Volume 2, No. 6, June 2011

Journal of Global Research in Computer Science

ISSN-2229-371X

RESEARCH PAPER

Available Online at www.jgrcs.info

ATTACK DESCRIPTION LANGUAGE FOR COLLABORATED ALERTS-USING XML AND UML

[1*]K.V.S.N. Rama Rao, [2]Manas Ranjan Patra
[1]Bandari Srinivas Inst.of Technology Hyderabad,India
kvsnramarao@yahoo.co.in
[2]Berhampur University Berhampur,Orissa
mrpatra12@gmail.com

***Abstract*—**Statistics of Internet usage are increasing enormously. In harmony, the attacks are also escalating. In the recent era, IDS have gained more popularity in connection to network security. IDS deployed in the network will scan the hosts and the network. It will try to sense misuse detection or anomaly detection. Whenever there is any suspicious activity, IDS will immediately raise alarm. It would be apt to capture the complete description of the new attack as soon as alarm rises. This information to be collected may be heterogeneous because it may be from multiple users, process or hosts. Hence there is a need for common standard language that will work across various domains and platforms. XML is one such language.Writing an XML schema directly would be difficult and inconvenient. The best way to write XML schemas is to useUML models. Hence in this paper, we propose alert collbaration modeling architecture and attack description language using XML notion, which uses UML modeling.

Keywords- *IDS; Attack description; XML;UML*

INTRODUCTION

Intrusion Detection is the process of identifying and responding to intrusion activities .IDS can be at host or network level. IDS will aim to spot known and unknown attacks namely misuse detection and anomaly detection. Anomaly detection can be categorized as unknown attack identification. In anomaly detection, the system will have historic information about normal behavior of the system under certain conditions. Whenever any activity is violating this behavior and behaving abnormally then IDS will alert. Misuse detection is known attack identification. In this it has a huge database of signatures, which are attack descriptions. The incoming network packet stream is matched against these known patterns of signatures. If match is found then alert is produced. Alerts are raised autonomously arriving from various resources such as Firewalls, Hosts, File system integrity checkers, system call traces etc. Since these are arrived from various resources, they will be in different formats. So different formats will make it difficult to build a unique representation of attack. Hence we can say that, intrusion detection systems are facing the problem of evaluating large number of alerts in dissimilar formats added with high false alarm rate. Alerts may correspond to multiple stages of a single attack. In reality there will be a rational association between the alerts. If we can associate alerts, we can try to describe the attack scenario.

In recent times, alert correlation and alert verification have become common promising techniques for describing attack scenarios. Alert correlation can be referred as understanding and study of intrusion alerts with the purpose of intrusion alert fine-tuning and intrusion scenario construction.

There are many languages developed to describe the attack or event. But those languages are suited only for few environments. So there is a need for the development of an attack description language that suits for variety of platforms and domains. Hence in this paper we propose a XML based attack description language that suits for any platform and domain.

ATTACK COLLABORATION MODELLING ARCHITECTURE

Figure 1 shows the proposed architecture for attack collaboration modelling. The proposed architecture shows several important components.

Alert collector:

This component will assume the responsibility of collecting the alerts that were generated by various external sources such as Firewalls, Network management systems, Files, Business Rules etc. As mentioned earlier, these alerts will be different formats. Significant characteristics such as source host, target host, file accessed port numbers, IP addresses are gathered and maintained in the form of a Relational Database.

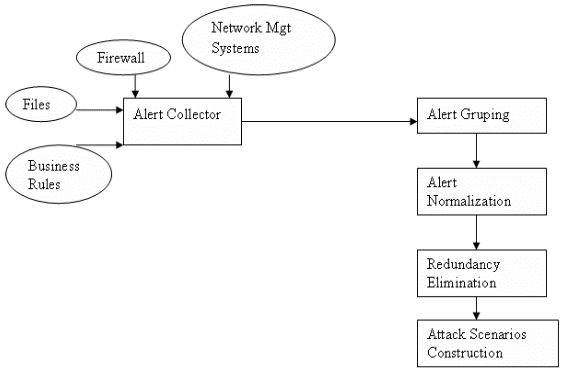

Figure 1. Alert collabaration modelling architecture

Alert grouping:

This component will group the alerts. The alerts that have analogous characteristics are retrieved from the database and were made as a group.

Normalization:

The alerts received are in different formats. Normalization will render a standard format for each alert. This standard format consist various parameters that make up alert and environment that led to attack.

Duplicate elimination:

This component will eliminate if the same alert is present multiple times. This process will greatly reduce the number of alerts.

Alert aggregation:

This will specify the association of an alert with a specific attack. An aggregation is a strong form of association in which the aggregate object is made up of constituent parts.

OUR APPROACH FOR ATTACK SCENARIO DESCRIPTION

Intrusion Detection systems need to collect huge network traffic information from heterogeneous sources. Gradually this database will be growing, making it more difficult to analyze the network events. Hence we present a concept that makes understanding, analyzing and representing attack scenarios in a flexible manner using UML and XML.

Why UML: UML (Unified Modeling Language) is an object-oriented analysis and design language defined by the Object Management Group (OMG). UML is used as a graphical tool to create abstract models. Conceptual models support a good understanding of the application domain UML can be used for this purpose.

Why XML: XML (eXtensible Markup Language) is a standard method for data interchange in the Internet.

XML schemas will be used to define and constrain the nature of XML exchanged. XML Schema is a text-based document. However, creating XML Schema manually is error-prone and inconvenient. A better way to design an XML Schema is through UML.

So our approach is a 3-step process

Step1: Modeling the concept using UML class diagrams so that we can identify objects and their relations.

Step2: Using these UML class diagrams, XML schemas were written.

Step3: Then the final version of XML is created.

Our model is based on following assumptions

A) An attack scenario is a collection of set of events.

B) Each event will have initial state and final state.

C) Each event is associated with certain actions.

D) Based on these actions, transitions from state to state will occur.

Step 1: Modeling the concept using UML class diagrams

Several classes like several classes like scenario, event, actions, vulnerability, state, initial state and final state are shown in figure 2.

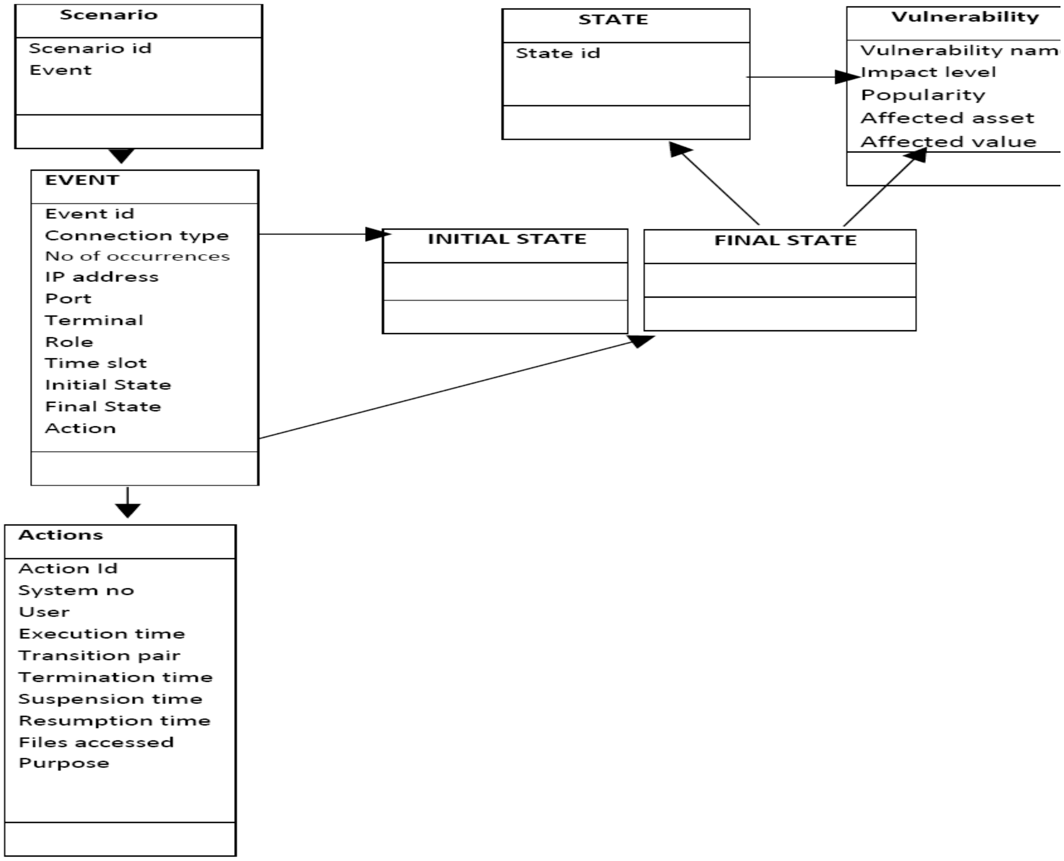

Figure.2 UML Class diagram

Step 2: Next step will be to write XML schemas based on the above UML diagram

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"elementF
ormDefault="qualified"
attributeFormDefault="unqualified">
<xs:element name="Scenario">
<xs:complexType>
<xs:sequence>
<xs:element name="scenario id"type="int"/>
<xs:element name="Event " type="Event"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

In the above lines, the Scenario class in UML is represented by the complex type Scenario in the schema. Scenario class diagram consists of scenarioId and Event as attributes which are represented in the above XML schema. Similarly, we can represent XML schema for an Event class as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"elementF
ormDefault="qualified"
attributeFormDefault="unqualified">
<xs:element name="Event">
<xs:complexType>
<xs:sequence>
<xs:element name="event id"type="int"/>
<xs:element name="Action " type="Action"/>
<xs:element name="Initial state " type=" Initial state "/>
<xs:element name=" Final state " type=" Final state "/>
<xs:element name=" Connection type " type=" string "/>
………………………………………………………
---------------------
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

In the above figure, State class is an abstract type. The classes Initial state and Final state inherit the characteristics of State class. Hence State class can be called as reusable type class. In XML schemas, to represent reusable classes or abstract types we can use the keyword abstract="true".

```
<xs:complexType name="State" abstract="true"> <xs:
sequence>
```

While writing schemas for Initial and Final state classes, we can use extension base="State" to show that they inherit the characteristics of State. The Schema can written as

```
<xs:complexType name="Initial State">
<xs:complexContent>        <xs:extension base="State"/>
</xs:complexContent>
</xs:complexType>
```

In addition, we can also capture in the model that a Scenario can consist of many Events (as per assumption 1) with the code type="Events" maxOccurs="unbounded".

```
<xs:complexType name="Scenario">
<xs:sequence>
<xs:element         name="Event"         type="Event"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
```

The following is a resulting instance code for a scenario, given the schemas that we created earlier.

Step 3:

Scenario.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Scenario
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="D:\schemas\
Scenario.xsd">
< ScenarioId> </ ScenarioId>
<Event 1>
<Event id> </ Event id >
<Connectiontype>TCP/IP</Connection     type>
<No of occurrences> </ No of occurrences >
<IP Address>192.160.23.2</IP Address>
<Port>8080</Port>
<Terminal> </Terminal>
<Role>user</Role>
<Time slot> </Time slot>
<Action 1>
<Action Id> </Action Id>
<System No> </System No>
<User> </User>
<Execution time> </Execution time>
<Transition pair> </Transition pair >
<Termination time > </Termination time >
<Suspension time> </Suspension time >
<Resumption time> </Resumption time >
<Files accessed> </Files accessed >
<Purpose> </Purpose>
</Action 1>
<Action  2>
……………
</Action 2>
<Initial State>
<state id>  </stateid>
</Initial State>
<Final State>
<state id>  </stateid>
</Final State>
</Event 1>
<Event  2>
…………………….
…………………….
</Event 2>
</Scenario>
```

Based on this Meta data, we can identify how an attacker has got access to victim system. This can be known by Port number, IP address and connection type metadata. Using the “Files accessed” metadata we can identify which file or attachment has activated this vulnerability. For example opening an email attachment may activate vulnerability. We can also know during the transition from which state the vulnerability has occurred. The vulnerability class will capture the vulnerability name; impact level, affected asset and value. So in this manner our model helps in describing attack.

RELATED WORK

Recently, intrusion alert correlation and event correlation areas are gaining the attention of the researchers. These two areas are interrelated to each other. Amoroso [11] mostly focuses on intrusion event correlation. According to [11], three types of correlation are identified.

A) Single-session versus multiple-session network correlation: Single session network correlation refers to the correlation of information related to a stream of packets between two endpoints. Context management, memory management and state information maintenance are several problems identified for single session.

In the case of multiple session network correlation, the problems may be Remote sessions, Same source or destination end points, Time inconsistencies, Patterns in unrelated sessions and Connecting unrelated sessions.

B) Real-time versus after-the-fact correlation: Real-time analysis cannot use a “look forward” function, i.e. use the ability to “jump forward in time” during batch-processing of stored event logs for instance.

C) In-band versus all-band information:The target system has inbuilt computing and networking activity which is refered as inband.All other information will be out band.Combination of in band and out band will form all band.A common format is a problem in all band.

Regarding intrusion alert correlation, IETF is developed the Intrusion Detection Message Exchange Format (IDMEF) draft standard [12].This has several classes and sub classes. Analyzer class may be used to identify which sensor has generated the alert. The classes detect, create and analyzer time will represent various aspects of date and time. The source class will identify the offending party while the target class is to specify possibly affected entities. The assessment class has in turn aggregate classes such as impact, action and confidence.The classification class will provide additional details regarding the alert.Additional data class is provided for heterogeneous alert correlation.

IBM used the concept of aggregation for grouping alerts as per selection criterion. They named the product as Aggregation and Correlation Component (ACC) [13]. A formal data model for the alert correlation process is developed in M2D2 [14].Vulnerability scanners will provide some information, which will be correlated with alerts from IDS. This will help to gain additional information about monitored system and security tools.

In the EMERALD architecture [15][16][17], Hierarchical correlation is performed using four different concepts such as minimum similarity , feature overlap, feature similarity and expectation of similarity. M-Correlator is an extension to the EMERALD architecture which provides additional information about the target operating system, priority with respect to criticallity alert type as well as incident ranking based on Bayes networks [18]. MIT Lincoln Laboratory presents an alert correlation model [19] by making use of automatic optimization of correlation parameters using training sets of tagged alerts.

The “Prerequisite” Approach presents the concept of a hyper-alert, facts, prerequisites, and consequences of intrusions [20][21][22][23]. The MIRADOR project includes an intrusion alert correlation component called CRIM [24][25]. Inward alerts are clustered and combined and then processed.

CONCLUSION

XML has become a widely accepted standard for information exchange. Greatest advantage of XML is that it has no barriers. It can work in any platform and in any domain. So representing attack information in XML would be more beneficial because it will be originated from multiple users, process & hosts. Straight away writing an XML schema would be difficult and inconvenient. Hence, the superior way to write XML schemas is to use UML models. So in this paper, we used UML class diagram to identify objects and their relationships. Then an XML schema is designed and finally its XML instance is written. This metadata is used to describe attack.

REFERENCES

[1] Karine P. Peralta, Alex M. Orozco, Avelino F. Zorzo, Fl¶avio M. Oliveira “Specifying Security Aspects in UML Models”

[2] Siv Hilde Houmb and Kine Kvernstad Hansen “Towards a UML Profile for Security Assessment”

[3] Hong-Bae Jun, Dimitris Kiritsis,and Paul Xirouchakis “Product usage information modelling in a ubiquitous environment”.

[4] Marcus Fontoura, Carlos J. Lucena, Alexandre Andreatta, Sérgio E. Carvalho, and Celso C. Ribeiro” Using UML-F to Enhance Framework Development: a Case Study in the Local Search Heuristics Domain”

[5] Ebenezer A. Oladimeji, Sam Supakkul, Lawrence Chung “Representing Security Goals, Policies, and Objects”

[6] Booch, Grady; Rumbaugh, James; Javobson, Ivar. “The Unified Modeling Language User Guide”. Prentice-Hall, Inc,.

[7] Carlson, David. “Modeling XML Application with UML: Practical e-Business Application”. Addison-Wesley.

[8] Deitel, H.M.; Deitel, P.J.; Nieto, T.R.; Lin, T.M.; Sadhu, P. “XML How to Program”. Prentice Hall, Inc, 2001.

[9] W3C. “Extensible Markup Language (XML) 1.0 (Third Edition)”.http://www.w3.org.

[10]W3C. “XML Schema” http://www.w3.org

[11] E. Amoroso. Intrusion Detection - An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response. Intrusion.Net Books. ISBN 0-9666700-7-8.1999.

[12] D. Curry and H. Debar. Intrusion Detection Message Exchange Format. http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-10.txt. August 2003.

[13] H. Debar and A. Wespi. Aggregation and Correlation of Intrusion-Detection Alerts.In Proceedings of the 4th International Symposium, Recent Advances in Intrusion

Detection (RAID) 2001, Springer-Verlag Lecture Notes in Computer Science, October 2001.

[14] B. Moring, L. Me, H. Debar, and M.Ducassé. M2B2: A formal Data Model for IDS Alert Correlation. In Proceedings of the 5th International Symposium, Recent Advances in Intrusion Detection (RAID) 2002, Springer-Verlag Lecture Notes in Computer Science, October 2002.

[15] A. Valdes and K. Skinner. Probabilistic Alert Correlation. In Proceedings of the 4th International Symposium, Recent Advances in Intrusion Detection (RAID) 2001,Springer-Verlag Lecture Notes in Computer Science

[16] A. Valdes and K. Skinner. Adaptive, Model-Based Monitoring for Cyber Attack Detection. In Proceedings of the third International Workshop, Recent Advances in Intrusion Detection (RAID) 2000, Springer-Verlag Lecture Notes in Computer Science, October 2000.

[17] D. Andersson, M. Fong, and A. Valdes. Heterogeneous Sensor Correlation: A Case Study of Live Traffic Analysis. In IEEE Information Assurance Workshop, United States Military Academy, West Point, NY, June 2002.

[18] P. A. Porras, M. W. Fong, and A. Valdes. A Mission-Impact-Based Approach to INFOSEC Alarm Correlation. In Proceedings of the 5th International Symposium,Recent Advances in Intrusion Detection (RAID) 2002, Springer-Verlag Lecture Notes in Computer Science, October 2002.

[19] O. M. Dain and R. K. Cunningham. Building Scenarios from a Heterogeneous Alert Stream. In IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 5-6 June 2001.

[20] P. Ning, D.S. Reeves, and Y. Cui. Correlating Alerts Using Prerequisites of Intrusions. Technical Report TR-2001-13, North Carolina State University, Department of Computer Science, 2001.

[21] P. Ning and Y. Cui. An Intrusion Alert Correlator Based on Prerequisites of Intrusions.Technical Report TR-2002-1, North Carolina State University, Department of Computer Science, 2002.

[22] P. Ning, X.S. Wang, and S. Jajodia. Modelling Requests among Cooperating Intrusion Detection Systems. Computer Communications 23(17):1702-1715, Elsevier Science, 2000.

[23] P. Ning, Y. Cui, and D. S. Reeves. Analyzing Intensive Intrusion Alerts via Correlation.In Proceedings of the 5th International Symposium, Recent Advances in Intrusion Detection (RAID) 2002, Springer-Verlag Lecture Notes in Computer Science, October 2002.

[24] F. Cuppens and A. Miège. Alert Correlation in a Cooperative Intrusion Detection Framework. In Proceedings of 2002 IEEE Symposium on Security and Privacy, 2002.

[25] F. Cuppens. Managing alerts in multi-intrusion detection environments. In 17th Annual Computer Security Applications Conference (ACSAC). New-Oreans, December 2001.