Urban Studies and Public Administration
Vol. 2, No. 1, 2019
www.scholink.org/ojs/index.php/uspa
ISSN 2576-1986 (Print) ISSN 2576-1994 (Online)

Original Paper

Risk Rank Analysis Method for Vulnerabilities in a Network System

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] Department of Mathematics and Statistics, University Of South Florida, Tampa, Florida, USA

[3] Department of Mathematics, University Of Dayton, Dayton, Ohio, USA

[*] Pubudu Kalpani Kaluarachchi, Department of Mathematical and Physical Sciences, Miami University, Middletown, Ohio, USA

Received: November 8, 2018 Accepted: November 30, 2018 Online Published: January 10, 2019

doi:10.22158/uspa.v2n1p22 URL: http://dx.doi.org/10.22158/uspa.v2n1p22

Abstract

Prioritizing on most critical weaknesses in a network system at the correct time is a very important role in network security administration. Due to the complexity and high unpredictability of exploitations it is hard to decide which vulnerabilities and which IP s are at the highest risk at a particular time. Present study proposes a new methodology that enables network administrators to rank vulnerabilities based on the probability of being exploited at a given time using the Markovian process. Markovian process allows us to iterate a transition probability matrix for a network system consisting identified or discovered vulnerabilities. This process result in a steady state with probabilities that a vulnerability will be exploited. Similar approach is used here to develop a risk rank model. Well known Google Page Rank Algorithm also uses a similar approach in estimating the probability of a web surfer reaching a particular webpage. Same concept can be used with several modifications to estimate and rank the risk level of each vulnerability in a network system. New methodology is presented with an example of a small network model with three vulnerabilities.

Keywords

Markov chain, vulnerability, network security, google page rank, risk

1. Introduction

A Network systems could have numerous vulnerabilities. We understand the process of generating vulnerabilities is highly stochastic and outcomes are hard to predict. Similarly the behavior of attacks and attackers also have higher level unpredictability. When considering a particular system based on

the discovered vulnerabilities the analysis must consider the dynamic nature of the effect of vulnerabilities over time. As we observed in our previous studies (Kaluarachchi, Tsokos, & Rajasooriya, 2016), the effect of the vulnerabilities vary with the time through their life cycle. Therefore, for a particular system, the most threatening vulnerability (2016 U.S Government Cybersecurity report and NVD Data base) at time t_1 might not be the same at time t_2. Hence, it would be very useful to have analytical models to observe the behavior of the rank of vulnerabilities based on the magnitude of the threat with respect to time for a given network system.

Such ranking distribution over time would empower the defenders by giving the priority directions to attend on fixing vulnerabilities. In this paper we attempt to address this need.

2. Methodology

2.1 Google Page Rank Algorithm

This section provides a background for our quantitative analysis of Risk Rank Algorithm method. Ranking web pages is an important function of an internet search engine (Kijsanayothin, 2010; Sawilla, 2007). Google Page Rank Algorithm (Gleich, 2015) is one of the most accurate and efficient page ranking methods in use. Methodology behind this algorithm will be briefly discussed below.

Output of this algorithm gives a probability distribution which is used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. Using this method we can rank the likelihood of clicking on any web link. This can be calculated for any number of web links. In this algorithm, the sum of the page rank values of all the considered web links is equal to be one and it is assumed that the probability of selecting a web page initially is equal for any available option.

Google page rank algorithm simulates the clicking behavior of a web link in two ways. First is to visit a web link via an incoming link to the current web page and second way is to pick a web page randomly. Google page rank theory holds that an any surfer who is randomly clicking on web links will eventually stop clicking. At any of these stages, a damping factor d is the probability that the web surfer will continue surfing. Many researchers have tested various damping factors but in generally it is assumed that the damping factor will be set around 0.85.

Let $p_t(v)$ be the probability of visiting web page v at time t and V be a set of all web pages under consideration. Here $out(v)$ represents the set of web pages in v with an outgoing link from v, and $in(v)$ represents the set of incoming link to v. The page rank computation can be viewed as a Markov process whose states are pages and the links between pages represent state transitions (Kijsanayothin, 2010). This computation is given in the Equation (1) below.

$$P_{t+1} = (1-d) * \sum_{\forall u \in V} \frac{P_t(u)}{|V|} + d * \sum_{\forall u \in in(v)} \frac{P_t(u)}{|out(u)|} \quad (1)$$

Let, $|V|$ be the number of pages considered. Surfer will stop clicking on any link with probability $1-d$. Since there are $|V|$ number of pages and probability of visiting v from any page is equally likely, the probability for each case is equal to $\frac{1}{|V|}$.

Here $d * \sum_{\forall u \in in(v)} \frac{P_t(u)}{|out(u)|}$ represents the case when the surfer continues clicking links with probability d and goes to page v at time $t + 1$ from page u that has an incoming link to v.

Initially at $t = 0$, each page has the same ranking value probability which is equal to $\frac{1}{|V|}$. Then iterations are executed over time until the stability is achieved. Once the probability distribution for each page becomes stable, considering high to low probabilities ranks are assigned.

3. Risk Rank Algorithm

By developing the concept applied in Google Page Rank Algorithm here we introduce a ranking method for risk of vulnerabilities (Frei, 2009) in a network system.

To estimate the probabilities in Risk Rank Algorithm Markov model techniques can be applied similarly as in Google Page Rank Algorithm (Gleich, 2015). However there is a difference between web surfing behavior and Cyber attacking behavior. A web surfing user can randomly select a web page but in cyberattacks an attacker doesn't have the same freedom. In web surfing user can arrive at any web page in one single step by using it's URL. But attacker has many restrictions. In computer net-work system an attacker doesn't have the access to all vulnerabilities in the network system. To achieve attacker's target state he must exploits several vulnerabilities in a particular order and enters in to the target system.

In the attacking process an attacker has two options. He can either continue or quit from his current path. If it is too difficult for him to achieve his goal state he can quit on the current path and try an alternative path by starting over from one of the set of initial states. Base on these assumptions here we propose our model to calculate the probability distribution of a given security attack model.

To obtain the risk rank (Alhazmi & Joh, n.d.) we used the risk factor $R(v(t))$ (Rajasooriya, n.d.) of each vulnerability at each state and calculated normalized risk factor matrix $A(V, R)$ for the attack net-work system by using $\varphi(u, v)$ transition probabilities from state u to v. Thus, we can calculate transition probabilities using the equation $\varphi(u, v) = \frac{R(v(t))}{\sum_{\forall w \in out(u)} R(w(t))}$

Let $P_k(v)$ be the probability of exploiting state v at time k and V be a set of all states under consideration. Here $out(v)$ represents the set of states in V with an outgoing link from v, and $in(v)$ represents the set of incoming link to v. The Risk rank computation can be viewed as a Markov process (Lawler, 2006; Abraham, 2014) whose state are vulnerabilities and the links between vulnerabilities represent state transitions. This computation is given in the Equation (2) below.

$$P_{k+1}(v) = \begin{cases} d * \sum_{\forall u \in in(v)} P_k(u) * \varphi(u,v) + \frac{(1-d)}{|I|}, & if\ v\ is\ an\ intial\ state \\ d * \sum_{\forall u \in in(v)} P_k(u) * \varphi(u,v), & if\ v\ is\ not\ an\ intial\ state \end{cases} \quad (2)$$

Let $|I|$ be the number of initial states and attacker will stop his current path with probability $1-d$. Since there are $|V|$ number of states and probability of exploiting v from any other state is equally likely, the probability for each case is $\frac{1}{|V|}$.

Here in equation (2) $d * \sum_{\forall u \in in(v)} P_k(u) * \varphi(u,v)$ represents the case when the attacker continues his current path with probability d and attack to state v at time $t+1$ from state u that has an incoming link to vulnerability v.

Initially at t=0 each state has the same ranking value which is equal to $\frac{1}{|V|}$. Then computing iterations over time once the stability achieved. Once the probability distribution for each state become stable, assigning rank to each vulnerability (Noel et al., 2005; Mehta et al., 2006).

Algorithm 1 Risk Rank Algorithm

1: **procedure** Risk Rank Algorithm (Input)

2: Determine the input values.

3: $A(V, R)$= Probability transition matrix, V - Set of states in the attack graph, $R(v(t))$-Risk factor of vulnerability v at time t, $\varphi(u, v)$- transition probability from state u to v, I - Set of initial states. $out(v)$ - Set of all states out going from state v,

$|V|$- Number of states.

4: begin each $v \in V$

5: set $P_o(v) = \frac{1}{|V|}$

6: for $\forall u \in out(v)$ do

7: $\varphi(u,v) = \frac{R(v(t))}{\sum_{\forall w \in out(u)} R(w(t))}$

8: end

9: for $k=k+1$

10: $P_{k+1}(v) = d * \sum_{\forall u \in in(v)} P_k(u) * \varphi(u,v)$

11: if $v \in I$ then

12: $P_{k+1}(v) = P_{k+1}(v) + \frac{(1-d)}{|I|}$

13: end if

14: end for

15: break if $P_{k+1}(v)=P_k(v)$, $\forall v \in V$

16: **end procedure**

This procedure is illustrated by the following schematic network.

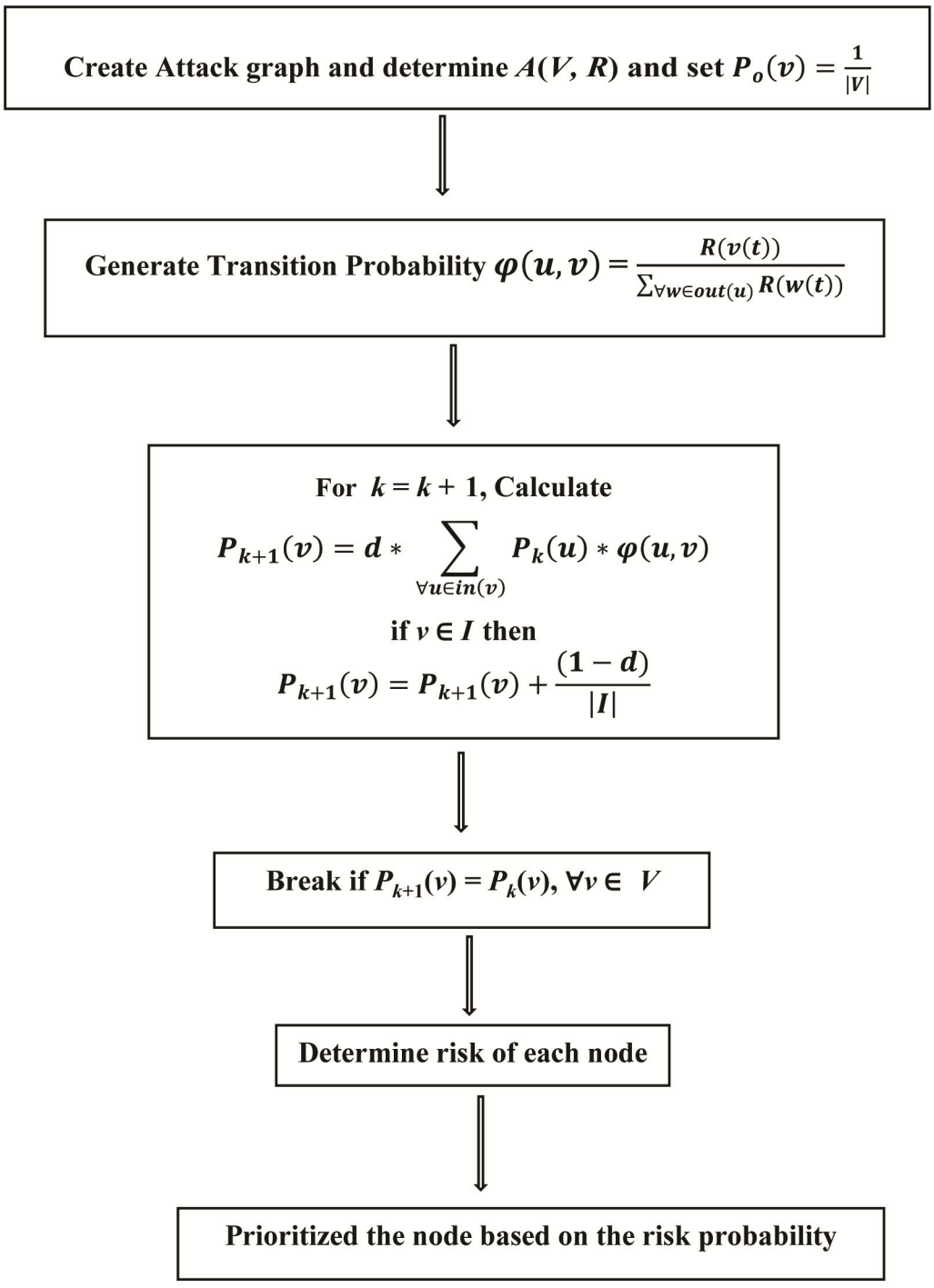

Figure 1. Key Steps of the Risk Rank Algorithm

4. Illustration of Applying of the Risk Rank Algorithm

To illustrate the proposed analytical approach model that we have developed as discussed above, we considered a Network Topology (Kaluarachchi et al., 2016), given by Figure 2, below.

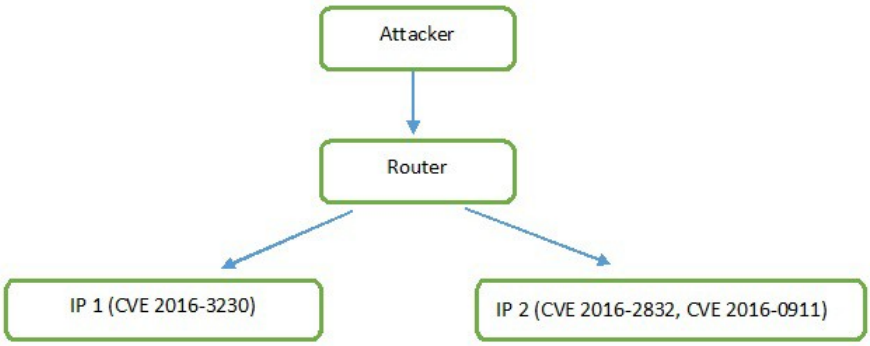

Figure 2. Network Topology

The computer network consists of two service hosts IP 1, IP 2 and an attacker's workstation. Attacker is 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 as CVE 2016-2832 and CVE 2016-0911. Let's denote them as V_2 and V_3, respectively.

We proceed using the CVSS score [11], [12] of the above vulnerabilities in our analysis.

The exploitability score ($e(vj)$) and Risk factor of each vulnerability are given in Table 1, below.

Table 1. Vulnerability Scores

Vulnerability	Published date	CVSS score	$e(vj)$	(tj)	$R(vj(t))$
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

For example we can calculate the Risk Factor of V_1 as follows.

$R(vj(t))=Y(t) \times e(vj)$,

$R(v1(t))=[0.1917010.383521(1/t)-0.00358 \ln(\ln(t))] \times 8$,

and

$R(v1(9))=1.702$.

Although the method we propose can be applied to any form of network system, for simplicity we will use our host centric attack graph model (Kaluarachchi et al., 2016; Mehta et al., 2006) introduced below to illustrate the process. The host centric attack graph is shown by Figure 3, below. Here, we consider that the attacker can reach the goal state only by exploiting Vulnerability $V3$. The graph shows all the possible paths that is available for the attacker to reach the goal state.

Note that IP 1, 1 state represents vulnerability $V1$ and states IP 2,1 and IP 2,2 represent vulnerabilities $V2$ and $V3$ respectively. The attacker can reach each state by exploiting the relevant Vulnerability.

With this methodology for the Host Centric Attack graph (Wang et al., 2007), we can obtain the Adjacency Matrix as follows. Applying the information given in Table 1, the matrix $A(V, R)$ is derived. The matrix the transition probabilities from one state to another state.

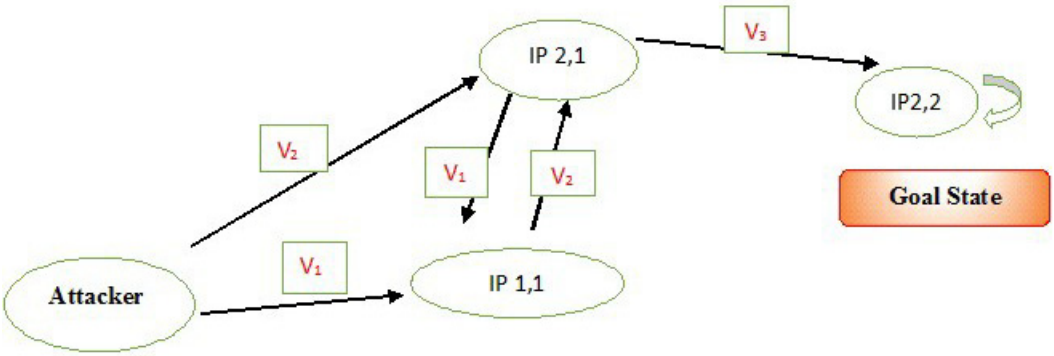

Figure 3. Host Centric Attack Graph

$$A = \begin{bmatrix} 0 & .76 & .24 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & .83 & 0 & .17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Applying this normalized risk matrix into Algorithm 1, we can obtain steady state probabilities for each state in the network which represent risk of being exploited (Frei, 2009; Joh & Malaiya, 2010). Results we obtained for each state are shown in the Table 2.

Table 2 results are in the order of being exploited by attacker at time t. Order of vulnerabilities based on the rank we obtained is $s1$, $s2$, $s3$, $s0$.

Table 2. Ranking Results in Attack States

States	Rank probability	Rank
$s0$	0.15	4
$s1$	0.293669	1
$s2$	0.279731	2
$s3$	0.2766	3

This result suggests that $s1$ has the highest likelihood of being attacked. This means at time t, $s1$ is the most vulnerable state. However according to Table 1 risk factor values for vulnerability $v1$ is 1.702 which is higher than the risk factor values of vulnerabilities $v2$ and $v3$. Therefore it is reasonable to assume that reaching state $s1$ from initial state $s0$ (attacker's state) by exploiting $v1$ vulnerability is easier than reaching states $s2$ and $s3$. Therefore, the risk rank of the state $s1$ is higher than other states.

5. Conclusion and Future Works

In this paper a new Ranking Algorithm was introduced to rank the vulnerabilities in a particular computer network system. The methodology of well-known Google Page Rank Algorithm (Gleich, 2015) was used and we further developed it to fit a computer network environment. General assumptions used in Google Page Rank Algorithm with respect to the probability of selecting a particular web link were changed according to the probability distributions we obtained by normalized vulnerability scores in subject computer network system. Ranks were obtained for each vulnerability based on the likelihood of those vulnerabilities getting exploited (Rajasooriya, Tsokos, & Kaluarachchi, 2016). This new methodology will greatly help relevant parties to make better decisions to protect network systems because at a particular network system, the algorithm will indicate which vulnerabilities are most vulnerable and needed immediate attention or priority.

In future we are planning to develop the algorithm so that the Distribution of Ranks of Vulnerabilities in the subject computer network system is given as a function of time. That is, using our new algorithm, a user (a network system administrator or a researcher) would be able to observe the behavior of the ranks of vulnerabilities with respect to time.

References

Abraham, S., & 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

Alhazmi, O. H., Malaiya, Y. K., & Ray, I. (2007). Measuring, analyzing and predicting security vulnerabilities in software systems. *Computers and Security Journal*, *26*(3), 219-228. https://doi.org/10.1016/j.cose.2006.10.002

Alhazmi, O. H., & Malaiya, Y. K. (2008). Application of Vulnerability Discovery Models to Major Operating Systems. *IEEE Transactions on Reliability*, *57*(1), 14-22. https://doi.org/10.1109/TR.2008.916872

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

CVE details. (n.d.). http://www.cvedetails.com/

Frei, S. (2009). *Security Econometrics: The Dynamics of (IN) Security* (Ph.D. dissertation at ETH Zurich).

Gleich, D. F. (January 2015). PageRank Beyond the Web. *SIAM Review*, *57*(3), 321-363. https://doi.org/10.1137/140976649

Jajodia, S., & Noel, S. (2005). Advanced Cyber Attack Modeling, Analysis, and Visualization, 14th USENIX Security Symposium. In *Technical Report*. George Mason University, Fairfax, VA.

Joh, H., & 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.

Kaluarachchi, P. K., Tsokos, C. P., & 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

Kaluarachchi, P. K., Tsokos, C. P., & 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

Kijsanayothin, P. (2010). *Network Security Modeling with Intelligent and Complexity Analysis* (Ph.D. Dissertation). Texas Tech University.

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

Mehta, V., Bartzis, C., Zhu, H., Clarke, E. M., & Wing, J. M. (2006). Ranking attack graphs. In D. Zamboni, & C. Krugel (Eds.), *Recent Advances in Intrusion Detection* (Vol. 4219 of Lecture Notes in Computer Science, pp. 127-144). Springer. https://doi.org/10.1007/11856214_7

Noel, S., Jacobs, M., Kalapa, P., & Jajodia, S. (2005). Multiple Coordinated Views for Network Attack Graphs. In *VIZSEC'05: Proc. of the IEEE Workshops on Visualization for Computer Security* (pp. 99-106). Minneapolis, MN.

National vulnerability database (NVD). (n.d.). Retrieved from http://nvd.nist.gov/

Rajasooriya, S. M., Tsokos, C. P., & 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

Rajasooriya, S. M., Tsokos, C. P., & 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

Sawilla, R., & X. Ou. (2007). Googling Attack Graphs. In *Technical Report TM-2007-205, Defense Research and Devel-opment Canada, September 2007*.

Symantec, Internet security threat report. (2016). *Symantec, Internet security threat report* (Vol. 21). Retrieved from https://resource.elq.symantec.com

Schiffman, M. (n.d.). *Common Vulnerability Scoring System (CVSS)*. Retrieved from http://www.first.org/cvss/

2016 U.S Government Cybersecurity report. (2016). Retrieved from https://cdn2.hubspot.net/hubfs/533449/SecurityScorecard_2016_Govt_Cybersecurity_Report

Wang, L., Singhal, A., & Jajodia, S. (2007). Measuring overall security of network configurations using attack graphs. *Data and Applications Security XXI*, *4602*, 98-112. https://doi.org/10.1007/978-3-540-73538-0_9

Wang, L., Islam, T., Long, T., Singhal, A., & Jajodia, S. (2008). *An attack graph-based probabilistic security metric* (DAS 2008, LNCS 5094, pp. 283-296).