Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

HTTP attack detection using *n*-gram analysis

Aditya Oza [a], *Kevin Ross* [a], *Richard M. Low* [b], *Mark Stamp* [a,*]

[a] *Department of Computer Science, San Jose State University, USA*
[b] *Department of Mathematics, San Jose State University, USA*

ARTICLE INFO

Article history:
Received 29 May 2013
Received in revised form
11 April 2014
Accepted 2 June 2014
Available online 19 June 2014

Keywords:
HTTP attacks
n-gram analysis
Hidden Markov model
Network traffic analysis
Byte analysis
WordPress

ABSTRACT

Previous research has shown that byte-level analysis of network traffic can be useful for network intrusion detection and traffic analysis. Such an approach does not require any knowledge of applications running on web servers or any pre-processing of incoming data.

In this paper, we apply three *n*-gram techniques to the problem of HTTP attack detection. The goal is to provide a first line of defense by filtering the vast majority of benign HTTP traffic, leaving only a relatively small amount of suspect traffic for more costly processing. We analyze these *n*-gram techniques in terms of accuracy and performance. Our results show that we can attain equal or better detection rates at considerably less cost, in comparison to a previously developed HMM-based technique. We also apply these techniques to a highly realistic dataset consisting of four recent attacks and show that we obtain equally strong results in this case. Overall, these results indicate that this type of byte-level analysis is highly effective and practical.

1. Introduction

According to (Symantec internet security threat), approximately 4500 new web based attacks were introduced each day in 2011. Web application servers are common targets for such attacks, in part because they communicate with many other systems and are therefore a prized target for malware (Aycock, 2006).

Much HTTP traffic consists only of printable ASCII characters. In particular, we do not expect any executable code in incoming HTTP packets—executable code in HTTP is an indicator of a possible malware injection attack (Ahmed and Lhee, 2011). The byte distribution of printable ASCII is much more restricted than that of executable code. Therefore, analysis of the distribution of bytes in HTTP packets may be useful for detecting certain classes of attacks. In general, such an approach does not require knowledge of applications running on web servers or any significant pre-processing of incoming data.

Techniques based on hidden Markov models and *n*-gram analysis have previously been used to successfully detect HTTP attacks (Ariu et al., 2011; Perdisci et al., 2009; Wang and Stolfo, 2004). In Wang and Stolfo (2004) an *n*-gram model is applied to raw bytes in HTTP packets. This *n*-gram based technique was applied to the problem of file-type classification in Ahmed and Lhee (2011), where byte distributions are used to classify files as executable, text, or multimedia. The paper (Ahmed and Lhee, 2011) also considers a pattern counting technique to classify text and executable files. Hidden Markov model analysis of HTTP traffic is the focus of the research in Ariu et al. (2011). The paper (Toderici and Stamp, 2013) proposes a malware detection technique based on a χ^2 statistic.

* *Corresponding author.*
E-mail address: stamp@cs.sjsu.edu (M. Stamp).

http://dx.doi.org/10.1016/j.cose.2014.06.002

In this paper, we consider n-gram analysis similar to that in Ahmed and Lhee (2011) and Wang and Stolfo (2004), a χ^2 test analogous to that in Toderici and Stamp (2013), and the pattern counting technique in Ahmed and Lhee (2011). We apply these techniques to the problem of HTTP attack detection, and compare our results to those obtained using an HMM-based approach (Ariu et al., 2011). Note that the goal of any such approach is to provide an efficient first line of defense that will filter out the vast majority of innocent HTTP traffic, so that potentially malicious traffic can be further analyzed with more costly methods. Here, we provide extensive experimental results comparing both the effectiveness and the efficiency of these various techniques. Our primary objective is to determine the strengths and weaknesses of relatively straightforward n-gram analysis as compared to the HMM-based technique in Ariu et al. (2011).

This paper is organized as follows. In Section 2, we give an overview of network intrusion detection and discuss examples of byte-level analysis techniques that have been applied to web traffic. We also discuss Pearson's χ^2 distance, which is the basis of one of the attack detection techniques considered here. In Section 3, we discuss the n-gram techniques that we consider in this paper. Section 4 gives details about the datasets we use in our experiments. In Section 5, we present our detection results and evaluate the performance of our proposed techniques. Finally, in Section 6, we conclude the paper and discuss future work.

2. Background

In this section, we consider network intrusion detection and review select strategies that have been applied to this problem. We also briefly cover Pearson's χ^2 statistic.

A network intrusion detection system examines traffic in real time and raises alarms when potentially malicious data is detected (Scarfone and Mell, 2007). Broadly speaking, intrusion detection systems can be categorized as signature-based or anomaly-based. A signature-based system looks for known patterns of attack, while an anomaly-based system looks for unusual behavior. Examples of signature-based network intrusion detection systems include (Bro network security monir; Snort). While such systems are able to detect known attacks effectively and efficiently, they cannot prevent zero day attacks or variants of known attacks.

The advantage of an anomaly based approach is that it has the potential to detect previously unknown attacks. But such techniques tend to be complex, costly, and subject to false positives. Designing an anomaly based detection strategy which is accurate and efficient poses a significant challenge to security researchers.

2.1. *Related research*

In this section, we discuss some examples of relevant research. Our focus is on intrusion detection techniques that employ byte level analysis of web traffic.

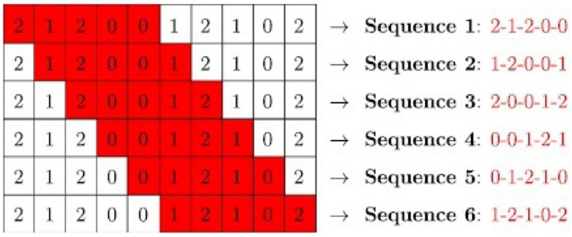
The paper (Ariu et al., 2011) proposes a technique, "HMMPayl," that models the structure of benign HTTP traffic using hidden Markov models (HMM). HMMs are a well-known machine learning technique that are applicable to a wide range of real-world problems (Rabiner, 1989; Stamp).

In HMMPayl, overlapping n-grams of bytes are extracted from benign HTTP traffic packets, as illustrated for 5-g in Fig. 1. These n-grams are then used to train multiple hidden Markov models, using different initial starting points for each. In the detection phase, each of the trained HMMs is used to score observed n-grams, as illustrated in Fig. 2. In Ariu et al. (2011), various combinations of the HMM scores are tested. Specifically, the score combinations analyzed are the mean, the maximum, the minimum, and a geometric mean. The resulting score is used to classify a packet as benign or malicious.

It is worth noting that the HMMPayl method in Ariu et al. (2011) is relatively costly, in part due to the multiple HMM scores per n-gram. The HMM training process can be viewed as a discrete hill-climb, where we are only assured of a local maximum. Therefore, training multiple HMMs with different starting points is a reasonable strategy.

In HMMPayl, the number of hidden states is selected to be equal to the length of the n-gram under consideration. However, it is not clear that there is any logical connection between the length of an n-gram and the optimal number of hidden states. In fact, for many applications where the number of observations and/or sequence length is large, HMMs are successfully constructed using a very small number of hidden states N; often, $N = 2$ or $N = 3$ suffices. For example, in Wong and Stamp (2006), an HMM is used to classify malware based on extracted opcode sequences. In Wong and Stamp (2006), $N = 3$ is shown to be optimal (and only marginally better than $N = 2$) yet the number of distinct opcodes is large, and the scored sequences contain hundreds of observations. As another example, in (Stamp), an HMM is trained on English text using 27 distinct symbol (lower-case letters and word-space). With $N = 2$ hidden states, the structure of English is revealed (i.e., consonants and vowels). However, this information is "hidden" from the perspective of model development, and the actual training and scoring requires long sequences involving these 27 distinct symbols. These results are typical of HMM applications, where the number of hidden states bears no connection to the length of the training or scored sequences.

The paper (Wang and Stolfo, 2004) presents a detection technique based on n-gram analysis of bytes in web traffic. A model of benign traffic is constructed based on expected frequencies of bytes. Detection is then based on the Mahalanobis distance (Mahalanobis, 1936) between this model and the frequency vector of an incoming packet. The authors Wang and Stolfo (2004) present an improved version of their



2 1 2 0 0 1 2 1 0 2 → Sequence 1: 2-1-2-0-0
2 1 2 0 0 1 2 1 0 2 → Sequence 2: 1-2-0-0-1
2 1 2 0 0 1 2 1 0 2 → Sequence 3: 2-0-0-1-2
2 1 2 0 0 1 2 1 0 2 → Sequence 4: 0-0-1-2-1
2 1 2 0 0 1 2 1 0 2 → Sequence 5: 0-1-2-1-0
2 1 2 0 0 1 2 1 0 2 → Sequence 6: 1-2-1-0-2

Fig. 1 – HMMPayl feature extraction (Ariu et al., 2011).