Attack modelling in open network environments

S.K. Katsikas
University of the Aegean
Department of Mathematics
Karlovassi 83200
Greece
Phone: +30-273-33919
Fax: +30-273-35482
E-mail: ska@aegean.gr

D. Gritzalis
Athens University of Economics & Business
Department of Informatics
76 Patission St., Athens 10434
Greece
Phone: +30-1-8212532
Fax: +30-1-8212532
E-mail: dgrit@aegean.gr

P. Spirakis
University of Patras
Department of Computer Engineering & Informatics
Patras 26500
Greece
Phone: +30-61-997703
Fax: +30-61-991909
E-mail: spirakis@cti.gr

Abstract

In this paper, the beginnings of a formal generic model describing the process of a malicious attack against a computer network, has been proposed, leading to a Markov chain description. This model can be used for better understanding the nature of malicious attacks

P. Horster (ed.), *Communications and Multimedia Security II*

against open networks, for defining a simulator to analyse the impact of malicious attacks against computer networks, or for analytically and quantitatively studying the power of several attacks versus the effectiveness of protection mechanisms. A worked out example of the description of a virus attack against a computer network is also given.

Keywords

Intrusion, Computer Virus, Worm, Trojan Horse, Markov Chain

1 INTRODUCTION

For the purposes of this paper, the term *malicious attack* is taken to mean the carrying out of unauthorised activities. Malicious attacks are classified into two broad categories, namely, viruses and intrusions.

A computer virus attack may be informally described as a sequence of symbols in a machine's memory; what makes such a sequence of symbols an element of a "viral set" (Cohen, 1989), is that when the machine interprets this sequence, it causes some other element of that viral set to appear somewhere else in the system, at a possibly later time. Formal descriptions of computer viruses have been provided for, either through a computational approach (Cohen, 1989, Cohen, 1987) or through an abstract formalism (Adleman, 1990, Kephart, 1991, Ostrowski, 1991).

Even though several issues pertaining to viruses (behaviour, characteristics, antidotes, spreading mechanisms, etc.) have been examined in detail in the literature, there is very limited published work on the issue of how a virus spreads into a network, and how a network is affected by this spreading.

Virus spread models in non-distributed environments have been reported in (Cohen, 1989). In (Guinier, 1991) it was shown that a virus that has infected a single file quickly spreads and incapacitates all files within a computer system, if the system has no ability to apply countermeasures against the virus. In (Kephart, 1991), a statistical-epidemiological model of virus spreading processes is proposed, that allows for antidotes to be modelled as well.

In a network environment, Giess (Giess, 1990) proposed an early model for assessing the performance of a network - equipped with antidotes - under a viral attack.

In (Katsikas, 1996), the issue of what happens when a virus is injected into a stable computer network, given that some nodes of the network have been equipped with antidotes against this type of attacks was studied. This issue was addressed by assuming a model of the network, a model for a hypothetical virus resembling those most commonly encountered in viral attacks, by establishing the rules of the virus spreading process between nodes in the network, by establishing the rules of the virus spreading process within a network node and by establishing the rules of the antidote virus removal process.

An intrusion attack is based on stand-alone program(s), or commands entered directly by the keyboard. In many, the intruder's first goal is to gain access to hosts for which he/she has no access permission.

Most published work on intrusions covers the issue of intrusion *detection* rather than modelling of the intrusion process itself. Indeed, the work of Denning (Denning, 1987) led the way to a series of papers on the issue of how one can derive models useful for detecting intrusions in computerised environments, networked or not (see (Lunt, 1993) and references therein).

Informal models of the intrusion process, which merely classify the methods of attack or describe the intrusion process in a structured -yet informal- way have also appeared in the literature (Heberlein, 1990). It was only recently that a Markov chain model of the intrusion process for a stand-alone system was proposed and studied (Soh, 1995).

It appears then than no real formal models of either the virus attack or the intrusion process for networked environments are available. Such models not only provide very useful insight into the workings of the processes - thus allowing their better understanding - but can be instrumental in the design of practical countermeasures, e.g. by allowing the simulated (hence controlled) study of the attack process.

In this paper, initial work towards the development of such formal models is described. The paper is structured as follows: Section 2 describes the assumed network environment; section 3 describes the proposed generic formal model used to describe the attack process; section 4 describes a more detailed model of a virus attack against a network which results from the application of the generic model to the virus case. Finally, section 5 summarises our conclusions.

2 THE NETWORK ENVIRONMENT

Any network environment consists primarily of nodes, which are points where information flows come together or diverge, and of links which are bi-directional paths along which information is transmitted.

The properties of the links are determined by the nodes to which they are connected; if both nodes are operational, then the link is operational. If one node has been disabled, then the link is not operational, meaning that communication between the two nodes attached to the link is impossible.

When two network nodes communicate, they exchange data or one of them calls executable files resident at the other. The precise mode of communication is governed by specific communication protocols used in the network, as well as by the operating systems employed by the nodes involved.

We assume no specific protocol nor operating system herein, even though the precise evolution of the attack in the network does depend - to some extent - on both. The reason for this is that we are interested mainly in assessing the overall network behaviour rather than the attack spread within each of its nodes.

Every node in the network can be assumed to resemble a user in a non-distributed computer system. How often calls are made to other nodes and how often executable files residing in the same node are called, constitute, among others, characteristics of what is termed the node profile; this is assumed to vary with each node. Thus, some node may make heavy use of the network by making frequent calls to other nodes, the node may make heavy use of its own files, or, conversely, it may make light use of the network or of its own files.

Every node in the network can be at exactly one of the following four allowable states:

1. *Disabled:* The node has lost its capability to communicate with other nodes as a result of the attack against the network. This may have happened either because the node was incapacitated as a result of the attack or because the node was isolated by the network defender following detection of extensive compromise. The only way that this node may be reinstated to a normal state is by human intervention.
2. *Normal:* The node performs its designed operation.

3. *Penetrated/Infected:* At least one of the files residing at the node has been infected by a virus, or the node has been compromised due to an intrusion. The node still functions, but, when communicating with some normal node, it may either infect the latter or be used as an intermediate for launching an attack against it.
4. *Protected:* A normal node equipped with some protection and correction mechanism. In addition to its ordinary functions, the node can check the condition of other nodes and can also cure penetrated/infected nodes by moving its correcting capability to them.

The fourth allowable state implies that nodes have the capability to initiate sequences of actions at other nodes. Even though this may seem unrealistic, it can in fact happen if one considers that instead of automatically initiating actions at some remote node, a message could be sent to the human operator of the remote node, who could then proceed with carrying out the required actions.

Initially, the network is stable, i.e. all nodes are operating normally. At some point in time, an intruder launches an attack against one or more nodes of the network, thus turning the latter into the penetrated/infected state.

However, the network's defender (the security officer), can have some of the nodes in the network equipped with protection mechanisms.

Therefore, the initial network configuration is as follows: out of the total number N of the nodes in the network, N_n are normal, N_i are penetrated/infected and N_p are protected, where $N = N_n + N_i + N_p$.

The percentage of penetrated/infected nodes is a parameter depending on the intruder's intentions and on the type of the attack, whereas the percentage of the protected nodes is a parameter depending on the wishes of the network defender.

The initial situation evolves in time because network nodes will change state. Hence, the rules of the game are completely specified by specifying the rules governing the transition between states for the nodes of the network. These transitions are events taking place at random times and with certain probabilities.

3 ELEMENTS OF A FORMAL GENERIC MODEL OF AN ATTACK

3.1 The basic entities

For the purposes of this paper let:

- $at \in AT$ be a set of attack types AT={virus, Trojan horse, Logic bomb, insider attack, password guessing, system programming attack, outsider access violation, denial of service, eavesdropping, active network attack, worm, chain letter, antivirus dropping Trojan, ...}
- $an \in AN$ be a set of attacker's intentions, AN={accidental, intentional}
- $e \in E$ be a set of possible effects of the attack, E={passive, active, lethal}
- $m \in M$ be a set of attack methods
- M={impersonating, wiretapping, traffic flow analysis, replay, covert channel, ...}
- $p \in P$ be a set of possible attack phases
- P={preparation, execution, post attack} and
- $id \in ID$ be a set of attacker identities, including the value "unknown"

Definition: An *attack instance* (*ai)* is a sixtuple *ai=<at,an,e,m,p,id>*.

Definition: An *attack salient description* (*asd*) is a quadruple *asd=<at,an,e,m>* of the first four components of an attack instance.

Definition: An open network (*on*) is a quadruple *on=<network_type, connectivity, flaws, services>*, where:

- *network_type* is a value of a set of types (e.g. IBC, B-ISDN, etc.),
- *connectivity* is a value of a set of connectivities (e.g. ring, bus, tree, general, hierarchical, etc.),
- *flaws* is a set of network flaws or weaknesses (e.g. remote debugging allowed, etc.), and
- *services* is a set of fundamental network services (e.g. network management services, etc.).

Definition: Let *AI* be a set of attack instances and *ON* a set of open networks.

3.2 How the entities interact - compatibility issues

Not all types of attack methods can have a possibility of success with all network types. Similarly, not all attacks can exploit all network flaws.

Two binary relations are used to describe exactly this compatibility information:

1. Relation **Exploits**(*attack, flaw*). It has two attributes, namely, the attribute *attack* is an attack instance *ai*, and the attribute *flaw* is an element of the set of flaws of an open network *on*. Thus, *attack*$\in$*AI* and *flaw*$\in$*flaws,* where *on*$\in$*ON.*
2. Relation **Applies**(*open_network, attack_method*). It has two attributes: the attribute *open_network*$\in$*ON* and the attribute *attack_method* is such that *attack_method*$\in$*M.*

3.3 Dynamic and stochastic aspects of the interaction of entities

Each network node or subnetwork can be at one of the following states: S_1=disabled, S_2=normal, S_3=infected/penetrated, S_4=protected. Let n_1, n_2, n_3, n_4 be the number of network nodes and $S_1 \ldots S_4$ the sets of network nodes in each of the above states. Obviously, $|S_i| = n_i$ and $|N| = n_1 + n_2 + n_3 + n_4$ where N is the set of network nodes.

Definition: An instantaneous network description (*ind*) of an open network *on* is a vector $ind = < S_1, S_2, S_3, S_4 >$.

The possible spread of an attack can be characterised by two probability distributions. In addition, let:

1. $i \in I$ be a set of useful specific information about a network. Let *f*(*ai,on,i*)=**Prob**{attack instance *ai*, applied to open network *on* will first obtain the information *i*}. Usually, *f*(*ai,on,i*)=0 when the phase of *ai* is not the preparation phase.
2. *ai* be an attack instance (at its execution phase), $N_1 \subseteq N$ be a set of connected network nodes already being at state *infected/penetrated* and $n \in N$ a node at state *normal.*

Also let $g(ai, on, N_1, n) \overset{\Delta}{=}$ **Prob** {attack instance *ai* will succeed at the next time instance (equal to a network line transmission delay) to convert the state of *n* to *infected/penetrated*}.

Note that $g(ai, on, N_1, n)=0$ when *n* is not a neighbour of any node of N_1. Also, note that $f=0$ when *ai* and *on* do not respect the compatibility relations defined previously. Depending on the details, *f* and *g* can be defined implicitly. For example, it could be the case that $g(ai, on, N_1, n)$ is a constant <1 independent of N_1. Then the probability of the attack spreading to a radius of more than logarithmic number of nodes (in $|N|$) from its origin, is very small.

Together, *f* and *g* define the probabilities of transitions from an *ind* to another *ind*. Thus, a Markov chain is defined, with instantaneous network descriptions as its states and transitions defined according to *f* and *g*. The chain is clearly irreducible and aperiodic and hence it has a uniquely defined steady state.

The steady state probability vector determines the possibility of success of an attack. The time duration of the transient is another useful piece of information, since it provides us with the time frame for countermeasures.

Note that the connectivity of *on* is implicitly affecting the behaviour of the chain, and therefore one can study the possibility of success of an attack with respect to various network topologies.

Definition: A network state *T* at time *t* is the vector $(t) = <|S_1(t)|, |S_2(t)|, |S_3(t)|, |S_4(t)|>$.

Let *p(state1,state2)* denote the transition probabilities between the four allowable states of a node, e.g. *p(2,4)* represents the probability of a transition from *normal* to *protected* state.

By assuming that transitions at each node are independent of each other, the network state at time *(t+1)* denoted by *T(t+1)* will be determined by the population changes per state as guided by the transition probabilities. For example:

$$N_2(t+1) = N_2(t) - \sum_{i=1,3,4} \Delta N_{2i}(t) + \sum_{i=1,3,4} \Delta N_{i2}(t),$$

where $\Delta N_{ij}(t)$ denotes the number of nodes changing state from state *i* to state *j* between time instants *t* and *(t+1)*. $\Delta N_{ij}(t)$ are random variables which are Bernoulli distributed, i.e.

$$\text{Prob}\{\Delta N_{ij}(t) = x\} = \binom{N_i(t)}{x} P^x(i,j)(1 - P(i,j))^{N_j(t)-x}$$

Thus, an irreducible and aperiodic Markov chain is defined, whose states are the network states (initially $N_1(0) = 0$) and whose transitions are governed by the Bernoulli distributions of the relative population changes. The steady state vector of *T* gives information about how the attack power and the protection scheme affect the attack spread in the network. The duration of the transient phase determines the time that has to pass until the network stabilises at some steady state. This time information is then valuable because it gives us the allowable time frames for external actions (countermeasures, alerts, etc.).

4 AN EXAMPLE: A VIRUS ATTACK

In order to illustrate the use of the generic model described above, assume a virus attack against an open network with bus topology. For this case, the transition rules between the four allowable states have been defined (Katsikas, 1996) to be (the symbol a→b, where $a, b \in \{1, 2, 3, 4$ denotes the transition from state a to state b.):

❶→❷: This may only happen by means of human intervention. It is a random event happening with certainty, taking time distributed exponentially.

❷→❸: This is a random event, depending on whether a normal node will establish contact with an infected node. We assume that any node equiprobably communicates with any other node, which is not always the case. Hence, the probability that a normal node will establish communication with an infected node is $N_i/(N-1)$, where N_i is the total number of infected nodes at any time and N is the total number of nodes in the network. The distribution of the time required to complete the transition depends on the node profile (i.e. how often the node communicates with other nodes), but it can be assumed to be uniform, its limits being allowed to vary from node to node.

❷→❹: This transition may occur in two distinct cases: when a disabled node is serviced and when a normal node communicates with a protected node and it is decided to install the protection mechanism to that node as well. When a disabled node is serviced, thereby becoming normal, it is decided to manually install the protection mechanism as well. The probability of this happening is a parameter controllable by the network defender and it is allowed to vary, whereas the time distribution is exponential. When a normal node communicates with a protected node and it is decided to install the protection mechanism to that node as well. The probability of this happening is the product of two factors: First the probability of the node to communicate with a protected node, which is equal to $N_p/(N-1)$, where N_p is the total number of protected nodes in the network at the time; second the percentage of normal nodes that the network defender wishes to transform to protected, which is a parameter controllable by the network defender and is allowed to vary. The time required to perform the transition in this case is assumed to be negligible.

❸→❶: This is a random event, depending upon the virus activation process and the process of virus spread within a single node. For our purposes, we assume the following regarding this process: Each node i stores n^i executable files, each of them equiprobable to be executed. Thus, the probability of an infected file to be executed (thus making the virus memory resident) is equal to k_i^i/n^i, where k_i^i is the number of infected files in node i. The probability of an uninfected file to be executed (thus turning to infected) is k_n^i/n^i, where k_n^i is the number of normal files in node i. It is assumed that once the virus is memory resident, it infects all subsequently executed files with probability one and takes negligible time. The distribution of time intervals between successive file executions is uniform and depends upon the profile of the user associated with the node, hence it varies with each node. As discussed previously, the virus is activated (i.e. disables the node), with certainty, when a given number of files have been infected or after a given number of hours of work, whichever comes first.

❸→❷: Even though a node may be infected, it still communicates with other nodes. If it establishes communication with a protected node, the latter may disinfect the former, thus reinstating it to the normal state. This is a random event happening with probability $k_\alpha/(N-1)\times eff$, where eff is the efficiency of the protection mechanism, i.e. the percentage of cases when the protection mechanism successfully detects the presence of a virus and removes it; clearly $0 \leq eff \leq$. The time distribution for the transition is uniform, its limits being determined by the node profile (i.e. how often the node communicates with other nodes), as well as by the time taken to scan and clean all files in the node.

❹→❸: Even though a node is protected, it may still become infected, since the protection mechanism has not been assumed to be totally foolproof. This is a random event, happening with probability $N_i \times (1-eff)/(N-1)$. The time required to perform the transition is again uniformly distributed with limits determined by the node profile.

However, one should note that there are some impossible transitions. In detail:

❶→❸: This transition is impossible.

❶→❹: This transition is also impossible; a disabled node may not become protected without becoming normal first.

❷→❶: This transition is also impossible; a normal node cannot be disabled unless it has first been infected.

❸→❹: This transition is impossible; an infected node cannot be made protected without becoming normal first.

❹→❶: This transition is impossible; a protected node cannot become disabled unless it becomes infected first.

❹→❷: This transition is impossible; a protected node has no reason to become normal unless the network defender decides so, perhaps due to an overload in the network traffic. However, we have arbitrarily assumed that this may not happen, even though the model does allow for such interventions.

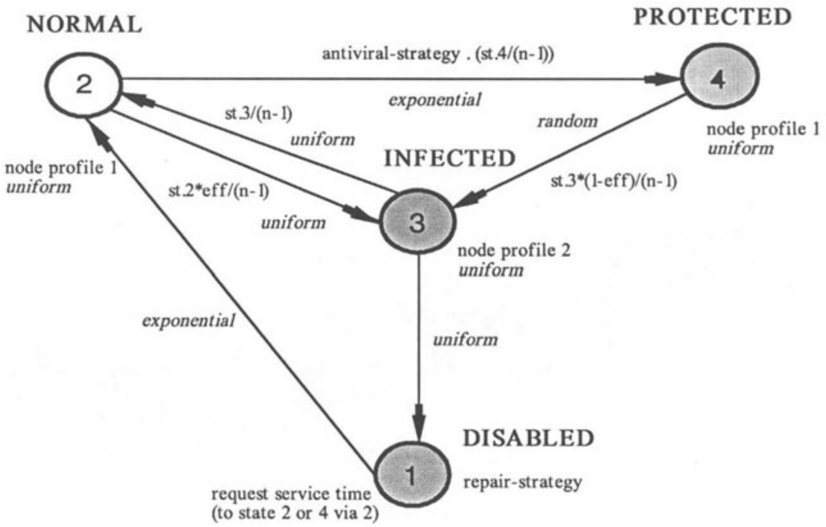

Figure 1 State transition diagram

The full state transition situation is graphically depicted in Figure 1. The model described above has been studied in considerable detail by simulation. The main results of this study have been reported in (Katsikas, 1996).

5 CONCLUSIONS

Malicious attacks against open networks constitute an area of growing interest, since their impact can be quite disastrous for every open environment.

In this paper, the beginnings of a formal generic model describing the process of a malicious attack against a computer network have been proposed, leading to a Markov chain description. The proposed model can be used for:

- Better understanding the nature of this threat.
- The preparation of a simulator, capable of analysing the impact of attacks against a computer network.
- The analytic and quantitative study of the power of several attacks versus the effect of protection mechanisms.

6 REFERENCES

Adleman, L. (1990) An abstract theory of computer viruses, in Hoffman L. (Ed.), *Rogue Programmes: Viruses, Worms and Trojan Horses,* Van Nostrand, pp. 307-323.

Cohen, F. (1987) Computer viruses: Theory and experiments, *Computers & Security,* Vol. 6, no. 1, pp. 22-35.

Cohen, F. (1989) Computational aspects of computer viruses, *Computers & Security,* Vol. 8, no 4, pp. 325-344.

Denning, D. (1987) An intrusion-detection model, in *IEEE Transactions on Software Engineering,* Vol. SE-13, pp. 222-232.

Giess, S. (1990) *Network stability under viral attack,* Royal Signals & Radar Establishment, NTIS AD-A229 274 Report, United Kingdom.

Guinier, D. (1991) Prophylaxis for virus propagation and general computer security policy, *ACM SIGSAG Review,* Vol. 9, no. 2, pp. 1-10.

Heberlein, L., Dias, G., Levitt, K., Mukherjee, B., Wood, J., Wolber, D. (1990) A network security monitor, in *Proc. of the 1990 IEEE Symposium on Research in Security and Privacy.*

Katsikas, S., Spirou, T., Gritzalis, D., Darzentas J. (1996) A model for network behaviour under viral attack, *Computer Communications* (to appear).

Kephart, J. and White, S. (1991) Directed graph epidemiological models of computer viruses, in *Proc. of the 1991 IEEE Symposium on Research in Security and Privacy,* pp. 343-359.

Lunt, T. (1993) A survey of intrusion detection techniques, *Computers & Security,* Vol. 12, no. 6, pp. 405-418.

Ostrowski, R. and Yung, M. (1991) How to withstand mobile virus attacks, in *Proc. of the 10th ACM Symposium on Principles of Distributed Computing,* pp. 51-59.

Soh, B.C. and Dillon T.S. (1995) Setting optimal intrusion-detection thresholds, *Computers & Security,* Vol. 14, pp. no. 8, 621-631.

7 BIOGRAPHIES

Sokratis Katsikas holds a Diploma (Electrical Engineering) from the Univ. of Patras, Greece, an MSc (Electrical and Computer Engineering) from the Univ. of Massachusetts at Amherst, USA, and a PhD (Computer Engineering) from the Univ. of Patras, Greece. He is an Associate Professor of Informatics with the Dept. of Mathematics of the Univ. of the Aegean, Greece. Prof. Katsikas is the Vice-president of the Greek Computer Society and the representative of Greece to CEPIS Special Interest Group on Security and Legal Issues.

Dimitris Gritzalis holds a BSc (Mathematics) from the Univ. of Patras, Greece, an MSc (Computer Science) from the City University of New York, USA, and a PhD (Informatics) from the Univ. of the Aegean, Greece. He is a Lecturer with the Dept. of Informatics of the Athens Univ. of Economics & Business, Greece. Dr. Gritzalis is the President of the Greek Computer Society and the representative of Greece to IFIP Technical Committee 11 (Security in Information Processing Systems).

Paul Spirakis holds a Diploma (Electrical Engineering) from the National Technical Univ. of Athens, Greece, and an MSc and a PhD (both in Computer Science) from the Univ. of Harvard, USA. He is a Professor with the Dept. of Computer Engineering & Informatics of the Univ. of Patras, Greece.