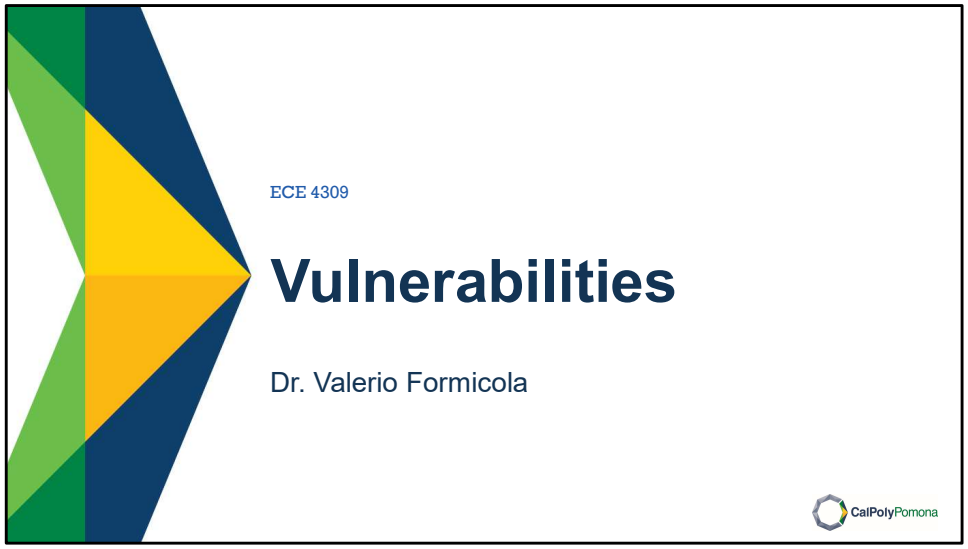
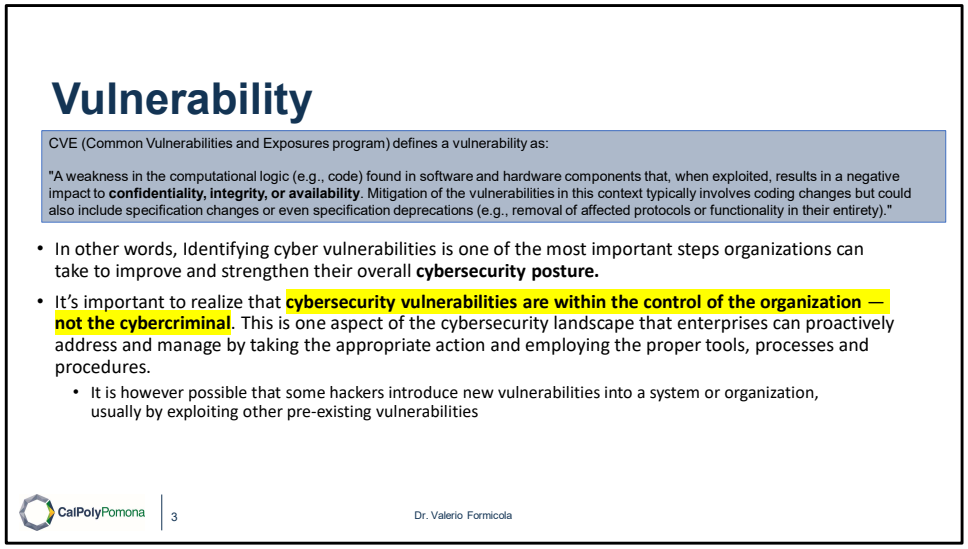

ECE 4309
Vulnerabilities
Dr. Valerio Formicola
CalPolyPomona

Cyber security posture

- Organization's security posture is the **big-picture view of organization's cybersecurity strength** and resilience — a measure of how prepared you are to defend against and respond to cyber threats.
- It considers the collective status of your **organization's security mechanisms, policies, and procedures**.
- Security posture is **not static** — it constantly evolves alongside your organization, developments in technology, and emerging threats.

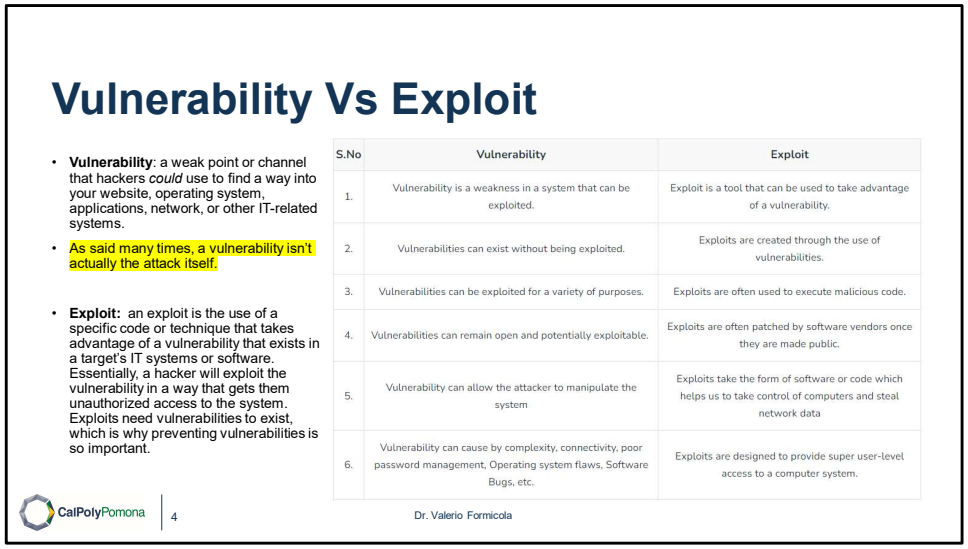
Here are the main elements:

- **Risk management:** Identifying potential security risks and implementing strategies to mitigate them. Proper risk management ensures your ability to handle threats that might compromise your security.
- **Incident response**: How you respond to a security breach or attack. Incident response includes the plans and procedures in place to minimize the damage of an incident, recover, and then learn from it.
- **Compliance and governance:** How closely you adhere to established industry standards, regulations, and laws related to data security. Compliance and governance measures demonstrate accountability to regulatory bodies and help you earn trust in your industry.
  - E.g., ISO 27001, Zero Trust, PCI-DSS, FERPA, etc.
- **Security architecture:** The design and implementation of security controls and measures to protect data and resources throughout your networks and systems.
- **Employee training and awareness**: Educating employees, ensuring they understand and follow security protocols.



A vulnerability in cybersecurity is a **weakness** in a host or system, such as a missed software update or system misconfiguration, that can be exploited by cybercriminals to compromise an IT resource and advance the attack path.

Remember the distinctions between vulnerability, threat and risk:
As noted above, a **vulnerability** is a weakness that can be exploited by a malicious actor. For example, unpatched software or overly permissive accounts can provide a gateway for cybercriminals to access the network and gain a foothold within the IT environment.
A **threat** is a malicious act that can exploit a security vulnerability.
A **risk** is what happens when a cyber threat exploits a vulnerability. It represents the damage that could be caused to the organization in the event of a cyberattack.



Example 1

Vulnerability: You did not update your WordPress plugin, which has a code error.

Exploit: A hacker uses the vulnerability to launch a SQL injection attack.

Example 2

Vulnerability: A web admin has a weak password that lacks complexity and doesn't meet NIST passwords standards. (See NIST SP 800-63B Authentication and Lifecycle Management, section 5.1.1.) Some general password creation best practices include using long passwords that include a combination of uppercase and lowercase characters, and at least one special character and number.

Exploit: A hacker uses a "cracker tool" to crack the password and now controls your website. It should be noted should how accessible these "cracker tools" are. There are literally top 10 lists that rank password cracking tools that range from ones that assists with brute force attacks to tools that can crack LM and NTLM hashes!

Example 3

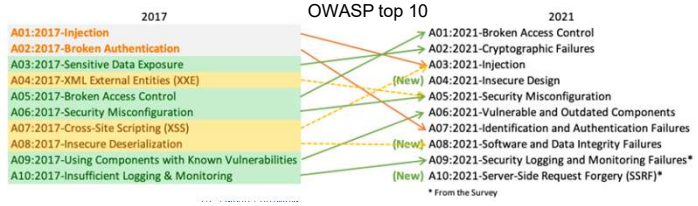
Vulnerability: A website has an area that allows users to upload unvalidated files with no filters or limits.

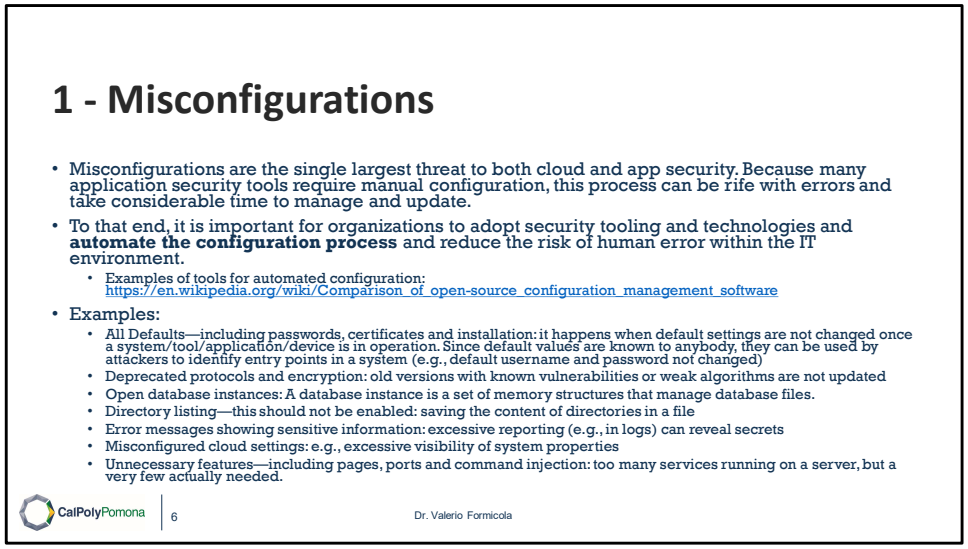
Exploit: A hacker uploads a file that contains executable code and now has access to your website source code and database credentials (basically controlling your website).

Remember also the Fault-Error-Failure sequence:
A vulnerability is a Fault in the system.
An error is generated with the activation/use of that fault (i.e., with the exploitation of vulnerability for driving the system into incorrect operation)
A failure is consequence of exploiting a vulnerability, for example with a damage or problems in the entire or part of the system.

Categorization of vulnerabilities

- Many models exist, often very similar
  - E.g., OWASP top 10, updated almost yearly, very famous
    - https://owasp.org/www-project-top-ten/
  - Commercial white papers/blogs:
    - E.g., crowdstrike, https://www.crowdstrike.com/





Another list of misconfigurations from CISA, overlapping here with other categories we use

https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-278a

2 - Outdated or Unpatched Software

- Software vendors periodically release application updates to either add new features and functionalities or patch known cybersecurity vulnerabilities. Unpatched or outdated software often make for an easy target for advanced cybercriminals.
  - See the slide *Vulnerability lifecycle* at the end of this set of slides
- While software updates may contain valuable and important security measures, it is the responsibility of the organization to update their network and all endpoints.
- Software vulnerabilities can be due to 1) design flaws or 2) implementation errors.
- May result in attackers installing a **malware payload**
- Organizations should develop and implement a process for **prioritizing software updates and patching**. To the extent possible, the team should also automate this activity so as to ensure systems and endpoints are as up to date and secure as possible.
  - ITIL management framework
  - More recently, DevOps and DevSecOps

ITIL update matrix

Urgency		Impact: High-System Wide (Business Unit, Department, Location)	Impact: Medium-Multiple Users (Number of Users)	Impact: Low-Single User (Single User)
	High (Can no longer perform primary work functions)	1 - Critica	2 - High	3 - Moderate
	Medium (Work functions impaired, work around in place)	2 - High	3 - Moderate	4 - Low
	Low (Inconvenient)	3 - Moderate	4 - Low	4 - Low

https://www.cisa.gov/news-events/news/understanding-patches-and-software-updates

CalPolyPomona | 7 Dr. Valerio Formicola

Unfortunately, because updates from different software applications can be released daily and IT teams are typically overburdened, it can be easy to fall behind on updates and patching, or miss a new release entirely. Failing to update even one machine can have potentially disastrous consequences for the organization, providing an attack path for ransomware, malware and a host of other security threats

ITILv3 defines impact as a measure of the **effect** of an incident, problem, or change on business processes.

ITIL (Information Technology Infrastructure Library) is a set of guidelines for IT service management. The guidelines cover best practices and tried-and-true processes for everything from incident management to problem management to change management.

DevOps is the practice of bridging the gap between development and operations. Its core principles are open communication, collaboration, and shared goals. “Unlike frameworks like ITIL, there is no ‘official’ document of best

practices for a DevOps team. But we can generally agree that, at its core, DevOps is about delivering business value to an organization by breaking down organizational silos, increasing transparency, and fostering open communication between developers and IT operations teams."

DevSecOps—short for *development, security,* and *operations*—automates the integration of security at every phase of the software development lifecycle, from initial design through integration, testing, deployment, and software delivery. DevSecOps represents a natural and necessary evolution in the way development organizations approach security. In the past, security was 'tacked on' to software at the end of the development cycle (almost as an afterthought) by a separate security team and was tested by a separate quality assurance (QA) team. This was manageable when software updates were released just once or twice a year. But as software developers adopted Agile and DevOps practices, aiming to reduce software development cycles to weeks or even days, the traditional 'tacked-on' approach to security created an unacceptable bottleneck.

3 - Unsecured APIs

- Another common security vulnerability is **unsecured application programming interfaces (APIs).** APIs provide a digital interface that enables applications or components of applications to communicate with each other over the internet or via a private network.
- APIs are one of the few organizational assets with a public IP address (usually with HTTP protocol in cloud). If not properly and adequately secured, they can become an easy target for attackers to breach.
  - https://www.trio.dev/back-end/resources/api-examples
  - https://learn.microsoft.com/en-us/industry/retail/intelligent-recommendations/sample-api
- As with misconfigurations, securing APIs is a process prone to human error. While rarely malicious, IT teams may simply be unaware of the unique security risk this asset possesses and rely on standard security controls. Conducting a security awareness training to educate teams on security best practices specific to the cloud — such as how to store secrets, how to rotate keys and how to practice good IT hygiene during software development — is critical in the cloud, just as in a traditional environment.
- Some famous examples of successful attacks (breaches): https://techblog.cisco.com/blog/real-world-api-security

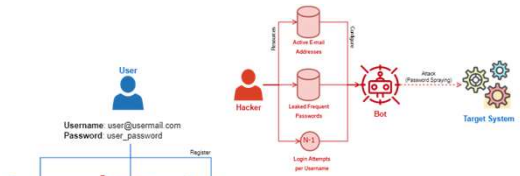
```
def get_access_token():
    """
    If you are

    1. an approved LinkedIn developer
    2. on a paid subscription to their Consumer Product

    You can use this function to fetch an `access_token` to access the API.
    """
    LI_ACCESS_TOKEN_EXCHANGE_URL = 'https://www.linkedin.com/oauth/v2/acces
    access_token = requests.post(LI_ACCESS_TOKEN_EXCHANGE_URL, params={
        'grant_type': 'client_credentials',
        'client_id': LINKEDIN_CLIENT_ID,
        'client_secret': LINKEDIN_CLIENT_SECRET,
    }).json()['access_token']
    return access_token
```

3 - Weak or Stolen User Credentials

- Many users **fail to create unique and strong passwords** for each of their accounts. Reusing or recycling passwords and user IDs creates another potential avenue of exploitation for cybercriminals.
- To address this particular cybersecurity vulnerability, organizations should set and enforce clear policies that require the use of strong, unique passwords and prompt users to change them regularly.
- Organizations should also consider implementing a multifactor authentication (MFA) policy, which requires more than one form of identification, such as both a password and a fingerprint or a password and a one-time security token, to authenticate the user.



CalPolyPomona | 9

Dr. Valerio Formicola

Weak user credentials are most often exploited in brute force attacks when a threat actor tries to gain unauthorized access to sensitive data and systems by systematically trying as many combinations of usernames and guessed passwords as possible. If successful, the actor can enter the system and masquerade as the legitimate user; the adversary can use this time to move laterally, install back doors, gain knowledge about the system to use in future cyberattacks, and, of course, steal data.

Vulnerable to brute force attacks, e.g.:
Dictionary attacks is an attack using a restricted subset of a keyspace to defeat a cipher or authentication mechanism by trying to determine its decryption key or passphrase, sometimes trying thousands or millions of likely possibilities often obtained from lists of past security breaches.

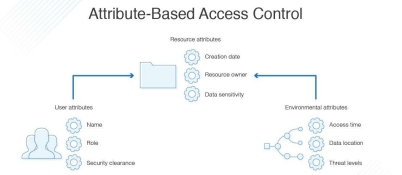
Credential stuffing attacks consist of using leaked credentials of a given system to try to access other systems.

Password spraying attacks are dictionary attacks that employ frequently used passwords to try to get access to systems through a list of potential usernames.

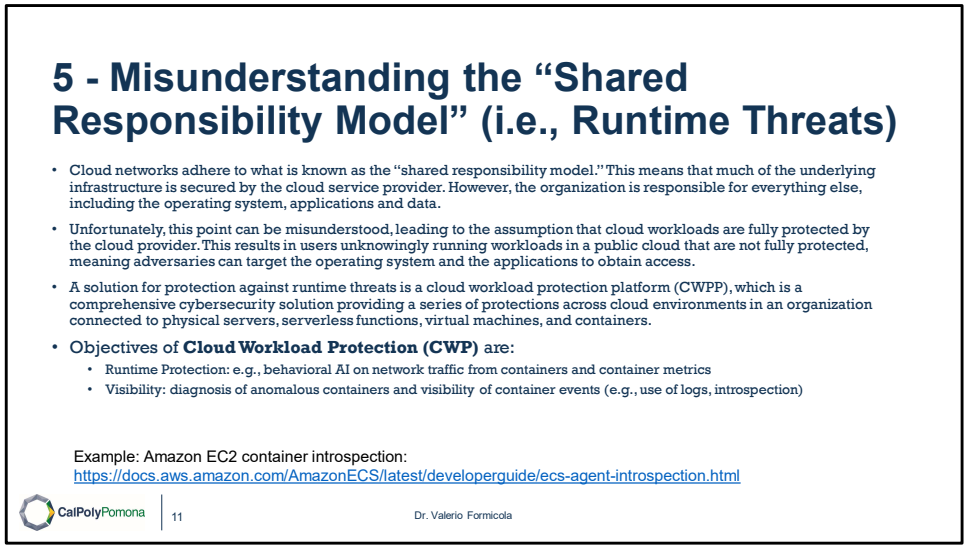
4 - Access Control or Unauthorized Access

- Companies often grant employees more access and permissions than needed to perform their job functions. This increases identity-based threats and expands access to adversaries in the event of a data breach.
- To address this issue, organizations should implement the **principle of least privilege (POLP),** a computer security concept and practice that gives users limited access rights based on the tasks necessary to their job. POLP ensures only authorized users whose identity has been verified have the necessary permissions to execute jobs within certain systems, applications, data and other assets.
- POLP is widely considered to be one of the most effective practices for strengthening the organization's cybersecurity posture, in that it allows organizations to control and monitor network and data access.
- Example of technologies that support POLP strategies are **Role Based Access Control (RBAC)** and **Attribute-based access control (ABAC),** also known as **policy-based access control**

ABAC Defines an access control paradigm whereby a subject's authorization to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment attributes.



https://frontegg.com/guides/rbac



Organizations that are using the cloud or shifting to a cloud or hybrid work environment must update their cybersecurity strategy and tooling to ensure they are protecting all areas of risk across all environments. Traditional security measures do not provide security in a cloud environment and must be supplemented to provide enhanced protection from cloud-based vulnerabilities and threats.

Example: A compromised container can affect another process, which can compromise other systems or containers. These threats can be dealt with by regularly monitoring and conducting container threat detection activities. You should pay special attention to network traffic.

Amazon EC2 monitoring: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs_monitoring.html

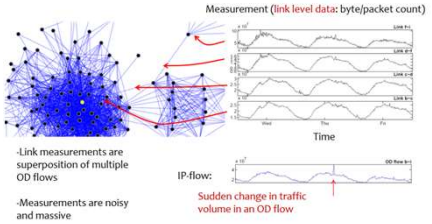
6 - Network vulnerabilities

- Network vulnerabilities can include any vulnerabilities within the software, hardware and processes that govern the **flows of data workloads, user traffic and computing requests within the IT networks**. Network vulnerabilities range from the hardware components in the physical layer and all the way up the stack to the application layer of the OSI model.

- **Example:** Even when all device software and firmware are maintained and up to date, the network fabric may be vulnerable to **unauthorized access due to misconfigured firewall and traffic routing**.

Example from a paper:
https://ieeexplore.ieee.org/abstract/document/8254833

Problem: Detection of anomalous traffic in large network



The extensive nature of the technologies that constitute an IT network makes it challenging to keep track of networking vulnerabilities: every hardware product, every software service is from a different vendor and is exposed to its own set of security risks.

7 – Physical vulnerabilities

- In the context of cybersecurity vulnerabilities, physical security is particularly relevant to cloud infrastructure vendors and large organizations operating in-house data center systems. A physical vulnerability may include:
  - The ability to access server rooms
  - Camera blind spots
  - Inadequate documentation
  - Recording of physical activities performed in the data center, such as replacing storage devices
- However, any insider threat within the physical office premise, or theft or loss of a **BYOD (Bring Your Own Device)** device can expose security risks to the organization.

In order to address these physical vulnerabilities, organizations must enforce strict policy controls governing the use of business information on BYOD devices and access to corporate apps, services and networks from outside of the physical premise of the organization.

- Example of technology: **PSIM,** Physical Security and Information Management system

Example of solutions for datacenters:
https://www.hw-group.com/solution/datacenter-environment-monitoring-and-access-control-systems

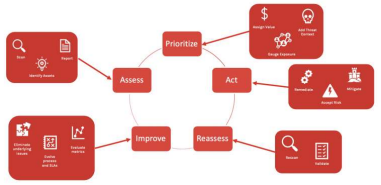
Bring your own device (**BYOD**)—also called **bring your own technology** (**BYOT**), **bring your own phone** (**BYOP**), and **bring your own personal computer** (**BYOPC**)—refers to being allowed to use one's personally owned device, rather than being required to use an officially provided device.

8 – Insiders as a vulnerability

- An insider will usually be working alone and have limited financial resources and time. However, the fact that they are insiders gives them an automatic advantage. They already have some access to your network and some level of knowledge. Depending on the insider's job role, they might have significant access and knowledge.
  - In some cases, competitors will use a disgruntled insider to get information from your company. They may also seek out insider information available for purchase on the dark web, a shadowy anonymous network often engaging in illicit activity.
- **Behavioral assessments** are a powerful tool in identifying insider attacks. Cybersecurity teams should work with human resources partners to identify insiders exhibiting unusual behavior and intervene before the situation escalates.
- Another insider risk is **Shadow IT**: any software, hardware or IT resource used on an enterprise network without the IT department's approval and often without IT's knowledge or oversight. Sharing work files on a personal Dropbox account or thumb drive, meeting on Skype when the company uses WebEx, starting a group Slack without IT approval—these are examples of shadow IT. Shadow IT does not include malware or other malicious assets planted by hackers. It refers only to unsanctioned assets deployed by the network's authorized end users.

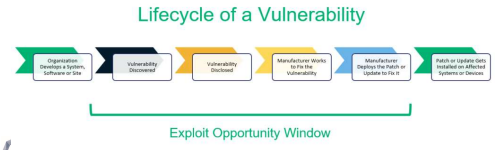
https://www.ibm.com/topics/shadow-it

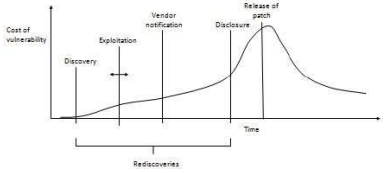
Vulnerability lifecycle



Stages of vulnerability management *:

- Step 1. Assess: narrow down and define the assets to be assessed for vulnerabilities (network or hosts agents).
- Step 2. Prioritize: evaluate risk based on assigned value and threat exposure (in collaboration with Threat Intelligence, usually big vendors). Includes reputation and cost analysis.
- Step 3. Act: decide to accept, mitigate or remediate with patches and fixes
- Step 4. Reassess: validate previous work (tracking, reporting, metrics) and evaluate introduction of new risks or changes
- Step 5. Improve: attempt to solve the problem at deeper levels, for example during production stages.





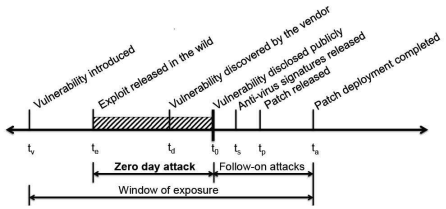
*Example adapted from crowdstrike:

https://www.crowdstrike.com/cybersecurity-101/vulnerability-management/vulnerability-management-lifecycle/
https://sectigostore.com/blog/exploit-vs-vulnerability-whats-the-difference/
https://ieeexplore.ieee.org/document/6165770

Zero-day vulnerability and exploit

A **zero-day** (also known as a 0-day) is a vulnerability in a computer system that was previously unknown to its developers or anyone capable of mitigating it. Until the vulnerability is mitigated, threat actors can exploit it.
An exploit taking advantage of a zero-day is called a **zero-day exploit**, or **zero-day attack**.

In most severe cases, Zero-day exploit occur **on the same day** the software, hardware or firmware flaw is detected by the manufacturer.
As it's been zero days since the security flaw was last exploit, the attack is termed as zero-day exploit or zero-day attack.
This kind of cyber-attacks are considered dangerous because the developer have not had the chance to fix the flaw yet.
Zero-day exploit typically targets large organizations, government departments, firmware, hardware devices, IoT, users having access to valuable business data, etc.



Zero Days are very common in Advanced Persistent Threats (APTs)

https://securityaffairs.co/9566/hacking/wrong-response-to-zero-day-attacks-exposes-to-serious-risks.html

Vulnerability publication models

- **CVE** (Common Vulnerabilities and Exposures) is one of the most authoritative: CVE® is a list of information security vulnerabilities and exposures that provides common identifiers for publicly known cybersecurity vulnerabilities. CVE makes it possible to share data across separate vulnerability capabilities (cybersecurity tools, repositories, and services) with this common enumeration. The use of CVE Records ensures that two or more parties can confidently refer to a **CVE Identifier (CVE ID)** when discussing or sharing information about a unique vulnerability. In this way, CVE is fundamental to the vulnerability management infrastructure

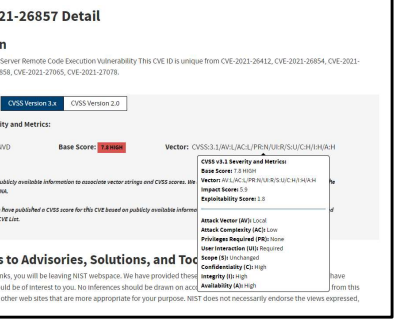
Mitre Corporation maintains an incomplete list of publicly disclosed vulnerabilities in the Common Vulnerabilities and Exposures, shared with the National Institute of Standards and Technology (NIST).

https://cve.mitre.org/

Each vulnerability is given a risk score using **Common Vulnerability Scoring System (CVSS),** Common Platform Enumeration (CPE) scheme, and Common Weakness Enumeration.

National Vulnerability Database (NVD) is the U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP)

https://nvd.nist.gov/search

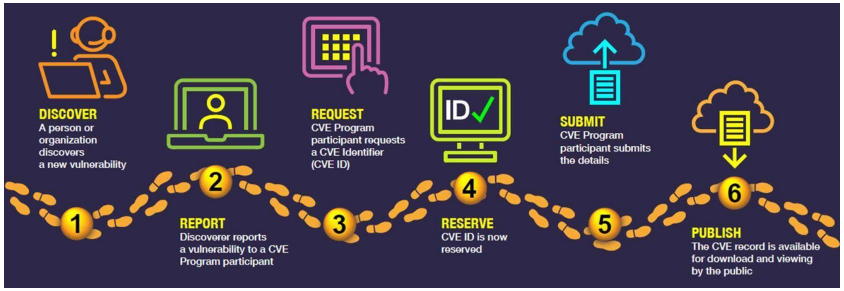


For detailed information regarding CVE please refer to
https://cve.org/
or the CNA Rules at
https://www.cve.org/ResourcesSupport/AllResources/CNARules.

SCAP: https://www.lifewire.com/what-is-scap-2487459

Example: CVE record creation

https://www.cve.org/ResourcesSupport/Glossary



Summary of process here: https://nvd.nist.gov/general/cve-process

https://www.cve.org/ResourcesSupport/Glossary

Other models or lists

- OWASP: https://owasp.org/www-community/vulnerabilities/
- Communities and Govs from different countries might have their local models or databases of vulnerabilities
  - Most often they refer to CVE and NVD

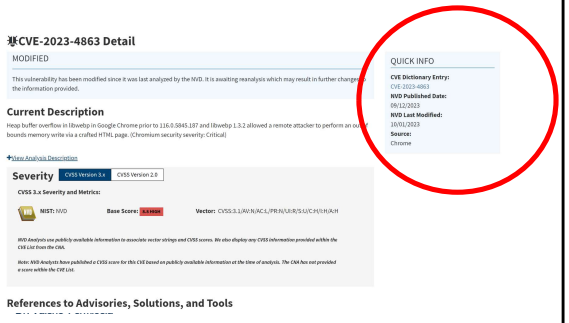
Examples of CVEs

https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-117a

https://www.hackthebox.com/blog/most-popular-cybersecurity-vulnerabilities-and-exploits-from-2022

Example: cve-2023-5129

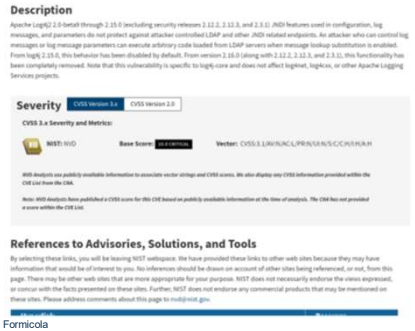
- https://nvd.nist.gov/vuln/detail/CVE-2023-4863



Example top scored: CVE-2021-44228 aka Log4Shell

- https://nvd.nist.gov/vuln/detail/CVE-2021-44228
- Score: 10.0

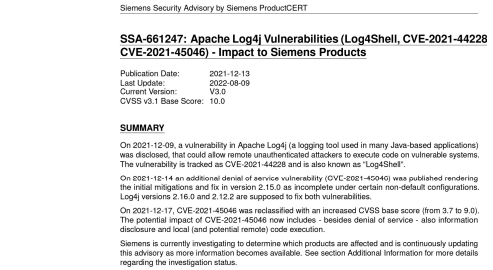
https://en.wikipedia.org/wiki/Log4Shell



Example of impacted systems due to Log4Shell

- https://cert-portal.siemens.com/productcert/pdf/ssa-661247.pdf

Similar analysis is done for all the products and systems in the world that used Log4J



Siemens Security Advisory by Siemens ProductCERT

SSA-661247: Apache Log4j Vulnerabilities (Log4Shell, CVE-2021-44228 CVE-2021-45046) - Impact to Siemens Products

Publication Date: 2021-12-13
Last Update: 2022-08-09
Current Version: V3.0
CVSS v3.1 Base Score: 10.0

SUMMARY

On 2021-12-09, a vulnerability in Apache Log4j (a logging tool used in many Java-based applications) was disclosed, that could allow remote unauthenticated attackers to execute code on vulnerable systems. The vulnerability is tracked as CVE-2021-44228 and is also known as "Log4Shell".

On 2021-12-14 an additional denial of service vulnerability (CVE-2021-45046) was published rendering the initial mitigations and fix in version 2.15.0 as incomplete under certain non-default configurations. Log4j versions 2.16.0 and 2.12.2 are supposed to fix both vulnerabilities.

On 2021-12-17, CVE-2021-45046 was reclassified with an increased CVSS base score (from 3.7 to 9.0). The potential impact of CVE-2021-45046 now includes - besides denial of service - also information disclosure and local (and potential remote) code execution.

Siemens is currently investigating to determine which products are affected and is continuously updating this advisory as more information becomes available. See section Additional Information for more details regarding the investigation status.

CISA: Current authority for cyber-security in USA

- https://www.cisa.gov/
- The Cybersecurity and Infrastructure Security Agency (CISA) is a component of the United States Department of Homeland Security (DHS) responsible for cybersecurity and infrastructure protection across all levels of government, coordinating cybersecurity programs with U.S. states, and improving the government's cybersecurity protections against private and nation-state hackers.

https://en.wikipedia.org/wiki/Cybersecurity_and_Infrastructure_Security_Agency