# Cybersecurity: A Predictive Analytical Model for Software Vulnerability

**Netra Khanal**

The University of Tampa

Frontiers of Statistics Conference at USF
Tampa, FL

May 11, 2018

The University Of
TAMPA

Collaborative work with

- Mr. Nawa Raj Pokhrel, a PhD candidate at USF
- Dr. Keshav Pokhrel, Assistant Professor, University of Michigan Dearborn
- Dr. Chris P. Tsokos, Distinguished University Professor at USF

# Outline of the talk

1. Introduction
   - Overview
   - Existing Models
2. Modeling Approach
   - Proposed Differential Equation Model
   - Solution of Differential Equation
3. Results
   - An application to the vulnerability data
   - Model validation and comparison
   - Prediction accuracy
4. Conclusions

# Overview

- A software vulnerability is defined as a flaw that exists in computer resources that can be exploited by one or more threats
- A loophole that allows an attacker to compromise the system
- No software or operating system with no vulnerability
- The existence of vulnerabilities possess high risk to all the stakeholder of the software
- They are discovered during the entire life cycle of the software

The University Of
TAMPA

# Overview

- A software vulnerability is defined as a flaw that exists in computer resources that can be exploited by one or more threats
- A loophole that allows an attacker to compromise the system
- No software or operating system with no vulnerability
- The existence of vulnerabilities possess high risk to all the stakeholder of the software
- They are discovered during the entire life cycle of the software

# Overview

- A software vulnerability is defined as a flaw that exists in computer resources that can be exploited by one or more threats
- A loophole that allows an attacker to compromise the system
- No software or operating system with no vulnerability
- The existence of vulnerabilities possess high risk to all the stakeholder of the software
- They are discovered during the entire life cycle of the software

The University Of TAMPA

# Overview

- A software vulnerability is defined as a flaw that exists in computer resources that can be exploited by one or more threats
- A loophole that allows an attacker to compromise the system
- No software or operating system with no vulnerability
- The existence of vulnerabilities possess high risk to all the stakeholder of the software
- They are discovered during the entire life cycle of the software

The University Of
TAMPA

# Overview

- A software vulnerability is defined as a flaw that exists in computer resources that can be exploited by one or more threats
- A loophole that allows an attacker to compromise the system
- No software or operating system with no vulnerability
- The existence of vulnerabilities possess high risk to all the stakeholder of the software
- They are discovered during the entire life cycle of the software

The University Of
TAMPA

# Existing Models

- There are some existing models for software vulnerabilities: Musa-Okomoto Model (MO), Anderson Thermodynamic Model (AT)
- Rescorla Linear Model (RL): $\Omega(t) = Bt^2 + Kt$, obtained from the vulnerability rate $\omega(t) = Bt + K$, where B is the slope, and K is a constant
- Rescorla Exponential Model (RE): $\Omega(t) = N(1 - e^{\lambda t})$ where $N$ is the total number of vulnerabilities, and $\lambda$ is the rate constant
- Alhazmi-Malaiya Logistic Model (AML): $\Omega(t) = \frac{B}{BCe^{-ABt}+1}$

The University Of
TAMPA

# Existing Models

- There are some existing models for software vulnerabilities: Musa-Okomoto Model (MO), Anderson Thermodynamic Model (AT)
- Rescorla Linear Model (RL): $\Omega(t) = Bt^2 + Kt$, obtained from the vulnerability rate $\omega(t) = Bt + K$, where B is the slope, and K is a constant
- Rescorla Exponential Model (RE): $\Omega(t) = N(1 - e^{\lambda t})$ where $N$ is the total number of vulnerabilities, and $\lambda$ is the rate constant
- Alhazmi-Malaiya Logistic Model (AML): $\Omega(t) = \frac{B}{BCe^{-ABt}+1}$

The University Of
TAMPA

# Existing Models

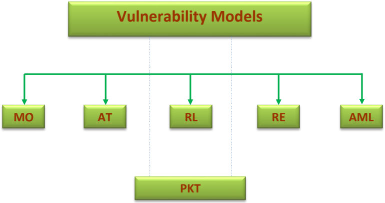
- There are some existing models for software vulnerabilities: Musa-Okomoto Model (MO), Anderson Thermodynamic Model (AT)
- Rescorla Linear Model (RL): $\Omega(t) = Bt^2 + Kt$, obtained from the vulnerability rate $\omega(t) = Bt + K$, where B is the slope, and K is a constant
- Rescorla Exponential Model (RE): $\Omega(t) = N(1 - e^{\lambda t})$ where $N$ is the total number of vulnerabilities, and $\lambda$ is the rate constant
- Alhazmi-Malaiya Logistic Model (AML): $\Omega(t) = \frac{B}{BCe^{-ABt}+1}$

# Existing Models

- There are some existing models for software vulnerabilities: Musa-Okomoto Model (MO), Anderson Thermodynamic Model (AT)
- Rescorla Linear Model (RL): $\Omega(t) = Bt^2 + Kt$, obtained from the vulnerability rate $\omega(t) = Bt + K$, where B is the slope, and K is a constant
- Rescorla Exponential Model (RE): $\Omega(t) = N(1 - e^{\lambda t})$ where $N$ is the total number of vulnerabilities, and $\lambda$ is the rate constant
- Alhazmi-Malaiya Logistic Model (AML): $\Omega(t) = \frac{B}{BCe^{-ABt}+1}$

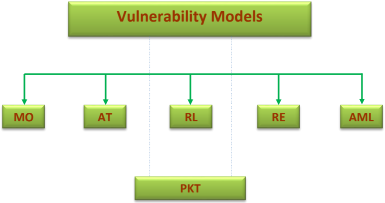
# Existing Models

- The AML model assumes that the vulnerability discovery rate increases at the beginning, reaches a steady rate, and then starts to decline
- It was discovered that the models such as RL, RE,and AT failed the goodness of fit tests except the AML model
- Existing and Proposed Models shown pictorially as follows:



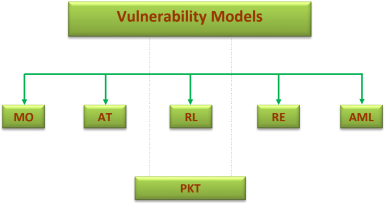
# Existing Models

- The AML model assumes that the vulnerability discovery rate increases at the beginning, reaches a steady rate, and then starts to decline
- It was discovered that the models such as RL, RE,and AT failed the goodness of fit tests except the AML model
- Existing and Proposed Models shown pictorially as follows:



# Existing Models

- The AML model assumes that the vulnerability discovery rate increases at the beginning, reaches a steady rate, and then starts to decline
- It was discovered that the models such as RL, RE,and AT failed the goodness of fit tests except the AML model
- Existing and Proposed Models shown pictorially as follows:



# Why differential equation?

- A careful reading of the scatter plots for the three different operating systems do not support the claim that the vulnerability attains a saturation phase

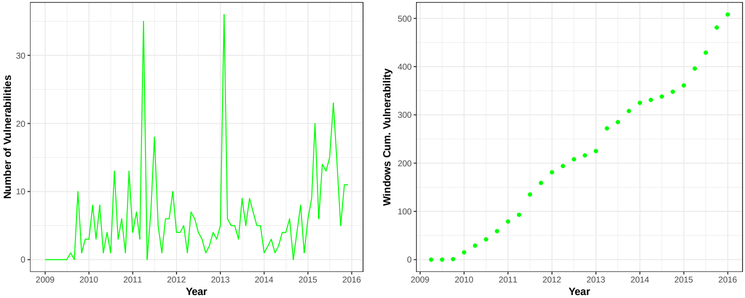


Figure: The monthly time series and cumulative quarterly scatter plot for Windows 7

# Why differential equation?

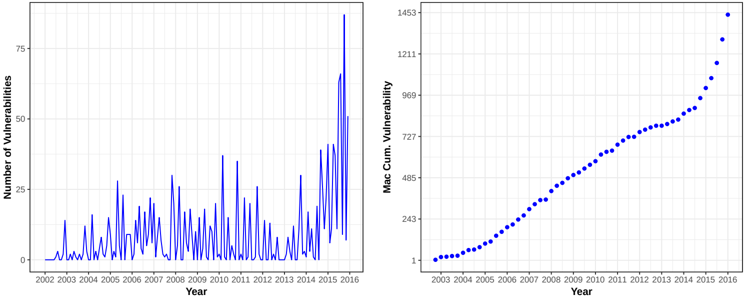


Figure: The monthly time series and cumulative quarterly scatter plot for Mac OS X

# why differential equation?

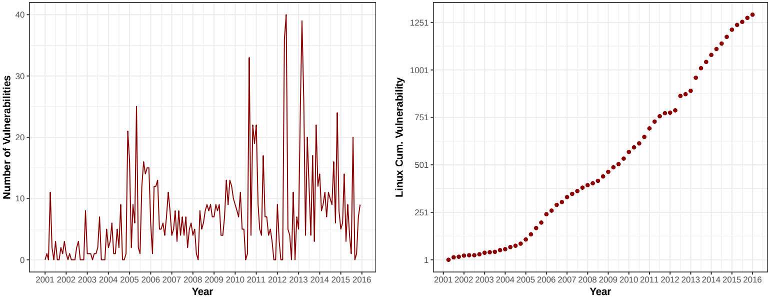
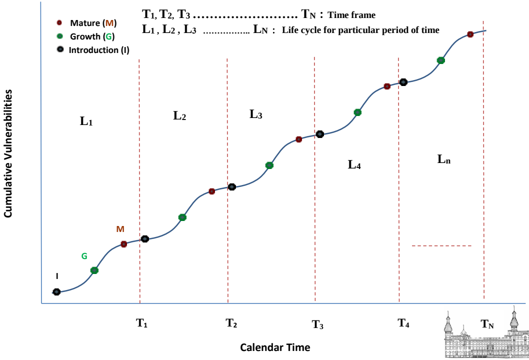


Figure: The monthly time series and cumulative quarterly scatter plot for Linux Karnel

The University Of TAMPA

# why differential equation?

- Existing models are developed based on three transition phases of vulnerability life cycle (introduction, growth, and mature) but we claim that this is a local phenomemon


Mature (M)
Growth (G)
Introduction (I)
$T_1, T_2, T_3$ ........................ $T_N$ : Time frame
$L_1, L_2, L_3$ ................ $L_N$ : Life cycle for particular period of time
$L_1$
$L_2$
$L_3$
$L_4$
$L_n$
I
G
M
Cumulative Vulnerabilities
$T_1$
$T_2$
$T_3$
$T_4$
$T_N$
Calendar Time

# differential equation model

- We plan to develop an analytic model that more accurately captures the dynamics of the total cumulative vulnerabilities of a given OS
- We propose a new time based nonlinear differential equation model given by

$$\Omega''(t) + \omega^2\Omega(t) = f(t), \tag{1}$$

  where $\Omega(t)$ is the cumulative vulnerability count at time $t$, and $f(t)$ is the quadratic forcing term
- This type of model is suitable for any data that has increasing cyclic behavior. The only issue is to find the suitable $w$

The University Of
TAMPA

# differential equation model

- We plan to develop an analytic model that more accurately captures the dynamics of the total cumulative vulnerabilities of a given OS
- We propose a new time based nonlinear differential equation model given by

  $$\Omega''(t) + \omega^2\Omega(t) = f(t), \qquad (1)$$

  where $\Omega(t)$ is the cumulative vulnerability count at time $t$, and $f(t)$ is the quadratic forcing term
- This type of model is suitable for any data that has increasing cyclic behavior. The only issue is to find the suitable $w$

The University Of
TAMPA

# differential equation model

- We plan to develop an analytic model that more accurately captures the dynamics of the total cumulative vulnerabilities of a given OS
- We propose a new time based nonlinear differential equation model given by

  $$\Omega''(t) + \omega^2 \Omega(t) = f(t), \tag{1}$$

  where $\Omega(t)$ is the cumulative vulnerability count at time $t$, and $f(t)$ is the quadratic forcing term
- This type of model is suitable for any data that has increasing cyclic behavior. The only issue is to find the suitable $w$

# solution of differential equation

- The differential equation is a second order, linear, nonhomogeneous differential equation
- A general solution of the differential equation 1 is given by

$$\Omega(t) = c_1 \cos(\omega t) + c_2 \sin(\omega t) + c_3\, t^2 + c_4\, t + c_5, \tag{2}$$

where $c_1$, $c_2$,...,$c_5$ are the coefficients that derives the model. The model 2 is considered as the final mathematical model, named as Pokhrel-Khanal-Tsokos differential equation model (PKT Model)

# solution of differential equation

- The differential equation is a second order, linear, nonhomogeneous differential equation
- A general solution of the differential equation 1 is given by

$$\Omega(t) = c_1 \cos(\omega t) + c_2 \sin(\omega t) + c_3\, t^2 + c_4\, t + c_5, \qquad (2)$$

where $c_1$, $c_2$,...,$c_5$ are the coefficients that derives the model. The model 2 is considered as the final mathematical model, named as Pokhrel-Khanal-Tsokos differential equation model (PKT Model)

The University Of
TAMPA

# solution of differential equation

- The differential equation is a second order, linear, nonhomogeneous differential equation
- A general solution of the differential equation 1 is given by

$$\Omega(t) = c_1 \cos(\omega t) + c_2 \sin(\omega t) + c_3\, t^2 + c_4\, t + c_5, \qquad (2)$$

where $c_1$, $c_2$,...,$c_5$ are the coefficients that derives the model. The model 2 is considered as the final mathematical model, named as Pokhrel-Khanal-Tsokos differential equation model (PKT Model)

# Vulnerability Data

- We have extracted the vulnerability data from the National Vulnerability Database (NVD)
- NVD is a product of the National Institute of Standards and Technology (NIST)
- We have collected the vulnerabilities for three Operating System namely Mac OS X, Linux Kernel, and Windows 7
- We find quarterly sum of vulnerability counts
- The vulnerability data of four quarters of 2016 is used as testing data to validate our analytic model

The University Of
TAMPA

# Vulnerability Data

- We have extracted the vulnerability data from the National Vulnerability Database (NVD)
- NVD is a product of the National Institute of Standards and Technology (NIST)
- We have collected the vulnerabilities for three Operating System namely Mac OS X, Linux Kernel, and Windows 7
- We find quarterly sum of vulnerability counts
- The vulnerability data of four quarters of 2016 is used as testing data to validate our analytic model

# Vulnerability Data

- We have extracted the vulnerability data from the National Vulnerability Database (NVD)
- NVD is a product of the National Institute of Standards and Technology (NIST)
- We have collected the vulnerabilities for three Operating System namely Mac OS X, Linux Kernel, and Windows 7
- We find quarterly sum of vulnerability counts
- The vulnerability data of four quarters of 2016 is used as testing data to validate our analytic model

The University Of
TAMPA

# Vulnerability Data

- We have extracted the vulnerability data from the National Vulnerability Database (NVD)
- NVD is a product of the National Institute of Standards and Technology (NIST)
- We have collected the vulnerabilities for three Operating System namely Mac OS X, Linux Kernel, and Windows 7
- We find quarterly sum of vulnerability counts
- The vulnerability data of four quarters of 2016 is used as testing data to validate our analytic model

# Vulnerability Data

- We have extracted the vulnerability data from the National Vulnerability Database (NVD)
- NVD is a product of the National Institute of Standards and Technology (NIST)
- We have collected the vulnerabilities for three Operating System namely Mac OS X, Linux Kernel, and Windows 7
- We find quarterly sum of vulnerability counts
- The vulnerability data of four quarters of 2016 is used as testing data to validate our analytic model

# Evaluation of parameters

- In the proposed model $\Omega(t) = c_1 \cos(\omega t) + c_2 \sin(\omega t) + c_3 t^2 + c_4 t + c_5$, $\omega = \frac{2\pi}{T}$ depends upon the time period $T$
- For Mac OS X, we consider just one cycle for simplicity and use $\omega \approx 0.116355283466$ in our model
- Using nls tools in R, a nonlinear modeling approach, the estimated values of the parameters are: $c_1$=26656.79, $c_2$=32220.98, $c_3$=272.47, $c_4$=-4033.11, and $c_5$=-26376.42
- Therefore, the model for Mac OS X is given by

$$\Omega(t) = 26656.79 \cos(0.12t) + 32220.98 \sin(0.12t) + 272.47t^2 - 4033.11 \tag{3}$$

# Evaluation of parameters

- In the proposed model
  $\Omega(t) = c_1 \cos(\omega t) + c_2 \sin(\omega t) + c_3 t^2 + c_4 t + c_5$,
  $\omega = \frac{2\pi}{T}$ depends upon the time period $T$
- For Mac OS X, we consider just one cycle for simplicity and use $\omega \approx 0.116355283466$ in our model
- Using nls tools in R, a nonlinear modeling approach, the estimated values of the parameters are: $c_1$=26656.79, $c_2$=32220.98, $c_3$=272.47, $c_4$=-4033.11, and $c_5$=-26376.42
- Therefore, the model for Mac OS X is given by

$$\Omega(t) = 26656.79 \cos(0.12t) + 32220.98 \sin(0.12t) + 272.47t^2 - 4033.11 \qquad (3)$$

# Evaluation of parameters

- In the proposed model
  $\Omega(t) = c_1 \cos(\omega t) + c_2 \sin(\omega t) + c_3 t^2 + c_4 t + c_5$,
  $\omega = \frac{2\pi}{T}$ depends upon the time period $T$
- For Mac OS X, we consider just one cycle for simplicity and use $\omega \approx 0.116355283466$ in our model
- Using nls tools in R, a nonlinear modeling approach, the estimated values of the parameters are: $c_1$=26656.79, $c_2$=32220.98, $c_3$=272.47, $c_4$=-4033.11, and $c_5$=-26376.42
- Therefore, the model for Mac OS X is given by

$$\Omega(t) = 26656.79\cos(0.12t) + 32220.98\sin(0.12t) + 272.47t^2 - 4033.11 \tag{3}$$

The University Of
TAMPA

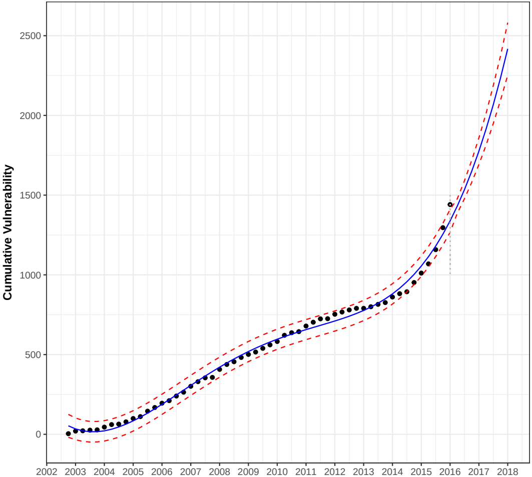
# Evaluation of parameters

- In the proposed model $\Omega(t) = c_1\cos(\omega t) + c_2\sin(\omega t) + c_3 t^2 + c_4 t + c_5$, $\omega = \frac{2\pi}{T}$ depends upon the time period $T$
- For Mac OS X, we consider just one cycle for simplicity and use $\omega \approx 0.116355283466$ in our model
- Using nls tools in R, a nonlinear modeling approach, the estimated values of the parameters are: $c_1$=26656.79, $c_2$=32220.98, $c_3$=272.47, $c_4$=-4033.11, and $c_5$=-26376.42
- Therefore, the model for Mac OS X is given by

$$\Omega(t) = 26656.79\cos(0.12t) + 32220.98\sin(0.12t) + 272.47t^2 - 4033.11 \tag{3}$$

# 95% Confidence Band using PKT Model

- The fitted values given by the PKT model together with cumulative vulnerability data and 95% confidence and prediction band for Mac OS X



# 95% Confidence Band using PKT Model

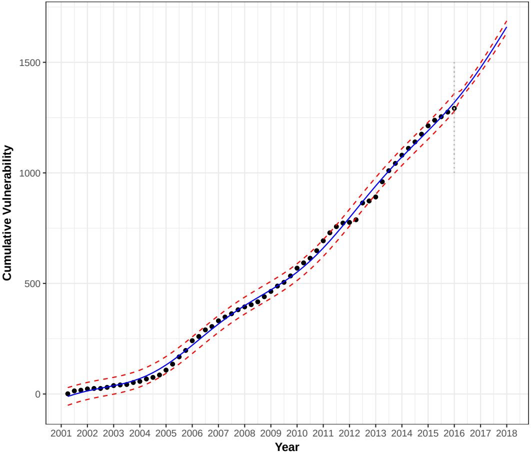
- The PKT model is also applied to develop the nonlinear models for Linux Kernel and Windows 7 OS. The final model for Linux and Windows 7 are given by the following equations:

$$\begin{aligned}\Omega(t) = {} & 1.13\cos(1.05t) + 23.55\sin(1.05t) + 3.97t^2 \\ & + 24.57t - 71.04,\end{aligned}$$

$$\begin{aligned}\Omega(t) = {} & 18.94\cos(1.15t) - 27.79\sin(1.15t) + 7.57t^2 \\ & + 4.54t + 7.57.\end{aligned}$$

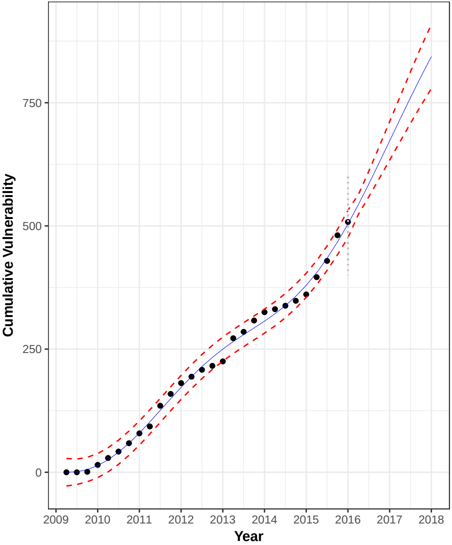
# 95% Confidence Band using PKT Model

- 95% confidence and prediction band for Linux Kernel



# 95% Confidence Band using PKT Model

- 95% confidence and prediction band for Windows 7



# RSS and AIC Comparison

- We compare PKT model with the other existing vulnerability discovery models, namely RL, RE, and AML
- The comparison is based on Sum of Squares (RSS) and Akaike Information Criteria(AIC)
- PKT model depicts lower RSS and AIC values

The University Of
TAMPA

# RSS and AIC Comparison

- We compare PKT model with the other existing vulnerability discovery models, namely RL, RE, and AML
- The comparison is based on Sum of Squares (RSS) and Akaike Information Criteria(AIC)
- PKT model depicts lower RSS and AIC values

# RSS and AIC Comparison

- We compare PKT model with the other existing vulnerability discovery models, namely RL, RE, and AML
- The comparison is based on Sum of Squares (RSS) and Akaike Information Criteria(AIC)
- PKT model depicts lower RSS and AIC values

The University Of
TAMPA

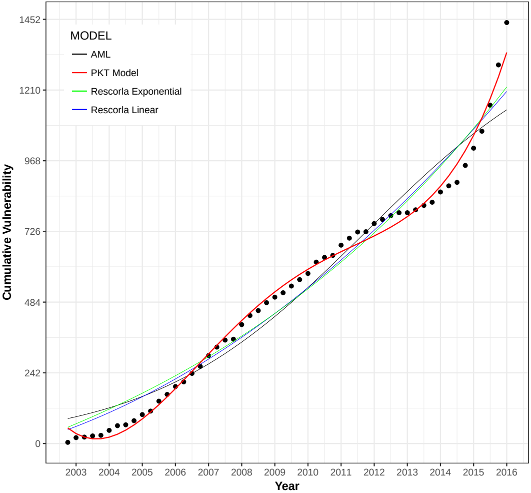
# RSS and AIC Comparison

- We compare PKT model with the other existing vulnerability discovery models, namely RL, RE, and AML
- The comparison is based on Sum of Squares (RSS) and Akaike Information Criteria(AIC)
- PKT model depicts lower RSS and AIC values

| Operating Systems | Models | RSS | AIC |
|---|---|---|---|
| MAC | RL | 241314.7 | 633.5574 |
| | RE | 262502.3 | 638.2703 |
| | AML | 334296.1 | 653.8092 |
| | PKT | **45584.43** | **529.1156** |
| Linux Kernel | RL | 48998.95 | 578.5852 |
| | RE | 124456.7 | 634.5147 |
| | AML | 78961.47 | 609.2149 |
| | PKT | **18451.8** | **525.987** |
| Windows 7 | RL | 16595.93 | 264.2324 |
| | RE | 22418.38 | 272.6527 |
| | AML | 17965.02 | 268.4519 |
| | PKT | **2808.963** | **220.4948** |

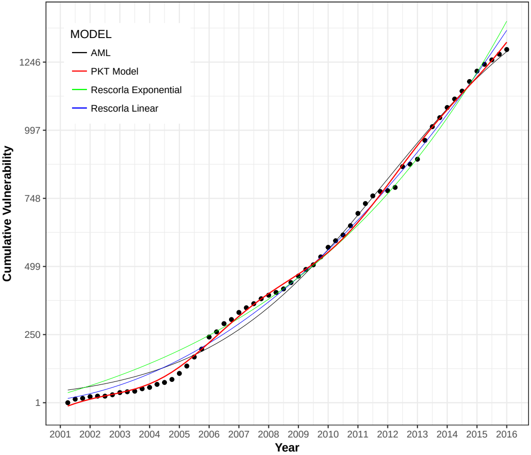
# Estimated fit given by different models

- The PKT model comparing with RL, RE, and AML for Mac OS X



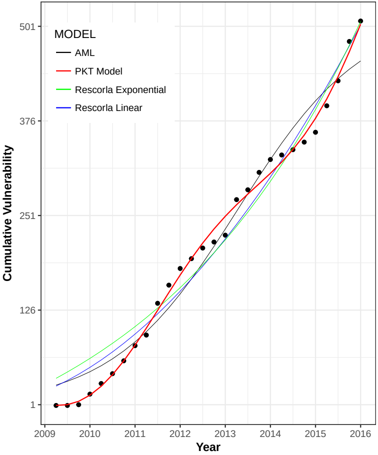
# Estimated fit given by different models

- The fitted PKT model captures the cyclic trend reasonably better than the other models



# Estimated fit given by different models

- The variability of data is higher towards the right tail PKT model stands out to capture the trend



The University Of TAMPA

# Prediction by PKT Model

- The proposed analytical model for software vulnerability and other existing models can be used to project the future vulnerability trends
- The models are fitted using data up to the last quarter of 2015 and vulnerability counts of 2016 and 2017 are estimated by using the fitted model
- We used actual vulnerability data of 2016 and 2017 for validation purpose
- The following Table shows that the prediction is very accurate for all quarters in case of Linux and Windows

The University Of
TAMPA

# Prediction by PKT Model

- The proposed analytical model for software vulnerability and other existing models can be used to project the future vulnerability trends
- The models are fitted using data up to the last quarter of 2015 and vulnerability counts of 2016 and 2017 are estimated by using the fitted model
- We used actual vulnerability data of 2016 and 2017 for validation purpose
- The following Table shows that the prediction is very accurate for all quarters in case of Linux and Windows

The University Of
TAMPA

# Prediction by PKT Model

- The proposed analytical model for software vulnerability and other existing models can be used to project the future vulnerability trends
- The models are fitted using data up to the last quarter of 2015 and vulnerability counts of 2016 and 2017 are estimated by using the fitted model
- We used actual vulnerability data of 2016 and 2017 for validation purpose
- The following Table shows that the prediction is very accurate for all quarters in case of Linux and Windows

# Prediction by PKT Model

- The proposed analytical model for software vulnerability and other existing models can be used to project the future vulnerability trends
- The models are fitted using data up to the last quarter of 2015 and vulnerability counts of 2016 and 2017 are estimated by using the fitted model
- We used actual vulnerability data of 2016 and 2017 for validation purpose
- The following Table shows that the prediction is very accurate for all quarters in case of Linux and Windows

# Prediction by PKT Model

- The proposed analytical model for software vulnerability and other existing models can be used to project the future vulnerability trends
- The models are fitted using data up to the last quarter of 2015 and vulnerability counts of 2016 and 2017 are estimated by using the fitted model
- We used actual vulnerability data of 2016 and 2017 for validation purpose
- The following Table shows that the prediction is very accurate for all quarters in case of Linux and Windows

# Predicted vulnerability by PKT Model

| **Operating Systems** | | **2016** | | | |
|---|---|---|---|---|---|
| | | **Q1** | **Q2** | **Q3** | **Q4** |
| Linux Kernel | Predicted Interval | [1340-1370] | [1377-1410] | [1416-1452] | [1457-1496] |
| | Predicted Vulnerability | 1354 | 1393 | 1433 | 1475 |
| | Actual Vulnerability | 1306 | 1407 | 1443 | 1522 |
| Windows 7 | Predicted Interval | [503-556] | [536-602] | [569-650] | [637-749] |
| | Predicted Vulnerability | 537 | 573 | 609 | 644 |
| | Actual Vulnerability | 538 | 569 | 596 | 642 |
| Mac OS X | Predicted Interval | [1252-1497] | [1319-1614] | [1392-1742] | [1472-1888] |
| | Predicted Vulnerability | 1431 | 1534 | 1649 | 1775 |
| | Actual Vulnerability | 1499 | 1573 | 1656 | 1756 |

# SSE of predicted vulnerabilities

- On SSE scale, PKT model has lower SSE in terms of predictive capabilities.

| **Operating Systems** | **SSE** | | | |
|---|---|---|---|---|
| | **PKT** | **RL** | **RE** | **AML** |
| Linux Kernel | **1603** | 4259.33 | 13839.33 | 13710 |
| Windows 7 | **63.33** | 179.33 | 109.67 | 17494.67 |
| Mac OS X | **2185** | 151149 | 128300.3 | 260835.3 |

# Conclusions

- We have developed an effective differential equation model for software vulnerabilities
- The proposed analytical model is significantly much better among the existing models in terms of excellent fitting and prediction accuracy
- Next goal is to study the software reliability analysis

# Conclusions

- We have developed an effective differential equation model for software vulnerabilities
- The proposed analytical model is significantly much better among the existing models in terms of excellent fitting and prediction accuracy
- Next goal is to study the software reliability analysis

The University Of
TAMPA

# Conclusions

- We have developed an effective differential equation model for software vulnerabilities
- The proposed analytical model is significantly much better among the existing models in terms of excellent fitting and prediction accuracy
- Next goal is to study the software reliability analysis

# Thank You

The University Of
TAMPA