ANNALS of Faculty Engineering Hunedoara
– International Journal of Engineering
Tome XIV [2016] – Fascicule 3 [August]
ISSN: 1584-2665 [print; online]
ISSN: 1584-2673 [CD-Rom; online]
a free-accessmultidisciplinarypublication
of theFaculty of Engineering Hunedoara

[1.]Petar CISAR, [2.]Zoltan RAJNAI,
[3.]Sanja MARAVIC CISAR, [4.]Robert PINTER

SCORING SYSTEM AS A METHOD OF IMPROVING IT VULNERABILITY STATUS

[1.]Department of Information technology, Academy of Criminalistic & Police Studies, Zemun, SERBIA
[2.]Obuda University, Donat Banki Faculty, HUNGARY
[3-4.]Subotica Tech, Subotica, SERBIA

ABSTRACT: The Common Vulnerability Scoring System (CVSS) represents an open structure for linking the characteristics and effects of IT vulnerabilities. The National Vulnerability Database (NVD) formulated particular scores for known vulnerabilities. Government institutions can utilize the Federal Information Processing Standards (FIPS) 199 security classifications with the NVD CVSS scores to acquire impact scores that are customized to concrete environment. CVSS is comprised of three components: base, temporal and environmental. Every component generates a number ranging from 0 to 10 and a textual form that defines the parameters used to determine the score (called vector). The base group describes the internal characteristics of a vulnerability. The temporal component refers to the attributes of a vulnerability that change after some time. The environmental component speaks to the attributes of a vulnerability that are remarkable to any client's environment. CVSS empowers IT experts, security and application vendors and scientists to all advantage by accepting this common approach of scoring IT vulnerabilities.

Keywords: vulnerability, scoring system, metrics, vectors

1. INTRODUCTION

This chapter describes the main approach to the Common Vulnerability Scoring System (CVSS) and is based on the NIST Interagency Report 7435 [1].

Different organizations from the sphere of security (vendors, coordinators, researchers, users) have different roles, motivations, priorities, resources etc. Nowadays, IT experts must recognize and evaluate vulnerabilities crosswise over numerous specific hardware and software configurations. They have to regulate these vulnerabilities and remediate those that represent the most serious danger. The key problem is to generate appropriate actionable information in a situation of enormous vulnerability data. The CVSS is a vendor-independent, industry standard that evaluates vulnerability severity and helps determine urgency and priority of reaction. It tackles the issue of various, contradictory scoring frameworks and is usable and understandable by anybody. CVSS is an open structure that addresses this issue. It offers several advantages:

- Standardization of scores: At the point when an organization standardizes vulnerability scores across all of its software and hardware platforms, it can influence a single vulnerability management strategy. This strategy may be like a service level agreement (SLA) that states how rapidly a specific vulnerability must be accepted and remediated.
- Open framework: With CVSS, anybody can see the individual characteristics used to infer a score.
- Risk priority: At the point when the environmental score is calculated, the vulnerability tends to be relevant. That is, vulnerability scores are now illustrative of the real risk to a firm. Clients know how imperative a given vulnerability is in relation to different vulnerabilities.

It is important to emphasize that CVSS is not a threat scoring system, a vulnerability database (for example, NVD - the U.S. government collection of standards based vulnerability management data) or a real-time attack scoring system.

CVSS is composed of three metric groups: base, temporal and environmental, each comprising of a set of metrics, as shown in Figure 1.

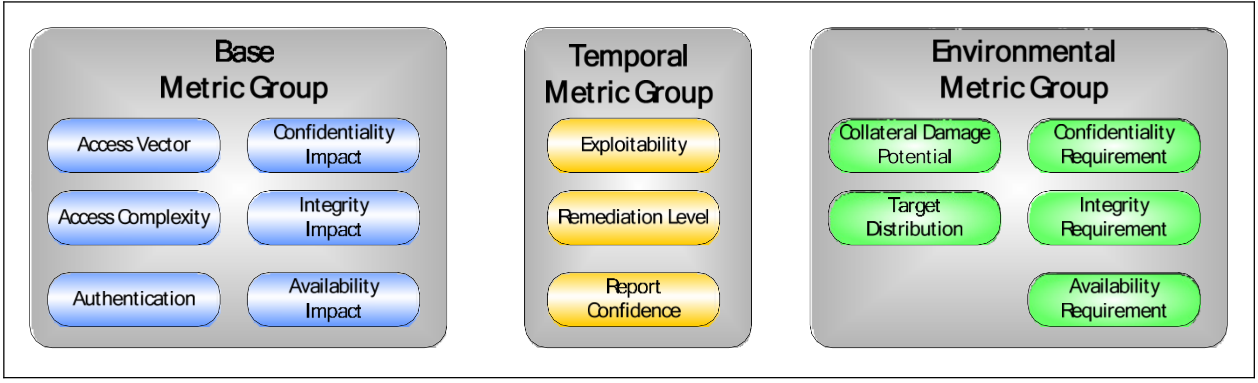


Figure 1. CVSS Metric Groups (source: NIST Interagency Report 7435)

The main differences between the groups are as follows:

- Base - fundamental characteristics of a vulnerability that are constant over time and user environments.
- Temporal - the characteristics of a vulnerability that change over time but not bet user environments.
- Environmental - the characteristics of a vulnerability that are relevant and unique to a particular user's environment.

1.1. Other vulnerability scoring systems

There are a number of other vulnerability scoring systems managed by commercial and non-commercial organizations. They each have their merits, but they differ in what they measure. For example, the coordinator CERT/CC scoring produces a numeric value between 0 and 180 that assigns an approximate severity to the vulnerability. This number considers several factors, including [2]:

- » F1: Is the information about the vulnerability widely available or known?
- » F2: Is the vulnerability being exploited in the incidents reported?
- » F3: Is the Internet infrastructure at risk because of this vulnerability?
- » F4: How many systems on the Internet are at risk from this vulnerability?
- » F5: What is the impact of exploiting the vulnerability?
- » F6: How easy is to exploit the vulnerability?
- » F7: What are the preconditions required to exploit the vulnerability?

The formula which is used in calculations: 3*(F1 + F2 + F3) * (F4 * F5 * F6 * F7) / 20^4.

The SANS vulnerability analysis scale considers whether the weakness is found in default configurations or client or server systems [3].

Vendor Microsoft's proprietary scoring system uses four rating categories [4].

Table 1. Microsoft's Vulnerability Rating

Rating	Definition
Critical	A vulnerability whose exploitation could allow the propagation of an Internet worm without user action.
Important	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity or availability of users data or of the integrity or availability of processing resources.
Moderate	Exploitability is mitigated to a significant degree by factors such as default configuration, auditing or difficulty of exploitation.
Low	A vulnerability whose exploitation is extremely difficult or whose impact is minimal.

Table 2. Secunia's Vulnerability Rating

Rating	Definition
Extremely critical	Typically used for remotely exploitable vulnerabilities, which can lead to system compromise. Successful exploitation does not normally require any interaction and exploits are in the wild.
Highly critical	As above, no known exploits
Moderately critical	As above, but DoS only or requiring user interaction
Less critical	XSS, privilege escalation, sensitive data exposure
Not critical	Very limited privilege escalation, locally exploitable DoS, non – sensitive data exposure

Researcher scoring: Secunia (https://www.first.org/cvss/cvss_basic-2.0.pdf)

1.2. The Working Principle of CVSS

When values are allotted to the base metrics, the calculation is performed by the base equation resulting in a score ranging from 0 to 10, and thus a vector is created (Figure 2). The vector is a text string form containing the values assigned to each metric, it therefore ensures the framework's "open" nature. It is used to communicate the form of determination of the score for each vulnerability, the aim being for anyone to see they how calculate the score. Therefore, the vulnerability score should be shown together with the vector.

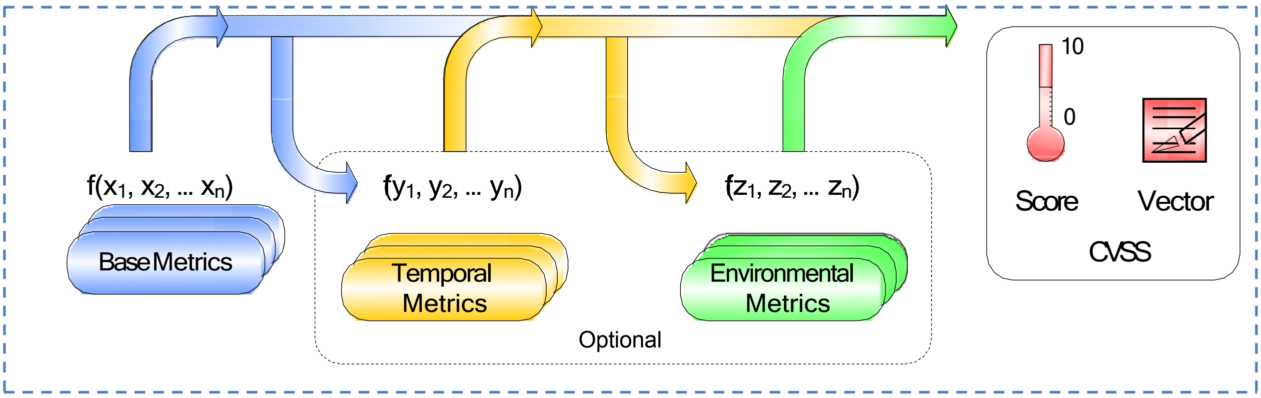


Figure 2. CVSS Metrics and Equations (source: NIST Interagency Report 7435)

Conversely, one may refine the base score by assigning values to the temporal and environmental metrics. This is important in order to provide further context for vulnerability by giving a more exact description of the existing risk by the vulnerability to a user's environment. Depending on purpose, it could be sufficient to have the base score and vector.

If one needs a temporal score, the temporal metrics will be combined with the base score by the temporal equation in order to create a temporal score ranging from 0 to 10. In a similar vein, if one needs an environmental score, the environmental metrics will be combined with the temporal score by the environmental equation so as to create an environmental score ranging from 0 to 10.

2. CVSS METRICS AND METRIC GROUPS

This section defines the metrics that comprise the CVSS version 2 and is based on [5]. These metrics are organized into three groups: base, temporal and environmental metrics.

2.1. Base Metrics

Those characteristics of the vulnerabilities that are constant with time and across user environments are captured by the base metric group. The Access Vector, Access Complexity, and Authentication metrics capture the form of access of the vulnerability and if some extra conditions are needed for its exploitation. The three impact metrics measure how vulnerability, if exploited, will directly affect an IT asset, where the impacts are independently defined as the degree of loss of confidentiality, integrity, and availability. For example, one of the results of a vulnerability could be a partial loss of integrity and availability, but no loss of confidentiality.

Access Vector (AV)

This metric reflects how the vulnerability is exploited. It measures whether vulnerability is exploitable locally or remotely.

» Local: The vulnerability is only exploitable locally

» Remote: The vulnerability is exploitable remotely

The more remote an attacker is to attacking a host, the greater the vulnerability score.

Access Complexity (AC)

This metric measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. The lower the required complexity, the higher the vulnerability score.

» High - specialized access conditions exist for

□ specific windows of time (a race condition)

□ specific circumstances (non-default configurations)

□ victim interaction (tainted e-mail attachment)

» Low - specialized access conditions or extenuating circumstances do not exist

□ always exploitable

Authentication (AU)

This metric measures whether or not an attacker needs to be authenticated to the target system in order to exploit the vulnerability. The fewer the authentication instances that are required, the higher the vulnerability score.

- Required: Authentication is required to access and exploit the vulnerability
- Not Required: Authentication is not required to access or exploit the vulnerability

Confidentiality Impact (C)

This metric measures the impact on confidentiality of a successful exploit of the vulnerability on the target system. The increased confidentiality impact increases the vulnerability score.

- None: No impact on confidentiality
- Partial: There is considerable informational disclosure
- Complete: A total compromise of critical system information

Integrity Impact (I)

This metric measures the impact on integrity of a successful exploit of the vulnerability on the target system. The increased integrity impact increases the vulnerability score.

- None: No impact on integrity
- Partial: Considerable breach in integrity
- Complete: A total compromise of system integrity

Availability Impact (A)

This metric measures the impact on Availability of a successful exploit of the vulnerability on the target system. The increased availability impact increases the vulnerability score.

- None: No impact on availability
- Partial: Considerable lag in or interruptions in resource availability
- Complete: Total shutdown of the affected resource

2.2.Temporal Metrics

These metrics describe the time dependent qualities of vulnerability (the threat posed by vulnerability may change over time): exploitability, remediation status and report confidence. Since temporal metrics are optional, they each include a metric value that has no effect on the score. This quality is utilized when the user feels the specific metric does not make a difference and wishes to bypass it.

Exploitability (E)

This metric measures how complex the process is to exploit the vulnerability in the target system once it has been accessed. The more easily vulnerability can be exploited, the higher the vulnerability score.

- Unproven: No exploit code is yet available
- Proof of Concept: Proof of concept exploit code is available
- Functional: Functional exploit code is available
- High: Exploitable by functional mobile autonomous code or no exploit required (manual trigger)

Remediation Level (RL)

This metric measures the level of solution available. The less official and permanent a fix, the higher the vulnerability score is.

- Official Fix: Complete vendor solution available
- Temporary Fix: There is an official temporary fix available
- Workaround: There is an unofficial non-vendor solution available
- Unavailable: There is either no solution available or it is impossible to apply

Report Confidence (RC)

This metric measures the degree of confidence in the existence of the vulnerability and the credibility of its report. The urgency of vulnerability is higher when vulnerability is known to exist with certainty. The more vulnerability is validated by the vendor or other reputable sources, the higher the score.

- Unconfirmed: A single unconfirmed source or possibly several conflicting reports
- Uncorroborated: Multiple non-official sources; possibly including independent security companies or research organizations
- Confirmed: The vendor has reported/confirmed a problem with its own product

2.3.Environmental Metrics

This metric is related to implementation and environment-specific qualities of vulnerability. Since environmental metrics are optional, they each include a metric value that has no effect on the score. This quality is utilized when the user feels the specific metric does not make a difference and wishes to bypass it.

Collateral Damage Potential (CDP)

This metric measures the potential for a loss in physical equipment, property damage or loss of life or limb. Naturally, the greater the damage potential, the higher the vulnerability score.

- None: There is no potential for property damage.
- Low: A successful exploit of this vulnerability may result in light property damage or loss
- Medium: A successful exploit of this vulnerability may result in significant property damage or loss
- High: A successful exploit of this vulnerability may result in catastrophic property damage and loss

Target Distribution (TD)

This metric measures the relative size of the field of target systems susceptible to the vulnerability. The greater the proportion of vulnerable systems, the higher the score.

- None: There are no target systems, or targets are so highly specialized that they only exist in a laboratory setting (0%)
- Low: Targets exist inside the environment, but on a small scale (1% - 15%)
- Medium: Targets exist inside the environment, but on a medium scale (16% - 49%)
- High: Targets exist inside the environment on a considerable scale (50% - 100%)

2.4.Base, Temporal, Environmental Vectors

Each metric in the vector is made up of the abbreviated metric name, followed by a ":" (colon), then the abbreviated metric value. These metrics are listed by the vector in a predetermined order, using the "/" (slash) character for metrics separation. If one does not wish to use a temporal or environmental metric, the value of "ND" (not defined) is assigned. Table 3 below presents the base, temporal, and environmental vectors.

Table 3. Base, Temporal and Environmental Vectors

Metric Group	Vector
Base	AV:[L,A,N]/AC:[H,M,L]/Au:[M,S,N]/C:[N,P,C]/I:[N,P,C]/A:[N,P,C]
Temporal	E:[U,POC,F,H,ND]/RL:[OF,TF,W,U,ND]/RC:[UC,UR,C,ND]
Environmental	CDP:[N,L,LM,MH,H,ND]/TD:[N,L,M,H,ND]/CR:[L,M,H,ND]/IR:[L,M,H,ND]/AR:[L,M,H,ND]

For example, a vulnerability with base metric values of "Access Vector: Low, Access Complexity: Medium, Authentication: None, Confidentiality Impact: None, Integrity Impact: Partial, Availability Impact: Complete" would feature the following base vector: "AV:L/AC:M/Au:N/C:N/I:P/A:C."

3. SCORING

Scoring is the process of combining metric values. It defines the equations used for base, temporal, and environmental score generation.

- The base score is the "foundation" - modified by temporal and environmental metrics
- The base and temporal scores are computed by vendors and coordinators with the intent of being published
- The environmental score is optionally computed by the end-user /organization

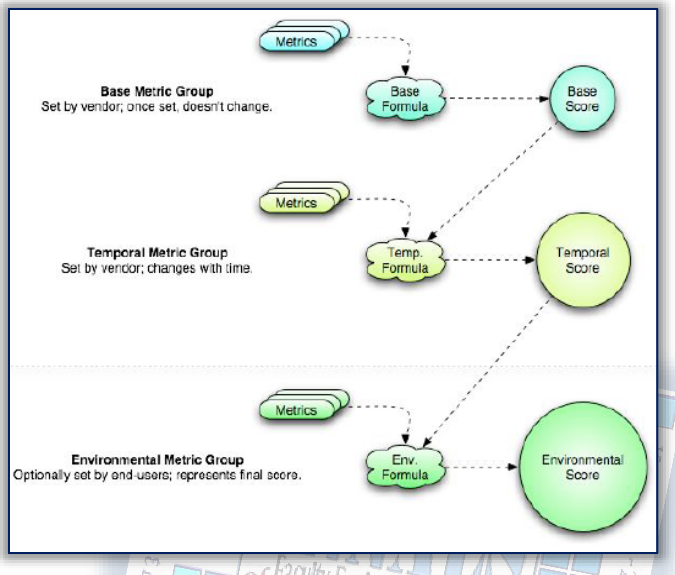

Metrics
Base Metric Group
Set by vendor; once set, doesn't change.
Base Formula
Base Score
Metrics
Temporal Metric Group
Set by vendor; changes with time.
Temp. Formula
Temporal Score
Metrics
Environmental Metric Group
Optionally set by end-users; represents final score.
Env. Formula
Environmental Score

Figure 3. CVSS – Scoring view

3.1.Base scoring

The base score is computed by vendors and coordinators. It combines the innate characteristics of the vulnerability. The base score has the largest bearing on the final score - computed primarily from the Impact Metrics. It represents the severity of the vulnerability.

The base equation is the foundation of CVSS scoring. The base equation is the following (NIST Interagency Report 7435):

```
BaseScore = round_to_1_decimal(((0.6*Impact)+(0.4*Exploitability)–1.5)*f(Impact))

Impact = 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact))

Exploitability = 20* AccessVector*AccessComplexity*Authentication f(impact)= 0 if Impact=0,

1.176 otherwise

AccessVector          = case AccessVector of
                               requires local access: 0.395 adjacent network
                               accessible: 0.646 network accessible: 1.0


AccessComplexity = case AccessComplexity of
                               high: 0.35
                               medium: 0.61
                               low: 0.71

Authentication        = case Authentication of
                               requires multiple instances of authentication: 0.45 requires single
                               instance of authentication: 0.56 requires no authentication: 0.704


ConfImpact            = case ConfidentialityImpact of
```

```
                 none: partial: complete:        0.0
                                                 0.275
                                                 0.660
IntegImpact      = case IntegrityImpact of none:      0.0
                 partial:                             0.275
                 complete:                            0.660

AvailImpact      = case AvailabilityImpact of none:   0.0
                 partial:                             0.275
                 complete:                            0.660
```

3.2.Temporal Scoring

The temporal score is computed by vendors and coordinators. It modifies the base score and it also allows for the introduction of mitigating factors to reduce the score of vulnerability. It is designed to be re-evaluated at specific intervals as the vulnerability ages. Further, it represents urgency at specific points in time. The temporal equation will produce a temporal score no higher than the base score, and no less than 33% lower than the base score. The temporal equation is shown below (NIST Interagency Report 7435).

```
TemporalScore =     round_to_1_decimal(BaseScor e *Exploitability
                    *RemediationLevel*ReportCon f idence)

Exploitability        = case Exploitability of
                            unproven:               0.85
                            proof-of-concept:       0.9
                            functional:             0.95
                            high:                   1.00
                            not defined:            1.00

RemediationLevel = case RemediationLevel of

                            official-fix:           0.87
                            temporary-fix:          0.90
                            workaround:             0.95
                            unavailable:            1.00
                            not defined:            1.00

ReportConfidence = case ReportConfidence of

                            unconfirmed:            0.90
                            uncorroborated:         0.95
                            confirmed:              1.00
                            not defined:            1.00
```

3.3.Environmental Scoring

The environmental score is computed by the end users. It adjusts the combined base-temporal score and should be considered as the final score. It represents a snapshot in time, tailored to an environment. User organizations will use this to prioritize the responses within their own environments. The environmental equation will produce a score no higher than the temporal score. The environmental equation is shown here (NIST Interagency Report 7435):

EnvironmentalScore = round_to_1_decimal((AdjustedTemporal+
*(10-AdjustedTemporal)*CollateralDamagePotential)*TargetDistribution)*

AdjustedTemporal = TemporalScore recomputed with the BaseScore's Impact sub- equation replaced with the AdjustedImpact equation

AdjustedImpact = min(10,10.41(1-(1-ConfImpact*ConfReq)*(1-IntegImpact*IntegReq)*
**(1-AvailImpact*AvailReq)))*

CollateralDamagePotential = case CollateralDama
none: low: *gePotential of 0*
low-medium: medium-high: high: *0.1*
not defined: *0.3*
0.4
TargetDistribution = case TargetDistribu *0.5*
none: low: medium: high: *0*
not defined:
tion of 0
0.25
0.75
1.00
1.00

ConfReq = case ConfReq of low: medium: high:
not defined:
0.5
1.0
1.51
1.0

IntegReq = case IntegReq of low: medium: high:
not defined:
0.5
1.0
1.51
1.0

AvailReq = case AvailReq of low: medium: high:
not defined:
0.5
1.0
1.51
1.0

3.4.Examples

1. The CVSS score distribution for all vulnerabilities[1]:

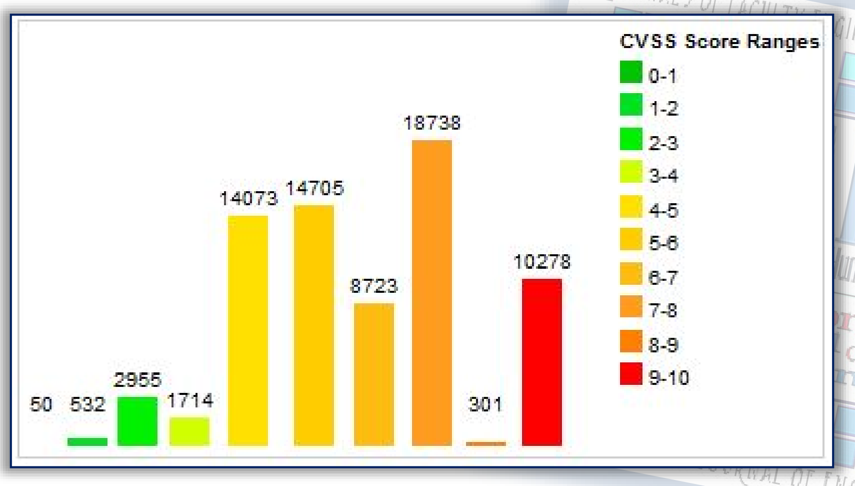


Figure 4. Vulnerability distribution by CVSS score (source: CVE Details)

[1] CVE Details, https://www.cvedetails.com/cvss-score-distribution.php

2. Consider the vulnerability CVE-2015-1337: Simple Streams does not properly verify the GPG signatures of disk image files, which allows remote mirror servers to spoof disk images and have unspecified other impact via a 403 (aka Forbidden) response.

The base vector for this vulnerability is [6]: AV:N/AC:M/Au:N/C:P/I:P/A:P.

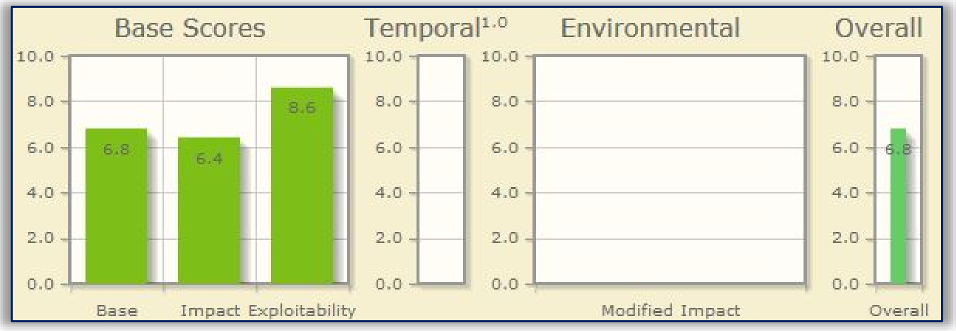


Figure 5. Base scores and overall score for CVE-2015-1337 (source: National Vulnerability Database)

CVSS base score equation (CVSS v2.10 Equations)

```
BaseScore = (.6*Impact +.4*Exploitability-1.5)*f(Impact)

Impact = 10.41 * (1 - (1 - ConfImpact) * (1 - IntegImpact) * (1 - AvailImpact))

Exploitability = 20 * AccessComplexity * Authentication * AccessVector

f(Impact) = 0 if Impact=0; 1.176 otherwise

AccessComplexity = case AccessComplexity of
                    high:   0.35
                    medium: 0.61
                    low:    0.71
Authentication   = case Authentication of
                    Requires no authentication:                 0.704
                    Requires single instance of authentication:    0.56
                    Requires multiple instances of authentication: 0.45

AccessVector     = case AccessVector of
                    Requires local access:    0.395
                    Local Network accessible: 0.646
                    Network accessible:       1

ConfImpact       = case ConfidentialityImpact of
                    none:        0
                    partial:     0.275
                    complete:      0.660

IntegImpact      = case IntegrityImpact of
                    none:        0
                    partial:     0.275
                    complete:      0.660

AvailImpact      = case AvailabilityImpact of
                    none:        0
                    partial:     0.275
                    complete:      0.660

CVSS Temporal Equation
 TemporalScore = BaseScore
            * Exploitability
            * RemediationLevel
            * ReportConfidence
 Exploitability   = case Exploitability of
                    unproven:          0.85
                    proof-of-concept:    0.9
                    functional:        0.95
                    high:            1.00
                    not defined         1.00
```

```
RemediationLevel = case RemediationLevel of
                     official-fix:       0.87
                     temporary-fix:      0.90
                     workaround:         0.95
                     unavailable:        1.00
                     not defined         1.00

ReportConfidence = case ReportConfidence of
                     unconfirmed:        0.90
                     uncorroborated:     0.95
                     confirmed:          1.00
                     not defined         1.00

CVSS Environmental Equation

EnvironmentalScore = (AdjustedTemporal
                     + (10 - AdjustedTemporal)
                     * CollateralDamagePotential)
                   * TargetDistribution

AdjustedTemporal = TemporalScore recomputed with the Impact sub-equation
                 replaced with the following AdjustedImpact equation.

AdjustedImpact = Min(10,
                   10.41 * (1 -
                          (1 - ConfImpact * ConfReq)
                         * (1 - IntegImpact * IntegReq)
                         * (1 - AvailImpact * AvailReq)))

CollateralDamagePotential = case CollateralDamagePotential of
                         none:          0
                         low:           0.1
                         low-medium:    0.3
                         medium-high:   0.4
                         high:          0.5
                         not defined:   0

TargetDistribution     = case TargetDistribution of
                         none:          0
                         low:           0.25
                         medium:        0.75
                         high:          1.00
                         not defined:   1.00

ConfReq      = case ConfidentialityImpact of
                     Low:          0.5
                     Medium:       1
                     High:         1.51
                     Not defined   1

IntegReq     = case IntegrityImpact of
                     Low:          0.5
                     Medium:       1
                     High:         1.51
                     Not defined   1

AvailReq     = case AvailabilityImpact of
                     Low:          0.5
                     Medium:       1
                     High:         1.51
                     Not defined   1
```

4. CVSS VERSION 3

CVSS v2 has been used by many organizations over the past years to rate vulnerabilities, but experts say this version has many faults and shortcomings. “While CVSSv2 saw improvements over CVSSv1, the scheme is still not adequately supporting real life usage, as it suffers from being too theoretical in certain aspects. Specific vulnerability types and vectors are not properly supported while others are not properly described, leading to subjective and inconsistent scoring, which CVSS was designed to prevent.” [7].

On June 10, 2015, following three years of receiving input from the representatives of a wide range of industries, FIRST made an announcement regarding the availability of CVSS v3, with the aim of providing a more robust and useful scoring system for vulnerabilities [8].

Enhancements are encompassed in the updated including: the promotion of consistency in scoring, the replacement of scoring tips so as provide clearer guidance for the end users of CVSS, and consideration of the system so it becomes more applicable to modern concerns [9].

Seth Hanford, co-chair of the FIRST CVSSv3 working group said "We hope that CVSS version 3 is clear, consistent and repeatable, and able to support the work of those who seek to understand, describe, compare, or evaluate IT vulnerabilities via a common scoring system."

CVSS v2.0 and v3.0[2]

The differences between two versions of CVSS are shown in the following table.

Table 4. CVSS v.2 and v.3

Version 2	Version 3
Vulnerabilities are scored relative to the overall impact to the host platform.	Vulnerabilities now scored relative to the impact to the impacted component.
No awareness of situations in which a vulnerability in one application impacted other applications on the same system.	A new metric, Scope, now accommodates vulnerabilities where the *thing suffering the impact* (the impacted component) is different from *the thing that is vulnerable* (the vulnerable component).
Access Vector may conflate attacks that require local system access and physical hardware attacks.	Local and Physical values are now separated in the Attack Vector metric.
In some cases, Access Complexity conflated the system configuration and user interaction.	This metric has been separated into Attack Complexity (accounting for system complexity), and User Interaction (accounting for user involvement in a successful attack).
In practice, the Authentication metric scores were biased toward two of three possible outcomes, and not effectively capturing the intended aspect of a vulnerability.	A new metric, Privileges Required, replaces Authentication, and now reflects the greatest privileges required by an attacker, rather than the number of times the attacker must authenticate.
Impact metrics reflected percentage of impact caused to a vulnerable application.	Impact metric values now reflect the degree of impact, and are renamed to None, Low and High.
The Environmental metrics of Target Distribution and Collateral Damage potential were not found to be useful.	Target Distribution and Collateral Damage potential have been replaced with Mitigating Factors.
CVSS v2.0 could not accommodate scoring multiple vulnerabilities used in the same attack.	While not a formal metric, guidance on scoring multiple vulnerabilities is provided with Vulnerability Chaining.
No formal qualitative scoring guidelines were provided.	Numerical ranges have been mapped to a 5-point qualitative rating scale.

There are several practical examples of numerical differences between versions 2 and 3:[3]

Table 5. Examples of Numerical Differences Between CVSS v2 and v3

Vulnerability	CVSS v2 Base Score	CVSS v3 Base Score
SSL/TLS MITM (CVE-2014-0224)	6.8	7.4
DokuWiki Reflected Cross-site Scripting Attack (CVE-2014-9253)	4.3	5.4
SearchBlox Cross-Site Request Forgery (CVE-2015-0970)	6.8	7.8
Apple iWork Denial of Service (CVE-2015-1098)	6.8	8.8

5. AGGREGATING INDIVIDUAL SCORES

CVSS is globally oriented towards the determination of individual vulnerabilities. It does not provide a direct procedure for aggregating individual scores into an overall metric of the targeted network. The existing approaches to aggregate CVSS scores usually cause loss of useful semantics of individual scores in the aggregated result.

The typical network topology that contains multiple individual results is given by the following image. Two special cases are considered. In case 1, it is supposed that host 1 runs a telnet service while host 2 runs the Universal Plug and Play (UPP) service. In case 2, host 1 and 2 change their

[2] https://www.first.org/cvss/user-guide

[3] https://www.first.org/cvss/examples

operating systems and corresponding services. In both cases, the firewalls stop any traffic except accesses to those services [10].

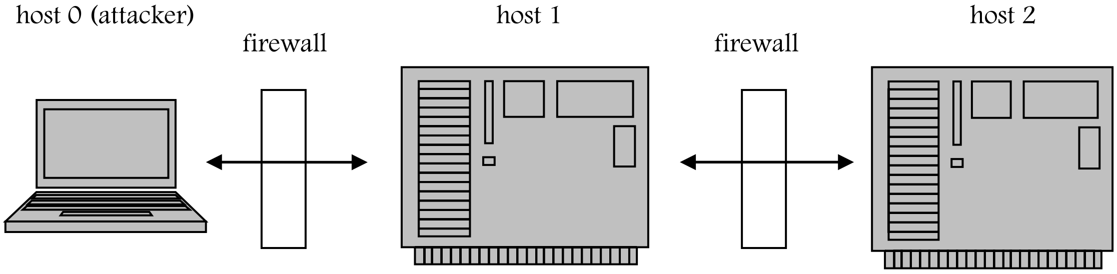


Figure 6. Network configuration (example)

The assumption is that the telnet service contains the vulnerability CVE-2007-0956 [6] (denoted by v_{telnet}), enabling remote attackers to avoid authentication and gain system accesses by via supplying the service with special usernames. The UPP service includes the vulnerability CVE-2007-1204 [6] (denoted by v_{UPP}), a stack overflow enabling attackers on the same subnet to execute arbitrary codes when they send specially crafted requests. Their CVSS base metrics [6] are presented in Table 6. Determination of the base score happens by implementing the CVSS calculator: 7.6 for v_{telnet} and 6.8 for v_{UPP} [10].

Table 6. CVSS Base Metrics and Scores of Analyzed Vulnerabilities [10]

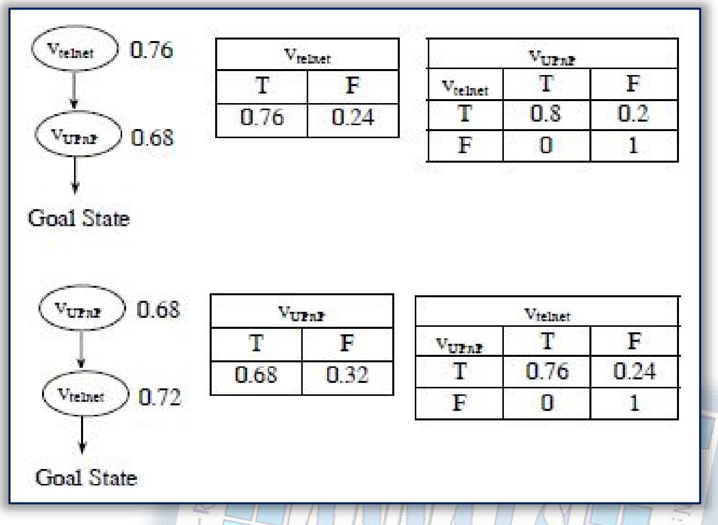
Metric Group	Metric	Value of v_{telnet}	Value of v_{UPP}
Exploitability	Access Vector Access Complexity Authentication	Network (1.00) High (0.35) None (0.704)	Adjacent Network (0.646) High (0.35) None (0.704)
Impact	Confidentiality Integrity Availability	Complete (0.660) Complete (0.660) Complete (0.660)	Complete (0.660) Complete (0.660) Complete (0.660)
Base Score		7.6	6.8

Existing approaches in aggregating obtained scores [10]:

- Average and maximum (naive approach): if one takes the average value (7.2 in both cases) and maximum value (7.6 in both cases).

- Attack graph-based approach: the CVSS base scores are changed into the attack graph-based approach [11]. Following this, the probabilities are aggregated based on these causal relationships: one can only reach an exploit if all preconditions are met; meeting a condition means there is at least one reachable exploit that the given condition has as its post-condition (i.e., a disjunction).

In case 1 of the previous example, it would be assigned 7.6/10 = 0.76 to (v_{telnet}, 0, 1), and 6.8/10 = 0.68 to (v_{UPP}, 1, 2) (and 1 to both conditions). Then the new value for (root, 1) is 0.76 and (v_{UPP}, 1, 2) and for (root, 2) is 0.76 × 0.68 = 0.52. Similarly, the same result is obtained for case 2.



V_{telnet}	
T	F
0.76	0.24

	V_{UPnP}	
V_{telnet}	T	F
T	0.8	0.2
F	0	1

V_{UPnP}	
T	F
0.68	0.32

	V_{telnet}	
V_{UPnP}	T	F
T	0.76	0.24
F	0	1

Figure 7. Bayesian network-based approach [10]

- Bayesian network (BN)-based approach [12]: This approach is illustrated by Figure 7. The left-hand side of the figure represents the BN, while the right-hand side shows the corresponding Conditional Probability Table. The upper graph and the tables are related to case 1 and the lower correspond for case 2.

In addition to these approaches, there are some improved variants described in [10].

6. CONCLUSION

The Common Vulnerability Scoring System gives a standard technique to government institutions and different organizations to rate the seriousness of vulnerabilities inside of their frameworks. The National Vulnerability Database provides a standard set of approved scores. When implemented into security items, NVD and CVSS empower organizations to comprehend the vulnerabilities' effect on their environments. Besides, the effect evaluations will be the same notwithstanding when the vulnerabilities are detected by various security tools utilized as a part of different subjects. This empowers logical correlation of the seriousness of vulnerabilities between government frameworks, and even organizations. Viewing the scores of the detected vulnerabilities after some time can help in identifying security trends. In that case, with a successful security strategy, organizations will experience upgrades in their vulnerability status over time.

References

[1.] U.S. Department of Commerce, National Institute of Standards and Technology (NIST), NIST Interagency Report 7435, The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems

[2.] United States Computer Emergency Readiness Team (US-CERT). US-CERT Vulnerability Note Field Descriptions. 2006, http://www.kb.cert.org/vuls/html/fieldhelp.

[3.] SANS Institute. SANS Critical Vulnerability Analysis Archive, http://www.sans.org/newsletters/cva/.

[4.] Microsoft Corporation. Microsoft Security Response Center Security Bulletin Severity Rating System. November 2002, http://www.microsoft.com/technet/security/bulletin/rating.mspx.

[5.] M. Schiffman, “The Common Vulnerability Scoring System”, The RSA conference, 2005, http://packetfactory.openwall.net/papers/CVSS/cvss-ppt.pdf

[6.] National Vulnerability Database (NVD), https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-1337

[7.] C. Eiram, B. Martin, “The CVSSv2 Shortcomings, Faults, and Failures Formulation, An Open Letter to FIRST”, https://www.riskbasedsecurity.com/reports/CVSS-ShortcomingsFaultsandFailures.pdf

[8.] SecurityWeek, FIRST Releases CVSS Version 3, http://www.securityweek.com/first-releases-cvss-version-3

[9.] FIRST, https://www.first.org/cvss

[10.] P. Cheng, L. Wang, S. Jajodia, A. Singhal, “Aggregating CVSS Base Scores for Semantics-Rich Network Security Metrics”, 31st International Symposium on Reliable Distributed Systems, 2012, pp. 31-40.

[11.] L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia, "An attack graph-based probabilistic security metric", In Proceedings of The 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec’08), 2008.

[12.] M. Frigault, L. Wang, A. Singhal, S. Jajodia, "Measuring network security using dynamic bayesian network", In Proceedings of ACM workshop on Quality of protection, 2008.

ANNALS of Faculty Engineering Hunedoara
– International Journal of Engineering