Quantitative Cyber-Security

Colorado State University
Yashwant K Malaiya

CS559

Midterm Review



CSU Cybersecurity Center
Computer Science Dept

Midterm

- Sec 001: Coming Tuesday Nov 3 during class time
- Sec 801 (online): 1 hr 15 min, Wed 12:10 AM-11:50 PM
- Closed book, closed notes
- Must bring/use laptop with camera with Respondus lockdown browser installed and tested. Access code will be provided in the room (S001) or by email (S801).
- Use calculator included in the browser
- One blank sheet permitted for scratch work.
  - Sec 001: Provided, turn in before leaving
  - Sec 801: Show both sides at the beginning and at the end. Must be destroyed in front of the camera.
- We cannot define terms, explain concepts, provide hints, or help in any way that will benefit a specific student. Questions on typos and language can be asked but none during the first 15 minutes.
- You cannot leave the room without permission.

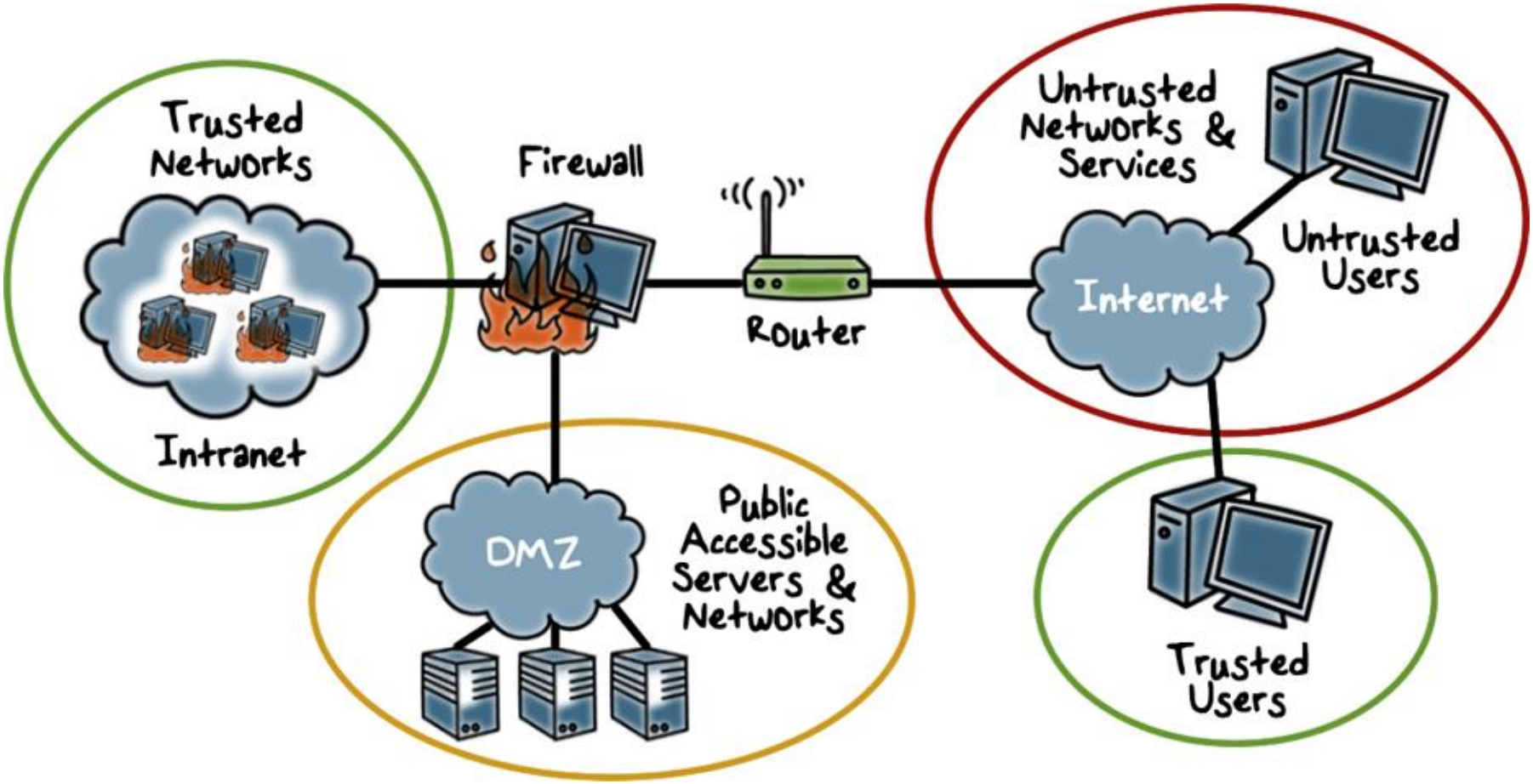
Quick Review

- The midterm covers everything we have done so far.
- Lectures, assignments, lecture notes, quizzes etc. are all included.
- Today's lecture is a sampler.
- How to prepare:
  - Listen to the lectures carefully
  - Reflect while working on the quizzes and assignments.

Main topics L1, L2

- Quantitative Perspective of the problem
- Security system architecture
  - Internet, trusted systems, firewalls, OSs, virtualization
- Assets, Threats, Vulnerabilities
- Cyber attack types, attack surfaces
- Malware: Viruses, worms etc
- Access Control:
  - Subjects, Objects, and Access Rights
  - Access Control Schemes
- Authentication

Firewalls



DMZ: “Demilitarized zone”, distributed firewalls, From Georgia Tech
Note multiple levels of trust.

Colorado State University

Example: Access Control Matrix

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Access Control List (ACL): Every object has an ACL that identifies what operations subjects can perform. Each access to object is checked against object's ACL.

May be kept in a relational database. Access recorded in file metadata (inode).

Main topics L3

- How to do research
  - Literature search, sources, reading papers
  - Original research
  - Publication, significance, citations
- Security frameworks
- NIST Cybersecurity Framework
  - Functions and categories
  - Implementations and priorities
- CIS Critical Security Controls
  - Basic, Foundational, Organizational

Main topics L3

- Riski = Likelihoodi x Impacti
- Risk: Possible Actions
  - Acceptance, mitigation, avoidance, transfer

$Likelihood_i = P\{A\ security\ hole_i\ is\ exploited\}.$

$= P\{hole_i\ present\}.\ P\{exploitation \mid hole_i\ present\}$

- Annual loss expectancy (ALE)

  ALE = SLE x ARO

  - Single loss expectancy SLE = AV x EF
    - AV value of the asset. EF exposure factor
  - ARO is Annualized rate of occurrence

Main topics L3

- COUNTERMEASURE_VALUE

  = (ALE_PREVIOUS – ALE_NOW) –COUNTERMEASURE_COST

- Return on Investment

  = COUNTERMEASURE_VALUE/COUNTERMEASURE_COST

L3

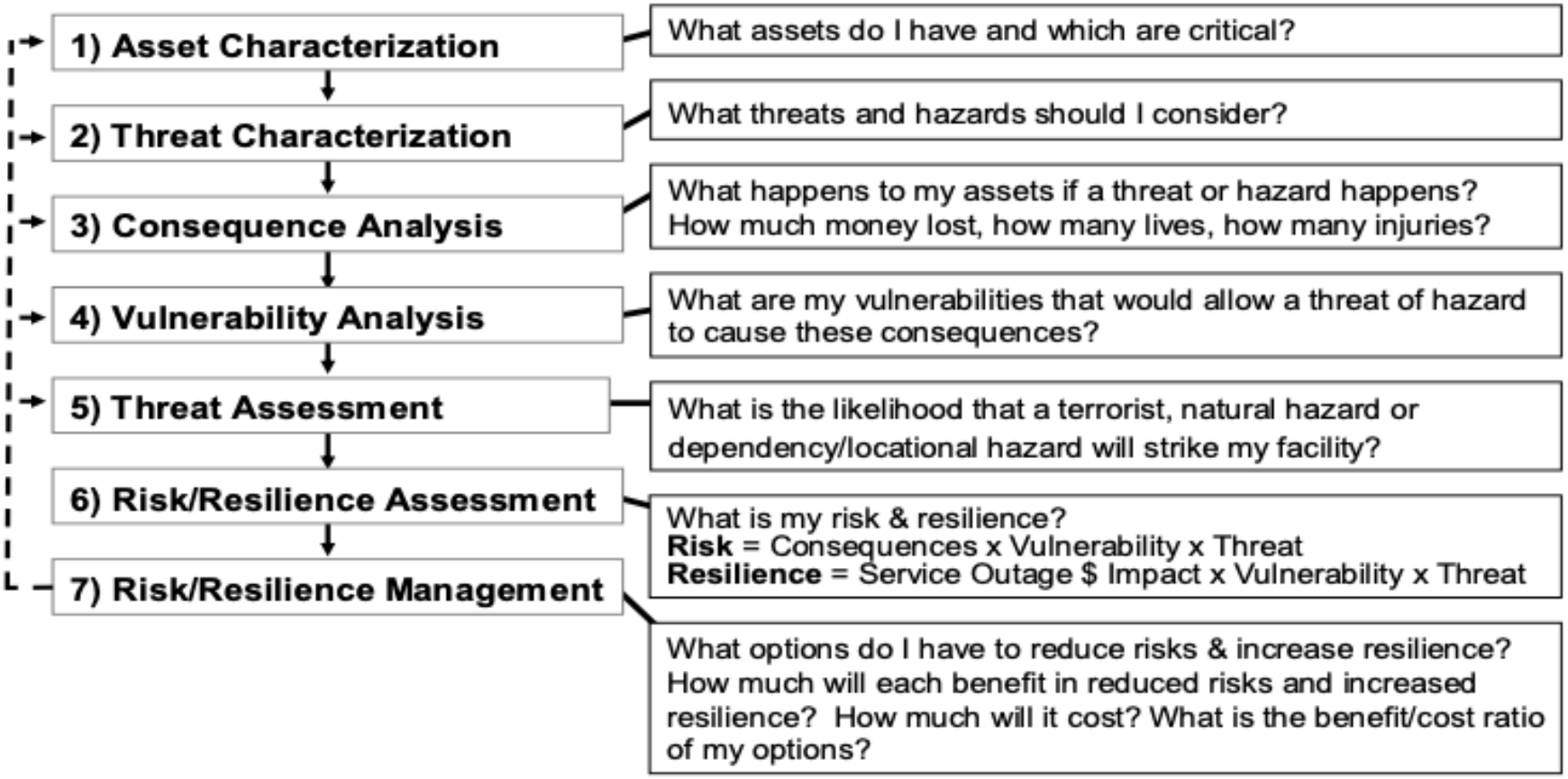
- Log(Risk) = Log(Likelihood) + Log(Impact)
  - Risk score = Likelihood score + Impact score

Likelihood	Consequences				
	Insignificant	Minor	Moderate	Major	Severe
Almost certain	M	H	H	E	E
Likely	M	M	H	H	E
Possible	L	M	M	H	E
Unlikely	L	M	M	M	H
Rare	L	L	M	M	H

Colorado State University

L4: RAMCAP

- RAMCAP Framework
  - Risk = Threat x Vulnerability x Consequence



L4: FAIR Framework

- **Factor Analysis for Information Risk**
- Risk = Probably Loss Magnitude x estimated Loss Event Frequency
  - Loss Event Frequency (LEF) = Threat Event Frequency x Vulnerability
    - Threat Event Frequency: table
  - Vulnerability (Vuln) = Threat Capability x lack of Control Strength
    - Threat Capability: table
    - Control Strength: table
- "Multiplication" achieved by using Matrices.

L4/5: **Risk management strategies**

- **Insurance: need**
- Law of large numbers
- **Actuarially fair Premium**: equal to expected claims

  = probability of illness in a year x average no. of utilization of services per year x unit cost of each utilization
- The **loss ratio** is the ratio of incurred losses and loss adjustment expenses to premiums earned.
- Asymmetric information
- Cyber Insurance: coverage, market, costs

Random Variables

- A random variable (r.v.) may take a specific random value at a time. For example
  - X is a random variable that is the height of a randomly chosen student
  - x is one specific value (say 5'9")
- A random variable is defined by its density function.
- A r.v. can be continuous or discrete

		continuous	*discrete*
Density function	$f(x)dx$	$P\{x \le X \le x+dx\}$	$p(x_i)$
"Cumulative distribution function" (cdf)	$F(x)$	$\int_{x\,\min}^{x} f(x)dx$	$\sum_{i=i\,\min}^{i\,\max} p(x_i)$
Expected value (mean)	$E(X)$	$\int_{x\,\min}^{x\,\max} x f(x)dx$	$\sum_{i=i\,\min}^{i\,\max} x_i p(x_i)$

L5: Probability

- Disjoint, independent, conditional prob.
- Confusion matrix
  - Sensitivity = TP/(TP+FN)
  - Specificity = TN/(FP+TN)
  - Precision = TP/(TP+FP)
  - Area under the ROC curve

		Actual	
		Disease +	Disease -
Predicted	Test +ve	TP	FP
	Test –ve	FN	TN

Colorado State University

Bayes' Rule

- Conditional probability

$$P\{A \mid B\} = \frac{P\{A \bigcap B\}}{P\{B\}} for P\{B\} > 0$$

P{A|B} is the probability of A, given we know B has happened.

- Bayes' Rule

$$P\{A \mid B\} = \frac{P\{B \mid A\}P\{A\}}{P\{B\}} for\, P\{B\} > 0$$

- Example: A drug test produces 99% true positive and 99% true negative results. 0.5% are drug users. If a person tests positive, what is the probability he is a drug user?

$$P\{DU \mid P\} = \frac{P\{P \mid DU\}P\{DU\}}{P\{P \mid DU\}P\{DU\} + P\{P \mid nDU)P\{nDU\}}$$

$$= \quad 33.3\%$$

L5: Distributions

- **Density and distribution functions**
  - Binomial, Poisson
  - Uniform
  - Normal, Lognormal
  - In Excel
  - Exponential, Weibull
- Variance & Covariance
- **Stochastic processes**
  - Markov process
  - Poisson process
  - Time between Two Events

L6: Intrusion detection Systems

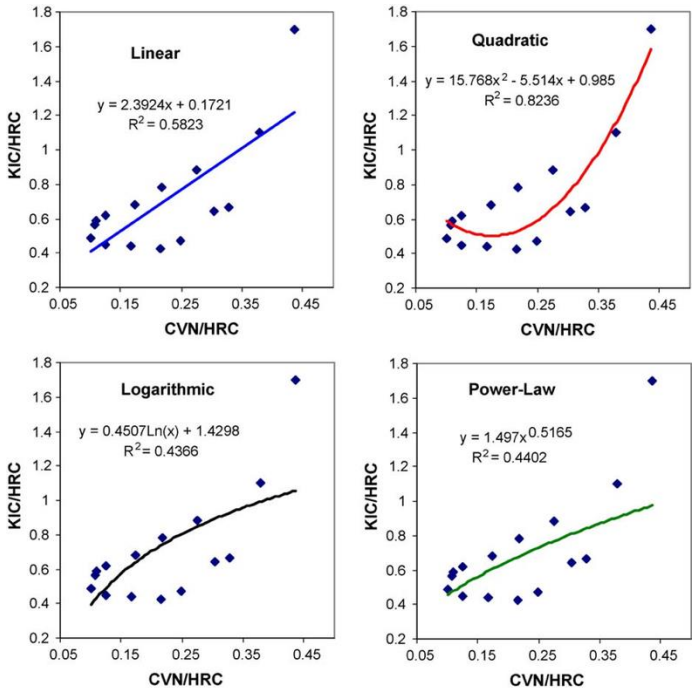
- IDS approaches
  - Anomaly detection: Is this the normal behavior?
  - Signature based detection: Does it match known bad behavior?
    - No clear diving line between intruder vs authorized user activity
- Rule-based heuristic
- Detections vs prevention (IPS in the path of information flow)
- Host-Based Intrusion Detection (HIDS) vs Network based (NIDS)

L9: Modeling and regression

- Models: what (derived/empirical) and why
- Curve fitting, tools
- Visualization
- Linear and non-linear: polynomial, exponential, power
- Log for linearization

Empirical models

- Look at data
- See if it resembles a function
  - Linear, quadratic, logarithmic, exponential..
  - Involving 1, 2 or more parameters
- See if it fits
  - If not try something more complex
- If it fits, see if an interpretation of the parameters is possible
  - Not necessary but will be good.



20

Vulnerabilities

- Defects vs vulnerabilities
- Types: software, system/physical, Personnel/procedures
- Components of Likelihood of Exploitation
  - Internal, external, interface
- Annual trends
- Vulnerability Lifecycle
- Vulnerability density and defect density
- Who discovers vulnerabilities?
- Classification of vulnerabilities

L10

- CVE numbering system
- Is it a vulnerability?
- Responsible Disclosure
  - Reward programs
  - Vulnerabilities for sale
- Data bases
- Vulnerability Lifecycle
  - Stochastic modeling
  - Zero-day attacks

L11/12

- Qualys “Laws of Vulnerabilities
  - Half-life, persistence, exploitation
- Modeling Vulnerability Discovery
- Using calendar time
  - AML model: derivation
  - Windows 98, NT
- Using equivalent effort
  - Market share
- Vulnerability density vs defect density

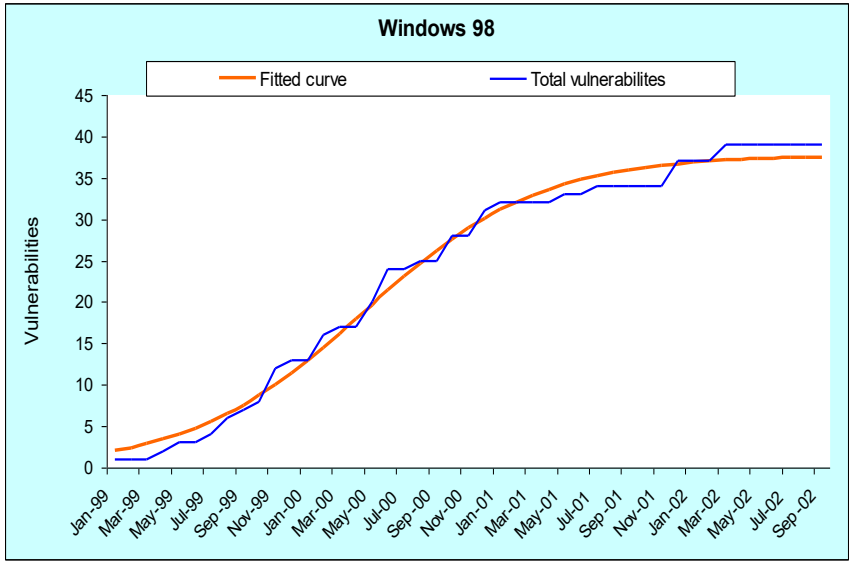
Time–vulnerability Discovery model

3 phase model S-shaped model.

- Phase 1:
  - Installed base –low.
- Phase 2:
  - Installed base–higher and growing/stable.
- Phase 3:
  - Installed base–dropping.

$$\frac{dy}{dt} = Ay(B - y)$$

$$y = \frac{B}{BCe^{-ABt} + 1}$$



Colorado State University

L12: Software Reliability Modeling

- Static metrics
- Exponential SRGM
- Usage –based vulnerability Discovery model
- Nonlinear regression using solver
- Factors Impacting Vulnerabilities
- Seasonality: testing for seasonality
  - Seasonal index analysis with test
  - Autocorrelation Function analysis

L12/13

- Is hacking legal?
- Dimensions and Approximations
- What you should question
- Software Reuse
  - Software Evolution
- Vulnerability Discovery & Evolution
  - Code Sharing & Vulnerabilities
- Multi-version Vulnerability Discovery
  - Humps vs extended linear
- Linear model
- Long Term Trends
  - Size evolution: Linus kernel

L15 Metrics

- Scales: Nominal, ordinal, Interval, Ratio
- Pendleton et al's Survey on Security metrics
- Vectors: entities, vulnerabilities, security state
- Attack-defense interactions in a computer
- Metrics Classification
  - *1. system vulnerabilities*,
  - *2. defense strength and*
  - *3. attack severity*,
  - *4. situation*

L15

Metrics for

- Measuring User (people) Vulnerabilities
- Measuring Interface-Induced Vulnerabilities
- Measuring Software Vulnerabilities
  - Evolution, lifetime, CVSS
- Measuring the Strength of Defenses
- Attack metrics
- Situation metrics
  - Incidents
  - Damage
  - investment

L16: CVSS

Objective: prioritize effort to address vulnerabilities

- Metrics: Components by levels translated into a numerical metric
- Scores: Computed score using a set of metrics as given by formulas
- 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

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.

30

L16

- Exploitability components:
  - Attack Vector (AV)
  - Attack Complexity (AC)
  - Privileges Required (PR)
  - User Interaction (UI)
  - Scope change
- Impact
  - Confidentiality Impact (C)
  - Integrity Impact (I)
  - Availability Impact (A)

CVSS system: How useful it is?

- What if they had multiplied exploitability and impact sub-scores instead of adding?
- Correlation among
  - CVSS Exploitability, Microsoft Exploitability metric, Presence of actual exploits: small or negative correlations
- Time to discovery? Some possible correlation
- Reward program? Significant correlation
- Time to patch: correlation
- Can metric/score determination be automated? Perhaps.
- VRP Cost effectiveness?

CVSS Base Score vs Vulnerability Rewards Programs

Chrome		
Vulnerabilities	Rewarded	Not rewarded
1012	584	428
VRP severity	Rewarded	Not rewarded
Critical & High	441	175
Medium	136	137
Low	7	116

The results results show that CVSS Base Score may have some correlation with the vulnerability reward program.

L16: Detection Probability

- **Detection probability** of a fault: if there are N distinct possible input vectors, and if a fault is detected by k of them, then its *detection probability* is k/N.
- A fault with detection probability (dp) 1/N would be hardest to test, since it is tested by only one specific test and none other.
- A fault which is detected by almost all vectors, would have a detection probability close to 1 and will be found with minimal texting effort. It is a low hanging fruit.



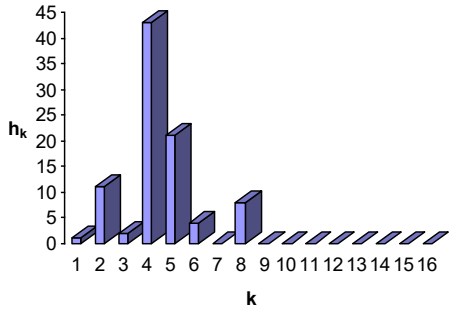
Detectability Profile of a unit under test

- The Detectability Profile of a unit under test describes how the defects are distributed relative to their detectability.
- Total M faults, total N possible input combinations. The set of faults can be partitioned into these subsets:
- $H = \{h_1, h_2, \ldots h_N\}$
- Where h_k is the number of faults detectable by exactly k inputs. The vector H describes the detectability profile.
  - h_1 is the number of faults that are hardest to find.
  - As testing and debugging continues, harder to find faults will tend to remain. Easy to find faults will get eliminated soon.

Ex: CECL Full adder Inputs=4 (N=16)

$H=(h_1,h_2,h_3,h_4,h_5,h_6,h_8)$

$=(1,11,2,43,21,4,8)$

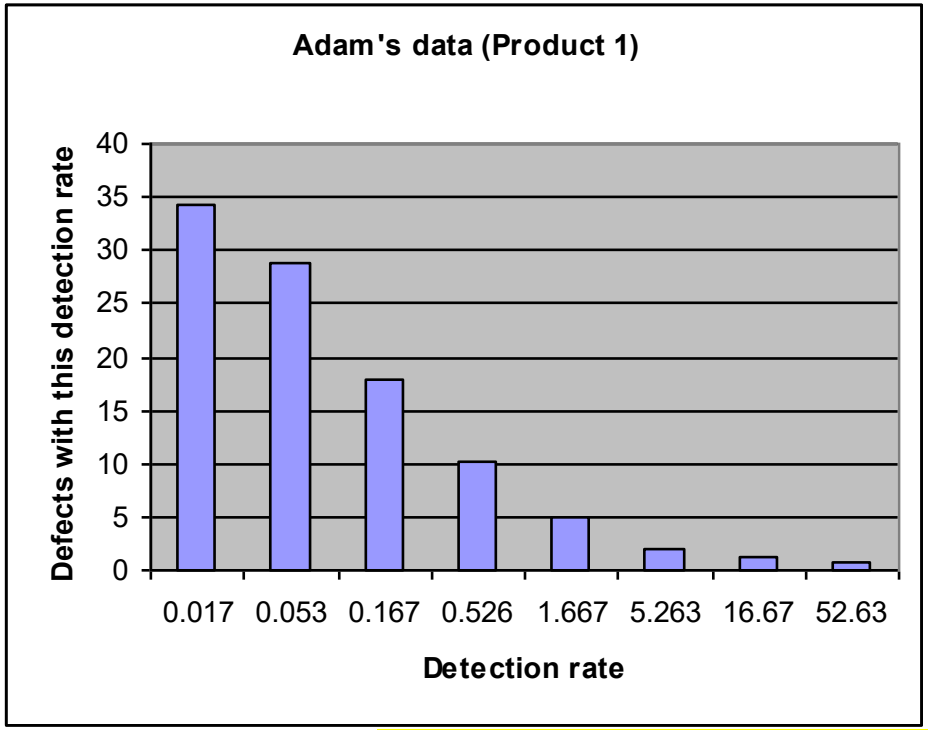


Y. K. Malaiya, S. Yang, "The Coverage Problem for Random Testing," IEEE Int Test Conf. 1984, pp. 237-245.

K. Wagnor, C. Chin, and E. McCluskey, "Pseudorandom testing". IEEE Trans. Computer, Mar. 1987, pp. 332—343.

Detectability Profile: software

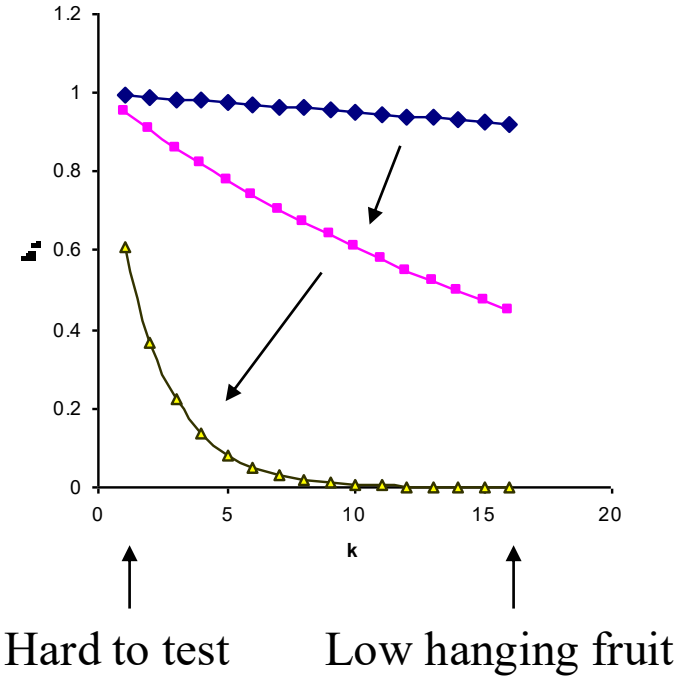
- Adam's Data for a large IBM software product. Note bugs with high detection rates are mostly gone.



Adams, IBM Journal of Research and Development, Jan. 1984

Detectability Profile of a unit under test

- Total M faults, total N possible input combinations. The set of faults can be partitioned into these subsets:
- $H = \{h_1, h_2, \ldots h_N\}$
- Where h_k is the number of faults detectable by exactly k inputs. The vector H describes the detectability profile.
  - h_1 is the number of faults that are hardest to find.
  - As testing and debugging continues, harder to find faults will tend to remain. Easy to find faults will get eliminated soon.



SRGMs

- **Exponential SRGM**: assumes bug finding rate $\lambda(t)$ is proportional to remaining bugs at time N(t).

$$\lambda(t) = -\frac{dN(t)}{dt} = \beta_1 N(t)$$

- Exponential defect finding model is

$$\lambda(t) = \beta_0 \beta_1 e^{-\beta_1 t}$$

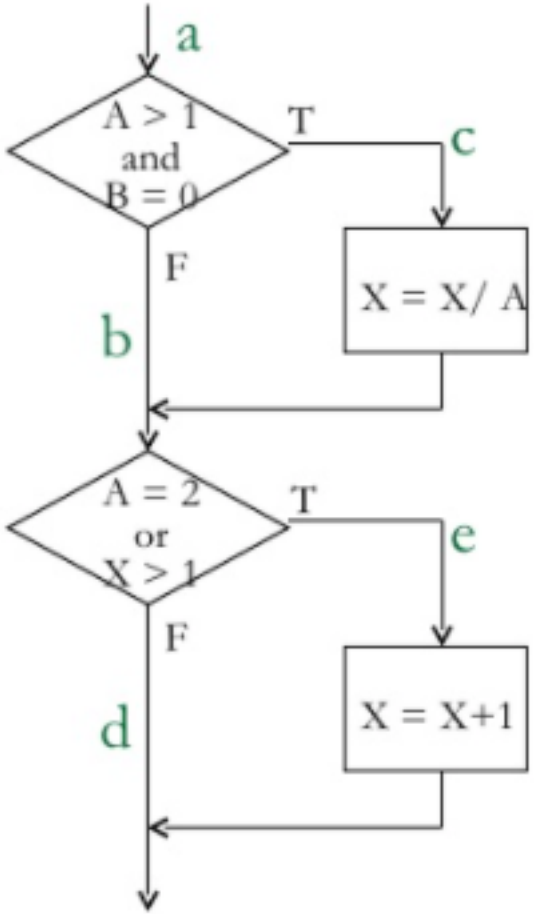
- β_0 represents the initial number of bugs.
- If the initial *defect density* is D(0), and the software size (measured in 1000 lines of code, i.e. KLOC) is S, then

$$\beta_0 = D(0) \times S$$

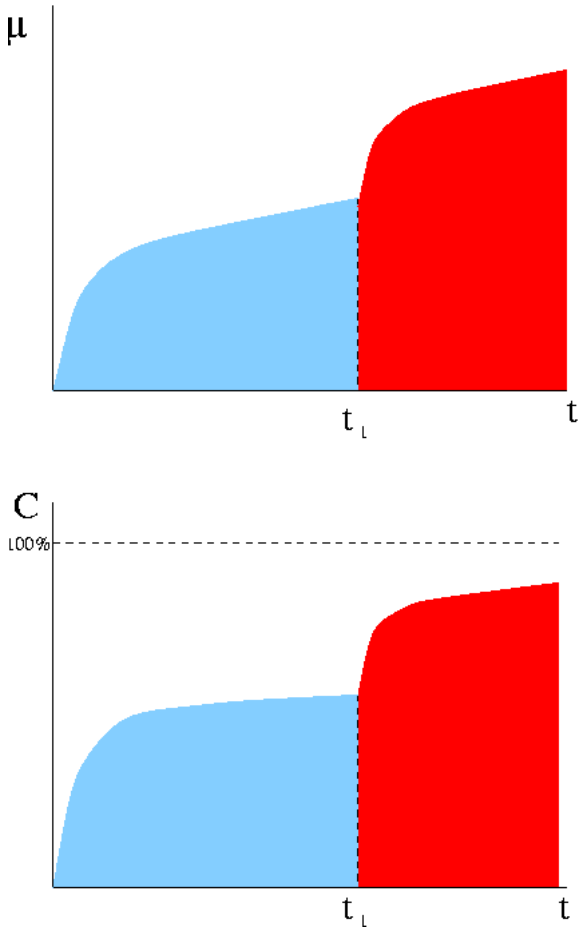
- The initial defect density is a function of the software development process and the degree of prior defect removal.

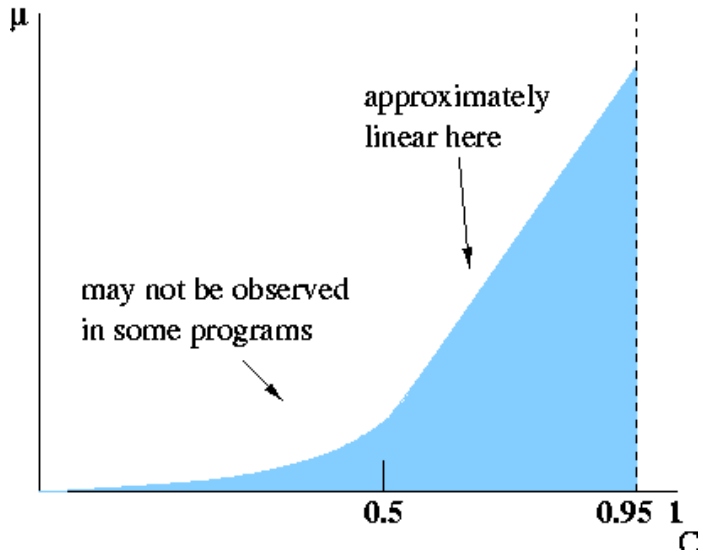
L17 Test Coverage Measures

- Test case A =2, B=0, X=4
  - Covers branches a, c, e
  - Covers all the statements
- Test case A =1, B=1, X=1
  - Covers branches a, b, d
- Two test cases for 100% branch coverage.



Modeling : Defects, Time, & Coverage





Malaiya, Li, Bieman, Karcich, Skibbe, 1994
Li, Malaiya, Denton, 1998

Log-Expo Coverage Model

- Eliminating t and rearranging,

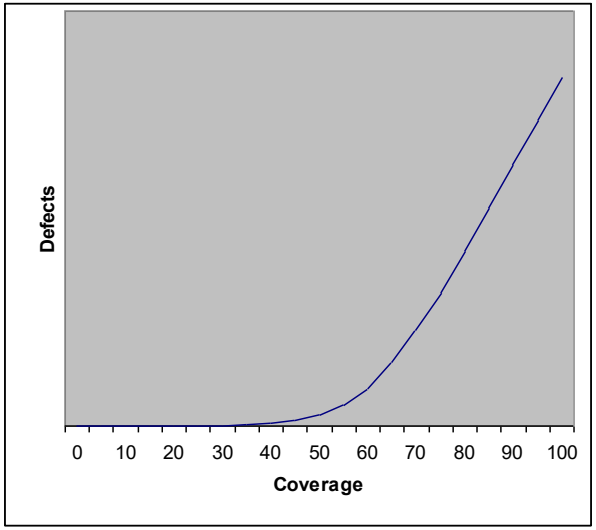
$$C^0 = a_0^i \ln[1 + a_1^i(\exp(a_2^i C^i) - 1)], \quad C^0 \le 1$$

where C^0 : defect coverage, C^i : test coverage

a_0^i, a_1^i, a_2^i : parameters ; i : branch cov, p - use cov etc.

- For "large" Ci, we can approximate

$$C^0 = -A^i + B^i C^i$$

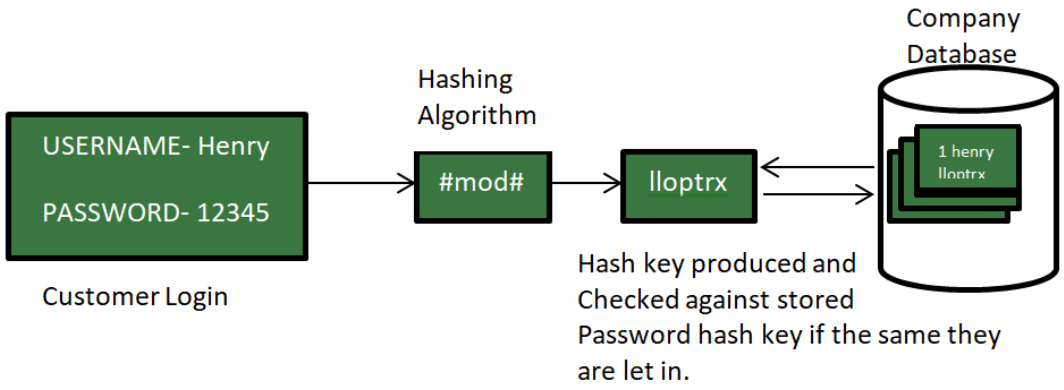


L 18: Problem: Password guessing

- If your keyboard has R= 95 unique characters, 12-character password, then L = 12.
- 95^{12}= 540,360,087,662,636,962,890,625 passwords

Entropy = $\log_2(R^L)$ = 78.9 bits assuming passwords are created randomly

- Non-randomness makes password guessing easier.
- Measures of password strength proposed and used



Colorado State University

Attacking Salted Passwords



hash()

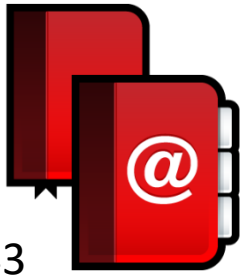
List of possible password hashes

No matches



hashed_ and_salted_ password.txt

cbw	a8
sandi	0X
amislove	hz
bob	K@



hash('0X' + word)

List of possible password hashes w/ salt *0X*

c

sandi YYYY

Cloud vs Local Password Storage

Password vaults is stored either

- Locally on the device (or on premise), may be synched with other devices
  - Will be lost with the device
  - May be hacked by malware
  - More secure?
- On the cloud, log in from any device
  - Recoverable if device lost, can be changed
  - May be stolen with a cloud breach
  - May be encrypted/decrypted at the device level (zero-knowledge architecture)

Master Password

Needed to login into the Password Vault

- Passwords in the vault may be lost if forgotten
- Might be possible to reset the Master Password if the recovery code is available.
- Guidelines:
  - Is unique to your password manager
  - Has not been used in other places online
  - Something you can remember, or write down to remember and put in a secure place
  - Strong: long, random, hard to guess
  - Enable two factor authentication
- Not needed to change frequently, unless evidence of a compromise

Password Generation

- Password Managers generate strong passwords with desirable attributes
  - Random combination of characters with symbols. It will be stored and applied thus no need to store
  - 16 characters better than 12
  - `kdH/3yBqXa*4n7r}`
  - `randomescapeemountoptimal`
  - Can also generate answers to questions

    MOTHER'S MAIDEN NAME? `Smith1980`

Password manager security

Lastpass breach August 2022:

- limited to the service's development environment that isn't connected to customers' data and encrypted vaults.
- LastPass has no access to users' master passwords, which are needed to decrypt their vaults.

Lastpass breach on June 14, 2015

- user email addresses, authentication hashes, password reminders and server per user salts were compromised
- no evidence that user vault data (a user's stored passwords) was taken or that accounts were accessed

https://www.engadget.com/lastpass-hacked-no-user-data-was-compromised-064640557.html

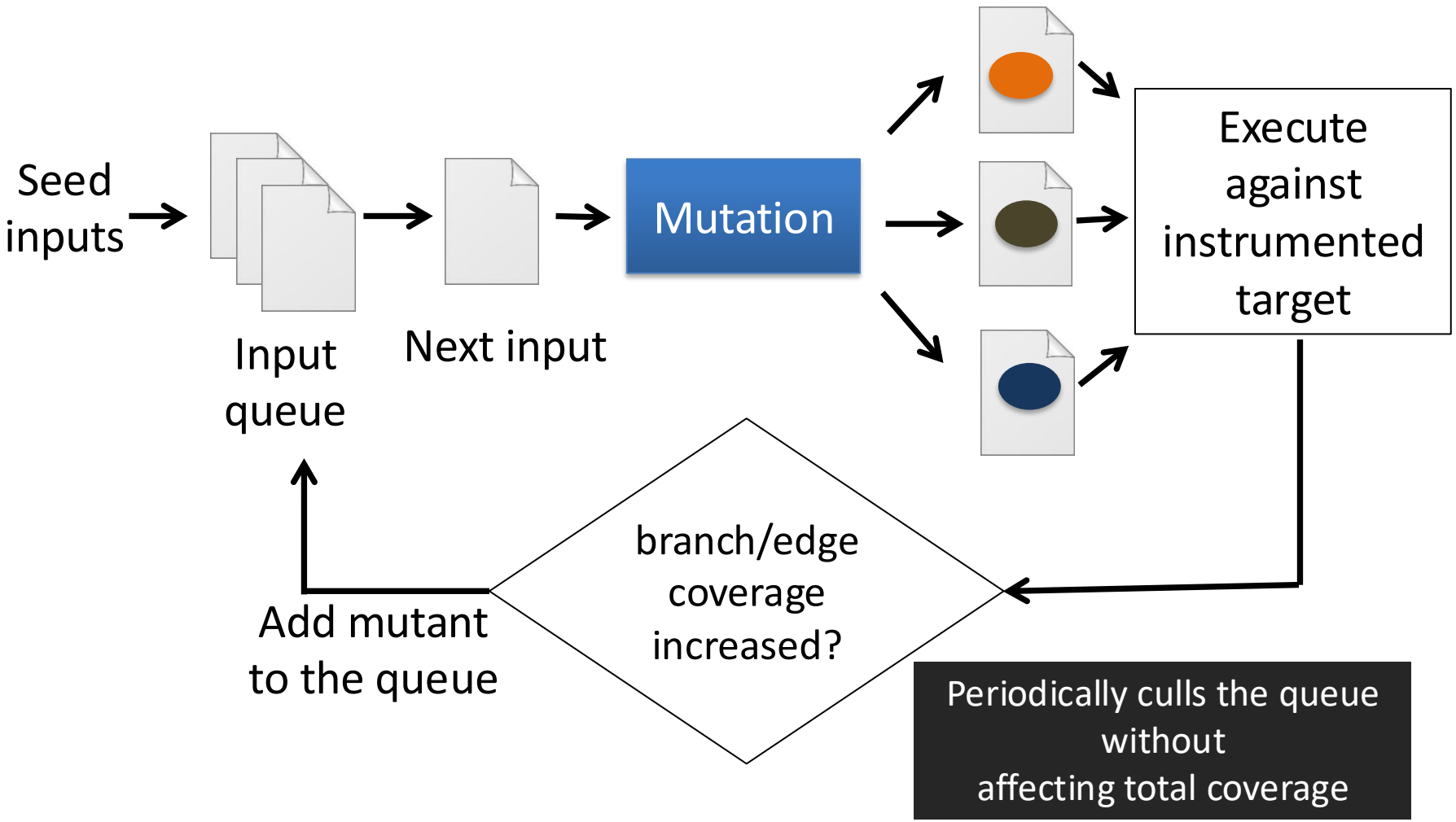
L 19: Fuzzing PDF Reader

- Download 100s of random PDF files
- Mutate content in the PDF file:
  - flip bits
  - increase size of integers or strings
  - remove data
- Limited by the functionality that the existing files happened to use — unlikely to hit less commonly tested code paths





American Fuzzy Lop (AFL)

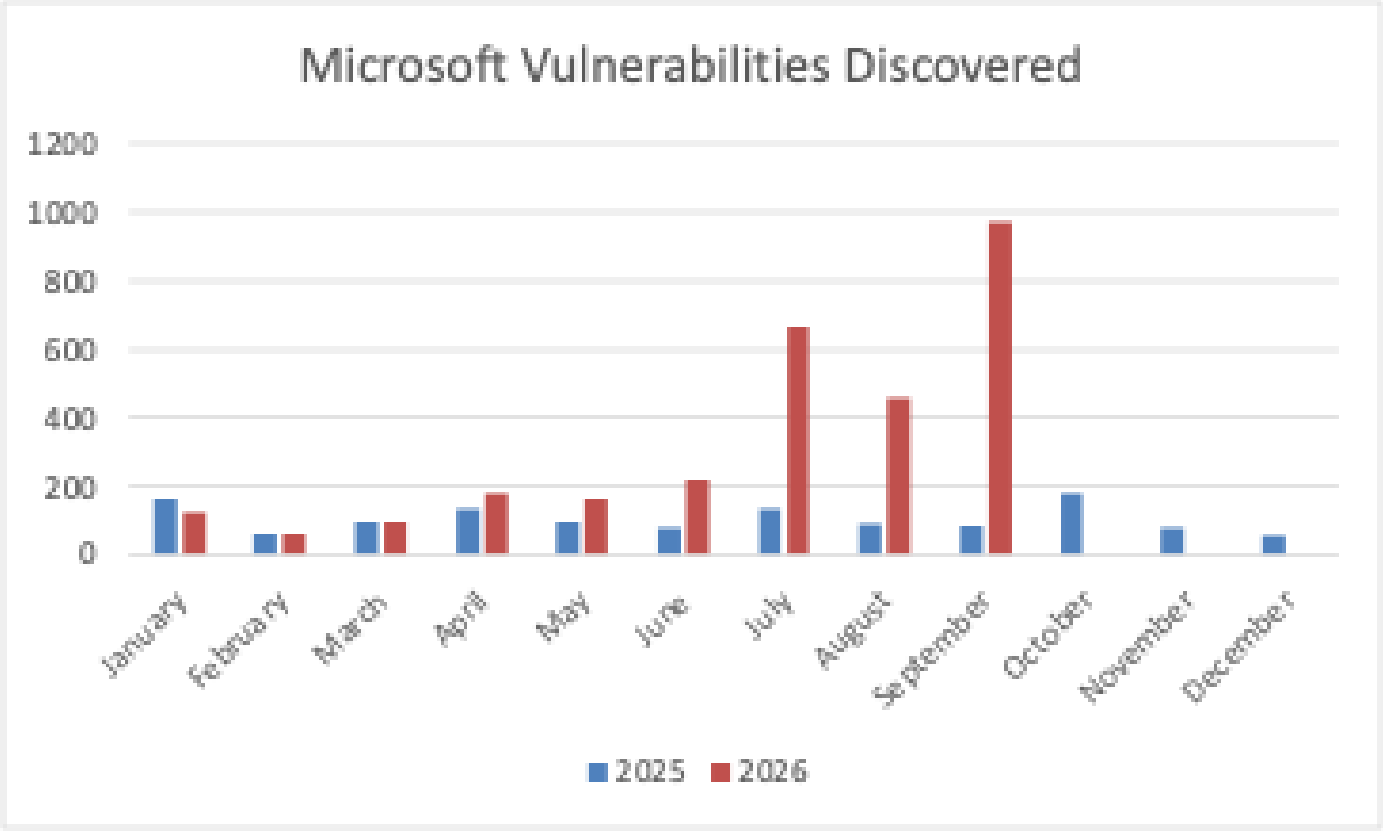


New Cyber security Crisis

- **The "PHP2Shell" Discovery**
- Researchers at OpenAI, Wiz, and Calif used frontier AI models (GPT-5.6 Sol and Gemini 4 Argon) to discover a catastrophic series of vulnerabilities hidden in PHP code for over a decade.
- CVE-2026-109618 and CVE-2026-109619 both with CVSS of 9.2. Allow remote code execution.
- The flaw affects roughly 70% of the internet and has forced many websites to go offline today to patch it.

AI for Vulnerability Discovery

- Use of AI has dramatically speeded up vulnerability discovery.



Colorado State University

AI-driven remediation

Patching Phase	Traditional / Without AI	Autonomous / With AI
Typical Enterprise Cycle	30 to 90 days (Governance SLAs typically give 30 days for critical, 60 for high)	6 to 13 days (Autonomous endpoint patching tools)
Average Real-World MTTP	43 days (up from 32 days in 2024 due to increased complexity)	Under 3 days (When integrating AI scanners with virtual patching/WAAP solutions)
Testing & Verification	8 to 14 days of traditional siloed testing	Automated validation and immediate deployment orchestration

Defensive asymmetry

AI has broken traditional, patch-centric vulnerability management.

- **Centralized Bottlenecks: The National Vulnerability Database (NVD) Crisis** NVD, has been overwhelmed by record-breaking CVE growth driven heavily by AI-assisted submissions. Of the more than 53,000 CVEs published by late 2026, nearly 79% sit in an "Awaiting Analysis" backlog. Roughly 80% of new CVEs are now labeled "Not Scheduled.
- **The Open-Source & Maintainer Chokepoint** Anthropic's Project Glasswing utilized AI to surface 26,153 vulnerabilities. Out of more than 2,000 officially disclosed to maintainers, fewer than 1% have actually been patched.
- **Enterprise Triage Overwhelm** :
  - **Signal vs. Noise:** Poorly implemented AI code scanners routinely drop massive lists of unvalidated flaws into development backlogs, sometimes yielding true-positive rates below 7%.
  - **Fixing vs. Breaking:** more than 53% of AI-generated patches either failed to fix the core vulnerability, broke application logic entirely, or introduced brand-new security flaws.

Quantitative Security

Colorado State University

Yashwant K Malaiya

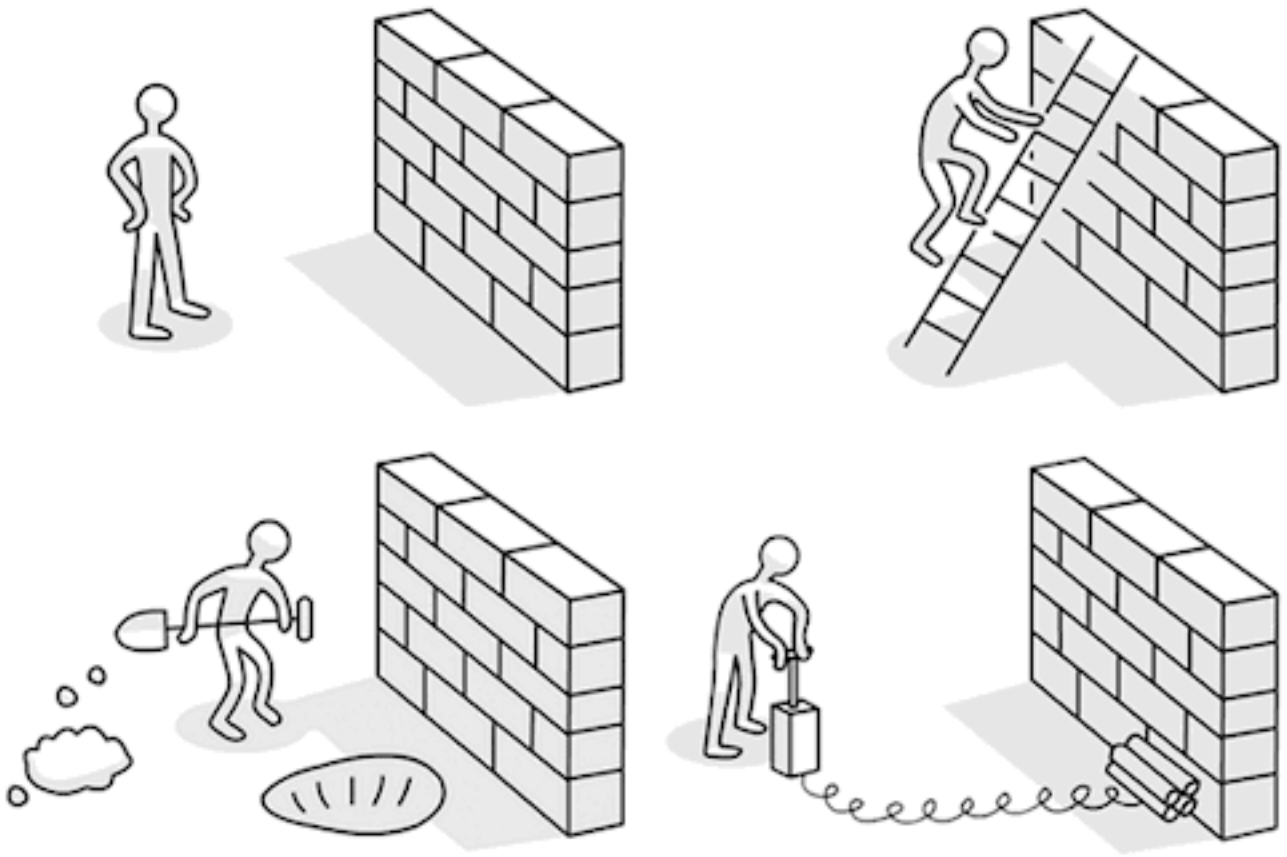
CS 559

Penetration Testing



CSU Cybersecurity Center
Computer Science Dept

Penetration Testing



https://www.cobalt.io/blog/how-to-choose-a-pentest-service

Colorado State University