UNIVERSITÀ DEGLI STUDI
DI TRENTO

Network Security

AA 2015/2016

Vulnerability measurement

Dr. Luca Allodi

Why to grade vulnerabilities?

- Central question:
  - → How severe are the security problems affecting my software configuration?
- Not all vulnerabilities are the same
  - XSS vs BoF vs SQLi vs Privilege escalation vs …
  - Vulnerability counting can NOT be a measure of severity
  - → What is the threat level of your systems?
  - → Clients and users should be informed too
    - → Not all users are "security experts"
    - → "IT knowledge" can be assumed
  - → How to measure communicate a security issue?

Best practice

- Listen to the U.S. Government….
  - US Cyber Security Order (Press release Feb’2013)
    - “NIST will work collaboratively with critical infrastructure stakeholders to develop the framework relying on **existing international standards**, practices, and procedures that have proven to be effective”
  - U.S. NIST SCAP Protocol v1.2(Draft Jan 2012)
    - “Organizations should use **CVSS base scores** to assist in prioritizing the remediation of known security-related software flaws based on the relative severity of the flaws.”
  - PCI-DSS v2 (June 2012)
    - “Risk rankings should be based on industry best practices. For example, criteria for ranking ―High‖risk vulnerabilities may include a **CVSS base score** of 4.0 or above”
  - U.S. Government Configuration Baseline (USGCB)
    - Supported by the industry→ Rapid7, Telos, VmWare, Symantec, Qualys, Retina etc. etc.

The Common Vulnerability Scoring System

- CVSS is an open framework for communicating the characteristics and severity of software vulnerabilities.
- Goal is to have a **shared system of metrics** to **analyze and measure vulnerabilities**
  - Different users score the same vuln in the same way → severity assessment
  - Different people "read" the same vuln and understand the same thing → severity communication

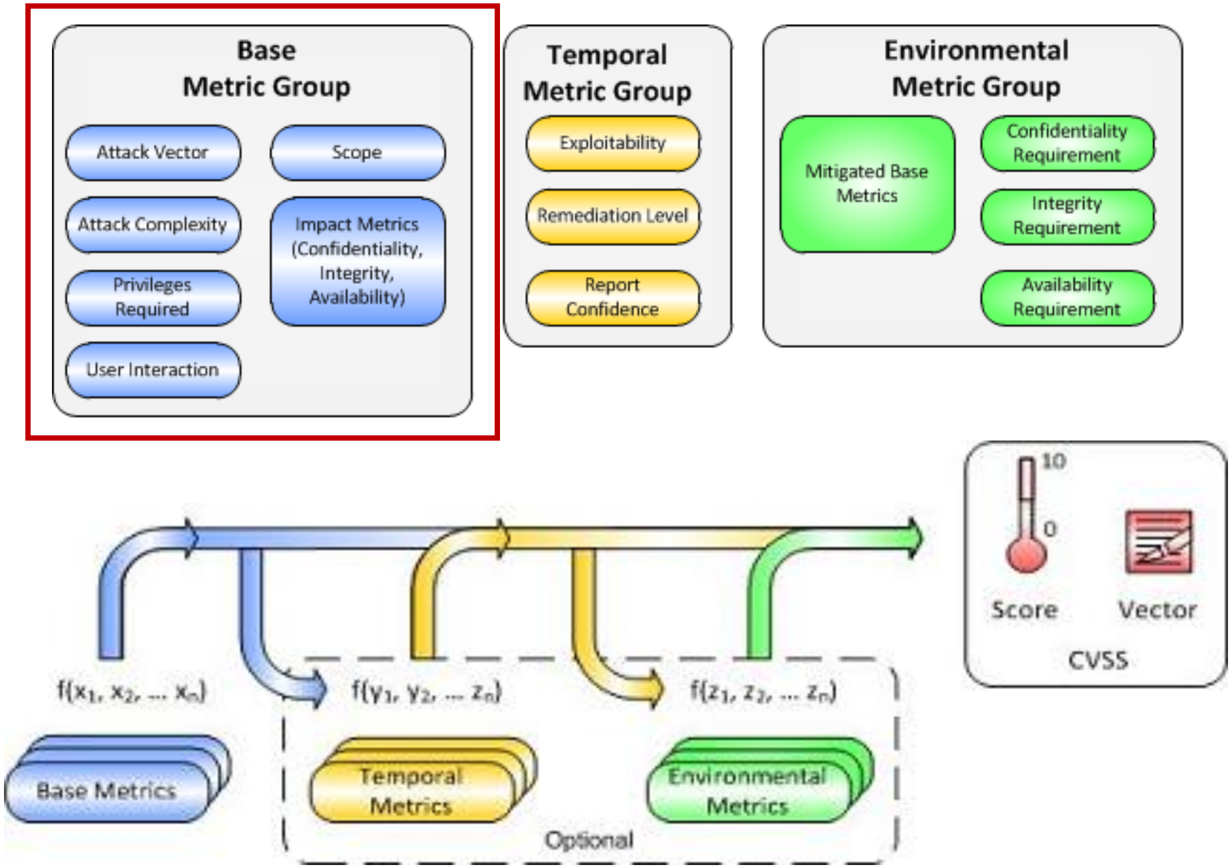
CVSS v(x) walkthrough

- CVSS v(1) introduced back in 2004 by First.org
  - Reception was good but implementation was confusing
  - Not peer-reviewed
- CVSS v(2) workings started in 2005, released in 2007
  - Peer-reviewed, industry feedback
  - Became *standard-de-facto* vulnerability scoring system in the industry
- CVSS v(3) workings started in 2012, released in 2015
  - Builds on top of v2
  - Changes the “scoring philosophy”
  - Further step toward a precise scoring system

CVSS v3

http://www.first.org/cvss/v3/development

- CVSS is based on three metric groups



CVSS Base metric overview

- Exploitability metrics
  - Attack Vector
  - Attack Complexity
  - User Interaction
  - Privileges Required
- Scope metric
- Impact metrics
  - Confidentiality
  - Integrity
  - Availability

Measured over the vulnerable component

Auth. Authority of Vulnerable Component =
Auth. Authority of Impacted Component?

Measured over the impacted component

Expl. Metrics: Attack Vector

- This metric reflects the context in which the vulnerability exploitation occurs.
- The more remote an attacker (or the attack) can be from the target, the greater the vulnerability score.
- Possible values:
  1. **Network**: exploitation is bound to the network stack
  2. **Adjacent Network**: attacker needs to be in same subnet
  3. **Local**: attack is not bound to network stack, but rather to I/O on system. In some cases, the attacker may be logged in locally in order to exploit the vulnerability, otherwise, she may rely on User Interaction to execute a malicious file.
  4. **Physical**: attacker must be physically operating over the vulnerable component

Expl. Metrics: Attack Complexity

- This metric describes the conditions beyond the attacker’s control that must exist in order to exploit the vulnerability.
- Possible values:
  1. **High**: A successful attack depends on conditions outside the attacker's control. That is, **a successful attack** cannot be accomplished , but **requires the attacker to invest in some measurable amount of effort in preparation or execution against the vulnerable component** before a successful attack can be expected.
  2. **Low:** Specialized access conditions or extenuating circumstances do not exist. An attacker can expect repeatable exploit success against a vulnerable target

Examples for Attack Complexity: High

- For example, a successful attack may depend on an attacker overcoming any of the following conditions:
  1. The attacker must conduct **target-specific reconnaissance**. For example, on target configuration settings, sequence numbers, shared secrets, etc.
  2. The attacker must **prepare the target environment** to improve exploit reliability. For example, repeated exploitation to win a race condition, or overcoming advanced exploit mitigation techniques.
  3. The attacker **injects herself into the logical network path** between the target and the resource requested by the victim in order to read and/or modify network communications (e.g. man in the middle attack).

Expl. Metrics: Privileges Required

- This metric describes the level of privileges an attacker must possess before successfully exploiting the vulnerability.
- Possible values:
  1. High: The attacker is authorized with (i.e. requires) privileges that provide significant (e.g. administrative) control over the vulnerable component that could affect component-wide settings and files.
  2. Low: The attacker is authorized with (i.e. requires) privileges that provide basic user capabilities that could normally affect only settings and files owned by a user. Alternatively, an attacker with Low privileges may have the ability to cause an impact only to non-sensitive resources.
  3. None: The attacker is unauthorized prior to attack, and therefore does not require any access to settings or files to carry out an attack.

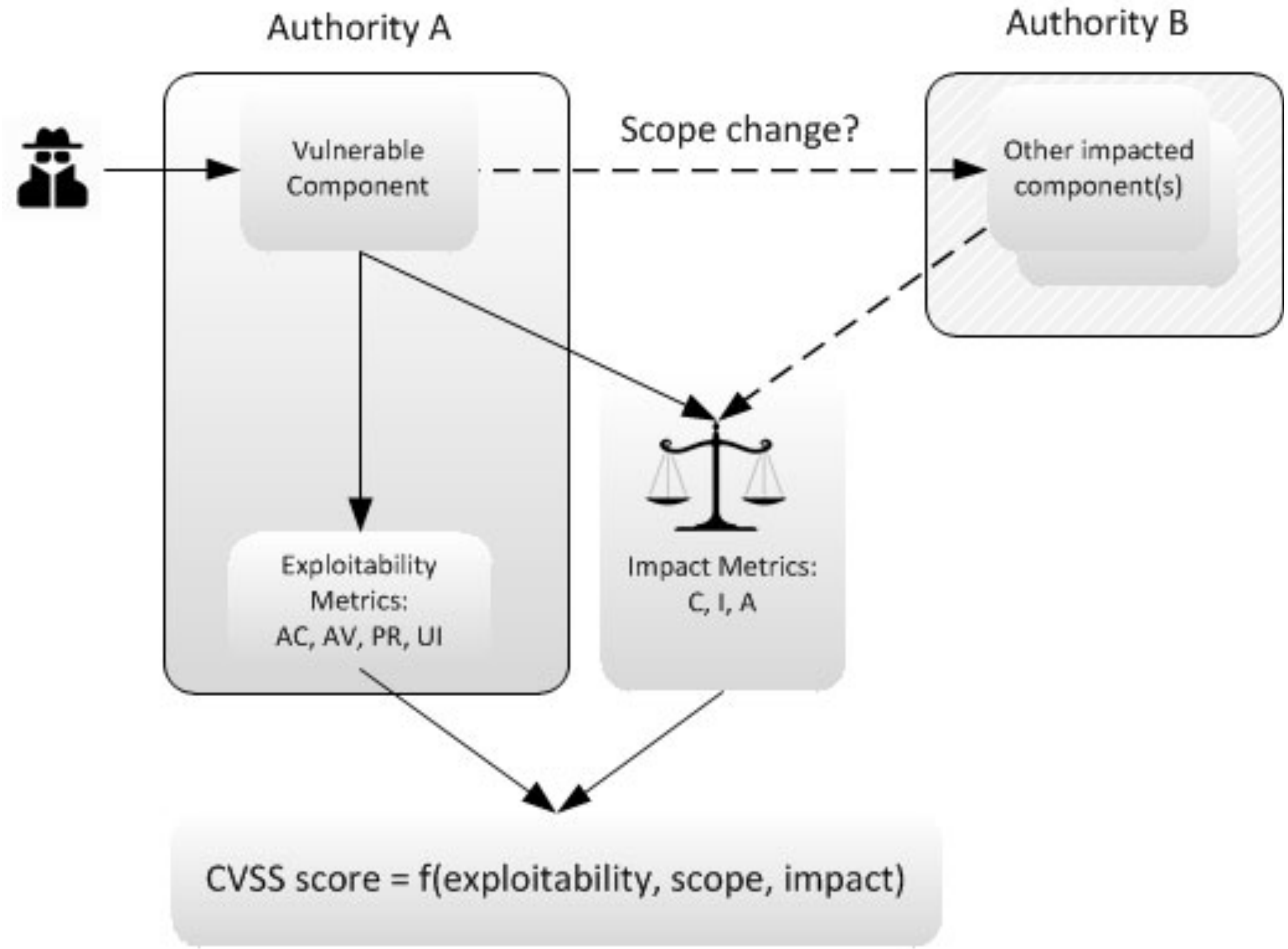
Expl. Metrics: User Interaction

- This metric captures the requirement for a user, other than the attacker, to participate in the successful compromise the vulnerable component.
- This metric determines whether the vulnerability can be exploited solely at the will of the attacker, or whether a separate user (or user-initiated process) must participate in some manner.
- Possible values:
  1. Required: Successful exploitation of this vulnerability requires a user to take some action before the vulnerability can be exploited. For example, a successful exploit may only be possible during the installation of an application by a system administrator.
  2. None: The vulnerable system can be exploited without any interaction from any user.

Scope (1)

- Scope refers to the collection of privileges defined by a computing authority (e.g. an application, an operating system, or a sandbox environment) when granting access to computing resources (e.g. files, CPU, memory, etc). These privileges are assigned based on some method of identification and authorization.
- When the vulnerability of a software component governed by one authorization scope is able to affect resources governed by another authorization scope, a Scope change has occurred.

Scope (2)


Authority A
Authority B
Vulnerable Component
Scope change?
Other impacted component(s)
Exploitability Metrics: AC, AV, PR, UI
Impact Metrics: C, I, A
CVSS score = f(exploitability, scope, impact)

Scope (3)

- Possible values:
  - Unchanged: An exploited vulnerability can only affect resources managed by the same authority. In this case the vulnerable component and the impacted component are the same.
  - Changed: An exploited vulnerability can affect resources beyond the authorization privileges intended by the vulnerable component. In this case the vulnerable component and the impacted component are different.

Impact metrics

- Measures the losses on
  - Confidentiality, → impact on confidentiality of **data**
    - *property that information is not made available or disclosed to unauthorized individuals, entites, or processes*
  - Integrity, → impact on integrity of **data**
    - *the "property of accuracy and completeness" of information*
  - Availability → impact on availability of **the component**
    - is the "property of being accessible and usable upon demand by an unauthorized entity"
- Each metric measures the losses **suffered by the impacted component**
- Possible values:
  1. High → total loss
  2. Low → partial loss
  3. None → no loss

Scoring Guide/Philosophy

- Access Vector → is the attack bound to the network stack?
- Attack Complexity → can the attacker control all factors relevant to the exploitation?
- Privileges Required → does the attacker need be authenticated?
- User Interaction → does the victim user need to interact with the attack?
- Scope → is the authorisation authority under which the vulnerable component is the same as the impacted component?
- Impact
  - Confidentiality, Integrity → Data
  - Availability → Service
- **Scoring rule: When more than one assessment is possible, go with the more severe one**
  - **e.g. exploitation can happen both though local I/O and on network stack → go with network**

Scoring Exercise (1)

- MS Word Denial-of-Service attack (CVE-2013-6801)
  - Microsoft Word 2003 SP2 and SP3 on Windows XP SP3 allows remote attackers to cause a denial of service (CPU consumption) via a malformed .doc file containing an embedded image, as demonstrated by word2003forkbomb.doc, related to a "fork bomb" issue.

Access Vector	
Access Complexity	
Privileges Required	
User Interaction	
Scope	
Confidentiality	
Integrity	
Availability	

Scoring Exercise (2)

- CISCO host crash (CVE-2011-0355)
  - Cisco Nexus 1000V Virtual Ethernet Module (VEM) 4.0(4) SV1(1) through SV1(3b), as used in VMware ESX 4.0 and 4.1 and ESXi 4.0 and 4.1, does not properly handle dropped packets, which allows guest OS users to cause a denial of service (ESX or ESXi host OS crash) by sending an 802.1Q tagged packet over an access vEthernet port, aka Cisco Bug ID CSCtj17451.

Access Vector	
Access Complexity	
Privileges Required	
User Interaction	
Scope	
Confidentiality	
Integrity	
Availability	

Scoring Exercise (3)

- CVE-2009-0927
  - Stack-based buffer overflow in Adobe Reader and Adobe Acrobat 9 before 9.1, 8 before 8.1.3 , and 7 before 7.1.1 allows remote attackers to execute arbitrary code via a crafted argument to the getIcon method of a Collab object, a different vulnerability than CVE-2009-0658.

Access Vector	
Access Complexity	
Privileges Required	
User Interaction	
Scope	
Confidentiality	
Integrity	
Availability	

Scoring Exercise (4)

- Libvirt USB handling (CVE-2012-2693)
  - libvirt, possibly before 0.9.12, does not properly assign USB devices to virtual machines when multiple devices have the same vendor and product ID, which might cause the wrong device to be associated with a guest and might allow local users to access unintended USB devices.

Access Vector	
Access Complexity	
Privileges Required	
User Interaction	
Scope	
Confidentiality	
Integrity	
Availability	

Next time - Scoring exercise

- **Bring your laptop → exercise on google classroom**
  - The scoring exercise will be with different vulns from those @ Sec Engineering, + 1 metric (Scope)
  - People that already did CVSS assessments will be considered as “experts”
  - This is NOT graded → not part of your final grade
- Exam may require to score a vulnerability using CVSS v3 and justify decision
- We will have 4 groups: A,B,C,D
  - Each student will be assigned to a group randomly
  - Each group differs only for the arrangement of the vuln description
  - All have identical vulnerabilities to score
- 1 hour for the exercise, remaining time to discuss scores

Scoring exercise

- **We have 4 groups: A,B,C,D**
  - **Each group differs only for the arrangement of the vuln description**
  - **All have identical vulnerabilities to score**
- You get a spreadsheet with 16 vulnerability descriptions
- Estimated score: 1-10 with 10 very bad, 1 not so bad
- Confident?
  - Yes=the vuln is clear to me
  - No= I'm not sure
- DK → Domain Knowledge:
  - 0: I have bearly heard of that software, don't know it
  - 1: I have some knowledge on what the software does
- Comments
  - Leave comments on the vulnerability.
    - Was the provided information sufficient?
      - If not, what additional info you deem necessary?
    - Is there something you did not understand?