# OntoLog: Using Web Semantic and Ontology for Security Log Analysis

Clovis Holanda do Nascimento[1], Felipe Silva Ferraz[2],
Rodrigo Elia Assad[2], Danilo Leite e Silva[1], Victor Hazin da Rocha[2]

[1]CESAR – Recife Center for Advanced Studies and Systems
{clovishn,daniloleite2}@gmail.com
Informatics Center
[2]Federal University of Pernambuco (UFPE) Recife – PE, Brazil
{fsf3,rea,vhr}@cin.ufpe.br

***Abstract*—Along with the growth of available information on the internet, grows too the number of attacks to the Web systems. The Web applications became a new target to those invaders, due to the popularization of the Web 2.0 and 3.0, as well as the access to the social networks system's API's, the cloud computing models and SaaS. In this context, the identification of an eventual attack to those applications has become an important challenge to the security specialists. This article presents a proposition of using Semantic Web and Ontology concepts to define an approach to analyze Security logs with the goal to identify possible security issues.**

***Keywords-Security; Log Analysis; Ontology***

## I. INTRODUCTION

Log Analysis to search for information that can provide data about the process of identifying evidence, events and user profiles related to the system, consists in an ordinary and necessary activity for the teams that administer and manage systems. With the growth and popularization of Web systems [7] [12], the volume of information generated in logs has grown considerably.

The growth of generated logs made the most common techniques used to analyze them, such as looking for evidence of certain attacks and even compromising those by finding patterns in the logs, not as effective as they were before [4]. This scenario become even more complex when there is the need to identify co-relation between the events that are in the logs, such as identifying which operations a determined user in which systems in the last 3 days?

Alongside the problems described, we are maturing the definition of what can be defined as an attack and how an eventual attacker would use it [17] [24], what allowed to be adopted more sophisticated mechanisms, generating detailed data about the event, but making the log analysis more complicated.

In this context, the use of Semantic Web technologies, specifically, the use of ontologies, in the context of security log analysis, showed itself as a possibility of improving the results of the searches in the log files. Generally, is expected that the ontologies can help in the interpretation process of interpretation of the great diversity of information that are present in this kind of archive [3] [5] [6].

Fundamentally, the role of ontology is to enable the construction of a representation model of a given area through the representation of a terms and relations vocabulary [21]. According to Gruber [9] , Ontology is a formal and explicit specification of a shared conceptualization. Thus, as a formal specification, the ontology can be processed by computer software with precision in the analysis, facilitating the search in the log files and thus improving the efficiency of the results analysis [8].

This article aims to concisely present the proposal for the use of ontologies to analyze and process logs generated by web application firewall [15], identifying the generated types of information, its importance and the problems related to the log files.

The remaining sections of this paper are divided as follows: Section 2 presents the difficulties related to log analysis and security. Section 3 presents the use of ontologies for log analysis. Section 4 presents the results of the experiment. Finally, Section 5 presents the conclusions.

## II. LOG ANALYSIS DIFFICULTIES AND SECURITY

The information stored in the logs are invaluable to the security area, for they have the attacks records, input ports, IP numbers, evidence of invasion, typed commands, among others. In addition, logs can be considered as a source in constant growth, due to the use of systems on a daily basis. Kimball and Merz [11] present some problems found in the log files, such as; multiple file formats, dispersed information, incomplete, inconsistent and irrelevant data, which makes the analysis and extraction of information from these files harder to accomplish.

The security auditor or system administrator has as part of their daily routine duties a rather hard activity, the research and analysis of logs. This task is considered difficult and time consuming, because the log files are created without a semantic description of their format, making the extracting of the meaning of the data impracticable, showing only the words typed in the search, resulting in poor quality results. According to Guarino [10], this limitation occurs because when the data is generated without the use of ontology, it can present ambiguities and vagueness of

 ISBN: 978-1-61208-165-6 

elements. This situation becomes more serious when we face major files with gigabytes of information.

## III. THE USE OF THE ONTOLOGY FOR LOG ANALYSIS

There are various definitions found in literature about what is ontology. Originally the term was born in the field of philosophy, being a word of Greek origin, which deals with the nature of being and its existence. In the field of Computer Science, it was first applied in the artificial intelligence field to computational representation of knowledge, being considered an abstract model of a domain. Below are some of the most used definitions for the term ontology:

- According to Gruber [9], "ontology is a formal and explicit specification of a shared conceptualization."
- The W3C consortium [25] defines ontology as: "the definition of terms used to describe and represent an area of knowledge."
- According to Noy and McGuinness [16], there is no one correct way to model a domain, meaning that there is more than one way to develop an ontology.

The basic components of ontology are classes (organized in taxonomy), relations (used to connect the domain concepts), axioms (used to model sentences that are always true), properties (describe characteristics common to the instances of a class or relationships between classes) and instances (used to represent specific data).

Ontologies are used for modeling data from specific domains and also allow inferences to discover implicit knowledge in these. Despite the considerable increase in the use of ontologies, build a complete ontology covering all the expressiveness of the domain continues to be a hard work, making the work of a multidisciplinary team a necessity, in which case it would be an ontology engineer, a security expert, among others, and acting in a participatory and integrated [22] [24].

More specifically, in this work, we are interested in building ontology for the representation of data available in security logs of web applications. In this context, ontologies can be useful for improving the classification of the attacks occurred and the identification of related events.

In the next session, we present an overview of ontology, and describe the methodology used for its creation.

### A. General description of the proposed ontology

The proposed ontology, OntoSeg, has as main objective the representation of data generated by the application firewall log ModSecurity on this work. From a detailed analysis of several samples of the log we identified various classes and their relations. Table 1 presents a brief description of the main classes that compose the ontology for the representation of the security log.

TABLE 1. MAIN CLASSES OF PROPOSED ONTOLOGY

| Class | Definition |
|---|---|
| Audit log header | Represents the log header, and contains the following information: date and time, ID (transaction ID, with a unique value to each transaction log), source IP, source port, destination IP and port of destination. |
| Request Headers | Represents the request Header, contain the information of the request and the header of the solicitation that was sent by the client. |
| Request Body | Represents the body of the transaction, contains the contents of the client request. |
| Intended Response Headers | Represents the status line and headers, this part is reserved for future use, so it is not implemented |
| Intended Response Body | Represents the body of the response of the transaction, the response body contains the actual case the transaction is not intercepted. |
| Response Headers | Represents the header of the actual response sent to the the client, contains data and headers |
| Response Body | Represents the body's effective response to the request, but this part is reserved for future use, so it is not implemented |
| Audit Log Trailer | Represents additional data, contains additional meta data of the transaction obtained from the web server or ModSecurity |
| Reduced Multipart Request Body | Represents the body of the request reduced, Part I is designed to reduce the size of the request body with irrelevant content from the standpoint of security, transactions that deal with uploading large files tend to be big |
| Multipart Files Information | Represents information about the files, the objective of this part is to write the metadata information about the files contained in the request body, this part is reserved for future use, so it is not implemented |
| Matched Rules Information | Figure 1. Represents the rules, contains a record with all the rules that occurred during the processing of transactions ModSecurity |
| Audit Log Footer | Represents the end of the log, your goal is just to indicate the end of a log file |

Figure 1 represents the relationships proposed in OntoSeg. As can be noted several branches show the complexity of the domain.

The basic relationships between the classes are performed using the property ID (transaction ID, with unique value to each transaction), derived from the log ModSecurity [15] that defines the ModSecurity_log main class, and from this we have the following derived classes:

- Response_header: contains all the information related to the HTTP header response
- Request_headers: contains all the information related to the HTTP request header
- Audit_log_header: contains all the information related to the header of IP and TCP
- There still is a set of information that derive from ModSecurity_log class that contains information about the HTTP message body from these subclasses we have the derivation of other subclasses that contains each of the basic elements of ontology OntoSeg.

 ISBN: 978-1-61208-165-6 

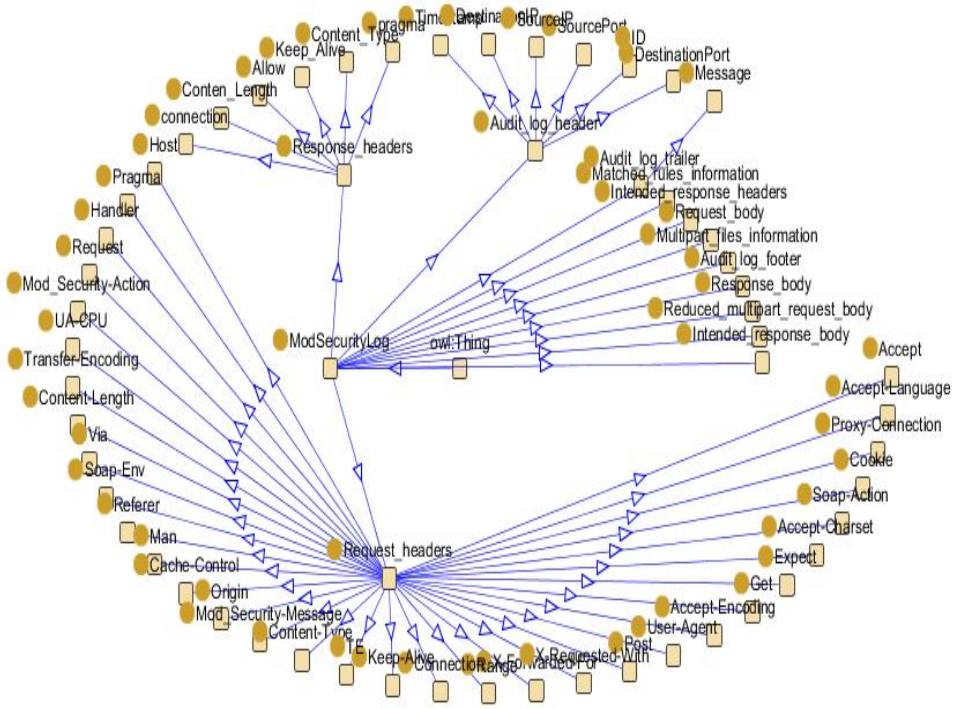


Figure 2. Figure 1. Conceptual model of ontology in protégé - radial view. [25]

Figure 2 represents the definition of the SourcePort, Get, and SourceIP Content-Type subclasses, with their classes, respectively, Audit_log_header, Request_headers, and Request_headers Audit_log_header:

```
<owl:Class rdf:ID="SourcePort">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Audit_log_header"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Get">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Request_headers"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Content-Type">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Request_headers"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SourceIP">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Audit_log_header"/>
  </rdfs:subClassOf>
</owl:Class>
```

Figure 2. Excerpt from the code of the ontology developed in the owl language [13]

### B. *Creation proccess of the proposed ontology*

The experiment with the use of ontologies for log files analysis was developed using an actual log of Web applications developed by the company CESAR (Center for Advanced Studies and Systems of Recife) that contained a series of attacks by these applications. This server has about 212,000 page views per month from 35,900 different addresses. The Web server used is Apache running in the Linux operational system that uses the ModSecurity [15] program as a firewall of web applications. The filter rules include several filters and for most of the known WEB attacks, for security reasons we cannot detail how they are configured.

From the analysis of logs generated by the application firewall, it was possible to identify the major classes of the ontology and their relationships. The universe of terms that compose the proposed ontology was defined based on the log of ModSecurity that records all the transaction data from the CESAR WEB systems. The classes were defined according to the division of parts of the ModSecurity log and the subclasses were established based on the configuration of keywords enabled by the administrator of ModSecurity, which were 41 words. This amount could be higher or lower depending on the established configuration; the instances are data that are part of the logs.

 ISBN: 978-1-61208-165-6 

Figure 3 shows an excerpt of the ModSecurity log, where you can see that there is no semantic description associated with its data, thus limiting the expressiveness of the concepts.


Figure 3. Example of an excerpt from the log of ModSecurity, which represents a real SQL injection attack (some data has been changed to preserve confidentiality of the company)

For the creation of the ontology were chosen: a language for representing ontologies, a specific tool for working with ontologies, and some methodologies for the constructions of ontologies with defined roles. The below summarizes the choices that were made during the creation of ontology:

- Language for definition of the ontology:
  In order to create an ontology that can be semantically processed by computers, the OWL language was adopted, that has these characteristics, and currently is the language recommended by the W3C consortium [1].

- Methodologies for building ontologies:
  In the proposal, the following methodologies were used to develop ontologies [2]:

  - 101: Methodology of the simplified and interactive process, which has the following steps: Determine the domain and scope, consider reuse, list relevant terms, define classes and their hierarchy, define classes properties, set property values and creating instances.
  - Uschould and King: Methodology consists in four distinct stages: Identifying the purpose of the ontology, construction, evaluation and documentation.
  - Methontology: Methodology that suggests a life cycle of evolutionary model, composed by the following phases: planning, specification, knowledge acquisition, conceptualization, formalization, integration, implementation, evaluation, documentation and maintenance.

- Tool for the creation of the ontology:
  The Protégé [20] tool was chosen that allows the construction and edition of ontologies through a graphical interface of easy interaction. It also allows the insertion of new capabilities by installing plug-ins. In our case the OWL and Jambalaya plug-ins were installed. Other features of this tool are the importing and exporting of ontologies, open source and be developed in Java. Based on the comparative study of SILVA, Daniel Lucas; ROCHA SOUZA, Renato; ALMEIDA, Mauricio Barcelos [2], which presents the methodologies for building ontologies, three methods were selected according to the activities involved in the proposal, as described in Table 2.

TABLE II. METHODOLOGIES USED IN EACH STEP OF THE LIFE CYCLE OF THE ONTOLOGY

| | **METHODOLOGY** | | |
|---|---|---|---|
| **ACTIVITY** | Uschould e King | 101 | Methontology |
| Determination of the propose of the ontology. | ■ | | |
| Definition of classes, properties and instances. | | ■ | |
| Construction of the conceptual model. | | | ■ |
| Implementation. | | ■ | |
| Verification and validation. | | | ■ |
| Maintenance. | | | ■ |

## IV. RESULTS AND TESTS

Among the benefits of using ontologies are: the classification of the terms of the logs, relationships, inferences, formalization, reuse, sharing, among others, we will show some gains in the context of research.

To prove the improvements in research in the generated logs, was used the ontology described in the previous sections to analyze the logs. In addition, the ontology is necessary to use a query language, called SPARQL [18], which since January 2008 is recommended as the standard by the W3C Consortium [14] [19].

This language allows querying ontologies through clauses, which can combine several classes at the same time

 ISBN: 978-1-61208-165-6 

and also make filters in the searches. A compiler for SPARQL [18] is already integrated in the Protégé tool [20].

To strengthen the evidence of the improvements in research involving the use of ontology in comparison with the traditional keyword searches, see below some situations:

1. A security auditor must analyze the log events that occurred in the 10th day of a month until the 10th day of the following month, that has the ip address range of 192.168.0.13 to 192.168.0.97 with destination ports 3306 and 443, and is on schedule from 21:30 to 24:00 h.
2. A security auditor wishes to know what IP address or group of addresses generated more attacks and in what times.

Considering the situations above, we have one, between two ways to proceed:

1. Search with the use of Ontology: It would be enough to use the select command with the filter clause containing the classes mentioned above, and be answered with only the desired data to analyze.

   It could also create a search interface using the language SPARQL query language or another, to prevent the auditor to need to know and type the commands queries.

2. Searches without the use of ontologies :

   Quite difficult to be generated because that for the auditor to make the analysis he wants, he would first have to read many logs manually separating by keyword and then make de co-relation, with the risk of losing data, since only the search for information, he will be ignoring the other data you want.

   Below are the consult solutions in SPARQL to the situations described above:

```
SELECT ?ID ?DestinationIP ?DestinationPort ?SourceIP
?SourcePort ?Timestamp ?Cookie ?mod_security-message
WHERE { ?ID rdf:type :ID . ?DestinatioIP rdf:type
:DestinationIP .
?DestinationPort rdf:type :DestinationPort . ?SourceIP
rdf:type :SourceIP . ?SourcePort rdf:type :SourcePort .
?Timestamp rdf:type :Timestamp . ?Cookie rdf:type
:Cookie . ?mod_security-message rdf:type :mod_security-
message
FILTER((?TimeStamp > xsd:date(“2010-07-09”) &&
?TimeStamp < xsd:date(“2010-08-11”) &&
(?DestinationIP > 192168012 && ?DestinationIP <
192168098) && (?DestinationPort = 3306 ||
?DestinationPort = 443) && (?TimeStamp >
xsd:time(“21:29:00”) && ?TimeStamp <
xsd:time(“24:00:01”))) }
```

Figure 4. SPARQL [18] consult in the Ontology, Situation 1

```
SELECT ?ID ?SourcerIP ?DestinationIP ?Timestamp
WHERE { ?ID rdf:type :ID . ?SourceIP rdf:type
:SourceIP . ?DestinationIP rdf:type :DestinationIP .
?Timestamp rdf:type :Timestamp GROUP BY
?SourceIP } ORDER BY ASC(?SourceIP)
```

Figure 5. SPARQL Consult in the Ontology, situation 2

In this sense, the implemented ontology fulfilled its role very well, according to what was previously planned, the searches were carried out in a simple way in the ontology producing the most interesting and visible results in comparison with the traditional consultations using only key words, obtaining better results for event logs identification.

It is seen that with the approach proposed in this paper, the activity log analysis was made simple and independent of the complexity of the log and the generated data volumes allowing the realization of co-relations between events more efficiently

## V. CONCLUSION AND FUTURE WORKS

This study aimed to take the first steps in using ontologies for analysis of security logs. For that purpose the logs generated by the program ModSecurity were initially used. As a starting point the log that was generated by this tool on a web server of CESAR (Center for Advanced Studies and Systems of Recife) was used. The ontology modeling was accomplished from the understanding of the logs the model the ontology.

The performed tests proved that there was an easier log interpretation and analysis, allowing the performing of more complex consultations and the implementation of co-relation of events very effectively.

Finally, we proved that the demand for log audits that use ontologies is very large, for the tools and current research procedure is very limited, constituting a critical point in analyzing logs. In this context, this work was made to contribute to the attending of this demand.

## ACKNOWLEDGMENT

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES http://www.ines.org.br), funded by CNPq and FACEPE, grants 573964/2008-4 and APQ-1037-1.03/08.

## REFERENCES

[1] D. Allemang and J. Hendler, Semantic Web for the Working Ontologist, Effective Modeling. in: RDFS and OWL, Cambridge, Morgan Kaufmann, 2008.

[2] M. B. Almeida and M. P. Bax, Uma Visão Geral sobre Ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção in Cin:Inf., Brasilia, 2003.

[3] A. Grigoris and V. H. Frank, A Semantic Web Primer, Second Edition. London, The Mit Press, 2008.

[4] T. Berness-lee, J. Hendler, and O. Lassila, The Semantic Web, Scientific American Publishing, New York, 2001.

[5] A. Brandão, A. R. A. Franco, F. Lucena and C. J. Pereira, Uma Introdução à Engenharia de Ontologias no contexto da Web Semântica, Rio de Janeiro, 2002.

[6] K. Breitnam., Web Semântica, a Internet do Futuro. Rio de Janeiro, LTC Publishing, 2005.

[7] R Cannings, D. H. Lackey, and H. Zane. Hacking Exposed™ WEB 2.0: Web 2.0 Security Secrets And Solutions. New York: Mc Graw Hill, 2008. - DOI: 10.1036/0071494618

[8] M. C. Silveira. Um estudo sobre XML, Ontologias e RDF(S),http://www.inf.pucrs.br/~mchaves/pg_portugues/tc/paperxml .pdf, Acessed on 01/10/2010.

[9] T. Gruber and R.Toward, principles for the design of ontologies used for knowledge sharing. In Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, 1996.

[10] P. N. Guarino and A. Roberto, Formal ontology, conceptual analysis and knowledge representation. International Journal of Human-Computer Studies, Volume 43 Issue 5-6, Nov./Dec. 1995

[11] R. Kimball and R. Merz, The Data Webhouse Toolkit, New York, John Wiley and Sons, Inc, Wiley Computer Publishing 2000.

[12] M. Lytras, D. Damiani, P. Ernesto and Patricia O. WEB 2.0 The Business Model. New York: Springer, 2009. ISBN-13: 978-0-387-85894-4

[13] D. L. Mcguinness and F. V. Harmelen. OWL Web Ontology Language , http://www.w3.org/TR/owl-guide/, Accessed on 10/10/2011

[14] Overview. W3C World Wide Web Consortium (http://www.w3.org/TR/owl-features/), 2004. Accessed on 08/16/2011

[15] Modsecurity. Breach, ModSecurity 2 Data Formats, 2009, Copyright © 2004-2009 Breach Security, Inc. (http://www.breach.com). Accessed on 08/16/2011

[16] N.F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Create Your First Ontology , http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf, Acceed on 10/10/2011

[17] Owasp, 2008, owasp testing guide 2008 v3.0.: http://www.owasp.org/index.php/category:owasp_testing_project Accessed on 08/16/2011

[18] E. Prud'hommeaux and A. Seaborne, SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/ , Acceed on 10/10/2011

[19] W3C- World Wide Web Consortium (http://www.w3.org/TR/rdf-sparql-query/). Accessed on 08/16/2011.

[20] Protégé; Ontology Editor and Knowledge Acquisition System (http://protege.stanford.edu/>). Accessed on 08/16/2011

[21] R. Studer, et al., Situation and Perspective of Knowledge Engineering, Stanford University, http://infolab.stanford.edu/~stefan/paper/2000/ios_2000.pdf, Accessed 10/10/2011 .

[22] D. Stuttard and M. Pinto, The Web Application Hacker's HandBook:Discovering and Exploiting Security Flaws, 2008.

[23] Us-cert - technical cyber security alerts, 2009. http://www.us-cert.gov/cas/techalerts/ . Accessed on 08/16/2011

[24] W3C, 2010, http://www.w3.org/. Accessed on 08/16/2011.

[25] Stanford Jambalaya plug-in, http://protege.stanford.edu/plugins/jambalaya/jambalaya-simple-backup.htm , Accessed on 10/10/2011

 ISBN: 978-1-61208-165-6 