Behavior Analysis of Web Service Attacks

Abdallah Ghourabi, Tarek Abbes, Adel Bouhoula

Behavior Analysis of Web Service Attacks

Abdallah Ghourabi, Tarek Abbes, and Adel Bouhoula

Higher School of Communication of Tunis SUP'COM, University of Carthage, Tunisia
abdallah.ghourabi@supcom.rnu.tn, tarek.abbes@isecs.rnu.tn,
adel.bouhoula@supcom.rnu.tn

Abstract. With the rapid development of Internet and its services, cyber attacks are increasingly emerging and evolving nowadays. To be aware of new attacks and elaborate the appropriate protection mechanisms, an interesting idea is to attract attackers, then to automatically monitor their activities and analyze their behaviors. In this paper, we are particularly interested in detecting and learning attacks against web services. We propose an approach that describes the attacker's behavior based on data collected from the deployment of a web service honeypot. The strengths of our approach are that (1) it offers a high interaction environment, able to collect valuable information about malicious activities; (2) our solution preprocesses the set of data attributes in order to keep only significant ones (3) it ensures two levels of clustering in order to produce more concise attack scenarios. In order to achieve these contributions, we employ three analysis techniques: Principal Component Analysis, Spectral Clustering and Sequence Clustering. Our experimental tests allow us discovering some attacks scenarios, such as SQL Injection and Denial of Services (DoS), that are modeled in Markov chains.

Keywords: Honeypot, Web Service, Attacker's Behavior, Clustering, Data Analysis

1 Introduction

The Web Service technology is increasingly used in companies due to its simplicity and interoperability. It is based on several standards such as SOAP and XML for exchanging information between applications in heterogeneous environments through the Internet. Among the security problems of Web Services are the exchanged data that can convey many threats to the target system. To address these security issues, administrators have to carefully supervise the execution and utilization of web services. The Honeypot technology constitutes an ideal solution to ensure this kind of monitoring. It allows discovering new attacks methods, intrusion scenarios and attackers' objectives and strategies. In this context, we have proposed in an earlier work [3] a honeypot solution, called WS Honeypot, designated to attract and monitor web service attacks.

Honeypots are very useful for collecting valuable information about the attackers and their techniques. However, the volume of collected data increases

rapidly due to frequent and repetitive data. The large amount of data complicates the analysis task and overwhelms rapidly the human analyst. Thus, a manual tracking of attackers' activities on the honeypot seems to be very difficult and tedious work. To improve the performance of the data analysis in our WS Honeypot, we propose in this paper an automatic approach to analyze the collected data and describe attackers' behaviors on the honeypot. Our approach combines 3 analysis techniques: Principal Component Analysis to select the features to be extracted from the captured data; Spectral Clustering to cluster the collected requests and Sequence Clustering to regroup the similar activities performed by the attackers and describe their behavior.

The remaining parts of the paper are organized as follows: Section 2 reviews the related works. Section 3 defines the analysis techniques used in our approach. Section 4 presents the architecture of our analysis approach and describes the working principle of the implemented algorithms. Section 5 reports our experimental results. Finally, we conclude the paper in Section 6.

2 Related Work

The approach proposed in this paper is positioned among other related works that aim to characterize attacks in the honeypots. Despite the common purpose, these works have used various analytical techniques. For instance, Pouget and Dacier [8] proposed a simple clustering approach to analyze the data collected from the honeypot project Leurre.com. Their objective is to characterize the root causes of attacks targeting their Honeypots. The aim of this algorithm is to gather all attacks presenting some common characteristics (duration of attack, targeted ports, number of sent packets, etc.) based on generalization techniques and association-rule mining. Resulting clusters are further refined using Levenshtein distance. The final goal of their approach is to group into clusters, all attacking sources sharing similar activity fingerprints, or attack tools. Alata et al. [1] presented some results obtained from their project CADHo (Collection and Analysis of Data from Honeypots). The purpose of this project is to analyze the data collected from the environment Leurre.com and to provide models for the observed attacks. They proposed simple models describing the time-evolution of the number of attacks observed on different honeypot platforms. Besides, they studied the potential correlations of attack processes observed on the different platforms taking into account the geographic location of the attacking machines and the relative contribution of each platform in the global attack scenario. The correlation analysis is based on a linear regression models. Thonnard and Dacier proposed a framework for attack patterns' discovery from the honeynet collected data [11]. The aim of this approach is to find, within an attack dataset, groups of network traces sharing various kinds of similar patterns. In this work, the authors applied a graph-based clustering method to analyze one specific aspect of the honeynet data (the time series of the attacks). The results of the clustering applied to time-series analysis enable to identify the activities of several worms and botnets in the traffic collected by the honeypots.

Compared to above works, our approach offers two main advantages. Firstly, we use the Principal Component Analysis to select the pertinent features unlike other works where this selection is done manually. Secondly, to cluster observed attacks, we employ a sequence clustering method which allows tracking the transition between the attacker's activities taking into account the order of realization.

3 Data analysis techniques

To analyze the large amount of data collected from the honeypot, several techniques of data analysis can be applied, such as statistics, data mining and machines learning. Generally, researchers apply such methods to data for two main reasons: to better understand the existing data and to predict something about new data [7]. In this paper, we propose a hybrid approach composed of three analysis techniques (Principal Component Analysis, Spectral Clustering and Sequence Clustering) to analyze the data collected from our WS Honeypot and describe the behavior of the captured attacks. Before presenting our approach, we devote this section to introduce these three techniques.

3.1 Principal Component Analysis

The Principal Component Analysis (PCA) is a data analysis method used for dimensionality reduction and multivariate analysis. The central idea of principal component analysis is to reduce the dimensionality of a data set consisting of a large number of possibly correlated variables, while retaining as much as possible of the variation present in the data set [4]. This is achieved by transforming these variables into a smaller number of uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component is a linear combination of the original variables with the largest possible variance. The second principal component is the linear combination of the original variables with the second largest variance and orthogonal to the first principal component, and so on [12].

To describe the principle of PCA, let us consider a set of observations (x_1, x_2, ..., x_n) where each observation is represented by a vector of length m. The dataset is represented by a matrix X of dimensions $n \times m$. The transformation into principal components is mainly based on the eigenvalues and the eigenvectors of the covariance matrix C (formed from X). We suppose that $(\lambda_1, u_1), (\lambda_2, u_2), ..., (\lambda_m, u_m)$ are the pairs (eigenvalue, eigenvector) of the covariance matrix C. We choose the k eigenvectors having the largest eigenvalues. Afterwards, we form a $m \times k$ matrix U whose columns consist of the k eigenvectors. The representation of the data by principal components is made by projecting the original data onto the k-dimensional subspace according to the following rule [12]:

$$y_i = U^T(x_i - \mu)$$

Where $\mu = \frac{1}{n}\sum_{(i=1)}^{n} x_i$ and y_i is a k-dimensional vector that represents the projection of the original m-dimensional data vector x_i.

3.2 Spectral Clustering

Clustering is an unsupervised learning method that seeks to assign a set of objects into homogeneous groups (called clusters). Several algorithms can be used to cluster data such as k-means, hierarchical clustering, spectral clustering, etc. In recent years, spectral clustering has become one of the most popular modern clustering algorithms. It is simple to implement and very often outperforms traditional clustering algorithms such as the k-means algorithm [6]. Unlike other clustering algorithms, the spectral clustering algorithm is based on the concept of similarity between each pair of points instead of distance.

Given a set of data points $x_1, ..., x_n$, the similarity matrix may be defined as a matrix S, where S_{ij} represents a measure of the similarity between all pairs of data points x_i and x_j. The main goal of clustering is to divide the data points into several groups in a way that the data points of the same group exhibit similar properties. To do that, the Spectral clustering represents the data in the form of the similarity graph $G = (V, E)$. Each vertex V_i in this graph represents a data point x_i. Two vertices are connected if the similarity S_{ij} between the corresponding data points x_i and x_j is positive or larger than a certain threshold, and the edge is weighted by S_{ij} [6].

After constructing the similarity graph, the next step is to find a partition of the graph in a way that the edges between different groups have very low weights and the edges within a group have high weights. The basic idea is to calculate the graph Laplacian matrix L, and to extract the k (number of clusters to construct) first eigenvectors of L. These eigenvectors allow obtaining a projection space with a smaller dimension (k dimensions). Afterwards, a clustering algorithm (e.g. k-means) can be applied to assign each data point into a cluster among the k ones.

3.3 Sequence clustering

The role of sequence clustering is to group a set of sequences in such a way that sequences in the same group (cluster) are more similar to each other than to those in other clusters. The working principle of a sequence-clustering algorithm differs depending on the used implementation. In our work, we are particularly interested in Microsoft Sequence Clustering algorithm [10] which is included in the data mining tools of Microsoft SQL Server.

Microsoft Sequence Clustering

The Microsoft Sequence Clustering algorithm is a hybrid algorithm that uses Markov chain to analyze the sequences and partitions them using the Expectation Maximization (EM) method of clustering. The Markov chain can be defined as a mathematical system that describes the transition probabilities between a

set of states. In Microsoft Sequence Clustering algorithm, each generated cluster is associated to an n-order Markov chain.

To describe the basic principles of this algorithm, consider the case of a first-order Markov chain. For an observed sequence, the probability of belonging to a given cluster is in effect the probability that the observed sequence was produced by the Markov chain associated with that cluster [2]. Consider a sequence $x = x_1, x_2, ..., x_l$ of length l, the probability of belonging to a given cluster C_k is calculated using the following formula:

$$p(x|C_k) = p(x_1, C_k). \prod_{(i=2)}^{l} p(x_i|x_{(i-1)}, C_k) \quad (1)$$

where $p(x_1, C_k)$ is the probability that x_1 is the first state in the Markov chain associated with the cluster C_k and $p(x_i|x_{(i-1)}, C_k)$ is the transition probability of state $x_{(i-1)}$ to x_i in the same Markov chain. To calculate the parameters of this model, the Microsoft Sequence Clustering implements the iterative algorithm Expectation and Maximization (EM).

4 Automatic analysis of Web service Honeypot data

The data collected from a honeypot contains interesting information about the attacker and his activities. This information is useful to understand the hackers' strategies and to learn the modus operandi of their attacks. To obtain a better knowledge, an exhaustive analysis of collected data is very important. In this paper, we propose an automatic approach to analyze attacks traces collected from the deployment of a Web service Honeypot called WS Honeypot, previously presented in [3].

WS Honeypot is a high interaction honeypot that simulates the behavior of a Web service. The role of this honeypot is to attract and monitor attackers targeting web service applications. WS Honeypot provides real web services to ensure a real interaction with attackers. The services offered by the honeypot can be deployed by using two technologies, Axis and .Net. The approach proposed in this paper aims at describing the behavior of attacks captured by WS Honeypot, using several analysis techniques.

4.1 WS Honeypot data Analysis workflow

The overall architecture of our approach is shown in Figure 1. The WS Honeypot captures the SOAP messages sent by the attackers. Afterwards, we extract from each SOAP message its characteristic parameters (Feature extraction). Then, we apply a statistical method called "Principal Component Analysis" to select the most relevant features that ensure reliable identification of attacks. Thus, we obtain a dataset of observations (DS1) incorporating significant parameters extracted from captured SOAP messages. The next step consists in applying a

spectral clustering on dataset DS1 in order to cluster the captured SOAP messages into homogeneous groups; each group is supposed to represent a message type. As each SOAP message sent by the user is designated to perform an activity in the web service, we consider in the following that each resulting cluster represents an activity type that an attacker can accomplish with different manners at the WS Honeypot. Hence, the spectral clustering analysis allows us to characterize the session of an attacker by determining his activity types in the honeypot in a period of 24 hours. These data are stored on a second dataset DS2 called "session characteristics". The final step of our analysis process consists in applying a sequence clustering on the second dataset DS2 in order to form homogeneous clusters. Each cluster is designed to describe the attacker's behavior for each attack type. The cluster, containing related activity types, can be modeled by a Markov chain to outline the attack scenario.

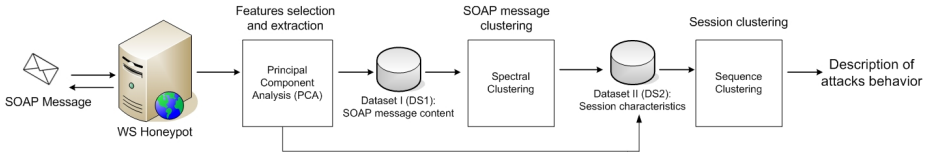


Fig. 1. Data analysis workflow in WS Honeypot

4.2 Features selection and extraction

Features selection and extraction is a preprocessing step, which is very important in our approach. It allows the selection of relevant features to be extracted from the SOAP messages in order to improve the performance of the clustering process. Although the choice of these features can be made manually based on expert knowledge, the use of automated algorithms for feature selection is very interesting to obtain better results. In the case of our approach, we employ a statistical method called "Principal Component Analysis" to select the most pertinent features.

The selection process is as follows: Firstly, we prepare a preliminary list of all possible features that we can extract from a SOAP message. Then, we apply the PCA algorithm on a set of instances representing some SOAP messages according to the chosen features. Finally, we exploit the results of PCA to reduce the preliminary list of features and keep only those which are relevant.

Preliminary list of features

First of all, we describe all features that we are able to extract from a SOAP message. The Web service requests are formatted in XML language and encapsulated in SOAP messages with the use of HTTP as a transport protocol. A malicious user, trying to attack the Web service, can inject a malicious content in the SOAP message in order to exploit the target. Therefore, the inspection

of the SOAP messages contents is essential to identify attacks. Moreover, we extract extra parameters characterizing the SOAP exchange and the amount of consumed resources for processing. This list includes the source IP address of the sender machine, the destination IP address, the encoding type, the header size, the overall size of the request, the protocol type, the port number, the input parameters of the web service operations and some information related to the request processing (execution time, memory and CPU usage). We present in Table 1 the preliminary list of the selected features during this step.

Table 1. List of initial features

No	Feature	Description
1	Src_IP	source IP address
2	Dest_IP	destination IP address
3	Protocol	type of used protocol
4	Port	destination port number
5	Soap_msg_size	size of the SOAP message
6	Soap_header_size	size of the header
7	Soap_request_type	request type
8	Soap_response_type	response type
9	Encoding	used encoding type
10	Operation	called operation name
11	in1	input for parameter number 1
12	in2	input for parameter number 2
13	in3	input for parameter number 3
14	in4	input for parameter number 4
15	in5	input for parameter number 5
16	in6	input for parameter number 6
17	in7	input for parameter number 7
18	in8	input for parameter number 8
19	in9	input for parameter number 9
20	in10	input for parameter number 10
21	Time	request processing time
22	Mem	memory occupation
23	CPU	request CPU usage

Selection of pertinent features

The objective of this step is to apply the Principal Component Analysis (PCA) method in order to reduce the list of preliminary features described in Table 1 by eliminating useless features. For this purpose, we collect several types of SOAP requests. In addition to normal requests, we gather others messages with malicious code causing several attacks such as SQL/XML injection, denial of service, parameter tampering, etc. From each SOAP message, we extracted the features described in Table 1 and we stored them in a dataset. Each instance (observation) of the dataset is designed to characterize a SOAP request. On this dataset, we applied a PCA algorithm implemented in Tanagra software [9].

The PCA replaces the old axes with new axes (called factorial axes). The latter are associated with new variables (called principal components) obtained by linear combinations of old variables. To interpret the results, we report in

Table 2 the obtained factorial axes, the associated eigenvalues, the proportion of inertia (data dispersion) explained by each factorial axis and the cumulative inertia. For the choice of factorial axes (or principal components) to be retained, we are based on the Kaiser criterion [5] which consists in only retaining the axes that have an eigenvalue greater or equal to 1 (i.e. the first 6 axes). Nevertheless, as the axes 7 and 8 have an eigenvalue very close to 1, we also retained them. From Table 2 we can see that the first 8 factorial axes dispose 77.84% of the available information.

The last step in this process consists in projecting the original data on the 8 first factorial axes and determining the variables that correlate better with these axes. Based on this projection, we selected 12 variables that have a high correlation with the factorial axes: Operation, Soap_msg_Size, Mem, CPU, Time, in1, in2, in3, in4, in5, in6, and in7. This analysis show that these 12 variables are the most important (and hence the most relevant) compared to the rest of variables.

The selected variables (features) are used to construct the first dataset "SOAP message content" noted DS1. From each SOAP request captured by the WS Honeypot, we extract the 12 features to characterize the request and we store them in the dataset DS1.

Table 2. Factorial axes issued from PCA

Axis	Eigen value	Proportion (%)	Cumulative (%)
1	2,770407	17,32 %	17,32 %
2	2,182733	13,64 %	30,96 %
3	1,689447	10,56 %	41,52 %
4	1,524927	9,53 %	51,05 %
5	1,230749	7,69 %	58,74 %
6	1,105892	6,91 %	65,65 %
7	0,989987	6,19 %	71,84 %
8	0,960595	6,00 %	77,84 %
9	0,844701	5,28 %	83,12 %
10	0,802064	5,01 %	88,13 %
11	0,419877	2,62 %	90,76 %
12	0,350289	2,19 %	92,95 %
13	0,332850	2,08 %	95,03 %
14	0,301130	1,88 %	96,91 %
15	0,267431	1,67 %	98,58 %
16	0,226920	1,42 %	100,00 %
17	0,000000	0,00 %	100,00 %
...			

4.3 SOAP messages clustering

Intruders employ different methods to penetrate the target system. When exploiting web services, they have several ways to succeed an attack (SQL injection, Path traversal, XML Injection, DoS, etc.). Thus, clustering similar attacks is useful to characterize the different categories of intrusion. For this process, we used the spectral clustering algorithm. To cluster the collected SOAP messages

(requests), we apply the clustering algorithm on the first dataset DS1 of "SOAP messages content". The execution of spectral clustering algorithm includes essentially three steps: construction of the similarity matrix S, extracting the k first eigenvectors from the Laplacian matrix L, and partitioning the data.

Construction of similarity matrix

The first step in spectral clustering is to construct a similarity matrix that measures the similarity between the data points of our dataset DS1. We construct a graph where pairs of objects are connected by links weighted by similarity values. The final objective is to find a partition of the graph such that edges between different groups have a very low weight (similarity) and the edges within a group have a high weight. Since SOAP messages are viewed as a set of string, we choose the Levenshtein distance instead of the Euclidean distance to compute the similarity. Having two strings, the chosen distance is equal to the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character. Once we calculate this value, the resulting similarity function is a Gaussian measure $S_{ij} = exp(\frac{-d^2(x_i,x_j)}{2\sigma^2})$, where d is a distance function (Levenshtein distance) and σ is a user specified scaling factor. At the end of this step, we obtain a similarity matrix $S \in R^{n \times n}$, where n is the size of the objects set.

Extracting the k first eigenvectors

The next step of spectral clustering is to construct the similarity graph $G = (V, E)$ based on the similarity matrix S. V denotes the set of vertices V_i representing the data point x_i; E is the set of edges connecting the pairs of vertices. In the case of our implementation, each data point x_i contains the characteristic parameters of each SOAP message captured by the WS Honeypot.

Afterwards, we need to calculate the graph Laplacian matrix L and extract the k (number of clusters to construct) first eigenvectors $(u_1, ..., u_k)$ of L as described in Subsection 3.2. These eigenvectors allow obtaining a projection space with smaller dimensions (equal to k). We use these vectors $(u_1, ..., u_k)$ as a set of columns to obtain a matrix U whose rows will be classified in the next and final step.

For the choice of the number of clusters k, we refer to the idea presented by U. Luxburg [6], which is based on the eigengap heuristic. Here, the goal is to choose the number k in such a way that all the $\lambda_1, ..., \lambda_k$ eigenvalues are very small, but λ_{k+1} is relatively large.

Data partitioning

At the final step of spectral clustering, we consider each row of the matrix U as a point in R^k and we cluster it via K-means (in our case) or other algorithm. Finally, we assign the original object x_i (SOAP message i) to cluster j if and only if row i of the matrix U is assigned to cluster j.

4.4 Sequence clustering

Web service attacks are not always too simple as sending a single request containing malicious parameters. Attackers are now looking for solutions to generate more complicated attacks that are performed on several steps, hence involving multiple Web service requests. For example, the denial of service attack is performed by submitting multiple requests from one or several sources in the same time. In other cases, an attacker must initiate other attacks in order to succeed his final attack. To follow and characterize the entire attack scenario, the security expert needs to monitor the interaction between the attacker and the system. For this reason, we choose in our approach to monitor the entire sessions captured by the WS Honeypot. The proposed idea consists in collecting the sequence of activities occurred during a user session and applying a clustering algorithm to form homogenous clusters. We define a session as the set of SOAP requests submitted to the WS Honeypot from one source during a period not exceeding 24 hours. Each formed cluster is designed to describe the behavior of attackers that have performed similar activities in the honeypot.

The sequence clustering algorithm is applied on the second dataset "session characteristics", noted DS2. Each instance of the dataset contains: the source IP address of the session, the sequence of activities performed during the session and the average inter-arrival time. The sequence of activities is created based on the results of spectral clustering. Knowing that each cluster formed by the spectral clustering represents an activity type, we can determine the sequence of activities in each session by checking each received request to which activity cluster is belonging. For example, suppose that during a given session we captured 5 SOAP requests (req1, req2, req3, req4 and req5). After the spectral clustering, we concluded that req1, req2 and req3 are belonging to Cluster 1 because they are very similar. On the other hand, req4 and req5 belong respectively to Cluster 2 and Cluster 3. The sequence of activities related to this session is then equal to {A1, A1, A1, A2, A3}.

The sequence clustering method used in this approach is based on Microsoft Sequence Clustering algorithm described in Subsection 3.3. In this clustering process, we collect all the sequences of activities captured from the WS Honeypot and we form homogenous clusters. Each cluster is supposed to describe the attacker's behavior for an attack type. In some clusters we can find several attacker sequences. This means that these attackers have performed similar activities in the honeypot. Each resulting cluster is represented by a Markov chain. The chain is used to describe the attacker's behavior and the transitions between the activities that s/he has carried out.

5 Experimental results

5.1 Experimental data

The experimental tests described in this section are performed on the data collected from the deployment of our WS Honeypot on Internet. The period of

deployment has lasted five months starting from February 2012 until June 2012. The honeypot was configured to simulate a Web service for online shopping. This type of service attracts many attackers especially those seeking for confidential information like credit card numbers and user passwords.

To enrich the data upon which WS Honeypot can work and give it the opportunity to receive other types of attack, we have chosen to simulate multiple attacks and to use some tools of penetration test and vulnerabilities discovery in order to attack the WS Honeypot. In this way, the overall data collected from the honeypot were coming from different sources:

- Real traffic from the Internet
- Fuzzing penetration testing tool (WSFuzzer)
- Web vulnerability scanner (Acunetix)
- Simulated attacks from different sources

In total, we have collected an amount of data equal to 38.7 MB, which are generated by 2317 requests coming from 113 different IP sources. In Table 3, we give a brief summary of the collected dataset.

Table 3. Dataset Characteristics

Dataset size	Number of captured requests	Number of sources	Number of sessions	Protocol distribution
38.7 MB	2317	113	451	SOAP (96%) Other (4%)

5.2 SOAP messages clustering

Most often, attackers send SOAP requests with malicious contents in order to attack a web service. The purpose of these requests is to generate harmful activities on the target service. The content and the nature of a malicious request depend on the attack type and goal. For example, the added code within a SOAP request to launch an SQL injection attack differs from that used in an XSS (Cross-site scripting) attack. To assign each SOAP request to an attack class, we employed the spectral clustering. The goal of this clustering is to categorize the activities performed by the attacker during its interaction with the WS Honeypot.

To test this technique we applied the spectral clustering over our collected dataset. In Table 4, we describe the result of this task by presenting some formed groups (clusters) and the malicious content found in these groups.

5.3 Description of the attacker's behavior

All activities performed by the users in the WS Honeypot, whatever malicious or not, are stored in our datasets. To characterize these activities, we firstly apply a spectral clustering. For example, if an attacker sent a SOAP request to the

honeypot and the clustering process reveals that the request belongs to group 1 (in Table 4), we can conclude that the attacker performed the activity type 'A1', i.e. it is most likely to be an SQL injection. Afterward, we determine the list of activities for each user within the same session, and we apply a sequence clustering on the constructed dataset.

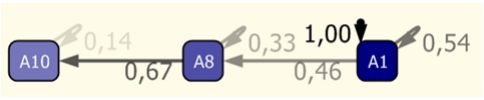
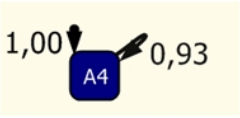
In Table 5, we present some examples from the obtained results. Among the clusters formed by the sequence clustering, we describe Cluster 5 and Cluster 12. In cluster 5, we find 6 similar sequences of 6 different attackers. There are 3 types of activity: A1, A8 and A10. The attacker number 1 has performed during the same session the following activities: A1, A1, A8, A8 and finally A10. A1 means an SQL injection; A8 and A10 refer to 2 variants of parameters tampering attack. To describe the behavior of the intruder in this cluster, we employ the Markov chain. In this chain, the states represent the activities types and the links describe the transition probabilities between these states. By analyzing the Markov Chain derived from Cluster 5, we can conclude that the attack begins with the activity A1 (which may be repeated several times) and then passes to A8 and A10 with a given transition probability. In Cluster 12, there are 2 sequences formed by the activity A4. Here, A4 designates the submission of a large SOAP request with a repetitive content. An attacker, who generates at several times this type of activity, essentially aims to conduct a denial of service attack.

Table 4. Example of groups formed by the Sectral Clustering

Group number	Malicious sequence	Probable attack type
Group 1	0 or 1=1 ' or 0=0 – " or 0=0 – or 0=0 – ' or 0=0 ' or 1=1– " or 1=1- ' or 'x'='x 0 or 1=1	SQL injection
Group 2

'
/' | Meta-Character Injection |
Group 3	../../../../../../../../../../etc/passwd%00 ../../../../../../../../../../etc/passwd ../../../../../../../../../../etc/shadow%00 ../../../../../../../../../../etc/shadow \..\..\..\..\..\..\..\..\..\..\..\..\passwd ../../../../../../../../conf/server.xml	Path Traversal
Group 4	AAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAA.....AA AAAAAAAAAAAAAAAAAAAA BBBBBBBBBBBAAAAAAAA.....AAA BBBBBBBBBBBBBBBBBBBBBBBB BBBBBBBBBBBBBB	Unknown
...		

Table 5. Example of clusters obtained from sequence clustering

Cluster description	Associated Markov Chain
Cluster 5 Size: 6  1. A1,A1,A8,A8,A10 2. A1,A1,A8,A8,A10 3. A1,A1,A8,A10 4. A1,A1,A8,A8,A10 5. A1,A1,A1,A8,A10 6. A1,A1,A8,A10	
Cluster 12 Size: 2  1. A4, A4, A4, A4, A4, A4, A4, A4, A4, A4, A4 2. A4, A4, A4, A4, A4, A4, A4, A4, A4, A4, A4, A4, A4, A4, A4, A4	

Each cluster can be summarized by a signature describing the attack process. For example, we present in Figure 2 the attack signature of Cluster 5.

6 Conclusion

In this paper, we proposed an analysis method to explore data collected from a web service honeypot. The analysis process is based on the use of a statistical technique "Principal Component Analysis" and two clustering methods: Spectral Clustering and Sequence Clustering. The described approach is divided into three main steps. Firstly, we selected the pertinent features from collected data by the use of Principal Component Analysis. Afterwards, we applied a spectral clustering to extract groups of activities sharing common characteristics. Finally, we gathered the sequence of activities for each attack session and we used the sequence clustering to form homogeneous groups describing the behavior of attackers having similar attitudes. We evaluated our approach by experimental tests applied on data collected from our Web Service Honeypot. The obtained results describe the attacker's behavior in the form of a Markov Chain representing the transition between its activities.

As a future work, we plan adding another functionality to automatically construct enriched attacks signatures to be used by Intrusion Detection and Prevention Systems (IDPS). We envisage also expanding the detection features of our Honeypot so that it can support attacks targeting web applications in general.

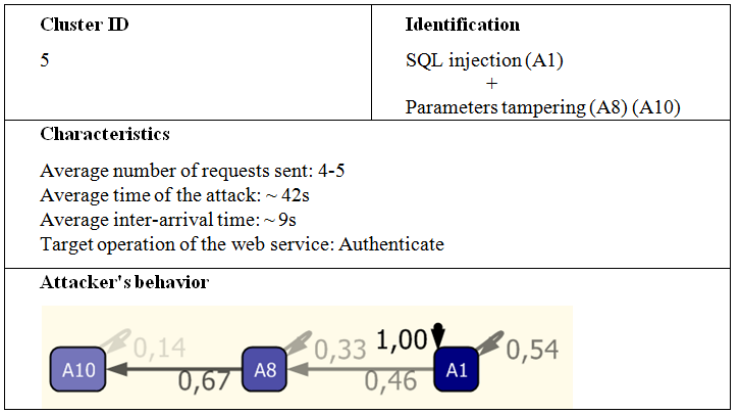


Fig. 2. Example of a Cluster signature

References

1. Alata, E., Dacier, M., Deswarte, Y., Kaniche, M., Kortchinsky, K., Nicomette, V., Pham, V.H., Pouget, F.: Collection and analysis of attack data based on honeypots deployed on the Internet. In: First Workshop on Quality of Protection, Security Measurements and Metrics, Milan, Italy (2005)
2. Ferreira, D., Zacarias, M., Malheiros, M., Ferreira, P.: Approaching Process Mining with Sequence Clustering: Experiments and Findings. In: Proceedings of the 5th International Conference on Business Process Management (BPM). LNCS, vol. 4714, pp. 360-374 (2007)
3. Ghourabi, A., Abbes, T., Bouhoula, A.: Design and implementation of web service honeypot. In: 19th International Conference on Software, Telecommunications and Computer Networks, Split, Croatia (2011)
4. Jolliffe, I.T.: Principal Component Analysis. 2nd Ed., Springer-Verlag, NY (2002)
5. Kaiser, H.F.: The application of electronic computers to factor analysis. Educational and Psychological Measurement, 20, 141-151 (1960)
6. Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing, Vol. 17 No. 4, pp. 395 416 (2007)
7. Maloof, M. A.: Machine Learning and Data Mining for Computer Security: Methods and Applications. Springer (2006)
8. Pouget, F., Dacier, M.: Honeypot-based Forensics. In: AusCERT Asia Pacific Information Technology Security Conference (AusCERT'2004), Brisbane, Australia (2004)
9. Rakotomalala, R.: TANAGRA: a free software for research and academic purposes. In: Proceedings of EGC'2005, RNTI-E-3, vol. 2, pp.697-702 (2005)
10. Tang, Z., MacLennan, J.: Data Mining with SQL Server 2005. Wiley (2005)
11. Thonnard, O., Dacier, M.: A framework for attack patterns' discovery in honeynet data. Digital Investigation, vol. 8, pp. S128S139 (2008)
12. Wang, W., Battiti, R.: Identifying Intrusions in Computer Networks with Principal Component Analysis. In: Proceedings of the First International Conference on Availability, Reliability and Security (2006)