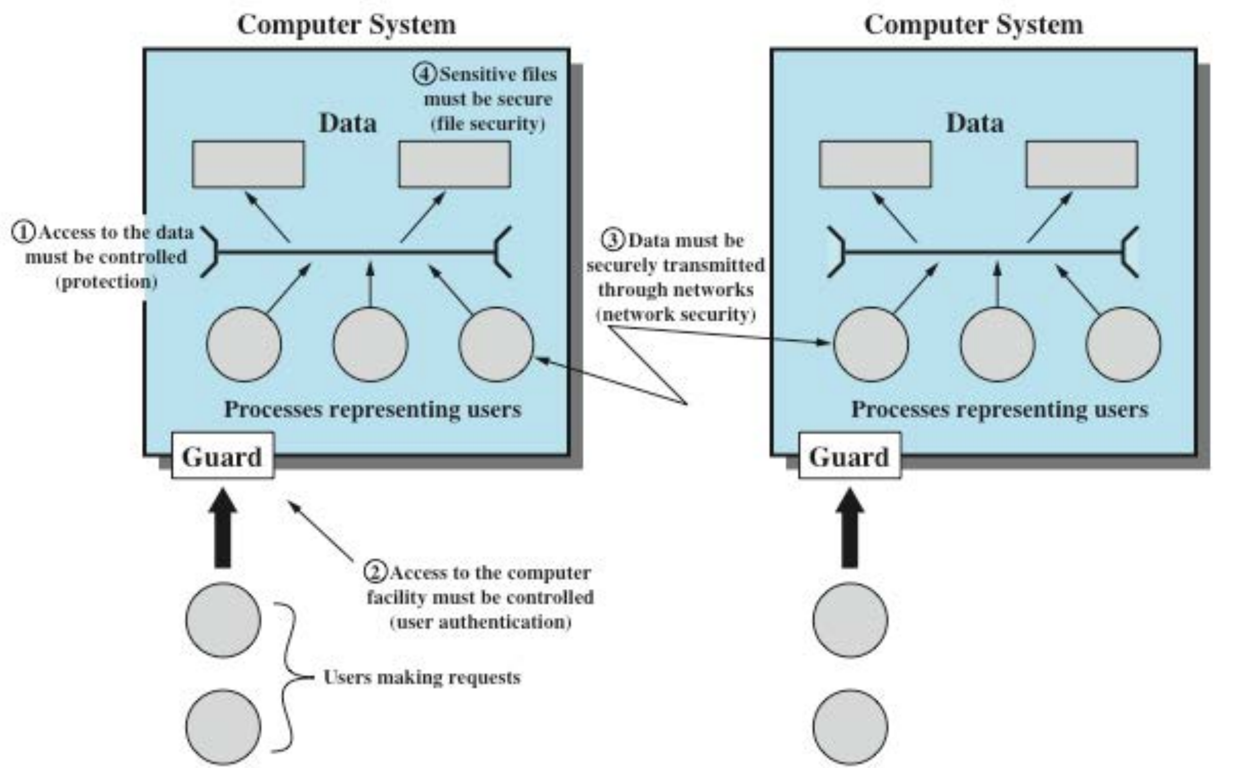
The loss could be expected to have a **serious adverse**

HIGH

The loss could be expected to have a **severe or catastrophic adverse effect**

Architecture of a Computer Systems from a Security Perspective

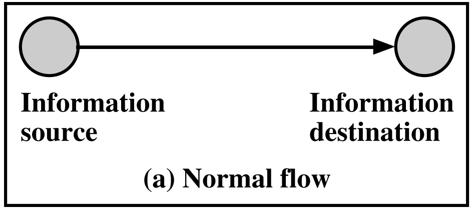
W. Stallings


Computer System
Computer System
④Sensitive files must be secure (file security)
Data
Data
①Access to the data must be controlled (protection)
③Data must be securely transmitted through networks (network security)
Processes representing users
Processes representing users
Guard
Guard
②Access to the computer facility must be controlled (user authentication)
Users making requests

Co-funded by the Erasmus+ Programme of the European Union

Threat Model

Any action performed by a computer system can be **modelled** as an **information flow** from a source to a sink

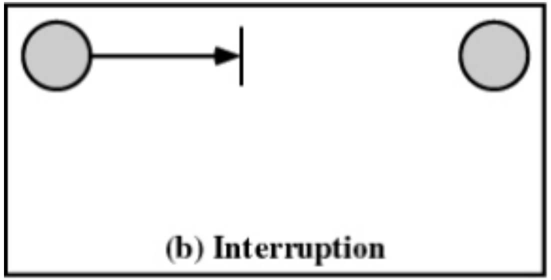


(a) Normal flow

- Computer attacks aim at modifying the information flow
- Four main categories of attacks can be defined

1. Interruption

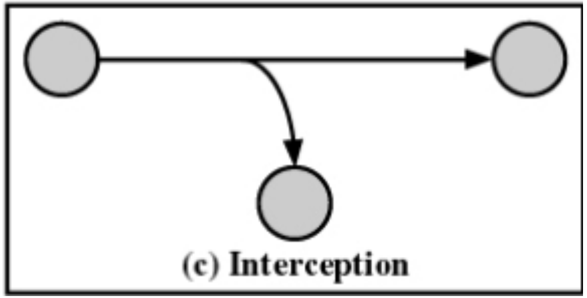
- An asset is destroyed or disabled
  - hardware damages
  - interruption of communication lines
  - exhausting all the available resources
  - disabling core services


(b) Interruption

- This kind of attack is called Denial of Service (DoS) as the attack threats the **availability**

2. Interception

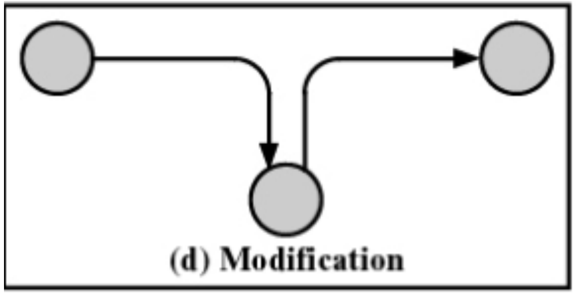
A third unauthorised party gain access to information flows



This attack is a threat to **confidentiality**

3. Modification

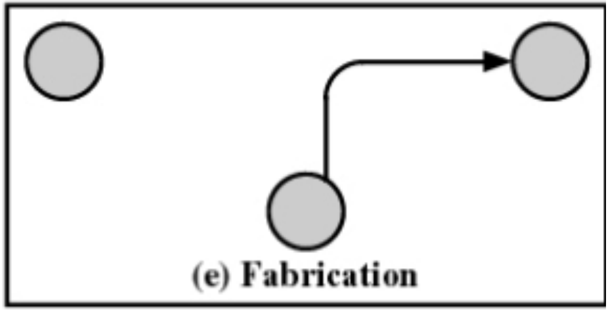
- A third unauthorised party
  - intercepts the information flow by *spoofing* the identity of the destination (this is an attack per se)
  - sends a *modified* flow to the destination



This attack is a threat to **confidentiality** and **integrity**

4. Fabrication

A third unauthorised party produces information flows by *spoofing* the identity of the source



This attack is a threat to **authenticity**

Summary

	Availability	Confidentiality	Integrity/Authenticity
Hardware	Equipment is stolen or disabled, thus denying the device		
Software	Programs are deleted, denying access to users	An unauthorised copy of software is made	A working program is modified, either to cause it to fail during execution ot to cause it to do some unintended task
Data	Files are deleted, denying access to users	An unauthorised read of data is performed. An analysis of statistical data reveals underlying data	Existing files are modified or new files are fabricated
Communication lines	Messages are destroyed or deleted. Communication lines or networks are rendered unavailable	Messages are read. The traffic pattern of messages is observed	Messages are modified, delayed, reordered, or duplicated. False messages are fabricated

Co-funded by the
Erasmus+ Programme
of the European Union

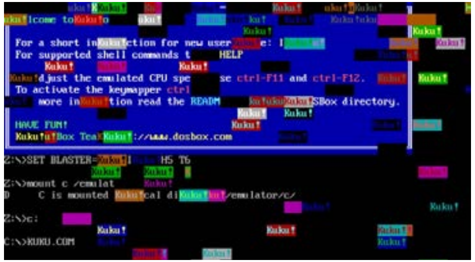
Threat consequences (RFC2828)

Threat Consequence	Threat Action (Attack)
Unauthorized Disclosure An entity gains access to data for which the entity is not authorized	**Exposure**: Sensitive data are directly released to an unauthorized entity. **Interception**: An unauthorized entity directly accesses sensitive data traveling between authorized sources and destinations. **Inference**: A unauthorized entity indirectly accesses sensitive data (but not necessarily the data contained in the communication) by reasoning from characteristics or byproducts of communications. **Intrusion**: An unauthorized entity gains access to sensitive data by circumventing a system's security protections.
Deception An authorized entity receiving false data and believing it to be true.	**Masquerade**: An unauthorized entity gains access to a system or performs a malicious act by posing as an authorized entity. **Falsification**: False data deceive an authorized entity. **Repudiation**: An entity deceives another by falsely denying responsibility for an act.

Threat consequences (RFC2828)

Threat Consequence	Threat Action (Attack)
Disruption The correct operation of system services and functions are interrupted or prevented.	**Incapacitation**: Prevents or interrupts system operation by disabling a system component. **Corruption**: Undesirably alters system operation by adversely modifying system functions or data. **Obstruction**: A threat action that interrupts delivery of system services by hindering system operation.
Usurpation Control of system services or functions by an unauthorized entity.	**Misappropriation**: An entity assumes unauthorized logical or physical control of a system resource. **Misuse**: Causes a system component to perform a function or service that is detrimental to system security.

Co-funded by the
Erasmus+ Programme
of the European Union

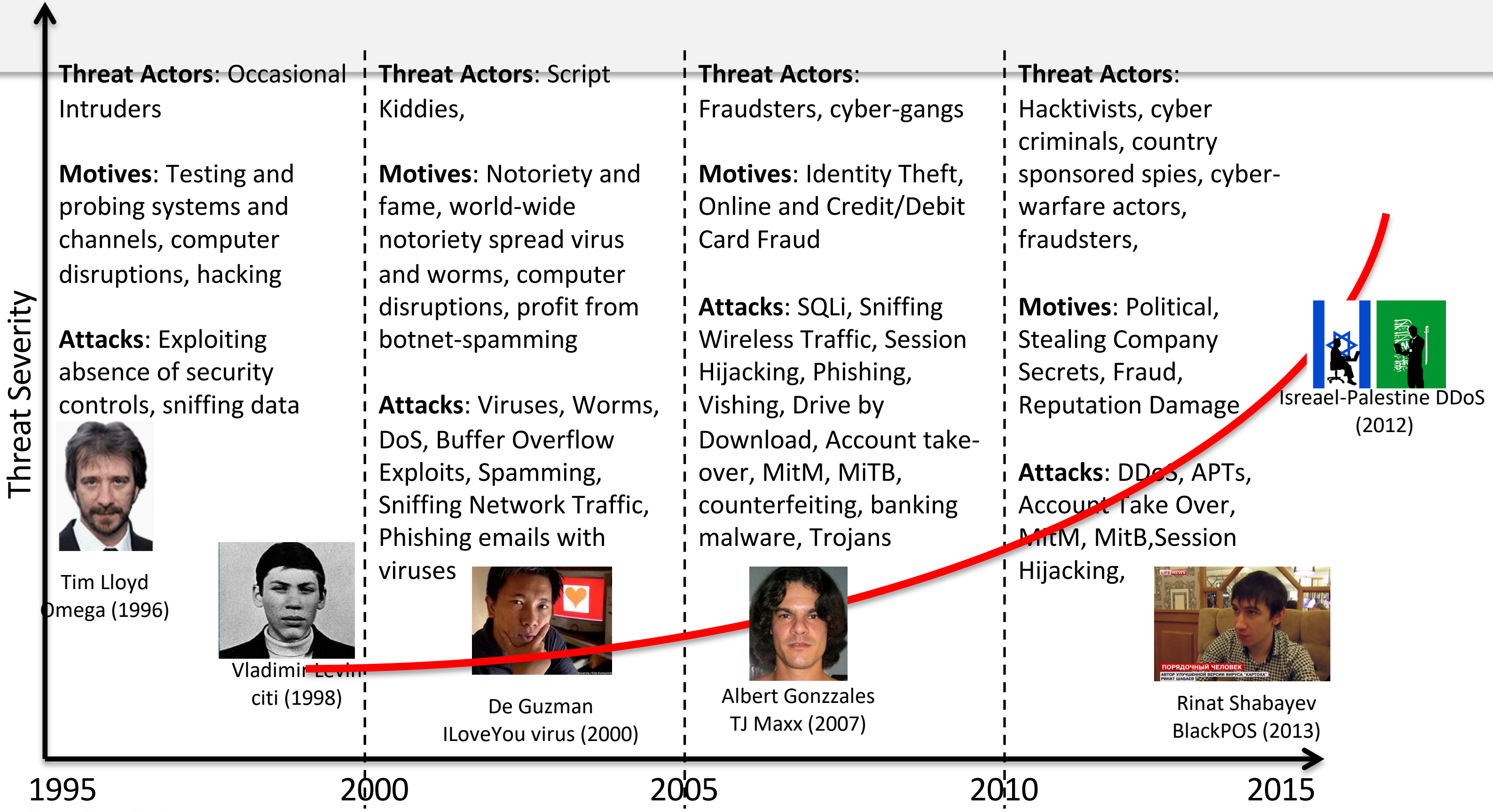


```
         uku!KKuku!                      Kuku!      uku!Kuku!
uku!lcome toKuku!o     uku!             ku!        Kuku!uku
                                     Kuku!                    Kuku!
 For a short inKuku!ction for new user     e: I                Kuku!
 For supported shell commands t     HELP
      Kuku!            Kuku!
Kuku!djust the emulated CPU spe     se ctrl-F11 and ctrl-F12.     Kuku!
 To activate the keymapper ctrl
     more inKuku!tion read the READM     ku!ukuKuku!SBox directory.
                                      Kuku!   Kuku!
 HAVE FUN!                           Kuku!
 Kuku!u!Box TeaKuku!://www.dosbox.com

Z:\>SET BLASTER=Kuku!I      H5 T6
                 Kuku!   Kuku!
Z:\>mount c /emulat          Kuku!
D     C is mounted Kuku!cal diKuku!ku!/emulator/c/
                                                           Kuku!
Z:\>c:
                  Kuku!                      Kuku!          Kuku!
C:\>KUKU.COM                                                Kuku!
```

History of Computer Attacks

Evolution of attacker's motivations



Threat Actors: Occasional Intruders

Motives: Testing and probing systems and channels, computer disruptions, hacking

Attacks: Exploiting absence of security controls, sniffing data

Tim Lloyd
Omega (1996)

Vladimir Levin
citi (1998)

Threat Actors: Script Kiddies,

Motives: Notoriety and fame, world-wide notoriety spread virus and worms, computer disruptions, profit from botnet-spamming

Attacks: Viruses, Worms, DoS, Buffer Overflow Exploits, Spamming, Sniffing Network Traffic, Phishing emails with viruses

De Guzman
ILoveYou virus (2000)

Threat Actors: Fraudsters, cyber-gangs

Motives: Identity Theft, Online and Credit/Debit Card Fraud

Attacks: SQLi, Sniffing Wireless Traffic, Session Hijacking, Phishing, Vishing, Drive by Download, Account take-over, MitM, MiTB, counterfeiting, banking malware, Trojans

Albert Gonzzales
TJ Maxx (2007)

Threat Actors: Hacktivists, cyber criminals, country sponsored spies, cyber-warfare actors, fraudsters,

Motives: Political, Stealing Company Secrets, Fraud, Reputation Damage

Attacks: DDoS, APTs, Account Take Over, MitM, MitB,Session Hijacking,

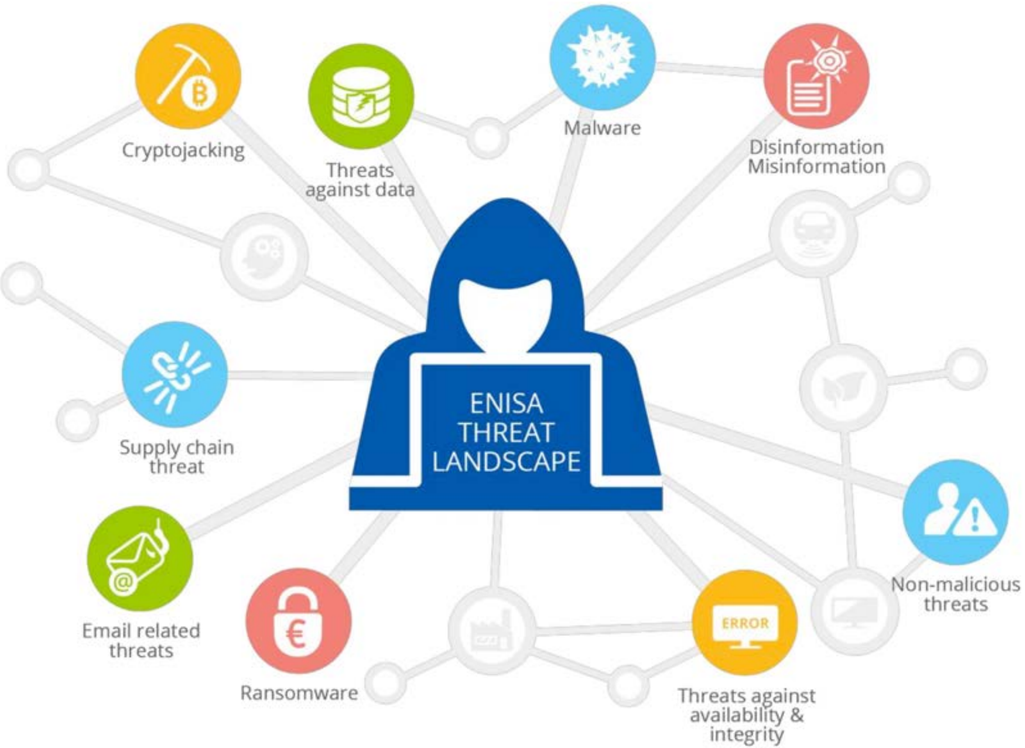
Rinat Shabayev
BlackPOS (2013)

Isreael-Palestine DDoS
(2012)

Credits: Marco Morana

Threat Landscape 2021

https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021

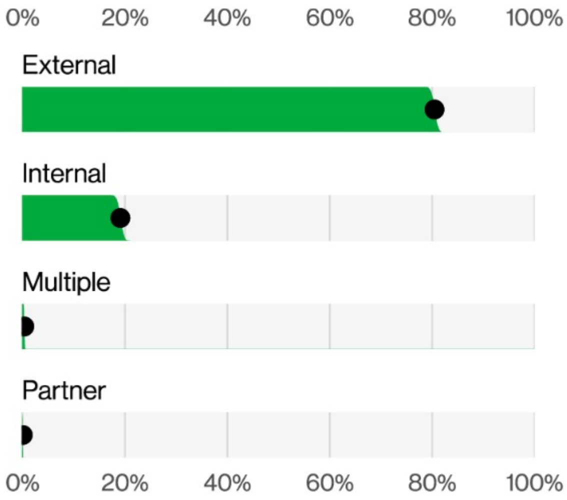


THREAT ACTOR TRENDS

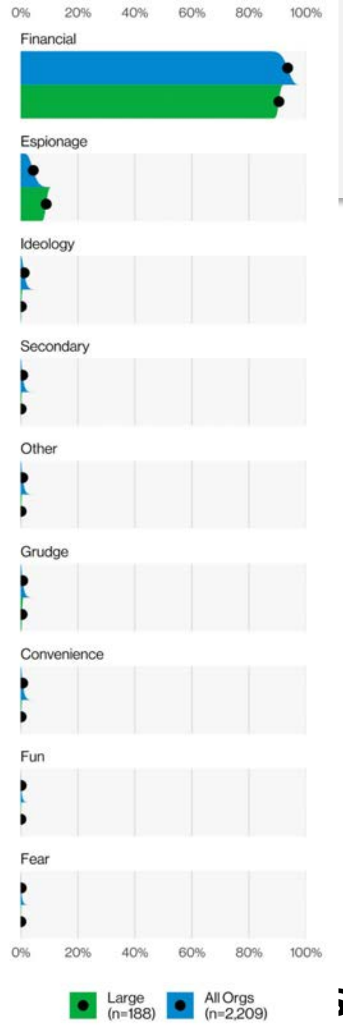
- State-sponsored actors
- Cybercrime Actors
- Hacker-for-hire actors
- Hacktivists

Threat Actors and Their Motives

ACTORS IN BREACHES



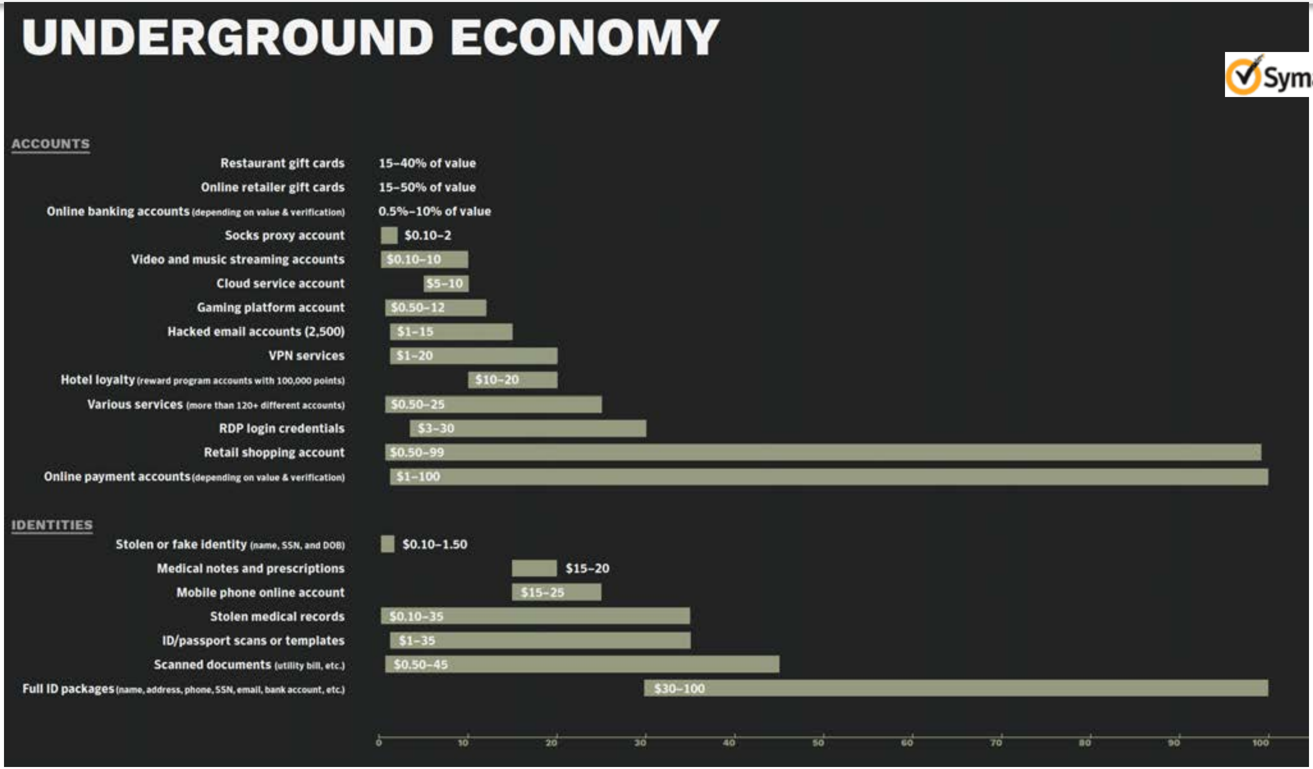
MOTIVES IN EXTERNAL ACTORS



Verizon – 2022DBIR (Data Breach Investigations Report)

Co-funded by the Erasmus+ Programme of the European Union

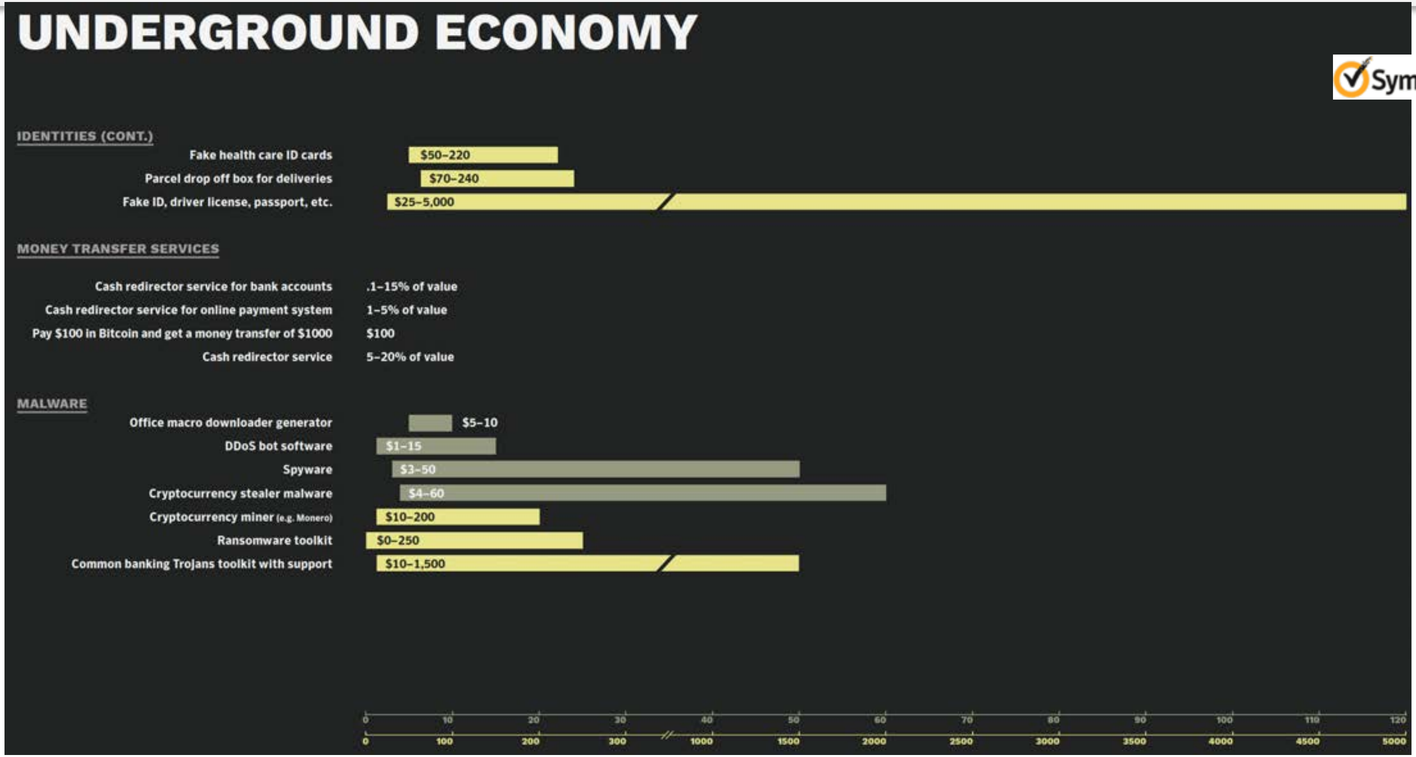
Economic motivations



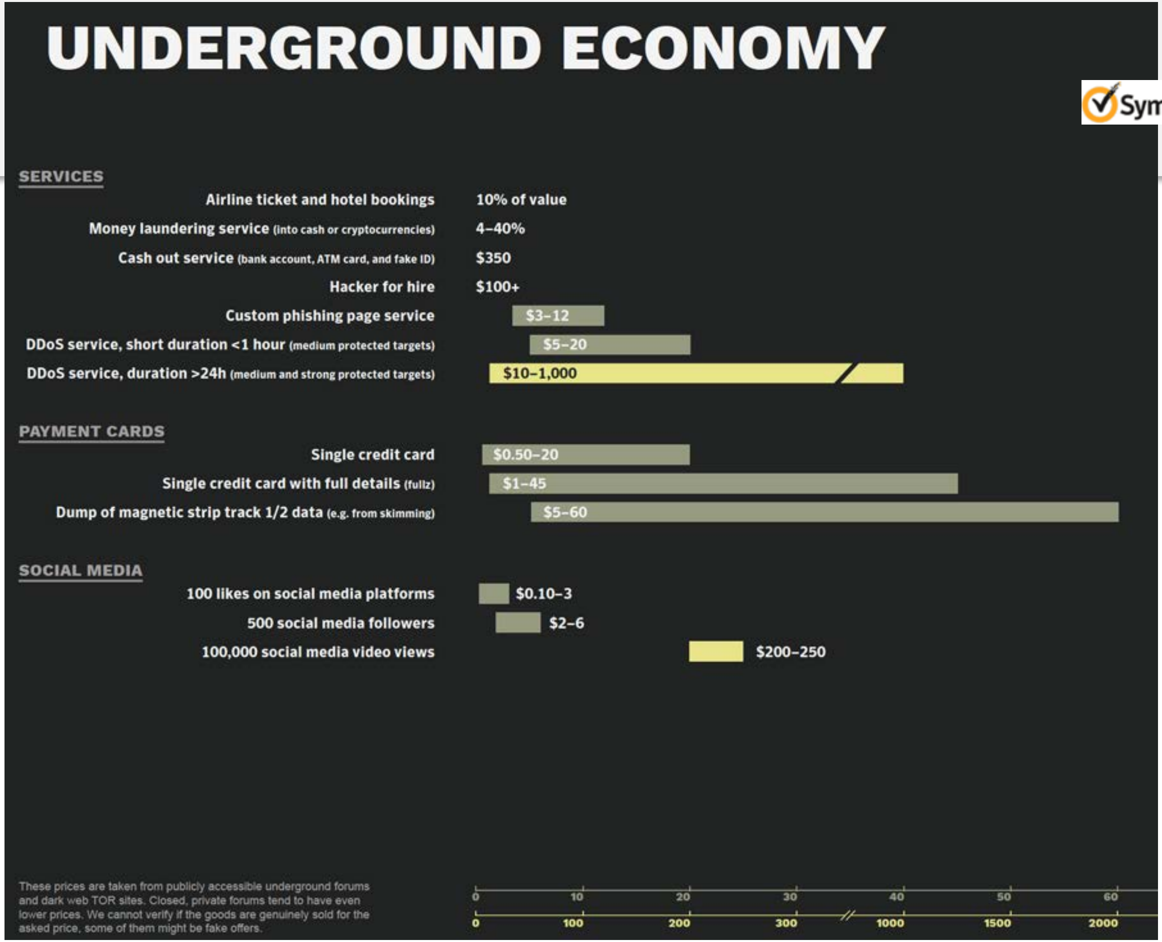
Symantec.

ISTR
Internet Security Threat Report
Volume 24 | February 2019

Economic motivations



Co-funded by the
Erasmus+ Programme
of the European Union



UNDERGROUND ECONOMY

SERVICES

Airline ticket and hotel bookings	10% of value
Money laundering service (into cash or cryptocurrencies)	4–40%
Cash out service (bank account, ATM card, and fake ID)	$350
Hacker for hire	$100+
Custom phishing page service	$3–12
DDoS service, short duration <1 hour (medium protected targets)	$5–20
DDoS service, duration >24h (medium and strong protected targets)	$10–1,000

PAYMENT CARDS

Single credit card	$0.50–20
Single credit card with full details (fullz)	$1–45
Dump of magnetic strip track 1/2 data (e.g. from skimming)	$5–60

SOCIAL MEDIA

100 likes on social media platforms	$0.10–3
500 social media followers	$2–6
100,000 social media video views	$200–250

These prices are taken from publicly accessible underground forums and dark web TOR sites. Closed, private forums tend to have even lower prices. We cannot verify if the goods are genuinely sold for the asked price, some of them might be fake offers.

ISTR

Symantec.

et Security Threat Report

ume 24 | February 2019

SECURE CODING

Co-funded by the
Erasmus+ Programme
of the European Union

Security Failures and Vulnerabilities

- **Software Security** is defined by the requirements in terms of **Confidentiality**, **Integrity** and **Availability**.

- A **Security Failure** is a scenario where the software does not achieve its **security objective.**

- A **Vulnerability** is the underlying cause of a security failure.

- There are well known classes of **implementation weaknesses** that an attacker can trigger to cause a substantial disruption in the behaviour of the software, thus breaking whatever security objective has been defined.

Writing Safe Program Code

- High-level languages are typically compiled and linked into machine code which is then directly executed by the target processor

- Security issues
  - Correct algorithm implementation
  - Correct machine instructions for algorithm
  - Valid manipulation of data

Co-funded by the Erasmus+ Programme of the European Union

Correct Algorithm Implementation

- Failures in software development
  - The algorithm may **not correctly handle all problem variants**
  - Consequently, the resulting program could be exploited

- Another type of failure is when the programmers deliberately include **additional code to help test and debug** it
  - often code remains in production release of a program and **could inappropriately release information**
  - **may permit a user to bypass security checks** and perform actions they would not otherwise be allowed to perform

Co-funded by the Erasmus+ Programme of the European Union

Ensuring Machine Language Corresponds to Algorithm

- Programmers often **assume that the compiler** or interpreter **generates** or executes **code** that validly **implements** the language **statements**

- Requires comparing machine code with original source
  - slow and difficult

- Development of computer systems with very **high assurance level** is the one area where this level of checking is required

Correct Data Interpretation

- **Data stored as bits/bytes in computer**
  - Grouped as words or longwords
  - Accessed and manipulated in memory or copied into processor registers before being used
  - Interpretation depends on machine instruction executed

- Different languages **provide different capabilities for restricting and validating interpretation of data** in variables
  - Strongly typed languages are more limited, but safer
  - Other languages allow more liberal interpretation of data and permit program code to explicitly change their interpretation

Correct Use of Memory

- Dynamic memory allocation
  - Unknown amounts of data
  - Allocated when needed, released when done
  - Used to manipulate memory leak
  - Steady reduction in memory available on the heap to the point where it is completely exhausted

- Older languages have no explicit support for dynamic memory allocation
  - Use standard library routines to allocate and release memory

- Modern languages handle automatically

Use of the Least Privilege Principle

- **Least privilege**
  - Run programs with least privilege needed to complete their function

- Determine appropriate user and group **privileges required**
  - Decide whether to grant extra user or just group privileges

- Ensure that privileged programs has a **limited scope**

- Privilege **escalation**
  - When attackers can gain high privileges by exploiting flaws in privilege management

Co-funded by the
Erasmus+ Programme
of the European Union

Management of Temporary Files

- Many programs use temporary files
- They are often stored in common, **shared** system areas
- Must be unique, not accessed by others
- Commonly the **name** is created using the process ID
  - Unique, but predictable
  - Attacker might guess and attempt to create own file between program checking and creating
- Secure **temporary file** creation and use requires the use of random names

CWE – common weakness enumeration

http://cwe.mitre.org

- A Community-Developed List of Software & Hardware Weakness Types.
- The current version is 4.8 and 927 weaknesses are listed
- They are organised as a hierarchy of classes and subclasses.
- Three views are available:
  - by Software Development
  - by Hardware Design
  - by Research Concepts

2021 CWE Top 25 Most Dangerous Weaknesses

Rank	ID	Name
[1]	CWE-787	Out-of-bounds Write
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[3]	CWE-125	Out-of-bounds Read
[4]	CWE-20	Improper Input Validation
[5]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[7]	CWE-416	Use After Free
[8]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
[9]	CWE-352	Cross-Site Request Forgery (CSRF)
[10]	CWE-434	Unrestricted Upload of File with Dangerous Type
[11]	CWE-306	Missing Authentication for Critical Function
[12]	CWE-190	Integer Overflow or Wraparound
[13]	CWE-502	Deserialization of Untrusted Data
[14]	CWE-287	Improper Authentication
[15]	CWE-476	NULL Pointer Dereference
[16]	CWE-798	Use of Hard-coded Credentials
[17]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer
[18]	CWE-862	Missing Authorization
[19]	CWE-276	Incorrect Default Permissions
[20]	CWE-200	Exposure of Sensitive Information to an Unauthorized Actor
[21]	CWE-522	Insufficiently Protected Credentials
[22]	CWE-732	Incorrect Permission Assignment for Critical Resource
[23]	CWE-611	Improper Restriction of XML External Entity Reference
[24]	CWE-918	Server-Side Request Forgery (SSRF)
[25]	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')

Finding Vulnerabilities

- Any computer program or protocol may contain **weaknesses**
  - originating from the programming **language**
  - causing unexpected outputs from unexpected **inputs**
  - that allow for the arbitrary modification of the **program flow**
- The maliciousness depends on the **context**
  - input values, API usage, etc. cannot be considered malicious per se but the maliciousness is related to the context and the related consequences
  - **ambiguity** and **misinterpretation** may occur when data and instructions are passed from one component to another
- The detection of weaknesses is a very difficult task
  - Requires deep knowledge of languages and protocols
  - Multiple information sources (network traffic, application logs, system calls, etc.)
  - Static or dynamic analysis

Top 15 Routinely Exploited Vulnerabilities in 2021

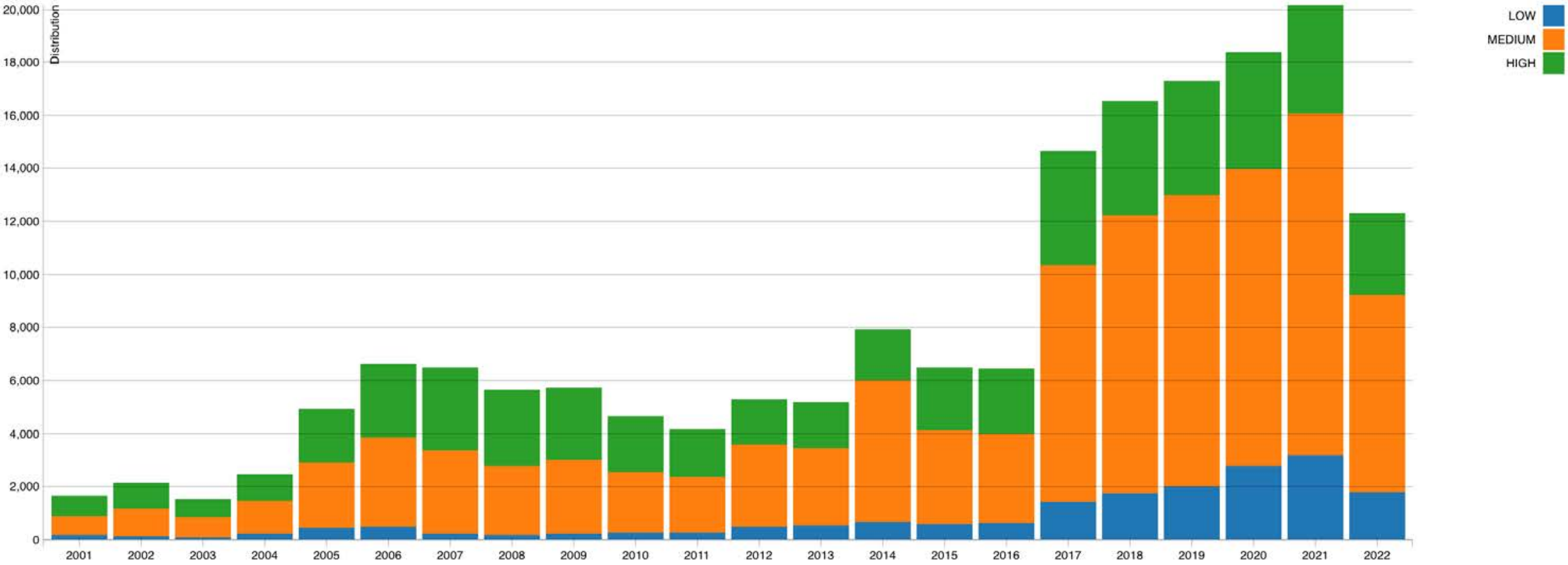
https://www.cisa.gov/uscert/ncas/alerts/aa22-117a - April 27, 2022 - US Cybersecurity & Infrastructure Security Agency

CVE	Vulnerability Name	Vendor and Product	Type
CVE-2021-44228	Log4Shell	Apache Log4j	Remote code execution (RCE)
CVE-2021-40539		Zoho ManageEngine AD SelfService Plus	RCE
CVE-2021-34523	ProxyShell	Microsoft Exchange Server	Elevation of privilege
CVE-2021-34473	ProxyShell	Microsoft Exchange Server	RCE
CVE-2021-31207	ProxyShell	Microsoft Exchange Server	Security feature bypass
CVE-2021-27065	ProxyLogon	Microsoft Exchange Server	RCE
CVE-2021-26858	ProxyLogon	Microsoft Exchange Server	RCE
CVE-2021-26857	ProxyLogon	Microsoft Exchange Server	RCE
CVE-2021-26855	ProxyLogon	Microsoft Exchange Server	RCE
CVE-2021-26084		Atlassian Confluence Server and Data Center	Arbitrary code execution
CVE-2021-21972		VMware vSphere Client	RCE
CVE-2020-1472	ZeroLogon	Microsoft Netlogon Remote Protocol (MS-NRPC)	Elevation of privilege
CVE-2020-0688		Microsoft Exchange Server	RCE
CVE-2019-11510		Pulse Secure Pulse Connect Secure	Arbitrary file reading
CVE-2018-13379		Fortinet FortiOS and FortiProxy	Path traversal

Co-funded by the Erasmus+ Programme of the European Union

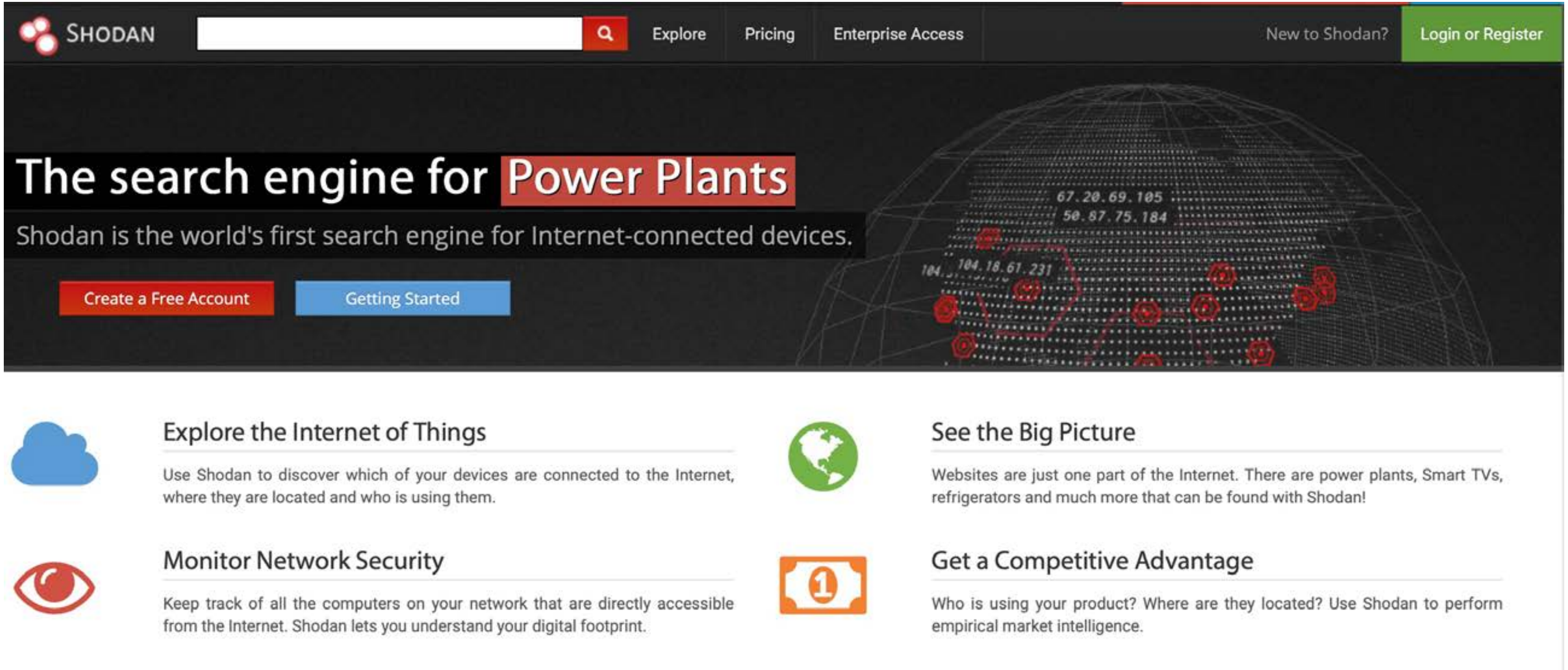
Critical vulnerabilities

CVSS - Common Vulnerabilities Scoring System



https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cvss-severity-distribution-over-time

The search engine for exposed devices



The search engine for Power Plants

Shodan is the world's first search engine for Internet-connected devices.

Explore the Internet of Things

Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.

See the Big Picture

Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan!

Monitor Network Security

Keep track of all the computers on your network that are directly accessible from the Internet. Shodan lets you understand your digital footprint.

Get a Competitive Advantage

Who is using your product? Where are they located? Use Shodan to perform empirical market intelligence.

Co-funded by the
Erasmus+ Programme
of the European Union

Authentication

Co-funded by the
Erasmus+ Programme
of the European Union

Authentication and Authorization

- **AUTHENTICATION**
  verification of a person (or process)
  - the act of proving the identity of a user, that she is who she claims to be

  The process of establishing confidence in user identities that are presented electronically to an information system
  NIST SP 800-63-3

- **AUTHORIZATION**
  verification of the **privileges** of a user on the resources he has access to
  - Access matrix

Co-funded by the
Erasmus+ Programme
of the European Union

NIST SP 800-63-3

- Identity proofing establishes that a subject is **who they claim to be**.
- Digital authentication is the process of determining the **validity** of **one or more authenticators** used to claim a digital identity.
- Successful authentication provides **reasonable risk-based assurances** that the subject accessing the service today is the same as that which previously accessed the service.
- **Digital identity** is the **unique representation of a subject** engaged in an online transaction.
- A digital identity is **always unique in the context** of a digital service, but **does not necessarily need to uniquely identify the subject in all contexts**.
  In other words, accessing a digital service may not mean that the subject's real-life identity is known

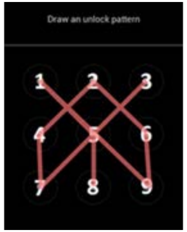
Co-funded by the
Erasmus+ Programme
of the European Union

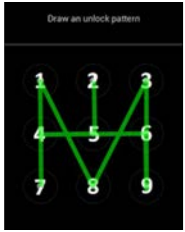
Authentication mechanisms

- **WHAT YOU ARE**
  biometrics (fingerprints, face, iris, etc.)

- **WHAT YOU HAVE**
  card, keys, etc.

- **WHAT YOU KNOW**
  a secret, such as a password, security question, PIN, etc.





- **Multifactor authentication** (**MFA**) when multiple methods are used at the same time
  - e.g., card + PIN

Co-funded by the
Erasmus+ Programme
of the European Union

Attacks against authentication systems

Attack type	Authentication Factor	Example	Mitigation
Client Attack	Password	Guessing, trial & error	Password complexity, limited attempts
	Token	Exhaustive search	Limited attempts
	Biometrics	False match	Biometric complexity, *liveness detection*
Host Attack	Password	Password theft	Cryptography, direct attack protection
	Token	Passcode theft	1-time Passcode
	Biometrics	Template theft	Capture-device authentication

Co-funded by the
Erasmus+ Programme
of the European Union

Attacks against authentication systems

Attack type	Authentication Factor	Example	Mitigation
Eavesdropping, theft, copy	Password	Shoulder surfing, keylogger	Personal password storage, weak password check, multi-factor authentication
	Token	Theft, clone, counterfeit	Tamper-resistant token, multi-factor authentication
	Biometrics	Fake biometric traits	Copy detection at the physical device, liveness detection

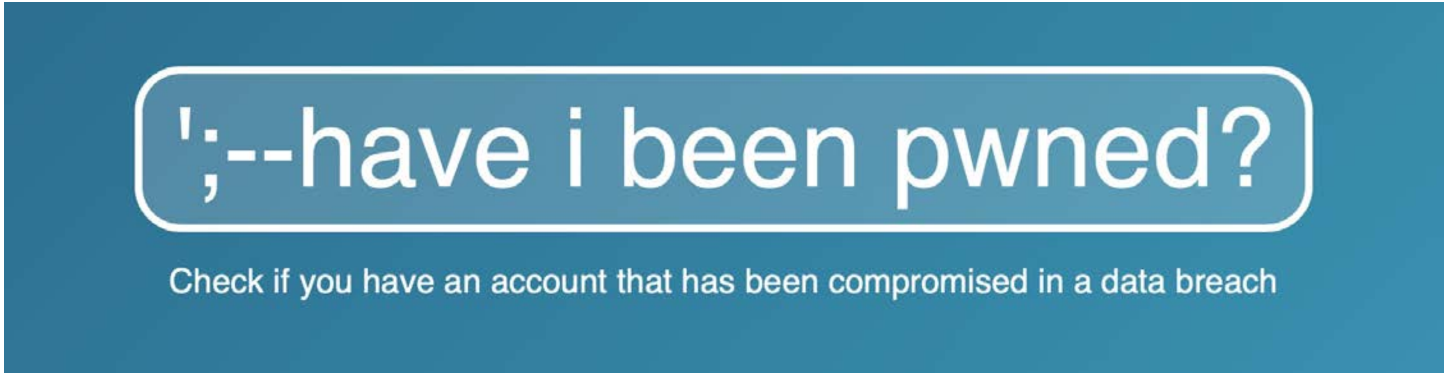
Co-funded by the Erasmus+ Programme of the European Union

Attacks against authentication systems

Attack type	Authentication Factor	Example	Mitigation
Replay	Password, Token, Biometrics	Replay stolen password, passcode, template	challenge-response, OTP
Trojan Horse	Password, Token, Biometrics	Rogue client or capture devices	Trusted Locations. Trusted Devices
Denial of Service	Password, Token, Biometrics	Lockout by multiple failed authentication attempts	Multi-factor authentication with physical devices

Co-funded by the Erasmus+ Programme of the European Union

Have I Been Pwned?



';--have i been pwned?

Check if you have an account that has been compromised in a data breach

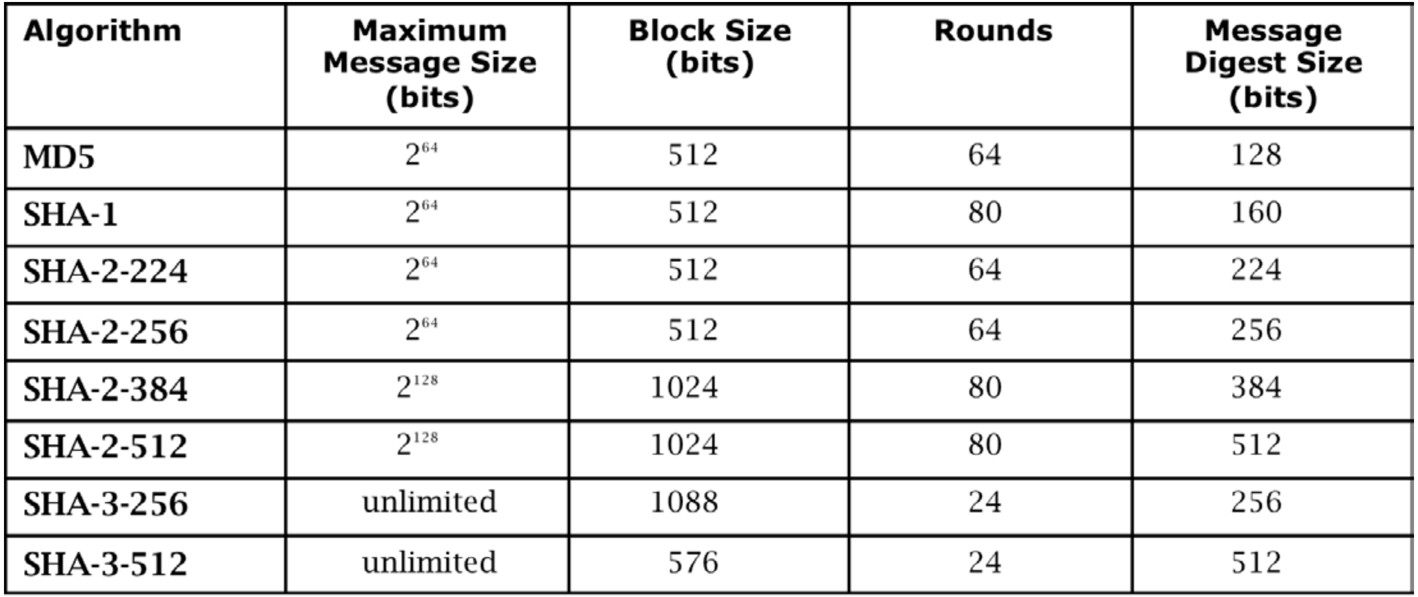
https://haveibeenpwned.com

Password encryption

- Passwords are never stored or checked in clear, **password hashes** are used instead.
- **One-way hash functions** are cryptographic functions with multiple uses
  - They are used in **integrity** checking
  - They are used in **authentication**
  - They are used in **communications protocols**
- They are based **on *one-way* random functions**. Given an input sequence of bytes of arbitrary length, hash functions produce a **fixed-length** string
  - It is infeasible to **infer the input** given a **hash** value
  - it is infeasible to find a pair of inputs that produce the same hash
- There are **dictionaries** of hashes that match with the corresponding plaintext
  - hashes.com, crackstation.net

Co-funded by the
Erasmus+ Programme
of the European Union

Properties of Current Hash Standards



Algorithm	Maximum Message Size (bits)	Block Size (bits)	Rounds	Message Digest Size (bits)
MD5	2^{64}	512	64	128
SHA-1	2^{64}	512	80	160
SHA-2-224	2^{64}	512	64	224
SHA-2-256	2^{64}	512	64	256
SHA-2-384	2^{128}	1024	80	384
SHA-2-512	2^{128}	1024	80	512
SHA-3-256	unlimited	1088	24	256
SHA-3-512	unlimited	576	24	512

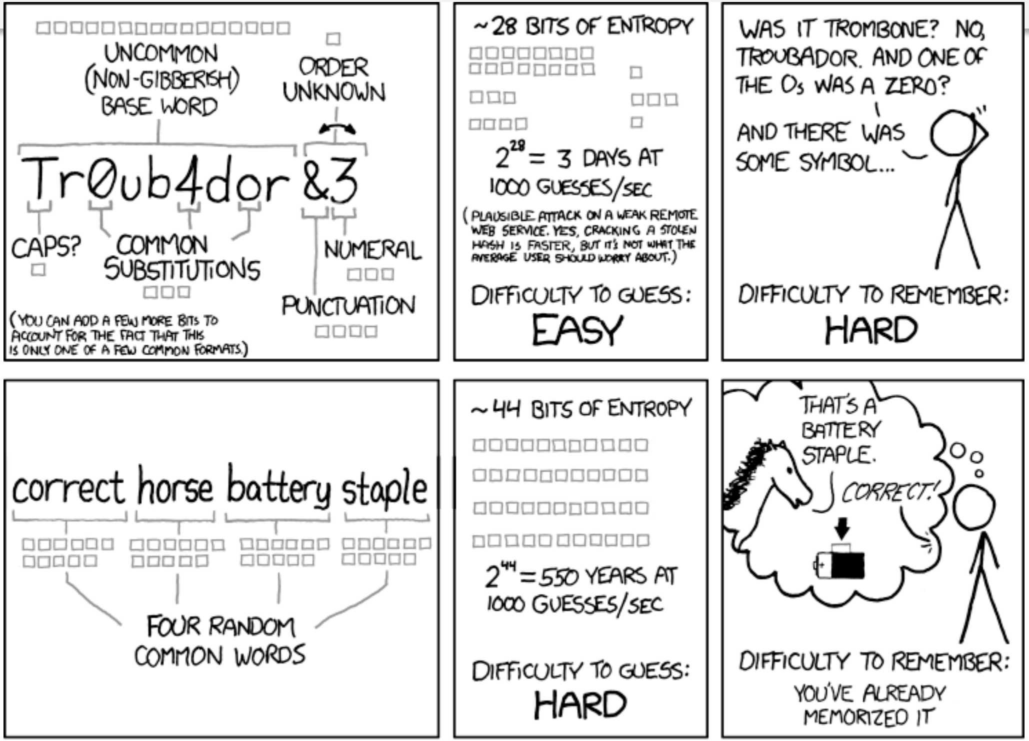
Weak passwords

- Guessed though
  - Dictionary Attack
  - Inference (e.g., social engineering, open source intelligence)
- Brute Force
- Defeating Encryption
- Popular algorithms
  - John the Ripper password cracker http://www.openwall.com/john/
  - Hashcat https://hashcat.net/hashcat/
- Hashes.com
  - repository of leaked hashed password with the recovered plaintext

Co-funded by the Erasmus+ Programme of the European Union

Passphrases

Credit: Randall Munroe, xkcd.com, CC 2.5


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^{28} = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^{44}=550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

NIST SP 800-63

https://www.nist.gov/blogs/taking-measure/easy-ways-build-better-p5w0rd

Password Managers

https://pages.nist.gov/800-63-FAQ/#q-b12

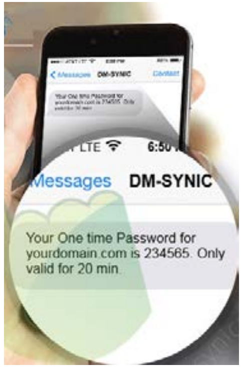
- One solution to
  - set difficult-to-guess password
  - avoid storing strong passwords in unsecure archives such as paper notes, unencrypted files, etc.

is using password manager applications

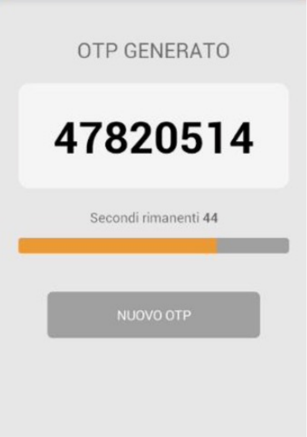
- you need to set only **one strong master password** for the application, so that you have to remember just 1 password
- the application **generates random** strong passwords
- the **password archive** is **encrypted** and stored in your device and/or in a cloud service

One-Time Password

- OTP
  A random password is generated by the server for one-time use (very short time-to-live)
  - either the client runs the same algorithm and generates the same random password
  - or the OTP is sent “out-of-band” (i.e., via SMS)



Your One time Password for yourdomain.com is 234565. Only valid for 20 min.



47820514

Secondi rimanenti 44

Challenge-response

- During the enrolment phase, the user is asked to provide more than 1 secret
  - Secret questions
  - Multiple fingerprints
  - Long codes

- At access time, the system chooses at random one or more *questions*

Biometrics

- More difficult to spoof

- Problem: user acceptance (intrusiveness)
- Need for advanced (expensive) sensors and algorithms for high accuracy

Co-funded by the
Erasmus+ Programme
of the European Union

Multi-Factor Authentication (MFA)

- Mitigate the risk of one-factor authentication

- Two or more factors *simultaneously*
  - e.g., card + PIN, card + biometrics

- Two or more factors in cascade
  - e.g., PIN, then OTP or smartphone

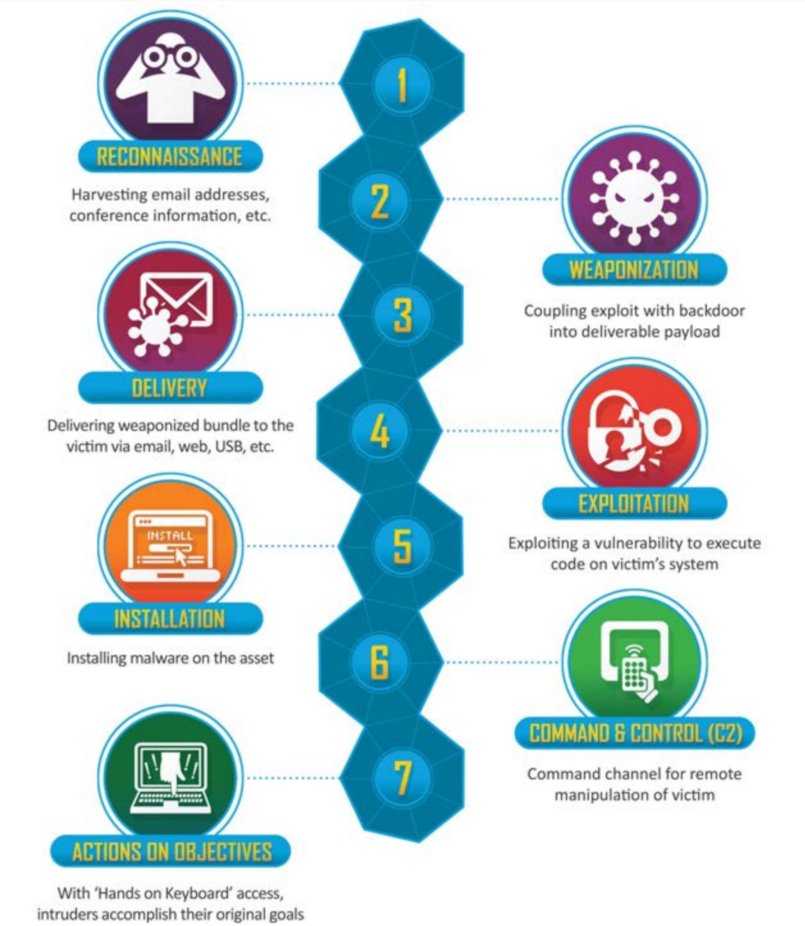
Cyber Threat Intelligence

Cyber Kill Chain

Released by **Lockheed Martin** in **2011**.

The rationale is that by understanding each of these stages, defenders can better identify and stop attackers at each of the respective stages.

Since 2011, various versions of the "Cyber Kill Chain" have been released

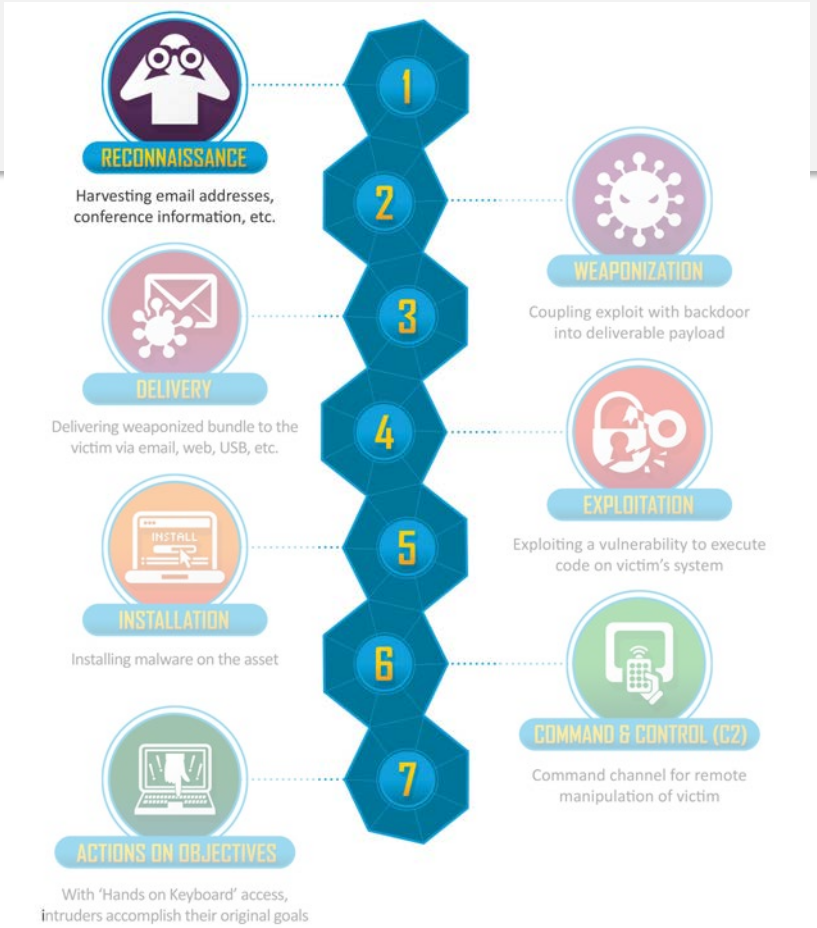

1
RECONNAISSANCE
Harvesting email addresses, conference information, etc.
2
WEAPONIZATION
Coupling exploit with backdoor into deliverable payload
3
DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.
4
EXPLOITATION
Exploiting a vulnerability to execute code on victim's system
5
INSTALLATION
INSTALL
Installing malware on the asset
6
COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim
7
ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Co-funded by the Erasmus+ Programme of the European Union

Cyber Kill Chain

Harvesting email addresses, conference information, etc.



1 RECONNAISSANCE: Harvesting email addresses, conference information, etc.

2 WEAPONIZATION: Coupling exploit with backdoor into deliverable payload

3 DELIVERY: Delivering weaponized bundle to the victim via email, web, USB, etc.

4 EXPLOITATION: Exploiting a vulnerability to execute code on victim's system

5 INSTALLATION: Installing malware on the asset

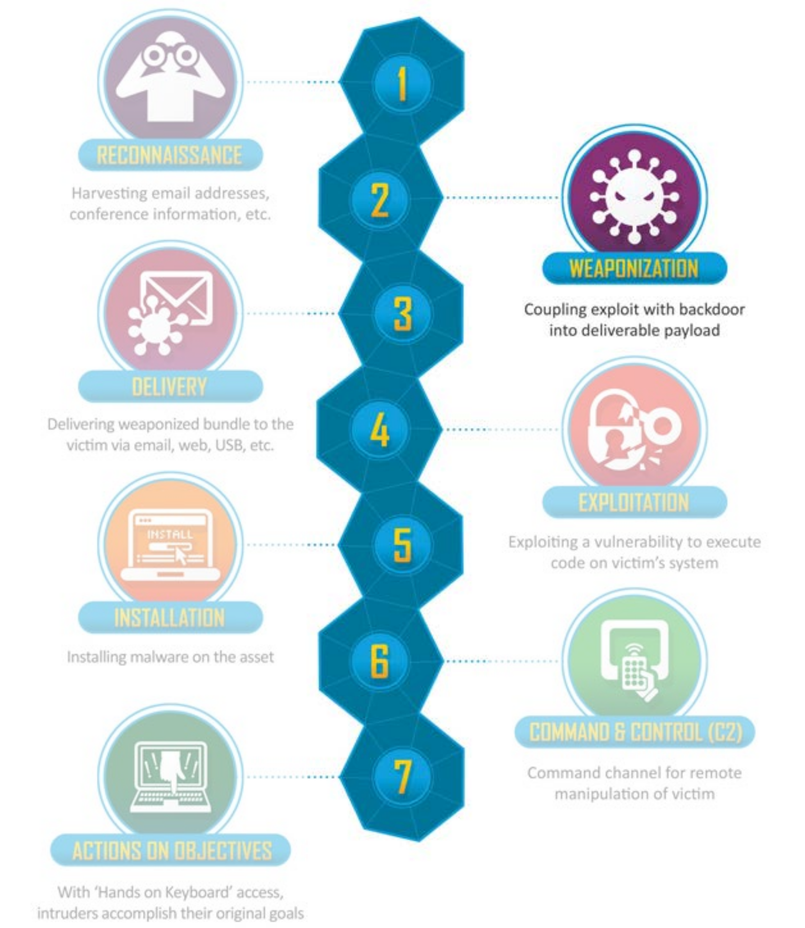
6 COMMAND & CONTROL (C2): Command channel for remote manipulation of victim

7 ACTIONS ON OBJECTIVES: With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Cyber Kill Chain

Coupling exploit with backdoor into deliverable payload



1 RECONNAISSANCE
Harvesting email addresses, conference information, etc.

2 WEAPONIZATION
Coupling exploit with backdoor into deliverable payload

3 DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.

4 EXPLOITATION
Exploiting a vulnerability to execute code on victim's system

5 INSTALLATION
Installing malware on the asset

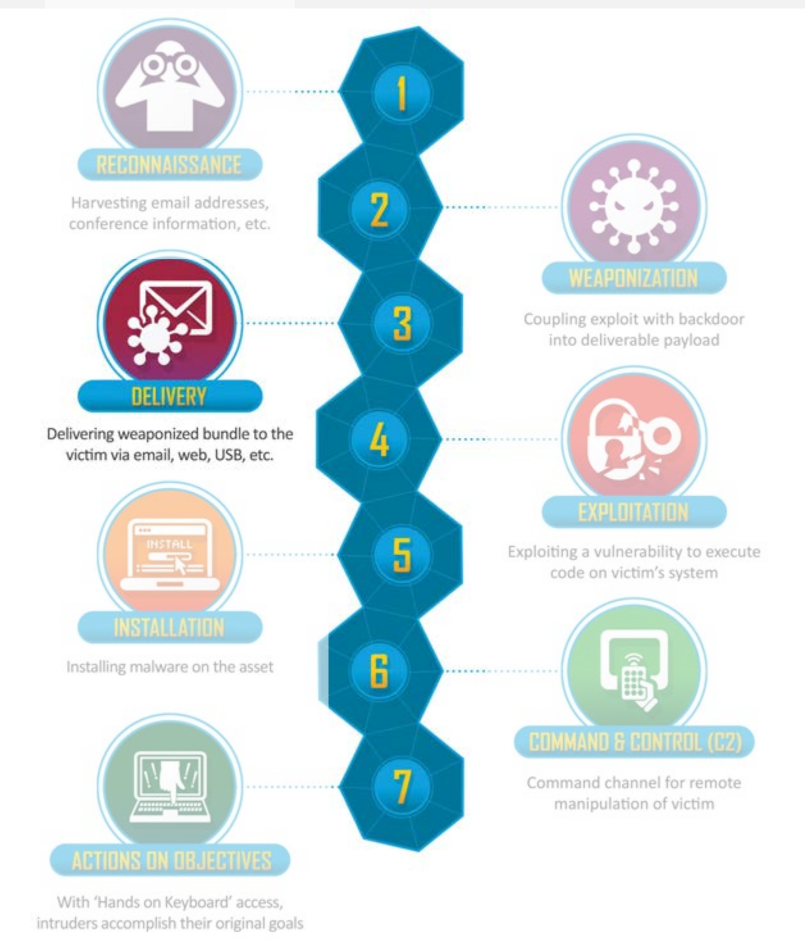
6 COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim

7 ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Cyber Kill Chain

Delivering weaponized bundle to the victim via email, web, USB, etc.



1 RECONNAISSANCE

Harvesting email addresses, conference information, etc.

2 WEAPONIZATION

Coupling exploit with backdoor into deliverable payload

3 DELIVERY

Delivering weaponized bundle to the victim via email, web, USB, etc.

4 EXPLOITATION

Exploiting a vulnerability to execute code on victim's system

5 INSTALLATION

Installing malware on the asset

6 COMMAND & CONTROL (C2)

Command channel for remote manipulation of victim

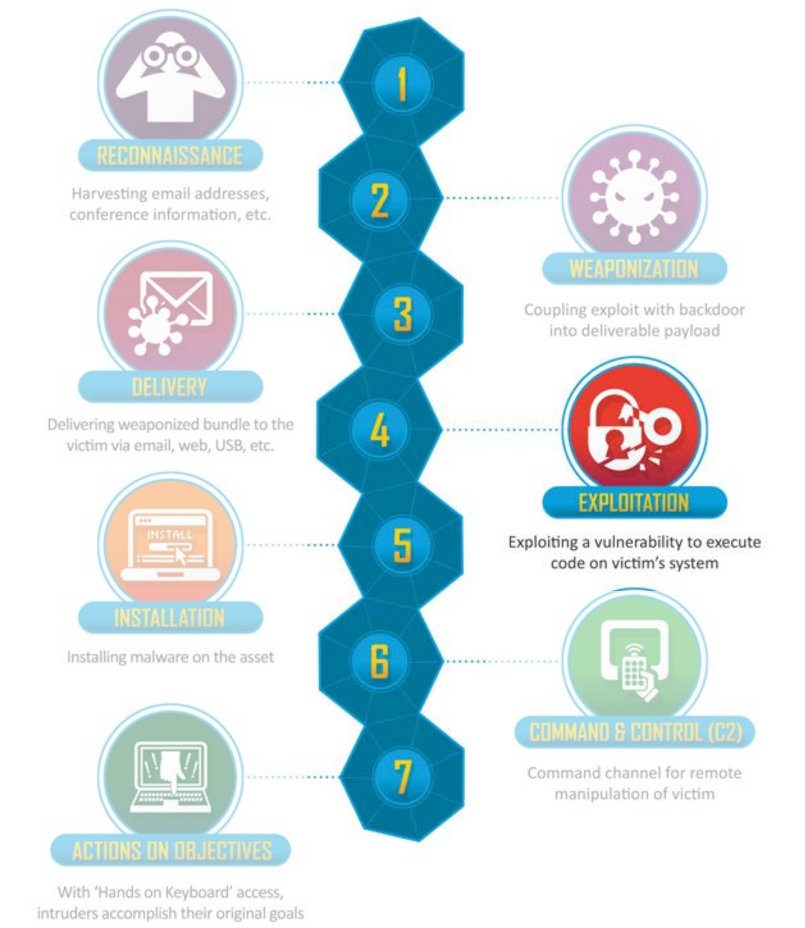
7 ACTIONS ON OBJECTIVES

With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Cyber Kill Chain

EXPLOITATION
Exploiting a vulnerability to execute code on victim's system

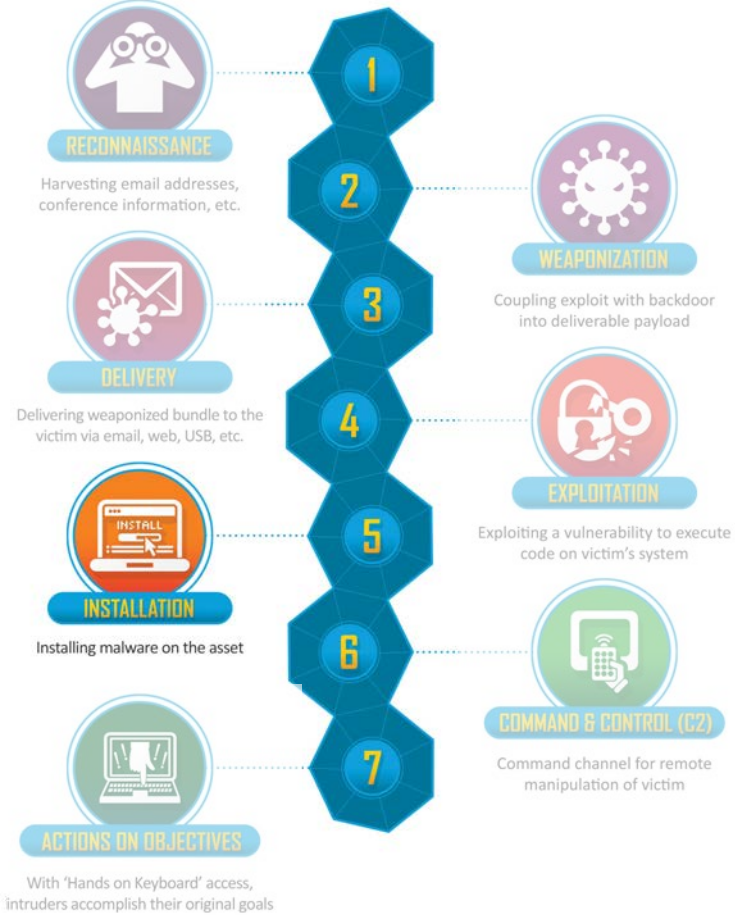

1
RECONNAISSANCE
Harvesting email addresses, conference information, etc.
2
WEAPONIZATION
Coupling exploit with backdoor into deliverable payload
3
DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.
4
EXPLOITATION
Exploiting a vulnerability to execute code on victim's system
5
INSTALL
INSTALLATION
Installing malware on the asset
6
COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim
7
ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Co-funded by the Erasmus+ Programme of the European Union

Cyber Kill Chain

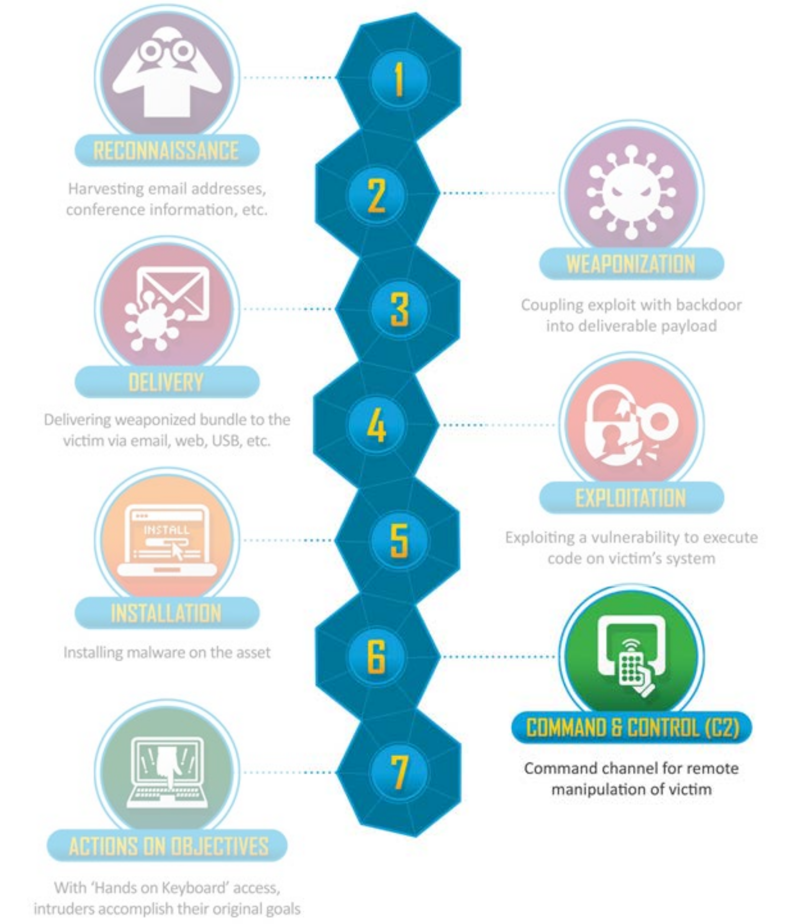
Installing malware on the asset


1
RECONNAISSANCE
Harvesting email addresses, conference information, etc.
2
WEAPONIZATION
Coupling exploit with backdoor into deliverable payload
3
DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.
4
EXPLOITATION
Exploiting a vulnerability to execute code on victim's system
5
INSTALL
INSTALLATION
Installing malware on the asset
6
COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim
7
ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Cyber Kill Chain

Command channel for remote manipulation of victim



1 RECONNAISSANCE

Harvesting email addresses, conference information, etc.

2 WEAPONIZATION

Coupling exploit with backdoor into deliverable payload

3 DELIVERY

Delivering weaponized bundle to the victim via email, web, USB, etc.

4 EXPLOITATION

Exploiting a vulnerability to execute code on victim's system

5 INSTALLATION

Installing malware on the asset

6 COMMAND & CONTROL (C2)

Command channel for remote manipulation of victim

7 ACTIONS ON OBJECTIVES

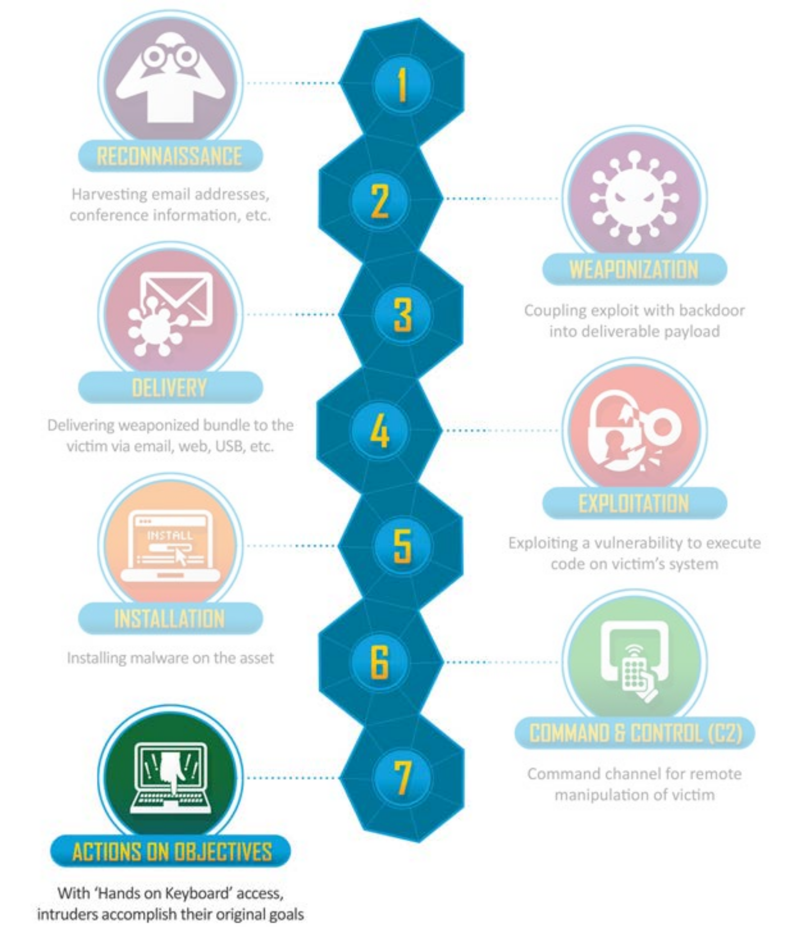
With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Cyber Kill Chain

ACTIONS ON OBJECTIVES

With 'Hands on Keyboard' access, intruders accomplish their original goals



RECONNAISSANCE

Harvesting email addresses, conference information, etc.

WEAPONIZATION

Coupling exploit with backdoor into deliverable payload

DELIVERY

Delivering weaponized bundle to the victim via email, web, USB, etc.

EXPLOITATION

Exploiting a vulnerability to execute code on victim's system

INSTALLATION

Installing malware on the asset

COMMAND & CONTROL (C2)

Command channel for remote manipulation of victim

ACTIONS ON OBJECTIVES

With 'Hands on Keyboard' access, intruders accomplish their original goals

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Co-funded by the
Erasmus+ Programme
of the European Union

Cyber Threat Intelligence Libraries

- Categorisation of Attack Patterns, Weaknesses, Tactics, and Techniques
  - ATT&CK (MITRE)
    knowledge base of adversary tactics and techniques based on real-world observations
    V11.2 (April 2022 - 14 Tactics, 191 Techniques, and 386 Sub-techniques)
  - CAPEC (MITRE)
    Common Attack Pattern Enumeration and Classification
    V3.7 (February 2022 - 546 attack patterns)
  - OWASP Cheat Sheet Series
    a concise collection of high value information on specific web application security topics

giacinto@unica.it

Thank you for your attention!

Co-funded by the
Erasmus+ Programme
of the European Union