Journal of Applied Computing Research, 1(2):111-120
July-December 2011

Anomaly-based Techniques for Web Attacks Detection

Bruno Augusti Mozzaquatro, Renato Preigschadt de Azevedo, Raul Ceretta Nunes, Alice de Jesus Kozakevicius

Universidade Federal de Santa Maria, Centro de Tecnologia. Av. Roraima, 1000, 97105-900, Santa Maria, RS, Brasil.
brunomozza@inf.ufsm.br, renato.azevedo@inf.ufsm.br, ceretta@inf.ufsm.br, alicek@smail.ufsm.br

Cristian Cappo, Christian Schaerer

Polytechnic School, National University of Asuncion. P.O. Box: 2111 SL, San Lorenzo, Paraguay.
ccappo@pol.una.py, cschaer@pol.una.py

Abstract: The widespread use of the Internet comes accompanied with severe threats for web applications security. Intrusion Detection Systems (IDS) have been considered to deal with the diversity and complexity of web attacks. In this context, this work proposes an algorithm for web attack detection, exploring an anomaly-based technique: the wavelet transform. The proposed algorithm analyzes anomalies within variations on characters frequencies in web requests. Experimental results show high rates of detection without false positive occurrences.

Keywords: Web Attacks, Anomaly Detection, Wavelet Transform, Web Applications.

Introduction

Internet has become an interconnection system between computer networks and web applications, that serves billions of users and mainly provides communication and information dissemination.

A consequence of massive usage and expansion of web applications is the continual identification and appearance of new vulnerabilities (OWASP, 2010), that are explored with the aim of violating the security and prejudicing the availability of applications (Fonseca *et al.*, 2010).

In this context, Intrusion Detection Systems (IDS) are necessary to guarantee information security. These systems can be distinguished according to two different approaches: signature- and anomaly-based.

Signature-based IDS identify attacks through a group of information containing previously defined patterns, i.e. signatures. (Kruegel *et al.*, 2004). The advantage of this approach is to detect attacks with low rate of false positives. Nevertheless, only attacks with respect to known patterns are identified, demanding frequent signature updates in order to keep detection in reliable standards.

Alternatively, anomaly-based IDS identify attacks through the observation of behaviour variations in characteristics of the network (Chandola *et al.*, 2009). The main advantage of this approach is the ability to identify new attacks that are not necessarily associated to previously known patterns, avoiding the usage of signatures (Chandola *et al.*, 2009).

Web attacks are performed exploring many different contexts. Among them, attacks that manipulate web requests have received wide attention (OWASP, 2010) in the last years. These attacks explore vulnerabilities of parameter manipulations to change values of a web request (Álvares and Petrović, 2003). In this context, special characters are commonly inserted, resulting in serious degradation of integrity, authentication, availability and authorization of internet services.

Several works explore anomaly-based methodologies to determine anomalies restrained in the web traffic (Kruegel and Vigna,

2003; Wang and Stolfo, 2004; Estevez-Tapiador *et al.,* 2005), to mention a few examples. The most widespread techniques are those based on n-grams (Wang and Stolfo, 2004), deterministic finite automata (Ingham and Inoue, 2007), statistics information (Kruegel and Vigna, 2003) and Markov chains (Estevez-Tapiador *et al.,* 2005).

Signal processing techniques have been recently applied to anomalies detection in computer networks (de Azevedo *et al.,* 2011; Lu and Ghorbani, 2009), providing a promising toolset, with big potential for anomalies recognition.

This work explores a new approach for web attacks detection also based on signal processing techniques, considering well-established wavelets methods (Stollnitz *et al.,* 1995). In the initial study, presented in Cappo *et al.* (2009), the usage of the Haar wavelet transform for attack detection on web application was investigated. Now in this work the application of the same discrete wavelet transform is considered to analyze the characters frequency distribution of web requests. The wavelet transform enables the extraction of information on different resolution levels and discards the training phase. This is an alternative to techniques based on machine learning (Kruegel and Vigna, 2003; Kiani *et al.,* 2008), which are the most used approach to detect web attacks.

This paper is organized as follows: Section II briefly summarizes the most commonly techniques used to detect anomalies on web requests. Section III presents aspects with respect to web attacks. In Section IV the one dimensional Haar wavelet transform is introduced, along with the corresponding bi-dimensional version. Section V contains the approach for detecting web attacks using the Haar wavelet transform. The validation set of experiments is presented in Section VI. Finally, the conclusion of the work is presented in Section VII.

Anomaly-based Detection Techniques for Web Traffic

In this section a brief review about anomalies detection in the context of web traffic security is presented. The most common techniques are based on machine learning approaches, which explore anomalies in web traffic by a training phase, to model the ordinary behaviour of a system (Kruegel *et al.*, 2004). Three commonly used approaches for the learning phase from web requests: packet payload, extracted tokens and character distribution analysis.

Wang and Stolfo (2004) observed packet payload to model byte (n-gram) frequency distributions. A clustering algorithm was then applied to merge similar distributions by port and stream directions. The detection phase consisted in comparing the cluster distances to a certain threshold.

Ingham and Inoue (2007) tokenized HTTP requests. They employed Deterministic Finite Automata (DFA) machine induction combined with rules and heuristics for modelling and detecting attacks on traces of tokens with benign attacks. Ingham and Inoue (2007) compared the DFA and n-grams performances, signalling that the first approach was more efficient than the former one.

Character distribution was also explored by the usage of statistic techniques. Kruegel and Vigna (2003) modelled the relative frequency of the characters and arranged into six arbitrarily chosen groups. Finally the Pearson's chi-squared statistical test was applied to detect attacks. Kiani *et al*. (2008) also utilized the test. However, in their work cumulative frequency of the characters in the training phase has been considered, and the characters from Yate's correction method were grouped. Another group clustering was proposed by Sriraghavan and Lucchese (2008), where three groups of characters were formed according to type as alphabetic, numeric, and special characters.

Another behaviour modelling was proposed by Corona *et al.* (2009). The authors, using Markov's hidden models for parameter analysis of the web traffic communication (GET and POST), demonstrated good results in the detection of the SQL Injection and Cross Site-scripting (XSS) attacks. Markov's hidden models were also employed by Airu and Giacinto (2010) to explore the HTTP payload byte sequence for analysing attacks between web application and the web server.

From the authors' knowledge, signal processing techniques have been applied for detecting network anomalies, but not to explore HTTP requests, as proposed in this work.

Web Attacks

Within the reported attacks in the last years, a representative amount was related to web applications (OWASP, 2010). This scenario is a consequence from the increasing utilization of these systems on the web and also from

failures on the software development phase, since programmers commonly skip some security issues on the software life cycle design (Wagner *et al*., 2011).

From OWASP (2010), vulnerabilities related to content injections are on the top of the technical vulnerabilities list. The amount of injection based web attacks, as Code Injection and Cross Site-scripting (XSS), are consequently increasing and causing damages for many organizations.

This work looks at web attacks that insert malicious content by changing the character frequency on web requests. Figure 1 shows an example of this type of attack, where characters "." and "/" were injected on a GET request.

Wavelet Transform

The wavelet transform decomposes a signal on many resolution levels with the goal to highlight and recognize relevant information on the signal. The wavelet decomposition is hierarchical and is formed by a coarsest level of representation and many refinement levels of information (Stollnitz *et al*., 1995). The coarsest level is associated to the scaling coefficients, which represent the main trend of the signal. The refinement levels are composed by the wavelet coefficients, also called details, which are complementary information of the coarse representation and are needed to reconstruct the original data. In signal analysis applications they are manipulated in order to allow noise removal, compression (Kozakevicius *et al*., 2005), or other feature extractions from the analyzed data.

There is a diversity of wavelet families, and the Haar wavelet is the one adopted in this work, mainly due to the facility to preserve the location of sudden variations within data (Mallat, 2008). Furthermore, the Haar wavelet transform dismisses special treatment for the signal borders and its implementation is straightforward. These are important features when considering discrete input data (with finite dimension).

One-dimensional Wavelet

The application of the Haar wavelet transform involves high-pass (H) and low-pass filters (L), which are convolved with the signal to obtain the transformation coefficients. The application for one decomposition level of the pyramidal algorithm (Mallat, 2008), provides two groups of coefficients: a set of scaling, $c_{j-1,i}$, and a set of wavelets coefficients, $d_{j-1,i}$. The coefficients $c_{j-1,i}$ are associated to lower frequencies, keeping the mean information of the signal at each level. The wavelets coefficients (or details) $d_{j-1,i}$ are associated to higher frequencies within the signal. Their computation considered as input data a vector with values on the finest resolution level. Therefore the one scale decomposition is done as follow:

$$c_{j-1,i} = \sum_{k=0}^{2N-1} L_k c_{j,2i+k},\ i = 0,\ldots,M_{j-1}-1 \quad (1)$$

$$d_{j-1,i} = \sum_{k=0}^{2N-1} H_k c_{j,2i+k},\ i = 0,\ldots,M_{j-1}-1 \quad (2)$$

Note that the difference between them is basically due to the considered filter type. One of the most important properties of the wavelet transform is its capability to identify jumps and peaks on data through the set of details (Grané and Veiga, 2009). This property allows the application of wavelet transform for abrupt changes identification and localization, noise removal and signal compression. In this paper we explore the algorithms presented in Stollnitz *et al*. (1995) for the alarm designing.

In Figure 2, an algorithm to compute the one dimensional wavelet transform (1DWT) is represented. In the input procedure the input data is processed until the coarsest decomposition level is achieved (line 8). On the level decomposition step (line 1), only data associated to mean values in each level is decomposed, generating a new set of means (denoted by C′[i] on Figure 2) and a new set of details (denoted by $C'\left[\frac{h}{2}+i\right]$), both with half size of the original block. One aspect to observe is that only blocks associated to means (scaling coefficients) are further decomposed when the standard wavelet transform is con-

```
121.112.21.23 " GET /index.php?v1=20112320&v2=admin HTTP/1.0" 200
154.127.15.33 " GET /index.php?v1=../../../../../../../../etc/passwd%00 HTTP/1.0" 200
163.12.165.12 " GET /index.php?v1=20112523&v2=user HTTP/1.0" 200
```

Figure 1. Example of web attack with malicious code injection.

```
1  procedure DecompositionStep(C: array [1..h] of reals)
2      for i ← 1 to h/2 do
3          C'[i] ← (C[2i − 1] + C[2i])/√2
4          C'[h/2 + i] ← (C[2i − 1] − C[2i])/√2
5      end for
6      C ← C'
7  end procedure

8  procedure Decomposition(C: array [1..h] of reals)
9      while h > 1 do
10         DecompositionStep(C[1..h])
11         h ← h/2
12     end while
13 end procedure
```

Figure 2. Algorithm for 1DWT decomposition (Adapted from Stollnitz et al., 1995).

sidered. Another aspect is that in the proposed algorithm all decompositions are stored on the same array *C*, instead of using two different arrays as suggested by pyramidal algorithm equations. The level of transform corresponds to the number of interactions on decomposition procedure. Each level of Haar wavelet transform is performed here in a dyadic form, i.e. the input data is always split in sets with half size. At least two values must be kept on the coarsest level to run this algorithm.

Bi-dimensional Wavelet

The bi-dimensional wavelet transform (2DWT) allows data analysis and identification of variations with respect to lines and columns of two dimensional input data.

The bi-dimensional wavelet transform (2DWT) allows data analysis and their variations recognition with respect to lines and columns of a bi-dimensional input data. In fact, when processing the decomposition with respect to lines and to columns, another set of variations is captured as a by-product, which here we interpret as being in the diagonal direction. This set is also a linear combination to lines and columns and is associated again to high frequencies of the signal. In other words the 2DWT generates three sets of wavelet coefficients (sometimes called as sub-bands) for representing variations in the input data, differently of the 1DWT, which generates only one set of wavelet coefficients.

In practice, there are many ways to perform this transformation (Stollnitz *et al.*, 1995), since the order which the many 1DWT are performed implies in a different system for data representation. One simple form is to compute initially the 1DWT for all lines of the input matrix and then compute the 1DWT for all columns of the resulting matrix. According to the classification presented in Stollnitz *et al.* (1995) this is a Non-Standard formulation, which is presented in Figure 3.

In Figure 4 the effect of applying one complete level of the 2DWT is presented for one sample image. First in panel (a) the intermediate stage after the 1DWT by lines is shown, where data is decomposed in scaling and wavelet coefficients. In panel (b) the final step is shown, when all three sets of wavelet coefficients are finally obtained as well as the set of the scaling coefficients (one level coarser).

As done for the one dimensional transform, again in a free interpretation, the scaling coefficients are seen as a block formed by means, (means over means), and the three wavelet coefficient sub-bands could be viewed as details of means (*dc*), means of details (*cd*) and details of details (*dd*), as a try to identify the origin of the wavelet coefficient generated during the transformation process.

In the application proposed here in this paper the lines of input matrix are associated to ASCII characters and web requests to the columns. This paper explores the 2WDT to accomplish analysis of the character frequencies variation kept in the web requests in more than one direction, within a request and among requests.

In order to perform more levels of the 2DWT, only the set of the scaling coefficients is analyzed, being considered as the new in-

```
1  procedure NonstandardDecomposition(C: array [1..h, 1..h] of reals)
2      while h > 1 do
3          for row ← 1 to h do
4              DecompositionStep(C[row, 1..h])
5          end for
6          for col ← 1 to h do
7              DecompositionStep(C[1..h, col])
8          end for
9          h ← h/2
10     end while
11 end procedure
```

Figure 3. Nonstandard 2DWT algorithm: 1DWT applied for all lines and then applied for all columns (Adapted from Stollnitz et al., 1995).

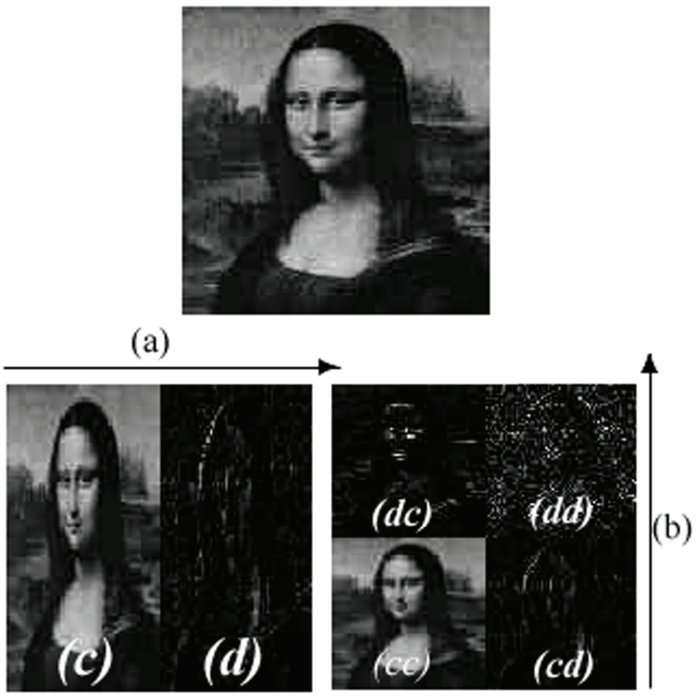

Figure 4. NonStandard decomposition of an image (Adapted from Stollnitz et al., 1995).

put data, which will again be decomposed in four new coefficient sets (sub-bands) according to the same algorithm. In each stage of the transform, the wavelets sub-bands remain unaltered.

Threshold

After the application of the bi-dimensional wavelet transform, the signal is represented by four different sub-bands *cc*, *cd*, *dc* and *dd*. The sub-bands associated to wavelet coefficients represent the data variations captured in three different directions.

According to one of the main properties of wavelet decompositions, each time the wavelet coefficients are small, the corresponding coarse information (mean coefficients) can accurately represent the original data. Therefore the determination of the size (significance) of the wavelet coefficients is a tool to discard irrelevant data from the representation or to detect relevant variations within the analyzed data. Thus, the criteria to discard wavelet coefficients are basically based on comparing the coefficients from sub-bands *cd*, *dc* and *dd* with some reference value, called threshold value. One of the most utilized values in filtering

applications is the universal threshold proposed by Donoho and Iain (1995) that is determined by $\lambda = \sigma\sqrt{2\log(N)}$, where σ is the standard deviation of the coefficients, **N** is the number of samples at each sub-band.

When the threshold value is consistently chosen, the threshold operation does not destroy fundamental properties of the signal (Donoho and Iain, 1995). So, the main analysis question is how to determine the most coherent threshold, capable to separate relevant and irrelevant information, depending on the application properties.

Donoho and Iain (1995) proposed two ways to filter wavelet coefficients: one by applying a Hard threshold (Equation 3) and the other by applying a Soft threshold (Equation 4). The Hard threshold discards wavelet coefficients without modifying coefficients bigger than the threshold value (those judged as significant) and the Soft threshold provides smoothness the entire set of wavelet coefficients.

$$d(k) = \begin{cases} 0 & , \ |d(k)| < \lambda, \\ d(k) & , \ |d(k)| \geq \lambda, \end{cases} \quad (3)$$

$$d(k) = \begin{cases} 0 & , |d(k)| < \lambda, \\ d(k) - \lambda & , |d(k)| \geq \lambda \wedge d(k) \geq 0, \\ d(k) + \lambda & , |d(k)| \geq \lambda \wedge d(k) < 0. \end{cases} \quad (4)$$

In this work, the Hard threshold is employed, since the main information necessary for modelling the proposed attack algorithm detection is the significance of the wavelet coefficient instead of its absolute value.

For a detailed explanation about different threshold definitions, as well as different criteria for the threshold value choice, see (Donoho and Iain, 1995).

Wavelet based Algorithm for Web Attacks Detection

Anomalies can occur in web traffic communications, especially during a web attack occurrence, causing several inconveniences for clients and web servers. The web traffic can be analyzed on different ways, from packet amount to protocols usage or server requests, considering point-to-point streams or not. This work focuses on web attacks that generate anomalies in the web request character distribution, observed from the server side (requests from clients to server).

Attack-free web requests present two important features (Kruegel and Vigna, 2003) they have a regular structure with respect to characters frequency distribution and they commonly use readable characters. Thus, variations on the regular structure of the commonly used characters can indicate undesirable events, such as a possible attack. Since wavelet analysis is an efficient tool to detect variations, this capability is explored here for detecting web attacks that inject code or malicious content in web requests. The proposed wavelet-based algorithm analyzes the request characters frequency by using the two dimensional Haar wavelet transform.

The proposed algorithm has two data inputs: the requests sent to a specific web server and a constant correction factor ρ to adjust the universal threshold value. The requests are organized in a matrix defined by M*[n]*[*m*], where *n* is the number of ASCII characters (256). The value of *m* is the amount of requests that are considered in the wavelet analysis. In the matrix, each row represents a character and each column represents a request. Thus, the element M*[i]*[*j*] contains the amount of times the i^{th} ASCII character appears in request *j*.

Figure 5 presents the proposed algorithm. For the anomalies detection approach in this work, only one level of the 2DWT is performed, since the significant abrupt changes are associated to high frequencies wavelet coefficients. Through the transformation of M*[n]*[*m*], four sub-bands of coefficients are obtained: *cc*, *dc*, *cd* and *dd*. For each one of the wavelets sub-bands, a corresponding threshold value (λdc, λcd and λdd) is computed as the multiplication of the constant correction factor ρ and the universal threshold value for the sub-band. This correction of the universal threshold value provides more freedom in adjusting the truncation parameter in the filtering process since the Donoho threshold was planned considering image denoising and this paper threshold must be applied to highlight only the values that override the higher value of a data without attacks. The threshold strategy is then processed (Line 4-6).

After the hard threshold application, characters with the most significant variations for their occurrence distribution can be identified. Note that the wavelet-based algorithm enables peaks verification on the occurrence distributions among characters on the same requisition (M*[n]*[*m*] rows) and among characters of different requests (M*[n]*[*m*] columns). To decide whether the variation is an anomaly or not (Line 9 of the algorithm), the variation

Input : M : Matrix of Data
ρ : Constant Factor
Output: A : Set of attack positions
1 $A \leftarrow \emptyset$
2 $n \leftarrow |M|$
3 $(cc, dc, cd, dd) \leftarrow TW[M]$ [one level]
4 $\lambda_{cd} \leftarrow \rho \cdot \sigma_{cd} \sqrt{2 \log(\frac{n}{2})}$
5 $\lambda_{dc} \leftarrow \rho \cdot \sigma_{dc} \sqrt{2 \log(\frac{n}{2})}$
6 $\lambda_{dd} \leftarrow \rho \cdot \sigma_{dd} \sqrt{2 \log(\frac{n}{2})}$
7 **for** $i \leftarrow 1$ **to** $\frac{n}{2}$ **do**
8 **for** $j \leftarrow 1$ **to** $\frac{n}{2}$ **do**
9 **if** $(cd_{i,j} >= \lambda_{cd}$ *AND* $dc_{i,j} >= \lambda_{dc})$ *OR*
$(cd_{i,j} >= \lambda_{cd}$ *AND* $dd_{i,j} >= \lambda_{dd})$ *OR*
$(dc_{i,j} >= \lambda_{dc}$ *AND* $ddi, j >= \lambda_{dd})$ **then**
10 $A \longleftarrow A + (i, j)$
11 **end**
12 **end**
13 **end**
14 **return** A

Figure 5. Pseudocode of algorithm proposed.

detection between each two pairs of sub-bands *cd*, *dc* and *dd* is analysed. A peak on a character distribution is associated to a peak on the corresponding wavelet coefficient. Therefore, anomalous events are associated to significant coefficients, classified after the hard threshold process. Moreover, according to the proposed heuristic, anomaly on a characters distribution should be captured by at least two sub-bands of details to be considered an attack (Line 6).

Finally, another point to be highlighted is that the Haar wavelet transform is characterized as a simple and fast tool, specially because it's algorithmic formulation (Mallat, 2008). In practice, it is due to the number of operations being linearly proportional to the quantity of data contained in the input matrix, resulting in an algorithm of complexity O(*nm*).

Experiments and Results

Data collected from web application on the web server of the Polytechnic of the National University of Asuncion (FPUNA), Paraguay is considered to validate the proposed methodology. The data acquisition period was from January to March 2009, building 71 days of web traffic, captured from log files of web server. Altogether, a total amount of 59.248 web requests on the collecting phase was obtained.

For the validation process of any anomaly detection procedure, the capability of detecting true attacks has to be proved. Therefore, a data set with attacks inserted in known positions is considered for this verification.

The attacks detected on the FPUNA collected data set were manually removed by visual inspection, generating a so called "attack-free" signal, whose status was verified by using actualized signature-based IDS. Eight test data sets were built by the injection of 0, 1, 2, 3, 4, 5, 10 and 20 attacks in predefined places in the "attack-free" constructed signal. The data set with 0 attacks presents regularity among requests and low variability within the same character frequency by request. According to the description given on Section 3, the injected attacks were of types: directory transversal, cross site scripting and buffer overflow (the most popular attacks against web servers).

Based on the eight benchmark examples, the detection rates were computed, since the attacks location and distributions are previously known. For the experiments executed in this work, all event detections on unknown places are considered false positives. In other words, a false positive is characterized by the detection of an attack in a normal web request.

Table 1 presents the results of the detection when varying the number of attacks and the constant factor of universal threshold value. The algorithm was run considering the Hard

Table 1. Results obtained with proposed approach.

Attacks	# attacks detected with differents ρλ									
	1λ		2λ		3λ		4λ		5λ	
	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	15971	0	1939	0	7	0	1	0	0	0
1	15970	1	1934	1	8	1	2	1	0	1
2	15969	2	1934	2	7	2	1	2	0	2
3	15911	3	1934	3	7	2	1	3	0	3
4	15909	4	1934	4	7	4	1	4	0	4
5	15887	5	1874	5	3	5	1	5	0	5
10	15774	10	1940	10	7	10	1	10	0	10
20	15420	20	1876	20	3	20	1	20	0	20

FP = False Positive **TP** = True Positive

threshold approach and increasing the value of the threshold by multiplying it by factor ρ equal to 1, 2, 3, 4 and 5. The results show that for all data sets the number of false positives (FP) is very high when constant factor is 1 and 2 times the universal threshold λ. Otherwise, when the threshold overrides 3λ the number of false positives is small. The number is 0 when the adopted threshold is equal to 5λ.

It is important to notice that the determination of the threshold value is a key point in analysis using wavelet transforms associated to the Hard or Soft threshold strategies. Here the universal threshold value given by Donoho and Iain (1995) was considered as a possible reference choice for the cutting point, since it was an efficient choice for denoising processes when the signal is affected by White Gaussian noise. In other words, the Hard threshold with the universal threshold value is designed to be efficient under a big amount of variability. Thus, the experiment designed here shows that the universal threshold value is only one possible candidate to be considered as an alarm detector, and the correction factor plays an important role in fitting the cutting parameter according to the magnitude of the analyzed data, especially in the case of noise absence. Here five times the universal truncation value produced an efficient detector, since all the attacks were detected without occurrence of false positives. However, according to the web application, an analysis for the determination of the threshold value has to be performed. In this paper the FPUNA dataset was collected from a web application server and it has low variability and regularity between requests and characters frequencies.

The Table 2 compares our algorithm results with the results obtained by the Pearson chi-squared statistical χ^2 test. The χ^2 test is frequently used by detection techniques (Kruegel and Vigna, 2003; Kiani *et al.*, 2008) with the same proposal as presented in this work.

The χ^2 test makes use of a training phase and it was set to use a threshold of 10% higher than the maximum value obtained on the training phase. In the comparison, our algorithm has utilized the threshold of 5λ. The results obtained comparing of the proposed algorithm with the χ^2 show the wavelet-based algorithm has better results with respect to the number of false positives and correct detections.

For the measure the computational cost are performed various performance tests. The algorithm was implemented with ANSI C in a sequential version (i.e. using a single processor). The version of the Wavelet Transform algorithm is not in-line requiring an additional space of order of $O(nm)$. In the experiments the total memory required for the algorithm was approximately 640 KB.

The analysed dataset contains about 232 groups of 256 requests. The processing average time of 256 requests is presented for three different platforms in the Table 3. This processing time of complete dataset (FPUNA), with 59248 requests, considering the platforms presented was 1 second, 1.5 seconds and 2.5 seconds, respectively. Thus, the obtained performance in requests per second was approximately between 59000 and 24000 requests per second.

Conclusion

Internet became an essential component for the current society and numerous organi-

Table 2. Comparison of algorithm proposed with test Person χ^2

Algorithm	# attacks inserted in the data set FPUNA															
	0		1		2		3		4		5		10		20	
	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
χ^2	12	0	12	0	12	0	12	3	12	4	12	5	12	10	12	20
*	0	0	0	1	0	2	0	3	0	4	0	5	0	10	0	20

* Algorithm proposed (5λ) **FP** = False Positive **TP** = True Positive

Table 3. The computational cost for processing of the algorithm proposed.

Plataform	Average time in ms*
Intel Pentium 4 3.0 GHz Fedora – Linux Kernel 2.6 - gcc 4.3.0	3.225207
Intel Core 2 Duo 2.0 GHz Windows Vista 64 bits Cygwin - gcc 4.3.4	1.586207
Intel Xeon Quad-Core 2.4GHz Ubuntu Server – Linux Kernel 2.6 - gcc 4.5.2	0.769847

***ms** = milliseconds

zations perform daily operations through web applications. However, security of the information plugged in the web, dependability and reliability of the users should be guaranteed. Attacks damaging web applications or accessing restricted information can cause several problems for an organization and, consequently, for its users.

This paper proposed a new approach to detect web attacks by using signal-processing techniques. The performed experiments have demonstrated that the utilization of the wavelet transform associated to a threshold criteria allows web attack detection, since attacks associated of characters injection cause significant variation in the characters distribution within the requests.

Moreover, the issue related to the choice of the threshold value is also addressed. The rate of false positives was null when the correction factor ρ was considered bigger than 5, generating a threshold value 5 times the universal value of Donoho and Iain (1995). The Hard threshold procedure was applied for construction of the proposed alarm detector.

Using the properties of the wavelet transform it was possible to analyze the variation in the characters distribution in different directions, resulting in a high rate of correct web attacks detection. Further, as the wavelet approach is based on low complexity calculus and it does not consider the training phase, an efficient and fast approach for detecting injection-based web attacks is obtained.

For future works, a new set of experiments to demonstrate the use feasibility in real time applications will be designed. The proposed approach so far has demonstrated good detection rates for web attacks in off-line scenarios.

References

ÁLVAREZ, G.; PETROVIĆ, S. 2003. A new taxonomy of Web attacks suitable for efficient encoding. *Computers & Security*, **22**(5):435-449. http://dx.doi.org/10.1016/S0167-4048(03)00512-1

ARIU, D.; GIACINTO, G. 2010. HMMPayl: an application of HMM to the analysis of the HTTP Payload. *In:* WORKSHOP ON APPLICATIONS OF PATTERN ANALYSIS, Cumberland Lodge, 2010. *Proceedings...* Cumberland Lodge, WAPA. p. 81-87.

CAPPO, C.; NUNES, R.C.; SCHAERER, C. 2009. On using wavelets for detecting attacks to web-based applications. *In:* CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, XXXII, Cuiabá, 2009. *Proceedings...* Cuiabá, CNMAC, p. 1040-1041.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys*, **41**(3):15-58. http://dx.doi.org/10.1145/1541880.1541882

CORONA, I.; ARIU, D.; GIACINTO, G. 2009. HMM-Web: A Framework for the Detection of Attacks Against Web Applications. *In:* IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 9, Dresden, 2009. *Proceedings...* Dresden, ICC. p. 1-6. http://dx.doi.org/10.1109/ICC.2009.5199054

DE AZEVEDO, R.P.; MOZZAQUATRO, B.; CAPPO, C.; NUNES, R.C.; SCHAERER, C.E.; KOZAKEVICIUS, A. 2011. A Bidimensional Wavelet Transform based Algorithm for DoS Attack Detection. *In:* LATIN-AMERICAN SYMPOSIUM ON DEPENDABLE COMPUTING, 5, São José dos Campos, 2011. *Proceedings...* São José dos Campos, LADC. **1**:1-2.

DONOHO, L.D.; IAIN, M.J. 1995. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association.* **90**(432):1200-1224. http://dx.doi.org/10.2307/2291512

ESTEVEZ-TAPIADOR, J.M.; GARCIA-TEODORO, P.; DIAZ-VERDEJO, J.E. 2005. Detection of Web-Based Attacks through Markovian Protocol Parsing. *In:* IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS, 10, Madrid, 2005. *Proceedings...* Madrid, ISCC. p. 457-462. http://dx.doi.org/10.1109/ISCC.2005.51

FONSECA, J.; VIEIRA, M.; MADEIRA, H. 2010. The Web Attacker Perspective - A Field Study. *In:* INTERNATIONAL SYMPOSIUM ON SOFTWARE RELIABILITY ENGINEERING, 21, San Jose, 2010. *Proceedings...* San Jose, ISSRE. p. 299-308. http://dx.doi.org/10.1109/ISSRE.2010.21

GRANÉ, A.; VEIGA, H. 2009. Wavelet-based detection of outliers in volatility models. *In:* STATISTICS AND ECONOMETRICS WORKING PAPERS. Madrid, 2009. *Proceedings...* Madrid, p. 01-22.

INGHAM, K.L.; INOUE, H. 2007. Comparing Anomaly Detection Techniques for HTTP. *In:* INTERNATIONAL CONFERENCE ON RECENT ADVANCES IN INTRUSION DETECTION, 10, Gold Goast, 2007. *Proceedings...* Gold Goast, RAID. p. 42-62. http://dx.doi.org/10.1007/978-3-540-74320-0_3

KIANI, M.; CLARK, A.; MOHAY, G. 2008. Evaluation of Anomaly Based Character Distribution Models in the Detection of SQL Injection Attacks. *In:* INTERNATIONAL CONFERENCE ON AVAILABILITY, RELIABILITY AND SECURITY, 3, Barcelona, 2008. *Proceedings...* Barcelona, ARES. p. 47-55. http://dx.doi.org/10.1109/ARES.2008.123

KOZAKEVICIUS, A.J.; RODRIGUES, C.R.; NUNES, R.C.; GUERRA FILHO, R. 2005. Adaptive ECG Filtering and QRS Detection using Orthogonal Wavelet Transform. *In:* INTERNATIONAL CONFERENCE ON BIOMEDICAL ENGINEERING. Innsbruck, 2005. *Proceedings...* Innsbruck, IASTED. p. 109-114.

KRUEGEL, C.; VIGNA, G. 2003. Anomaly detection of web-based attacks. *In:* ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 10, Washington, 2003. *Proceedings...* Washington, CCS. p. 251-261. http://doi.acm.org/10.1145/948109.948144

KRUEGEL, C.; VALEUR, F.; VIGNA, G. 2004. *Intrusion Detection and Correlation: Challenges and Solutions.* Santa Clara, Springer, 136 p.

LU, W.; GHORBANI, A.A. 2009. Network anomaly detection based on wavelet analysis. *EURASIP J. Adv. Signal Process*, **4**(1)4-16. http://dx.doi.org/10.1155/2009/837601

MALLAT, S. 2008. *A wavelet tour of signal processing*, 3 ed., Amsterdam, Academic Press. 832 p.

OWASP. 2010. Top 10 Web Application Security Risks. Available at: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. Acessed on: October 2th, 2011.

SRIRAGHAVAN, R.G.; LUCCHESE, L. 2008. Data processing and anomaly detection in web-based applications. *In:* IEEE WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING. Cancun, 2008. *Proceedings...* Cancun, p. 187-192. http://dx.doi.org/10.1109/MLSP.2008.4685477

STOLLNITZ, E.J.; DEROSE, A.D.; SALESIN, D.H. 1995. Wavelets for computer graphics: a primer 1. IEEE COMPUTER GRAPHICS AND APPLICATIONS. Seattle. 1995. *Proceedings...* Seattle, **15**(6):76-84. http://dx.doi.org/10.1109/38.376616

WAGNER, R.; FONTOURA, L.M.; NUNES, R.C. 2011. Tailoring Rational Unified Process to Contemplate the SSE-CMM. *In:* LATIN AMERICAL CONFERENCE ON INFORMATICS, Quito, 2011. *Proceedings...* Quito, CLEI. p. 128-141.

WANG, K.; STOLFO, S.J. 2004. *Anomalous Payload-Based Network Intrusion Detection*. New York. Springer Berlin, Heidelberg. 3224:203-222. http://dx.doi.org/10.1007/978-3-540-30143-1_11

Submitted on October 10, 2011.
Accepted on December 20, 2011.