Security Vulnerabilities: Risks from Discovery to Exploitation

Yashwant K. Malaiya
Colorado State University

Outline

- Vulnerabilities and the society
- Risk as Likelihood x Impact product
- Conditional components of Likelihood
  - Internal and External
- Vulnerability discovery in lifecycle
- CVSS as a risk measure
- Vulnerability markets
- Measuring impact

Magnitude of Security Risks

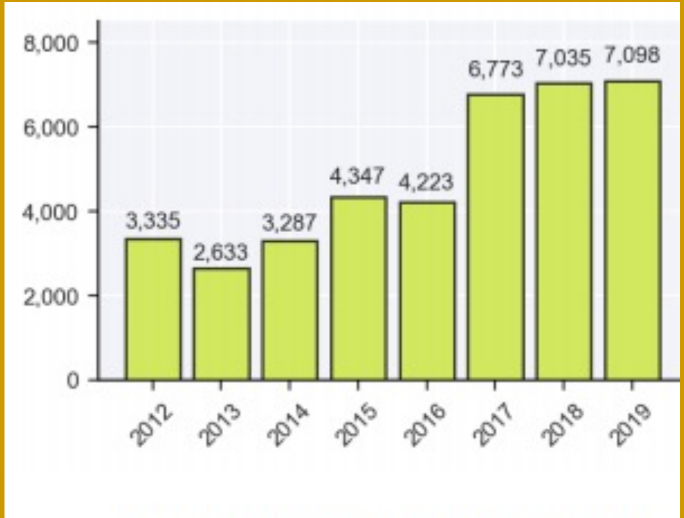


Figure 1: Number of breaches reported each year

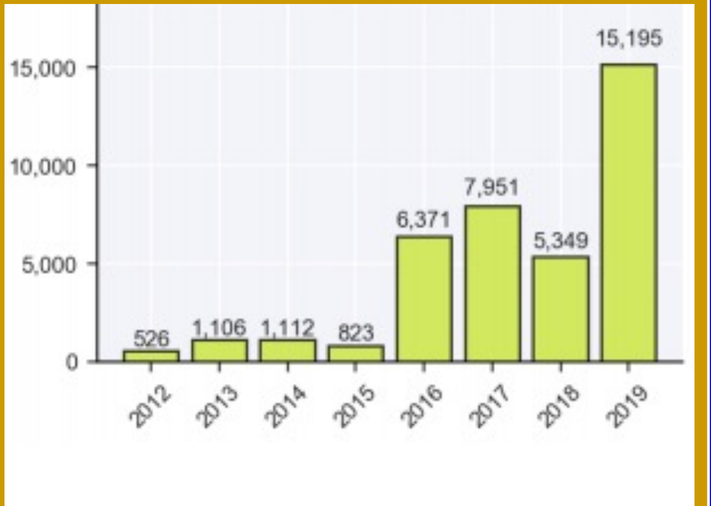


Figure 2: Number of records lost (in millions) each year

2019 Year End Data Breach QuickView Report

Exposed Records by Country

Ranking	# of Breaches	Country	Total Exposed	Records Average Records per Breach	Median Number of Records	Percentage of Exposed Records
1	27	China	3,822,021,911	141,556,367	11,748,417	52.01%
2	2330	UnitedStates	2,317,065,126	994,449	1,458	31.53%
3	16	Netherlands	711,794,171	44,487,136	4,021	9.69%
4	78	India	301,422,538	3,864,392	216	4.10%
5	11	SouthAfrica	67,023,831	6,093,076	6,700,000	0.91%
6	3	Philippines	55,245,020	13,811,255	-	0.75%
7	6	Argentina	28,741,292	4,790,215	2,516	0.39%
8	12	Republic Of Korea	17,372,292	1,447,691	1,000,000	0.24%
9	11	Israel	14,001,285	1,272,844	131	0.19%
10	1	Bermuda	13,400,000	13,400,000	-	0.18%

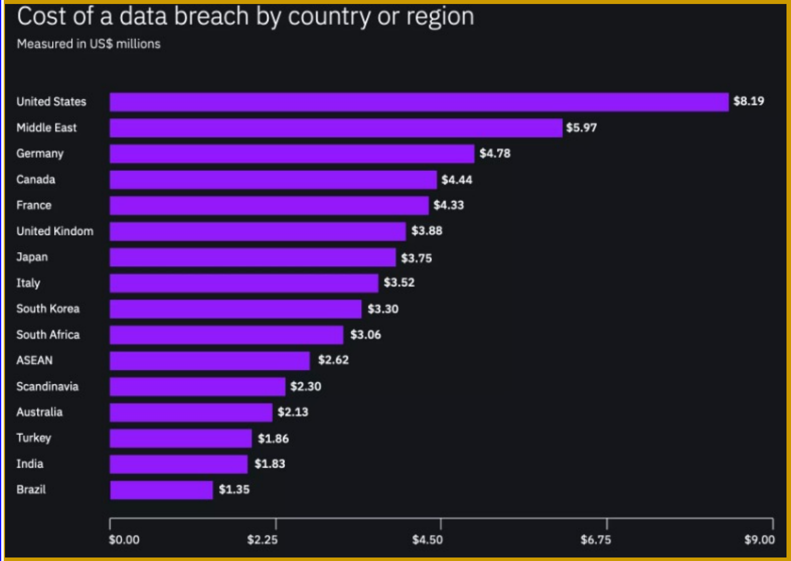
Data Breach QuickView Report: 2017 Data Breach Trends – Year In Review

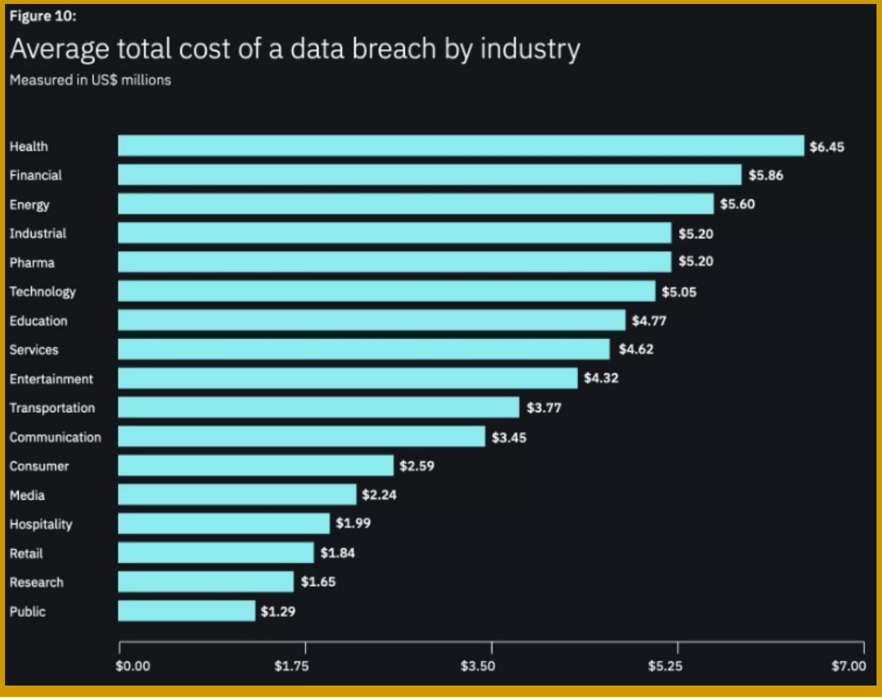
Cost of security Incidents

Business Size	BusinessSize in $	Million $/incident
Small	<100 M	0.41
Medium	100 M to 1 B	1.3
Large	>1 B	5.9
National Economy		? (**Gingrich IP** $360B) '16
National Security		? (Stuxnet type attack $1T) '15
National Democracy		? (Clinton campaign: 1.2B, DNC) '16

Source: Global State of Information Security Survey 2015 (and others)

Cost of security Incidents





What's the Cost of a Data Breach in 2019? Chris Brook July 30, 2019

Objectives and Challenges

Coming up with

- a standard and comprehensive terminology
- and then develop models for risk components

Challenges

- There exist numerous measures of risk, most of them partial measures based on limited perspectives (network accessibility, attack surface, CVSS etc)
- Different measures of “cost”
- Data does not come from controlled experiments
  - Real life data
  - Limited data from diverse sources collected without mutual coordination
  - Need to reconcile apparent mismatch/contradictions

Extent of the problem: IoT

Risk as a composite measure

Formal definition:

- Risk due to an adverse event e_i

  $Risk_i = Likelihood_i \times Impact_i$

- Sometimes likelihood is split in two factors

  $Likelihood_i = P\{hole_i\ present\}.$

  $P\{exploitation|hole_i\ present\}$

- A specific time-frame, perhaps a year, is presumed for the likelihood.

In classical risk literature, the internal component of Likelihood is termed “Vulnerability” and external “Threat”. Both are probabilities. There the term “vulnerability” does not mean a security bug, as in computer security.

Likelihood & Impact scales

- Quantitative or descriptive levels
  - Number of levels may depend on resolution achievable
- Scale: Logarithmic, Linear or combined
- Risk = Likelihood x Impact
  - Log(Risk) = Log(Likelihood) + Log(Impact)
- If “Score” is proportional to Log value
  - Risk score = Likelihood score + Impact score
  - Adding scores valid if scores represent logarithmic values.
  - Example:
    - Likelihood = 10%, impact = $100,000 ⇒ **Risk = $10,000**
    - Scores: Log(0.10) = -1, log (100000) = 5 ⇒ **Risk score = 4**

Risk Matrix

- Likelihood and Impact divided into levels
  - Each level quantitatively/qualitatively defined
- Cells marked by the overall risk
  - Low, Medium, High, Extreme etc.
- Equal risk regions along the diagonal, valid provided score scales are logarithmic.

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

LIKELIHOOD (probability) How likely is the event to occur at some time in the (Linear Scale time specific matrix)	CONSEQUENCES				
	What is the Severity of injuries /potential damages / financial impacts (if the risk event actually occurs)? *(Logarithmic Scale, property industry specific matrix)*				
	Insignificant	Minor	Moderate	Major	Catastrophic
	No Injuries First Aid No Envir Damage << $1,000 Damage	Some First Aid required Low Envir Damage << $10,000 Damage	External Medical Medium Envir Damage <<$100,000 Damage	Extensive injuries High Envir Damage <<$1,000,000 Damage	Death or Major Injuries Toxic Envir Damage >>$1,000,000 Damage
Almost certain - expected in normal circumstances (100%)	MODERATE RISK	HIGH RISK	HIGH RISK	CRITICAL RISK	CRITICAL RISK
Likely – probably occur in most circumstances (10%)	MODERATE RISK	MODERATE RISK	HIGH RISK	HIGH RISK	CRITICAL RISK
Possible – might occur at some time. (1%)	LOW RISK	MODERATE RISK	HIGH RISK	HIGH RISK	CRITICAL RISK
Unlikely – could occur at some future time (0.1%)	LOW RISK	MODERATE RISK	MODERATE RISK	HIGH RISK	HIGH RISK
Rare - Only in exceptional circumstances 0.01%)	LOW RISK	LOW RISK	MODERATE RISK	MODERATE RISK	HIGH RISK

Security Holes: Types

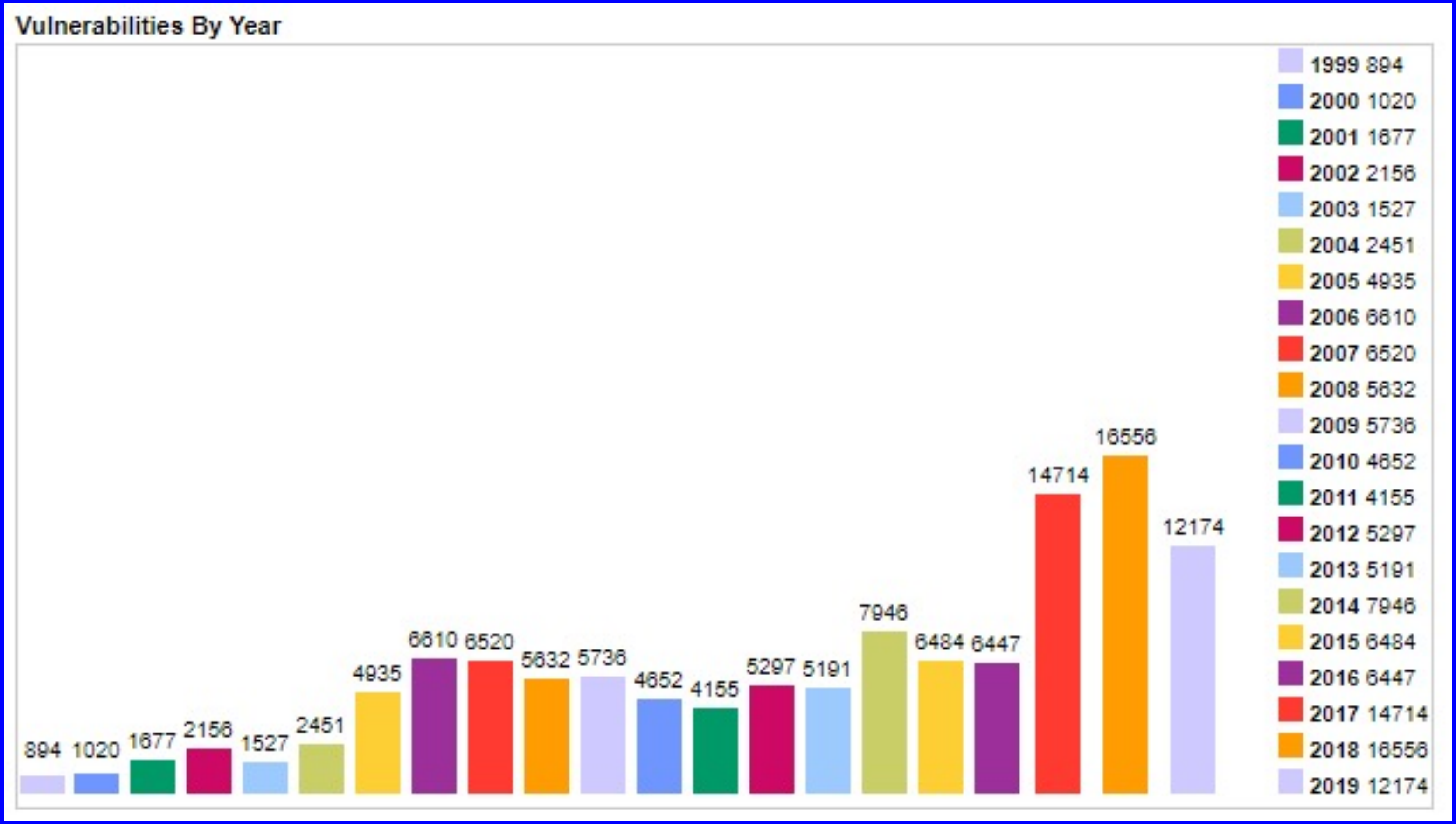
- Software holes: Vulnerabilities
  - CVSS scores involving *exploitability* and *impact* is a type of risk measure.
- System/physical holes
- Personnel/Procedural holes:
  - e.g. Phishing
- Exploitation may involve multiple holes, perhaps of different types
- Classify them:
  - Target 2013 breach: credentials stolen from a HVAC contractor
  - Equifax 2017 breach: vulnerability patch not applied

Components of Likelihood of Exploitation

- Internal
  - Presence of a vulnerability (Vulnerability Discovery*)
  - Vulnerability not patched
- External
  - Attacker's motivation level
  - Technical capabilities, exploit availability*
  - Network access to vulnerable system
- Interface
  - Attack surface* of vulnerable system
  - Reachability* of vulnerability

We have published research related to these. See publications 2005-2016.

Vulnerabilities Trend



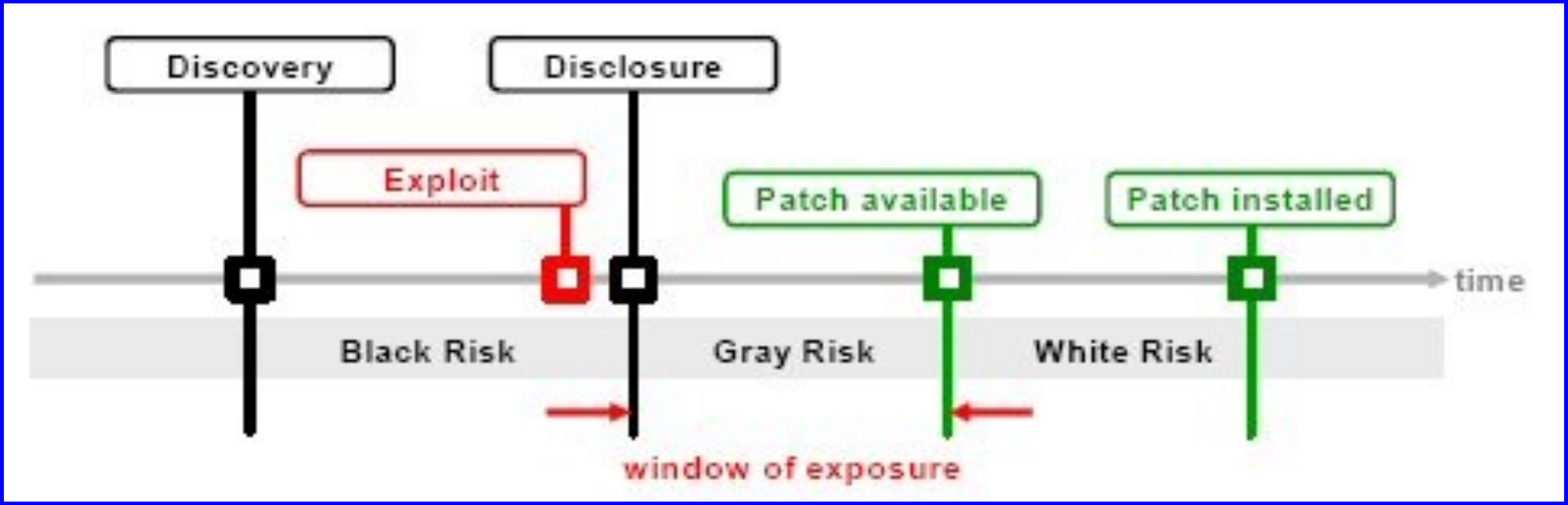
https://www.cvedetails.com/browse-by-date.php

Vulnerability Data-bases

- NIST NVD (National Vulnerability Database) U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP)
- CVE Details web interface to CVE vulnerability data. Browse for vendors, products and versions and view cve entries, vulnerabilities, related to them
- VulnDB Identified and cataloged over 73,969 vulnerabilities not found in CVE/NVD
- ExploitDB CVE compliant archive of public exploits and corresponding vulnerable software, developed for use by penetration testers and vulnerability researchers

Vulnerability Lifecycle

Vulnerabilities: “defect which enables an attacker to bypass security measures” [Schultz et al]



Exploit code (“exploit”) : usually available after disclosure

Modeling Vulnerability Discovery

- Quantitative Vulnerability Assessment Alhazmi 2004-2008
- Discovery in Multi-Version Software Kim 2006,2007
- Seasonality in Vulnerability Discovery Joh 2008,2009

Motivation

- For defects: Reliability modeling and SRGMs have been around for decades.
- Assuming that vulnerabilities are special faults will lead us to this question:
  - To what degree reliability terms and models are applicable to vulnerabilities and security? [Littlewood et al].
  - The need for quantitative measurements and estimation is becoming more crucial.

Goal: Modeling Vulnerability Discovery

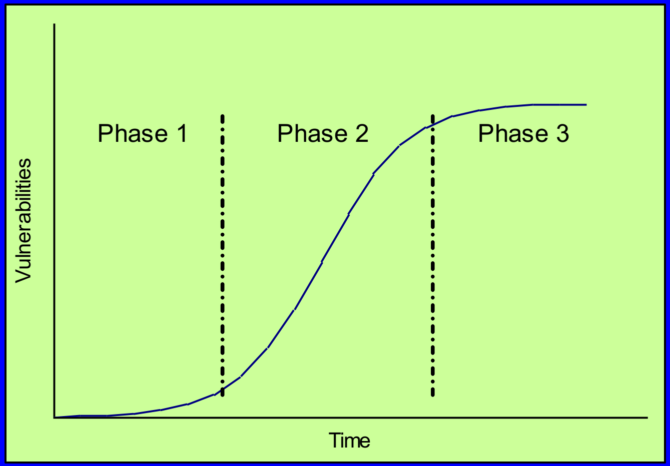
- Developing a quantitative model to estimate vulnerability discovery.
    - Using *calendar time*.
    - Using *equivalent effort*.
- Validate these measurements and models.
    - Testing the models using available data
- Identify security Assessment metrics
    - *Vulnerability density*
    - *Vulnerability to Total defect ratio*

Time – vulnerability discovery model

- What factors impact the discovery process?
  - The changing environment
    - The share of installed base.
    - Global internet users.
  - Discovery effort
    - Discoverers: Developer, White hats or black hats.
    - Discovery effort is proportional to the installed base over time.
    - Vulnerability finders' reward: greater rewards, higher motivation.
  - Security level desired for the system
    - Server or client

Time – vulnerability discovery model

- Each vulnerability is recorded.
  - Available [NVD, vender etc].
  - Needs compilation and filtering.
- Data show three phases for an OS.
- Alhazmi-Malaiya Logistic model (AML)
  - Assumptions:
    - The discovery is driven by the rewards factor.
    - Influenced by the change of market share.



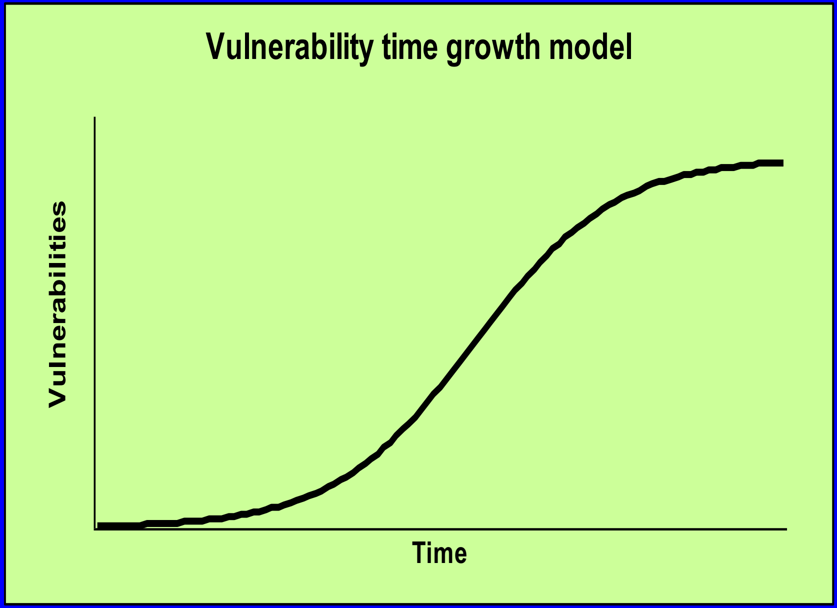
Time–vulnerability Discovery model

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

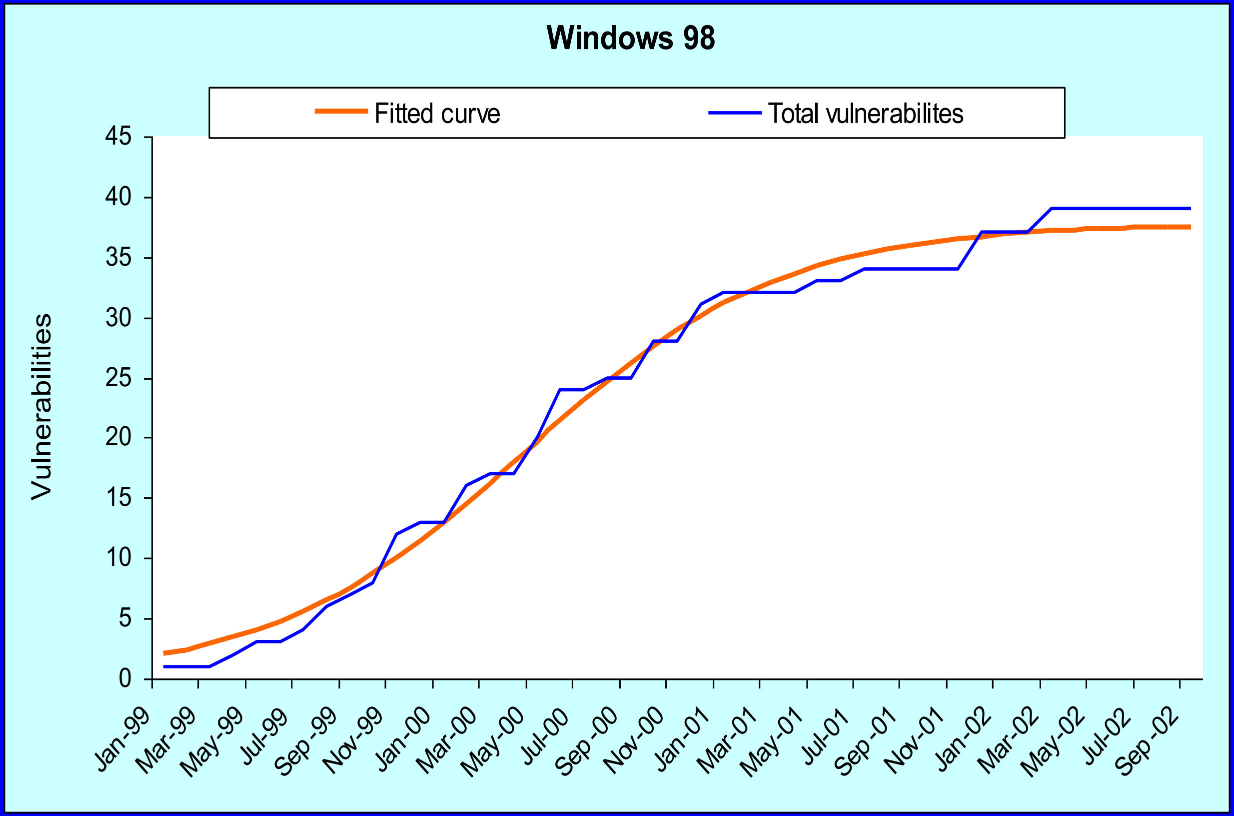
$$y = \frac{B}{BCe^{-ABt} + 1}$$

3 phase model S-shaped model.

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

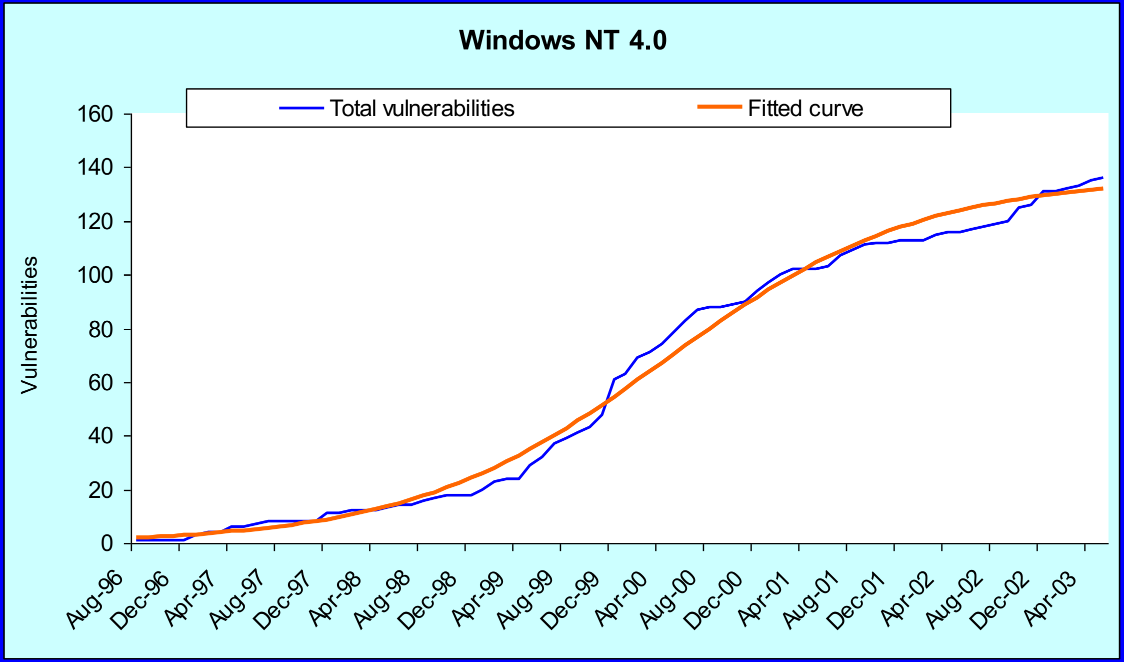


Time–based model: Windows 98



	Windows 98
A	0.004873
B	37.7328
C	0.5543
X^2	7.365
$X^2_{critial}$	60.481
P-value	1- $7.6\text{x}10^{-11}$

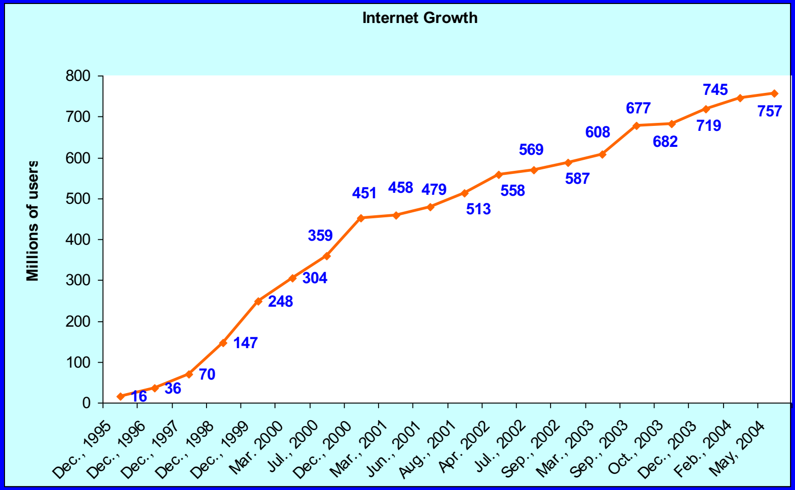
Time–based model: Windows NT 4.0

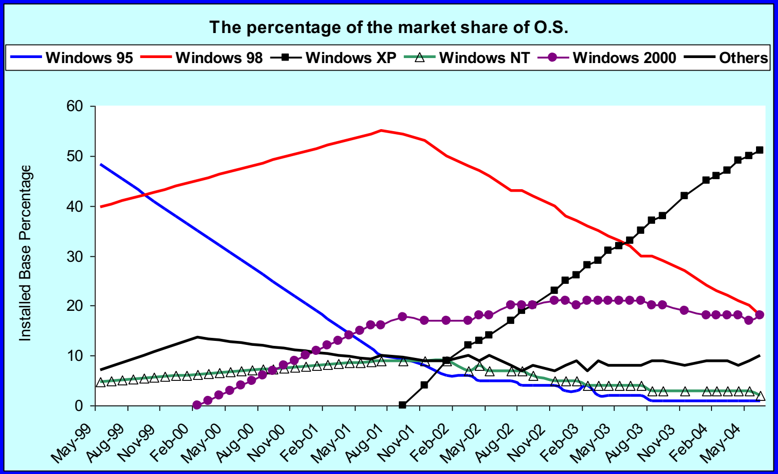


	Windows NT 4.0
A	0.000692
B	136
C	0.52288
χ^2	35.584
$\chi^2_{critial}$	103.01
P-value	0.9999973

Usage –vulnerability Discovery model

- The data:
  - The global internet population.
  - The market share of the system during a period of time.
- *Equivalent effort*
  - The real environment performs an intensive testing.
  - Malicious activities is relevant to overall activities.
  - Defined as $E = \sum_{i=0}^{n} (U_i \times P_i)$





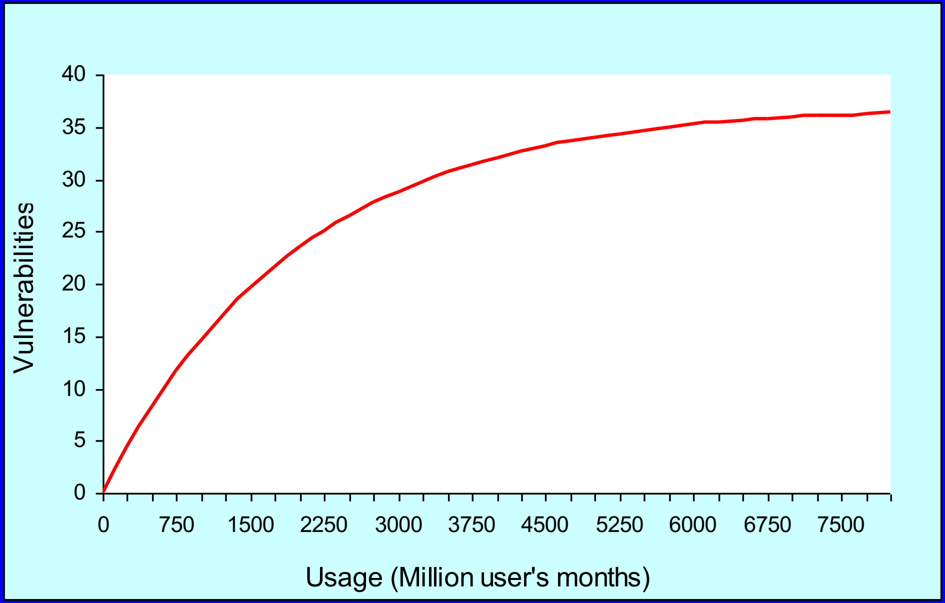
O. H. Alhazmi and Y. K. Malaiya, "Quantitative Vulnerability Assessment of Systems Software," Proc. Ann. IEEE Reliability and Maintainability Symp., 2005, pp. 615-620

Usage –vulnerability Discovery model

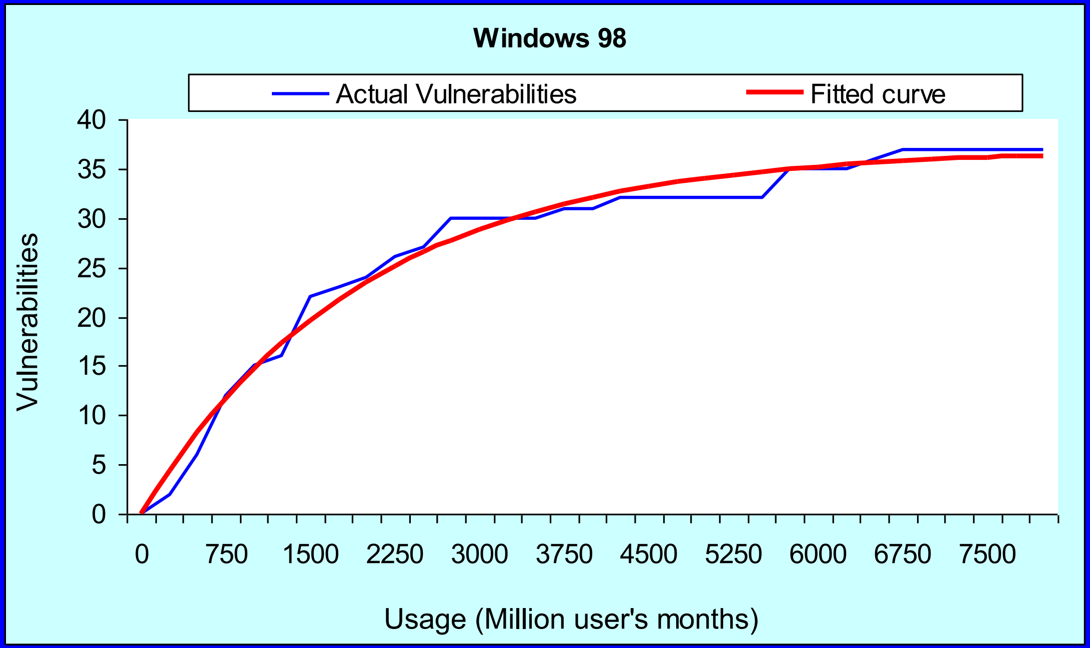
- The model:

$$y = B(1 - e^{-E\lambda_{vu}})$$

- Exponential growth with effort.
- The basic reliability model [Musa].
- Time is eliminated.

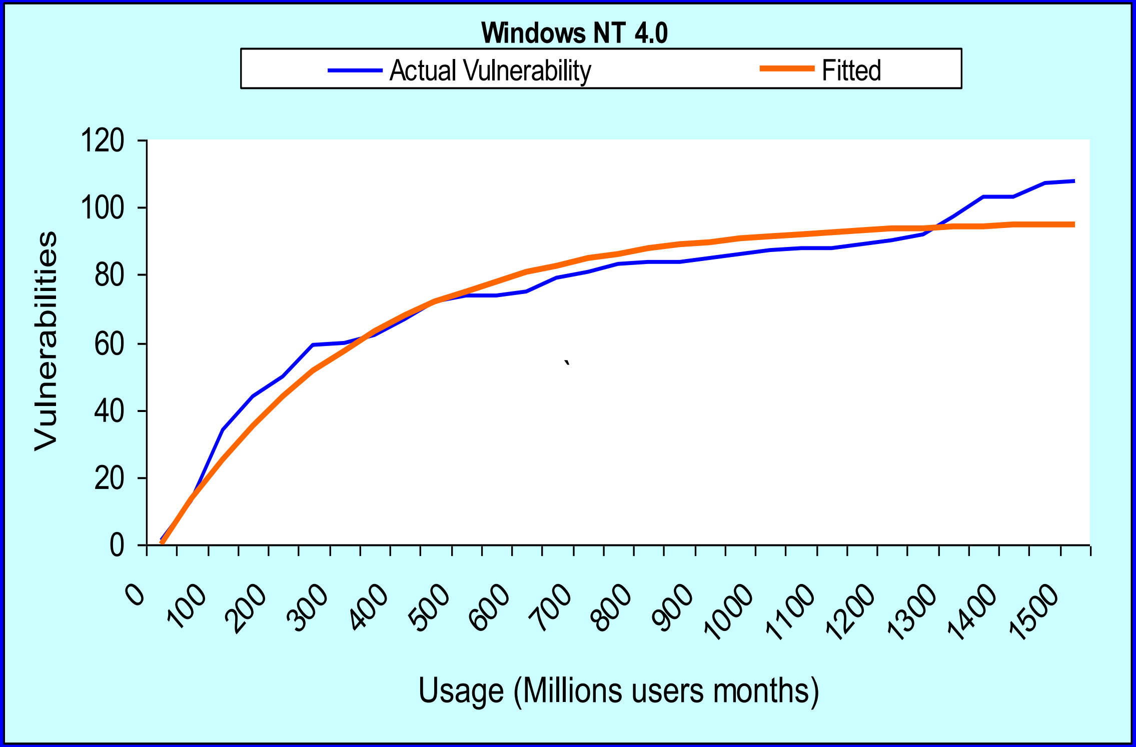


Effort-based model: Windows 98



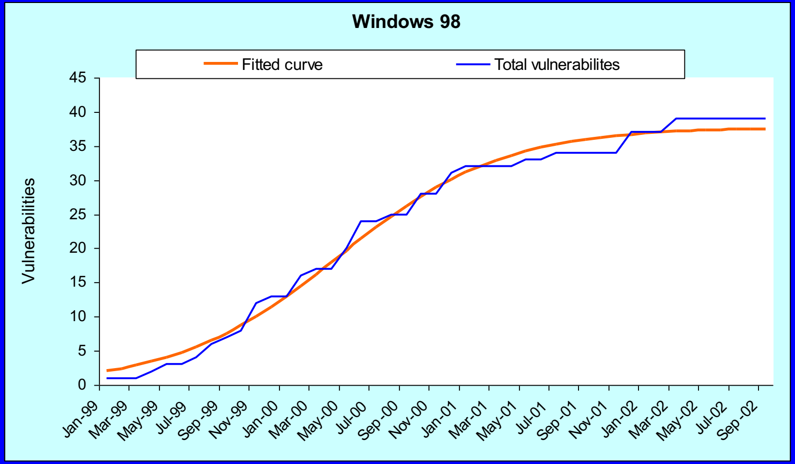
	Windows 98
B	37
λ_{vu}	0.000505
X^2	3.510
$X^2_{critial}$	44.9853
P-value	$1- 3.3\text{x}10^{-11}$

Effort-based model: Windows NT 4.0



	Win NT 4.0
B	108
λ_{vu}	0.003061
X^2	15.05
$X^2_{critial}$	42.5569
P-value	0.985

Discussion



- Excellent fit for Windows 98 and NT 4.0.
- Model fits data for all OSs examined.
- Deviation from the model caused by overlap:
  - Windows 98 and Windows XP
  - Windows NT 4.0 and Windows 2000
- Vulnerabilities in shared code may be detected in the newer OS.
- Need: approach for handling such overlap

Vulnerability density and defect density

- Defect density
  - Valuable metric for planning test effort
  - Used for setting release quality target
  - Some data is available
- Vulnerabilities are a class of defects
  - Vulnerability data is in the public domain.
  - Is vulnerability density a useful measure?
  - Is it related to defect density?
    - Vulnerabilities = 5% of defects [Longstaff]?
    - Vulnerabilities = 1% of defects [Anderson]?
- Can be a major step in measuring security.

Vulnerability density and defect density

- **Vulnerability densities**: 95/98: 0.003-0.004 NT/2000/XP: 0.01-0.02
- V_{KD}/D_{KD}: 0.68-1.62% about 1%

System	MSLOC	Known Defects (1000s)	D_{KD} (/Kloc)	Known Vulner - abilies	V_{KD} (/Kloc)	Ratio V_{KD} /D_{KD}
Win 95	15	5	0.33	46	0.0031	0.92%
NT 4.0	16	10	0.625	162	0.0101	1.62%
Win 98	18	10	0.556	84	0.0047	0.84%
Win2000	35	63	1.8	508	0.0145	0.81%
Win XP	40	106.5*	2.66*	728	0.0182	0.68%*

Summary and conclusions

We have introduced:

- Models:
  - Time – vulnerability model.
  - Usage – vulnerability model.
  - Both models shown acceptable goodness of fit.
    - Chi-square test.
- Measurements:
  - vulnerability density.
  - Vulnerability density vs. defect density.

Vulnerability Discovery in Multi-Version Software Systems

- Motivation
- Software Evolution
- Multi-version Software Discovery Model
  - Apache, Mysql and Win XP data

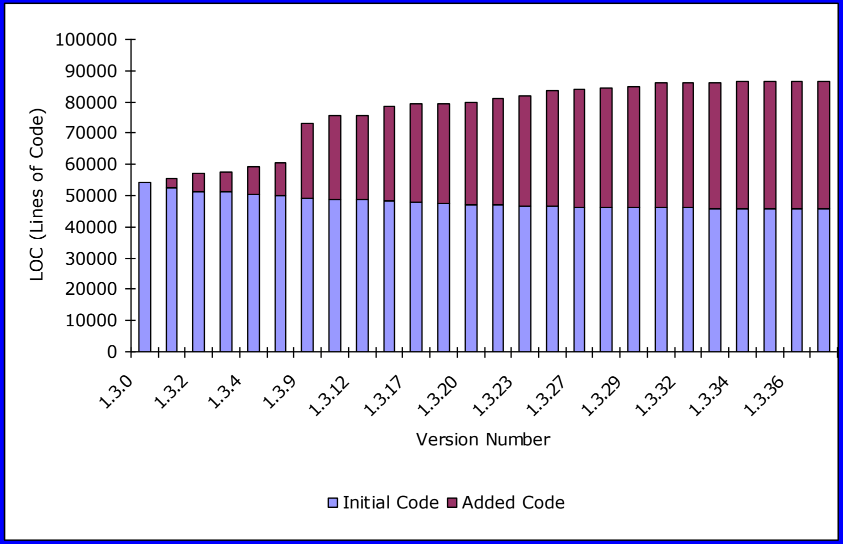
Motivation for Multi-version VDMs

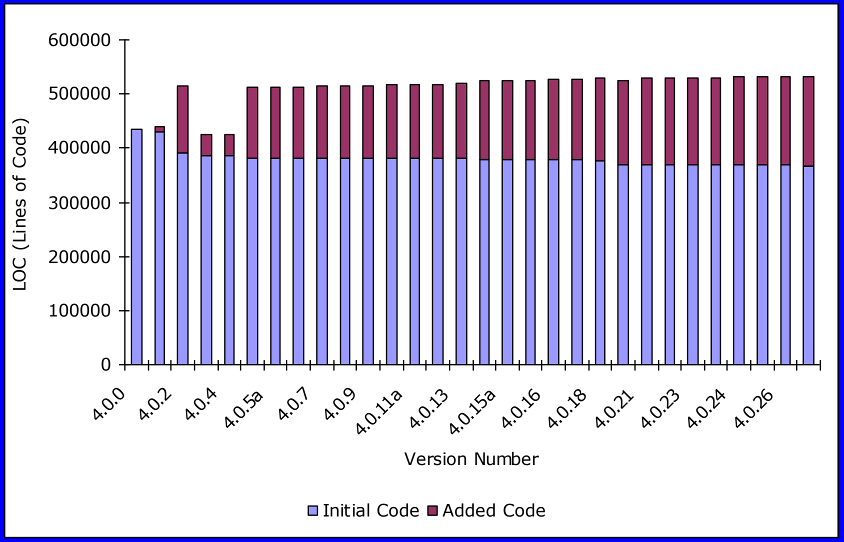
- Superposition effect on vulnerability discovery process due to shared code in successive versions.
- Examination of software evolution: impact on vulnerability introduction and discovery
- Other factors impacting vulnerability discovery process not considered before

Software Evolution

- The modification of software during maintenance or development:
  - fixes and feature additions.
  - Influenced by competition
- Code decay and code addition introduce new vulnerabilities
- Successive version of a software can share a significant fraction of code.

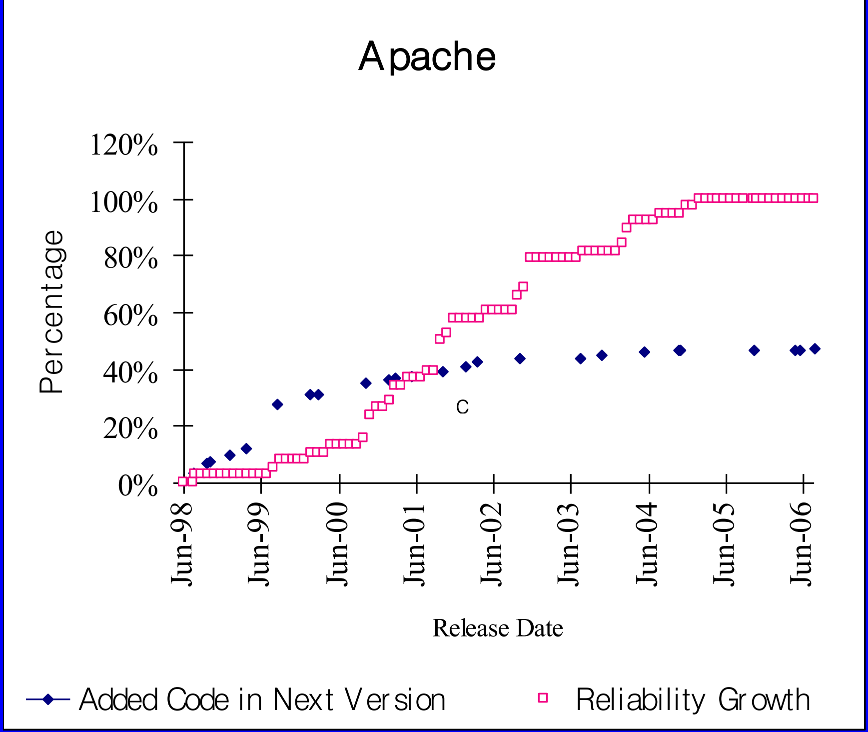
Software Evolution: Apache & Mysql

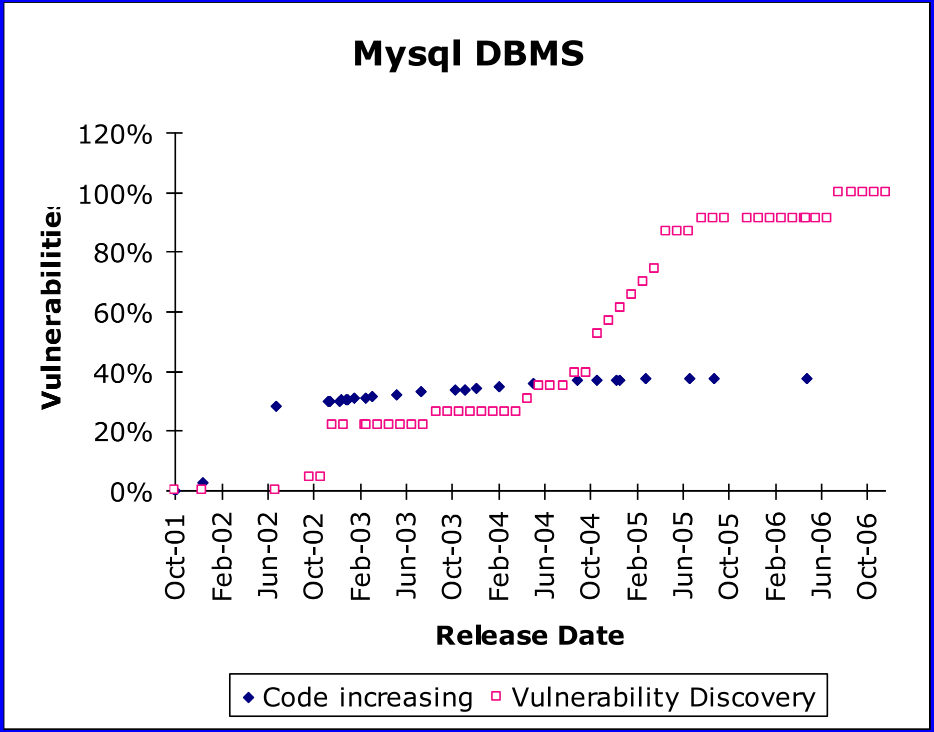




Modification: Apache 43%, Mysql 31%

Vulnerability Discovery & Evolution: Apache & Mysql

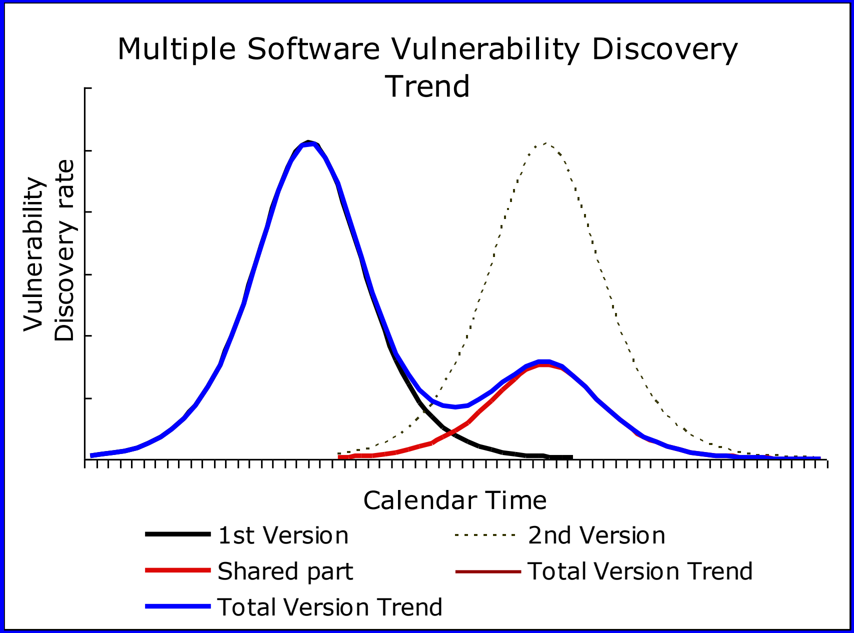




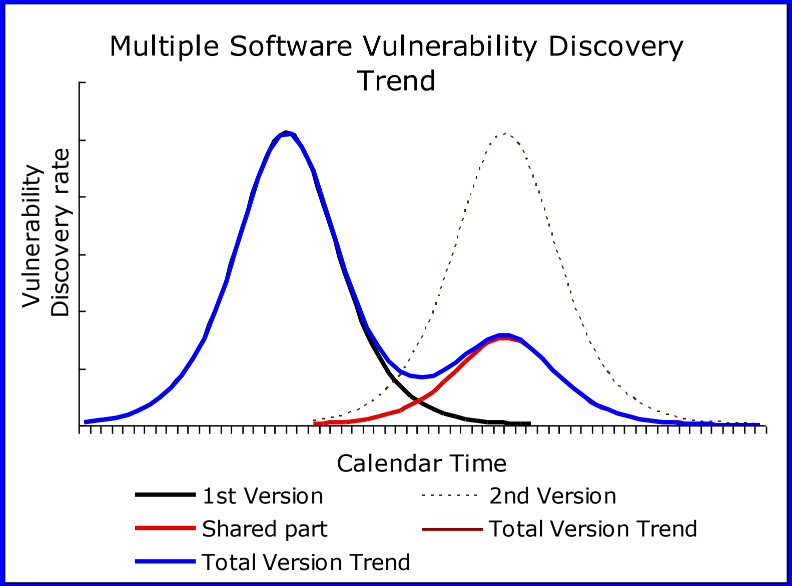
Some vulnerabilities are in added code, many are inherited from precious versions.

Code Sharing & Vulnerabilities

- Observation
  - Vulnerability increases after saturation in AML modeling
- Accounting for Superposition Effect
  - Shared components between several versions of software



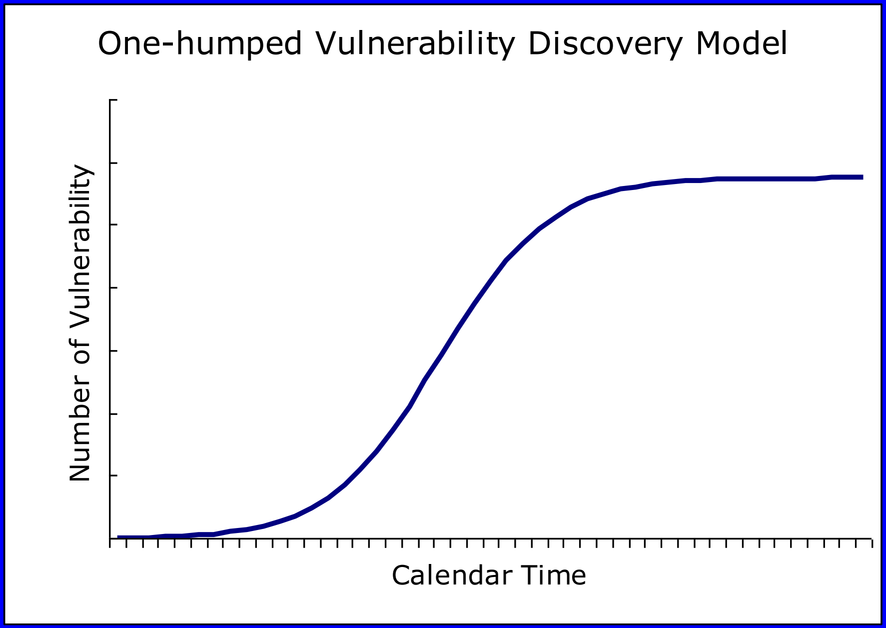
Multi-version Vulnerability Discovery Model

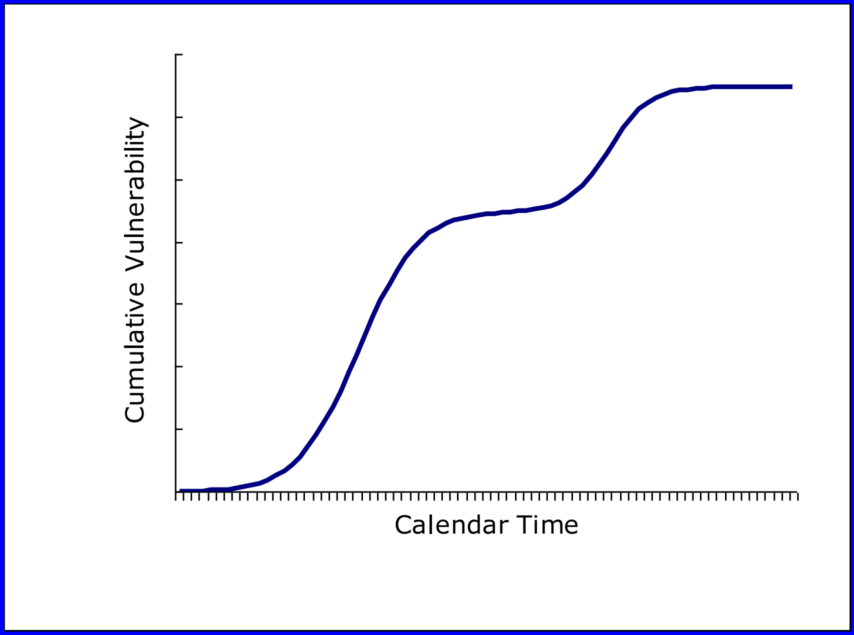


$$\Omega(t) = \frac{B}{BCe^{-ABt}+1} + \alpha \frac{B'}{B'C'e^{-A'B'(t-\varepsilon)}+1}$$

	Previous Version	Next Version	Shared Code Ratio α
Apache	1.3.24 (3-21-2002)	2.0.35 (4-6-2002)	20.16%
Mysql	4.1.1 (12-1-2003)	5.0.0 (12-22-2003)	83.52%

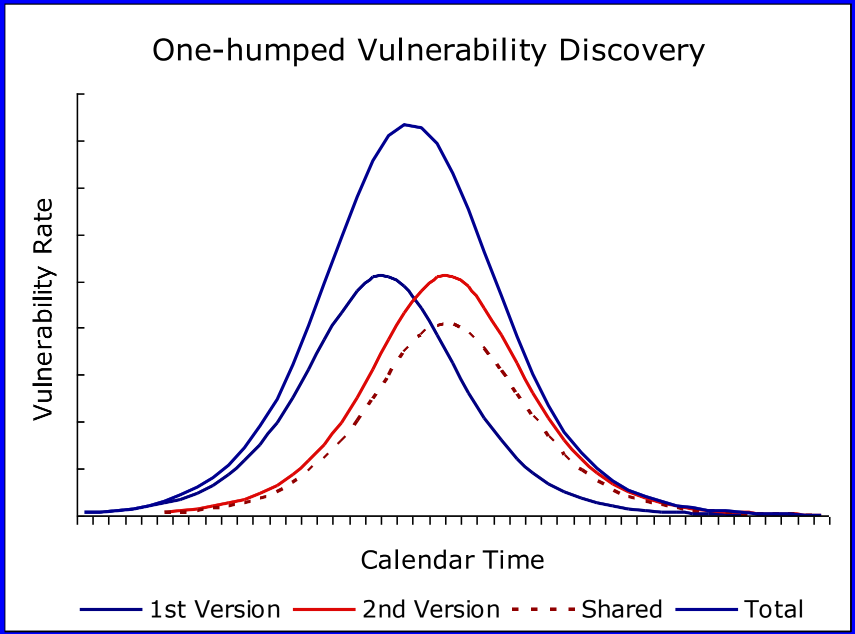
One vs Two Humps

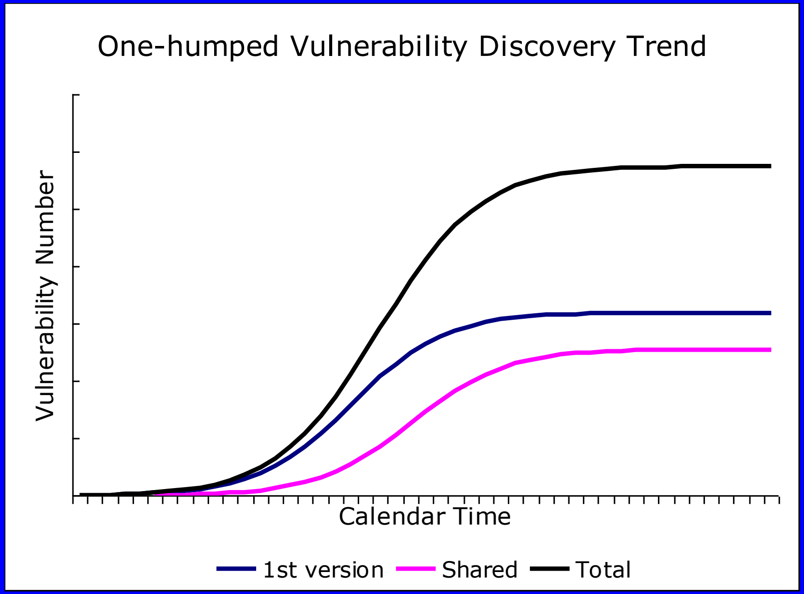




Superposition affect

Multi-version Vulnerability Discovery Model





- May result in a single hump with prolonged linear period

Seasonality in Vulnerability Discovery in Major Software Systems

- Vulnerability Discovery Model (VDM):
  - a probabilistic methods for modeling the discovery of software vulnerabilities [Ozment]
  - Spans a few years: introduction to replacement
- Seasonality: periodic variation
  - well known statistical approach
  - quite common in economic time series
    - Biological systems, stock markets etc.

Halloween indicator:
Low returns in May-Oct.

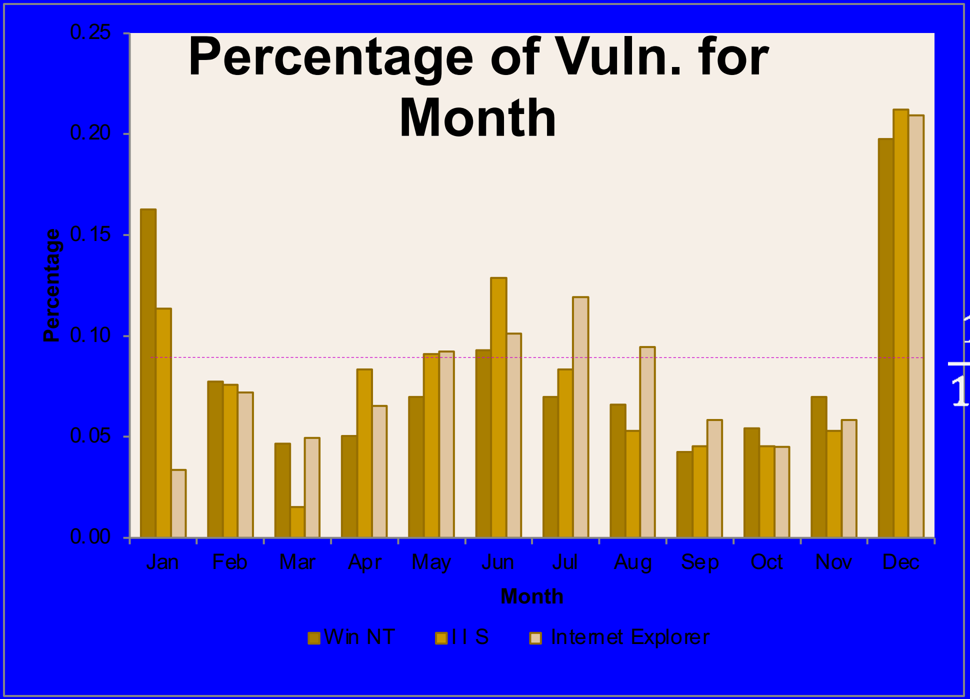
Examining Seasonality

- Is the seasonal pattern statistically significant?
- Periodicity of the pattern
- Analysis:
  - Seasonal index analysis with χ^2 test
  - Autocorrelation Function analysis
- Significance
  - Enhance VDMs' predicting ability

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



$\frac{1}{12}$

Seasonal Index

Seasonal Index Values

	WinNT	IIS	IE
Jan	1.95	1.36	0.41
Feb	0.93	0.91	0.86
Mar	0.56	0.81	0.59
Apr	0.60	1.00	0.78
May	0.84	1.09	1.11
Jun	1.12	1.55	1.22
Jul	0.84	1.00	1.43
Aug	0.79	0.64	1.14
Sep	0.51	0.55	0.70
Oct	0.65	0.55	0.54
Nov	0.84	0.64	0.70
Dec	2.37	2.55	2.51
χ^2_c	19.68	19.68	19.68
χ^2_s	78.37	46	130.43
p-value	3.04e-12	3.23e-6	1.42e-6

- **Seasonal index**: measures how much the average for a particular period tends to be above (or below) the expected value
- **H_0: no seasonality is present**. We will evaluate it using the monthly seasonal index values given by [4]:

$$s_i = \frac{d_i}{d}$$

where, s_i is the seasonal index for i^{th} month, d_i is the mean value of i^{th} month, d is a grand average

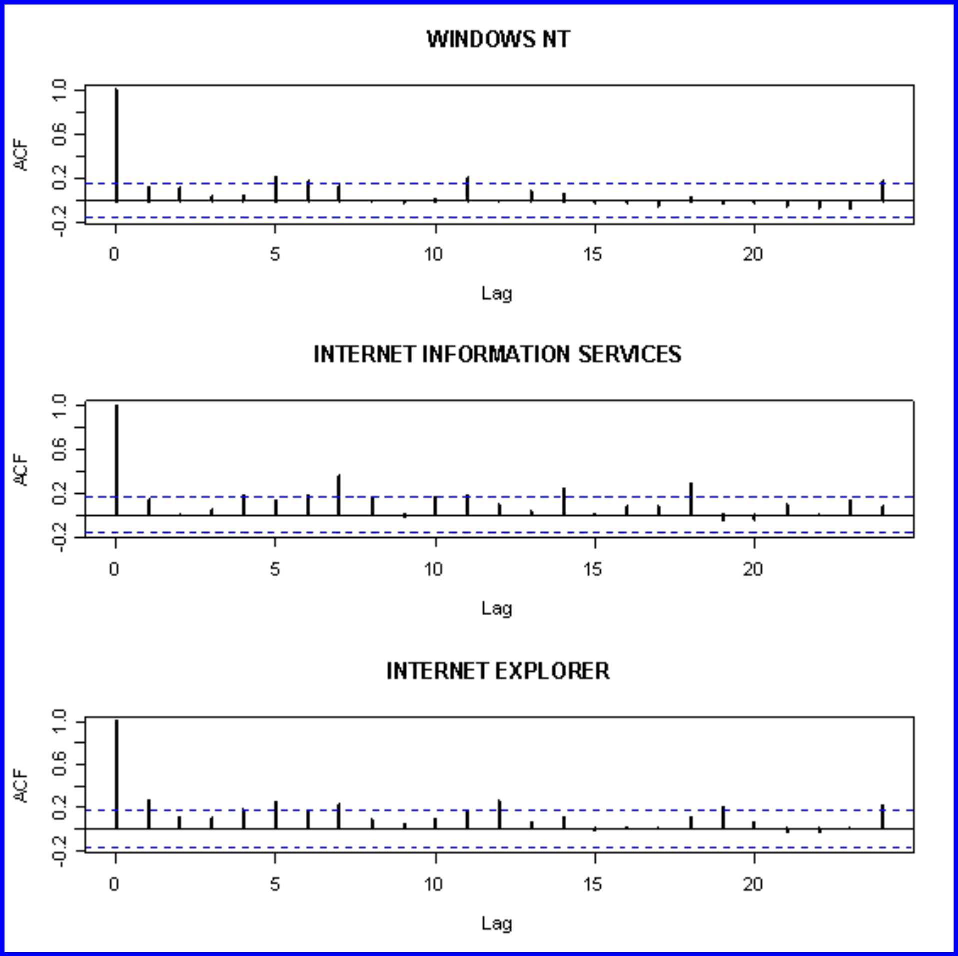
[4] Hossein Arsham. Time-Critical Decision Making for Business Administration. Available: http://home.ubalt. edu/ntsbarsh/Business-stat/stat-data/Forecast.htm#rseasonindx

Autocorrelation function (ACF)

- Plot of autocorrelations function values
- With time series values of z_b, z_{b+1}, …, z_n, the ACF at lag k, denoted by r_k, is [5]:
- $$r_k = \frac{\sum_{t=b}^{n-k}(z_t - \bar{z})(z_{t+k} - \bar{z})}{\sum_{t=b}^{n}(z_t - \bar{z})^2}$$
- , where $$\bar{z} = \frac{\sum_{t=b}^{n} z_t}{(n - b + 1)}$$
- Measures the linear relationship between time series observations separated by a lag of time units
- Hence, when an ACF value is located outside of confidence intervals at a lag t, it can be thought that every lag t, there is a relationships along with the time line

[5] B. L. Bowerman and R. T. O'connell, Time Series Forecsting: Unified concepts and computer implementation. 2nd Ed., Boston: Duxbury Press, 1987

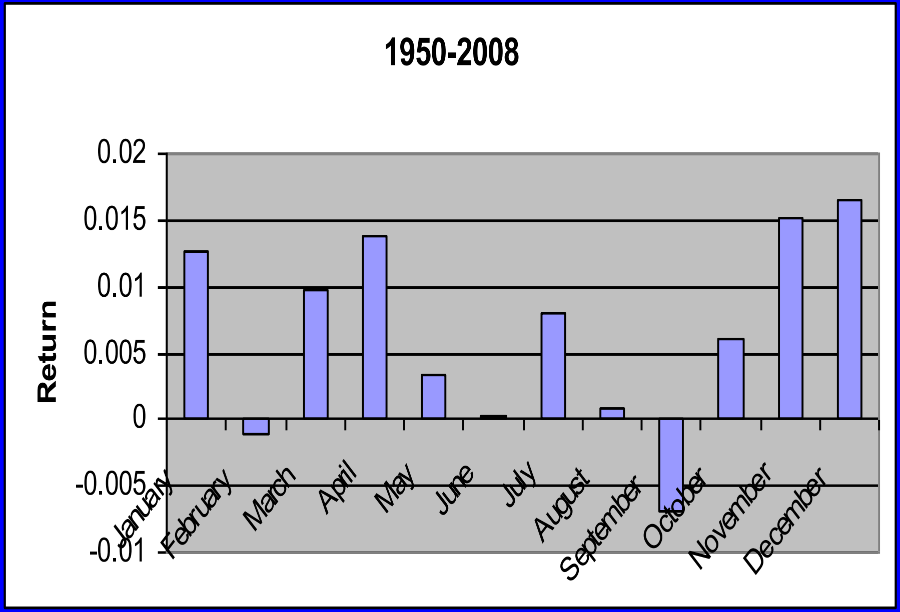
Autocorrelation (ACF):Results



- Expected lags corresponding to 6 months or its multiple would have their ACF values outside confidence interval
- Upper/lower dotted lines: 95% confidence intervals.
- An event occurring at time t + k (k > 0) lags behind an event occurring at time t.
- Lags are in month.

Halloween Indicator

- "Also known as "Sell in May and go away"
- Global (1973-1996):
  - Nov.-April: 12.47% ann., st dev 12.58%
  - 12-months:10.92%, st. dev. 17.76%
- 36 of 37 developing/developed nations
- Data going back to 1694
- "No convincing explanation"



Jacobsen, Ben and Bouman, Sven,The Halloween Indicator, 'Sell in May and Go Away': Another Puzzle(July 2001). Available at SSRN: http://ssrn.com/abstract=76248