Data Mining Methods for Anomaly Detection of HTTP Request Exploitations

Xiao-Feng Wang, Jing-Li Zhou, Sheng-Sheng Yu, and Long-Zheng Cai

Department of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan 430074, China
xfwang@wtwh.com.cn

Abstract. HTTP request exploitations take substantial portion of network-based attacks. This paper presents a novel anomaly detection framework, which uses data mining technologies to build four independent detection models. In the training phase, these models mine specialty of every web program using web server log files as data source, and in the detection phase, each model takes the HTTP requests upon detection as input and calculates at least one anomalous probability as output. All the four models totally generate eight anomalous probabilities, which are weighted and summed up to produce a final probability, and this probability is used to decide whether the request is malicious or not. Experiments prove that our detection framework achieves close to perfect detection rate under very few false positives.

1 Introduction

Web servers and Web-based programs are suffering sharply increasing amounts of web-based attacks. Generally, attacks against either web servers or web-based applications must be carried out through forging specially formatted HTTP requests to exploit potential web-based vulnerabilities, so we name this kind of attacks as HTTP request exploitations.

We provide a novel detection framework dedicated to detecting HTTP request exploitations. The framework is composed of four detection models. Each of the four detection models contains at lease one classifier. The classifiers calculate the anomalous probabilities for HTTP requests upon detection to decide whether they are anomalous or not. Eight classifiers of the framework present eight independent anomalous probabilities for a HTTP request. We assign a weight for each probability and calculate the weighted average as the final output.

2 Methodology

Web log file is composed of a large amount of entries that record basic information of HTTP requests and responses. We extract the source IP and URL of each entry and translate them into serials of name-value pairs. Let *avp=(a, v)* represent a name-value pair and in this paper we call the name-value pair an attribute which comprises name

L. Wang and Y. Jin (Eds.): FSKD 2005, LNAI 3614, pp. 320 – 323, 2005.

$$\underbrace{program = search}_{attribute\ 1}\ \underbrace{ip = 211.67.27.194}_{attribute2}\ \underbrace{hl = zh\text{-}CN}_{attribute3}\ \underbrace{q = computer + science}_{attribute4}$$

Fig. 1. A four-element attribute list is derived from an entry of a web server log file

a and attached value *v*. Figure 1 shows the example of an entry from a web server log file (ignoring some parts). Attribute list is a set of attributes derived from a web log entry. So a web log file can be mapped into a set of attribute lists. Let *PROB* be the output probability set derived from classifiers and let *WEIGHT* be related weight set. The final anomalous probability is calculated according to Equation 1. The *WEIGHT*, as a relatively isolated subject, is not introduced in this paper.

211.67.27.194 - [Time] "GET search?hl=zh-CN&q=computer+science" 200 2122

$$Final\ Anomalous\ probability = \sum_{\substack{p_m \in PROB \\ w_m \in WEIGHT}} p_m * w_m \tag{1}$$

3 Detection Models

The framework is composed of four independent detection models: attribute relationship, fragment combination, length distribution and character distribution. The corresponding principles for these models are introduced in this section.

3.1 Attribute Relationship

General speaking, there must be some fixed inter-relationships between the attribute lists derived from web log entries. For example, suppose that a web form page contains a hidden tag with a given value. Once a user submits this web form to a specified web program (say, a CGI program dealing with the web form), the given value will be presented in the submitted request as a query attribute instance. Apparently, the specified web program only accepts the given value attached to the hidden tag. This fact is described as an enumeration rule. If malicious users tamper with the given value attached to the hidden tag, the rule is broken. Further more, a HTTP request missing necessary query attribute breaks integrality rule and a HTTP request containing faked query attribute breaks validity rule. Interdependence rule prescribes the interdependence of two attribute instances. Suppose that a web program dealing with an order form accepts two arguments: *id* and *discount*. id is client's identity and discount is client's shopping discount ratio. If *Jason* enjoys 10% discount, an interdependence rule will be discovered between two attribute instances: (*id*, *Jason*) and (*discount,* 10%). If *Jason* tampers with his discount to 50%, this interdependence rule is broken. A set of attribute lists derived from a given web log file is used to mine the four types of rules. According to the respective kinds of rules, four classifiers calculate four anomalous probabilities: $P_{integrality}$, $P_{validity}$, $P_{enumeration}$ and $P_{interdependence}$. The first three are Boolean values and the last one is continuous value between 0 and 1.