Quantitative Security

Colorado State University

Yashwant K Malaiya

CS 559

L 14: Security Metrics, CVSS

CSU Cybersecurity Center
Computer Science Dept

Quantitative Security

Colorado State University

Yashwant K Malaiya

CS 559

Security Metrics

CSU Cybersecurity Center
Computer Science Dept

To measure is to know.

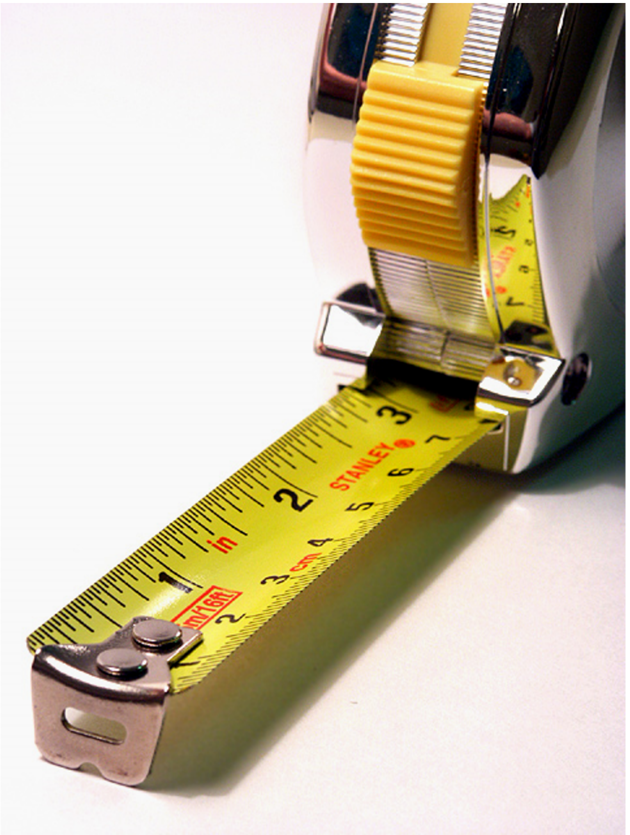
James Clerk Maxwell, 1831-1879

Measurement motivates.

John Kenneth Galbraith. 1908-2006

Metrics do matter

1. Metrics quantify the otherwise unquantifiable
2. Metrics can show trends and trends matter more than measurements do
3. Metrics can show if we are doing a good or bad job
4. Metrics can show if you have no idea where you are
5. Metrics establish where “You are here” really is
6. Metrics build bridges to managers
7. Metrics allow cross sectional comparisons
8. Metrics set targets
9. Metrics benchmark yourself against the opposition
10. Metrics create curiosity



Source: Andy Jaquith, Yankee Group, Metricon 2.0

Metrics

- Things that can be measures or predicted
  - Some metrics are more important than others
  - Some metrics can be correlated or redundant
    - Cyclomatic complexity and Lines of code
- Security metrics: contribute to determination of system security risk
- Classification by Pendleton et al
  - metrics of *system vulnerabilities*,
  - metrics of *defense strength*,
  - metrics of *attack (or threat) severity*,
  - metrics of *situation understanding*.

Marcus Pendleton, Richard Garcia-Lebron, Jin-Hee Cho, and Shouhuai Xu. 2016. A survey on systems security metrics. ACM Comput. Surv. 49, 4, Article 62 (December 2016), 35 pages. DOI: http://dx.doi.org/10.1145/3005714

Scales of Metrics

- **Nominal scale** is a naming scale, where variables are simply "named" or labeled, with no specific order.
  - Ex: blood types
- **Ordinal scale** has all its variables in a specific order, beyond just naming them.
  - Ex: “low income”, ”middle income”, ”high income”
- **Interval scale** offers labels, order, as well as, a specific interval between each its variable options.
  - Ex: temperature (Fahrenheit)
- **Ratio scale** bears all the characteristics of an interval scale, in addition to that, it can also accommodate the value of "zero" on any of its variables.
  - Weight, temp. in Kelvin

Terminology

- ***Enterprise systems*** refer to networked systems of multiple computers/devices, clouds, or even the entire cyberspace.
- ***Computer systems*** represent individual computers/devices. We interchangeably use the terms node, device, or computer to refer to a single entity.
- **Attackers:** These are *attacking entities* representing computers or IP addresses from which cyber attacks are launched against other normal entities.
- **Incident:** It represents a successful attack (e.g., malware infection or data breach).

Attack-defense interactions in an enterprise system at time *t*

At time *t*, the enterprise system consists of *n* entities (i.e., computers), denoted by the vector

$C(t) = \{c1(t), \ldots, cn(t)\}$

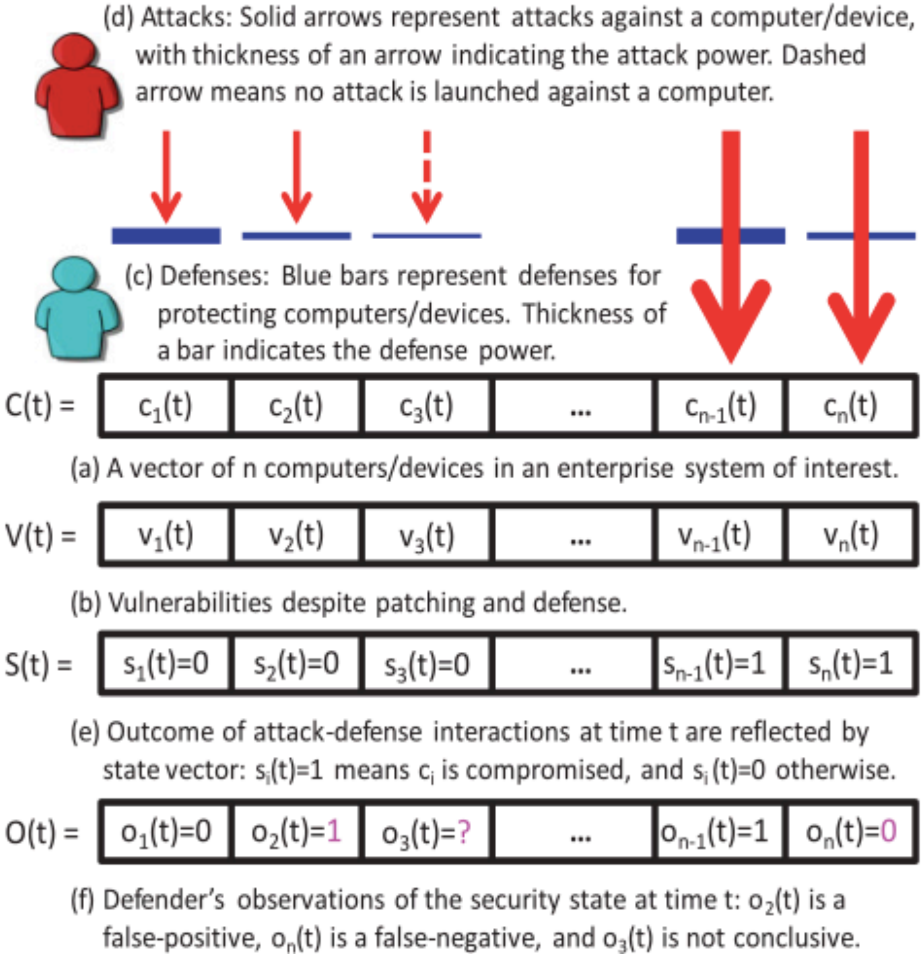
n may vary with time.

Each entity, *ci*(*t*), has a vector v*i*(*t*) of vulnerabilities, such as zero-day and/or some unpatched software vulnerabilities.

The outcome of the attack-defense interaction reflected by a global **security state vector**

$S(t) = \{s1(t),...,sn(t)\}$,

where *si*(*t*) = 0 means entity *ci*(*t*) is secure at time *t* and *si*(*t*) = 1 means entity *ci*(*t*) is compromised at time *t*



(d) Attacks: Solid arrows represent attacks against a computer/device, with thickness of an arrow indicating the attack power. Dashed arrow means no attack is launched against a computer.

(c) Defenses: Blue bars represent defenses for protecting computers/devices. Thickness of a bar indicates the defense power.

$C(t) =$ $c_1(t)$ | $c_2(t)$ | $c_3(t)$ | ... | $c_{n-1}(t)$ | $c_n(t)$

(a) A vector of n computers/devices in an enterprise system of interest.

$V(t) =$ $v_1(t)$ | $v_2(t)$ | $v_3(t)$ | ... | $v_{n-1}(t)$ | $v_n(t)$

(b) Vulnerabilities despite patching and defense.

$S(t) =$ $s_1(t)=0$ | $s_2(t)=0$ | $s_3(t)=0$ | ... | $s_{n-1}(t)=1$ | $s_n(t)=1$

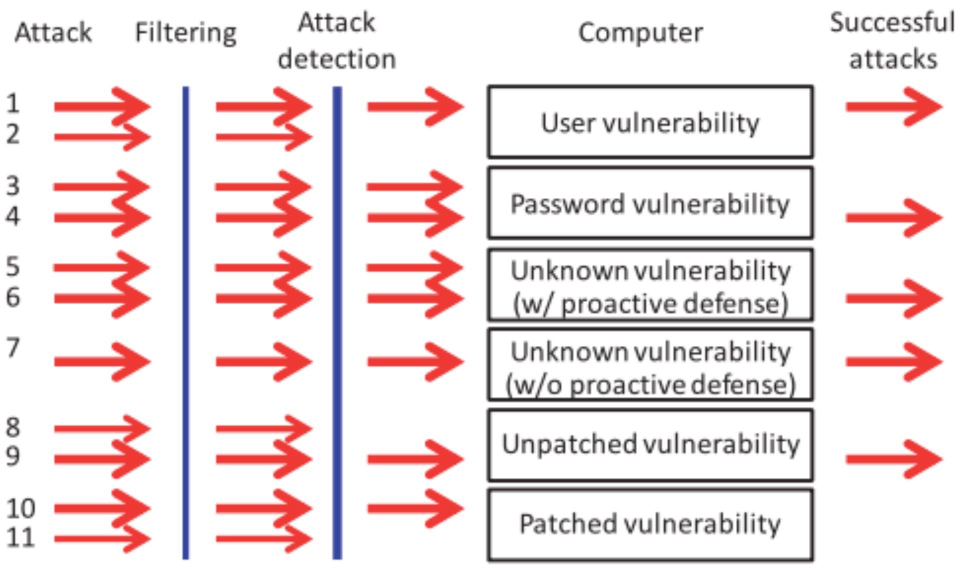
(e) Outcome of attack-defense interactions at time t are reflected by state vector: $s_i(t)=1$ means c_i is compromised, and $s_i(t)=0$ otherwise.

$O(t) =$ $o_1(t)=0$ | $o_2(t)=1$ | $o_3(t)=?$ | ... | $o_{n-1}(t)=1$ | $o_n(t)=0$

(f) Defender's observations of the security state at time t: $o_2(t)$ is a false-positive, $o_n(t)$ is a false-negative, and $o_3(t)$ is not conclusive.

Marcus Pendleton, Richard Garcia-Lebron, Jin-Hee Cho, and Shouhuai Xu. 2016. A survey on systems security metrics. ACM Comput. Surv. 49, 4, Article 62 (December 2016), 35 pages. DOI: http://dx.doi.org/10.1145/3005714

Attack-defense interactions in a computer (or device), *ci* (*t*)



Filtering: mechanisms deployed at the enterprise system perimeter to block traffic from malicious or blacklisted IP addresses,
Use of some *attack detection* mechanisms to detect and block attacks before they reach *ci* (*t*),
Use of proactive defense mechanisms (e.g., address space randomization) to mitigate vulnerabilities exploitable by attackers (i.e., exploitability).

Situations

- *Situation*(*t*)= *f*(*V*(*t*), *D*(*t*), *A*(*t*))
  - where *V*(*t*) is a function of *vulnerabilities* at time *t*,
  - *D*(*t*) is a function of *defenses* at time *t*, and
  - *A*(*t*) is a function of *attacks* at time *t*.
- *S*(*t*) is naturally affected by *V* (*t*), *D*(*t*), and *A*(*t*) as well.

Security Metrics

- Most researchers have focused on a specific component of he overall risk and have used various metrics for their work.
  - A single component is not the whole risk.
  - $\text{Risk}_i = \text{Likelihood}_i \text{ x Impact}_i$
  - Likelihood_i = P{A security hole_l is exploited}.

    = P{hole_i present}. P{exploitation|hole_i present}
- Pendleton, Garcia-Lebron, Cho, and Xu. 2016. A survey on systems security metrics.
  - *1. system vulnerabilities,*
  - *2. defense strength and 3. attack severity,*
  - *4. situation*
- Four set of metrics is examined next.

Metrics for Measuring User Vulnerabilities

- Metrics for Measuring Phishing Susceptibility.
  - Typical metrics are false positives (FP) or false negatives (FN), where FP indicates the percentage of flagging genuine email as phishing email while FN captures the percentage of detecting a phishing email as a genuine email.
- Metrics for Measuring Malware Susceptibility.
  - closely related to a user’s online behavior. Users who often install many applications are more likely exposed to malware. If users visit many websites, then there is a higher vulnerability for malware infection
- Metrics for Measuring Password Vulnerabilities.
  - *Entropy* is the most intuitive metric to measure the strength of a password. Also *password guessability etc.*

Metrics for Measuring Interface-Induced Vulnerabilities

- *Attack surface* metrics [Manadhata and Wing 2011] aim to measure the ways by which an attacker can compromise a targeted software.
  - Many attacks against a software can be conducted by entering data from the environment (in which the software runs) to the software (e.g., buffer overrun) or by receiving data via interactions with the software.
  - These attacks typically interact with the software by connecting to a channel (e.g., socket) or invoking a method (e.g., API) offered by the software or sending/receiving data items to/from the software.

Metrics for Measuring Software Vulnerabilities

- *Metrics for measuring the evolution of vulnerabilities*
  - Historical: vulnerabilities, exploited vulnerabilities
  - Future: vulnerabilities, exploited vulnerabilities
  - *Tendency-to-be-exploited metric* measures the tendency that a vulnerability may be exploited, which may be derived from information sources such as posts on Twitter before vulnerability disclosures
- *Metrics for measuring vulnerability lifetime*
  - measures how long it takes to patch a vulnerability since its disclosure
  - Higher severity vulnerability may be patched faster
- *Metrics for measuring severity of individual vulnerabilities*
  - *CVSS etc.*
- *Metrics for Measuring Severity of a Collection of Vulnerabilities*
  - Attacks requiring multiple vulnerabilities

Metrics for Measuring the Strength of Defenses

- **Metrics for Measuring the Strength of Preventive Defenses**
  - Metrics for Blacklisting
    - ***Reaction time metric*** captures the delay between the observation of the malicious entity at time *t* and the blacklisting of the malicious entity at time *t'*
    - ***Coverage metric*** estimates the portion of blacklisted malicious players
  - Others
- **Metrics for Measuring the Strength of Reactive Defenses**
  - Metrics for Monitoring
    - Coverage, redundancy, fault-tolerance
    - cost

Metrics for Strength of Defense Mechanisms

- **Metrics for the Individual Strength of Defense Mechanisms**
  - ***Detection time***
  - ***Intrusion detection metrics***
    - ***True-positive rate, False-negative rate etc***
    - ***Receiver operating characteristic*** (ROC)
    - ***Cost metric***
- **Metrics for systems with Defense Mechanisms**
  - **Relative Strength of Defense Mechanisms**
  - **Collective Strength of Defense Mechanisms**

Metrics for Measuring the Strength Defenses

- **Metrics for Measuring the Strength of Proactive Defenses**
  - Address Space Layout Randomization (ASLR)
  - Moving Target Defense (MTD)
- **Metrics for Measuring the Strength of Overall Defenses**
  - **Penetration resistance** (PR) can be measured by running a penetration test to estimate the level of effort (e.g., person-day or cost) required for a red team to penetrate into a system

Red teams are offensive security professionals who are experts in attacking systems and breaking into defenses. Blue teams are defensive security professionals responsible for maintaining internal network defenses against all cyber attacks and threats. Red teams simulate attacks against blue teams to test the effectiveness of the network's security.

ATTACK METRICS

- **Metrics for Measuring Zero-Day Attacks**
  - ***Lifetime of zero-day attacks*** measures the period of time between when an attack was launched and when the corresponding vulnerability is disclosed to the public.
  - ***Victims by zero-day attacks*** measures the number of computers compromised by zero-day attacks.
- **Metrics for Measuring Targeted Attacks**
  - ***targeted threat index =*** social engineering tactic sophistication x technical sophistication of the malware in the attacks
- **Metrics for Measuring Botnets**
  - ***Botnet size*** - the number of bots
  - ***Network bandwidth*** – that a botnet can use
  - ***Botnet efficiency*** -the network diameter of the botnet network topology
  - ***Botnet robustness*** the robustness of botnets under random or intelligent disruptions

Attack Metrics

- **Metrics for Measuring Malware Spreading**
  - The ***infection rate*** metric, denoted by γ , measures the average number of vulnerable computers that are infected by a compromised computer (per time unit) at the early stage of spreading.
- **Metrics for Measuring evasion of Attack Evasion Techniques**
  - Metrics for Measuring Adversarial Machine Learning Attacks
  - Metrics for Measuring Obfuscation Attacks

SITUATION METRICS

- **Metrics for Measuring Security State**
  - Data-Driven State Metrics
    - ***Network maliciousness metric*** estimates the fraction of blacklisted IP addresses in a network
    - ***Rogue network metric*** captures the population of networks used to launch drive-by download or phishing attacks
    - ***ISP badness metric*** quantifies the effect of spam from one ISP or Autonomous System (AS) on the rest of the Internet
    - more
  - Model-Driven Metrics
    - ***Fraction of compromised computers***
    - ***Probability a computer is compromised at time t***

Metrics for Measuring Security Incidents

- Measuring Frequency of Security Incidents
  - ***Encounter rate*** measures the fraction of computers that encountered some malware or attack during a period of time
  - ***Incident rate*** measures the fraction of computers successfully infected or attacked at least once during a period of time
  - ***Blocking rate*** is the rate an encountered attack is successfully defended by a deployed defense
  - ***Breach frequency metric*** measures how often breaches occur
  - ***Breach size metric*** gives the number of records breached in individual breaches
  - ***Time-between-incidents metric*** measures the period of time between two incidents
  - ***Time-to-first-compromise metric*** estimates the duration of time between when a computer starts to run and the first malware detection alarm is triggered

Measuring Damage of Security Incidents

- ***Delay in incident detection*** measures the time between the occurrence and detection implying that a longer delay is a higher damage.
- ***Cost of incidents*** may include both the direct cost (e.g., the amount of lost money) and the indirect cost (e.g., negative publicity and/or the recovery cost)
  - More on this later.

Metrics for Measuring Security Investment

- ***Security spending*** indicates a percentage of IT budget. This metric is important
  - enterprises want to know whether their security expenditure is justified by the security performance and
  - is comparable to other organizations' security investments.
- ***Security budget allocation*** estimates how the security budget is allocated to various security activities and resources
- ***Return on security investment*** (ROSI) measuring the financial net gain of an investment based on the gain from investment minus the cost of investment
  - Since security is not a *real* investment (i.e., not generating a revenue), the ROSI metric actually measures the reduction in the loss caused by incompetent security
  - ***Net present value*** measures the difference between the present economic value of future inflows and the present economic value of outflows with respect to an investment

Quantitative Security

Colorado State University

Yashwant K Malaiya

CS 559

CVSS

CSU Cybersecurity Center
Computer Science Dept

Topic Outline

CVSS Metrics/Scores Outline

- Introduction/history
- The Base Score
  - The Base metrics and their values
- Temporal Score
- Environmental Score
- Question of validation

CVSS: Common Vulnerability Scoring System

- How important is a specific vulnerability?
  - Essentially a risk measure
  - Vulnerabilities with highest scores need addressing quickly. Those with lowest scores are low priority.
- CVSS v1: *National Infrastructure Advisory Council* (NIAC), 2005
- CVSS v2: *Forum of Incident Response and Security Teams* (FIRST)
  - 2007
  - Still common
- CVSS V3: 2015
  - Getting common

CVSS Scoring System

- How important is a specific vulnerability?
  - Essentially a risk measure*
  - Vulnerabilities with highest scores need addressing quickly. Those with lowest scores are low priority.
- CVSS v1: *National Infrastructure Advisory Council* (NIAC), 2005
- CVSS v2: *Forum of Incident Response and Security Teams* (FIRST)
  - 2007
  - Still common
- CVSS V3: 2015
  - Getting common

* As we will see

CVSS Metrics and Scores

- De facto standard for assessing severity of software vulnerabilities
- Objective: prioritize effort to address vulnerabilities
- Metrics: Components by levels, each translated into a numerical metric
- Scores: Computed score using a set of metrics as given by formulas
  - Formulas based on committee judgement, not derived or proven
- Three metric groups and associated scores;
  - Base (mandatory): intrinsic to the vulnerability
  - Temporal: time-dependent variation in risk
  - Environmental: risk component dependent on the organization’s environment

* As we will see

CVSS Base Metrics & Score: Overview

- CVSS Base Score = f(Exploitability sub-score, Impact sub-score,)
  - This is the score used to prioritize vulnerabilities*.
  - Ranges from 0 to 10.
- Exploitability metrics: how easy is the vulnerability to exploit
  - Exploitability sub-score = f(Attack Vector, Attack Complexity, Privileges Required, User Interaction, Scope)
- Impact sub-score: what is the extent of the impact.
  - Impact sub-score = f(Confidentiality Impact, Integrity Impact, Availability Impact)
- Intrinsic value of a raw metric ranges from 0 to 1.
- Our observation: exploitability sub-score attempts to measure *likelihood*$_i$, Impact sub-score attempts to measure *impact*$_i$ due to a *vulnerability i.*

* Details to follow

CVSS Base Scores: Ratings

All recognized vulnerabilities are assigned Base Scores*. The complete data for recognized vulnerabilities is available at the National Vulnerability Database.

Ranking according to CVSS 3.0.

V 3.0 Severity Rating	Base Score Range
None	0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critial	9.0 - 10.0

V 2.0 Severity Rating	Base Score Range
Low	0.0-3.9
Medium	4.0-6.9
High	7.0-10.0

* Has this measure been validated? Interesting questions.

CVSS Metrics: Data bases

NVD (National Vulnerability database): U.S. government repository of standards based vulnerability management data. represented using the Security Content Automation Protocol (SCAP).

- sponsored by the Department of Homeland Security (DHS), NCCIC and US-CERT.

Documentation/Data hosted at NVD

- Example:
- MySQL Stored SQL Injection (CVE-2013-0375)
- Oracle MySQL 5.1.66 and earlier, and 5.1.28 and earlier
- CVSS v2 Base Score: 4.3
- CVSS v3.0 Base Score: 6.4 medium
- Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:N

* Has this measure been validated? Interesting questions.

CVE Statuses in NVD

- Received: CVE has been recently published to the CVE dictionary and has been received by the NVD.
- Awaiting Analysis: about 24 hours
- Undergoing Analysis: CVE is currently being analyzed by NVD staff, results in association of reference link tags, CVSS scores
- Analyzed
- Modified: CVE has been amended by a source (CVE Primary CNA or another CNA)
- Deferred
- Rejected: These CVEs are in the NVD, but currently do not show up in search results.

*

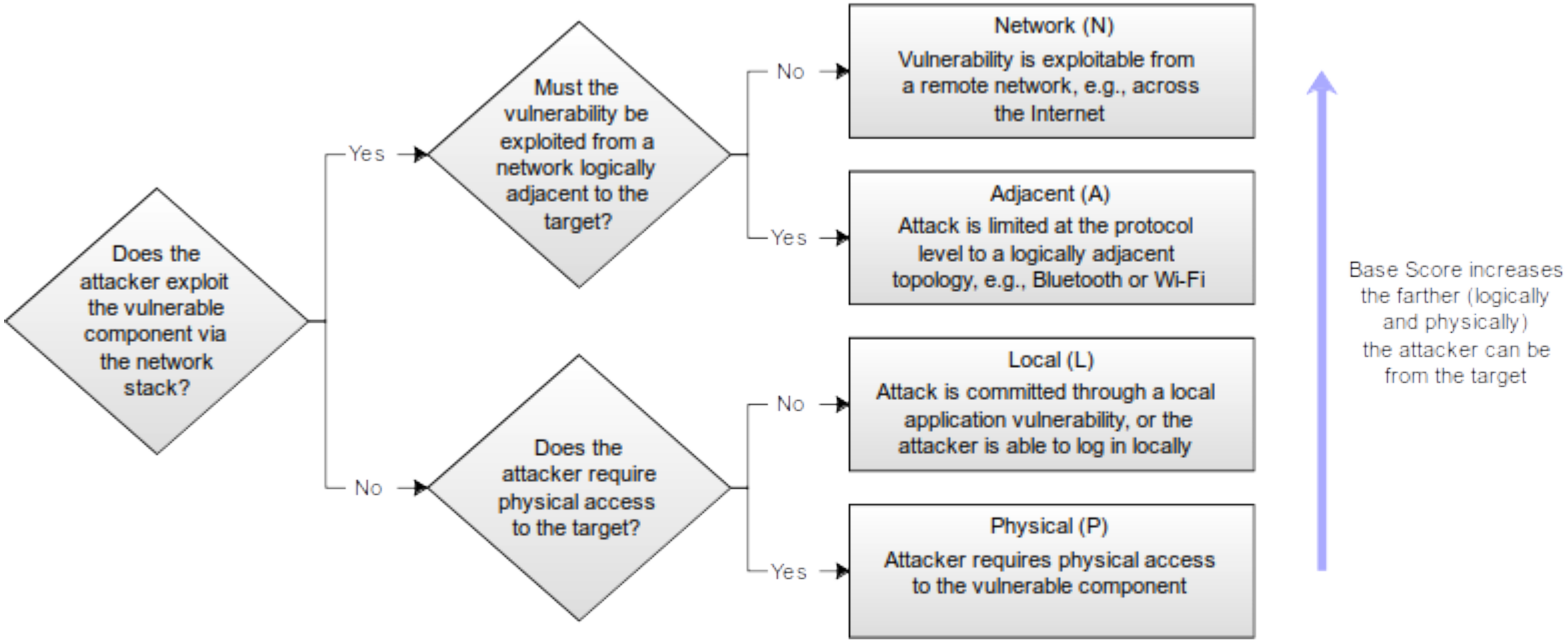
Details: Base Metrics

Exploitability components:

Attack Vector (AV)	**context by which vulnerability exploitation is possible**	
Metric level	**Represents**	**Value V3.0**
Network (N)	“remotely exploitable” one or more network hops away	0.85
Adjacent (A)	attack is limited to the same shared physical/logical network	0.62
Local (L)	exploitable with local access	0.55
Physical (P)	requires the attacker to physically access vulnerable component	0.2

Attack Complexity (AC)	**conditions beyond the attacker’s control that must exist in order to exploit the vulnerability**	
Metric level	**Represents**	**Value V3.0**
Low	No specialized access conditions required	0.77
High	A successful attack depends on specific conditions beyond the attacker's control.	0.44

Graphical visualization: Attack vector



Does the attacker exploit the vulnerable component via the network stack?
Yes
Must the vulnerability be exploited from a network logically adjacent to the target?
No
Does the attacker require physical access to the target?
No
Yes
No
Yes
Network (N)
Vulnerability is exploitable from a remote network, e.g., across the Internet
Adjacent (A)
Attack is limited at the protocol level to a logically adjacent topology, e.g., Bluetooth or Wi-Fi
Local (L)
Attack is committed through a local application vulnerability, or the attacker is able to log in locally
Physical (P)
Attacker requires physical access to the vulnerable component
Base Score increases the farther (logically and physically) the attacker can be from the target

For other flow-charts see: [CVSS User Guide]

Details: Base Metrics

Exploitability components (cont):

Privileges Required (PR)	**the level of privileges an attacker must possess before successfully exploiting the vulnerability.**	
Metric level	**Represents**	**Value V3.0**
None (N)	attacker is does not require any access to settings or files to carry out an attack	0.85
Low (L)	The attacker needs basic user privileges for the attack	0.62 (0.68 if scope/ modified scope is changed)
High (H)	attacker needs administrative privileges	.27 (0.50 if scope/ modified scope is changed)

User Interaction (UI)	**whether a separate user (or user-initiated process) must participate in some manner**	
Metric level	**Represents**	**Value V3.0**
None (N)	can be exploited without interaction from any user	0.85
Required (R)	requires a user to take some action before the vulnerability can be exploited	0.62

Details: Base Metrics

Scope change : When the vulnerability governed by one authorization scope is able to affect resources governed by another authorization scope.

Scope	impact to the confidentiality/Integrity/Availability of the information resources	
Metric level	**Represents**	**Value V3.0**
Unchanged (U)	An exploited vulnerability can only affect resources managed by the same authority.	-
Changed (C)	An exploited vulnerability can affect resources beyond the authorization privileges intended by the vulnerable component	Modified scope

Impact : The Impact metrics refer to the properties of the impacted component.

Confidentiality Impact (C) Integrity Impact (I) Availability Impact (A)	impact to the confidentiality/Integrity/Availability of the information resources	
Metric level	**Represents**	**Value V3.0**
High (H)	Total loss	0.56
Low (L)	loss is constrained	0.22
None (N)	no loss	0

Base Score: Formulas

Exploitability sub-score

- Exploitability = $8.22 \times Attack\ Vector \times Attack\ Complexity$
  $\times Privilege\ \ Required \times User\ Interaction$

Impact sub-score (ISC)

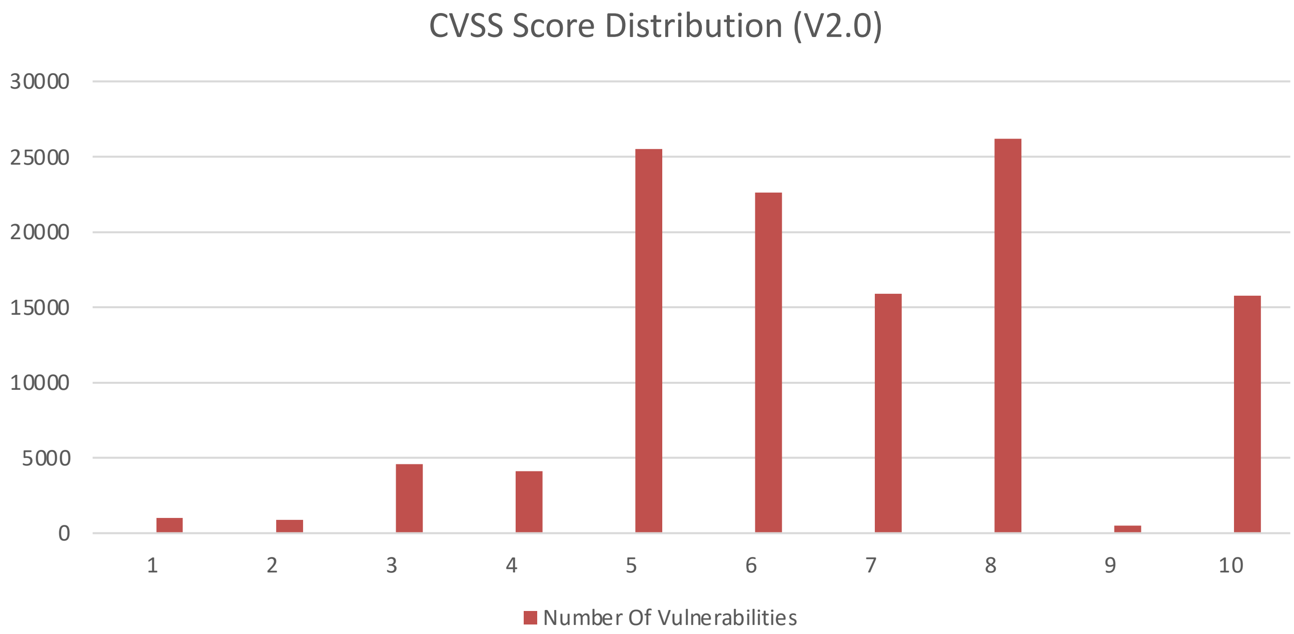
- $ISCBase = 1 - [\ (1 - ImpactConf) \times (1 - \ ImpactInteg) \times (1 - \ ImpactAvail)]$

Base Score is

- If (Impact sub score =< 0)
  = 0 else,
- Scope Unchanged
  $= Round\ up\ (Minimum\ [(Impact + Exploitability), 10])$
- Scope Changed
  $= Round\ up\ (Minimum\ [1.08 \times (Impact + Exploitability), 10])$

All the subjects and objects under the jurisdiction of a single *security authority* are considered to be under one *security scope*. If a vulnerability in a vulnerable component can affect a component which is in a different *security scope* than the vulnerable component, a Scope change occurs

Base Score: Distribution



Using CVEDetails Data for 117,132 vulnerabilities

V 2.0 Severity Rating	Base Score Range
Low	0.0-3.9
Medium	4.0-6.9
High	7.0-10.0

Temporal Metrics

Temporal metrics measure the current state: exploit availability, patches or workarounds, or the confidence in the description of the vulnerability.

Note: An exploit is a piece of code that exploits the vulnerability. Available exploits are documented at the Exploit databse.

My Comment: Good idea, but the concept needs some work.

Exploit Code Maturity (E)	**context by which vulnerability exploitation is possible**	
Metric	**Represents**	**Value V3.0**
Not Defined (X)	No information available	1
High (H)	widely available exploit code works in every situation	1
Functional (F)	exploit code works in most situations	0.97
Proof-of-Concept (P)	Proof-of-concept exploit available but an attack demonstration is not practical for most systems	0.94
Unproven (U)	No exploit code is available, or an exploit is theoretical	0.91

Temporal Metrics

Remediation Level (RL): patched yet?

Remediation Level (RL)	context by which vulnerability exploitation is possible	
Not Defined (X)	No information available	1
Unavailable (U)	no solution available	1
Workaround (W)	an unofficial, non-vendor solution available	0.97
Temporary Fix (T)	official but temporary fix available	0.96
Official Fix (O)	official patch, or an upgrade is available	0.95

Report Confidence:

Report Confidence (RC)	the degree of confidence in the existence of the vulnerability	
Not Defined (x)	No information available	1
Confirmed (C)	Detailed reports exist	1
Reasonable (R)	researchers do not have full confidence in the root cause	0.96
Unknown (U)	questionable report	0.92

Temporal Metrics

Remediation Level (RL): patched yet?

Remediation Level (RL)	context by which vulnerability exploitation is possible	
Not Defined (X)	No information available	1
Unavailable (U)	no solution available	1
Workaround (W)	an unofficial, non-vendor solution available	0.97
Temporary Fix (T)	official but temporary fix available	0.96
Official Fix (O)	official patch, or an upgrade is available	0.95

Report Confidence:

Report Confidence (RC)	the degree of confidence in the existence of the vulnerability	
Not Defined (x)	No information available	1
Confirmed (C)	Detailed reports exist	1
Reasonable (R)	researchers do not have full confidence in the root cause	0.96
Unknown (U)	questionable report	0.92

Temporal score and comments

Temporal score

$$= Round\ up(Base\ Score \times Exploit\ Code\ Maturity \times Remediation\ Level \times Report\ Confidence)$$

Comments*: The significance of the presence of an Exploit and a Patch is obvious. It is not clear by all the values are close to 1.0, i.e. the levels did not appear to matter for the FIRST people.*

Note that

- Most vulnerabilities are declared along with patch releases. However users often do not apply the patches immediately.
- Known exploits often appear quickly, if the vulnerability is significant.
  - *However we have found a lack of correlation between vulnerability severity and delay in exploit appearance. It may be explained that a high severity vulnerability may already have disappeared by in new release when it is disclosed, making exploit development largely useless.*

Environmental Metrics

The environmental score

= If (Modified Impact Sub score =< 0) 0

else,

Modified Scope Unchanged

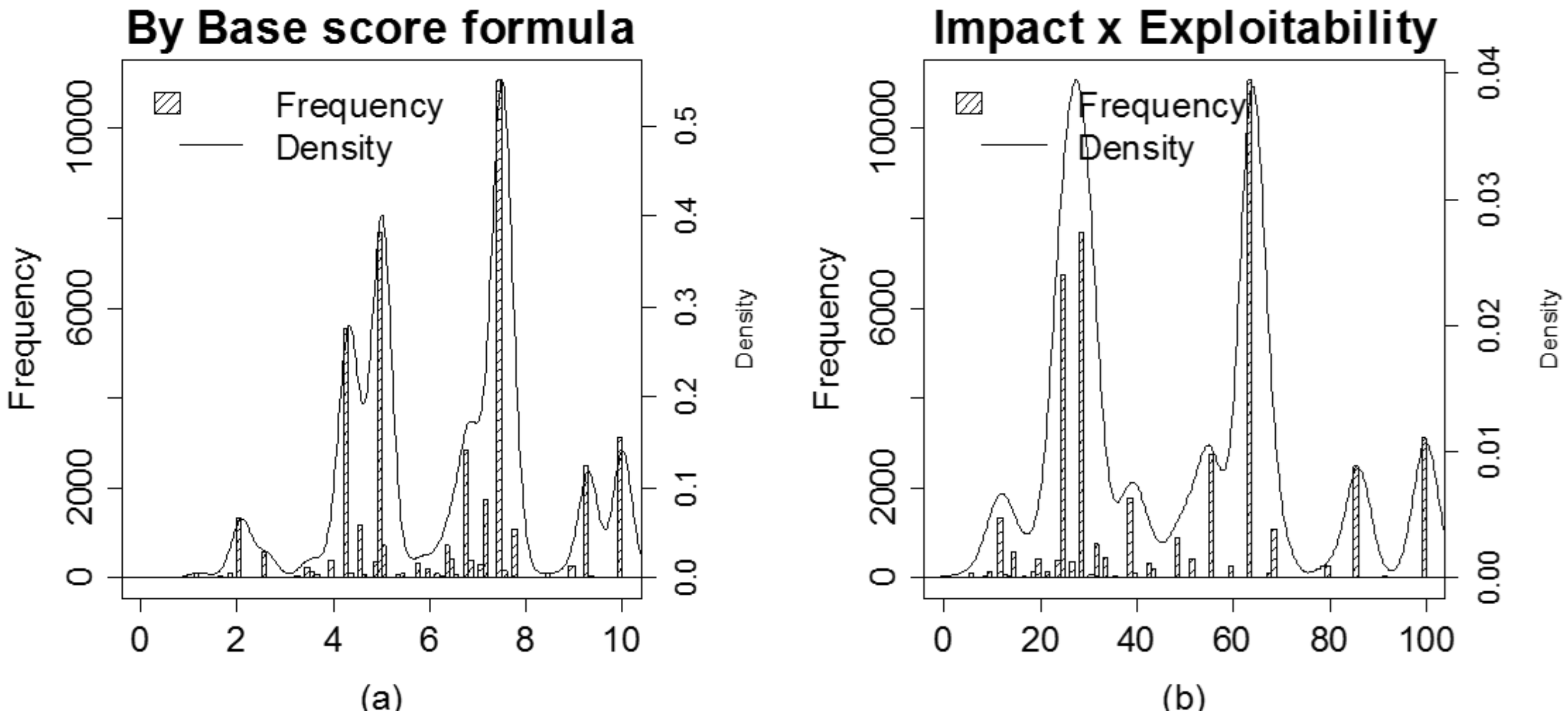
$= Round\ up(Minimum\ [(M.Impact + M.Exploitability) \times Exploit\ Code\ Maturity \times Remediation\ Level \times Report\ Confidence, 10])$

Modified Scope Changed

$= Round\ up(\ Round\ up[Minimum[1.08 \times (M.Impact + M.Exploitability)], 10] \times Exploit\ Code\ Maturity \times Remediation\ Level \times Report\ Confidence))$

44

CVSS: How useful it is?



(a)

(b)

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Combinations
(a)	0	5	6.8	6.341	7.5	10	63
(b)	0	29	49	48.59	64	100	112

H. Joh and Y. K. Malaiya, "Defining and Assessing Quantitative Security Risk Measures Using Vulnerability Lifecycle and CVSS Metrics," SAM'11, The 2011 International Conference on Security and Management, pp.10-16, 2011

Has CVSS worked?

- Windows 7 Correlation among
  - CVSS Exploitability
  - Microsoft Exploitability metric
  - Presence of actual exploits
- No significant correlation found.
- Continuing research

Variables	Exploit Existence	MS-EXP	CVSS-EXP
Exploit Existence	1	-0.078	-0.146
MS-EXP	**-0.078**	1	-0.116
CVSS-EXP	**-0.146**	-0.116	1

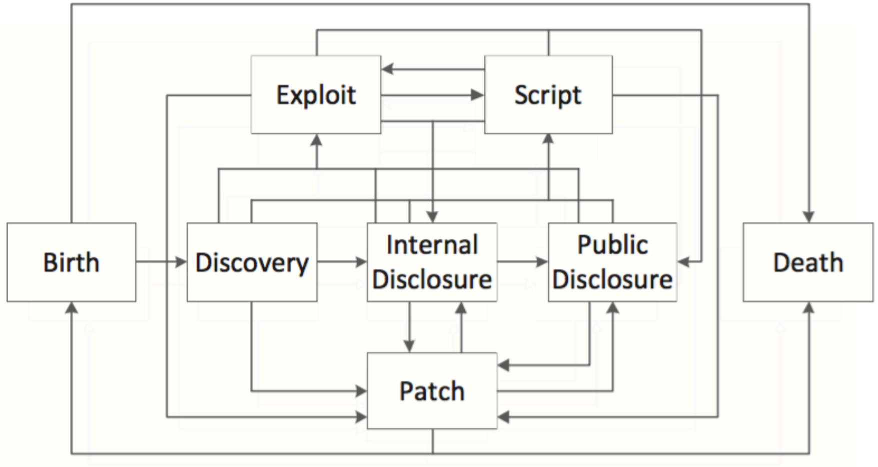
A. Younis and Y.K. Malaiya, "Comparing and Evaluating CVSS Base Metrics and Microsoft Rating System", The 2015 IEEE Int. Conf. on Software Quality, Reliability and Security, pp. 252-261

Colorado State University

Likelihood of Individual Vulnerabilities Discovery

- **Ease of discovery**
  - Human factor (skills, time, effort, etc.), Discovery technique, Time

- Time:



- Apache HTTP server
- CVE-2012-0031, **(01/18/2012)**
- V. 1.3.0→1998-06-06

Time to Discovery = Discovery Time Date – First Effected version Release Date

Colorado State University

Correlation: Access complexity vs Time to Discover

❖ AC vs Time

- AC= Low

2.000

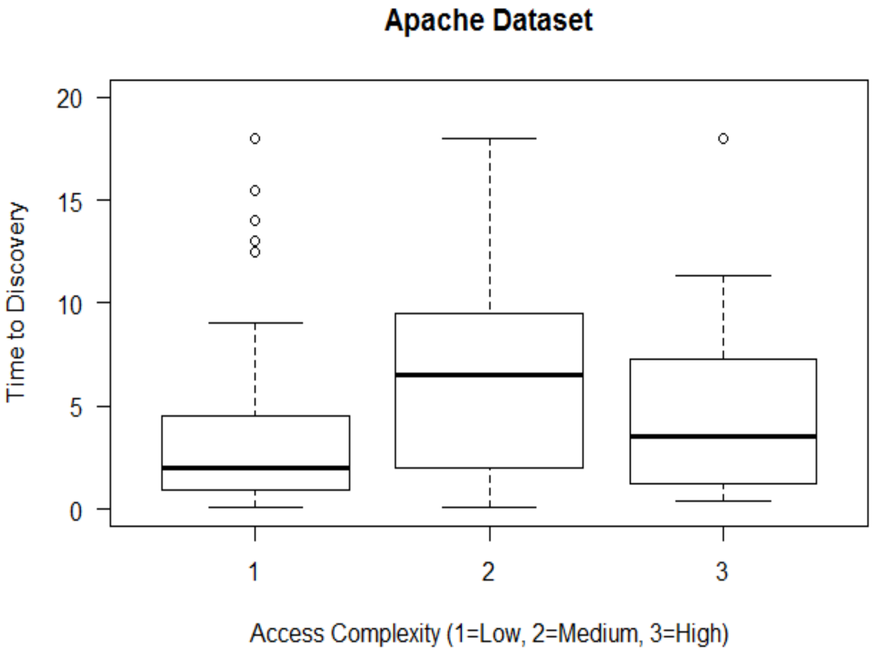
- AC= Medium

6.500

- AC= High (very few points)

3.500

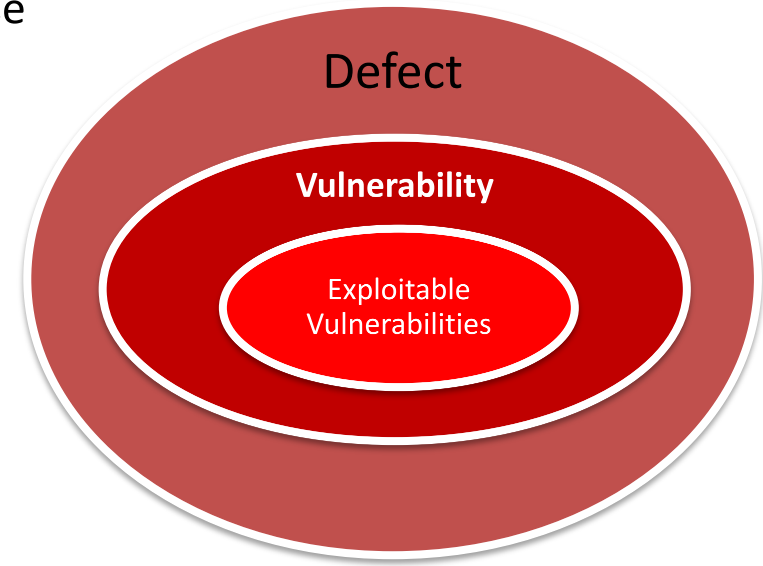
- Some correlation between Access Complexity and Time to Discover



Colorado State University

Characterizing Vulnerability with Exploits

- **1 to 5 %** of defects are vulnerabilities.
- Finding vulnerabilities can take considerable expertise and effort.
- Out of 49599 vulnerabilities reported by NVD, **2.10% have an exploit**.
- A vulnerability with an exploit written for it presents more risk.
- **What characterizes a vulnerability having an exploit?**



Vulnerability	In-Degree	Out-Degree	CountPath	ND	CYC	Fan-In	No of Invocation	SLOC	Exploit Existence
CVE-2009-1891	1	9	9000	6	68	45	2	211	NEE
CVE-2010-0010	4	9	145	4	11	16	4	38	EE
CVE-2013-1896	26	5	8	1	5	37	3	29	EE