Software Security Course

Lecture #01 Supplement: Rating security issues with CVSS 3.0

Dimitrios A. Glynos
{ daglyn at unipi.gr }

Department of Informatics
University of Piraeus

Part I

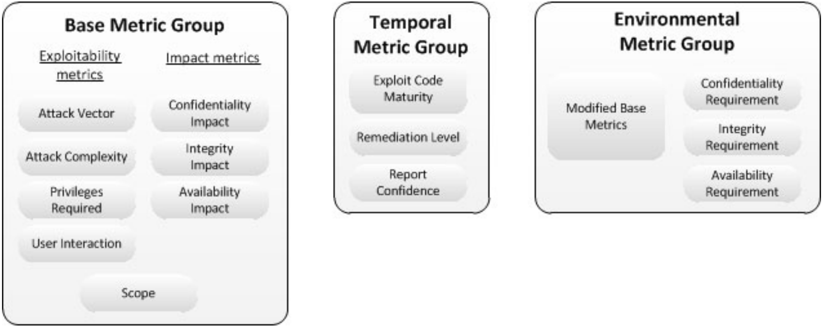
Vulnerability Scoring

Vulnerability scoring

- CVSS[1] - Common Vulnerability Scoring System
- A standard for software vulnerability scoring.
- Latest version is 4.0
- Allows to compare vulnerabilities by criticality.
- Allows to prioritize the fixing of critical vulnerabilities.
- We will explore version 3.0, as it is the one most widely adopted.

[1] https://www.first.org/cvss/

CVSS v3.0 factors



Base Metric Group

Exploitability metrics

Attack Vector

Attack Complexity

Privileges Required

User Interaction

Impact metrics

Confidentiality Impact

Integrity Impact

Availability Impact

Scope

Temporal Metric Group

Exploit Code Maturity

Remediation Level

Report Confidence

Environmental Metric Group

Modified Base Metrics

Confidentiality Requirement

Integrity Requirement

Availability Requirement

- Base, Temporal and Environmental metrics
- Temporal is influenced by Base
- Environmental is influenced by Temporal
- Temporal and Environmental metrics are optional

Base metrics

- **Attack Vector**: Network / Adjacent / Local / Physical
- **Attack Complexity**: Low / High
- **Privileges Required**: None / Low / High
- **User Interaction**: None / Required
- **Scope**: Unchanged / Changed
- **Confidentiality**: None / Low / High
- **Integrity**: None / Low / High
- **Availability**: None / Low / High

Temporal metrics

- **Exploit Code Maturity**: Not Defined / Unproven / PoC / Functional / High
- **Remediation Level**: Not Defined / Official Fix / Temporary Fix / Workaround / Unavailable
- **Report Confidence**: Not Defined / Unknown / Reasonable / Confirmed

Environmental metrics

- **Confidentiality Requirement**: Not Defined / Low / Medium / High
- **Integrity Requirement**: Not Defined / Low / Medium / High
- **Confidentiality Requirement**: Not Defined / Low / Medium / High
- **Modified Attack Vector**: Not Defined / Network / Adjacent / Local / Physical
- **Modified Attack Complexity**: Not Defined / Low / High
- **Modified Privileges Required**: Not Defined / None / Low / High
- **Modified User Interaction**: Not Defined / None / Required
- **Modified Scope**: Not Defined / Unchanged / Changed
- **Modified Confidentiality**: Not Defined / None / Low / High
- **Modified Integrity**: Not Defined / None / Low / High
- **Modified Availability**: Not Defined / None / Low / High

Using the score

- Each metric contributes to the metric group score with a certain weight (depending on the option selected).
- Stakeholders may further classify (e.g. 0 ”informational”, 1-3 ”low”, 4-6 ”medium”, 7-10 ”high”) the following scores:
    - Base Score
    - Temporal Score
    - Environmental Score
- A default action can be selected for scores belonging to a particular class
    - *We can proceed with a release if all issues are of **informational** or **low** class*.
- Stakeholders examine the severity of issues in order to take decisions
    - release without a patch (i.e. make it ”an accepted risk”)
    - schedule the patch for the next planned release
    - release an urgent security update

Accepted Risk

- *”Accepting risk occurs when the cost of managing a certain type of risk is accepted, because the risk involved is not adequate enough to warrant the added cost it will take to avoid that risk.”*
  (source: `investopedia.com`)
- *Accepted risks* need to be tracked and documented.

Part II

Issue Tracking

Tracking Vulnerabilities

- Tracking security vulnerabilities within the lifetime of a project is essential as:
  - It allows for in-depth documentation of discovered security vulnerabilities which is crucial for developers.
  - It allows for prioritizing vulnerability fixing tasks.
  - It enables the management of risk throughout the lifecycle of a project.
  - It builds a knowledgebase on issues affecting the project.

The Issue Tracker

- Vulnerabilities are typically tracked through an Issue Tracker.
- This can be a spreadsheet or an online bug tracking system, where vulnerabilities and their properties are recorded.
- Each issue gets a single record, that describes the issue and its current state in the project.
- An issue's vulnerability score may change over time, due to new security measures being introduced, new research findings, due to temporal factors or environmental factors.