ALERT CORRELATION BASED ON A LOGICAL HANDLING OF ADMINISTRATOR PREFERENCES AND KNOWLEDGE

Salem Benferhat and Karima Sedki
CRIL-CNRS UMR-8188, Universit d'Artois, Rue Jean Souvraz SP 18 62307 Lens, Cedex, France

Keywords: Alert correlation, preferences logic, administrator's preferences and knowledge.

Abstract: Intrusion detection systems (*IDSs*) are important tools for infortation systems security. However, they generate a large number of alerts which complicate the task of network administrator to understand these triggered alerts and take appropriate actions. In this paper, we present a logic-based approach to alert correlation. This logic allows to integrate administrator's preferences and knowledge. Our logic, called Extended Qualitative Choice Logic ($\mathcal{EQCL}$), is an extension of a fragment of first order logic. It adds a new connector, denoted $\vec{\times}$, that allows to represent administrator preferences. The objective of our logic-based alert correlation approach is to rank-order alerts generated by IDS on the basis of administrator preferences and knowledge. Only alerts that fully fit administrator's preferences and knowledge are first presented. Then if needed, less preferred alerts (which falsify less important preferences) will be presented, and so on.

1 INTRODUCTION

Intrusion detection system (IDS) is an active area of research for almost thirteen years (Anderson, 1980). Actually, there are two main intrusion detection approaches used to detect attacks on the network. Anomaly detection (Lunt, 1990), (Porras and Neumann, 1997) is based on defining a profile representing normal activities. Any action or behavior that deviates from the normal profile is considered as anomaly or possible intrusion. This approach has the advantage of being able to detect unknown attacks. However, it generates a large volume of false alerts. Misuse detection (Kumar and Spafford, 1995), (Paxson, 1999)focuses on the characteristics or signatures of known attacks that exploit weaknesses in system and application software. They have the advantage that they achieve very high detection rates on known attacks. However, they are not able to detect new attacks that do not follow predefined patterns. Clearly, whatever the used approach, an administrator is confronted with a large amount of alerts produced by IDS. In this context, several alert correlation approaches have been proposed in recent years to address this problem. Alert correlation tools are important for reducing the large volume of alerts that are generated by IDSs. Generally, they can be classified into three categories: approaches based on similarity between alert attributes (Valdes and Skinner, 2001), (Julisch, 2003) , alert correlation based on known attacks scenarios (Eckmann et al., 2000), (Morin et al., 2002)and approaches that are based on prerequisites and consequences relationship (Cuppens and Miège, 2002), (Ning et al., 2002).

Even if existing alerts correlation approaches are quite efficient to remove redundant information or to detect complex coordinated attacks, they still produce a large number of alerts. Moreover to the best of our knowledge, none of existing alert correlation tools take into account administrator preferences and knowledge. The alert correlation proposed in this paper follows another direction of the current correlation techniques. It is also complementary to existing alert correlation approaches. It consists in modeling administrator knowledge (on system, network, IDS) and his preferences. The main idea is to develop a logic that ranks-order alerts and classifies them in different categories that fit more or less administrator knowledge or preferences. Alerts in the first category are those that should be first presented to administrator. And if needed alerts of the second category will be presented, and so on. The rest of this paper is organized as follows: Section 2 describes our new logic and Section 3 presents our alert correlation approach. Section 4 concludes the paper.

Benferhat S. and Sedki K. (2008).
ALERT CORRELATION BASED ON A LOGICAL HANDLING OF ADMINISTRATOR PREFERENCES AND KNOWLEDGE.
In *Proceedings of the International Conference on Security and Cryptography*, pages 50-56
DOI: 10.5220/0001924000500056

2 A NEW LOGIC FOR HANDLING PREFERENCES

We present our new logic to be used to represent administrator preferences. The starting point is an existing logic called *Qualitative Choice Logic (QCL)* (Brewka et al., 2004). It is an extension of propositional logic. The non-standard part of *QCL* logic is a new logical connective $\vec{\times}$, called *ordered disjunction*. Intuitively, if A and B are propositional formulas then A $\vec{\times}$ B means: "if possible A, but if A is impossible then at least B". *QCL* logic can be very useful for representing preferences. However, its inference relation is less satisfactory. This is due on the one hand to the way rules are handled, where ordered disjunction is lost when preferences are associated with negation (see (Benferhat and Sedki, 2007) for more details). On the other hand propositional logic is not appropriate to express complex knowledge. It cannot express generic knowledge that involve variables. The extensions of *QCL* proposed in (Benferhat and Sedki, 2007) are not satisfactory for our application, in particular they are defined within propositional language. What is needed is a richer language to express general pieces of information. This is the aim of the following subsections.

2.1 $\mathcal{EQCL}$ Language

$\mathcal{EQCL}$ language is composed of three encapsulated sub-languages: Propositional Logic Language, the set of Basic Choice Formulas (*BCF*) and the set of General Choice Formulas (*GCF*).

1. **Basic Choice Formulas (*BCF*):** Let *PS* denotes a set of propositional symbols and $PROP_{PS}$ denotes the set of propositional formulas. *BCF* formulas are ordered disjunctions of propositional formulas. The language composed of *BCF* formulas is denoted by BCF_{PS}, is defined as follow:
   (a) If $\phi \in PROP_{PS}$ then $\phi \in BCF_{PS}$
   (b) If $\phi, \psi \in BCF_{PS}$ then $(\phi \vec{\times} \psi) \in BCF_{PS}$
   (c) Every *basic choice formula* is only obtained by applying the two rules above a finite number of times.
2. **General Choice Formulas (*GCF*):** *GCF* formulas represent any formula that can be obtained from *PS* using connectors $\vec{\times}, \wedge, \vee, \neg$. The language composed of *GCF* formulas, denoted by GCF_{PS}, is defined inductively as follows:
   (a) If $\phi \in BCF_{PS}$ then $\phi \in GCF_{PS}$
   (b) If $\phi, \psi \in GCF_{PS}$ then $(\phi \wedge \psi), \neg(\psi), (\phi \vee \psi), (\phi \vec{\times} \psi) \in GCF_{PS}$.
   (c) The language of GCF_{PS} is only obtained by applying the two rules above a finite number of times.

Example 1. *The formula "Bro-Alerts $\vec{\times}$ Snort-Alerts" is a BCF formula, while the formula "(Bufferoverflow-Alerts$\vec{\times}$ Ping-Alerts) $\wedge$ (Scan-Alerts$\vec{\times}$ Badtraffic-Alerts)" is a GCF formula.*

2.2 Inference Relation of Basic Choice Formulas

The semantics of $\mathcal{EQCL}$ formulas is based on the degree of satisfaction of a formula in a particular model *I*. An interpretation *I* is an assignment of the classical truth values T,F to the atoms in *PS*. Inference relation of *BCF* formulas is given in the following definition:

Definition 1. *1. Let $\phi = a_1 \vec{\times} a_2 \vec{\times} \ldots \vec{\times} a_n \in BCF_{PS}$, $I \models_k \phi$ iff $I \models a_1 \vee a_2 \vee \ldots \vee a_n$ and $k = min\{j \mid I \models a_j\}$.*

2. Let $\phi \in PROP_{PS}$. $I \models_1 \phi$ iff $I \models \phi$.

2.3 Normalization Form and Inference from General Choice Formulas

This section proposes our inference relation for *GCF* formulas which departs from (and overcomes limitations of) the one defined in (Brewka et al., 2004). Inference relation of *GCF* simply reuses Definition 1 after a normalization step which transforms a *GCF* formula into a *BCF* formula. Namely, first, an equivalent *BCF* formula of each *GCF* formula is provided, and then Definition 1 applied. The following introduces the notion of normal form function denoted by $\mathcal{N}_{\mathcal{EQCL}}$ which associates with each *GCF* formulas (complex form of preferences), its corresponding *BCF* formulas (simple form of preferences).

Definition 2. *We define a normal function denoted by $\mathcal{N}_{\mathcal{EQCL}}$, a function from $GCF_{PS} \rightarrow BCF_{PS}$, such that:*

1. *Normal form of basic choice formulas and propositional formulas are these formulas themselves: $\forall\, \phi \in BCF_{PS}, \mathcal{N}_{\mathcal{EQCL}}(\phi) = \phi$.*
2. *The normal form with respect to negated formulas: $\forall\, \phi \in GCF_{PS}$, and $(\phi \notin BCF_{PS})$, $\mathcal{N}_{\mathcal{EQCL}}(\neg\phi) \equiv \mathcal{N}_{\mathcal{EQCL}}(\neg\mathcal{N}_{\mathcal{EQCL}}(\phi))$.*
3. *The normal form is decomposable with respect to conjunction, disjunction and ordered disjunction of GCF formulas:*
   (a) $\forall\ \phi, \psi \in GCF_{PS}$ and $(\phi \notin BCF_{PS}$ or $\psi \notin BCF_{PS})$, $\mathcal{N}_{\mathcal{EQCL}}(\phi \wedge \psi) \equiv \mathcal{N}_{\mathcal{EQCL}}(\mathcal{N}_{\mathcal{EQCL}}(\phi) \wedge \mathcal{N}_{\mathcal{PQCL}}(\psi))$.
   (b) $\forall\ \phi, \psi \in GCF_{PS}$ and $(\phi \notin BCF_{PS}$ or $\psi \notin BCF_{PS})$, $\mathcal{N}_{\mathcal{EQCL}}(\phi \vee \psi) \equiv \mathcal{N}_{\mathcal{EQCL}}(\mathcal{N}_{\mathcal{EQCL}}(\phi) \vee \mathcal{N}_{\mathcal{EQCL}}(\psi))$.
   (c) $\forall\ \phi, \psi \in GCF_{PS}$ and $(\phi \notin BCF_{PS}$ or $\psi \notin BCF_{PS})$, $\mathcal{N}_{\mathcal{EQCL}}(\phi \vec{\times} \psi) \equiv \mathcal{N}_{\mathcal{EQCL}}(\mathcal{N}_{\mathcal{PQCL}}(\phi) \vec{\times} \mathcal{N}_{\mathcal{EQCL}}(\psi))$.

4. *Normal form of negated, conjunction and disjunction of BCF formulas are: Let* $\phi = a_1 \vec{\times} a_2 \vec{\times} \ldots \vec{\times} a_n$, *and* $\psi = b_1 \vec{\times} b_2 \vec{\times} \ldots \vec{\times} b_m$ *such that* a_i*'s and* b_i*'s are propositional formulas,*
   (a) $\mathcal{N}_{\mathcal{EQCL}}((a_1 \vec{\times} a_2 \vec{\times} \ldots \vec{\times} a_n) \wedge (b_1 \vec{\times} b_2 \vec{\times} \ldots \vec{\times} b_m)) \equiv c_1 \vec{\times} c_2 \vec{\times} \ldots \vec{\times} c_k$, *where k = max(m, n), and*
   $$c_i = \begin{cases} [(a_1 \vee \ldots \vee a_i) \wedge b_i] \vee [a_i \wedge (b_1 \vee \ldots \vee b_i)] \\ \quad if \quad i \leq min(m,n) \\ ((a_1 \vee \ldots \vee a_n) \wedge b_i) \quad if \quad n < i \leq m \\ (a_i \wedge (b_1 \vee \ldots \vee b_m)) \quad if \quad m < i \leq n \end{cases}$$
   (b) $\mathcal{N}_{\mathcal{EQCL}}(\phi \vee \psi) \equiv c_1 \vec{\times} c_2 \vec{\times} \ldots \vec{\times} c_k$ *where k = max(m, n), and*
   $$c_i = \begin{cases} (a_i \vee b_i) & if \quad i \leq min(m,n) \\ a_i & if \quad m \leq i \leq n \\ b_i & if \quad n \leq i \leq m \end{cases}$$
   (c) $\mathcal{N}_{\mathcal{EQCL}} \neg(\quad a_1 \vec{\times} \quad a_2 \vec{\times} \ldots \vec{\times} \quad a_n) \equiv \neg\, a_1 \vec{\times} \neg\, a_2 \vec{\times} \ldots \vec{\times} \neg\, a_n$.

Example 2. *Assume that the network administrator wants to analyze R2L alerts that are preferred to Probe ones and also U2R alerts that are preferred to Dos ones. These preferences are represented by the following GCF formula* ψ *= (R2L-Alerts* $\vec{\times}$ *Probe-Alerts)* $\wedge$ *(U2R-Alerts* $\vec{\times}$ *DoS-Alerts).*
To give the satisfaction degree with respect to a given model of ψ*, we first normalize the formula* ψ *(namely, we transform* ψ *in the equivalent BCF formula) as indicated in Definition 2, then we use Definition 1 to give the inference relation of obtained BCF formula.*

1. *Normalization of* ψ*: Using item 4-(a) of Definition 2, we obtain:*
   $\mathcal{N}_{\mathcal{EQCL}}(\psi)$=$\mathcal{N}_{\mathcal{EQCL}}$*((R2L-Alerts* $\vec{\times}$ *Probe−Alerts)* $\wedge$*(U2R-Alerts*$\vec{\times}$*DoS-Alerts))*
   $\equiv$ *((R2L-Alerts* $\wedge$ *U2R-Alerts)* $\vee$ *(R2L-Alerts* $\wedge$ *U2R-Alerts))* $\vec{\times}$*((R2L-Alerts* $\vee$ *Probe-Alerts)*$\wedge$ *DoS-Alerts)* $\vee$*(Probe-Alerts* $\wedge$ *(U2R-Alerts* $\vee$ *DoS-Alerts)).*
   $\equiv$ *(R2L-Alerts* $\wedge$ *U2R-Alerts)*$\vec{\times}$ *((R2L-Alerts* $\wedge$ *DoS-Alerts)* $\vee$ *(Probe-Alerts* $\wedge$ *DoS-Alerts)* $\vee$*(Probe-Alerts*$\wedge$ *U2R-Alerts)).*

2. *Satisfaction degree of* ψ*:*
   - *Let* I_1 *={Probe-alerts, DoS-Alerts}. Using item 2 of Definition 1, we have* $I_1 \not\models$*R2L-Alerts*$\wedge$*U2R-Alerts,* $I_1 \not\models$*R2L-Alerts*$\wedge$*DoS-Alerts,* $I_1 \not\models$*Probe-Alerts*$\wedge$*U2R-Alerts, and* $I_1 \models$*Probe-Alerts*$\wedge$*DoS-Alerts, so* $I_1 \models$ *(R2L-Alerts* $\wedge$ *DoS-Alerts)*$\vee$*(Probe-Alerts* $\wedge$ *DoS-Alerts)*$\vee$*(Probe-Alerts* $\wedge$ *U2R-Alerts). Using item 1 of Definition 1, we have* $I_1 \models_2 \mathcal{N}_{\mathcal{EQCL}}(\psi)$*. We conclude that* I_1 *satisfies* ψ *to degree 2.*
   - *If* I_2 *={DoS-Alerts}, then* $I_2 \not\models \mathcal{N}_{\mathcal{EQCL}}(\psi)$*.* I_2 *does not satisfy* ψ*.*
   - *If* I_3 *={R2L-Alerts, U2R-Alerts}, then* $I_3 \models_1 \mathcal{N}_{\mathcal{EQCL}}(\psi)$*. We conclude that* I_3 *satisfies* ψ *to degree 1. Namely,* ψ *is fully satisfactory in* I_3*.*

The following defines preferred models when we have a set of knowledge and a set of preference. Let K be a set of propositional formulas which represents knowledge, and let T be a set of preferences that contains only simple form preferences (*BCF* formulas). We suppose that all complex form preferences (*GCF* formulas) are first transformed into simple form preferences, using Definition 2.

Definition 3. *Let* $M^k(T)$ *denote the subset of basic choice formulas of T satisfied by a model M to a degree k (using Definition 1). A model* M_1 *is* $\{K \cup T\}$*-preferred over a model* M_2 *if there is k such that* $\mid M_1^k(T)\mid > \mid M_2^k(T)\mid$ *and for all* $j < k$*:* $\mid M_1^j(T)\mid = \mid M_2^j(T)\mid$*. M is a preferred model of* $\{K \cup T\}$ *iff :*

1. *M is model of K,*
2. *M is maximally* $\{K \cup T\}$*-preferred.*

2.4 Universally Quantified First Order $\mathcal{EQCL}$

Propositional logic is not appropriate to express complex knowledge. It can only deal with pieces of information regarding particular situations or properties, and cannot express knowledge that involve variables. So, what is needed is a richer language to express general information. In the following, we slightly extend our logic to a fragment of universally quantified first order formulas that do not involve function symbols. For the purpose of our application there is no need to consider the full first order logic. More precisely, let $\mathcal{X} = \{x, y, z, ...\}$ be a set of variables. Let $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ be a set of constants. We define a term as either a constant of $\mathcal{C}$ or a variable of $\mathcal{X}$. Let us denote by $\mathcal{P} = \{p, q, r, ...\}$ a set of predicate symbols. The following defines the notion of unquantified first order $\mathcal{EQCL}$ formulas:

Definition 4. *Unquantified first order* $\mathcal{EQCL}$ *formulas are defined as follows:*

1. *If p is a predicate symbol of arity n, and* $t_1, ..., t_n$ *are terms, then* $p(t_1, ..., t_n)$ *is an unquantified first order* $\mathcal{EQCL}$ *formula.*
2. *If* ϕ *and* ψ *are two unquantified first order* $\mathcal{EQCL}$ *formulas, then* $\phi \wedge \psi$*,* $\phi \vee \psi$*,* $\neg\phi$*,* $\phi \vec{\times} \psi$ *are unquantified first order* $\mathcal{EQCL}$ *formulas.*

In this paper, for sake of simplicity, we only restrict ourselves to universally quantified first order $\mathcal{EQCL}$ formulas given by the following definition:

Definition 5. *Universally quantified first order* $\mathcal{EQCL}$ *formulas are obtained by:*

- *If ϕ is an unquantified first order QCL formula, and $\{x_1,..., x_n\}$ are the set of variables used in ϕ, then $\forall$ x_1,..., $\forall$ x_n ϕ is a universally quantified first order $\mathcal{EQCL}$ formula.*

Example 3. *Assume that the network administrator prefers alerts that are issued by Bro IDS to those issued by Snort IDS. We use the universally quantified $\mathcal{EQCL}$ formula ψ to define this preference: $\psi = \forall x$, $\forall y\ Alert\text{-}Bro(x)\ \overrightarrow{\times} Alert\text{-}snort(y)$.*

Universally quantified first order $\mathcal{EQCL}$ language offers flexibility for expressing knowledge and preferences. However, from reasoning point of view, it is better to work on propositional level in order to exploit existing inference tools. Namely, it is important to instantiate first order knowledge's bases to propositional ones. The instantiation steps are:
1. Let K and T be two sets of universally quantified first order $\mathcal{EQCL}$ formulas, where K does not involve the ordered disjunction symbol $\overrightarrow{\times}$.
2. Let $\mathcal{E}(K,T)$ be the set of constant symbols used in K and T.
3. Let $\psi = \forall$ x_1,..., $\forall$ x_n ϕ be a universally quantified first order $\mathcal{EQCL}$ formula. Define $\mathcal{D}(\psi)$ as the domain associated with ψ. $\mathcal{D}(\psi)$ is a subset of $\mathcal{E}(K,T)^n$ composed of feasible n-uplets of $\mathcal{E}(K,T)^n$. It is either set by an expert, or initialized by default to $\mathcal{E}(K,T)^n$.
4. Define $Instantiation(\psi)$ as the set of all grounded $\mathcal{EQCL}$ formulas obtained by replacing (x_1,..., x_n) in ψ respectively by an element (c_1,..., c_n) of $\mathcal{D}(\psi)$.
5. Define $Inst(K)$ (resp. $Inst(T)$) as the result of instantiating each formula of K (resp. of T).
The inference relation for $\mathcal{EQCL}$ formulas is given by the following steps:
* Apply Definition 2 to transform universally quantified first order *GCF* formulas into universally quantified first order *BCF* formulas.
* Apply the instantiation steps for each formula as indicated above for K and T.
* Apply Definition 1 (item 1) for all obtained *BCF* formulas or (item 2) for propositional ones.
* Compute preferred models of $\{ K \cup T\}$ using Definition 3.

3 APPLICATION OF $\mathcal{EQCL}$ TO ALERT CORRELATION

3.1 Description of Inputs

The inputs of our model are:

1. **A Group of Alerts *G* Produced by IDSs:** Each alert is characterized by a set of attributes called "basic attributes". Examples of basic attributes are: Signature Identifier (SID), messages associated with alerts, Protocol, TTL (Time To Live), etc. Each attribute will be represented by a predicate symbol. The sets of predicate facts containing values of alert attributes of G will be represented by K_1.

   Example 4. *Assume that G contains one alert identified by $id1$. Assume that the attributes concerning this alert are: IDS identity is Snort, the used protocol is TCP and the class of attack is DoS. These facts will be represented by $K_1 = \{IDS(id1,Snort),\ Protocol(id1,TCP),\ Class(id1,DoS)\}$. Note that in general, some attributes may not be informed (known) by an IDS.*

   Types of Facts: We distinguish two kinds of facts:

   (a) **Alerts Facts:** These facts are directly defined on basic attributes of alerts. $Protocol(A_1,TCP)$ is an example of alert fact which indicates that the attribute protocol of alert A_1 is TCP.

   (b) **Other Facts:** These facts concern attributes that are not known by the IDS from which the alerts are issued. *Direction* of alert is an example of this kind of facts. It is based on source and target IP addresses. These information allow to know the direction of concerned alerts on the system (inbound, outbound, inside).

2. **Knowledge of the Network Administrator:** Administrator network can provide some knowledge or beliefs on networks, on system, etc. This knowledge base is denoted by K_2, it contains a set of universally quantified propositional formulas (namely, formulas that do not involve $\overrightarrow{\times}$).

3. **Preferences of the Network Administrator:** The network administrator can express his preferences according to what he wants to first analyze and what he would like to ignore. This will be represented with a set of $\mathcal{EQCL}$ formulas T.

3.2 Output of our Model

The output of our model is a subset $G' \subseteq G$. More precisely, the subset of alerts in G to be first presented to the network administrator. The objective of our alert correlation, is to first present only alerts that satisfy knowledge and preferences of the network administrator, namely, we only present preferred alerts. Then if needed second preferred alerts will be presented, etc. So, we need to preprocess available alerts and encode them in our logical framework. Namely, we need to:
- Extract the set of facts K_1 from the given alerts.

Table 1: Group of inputs alerts.

Alerts	Timestamp	Sid	Message	Protocol	IPsrc	Portsrc	IPdst	Portdst	TTL
A_1	3/8/16/39/12.024	1156	WEB-MISC	TCP	199.174.194.16	1028	172.16.114.50	80	62
A_2	3/8/16/39/12.0267	1156	WEB-MISC	TCP	199.174.194.16	1028	172.16.114.50	80	62
A_3	3/8/16/39/12.030	1156	WEB-MISC	TCP	199.174.194.16	1028	172.16.114.50	80	62
A_4	3/11/17/50/16.384	469	ICMP PING	ICMP	207.103.80.104	8 (Type)	172.16.114.50	0 (Code)	63
A_5	4/1/02/19/43.008	1893	SNMP missing	UDP	172.16.0.1	1336	207.181.92.211	161	63
A_6	3/8/15/01/13.344	1648	*WEB-CGI* perl.exe	TCP	206.48.44.18	1061	172.16.112.100	80	127

- Represent each preference of the administrator as a universally quantified first order $\mathcal{EQCL}$ formula. This will be represented by T.
- Represent each knowledge of the administrator as a universally quantified propositional formula. This will be represented by K_2. Then,
- Use $\mathcal{N}_{\mathcal{EQCL}}$ to normalize the *GCF* formulas.
- Apply the steps of instantiation according to the all constants in $\mathcal{E}(K_1 \cup K_2, T)$.
- Select only instantiated formulas that concern considered facts.
- Provide the inference relation (satisfaction degree) of the instantiated formulas.
- Determine preferred models. Finally,
- Define G' as a set of alerts which are true in preferred models.

3.3 A Detailed Example

Assume that the input is a set of alerts given in Table 1. This input contains 6 alerts given by Snort IDS on some traffic data. Note that in general, some attributes may not be informed by the IDS. For instance, Snort does not provide the attribute *Direction* (which is derived). Because of the important number of alerts and presence of false alerts, it is not useful to analyze all alerts. So, we assume that, a network administrator provides the following set of knowledge and preferences to select and choose preferred ones:

1. **Administrator's Preferences:**
   - The network administrator prefers alerts which are identified with *TCP* protocol than alerts identified with *UDP* protocol than those identified with *ICMP* protocol. Namely, if x,y,z are three alerts, and if the protocol of x (resp. y,z) is TCP (resp. UDP, ICMP), then it is preferred to first present x, then y and lastly z.
   - He prefers inbound alerts than outbound ones.
2. **Administrator's Knowledge:**
   - He wants to ignore all *ICMP* alerts.
   - He wants to ignore redundant alerts.

The predicate symbols of our language are associated with alerts attributes Timestamp, Sid, Portdst,..., and:
* $Differ(A_i,A_j)$ which means that alerts A_i and A_j have different identities.
* $Present\text{-}alert(id)$ when it is true, means that the alert should be presented to the administrator.

1. **Extract the Fact Base K_1:**
   - Alert facts are directly extracted from the given alerts (see Table 1). $Protocol(A_1,TCP)$ is an example of alerts fact.
   - Other facts that we may defined: The fact that concerns direction of alerts. In this example, inbound alerts are alerts where the source IP address is different to ”172.16.x.y” and the destination IP address in equal to ”172.16.x.y”. Outbound alerts are alerts where the source IP address is equal to ”172.16.x.y” and the destination IP address is different to ”172.16.x.y”. The facts *SameIPsrc*, *SameIPdst*, *SamePortsrc*, *SamePortdst*, *SameSid*, *Timestamp-samaller-than* are corresponding respectively to alerts that have the same attributes (Sid, protocol, source and target IP addresses, source and destination ports, smallest timestamp). These facts are applied on alerts A_1, A_2, A_3. A_2 and A_3 are repetitive alerts. We only specify facts that are relevant to knowledge and preferences of the network administrator:

$$K_1 = \begin{cases} Protocol(A_1,TCP), Protocol(A_2,TCP), Protocol(A_3,TCP) \\ Protocol(A_4,ICMP), \\ Protocol(A_5,UDP), Protocol(A_6,TCP), Direction(A_1 inbound) \\ Direction(A_2,inbound), Direction(A_3,inbound) \\ Direction(A_4,inbound) \\ Direction(A_5,outbound), Direction(A_6 inbound) \\ SameIPsrc(A_1,A_2), SameIPsrc(A_1,A_3), SameIPsrc(A_2,A_3) \\ SameIPdst(A_1,A_2) \\ SameIPdst(A_1,A_3), SameIPdst(A_2,A_3), SamePortsrc(A_1,A_2) \\ SamePortsrc(A_1,A_3) \\ SamePortsrc(A_2,A_3), SamePortdst(A_1,A_2), SamePortdst(A_1,A_3 \\ SamePortdst(A_2,A_3) \\ SameSid(A_1,A_2), SameSid(A_1,A_3) \\ Timestamp\text{-}Smaller\text{-}than(A_1,A_3) \\ Timestamp\text{-}Smaller\text{-}than(A_2,A_3), SameSid(A_2,A_3) \\ Timestamp\text{-}Smaller\text{-}than(A_1,A_2) \end{cases}$$

2. **Formalize the Preferences Base T:**
   Administrator's preferences given above are formalized respectively by ϕ_1 and ϕ_2 as follows:
   - $\phi_1 = \forall x, \forall y, \forall z\ Protocol(x,TCP) \wedge Protocol(y,UDP) \wedge Protocol(z,ICMP) \wedge Differ(x,y,z) \rightarrow Present\text{-}alert(x) \vec{\times} Present\text{-}alert(y) \vec{\times} Present\text{-}alert(z)$.

- $\phi_2 = \forall x, \forall y \; \mathit{Direction}(x, \mathit{inbound}) \wedge \mathit{direction}(y, \mathit{outbound}) \wedge \mathit{Differ}(x,y) \rightarrow \mathit{Present\text{-}alert}(x) \vec{\times} \mathit{Present\text{-}alert}(y)$.

3. **Formalize the Knowledge Base K_2:**

- $\phi_3 = \forall x \; \mathit{Protocol}(x, \mathit{ICMP}) \rightarrow \neg \; \mathit{Present\text{-}alert}(x)$.
- $\phi_4 = \forall x, \forall y \; \mathit{SameIPsrc}(x,y) \wedge \mathit{SameIPdst}(x,y) \wedge \mathit{SamePortsrc}(x,y) \wedge \mathit{SamePortdst}(x,y) \wedge \mathit{SameSid}(x,y) \wedge \mathit{Timestamp\text{-}Smaller\text{-}than}(x,y) \wedge \mathit{Differ}(x,y) \rightarrow \mathit{Present\text{-}Alert}(x) \wedge \neg \mathit{Present\text{-}Alert}(y)$.

4. **Define $\mathit{Inst}(T)$ and $\mathit{Inst}(K_2)$:** To compute $\mathit{Inst}(T)$ and $\mathit{Inst}(K_2)$, a direct way is to consider all different possibilities regarding different values of alert identities. For instance, for ϕ_1, we need to consider all pairs (x, y, z) from $\{A_1,A_2,A_3,A_4,A_5,A_6\}^3$. However, with the help of K_1, some instantiations are simply impossible. For instance, in ϕ_1, there is no need to consider (A_1,A_4, A_5) since this will lead: $\mathit{Protocol}(A_4, \mathit{TCP}) \wedge \mathit{Protocol}(A_5, \mathit{UDP}) \wedge \mathit{Protocol}(A_1, \mathit{ICMP}) \rightarrow \mathit{Present\text{-}alert}(A_4) \vec{\times} \mathit{Present\text{-}alert}(A_5) \vec{\times} \mathit{Present\text{-}alert}(A_1)$.
   Since together with $\mathit{Protocol}(A_1, \mathit{ICMP}) \in K_2$, this rule will never be applied, and can be removed. Given this observation, $\mathit{Inst}(K_2)$ and $\mathit{Inst}(T)$ are given in the following:

$$\mathit{Inst}(K_2) = \begin{cases} \psi_1 = \mathit{Protocol}(A_4, (\mathit{ICMP}) \rightarrow \neg \mathit{Present\text{-}alert}(A_4) \\ \psi_2 = \mathit{SameIPsrc}(A_1,A_2) \wedge \mathit{SameIPdst}(A_1,A_2) \\ \wedge \mathit{SamePortsrc}(A_1,A_2) \wedge \mathit{SamePortdst}(A_1,A_2) \\ \wedge \mathit{SameSid}(A_1,A_2) \wedge \mathit{Timestamp\text{-}Smaller\text{-}than}(A_1,A_2) \\ \rightarrow \mathit{Present\text{-}alert}(A_1) \wedge \neg \mathit{Present\text{-}alert}(A_2) \\ \psi_3 = \mathit{SameIPsrc}(A_1,A_3) \wedge \mathit{SameIPdst}(A_1,A_3) \\ \wedge \mathit{SamePortsrc}(A_1,A_3) \wedge \mathit{SamePortdst}(A_1,A_3) \\ \wedge \mathit{SameSid}(A_1,A_3) \wedge \mathit{Timestamp\text{-}Smaller\text{-}than}(A_1,A_3) \\ \rightarrow \mathit{Present\text{-}alert}(A_1) \wedge \neg \mathit{Present\text{-}alert}(A_3) \\ \psi_4 = \mathit{SameIPsrc}(A_2,A_3) \wedge \mathit{SameIPdst}(A_2,A_3) \\ \wedge \mathit{SamePortsrc}(A_2,A_3) \wedge \mathit{SamePortdst}(A_2,A_3) \\ \wedge \mathit{SameSid}(A_2,A_3) \wedge \mathit{Timestamp\text{-}Smaller\text{-}than}(A_2,A_3) \\ \rightarrow \mathit{Present\text{-}alert}(A_2) \wedge \neg \mathit{Present\text{-}alert}(A_3). \end{cases} .$$

To compute the set of preferred models of $\{K_2 \cup T\}$, namely to give preferred alerts, we provide the satisfaction degree of all formulas in $\mathit{Inst}(K_2)$ and $\mathit{Inst}(T)$, namely formulas ψ_1, ψ_2, ψ_3, ψ_4, ψ_5, ψ_6, ψ_7, ψ_8, ψ_9, ψ_{10}, ψ_{11}, ψ_{12}, ψ_{13} for each interpretation. For lack of space, we do not give the table of preferred models. One can check that all preferred models contain $\{\mathit{Present\text{-}alert}(A_1), \mathit{Present\text{-}alert}(A_6)\}$. Hence G' = $\{A_1, A_6\}$ and this will be first presented to the administrator.

and $\mathit{Inst}(T)$ =

$$\begin{cases} \psi_5 = \mathit{Protocol}(A_1, \mathit{TCP}) \wedge \mathit{Protocol}(A_5, \mathit{UDP}) \wedge \mathit{Protocol}(A_4, \mathit{ICMP}) \\ \rightarrow \mathit{Present\text{-}alert}(A_1) \vec{\times} \mathit{Present\text{-}alert}(A_5) \vec{\times} \mathit{Present\text{-}alert}(A_4) \\ \psi_6 = \mathit{Protocol}(A_2, \mathit{TCP}) \wedge \mathit{Protocol}(A_5, \mathit{UDP}) \wedge \mathit{Protocol}(A_4, \mathit{ICMP}) \\ \rightarrow \mathit{Present\text{-}alert}(A_2) \vec{\times} \mathit{Present\text{-}alert}(A_5) \vec{\times} \mathit{Present\text{-}alert}(A_4) \\ \psi_7 = \mathit{Protocol}(A_3, \mathit{TCP}) \wedge \mathit{Protocol}(A_5, \mathit{UDP}) \wedge \mathit{Protocol}(A_4, \mathit{ICMP}) \\ \rightarrow \mathit{Present\text{-}alert}(A_3) \vec{\times} \mathit{Present\text{-}alert}(A_5) \vec{\times} \mathit{Present\text{-}alert}(A_4) \\ \psi_8 = \mathit{Protocol}(A_6, \mathit{TCP}) \wedge \mathit{Protocol}(A_5, \mathit{UDP}) \wedge \mathit{Protocol}(A_4, \mathit{ICMP}) \\ \rightarrow \mathit{Present\text{-}alert}(A_6) \vec{\times} \mathit{Present\text{-}alert}(A_5) \vec{\times} \mathit{Present\text{-}alert}(A_4) \\ \psi_9 = \mathit{Direction}(A_1, \mathit{inbound}) \wedge \mathit{Direction}(A_5, \mathit{outbound}) \\ \rightarrow \mathit{Present\text{-}alert}(A_1) \vec{\times} \mathit{Present\text{-}alert}(A_5) \\ \psi_{10} = \mathit{Direction}(A_2, \mathit{inbound}) \wedge \mathit{Direction}(A_5, \mathit{outbound}) \\ \rightarrow \mathit{Present\text{-}alert}(A_2) \vec{\times} \mathit{Present\text{-}alert}(A_5) \\ \psi_{11} = \mathit{Direction}(A_3, \mathit{inbound}) \wedge \mathit{Direction}(A_5, \mathit{outbound}) \\ \rightarrow \mathit{Present\text{-}alert}(A_3) \vec{\times} \mathit{Present\text{-}alert}(A_5) \\ \psi_{12} = \mathit{Direction}(A_4, \mathit{inbound}) \wedge \mathit{Direction}(A_5, \mathit{outbound}) \\ \rightarrow \mathit{Present\text{-}alert}(A_4) \vec{\times} \mathit{Present\text{-}alert}(A_5) \\ \psi_{13} = \mathit{Direction}(A_6, \mathit{inbound}) \wedge \mathit{Direction}(A_5, \mathit{outbound}) \\ \rightarrow \mathit{Present\text{-}alert}(A_6) \vec{\times} \mathit{Present\text{-}alert}(A_5) \end{cases}$$

4 CONCLUSIONS

In this paper, we presented our alert correlation approach which follows another direction of existing correlation techniques. It is complementary to existing alert correlation approaches. It consists in modeling administrator knowledge (on system, network, IDS) and its preferences in order to classify alerts, discard information that are less important and present alerts that fit administrator preferences. A future work is to enrich knowledge bases and preferences in our experimental studies. A first experimental study on real network traffic shows that reduction of more than 60 % of generated alerts can be performed.

ACKNOWLEDGEMENTS

This works is supported by the SETIN06 project PLACID (Probabilistic graphical models and Logics for Alarm Correlation in Intrusion Detection).

REFERENCES

Anderson, J. (1980). *Computer security threat monitoring and surveillance*. Fort Washington, Pennsylvania.

Benferhat, S. and Sedki, K. (2007). A revised qualitative choice logic for handling prioritized preferences. In *ECSQARU*, pages 635–647.

Brewka, G., Benferhat, S., and Berre, D. L. (2004). Qualitative choice logic. *Artificial Intelligence Journal(AIJ)*, 157(1-2):203–237.

Cuppens, F. and Miège, A. (2002). Alert correlation in a cooperative intrusion detection framework. In *SP*, USA.

Eckmann, S., Vigna, G., and Kemmerer, R. (2000). Statl: An attack language for state-based intrusion detection.

Julisch, K. (2003). Clustering intrusion detection alarms to support root cause analysis.

Kumar, S. and Spafford, E. (1995). A software architecture to support misuse intrusion detection. In *Proceedings of the 18th National Information Security Conference*.

Lunt, T. (1990). Detecting Intruders in Computer Systems. In *In Proc of the Sixth Annual Symposium and Technical Displays on Physical and Electronic Security*.

Morin, B., Ludovic, M., Debar, H., and Ducassé, M. (2002). M2d2: A formal data model for ids alert correlation.

Ning, P., Cui, Y., and Reeves, D. S. (2002). Constructing attack scenarios through correlation of intrusion alerts. In *CCS'2002*, pages 245–254.

Paxson, V. (1999). Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23–24):2435–2463.

Porras, P. A. and Neumann, P. G. (1997). EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *NIST-NCSC*, pages 353–365.

Valdes, A. and Skinner, K. (2001). Probabilistic alert correlation. In *RAID 200*, number 2212.