A PREVENTION STRATEGY FOR SECURITY: A BAYESIAN APPROACH TO ANOMALIES DETECTION

Franco Arcieri and Andrea Dimitri
Nestor Laboratory
University of Rome "Tor Vergata", Italy
{arcieri, dimitri}@nestor.uniroma2.it

Abstract: Intrusion detection is one of the new frontiers in network security, but almost every implemented system is in trouble when it has to deal with new kind of attacks or when it has to give a real time response to predefined attacks. In this work, we assert that the way of improving intrusion detection is to consider the semantic aspects of the communication protocols. Furthermore, we analyze an intrusion detection model that tries to reach this goal putting together database logical design rules and new rules from Bayesian reasoning. In the final section, we sketch some application of the model and we show how to implement the model and how to face existing attacks using the model itself.

1. INTRODUCTION

Network Intrusion Detection is the process of identification of malicious behaviours that damage a network and its resources. Intrusion Detection systems are usually classified in misuse-based and anomaly-based.

Solutions using misuse-based techniques contain a certain number of attack descriptions, called "signatures". When these systems recognize a signature instance in the data flows audited, they judge it as an attack. The positive characteristic of the misuse-based systems is that they usually generate very few false alarms, the negative one is that they recognize only previously defined attacks (the "signatures").

The anomaly–based techniques follow a complementary approach: they are based on models or profiles that capture the "normal" behaviour of a

E. Nardelli et al. (eds.), *Certification and Security in Inter-Organizational E-Service*

system. When something in the data flows is different from those models or profiles, they judge it as an attack. As known in literature, these systems are able to identify new attacks with unknown patterns, causing many false alarms.

For these reasons, both misuse-base systems and anomaly-based systems are not so frequently used: the system manager and the security manager have to cooperate every time to classify anomalous network events distinguishing real attacks from not dangerous behaviours. It is therefore evident that anomaly-based techniques depend exclusively from the ability to define their underlying models/profiles.

The challenge is to build a model that allows adding, within itself, rules and semantics that do not belong only to a static, predefined data set.

In that sense, some intrusion detection systems provide a sort of semantic functionalities in the Protocol Anomaly Filters: these filters search for a bad use of a communication protocol, according to the standard protocol rules.

An existing example of semantic based anomaly detection system is the human immune system.

The goal of the immune system of an organism is to defend it against harmful diseases and infections. The immune system is virtually able to recognize any foreign cell or molecule and eliminating it from the body. To do this, the immune system performs an operation of pattern recognition, with the perspective to distinguish molecules and cells of the body (called "self") from foreign ones (called "nonself"), potentially dangerous. The architecture of the immune system is multilayered, with defense systems provided in every one of the levels. The outmost layer, the skin, is the first barrier against infections. A second barrier is physiological: there are conditions such as pH and temperature that provide inappropriate living conditions for some foreign organisms (pathogens). Once pathogens have entered the body, they are handled with two types of approach: the innate immune system and the adaptive immune system. Examples of the first defense system are macrophages, that circulate in the organism and eat extra cellular molecules and foreign materials cleaning the body. Nevertheless, the most sophisticated system is the second one. It is called "adaptive" because it is responsible for immunity that is adaptively acquired during lifetime of the organism. Because the adaptive immune system provides the most potential from a computer security viewpoint, we will focus on it. The adaptive immune system can be viewed as a distributed detection system. It is constituted primarily of white blood cells, called lymphocytes. These small independent detectors circulate through the body in the blood and the lymph systems. They are negative detectors, because they act against foreign patterns, ignoring self patterns. Detection or recognition is caused by the affinity of the receptor that covers the surface of the lymphocyte and the pathogens. Detection is approximate: hence, a lymphocyte will bind with several different kinds of (structurally related) pathogens. To recognize most different pathogens is required a huge diversity of lymphocyte receptors and a great number of them. This diversity is partly achieved by generating lymphocyte receptors through a genetic process that introduces a huge amount of randomness. Even if receptors are randomly generated, in the organism there are not enough lymphocytes to provide a complete coverage of the space of all pathogen patterns. One estimate is that there are 10^8 different lymphocyte receptors in the body at any given time, which must detect potentially 10^16 different foreign patterns. To address this problem, the immune system has several mechanisms that make it dynamic and specific. Protection is made dynamic by the continual circulation of lymphocytes through the body and by a continual turnover of the lymphocyte population. The life of lymphocytes is typically a few days: the random renovation process will replace them by new ones. Dynamic protection increases the coverage provided by the immune system over time: the longer a pathogen is present in the body, the more likely it will be detected because it will meet a greater diversity of lymphocytes. Protection is made more specific by mechanism of learning and by memory. If the immune system detects a pathogen that it has not encountered before, it undergoes a primary response, during which it "learns" the structure of the specific pathogen, i.e. it evolves a set of lymphocytes with high affinity for that pathogen, through a process called affinity maturation. Affinity maturation produces a large number of lymphocytes that have high affinity for a particular pathogen, which accelerates its detection and elimination. Speed of response is important in the immune system because most pathogens are replicating and will cause increasing damage as their numbers increase. Last remark, each individual in a population has his tribe of lymphocytes, specific for each individual and different from another individual. This diversity of immune systems across a population greatly enhances the survival of the population as a whole.

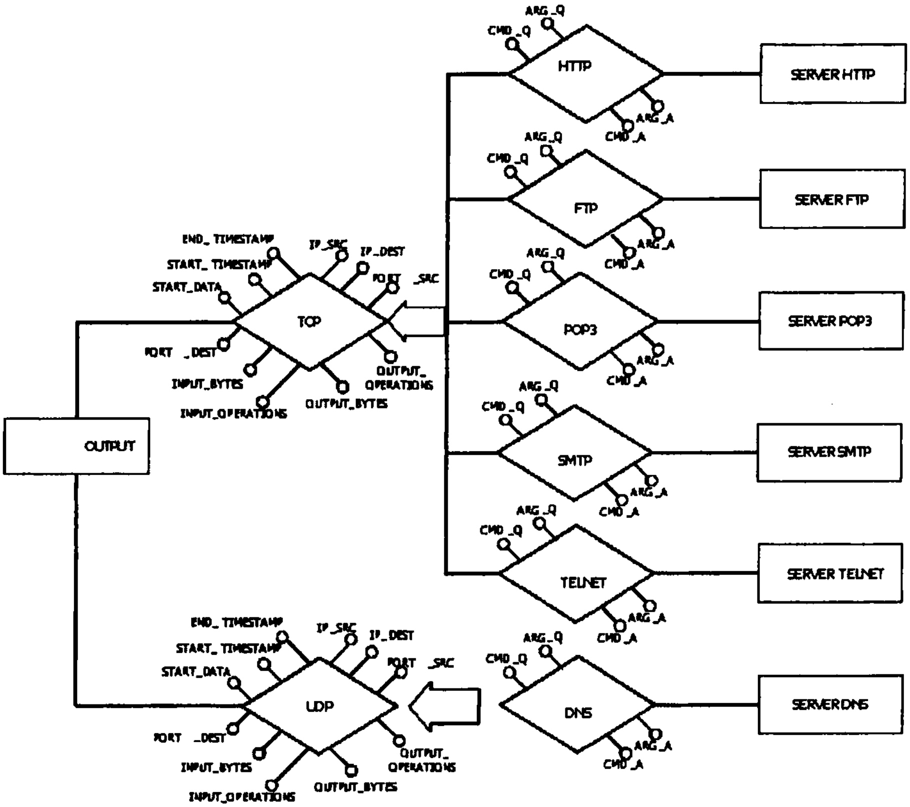
Another reference that is similar to our approach could be the observation of a behaviour model. Let us suppose to observe a network security expert when he analyses outputs of some documentation system (router and firewall logs, sniffer output, ...) with the goal to discover network anomalies. He has in his mind a rule system that comes from his experience and from his background knowledge. This system is updated continuously because the expert continuously observes the world around him. When a network pattern does not match with one of the rules of his system, the expert starts a more detailed observation activity around the non-matching event.

2. THE MODEL

In the context of Protocol Anomaly Detection is often said that most attacks are violations of the rules that define a protocol. This statement must be understood in a general context: let recall the DOS (denial of service) attack, that is based on a rapid succession of TCP connect operations. A TCP connect operation, not followed by other TCP operations, it is not a wrong operation in the context of the TCP protocol: we can assert that it is an anomalous operation. Some times no other TCP operations follow a TCP connect because the other peer of the connection cannot complete his work session. In addition, well-known worms or backdoors are difficult to recognize as TCP protocol violations.

An attack is a violation of an expected behaviour. This behaviour must be defined in a rule system.

Starting from this definition, we will build our model. We want to refer to a more general context: the context of the relational model. The relational model allows us to design complex scenarios. Network attacks, when they are not only simple service negation, are made by complex phases that cross the communication protocol stack. Therefore, to design complex scenarios we use the tools of the relational model. This model has a drawback: it hasn't tools to manage the semantic of the world we are representing, in particular, the semantic of the tuple to be inserted in the database. The logical design of databases and, in particular, the integrity constraint theory, could support us facing this limit. In this way we add the network rules to our model. However, this is not enough; the next step is to extend the classical concept of integrity constraint. To obtain this, we will use solutions deriving from Bayesian reasoning.



2.1 Bayesian reasoning

In a nutshell, the Bayesian probability of an event x is a person's degree of belief in that event (subjective approach). Whereas in classical approach probability is a physical property of the world (e.g. the probability that a coin will land heads) in the Bayesian approach probability is a property of the person who assigns it. Here we want to notate that the Bayesian approach doesn't neglect the classical one. When assigning a probability to an event of flip of a coin, a person can take in account only physical properties of the context of the flip. The classical concept of probability can become the starting point for building a Bayesian probabilistic model. But in the model have to enter also other source of knowledge, such as prior knowledge and historical information.

If we have a dataset that we consider significant, by the Bayesian approach we can transform the problem of probability assignment in a learning problem starting from a dataset. To show this fact, consider an irregular coin or a coin we are not sure it is regular. If we throw the coin up in the air, the result can be head or tail. Suppose we repeat the trial N + 1 times, making sure that the physical properties of the coin and the conditions under which it is thrown out remain stable over time. From the first N observations, we want to determine the probability of heads on the N-1h toss. In the classical analysis of this problem, we assert that there is some physical probability of heads, which is unknown. Probability does exist and it is defined by the physical conditions of the trial, but it is unknown. Therefore it is object of estimation using classical statistical inference procedures. We then use this estimate as our probability for heads on the N+1th toss. In the Bayesian approach, we also assert that there is some physical probability of heads, but we encode out uncertainty about this physical probability using (Bayesian) probabilities, and use the rules of probability to compute our probability of heads on the N+1th toss.

We need some notation. We denote a variable by an upper-case letter (eg. X; Y;Xi; Θ), and the current instance, or a state or a value of the corresponding variable by the same letter in lower case (e.g., x; y; xi; θ). We denote a set of variables by a bold-face upper-case letter (e.g. **X;Y;Xi**) and we use a corresponding bold-face lower-case letter (e.g., **x**; **y**; **xi**) to denote an assignment of state or value to each variable in a given set. We use p(X = x / ξ) (or p(x/ ξ) as a shorthand) to denote the probability of the event X = x for a person with a state of information ξ. Other consideration, with the same formulas p(x / ξ) we denote the density function of variable X and the mass or cumulative function F(X). Whether p(x / ξ) refers to a probability, a probability density or a probability distribution will be clear from the contest.

Let's return to the irregular coin problem. Let Θ to be the variable whose values θ correspond to the possible true value of the physical probability of the event heads. The probability θ is a random variable and we express the uncertainty about Θ with the density function p(θ/ ξ). In addition, we use Xl to denote the variable representing the outcome of the l-th toss, for l = 1,........,N + 1, and D = {X1=x1, ... XN = xN } to denote the set of our observations.

Thus, in Bayesian terms, the irregular coin problem reduces to computing p(xN+1/D, ξ) from p(θ/ ξ). To do so, we first use Bayes theorem to obtain the probability distribution for Θ given D and background knowledge ξ:

$$p(\theta|D,\xi) = \frac{p(\theta|\xi)\ p(D|\theta,\xi)}{p(D|\xi)}$$

where

$$p(D|\xi) = \int p(D|\theta,\xi)\ p(\theta|\xi)\ d\theta$$

p(D/ θ, ξ) is the convergence point between the classical approach and the Bayesian approach. For the conditions of the trials we have a binomial distribution:

p(D/ θ, ξ)= θ(1- θ)

therefore first equation becomes

$$p(\theta|D,\xi) = \frac{p(\theta|\xi)\ \theta^{h}\ (1-\theta)^{t}}{p(D|\xi)}$$

where h and t are the number of heads and tails observed in D, respectively. p(θ / ξ) is the prior probability and p(θ /D,ξ) is the posterior probability, the effective object of dispute between classics and Bayesians. The quantities h and t are said to be sufficient statistics for binomial distribution. Finally, we average over the possible values of Θ (using the expansion rule of probability) to determine the probability that the N+1th toss of the coin will come up heads:

$$\begin{aligned} p(X_{N+1} = heads|D,\xi) &= \int p(X_{N+1} = heads|\theta,\xi)\ p(\theta|D,\xi)\ d\theta \\ &= \int \theta\ p(\theta|D,\xi)\ d\theta \equiv E_{p(\theta|D,\xi)}(\theta) \end{aligned}$$

where E means expectation or average, and p(θ / D,ξ) is the weight element.

To complete the exposure we have to spend some world about the prior probability p(Θ/ ξ). In the Bayesian approach this is a predefined input. One possible way is to get a standard distribution with some known properties.

Another way is the complete building of the probability distribution by a predefined knowledge state. We will see that in our model prior probability distribution is a basic concept. Prior distribution is often ignored where can become the right tool to introduce rules in a intrusion detection system.

Let's start to see how the above exposure enter in our model. In the field of logic design of a database there is the key declaration phase. A set of attribute is a key for a given entity set. We can try to review the key concept in the Bayesian approach. Generally speaking, for a given set of attributes **X**, parameter θ that includes the fact that the given set is a key set for the database, is a Boolean variable. If **X** is really a key $\theta = 1$; in the other case $\theta = 0$. In the Bayesian world θ become Θ; it is a random variable. An example: we take into consideration a teacher having an exam session. In the universe of persons, Name and Surname aren't a key attribute set. Supposing that in the same exam session arrive three persons with the same Name and Surname. This fact can become suspicious for the teacher. This last statement tells us that in the mind of the teacher {Name, Surname} could be a sort of key.

2.2 Generalized integrity constraints

2.2.1 Domain constraints

Every attribute or set of attributes has to be linked with a domain of possible values. This is the most elementary form of integrity constraint. But it is very important in the field of network information systems.

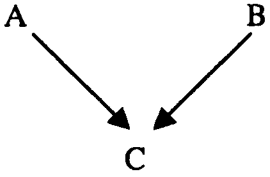
In the Bayesian world, in the space of admitted values for the domain we have to build a probability distribution. Almost every attack can be identified by some of its steps that are violation of one or more integrity constraints. We can think at backdoors, or various forms of worm viruses, or techniques that use man in the middle attacks to intermediate a communication.

2.2.2 Functional dependencies

It is easy to translate the concept of functional dependence in the direction of the concept of probabilistic dependence, and Bayesian inference methods supply all needed tools to express this new type of dependence.

Some observations can be done about the DOS attack. We want to demonstrate that DOS attack does not agree with the following two constraints: IP→TCP-CONNECTION and TCP-SEND→TCP-CONNECT. Consider the first functional dependence: in the TCP protocol the tuple (client ip, server ip, client service, server service) represents, at a given time, a key tuple for the TCP connection. In a static Local Area Network there are not two connections having the same key tuple. We said that in our Bayesian world this tuple is a random variable Θ(client ip, server ip, client service, server service). Seen as an event, the fact of to be or not to be a key it is a trivial random variable, because it takes value "true" with probability 1. But we can study the random variable Θ(client ip) at a given time. In this way, we formalize the fact that in some DOS attacks there are many open connections starting from the same ip address.

To better understand the second functional dependence violation, we will introduce the Bayesian concept of *explaining away*. Consider this simple probabilistic graph:



There are two causes A and B that compete to explain the effect C. Note that variables A and B are marginally independent but conditionally dependent. In terms of probabilities we have

$$P(A/B,C) != P(A)$$

This is known as Berkson paradox or *explaining away.*

Let us return to the DOS attack. In the TCP protocol, we have the functional dependency SEND→TCP-CONNECT. A TCP send operation functionally implies a TCP connect operation. In other words, the send operation is one of the possible effects of the TCP connect operation. Following the Bayesian approach we can assert that in the HTTP protocol, the WEB client/server protocol, TCP connect is the cause and TCP send is the effect. This means that, considering the *explaining away* concept, two TCP connect operations are independent *a priori,* but they become dependent *a posteriori* because a TCP send operation follows all TCP connect operations. In other words, a connect event is bound to the final result of the other previous TCP connection: a dialogue cannot continue if the present phases are not going ok. A large number of TCP connect without TCP send, in a client/server protocol is a clear signal of network anomalies.

2.2.3 Generalized Functional dependencies

In the field of logical db modeling, there are many types of constraints or dependencies (multi-valued dependencies, inclusion dependencies, ...). In a more general context, we introduce the concept of generalized functional dependencies. The basic feature of all forms of constraint says: if there are some tuples in the instance, then there are also other tuples, or some components of these tuples have some features. Using first order logic we can assert that a constraint is a sentence of form:

□x1...□xn [φ(x1....xn) -> □*z1...zk ψ(y1...ym)]*

where {z1...zk} = {y1...ym}-{x1....xn} and where φ is a atoms conjunction, possibly empty.

2.2.4 Probabilistic graphs

All above showed constraints can be mapped in the Bayesian world by the concept of probabilistic graph or Bayesian network.

A Bayesian network for a set of variables X = {X1,..........,Xn} consists of (1) a network structure S that encodes a set of conditional independence assertions about variables in X, and (2) a set P of local probability distributions associated with each variable. Together, these components define the joint probability distribution for X. The network structure S is a directed acyclic graph. The nodes in S are in one-to-one correspondence with the variables X. We use Xi to denote both the variable and its corresponding node, and Pai to denote the parents of node Xi in S as well as the variables corresponding to those parents. The lack of possible arcs in S encodes conditional independencies. In particular, given the structure S, the joint probability distribution for X is given by

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i)$$

The local probability distributions P are the distributions corresponding to the terms in the product of the above equation. Consequently, the pair (S; P) encodes the joint distribution p(x).

In our context, we refer to a probabilistic graph as a GIC (generalized integrity constraint).

2.2.5 Two layer, probabilistic model

A probabilistic model can be introduced as:

$$M = \{ \Omega, \beta(\Omega), m(\beta(\Omega)) \}$$

where Ω is an event space, $\beta(\Omega)$ is an algebra in that space and m is a probabilistic measure in the algebra. With a mapping between a semantic model (the logical model) and the Bayesian definition of probability we have introduced a probabilistic model. Explicitly:

$$M = \{ [DS], [DAG \text{ over } DS], m \}$$

DS = Database Schema
DAG over DS = Direct Acyclic Graph over the database schema
m = probabilistic measure over [DAG over DS].

We introduce another derived model:

$$M' = \{ [DAG \text{ over } DS], \beta([DAG \text{ over } DS]), m' \}$$

We have to explain in more details m and m'. For m construction we take in account Bayesian learning methods. For m' construction we consider a concept borrow from information theory.

2.2.6 Learning Probabilities in a Bayesian Network

In literature there are many source about this problem. Here we want just recall them (see bibliography) and recall principles about the problem. Methodology in learning probabilities for a Bayesian network is that of refining a defined prior distribution with other source of information. For example data capturing with an advanced sniffer.

We fix now two aspects, first one: the prior probability means rules. If a DAG represents a protocol or a work section, we know rules about it and we state that rules in the form of prior probabilities.

Second aspect: in our world we have simple DAG. We have many simple DAG. An open problem in the classical approach to IDS is the complexity to learn one big abstract and only historic model. Many data, a big dataset but many difficulties to work with it. An opposite approach is to have many small rules an to try to match them progressively.

2.2.7 A model in DAG space

The need of a model in DAG space and not only in event space, rises because we are building a dynamic model. To show this dynamism we need a correlation measure in DAG space. To do this we have to borrow some concepts from information theory.

Definition 1 (Information Gap) Let P=(p1,....,pk) and Q=(q1,.......,qk) two probability distributions. Information delay between P and Q is defined as follows:

$$D(Q/P) = \sum pi \log(pi/qi)$$

Conventionally we set 0*log(0)=0. In particular:

$$0 \leq D(Q/P) \leq \infty$$

$$D(Q/P) = 0 \leftrightarrow P \equiv Q$$

$$D(Q/P) = \infty \leftrightarrow \square i\ t..\ qi = 0$$

$$D(Q/P) \neq D(P/Q)$$

Definition 2 (Mutual Information) The information gap between the join probability distribution Pxy and the distribution PxPy is the mutual information between X and Y.

$$I(X/Y) = D(Pxy/PxPy)$$

Mutual information can be viewed as a measure of conditional dependence between variables X and Y. So we have I(X/Y)=0 if Pxy = Px*Py. Mutual information is symmetric in variables exchange. In information theory two variables are dependent if they exchange information. This a dynamic view of variables correlation.

Another important concept is the auto- mutual information, i.e. the mutual information of X and X.

$$P(xi, xj) = 0 \text{ if } I \mathrel{!=} j$$

$$P(xi,xi) = xi$$

So we have:

$$I(X/X) = \sum pij \log(pij/pi\ pj) = - \sum pi \log(pi).$$

Auto mutual information is called Shannon entropy.

Definition 3 (DAGs Mutual Information) The mutual information between two DAGs is the max mutual information between all couples of nodes one for DAG.

So we define m' as the DAG Mutual Information. We shall see that when we will explain model dynamic we will use DAGs Mutual Information concept and entropy concept. Return to immune systems. They have a mechanism to generate new lymphocytes in a normal situation (dynamism), and a mechanism to front an attack (specialization). The concept of entropy allows to generate random GIC; only those GIC with a small entropy will be considered valid, i. e. GIC that reflects the semantic of the model. The concept of DAGs Mutual Information introduces a metric in the space of DAGs, so when an event doesn't correspond to a gived DAG, the system will analyze near DAGs.

3. APPLICATIONS

3.1 Sniffers

We must have an proper environment allowing to start with design. The input database and its structure defines what we could build over it and subsequently, what type of surveys we could do. The great advantage of our approach is that we developed our analysis starting from a robust platform that is database designing tool. But this is true if there exists a tool allowing to write in the designed DBMS. Common sniffer or common log files can't develop this job. Lab Nestor developed an environment to do this. The tool is not only a sniffer but a protocol recognizer and rebuilder so that it can be used to insert directly output data in the planned database.

3.2 IDS system organization

The designed IDS is a global filter in Database insert query. An insert query starts with a network event. Before the insert step can start IDS verify all its constraints. There are two other aspects to focus. First, there is a parallel task that refresh continually the constraints set. It builds new constraint proposal, validates it with the current set of constraint (coherence check) and the current historical data and, if the validation step is ok, puts it in the constraints set. It review old constraints to validate their current validity. Second, if the system locates a constraint violation, it starts a new task of generation of adjacent constraints and check with data object of violation. Concepts introduced in the below section define how all this operations can be achieved.

3.3 IDS configuration steps

The crucial step in the configuration phase is the constraints definition. There is a set of constraint that depends on the rules of the networks. In this case we refer to classical constraints. For the other part of the constraints we can consider three step: protocols analysis, work sessions analysis and applications analysis. We remember that we have to state rules; habits are an output of the model. So the rule is: "usually there is a DNS request and after an HTTP request". The model learns that the host name of DNS answer and host called in HTTP request are the same. Then, this type of event sequences are regular, other different types of sequences are to be validated.

3.4 Attacks

Every attack is a violation of a constraint, but a crucial point is that we have to consider where the violation is located. Many attacks have a first step concerning substitution of an host with another one. We are thinking at many types of TCP protocol violations, at Mitnick attack, at idle scanning, at some types of man in the middle attacks. In all this cases violation is located in the client network. If IDS is located in server network to protect final services locations, then we note nothing. If an intruder extorts a password from the PC of a regular user, floods this PC and manages TCP connection in its PC, there are many violations but they all are client side. From the Server side is all regular. Security is a complex task and an organization task. IDS has to be a part of a system.

Here we can state that we built our IDS on a robust platform and this is important. There are many studies in functional dependencies and implication problem that define environment where a set of constraints is complete and close. In our world we can state exactly what we are looking, and what we aren't.

4. CONCLUSIONS

The main goal of this paper is to explain how to add semantic in the field of IDS and how this is a crucial step in the perspective of adding learning in IDS. The approach we follow has many differences from the standard one.

We start from an existing and robust model, that is conceptual and logic database design. All development in this field can potentially be translated in the IDS field. Future works could analyze the implication problem and the goal to define a model sound and complete. An IDS can't continue to be thought as an oracle that, given a network event, output an answer. Every IDS has to be characterized with its limits and properties.

Prior knowledge strongly enters in the firsts phase of designing. This is true because we have a concept of constraint simple, so every prior information it is easily introduced in our model.

The output of the designing phase is a set of simple constraints, a set that is refreshed during the life of IDS.

Another important feature of our proposal is in its dynamism facing a constraint violation. Rarely an attack is something of punctual and when it is punctual cannot be really dangerous. A final attack is a process and the true goal of an IDS is to understand it. Introducing a metric in the network space is a step in this direction. We formalized a new concept of network as interconnected system. This is the base to do complete intrusion detection. Only in this direction we can avoid the problem of false alarms.

References

[Coh87] F. Cohen Computer viruses. Computers & Security, 6:22-35,1987.

[FHS96] S. Forrest, S. Hofmeyr, A. Somayaji. Computer immunology. Communications of the ACM (Dec. 1996).

[Hec96] D. Heckerman, A tutorial on learning with Bayesian networks, Microsoft Research tech. report, MSR-TR-95-06, 1996.

[Das01] K. Das, Protocol Anomaly Detection for Network-based Intrusion Detection, GSEC, 2001.

[AGNT01] F. Arcieri, R. Giaccio, E. Nardelli, M. Talamo. A framewok for Inter-Organizational Public Administration Network Services. Proc. Of International Conference Advances in Infrastructures for Electronic Business, Science, and Education on the Internet (SSGRR 2001), L'Aquila, August 2001.

[Axe00] S. Axelsson, Intrusion Detection Systems: A taxonomy and Survey, Technical Report No 99-15, Dept of Computer Engineering. Chalmers University of Technology, Sweden, March 2000.

[DuM] W. DuMouchel, Computer Intrusion Detection Based on Bayes Factors for Comparing Command Transition Probabilities, AT&T Labs, Research.