McPAD and HMM-Web: two different approaches for the detection of attacks against Web applications

Davide Ariu, Igino Corona, Giorgio Giacinto, Fabio Roli
University of Cagliari, Dept. of Electrical and Electronic Engineering
Piazza d'Armi, 09123 Cagliari (Italy)
{davide.ariu, igino.corona, giacinto, roli}@diee.unica.it

1 Introduction

Intrusion Detection Systems (IDS) are valuable tools for the defense-in-depth of computer networks. Two main approaches to intrusion detection are used, namely *misuse* and *anomaly* detection. Misuse detectors are based on a description of known malicious activities. This description is often modeled as a set of rules referred to as *attack signatures*. Activities that match an attack signature are classified as malicious. This makes misuse based IDS able to detect known attacks with a very low false positive rate. On the other hand, anomaly detectors are based on a description of *normal* or *benign* activities. As malicious activities are expected to be different from normal activities, a suitable distance measure allows anomaly-based IDS to detect attack traffic. Anomaly-based detection systems usually produce a relatively higher number of false positives, compared to the misuse-based or *signature-based* detection systems, because only a fraction of the anomalous traffic actually derives from intrusion attempts. Nevertheless, anomaly detectors are able to detect *zero-day* (i.e., never-seen-before) attacks, whereas signature-based systems are not. As obtaining a labeled dataset representative of normal network activities is difficul and expensive, we might work in a scenario that in intrusion detection is usually referred as "unlabeled anomaly detection" (it is what in machine learning is actually known as "outlier detection" or "one-class classification") [9].
In this paper we propose two different anomaly based IDS for the detection of attacks against Web Applications. The first one, that we called McPAD, is a network based intrusion detection system that analyses the payload of HTTP packets containing requests toward a web server. The second one (HMM-Web), is an IDS that detects attacks against web application analyzing the request URI coming to the web server.

2 Payload-based Anomaly Detection

Recent work on unlabeled anomaly detection focused on *high speed* classification based on simple *payload* statistics [6, 11, 12]. For example, PAYL [11, 12] extracts

256 features from the payload. Each feature represents the occurrence frequency in the payload of one of the 256 possible byte values. A simple model of normal traffic is then constructed by computing the average and standard deviation of each feature. A payload is considered anomalous if a *simplified Mahalanobis distance* between the payload under test and the model of normal traffic exceeds a predetermined threshold. A more sophisticated version of PAYL has been also proposed, where the statistics were calculated considering sequences of bytes instead of single bytes. In [8] Perdisci et al. proposed a new approach to construct a *high speed* payload-based anomaly IDS by combining multiple one-class SVM classifiers using a majority voting rule. In order to solve the curse of dimensionality problem of the n-gram analysis, in [8] the authors proposed to measure the features by using a sliding window that "covers" two bytes which are ν positions apart from each other in the payload. By varying the parameter ν, a representation of the payload in different feature spaces was constructed.

2.1 Our work: McPAD

McPAD extends the results reported in [8]. In particular, in McPAD classifiers are combined using combination schemes different than Majority Voting. We experiment on large datasets of both normal traffic and attacks. In particular, we perform experiments on the first week of traffic from the DARPA'99 dataset [7], and on seven days of real HTTP traffic collected at an academic institution. IDS detection capabilities regarding detection accuracy for shell-code attacks [1] and polymorphic shell-code attacks have been tested. As attack dataset, we use a large dataset of "standard" HTTP attacks provided by the authors of [5], which is publicly available. Furthermore, we construct a large number of polymorphic attacks, which include attacks generated with the polymorphic shell-code engine CLET [3], a set of Polymorphic Blending Attacks (PBA) [4] designed to evade PAYL [11], and a set of PBA specifically designed with the intent to evade our detection system. We compare McPAD to PAYL [11], and we show that our IDS has a much higher detection accuracy than PAYL on shell-code attacks at low false positive rates, and is in some cases resistant to even the sophisticated polymorphic blending attacks specifically designed to evade it. Furthermore, we released the source code of McPAD and the attack datasets we used for our experiments, with the hope of making the results we obtained reproducible. Both McPAD and the attack datasets can be downloaded from `http://prag.diee.unica.it/n3ws1t0/node/201`.

3 WEB Applications security

Nowadays, the web-based architecture is the most frequently used for a wide range of internet services, as it allows to easily access and manage information and software on remote machines. The input of web applications is made up of queries, i.e. sequences of pairs *attribute←value*. A wide range of attacks exploits web application vulnerabilities, typically derived from input validation flaws [13, 10]. HMM-Web exploits a new formulation of query analysis through Hidden Markov Models (HMM) and show that HMM are effective in detecting a wide range of either known or unknown attacks on web applications. In addition, despite previous related works [2, 5], HMM-Web esplicitly address the problem related to the presence of noise (i.e., attacks) in the training set. Also, HMM-Web performances can be increased when a sequence of symbols is modelled by an *ensemble* of HMM. Experimental results on real world data, show the effectiveness of the proposed system in terms of very high detection rates and low false alarm rates.

A detailed description of HMM-Web is presented in section 3.1. Experimental results are reported in section 3.2. Conclusions and future work are drawn in section 4.

3.1 Our work: HMM-Web

Our aim is detecting both simple and sophisticated attacks against web applications. Thus, we exploit the powerful of HMM to thoroughly model web application queries. Focusing on this goal, each web application is analysed *independently* to provide for a specialised modelling of its queries. Thus, the IDS is composed by a set of (independent) application-specific modules, each one dedicated to the analysis of queries on a specific web application.

When a request on a certain web application must be analysed, first it is checked whether a module specialised on that web application is present or not. If such a module is not present, the related query is considered suspicious, because on a "unknown" web application (that is, an application which was not contained in the training set). Otherwise, the query is sent to the module specialised on that web application, which outputs the probability of such a query. Furthermore, a decision module classifies the request as suspicious (a possible attack) or legitimate, applying a threshold to the probability assigned to the query.

Each application-specific module computes the probability of a query by correlating different HMM ensembles, i.e., for the analysis of the attribute sequence and the input of each attribute, using *minimum* rule, so as to detect any anomaly in a query (see figure 1). HMM inside a generic ensemble are correlated using *maximum* rule, aiming at selecting the HMM which better models the analysed sequence.

During the operational phase of the IDS, the fraction of queries on each web application can be viewed as the probability of (a query on) the web application. The web application probability reflects in some way how strong is the assumption that its queries (inside the training set) are actually legitimate. So, thresholds inside the decision module are in inverse proportion with respect to web application probability, in agreement with the overall fraction of non-legitimate queries α we expect in the training set.

For a generic query, the sequence of attributes is modelled by considering each attribute as a different symbol. Conversely, to light out the typical relevance of meta-characters in the *semantics* of attribute inputs, we generalise each letter with a special symbol $\underline{\mathtt{A}}$ and each digit with $\underline{\mathtt{N}}$.

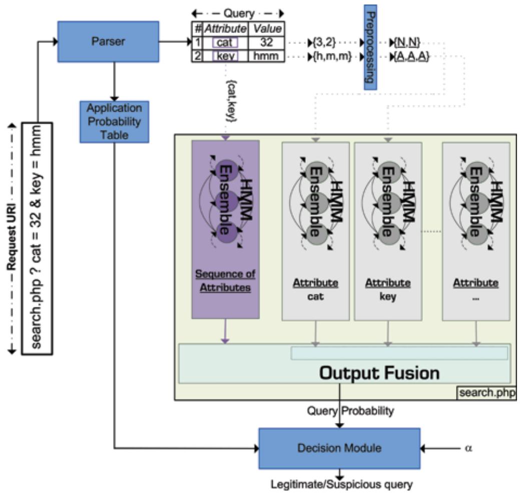

Figure 1: Example scheme of the IDS functioning. In the picture, a parser processes the request URI (obtained from a GET request) and identifies the web application (i.e. search.php) and its input query. The codified query is analysed by the specific module for the web application, which outputs a probability value. The decision module, using a threshold which depends on the web application probability and the α parameter, classifies the query as legitimate or suspicious.

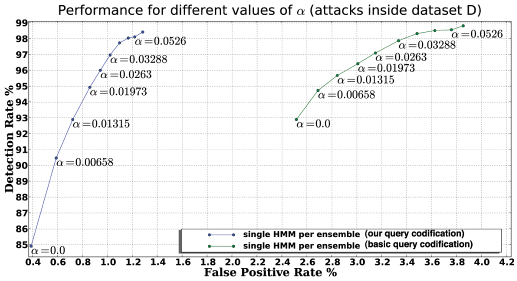

Figure 2: IDS performance for different values of fraction of training queries classified as suspicious α, starting from $\alpha = 0$. Picture refers to dataset D, whose splits are used for training. Each point in this graphic is the average detection rate and false positive rate for a specific value of α. A single HMM per ensemble is used, either with the query codification used in [2] or with our codification.

Figure 3: HMM-Web overview and performances.

3.2 HMM-Web: Experimental Results

We collected about 150,000 queries for training the system (dataset D) in a time interval of six months, from a production web server used by our University. Queries were distributed over a total of 52 web applications. A substantial set of these applications (24) was related to administration services, and all others to public services (28). Using as a reference the authoritative archive of the MITRE Corporation (`http://cve.mitre.org`), we performed 19 SQL Injection attacks and 19 Cross-site Scripting attacks, on a subset of 18 web applications, for a total of 38 attacks, stored on dataset A.

For a specific setting of our IDS, we evaluated its performance through a cross validation technique. Dataset D is randomly divided in five parts (splits), containing the same number of queries. Queries inside dataset D are then labeled as legitimate or attack queries in a semiautomatic way, with a preliminary training of HMM-Web. Each split of D is used as an indipendent training set and, exploiting the previous labelling, the false alarm/detection rate is evaluated on the remaining splits. In such a way, we are able to both evaluate false alarm rate of our IDS and its effectiveness to spot attack queries, *even if very similar (attack) queries are contained in the training set.* As *all* these attacks inside data set A were detected by HMM-Web, we focused the performance analysis on dataset D. As we can see from figure 2 our query codification is really effective to heavily reduce the false alarm rate. On the other hand, with a little expense in terms of false alarm rate, a positive value of α definitely enhance the detection rate of attacks, even if similar attacks are present in the training set. It is interesting to note that even with a large fraction of training queries classified as suspicious, i.e. $\alpha \sim 3\%$ (we know that attacks are about the 1%), we are able to both detect 96% of attacks similar (maybe equal) to those inside the training set, and raise a fraction of false alarms lower than 1%. By setting $\alpha = 0$, a lower amount of false alarms can be obtained, i.e. about 0.4%, but about 15% of attacks inside D cannot be detected. However, as in our case, it may be fundamental to spot these attacks, as they may light out vulnerabilities of web applications which are currently exploited.

IDS performance is improved when an *ensemble* of HMM, w.r.t. a single HMM, is used to model the structure of a generic sequence. However, in our experiments we found that the higher values of α the lower the improvement of this setting. So, besides the increased complexity of the learning task, it is necessary to take into account this behaviour when using more than one HMM per ensemble.

4 Conclusions and future work

In this paper we proposed two different solutions for the detection of attacks against web applications. Even if results are good, both solution can be improved. In the case of McPAD we are working to extract as more information as possible from the payload structure, in order to make more difficult for the attacker evading the IDS. In the case of HMM-Web a possible improvement of the system may be related to the automatic cleaning of the training set. To this end, we are studying how to automatically retrain the IDS discarding (a little amount of) training queries which may be related to attacks (i.e. with lowest probability). Moreover, we are going to compare performance of our system w.r.t. the IDS proposed in [2], to evidence more thorougly our contribution.

References

[1] I. Arce. The shellcode generation. *IEEE Security and Privacy*, 2(5):72–76, 2004.

[2] Kruegel C., Vigna G., and Robertson W. A multi-model approach to the detection of web-based attacks. *Computer Networks*, 48(5):717–738, 2005.

[3] T. Detristan, T. Ulenspiegel, Y. Malcom, and M. Underduk. Polymorphic shellcode engine using spectrum analysis. *Phrack Issue 0x3d*, 2003.

[4] P. Fogla, M. Sharif, R. Perdisci, O. M. Kolesnikov, and W. Lee. Polymorphic blending attack. In *USENIX Security Symposium*, 2006.

[5] Ingham K.L., Somayaji A., Burge J., and Forrest S. Learning dfa representations of http for protecting web applications. *Computer Networks*, 51:1239–1255, 2007.

[6] C. Kruegel, T. Toth, and E. Kirda. Service specific anomaly detection for network intrusion detection. In *ACM Symposium on Applied Computing (SAC)*, 2002.

[7] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. The 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, 2000.

[8] R. Perdisci, G. Gu, and W. Lee. Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 488–498, 2006.

[9] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *ACM CSS Workshop on Data Mining Applied to Security*, 2001.

[10] www.sans.org Roger Meyer, SANS institute. Detecting attacks on web applications from log files, 2008.

[11] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Advances in Intrusion Detection (RAID)*, 2004.

[12] K. Wang and S. Stolfo. Anomalous payload-based worm detection and signature generation. In *Recent Advances in Intrusion Detection (RAID)*, 2005.

[13] Open Web Application Security Project (OWASP) www.owasp.org. The ten most critical web application vulnerabilities, 2007.