Measuring Software Security Using SAN Models

Sadegh Dorri Nogoorani, Mohammad Ali Hadavi, Rasool Jalili
Department of Computer Engineering,
Sharif University of Technology,
Tehran, I.R.Iran
{dorri, mhadavi}@ce.sharif.edu, jalili@sharif.edu

***Abstract*—Security is one of the important issues in developing and implementing software systems especially in highly critical applications. Quantification and measurement of security is one of the approaches adopted to achieve the desired degree of security. In this paper, Stochastic Activity Networks (SANs) are used to formally model the attacks on the system under investigation. To this end, the semi-Markov attack model is sketched. Having the semi-Markov model, Probability of Attack Success (PAS), Mean Time to First Breach (MTFB), and System Misuse Proportion (SMP) are measured according to the appropriate transformation of the model to a SAN model. As a case study, we have studied a high-level attack on a password authentication subsystem. The results prove the applicability and suitability of our proposed method in making the compromise between security and other system requirements.**

Keywords- security quantification; security measurement; Stochastic Activity Network; password authentication

I. INTRODUCTION

Network-based software systems are becoming increasingly popular with the wide-spread coverage of computer networks and the Internet. In parallel with this popularity, security of these systems is becoming a major issue. The well-known security services such as authentication, authorization, and auditing are provided in a software system to fulfill its security needs. However, the efficiency of these services against attackers or in other words, the measurement of the security they provide, is the focus of this paper.

Modeling and formal verification of a software system as a whole is very complex and complicated, since modeling every detail of the system is very time-consuming, error-prone, and impractical. It is due to the huge amount of resources needed, and the problem of state space explosion. The complexity of attacker modeling, and dependency of the system behavior on the application environment must be added to the aforementioned challenges. Hence, usually a high-level model of the system is verified using formal methods. These methods can be utilized in two manners: verification and measurement. In the verification approach, some properties are specified with reference to the system specification. Then, verification techniques (such as model checking or theorem proving) are utilized to approve or disapprove the satisfaction of the properties by the system. In the measurement approach, the properties are graded instead of being completely true or false.

We have followed the measurement approach in this paper. More specifically, we are to quantify security of a system by means of formally measuring three security metrics (namely Probability of Attack Success, Mean Time to First Breach, and System Misuse Proportion). To this end, the system is first modeled using a semi-Markov model. Then this model is transformed to three different SAN models corresponding to the security metrics. Finally, the security metrics are measured according to the corresponding SAN model.

As a case study, we measure the three security metrics of an authentication subsystem. This choice has been made because this subsystem is the first, and probably the most important security service that a software system provides. We model the steps an attacker should take to compromise a password in a password-based authentication subsystem of a network-based software (attack model). Having the model in hand, the three security metrics are measured by transforming the model to appropriate SAN models. Our analyses on the results of the study confirm the applicability of our method. In addition, we have shown how a system architect can utilize these three measures to tune the system security with respect to other requirements such as ease-of-use and maintenance cost.

The rest of the paper is organized as follows. The background and related works are presented in Section II. Then, the general attack model and the security metrics are proposed in Section III. Section IV is devoted to a case study and the analysis of its security measurement results. Finally, the paper is concluded in Section V.

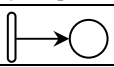
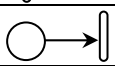
II. BACKGROUND AND RELATED WORKS

In this section we first give a brief overview of the Stochastic Activity Networks (SANs) and our notation in the SAN diagrams. Then we review the related works.

A. Stochastic Activity Networks (SANs)

Constructing a formal model of a system to evaluate its performance or dependability properties has a long history. System availability, processor utilization, duration of downtime, and system reliability can then be measured according to the formal model. Different variants of Petri nets are examples of state transition systems, developed to evaluate performability characteristics of computer systems. Stochastic Activity Networks (SANs) are probabilistic extensions of *activity networks*, or a stochastic generalization of Petri nets, that have been used to model the stochastic behavior of systems since mid-1980 [1,2]. The graphical representation of a SAN can facilitate the process of modeling. Table I shows the main graphical components of SANs. Some of these components have been used in this paper for our modeling.

TABLE I. GRAPHICAL COMPONENTS OF SANS

Name	Graphical Representation	Short Description
Places		Similar to variables and can contain tokens
Tokens (in a place)		The state of places (as the value of variables)
Transitions (timed, untimed)		change the number of tokens in places
Input arcs		Connect places to transitions
Output arcs		Connect a transition to a place
Input gates		Define complex enabling predicates and completion functions
Output gates		Define complex completion functions
Cases (small circles on activities)		Specify probabilistic choices for transitions
Instantaneous activities		Specify zero-time events

SANs support different activity time distributions such as exponential, deterministic, normal, gamma, beta, and uniform distributions, and are not restricted to uniform or exponential distributions. When a transition fires, for each input arc, a token is removed from the corresponding place. At the same time, a token is added to the place corresponding to the output arc. Each input gate is assigned a Boolean predicate as its enabling function, and an input function which specifies the rule for changing the state.

B. The Related Works

The related works on formal modeling and verification of the security of software systems can be classified into multiple categories. Mandatory Access Control (MAC) and Role-Based Access Control (RBAC) policies have previously been modeled using Colored Petri Nets [3,4]. These models are then analyzed based on their reachability graph. The reachability graph is also useful in detection of possible security policy breaches.

An Aspect Oriented extension of Petri nets has been proposed in [5] to verify security of software systems. This Aspect Oriented approach to security mitigations enables one to incrementally verify the security of the system. To this end, the potential threats to the insecure design are validated first of all. Then, it is proved that the final design is secure against those threats. Another advantage of this approach is that different security solutions can be compared with each other in order to choose the best alternative.

Another category of the related works is dedicated to the quantification of security with focus on modeling techniques. A general Colored Petri Net model has been proposed in [6] to quantify security system breaches from two aspects: system failure due to usual or random use, or intended attacks. In [7], Jonsson and Olovsson have quantitatively analyzed the attacker behavior. The analysis is based on empirical data, and shows that the attacker behavior usually consists of three phases of *learning*, *standard*, and *innovative*. According to this classification, the possibility of success of the attacker in the standard phase is greater than the other two phases. The study shows that the time between the detection of security breaches in the standard phase has exponential distribution.

Almasizadeh *et al*. have proposed a method for modeling and quantifying the effects of attacks on computer systems [8]. Attacks in [8] are defined as a series of consecutive steps. The transition times between these steps are defined to be uniform within an interval proportional to their required effort. In each transitive step of an intrusion, the attacker gains new privileges. The transitions between the steps form a semi-Markov model which is then transformed to a Discrete Time Markov Chain (DTMC). Finally, the DTMC is analyzed and the probability of attacker success is calculated.

Almasizadeh *et al*. have proposed an improved model over [8] in [9]. In this improved model, the possibility of system recovery after security breach has been considered thanks to a transition from the *security breach* state to the *secure* state. The parameters of this new model are setup according to the type of attack, the expertise of the attacker, and the security of the system. In addition to the more realistic model, Almasizadeh *et al*. have measured a more comprehensive set of security metrics. Namely, Mean Time to First Security Failure (MTFSF) and the steady state probabilities of being in each state. In order to evaluate these factors, the semi-Markov model is transformed to its embedded DTMC. For steady-state analysis, all states are converted to transient states and the steady-state probabilities are calculated.

Wang *et al.* [10] propose a multiple queuing model to mathematically analyze the performance metrics of an email system under three types of attacks. They have considered three types of attacks (cracking password, malicious emails, and email bombs) in addition to ordinary emails as the input and output processes of a multiple queuing system. Hence, there are four waiting lines consisting of ordinary emails and the attacks, each of which has an independent Poisson arrival rate. The queuing model is an irreducible continuous-time QBD process. In order to analyze the security, email system *availability* is measured against email bombs. Moreover, *the probability of information leakage* (a key metric of an email system security) is measured in face of cracking password and malicious email attacks. Wang *et al.* use numerical examples to show the effectiveness and efficiency of their solution for security analysis of an email system.

The closest related works to our proposal are [8,9]. In contrast to these proposals, we do not transform our models to DTMCs but instead, we use SANs to analyze the semi-Markov models. SANs have greater flexibility and there are automated tools to calculate the desired model outputs. Because of the more flexible underlying model, our attack models are not restricted to the ones suggested in [8,9]. In particular, transitions between every two states are possible (according to the attack model). In addition, the transition times in a SAN are not restricted to particular distributions, and can be tailored to real modeling needs.

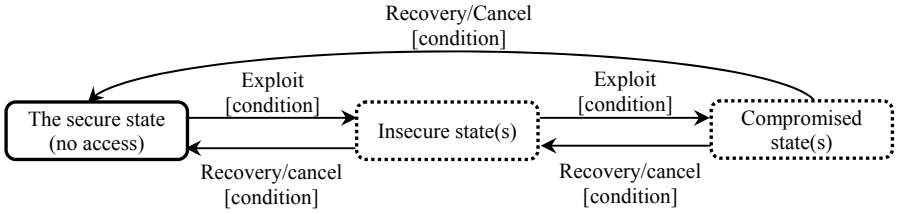


Figure 1. The general semi-Markov attack model

III. THE GENERAL ATTACK MODEL

An attack model is a semi-Markov process. States in this model correspond to the privileges the attacker holds. A transition between states indicates gaining or losing privileges by the attacker. In particular, there are three classes of states (secure, insecure, compromised), and three classes of transitions (exploit, recover, cancel) in an attack model (see Fig. 1). The initial state of the model is assumed to be the only *secure* state, and a *compromised* state corresponds to a successful security breach, and in-between states are all *insecure* ones. A transition of an attacker from a less privileged state to a more privileged one is called an *exploit* transition. The auditing and security response actions in the system may intentionally *recover* the system and push the attacker to a less privileged, or to the secure state. Unintended prevention actions taken by users or administrators, such as regularly changing passwords, *cancel* exploits and push the attacker back to less privileged states. The transitions can be conditional and their activation dependent on the context and environment.

For example, Fig. 2 illustrates a password compromise attack. In this figure, the *No access* state is the *secure* state, *Pwd compromised* is the only *compromised* state, and the others are *insecure* ones. All transitions from insecure and compromised states to the *No access* state are *recovery* transitions, while *Pwd change* transitions are *cancel* transitions, and the others are *exploit* ones.

The transition time from the *secure* to an *insecure* or *compromised* state represents the startup time of an attack. The distribution time of *recovery* transitions account for the time taken to discover and recover from an *insecure* or *compromised* state to a safer state, and is directly dependent on the effectiveness of auditing and recovery mechanisms. On the other hand, the distribution time of the *exploit* transitions reflect the security of the system as well as the expected expertise of an attacker. All these parameters should be set with reference to empirical data, or a domain and security expert.

A. Security Metrics

Having an attack model in hand, the following security metrics are measured:

1. *Probability of Attack Success (PAS)*: The probability that an attack is successful. In other words, the probability that an attacker starting from the *secure* state is not trapped by auditing and protection mechanisms, and reaches the *compromised* state. In this regard, every attack is either successful or failed.
2. *System Misuse Proportion (SMP)*: The proportion of time that the system is in the *compromised* state and the attacker can misuse the system.
3. *Mean Time to First Breach (MTFB)*: The expected time for the first successful security breach. In other words, the expected time for an attacker to reach the *compromised* state (for the first time) starting from the *secure* state.

B. Measurement

In order to evaluate the security metrics, the attack model is transformed to a corresponding SAN model. MTFB and PAS are transient in nature, while SMP must be evaluated in steady state. Hence, different SAN models are required for each metric.

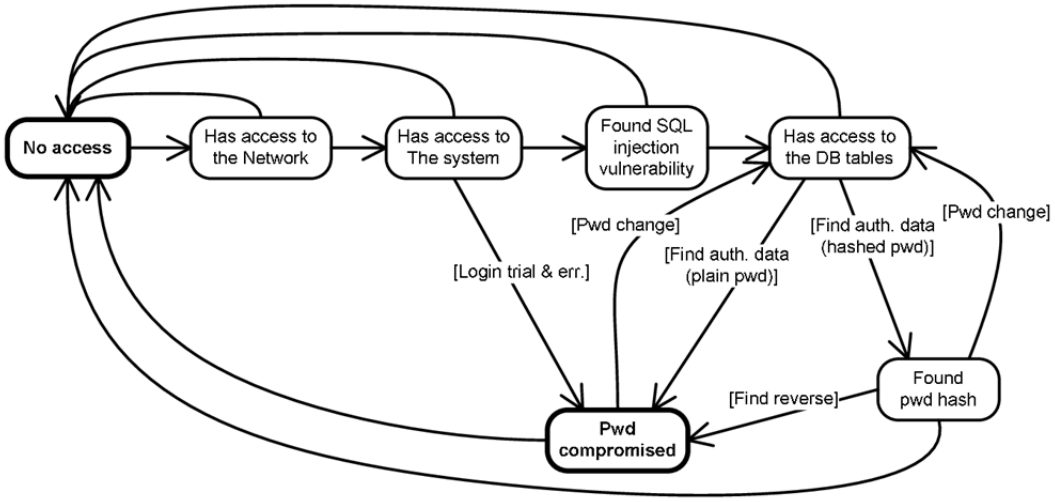


Figure 2. The semi-Markov model of password compromise via access to the system database

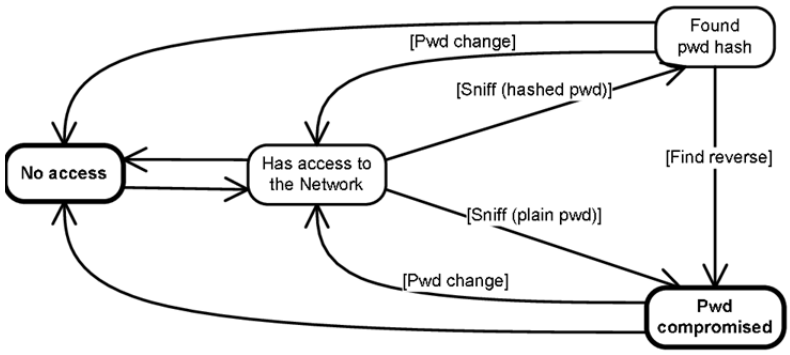


Figure 3. The semi-Markov attack model

The SMP-SAN is the most resembling SAN model to the attack model. It is the result of the following transformation:

1. Each state in the attack model has a corresponding state in the SAN model.
2. Unconditional transitions in the attack model translate to a timed transition in the SAN model and connected to the source and destination states via input and output arcs.
3. Conditional transitions in the attack model translate to cases and gates in the SAN model.

The following step is additionally required to construct the MTFB-SAN:

4. All *compromised* states are merged into one trapping state (no out-going transition).

One more step is required to create the PAS-SAN:

5. All recovery transitions are redirected to a new trapping state which is the *Attack failed* state.

All SAN models are initialized with one token in the *secure* state (one attacker) and the metrics are measured as follows:

1. PAS (transient): The token transits between states until it is trapped in the *compromised* state (successful attack) or in the *Attack failed* state (failed attack). PAS is calculated according to (1) where n_s is the number of successful attacks and n is the total number of trials.

$$PAS = \frac{n_s}{n} \quad (1)$$

2. SMP (steady-state): The token transits between the states and the steady-state probabilities of being in one of the *compromised* states are calculated. Hence, SMP is calculated according to (2) where S_{comp} is the set of *compromised* states.

$$SMP = \sum_{q \in S_{comp}} P_q \quad (2)$$

3. MTFB (transient): The token transits between the states until it reaches the *compromised* state (trapping). MTFB is calculated according to (3) where t_{FB} is the time to first breach.

$$MTFB = E[t_{FB}] \quad (3)$$

IV. Case Study

We model a simple password-guess attack as a case study. The attacker in our model sniffs network packets and tries to reverse hashed passwords. We measure the security of an example system under this type of attack with various configuration settings, depending on the level of auditing and password policies of the system. The measurements enable one to trade-off between ease-of-use, auditing and maintenance cost, and security.

A. The Attack Model

The semi-Markov attack model is depicted in Fig. 3. The *No access* state is the *secure* state, *Pwd compromised* the only *compromised* state, and the other states are *insecure* ones. The transition distributions are assumed to be as in Table II. The distributions in Table II are some reasonable guesses and can be replaced with empirical distributions suggested by real-world data.

The transition times are assumed to be uniform and in some cases deterministic (as assumed in [8,9]). The uniform distribution is an Increasing Failure Rate (IFR) distribution. This property is in harmony with the fact that the probability of success is increasing with the amount of time spent on the attack. In addition, the bounds of the distribution can be determined empirically or intuitively. The exploit transitions are dependent on the assumed expertise of the attacker, and the *cancel* and *recovery* transitions account for the strength of the system and effectiveness of the mitigation mechanisms.

B. SAN Models

The SMP-SAN is depicted in Fig. 4. This SAN is closely analogous to the model of Fig. 3. The other SANs in Fig. 5 and Fig. 6 have slightly different transitions, or states.

C. Measurement Results

In this section the results of measuring the three security metrics of the case under study are presented. The *PwdChange* and *Reverse* transitions have uniform time distributions which are determined according to the password-change policy of the system. In our study, three different policies are assumed and the time bounds are determined according to the policy.

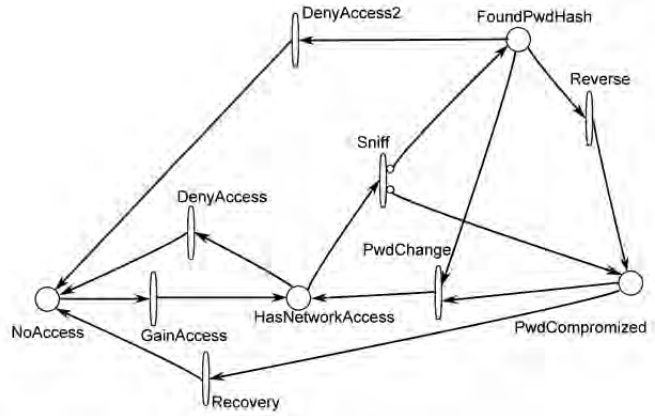


Figure 4. SMP-SAN

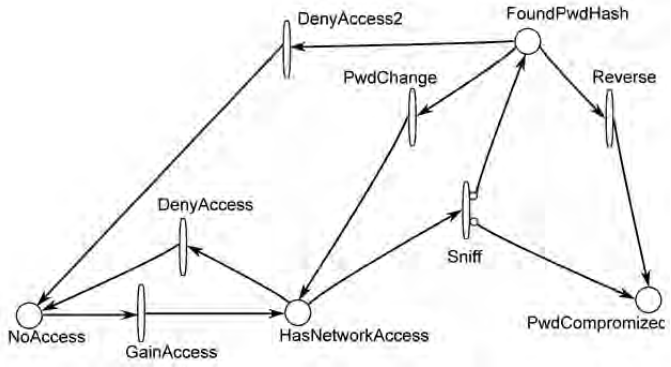


Figure 5. MTFB-SAN

The *DenyAccess*, *DenyAccess2*, and *Recovery* transitions are dependent on the auditing and intrusion detection policies of the system. We have evaluated the security metrics with three different auditing levels (weak, mild, strict) and compared the results. We assumed that passwords are hashed before being transmitted over the network.

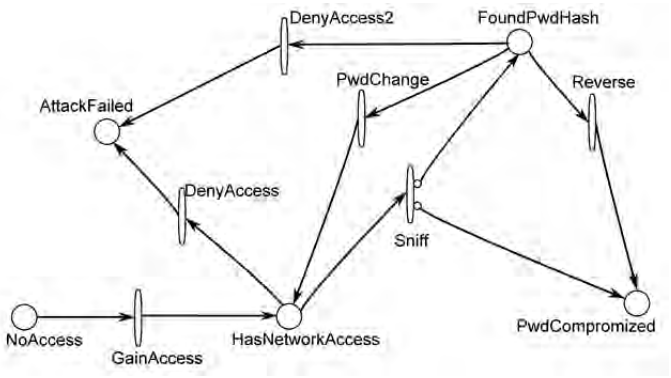


Figure 6. PAS-SAN

Fig. 7 illustrates the results of measuring PAS with reference to PAS-SAN (Fig. 6). In this figure, the effects of varying *Auditing Policies* (AP) and *Password Strength* (PS) can be seen. The three values of 0, 1, and 2 correspond to weak, mild, and strong passwords, respectively. According to the results, *Auditing Policies* and *Password Strength* have direct effect on PAS. However, as can be seen in Fig. 8, *Password Change Policy* (PCP) has no effect on PAS. Our evaluations show that PCP has no effect on the other security metrics, too.

Fig. 9 depicts the changes in SMP by varying AP and PS. SMP has inverse relation to AP and PS.

MTFB of the system with various system settings are evaluated and depicted in Fig. 10. According to this figure, PS has greater impact on MTFB than AP. It should be noted that 8,000 hours was the upper bound on the run time of the SAN model. Hence, MTFB with strong AP and PS is greater than 8,000 hours.

TABLE II. TRANSITION TIME DISTRIBUTIONS (IN HOURS)

Transition	**Distribution**	**Distribution Parameters**		
		Description	***Lower Bound***	***Upper Bound***
GainAccess	Uniform	--	50	300
PwdChange	Uniform/deterministic	(0) Weak: Uniform	6 months	5 years
		(1) Mild: Deterministic	6 months	
		(2) Strict: Deterministic	1 month	
Sniff	Uniform	--	10	50
Reverse	Uniform (dependent on Password Strength)	(0) Weak PS	1	10
		(1) Mild PS	10	200
		(2) Strict PS	200	1000
DenyAccess	Uniform (dependent on Auditing Policy)	(1) Weak AP	450	3450
		(2) Mild AP	10	450
		(3) Strict AP	0	50
DenyAccess2	Uniform (dependent on Auditing Policy)	(1) Weak AP	400	3300
		(2) Mild AP	0	400
		(3) Strict AP	0	20
Recovery	Uniform (dependent on Auditing Policy)	(1) Weak AP	600	3600
		(2) Mild AP	140	600
		(3) Strict AP	20	200

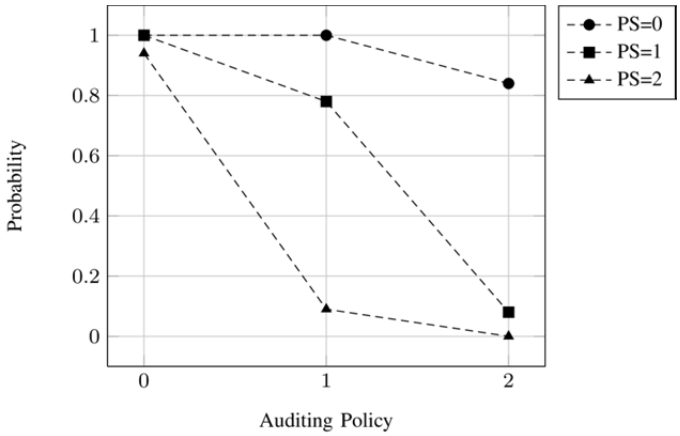


Figure 7. Probability of Attack Success (PAS)

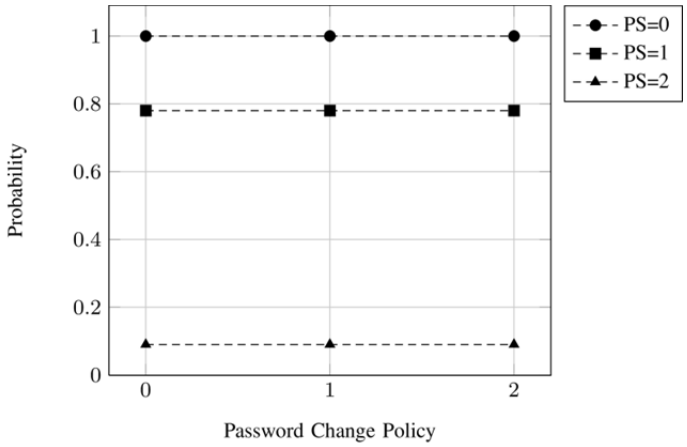


Figure 8. Probability of Attack Success (PAS) with Auditing Policy = 1

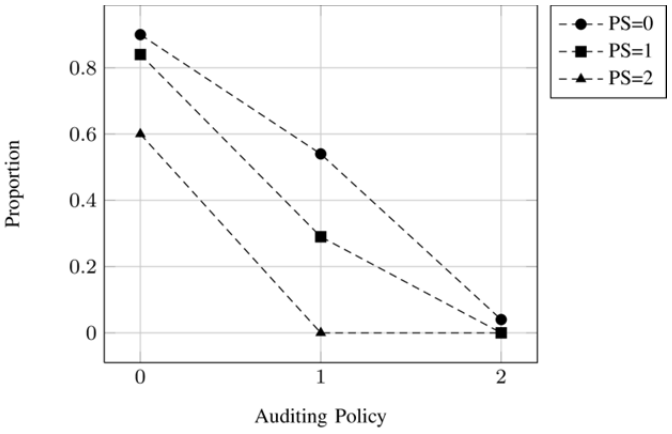


Figure 9. System Misuse Proportion (SMP)

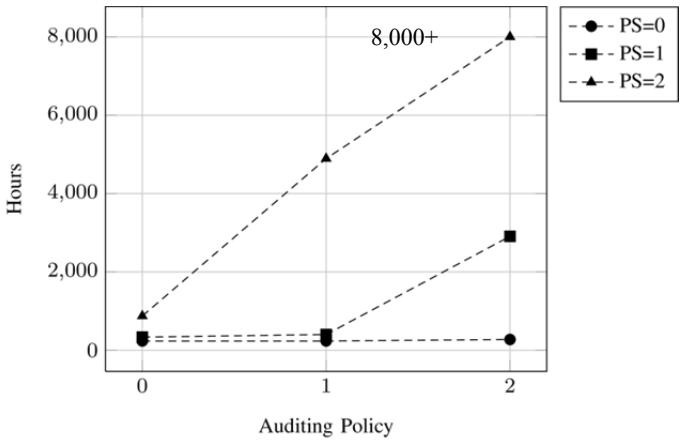


Figure 10. Mean Time to First Breach (MTFB) in hours

D. Discussions

Three configuration settings were changed during the study: Auditing Policy (AP), Password Strength (PS), and Password Change Policy (PCP). According to the results, PCP doesn't affect the security metrics. It may be because even a strict PCP forces the users to change their passwords at most once in a month. Comparing this setting with the other settings, they have more immediate impact than PCP. On the other hand, mandating the users to change their passwords with a higher frequency has a bad effect on user friendliness, and may make the policy ineffective because users will reuse their passwords with high probability. This result can hint the administrators to focus on more effective ways of improving system security while not annoying the users with the PCP.

The other two settings have meaningful impacts on the security metrics. We can conclude that the weak AP or PS is not reasonable, and using strong passwords leads to a good level of security even with the mild AP.

V. CONCLUSIONS AND FUTURE WORK

Security of network-based software systems is becoming increasingly important as they are being used in different aspects of our daily life such as business, information, and learning systems. It is even more critical in some specific areas such as financial and military applications. Quantitative analyses of security metrics, e.g. misuse probability of a system or mean time to first failure, provide a more reliable view to the system security, compared to traditional subjective qualitative evaluations.

We proposed to model an attack scenario by means of a semi-Markov model. The model is then transformed to various SAN models. These SAN models enabled us to evaluate system security metrics. The results of measuring these metrics in a password compromise attack case highlighted the applicability and usefulness of our proposed method. In addition, we showed how a system administrator can utilize the results in order to configure system settings.

Our approach can account for different environmental settings, e.g. the skill level of adversaries or the level of system auditing, by changing the transition distributions or parameter values according to empirical data. For example, the upper bound of the uniform distribution of a transition time corresponds to the level of system exposure for attackers,

and its lower bound corresponds to the effectiveness of preventive techniques.

Our password compromise case study can also be improved by modeling attackers' behavior with more accurate estimates of distributions and their parameters, and consequently more accurate analyses and measurements. Exploring different attack scenarios with respect to other software security services as well as the authentication service is another future work to extend our current research.

ACKNOWLEDGMENT

We would like to hereby acknowledge Prof. Ali Movaghar who made us familiar with SANs and provided us with their software tools.

REFERENCES

[1] J.F. Meyer, A. Movaghar, and W. H. Sanders, "Stochastic activity networks: structure, behavior and application," *Int. Workshop on Timed Petri Nets*, 1985, pp. 106-115.

[2] W.H. Sanders and J. F. Meyer, "Stochastic activity networks: formal definitions and concepts," *Lec. Formal Methods and Performance Analysis, LNCS*, vol. 2090, Springer-Verlag, 2001, pp. 315-343.

[3] K. Juszczyszyn, "Verifying enterprise's mandatory access control policies with coloured Petri nets," *12th IEEE Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003, pp. 184-189.

[4] A. Walvekar et al., "Using Petri nets to detect access control violations in a system of systems," *Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis*, 2006.

[5] D. Xu and K.E. Nygard, "Threat-driven modeling and verification of secure software using aspect-oriented Petri nets," *IEEE Trans. Software Eng.*, vol. 32, no. 4, April 2006, pp. 265-278.

[6] S.H. Houmb and K. Sallhammar, "Modeling system integrity of a security critical system using coloured Petri nets," *1st Int. Conf. Safety and Security Eng.*, 2005, pp. 3-12.

[7] E. Jonsson and T. Olovsson, "A quantitative model of the security intrusion process based on attacker behavior," *IEEE Trans. Software Eng.*, vol. 23, no. 4, April 1997, pp. 235-245.

[8] J. Almasizadeh and M. A. Azgomi, "A new method for modeling and evaluation of the probability of attacker success," *Int. Conf. Security Technology*, 2008, pp. 49-53.

[9] J. Almasizadeh and M. A. Azgomi, "Intrusion process modeling for security quantification," *4th Int. Conf. Availability, Reliability and Security*, 2009, pp. 114-121.

[10] Y. Wang, C. Lin, and Q.-L. Li, "Performance analysis of email systems under three types of attacks," *Performance Evaluation*, vol. 67, 2010, pp. 485-499.