

# Security modeling and quantification of intrusion tolerant systems \*

B.B. Madan and K.S. Trivedi †

CACC, Department of Electrical & Computer Engineering,  
Pratt School of Engineering, Duke University, Durham, NC

## 1 Introduction

In the past, the security of a software or an information system has been treated as a qualitative attribute. Recently however, several researchers [1, 2, 3] have suggested that security be treated as a quantifiable QoS attribute. The question that naturally arises is how to formally define security? Security analysis of a software or an information system is a 5-fold framework of (1) Availability, (2) Integrity, (3) Confidentiality, (4) Authenticity, and (5) Non-repudiation as depicted in Figure 1:

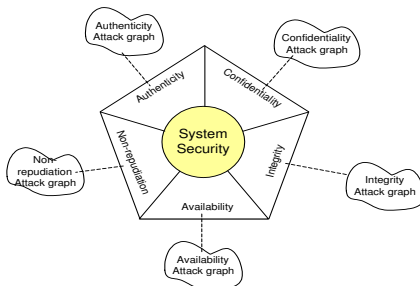


Figure 1. 5-fold Security Framework

## 2 Attack-Response Graph

Computer systems may be subjected to a wide variety of attacks leading to security being compromised. While these attacks numerous in numbers, all follow one strategy, namely to build an attack with the help of a series of smaller or atomic intrusions. Swiler et. al. [4] suggested the use of a graph (which they call the *attack graph*) to not only model a security attack but also suggested building such a graph

\*This work is sponsored by the U.S. Department of Defense Advanced Research Projects Agency (DARPA) under contract N66001-00-C-8057 from the Space and Naval Warfare Systems Center - San Diego (SPAWARSYSCEN). The views, opinions and findings contained in this paper are those of the authors and should not be construed as official DARPA or SPAWARSYSCEN's positions, policy or decision

† work was done as a Visiting Professor holding the Poonam and Prabhu Goel Chair, Dept of Comp. Sc. and Eng., I.I.T, Kanpur

automatically using a database of atomic attacks. Dacier et. al. [2] suggested a similar technique, which they refer to as the *privilege graph*. They also suggest techniques for converting a privilege graph to a Markov chain as well a stochastic Petri net. The resulting Markov chain model is also used for quantifying the security attribute of a system by computing the mean time (or effort) required to send the system into a security failed state. More recently, Jha et. al. [5] and [6] have suggested algorithms for automatic generation, minimization and verification of an attack. Attack graph (privilege graph) has been successfully used to model a system's susceptibility to security attacks or successful intrusions into a system. In this approach, a successful attack causes the system to reach a security failed state. Moreover, a system does not reach the security failed state in a single step. Instead, all attacks are made up of a number of atomic attacks. Atomic attacks defining the path from the normal start state to the final security failed state have to succeed in order for the security failure to result. From the system's perspective of ensuring intrusion tolerance, this observation can be made to play a crucial role. If we compare this situation with reliability theory, it is easy to see that we are dealing with a *series system*. In other words, if the system has some mechanism in place to thwart any one of the atomic attacks, the system will be able to defeat the attacker.

The attack or the privilege graph models described in [2, 5, 6] model only the attacker's actions. However, to model an intrusion tolerant system, we also need to introduce system's response to (some of) the atomic attacks. A graph that incorporates attacker's actions as well as the system's response will henceforth be referred to as the *Attack Response Graph (ARG)*. Note that it is not practical for a system to respond to each and every atomic attack. Instead, it would suffice to identify one arc (atomic attack) along an attack path (attack foot print) to which the system can respond. To facilitate identification of such an arc, we propose using a *Depth First Search (DFS)* on the attack graph. The DFS graph traversal will divulge all the possible paths (in terms of atomic attacks) that the attackers may follow to inject a successful attack into the system.

```
set T of attack foot prints ← DFS(G)
```

If the system is now able to break any edge of in an attack footprint by initiating intrusion tolerant measures, then the system is effectively able to tolerate such an intrusion. This is clarified by the following two examples:

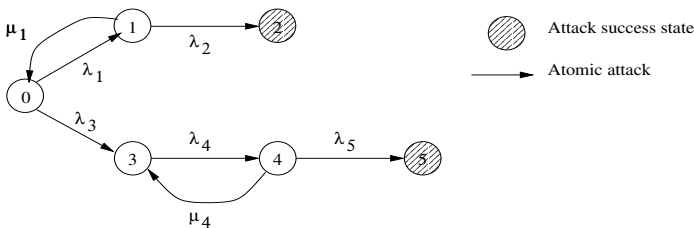
Syn flood DoS attack: The attacker fills up the listening socket queue with half-open TCP connections. This will prevent a valid TCP service user from making a TCP connection. However, if the system monitors this listening queue state and is able to conclude that it is being flooded with half open TCP connections, it can initiate intrusion tolerance measures, such as, either flushing this queue or closing this service for some small amount of time, thus preventing compromise of system’s availability measure.

Login ID theft: This attack is made up of several atomic attacks as enumerated below : (1)  $host_{attacker}$  creates a root level shell on  $host_1$  using *sshd* buffer vulnerability. (2) Attacker creates remote login trust between  $host_1$  and  $host_2$  by overwriting *.rhost* file using *ftpd* vulnerability. (3) Using *'rsh -l user1 host2'* (within the root shell created in step 1), attacker creates a shell on the target  $host_2$ , thus stealing user1’s identity.

Depending on the privileges of user1 on  $host_2$  the attacker may be able to cause compromise of integrity, confidentiality or availability. However, if the system is able to counter any of the atomic attacks, e.g., cleaning up step 2 above, then the system will be able to tolerate such attacks without any real damage to the system.

### 3 Preliminary Results

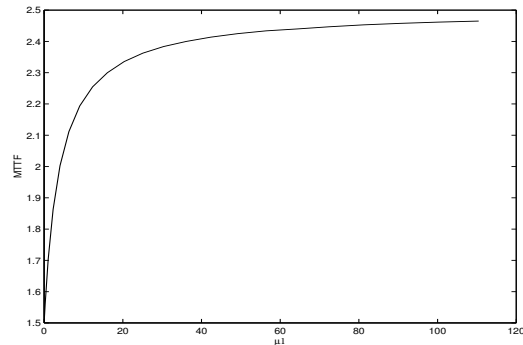
Consider a sample Markov chain shown in Figure 2 that can be derived from an attack-response graph using the procedure described in [2]. The return arcs labeled  $\mu_1$  and



**Figure 2. Attack-response Markov Chain**

$\mu_4$  represent the system’s response to the corresponding atomic attacks. Figure 3 shows the mean *MTTF* for the system to reach the security failed (or absorbing) states 2 and 5 as a function of  $\mu_1$ . SHARPE tool [7] was used to solve the Markov chain shown Figure 2 to compute the *MTTF* to reach the absorbing states. This *MTTF* is commonly referred to as the unconditional *MTTF*. The sys-

tem’s *MTTF* behavior as shown in Figure 3 is intuitively satisfying, in that, as  $\mu_1$  increases, i.e., systems reacts more frequently to attacks, the *MTTF* measure also increases.



**Figure 3. MTTF behavior ( $\lambda_1 = 1.0, \lambda_3 = 1.0, \lambda_2 = 2.0, \lambda_4 = 1.5, \lambda_5 = 2.0, \mu_4 = 1.0$ )**

### References

- [1] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, and D. Wright. Towards operational measures of computer security. *Journal of Computer Security*, 2:211–229, 1993.
- [2] M. Dacier, Y. Deswarte, and M. Ka nliche. Quantitative assessment of operational security: Models and tools. Technical report, LAAS research Rept. 96493, May, 1996.
- [3] B.B. Madan, K. Go seva-Popstojanova, K. Vaidyanathan, and K.S. Trivedi. Modeling and quantification of security attributes of software systems. In *Proc. Int. Conf. DSN, (IPDS stream)*, volume 2, pages 505–514, 2002.
- [4] L.P. Swiler, C. Phillips, and T. Gaylor. A graph-based network vulnerability analysis system. Technical report, SANDIA Report, SAND97-3010/1, Jan, 1998.
- [5] S. Jha, O. Sheyner, and J.M. Wing. Minimization and reliability analysis of attack graphs. Technical report, CMU Tech. Report, CMU-CS-2-109, May, 2002.
- [6] O. Sheyner, J. Haines, S. Jha, and R. Lippmann and J.M. Wing. Automated generation and analysis of attack graphs. Technical report, May, 2002.
- [7] 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 Publisher, 1995.