Towards Quantifying the (In)Security of Networked Systems

Xiaohu Li T. Paul Parker Shouhuai Xu
Department of Computer Science, University of Texas at San Antonio
{xli,tparker,shxu}@cs.utsa.edu

Abstract

Traditional security analyses are often geared towards cryptographic primitives or protocols. Although such analyses are absolutely necessary, they do not provide much insight for answering an equally important question: What is the security assurance of a physically or logically networked system when we consider it as a whole? *This question is known to be notoriously difficult, and the state-of-the-art is that we know very little about it. In this paper, we make a step towards resolving it with a new modeling approach.*

1 Introduction

Traditional security analyses are typically geared towards cryptographic primitives or protocols. Although such analyses are absolutely necessary, they do not provide much insight for answering an equally important question: *What is the security assurance of a physically or logically networked system when we consider it as a whole?* By "physically networked system" we mean a physical network, where nodes are computers and edges are point-to-point links. By "logically networked system" we mean a logical network, where the nodes are users and edges represent certain (e.g., trust) relationships between the nodes. The state-of-the-art is that we know very little about the answer to the above question, which is notoriously known to be difficult but has important practical value.

1.1 Our Contributions

In this paper we make a step towards quantifying the (in)security of networked systems in terms of metrics such as the number of compromised nodes, private keys or digital identities. Our approach is stochastic modeling, which is appropriate because successful attacks as well as their detections do not seem to be deterministic events in real-life systems.

Specifically, we present three concrete models for investigating the (in)security of physically or logically networked systems. In the first model (Section 3), we make some strong (but still realistic in some cases) assumptions about the system attributes, which allows us to derive some concise analytic result. The second model (Section 4) is more powerful than the first one, in that it can capture the heterogeneity that different nodes may have different degrees. The third model (Section 5) is even more powerful because it can accommodate full heterogeneity. For the second and third model, it is very difficult to derive closed-form analytic results. Nevertheless, we manage to derive some useful results: (1) It is unlikely that, in realistic networked systems, one can ever control the number of the compromised nodes to below an arbitrary threshold. (2) We give some sufficient conditions under which the expected number of compromised nodes doesn't oscillate. This means that the (in)security of a networked system can be measured in terms of the number of compromised nodes when the system enters the steady state.

We conduct a case study based on the PGP certificate graph [14]. We treat the public keys as the nodes, and the certification relationship between two nodes as a directed edge. As we will see, our models are accurate. Further, for certain scenarios, we derive the optimal investment strategies to minimize the number of compromised nodes when a system enters the steady state.

Outline: In Section 2 we specify the system setting. In Sections 3 - 5 we explore the three models from the simplest to the most complex. In Section 6 we conduct a case study based on PGP. We discuss related work in Section 7, and conclude the paper in Section 8. Due to space limitation, many details are deferred to the full version of the present paper [8].

2 System Setting and Notations

A system is modeled as $G = (V, E)$, a finite graph with $V = \{1, 2, \ldots, n\}$ the set of nodes (i.e., users or vertices) and E the set of edges. We stress, however, that an edge $(u, v) \in E$ does *not* necessarily correspond to the *physical* link between nodes u and v. Instead, it could be a logical concept that captures, for instance, that the compromise of

node u may lead to the compromise of node v with a non-zero probability. A graph G may be directed or undirected; all the results in this paper are equally applicable to both cases. For $u, v \in V$, let I be the predicate that $I(u,v) = 1$ if $(u,v) \in E$, and 0 otherwise. Let A be the adjacency matrix of graph G. For a matrix M, M^T denotes its transpose.

We are interested in discrete time models for time $t = 0, 1, 2, \ldots$. At any time t, a node is always in one of two states: secure and compromised. For simplicity, we assume that once a node becomes compromised, all its secrets (including its cryptographic keys) are compromised. We also assume that a compromised node remains so until the attack has been detected (e.g., by an intrusion detection system) and thus appropriate measures have been taken (e.g., the cryptographic keys are revoked). On one hand, a node may become compromised because of its own reasons (e.g., the user downloads and runs a malicious code) or because some of its trusted nodes have become compromised. To capture the former, we associate a node v with a parameter α_v, which is the probability that a secure node becomes compromised at a discrete time step because of its own reasons. To capture the latter, we associate each edge $(u,v) \in E$ with a parameter, γ_{vu}, which may reflect v's degree of trust in u, or the capability of an infection from a compromised u to a secure v. Naturally, in the case of undirected graphs, $\gamma_{vu} = \gamma_{uv}$ for all $(u,v) \in E$, whereas it is not necessarily true in the case of directed graphs. On the other hand, a compromised node v may become secure because of the detection of the attack. For this we associate v with a parameter β_v, which is the probability that a compromised node v becomes secure at a discrete time step.

Let S_t be the random variable indicating the total number of secure nodes at time t, and C_t be the random variable indicating the total number of compromised nodes at time t. Further, let $s_v^{(t)}$ be the probability that node v is in the state of secure at time t, and $c_v^{(t)}$ be the probability that node v is in the compromised state at time t. We note that for all t, $s_v^{(t)} + c_v^{(t)} = 1$ and $S_t + C_t = n$.

3 Model I: Basic Case

In this model, we assume that $E = \emptyset$, meaning that compromise of a node does not increase the probability of another node being compromised. Equivalently, this assumption can be interpreted as $\gamma_{vu} = 0$ for all $(u,v) \in E$. Further, we assume that $\alpha_v = \alpha$ and $\beta_v = \beta$ for all $v \in V$.

Figure 1 shows the state transition of a node v in this model. Specifically, at each discrete time step, each node independently changes from secure to compromised with probability α, or from compromised to secure with probability β. As a result, we have

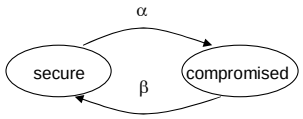


Figure 1. State transition in the basic case

$$\begin{cases} s_v^{(t+1)} &= (1-\alpha)s_v^{(t)} + \beta c_v^{(t)} \\ c_v^{(t+1)} &= \alpha s_v^{(t)} + (1-\beta)c_v^{(t)}. \end{cases} \tag{3.1}$$

Denote by

$$\Lambda = \begin{pmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{pmatrix},$$

then (3.1) can be rephrased as

$$\begin{pmatrix} s_v^{(t)} \\ c_v^{(t)} \end{pmatrix} = \Lambda \begin{pmatrix} s_v^{(t-1)} \\ c_v^{(t-1)} \end{pmatrix} = \Lambda^t \begin{pmatrix} s_v^{(0)} \\ c_v^{(0)} \end{pmatrix} = \Lambda^t \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

It holds that

$$c_v^{(t)} \;=\; \frac{1}{1+\frac{\beta}{\alpha}}\left[1-(1-\alpha-\beta)^t\right]. \tag{3.2}$$

Hence, the total number of compromised nodes at time t, C_t, follows the binomial distribution with parameter $(|V|, c_v^{(t)})$, namely

$$\Pr[C_t = i] = \binom{|V|}{i}\left[c_v^{(t)}\right]^i\left[s_v^{(t)}\right]^{n-i}, \quad i = 0, 1, 2 \cdots, n, \tag{3.3}$$

and its mean and variance are

$$\mathsf{E}[C_t] = \mathsf{E}\left[\sum_{v\in V} c_v^{(t)}\right] = |V| \cdot c_1^{(t)}, \tag{3.4}$$

$$\mathsf{Var}[C_t] = \mathsf{Var}\left[\sum_{v\in V} c_v^{(t)}\right] = |V| \cdot c_1^{(t)} \cdot s_1^{(t)}. \tag{3.5}$$

3.1 Analysis

Observe that $0 < \alpha+\beta < 2$, meaning that $|1-\alpha-\beta| < 1$ is always valid in (3.2). Thus, as $t \to \infty$,

$$c_v^{(t)} \to \frac{\alpha}{\alpha+\beta}. \tag{3.6}$$

The above brings us the following insight: $\mathsf{E}[C_t] \to |V| \times \frac{\alpha}{\alpha+\beta}$, namely that at any sufficiently large time t, the number of compromised nodes in the system becomes steady. Furthermore,

$$\frac{\partial(\lim_{t\to\infty} c_v^{(t)})}{\partial\alpha} = \frac{\beta}{(\alpha+\beta)^2} > 0,$$

which means that the system becomes more secure as α decreases. Also, it is clear that the system becomes more secure as β increases (meaning, for example, that the users deploy some global alert service and patching system). In addition to the aforementioned important aspect of the system's overall security, as we will see in Section 6, this also facilitates the investigation of the optimal strategy in investing a given budget to reduce α, increase β, or both.

4 Model II: Semi-heterogeneous Case

In Model I, we abstracted away the attributes of the edges in the systems. In this section, we present a model for systems that can be modeled as semi-heterogeneous graphs, where by "semi-heterogeneous" we mean that different nodes may have different degrees. Further, we assume that $\gamma_{vu} = \gamma$ for any $(u, v) \in E$.

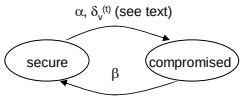


Figure 2. State transition in the semi-heterogeneous case

Figure 2 shows the state transition of a node v in this model. Specifically, at any time $t+1$, v may become compromised because of its own reason (with probability α) or some of its trusted nodes (with probability $\delta_v^{(t)}$). Then,

$$\delta_v^{(t)} = 1 - \prod_{(u,v)\in E} \left[1 - \gamma c_u^{(t)}\right]. \tag{4.1}$$

Further, we have

$$\begin{cases} s_v^{(t+1)} &= \left[(1-\alpha)(1-\delta_v^{(t)})\right] s_v^{(t)} + \beta c_v^{(t)} \\ c_v^{(t+1)} &= \left[1-(1-\alpha)(1-\delta_v^{(t)})\right] s_v^{(t)} + (1-\beta) c_v^{(t)}. \end{cases} \tag{4.2}$$

This formula gives the expected number of compromised nodes at any time t, namely $\mathsf{E}[C_t]$.

4.1 Analysis

Besides being able to accurately predict the metrics of interest, we observe that $\mathsf{E}[C_t]$ in the present model is strictly larger than $\mathsf{E}[C_t] \to |V| \cdot \frac{\alpha}{\alpha+\beta}$ as given by (3.4) and (3.6). This means that there are always compromised nodes, unless $\alpha = 0$. In what follows we draw a sufficient condition, under which $\mathsf{E}[C_t]$ converges as $t \to \infty$.

By omitting those items of power no less than 2, (4.2) can be rearranged as below,

$$c_v^{(t)} \approx \alpha + (1-\alpha-\beta)c_v^{(t-1)} + \gamma \sum_{(u,v)\in E} c_u^{(t-1)}.$$

Denote by

$$C(t) = \begin{pmatrix} c_1^{(t)} \\ \vdots \\ c_n^{(t)} \end{pmatrix}, \ \mathbf{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \ \mathsf{I} = \begin{pmatrix} 1 & \cdots & 0 \\ 0 & \vdots & 0 \\ 0 & \cdots & 1 \end{pmatrix},$$

then, we have

$$C(t) = \alpha \left(\sum_{j=0}^{t-1} \mathsf{S}^j \right) \mathbf{1} + \mathsf{S}^t \times C(0)$$

where

$$\mathsf{S} = (1-\alpha-\beta)\mathsf{I} + \gamma \mathsf{A}^T \tag{4.3}$$

and A is the adjacency matrix of the network.

Denote by $\lambda_{1,\mathsf{A}} \geq \lambda_{2,\mathsf{A}} \geq \cdots \geq \lambda_{n,\mathsf{A}}$ the ordered eigenvalues of A (i.e., of A^T). It is straightforward to see, for $i = 1, \cdots, n$, if $\lambda_{i,\mathsf{A}}$ is the i^{th} eigenvalue of A then

$$\lambda_{i,\mathsf{S}} = (1-\alpha-\beta) + \gamma \lambda_{i,\mathsf{A}} \tag{4.4}$$

corresponds to the i^{th} eigenvalue of S. Denote by $\mathsf{u}_{1,\mathsf{A}}, \ldots, \mathsf{u}_{n,\mathsf{A}}$ the orthogonal unit eigenvectors corresponding to the eigenvalues $\lambda_{1,\mathsf{A}}, \ldots, \lambda_{n,\mathsf{A}}$. Note that (4.4) implies that, for any $1 \leq i \leq n$, if x is the eigenvector corresponding to eigenvalue $\lambda_{i,\mathsf{A}}$, then x is also the eigenvector corresponding to $\lambda_{i,\mathsf{S}}$. In other words, S and A have common eigenvectors. By the spectral decomposition, it holds that

$$\mathsf{S} = \sum_{i=1}^{n} \lambda_{i,\mathsf{S}} \left(\mathsf{u}_{i,\mathsf{A}} \times \mathsf{u}_{i,\mathsf{A}}^T \right)$$

and hence

$$\mathsf{S}^j = \sum_{i=1}^{n} \lambda_{i,\mathsf{S}}^j \left(\mathsf{u}_{i,\mathsf{A}} \times \mathsf{u}_{i,\mathsf{A}}^T \right).$$

It can be shown that

$$\begin{aligned} C(t) &= \alpha \mathbf{1} + \alpha \left[\sum_{i=1}^{n} \frac{\lambda_{i,\mathsf{S}}(1-\lambda_{i,\mathsf{S}}^{t-1})}{1-\lambda_{i,\mathsf{S}}} \left(\mathsf{u}_{i,\mathsf{A}} \times \mathsf{u}_{i,\mathsf{A}}^T \right) \right] \times \mathbf{1} \\ &\quad + \left(\sum_{i=1}^{n} \lambda_{i,\mathsf{S}}^t \left(\mathsf{u}_{i,\mathsf{A}} \times \mathsf{u}_{i,\mathsf{A}}^T \right) \right) \times C(0). \end{aligned}$$

In order for the probability vector $C(t)$ to converge (i.e., the system eventually enters the steady state), a *sufficient* condition is that $\lambda_{i,\mathsf{S}}^t \to 0$ as $t \to \infty$. This is equivalent to requiring

$$\max_{1\leq i\leq n} |\lambda_{i,\mathsf{S}}| < 1, \tag{4.5}$$

which is equivalent to requiring

$$|\lambda_{1,\mathsf{S}}| < 1 \text{ and } |\lambda_{n,\mathsf{S}}| < 1. \tag{4.6}$$

Since the trace of A (i.e., the sum of the elements on the main diagonal of A) is 0, which equals to the sum of the eigenvalues of A, it holds that $\lambda_{1,\mathsf{A}} > 0$ and $\lambda_{n,\mathsf{A}} < 0$. Therefore, (4.6) is equivalent to

$$-1 < 1-\alpha-\beta+\gamma\lambda_{n,\mathsf{A}} < 1-\alpha-\beta+\gamma\lambda_{1,\mathsf{A}} < 1.$$

This means

$$\frac{\alpha+\beta-2}{\gamma} < \lambda_{n,\mathsf{A}} \text{ and } \lambda_{1,\mathsf{A}} < \frac{\alpha+\beta}{\gamma}.$$

Under the above sufficient condition, we have, as $t \to \infty$,

$$C(t) \to \alpha\mathbf{1} + \alpha\left[\sum_{i=1}^{n}\left(\frac{1}{\alpha+\beta-\gamma\lambda_{i,\mathsf{A}}}-1\right)\left(\mathsf{u}_{i,\mathsf{A}}\times\mathsf{u}_{i,\mathsf{A}}^T\right)\right]\times\mathbf{1}.$$

Therefore, the number of compromised nodes is

$$\mathsf{E}[C_t] = \mathbf{1}^T C(t) \to n\alpha + \alpha\sum_{i=1}^{n}\left[\left(\frac{1}{\alpha+\beta-\gamma\lambda_{i,\mathsf{A}}}-1\right)\left(\mathsf{u}_{i,\mathsf{A}}^T\times\mathbf{1}\right)^2\right].$$

The above brings us the following insight. Since

$$\frac{\partial\left(\lim_{t\to\infty}\mathsf{E}[C_t]\right)}{\partial\beta} = \alpha\sum_{i=1}^{n}\frac{-\left(\mathsf{u}_{i,\mathsf{A}}^T\times\mathbf{1}\right)^2}{\alpha+\beta-\gamma\lambda_{i,\mathsf{A}}} < 0,$$

for sufficiently large t, $\mathsf{E}[C_t]$ decreases as β (i.e., the curing capability) grows. Further, if A is positive definite (i.e., $\lambda_{i,\mathsf{A}} > 0$ for all i), then

$$\frac{\partial\left(\lim_{t\to\infty}\mathsf{E}[C_t]\right)}{\partial\gamma} = \alpha\sum_{i=1}^{n}\frac{\lambda_{i,\mathsf{A}}\left(\mathsf{u}_{i,\mathsf{A}}^T\times\mathbf{1}\right)^2}{\alpha+\beta-\gamma\lambda_{i,\mathsf{A}}} > 0,$$

meaning that for sufficient large t, $\mathsf{E}[C_t]$ increases as γ (i.e., the edges infection capability) grows.

5 Model III: Fully-heterogeneous Model

Model II captures semi-heterogenous systems. In this section, we present a model for fully-heterogeneous systems, where by "fully-heterogeneous" we mean that each edge $(u,v) \in E$ may be associated with γ_{vu} that reflects its own attributes, and each node v may be associated with α_v and β_v that reflect its own attributes.

Figure 3 shows the state transition of a node n. Specifically, v becomes compromised because of its own reason (with probability α_v), or because of some of its compromised trusted nodes (with probability $\delta_v^{(t)}$). Further, v may become secure with probability β_v. Then,

$$\delta_v^{(t)} = 1 - \prod_{(u,v)\in E}\left[1-\gamma_{vu}c_u^{(t)}\right]. \tag{5.1}$$

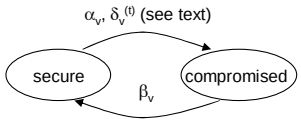


Figure 3. State transformation in the fully-heterogeneous case

Further, we have

$$\begin{cases} s_v^{(t+1)} &= \left[(1-\alpha_v)(1-\delta_v^{(t)})\right]s_v^{(t)} + \beta_v c_v^{(t)} \\ c_v^{(t+1)} &= \left[1-(1-\alpha_v)(1-\delta_v^{(t)})\right]s_v^{(t)} + (1-\beta_v)c_v^{(t)}. \end{cases} \tag{5.2}$$

This formula immediately gives the expected number of compromised nodes at any time t.

5.1 Analysis

In this subsection, we establish a sufficient condition for $\mathsf{E}[C_t]$ to converge as $t \to \infty$. As we will see, it is also true that $\mathsf{E}[C_t] > 0$ as $t \to \infty$, unless $\alpha_v = 0$ for all $v \in V$.

By omitting the nonlinear items, (5.2) implies

$$c_v^{(t)} \approx \alpha_v + (1-\alpha_v-\beta_v)c_v^{(t-1)} + \sum_{(u,v)\in E}\gamma_{vu}c_u^{(t-1)}.$$

Denote by

$$C(t) = \begin{pmatrix} c_1^{(t)} \\ \vdots \\ c_n^{(t)} \end{pmatrix}, \mathbf{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \mathsf{I} = \begin{pmatrix} 1 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & 1 \end{pmatrix}, \mathsf{B}_\alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix},$$

$$\mathsf{D}_\alpha = \begin{pmatrix} \alpha_1 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & \alpha_n \end{pmatrix}, \mathsf{D}_\beta = \begin{pmatrix} \beta_1 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & \beta_n \end{pmatrix},$$

$$\mathsf{D}_\gamma = \begin{pmatrix} 0 & \gamma_{12}I(2,1) & \cdots & \gamma_{1n}I(n,1) \\ \gamma_{21}I(1,2) & 0 & \cdots & \gamma_{2n}I(n,2) \\ \cdots & \cdots & \cdots & \cdots \\ \gamma_{n1}I(1,n) & \gamma_{n2}I(2,n) & \cdots & 0 \end{pmatrix},$$

then, we have

$$C(t) = \left(\sum_{j=0}^{t-1}\mathsf{S}^j\right)\mathsf{B}_\alpha + \mathsf{S}^t C(0).$$

where

$$\mathsf{S} = \mathsf{I} - \mathsf{D}_\alpha - \mathsf{D}_\beta + \mathsf{D}_\gamma. \tag{5.3}$$

Denote by $\lambda_{1,\mathsf{S}} \geq \lambda_{2,\mathsf{S}} \geq \cdots \geq \lambda_{n,\mathsf{S}}$ the ordered eigenvalues of S, and by $\mathsf{u}_{1,\mathsf{S}}, \ldots, \mathsf{u}_{n,\mathsf{S}}$ the orthogonal unit eigenvectors corresponding to the eigenvalues $\lambda_{1,\mathsf{S}}, \ldots, \lambda_{n,\mathsf{S}}$. By the spectral decomposition, it holds that

$$\mathsf{S} = \sum_{i=1}^{n}\lambda_{i,\mathsf{S}}\left(\mathsf{u}_{i,\mathsf{S}}\times\mathsf{u}_{i,\mathsf{S}}^T\right)$$

and hence

$$\mathsf{S}^t = \sum_{i=1}^{n} \lambda_{i,\mathsf{S}}^t \left(\mathsf{u}_{i,\mathsf{S}} \times \mathsf{u}_{i,\mathsf{S}}^T\right).$$

It can be shown that

$$\begin{aligned} C(t) &= \mathsf{B}_\alpha + \left[\sum_{i=1}^{n} \frac{\lambda_{i,\mathsf{S}}(1-\lambda_{i,\mathsf{S}}^{t-1})}{1-\lambda_{i,\mathsf{S}}} \left(\mathsf{u}_{i,\mathsf{A}} \times \mathsf{u}_{i,\mathsf{A}}^T\right)\right] \times \mathsf{B}_\alpha \\ &\quad + \left(\sum_{i=1}^{n} \lambda_{i,\mathsf{S}}^t \left(\mathsf{u}_{i,\mathsf{A}} \times \mathsf{u}_{i,\mathsf{A}}^T\right)\right) \times C(0). \end{aligned}$$

In order for the probability vector $C(t)$ to converge (i.e., the system eventually enters the steady state), a sufficient condition is that $\lambda_{i,\mathsf{S}}^t \to 0$ as $t \to \infty$ for all i. This is equivalent to requiring

$$\max_{1 \le i \le n} |\lambda_{i,\mathsf{S}}| < 1, \tag{5.4}$$

that is,

$$\lambda_{1,\mathsf{S}} < 1 \text{ and } \lambda_{n,\mathsf{S}} > -1. \tag{5.5}$$

Under the condition (5.4), it holds that, as $t \to \infty$,

$$C(t) \to \mathsf{B}_\alpha + \left[\sum_{i=1}^{n} \frac{\lambda_{i,\mathsf{S}}}{1-\lambda_{i,\mathsf{S}}} \left(\mathsf{u}_{i,\mathsf{S}} \times \mathsf{u}_{i,\mathsf{S}}^T\right)\right] \times \mathsf{B}_\alpha,$$

and thus

$$\mathsf{E}[C(t)] \to \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} \frac{\lambda_{i,\mathsf{S}}}{1-\lambda_{i,\mathsf{S}}} \left(\mathbf{1}^T \times \mathsf{u}_{i,\mathsf{S}}\right) \left(\mathsf{u}_{i,\mathsf{S}}^T \times \mathsf{B}_\alpha\right). \tag{5.6}$$

6 Case Study: on the (In)Security of PGP

Now we conduct a case study by applying our models to investigate the security of the PGP public key infrastructure [14], in which each user has a pair of public and private keys. A private key may be compromised because its owner's computer is broken into, and signatures can be arbitrarily forged until the public key has been revoked. In order to validate the accuracy of our models, we conduct simulations by utilizing the PGP web of trust graph (dated March 05 2006) of `http://dtype.org/keyanalyze/`. We use all keys in the data set, comprising 188,398 keys (i.e., nodes). We naturally interpret the trust relationships between the nodes as a directed graph. Specifically, if Alice certifies Bob's public key, then there is an arc from Alice to Bob. We note that self-loops (signing one's key by itself) had already been removed in the data set. This forms 565,542 directed edges. The simulations are done for 50 runs, and starting with the realistic setting of $\forall v \in V$, $s_v^{(0)} = 1$ and $c_v^{(0)} = 0$.

6.1 On the Accuracy of the Models

On the Accuracy of Model I. In Figure 4.(I.1)-4.(I.3) we compare the model predictions of $\mathsf{E}[S_t]$ and $\mathsf{E}[C_t]$, as indicated by Eq. (3.1), with the S_t and C_t averaged over the 50 simulation runs, respectively. We considered three sets of (arbitrary) parameters: (1) $\alpha = 0.001$ and $\beta = 0.1$; (2) $\alpha = 0.01$ and $\beta = 0.1$; (3) $\alpha = 0.1$ and $\beta = 0.1$.

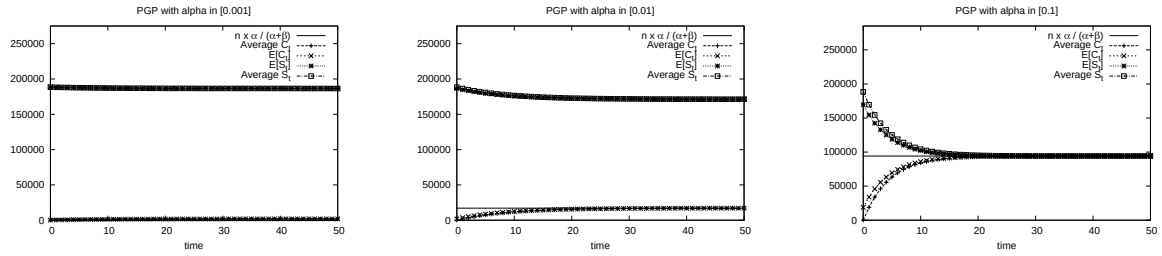
From Figure 4.(I.1)-4.(I.3), it is clear that S_t and C_t are symmetric or mirroring with respect to the horizontal line of $|V|/2$. This is because $S_t + C_t = n$. Note that in the case of $\alpha = 0.1$ and $\beta = 0.1$ (see Figure 4.(I.3)), this line coincides with $|V| \cdot \frac{\alpha}{\alpha+\beta}$. Moreover, $\mathsf{E}[S_t]$ and the averaged S_t (the two curves on the upper half of the picture) match almost perfectly in the case of $\alpha = 0.001$ and $\alpha = 0.01$. This also means that $\mathsf{E}[C_t]$ and the averaged C_t (the two curves on the lower half of the picture) match almost perfectly in the case of $\alpha = 0.001$ and $\alpha = 0.01$. Even in the case of $\alpha = 0.1$, $\mathsf{E}[S_t]$ and averaged S_t (therefore, $\mathsf{E}[C_t]$ and averaged C_t) match almost perfectly at about $t = 15$, even before the system enters the steady state. Finally, $\mathsf{E}[C_t]$ and the averaged C_t match $|V| \cdot \frac{\alpha}{\alpha+\beta}$ almost perfectly in the case of Figure 4.(I.1), and in the cases of Figure 4.(I.2) and Figure 4.(I.3) after the system enters the steady state (about $t = 15$).

On the Accuracy of Model II. In Figure 4.(II.1)-4.(II.3) we compare the model predictions of $\mathsf{E}[S_t]$ and $\mathsf{E}[C_t]$, as indicated by Eq. (4.2), with the S_t and C_t averaged over the 50 simulation runs, respectively. We consider three sets of (arbitrary) parameters: (1) $\alpha = 0.001$, $\beta = 0.1$, and $\gamma = 0.05$; (2) $\alpha = 0.01$, $\beta = 0.1$, and $\gamma = 0.05$; (3) $\alpha = 0.1$, $\beta = 0.1$, and $\gamma = 0.05$.

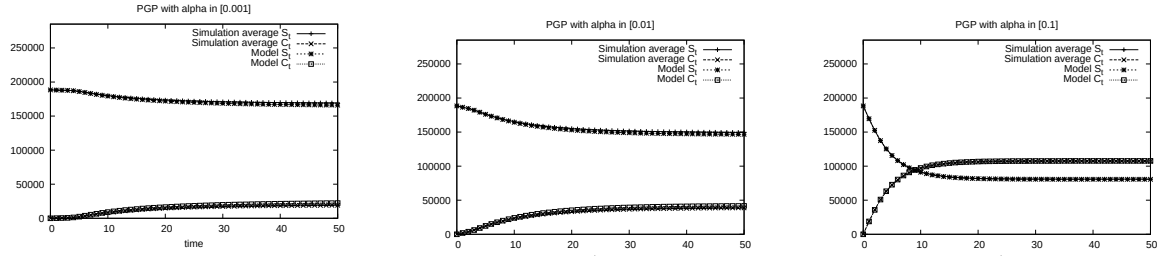
Figure 4.(II.1)-4.(II.3) clearly shows that S_t and C_t are symmetric with respect to the horizontal line of $|V|/2$. This is because $S_t + C_t = n$. Further, in all three cases, the model predictions of $\mathsf{E}[S_t]$ and $\mathsf{E}[C_t]$ almost perfectly match the averaged S_t and C_t, respectively. It is worthwhile to note that S_t and C_t are symmetric with respect to the horizontal line of $|V|/2$. This is because $S_t + C_t = n$.

On the Accuracy of Model III. In Figure 4.(III.1)-4.(III.3) we compare the model predictions of $\mathsf{E}[S_t]$ and $\mathsf{E}[C_t]$, as indicated by Eq. (5.2), with the S_t and C_t averaged over the 50 simulation runs, respectively. We consider three sets of (arbitrary) parameters: (1) $\alpha_v \in [0.001, 0.01]$ (i.e., each α_v is drawn from the interval $[0.001, 0.01]$ uniformly at random), $\beta_v \in [0.10, 0.20]$, and $\gamma_v \in [0.05, 0.10]$; (2) $\alpha_v \in [0.01, 0.1]$, $\beta_v \in [0.10, 0.20]$, and $\gamma_v \in [0.05, 0.10]$; (3) $\alpha_v \in [0.1, 0.2]$, $\beta_v \in [0.10, 0.20]$, and $\gamma_v \in [0.05, 0.10]$.

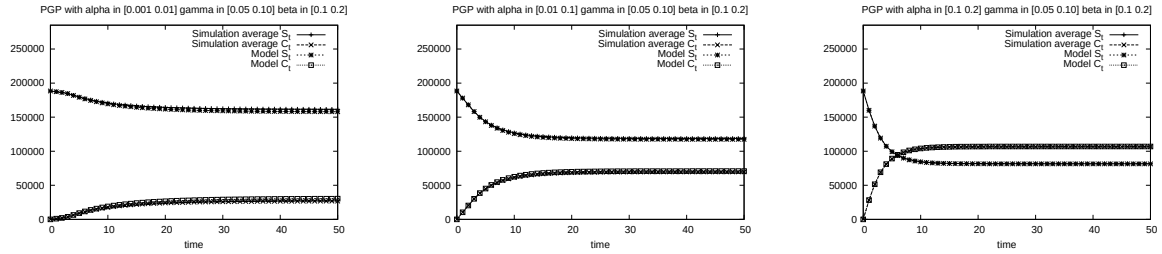
Figure 4.(III.1)-4.(III.3) clearly shows that S_t and C_t are symmetric with respect to the horizontal line of $|V|/2$. This is because $S_t + C_t = n$. Further, in all three cases, the model predictions of $\mathsf{E}[S_t]$ and $\mathsf{E}[C_t]$ almost perfectly match the averaged S_t and C_t, respectively. It is worthwhile to note



(I.1) $\alpha = 0.001, \beta = 0.1$ (I.2) $\alpha = 0.01, \beta = 0.1$ (I.3) $\alpha = 0.1, \beta = 0.1$



(II.1) $\alpha = 0.001, \beta = 0.1, \gamma = 0.05$ (II.2) $\alpha = 0.01, \beta = 0.1, \gamma = 0.05$ (II.3) $\alpha = 0.1, \beta = 0.1, \gamma = 0.05$



(III.1) $\alpha_v \in [0.001, 0.01], \beta_v \in [0.10, 0.20], \gamma_v \in [0.05, 0.10]$ (III.2) $\alpha_v \in [0.01, 0.1], \beta_v \in [0.10, 0.20], \gamma_v \in [0.05, 0.10]$ (III.3) $\alpha_v \in [0.1, 0.2], \beta_v \in [0.10, 0.20], \gamma_v \in [0.05, 0.10]$

Figure 4. Simulation vs. models using the PGP data

that S_t and C_t are symmetric with respect to the horizontal line of $|V|/2$. This is because $S_t + C_t = n$.

6.2 On the Usefulness of the Models

First, we have shown that our models can be used to quantify the security assurance when we consider the PGP graph as a whole. For example, Figure 4.(III.1) shows that under the given parameters there are always about 15% compromised private keys when the system enters the steady state. Such metrics can be used to help a signature verifier determine the trustworthiness of an incoming message that is accompanied with a valid digital signature. Obviously, in the case that a verifier needs a certain high confidence on a certain message, it can require, for instance, multiple signatures on the message. This can be done as follows. Suppose there is always a portion of private keys being compromised (e.g., $a = 0.15$ in the case of Figure 4.(III.1)), and a verifier needs $1 - \varepsilon$ confidence on the validity of a message. Then the verifier would achieve this by uniformly picking $\lceil \log_a \varepsilon \rceil$ nodes to certify the message, provided that the selected nodes are willing to do so.

Second, the analyses in the last sections suggested a sufficient condition for $\mathsf{E}[C_t]$ to converge, and useful advice towards lowering $\mathsf{E}[C_t]$. For example, if the users deploy host-based intrusion detection systems (i.e., which may increase β) and the users are cautious enough in executing codes (i.e., which may reduce α), the systems get more secure with a security gain that could even be quantified.

Third, our models can be used to help resolve the following important problem. Suppose B is a given budget. What is the optimal strategy for investing xB, where $0 \leq x \leq 1$, to reduce α (e.g., by installing an anti-virus software), and investing $(1-x)B$ to increase β (e.g., by deploying a global attack alert service mechanism)? In order to clarify the ideas, let's consider model I (a variant of model III is available in the full version of this paper [8]). In this model it holds that $\mathsf{E}[C_t] \rightarrow \frac{\alpha}{\alpha+\beta}$ when the system enters the steady state. We consider the following two cases.

Case I. Assume the outcome of an investment xB in reduc-

ing α is $\alpha(x) = \frac{\alpha}{1+k_1 x}$, and the outcome of an investment xB in increasing β is $\beta(x) = (1+k_2 x)\beta$, where k_1 and k_2 are some positive constants such that

$$\min_x \alpha(x) = \frac{\alpha}{1+k_1} \text{ and } (1+k_2)\beta < 1.$$

Note that $\alpha(0) = \alpha$ and $\beta(0) = \beta$. Given an investment xB, the steady state number of compromised nodes is

$$f(x) = \frac{n \cdot \alpha(x)}{\alpha(x) + \beta(1-x)} = \frac{n\alpha}{\alpha + (1+k_1 x)[1+k_2(1-x)]\beta}.$$

Our goal is to find some point x_{min} that minimizes $f(x)$, or equivalently, maximizes $h(x) = (1+k_1 x)[1+k_2(1-x)]$. Since

$$h'(x) \quad = \quad k_1 - k_2 + k_1 k_2 - 2k_1 k_2 x \begin{cases} \geq 0, & x \leq x_0 \\ < 0, & x > x_0, \end{cases}$$

where $x_0 = \frac{k_1 - k_2 + k_1 k_2}{2k_1 k_2}$, there are three scenarios.

1. If $x_0 \leq 0$, then $h'(x) < 0$ for $x \in [0,1]$. This means $x_{min} = 0$, namely that the entire budget should be invested to increase β.

2. If $0 < x_0 \leq 1$, then $h'(x) > 0$ for $x \in (0, x_0]$ and $h'(x) < 0$ for $x \in (x_0, 1]$. This means $x_{min} = x_0$, namely that $x_0 B$ should be invested to decrease α and $(1-x_0)B$ should be invested to increase β.

3. If $x_0 > 1$, then $h'(x) > 0$ for $x \in [0,1]$. This means $x_{min} = 1$, namely that the entire budget should be invested to decrease α.

Case II. Assume the outcome of an investment xB in reducing α is $\alpha(x) = \frac{\alpha}{1+k_1 x^2}$, and the outcome of an investment xB in increasing β is $\beta(x) = (1+k_2 x)\beta$, where k_1 and k_2 are some positive constants such that

$$\min_x \alpha(x) = \frac{\alpha}{1+k_1} \text{ and } (1+k_2)\beta < 1.$$

Note that $\alpha(0) = \alpha$ and $\beta(0) = \beta$.

Given an investment x, the steady state number of compromised nodes is

$$f(x) = \frac{n \cdot \alpha(x)}{\alpha(x) + \beta(1-x)} = \frac{n\alpha}{\alpha + (1+k_1 x^2)[1+k_2(1-x)]\beta}.$$

Our goal is to find some point x_{min} that minimizes $f(x)$, or equivalently maximizes $h(x) = (1+k_1 x^2)[1+k_2(1-x)]$. The practical value of x_{min} can be understood as follows: Given a budget B, it is optimal to invest $x_{min}B$ to reduce α, and $(1-x_{min})B$ to increase β. Since $h'(x) = -3k_2 k_1 x^2 + 2k_1(k_2+1)x - k_2$, $h'(0) = -k_2 < 0$. Denote by $\Delta = 4k_1^2(k_2+1)^2 - 12k_1 k_2^2$.

Suppose $\Delta > 0$. Then, $h'(x) = 0$ has two roots x_1, x_2, where $0 < x_1 < x_2$. There are three scenarios.

1. If $x_1 > 1$, then we have $h'(x) < 0$ for any $x \in [0,1]$. Thus, $h(x)$ decreases in $[0,1]$ and $x_{min} = 0$ minimizes $p(x)$ (i.e., maximizing $h(x)$) in $[0,1]$.

2. If $x_2 \leq 1$, then we have $h'(x) \leq 0$ when $x \in [0, x_1]$, $h'(x) > 0$ when $x \in [x_1, x_2]$, and $h'(x) \leq 0$ when $x \in [x_2, 1]$. Thus, $h(x)$ decreases in $[0, x_1]$ as well as $[x_2, 1]$, and increases in (x_1, x_2). Therefore, the point minimizing $f(x)$ is

$$x_{min} = \begin{cases} 0, & h(0) \geq h(x_2), \\ x_2, & h(0) < h(x_2). \end{cases}$$

3. If $x_1 \leq 1 < x_2$, then we have $h'(x) < 0$ when $x \in [0, x_1)$, and $h'(x) \geq 0$ when $x \in [x_1, 1]$. Thus, $h(x)$ decrease in $[0, x_1]$, but increases in $[x_1, 1]$. Therefore, the point minimizing $f(x)$ is

$$x_{min} = \begin{cases} 0, & h(0) \geq h(1), \\ 1, & h(0) < h(1). \end{cases}$$

Suppose $\Delta \leq 0$. Then $h'(x) \leq 0$ for any $x \in [0,1]$. Thus, $h(x)$ decreases in $[0,1]$, and $x_{min} = 0$ minimizes $f(x)$ (i.e., maximizing $h(x)$) in $[0,1]$.

7 Related Work

Existing graph-based analysis approaches vs. our modeling approach. There are three types of graph-based analysis approaches. The overall goal of these approaches are similar to ours, but there are also important differences as we elaborate below.

The first is based on *privilege graphs* [4, 9], where a node represents a set of privileges on some objects and an arc represents a vulnerability. An arc exists from one node to another if there is a method allowing a user owning the former node's privileges to obtain those of the latter.

The second is based on *attack graphs*, where a node represents the state of a network (i.e., the values assigned to relevant system attributes such as specific vulnerabilities on various hosts and connectivity between hosts), and an edge represents a step in an attack (cf. [10, 6, 1] and their references). A designated node (or set of nodes) represents the initial state(s), and each transition represents a specific exploit that an attack can carry out. This technique can identify the set of attack paths that have a high probability of success for the attacker.

The third is based on *key challenge graphs* [3], where a node represents a host, and an arc represents a *key challenge* that is an abstraction to capture access control. A key challenge is, for instance, a password authentication prior to accessing to a resource. The starting point of an attack could be one or more vertices, and the target of an attack

could be one or more vertices. A successful attack is a sequence of zero or more vertices not in the initial set but eventually containing all the target nodes. The cost of an attack is measured as the sum of the effort required to compromise individual vertices by attempting to counter the key challenges on the edges.

The main difference between privilege/attack graphs and ours lies in the model *assumptions* and *scalability*. First, privilege/attack graphs are based on the systems' *known* vulnerabilities that have not been patched. While this is certainly a realistic threat, we believe that it would be better resolved using vulnerability-specific countermeasures (e.g., [11]). In contrast, our modeling approach does not assume known vulnerabilities; instead, it is quite appropriate for modeling attacks that may be based on *zero-day or unknown* vulnerabilities — this further justifies our use of stochastic models. Second, privilege/attack graphs based approaches suffer from limited scalability, because of their inherent exponential state explosion; whereas our approach is scalable.

The main difference between the key challenge graph approach and ours lies in the model *purpose* (i.e., the questions targeted by the models) and *capability*. First, the key challenge graph approach emphasizes the algorithmic aspect of finding an optimal attack path, namely that the adversary can achieve its goal with minimal effort or cost. In contrast, our modeling approach aims to understand the dynamic behavior of system evolution, and to answer questions such as the number of compromised private keys in a public key infrastructure. Second, the key challenge graph can only capture the attack behaviors with respect to some specific starting points. In contrast, our modeling approach does not need to know the initially compromised nodes, nor even to have any.

Existing epidemic models vs. our model. Existing epidemic models (e.g., [7] and its numerous follow-ons) are typically adapted from the biological epidemiology models for *homogeneous* systems [2]. Such models assume that every individual has equal contact to everyone else in the population, and that the rate of infection is largely determined by the density of the infected individuals. Homogeneous models may be useful in some cases, e.g., when a worm spreads via *truly random scanning*. However, homogeneous models do not apply to heterogeneous systems, which are harder to model because topology-based spreading does not rely on random scanning.

Semi-heterogeneous networks are investigated in [13, 12] with discrete time models, and in [5] with an analogous continuous time model. The models of [13, 12] are perhaps the closest to ours. However, they have the following drawbacks. (1) They cannot express that the number of compromised nodes is persistently greater than zero, even if the ratio between the attack death rate and the attack birth rate is above the threshold. (2) They can only model semi-heterogeneous systems, where different nodes may have different degrees. However, both capabilities are offered in our models.

8 Conclusion

We presented a series of models for quantifying the (in)security of networked systems. Our models are powerful, accurate and useful. As a specific application, we showed how our models can be applied to study the (in)security of PGP based on some artificial parameters.

Acknowledgement. This work was supported in part by ARO and UTSA.

References

[1] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *ACM CCS'02*, pp 217–224.

[2] N. Bailey. *The Mathematical Theory of Infectious Diseases and Its Applications*. 2nd Edition. Griffin, London, 1975.

[3] R. Chinchani, A. Iyer, H. Ngo, and S. Upadhyaya. Towards a theory of insider threat assessment. In *Proc. IEEE DSN'05*, pp 108–117.

[4] M. Dacier and Y. Deswarte. Privilege graph: an extension to the typed access matrix model. In *Proc. of ESORICS'94*, pp 319–334.

[5] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proc. IEEE Infocom 2005*, 2005.

[6] S. Jha and J. Wing. Survivability analysis of networked systems. In *Proc. ICSE'01*, pp 307–317.

[7] J. Kephart and S. White. Directed-graph epidemiological models of computer viruses. In *IEEE Symposium on Security and Privacy*, pages 343–361, 1991.

[8] X. Li, P. Paul and S. Xu. Towards quantifying the (in)security of networked systems. Full version of this paper, available at `www.cs.utsa.edu/~shxu`.

[9] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Trans. Softw. Eng.*, 25(5):633–650, 1999.

[10] C. Phillips and L. Swiler. A graph-based system for network-vulnerability analysis. In *Proc. NSPW'98*, pp 71–79.

[11] H. Wang, C. Guo, D. Simon, and A. Zugenmaier. Shield: vulnerability-driven network filters for preventing known vulnerability exploits. In *Proc. ACM SIGCOMM'04*, pp 193–204.

[12] J. Wang, L. Lu, and A. Chien. Tolerating denial-of-service attacks using overlay networks – impact of topology. In *Proc. ACM workshop on survivable and self-regnerative systems*, 2003.

[13] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *Proc. IEEE SRDS'03*, pp 25–34.

[14] P. Zimmermann. *The official PGP user's guide*. MIT Press, Cambridge, MA, USA, 1995.