Cyber Security Analytics: A Stochastic Model for Security Quantification Using Absorbing Markov Chains

Subil Abraham and Suku Nair
Southern Methodist University, Dallas 75275, USA
Email: smabraham@mail.smu.edu; nair@lyle.smu.edu

Abstract — Most of the current research in security analysis has been centered on identifying threats and vulnerabilities and providing suitable defense mechanisms to improve the robustness of the network and systems. While this approach is attractive, it provides limited insight into understanding the impact these attacks have on the overall security goals of the network and the system. Attack Graph as a model lends itself nicely to the analysis of the security state of a network. Most of the Attack graph based metrics proposed in the literature are one dimensional; however the research community has acknowledged the fact that security needs to be treated as a multidimensional concept. In this paper, we utilize stochastic modeling techniques using Attack graphs to define a complementary suite of quantitative metrics to aid the security engineer in visualizing the current as well as future security state of the network and optimizing the necessary steps to harden the enterprise network from external threats. We present experimental results from applying this model on a sample network to demonstrate the practicality of our approach.

Index Terms—Attack graph, CVSS, markov model, security evaluation, cyber situational awareness

I. Introduction

Humans have intuitively used measurement as a means to understand the world around them and quantify it with a high level of precision. As Lord Kelvin put it, "when you can measure what you are speaking about and express it in numbers, you know something about it" [1]. Evaluating the security of an enterprise is an important step towards securing its system and resources. It can help security administrators make important decisions regarding how to design their systems as well modify resources dynamically to counteract any outside threats. In a recent 2013 survey conducted by Cyber-Ark [2], it was found that more than eighty percent of C-level executives and IT security professionals believed that their nation was far more susceptible to cyber-attacks than physical attacks. Several security forums have outlined the need for having metrics to evaluate the security of an enterprise. For example INFOSEC [3] has identified security metrics as being one of the top 8 security research priorities. Similarly Cyber Security IT Advisory Committee [4] has also identified this area to be among the top 10 security research priorities.

In this paper, we present an integrated view of security for computer networks within an enterprise. We introduce the concept of Cyber-Security Analytics to provide a high-level assessment of the current as well as future (predicted) security state of the network based on the aggregation of the security attributes at each node or subsystem within the network.

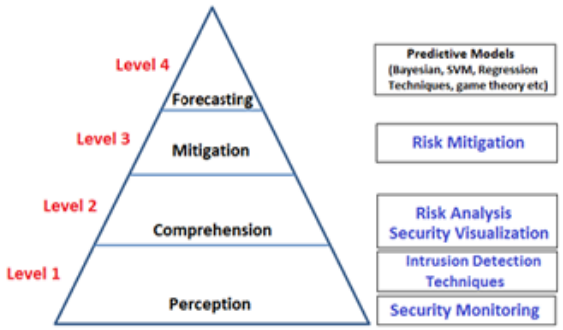

Fig. 1. Cyber-situational awareness model

The proposed research falls under the realm of "*Cyber Situational Awareness*" which provides a holistic approach to understanding threats and vulnerabilities, performing analysis (using data mining & predictive analytics) to evaluate the current security situation as well as perform a projection or forecast of the future security state to address potential situations. There are multiple levels to Situational Awareness as depicted in Fig. 1. Situational awareness is a universal concept and is the ability to identify, comprehend and forecast the integral features of a system. Situational awareness was defined by [5]-[7] as "the perception of the elements in the environment with a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.

The remainder of the paper is organized as follows. In Section 2, we discuss about attack graphs and provide a brief historical perspective on it. And then we discuss previous research proposed for security metrics. In Section 3, we explore the Cyber-Security Analytics Framework in more detail. In Section 4 and 5, we present our analysis and provide results from our simulation. Finally, we conclude the paper in Section 6.

II. Background and Related Work

Manuscript received August 18, 2014; revised December 24, 2014.
Corresponding author email: smabraham@mail.smu.edu.
doi:10.12720/jcm.9.12.899-907

Here we briefly discuss about Attack Graphs and then provide an overview of some of the most prominent works that have been proposed for quantifying security in a network.

A. Attack Graph

Computer attacks have been graphically modeled since the late 1980s by the US DoD as discussed in their paper [8]. Most of the attack modeling performed by analysts was constructed by hand and hence it was a tedious and error-prone process especially if the number of nodes were very large. In 1994 Dacier *et al* [9] published the one of the earliest mathematical models for a security system based on privilege graphs. By the late 1990's a couple of other papers [8], [9] came out which enabled automatic generation of attack graphs using computer aided tools. In [8] the authors describes a method of modeling network risks based on an attack graph where each node in the graph represented an attack state and the edges represented a transition or a change of state caused by an action of the attacker. Since then researchers have proposed a variety of graph-based algorithms to generate attack graphs for security evaluation.

B. Classes of Security Metrics

There are different classes under which network security metrics fall under. These classes are depicted in Fig. 2.

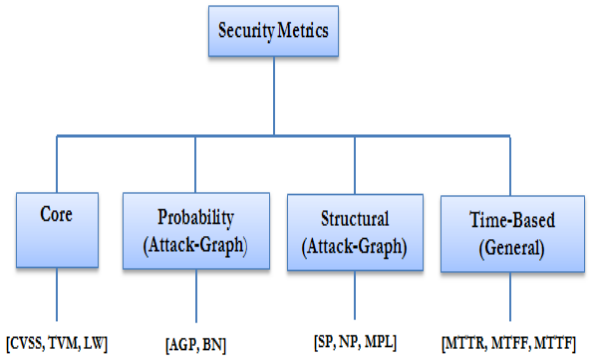

Fig. 2. Security metric classification

Here are some examples of metrics that fall under each category.

Core Metrics: These are aggregation metrics that typically don't use any structure or dependency to quantify the security of the network. A few examples that fall under this category are Total Vulnerability Measure (TVM) [10] and Langweg Metric (LW) [11]. TVM is the aggregation of two other metrics called the Existing Vulnerabilities Measure (EVM) and the Aggregated Historical Vulnerability Measure (AHVM). CVSS [12], [13] is an open standard for scoring IT security vulnerabilities. It was developed to provide organizations with a mechanism to measure vulnerabilities and prioritize their mitigation. For example the US Federal government uses the CVSS standard as the scoring engine for its National Vulnerability database (NVD) [14] which has a repository of over forty-five thousand known vulnerabilities and is updated on an ongoing basis.

Structural Metrics: These metrics use the underlying structure of the Attack graph to aggregate the security properties of individual systems in order to quantify network security. The Shortest Path (SP) [15] metric measures the shortest path for an attacker to reach an end goal. The Number of Paths (NP) [15] metric measures the total number of paths for an attacker to reach the final goal. The Mean of Path Lengths (MPL) metric [16] measures the arithmetic mean of the length of all paths to the final goal in an attack graph. The above structural metrics have shortcomings and in [17], Idika *et al* have proposed a suite of attack graph based security metrics to overcome some of these inherent weaknesses. In [18], Ghosh *et al* provides an analysis and comparison of all the existing structural metrics.

Probability-Based Metrics: These metrics associate probabilities with individual entities to quantify the aggregated security state of the network. A few examples that fall under this category are Attack Graph-based Probabilistic (AGP) and Bayesian network (BN) based metrics [19]-[21].

Time-Based Metrics: quantify how fast a network can be compromised or how quickly a network can take preemptive measures to respond to attacks. Common metric that fall in this category are Mean Time to Breach (MTTB), Mean Time to Recovery (MTTR) [22] and Mean Time to First Failure (MTFF) [23].

The drawback with all these classes of metrics is that they take a more static approach to security analysis. They do not adopt any stochastic modeling techniques which leverages the CVSS framework in order to assess current security situation and help locate critical nodes for optimization.

Cyber Situation Awareness: Tim Bass [24] first introduced the concept of cyberspace situation awareness and built a framework for it which laid the foundation for subsequent research in Network Security Situational Awareness [25]. In [26]-[29] the framework was extended to model security at a large-scale level where existing techniques have been integrated to gain richer insights. Researchers have also proposed many evaluation models and algorithms for NSSA [30], [31] to reflect the security environment and capture the trends of changes in network state. The drawback with most of these NSSA models is that they don't adopt a consistent integrated framework for describing the relationships between the vulnerabilities in the network nor do they use an open scoring framework such as CVSS for analyzing the dynamic attributes of a vulnerability using stochastic modeling techniques.

III. Cyber-Security Analytics Framework

In this section, we introduce a new set of enterprise security metrics by exploring the concept of modeling the Attack graph as a stochastic process. In order to truly

comprehend and visualize the strength of a network against attacks, we need to design a model that generates a complementary suite of quantitative metrics across multiple dimensions. The model we are proposing aids the security engineer to analyze the security properties of a network by considering the time needed to breach a security goal, the probability for a particular node to be compromised and the number of steps needed to reach a security goal. By providing a single platform and using an open vulnerability scoring framework such as CVSS it is possible visualize the current as well as future security state of the network and optimize the necessary steps to harden the enterprise network from external threats.

A. State Based Stochastic Modeling

Several research fields have long used Modeling and Simulation to study the behavior of a system under different varying conditions. By applying the same methodology in the area of security enables an IT security administrator to assess the current security state of the entire network as well as predict how this state will change based on new threat levels, new vulnerabilities etc. Another capability is analyzing what-if scenarios to calculate metrics based on making certain changes to system. Most IT departments are faced with limited budgets and hence applying patches to all systems in a timely manner may not be feasible. Hence it is necessary to optimize the application of such security controls without compromising the network or disrupting business operations. Markov model is one such modeling technique that has been widely used in a variety of areas such as system performance analysis [32] and dependability analysis [33], [34].

B. Model Representation

Fig. 3 shows a high level view of our proposed cyber-security analytics model where we have captured all the processes involved in building our security metric framework.

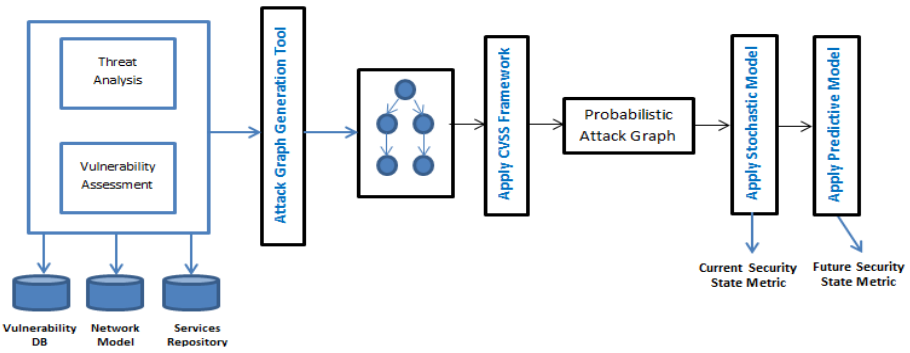

Fig. 3. Cyber security analytics model

The core component of our model is the Attack Graph which is generated using a network model builder by taking as input network topology, services running on each host and a set of attack rules based on the vulnerabilities associated with the different services. Given an Attack Graph, we associate probabilities to each of the edges of the graph. This probability represents the likelihood of a vulnerability to be exploited by an attacker. Then, a stochastic process is applied over the Attack Graph to describe the attacks. This allows us to captures various security metrics leading to useful insights into the current security state of the network. The model also takes into account zero-day attacks for vulnerabilities that have been disclosed by a Security Information provider (SIP).

We will model the Attack Graph (AG) described in the previous section as an absorbing Markov chain since it satisfies the following two properties.

- An attack graph has at least one absorbing state or goal state.
- In an attack graph it is possible to go from every state to an absorbing state.

The absorbing state is the node where the security goal is violated. Once the attacker reaches this state then the system is considered to be in a compromised state and the attacker will have met their objective. So the attacker will continue to remain in this state until preventive measures have been taken by the security team to remove the attacker's presence from the system. The transition matrix for an absorbing Markov chain has the following Canonical form**.**

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

Here $\boldsymbol{P}$ is the transition matrix, $\boldsymbol{R}$ is the matrix of absorbing states, and $\boldsymbol{Q}$ is the matrix of transient states. The set of states, S in the model represent the different vulnerabilities associated with services running on the nodes that are part of the network. The attacker moves from one state to another when the target state's vulnerability is exploited. Depending on the complexity of the vulnerability, some are more easily exploited than others. We define the exploitability score $e(v)$ as the measure of complexity in exploiting the vulnerability v. The CVSS standard provides a framework for computing these scores using the *access vector* (*AV*), *access complexity*(*AC*) and *authentication*(*Au*) as follows

$$e(v) = 20 \times AV \times AC \times Au$$

The constant 20 represents the severity factor of the vulnerability. As an example, consider CVE-2007-3039 which is stack-based buffer overflow vulnerability in Microsoft Windows. The access vector, authentication and access complexity for this vulnerability is 1, 0.56 and 0.71 respectively. Therefore the exploitability of CVE-2007-3039 is 8.0 which indicate that it has relatively high exploitability.

Given the exploitability scores for each of the vulnerabilities in the Attack Graph, we can estimate the transition probabilities of the Absorbing Markov chain by normalizing the exploitability scores over all the edges starting from the attacker's source state. Let p_{ij} be the probability that an attacker currently in state i exploits a vulnerability in state j. We can then formally define the transition probability below where n is the number of outgoing edges from state i in the attack model and e_j is the exploitability score for the vulnerability in state j.

$$p(i, j) = \frac{e_j}{\sum_{k=1}^{n} e_k}$$

The matrix P represents the transition probability matrix of the Absorbing Markov chain where, $p(i, j) \geq 0$ for all $i, j \in S$. In an absorbing Markov chain the probability that the chain will be absorbed is always 1. Further, each row of P is a probability vector, which requires that

$$\sum p(i, j) = 1 \text{ for all } i \in S$$

In an absorbing Markov chain the probability that the chain will be absorbed is always 1. Hence

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

Therefore for an absorbing Markov chain P, we can derive a matrix $N = (1-Q)^{-1} = I + Q + Q^2 + Q^3 + \cdots$ which is called the *fundamental matrix* for *P*. This matrix N provides considerable insight into the behavior of an attacker who is trying to penetrate the network. The elements of the fundamental matrix n_{ij} describe the expected number of times the chain is in state *j*, given that the chain started in state *i*. By using these elements we will measure several aspects of the network's security using an integrated approach.

IV. Security Analysis

In this section we present qualitative analysis and quantitative analysis using our cyber security analytics model. The focus of our analysis will on assessing the current/present security state of the network

A. Qualitative Analysis

Node Rank Analysis: It is important to determine the critical nodes in the Attack graph where the attacker is most likely to visit. Based on this insight, the necessary systems can be patched for improved security. The unique property associated with the Fundamental matrix N is that each element n_{ij} gives the expected number of times that the process is in the transient state s_j given that it started in the transient state s_i. This matrix is critical for a security engineer while analyzing the security situation of the network. Each element in the Fundamental matrix N gives the expected number of times an attacker will visit a state j given that he started at state i. By analyzing this matrix we can identify points in the network where vulnerabilities need to be patched if the attacker is more likely to target or visit a state which is critical to the function of the business.

B. Quantitative Analysis

Expected Path length (EPL) metric: measures the expected number of steps the attacker will take starting from the initial state to compromise the security goal. Using the Fundamental matrix N of the Absorbing Markov chain, we can compute the expected number of steps before the chain goes to the absorbed state. For example let t_i be the expected number of steps before the chain is absorbed, given that the chain starts in state s_i, and let *t* be the column vector whose i^{th} entry is t_i. Then

$$T = Nc \text{ where for all } j\ c_j \text{ is } 1$$

In addition to calculating the expected number of steps to absorption from the fundamental matrix, we can also estimate the probability that the chain will be absorbed if we were to start in the transient state s_i. This security metric is considering one dimension which is the number of steps or the resistance of the network.

Probabilistic Path (PP) metric: is another dimension which measures the likelihood of an attacker to reach the absorbing states of the graph. For this we will calculate the following matrix B where $B = NR$ where N is the fundamental Matrix and R is obtained from the Canonical form. The element b_{ij} in the matrix measures the probability of reaching the security goal state j given that the attacker started in state i. The *Probabilistic Path (PP)* metric also aids the security engineer in making decisions on optimizing the network and we will label this as the *Probabilistic Path (PP)* metric.

Temporal Attack Graph (TAG) Score: The third dimension we will look at will measure temporal aspect of security. We will extend our discrete model and make it continuous by taking into consideration the sojourn time in each state. A continuous time Markov process can be represented by a generator matrix *G* and an initial state probability vector $p = (p_1, p_2 \ldots p_n)$ where *n* is the number of states. The generator matrix G has the following form

$$G = \begin{pmatrix} g[1,1] & g[1,2] & \cdots & g[1,n] \\ g[2,1] & g[2,2] & \cdots & g[2,n] \\ \vdots & \vdots & g[i,j] & \vdots \\ g[n,1] & g[n,2] & \cdots & g[n,n] \end{pmatrix}$$

The time spent visiting each state *i* is the holding time H_i. It is exponentially distributed with parameter g_{ii}. Hence the expected holding time in each state is $1/g_{ii}$ and the distribution of the holding time is given by

$$F_{Hi}(a) = 1 - e^{-giia},\ 0<=a$$
$$0,\qquad a<0$$

If the transition matrix P and the holding times are available for a Markov process, then we can compute the generator matrix using the following equations.

$$gii = \varphi i$$

$$gij = giipij = \varphi ipij,\ j \neq 1$$

In a continuous time Markov chain, the transition probabilities can be described as a function of time as follows.

$$P_{ij}(t) = P_r(X_t = j \mid X_0 = i),\ t \geq 0$$

The term p_{ij} (t) is a transition function and denotes the probability that the process is in state j at time t, given that it was previously in state i at time 0. Analogous to the transition probabilities mentioned in Discrete Time Markov Chain, the transition functions $p_{ij}(t)$ can be organized into a transition function matrix as shown below.

The transition function matrix P (t) can be approximated as

$$P(t) = e^{-g^* t} \sum_{n=0}^{\infty} Q^n \frac{(g^* t)}{n!}$$

where $g^* = max_i \{gii\}$ and

$$Q = 1 + \frac{1}{g^*} G$$

The sojourn time for an attacker in each state is dependent on the attacker's profile as well as the exploitability score of the vulnerability. The lower the exploitability scores the longer the holding time in that state due to the level of difficulty in exploiting the vulnerability associated with that state. The transition rates associated with the generator matrix are calculated from the CVSS scoring framework. Hence by analyzing this generator matrix and plotting it on a graph, a security engineer can identify how long an attacker will take to breach the security goal state once the network is infiltrated by an attacker. This will also give the team adequate time to take preventive measures in order to subvert the attacker's effort. The longer it takes the stronger and resilient the network is. We will label this new metric as the *Temporal AG Score*.

V. Case Study and Analysis

In the previous section, we proposed a set of Attack graph (AG) based metrics to evaluate the security of networks and also forecast how the security of the network would evolve with time. In this section, we will present simulated results based on the analysis of these security metrics.

To illustrate the proposed approach in details, consider a simple but realistic network as shown in Fig. 5.

The network is comprised of 4 machines that are interconnected together and operating internally behind a firewall. The Attacker or threat is behind the firewall and is connected to the external network which is also connected to the firewall. The firewall has only one port open (port 80) to the outside network for access to its web-server. The machine hosting the web-server M1 is running Apache Webserver and is running on a Windows platform. Similarly M2 is running the ssh service using which the other 3 machines can connect to it using an authenticated account. M3 is running 2 services namely VMware server and ms-server service. And finally M4 is running an LDAP server and VNC server that allow authorized users to remotely control the machine for maintenance and configuration. The aim of the attacker is to infiltrate the network and gain root access on M4.

A. Environment Information

Table I contains a list of all the vulnerabilities that can be exploited by an attacker if certain conditions are met in our network. Each of the six vulnerabilities is unique and publicly known and is denoted by a CVE (Common Vulnerability and Exposure) identifier. For example Apache web-server was found to have vulnerability CVE-2002-0392 which allows remote attackers to execute arbitrary code via chunked encoding. Similarly the ftp service hosted by M1 had a vulnerability denoted by CVE-2006-5815 which allowed remote attackers, probably authenticated, to cause a denial of service and execute arbitrary code.

TABLE I: INTRUDER ACTIONS

Service Name	CVE-ID	Vulnerability	Host
apache	CVE-2002-0392	Chunked Encoding	M1
sshd	CVE-2008-4762	Stack-based buffer overflow	M2
vmware	CVE-2009-1147	Allows local users to gain privileges	M3
ms-server	CVE-2007-3039	Stack-based buffer overflow	M3
vncserver	CVE-2006-2369	Bypass Authentication	M4
openldap	CVE-2006-2754	Stack-based buffer overflow	M4

TABLE II: VULNERABILITY SCORES

Service Name	CVE-ID	Impact Subscore	Exploitability Subscore
apache	CVE-2002-0392	10	1.9
sshd	CVE-2008-4762	10	8.0
vmware	CVE-2009-1147	10	3.9
ms-server	CVE-2007-3039	10	8.0
vncserver	CVE-2006-2369	6.4	10
openldap	CVE-2006-2754	2.9	10

In Table II, each of the six vulnerabilities has been associated with an Impact score and an Exploitability score. Several public sites such as NVD (National

Vulnerability Database), MITRE, Secunia provide information about well-known vulnerabilities and also the severity of it using scores values adopted by the CVSS (Common Vulnerability Scoring System) framework.

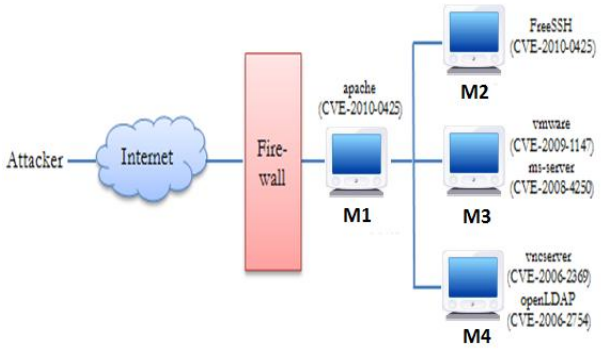

Fig. 4. Network Topology

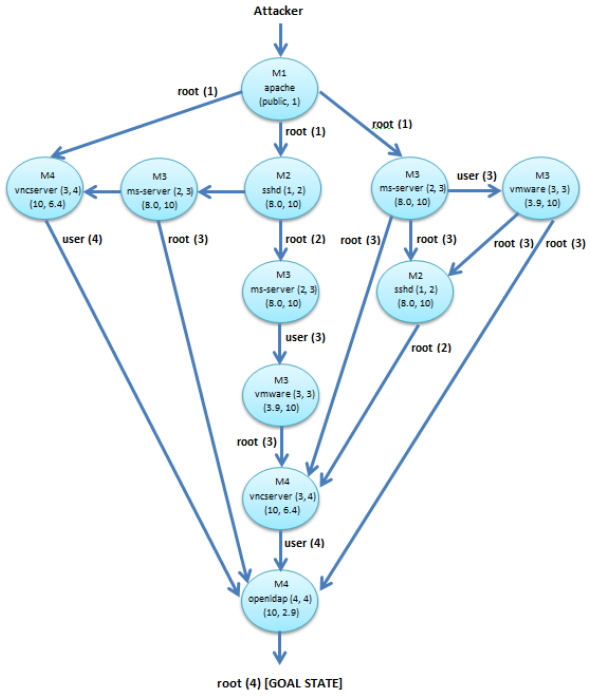

Fig. 5. Attack graph

B. Attack Graph Generation

By combining the vulnerabilities present in the network configuration (Fig. 4), we can build several scenarios whereby an attacker can reach a goal state. In this particular case, the attacker's goal state would be to obtain root access on Machine M4. Fig. 5 depicts the different paths an attacker can take to reach the goal state. By combining these different paths we are able to obtain an Attack Graph. A couple of practical approaches have been proposed [35-38] to automate generation of attack graphs without the intervention of a red team. In our case-study we have used the tool discussed in [36] to generate the attack graph from our sample network. The tool consists of two modules which is the network model builder and a scenario generator.

The model builder takes as input the network topology, services running on each host and a set of attack rules based on the vulnerabilities associated with the different services described in Table I and Table II. Using this information it constructs a finite model of the network which is encoded in XML. This specification is further extended with a security property which specifies the security requirements against which the attack tree is to be built. For example a security property that an attacker never gains root access to the openldap system [M4] can be specified as

$$AG\ (attacker.privilege[M4] < root)$$

The scenario graph generator takes as input the model in XML format and the security property and uses a symbolic model checker such as NuSMV [39] to generate a list of possible paths that violate this property which is the desired attack graph

C. Simulation Results

Based on the Attack Graph generated for network A, a simulation of the Absorbing Markov chain is conducted. In our experiment we model an attacker and simulate over 2000 attack paths over the Attack Graph based on the probability distribution of the nodes. We used the R statistic package [40] to generate the model and run the simulations. The transition probabilities are formulated from the CVSS scoring framework as described in section 3.

Each simulation run uses the transition probability row vector to move from one state to another until it reaches the final absorbing state. Fig. 6 shows 5 attack paths produced by this simulation by taking into account the probability associated with each transition from state i to state j. All the sample paths begin in state 1 and continue till it reaches the state 10 which is the state where the security goal is compromised. Since state 10 is an absorbing state, it will continue to remain in this state. In other words all sample/attack paths generated by the Markov chain will ultimately end in state 10. Notice that among the 5 attack paths depicted on the chart, the shortest path is of length 5 and the longest path is of length 8. Fig. 7 shows the histogram of the distribution of attack path lengths $X_1, X_2 \ldots\ldots X_{2000}$ from 2000 simulated sample paths. In other words this distribution models the behavior of an attacker and on average will require 4.689 steps to reach his objective. This is an important metric and we will denote this as the Expected Path Length (EPL). From the figure, we can see that the attacker is most likely to reach his goal or the absorbing state in 3 steps and the graph confirms the fact that the expected number of steps is in line with the expectation that was calculated using the Fundamental matrix described in section 4 which is 4.58.

Fig. 8 shows the histogram of the distribution of the states for the Attack Graph for all the 2000 sample paths that were simulated using the Absorbing Markov chain model. Based on the simulation result, if we were to exclude the start state(1) and the absorbing state(10), we can find that an attacker is most likely to visit state 2 and

least likely to be in state 7. Hence the attacker is most likely to exploit the vulnerability of the vnc server running on M4 and least likely to exploit the ms-server service on M3. This information is valuable for a security engineer to prioritize which exploit needs to be patched and how it will affect the strength of the network against attacks.

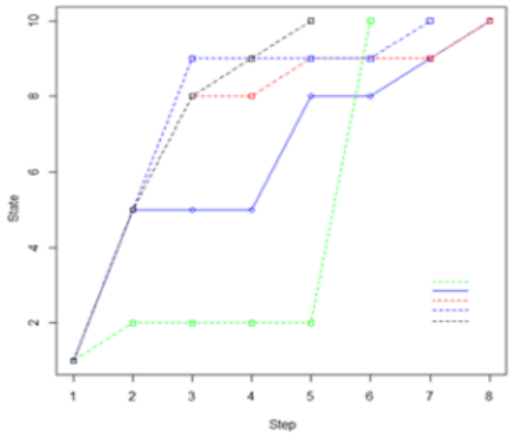

Fig. 6. State transitions

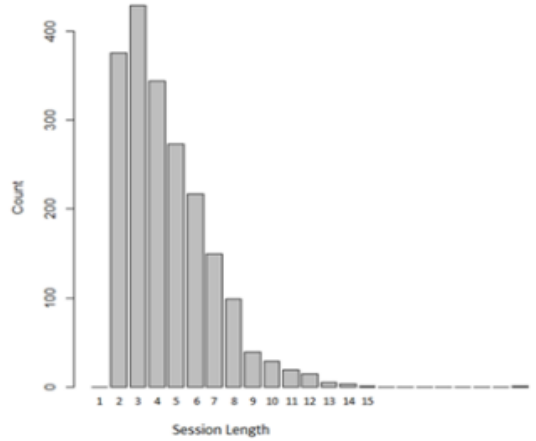

Fig. 7. Attack path length distribution

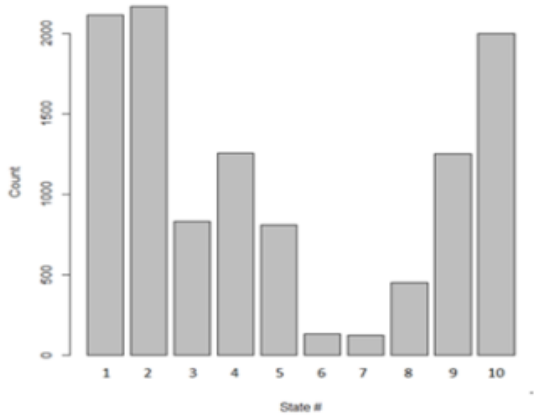

Fig. 8. State visit distribution

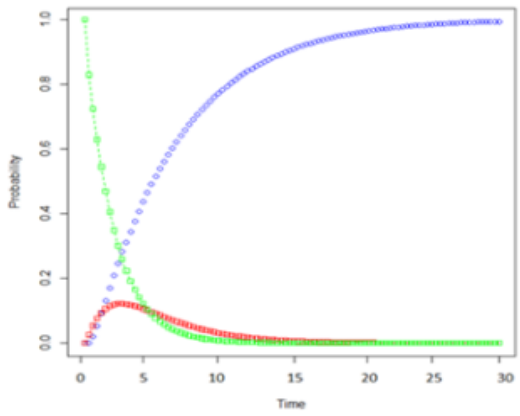

Fig. 9. TAG score distribution

In section 4, we had formulated a generator matrix for the Attack graph based on the transition matrix and holding times for the states in the Absorbing Markov chain. Fig. 9 shows a plot of $P_{1,1}(t)$ {green}, $P_{1,10}(t)$ {blue} and $P_{6,8}(t)$ {red} for t = 0, 1, 2….30. This is a function that traces the MTTA (Mean time to attack) for a network. At time t =0 the probability that an attacker is in state 1 is almost 1.0, but it drops rapidly by time t = 5 units. On the other hand the probability that the attacker has achieved his goal or is in the absorbing state 10 is nearly zero for the first 4 days and rises to 1.0 by time unit 20. Hence by analyzing this chart, a security engineer can identify how long an attacker will take to reach the critical state (10) once the network is infiltrated. This will also give the team adequate time to take counter measures to remove the attacker from the system. The larger the value for MTTA, the stronger and resilient the network is.

D. Comparison with Existing AG Security Metrics

Table III shows a comparison of the different structural Attack Graph metrics currently being used in the research community. These metrics consider only the path length dimension of security. The structural metrics shown in the table have been calculated using the example network configuration and the corresponding attack graph shown in Fig. 4 and Fig. 5. The metrics denoted by SP, NP, MPL, NMPL, MOPL, and MEPL uses the structure of the graph by considering the length and number of attack paths to compute how resistant the network is to external attacks. As mention in section 2, these scores don't take into account the relative difficulty in exploiting the vulnerabilities and instead makes the assumption that all exploits are of equal strength. In addition the metrics also don't factor in the profile of the attackers.

TABLE III: SEVERITY SCORES OF METRICS

Measure	Value
Shortest Path Metric (SP)	2
Number of Paths Metric (NP)	8
Means of Path lengths Metric (MPL)	3.625
Normalized Mean of Paths Lengths metric (NMPL)	0.4
Mode of Path Lengths Metric (MOPL)	3
Median of Path Lengths Metric (MEPL)	3.5
Probabilistic Security Metric (PSM)	0.61
Expected Path Length Metric (EPL)	4.58

The EPL metric we have defined for this particular dimension provides a more realistic estimate of the length an attacker must traverse before reaching his objective.

VI. Conclusion and Future Work

In this paper, we presented a stochastic model for cyber-security analytics using Attack graphs. Since existing metrics have potential short-comings for accurately quantifying the security of a network system, our framework employs a suite of complementary metrics to provide a more realistic and objective security evaluation of the network. What sets our model apart from the rest is our use of Absorbing Markov chains and the CVSS framework to comprehend and analyze the security situation from the structure of the network system. We conducted a simulation-based experiment to analyze the merits of using our model to evaluate security properties. For future work, we plan to extend the model by applying other predictive and forecasting techniques in order to create a more comprehensive, integrated approach to the evaluation of security.

References

[1] W. Thompson, *Popular Lectures and Addresses*, 1891-1894.

[2] C. Ark. (2013). *Global Advanced Threat Landscape*. [Online], Available: http://www.cyber-ark.com/landing-pages/global-advanced-threat-survey/index.asp

[3] INFOSEC Research Council Hard problem List, 2005.

[4] A Crisis of Prioritization, President's IT Advisory Committee, 2005.

[5] M. Endsley, "Toward a theory of situation awareness in dynamic systems," *In Human Factors Journal*, vol. 37, no. 1, pp. 32–64, March 1995.

[6] System Security Engineering Program Management Requirements, MIL-STD-1785, 1988.

[7] M. Dacier and Y. Deswarte, "Privilege graph: An extension to the typed access matrix model," in *Proc. ESORICS*, 1994.

[8] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proc. WNSP'98*, 1998, pp. 71-79.

[9] L. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," in *Proc. DISCEX01*, pp. 307-321.

[10] M. Abedin, S. Nessa, E. Al-Shaer, and L. Khan, "Vulnerability analysis for evaluating quality of protection of security policies," in *Pro. Quality of Protection*, October 2006.

[11] H. Langweg, "Framework for malware resistance metrics," in *Proc. Quality of Protection*, October 2006.

[12] M. Schiffman. Common Vulnerability Scoring System (CVSS). [Online]. Available: http://www.first.org/cvss/

[13] A. Ali, P. Zavarsky, D. Lindskog, and R. Ruhl, "A software application to analyze affects of temporal and environmental metrics on overall CVSS v2 score," Concordia University College of Alberta, Edmonton, Canada, October 2010.

[14] National Vulnerability Database. (2013). [Online]. Available: http://nvd.nist.gov/

[15] R. Ortalo, Y. Deswarte, and M. Kaaniche, "Experimenting with quantitative evaluation tools for monitoring operational security," *IEEE Transactions on Software Engineering*, vol. 25, pp. 633–650, September 1999.

[16] W. Li and R. Vaughn, "Cluster security research involving the modeling of network exploitations using exploitation graphs," in *Proc. Sixth IEEE International Symposium on Cluster Computing and Grid Workshops*, May 2006.

[17] N. Idika and B. Bhargava, "Extending attack graph-based security metrics and aggregating their application," *IEEE Transactions* on *Dependable and Secure Computing*, no. 99, pp. 1–1, 2010.

[18] A. Kundu, N. Ghosh, I. Chokshi, and S. K. Ghosh, "Analysis of attack graph-based metrics for quantification of network security," in *Proc. India Conference, IEEE*, 2012, pp. 530 – 535.

[19] L. Wang, A. Singhal, and S. Jajodia, "Toward measuring network security using attack graphs," in *Proc. ACM Workshop on Quality Protection*, 2007, pp. 49-54.

[20] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," in *Data and Applications Security XXII, Lecture Notes in Computer Science*, vol. 5094, 2008, pp. 283-296.

[21] L. Wang, A. Singhal, and S. Jajodia, "Measuring overall security of network configurations using attack graphs," *Data and Applications Security XXI*, vol. 4602, August 2007, pp. 98-112.

[22] A. Jaquith, *Security Metrics: Replacing Fear, Uncertainty, and Doubt*, Addison-Wesley, Pearson Education, 2007.

[23] K. Sallhammar, B. Helvik, and S. Knapskog, "On stochastic modeling for integrated security and dependability evaluation," *Journal of Networks*, vol. 1, 2006.

[24] T. Bass, "Intrusion detection system and multi-sensor data fusion," *Communications of the ACM*, vol. 43, no. 4 pp. 99-105, 2000.

[25] S. G. Bat sell, *etc.* (2005). Distributed Intrusion Detection and Attack Containment for Organizational Cyber Security. [Online]. Available: http://www.ioc.ornl.gov/projects/documents/containment.pdf

[26] J. Shifflet, "A technique independent fusion model for network intrusion detection," in *Proc. Midstates Conference on Undergraduate Research in Computer Science and Mat Hematics* , vol. 3, 2005, pp. 13-19.

[27] H. Q. Wang, *etc.*, "Survey of network situation awareness system," *Computer Science*, vol. 33, pp. 5-10, 2006.

[28] R. Ball and G. A. Fink, "Home-centric visualization of network traffic for security administration," in *Proc. ACM Workshop on Visualization and Data Mining for Computer Security*, Washington DC, October 2004, pp. 55-64.

[29] S. T. Teoh, K. L. Ma, S. Felix Wu, *et al.*, "Case study: Interactive visualization for internet security," in *Proc. IEEE VIS*, Boson, October 2002, pp. 505-508.

[30] C. Xiuzhen, Z. Qinghua, G. Xiaohong, and L. Chenguang, "Quantitative hierarchical threat evaluation model for network security," *Journal of Software*, vol. 17, no. 4, pp. 885−897, April 2006.

[31] S. Shaoyi and Z. Yongzheng, "A novel extended algorithm for network security situation awareness," in *Proc. International Conference on Computer and Management*, 2011, pp. 1-3.

[32] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing Networks and Markov Chain*s, New York: John Wiley & Sons, 1998.

[33] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science*, 2003, pp. 107.

[34] R. A. Sahner, K. S. Trivedi, and A. Puliafito, "Performance and reliability analysis of computer systems: An example-based approach using the sharpe software package," *Kluwer Academic Publishers*, 1996.

[35] O. Sheyner, J. W. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proc. IEEE Symposium on Security and Privacy*, 2002, pp. 273–284.

[36] O. Sheyner and J. Wing, "Tools for generating and analyzing attack graphs," in *Proc. Formal Methods for Components and Objects*, 2004, pp. 344–371.

[37] K. Lye and J. Wing, "Game strategies in network security," in *Proc. Foundations of Computer Security*, 2002.

[38] S. Jajodia and S. Noel, "Advanced cyber attack modeling, analysis, and visualization," Technical Report, George Mason University, Fairfax, VA, 2010.
[39] NuSMV. A New Symbolic Model Checker. [Online]. Available: http://nusmv.fbk.eu/
[40] R-Statistics Tool. The R Project for Statistical Computing. [Online]. Available: http://www.r-project.org/

Subil Abraham received his B.S. degree in Computer Engineering from the University of Kerala, India. He obtained his M.S. in Computer Science in 2002 from Southern Methodist University, Dallas, TX. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering at Southern Methodist University, Dallas, TX. His research interests include vulnerability assessment, network security, and security metrics.

Suku Nair received his B.S. degree in Electronics and Communication Engineering from the University of Kerala. He received his M.S. and Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana in 1988 and 1990, respectively.
Currently, he is the Chair and Professor in the Computer Science and Engineering Department at the Southern Methodist University at Dallas where he held a J. Lindsay Embrey Trustee Professorship in Engineering. His research interests include Network Security, Software Defined Networks, and Fault-Tolerant Computing. He is the founding director of HACNet (High Assurance Computing and Networking) Labs. He is a member of the IEEE and Upsilon Pi Epsilon.