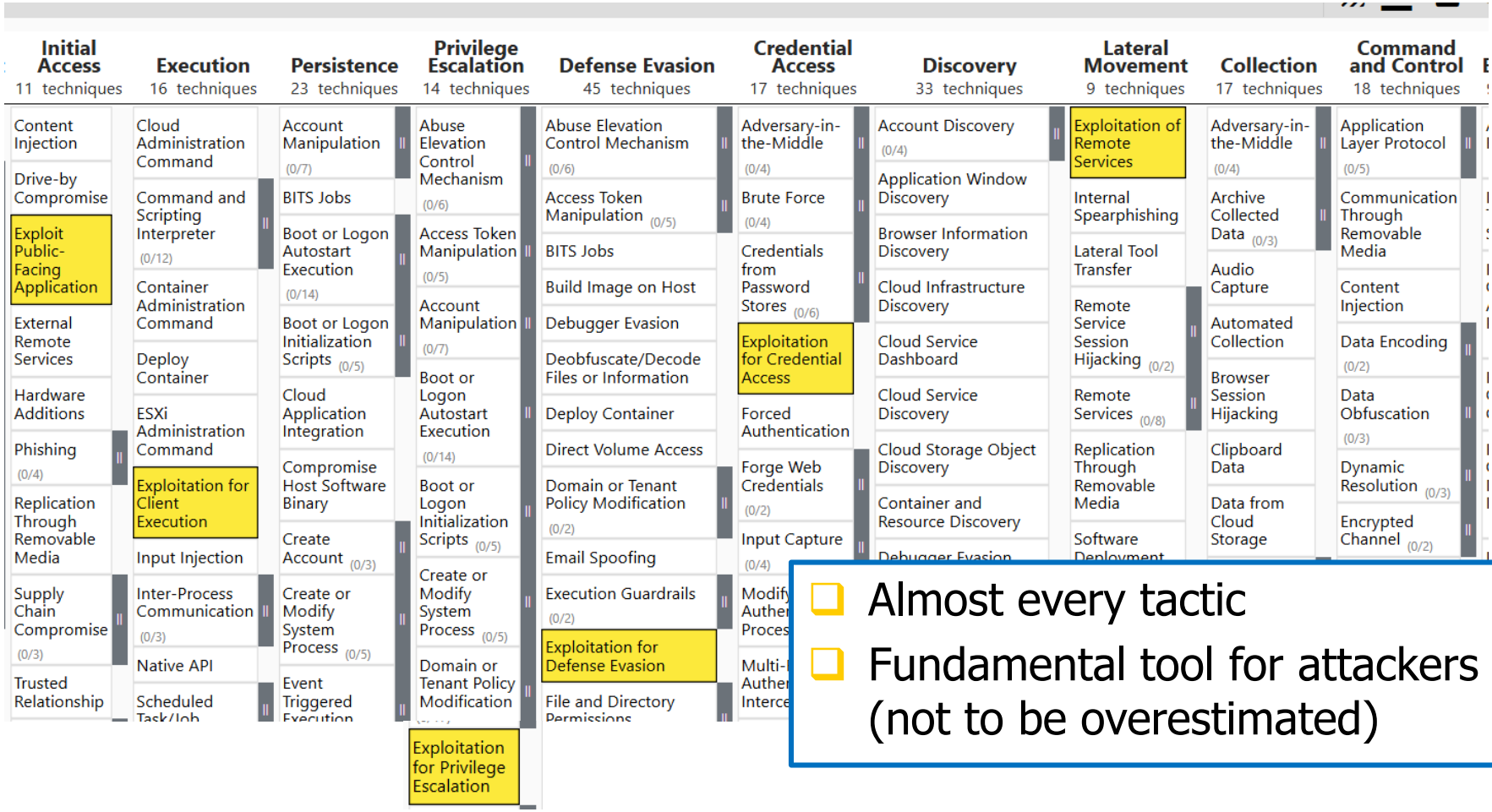
Vulnerabilities: Introduction

Role of Vulnerabilities in MITRE ATT&CK (REMIND)



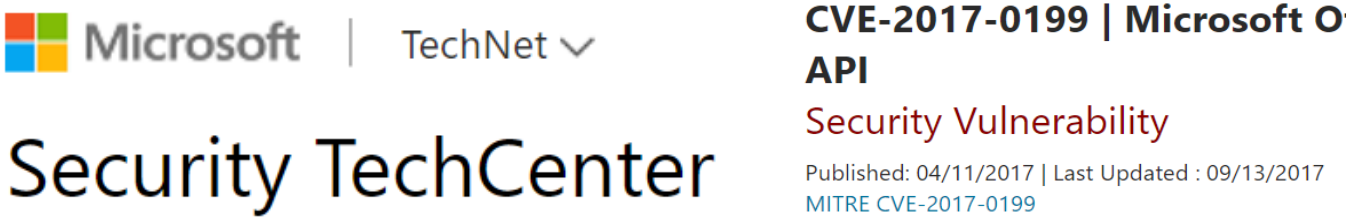
- Almost every tactic
- Fundamental tool for attackers (not to be overestimated)

What is it?

- A **mistake in software** that can be directly used **to gain access** to a system or network

- CVE Common Vulnerabilities and Exposures
  http://cve.mitre.org/

Example: Microsoft (Aprile 2017)



- A ...vulnerability exists in the way that Microsoft Office and WordPad 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

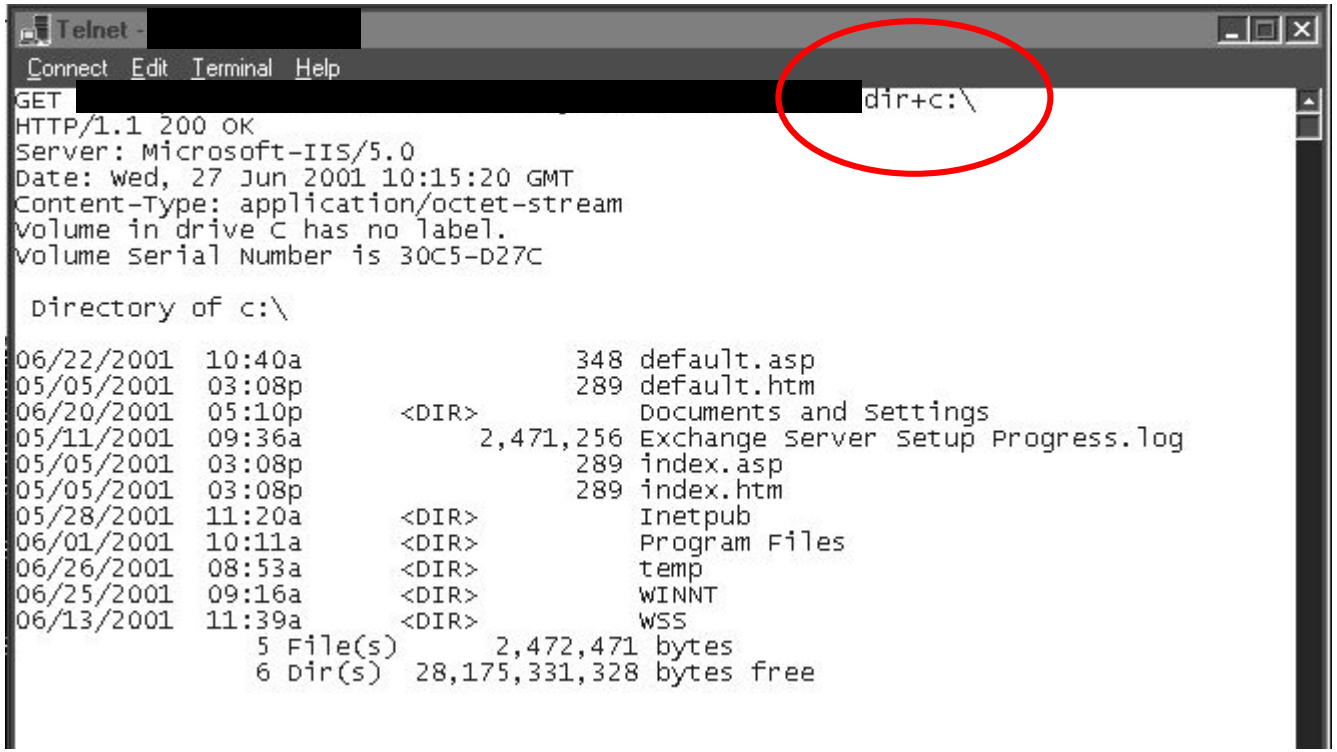
Example (Old but interesting) (I)

HTTP Request
with "long and wrong URL"
ending with command

Execute command

(with server privilege)

Example (Old but interesting) (II)

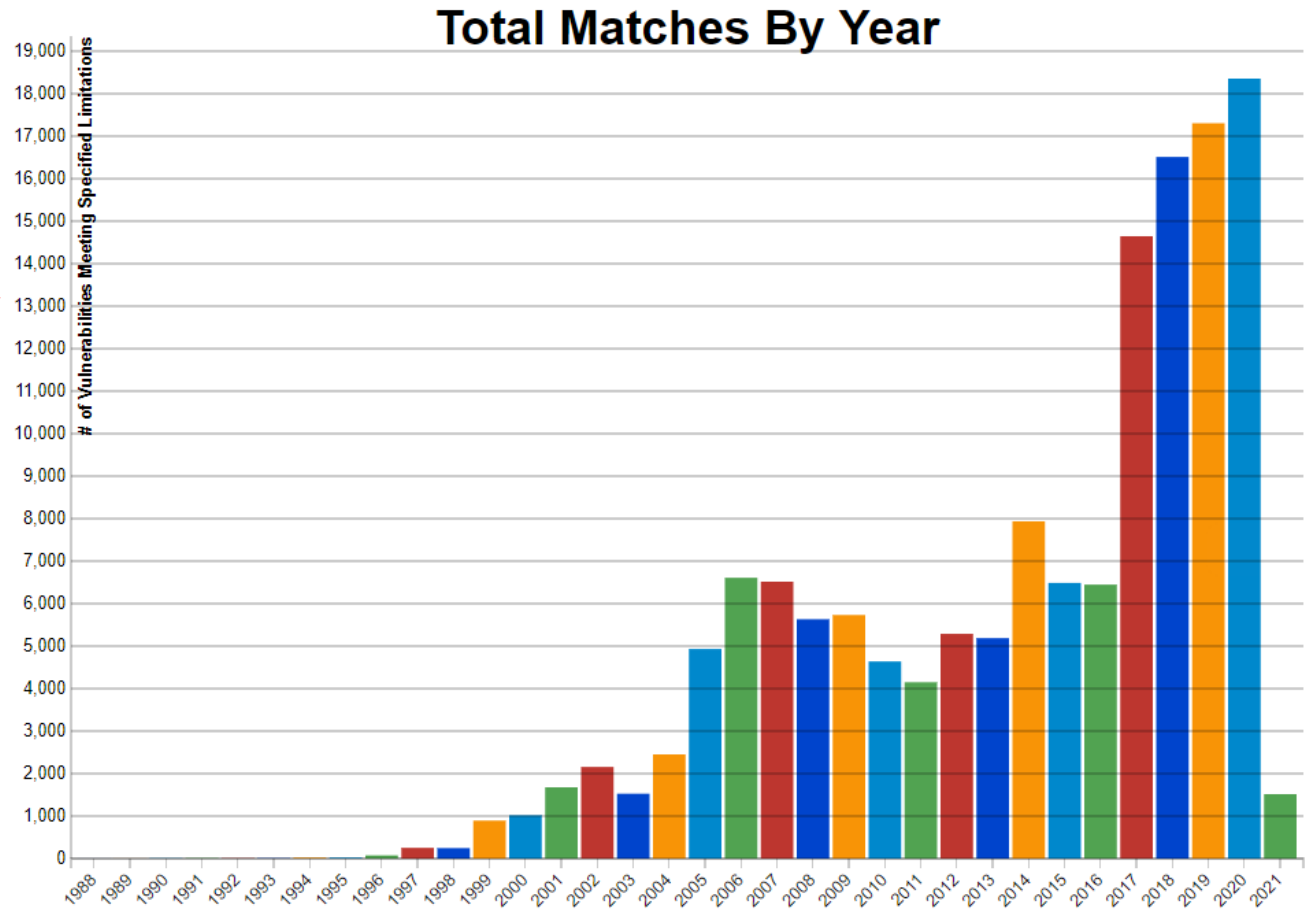


Which Impact?

- **Many** possibilities

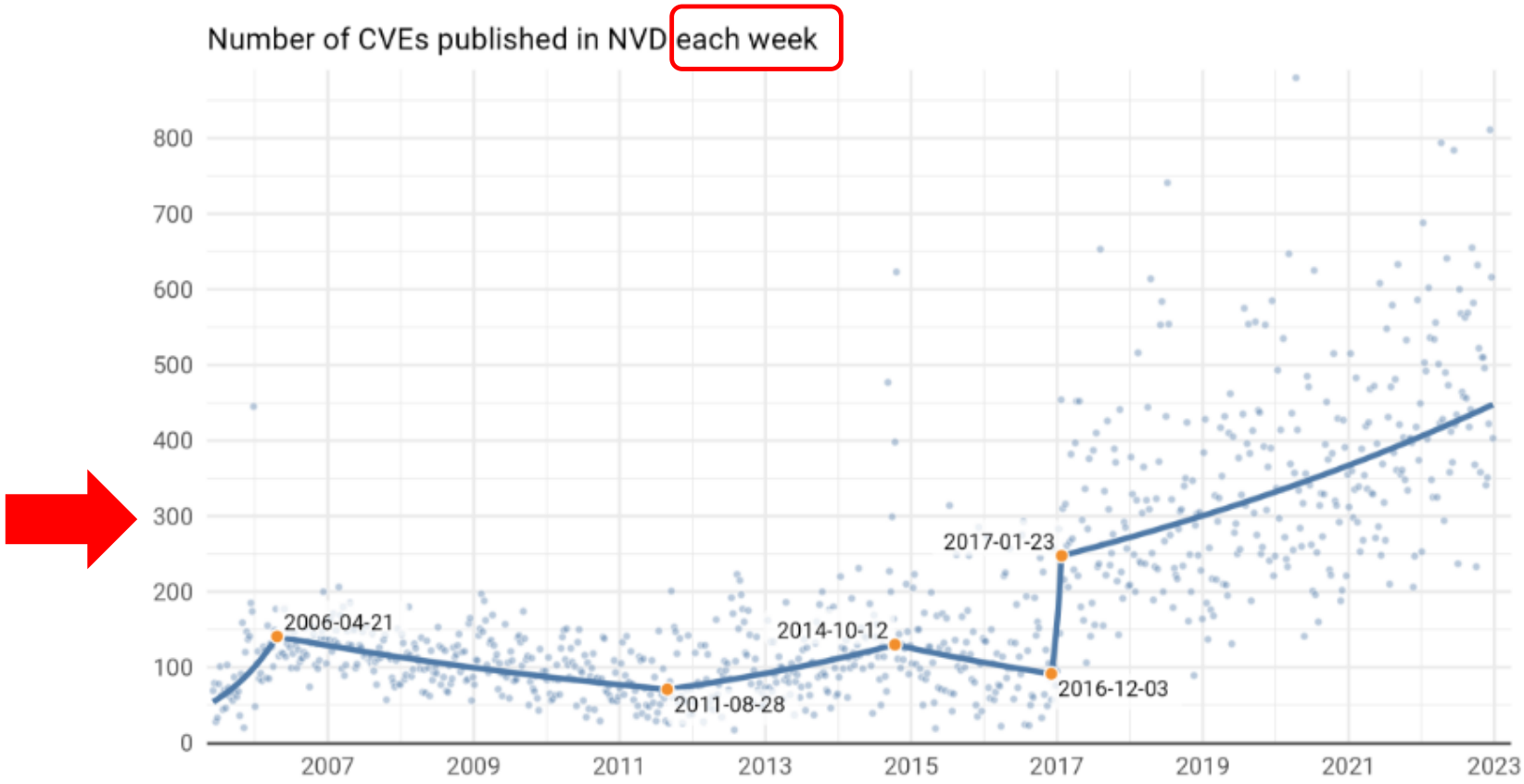
- In a nutshell:
  - Information disclosure
  - Privilege escalation
  - Denial of service
  - Code execution
    - Existing operations (access control vulnerability)
    - Command / code injection

Vulnerabilities: How many? (I)



https://nvd.nist.gov/vuln/search/statistics

Vulnerabilities: How many? (II)



Vulnerabilities: How many? (III)

Computer Security Resource Center

National Vulnerability Database

NIST National Institute of Standards and Technology

Search Parameters:

- Results Type: Overview
- Search Type: Search Last 3 Years
- Keyword (text search): android

There are **3,229** matching records.
Displaying matches **1** through **20**.

Search Parameters:

- Results Type: Overview
- Search Type: Search Last 3 Years
- Keyword (text search): apple

There are **1,770** matching records.
Displaying matches **1** through **20**.

Which software?

- **Every kind of software may suffer of vulnerabilities**
- Period
- SEE APPENDIX FOR SIGNIFICANT EXAMPLES

Takeaway

- NOT a sporadic / occasional phenomenon
- **Intrinsic** feature of software

Vulnerability: A Better Definition

Vulnerability Definition: Hhmmm…(I)

- A mistake in software that can be directly used by a hacker to gain access to a system or network

- What about mistakes in the **design**?
  (as opposed to the implementation)

Example 1

ICS Medical Advisory (ICSMA-21-215-01)

Swisslog Healthcare Translogic PTS

Original release date: August 03, 2021

CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY

Industrial Control Systems

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 buy **one** device, you **immediately** have passwords valid on **all** the devices in the world
- Passwords that **cannot** be modified

- https://cwe.mitre.org/data/definitions/259.html

Example 2

ICS Advisory (ICSA-21-012-02)

Siemens SCALANCE X Switches (Update B)

Original release date: September 14, 2021

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 buy **one** device, you reverse engineer its firmware and have the private key valid on **all** the devices in the world
- Private key that **cannot** be modified

- https://cwe.mitre.org/data/definitions/321.html

IoT Engineers please keep in mind!

- Never ever design systems that embed:
  - **The same secret in all their instances**
  - **A secret that cannot be modified**

Vulnerability Definition: Hhmmm…(II)

- A mistake in software that can be directly used by a hacker to gain access to a system or network

- What about mistakes useful **after** initial access? (e.g., lateral movement, privilege escalation)

- Many, many examples

Vulnerability Definition: Hhmmm…(III)

- A mistake in software that can be directly used by a hacker to gain access to a system or network

- What **about insecure default in configuration**?

- Many, many examples

My Preferred Vulnerability Definition

- A mistake in software that can be directly used by a hacker to gain access to a system or network
- 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**

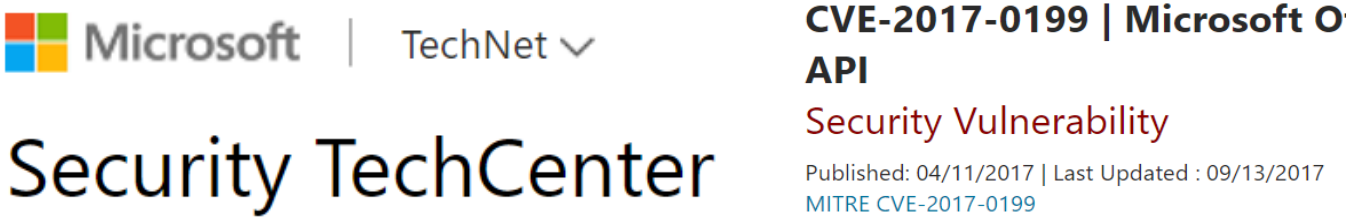
- The mistake may have occurred at **any of several phases**
- The mistake may be exploited for **many purposes**

Exploit and Injection

Exploit

- A software mistake does **not** provoke any damage **by itself.**
- The problem is when **execution** incurs in that mistake
- Always necessary a **carefully constructed input** (**exploit**)

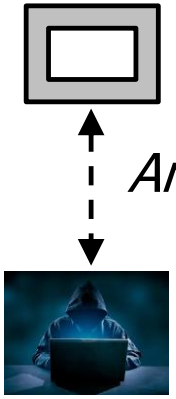
Example: Microsoft (Aprile 2017)



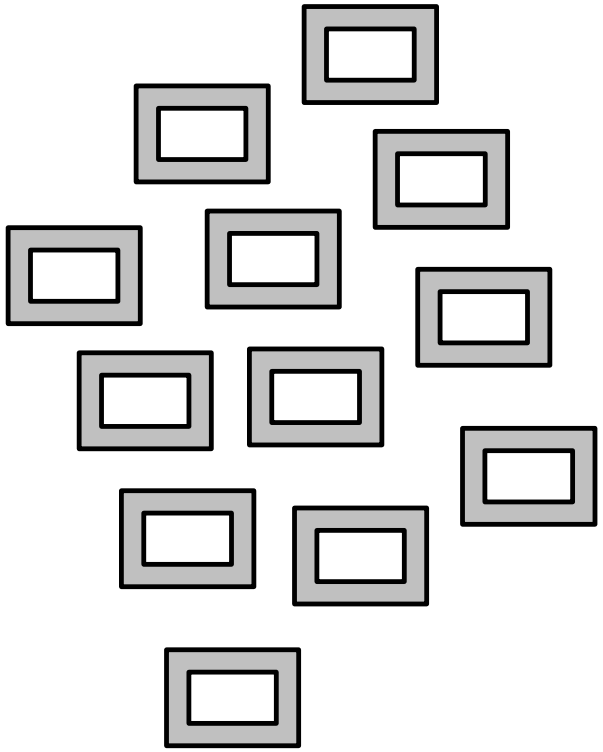
- A ...vulnerability exists in the way that Microsoft Office and WordPad parse specially crafted files
- An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights.

- An attacker could exploit the vulnerability by sending **a specially crafted file** to the user and then convincing the user to open the file

Exploit Development



Analyze & Experiment



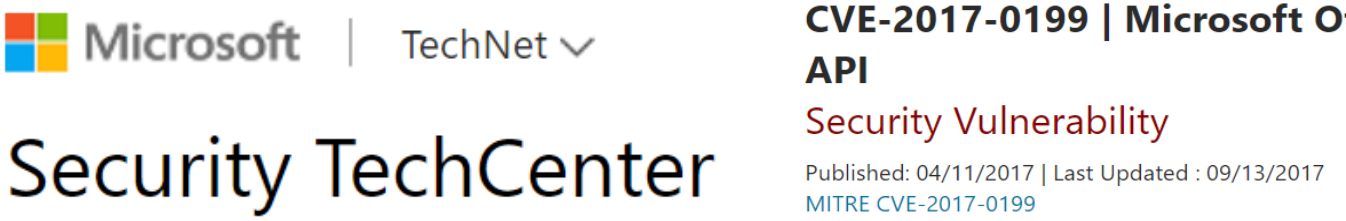
The same sequence of bytes
for **all** instances of vulnerable program

Exploit + **Injection**

- A software mistake does **not** provoke any damage **by itself.**
- The problem is when **execution** incurs in that mistake
- Always necessary a **carefully constructed input** (**exploit**)

- ...and the actual **injection** of the exploit into the vulnerable system

Example: Microsoft (Aprile 2017)



- A ...vulnerability exists in the way that Microsoft Office and WordPad 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**

Injection Categories (I)

- **User action required**:
  - "Successful exploitation of this vulnerability **requires a user to take some action** before the vulnerability can be exploited".
  - Example: open email attachment

- **No user action required**:
  - "The vulnerable system can be exploited **without any interaction from any user**."
  - Example: send network message to server

Example No User Action (March 2017)

Microsoft Security Bulletin MS17-010 - Critical

10/11/2017 • 12 minutes to read • Contributors

Security Update for Microsoft Windows SMB Server (4013389)

Published: March 14, 2017

- An attacker who successfully exploited the vulnerabilities could gain the ability to execute code on the target server.
- To exploit 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)

- To exploit this vulnerability, an attacker would need to **send a specially crafted RPC call to an RPC host**. This could result in remote code execution on the server side with the same permissions as the RPC service.
- The attacker **...** does not require any access to settings or files to carry out an attack.
- The vulnerable system can be exploited without **any interaction from any user**.

OUR categorization for injections (I)

USER ACTION **NOT** REQUIRED	
USER ACTION REQUIRED	

Injection Categories (II)

- **Local**
  - Can be done only by a program **already running on the target**
- **Remote**
  - Can be done by a program running remotely
    - **Authenticated** on the target
    - **Unauthenticated** on the target

Examples Local Injection (2022)

CVE-2022-0847 Detail

Current Description

A flaw was found in the way the "flags" member of the new pipe buffer structure was lacking proper initialization in copy_page_to_iter_pipe and push_pipe functions in the Linux kernel and could thus contain stale values. An unprivileged local user could use this flaw to write to pages in the page cache backed by read only files and as such escalate their privileges on the system.

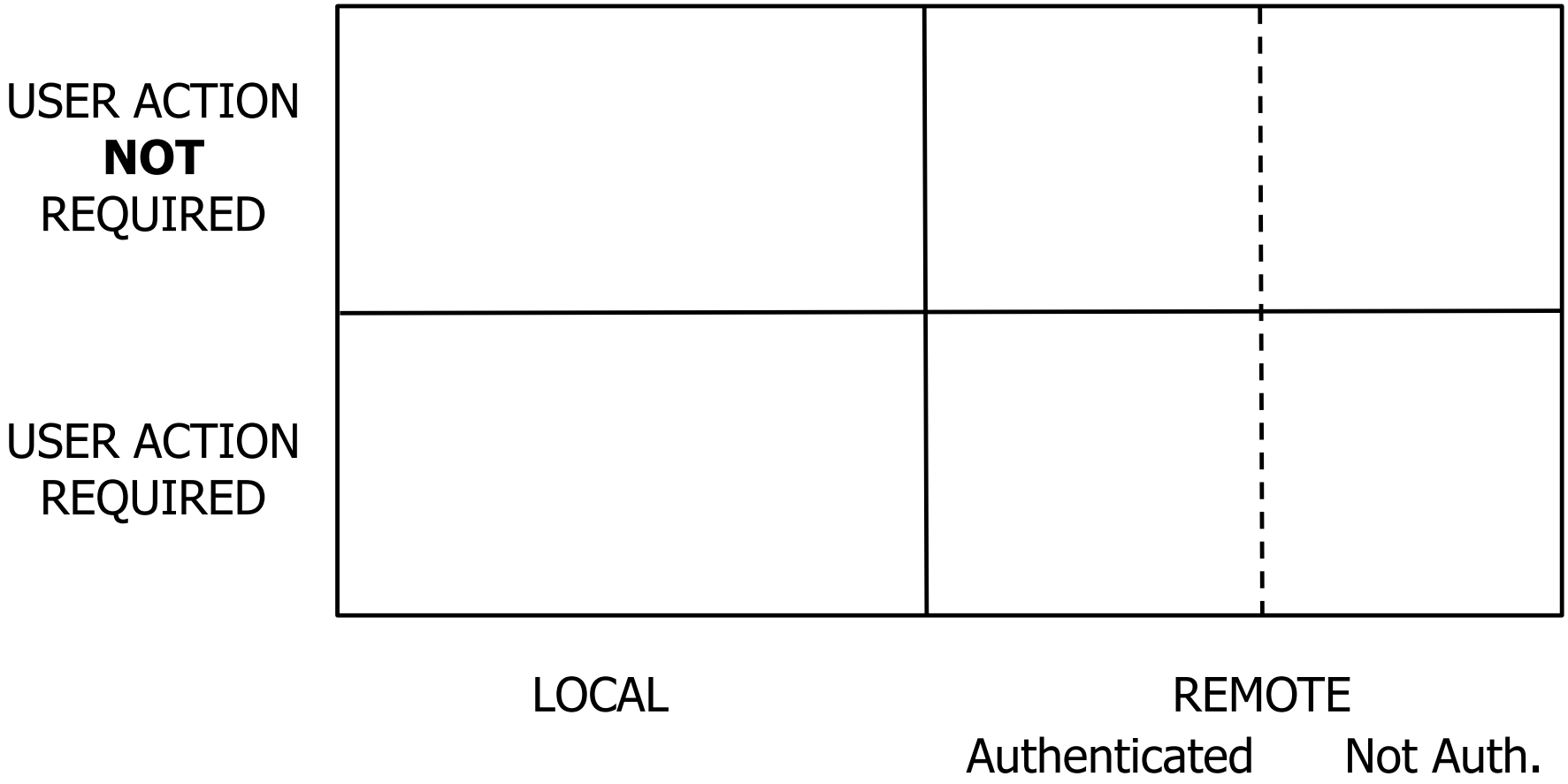
CVE-2021-42103 Detail

Current Description

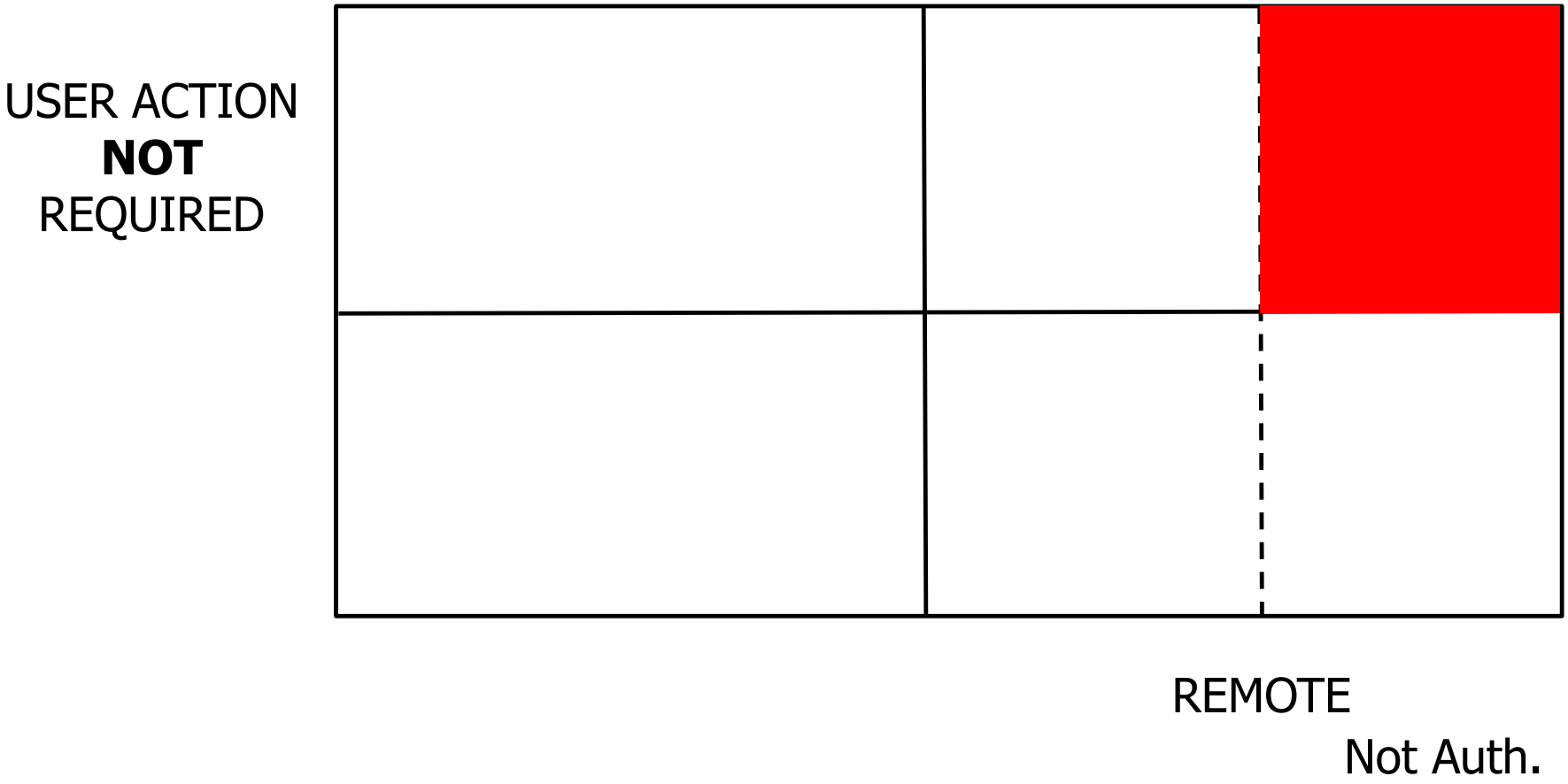
An antivirus!

An uncontrolled search path element vulnerabilities in Trend Micro Apex One and Apex One as a Service could allow a local attacker to escalate privileges on affected installations. An attacker must first obtain the ability to execute low-privileged code on the target system in order to exploit this vulnerability. This vulnerability is similar but not identical to CVE-2021-42101.

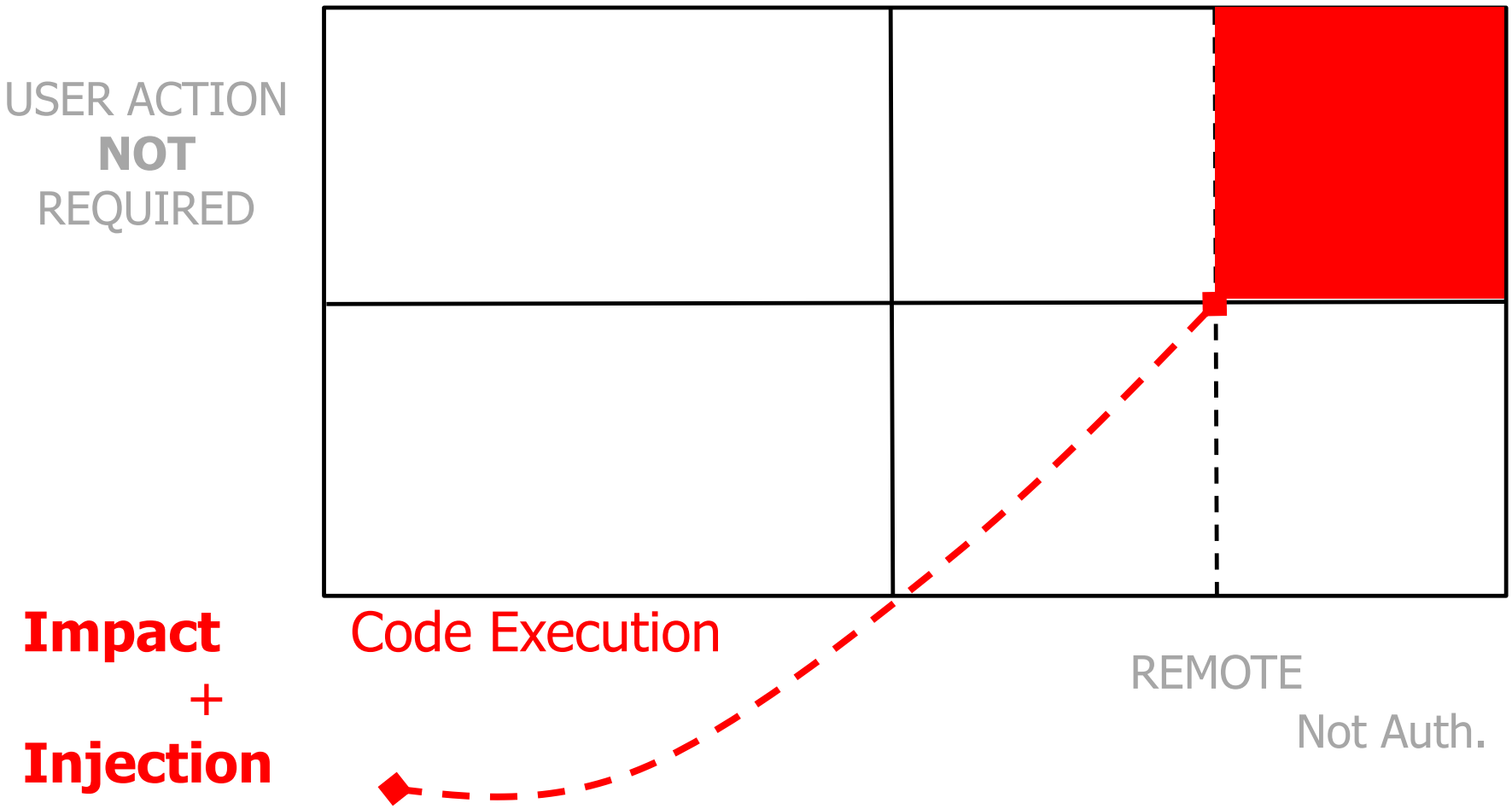
OUR categorization for injections (II)



Very important injection category



Wormable VuIn (Tragedy) (I)



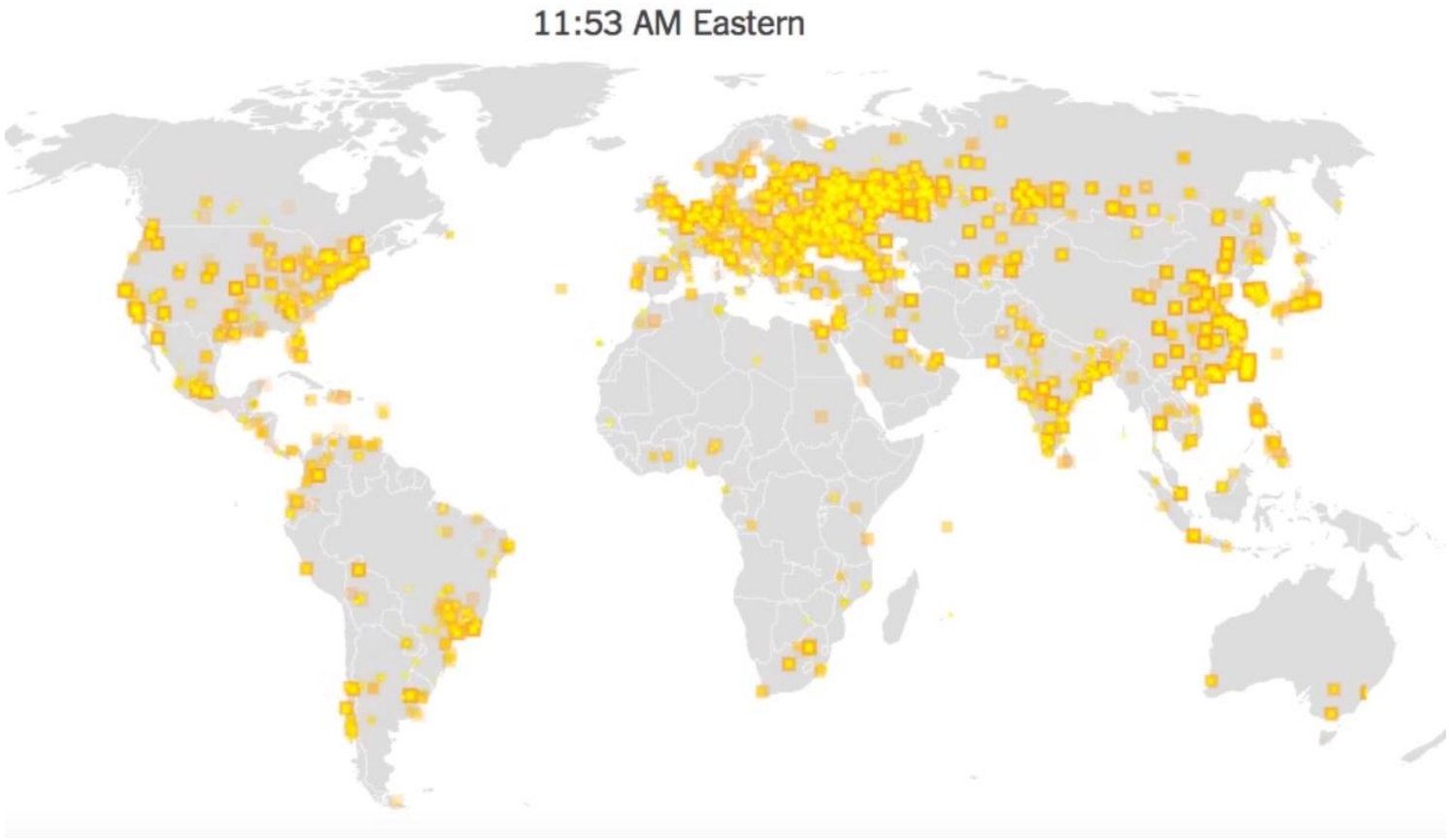
Wormable Vuln (Tragedy) (II)

- Exploit that **propagates** itself **automatically**:
  1. Attempts to connect to **all** IP addresses
  2. Inject a **copy of itself** on every IP address found

- Within **a few minutes**, **all** vulnerable systems reachable from patient zero will be infected

- One of the reasons why NAT & **tight firewall rules on users platforms** are important…

Wannacry (May 2017): Time 30 minutes



Exploit and Injection: Examples

- See APPENDIX

Vuln vs Exploit vs Injection

Vuln vs Exploit vs Injection (I)

- Actual exploitation of a vuln requires **all the three steps**:
  1. Vulnerability discovery
  2. Exploit creation
  3. Exploit injection in vulnerable system

- The exploit must be **developed**
- Its existence is not "automatic"

Vuln vs Exploit vs Injection (II)

- Actual exploitation of a vuln requires **all the three steps**:
  1. Vulnerability discovery
     - Difficult — Often a full-time job
  2. Exploit creation
     - **Very** difficult — Often a full-time job
  3. Exploit injection in vulnerable system
     - Local / Remote
     - User / No User
     - Varying levels of difficulty

Bugs vs Vulns vs Exploitable Vulns

Bugs

Error or flaw that causes the sw
- to produce an **incorrect** result, or
- to behave in **unintended ways**.

Vulnerabilities

- ...with **violation** of some **security** property

Exploitable Vuln.

PoC

- Actual exploitation of a vuln requires **all the three steps**:
  1. Vulnerability discovery
  2. Exploit creation

- Sometimes, **proof of concept** exploits are publicly available
- Can be modified and specialized easily

- search "`exploit db`"

Common Vulnerability and Exposures (CVE)

Common Vulnerabilities and Exposures (CVE) (I)

- **Database** of every **publicly known** vulnerability
- **Official** reference: search "cve mitre"
  - Link to a github repository updated daily
- **Practical** reference: search "NIST search nvd"
  - Webapp
- Very handy: search "cvedetails"
  - Webapp **browsable** by many criteria

CVE (I)

- Each CVE record ("CVE") is composed of:
  - **CVE-ID** CVE-YYYY-*NNNN..NN*
    Unique identifier of the vulnerability
  - Textual description and links
  - **CPE** (Common **Platform** Enumeration)
    Vendor, System, Version in standard format
  - **CWE** (Common **Weakness** Enumeration)
    Type of mistake(s) in standard taxonomy
  - ...

Example (I)

CVE-2025-6558 Detail

Description

Insufficient validation of untrusted input in ANGLE and GPU in Google Chrome prior to 138.0.7204.157 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-20	Improper Input Validation	Chrome

Known Affected Software Configurations Switch to CPE 2.2

Configuration 1 (hide)

cpe:2.3:a:google:chrome:*:*:*:*:*:*:*:*	**Up to (excluding)** **138.0.7204.157**
Hide Matching CPE(s) ▲	
• *cpe:2.3:a:google:chrome:9.0.597.5:*:*:*:*:*:*:**	
• *cpe:2.3:a:google:chrome:9.0.597.7:*:*:*:*:*:*:**	
• *cpe:2.3:a:google:chrome:9.0.597.8:*:*:*:*:*:*:**	
• *cpe:2.3:a:google:chrome:9.0.597.9:*:*:*:*:*:*:**	
• *cpe:2.3:a:google:chrome:9.0.597.40:*:*:*:*:*:*:**	

CVE (II)

- Each entry ("CVE") is composed of:
  - ...
  - **Injection** type
  - **Impact** type
  - **Attack** complexity
  - Each expressed in predefined categories

  - Distilled in a single severity score [0, 10]
    CVSS (Common Vulnerability **Scoring** System)

Example (II)

Base Score: 8.8 HIGH

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N)	Adjacent Network (AV:A)	Local (AV:L)	Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L)	High (AC:H)

Privileges Required (PR)*

None (PR:N)	Low (PR:L)	High (PR:H)

User Interaction (UI)*

None (UI:N)	Required (UI:R)

Impact Metrics

Confidentiality Impact (C)*

None (C:N)	Low (C:L)	High (C:H)

Integrity Impact (I)*

None (I:N)	Low (I:L)	High (I:H)

Availability Impact (A)*

None (A:N)	Low (A:L)	High (A:H)

Categorization **different** from ours

Common Vulnerabilities and Exposures (CVE) (II)

- Standard de facto
- Extremely important as **common** and **reliable** reference

- Mantained by a consortium of organizations (**CNA**: Common Numbering Authority)

1. Submission by a researcher
2. Analysis by a CNA
3. Insertion in CVE

What does it contain, exactly? (I)

- Public database of every publicly known vulnerability

- At any given time there exist vulnerabilities **exploited** in practice that are known **only** to Attackers
  - Criminal organizations
  - Nation-states
- Such "**zero-days**" are obviously **not** listed in the CVE

- More details later

What does it contain, exactly? (II-a)

- Public database of every publicly known vulnerability
- **Only** vulns of "**widely used"** products

What does it contain, exactly? (II-b)

- **Only** vulns of **"widely used"** products...and **not always**

- Sometimes, vulnerabilities of "widely used" products are **not** given any CVE
- Powerful vendors sometimes:
  - Claim it is a feature
  - Quietly modify the system to fix the vulnerability
  - Quietly modify the documentation
  - Quietly modify the default configuration

- Relatively common in cloud services

Common Vulnerability Scoring System (CVSS)

Vulnerability Severity

- Depends on **many** factors:
  - **Impact**
  - Difficulty of **developing exploit**
  - Difficulty of **injection**
    - Local vs Remote (Auth, NoAuth)
    - User intervention vs Automatic
  - When there will be a **patch**
  - What **workarounds** meanwhile
  - ...

Common Vulnerability Scoring System (CVSS) (I)

- Standard for **quantifying** severity by means of a **single number** (**score**)

- **Score** in range 1-10
- Discretized in 5 **categories**

None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Common Vulnerability Scoring System (CVSS) (II)

- Input:
  - 8 categorical features **part of the CVE**
  - ≈ Impact + Injection
- Processing:
  - Translate categories to numbers
  - Arithmetic formulas + IF-THEN-ELSE
- Output:
  - **Score** in range 1-10
  - Discretized in 5 **categories**

Example

CVE-2021-26855 Detail

Base Score: 9.8 CRITICAL

Current Description

Microsoft Exchange Server Remote Code Execution Vulnerability

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N) | Adjacent Network (AV:A) | Local (AV:L) | Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L) | High (AC:H)

Privileges Required (PR)*

None (PR:N) | Low (PR:L) | High (PR:H)

User Interaction (UI)*

None (UI:N) | Required (UI:R)

Scope (S)*

Unchanged (S:U) | Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*

None (C:N) | Low (C:L) | **High (C:H)**

Integrity Impact (I)*

None (I:N) | Low (I:L) | **High (I:H)**

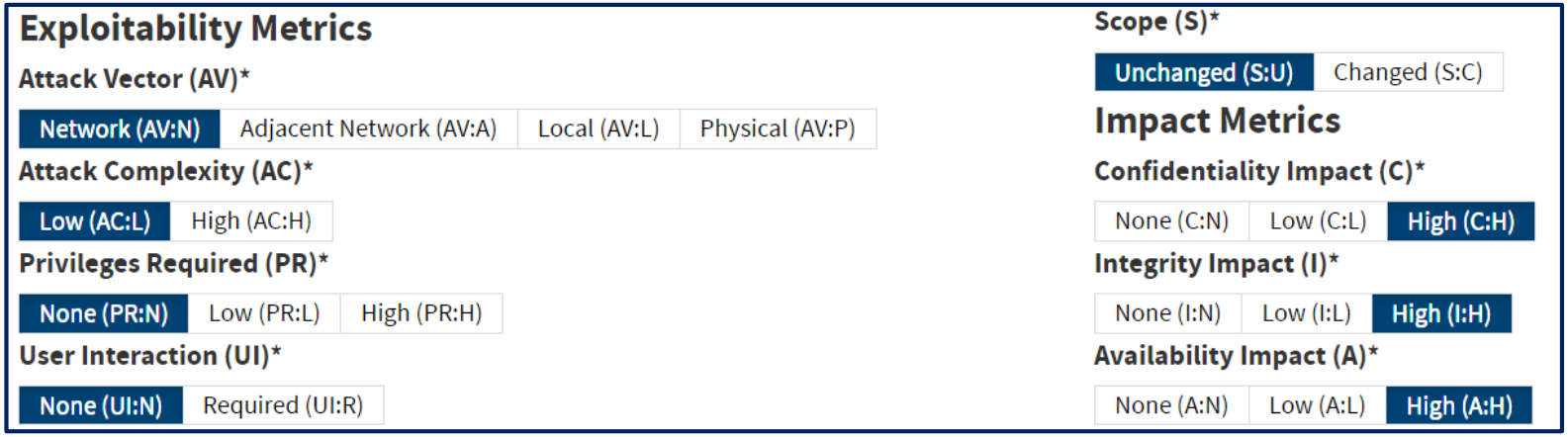
Availability Impact (A)*

None (A:N) | Low (A:L) | **High (A:H)**

Remark

- Input:
  - 8 categorical features part of the CVE
  - ≈ Impact + Injection

- **Our** categories for Impact and Injection were different



Understanding CVSS

CVSS Definition

- Defined by a consortium (and updated several times)
  - Which input categories
  - Which output range
  - How inputs are mixed to obtain the output
  - Criteria that **human operators** should follow for determining the inputs for the CVE being assessed

- Just **one** of the **many** possible ways of distilling **many qualitative properties** in a **single number**

CVSS is NOT a "physical property"

- CVSS is just **one** of the **many** possible ways of distilling **many qualitative properties** in a **single number**

- **"Quantitative"** does **not** necessarily mean "**Objective and Meaningful"**

- Can you **quantify** beauty of a person?
  - Yes, we could invent endless formulas
- Can you do that in a way that is both **objective and meaningful**?
  - Of course not

CVSS <> Risk

- CVSS estimates **severity** of a vulnerability
- CVSS is **not** a measure of the **actual risk** provoked by that vulnerability

- Key examples:
  - Vulnerability V1 low score in system that in my environment is vital
  - Vulnerability V2 high score in system that in my environment is not important at all
  - Vulnerability V3 high score in system that in my environment is almost unreachable

Risk assessment

- CVSS is **not** a measure of the **actual risk** provoked by that vulnerability
- Assessing the actual risk of a given vulnerability requires:
  - **Contextual** (environmental) information
    - Injection in **my** environment?
    - Impact in **my** environment?
  - Severity information **not** used for computing the CVSS
    - Is there a **patch**?
    - Is there a **mitigation**?
    - Are **exploits** easily available?

Vulnerability Severity (REMIND)

- Depends on **many** factors:
  - **Impact**
  - Difficulty of **developing exploit**
  - Difficulty of **injection**
    - Local vs Remote (Auth, NoAuth)
    - User intervention vs Automatic
  - When there will be a **patch**
  - What **workarounds** meanwhile
  - ...

CVSS considers **only** these properties

CVSS Base Score

- Input:
  - 8 categorical features **part of the CVE**
  - ≈ Impact + Injection

- **Immutable**
  - Does not depend on existence of exploit / patch
- **Independent of the context**
  - Does not depend on the specific environment where the vulnerable system is placed

CVSS Full Score

Temporal Score Metrics

Exploit Code Maturity (E)

Not Defined (E:X)	Unproven that exploit exists (E:U)	Proof of concept code (E:P)	Functional exploit exists (E:F)	High (E:H)

Remediation Level (RL)

Not Defined (RL:X)	Official fix (RL:O)	Temporary fix (RL:T)	Workaround (RL:W)	Unavailable (RL:U)

Report Confidence (RC)

Not Defined (RC:X)	Unknown (RC:U)	Reasonable (RC:R)	Confirmed (RC:C)

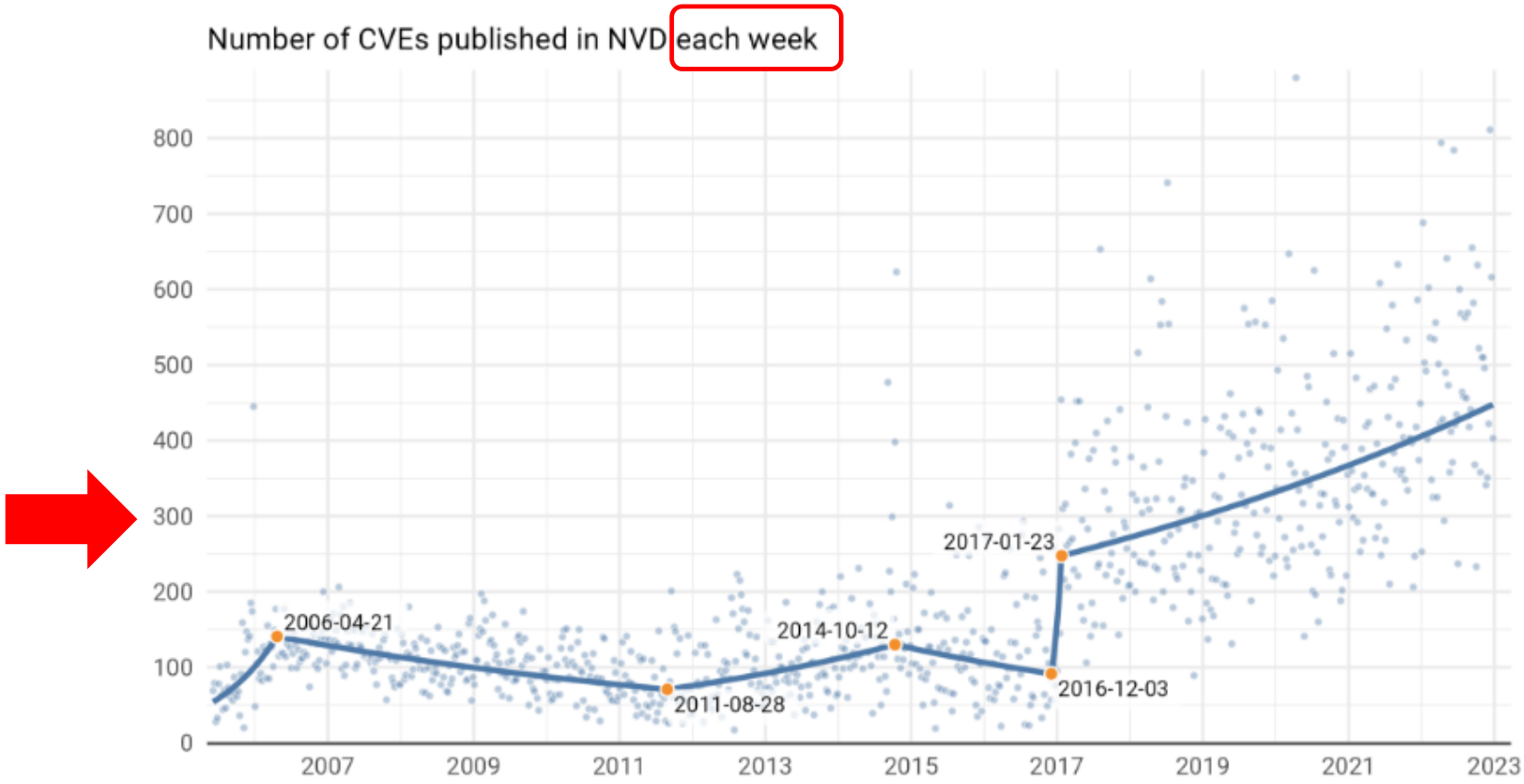
Environmental Score Metrics

Exploitability Metrics **Impact Metrics**

- One can define these **additional** input values and obtain the **full** score CVSS
- Not particularly useful / used

Understanding CVE

New CVEs (REMIND)



Basic Fact #1

- **Very few CVEs are actually exploited**
- Just to have an idea: **≈5%** of all CVEs **(!)**

Basic Fact #2 (I)

- CVSS is **not** a good predictor of which vulnerabilities will be **actually exploited**

- I will focus on vulns with CVSS > some-Threshold
  - "Low" precision:
    You will worry about many vulns **unnecessarily**
  - "Low" recall:
    You will wrongly **neglect** many vulns

Basic Fact #2 (II)

- CVSS is **not** a good predictor of which vulnerabilities will be **actually exploited**
- Predicting which vulnerabilities will be exploited is a **huge open problem**

- Every predictor you can think of turns out to be
  - "Low" precision:
    You worry about many vulns **unnecessarily**
  - "Low" recall:
    You wrongly **neglect** many vulns

KEV Catalog (I)

- CISA maintains a free public catalog of **Known Exploited** Vulnerabilities (**KEV**)
- Approximately 1 every 250 of newly published vulnerabilities (0.4%)
  - ≈180 from March 2023 to September 2024
- You may think of it as:
  - CVE
  - Filtered by **threat intelligence** information

KEV Catalog (II)

- Every predictor you can think of turns out to be
  - "Low" precision:
    You worry about many vulns **unnecessarily**
  - "Low" recall:
    You wrongly **neglect** many vulns

- True even if your predictor is "presence in KEV"

CVE Publication Process (I)

1. Submission by a researcher
2. Analysis by a CNA
3. Insertion in CVE

- Process requiring **careful analysis** by **human operators**

CVE Publication Process (II)

1. Submission by a researcher
2. Analysis by a CNA
3. Insertion in CVE
   a) **CVE-ID** + some textual information
   b) CVSS, CWE, CPE added as soon as available

- The content of a CVE record is **not** published as a **single** event

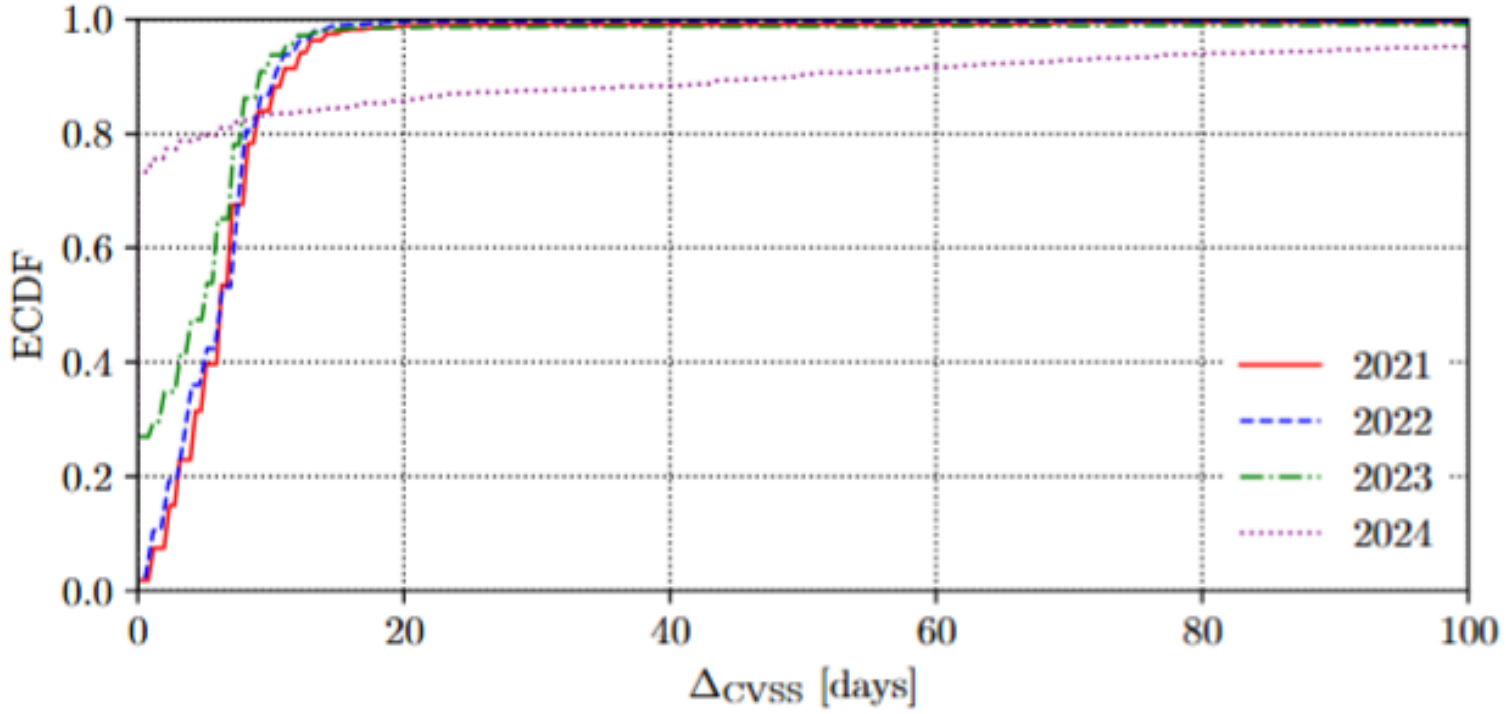
CVE Publication Process (III)

1. Submission by a researcher
2. Insertion in CVE
3. Analysis by a CNA
   a) **CVE-ID** + some textual information
   b) CVSS, CWE, CPE added as soon as available

- The duration of each step may take **many days**

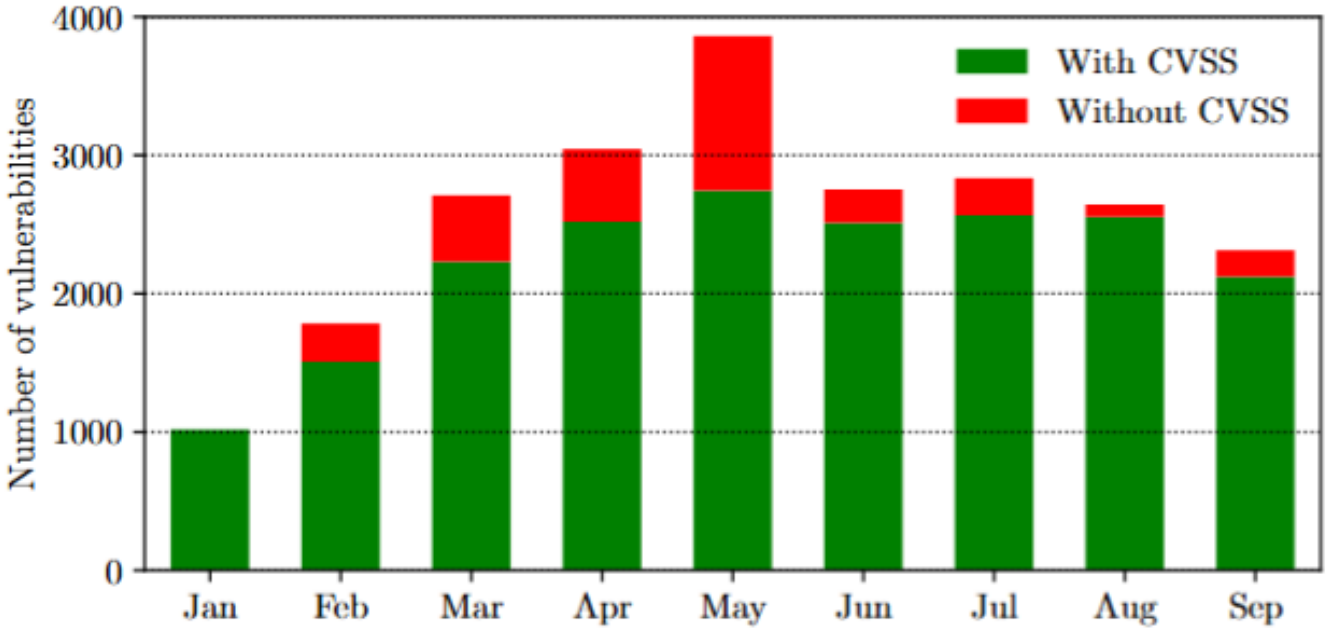
- Significant backlog accumulated in 2024

CVSS assignment delay (I)



- Cumulative distributions
- Number of days since the CVE record was published to obtain CVSS

CVSS assignment delay (II)



- Number of new reserved CVE in 2024
- With / without a CVSS on Sept 30, 2024

Basic Fact #3

- Increasing **evidence of exploitation before**
  - CVE publication
  - CVSS assignment
  - KEV insertion

- 1-H 2025: ≈**1/3 KEV** evidence of exploitation before CVE publication
- 1-H 2025: less than 1/4

Source: VulnCheck

CVE: Brutal (but meaningful) summary

- **Important information** but
- Mixed with a **lot of noise** that we do **not** know how to filter out
- Often available **too late**
- Problems increasingly more acute