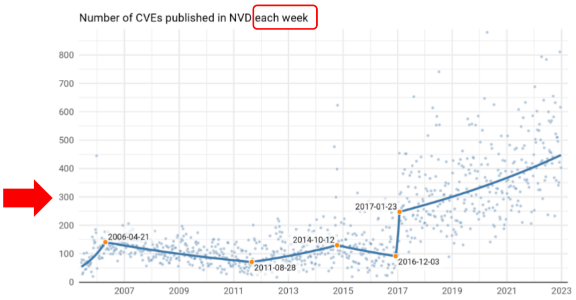
Vulnerabilities: introduction

- A vulnerability is a mistake in software that can be directly used to gain access to a system or network
    - Example microsoft 2017: vulnerability in the way that microsoft office and worded parse specially crafted files. An attacker could then install programs, view, change or delete data, or create new accounts with full user rights. An attacker could exploit the vulnerability by sending a specially crafted file to the user and then convincing the user to open the file
    - Example: stesso esempio del web server che se gli mandavi una richiesta HTTP con un URL lungo e sbagliato che termina con dei comandi, il server eseguiva i comandi con il suo livello di privilegio (che probabilmente era alto)
- Impact: there are many possibilities
    - Informations disclosure, privilege escalation, denial of service, code execution (existing operation (access control vulnerabilities) or arbitrary code)

NB software mistakes + configuration / procedural mistakes → every major incident exploits both kinds of vulnerabilities



- There are several public databases with publicly known vulns
    - This vulns are known to attackers, software manufacturers, defenders (users)
- There are a lot of vulnerabilities
- There are ZERO DAY vulns that are known only by attackers, not known by software manufacturers or defenders
- Vulns are not a sporadic or occasional phenomenon, they are an intrinsic feature of software, it is not just a matter of computer/software engineers
- Every kind of software may suffer of vulnerabilities (segue lista di servizi con esempi di vulnerabilità, ce ne sono tante altre, basta cercare)
    - End user software:
        - Browsers (search "ned search vulnerability database", then search any browser (chrome, edge, safari..), or search "cisa security" and then "security updates chrome")
        - Whatsapp: bug that could have let the hackers to secretly install Spyware on your devices → to remotely exploit the vulnerability, all an attacker needs is the phone number of targeted users and send them a maliciously crafted MP4 file over WhatsApp, which eventually can be programmed to install a malicious backdoor or spyware app on the compromised devices silently
    - Network devices:
        - home routers / SOHO D-LINK: D-Link DIR-816 A2 1.10 B05 devices allos arbitrary remote code execution without authentication
        - Cisco: CVE-2021-1141 multiple vulnerabilities in the web UI of Cisco smart software manager satellite could allow an unauthenticated, remote attacker to execute arbitrary code commands on the underlying operating system
    - Security software
        - Antivirus:
            - alert (TA16-187A) Symantec and Norton security products contain critical vulnerabilities, systems affects are all Symantec and Norton branded antivirus products, the overviews is exploitation of these vulnerabilities could allow a remote attacker to take control of an affected system
            - McAfee aggiornamento di sicurezza per il software
            - CVE-2021-1647 microsoft defender remote code execution vulnerability
        - Pulse secure VPN: contains multiple vulnerabilities that can allow remote unauthenticated remote attacker to compromise the VPN server and connected clients
        - TLS:
            - 'goto fail' could allow an attacker to capture or modify data that was supposed to be encrypted via SSL/TLS. The vulnerability affected OSX and iOS devices and was caused by one line typo in Apple's products
            - Heartbleed affects servers running OpenSSL, it could allow an attacker to grab all kinds of data, including SSL site keys, usernames and passwords…
        - Kerberos…a lot of CVEs
        - Firewall: CVE-2022-23176 WatchGuard Firebox and XTM appliances allows a remote attacker with unprivileged credentials to access the system with a privileged management session via exposed management access

Vulnerabilities: intro, exploit, injection

  - Industrial control systems (ICS):
    - Remotely monitor, manage and control industrial equipment over the net (LANTRONIX) → the UDS1100-IAP is a rugged and powerful tool which enables users to connect, manage and control just about any piece of industrial equipment from virtually anywhere over the ethernet or the internet → by sending a malformed request on port 30718, devices that have not been updated to the latest firmware version will reply back with their config, including the telnet password
    - ICS vulnerabilities → tante
    - Ci sono altri 3 esempi (Schneider Wlectric InduSoft web studio and intouch machine edition, Yokogawa CENTUM and Exaopc, Honewell Midas Gas Detector Vulnerabilities)
  - Medical devices (Insulet Omnipod, Hillrom Medical Device Management, Pacemakers)
  - Cars, ships, IoT (everything nowadays is a computer and has to be secured like computers)
- Remarks for non IT engineers:
  - We've normalized the fact that technology products are released to market with dozens, hundreds or thousands of defects, when such poor construction would be unacceptable in any other critical field (CISA director, 2023)
  - A manufacturer producing a system with a software component must think of itself as a software company and manage software security accordingly (Technology innovation management review 2017)

Vulnerability: a better definition

- What about the mistakes in the design (as opposed to the implementation)
  - Example: Swisslog Healthcare Translogic PTS → user and root accounts have hardcoded passwords that can be accessed remotely on the Nexus Control Panel. These accounts are enabled by default and cannot be turned off by native configuration of the system. You can buy one device and immediately have the passwords valid on all the devices in the world (passwords cannot be modified)
  - Same in Siemens SCALNCE X Switches: devices do not create a new unique private key after factory reset, devices use the hardcoded private RSA-key shipped with the firmware image. You can buy one device, reverse engineer its firmware and have the private key valid on all the devices in the world (private key cannot be modified)

NB Never ever design systems that embed the same secret in all their instances and a secret that cannot be modified

- What about mistakes useful after initial access (like lateral movement or privilege escalation) → there are many examples
- So, the new definition of vulnerability is: a flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy → the mistakes can occur at any of several phases and can be exploited for many purposes
- Prof's favorite definition di cosa non va fatto (?)
  - Design example: secrets embedded in code/devices can be used anywhere
  - Implementation example: certain inputs not handled properly
  - Operation/management example: insecure default

How to actually exploit

- Exploit:
  - A software mistake does not provoke any damage itself, the problem is when execution incurs in that mistake
  - It is always necessary a carefully constructed input → that is the exploit
- Exploit development: the attacker analyzes and experiments the vulnerability, then construct a sequence of byte that can be used for all instances of vulnerable program
- Once the attacker has the exploit, he has to inject it in the vulnerable system
- Injection categories:
  - User action required: the successful exploitation of this vulnerability requires a user to take some action before the vulnerability can be exploited (open an email attachment)
  - No user action required: the vulnerable system can be exploited without any interaction from any user (send network message to server)
  - Remote: ca ne done by a program running remotely
    - Authenticated on the target

Vulnerabilities: intro, exploit, injection

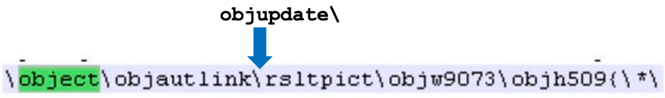
        - Unauthenticated on the target
    - Local: can be done only by a program already running on the target
    - Example No user action (march 2017): Microsoft security Bulletin MS17-010- critical → an attacker who successfully exploited the vulnerabilities could gain the ability to execute code on the target server, to export the vulnerability, in most situations, an unauthenticated attacker could send a specially crafted packet to a targeted SMBv1 server
    - Example No user action (April 2022): Remote procedure call runtime remote code execution vulnerability (Microsoft) → to exploit this vulnerability, an attacker would need to send a specially crafted RPC all to an RCP host. This could result in remote code execution on the server side with the same permissions as the RCP service. The attacker does not require any access to setting or files to carry out the attack, the vulnerable system can be exploited without any interaction from any user

	LOCAL	REMOTE	
		Authenticated	Not Auth.
USER ACTION **NOT** REQUIRED			
USER ACTION REQUIRED			

    - Examples local injection
        - CVE 2022-0847: a flaw was found in the way the flags member of the now pipe buffer structure was lacking proper initialization in copy_page_to_iter_pipe and push_pipe functions in the linux kernel and could contain stale values. An unprivileged local user could use this flaw to write to pages in the cache backed by read only files and such escalate their privileges on the system
        - CVE 2021-42103: an uncontrolled search path element vulnerabilities in Trend Micro Apex One and Apex One as a service allow a local attacker to escalate privilege on affected installations. An attacker must first obtain the ability to execute low-priveleged code on the target system in order to exploit this vulnerability. This vulnerability is similar but not identical to CVE 2021-42101
- Wormable Vuln (tragedy)
    - Remote code execution on a machine with not need of authentication and user action not required (impact injection → code execution)
    - Exploit propagated itself automatically: it attempts to connect to all the IP addresses and inject a copy of itself on every IP address found → within a few minutes, all the vulnerable systems reachable from the patient zero will be infected. This is one of the reasons why NAT and tight firewall rules on user platforms are important (Wannacry)

Exploit and injection examples

- **Example exploit 1:**
    - File word that downloads a malware upon opening
    - Impact: code execution
    - Injection: user action required, it is remote and not authenticated
    - It occurs when the file is opened → there is no visible effect (terrible!!)
    - The exploit for this vuln:
        - The user action is opening the word file
        - The word action is: analyze the opened file, if it finds a link to URL-X, it downloads the file
            - Word analyze that file and possibly execute it → if it finds a VBSCRIPT script, it execute the script and the shell commands (for downloading the malware for example)
- Exploit construction:
    - Write a script in VBSCRIPT language that created a shell and executed a command for downloading a malware
    - Create a file with HTLM content and that script
    - Name the file with .rtf extension
    - Set up and configure the web server such that the file is at URL-X and the content type is set to application/hta
    - Now create the deceiving word file, insert in it a word object with the link to URL-X
    - Modify the word file with binary editor → (necessary step for executing shell upon the file opening, otherwise is required the click on the object)



```
objupdate\
\object\objautlink\rsltpict\objw9073\objh509{\*\
```

  - That's it → l'apertura del file word provoca l'esecuzione dei comandi scelti dall'attaccante senza che lo user si accorga di nulla
- So the exploit is also an input that is not handled correctly by the vulnerable software
- **Example exploit 2:**
  - Code library placed in a certain directory
  - Impact: code execution and possible privilege escalation
  - Injection: user action required and local
  - The user must launch an executable that uses that library, the local access is needed for placing the library in a specific directory
  - Example: (April 2022) Qt allows for privilege escalation due to hard coding of qt_prfxpath value, impact: by placing a file in an appropriate location on a windows system, an unprivileged attacker may be able to execute arbitrary code with the privileges of the software that uses Qt. Exploit: code library placed in a certain directory
- Most software products are developed with Qt software development framework
  - when the product runs, it loads libraries from a list of predefined directories
  - Some of those products are installed to run with local admin privilege
- The vulnerability sta nel fatto che the list of predefined directories always starts with a directory that can be guessed easily and can be written by low privilege users (in default windows configurations)
- Exploit construction:
  - Write a malicious code, rename it QtCore.dll, create a directory named C:/Qt/5.3/msvc2013_64_opengl/pluins/
  - Place the malicious code in that directory
  - If a software with Qt runs then that software will execute also the malicious code

Vunl vs exploit vs injection

- The actual exploitation of a vuln requires all the three steps
  1. Vulnerability discovery → difficult, often is a full time job
  2. Exploit creation → is very difficult, also often is a full time job
  3. Exploit injection in vulnerable systems → there are different levels of difficulty, the injection can be local/remote, user/no user…

→ the exploit must be developed, its existence is not automatic

- Bugs vs vulns vs exploitable vulns: solo una parte dei bug sono delle vulnerabilità e solo una parte delle vulnerabilità può effettivamente essere sfruttata (exploitable vulns)
- PoC (proof of concept): exploits that can be publicly available, can be modified and specialized easily
  - Sono praticamente degli exploit che servono a dimostrare che una vulnerabilità può essere sfruttata, non sono malevoli, però quando sono disponibili pubblicamente possono essere modificati e usati dagli attaccanti