Quantitative Security

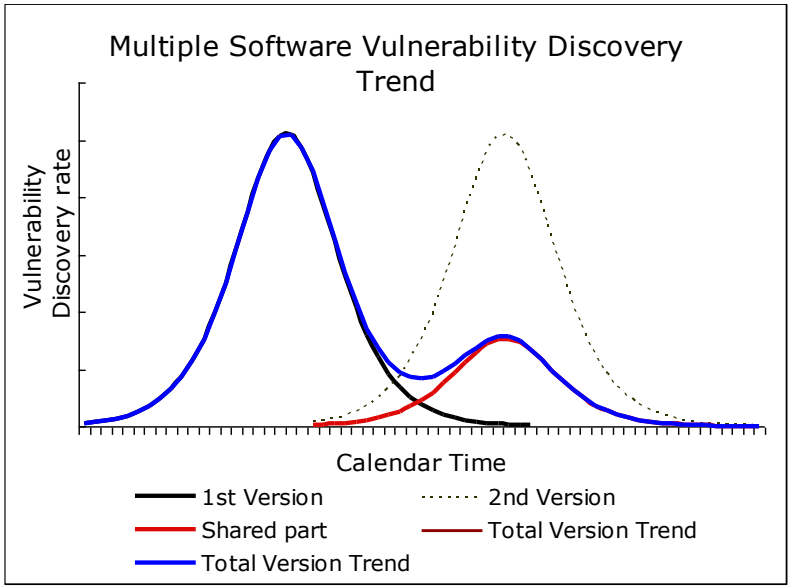
Colorado State University
Yashwant K Malaiya

CS 559
L 14: Security Metrics, CVSS



CSU Cybersecurity Center
Computer Science Dept

Multi-version Vulnerability Discovery Model



$$\Omega(t) = \frac{B}{BCe^{-ABt}+1} + \alpha \frac{B'}{B'C'e^{-A'B'(t-\varepsilon)}+1}$$

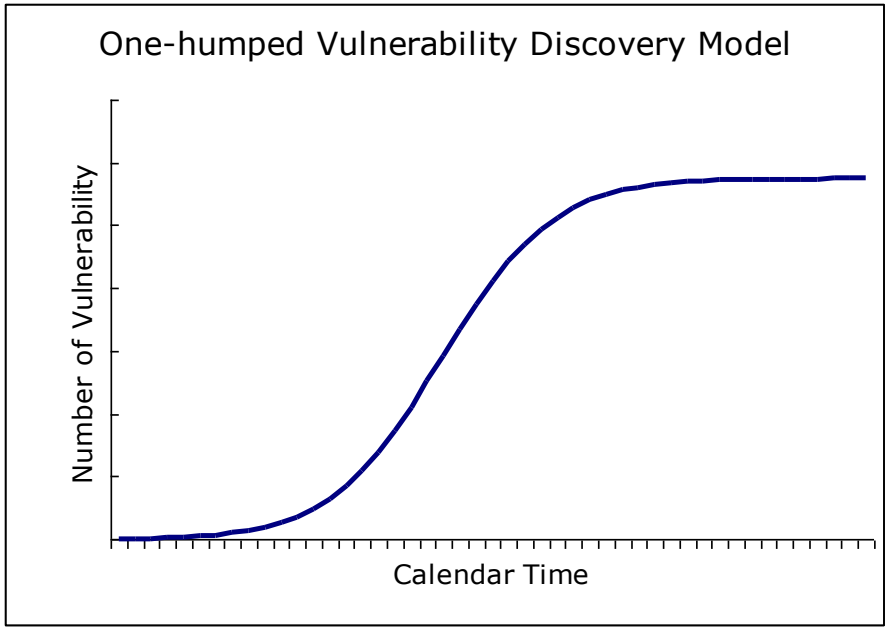
Parameter α indicates shared components such as shared code and shared functionality, and the parameter ε is the time lag between the release dates of the two versions.

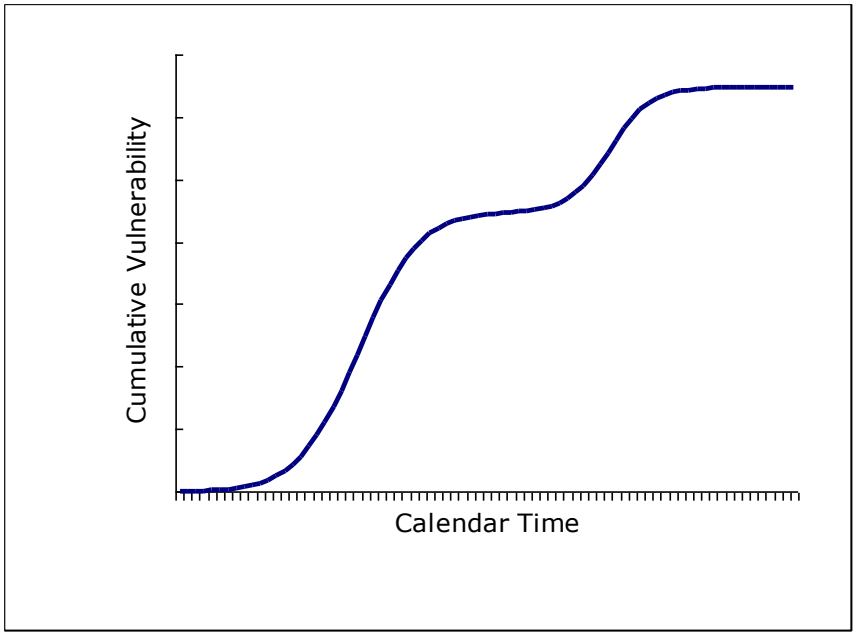
	Previous Version	Next Version	Shared Code Ratio α
Apache	1.3.24 (3-21-2002)	2.0.35 (4-6-2002)	79.84%
Mysql	4.1.1 (12-1-2003)	5.0.0 (12-22-2003)	83.52%

http://www.cs.colostate.edu/~malaiya/p/kim07.pdf

2

One vs Two Humps

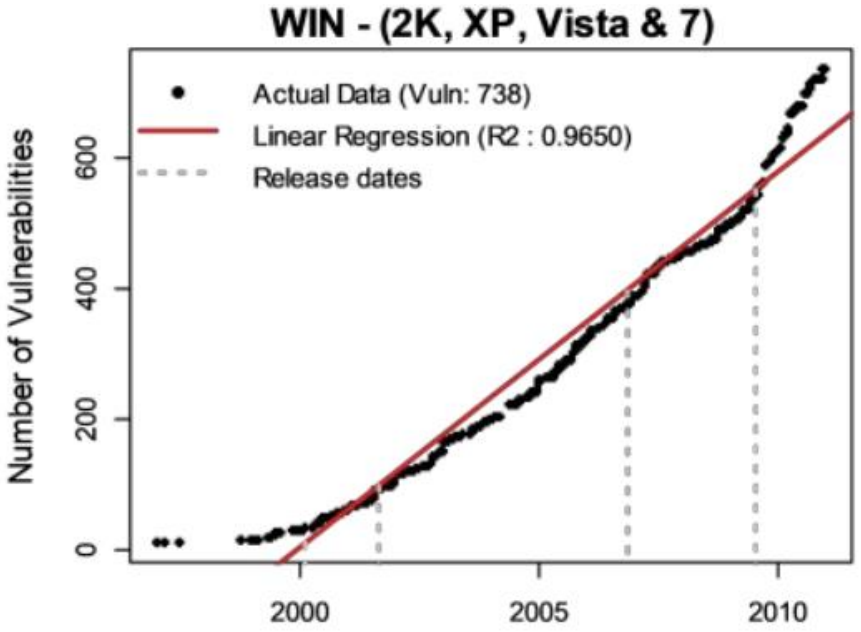


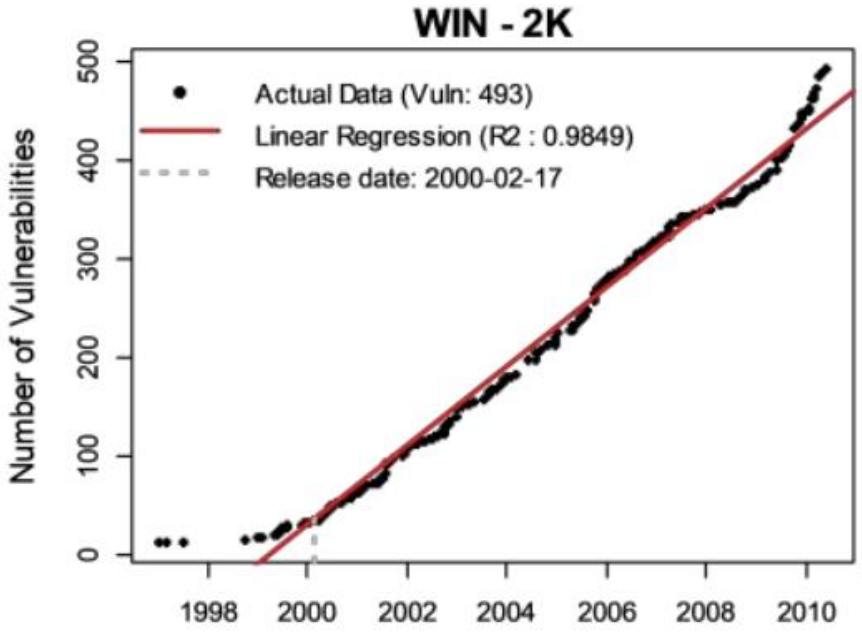


Superposition affect

Linear model

- Because of nearly continuous evolution, the linear phase may get stretched.





- If the evolution rate is steady, the size of the pool of undiscovered vulnerabilities stays the same (vulnerabilities removal rate = injection rate)
- If the market share is steady, the number of vulnerability finders remains steady.

Joh's thesis

4

Examining Seasonality

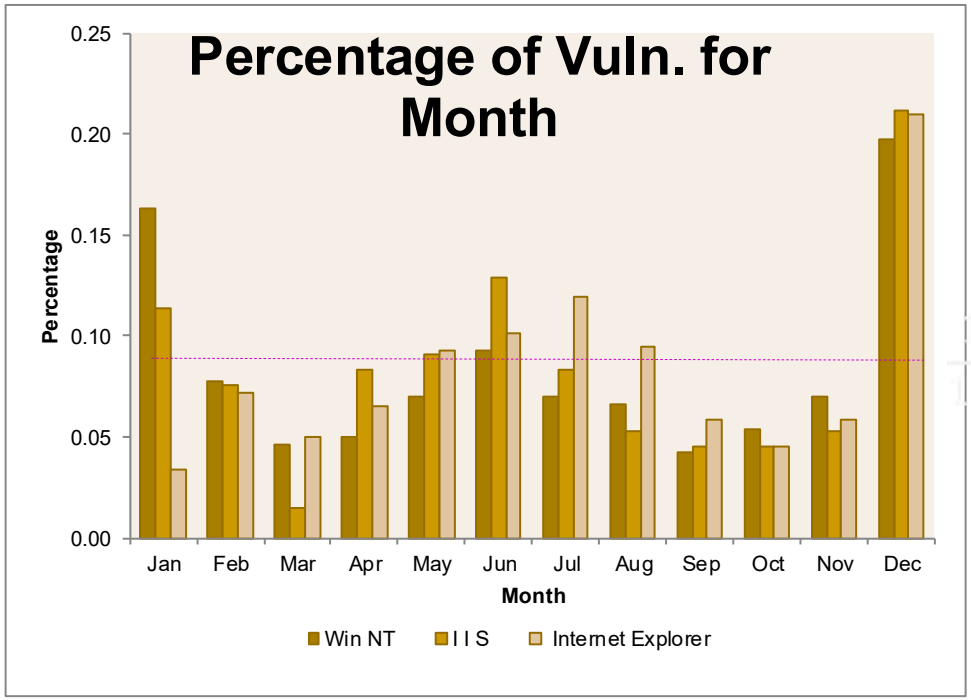
- Is the seasonal pattern statistically significant?
- Periodicity of the pattern
- Analysis:
  - Seasonal index analysis with test
  - Autocorrelation Function analysis
- Significance
  - Enhance VDMs' predicting ability
- Annual and Weekly seasonality

Colorado State University

Annual: Prevalence in Month

Vulnerabilities Disclosed

	WinNT '95~'07	IIS '96~'07	IE '97~'07
Jan	42	15	15
Feb	20	10	32
Mar	12	2	22
Apr	13	11	29
May	18	12	41
Jun	24	17	45
Jul	18	11	53
Aug	17	7	42
Sep	11	6	26
Oct	14	6	20
Nov	18	7	26
Dec	51	28	93
Total	258	132	444
Mean	21.5	11	37
s.d.	12.37	6.78	20.94



6

Colorado State University

‘Seasonality Timing System’

Norman Fosback 1970s, head of the Institute for Econometric Research

1. To exploit **positive seasonality around the turns of each month**: Buy at the close of the third-to-last trading of each month, and sell at the close of the fifth trading session of the following month.

2. To exploit **positive pre-holiday seasonality**: Buy at the close of the third-to-last trading day prior to exchange holidays, and sell at the close of the last trading day before a holiday.

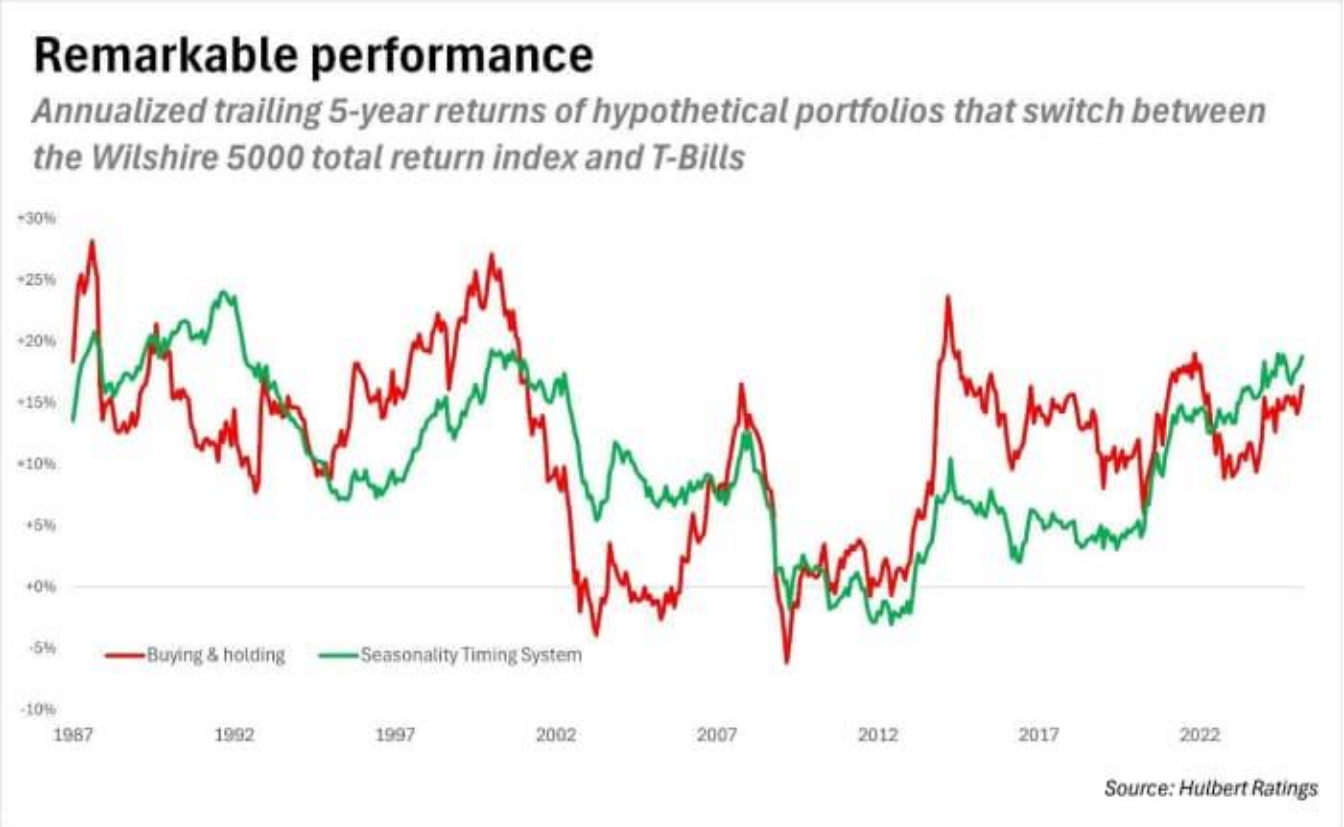
3. Exceptions: If the holiday falls on a Thursday, sell at Friday’s close rather than Wednesday’s. Also, if the last day before a holiday is the first trading day of the week, don’t sell until the day after the holiday. Finally, never sell on the first trading day after options expire; instead wait an extra day.

Beats buy-and-hold: Mark Hulbert

Sell in May (Halloween) and go away: Stay invested during November to April.

20 years . May-October 0.55 November-April 4.89

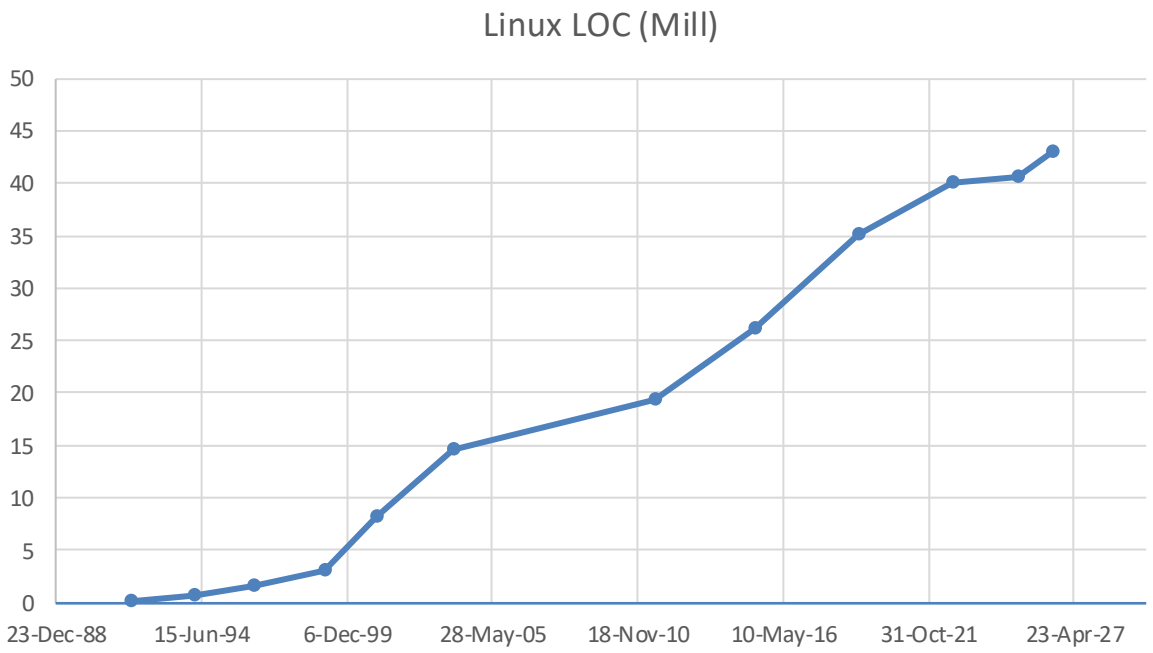
‘Seasonality Timing System’



This timing system beats the market using the calendar.
Mark Hulbert March 24, 2025

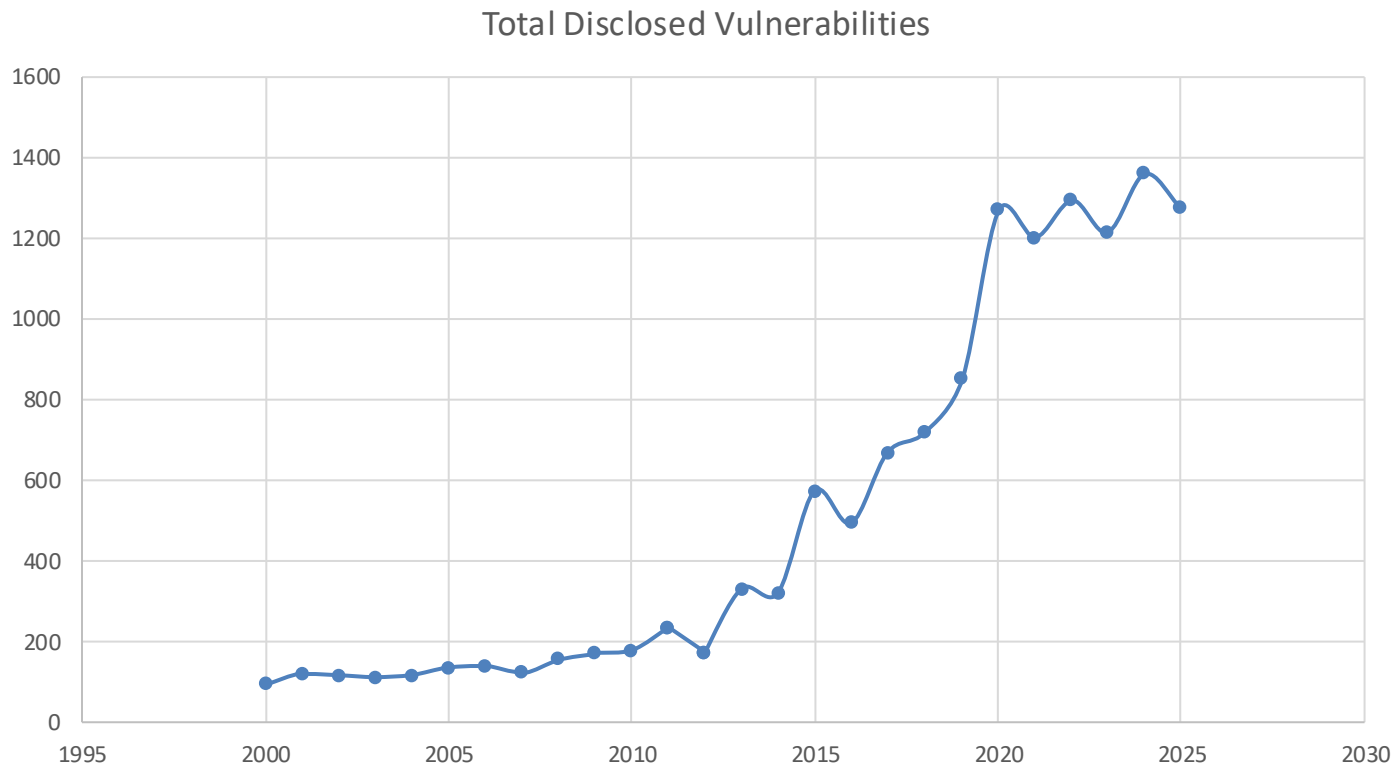
Colorado State University

Long Term Trends: software sizes



Size evolution: Linux kernel size
Linux 1.0-7.2: Oct 1991-June 2026

Long Term Trends: Microsoft Vulnerabilites



Long term Trends: Total vulnerabilities, Microsoft products

Note: Covid19: became common in January 2020

Long term trends

Likely factors that affect long-term trends

- Better understanding of safer coding practices
  - Fewer vulnerabilities injected?
- Better vulnerability discovery tools (fuzzers) and more finders
  - Higher vulnerability discovery rates
- More software products
  - More vulnerabilities to be found
- Increasing code sizes
  - Injection of new vulnerabilities
- More software products

11

Vulnerability Discovery and Risks

What factors impact risk?

- Not the vulnerabilities that have been found and patched
- Vulnerabilities that have been discovered but not patched
  - Before disclosure: black hat people/organizations
  - after disclosure: when patch development is taking time
  - Vulnerabilities with patches, but patches not applied
- Statistical modeling may be needed for assessing probability of breaches

12

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

Maxwell's equations for electromagnetism



Measurement motivates.

John Kenneth Galbraith. 1908-2006

Economist, Ambassador to India 1961-63



14

Scales of Metrics

- **Nominal scale** is a naming scale, where variables are simply "named" or labeled, with no specific order.
  - Ex: blood types A, B, AB, O
- **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, money, temp. in Kelvin

What you should question

- A claim should probably be tentatively accepted if
  - It is consistent with well established, carefully researched observations
  - Credibility of the researchers and publication
- Question a claim if
  - You think you can come up with a better idea
- Researchers (unlike managers) do not claim they know everything.



Colorado State University

Security Metrics

- Pendleton, Garcia-Lebron, Cho, and Xu. 2016. A survey on systems security metrics. Classification:
  - Enterprise system with n machines
  - *1. system vulnerabilities*, user, interface, software
  - *2. defense strength* proactive, reactive
  - *3. attack severity*, zero-day, targeted, botnets, malware etc
  - *4. situation (secure or compromised)* *internet, enterprise, breach rates etc*
  - Four sets of metrics is examined next. Search literature for details.

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

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

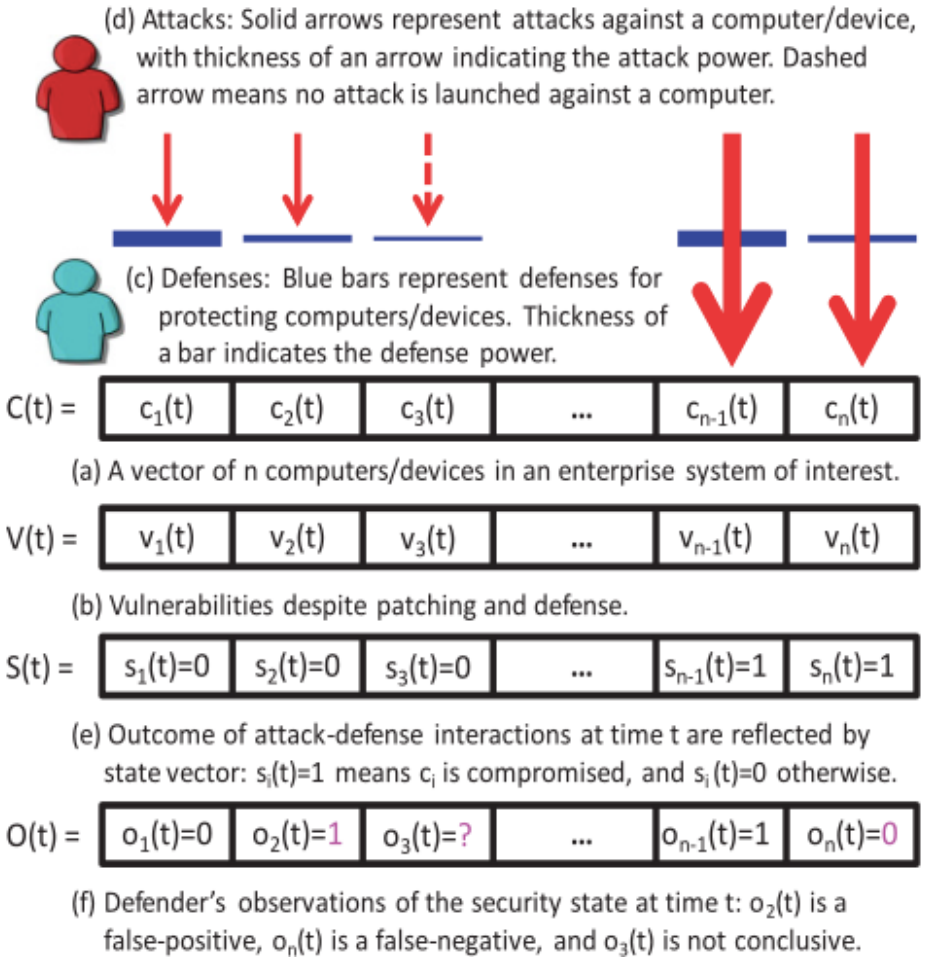
n may vary with time.

Each entity, $ci(t)$, has a vector $vi(t)$ 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**

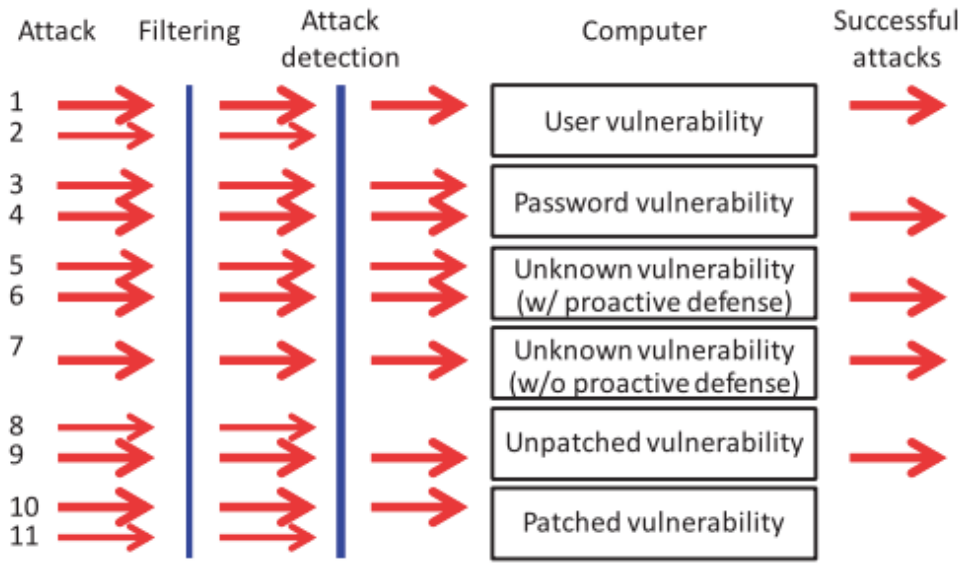
$S(t) = \{s1(t),\ldots,sn(t)\}$,

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*



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.
  - $Risk_i = Likelihood_i \times Impact_i$
  - $Likelihood_i = P\{\text{A security hole}_i \text{ is exploited}\}$.

    $= P\{\text{hole}_i \text{ present}\}.\ P\{\text{exploitation} | \text{hole}_i \text{ present}\}$
- 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

4. 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***

5. 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.

6. 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

36

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 found
- CVSS V3: 2015
  - Common
- CVSS V4: Nov 2023
  - Recently introduced

37

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
- CVSS V4: Nov 2023 coming into use

* As we will see

38

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 (detailed tables soon)
- 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
  - Thread ~~Temporal~~: time-dependent variation in risk (renamed Threat metrics in V4)
  - Environmental: risk component dependent on the organization's environment
  - Supplemental (V4)

* As we will see

39

CVSS Base Metrics & Score: Overview

CVSS Base Score = f(Exploitability sub-score, Impact sub-score,)

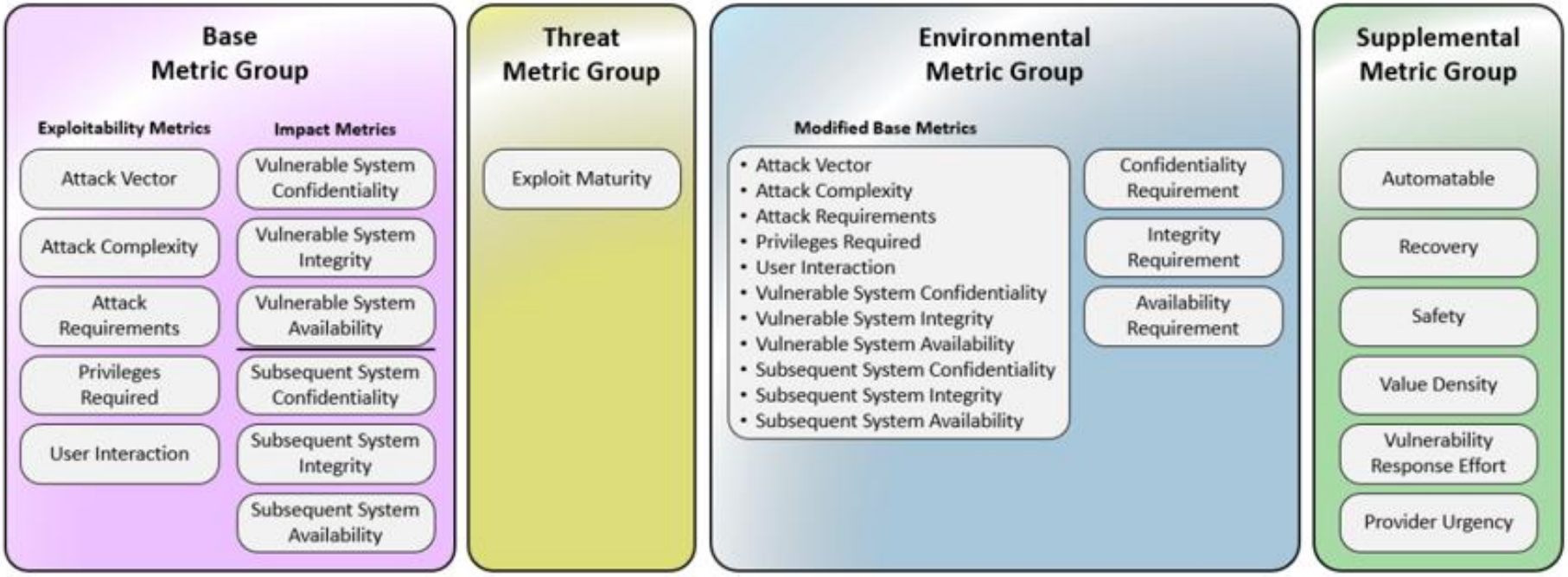
- This is the score used to prioritize vulnerabilities*.
- Ranges from 0 to 10.

- Exploitability metrics: how easy is the vulnerability to exploit
  - Exploitability sub-score = f(Attack Vector, Attack Complexity, Attack Requirements, Privileges Required, User Interaction~~, Scope~~)
- Impact sub-score: what is the extent of the impact.
  - Impact sub-score = f(Confidentiality, Integrity,, Availability, subsequent integrity and availability)
- 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

40

CVSS V4.0 Metrics



CVSS Metric Groups

FIRST: [Forum of Incident Response and Security Teams]

883 Teams in nine countries

Colorado State University

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 V3/V4.

Rating	CVSS Score	Response time
None	0.0	Next major release accept risk
Low	0.1 - 3.9	Next major release accept risk
Medium	4.0 - 6.9	90-day SLA maintenance window
High	7.0 - 8.9	30-day SLA planned sprint
Critical	9.0 - 10.0	24-72 hours Emergency

* Has this measure been validated? Interesting question.

42

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 v3.0 Base Score: 5.4 medium
  - Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:N
- CVSS v4.0 Base Score: not yet assessed

CVSS v3.1 Severity and Metrics:
Base Score: 5.4 MEDIUM
Vector: AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N
Impact Score: 2.5
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): Low
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): Low
Integrity (I): Low
Availability (A): None

* Has this measure been validated? Interesting questions.

43

CVE Statuses in NVD

NVD analyzes each vulnerability received. Possible status at a time:

- **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.

*

44