Scientific Research Publishing

Journal of Information Security, 2018, 9, 12-24
http://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Non-Homogeneous Stochastic Model for Cyber Security Predictions

Pubudu Kalpani Kaluarachchi[1*], Chris P. Tsokos[2], Sasith M. Rajasooriya[3]

[1]Department of Mathematical and Physical Sciences, Miami University, Middletown, Ohio, USA
[2]Distinguished University Professor, Department of Mathematics and Statistics, University of South Florida, Tampa, Florida, USA
[3]Department of Statistics, Miami University, Oxford, Ohio, USA
Email: *hitigap@miamioh.edu

How to cite this paper: Kaluarachchi, P.K., Tsokos, C.P. and Rajasooriya, S.M. (2018) Non-Homogeneous Stochastic Model for Cyber Security Predictions. *Journal of Information Security*, **9**, 12-24.
https://doi.org/10.4236/jis.2018.91002

Received: October 29, 2017
Accepted: November 27, 2017
Published: November 30, 2017

Abstract

Any computer system with known vulnerabilities can be presented using attack graphs. An attacker generally has a mission to reach a goal state that he expects to achieve. **Expected Path Length (EPL)** [1] in the context of an attack graph describes the length or number of steps that the attacker has to take in achieving the goal state. However, **EPL** varies and it is based on the "**state of vulnerabilities**" [2] [3] in a given computer system. Any vulnerability throughout its life cycle passes through several stages that we identify as "**states of the vulnerability life cycle**" [2] [3]. In our previous studies we have developed mathematical models using Markovian theory to estimate the probability of a given vulnerability being in a particular state of its life cycle. There, we have considered a typical model of a computer network system with two computers subject to three vulnerabilities, and developed a method driven by an algorithm to estimate the EPL of this network system as a function of time. This approach is important because it allows us to monitor a computer system during the process of being exploited. Proposed **non-homogeneous** model in this study estimates the behavior of the **EPL** as a function of time and therefore act as an index of the risk associated with the network system getting exploited.

Keywords

Vulnerability, Attack Graph, Markov Model, Security Evaluation, Expected Path Length (EPL), Common Vulnerability Scoring System (CVSS), Non Homogeneous Stochastic Model

1. Introduction

In 2016, the U.S. Government Cybersecurity report commences with the fol-

lowing paragraph. [4] "*In July* 2015, *hackers stole social security numbers, health data, and other highly sensitive data from* 21 *million Americans through the Office of Personnel Management in what, at the time, was the largest data breach in U.S. history. As a response, U.S. government agencies committed to making significant efforts to reinforce and expand existing security measures. Security Scorecard wanted to find out if these government agencies were successful in their commitment*". "Symantec corporation", in their "Internet security threat report 2016-Volume 21" [5] presents with records on rapidly increasing **vulnerabilities**, **security threats, susceptibility of systems** that motivates researchers to study these important issues on Cybersecurity measures. Cybersecurity is one of the critical issues that our global society is facing on daily basis. It is now a part of our daily life and culture and has become an index of personal security and integrity.

To address this scenario, many research efforts have been taken. However, due to the peculiar, voluminous and dynamic nature of the field, defending methods are still chasing behind the defending targets. Therefore, it is extremely important to integrate scientific efforts and develop strong theoretical basis aiming for rapid development of applications and system solutions.

In this study, we continue our research efforts in integrating Mathematical and Statistical theories into better understanding the complex behavior of computer network systems in the perspective of Cybersecurity. Thus, we propose a new method to estimate the **EPL** as a function of time "*t*". The EPL is a major factor in determining the risk level of a given computer system where with smaller EPL, the network system is more vulnerable and probable to be exploited.

In our recent studies, [1] [2] [3], we introduced several stochastic models to better understand the behavior of vulnerabilities, network systems with respect to cybersecurity. Initially, we introduced a stochastic model that can estimate the Expected Path Length of a system with any three vulnerabilities and two machines. Then, we introduced a new approach of estimating the probability of a given vulnerability being exploited at a time *t*, using Markovian approach with respect to the Vulnerability life cycle. We have further introduced a set of three stochastic time dependent models for each categories of vulnerabilities with **Low**, **Medium** and **High** exploitability scores [6] that can estimate the probability of a given vulnerability getting exploited without going through the Markovian process [1] [2] [7] each time. Additionally, the concept of "**Risk Factor**" [2] [3] that we introduced and its analytical formulation allowed us to present a more sophisticated way of estimating the risk associated with a specific vulnerability of a computer network system.

In the present study, we introduce a **Non Homogeneous Stochastic Model** that allows the computer system administrators to predict the time that the system is most vulnerable for an attack in terms of the EPL. This estimate is based on the assumption that a system is more susceptible to be exploited when the EPL is at a minimum at a particular time "*t*". In developing this model we have

used a network system of two IPs with three vulnerabilities as a base model.

With the introduction of this new approach we will be re-defending the capability to estimate the probability of getting exploited as a function of time for a computer network system with given set of vulnerabilities. Even though we have already developed a successful statistical model to find the EPL of a possible attack, it is more important to estimate the EPL as a function of time. Current study will address this need. Thus, for a system with a given set of vulnerabilities, estimating of most probable exploit times can be modelled on the logical assumption that a system is more susceptible to be exploited at a time where the Expected Path Length (number of steps that an attacker needs to pass before achieving the goal state) is at its minimum.

2. Methodology

2.1. Cybersecurity Analysis Method

The core component of this method is the attack graph [1] [7]. An **attack graph** for a cybersecurity system has several nodes, which represent both the vulnerabilities that exist in the system and the attacker's states [1] [7]. We consider that it is possible to go to a goal state starting from any other state in the **attack graph**. This possibility depends on several factors such as the **attacker's attacking strategy**, **recourses**, **system design**, **networking**, **authenticating protocols**, **human interface** and other environmental factors. An attack graph has at least one "**Absorbing state"** named "**Goal state",** which is, the state where the attacker will reach his objective and cannot go beyond. Therefore we will model the attack graph as an absorbing Markov chain [1] [2] [3].

Absorbing state or goal state is the security node which the attacker expects to reach and exploit. When the attacker has reached this goal state, the attack path is completed. Thus, the entire attack graph consists of these types of attack paths that will be illustrated in this study.

Given the **CVSS** score [8] [9] for each vulnerability in the **attack Graph**, we can estimate the transition probabilities of the absorbing Markov chain by normalizing the **CVSS** scores over all the edges starting from the attacker's source state (initial state). The analytical methodology that we used is explained below.

We define,

j = probability that an attacker is currently in state i and exploits a vulnerability in state.

n = number of outgoing edges from state i in the attack model.

v_j = CVSS score of the vulnerability in state j.

Thus formally we can define the transition probability given by,

$$p_{ij} = \frac{v_j}{\sum_{k=1}^{n} v_k}$$

Now, using these transition probabilities we can derive the absorbing transition probability matrix P, which possesses the properties defined under Markov

chain probability methods.

2.2. Risk Factor Model

p_{ij}, the transition probabilities for each state in an **attack graph** represent the risk of a particular state (for a given vulnerability) of being exploited. Therefore, it is logical to consider it as a risk variable. In our previous studies we have introduced a more convenient tool named "**Risk Factor**" [2] [3] that can estimate the risk associated with a particular state of a given vulnerability.

It is important to note that when we consider a given vulnerability, its exploitability factor should vary with time. But the exploitability factor calculated under the CVSS is a constant and is not suitable for inclusion in a non-homogenous model. However, our "**Risk Factor**" model is based on the **Vulnerability Life Cycle** [2] [3] [10] [11] [12] [13] [14] and it is time dependent. This allows us to develop a non-homogeneous model which is our objective in this study. Therefore, in this study we will extend the **Transition Probability Matrix Model**, replacing vulnerability with the CVSS, "***v***" by its **Risk Factor** "***r***".

The probability of an exploitation for a given vulnerability can be obtained using the three stochastic models given in Table 1 below. These time dependent stochastic models were developed in our previous study [3], and we used the general classification of vulnerability risks based on the CVSS identified as **Low**, **Medium** and **High**. Details of the process and methodology in developing the subject models along with their validation accuracy were given in our previous study [3].

In each of the equations, t is the age of vulnerability and is calculated by taking the difference between the dates that the vulnerability was first discovered and the attacking attempt started.

Thus, for a given vulnerability at a time we can obtain the probability of being exploited. We can now define the transition probability as follows.

$$p_{ij} = \frac{R\left(v_j(t)\right)}{\sum_{j=1}^{n} R\left(v_j(t)\right)}$$

$(v_j(t))$ = Risk Factor of a given vulnerability in state j at time t,

$e(v_j)$ = Exploitability sub score that is related to the **CVSS** score for the given vulnerability in state j.

And

$$R\left(v_j(t)\right) = Y(t) * e\left(v_j\right)$$

Table 1. Model equations of risk factors for three different categories of vulnerabilities.

Category	Model Equation	R^2	R^2_{adj}
Low (0 - 4)	$Y(t) = 0.135441 - 0.308532\,(1/t) - 0.002030\ln(\ln t)$	0.9576	0.9566
Medium (4 - 7)	$Y(t) = 0.169518 - 0.356821(1/t) - 0.007011\ln(\ln t)$	0.962	0.961
High (7 - 10)	$Y(t) = 0.191701 - 0.383521\,(1/t) - 0.00358\ln(\ln t)$	0.9588	0.9577

is the analytic form of the risk factor as a function of $Y(t)$ and $e(v_j)$ where $Y(t)$ is the exploitability probability factor as a function of time and $e(v_j)$ is the exploitability score taken from the CVSS.

2.3. Attack Prediction

Under the **Attack Prediction**, we consider two methods to predict the attacker's behavior.

2.3.1. Multi Step Attack Prediction

The absorbing transition probability matrix [15] shows the presence of each edge in a network attack graph. This matrix shows every possible single-step attack. In other words, the absorbing transition probability matrix shows attackers reachability within one attack step. We can navigate the absorbing transition probability matrix by iteratively matching rows and columns to follow multiple attack steps, and also raise the absorbing transition probability matrix to higher powers, which shows multi-step attacker reachability at a glance.

For a square $(n \times n)$ adjacency matrix P and a positive integer k, P^k is matrix P raised to the power of k. Since P is an absorbing transition probability matrix with respect to time, this matrix goes to some stationary matrix Π, where the rows of this matrix are identical as follows. That is,

$$\lim_{k\to\infty} P^k = \Pi$$

Once the stationarity is achieved, goal state column of this matrix Π has ones, so we can find the minimum number of steps (time) that the attacker will reach the goal state with probability 1. Once the attacker is in the goal state we can identify the probability of the system being exploited.

2.3.2. Prediction of Expected Path Length (EPL)

The **Expected Path Length** (EPL) measures the expected number of steps the attacker will need starting from the initial state to reach the goal state (the attacker's objective). As we discussed earlier P has the following canonical form,

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$$

Here, P is the transition matrix, Q is the matrix of transient states, R is the matrix of absorbing states and I is the identity matrix.

The matrix P represents the transition probability matrix of the absorbing Markov chain. In an absorbing Markov chain the probability that the chain will be absorbed is always 1. Thus, we have

$$Q^n \to 0 \text{ as } n \to \infty$$

This property implies that all the eigenvalues of Q have absolute values strictly less than 1. Thus, $I - Q$ is an invertible matrix and there is no problem in defining the matrix

$$M = (I - Q)^{-1} = I + Q + Q^2 + Q^3 + \cdots$$

Using this fundamental matrix M of the absorbing Markov chain we can compute the expected total number of steps to reach the goal state until absorption.

Taking the summation of the first row elements of matrix M gives us the expected total number of steps to reach the goal state which is defined as the Expected Path Length.

Given below is an application that illustrates a computer network system of our proposed analytic process to estimate the EPL of a hacker.

3. Attack Graph and Attack Risk Evaluation

In this section we present an example illustrating the application of the usefulness of our method. We combine the application of methodology with an attack graph relevant to a typical network exemplified with three different recorded vulnerabilities.

3.1. Application: The Attacker

To illustrate the proposed analytical approach model that we have developed as discussed above, we considered the **Network Topology** [1] [16]-[21], given by **Figure 1** below.

The computer network consists of two service hosts IP 1, IP 2 and an attacker's workstation, Attacker connecting to each of the servers via a central router.

In the server IP 1 the vulnerability is labeled as CVE 2016-3230 and shall be denoted as V_1.

In the server IP 2 there are two recognized vulnerabilities, which are labeled CVE 2016-2832 and CVE 2016-0911. Let's denote them as V_2 and V_3, respectively.

We proceed to use the **CVSS** score of the above vulnerabilities in our analysis. The exploitability score (e (v) in **Figure 1**) of each vulnerability is given in **Table 2** below.

Published date is in general considered as the date that a vulnerability is made known to the public. CVSS score is the score given to the vulnerability based on exploitability factors by the "Forum of Incident Response and Security Teams", (FIRST). Calculation of this score is established and updated time to time

Table 2. Vulnerability scores.

Vulnerability	Published date	CVSS score	Exploitability score	Time for the date 6/24/2016 (t_j)	Risk factor $R(v_i(t_j))$
V_1 (CVE 2016-3230)	6/15/2016	9 (High)	8	9	1.702
V_2 (CVE 2016-2832)	6/13/2016	4.3 (medium)	2.8	11	0.3667
V_3 (CVE 2016-0911)	6/19/2016	1.9 (Low)	3.4	5	0.2474

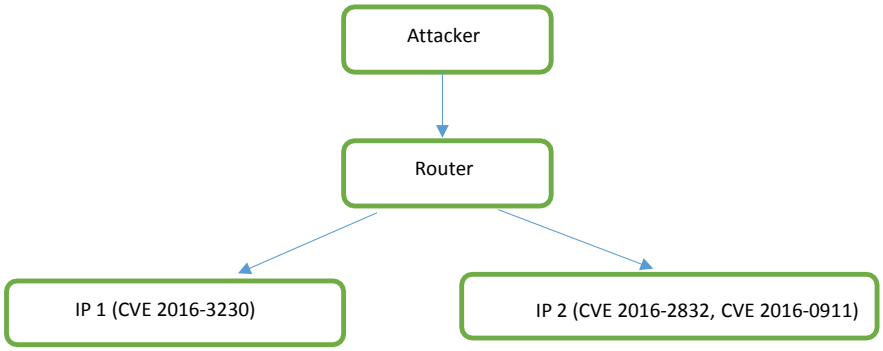

Figure 1. Network topology.

and the relevant details are available in the CVE detail and other relevant official websites.

June 24th was used as the date where a first attack attempt was made by an attacker. Risk factor is hence the Risk of being exploited on the 24th of June, calculated using the equation presented in the Section 2.2. That is,

$$\left(v_j(t)\right) = Y(t) * e\left(v_j\right)$$

For example, let's consider the vulnerability "V1 (CVE 2016-3230)". The CVSS score has given the exploitability score for this vulnerability as 8. Taking the difference between the published date (June 15th) and the attack date (June 24th), the age of this vulnerability is calculated as 9 days. Since this is a vulnerability of the category "High", we can now use our model given in the **Table 1** and calculate the "Risk Factor" as follows.

$$R\left(v_1(t)\right) = \left[0.191701 - 0.383521\left(\frac{1}{t}\right) - 0.00358\ln\left(\ln t\right)\right] * 8$$

$$R\left(v_1(9)\right) = 1.702$$

Similarly, Risk factors for two other vulnerabilities are also calculated and presented in the **Table 2** below.

3.2. Host Centric Attack Graph

The host centric attack graph is shown by **Figure 2**, below. Here, we consider that the attacker can reach the goal state only by exploiting V_3 vulnerability. The graph shows all the possible paths that the attacker can follow to reach the goal state.

Note that IP1.1 state represents V_1 vulnerability and IP2.1 and IP2.2 states represent vulnerabilities V_2 and V_3 respectively. Attacker can reach each state by exploiting the relevant Vulnerability.

3.3. Adjacency Matrix for the Attack Graph

In this section we will illustrate the process of developing Adjacency Matrix for the Attack Graph. Adjacency Matrix is a key analytical tool used in out methodology.

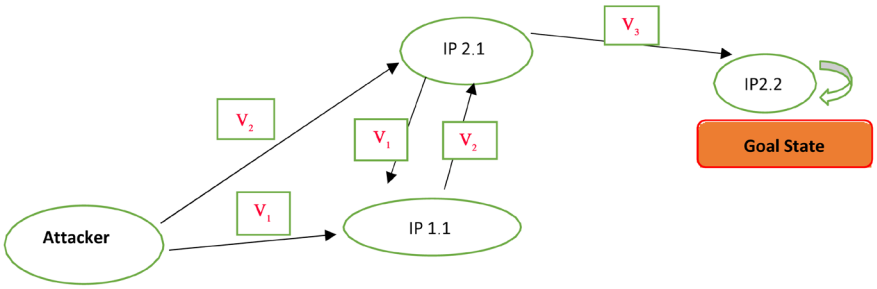

Figure 2. Host centric attack graph.

Let s_1, s_2, s_3, s_4, represent the attack states for Attacker, (IP1.1), (IP2.1) and (IP2.2), respectively.

To find the weighted value of exploiting each vulnerability from one state to another state, we divide the vulnerability score by summation of all out going vulnerability values from that state.

For our attack graph the weighted value of exploiting each vulnerability is given below. 1st row probabilities:

Weighted value of exploiting V_1 from s_1 to s_2 is $R_1/(R_1 + R_2)$ Weighted value of exploiting V_2 from s_1 to s_3 is $R_2/(R_1 + R_2)$ 2nd row probabilities:

Weighted value of exploiting V_2 from s_2 to s_3 is $R_2/(R_2)$ 3rd row probabilities:

Weighted value of exploiting V_1 from s_3 to s_2 is $R_1/(R_1 + R_3)$ Weighted value of exploiting V_3 from s_3 to s_4 is $R_3/(R_1 + R_3)$ 4th row probabilities:

Weighted value of exploiting V_3 from s_4 to s_4 is 1.

For the Host Centric Attack graph we can have the Adjacency Matrix as follows.

Applying the information given in Table 1, the matrix A can be obtained as follows.

$$A = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{array} \begin{array}{c} \begin{array}{cccc} s_1 & s_2 & s_3 & s_4 \end{array} \\ \begin{bmatrix} 0 & 0.7614 & 0.2386 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0.8255 & 0 & 0.1745 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Here, 0.7614 is the probability that attacker exploits vulnerability V_1 in the first step, the step from s_1 to s_2. Similarly, we can explain 0.1745 as the probability that attacker exploits the vulnerability V_3 in the step s_2 to s_3 in his first attempt. Similarly, each probability represents the likelihood to exploit relevant vulnerability from one state to another state in the first attempt.

We can use this matrix to answer several important questions in cyber security analysis. First, using the Adjacency Matrix we expect to find the Expected Path Length. Then, we can analyze the behavior of Expected Path Length over the time.

To calculate the EPL over the time we follow the steps given below.

Step 1: Calculate the **"Risk Factor"** of each vulnerability on the date of the first attack assumed (June 24th in our application). That is, calculate the **"age"** of

each vulnerability by taking the difference between the published date and the 24th of June. And, substitute this value of "t" in relevant model equation given in the Table 1.

Step 2: Using those **"Risk Factors"**, develop the transition matrix "A" and calculate the EPL.

Step 3: Repeat the same process for all the following dates that we need to calculate the

Expected Path Length.

From Table 3 below, we can identify that the number of days a hacker will take to reach his goal of exploitability for the given computer network system we have structured.

For example, let's consider the 20th day. Under **step 1**, we calculate the **Risk factors** for V_1, V_2 and V_3. For the 20th day age of three vulnerabilities V_1, V_2 and V_3 are, $t_1 = 9 + 20$, $t_2 = 11 + 20$ and $t_3 = 5 + 20$, respectively. Then, by substituting these ages in the respective model equation from the Table 1 and multiplying the answers by respective exploitability score, we calculate three risk factors as follows.

V_1 is a vulnerability of "High" category. Therefore, we use the 3rd model equation from Table 1 and obtain the **Risk factor** as follows.

Substituting, $t = 29$, in the model,

$$\left(v_1(t)\right) = Y(t) * e(v_1)$$

we obtain,

$$R_1 = 0.191701 - 0.383521 \times (1/29) - 0.00358 \ln(\ln 29) \times 8 = 1.393$$

Similarly for V_2 and V_3 we obtain the following Risk factors calculated using the relevant model equations.

For, $t = 31$,

$$\left(v_2(t)\right) = Y(t) * e(v_2)$$

$$R_2 = 0.169518 - 0.356821 \times (1/31) - 0.007011 \ln(\ln 31) \times 2.8 = 0.4182$$

For, $t = 25$,

$$\left(v_3(t)\right) = Y(t) * e(v_3)$$

$$R_3 = 0.135441 - 0.308532 \times (1/25) - 0.002030 \ln(\ln 25) \times 3.4 = 0.4105$$

Once we have calculated the "Risk Factors" for all the vulnerabilities in the network system, the second step is to develop the Transition Matrix "A" as given in the Figure 3.

The transition probability Matrix for this system on the 20th day after the first attack attempt is assumed to be made is given below.

$$A = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{array} \begin{array}{c} \begin{array}{cccc} s_1 & s_2 & s_3 & s_4 \end{array} \\ \begin{bmatrix} 0 & 0.7691 & 0.2309 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0.7724 & 0 & 0.2276 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$A = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{bmatrix} \frac{0}{R1+R2} & \frac{R1}{R1+R2} & \frac{R2}{R1+R2} & \frac{0}{R1+R2} \\ \frac{0}{R2} & \frac{0}{R2} & \frac{R2}{R2} & \frac{0}{R2} \\ \frac{0}{R1+R3} & \frac{R1}{R1+R3} & \frac{0}{R1+R3} & \frac{R3}{R1+R3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Figure 3. Adjacency matrix (Transition probability matrix).

Step 3 is to calculate the EPL. Applying the methodology we explained in the Section 2.3.2. we can calculate the EPL using the transition matrix "A" by obtaining the matrix "M".

The sum of the first row of matrix "M" is the EPL of this computer network system at the 20th day (from June 24th) from the first assumed attack attempt. We have obtained, EPL = 9.567 for the 20th day after the first attack created as given in the **Table 3**.

Expected Path length

The **Table 3** below shows us the EPL for this computer system for 100 days starting from 24th of June.

Figure 4 below illustrates the results shown in the **Table 3**, graphically.

By examining the distribution of **Expected Path Length** of the attacker over 100 days, it will take fewer steps for an attacker to compromise the security goal as the age of vulnerabilities increases. Security practitioners in a typical organization can establish a threshold score for the system and the security teams can planned in advance and identify the critical points to establish a strategy to defend the security of the computer system and introduce relevant patches before we approach such critical stages.

In the present system, it is clear that the threshold score of the EPL is approximately 9.5 steps and the defending professionals can conclude that the system in their network is relatively safe from exploits only for the next 21 days as EPL score is above the threshold value.

It is also clear that any vulnerability that exists creates a threat to the computer system and the risk of probable exploitation will increase over the time of its existence without being patched. In other words, for a particular network system, a higher Expected Path Length for an attacker to reach a goal state represents more difficulty for the hacker and would be reasonable to assume that the attacker has to face many defending measures with a higher Expected Path Length compared to a smaller Expected Path Length. Now, using the probabilistic models that we have developed in our previous studies, using the Vulnerability Life Cycle approach [2] [3] enables us to develop a time dependent stochastic models so that we could extend their application to develop a relevant and well defined process of monitoring the behavior of threats. Thus, our proposed analytic process illustrates its capability of estimating a **Risk Index** as a function of the attacking time for a given computer system with known vulnerabilities.

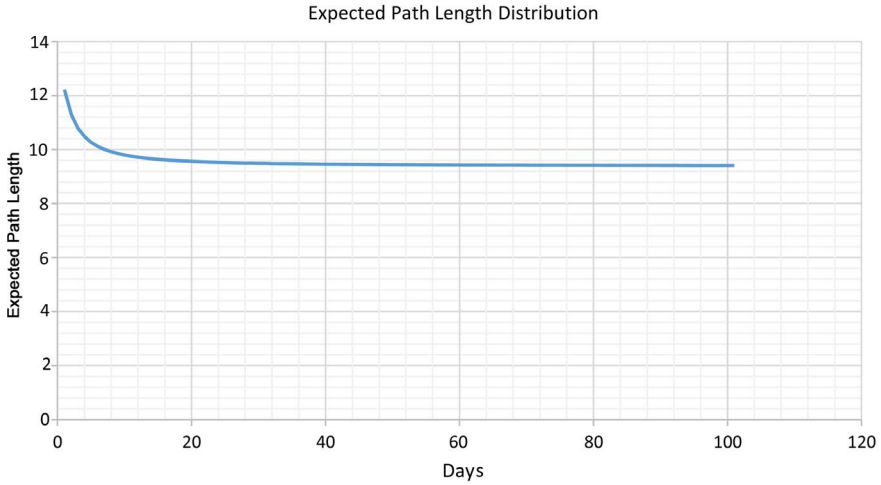

Figure 4. Behavior of expected path length over time.

Table 3. Expected path length relative to number of days after first attack.

Age (Days)	Expected Path Length	Age (Days)	Expected Path Length	Age (Days)	Expected Path Length	Age (Days)	Expected Path Length
1	12.2205398	26	9.517537	51	9.4453151	76	9.4239414
2	11.3052188	27	9.511655	52	9.4440137	77	9.4234008
3	10.7998722	28	9.506248	53	9.4427673	78	9.4228753
4	10.4850373	29	9.501261	54	9.4415727	79	9.4223643
5	10.2729754	30	9.49665	55	9.4404267	80	9.4218672
6	10.1220591	31	9.492376	56	9.4393265	81	9.4213834
7	10.0101501	32	9.488404	57	9.4382695	82	9.4209123
8	9.9244658	33	9.484705	58	9.4372532	83	9.4204536
9	9.8571518	34	9.481253	59	9.4362752	84	9.4200067
10	9.8031388	35	9.478024	60	9.4353336	85	9.4195711
11	9.7590231	36	9.474999	61	9.4344263	86	9.4191465
12	9.7224429	37	9.472158	62	9.4335516	87	9.4187324
13	9.6917134	38	9.469488	63	9.4327076	88	9.4183284
14	9.6656039	39	9.466972	64	9.4318929	89	9.4179342
15	9.643197	40	9.464599	65	9.431106	90	9.4175493
16	9.6237964	41	9.462358	66	9.4303454	91	9.4171736
17	9.606865	42	9.460237	67	9.4296099	92	9.4168066
18	9.5919829	43	9.458227	68	9.4288983	93	9.416448
19	9.5788176	44	9.456321	69	9.4282094	94	9.4160975
20	9.5671025	45	9.454511	70	9.4275421	95	9.415755
21	9.5566222	46	9.452789	71	9.4268956	96	9.41542
22	9.5472004	47	9.45115	72	9.4262687	97	9.4150924
23	9.5386921	48	9.449588	73	9.4256606	98	9.4147719
24	9.5309766	49	9.448098	74	9.4250706	99	9.4144583
25	9.5239531	50	9.446675	75	9.4244978	100	9.4141513

4. Conclusions

In the present study, we have developed a nonhomogeneous stochastic model for predicting the **Expected Path Length (EPL)** of a computer network system with a given set of vulnerabilities at time "t".

Knowing **EPL** as a function of time is extremely important in developing defending strategies for not being exploited. Such strategies will reduce the likelihood of the computer network system being hacked.

As we observe the behavior of the EPL over the time, it is possible to identify the time ranges where EPL reached a minimum. Small EPL implies higher chance for a hacker to be successful. In other words, a computer network system is more vulnerable to be exploited on the days where the EPL is the smallest. On such time "t", vulnerabilities and the system are hence more susceptible to be hacked. The same scenario from an attacker's point of view can be explained. That is, on the days where EPL is at its smallest, the likelihood of making a successful attack attempt is higher. Therefore, an attacker (hacker), who identifies the set of vulnerabilities in a given computer system would put more attempt on exploiting the system on such date where the EPL is at its smallest. This means that we can use this method as a prediction method of attacking (hacking) time.

By knowing this time for any computer network system, security engineers or IT architects can take the necessary actions in advance to protect their computer system.

Finally, we have developed our methodology based on a typical computer network system that exists in a real world situation with given vulnerabilities that identifies the **EPL** and actual time that the subject computer system could be exploited. Thus, industry can apply the developed methodology in their own computer network system with a given (known) vulnerabilities to predict the **EPL** and most probable time of being exploited.

References

[1] Kaluarachchi, P.K., Tsokos, C.P. and Rajasooriya, S.M. (2016) Cybersecurity: A Statistical Predictive Model for the Expected Path Length. *Journal of information Security*, **7**, 112-128. https://doi.org/10.4236/jis.2016.73008

[2] Rajasooriya, S.M., Tsokos, C.P. and Kaluarachchi, P.K. (2016) Stochastic Modelling of Vulnerability Life Cycle and Security Risk Evaluation. *Journal of information Security*, **7**, 269-279. https://doi.org/10.4236/jis.2016.74022

[3] Rajasooriya, S.M., Tsokos, C.P. and Kaluarachchi, P.K. (2017) Cybersecurity: Nonlinear Stochastic models for Predicting the Exploitability. *Journal of information Security*, **8**, 125-140. https://doi.org/10.4236/jis.2017.82009

[4] 2016 U.S Government Cybersecurity Report. https://cdn2.hubspot.net/hubfs/533449/SecurityScorecard_2016_Govt_Cybersecurity_Report.pdf

[5] Symantec, Internet Security Threat Report 2016-Volume 21. https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

[6] NVD, National Vulnerability Database. http://nvd.nist.gov/

[7] Kijsanayothin, P. (2010) Network Security Modeling with Intelligent and Complexity Analysis. Ph.D. Dissertation, Texas Tech University, Lubbock, Texas, U.S.

[8] Schiffman, M. Common Vulnerability Scoring System (CVSS). http://www.first.org/cvss/

[9] CVE Details. http://www.cvedetails.com/

[10] Frei, S. (2009) Security Econometrics: The Dynamics of (IN) Security, Ph.D. Dissertation at ETH Zurich.

[11] Joh, H. and Malaiya, Y.K. (2010) A Framework for Software Security Risk Evaluation Using the Vulnerability Lifecycle and CVSS Metrics, *Proc. International Workshop on Risk and Trust in Extended Enterprises*, November 2010, 430-434.

[12] Alhazmi, O.H., Malaiya, Y.K. and Ray, I. (2007) Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems. *Computers and Security Journal*, **26**, 219-228. https://doi.org/10.1016/j.cose.2006.10.002

[13] Alhazmi, O.H. and Malaiya, Y.K. (2008) Application of Vulnerability Discovery Models to Major Operating Systems. *IEEE Transactions on Reliability*, **57**, 14-22. https://doi.org/10.1109/TR.2008.916872

[14] Alhazmi, O.H. and Malaiya, Y.K. (2005) Modeling the Vulnerability Discovery Process. *Proceedings of* 16*th International Symposium on Software Reliability Engineering*, Chicago, 8-11 November 2005, 129-138. https://doi.org/10.1109/ISSRE.2005.30

[15] Lawler, G.F. (2006) Introduction to Stochastic processes. 2nd Edition, Chapman and Hall/CRC Taylor and Francis Group, London, New York.

[16] Noel, S., Jacobs, M., Kalapa, P. and Jajodia, S. (2005) Multiple Coordinated Views for Network Attack Graphs. *Proceedings of the IEEE Workshops on Visualization for Computer Security*, Minneapolis, October 2005, 99-106.

[17] Mehta, V., Bartzis, C., Zhu, H., Clarke, E.M. and Wing, J.M. (2006) Ranking Attack Graphs. In: Zamboni, D. and Krugel, C., Eds., *Recent Advances in Intrusion Detection*, Volume 4219 of Lecture Notes in Computer Science, Springer, Berlin, 127-144.

[18] Abraham, S. and Nair, S. (2014) Cyber Security Analytics: A Stochastic Model for Security Quantification using Absorbing Markov Chains. *Journal of Communications*, **9**, 899-907. https://doi.org/10.12720/jcm.9.12.899-907

[19] Jajodia, S. and Noel, S. (2005) Advanced Cyber Attack Modeling, Analysis, and Visualization. 14th USENIX Security Symposium, Technical Report 2010, George Mason University, Fairfax.

[20] Wang, L., Singhal, A. and Jajodia, S. (2007) Measuring Overall Security of Network Configurations using Attack Graphs. *Data and Applications Security*, **21**, 98-112. https://doi.org/10.1007/978-3-540-73538-0_9

[21] Wang, L., Islam, T., Long, T., Singhal, A. and Jajodia, S. (2008) An Attack Graph-Based Probabilistic Security Metric. DAS 2008, LNCS 5094, 283-296.