# Web Analyzing Traffic Challenge: Description and Results

Chedy Raïssi[1,2], Johan Brissaud[3],
Gérard Dray[2], Pascal Poncelet[2], Mathieu Roche[1], and Maguelonne Teisseire[1]

[1] LIRMM - UMR 5506, CNRS, Univ. Montpellier 2, France
[2] LGI2P, EMA Site EERIE Parc Scientifique Georges Besse, Nîmes, France
[3] BEE WARE Company, 210, Avenue Frederic Joliot, 13852 Aix-en-Provence, France
raissi@lirmm.fr,jbrissaud@bee-ware.net
{gerard.dray,pascal.poncelet}@ema.fr,{mroche,teisseire}@lirmm.fr

**Abstract.** This paper describes the Web Analyzing Traffic Challenge (Discovery Challenge of ECML/PKDD'07) and the results. Using the data from query logs it is possible to recognize an attack and define its class. Then the aim of this challenge is the filtering of attacks in Web traffic.

**Key words:** attack detection, classification.

## 1 Introduction

The number of computer attacks carried out grows in tandem with the web. According to the National Institute of Standards and Technology[4], American companies as early as 2004 suffered losses of up to 59.6 billion dollars following IT attacks. Considering the number of IT systems now deployed, intrusion detection is a significant research area for the purpose of assessing and forecasting system attacks as early as possible.

The OSI Model (*Open Systems Interconnection Basic Reference Model*) is usually represented by a diagram showing a column composed of stacked rectangular shapes, each one symbolizing a layer of the model. However, in reality the seventh layer is much wider and more diverse than the layers below it. This application layer is definitely the biggest, widest, and most complex of all. It contains more than just protocols and parameters, and is made up of languages, scripts, libraries and human concepts, etc. As a consequence, the OSI Model observed from a security perspective makes the diagram take on a reversed pyramid shape. So the higher the layers, the richer and more diverse is their content, which means they are also more complex to secure.

Trying to filter application traffic as diverse and dynamic as Web traffic can quickly bring awareness of the existence of several strong constraints and the necessity to fulfil specific requirements such as:

[4] http://www.nist.gov.

- **Unknown attack detection**: A major consequence of application diversity is that the potential for vulnerabilities is infinite. Experience has already revealed that the vast majority of application attacks consist in the unknown variety.
- **False positives**: Considering the richness and diversity of this domain, and seeing that the threshold of acceptance is user-dependent, avoiding and eliminating False Positives are critical issues when analyzing the application layer.
- **Ambiguous queries**: When looking at existing applications it becomes quickly obvious that they harbour weaknesses or vulnerabilities. Traffic addressing these resources will then appear to carry weaknesses, but cannot be blocked without stopping the application.
- **Abnormal behaviour detection**: Attacks are not the only danger prevalent. Securing Web traffic is a more complex task than mere intrusion prevention. There are various other types of requests that require supervision.

## 2 Main objectives

The issue being addressed by this challenge is the filtering of application attacks in Web traffic. This is a complex matter because of diversity in attack purposes and means, the quantity of data involved and technological shifts. Application attacks can be multi-class and undergo constant change. They do however maintain some distinguishing features (escaping, bypassing, keywords matching external entities, etc.).

To achieve this aim data sources available from HTTP query logs are used. Using this data we can not only recognize an attack but also define which class it belongs to. Participants would have to start with an HTTP query in context and deduce which class it belongs to and what is its level of relevance.

To efficiently address this issue, we divided the challenge in the two following tasks:

1. **Task 1: Multi-class and contextual classification**
   We have to be able to classify queries that may belong to different classes, and we have to do so according to context. A query in attack form that is not dangerous because made in the wrong context has to be properly labelled. The amount of data to process being considerable due to traffic density, any real-world classification application should be able to process the queries extremely quickly. Participants are judged on the classification performance but also on the time performance of their algorithm implementations.

2. **Task 2: Isolation of the attack pattern**
   We should be able to pinpoint in an attack query the shortest chain that conveys the attack[5].

[5] In this paper, this task will not be developed.

## 3 Dataset composition

The attacks of the dataset look like real attacks but have no chance to succeed because they are constructed blindly and do not target the correct entities. One sample can eventually target several classes (SQL injection, Command execution, etc.). Each example is totally independent of the others.

The data set are defined in XML (portable and standard format). Each sample is identified by a unique id, and contains the three following major parts: *Context* (stands for the environment in which the query is run), *Class* (describes how this sample is classified by an expert) and the *description* of the query itself.

- **Context:** It contains the following attributes:
  1. Operating system running on the Web Server (UNIX, WINDOWS, UNKNOWN).
  2. HTTP Server targeted by the request (APACHE, MIIS, UNKNOWN).
  3. Is the XPATH technology understood by the server? (TRUE, FALSE, UNKNOWN)
  4. Is there an LDAP database on the Web Server? (TRUE, FALSE, UNKNOWN)
  5. Is there an SQL database on the Web Server? (TRUE, FALSE, UNKNOWN)

- **Class:** It lists the different subdivision levels of HTTP query categorization (and how they are represented in the context part of the dataset).
  The ”type” element indicates which class this request belongs to:
  1. Normal Query (Valid)
  2. Cross-Site Scripting (XSS)
  3. SQL Injection (SqlInjection)
  4. LDAP Injection (LdapInjection)
  5. XPATH Injection (XPathInjection)
  6. Path Traversal (PathTransversal)
  7. Command Execution (OsCommanding)
  8. SSI Attacks (SSI)

  Moreover, a flag is added explaining whether a query is within the assigned context or not (element ”inContext” taking two values: TRUE or FALSE).

  Another element (”attackIntervall”) indicates where the attack is located on the query description. This element begins with the name of the element where the attack is located (uri, query, body, header) followed by ”:”. Thereafter the interval considered as an attack is specified. For headers, we also indicate the header name where the attack is located. The interval begins from the beginning of the considered header value.

- **Query:** It is described with its different components:
  1. Method
  2. Protocol
  3. Uri
  4. Query
  5. Headers
  6. Body

## 4 Evaluation Criterion

For this challenge, precision and recall (see formulae 1 and 2) are the basic measures used in evaluating search strategies.

$$Precision = \frac{\text{number of relevant attacks detected}}{\text{number of attacks detected}} \tag{1}$$

$$Recall = \frac{\text{number of relevant attacks detected}}{\text{number of relevant attacks}} \tag{2}$$

*Fmeasure* combines recall and precision in a single efficiency measure (see formula 3).

$$Fmeasure(\beta) = \frac{(\beta^2 + 1) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{3}$$

For the challenge, *Fmeasure* is calculated with $\beta = 1$, meaning that the same weight is given to precision and recall.

## 5 Results

Only two challengers sent a submission for this challenge. A discussion to explain this number of participants will be developed in section 6.

The evaluation of the challengers is based on the "*test*" dataset available at the end of June 2007. The "*learning*" dataset was available on April 15th, 2007. The description of the different datasets is given in the table 1.

The table 2 presents if the attacks are correctly detected. The challenger 1 provides the best result based on the *Fmeasure*[6]. For the two challengers the precision is better than the recall. Let us note that the *Fmeasure* obtained

[6] A new submission of the challenger 1 has be sent when the deadline was passed. This new submission (non-official submission) gives again a better result with a *Fmeasure* at 0.9345.

| | **Test dataset** | **Learning dataset** |
|---|---|---|
| **Number of examples** | 70,000 | 50,000 |
| **Ratio of attacks** | 60% | 70% |
| **Ratio of attacks in context** | 75% | 85% |
| **Type of Attacks** | | |
| post | 15% | 12% |
| get | 58% | 70% |
| cookie en post | 7% | 4% |
| header en post | 5% | 3% |
| cookie en get | 8% | 6% |
| header en get | 7% | 5% |

**Table 1.** Description of the two datasets used for the challenge

without context is low for the submission of the two challengers: 0.4826 for the challenger 1[7] and 0.1728 for the challenger 2.

The table 3 presents if the classes of attacks are correctly detected by the challengers. This table shows that the challenger 1 obtained the highest values of *Fmeasure* for all the classes of attacks. The results show that some types of attacks are quite easy to detect (*e.g.* XSS, XPathInjection, LdapInjection), while other ones are very difficult to find out (*e.g.* OsCommanding, SSI)[8]. We have to note that, for example, an instance of the SSI class is difficult to detect since this type of attack is usually a multi-class attack.

The 'Valid' class (*i.e.* normal queries) was easy to detect for the two challengers: *Fmeasure* at 0.8793 for the challenger 1 and 0.6900 for the challenger 2.

Finally, we can observe that multi-class attacks were not considered by all the challengers.

| | *Precision* | *Recall* | *Fmeasure* |
|---|---|---|---|
| **Challenger 1: LIFO<br>Orléans,<br>France** | 0.8229 | 0.7807 | 0.8012 |
| **Challenger 2: Department<br>of Informatics<br>Athens, Greece** | 0.4976 | 0.4721 | 0.4845 |

**Table 2.** The *Fmeasure* of the challengers

[7] However the value of the *Fmeasure* with the non-official submission of the challenger 1 is excellent: 0.9824.

[8] These conclusions are based on the most significant results of the challenger 1.

| **Group** | **Challenger 1** | **Challenger 2** |
|---|---|---|
| Valid | 0.8793 | 0.6900 |
| OsCommanding | 0.4093 | 0.0598 |
| SSI | 0.4216 | 0.0307 |
| SqlInjection | 0.6205 | 0.0358 |
| PathTransversal | 0.6819 | 0.0409 |
| XSS | 0.7597 | 0.0394 |
| XPathInjection | 0.7405 | 0.0487 |
| LdapInjection | 0.8811 | 0.0281 |

**Table 3.** The *Fmeasure* of the classes of attacks

## 6 Conclusion

As we have said in the introduction section, the problem of detecting intrusion in Web traffic is far away from trivial. This contest aims at providing different approaches for extracting such intrusions.

The contest is now finished and we have two observations. At the very beginning of the contest, 25 researchers from different countries such as China, Finland, Indonesia, Korea, Australia, Pakistan, Italy would like to apply their techniques (usually classification techniques) for learning intrusions. At the end, only two challengers sent their results and we would like to acknowledge them. We have asked the other challengers to know why they did not submit their results. The response was mainly they did not have a lot of knowledge about detection intrusion and when they tried to apply "traditional approaches" in order to characterize intrusion, their results were not good enough. We believe that this observation is very important in a data mining context since more and more we must integrate the expert earlier in the knowledge discovery process. The second observation is related to the problem of detection intrusion itself. When we proposed the contest, we knew that this problem was a very difficult topic but thanks to the results of the challengers it is clear now that detection intrusion becomes more and more a new hot topic since we have to deal not only with supervised, unsupervised classification but also with real time data mining.