X-log Incident-Monitor System for Internal Control

Tibor Fellegi, Andrásné Kovács

Budapest Tech
Budai út 45, H-8000 Székesfehérvár, Hungary
fellegi.tibor@roik.bmf.hu, kovacs.andrasne@roik.bmf.hu

Pokó István

Budai út 141, H-8000 Székesfehérvár, Hungary, poko.istvan@seacon.hu

Abstract: Many kinds of internal incidents can occur in the firms and institutes for electronic storage and transmission of information (by intranet and Internet). These incidents cause serious damage with consequences.
The risk of human factor is reducible by the X-log System, an internal control system based on the analysis of log-files. The X-log System collects operations with application programs and operating system (login to system, data manipulation, transaction, etc.), filters and analyses them by system of rules, and – if needed – it alerts the the competent person. We have used the objects of IDMEF (Intrusion Detection Message Exchange Format RFC 4765) and Guidelines for Evidence Collection and Archiving (RFC 3227) for recordstructure of database.
The topics of our presentation are X-log system's main points, advantages, scheduling of installation and application possibilities of the previous two standards.
This program permits to get an inside view of ordered data for inexpert persons. They draw conclusions for the incident, because they know the internal situation better.

Keywords: internal incident, X-log system, IDMEF, RFC 4765, RFC 3227

1 Reasons for Developing X-Log System

Firms and institutes are exposed to a number of incidents regarding their data storage in informatic systems. There can be many kinds of attacks, a part of which aims information collection (or sometimes they try to prove the intruders' skills – e.g. NASA faces hundreds of intrusion attempts daily, and some of them are succesful). It is an indirect but very significant damage (e.g. information outflow through industrial/corporate spying). The internal attack causes damage which is directly measurable (e.g. via forging data).

According to the bibliography item [1] based on an investigation concerning 1271 firms during the last two years, every second firm experienced financial losses due to security problems in information flow. **The cause of these losses – in two thirds of the cases – were the employees of the firms**, every second of them had no intention to do harm.

Regarding data security there are several impotant areas to be protected, but the factor which is most difficult to protect – is the human factor. Extra investment is needed to protect against human attacks, and it is not a single investment. A special informatic monitoring system is to be created, which records the changes performed by user programs and any communication steps with the operating system, and in case of suspicion it alerts the competent person. This program permits to get an inside view of ordered data for incompetent persons. Thus – knowing the internal situation better – they can make conclusions from the damage (e.g. transferring a larger amount to unknown customers). As a requirement the system should have a friendly user interface so that inexpert people could use it.

1.1 Danger Sources

Dangers threaten all companies regardless of the fact that the firm is connected or not to any external communication network. The sources of hazard can be:

- **Unauthorized persons' intrusion in the network.**
- Trojan programs, **spyprograms**, viruses in the system.
- False identity detection.
- **Abuse of acquired authority** (e.g. passing benefits on to friends through the violation of the contract).

With the spread of the Internet technologies internet processes like TCP/IP, HTTP protocols, standard data formats, other services are more extensively used in the networks inside firms. Besides the advantages there are several disadvantages, that is, the open standards are known to any other people, making the intrusion easier.

Surveys show that 40% of the staff using computer systems are honest, 30% of them takes the chance to abuse, and a further 30% utterly looks for the chance to damage.

Table 1.1 shows the potential violators.

Firms not having adequate **internal security measures face significant security risks**. Typical steps to be taken:

- Internal firewalls;
- Coding;

- Use of logins, passwords when entering the intranet and with certain user programs;
- Defining the authority structure with application programs;
- Use of IDS (Intrusion Detection System);

Due to the spread of the intranet there is a transmission from verbal and paperbased communication to electronic channels (e-mails, video conferences, news), as a consequence, the access to the stored data, the search in them and **the abuse** is easier.

Table 1.1
The circle of potential violators

Potential violators	Method of detection	Evidence
Current and previous employees	Monitoring attempts	Logs, notes on equipment use
	Questioning witnesses	
Contracted employees	Examining logs, notes on equipment use	Other physical evidences
External persons	Specialized computer programs, which analyse the attempt attributes	

1.2 Reducing Risks – the X-Log System

The designed **X-log system intends to reduce these risks** by development an internal checking system based on log analysis. We would like to collect, filter, analyse the movements related to operating system, at the beginning, later the user programs. If needed, the X-log system alerts the appointed person.

The system is based on **the IDS basic principles**, that is the system monitors many informatic actions (check-ins, data manipulations, transactions, etc.), and attempts to detect illegal actions. To do this **two basic methods** are used:

Detecting deviations from the regulations (e.g. a check-in password of an employee who – according to the check-in system - does not work that day – in case of excluding remote check in).

Indicating deviations from regular or contracted values (e.g. giving a 70% benefit to the customer; or providing another cable-TV package for a certain user, which is not included in the contract).

Considering the first method **we set rules**, analyse correlations, which are regularly supervised, corrected so that we could recognise irregularities.

Considering the second method **we store the regular and maximum values of the important data,** and compare them with the current numbers (e.g. value of gross salary; or value of transferred amount to a customer). In case of a transaction

we monitor the specific data and if it is suspicious we alert the authorized person. In this case after long preparatory work we define the value limits, this value library is static, but must be maintaned from time to time.

In both cases we can expect processing and storing a large number of data.

1.3 The Advantages of the X-Log Program System

The designed system has the following advantages

- Employees' movements can be monitored, which scares away ill-willed people (40%).
- The simple existance of the system **holds back** the 'trial and error'- type **intruders**.
- The system administrators' movements are logged - and so - monitored.
- Shadowing **increases the customers' and investors' confidence**, enhancing the company turnover.
- Introducing the system is obviously cost increasing, but it also means **significant saving**. Due to the above mentioned reasons it can reduce damage thus the application of the X-log system can be profitable.

1.4 Tasks Related to the Introduction of the System

The infrastructure of the system must be built up, the necessary data must be provided in the system:

- The log possibilities of the given **operating system** must be found and adjusted to the X-log system.
- If there is a **check-in system** at the company, the data included there must be analysed and stored.
- **The user programs must be adjusted to the system gradually** considering the order of importance and the technical implementation possibilities.
- **The system of authority** must be examined, check-ins must be compared with the stored authorities, in case of deviations a warning is needed.
- **The users of the system** (internal auditors, company executives) **must be trained** to handle the program.
- **The alert system** must be worked out (who is supposed to be alerted, what way, and in case of what event).

- The system needs **computers of high capacity, high power**, since a large amount of data is processed and stored.
- An **archiving system** is needed so that the archived data be available when it is necessary.
- **The duration of storing the data** must be defined.
- **The IDMEF standard** (see later) recommendations provide easier connection to the system for firms.

There are only few developed systems regarding Hungary therefore we have a low number of examples. Thus, the development of the system is much more difficult than in the case of making systems similar to already used program systems.

Since data collection is based on the log-files first we should deal with the role and structure of the log-files.

2 The Role and Structure of Logfiles

The operating systems or other applications send messages about their own status or about previously occured incidents.

Syslog has been developed by Eric Allman in 1980 as a part of the sendmail project. At first it was used by only the sendmail, and it proved to be so successful that other applications started to use it including Unix and Linux systems, and the Microsoft Windows operation system as well.

The Parts of a Syslog Message

A complete syslog message on the forwarding line has **three easily distinguishable parts.**

The first part is the PRI, the second is the header, the third is the MSG. The total length of the string should be 1024 bytes or shorter. Considering the length of the syslog message there is no minimum criterion, although a syslog message without content makes no sense and there is no need to forward it.

The PRI part must consist of four or five characters, and they represent the significance and severity of the facilities.

The header of the syslog package contains a TIMESTAMP, and gives the IP address of the hostname and the device.

The MSG part of the syslog package usually contains supplementary information. It consists of two fields: TAG and CONTENT. The value of the TAG field will be the name of the program or the process, which is generated by the message. The CONTENT contains the details of the message.

3 Standards for Incidence Monitoring and Risk Analyse

3.1 SIM Systems Overwiev

Primarily the SIM (Security Incident Management) systems are based on incident defense, in addition, they contain several principles which are useful for us. The following figure shows the schematic routine of the incident response by netForensics.

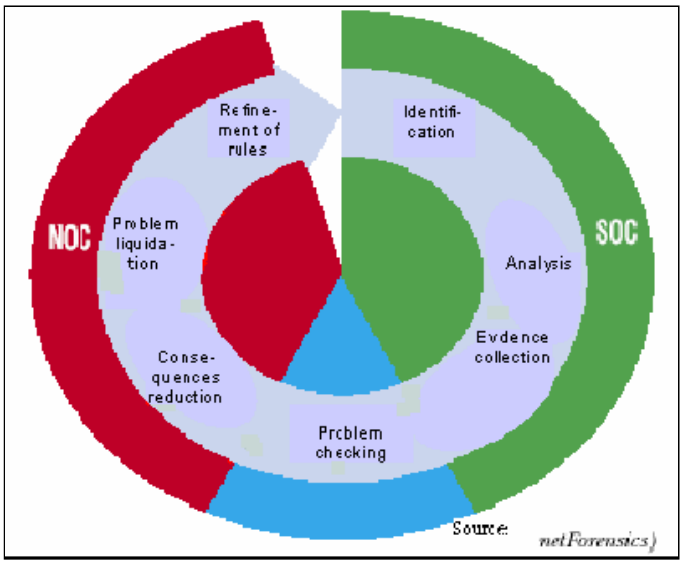

Figure 3.1
Schematic routine of the incident response on the netForensics nFx Open Security platform

3.1.1 SIM: Goals and Areas

The primary goal of handling security incidents is prevention, among them the proactive protection against known threats, the recognition of vulnerability and their liquidation, the prevention of security incidents and – of course - the defence and reduction of the consequences of unknown attacks.

3.1.2 IETF Intrusion Detection Workgroup (IDWG)

The aims of IDWG group is to create data format and data exchange processes which realise information exchange between intrusion detection systems without personal interference. In contrast to INCH the communication between systems is important (the defined data format is IDMEF).

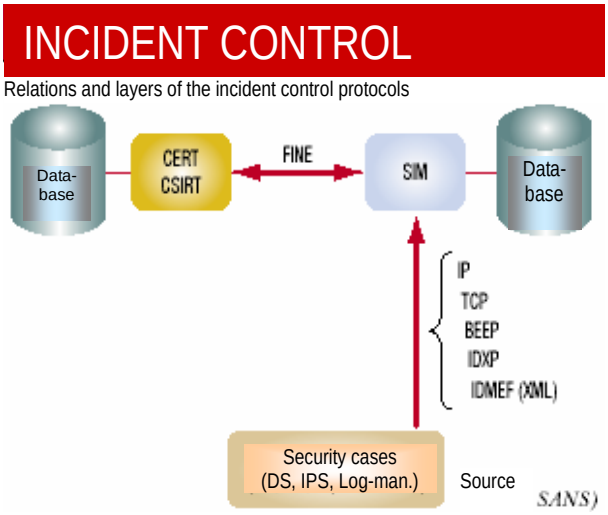

Figure 3.2
Relations and layers of the incident control protocols

3.1.3 The Standards Used for SIM Systems

IDMEF – Intrusion Detection Message Exchange Format. This standard defines dataformat and processes of communication, whis is based on XML (eXtensible Markup Language).

IDXP – Intrusion Detection Exchange Protocol. The data exchange standard by IDMEF.

BEEP – Blocks Extensible Exchange Protocol. Generic application layer protocol for trusted two-directed communication.

IODEF – Incident Object DetectionExchange Format. This is a data communication process of CERT and CSIRT, later changed by INCH (extended INCident Handling) standard.

FINE – Format for INcident report Exchange. The communication process used in data exchange between SIM and CERT/CSIRT promoted by INCH-workgroup and IETF.

3.2 Guidelines for Evidence Collection and Archiving (RFC 3227)

If needed, it is important to verify the suspicion by collected data of X-log system. The RFC 3227 standard [4] summarises the rules and steps of the evidence collection and archiving (2002).

3.2.1 Principles during Evidence Collection

- Adhere to our site's Security Policy and engage the appropriate Incident Handling and Law Enforcement personnel**.**
- Capture as accurate a picture of the system as possible.
- Keep detailed notes. These should include dates and times. If possible generate an automatic transcript. Notes and print-outs should be signed and dated.
- Note the difference between the system clock and UTC. For each timestamp provided, indicate whether UTC or local time is used.
- Be prepared to testify (perhaps years later) outlining all actions you took and at what times. Detailed notes will be vital.
- Minimise changes to the data as you are collecting it. This is not limited to content changes; you should avoid updating file or directory access times.
- When confronted with a choice between collection and analysis we should do collection first and analysis later.
- Our procedures should be implementable.
- We have to be methodical.
- Proceed from the volatile to the less volatile.
- We should make a bit-level copy of the system's media. If we wish to do forensics analysis we should make a bit-level copy of your evidence copy for that purpose, as our analysis will almost certainly alter file access times. Avoid doing forensics on the evidence copy.

3.2.2 Legal Considerations

Computer evidence needs to be:

- **Admissible:** It must conform to certain legal rules before it can be put before a court.
- **Authentic:** It must be possible to positively tie evidentiary material to the incident.
- **Complete:** It must tell the whole story and not just a particular perspective.
- **Reliable:** There must be nothing about how the evidence was collected and subsequently handled that casts doubt about its authenticity and veracity.
- **Believable:** It must be readily believable and understandable by a court.

3.2.3 The Collection Procedure

Our collection procedures should be as detailed as possible. The methods used to collect evidence should be transparent and reproducible. We should be prepared to reproduce precisely the methods we used, and have those methods tested by independent experts.

Where feasible we should consider generating checksums and cryptographically signing the collected evidence, as this may make it easier to preserve a strong chain of evidence. In doing so we must not alter the evidence.

3.2.4 The Archiving Procedure

Evidence must be strictly secured. In addition, the Chain of Custody needs to be clearly documented. We should be able to clearly describe how the evidence was found, how it was handled and everything that happened to it.

The following need to be documented

- Where, when, and by whom was the evidence discovered and collected.
- Where, when and by whom was the evidence handled or examined.
- Who had custody of the evidence, during what period. How was it stored.
- When the evidence changed custody, when and how did the transfer occur (include shipping numbers, etc.).

3.2.5 Tools We'll Need

We should have the programs we need to do evidence collection and forensics on read-only media (e.g., a CD). We should have prepared such a set of tools for each of the operating systems that you manage in advance of having to use it.

Our set of tools should include the following:

- a program for examining processes (e.g., 'ps').
- programs for examining system state (e.g., 'showrev', 'ifconfig', 'netstat', 'arp').
- a program for doing bit-to-bit copies (e.g., 'dd', 'SafeBack').
- programs for generating checksums and signatures (e.g.,'sha1sum', a checksum-enabled 'dd', 'SafeBack', 'pgp').
- programs for generating core images and for examining them (e.g., 'gcore', 'gdb').
- scripts to automate evidence collection (e.g., The Coroner's Toolkit).

3.6 Intrusion Detection Message Exchange Format (IDMEF) (RFC 4765)

Basically, the IDMEF defines the information interference of data format and data exchange processes in the intrusion detector and respondent system [2].

The Intrusion Detection Message Exchange Format (IDMEF) is intended to be a standard data format that automated intrusion detection systems can use to report alerts about events that they have deemed suspicious. The development of this standard format will enable interoperability among commercial, open source, and research systems, allowing users to mix-and-match the deployment of these systems according to their strong and weak points to obtain an optimal implementation.

4 The Recommended Object-oriented IDMEF Data Model

4.1 The Starting Point of X-Log Data Model Design

The data received by the data model are the messages, log-files of the different systems provided by **the X-log system's communication subsystem**. Standardization is normally the basis for the data model, since log data – in most cases – are stored and transmitted to the user in a producer-specific format. The starting point of the single format is the format of the **SYSLOG**. (Standard RFC 3164 is in accordance with the situation of June 14, 2007.) The data model should be able to analyse the records coming from the communication subsystem. The logics of the incident monitoring system:

- filters indifferent data,
- checks data records getting through the filter,
- must send alerts about suspicious transaction combinations.

At the initial state it must be able to process the central authority log (LDAP), the Seawing check-in system log, the ERP system abuse monitoring log and the database handling log, however, this range later should be extended.

Basically, the 'who, what, when, what for modified-type' information is needed also considering personal datahandling security, reconnaissance of unauthorized handling and access to these data.

4.2 Short Description of the IDMEF Data Model from the Aspect of Xlog

The data model is based on an object-oriented model of IDWG data description. The IDMEF basically is a dataformat standard for implementing SIM systems. Our goal is to expand it for understanding and processing logfiles.

According to the recommendation for all IDMEF messages the so called 'IDMEF-MESSAGE' class means the highest-level class. Each message is a subclass of this highest-level class. At present there are two subtypes defined: the **ALERT** and the **HEARTBEATS** subclasses. With each message the subclasses of the message classes provide the information transmission. It is interesting that the data model does not specify how to classify or identify an intrusion.

Furthermore, an incident is understood as such a single-format logfile record, which is received by the dataprocessing system as an input.

The understanding of the recommendation will not be disturbed if we use 'incident'in the meaning of 'alert'.

First let us describe the basic principles on which an IDMEF structure is based. As far as it is possible we keep the IDMEF terms, but instead of alert we use incident, that is log record, and we use the notion of 'tool' in a wider meaning (which in our case can be e.g. database program, operating system, ERP system' log, etc.), and this is the case with the notion of 'environment'.

The description of the incident information can be extremely different in the logfiles. Certain incident information contain only the origin, destination, name of the incident, and the time parameters, while other descriptions provide much more information. E.g.:

- Content correlations,
- Tools (ports) or services,
- Process information,
- User information, etc.

Therefore it is important that the description should be flexible. An object-oriented model of course can be extended with new classes and subclasses. The subclass and the aggregate provide the consistency of the model.

The tool environments can be different. Tools reporting the same incident but using different information sources will contain different information according to the IDMEF, this principle should be maintained in our case too. (It is possible that two logfiles contain the same information, only in a different value.) These principles are valid in our case too.

According to the IDMEF the classes which handle the incidents defined in the data model are located between the source and the target tools and a combination of the following:

NODE

USER

PROCESS

SERVICE classes.

It could be implemented and simplified but because of the uncertain future prospect it is better to stay with the IDMEF interpretation.

The environments are different, they depend on the type of the network, and on the used operating system, since the observed events, incidents have different attributes. From this aspect the definition of the NODE and SERVICE classes means the efficiency of the data model according to the IDMEF. If additional information is reported, we have to define a subclass in order to add these new attributes to our data model. It is a necessary principle with the X-log system too.

The tools can originate from different suppliers. The object-oriented model provides this attribute through heritance and through defining the subclasses. The problem and its solution is similar with the X-log too.

4.3 The Viewpoints of the Design

A brief overview of the IDMEF data model can be useful too.

4.3.1 Representing Events

The goal of the data model is to provide a standard representation of the information that an intrusion-detection analyzer detected an occurrence of some unusual activity. These alerts may be simple or complex, depending on the capabilities of the analyzer that created them.

4.3.2 Content-Driven

The design of the data model is content-driven. This means that new objects are introduced to accommodate additional content not semantic differences between the alerts. This is an important goal as the task of classifying and naming computer vulnerabilities is extremely difficult and subjective.

5 The IDMEF Data Model

To describe the data model the UML (Universal Modeling Language) is used. The UML provides a simple structure for the description of the classes and relationships between the classes. UML define entities as classes.

Aggregation is a form of association in which the whole is related to its parts. In this case the aggregate class contains all of its own attributes and as many of the attributes associated with its parts as required and specified in the multiplicity indicators. In this document the symbol ◆ is used to indicate aggregation.

Multiplicity defines the number of objects within a class that are linked to one another by an aggregation relationship. Typically multiplicity indicators are placed at each end of the association line. Default value is 1. Standard symbols, as used in this document, are:

1 = Exactly one
0..* = Zero or more
1..* = One or more
0..1 = Zero or One
5..8 = Specific range (5,6,7, & 8)

5.1 The IDMEF Data Model Overview

An overview of the data model is presented in Figure 5.1. The main component is the ALERT class, which bears minimum required information along with the ANALYZER and CLASSIFICATION classes. The ANALYZER class describes the sender of the alert, i.e. the analyzer; every alert must be associated with one and only one analyzer. The CLASSIFICATION class describes the subject of the alert, i.e. the reason for sending it; at least one of them MUST be provided in the alert.

Each alert is associated with zero or more TARGETs, and with zero or more SOURCEs. Each TARGET and SOURCE is described by a number of attributes. Provision of additional alert data is done by subclassing any of the model classes.

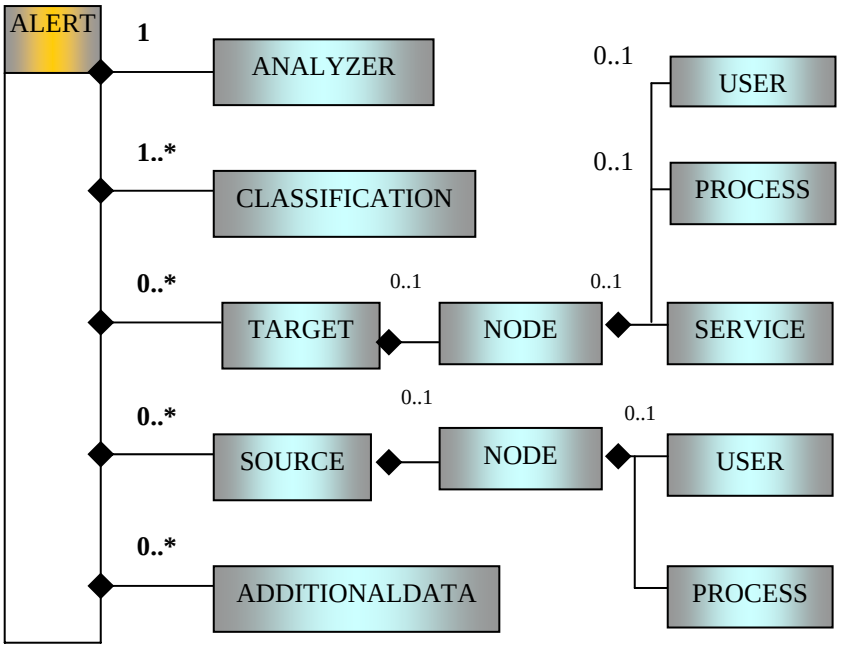

Figure 5.1
Data Model Overview

5.2 The Core of the Data Model

The core of the data model is the ALERT class. Every alert is associated with a single **analyzer** that generated it, and a single time. It is also associated with a list of 0 or more **targets**, and a list of 0 or more **sources**. This relationship is illustrated in Figure 5.2.

Figure 5.2 contains three classes that extend the ALERT class. These three classes have been included here because the information they carry appears frequently during the operation of intrusion-detection systems.

- **The ALERT class** is the central component of the data model. An IDWG-compliant intrusion-detection analyzer must generate at a minimum this set of information.
- **The TOOLALERT class** carries additional information related to the use of attack tools or Trojan horses.
- **The CORRELATIONALERT class** carries additional information related to the correlation of alert information.
- **The OVERFLOWALERT class** carries additional information related to overflow attacks.

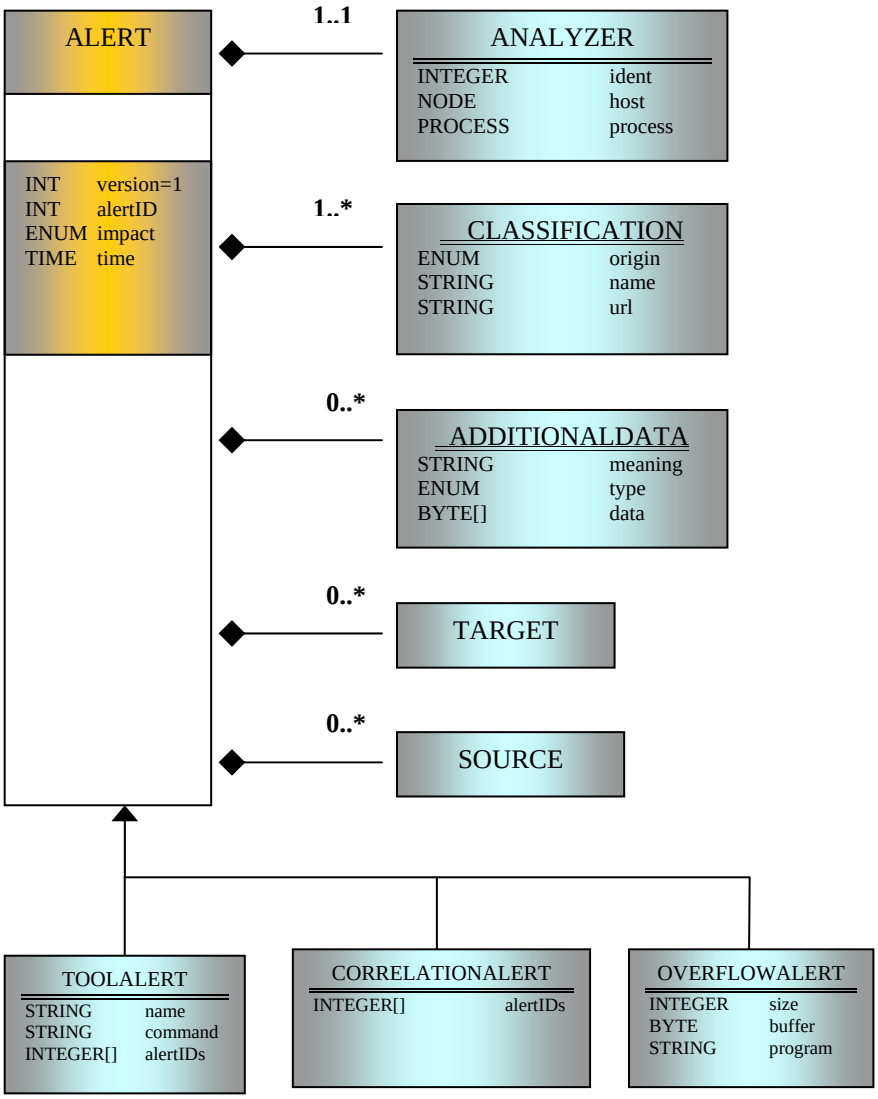

Figure 5.2Data model core

- **The ANALYZER class** identifies the intrusion detection analyzer that provided the alert. At the minimum, this is a unique identifier such as a serial number (unique over the organization where the IDS system is deployed). Additional identification information is provided.
- **The CLASSIFICATION class** names the vulnerability associated with the alert. One name MUST be provided, and additional, equivalent names MAY be added.

- **The ADDITIONALDATA class** provides a way to carry vendor-specific or implementation specific information.
- **The TARGET class** contains information about the target of the alert. It may consist of four classes: NODE, USER, PROCESS, SERVICE (the FILE class is optional).
- **The SOURCE class** contains information about the possible source or sources of the alert. It may consist of three classes: NODE, USER, PROCESS. An alert has more source (e.g. distributed DOS attack).

The IDMEF prefers **the XML implementation**. The XML is a reasonable choice, which is based on the DTD improvement and examples. (According to the IDWG recommendation of September, 1999; February 2000 the XML solution meets the requirements the best.)

References

[1] Othmar Kyas: Számítógépes hálózatok biztonságtechnikája, Budapest, 2000

[2] RFC 4765 The Intrusion Detection Message Exchange Format (IDMEF), http://www.rfc-editor.org/rfc/rfc4765.txt

[3] "Extensible Markup Language (XML)", W3C Recommendation, 1998 http://www. w3.org/TR/1998/REC-xml-19980210

[4] RFC 3227 Guidelines for Evidence Collection and Archiving http://rfc.net/rfc3227.html

[5] [FAR1999] Farmer, D., and W Venema, "Computer Forensics Analysis Class Handouts", http://www.fish.com/forensics/

[6] [RFC2350] Brownlee, N. and E. Guttman, "Expectations for Computer Security Incident Response", FYI 8, RFC 2350, June 1998