

Causality and intervention for alarm correlation :  
A Naive Bayes approach for detecting coordinated  
attacks

—

Délivrable n°8

---

Salem BENFERHAT – [benferhat@cril.univ-artois.fr](mailto:benferhat@cril.univ-artois.fr)  
CRIL – Université d'Artois

Tayeb KENZA – [kenaza@cril.univ-artois.fr](mailto:kenaza@cril.univ-artois.fr)  
CRIL – Université d'Artois

# Projet PLACID

## Livrable 8

### Causality and intervention for alarm correlation: A Naive Bayes approach for detecting coordinated attacks

Salem BENFERHAT  
benferhat@cril.univ-artois.fr

Tayeb KENZA  
kenaza@cril.univ-artois.fr

#### Abstract

Alert correlation is a very useful mechanism to reduce the high volume of reported alerts and to detect complex and coordinated attacks. Existing approaches either require a large amount of expert knowledge or use simple similarity measures that prevent detecting complex attacks. They also suffer from high computational issues due, for instance, to a high number of possible scenarios. In this paper, we propose a naive bayes approach to alert correlation. Our modeling only needs a small part of expert knowledge. It takes advantage of available historical data, and provides efficient algorithms for detecting and predicting most plausible scenarios. Our approach is illustrated using the well known DARPA 2000 data set.

**Keywords:** Intrusion detection, Alert correlation, Attack prediction, Naive bayes.

## 1 Introduction

Intrusion Detection Systems (IDSs) are usually considered to be a second line of defense to protect against malicious activities. Traditional IDSs usually focus on low-level attacks or anomalies, and process alerts individually and independently, though there may be logical connections between them. Hence, the result of the detection is a set of alerts that reports elementary attacks.

However, intruders may use complex attacks to achieve their objectives. Often, they perform a series of actions (elementary attacks) in a well-defined sequence, called a scenario or an attack plan. Most of these actions are reported by IDSs, but the logical relationships between these actions (sequence of actions) are not detected by standard tools. Thus, the intrusion analysts or the system administrators are often overwhelmed by an important

volume of alerts to correlate manually. To this end, the goal of the correlation is to search for relationships between alerts.

Alert correlation has been studied in last years by several researchers. We can distinguish two main categories of approaches:

1. The first category focuses on reducing the volume of reported alerts, by either using similarity measures between attributes such as: the classification of attacks, the source and the target addresses, the identity of users, the detection time, etc [28] or by using alert aggregation mechanisms [5, 10] and clustering mechanisms [17].
2. The second category aims to detect relationships between actions to build attack scenarios. This category of approaches either uses preconditions and postconditions of actions to implicitly construct attack scenarios [6, 27, 22] or simply introduces the description of scenarios on a system [7].

Existing correlation methods allow to reduce the volume of alerts and to detect achieved attack plans. However, for attack prediction they generate an important even exponential number of scenarios which is time consuming and very difficult for analysts to investigate each scenario. Moreover, they involve a large amount of expert knowledge, either for defining a complete attack scenarios or for defining preconditions and postconditions associated with attacks.

In this paper, we present a new approach for alert correlation based on naive bayes that allows to detect coordinated attacks. Naive bayes represent a simple form of general bayesian networks, which are graphical models that allows to efficiently deal with uncertain pieces of information. Naive bayes have been used in many applications including in intrusion detection [1, 2, 14, 21, 25]. However, few works have applied bayesian network to alert correlation [13, 24]. In fact, the few existing works that apply bayesian methods to detect coordinated attacks require expert knowledge that the scenario (under the form of attack trees) will be prealably defined. In our approach such explicit representation of scenario is not required, and even we do not require to explicitely determine the set of actions involved in a scenario. Everything is obtained from historical data.

Our approach is efficient for predicting attack scenarios and does not involve a large amount of expert knowledge. Alert correlation process will be viewed in this paper as a classification problem. Given a set of observed actions and a set of intrusion objectives, our aim is to determine the most plausible objectives of an intruder. When observations does not favour any attack plans, our approach is also able to confirm that the traffic is normal.

The rest of this paper is organized as follows : section 2 introduces naive bayes. in Section 3, we discuss alert correlation objective and the main ideas of our approach. Section 4 presents our approach in tree phases : data observation preprocessing, construction of naive bayes and intrusion objective prediction process. We also illustrate our approach using DARPA 2000 data set. The last section concludes the paper.

## 2 A refresher on Naive Bayes

Bayesian networks are one of the most widely used graphical models to represent and handle uncertain information [16, 23]. Bayesian networks are specified by two components:

- A graphical component composed of a directed acyclic graph (DAG) where vertices represent events and edges represent relations between events.
- A numerical component consisting in a quantification of different links in the DAG by a conditional probability distribution of each node in the context of its parents.

Naive Bayesian networks [18] are very simple Bayesian networks which are composed of DAG with only one parent, representing the unobserved node, and several children, corresponding to observed nodes, with the strong assumption of independence among child nodes in the context of their parent.

Naive Bayesian networks are appropriate to deal with classification problems [12]. In fact, classification is ensured by considering the parent node to be a hidden variable stating to which class each object in the database should belong and child nodes represent different attributes specifying this object.

Hence, in presence of a training set we should only compute conditional probability since the structure is unique. This computation can be summarized as follows:

- Conditional probability for discrete attributes probabilities are computed from frequencies by counting how many time each attribute-value pair occurs with each value of the parent node.
- Continuous attributes are usually handled by assuming that they have a Gaussian (i.e normal) probability distribution which supposes a continued attribute  $A_k$ , we should compute the mean  $\mu$  and the standard deviation  $\sigma$ . Using these two values we can compute a probability density function for any value  $a_k$  of  $A_k$ .

The Gaussian distribution assumption for numeric attributes can be considered as a restriction of naive Bayesian networks since some attributes are not normally distributed.

Thus we can use generalization of this method by using the kernel density estimation [11] which is a non parametric density estimates for classification. This estimates does not assume any particular distribution of the attribute values and it is based on localizing for each target point  $a_k$  the observations close to it via a weighting function or kernel  $K_\sigma(a_k, a_i)$  which assigns a weight to each  $a_i$  based on its distance from  $a_k$ . The more popular choice of  $K_\sigma$  is the Gaussian kernel  $K_\sigma = \phi|a_i - a_k|/\sigma$  where  $\sigma$  is the standard deviation. Another alternative for continuous variables is to simply discretize them.

Once the network is quantified, it is able to classify any new object giving its attributes' values using the Bayes rule expressed by:

$$P(c_i|A) = \frac{P(A|c_i).P(c_i)}{P(A)}$$

Where  $c_i$  is a possible value in the session class and  $A$  is the total evidence on attributes nodes. The evidence  $A$  can be dispatched into pieces of evidence, say  $a_1, a_2, \dots, a_n$  relative to the attributes  $A_1, A_2, \dots, A_n$ , respectively. Since naive Bayesian network work under the assumption that these attributes are independent (giving the parent node  $C$ ), their combined probability is obtained as follows:

$$P(c_i|A) = \frac{P(a_1|c_i).P(a_2|c_i)...P(a_n|c_i).P(c_i)}{P(A)}$$

Note that there is no need to explicitly compute the denominator  $P(A)$  since it is determined by the normalization condition. Therefore, it is sufficient to compute for each  $c_i$  its likelihood, i.e.  $P(a_1|c_i).P(a_2|c_i)...P(a_n|c_i).P(c_i)$  to classify any new object characterized by its attributes' values  $A_1, A_2, \dots, A_n$ .

### 3 Coordinated actions and alert correlation

During monitoring information systems, IDSs report alerts when suspicious actions are observed. Some actions may not be observed for different reasons such as: actions are outside observation field or IDSs are not reliable, etc. Alerts reported every day represent a simple instantiation of a finite set of actions modelled in the system. For example, hundreds "ICMP ping" alerts can be detected, after a network scan, and representing instantiations of a same action "scan". As we will see later, in our approach, actions will represent variables of interest of naive bayes.

Generally, an intruder performs actions in well predefined order called "attack plan". In an attack plan, early actions alter a system or provide knowledge to an intruder, in order to perform later ones. An attack plan is modeled as a planning process for actions that transform an information system from one state to another, until reaching some targets, which we call "Intrusion objective" [6]. To determine this sequence, some approaches used a precondition and postcondition mechanism, which require a large amount of expert knowledge to define the preconditions and postconditions associated with each action. Moreover, in [6] when some actions are not observed, some virtual alerts are produced. This increases the number of possible scenarios, and the weighted alert correlation proposed in [3] only limits consequences of this explosion of high number of scenarios.

Our approach does not need to determine preconditions and postconditions associated with actions. It allows to directly predict the most plausible intrusion objectives using available observation history. In fact, we are not interested in determining the exact order in which a set of actions has been executed in order to achieve a given intrusion objective. We are more interested in first determining which actions that may be involved (whatever is the

order) in intrusion objective, and a tool that predict, online, whether an intrusion objective may be compromised or not. It is very important to note that our approach does not require expert knowledge, namely it will neither require preconditions and postconditions of actions such as in [6, 22, 27], nor an explicit representation of attack scenarios such as in [7]. It does not require the set of actions involved in attacks. In fact, the set will be determined experimentally.

In following, we use a weak definition of an attack plan which is defined as a set  $S = \{A_1, A_2, \dots, A_n, O\}$ , where  $A_i$ 's are an instance of actions and  $O$  is an objective of intrusion such as:

*$A_i$  has an influence on  $O$*

This definition is weaker than the one used in [3], since no conditions is required on preconditions and postconditions of actions. One possible definition of influence is:

*$A_i$  has an influence on  $O$  if  $P(O|A_i) > P(O)$*

Namely, learning  $A_i$  increases the feasibility that the intrusion objective will be compromised.

Some actions can be involved in many attack plans, and some objectives can be achieved by many attack plans. For example, a Denial of Service attack can be performed by a simple Ping of Death or Synflood, or by more sophisticated attack plans like Smurf. The objective of our approach is to detect attack plans as early as possible and to predict the most plausible ones. Given an intrusion objective, we can distinguish three kinds of actions:

- Actions with negative influence which decrease the propability to reach the intrusion objective, such as:  $P(O|A_i) < P(O)$
- Actions with positive influence which increase the propability to compromise the intrusion objective without fully reaching it, such as:  $P(O|A_i) > P(O)$  and  $P(O|A_i) < Threshold$ . This means that the probability to reach the intrusion objective increase without exceeding some threshold (50% for instance).
- Actions with critical influence which allow directly to reach the intrusion objective, such as:  $P(O|A_i) > P(O)$  and  $P(O|A_i) > Threshold$ . This means that the probability to reach the intrusion objective exceeds a given threshold.

The following section presents our approach, namely the application of naive bayes to detect coordinated attacks. As we already pointed out, we do not require a large amount of expert knowledge. We only need from the expert the definition of the set of "intrusion objectives". The definition of actions involved in possible scenarios, as well as the prediction of attacks scenarios are obtained automatically.

## 4 Modeling alert correlation by Naive Bayes

In this section, we explain how to model alert correlation using naive bayes and exploiting observations history. Our approach includes tree main steps:

1. **Data observation preprocessing:** this preprocessing step concerns the history of observations. The result of this step is a set of formatted data.
2. **Construction of naive bayes:** in this step we compute the conditional probability distribution of each variable in naive bayes.
3. **Intrusion objective prediction:** in this step we predict intrusion objectives by applying inference mechanism of naive bayes.

Our approach will be illustrated on first scenario of DARPA 2000 data set [9]. DARPA 2000 first scenario includes a Distributed Denial of Service (DDoS) attack run by a novice attacker. The premise of the attack is that a relatively novice adversary seeks to show his/her prowess by using a scripted attack to break into a variety of hosts around the Internet, install the components necessary to run a DDoS, and then launch a DDoS at a US government site. As a part of the attack the adversary uses the Solaris Sadmin exploit, a well-known Remote-To-Root attack to successfully gain root access to three Solaris hosts at Eyrie Air Force Base (AFB) [9]. The phases of the attack scenario are:

1. IP sweep of the AFB from a remote site
2. Probe of live IP's to look for the sadmin daemon running on Solaris hosts
3. Breakins via the sadmin vulnerability, both successful and unsuccessful on those hosts
4. Installation of the trojan mstream DDoS software on three hosts at the AFB
5. Launching the DDoS

In phase 1, the intruder performs an IP sweep of multiple subnets on the Air Force Base. He sends ICMP echo-requests in this sweep and listens for ICMP echo-replies to determine which hosts are "up". In phase 2, the hosts discovered are probed to determine which hosts are running the "sadmin" remote administration tool. In phase 3, the intruder tries to break into the hosts running the sadmin service. The attack script attempts the sadmin Remote-to-Root exploit several times against each host, each time with different parameters. At the end of this phase, the intruder gets a root access on tree hosts. In phase 4, the script performs a telnet login on compromised hosts and install necessary components for DDoS attack (mstream server and mstream client). In last phase, the intruder launches the DDoS against the victim.

Let us now describe the tree steps of our model.

## 4.1 Data observation preprocessing

To construct naive bayes, we first have to preprocess observed data in order to learn naive bayes representing intrusion scenarios. The data contain a set of alerts that reports executed actions and also information on intrusion objectives (whether they have been reached or not). We first gather all observed intrusion objectives into a single class called “Objectives-Intrusion” and we assign to each objective a number from 0 to N, where zero represents no intrusion objective and N represents the maximum number of possible objectives to protect. So, the domain of Objectives-Intrusion is  $\{0, 1, 2, \dots, N\}$ . For instance, the first scenario of DARPA 2000 data set only contains one objective, a DDoS attack, so the class will contain two values  $Dom(class) = \{0, 1\}$  (0 means intrusion objective is not reached, and 1 means intrusion objective is compromised).

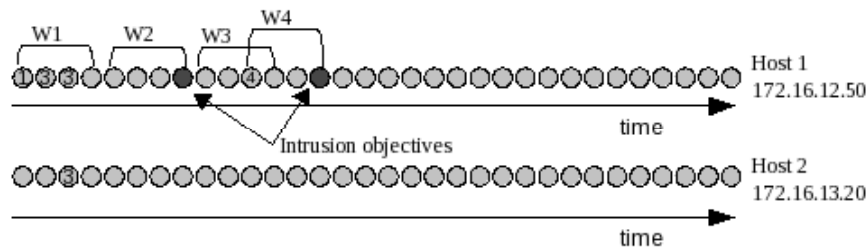


Figure 1: Data observation preprocessing

Then, we sort observed alerts according to chronological order and we split them into subgroups, according to a given window of time, determined experimentally (from few minutes until 02 hours). These windows represent usually the time required to achieve an attack plan (see figure 1). These windows are crucial to determine the set of actions involved in scenarios.

If an intrusion objective is observed inside a window, we move this window to left until it ends on this objective (see figure 1). We do this in order to ensure that all actions involved in each objective are present in a single window. Proceeding in this way means that some actions will be considered on two windows simultaneously. In figure 1 for example, action 4 will be considered on windows W3 and W4. W3 contains a normal data and W4 contains an attack plan (since at the end of window 4 an intrusion objective is compromised). So according to observation frequency of action 4 on normal or abnormal traffic we can determine if action 4 is suspicious or not.

Lastly, we will label each subgroup by a number corresponding to the observed intrusion objective. When no objective has been observed, the number 0 is assigned to say that this traffic does not contain any known attack plan. It is possible to observe more than one objective on a same window, so some subgroups can be labeled with several objectives number. Thus, we will get the observations under the form of vectors labeled by one or

more intrusion objectives from 0 to N (see table 1).

In fact, observations concern all hosts on the monitored network, so we have to apply the data preprocessing procedure described below for each host individually and merge results in a single table.

	<i>Action<sub>1</sub></i>	<i>Action<sub>2</sub></i>	...	<i>Action<sub>N</sub></i>	<i>Objectives</i>
<i>W<sub>1</sub></i>	<i>false</i>	<i>false</i>	...	<i>true</i>	1
<i>W<sub>2</sub></i>	<i>false</i>	<i>false</i>	...	<i>true</i>	0
...	...	...	...	...	...
<i>W<sub>n</sub></i>	<i>true</i>	<i>false</i>	...	<i>false</i>	2,4

Table 1: Preprocessed data observation

The preprocessed data observation procedure is summarized in the following pseudo-algorithm:

**Algorithm 1:** Data preprocessing

Data: Observation history

Result: Table of vectors;

**begin**

    Group all intrusion objectives in a single class called “Objectives-Intrusion”;

    Assign to each objective a number from 0 to N (0 represents no objective);

**for each host do**

        Sort observed actions chronologically and split them into groups, according to a given window of time;

**if an intrusion objective is observed inside a window then**

            └ Move this window to left until it ends on this objective;

        Label each subgroup (vector) by a numbers corresponding to the observed intrusion objectives;

    Merge vectors in a single table ;

**end**

Let us now illustrate this first step on a first scenario of DARPA 2000 data set. This data set contains a raw network traffic captured by a sniffer during the attack plan. We now need to describe actions, this is done with the help of an IDS (here Snort). After analyzing DARPA data set with Snort<sup>1</sup> we have observed that reported alerts concerns actions of table 2.

These actions will represent the set of variables of naive bayes. We also have observed a successful DDoS attack against some host, so this intrusion objective will represent the class of naive bayes.

In DARPA 2000 data set, the intruder tried to compromise every host in network. He got three compromised hosts after step 4 in DDoS scenario and he achieved the DDoS against the victim in the last phase. The DDoS attack has been achieved over a span of approximately 3 hours on 5 distinct phases, so we have taken 3 hours as a time window to split the reported alerts for every host individually. The preprocessing of DARPA 2000 dataset has resulted in 44 vectors labeled with DDoS when the window concerns a successful attack, or 0 when the window concerns a failure attack (normal traffic).

---

<sup>1</sup>Snort is an open intrusion detection system, <http://www.snort.org>

## 4.2 Construction of naive bayes

We construct one naive bayes for each intrusion objective. The reason why we consider one bayes network per intrusion objective instead of one bayesian network with a class variable containing all intrusion objectives is that intrusion objectives are not exclusive. It may happen that two different intrusion objectives  $O1$  and  $O2$  to be simultaneously compromised, namely  $P(O1) = P(O2) = 1$ . By defining one naive bayes per intrusion objective, it is possible to represent such situation. However, if only one naive bayes is used, one will only have  $P(O1) = P(O2) = 0.5$ . And if there are  $N$  intrusion objectives which are compromised, then one can not represent such situation and we will have  $P(O_i) = \frac{1}{N}$ , which means that the probability of each intrusion objective is weak. Now on the basis of this observation, we need to slightly modify Table 1, by splitting it in several tables, each of them only concerns a same intrusion objective. More precisely, for each intrusion objective we replace its number in the column “Objectives” by “true” and the other intrusion objectives by “false”. Thus, we get a table for each intrusion objective.

Table 3 shows preprocessed data for objective  $O1$ , the value “true” means that the action/objective has been observed on the corresponding window, the value “false” means that the action/objective has not been observed on the corresponding window.

	$Action_1$	$Action_2$	...	$Action_N$	$O_1$
$W_1$	<i>flase</i>	<i>false</i>	...	<i>true</i>	<i>true</i>
$W_2$	<i>flase</i>	<i>false</i>	...	<i>true</i>	<i>false</i>
...	...	...	...	...	...
$W_n$	<i>true</i>	<i>false</i>	...	<i>false</i>	<i>false</i>

Table 3: Preprocessed data for intrusion objective  $O1$

Figure 2 shows the naive bayes for first scenario of DARPA 2000 data set. The network structure is already defined, it only remains to compute the probability distribution (parameters).

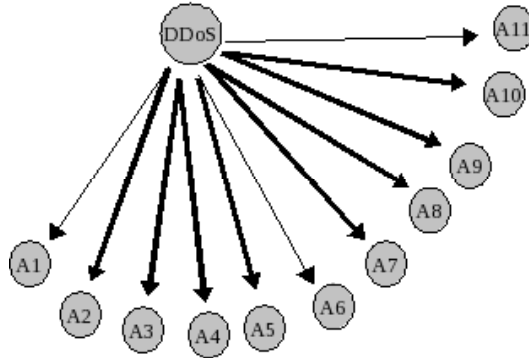


Figure 2: Naive bayesian network for DARPA 2000 dataset

The data allows us to estimate parameters that can be done by a simple frequency counts. However, when an attribute value does not occur together with a given class value produce a zero estimate for  $P(A|C)$ . This is problematic since it will wipe out all

information in the other probabilities when they are multiplied. To overcome this problem we will use Laplace estimator. Given a predefined factor  $f$ , if there are  $N$  matches of  $n$  instance for a  $K$  value problem, Laplace estimate the probability as  $(N+f)/(n+kf)$ . For two valued problem with  $f = 1$ , we get the well-known Laplace's of succession  $(N + 1)/(n + 2)$  [20].

Namely, once observations (alerts) are obtained and formatted as in table 3, we have to compute the CPT (Conditional Probability Table) for each variable.

The CPT associated with the class contains only two values, the probability to observe the intrusion objective ( $P(class = true)$ ) and the probability to not observe the intrusion objective ( $P(class = false)$ ), which can be computed as follows:

- $P(class = X) = \frac{NB(class=X)+1}{N+2}$  where :
- $X \in \{true, false\}$
- $NB(class = X)$  number of lines in table 3 where  $class = X$
- $N$  is the size of table 3

The CPT associated with variables in context of class will be computed as follows:

- $P(action_j = Y | class = X) = \frac{NB(action_j=Y \text{ and } class=X)+1}{NB(class=X)+2}$  where :
- $X, Y \in \{true, false\}$
- $NB(action_j = Y \text{ and } class = X)$  number of lines in table 3 where  $action_j = Y$  and  $class = X$
- $NB(class = X)$  number of lines in table 3 where  $class = X$

The naive bayes construction procedure is summarized in the following pseudo-algorithm:

**Algorithm 2:** Naive bayes construction

Data: Table of vectors

Result: Naive bayes;

**begin**

```

|   for each intrusion objective do
|   |   Replace its number in table of vectors by “true” and others by “false”;
|   |   Compute parameters for a corresponding bayes naive;
|   end

```

On DARPA 2000 data set, the a prior probability distribution of the intrusion objective DDoS and conditional probability distribution associated with variables are given in tables 4 and 5.

A priori there is a low probability that a DDoS will be observed. If DDoS is observed then it highly expected to have  $A_1, \dots, A_{11}$ . However, if DDoS is not observed then it is slightly expected to have  $A_1, \dots, A_{11}$ .

	False	True
DDoS	91.3%	8.7%

Table 4: Probability distribution of DDoS intrusion objective

		DDoS		
			False	True
A <sub>1</sub>	icmp_ping	False	4.65%	20%
		True	95.35%	80%
A <sub>2</sub>	rpc_sadmind_request	False	97.67%	20%
		True	2.33%	80%
A <sub>3</sub>	sadmind_ping	False	97.67%	20%
		True	2.33%	80%
A <sub>4</sub>	sadmind_root_query	False	97.67%	20%
		True	2.33%	80%
A <sub>5</sub>	sadmind_bof	False	97.67%	20%
		True	2.33%	80%
A <sub>6</sub>	icmp_reply	False	69.76%	20%
		True	30.24%	80%
A <sub>7</sub>	telnet_info	False	97.67%	20%
		True	2.33%	80%
A <sub>8</sub>	telnet_login_incorrect	False	97.67%	20%
		True	2.33%	80%
A <sub>9</sub>	telnet_bad_login	False	97.67%	20%
		True	2.33%	80%
A <sub>10</sub>	rsh_root	False	97.67%	20%
		True	2.33%	80%
A <sub>11</sub>	icmp_port_unreachable	False	55.81%	80%
		True	44.19%	20%

Table 5: Probability distribution of DARPA 2000 actions

### 4.3 Intrusion objective prediction

Our objective is to show how to infer (predict) intruder objectives given some recent observed actions.

The goal of inference is to estimate the values of hidden nodes, given the values of the observed ones. In naive bayes we are interested to infer the class values, given the values of some observed variables, this can be done by bayes formula :

$$P(class = x|y) = \frac{P(class = x).P(y|class = x)}{P(y)}$$

Where *class* is the hidden variable (in our case, the intrusion objective) and *y* is the observed evidence (in our case, the observed actions). When observed evidence concerns more than one variable, this formula can be written as follows:

$$\begin{aligned}
P(\text{class} = x|y_1, \dots, y_n) &= \frac{P(\text{class} = x, y_1, \dots, y_n)}{P(y_1, \dots, y_n)} \\
&= \frac{1}{\alpha} \cdot P(y_1, \dots, y_n|\text{class} = x) \cdot P(\text{class} = x)
\end{aligned}$$

Note that  $\alpha$  is a constant that can be obtained by normalization. Now,

$$\begin{aligned}
P(\text{class} = x, y_1, \dots, y_n) &= P(y_1, y_2, \dots, y_n|\text{class} = x) \cdot P(\text{class} = x) \\
&= P(y_1|y_2, \dots, y_n, \text{class} = x) \cdot P(y_2, \dots, y_n|\text{class} = x) \cdot P(\text{class} = x)
\end{aligned}$$

Recall that in naive bayes, by definition  $y_1$  in the context of *class* is independent of  $y_2, \dots, y_n$ . Hence :

$$P(\text{class} = x, y_1, \dots, y_n) = P(y_1|\text{class} = x) \cdot P(y_2, \dots, y_n|\text{class} = x) \cdot P(\text{class} = x)$$

And by iterating this process we get :

$$P(\text{class} = x, y_1, \dots, y_n) = P(y_1|\text{class} = x) \cdot P(y_2|\text{class} = x) \dots P(y_n|\text{class} = x) \cdot P(\text{class} = x)$$

In our context, the goal of inference is to compute the new probability of intrusion objectives given some observed actions. In a presence of a given observed action, there are three possible situations:

1. This action belongs to a single attack plan. In this case, we focus directly on other actions of attack plan.
2. This action belongs to several attack plans. In this case, we focus on the attack plan which this action has an important influence (positive or critical influence).
3. This action does not belong to any naive bayes. In this case, the prediction is only possible after the next update of table 3.

During the detection phase, we first initialize to zero some timeout variable. Each new reported alert will involve an evidence setting on each naive bayes. According to influence of this action on intrusion objectives, the probability of each intrusion objective will increase or decrease. We will focus on the attack plan (naive bayes) in which the probability of intrusion objective increases.

After each evidence setting, we verify the new probability to reach each intrusion objective. If the new probability to reach some intrusion objectives exceeds the threshold, we send an alarm to administrator or we proceed to an appropriate countermeasure. If none of intrusion objective probability exceeds the threshold, we wait for next alert. When the timeout expires and there is no objective such that its probability exceeds the threshold,

we can confirm that no attack plan is running. After the timeout expiration, we clear all evidences and we restart the detection phase.

The prediction procedure is summarized in the following pseudo-algorithm:

**Algorithm 3:** Objectives prediction

Data: Observed actions

Result: Prediction of intrusion objectives ;

```

begin
  Initialize timeout;
  while timeout is not expired do
    if an action A is observed then
      for Objective O = O1 to On do
        if Influence(A,O) = Negative then
          ⊥ No thing to do;
        if Influence(A,O) = Positive then
          ⊥ Focus in this intrusion objective;
        if Influence(A,O) = Critical then
          ⊥ Launch an appropriate counter measure;
      end
    end
  end

```

Let us now see how each action in DARPA 2000 first scenario influences the DDoS intrusion objective. From first view on DARPA 2000 naive bayes (figure 2), this structure does not seem to make the discovery of the attack plan, but after applying a simple computation of influence between variables and the class (the class is the intrusion objective), we can clearly identify the variables involved in the attack plan.

Table 6 shows the influence of each action with the new propability of intrusion objective. Namely, Table 6 shows how the probability that the DDoS will be observed, if one of actions  $A_1, A_2, \dots, A_{11}$  is observed.

The actions  $A_3, A_4, A_5, A_7, A_8, A_9$  and  $A_{10}$  have a critical influence on DDoS intrusion objective, because the probability to reach intrusion objective, given each one, exceeds 50% (see table 6). The actions  $A_2$  and  $A_6$  have a positive influence on intrusion objective, because the probability to reach intrusion objective, given each one, increases without exceeding 50%. Other actions have a negative influence on intrusion objective, because the probability to reach intrusion objective, given each one, decreases, so they are either false alarms, or they belong to another attack plan.

This analysis only concerns the first step of prediction, namely if one action will be observed. Now let us see how to expect this analysis on the basis of reported alerts.

	$P(DDoS A_j)$
$A_1$	7.4%
$A_2$	29.1%
$A_3$	76.6%
$A_4$	76.6%
$A_5$	76.6%
$A_6$	20.1%
$A_7$	76.6%
$A_8$	76.6%
$A_9$	76.6%
$A_{10}$	76.6%
$A_{11}$	4.1%

Table 6: Actions influence on DDoS

We have illustrated the individual influence of actions. Now, we will illustrate the prediction phase with two real complete scenarios, extracted from DARPA 2000 dataset. This two scenarios represent a successful and a failure cases of DDoS attack against two distinct hosts. These two scenarios were removed from learning step, namely from data preprocessing and naive bayes construction phases to be used for prediction phase. These two scenarios will be used to test our approach.

A prior probability (before receiving any reported alerts) that DDoS intrusion objective be reached is 8.7% (see table 4). After replaying the first scenario, Snort has detected this set of actions  $\{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}\}$ , which are sorted chronologically. After reporting each alert, we have set this evidence on naive bayes

	$P(DDoS A_j)$
$A_1$	7.4%
$A_1, A_2$	25.6%
$A_1, A_2, A_6$	47.6%
$A_1, A_2, A_6, A_{11}$	29.2%

Table 8: DDoS failure scenario

and we have inferred the new probability to reach DDoS intrusion objective (see table 7). According to new probabilities of intrusion objective, it is clear that after reporting  $A_3$ , we can confirm that DDoS intrusion objective can be directly reached, without waiting the timeout expiration. So we have to run a countermeasure at this time. This action has so influence on DDoS, because every host running Sadmin tool on DARPA 2000 dataset has been compromised.

After replaying the second scenario, Snort has detected this set of actions  $\{A_1, A_2, A_6, A_{11}\}$ , which are sorted chronologically. After reporting each alert, we have set this evidence on naive bayes and we have inferred the new probability to reach DDoS intrusion objective (see table 8). After reporting  $A_{11}$ , we have not observed any other action till the expiration of the timeout. Once the timeout is expired, we see that the intrusion objective probability did not exceed the threshold, so we can confirm that DDoS intrusion objective can not be reached and restart the detection phase. In both successful and failure scenarios, timeout was initialized when starting traffic replaying.

## 5 Related works

Bayesian networks have been introduced in intrusion detection area by several researchers, such as: Classifier for anomaly and misuse detection [2, 4, 19, 21, 25], Cybercrime detection [1], Plan recognition [13, 24], Distribute and multi-agent intrusion detection system [8, 14, 26], etc.

Axelsson [2] has proposed an interactive detection system based on simple Bayesian statistics combined with visualization component, in order to counteract the low detection rate and high rate of false alarms. His approach is based on the principles of Bayesian filtering like spam filtering in Email, and allows system to differentiate between normal and malicious access.

Abouzakhar et al [1] have proposed a bayesian learning networks approach to cyber-crime detection, in order to detect distributed network attacks as early as possible.

In [26] Scott has described a paradigm for designing network intrusion detection systems based on stochastic models. The principle is to base intrusion detection systems on stochastic models of user and intruder behavior combined using Bayes theorem.

Most recently, Gowdia et al [14] have developed a probabilistic agent-based intrusion detection system. This system is a cooperative agent architecture in which autonomous agents can perform specific intrusion detection tasks and also collaborate with other agents by sharing its beliefs on the same shared bayesian network. However, this bayesian network is provided by an expert.

Clearly, all above works apply bayesian networks to intrusion detection, but none of the cited works use bayesian networks to detect coordinated attacks. Now, among existing works the one of Qin and Lee [24] is close to our approach.

Qin and Lee [24] have proposed an approach for attack plans recognition and prediction using causal networks. In this approach, authors use attack trees to define attack plan libraries to correlate isolated alert sets. They then convert attack trees into causal networks on which they can assign probability distribution by incorporating domain knowledge to evaluate the likelihood of attack goals and predict future attacks.

Clearly, the main difference with our approach is that attacks trees should be explicitly defined by an expert in [24], which in our approach it is obtained automatically (we even do not to determine a priori the set of actions involved in attack scenarios). This is an important advantage of our approach. Our approach is easier to implement and does not need a large amount of expert knowledge. Analyst has just to determine the intrusion objectives to protect and labels when these objectives have been compromised in the observation history. Moreover, our approach implicitly filters the false alarms. Namely, every alert does not exceed the threshold will be considered as false alarm.

## 6 Conclusion

In this paper, we proposed a new alert correlation method based on naive bayes. Our approach used intrusion detection histories to build a naive bayes for each observed intrusion objective. During detection step, each observed action will provide an evidence that updates each naive bayes. According to the influence degree of this attack, probability of each intrusion objective will change positively or negatively.

Our approach has the advantage to make the prediction more easily thanks to simplicity and efficiency of naive bayes. It takes advantage of available data and only needs a small part of expert knowledge to determine the intrusion objectives. Moreover, attacks involved in the attack plan can be identified and false alarms are implicitly filtered by focussing on relevant actions.

Contrary to existing approach, attacks scenarios are not explicitly provided by experts but are learned automatically from observed data.

## References

- [1] ABOUZAKHAR N., GANI A., MANSON G., ABUITBEL M. and KING D., *Bayesian Learning Networks Approach to Cybercrime Detection*, proceedings of the 2003 PostGraduate Networking Conference, 2003.
- [2] AXELSSON S., *Combining a Bayesian Classifier with Visualisation: Understanding the IDS*, VizSEC/DMSEC-04, ACM, 99-108, 2004.
- [3] BENFERHAT S., AUTREL F. and CUPPENS F., *Enhanced Correlation in an Intrusion Detection Process*. Second International Workshop Mathematical Methods, Models and Architectures for Computer Networks Security, 157-170, St. Petersburg, Russia, September 20-24, 2003.
- [4] BEN AMOR N., BENFERHAT S. and ELOUEDI Z., *Naive Bayes vs decision trees in intrusion detection systems*. In Proceedings of the 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus, March 14-17, 420-424, 2004.
- [5] CUPPENS F., *Managing Alerts in a Multi-Intrusion Detection Environment*, In proceedings of Recent Advances in Intrusion Detection, 22-31, Davis, CA, USA, October 10-12, 2001.
- [6] CUPPENS F. and MIEGE A., *Alert correlation in a cooperative intrusion detection framework*. In Proceedings of the IEEE Symposium of Security and Privacy, 202-215, Berkeley, California 1215 May 2002.
- [7] DAIN and CUNNINGHAM, *Fusing a heterogeneous alert stream into scenario* , In Proceedings of the 2001 ACM Workshop on Data Mining for Security Application, 1-13, November 2001.
- [8] DANIEL J. B., Linda F. W. and GEORGE V. C., *Analysis of Distributed Intrusion Detection Systems Using Bayesian Methods*, 21th IEEE International Conference on Performance, Computing, and Communications, 329-334, 2002.
- [9] DARPA 2000, [http://www.ll.mit.edu/IST/ideval/data/data\\_index.html](http://www.ll.mit.edu/IST/ideval/data/data_index.html).
- [10] DEBAR H. and WESPI A., *Aggregation and Correlation of Intrusion-Detection Alerts*, In proceedings of Recent Advances in Intrusion Detection, 85-103, Davis, CA, USA, October 10-12, 2001.
- [11] DUDA R.O., HART P.E and STORK D.G. *Pattern Classification*, Hardcover, 2000.
- [12] FRIEDMAN N. and GOLDSZMIDT M., *Building classifiers using bayesiannetworks*, In Proceeding of American Association for Artificial Intelligence Conference (AAAI'96), Portland, Oregon, 1996.
- [13] GEIB C. and GOLDMAN R., *Plan Recognition in Intrusion Detection Systems*. In Proceeding of DARPA Information Survivability Conference and Exposition (DISCEX), Volume 1, 46-55, June 2001.
- [14] GOWDIA V., FARKAS C., VELTORTA M., *PAID: A Probabilistic Agent-Based Intrusion Detection system*, Computers & Security, Elsvier, volume 24, 529-545, 2005.
- [15] HEKERMANN D., *A Tutorial on Learning with Bayesian Networks*. Technical Report MSR-TR-95-06, Microsoft Corporation, 1995.
- [16] JENSEN F.V., *Introduction to Bayesian networks*, UCL Press, University college, London 1996.
- [17] JULISCH K. *Mining alarm clusters to improve alarm handling efficiency*, In Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC), 12-21, New Orleans, Louisiana, December 10-14, 2001.
- [18] LANGLEY P., IBA W. and THOMPSON K. *Decision making using probabilistic inference methods*, In Proceeding of the 18th National Information Security Conference, 194-204, 1995.
- [19] KANG D., FULLER D., HONAVAR V., *Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation*, In Proceedings of the 2005 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 118-125, 2005.

- [20] KOHAVI R., BEAKER B. and SOMMERFIELD D., *Improving simple Bayes*, In Proceedings of the European Conference on Machine Learning, Prague, Czech Republic, April 23-26, 1997.
- [21] KRUGEL C., MUTZ D., ROBERTSON W.K., and VALEUR F., *Bayesian Event Classification for Intrusion Detection*, In Proceeding of 19th Annual Computer Security Applications Conference (ACSAC 2003), Las Vegas, NV, USA , 14-23, 8-12 December 2003.
- [22] NING P. and CUI Y., *Analyzing intensive intrusion alerts via correlation*. In Processing of Recent Advances in Intrusion Detection, 74-94 Zurich, Switzerland, October 16-18, 2002.
- [23] PEARL J. *Probabilistic reasoning in intelligent systems: network of plausible inference*, Motgan Kaufmman, San Francisco (California), 1988.
- [24] QIN X. and LEE W., *Attack Plan Recognition and Prediction Using Causal Networks*, ACSAC-O4, 370-379, 2004.
- [25] RICARDO S. P., MARRAKCHI Z. and MÈ L., *A Bayesian Classification Model for Real-Time Intrusion Detection*, 22nd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering. AIP Conference Proceedings, Volume 659, pp. 150-162, 2003.
- [26] SCOTT L. S., *A Bayesian paradigm for designing intrusion detection systems*, Computational Statistics & Data Analysis, Elsevier, 69-83, 2004.
- [27] STEVEN J. T. and KARM L., *A requires/provides model for computer attacks*, In Proceedings of New Security Paradigms Workshop, 31-38, Cork, Ireland, September 19th - 21st, 2000.
- [28] VALDES A. and SKINNER K., *Probabilistic alert correlation*, In Recent Advances in Intrusion Detection, 54-68, Davis, CA, USA, October 10-12, 2001.