# Security Governance
# Master of Science in Cyber Security

# AA 2023/2024

CYBER-RISK MANAGEMENT

# Recap: Risk Identification

Observation

there are three elements without which there can be no risk
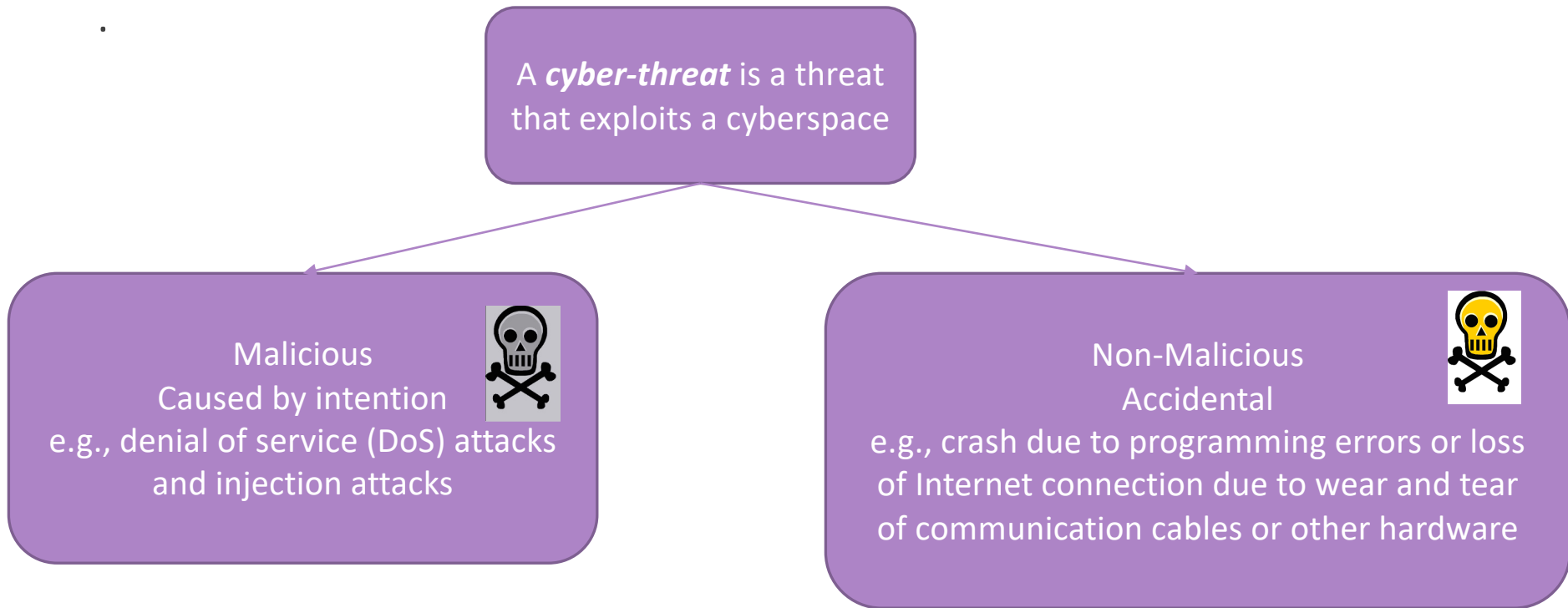
## Asset    Vulnerability    Threat

# Cyber-Systems

A **cyberspace** is a collection of interconnected computerized networks, including services, computer systems, embedded processors, and controllers, as well as information in storage or transit.

A **cyber-system** is a system that makes use of a cyberspace.

A **cyber-physical system** is a cyber-system that controls and responds to physical entities through actuators and sensors.
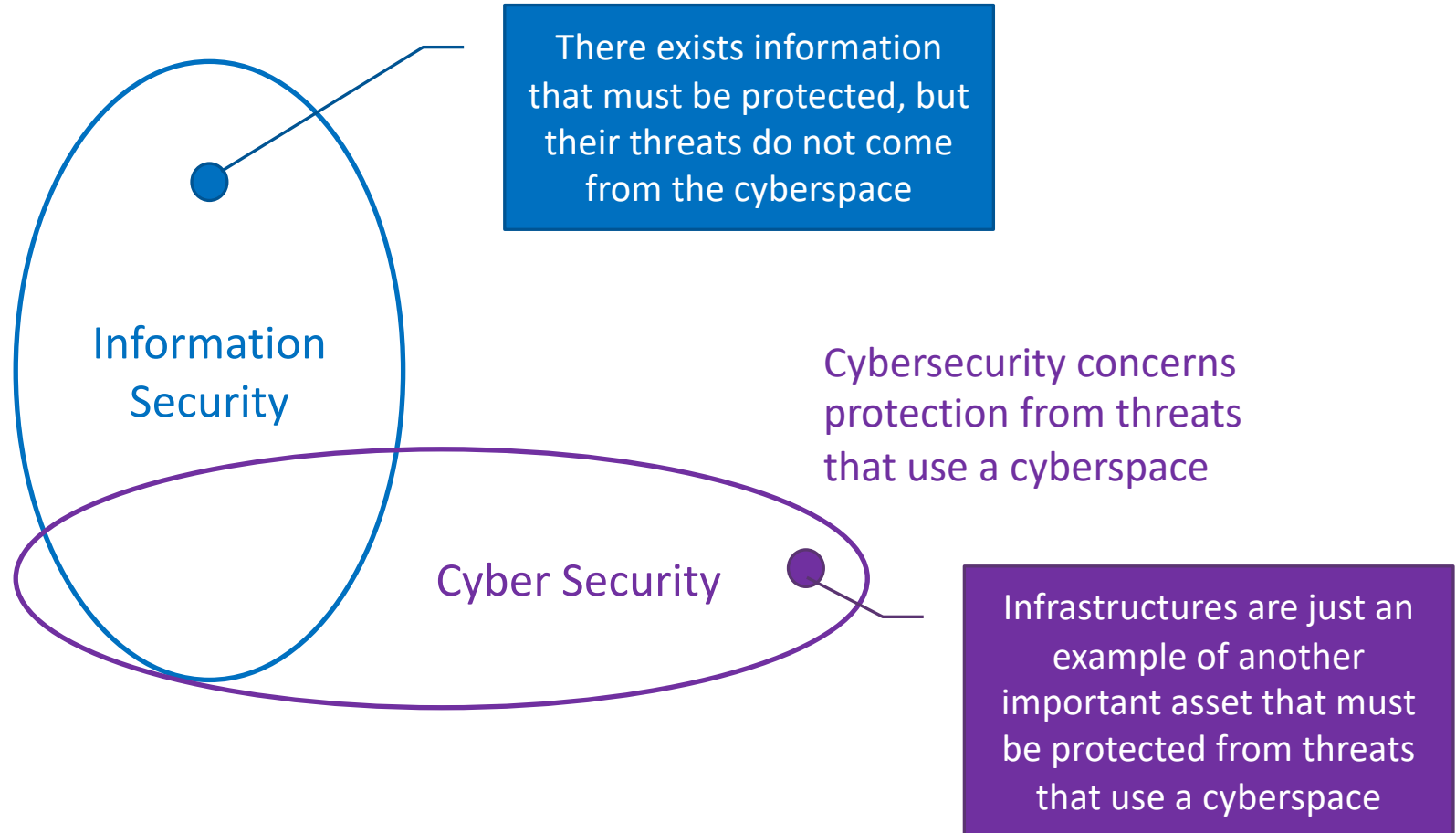
# Cyber-Security

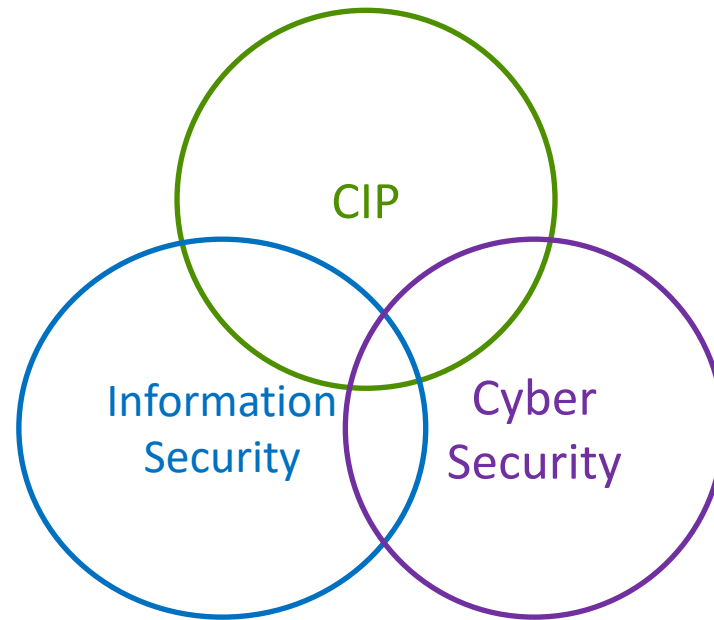*Cybersecurity* is the protection of cyber-systems against cyber- threats.

.

A *cyber-threat* is a threat that exploits a cyberspace

Malicious
Caused by intention
e.g., denial of service (DoS) attacks and injection attacks

Non-Malicious
Accidental
e.g., crash due to programming errors or loss of Internet connection due to wear and tear of communication cables or other hardware

# How Does Cybersecurity Relate to Information Security

Information security is the preservation of Information CIA properties

**Information Security**

**Cyber Security**

There exists information that must be protected, but their threats do not come from the cyberspace

Cybersecurity concerns protection from threats that use a cyberspace

Infrastructures are just an example of another important asset that must be protected from threats that use a cyberspace
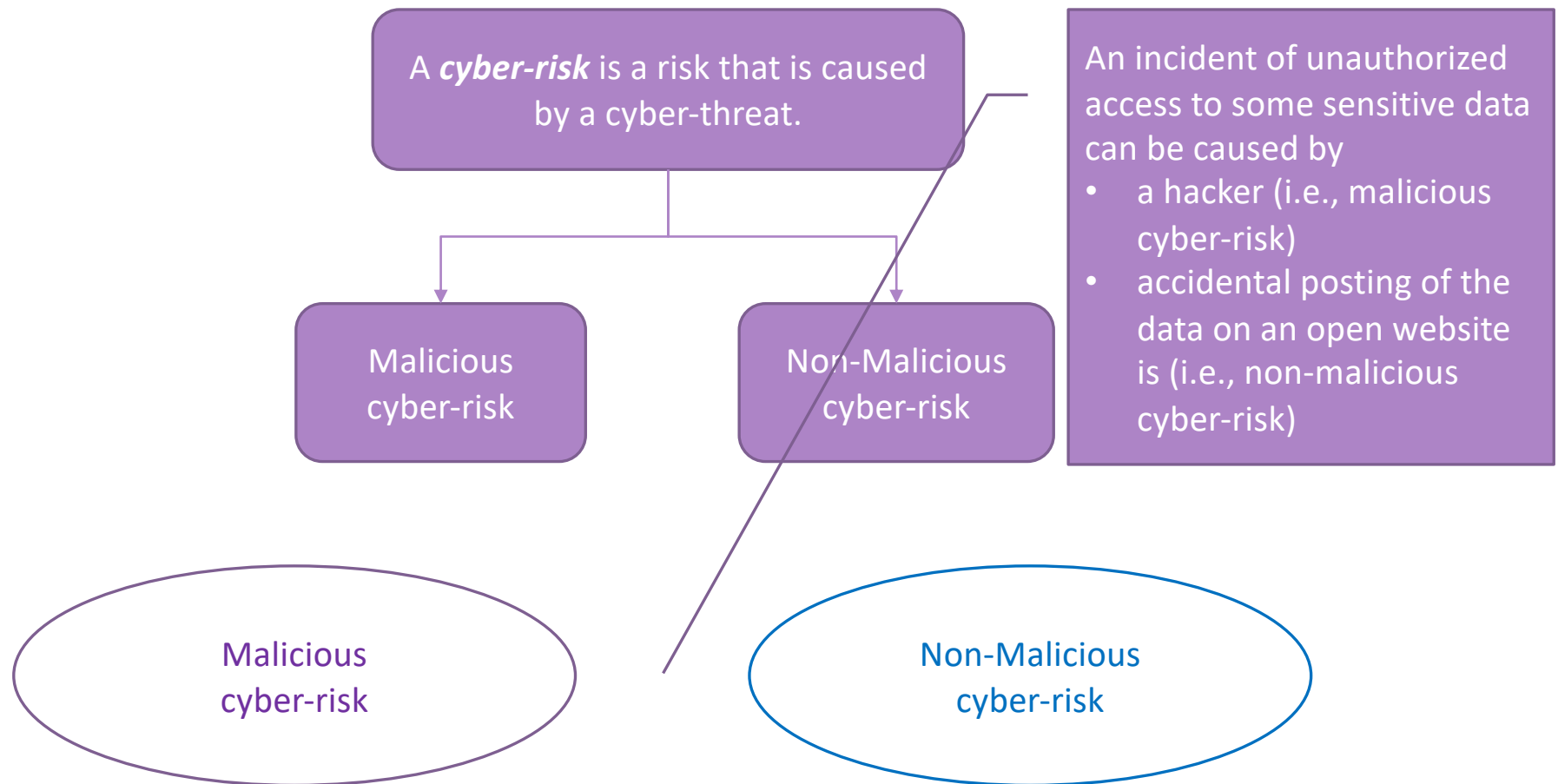
# How Does Cybersecurity Relate to Critical Infrastructure Protection

*Critical Infrastructure Protection* (CIP) and *Critical Information Infrastructure Protection* (CIIP), is concerned with the prevention of the disruption, disabling, destruction, or malicious control of infrastructure
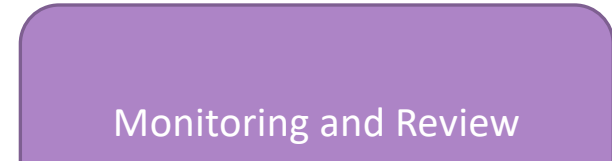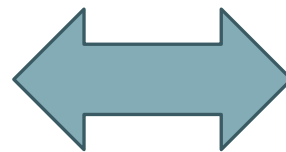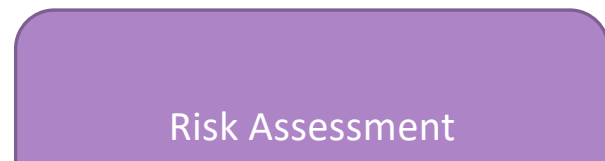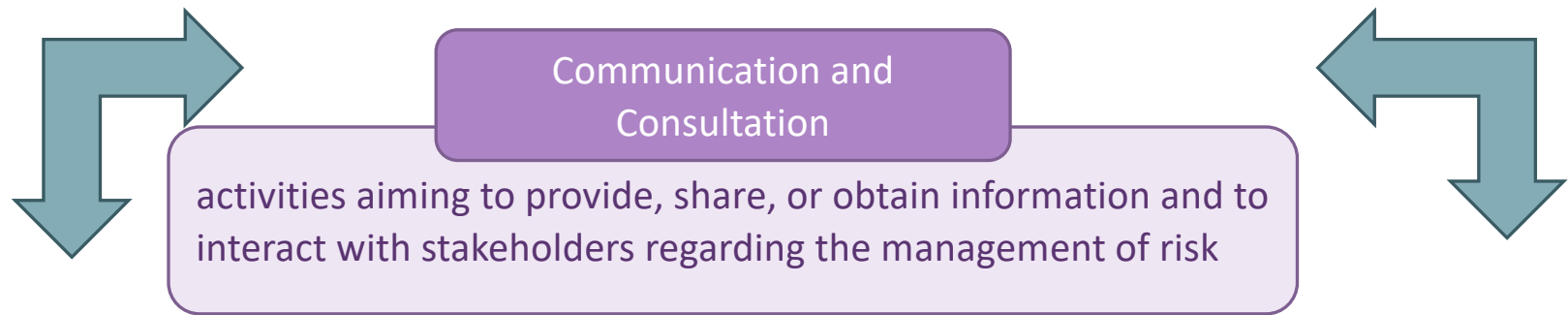
# Cyber-Risk

A **cyber-risk** is a risk that is caused by a cyber-threat.

Malicious cyber-risk

Non-Malicious cyber-risk

An incident of unauthorized access to some sensitive data can be caused by
- a hacker (i.e., malicious cyber-risk)
- accidental posting of the data on an open website is (i.e., non-malicious cyber-risk)

Malicious cyber-risk

Non-Malicious cyber-risk

# Recap: Risk Management General Process

It is composed by 3 main sub-processes

**Communication and Consultation**

activities aiming to provide, share, or obtain information and to interact with stakeholders regarding the management of risk

**Risk Assessment**

activities aiming to understand and document the risk picture for specific parts or aspects of a system or an organization

**Monitoring and Review**

Check the current status to identify deviations from the expected one and determine the effectiveness of the risk management process, framework, risks and treatments

# Communication and Consultation of Cyber-risk

Additional challenges are issued when considering cyber-systems

1. cyber-systems may potentially have stakeholders everywhere

2. there may potentially be adversaries everywhere

# An help comes form...

Repositories of up-to-date information regarding, for example,

- cyber-threats,
- vulnerabilities and incidents,
- potential and confirmed adversary profiles,
- current strategies and mechanisms for cyber-risk mitigation
- ...

# Why cyber-risk assessment is different from generic risk assessment?

1. the potentially far-reaching extent of a cyberspace implies that also the origins of threats are widespread, possibly global

2. the number of potential threat sources and threats, both malicious and non-malicious, is very large



## Need to define procedures and techniques that provide guidance and direction

# Cyber-Risk Assessment
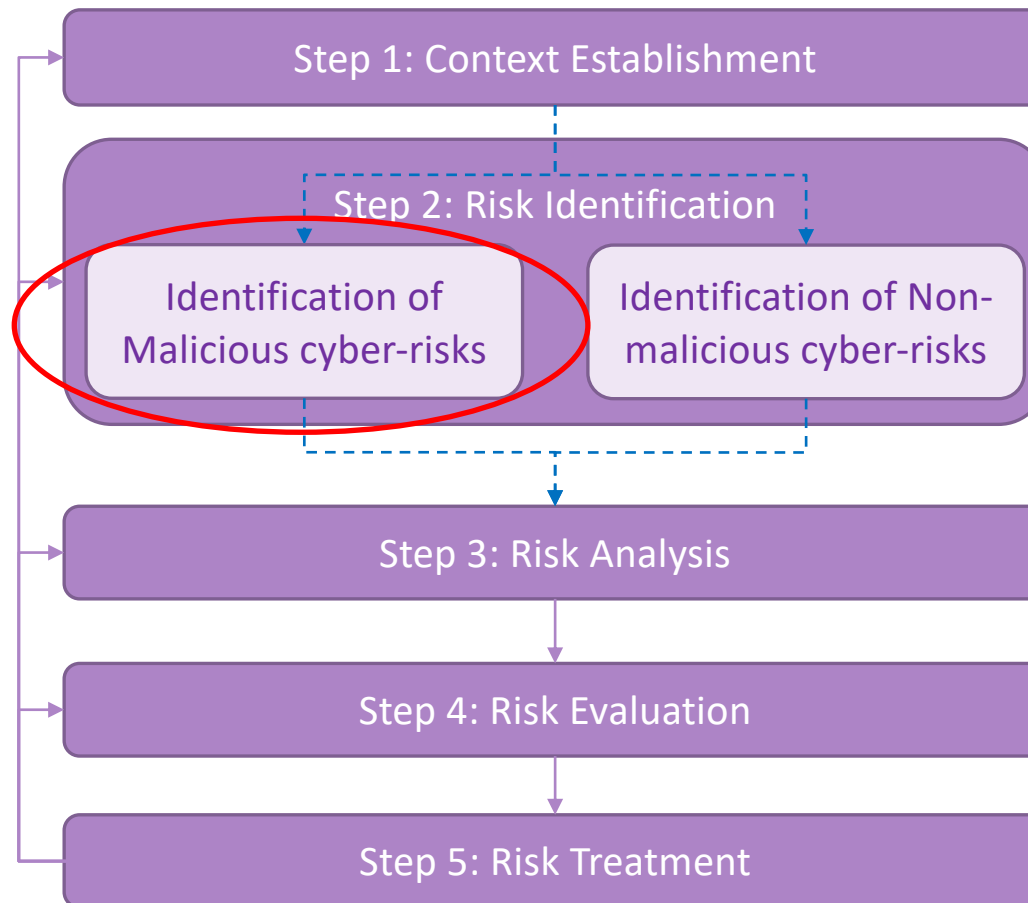
# Context Establishment for Cyber-risk

Understanding and documenting the interface to the cyberspace is important for cyber-risk management in general and for identification of cyber-risks in particular

The **attack surface** is all of the different points where an attacker or other threat source could get into the cyber-system, and where information or data can get out

Typical assets of concern in the setting of cyber-risk assessment are:
◦ information
◦ information infrastructures (including software, services, and networks)
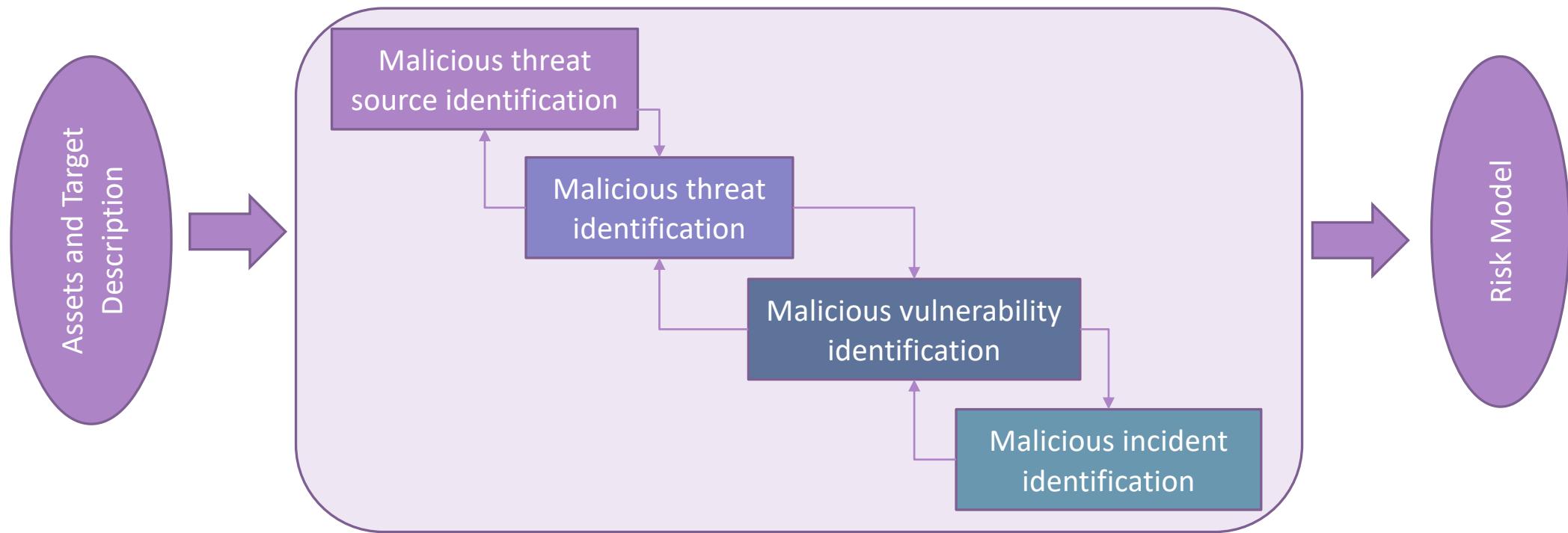
# Identification of Malicious Cyber-risk

Observation
What we can expect from the adversary depends
on the (i) motives, (ii) abilities, (iii) helpers and
(iv) resources available to the adversary

Risk Assessor

Attacker

Defender
(target and a set of assets)

# Identification of Malicious Cyber-risk

# Malicious threat source identification

It is important to understand

  ◦ who may want to initiate attacks

  ◦ what motivates them

  ◦ what their capabilities and intentions are

  ◦ how attacks can be launched

Threat and threat sources as not just external!
Consider that they may be inside your cyber-system

# Malicious threat identification

Pay particular attention to the interface to the cyberspace and the documented attack surface

Make active use of the description of the target of assessment, investigating where and how attacks can be launched

**Examples of helpful catalogues and repositories that concern cybersecurity (and cyber-threats in particular) are those that are provided by MITRE and OWASP, NIST etc.**

# Malicious vulnerability identification

By investigating existing controls and defence mechanisms to determine their strength and adequacy with respect to the identified threats and assets

By running  security testing i.e., penetration testing and vulnerability scanning
- to check whether or how easily a specific threat source can actually launch an attack
- to investigate the severity of known vulnerabilities,
- to look for potential vulnerabilities, and
- to look for possible incidents that the malicious threats may lead to

# Malicious incident identification

investigating how the malicious threats can cause harm to the identified assets given the identified vulnerabilities

Help and guidance come from
- event logs about previous incidents of relevance
- investigation of the kinds of incidents that the threats and vulnerabilities may lead to based on the result of penetration testing activities

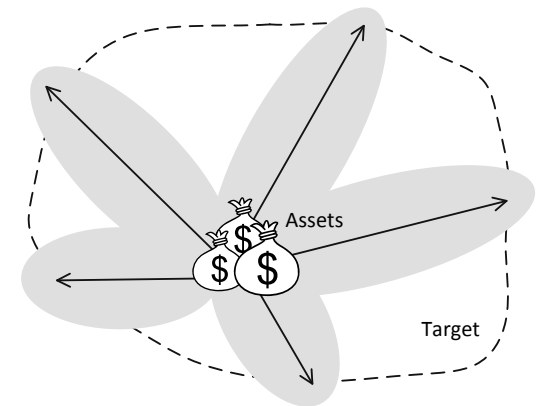# Identification of Non-malicious Cyber-risk

Normally there is no intent or motive behind non-malicious risks...

...It is not practical to start by identifying and documenting threat sources

It is recommend to start from the valuables to be defended

# Identification of Non-malicious Cyber-risk

1. Answer to the question **"In which way the considered asset may be directly harmed?"**
   - Each possibility corresponds to an incident

2. Identify the vulnerabilities and threats that may cause these incidents
   - focus only on the parts and aspects of the target that are of relevance to the identified incidents

3. identify the non-malicious threat sources that can cause the threats.

Assets

Target

# Identification of Non-malicious Cyber-risk
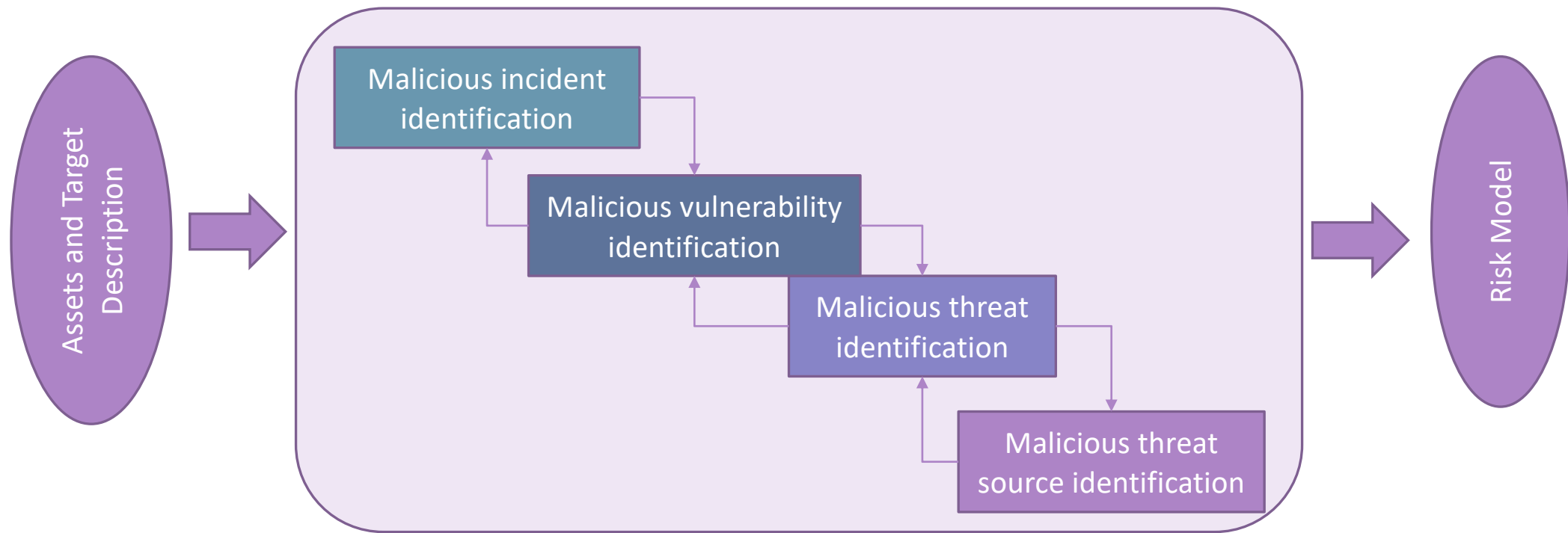
# Non-malicious Incident Identification

Start by investigating how the assets are represented and how they are related to the target of assessment

Accidents and unintended acts are often recurring and known

Use logs, monitored data, and other historical data to support the identification

# Non-malicious vulnerability identification

It is possible to investigate technical parts of the target of assessment, as well as the culture, routines, awareness, and so forth of the organization and personnel in question.

Open sources, such as the ISO 27005 standard, come with lists of typical vulnerabilities.

# ISO 27500 Annex Example

**Annex D**
(informative)

**Vulnerabilities and methods for vulnerability assessment**

**D.1  Examples of vulnerabilities**

The following table gives examples for vulnerabilities in various security areas, including examples of threats that might exploit these vulnerabilities. The lists can provide help during the assessment of threats and vulnerabilities, to determine relevant incident scenarios. It is emphasized that in some cases other threats may exploit these vulnerabilities as well.

| Types | Examples of vulnerabilities | Examples of threats |
|---|---|---|
| Hardware | Insufficient maintenance/faulty installation of storage media | Breach of information system maintainability |
| | Lack of periodic replacement schemes | Destruction of equipment or media |
| | Susceptibility to humidity, dust, soiling | Dust, corrosion, freezing |
| | Sensitivity to electromagnetic radiation | Electromagnetic radiation |
| | Lack of efficient configuration change control | Error in use |
| | Susceptibility to voltage variations | Loss of power supply |

# Non-malicious threat identification

We need to answer to the following question:

*Which unintended events may lead to the identified incidents due to the identified vulnerabilities, and how?*

We need to carefully consider the interface to the cyberspace to identify non-malicious threats that arise outside of the system.

Relevant sources on typical threats include, for example, the ISO 27005 standard and the NIST risk assessment guide, which provide representative examples of non-malicious threats.

# ISO 27005 Annex Example

**Annex C**
(informative)

**Examples of typical threats**

The following table gives examples of typical threats. The list can be used during the threat assessment process. Threats may be deliberate, accidental or environmental (natural) and may result, for example, in damage or loss of essential services. The following list indicates for each threat type where D (deliberate), A (accidental), E (environmental) is relevant. D is used for all deliberate actions aimed at information assets, A is used for all human actions that can accidentally damage information assets, and E is used for all incidents that are not based on human actions. The groups of threats are not in priority order.

| Type | Threats | Origin |
|---|---|---|
| Physical damage | Fire | A, D, E |
| | Water damage | A, D, E |
| | Pollution | A, D, E |
| | Major accident | A, D, E |
| | Destruction of equipment or media | A, D, E |
| | Dust, corrosion, freezing | A, D, E |
| Natural events | Climatic phenomenon | E |
| | Seismic phenomenon | E |
| | Volcanic phenomenon | E |

# Non-malicious threat source identification

*Who are the users of the system, and how can they cause the unintended or accidental events?*

Consider non-human threat sources, such as failure of hardware or other technical components, wear and tear, acts of nature, and so forth.

Event logs and historical data aid the identification of non-malicious threat sources, as do open sources such as the abovementioned ISO standard and NIST guide, which both provide categorized lists.

# Analysis of Cyber-risk

There are two main aspects that distinguish the analysis of cyber-risk from risk analysis in general.

1. for malicious threats behind which there is human intent and motive, it can be hard to estimate the likelihood of occurrence.

2. due to the nature of cyber-systems we have several options for logging, monitoring, and testing that can facilitate the analysis.

Use techniques for threat modelling to describe aspects such as attack prerequisites, attacker skills or knowledge required, resources required, attacker motive, attack opportunity, and so forth.

Similar descriptions can be made for vulnerabilities, such as ease of discovery and ease of exploit.

In combination, this information can be used to derive likelihoods of threats and incidents

# Evaluation of Cyber-risk

4 main steps (not too much different from the general case)

| | |
|---|---|
| **Consolidation of risk analysis results** | • Similarly to the general case, focus on the cyber-risks for which the estimates are uncertain<br>• Make the distinction between malicious and non-malicious risks |
| **Evaluation of risk level** | for convenience it is possible to evaluate malicious and non-malicious cyber-risks separately |
| **Risk aggregation** | As in the general case |
| **Risk grouping** | Risks are grouped based on the distinction between malicious and non malicious cyber-risk to improve the selection of the most appropriate treatments |

# Treatment of Cyber-risk

There are two features in particular that distinguish the risk treatment of cyber-systems from the general case:

1.  the highly technical nature of cyber-systems means that the options for risk treatment are also technical
    ◦ There is also the need to consider the sociotechnical aspects and human involvement.

2.  the distinction between malicious and non-malicious cyber-risks has implications for the most adequate risk treatment

# MITRE CAPEC[1]

*CAPEC is a publicly available catalogue of attack patterns along with a comprehensive schema and classification taxonomy created to assist in the building of secure software*

**CAPEC** Common Attack Pattern Enumeration and Classification
A Community Resource for Identifying and Understanding Attacks

ID Lookup: [_____] Go

**Home** | **About** | **CAPEC List** | **Community** | **News** | **Search**

Understanding how the adversary operates is essential to effective cyber security. CAPEC™ helps by providing a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defenses.

## View the List of Attack Patterns

( by Mechanisms of Attack )     ( by Domains of Attack )

## Search CAPEC

Easily find a specific attack pattern by performing a search of the CAPEC List by keywords(s) or by CAPEC-ID Number. To search by multiple keywords, separate each by a space.

[ Google Custom Search ]     [ 🔍 ]

# MITRE CAPEC

Understanding how the adversary operates is essential to effective cyber security.

CAPEC helps by providing a comprehensive dictionary of known **patterns of attack** employed by adversaries to exploit known weaknesses in cyber-enabled capabilities.

◦ "Attack Patterns" are descriptions of common methods for exploiting software providing the attacker's perspective and guidance on ways to mitigate their effect

It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defences

# CAPEC Attack Pattern Example

| Name | HTTP Response Splitting |
|---|---|
| **Typical Severity** | High |
| **Description** | HTTP Response Splitting causes a vulnerable web server to respond to a maliciously crafted request by sending an HTTP response stream such that it gets interpreted as two separate responses instead of a single one. This is possible when user-controlled input is used unvalidated as part of the response headers. An attacker can have the victim interpret the injected header as being a response to a second dummy request, thereby causing the crafted contents to be displayed and possibly cached. To achieve HTTP Response Splitting on a vulnerable web server, the attacker: 1. Identifies the user-controllable input that causes arbitrary HTTP header injection. 2. Crafts a malicious input consisting of data to terminate the original response and start a second response with headers controlled by the attacker. 3. Causes the victim to send two requests to the server. The first request consists of maliciously crafted input to be used as part of HTTP response headers and the second is a dummy request so that the victim interprets the split response as belonging to the second request. |
| **Attack Prerequisites** | User-controlled input used as part of HTTP header. Ability of attacker to inject custom strings in HTTP header. Insufficient input validation in application to check for input sanity before using it as part of response header |
| **Typical Likelihood of Exploit** | Medium |
| **Methods of Attack** | Injection. Protocol Manipulation |
| **Examples-Instances** | In the PHP 5 session extension mechanism, a user-supplied session ID is sent back to the user within the Set-Cookie HTTP header. Since the contents of the user-supplied session ID are not validated, it is possible to inject arbitrary HTTP headers into the response body. This immediately enables HTTP Response Splitting by simply terminating the HTTP response header from within the session ID used in the Set-Cookie directive.  CVE-2006-0207 |
| **Attacker Skill or Knowledge Required** | High - The attacker needs to have a solid understanding of the HTTP protocol and HTTP headers and must be able to craft and inject requests to elicit the split responses. |
| **Resources Required** | None |
| **Probing Techniques** | With available source code, the attacker can see whether user input is validated or not before being used as part of output. This can also be achieved with static code analysis tools. If source code is not available, the attacker can try injecting a CR-LF sequence (usually encoded as %0d%0a in the input) and use a proxy such as Paros to observe the response. If the resulting injection causes an invalid request, the web server may also indicate the protocol error. |

# CAPEC Attack Pattern Example

| | |
|---|---|
| **Indicators-Warnings of Attack** | The only indicators are multiple responses to a single request in the web logs. However, this is difficult to notice in the absence of an application filter proxy or a log analyzer. There are no indicators for the client |
| **Solutions and Mitigations** | To avoid HTTP Response Splitting, the application must not rely on user-controllable input to form part of its output response stream. Specifically, response splitting occurs due to injection of CR-LF sequences and additional headers. All data arriving from the user and being used as part of HTTP response headers must be subjected to strict validation that performs simple character-based as well as semantic filtering to strip it of malicious character sequences and headers. |
| **Attack Motivation-Consequences** | Execute unauthorized code or commands<br>Gain privileges/assume identity |
| **Context Description** | HTTP Response Splitting attacks take place where the server script embeds user-controllable data in HTTP response headers. This typically happens when the script embeds such data in the redirection URL of a redirection response (HTTP status code 3xx), or when the script embeds such data in a cookie value or name when the response sets a cookie. In the first case, the redirection URL is part of the Location HTTP response header, and in the cookie setting, the cookie name/value pair is part of the Set- Cookie HTTP response header. |
| **Injection Vector** | User-controllable input that forms part of output HTTP response headers |
| **Payload** | Encoded HTTP header and data separated by appropriate CR-LF sequences. The injected data must consist of legitimate and well-formed HTTP headers as well as required script to be included as HTML body. |
| **Activation Zone** | API calls in the application that set output response headers. |
| **Payload Activation Impact** | The impact of payload activation is that two distinct HTTP responses are issued to the target, which interprets the first as response to a supposedly valid request and the second, which causes the actual attack, to be a response to a second dummy request issued by the attacker. |
| **CIA Impact** | Confidentiality Impact: High    Integrity Impact: High    Availability Impact: Low |
| **Related Weaknesses** | CWE113  -  HTTP Response Splitting  -  Targeted<br>CWE74  -  Injection  -  Secondary<br>CWE697 - Insufficient Comparison - Targeted<br>CWE707 - Improper Enforcement of Message or Data Structure - Targeted<br>CWE713 - OWASP Top Ten 2007 Category A2 - Injection Flaws - Targeted |
| **Relevant Security Requirements** | All client-supplied input must be validated through filtering and all output must be properly escaped. |
| **Related Security Principles** | Reluctance to Trust |
| **Related Guidelines** | Never trust user-supplied input. |
| **References** | G. Hoglund and G. McGraw. Exploiting Software: How to Break Code. Addison-Wesley, February 2004. |
| **For enhanced descriptions of this example CAPEC-ID, see http://capec.mitre.org/data/definitions/34.html.** | |

# MITRE CWE

*CWE is a community-developed list of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.*

# MITRE CWE Example

**CWE Entries include:**

- name of the weakness type
- description of the type
- alternate terms for the weakness
- description of the behavior of the weakness
- description of the exploit of the weakness
- likelihood of exploit for the weakness
- description of the consequences of the exploit
- potential mitigations
- node relationship information
- source taxonomies
- code samples for the languages/architectures

| CWE ID 415 | Double Free |
|---|---|
| Description | The product calls free() twice on the same memory address, potentially leading to modification of unexpected memory locations. |
| Likelihood of Exploit | Low to Medium |
| Common Consequences | Access control: Doubly freeing memory may result in a write-what-where condition, allowing an attacker to execute arbitrary code. |
| Potential Mitigations | Architecture and Design: Choose a language that provides automatic memory management. <br> Implementation: Ensure that each allocation is freed only once. After freeing a chunk, set the pointer to NULL to ensure the pointer cannot be freed again. In complicated error conditions, be sure that clean-up routines respect the state of allocation properly. If the language is object oriented, ensure that object destructors delete each chunk of memory only once. <br> Implementation: Use a static analysis tool to find double free instances. |
| Demonstrative Examples | Example 1: The following code shows a simple example of a double free vulnerability. Double free vulnerabilities have two common (and sometimes overlapping) causes: - Error conditions and other exceptional circumstances - Confusion over which part of the program is responsible for freeing the memory Although some double free vulnerabilities are not much more complicated than the previous example, most are spread out across hundreds of lines of code or even different files. Programmers seem particularly susceptible to freeing global variables more than once. <br> Example 2: While contrived, this code should be exploitable on Linux distributions which do not ship with heap-chunk check summing turned on. |
| Observed Examples | CVE-2002-0059 - Double free from malformed compressed data. <br> CVE-2003-0545 - Double free from invalid ASN.1 encoding. <br> CVE-2003-1048 - Double free from malformed GIF. <br> CVE-2004-0642 - Double free resultant from certain error conditions. <br> CVE-2004-0772 - Double free resultant from certain error conditions. <br> CVE-2005-0891 - Double free from malformed GIF. <br> CVE-2005-1689 - Double free resultant from certain error conditions. |
| Node Relationships | Child Of - Operation on Resource in Wrong Phase of Lifetime (666) in View (1000) <br> Child Of - Duplicate Operations on Resource (675) in View (1000) <br> Child Of - Resource Management Errors (399) in View (699) <br> Peer Of - Use After Free (416) in View (699 & 1000) <br> Peer Of - Write-what-where Condition (123) in View (700) <br> Child Of - Indicator of Poor Code Quality (398) in View (700) <br> Child Of - Weaknesses that Affect Memory (633) in View (631) <br> Child Of - CERT C Secure Coding Section 08 – Memory Management (MEM) (742) in View (734) <br> Member Of - Weaknesses Examined by SAMATE (630) in View (630) <br> Peer Of - Signal Handler Race Condition (364) in View (1000) |

# Vulnerability Classification

Vulnerabilities may affect a system at different levels:
- Hardware, Software, Network, Personnel, Organizational…

Mitre Corporation maintains a list of disclosed vulnerabilities in a system called Common Vulnerabilities and Exposures (CVE), where vulnerability are classified (scored) using Common Vulnerability Scoring System (CVSS).

NIST collects and makes available scored vulnerabilities rough the NVD data-base (https://nvd.nist.gov/cvss.cfm)

# CVE Example

# Common Vulnerability Scoring System (CVSS)

The Common Vulnerability Scoring System (CVSS) captures the principal technical characteristics of software, hardware and firmware vulnerabilities

Its outputs include numerical scores indicating the severity of a vulnerability relative to other vulnerabilities

CVSS is composed of three metric groups:

◦ **Base metrics**: reflect the severity of a vulnerability according to its intrinsic characteristics which are constant over time and assumes the reasonable worst case impact across different deployed environments

◦ **Temporal metrics**: adjust the Base severity of a vulnerability based on factors that change over time, such as the availability of exploit code

◦ **Environmental metrics**: adjust the Base and Temporal severities to a specific computing environment. They consider factors such as the presence of mitigations in that environment

# CVSS 3.1
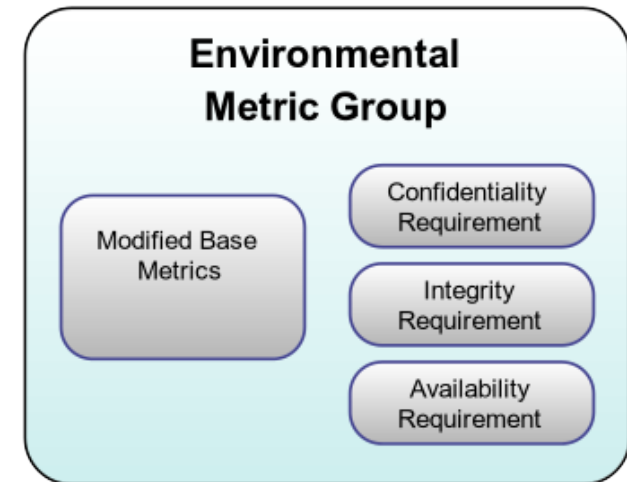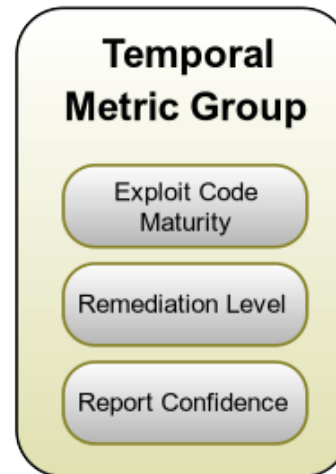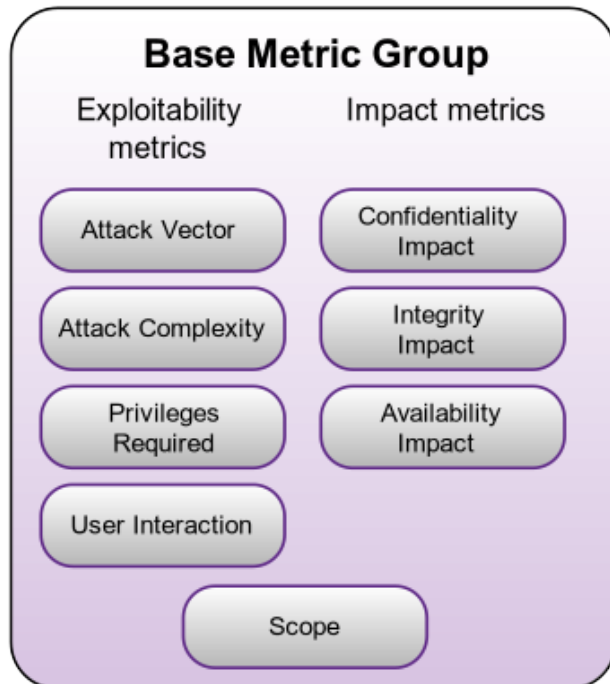
# CVSS Scoring

When the Base metrics are assigned values by an analyst, the Base equation computes a score ranging from 0.0 to 10.0
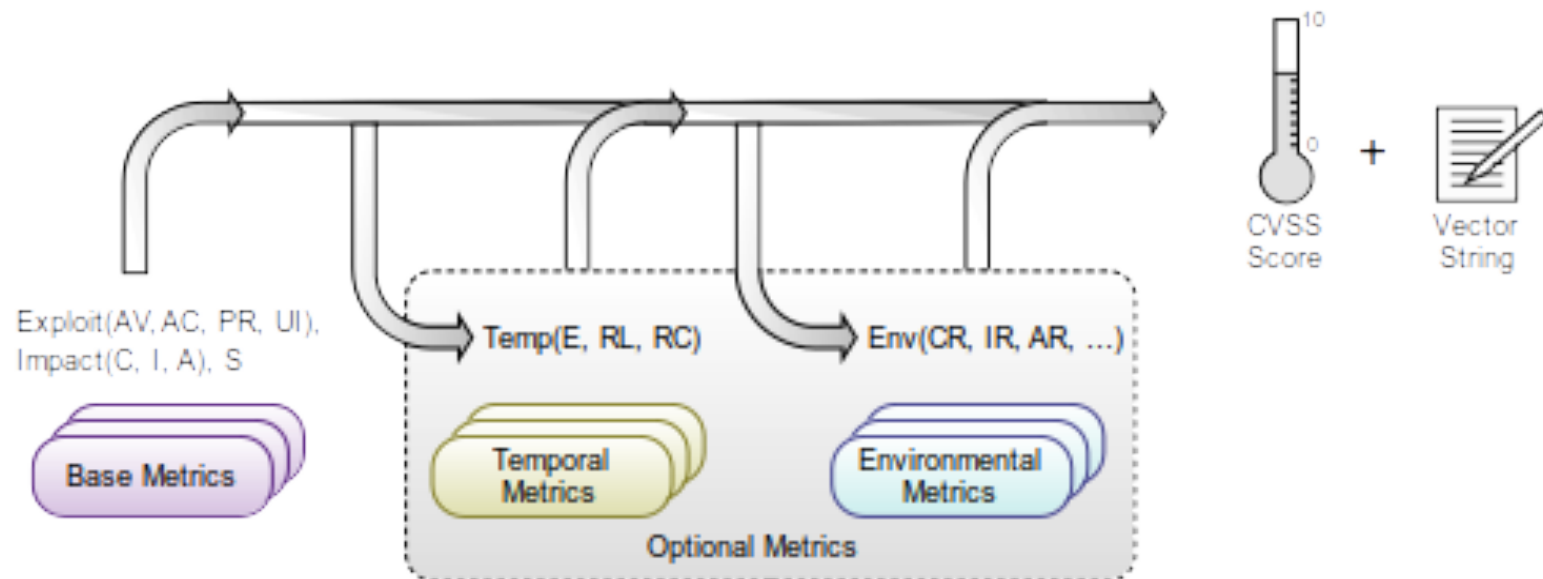
# Common Weakness Scoring System (CWSS)

The Common Weakness Scoring System (CWSS) provides a mechanism for prioritizing software weaknesses in a consistent, flexible, open manner

It is a collaborative, community-based effort that is addressing the needs of its stakeholders across government, academia, and industry

CWSS is distinct from (but not a competitor to) the Common Vulnerability Scoring System (CVSS).

◦ CVSS and CWSS have different roles, and they can be leveraged together

CWSS offers:

◦ Quantitative Measurements of the unfixed weaknesses that are present within a software application

◦ Common Framework for prioritizing security errors ("weaknesses") that are discovered in software applications

◦ Customized Prioritization

# CWSS Structure

CWSS is organized into three metric groups:

◦ Base Finding metric group: captures the inherent risk of the weakness, confidence in the accuracy of the finding, and strength of controls.

◦ Attack Surface metric group: the barriers that an attacker must overcome in order to exploit the weakness.

◦ Environmental metric group: characteristics of the weakness that are specific to a particular environment or operational context.
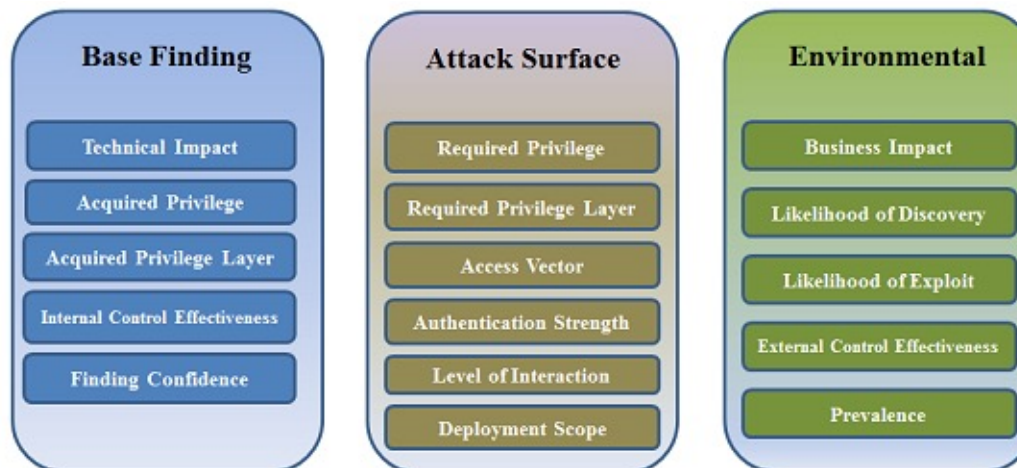
# CWSS Scoring

1. Each factor in the Base Finding metric group is assigned a value
   - These values are converted to associated weights, and a Base Finding sub-score is calculated
   - The Base Finding sub-score can range between 0 and 100

2. The same method is applied to the Attack Surface and Environmental metric group
   - their sub-scores can range between 0 and 1

3. The three sub-scores are multiplied together to get a CWSS score between 0 and 100