Proc. of 2nd Intern. Workshop "New Models of Business: Managerial Aspects and Enabling Technology", St. Petersburg State University, St. Petersburg, Russia, June 26-28, 2002, 212-220

Algebraic Modeling and Performance Evaluation of Business Processes

D. Guster[1]
St. Cloud State University, USA
N.K. Krivulin[2,3]
St. Petersburg State University, Russia

Abstract

An algebraic approach to the modeling and performance evaluation of business processes is developed based on fork-join queueing network formalism and idempotent algebra. As an illustration, a model of computer system security operation is considered. We introduce a related performance measure, and show how it may be used to analysis of actual systems.

Keywords: business process, computer system security, performance evaluation, fork-join queueing networks, idempotent algebra

1. Introduction

Most of the innovative activities under Continuous Improvement Efforts, Business Process Reengineering (BPR), and other programs companies try to implement to achieve better results in their operation are based on extensive use of information technology and systems. Among other analytical functions, the information systems normally provide for modeling of business processes on the basis of both mathematical methods and computer simulation. Although pure mathematical approaches can be inferior to computer simulation in versatility and flexibility, they allow one to get results easier and faster provided that there is an appropriate mathematical model and related solution methods. Of particular interest are the models that enable one to get closed-form solutions when evaluating business process performance measures and other quantitative characteristics. Clearly, the last models together with their solutions could be efficiently incorporated into any information system.

---

[1] St. Cloud State University, 720 4th Ave. S., St. Cloud, MN 56301-4442, USA, e-mail: Guster@stcloudstate.edu

[2] St. Petersburg State University, Universitetsky Ave. 28, Petrodvorets, St. Petersburg, 198504 Russia, e-mail: Nikolai.Krivulin@pobox.spbu.ru.

[3] The work is partially supported by the Russian Foundation for Basic Research – National Natural Science Foundation of China, Grant #99-01-39137.

One of the performance measures commonly used in analysis of business processes is the mean cycle time of a process. Such BPR activities as Reducing Cycle Time and Reducing Time to Market are directly involve the problem of evaluating the cycle time. In general, the above problem appears every time one is interested in evaluating performance of a recurring (cyclic) process of similar actions. As an illustration, one can consider service of a sequence of customers, repetitive corporate management routines, logistics operations, and others.

In this paper, we concentrate on evaluating cycle time for the process of computer security operation. In fact, the explosive growth in computer systems and networks has increased the role of computer security within organizations (Stallings, 1995). In many cases, ineffective protection against computer security treats leads to considerable damage, and even can cause an organization to be paralyzed. Therefore, the development of new models and methods of performance analysis of security systems seems to be very important.

We propose a model of computer security operation, and introduce its related performance measure that can be evaluated in a closed form. It is shown how the model can be applied to performance evaluation of actual systems. Finally, a technique of security system performance analysis is described and its practical implementation is discussed. In fact, the proposed models and methods are quite general, and they can be applied to analysis of many other business processes and systems.

We conclude with an appendix, which contains technical details concerning fork-join network representation of the model, idempotent algebra, and related results.

2. An Example: Security Operation Model

As an example of a business process we consider a computer security system operation in an organization. In fact, we deal with the current security activities (see Fig. 1) that mainly relate to the actual security threats rather than to strategic or long-term issues of security management (Guster and Krivulin, 2001).

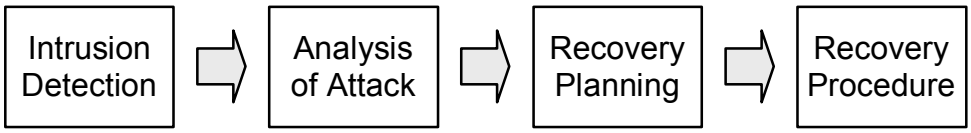

Figure 1. Computer systems security activities.

Consider the model of security operation in an organization, presented in Fig. 2. Each operational cycle starts with security attack detection based on audit records and system/errors log analysis, traffic analysis, or user reports. In order to detect an intrusion, automated tools of security monitoring are normally used including procedures of statistical anomaly detection, rule-based detection, and data integrity control (Stallings, 1995).

After security attack detection and identification, the integrity of system/application software and data in storage devices has to be examined to search for possible unauthorized

modifications or damages made by the intruder. The investigation procedure can exploit file lists and checksum analysis, hash functions, and other automated techniques.

In parallel, the system vulnerabilities, which allow the intruder to attack, should be identified and investigated. The vulnerability analysis normally presents an informal procedure, and therefore, it can hardly be performed automatically.

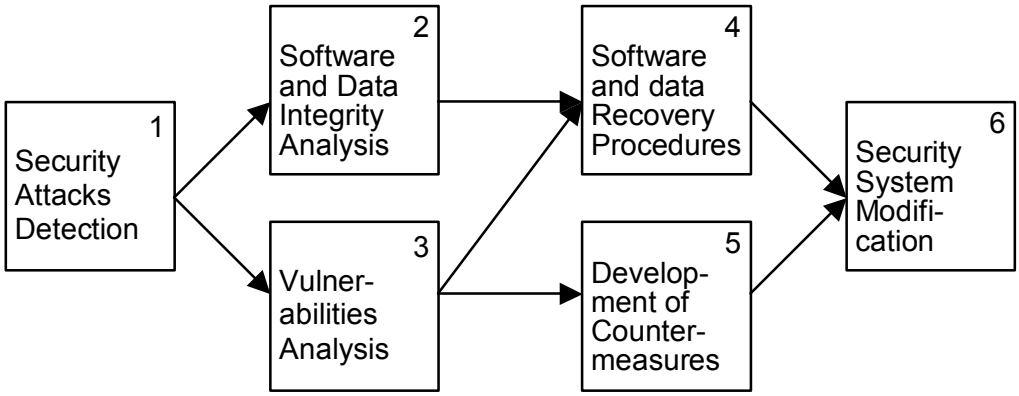

Figure 2. A Security analysis and modification model.

Based on the results of integrity analysis, a software and data recovery procedure can be initiated using back-up servers and reserving storage devices. It has to take into account the security vulnerabilities identified at the previous step, so as to provide for further improvements in the entire security system.

Along with the recovery procedure, the development of a complete set of countermeasures against similar attacks should be performed. Finally, the operational cycle is concluded with appropriate modifications of software, databases, system security policies and procedures.

We assume that the organization has appropriate personnel integrated in a Computer Emergency Response Team, available to handle the attack. The team would include at least two subteams working in parallel, one to perform integrity analysis and recovery procedures, and another to do vulnerability analysis and development of countermeasures. At any time instant, each subteam can deal with only one security incident. Any procedure may be started as soon as all prior procedures according to the model in Fig. 2, have been completed. If a request to handle a new incident occurs when a subteam is still working on a procedure, the request has to wait until the processing of that procedure is completed.

We denote by $\tau_1(k)$ a random variable (r.v.) that represents the time interval between detections of the k th attack and its predecessor. Furthermore, we introduce r.v.'s $\tau_i(k)$, $i = 2,\ldots,6$, to describe the time of the k th instant of procedure i in the model. We assume $\tau_i(1), \tau_i(2), \ldots$, to be independent and identically distributed (i.i.d.) r.v.'s with finite mean

and variance for each i, and denote $\tau_i = \tau_i(1)$, $i = 1,\ldots,6$. At the same time, we do not require of independence of $\tau_1(k),\ldots,\tau_6(k)$ for each k, $k = 1,2,\ldots$

3. Security Operation Performance Evaluation

In order to describe system performance, we introduce the following notations. Let $\overline{T}_A$ be the mean time between consecutive security attacks (the attack cycle time), and $\overline{T}_S$ be the mean time required to completely handle an attack (the recovery cycle time), as the number of attacks k tends to ∞.

In devising the security operation performance measure, one can take the ratio

$$R = \overline{T}_S / \overline{T}_A .$$

With the natural condition $\overline{T}_S \le \overline{T}_A$, one can consider R as the time portion the system is under recovery, assuming $k \to \infty$.

First note that the attack cycle time can immediately be evaluated as the mean: $\overline{T}_A = \mathrm{E}[\tau_1]$. Consider the cycle time of the entire system, which can be defined as the mean time between successive completions of security system modification procedures as $k \to \infty$. As one can show (see Appendix for further details), the system cycle time γ is given by

$$\gamma = \max\{\mathrm{E}[\tau_1],\ldots,\mathrm{E}[\tau_6]\} .$$

In order to evaluate the recovery cycle time, we assume the system will operate under the maximum traffic level, which can be achieved when all the time intervals between attacks are set to zero. Clearly, under that condition, the system cycle time can be taken as a reasonable estimate of the recovery cycle time.

Considering now that $\mathrm{E}[\tau_1] = 0$, we get the recovery cycle time in the form

$$\overline{T}_S = \max\{\mathrm{E}[\tau_2],\ldots,\mathrm{E}[\tau_6]\} .$$

4. Performance Analysis and Discussion

In fact, the above model presents a quite simple but useful tool for security system operation management. It may be used to make decision on the basis of a few natural parameters of the security operation process.

Let us represent the ratio R in the form

$$R = \max\{\mathrm{E}[\tau_2],\ldots,\mathrm{E}[\tau_6]\} / \mathrm{E}[\tau_1],$$

and assume the attack rate determined by $\mathrm{E}[\tau_1]$, to be fixed.

Taking into account that the above result has been obtained based on the assumption of an infinite number of attacks, we arrive at the following conclusion. As the number of attacks

becomes sufficiently large, the performance of the system is determined by the time of the longest procedure involved in the system operation, whereas the impact of the order of performing the procedures disappears.

It is clear that in order to improve system performance, the system security manager should first concentrate on decreasing the mean time required to perform the longest procedure within the security operation model, then consider the second longest procedure, and so on. The goal of decreasing the time can be achieved through partition of a whole procedure into subprocedures, which can be performed in parallel, or through rescheduling of the entire process with redistribution of particular activities between procedures.

In practice, the above model and its related ratio R can serve as the basis for efficient monitorization of organizational security systems. Because the introduction of new countermeasures may change the attack cycle time, the monitoring requires updating this parameter after each modification of the system.

Finally note, the above model can be easily extended to cover security operational processes, which consist of different procedures and precedence constraints. It is also quite applicable to analysis of many other business processes and systems.

Appendix

In order to describe the above systems in a formal way, we exploit the fork-join network formalism proposed in (Baccelli, 1989). The fork-join networks actually present a class of queueing systems, which allow for splitting a customer into several new customers at one node, and merging customers into one at another node.

To represent the dynamics of the networks, we use the (max,+)-algebra based approach developed in (Krivulin, 1995, 1996, 1998, 2000).

Idempotent Algebra

The (max,+)-algebra is the triple $\langle R_\varepsilon, \oplus, \otimes \rangle$, where $R_\varepsilon = R \cup \{\varepsilon\}$ is the set of real numbers with $\varepsilon = -\infty$ added, and $\oplus$ and $\otimes$ are binary operations defined as

$a \oplus b = \max(a, b)$, $a \otimes b = a + b$, for all $a, b \in R_\varepsilon$.

There are the null and identity elements in the algebra, namely ε and 0, to satisfy the conditions $a \oplus \varepsilon = \varepsilon \oplus a = a$, and $a \otimes 0 = 0 \otimes a = a$, for any $a \in R_\varepsilon$. The absorption rule involving $a \otimes \varepsilon = \varepsilon \otimes a = a$ for any $a \in R_\varepsilon$ is also valid.

The operations $\oplus$ and $\otimes$ retain most of the properties of the ordinary addition and multiplication, including associativity, commutativity, and distributivity of $\otimes$ over $\oplus$. However, the operation $\oplus$ is idempotent; that is, for any $a \in R_\varepsilon$, one has $a \oplus a = a$.

The algebra of matrices is introduced in the regular way. Specifically, for any $(n \times n)$-matrices $A = (a_{ij})$ and $B = (b_{ij})$, the entries of $C = A \oplus B$ and $D = A \otimes B$ are given by

$$c_{ij} = a_{ij} \oplus b_{ij} \text{ and } d_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes b_{kj} .$$

As the null and identity elements, the matrices

$$\mathcal{E} = \begin{pmatrix} \varepsilon & \cdots & \varepsilon \\ \vdots & \ddots & \vdots \\ \varepsilon & \cdots & \varepsilon \end{pmatrix}, \quad E = \begin{pmatrix} 0 & & \varepsilon \\ & \ddots & \\ \varepsilon & & 0 \end{pmatrix}$$

are respectively taken in the algebra.

Let $A = (a_{ij})$ be any $(n \times n)$-matrix. In the same way as in the conventional algebra, one can define $A^0 = E$ if $A \neq \mathcal{E}$, and $A^m = A \otimes A^{m-1} = A^{m-1} \otimes A$ for any integer $m \geq 1$.

Finally, one can define the matrix functions

$$\mathrm{tr}(A) = \bigoplus_{i=1}^{n} a_{ii} , \quad \|A\| = \bigoplus_{i=1}^{n} \bigoplus_{j=1}^{n} a_{ij} .$$

Algebraic Representation and Related Results

We consider a network with n single-server nodes and customers of a single class. The topology of the network is described by an oriented acyclic graph with its nodes representing servers, and its arcs determining the transition routes of customers. The nodes that have no predecessors are assumed to represent an infinite external arrival stream of customers. Each node without successors is considered as an output node, which releases customers from the network. An example of a network with $n = 6$ nodes is given in Fig. 3.

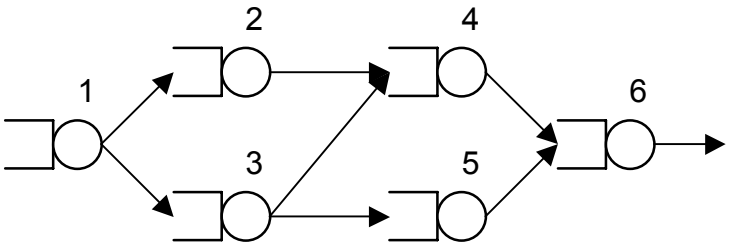

Figure 3. A network scheme.

Each node of the network includes a server and a buffer with infinite capacity, operating as a single-server queue under the first-come, first-served discipline. At the initial time, the servers and their buffers are assumed to be free of customers, except for the buffers in nodes with no predecessors, each assumed to have an infinite number of customers.

The operation of each node can include join and fork operations performed respectively before and after service. The join operation is actually thought to cause each customer coming into a node not to enter the queue but to wait until at least one customer from all preceding nodes arrives. Upon arrival, these customers are replaced by a new customer, which joins the queue. The fork operation at a node is initiated every time the service of a customer is completed. It consists in replacing the customer by several new customers, each intended to go to one of the succeeding nodes.

For the queue at node i, $i = 1,\ldots,n$, we denote the k th departure epoch by $x_i(k)$, and the service time of the k th customer by $\tau_i(k)$. Considering that the network starts operating at time zero, it is convenient to set $x_i(0) = 0$ and $x_i(k) = \varepsilon$ for all $k < 0$, $i = 1,\ldots,n$.

Now we introduce the notations:

$$\boldsymbol{x}(k) = \begin{pmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{pmatrix}, \quad T(k) = \begin{pmatrix} \tau_1(k) & & \varepsilon \\ & \ddots & \\ \varepsilon & & \tau_n(k) \end{pmatrix},$$

It has been shown in (Krivulin 1996) that the dynamics of the network can be described by the equation

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1)$$

with the matrix

$$A(k) = \bigoplus_{j=0}^{l} (T(k) \otimes G^T)^j \otimes T(k)$$

where l is the length of the longest path in the network graph, $G = (g_{ij})$ is the matrix with its entry $g_{ij} = 0$, if there exists arc (i, j) in the network graph, and $g_{ij} = \varepsilon$, otherwise.

The cycle time of the system is defined as

$$\gamma = \lim_{k \to \infty} \frac{1}{k} \| \boldsymbol{x}(k) \|$$

provided that the above limit exists.

Now suppose that $\tau_i(1), \tau_i(2), \ldots$, are i.i.d. r.v.'s with the finite mean and variance for each $i = 1,\ldots,n$. Under these assumptions, the next result presented in (Krivulin, 2002) is valid.

Theorem 1. *Let* $B = \mathrm{E}[A(1)]$ *be the matrix obtained from* $A(1)$ *by replacing all its entries with their mean values, considering that* $\mathrm{E}[\varepsilon] = \varepsilon$. *Then the cycle time is given by*

$$\gamma = \bigoplus_{i=1}^{n} \frac{1}{i} \mathrm{tr}(B^i) \quad \textit{with probability 1.}$$

The Security Operation Model

Clearly, the network depicted in Fig. 3 just represents the security operation model under consideration. The matrix G of the network graph takes the form

$$G = \begin{pmatrix} \varepsilon & 0 & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}.$$

Since for the network, we have $l = 3$, the matrix $A(k)$ is written as

$$A(k) = [E \oplus T(k) \otimes G^T \oplus (T(k) \otimes G^T)^2 \oplus (T(k) \otimes G^T)^3] \otimes T(k).$$

Consider the matrix $A(1)$, and define $\tau_i = \tau_i(1)$, $i = 1,\ldots,6$. Simple algebra gives us

$$A(1) = \begin{pmatrix} \tau_1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_1 \otimes \tau_2 & \tau_2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_1 \otimes \tau_3 & \varepsilon & \tau_3 & \varepsilon & \varepsilon & \varepsilon \\ \tau_1 \otimes (\tau_2 \oplus \tau_3) \otimes \tau_4 & \tau_2 \oplus \tau_4 & \tau_3 \oplus \tau_4 & \tau_4 & \varepsilon & \varepsilon \\ \tau_1 \otimes \tau_3 \otimes \tau_5 & \varepsilon & \varepsilon & \varepsilon & \tau_5 & \varepsilon \\ \tau_1 \otimes \tau_3 \otimes \tau_5 \otimes \tau_6 & \tau_2 \otimes \tau_5 \otimes \tau_6 & \tau_3 \otimes (\tau_4 \oplus \tau_5) \otimes \tau_6 & \tau_4 \otimes \tau_6 & \tau_5 \otimes \tau_6 & \tau_6 \end{pmatrix}.$$

It remains to proceed to the matrix $B = \mathrm{E}[A(1)]$, and then evaluate γ. First note that both $A(1)$ and B have the lower triangular form, and so any power of B takes the same form.

As it easy to verify, we have for any $i = 1,\ldots,6$,

$$A^i = \begin{pmatrix} i\mathrm{E}[\tau_1] & & \varepsilon \\ & \ddots & \\ \ldots & & i\mathrm{E}[\tau_6] \end{pmatrix},$$

where the entries below the diagonal are omitted for the sake of simplicity.

Furthermore, for each i, we get

$$\mathrm{tr}(A^i) = i(\mathrm{E}[\tau_1] \oplus \cdots \oplus \mathrm{E}[\tau_6]),$$

and thus

Proc. of 2nd Intern. Workshop “New Models of Business: Managerial Aspects and Enabling Technology”, St. Petersburg State University, St. Petersburg, Russia, June 26-28, 2002, 212-220

$$\gamma = \bigoplus_{i=1}^{n} \frac{1}{i} \operatorname{tr}(A^i) = \mathrm{E}[\tau_1] \oplus \cdots \oplus \mathrm{E}[\tau_6].$$

Turning back to the ordinary notations, we finally arrive at

$$\gamma = \max\{\mathrm{E}[\tau_1], \ldots, \mathrm{E}[\tau_6]\}.$$

References

Baccelli F. and Makowski A.M. (1989). Queueing Models for Systems with Synchronization Constraints, Proceedings of the IEEE, 77, No. 1, P. 138-160.

Guster D. and Krivulin N.K. (2001). Modeling and Performance Evaluation of Computer Systems Security Operation, Proceedings of the 4th St. Petersburg Workshop on Simulation (Simulation 2001), St. Petersburg, Russia, June 18-22, 2001 (S.M. Ermakov, Yu.N. Kashtanov and V.B. Melas, Eds.), St. Petersburg University, St. Petersburg, P. 233-238.

Krivulin N.K. (1995). A Max-Algebra Approach to Modeling and Simulation of Tandem Queueing Systems, Mathematical and Computer Modelling, 22, No.3, P. 25-37.

Krivulin N.K. (1996). Max-Plus Algebra Models of Queueing Networks, Proceedings of the International Workshop on Discrete Event Systems (WODES'96), 19-21 August 1996, University of Edinburgh, UK, IEE, London, P. 76-81.

Krivulin N.K. (1998). Monotonicity Properties and Simple Bounds on the Mean Cycle Time in Acyclic Fork-Join Queueing Networks, Recent Advances in Information Science and Technology (N. Mastorakis, Ed.), World Scientific, P. 147-152.

Krivulin N.K. (2000). Algebraic Modelling and Performance Evaluation of Acyclic Fork-Join Queueing Networks, Advances in Stochastic Simulation Methods, Statistics for Industry and Technology (N. Balakrishnan, V. Melas and S. Ermakov, Eds.), Birkhäuser, Boston, P. 63-81.

Krivulin N.K. (2002). Evaluation of the Mean Cycle Time in Fork-Join Networks, to appear in Vestnik Sankt-Peterburgskogo Universiteta. Ser. 1. (in Russian)

Stallings W. (1995). Network and Internetwork Security: Principles and Practice, Prentice-Hall, Englewood Cliffs.