SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 4, ISSUE 2
ISSN: 2054 -7420

Topological Analysis of Network Systems for Intrusion Detections

[1]Rohitha Goonatilake and [2]Susantha Herath
[1]Texas A&M International University, TX 78041, USA
[2]St. Cloud State University, MN 56301, USA
haragr@gmail.com; harag@tamiu.edu

ABSTRACT

An understanding of how well networks will respond to ongoing attack threats is an important task in formulating strategies to protect unauthorized network activities. The study of topological properties of network architecture sheds some light in this effort. The purpose of this paper is to study several scenarios that address topological structures and related analyses of network systems to begin the appropriate discussion towards this question. Analysis of the probabilistic state finite automation and its probability distribution theory play a pivotal role in the discussion.

Keywords: Intrusion, conditional probability, network system, regression, data analysis

1 Preliminaries

There is an urgency to harden the software used in network systems as increasing incidents of network intrusions have been reported. In addition, hardening software in general is designed based on the topological construction of the network systems. Malicious incidents on the Internet as reported to CERT (Computer Emergency Response Team) provide evidence that these attacks increased exponentially from 6 incidents reported in 1988 to 314,246 incidents reported in 2011 [1 & 2]. It appears that these incidents were not widely reported for general public since then. Furthermore, from 1988 to 2003, incidents exhibited exponential growth; after that the number of incidents increased sharply even with the increased resistances due to precautionary protective measures in place, according to Figure 1 [3].

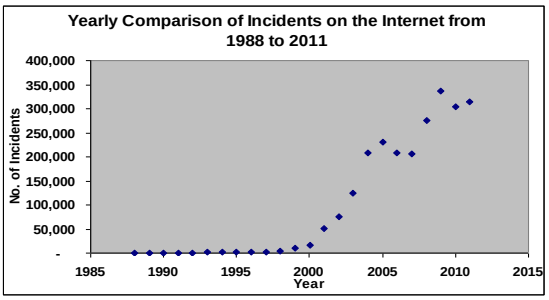


Figure 1. Exponential Growth of Malicious Incidents on the Internet

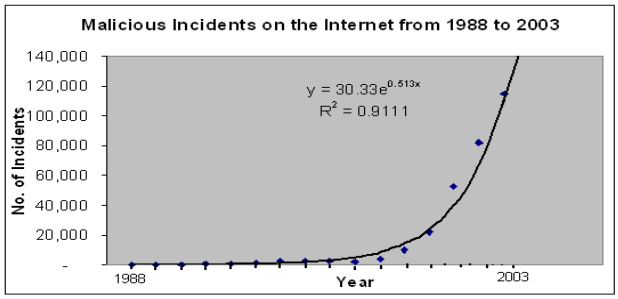


Figure 2. Exponential Growth of Malicious Incidents on the Internet

DOI: 10.14738/tnc.42.1989
Publication Date: 20th April 2016
URL: http://dx.doi.org/10.14738/tnc.42.1989

Figure 2 exhibits exponential growth of increasing malicious incidents on the Internet as plotted using the data in [4]. Moreover, in the recent past, these reported incidents grew significantly and have caused enormous disruptions for our banking and trading sectors causing unprecedented financial losses and privacy concerns.

Three aspects of the theory that are distinguished for analysis pertain to (a). the formal logic content, (b). the intuitive background, and (c). the applications. The character and applications of the structure as a whole cannot be appreciated without considering all three aspects in their relations for network traffic anomaly detections. Probability theory, on the other hand, is the mathematical theory of random (nondeterministic) phenomena. The probability distribution is derived by beginning with a statistical model, a set of assumptions about how responses are generated, and the calculations of associated probabilities. However, intrusion detection systems have widely been based on the characterization of an attack [5]. The primary focus in this effort will be the tracing of activities on the network to see if they match the known characterization. In addition, there are also a few flow-level detection schemes available [6]. Recently, intrusion detection systems based on previous known system data have appeared in the literature. In brief, the purpose is to inspect the network activities for suspicious activities that may indicate a system attack or an occurrence of misuse by unscrupulous network users [7]. An effective ISD logs actions executed by users or processes for investigation, alerts the system administrator when the activities are indicative of an attempted intrusion, and if appropriate, takes corrective measures such as expelling the intruder [8]. These vulnerabilities and bugs of information systems are often exploited by the intrusions. The extent of all possible scenarios for a network resulting from an intrusion can vary from none to an actual intrusion as depicted and appropriately color corded in Table 1. A unique obstacle eclipse effect in obstructed barriers has been observed that generates a sensor movement strategically performing definite obstructed barriers prevent intrusions [9].

Table 1. Vulnerability: Network Hardware vs. Outside Threats

Extent of intrusion	Network hardening					
		None	Mild	Reasonable	High	Very High
	None					
	Slight					
	Moderate					
	Extreme					
	Severe					

Optimal design of network topologies in multi-agent systems to facilitate effective communication on the network system is posed in the associated cost factors and the efficiency of performance [10]. This depends on the extent of network hardening software and severity of instruction in each component. The network vulnerability needs to be addressed in terms of the extent of intrusion, ranging from slight to extreme, and of network hardening which varies from mild to very high. Vulnerable devices, applications, and network software on an organization's network pose a great risk to the organization. The

Copyright © Society for Science and Education United Kingdom

determination of probability of vulnerability under these scenarios can be calculated using the conditional probabilities such as

$$\Pr(\text{Hardening} \mid \text{Volunerabi lty}) = \frac{\Pr(\text{Hardening and Volunerabi lty})}{\Pr(\text{Volunerabilty})} \quad \text{and}$$

$$\Pr(\text{Volunerabi lty} \mid \text{Hardening}) = \frac{\Pr(\text{Volunerabi lty and Hardening})}{\Pr(\text{Hardening})}.$$

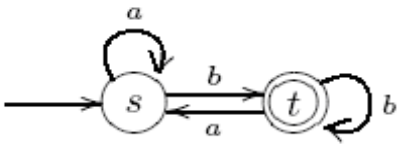
These calculations determine other vulnerabilities under consideration in the areas of threat detection and vulnerability analysis. Other scenario can also be considered using Theorem of Total Probability and Bayes' Theorem, respectively. Let $\{B_1, B_2, \cdots, B_n\}$ be a set of nonempty subsets of the sample space *S* of an experiment. If the events $B_1, B_2, \cdots, B_n$ are mutually exclusive and $B_1 \cup B_2 \cup \cdots \cup B_n = S$. For, a partition of *S*, $\{B_1, B_2, \cdots, B_n\}$: Theorem of Total Probability concludes if $B_1, B_2, \cdots$ is a partition of *S*, and *A* is any event, then $\Pr(A) = \sum_{i=1}^{\infty} \Pr(A|B_i)P(B_i)$ and that of Bayes' Theorem states if $B_1, B_2, \cdots$ is a partition of *S*, and *A* is any event, then $\Pr(B_i|A) = \frac{\Pr(B_i \cap A)}{\Pr(A)} = \frac{\Pr(B_i)\Pr(A|B_i)}{\sum_{i=1}^{\infty}\Pr(B_i)\Pr(A|B_i)}$.

The probability $\Pr(B_i)$ is called the priori probability and $\Pr(B_i \mid A)$ is called the posteriori probability. Accordingly, Bayes' Theorem determines the posteriori probability $\Pr(B_i \mid A)$ from the observation given that the event A has already occurred. This result is of many practical importances leading to Bayesian classification and Bayesian estimation.

Additionally, the attacks not only create havoc in the network system but also make the system highly congested holding the safety and the internal mechanism fails for hours and hours [11]. The characteristics of an efficient ISD that included 1) decentralized and distributed monitoring, 2) identification of coordinated attacks, and 3) passive network traffic analysis disrupting smooth automation networks [12].

2 Probabilistic State Finite Automata (PSFA)

A novelty approach that evolves around finite state automation is fascinating. A finite automaton is a mathematical model consisting of a set of states, a set of transitions between states, an input alphabet, an initial state, and a final state [13]. The following is a simple example of a transition diagram of finite automata.



	a	b
→ s	s	t
← t	s	t

Figure 3. An Example of a Transition Diagram of Finite Automata

Components in Figure 3 specify: the set of states as $S=\{s,t\}$, the input alphabet as $A=\{a,b\}$, the initial state as s, and the set of terminal states as $\{t\}$. The transition function $\delta: S\times A\to S$ given by $\delta(s,a)=s,\ \delta(s,b)=t,\ \delta(t,a)=s,$ and $\delta(t,b)=t.$

A PSFA is an extended finite automaton in which each state has an associated probability based on a user signature. Some familiar UNIX commands such as `login, lpr, cd, vi, ls, pico, mv, mail, www`, and `exit` are used to create an example of intrusion paths with probabilities calculated in Table 2, where p_j denotes the associated probabilities for each UNIX command listed.

Table 2. Probability Distribution of Traffic Paths

Traffics	Probability
`lpr/cd/vi`	$p_1\times p_2\times p_3$
`ls/pico/mv`	$p_4\times p_5\times p_6$
`ls/mail/www`	$p_4\times p_7\times p_8$

For the j number of traffic paths, each path has issued k number of commands $p_{j1}, p_{j2}, p_{j3},\ldots p_{j,k}\ldots.$, then $\sum_j\sum_k p_{j,k}=1.$ In general, computation of network reliability can often be calculated. Let ε be the set of all nodes and links in the network and p_e be the probability that the intruder is successful at some node, $e\in\varepsilon$. The subset $E_i\subseteq\varepsilon$ consists of these successful nodes and links. Thus, the probability that the intruder has succeeded in the network is $\Pr[E_i]=\prod_{e\in E_i} p_e\times\prod_{e\in\varepsilon\setminus E_i}(1-p_e)$. There are 2^n possible network paths for an intruder to be successful in achieving this exponential growth of possible network avenues, where n is the number of links in ε.

The extension of time-dependent deterministic finite automation (TDFA) enables us to not only study more than just the sequence of input characters, but also to consider the time intervals between receiving input characters in recognizing behavioral patterns as normal. As a result, the uses of automata to recognize denial of service (DoS) attack signatures between arriving network packets will prove to be a reliable technique in the intrusion detection process [14]. However, TDFA uses the actual difference of arrival times between two input characters, leading to an infinite set of possible differences. This creates Boolean values resulting from the comparison of each of the differences to constants or variables defined in the automata. If we consider the problem of modeling network transition patterns, it is common to assume that rhythms occur with more frequency than those that correspond to random phonemes [15]. Hidden Markov Models (HMMs) are frequently used in many areas of pattern recognition and more specifically, in network intrusion detection. It is based on the fact that web information learning retains the ability to recognize other pattern domains, such as the Reber grammar provided in Figure 4.

Copyright © Society for Science and Education United Kingdom

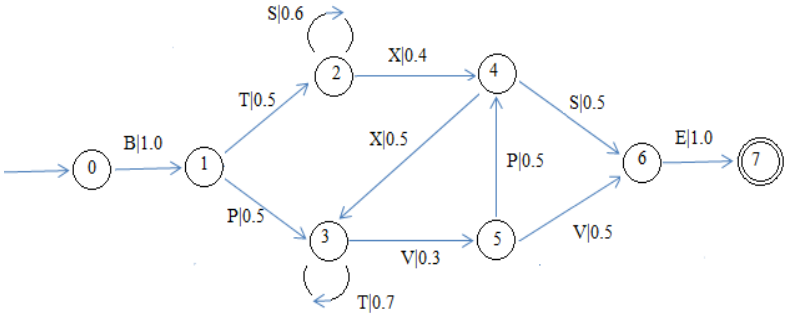


Figure 4. A SDFA Related to the Reber Grammar

A probabilistic automation defines a probability distribution over the set of strings of length n, for any particular n [16].

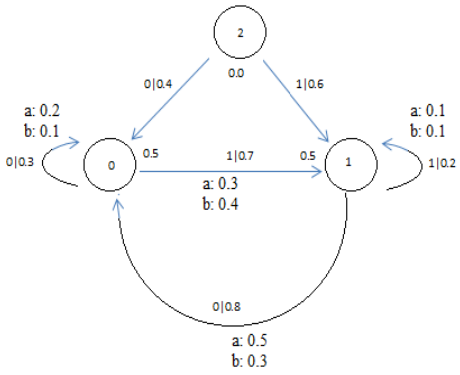


Figure 5. An Example of Probabilistic Automation

The probability assigned to this probabilistic automation in Figure 5 on the string $\omega = a \cdot a \cdot b$ is computed below:

$$P(\omega) = \pi_P(0).M_P(0,0,a).M_P(0,1,a).M_P(1,1,b) + \pi_P(0).M_P(0,1,a).M_P(1,1,a).M_P(1,0,b)$$
$$+ \pi_P(0).M_P(0,1,a).M_P(1,1,a).M_P(1,1,b)$$
$$+ \pi_P(1).M_P(1,1,a).M_P(1,1,a).M_P(1,0,b)$$
$$+ \pi_P(1).M_P(1,0,a).M_P(0,0,a).M_P(0,1,b)$$
$$+ \pi_P(1).M_P(1,1,a).M_P(1,0,a).M_P(0,0,b)$$
$$+ \pi_P(1).M_P(1,1,a).M_P(1,0,a).M_P(0,1,b)$$
$$+ \pi_P(1).M_P(1,0,a).M_P(0,0,a).M_P(0,0,b)$$

$$= 0.5 \times 0.2 \times 0.3 \times 0.1 + 0.5 \times 0.3 \times 0.1 \times 0.3 + 0.5 \times 0.3 \times 0.1 \times 0.1 + 0.5 \times 0.1 \times 0.1 \times 0.3 + 0.5 \times 0.5 \times 0.2 \times 0.4 + 0.5 \times 01 \times 0.5 \times 0.1 + 0.5 \times 01 \times 0.5 \times 0.4 + 0.5 \times 0.5 \times 0.2 \times 0.1$$

$$= 0.0480.$$

3 From Queuing Theory

Analysis of M/M/1 queue using a discrete time Markov chain (DTMC) during the times, 0, δ, 2δ, 3δ, $\cdots$, where δ is a small positive number, provides techniques for network intrusion detection. The transition

probabilities are up to an order of $o(\delta)$ terms. The transition probabilities $p_{ij} = \Pr[N_{k+1} = j \mid N_k = i]$ that are independent of k for a time-homogeneous DTMC are depicted in Figure 6.

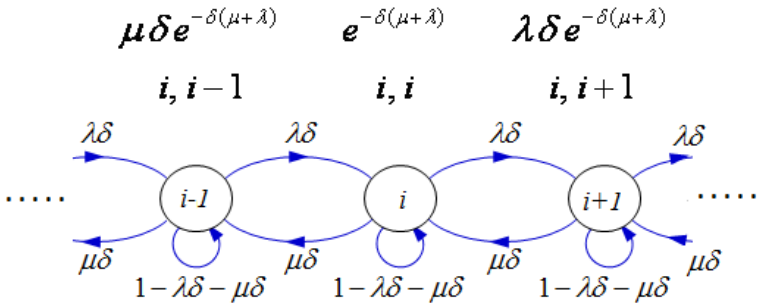


Figure 6. Transition Probabilities of DTMC for a M/M/1 queue from $i-1$ to $i+1$ States

One powerful, but simple formula in queuing theory [17], called Little's formula, has contributed to the study of intrusion detection. Little's formula of M/M/1 queues has uncovered an important phenomenon in cyber attacks. The expected number of attacks is proportional to the expected waiting time of an intruder, where the constant of proportionality is an average arrival time of attacks, $\lambda = \rho\mu$. In packet networks, the average packet delay caused by queuing is $T(c) = \dfrac{1}{\mu c - \lambda}$, where λ is the average packet arrival rate to a network link that follows a Poisson process, $\dfrac{1}{\mu}$ is the mean of average packet size that is exponentially distributed, and c is the link speed.

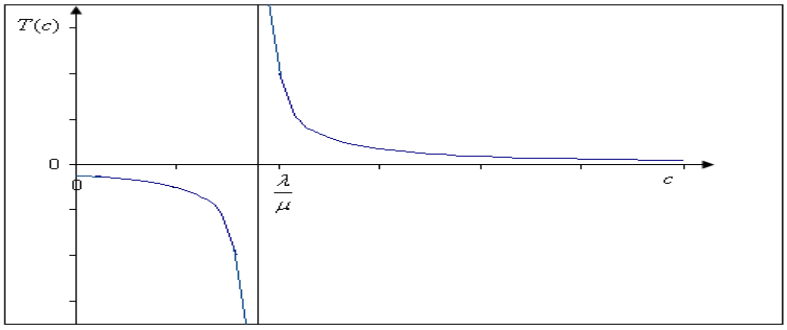


Figure 7. Average Packet Delay, $T(c)$ as a Function of Link Speed, c

Figure 7 concludes that for $c > \dfrac{\lambda}{\mu}$, average packet delay is exponentially decreasing and for $c < \dfrac{\lambda}{\mu}$, average packet delay stays negative, requiring some network justification.

Let us assume the arrival time has the geometric distribution with a parameter $(1-\rho)$ [18]. Let X be the number of unsuccessful attacks until the first successful attack has occurred. If p is the probability of successful attack then the probability that the k^{th} attack has been successful is $\Pr[X = k] = p(1-p)^k$, $k = 0, 1, 2, \cdots$ and $0 < p < 1$. This distribution has the same memory less property that has been held for Poisson distribution.

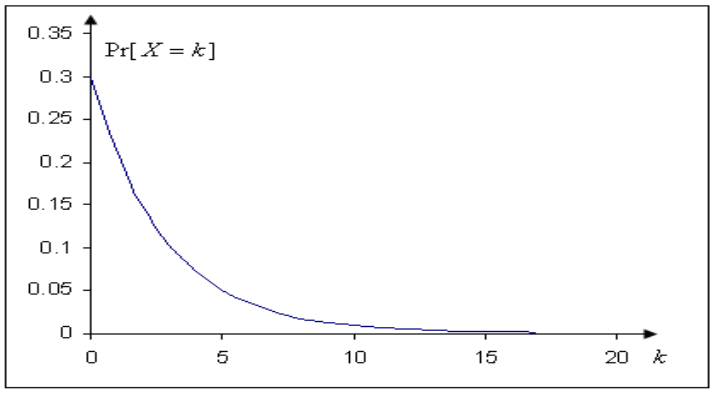


Figure 8. Probability that the k^{th} Attack has been Successful (taking $p = 0.3$)

Figure 8 exhibits the high probabilities for an attacker to be successful in his/her early attempts, as well as the diminished probability in subsequent attempts. If the attacker is selecting an address from the entire space of $N = 2^{32}$ addresses, the probability of detecting at least a single-packeted attack is $\Pr[\text{Detecting at least one - packeted attack}] = 1-(1-p)^k$, where $p = \frac{n}{N}$ is the probability of observing a single packet, assuming that the detector sensor monitors only n IP addresses [19]. As a result, the graph in Figure 6 is entirely flipped vertically, demonstrating the new probabilities. The probability of seeking j packets from the binomial distribution is $\Pr[j \text{ Packets}] = \binom{k}{j}(p)^j(1-p)^{k-j}$.

4 Profiles of Software Dynamics

Operational, functional, and module profiles have been the topics of much discussion [20]. Packet-based traffic monitoring is an application of multinomial distribution [21]. We observe certain parameters to estimate them for the population distribution. Let X be a random variable taking two values on the basis that an intrusion has occurred or otherwise. Thus, $X(\omega) = \begin{cases} 1 & if\ \omega \in A \\ 0 & if\ \omega \notin A \end{cases}$. This experiment is repeated as many as n times. Let Y be another random variable indicating the number of successes the intruder had. That is, $Y = \#\{i : X(\omega_i) = 1, \omega_i \in A\}$. Accordingly, $Y = \sum_{i=1}^{n} X(\omega_i) \rightarrow Y = B(n, p)$, where $B(n, p)$ is the binomial distribution with parameters n and p. The combined probability function of $Y_1, Y_2 \ldots, Y_n$ gives the multinomial distribution. For a given specific operation, say O_k, it will distribute its activity across the set of functionalities, $F^{(O_k)}$. At any arbitrary interval, n, during the expression of O_k, the program will be executing a functionality $f_i \in F^{(O_k)}$ with a probability, $\Pr[Y_n = i \mid X = k]$. From this conditional probability distribution for all operations, we derive the functional profile for the design specifications as a function of a user operational profile:

$$\Pr[Y = i] = \sum_j \Pr[X = j]\Pr[Y = i \mid X = j].$$

$\Pr[Y=i]=\sum_j p_j \Pr[Y=i \mid X=j]$.

$\Pr[Y=i]=\sum_j p_j \Pr[Y=i, X=j]/\Pr[X=j]$. Three possible traffic paths occur with probabilities p_i, $i=1,2,3,\ \sum_{i=1}^{3} p_i = 1$. Suppose that n independent replications of this traffic are initiated and let $X_i, i=1,2,3,$ denote the number of times outcome i appears. Now if $Y=i$ and $X=j$, then it follows that $Z=n-i-j$. However,

$\Pr[Y=i, X=j, Z=n-i-j]=\dfrac{n!}{i!\,j!(n-i-j)!}\,p_1{}^{i}\,p_2{}^{j}\,p_3{}^{(n-i-j)}$. This follows, since any particular sequence of n traffic paths having path 1 appearing i times, path 2 appearing j times, and path 3 appearing $(n-i-j)$ times has probability $p_1{}^{i}\,p_2{}^{j}\,p_3{}^{(n-i-j)}$ of occurring. Since there are $n!/[i!\,j!(n-i-j)!]$ such sequences, we have

$\Pr[Y=i]=\sum_j p_j \dfrac{\dfrac{n!}{i!\,j!(n-i-j)!}\,p_1{}^{i}\,p_2{}^{j}\,p_3{}^{(n-i-j)}}{\dfrac{n!}{j!(n-j)!}\,p_2{}^{j}(1-p_2)^{n-j}}$, where we have used the fact that Y has binomial distribution with parameter n and p_2. $\Pr[Y=i]=\sum_j p_j \binom{n-j}{i}\left(\dfrac{p_1}{1-p_2}\right)^{i}\left(1-\dfrac{p_1}{1-p_2}\right)^{n-j-i}$. For other profiles, more discussions are found in [22].

Let us now assume that an intruder attempts to attack a network system which has n components with probability p_2. The probability that he will be successful in each component is p_1. Let X_2 be the number of these components that are actually attacked. With this assumption, X_2 is then the binomial distribution with parameters x_1 and p_2, given x_1 components. The joint probability function for X_1, X_2 is the product given by

$p_{X_1,X_2}(x_1,x_2)=\binom{n}{x_1}p_1{}^{x_1}(1-p_1)^{n-x_1}\binom{x_1}{x_2}p_2{}^{x_2}(1-p_2)^{x_1-x_2}$, for $x_1=0,1,\ldots,n$ and $x_2=0,1,,\ldots,x_1$. The marginal probability function for X_2 is obtained by summing over the range of possible x_1 values for the given $x_2: x_1=x_2, x_2+1,\ldots,n$.

$$p_{X_2}(x_2)=\binom{n}{x_2}\left(\frac{p_2}{1-p_2}\right)^{x_2}\sum_{x_1=x_2}^{n}\binom{n-x_2}{x_1-x_2}\left(p_1(1-p_2)\right)^{x_1}(1-p_1)^{n-x_1}$$

$$p_{X_2}(x_2)=\binom{n}{x_2}\left(\frac{p_2}{1-p_2}\right)^{x_2}\sum_{j=0}^{n-x_2}\binom{n-x_2}{j}\left(p_1(1-p_2)\right)^{j+x_1}(1-p_1)^{n-x_1-j}$$

$$p_{X_2}(x_2) = \binom{n}{x_2}\left(\frac{p_2}{1-p_2}\right)^{x_2}\left(p_1(1-p_2)\right)^{x_2}\left(p_1(1-p_2)+(1-p_1)\right)^{n-x_2}$$

$p_{X_2}(x_2) = \binom{n}{x_2}(p_1p_2)^{x_2}(1-p_1p_2)^{n-x_2}$. Accordingly, X_2 has a binomial distribution with parameters n and p_1p_2.

In addition, majority of assumptions in the domain of cyber security naturally meet the properties of a Poisson process, namely, the number of intrusions can be modeled by a Poisson distribution and in fact, the time between intrusions is exponentially distributed. However, the log-normal distribution significantly fits the modeling in terms of the number of detected intrusions and the time between intrusions. The Pareto distribution is an alternative to both of these distributions that analyzing whether time-to-compromise (TTC) increase for each successful intrusion of network systems [23 & 24].

5 Conclusions and Future Work

The probability distribution theory assisted us in breaking down several parts of the analysis. The hardening hardware should be done at very appropriate levels using the topological nature of these analyses. In every case beginning with the identification of operating systems, network strengths, weaknesses, opportunities, and threats should be analyzed in a logical fashion. Assessing present strategies is done only when we are fully aware and conversant with the detective analysis of the systems. Necessary changes, improvements, and recommendations for any system will inevitably benefit the analysis and its mere purpose for preventing intrusion. Stating explicitly how to identify strengths of the methods to exploit the intrusion, rectifying the weaknesses, and preventing intrusion's threats to thus outline precautions to safeguard the network system is in fact needed.

REFERENCES

[1] Kumar, V. Anil (2004). *Sophisticated in Distributed Denial-of-Service Attacks on the Internet*, Current Science, Vol. 87, No. 7, pp. 885-888

[2] CERT Insider Threat Center (2009). Software Engineering Institute, Carnegie Mellon University, Last updated February 12, 2009, http://www.cert.org/stats/

[3] Internet Crime Report, Internet Crime Complaint Center (2012). http://www.ic3.gov/media/annualreport/2011_ic3report.pdf

[4] Chang, H.-Y., Wu, S. F. and Jou, Y. F. (2001). *Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks,* ACM Trans. Inf. Sys. Sec., Vol. 1, (2001), pp. 1-36

[5] Barbará, D., Couto, J., Jajodia, S., Popyack, L., and Wu, N. (2001). *ADAM: Detecting Intrusions by Data Mining*, Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West point, NY, pp. 5—6, June 2001

[6] Li, Z., Gao, Y., and Chen, Y. (2005). *Towards a High-speed Router-based Anomaly/Intrusion Detection System,* Northwestern University, http://www.sigcomm.org/sigcomm2005/poster-121.pdf

[7] Sebyala, A. A., Olukemi, T., and Sacks, L., (2002). *Active Platform Security through Intrusion Detection Using Naïve Bayesian Network for Anomaly Detection,* http://www.ee.ucl.ac.uk/lcs/papers2002/LCS116.pdf

[8] Kang, D.-K., Fuller, D., and Honavar, V. (2005). *Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation*, Proceedings of the 2005 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY

[9] Yang, G., Zhou, W., and Qiao, D. (2007). Defending against barrier intrusions with mobile sensors, the Proceedings of 2007 International Conference on Wireless Algorithms, Systems and Applications, 2007, pp. 113-120

[10] Rafiee, M. and Bayen, A. M. (2010). Optimal network topology design in multi-agent systems for efficient average consensus, Decision and Control (CDC), 2010 49th IEEE Conference on, Atlanta, GA, 2010, pp. 3877-3883

[11] Barbosa, R. R. R. and Pras, A. (2010). Intrusion Detection in SCADA Networks, the Proceedings of the Conference: Mechanisms for Autonomous Management of Networks and Services, 4th International Conference on Autonomous Infrastructure, Management and Security, AIMS 2010, Zurich, Switzerland, June 23-25

[12] Schuster, F. and Paul, A. (2012). A distributed intrusion detection system for industrial automation networks," Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on, Krakow, 2012, pp. 1-4

[13] Freeman, *S.,* Branch, J., Bivens, A., and Szymanski, *B*. (2002). *Host-Based Intrusion Detection Using User Signatures,* Proc. Research Conference, Troy, NY 12180-3590, http://www.cs.rpi.edu/~szymansk/papers/signature.pdf

[14] Petersson, K. M., Grenholm, P., and Forkstam, C. (2005). Artificial grammar learning and neural networks. In G. B. Bruna, L. Barsalou, & M. Bucciarelli (Eds.), Proceedings of the 27th Annual Conference of the Cognitive Science Society, pp. 1726-1731

[15] Kermorvant, C. and Dupont, P. (2002). Stochastic grammatical inference with multinomial tests. In 6th International Colloquium on Grammatical Inference: Algorithms and Applications (ICGI), Vol. 2484 of Lecture Notes in Computer Science, Springer, 2002, pp. 149–160

[16] Abe, N. and Warmuth, M. K. (1990). On the Computational Complexity of Approximating Distributions by Probabilistic Automata, Machine Learning, 1990, pp. 205-260

[17] Bose, K. Sanjay (2002). An Introduction to Queueing Systems Kluwer Academic/Plenum Publishers, New York, 2002

[18] Myers, J. L. and Well, A. D. (2003). *Research Design and Statistical Analysis,* second edition, 2003, Lawrence Erlbaum Associates, Inc. Publishers

[19] Wegman E. J. and Marchette, D. J. (2004). *Statistical Analysis of Network Data for Cybersecurity,* Chance, Vol. 17, No. 1 (2004), pp. 9-19

Copyright © Society for Science and Education United Kingdom

[20] Elbaum, S. and Munson, J. C. (1999). *Intrusion Detection: Through Dynamic Software Measurement*, Proceedings of the Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, USA, April 9—12, 1999, The USENIX Association, The Advanced Computing Systems Association

[21] Boodnah, J. and Scharf, E. M. (2005). *Applying Clustering to a Framework for Generating Trust,* http://www.ctr.kcl.ac.uk/iwwan2005/papers/39.pdf

[22] Munson, J. C. and Elbaum, S. (1999). Software Reliability as a Function of User Execution Patterns, Proceedings of the 32nd Hawaii International Conference on System Sciences – 1999, pp. 1-12, http://cse.unl.edu/~elbaum/papers/workshops/hawai99.pdf

[23] Van Oorschot, P. C., Robert, J.-M., and Martin, M. V. (2006). A monitoring system for detecting repeated packets with applications to computer worms Int. J. Inf. Secur. (2006) 5(3): pp. 186–199

[24] Holm, H. (2014). A large-scale study of the time required to compromise a computer system, Browse Journals & Magazines: Dependable and Secure Computing, IEEE Transactions, Vol. 11 Is. 1, 2014, pp. 2-15