Expert Systems with Applications 38 (2011) 12521–12530

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Anomaly detection techniques for a web defacement monitoring service

G. Davanzo *, E. Medvet, A. Bartoli

DI3 – Università degli Studi di Trieste, Via Valerio 10, Trieste, Italy

ARTICLE INFO

Keywords:
Security
Web defacement
Machine learning

ABSTRACT

The defacement of web sites has become a widespread problem. Reaction to these incidents is often quite slow and triggered by occasional checks or even feedback from users, because organizations usually lack a systematic and round the clock surveillance of the integrity of their web sites. A more systematic approach is certainly desirable. An attractive option in this respect consists in augmenting availability and performance monitoring services with defacement detection capabilities. Motivated by these considerations, in this paper we assess the performance of several anomaly detection approaches when faced with the problem of detecting web defacements automatically. All these approaches construct a profile of the monitored page automatically, based on machine learning techniques, and raise an alert when the page content does not fit the profile. We assessed their performance in terms of false positives and false negatives on a dataset composed of 300 highly dynamic web pages that we observed for 3 months and includes a set of 320 real defacements.

1. Introduction

Defacements are a common form of attack to web sites. In these attacks the legitimate site content is fully or partly replaced by the attacker so as to include content embarrassing to the site owner, e.g., disturbing images, political messages, forms of signature of the attacker and so on. Defacements are usually carried out by exploiting security vulnerabilities of the web hosting infrastructure, but there is increasing evidence of defacements obtained by means of fraudulent DNS redirections, i.e., by penetrating the DNS registrar rather than the web site. Recent examples of the two strategies include the massive breach suffered by a US hosting company (Hackers hit network solutions customers, 2010) and the redirection that affected a major search site in China (Baidu sues registrar over DNS records hack, 2010). Attackers may focus their efforts toward defacing a specific target site, but often they tend to follow a radically different pattern in which automated tools locate thousands of web sites that exhibit the same vulnerability and can thus be defaced simultaneously, with just a few keystrokes (Danchev, 2008; MultiInjector v0.3 released, 2008).

It seems fair to say that, unfortunately, defacements have gained a sort of first-level citizenship in the Internet. Nearly 1.7 million snapshots of defacements were stored during 2005–2007 at Zone-H, a public web-based archive (http://www.zone-h.org). Back in 2006, the annual survey from the Computer Security Institute observed that "defacement of web sites continues to plague organizations" (Gordon, Loeb, Lucyshyn, & Richardson, 2006). Not only the scenario did not change significantly in the following years, the latest version of this survey reports that the percent of responders which suffered this kind of attack in 2009 more than doubled with respect to 2008—14% vs. 6% (CSI, 2009).

Another side of the problem is the reaction time, i.e., the time it takes to an organization to detect that its site has been defaced and react appropriately. Anecdotal evidence suggests that this is a relevant issue, even at large organizations. To mention just a few examples, the Company Registration Office in Ireland was defaced in December 2006 and remained so until mid-January 2007 (CRO, xxxx). Several web sites of Congressional Members in the house.gov domain were defaced "shortly after President Obama's State of the Union address" and were still defaced at "4:10 am EST" (Congressional web site defacements follow the state of the union, 2010). A systematic study of the reaction time, performed by means of real-time monitoring of more than 60,000 defaced sites extracted on-the-fly from ZoneH, showed that 40% of the defacements in the sample lasted for more than 1 week and 37% of the defacements was still in place after 2 weeks (Bartoli, Davanzo, & Medvet, 2009). The cited study also showed that these figures do not change significantly in sites hosted by Internet providers (and as such presumably associated with systematic administration) nor by taking into account the importance of these sites as quantified by their PageRank index. These data confirm the intuition that web sites often lack a systematic surveillance of their integrity and that the detection of web defacements is usually demanded to occasional checks by administrators or to feedbacks

* Corresponding author.
E-mail addresses: giorgio.davanzo@gmail.com (G. Davanzo), emedvet@units.it (E. Medvet), bartoli.alberto@units.it (A. Bartoli).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.038

from users. Indeed, reaction to a defacement occurred recently at Poste.it, one of the largest financial institutions in Italy, was not triggered by site administrators but by a user who called the police because he happened to find the site defaced on Friday late afternoon (Le poste dopo l'attacco web Non violati i dati dei correntisti, 2009). Such an extemporaneous approach is clearly unsatisfactory. A more rigorous and systematic approach capable of ensuring prompt detection of such incidents is required.

An attractive option in this respect consists in augmenting availability and performance monitoring services (e.g., 13 free & cheap website monitoring services, 2008) with defacement detection capabilities (Bartoli & Medvet, 2006; Medvet & Bartoli, 2007). Since these services are cheap and non-intrusive, organizations of essentially any size and budget could indeed afford to exploit these services for performing a systematic and round the clock surveillance against defacements. Indeed, economics seems to play a key role in this scenario. Quantifying the cost of late detection of a defacement is very difficult and weighing this cost against the cost of better security-related skills, practices and technologies is even more difficult. In this respect, an external service that is cheap, can be joined with just a few clicks, without installing any software and without any impact on daily operating activities seems to be a sensible framework for promoting systematic surveillance and quicker detection on a large scale. A service of this kind would also be able to detect defacements induced by fraudulent DNS redirections. Attacks of this form are increasingly more diffused (Baidu sues registrar over DNS records hack, 2010; Google blames DNS insecurity for web site defacements, 2009; Hackers hijack DNS records of high profile new zealand sites, 2009; Puerto rico sites redirected in DNS attack security, 2009) and are very difficult to detect with detection technologies deployed locally on the monitored site.

A crucial problem for successful deployment of a defacement detection service consists in being able to cope with dynamic content without raising an excessive amount of false alarms. Site administrators could provide a description of the legitimate contents for their sites at service subscription time. This option requires defining a site-independent way for collecting this information, whose quality and amount should suffice to cover all relevant portions and content of the monitored pages. Moreover, the option assumes that site administrators indeed have time and skills for actually providing those descriptions. A radically different approach consists in extracting the relevant information automatically by means of *machine learning* techniques. The potential advantages of this approach are obvious, as site administrators would only need to provide the URL of the monitored page and simply wait for a few days—until the service will have constructed a profile of the legitimate content automatically. The implicit assumption is that *anomaly detection* (Denning, 1987; Gosh, 1998; Mutz, Valeur, Vigna, & Kruegel, 2006; Kruegel & Vigna, 2003) is indeed a feasible approach for a monitoring service of this kind, i.e., that defacements indeed constitute anomalies with respect to an established profile of the monitored resource and that false positives may indeed be kept to a minimum despite the highly dynamic nature of web resources.

In this paper we elaborate on this idea and assess the performance of several machine learning approaches when faced with the defacement detection problem. Clearly, by no means we intend to provide an extensive coverage of all the frameworks that could be used (Chandola, Banerjee, & Kumar, 2009; Patcha & Park, 2007; Tsai, Hsu, Lin, & Lin, 2009). We chose to restrict our analysis to key approaches that have been proposed for attack detection at host and network level (Boser, Guyon, & Vapnik, 1992; Breunig, Kriegel, Ng, & Sander, 2000; Kim & Kim, 2006; Lazarevic, Ertoz, Kumar, Ozgur, & Srivastava, 2003; Mukkamala, Janoski, & Sung, 2002; Ramaswamy, Rastogi, & Shim, 2000; Ye, Chen, Emran, & Vilbert, 2000; Ye, Emran, Chen, & Vilbert, 2002; Ye, Li, Chen, Emran, & Xu, 2001; Yeung & Chow, 2002). The analysis includes a detection algorithm that we have developed explicitly for defacement detection and that exploits a fair amount of domain-specific knowledge (Bartoli & Medvet, 2006; Medvet & Bartoli, 2007).

Our evaluation is based on a dataset composed of 300 highly dynamic web pages that we observed periodically for 3 months and on a sample of 320 defacements extracted from ZoneH. Each detection algorithm is hence tested against its ability in not raising false alarms or missing defacements.

2. Our test framework

We developed a prototype framework, which works as follows. We consider a source of information producing a sequence of *readings* $\{r^1, r^2, \ldots\}$ which is input to a *detector*. The source of information is a web page univocally identified by an URL; each reading r consists of the document downloaded from that URL. The detector will classify each reading as being *negative* (legitimate) or *positive* (anomalous). The detector consists internally of a *refiner* followed by an *aggregator*, as represented in Fig. 1.

2.1. Refiner

The refiner implements a function that takes a reading r and produces a fixed size numeric vector $v \in \mathbb{R}^n$. The refiner is internally composed by a number of *sensors*. A sensor is a component which receives as input a reading and outputs a fixed size vector of real numbers. The output of the refiner is obtained by concatenating outputs from the 43 different sensors of our prototype and corresponds to a vector of 1466 elements (Medvet & Bartoli, 2007). Sensors are functional blocks and have no internal state: v depends only on the current input r and does not depend on any prior reading.

Sensors are divided in five categories, accordingly to the way in which they work internally. Table 1 indicates the number of sensors and the corresponding size for the vector v portion in each category.

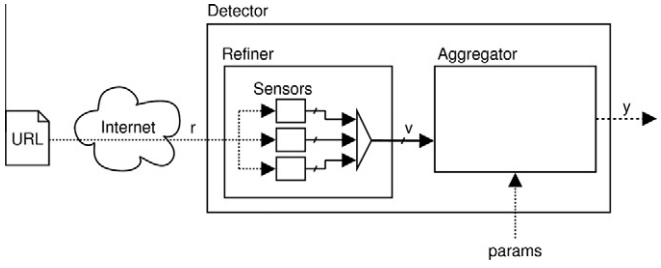

Fig. 1. Detector architecture. Different arrow types correspond to different types of data.

Table 1
Sensor categories and corresponding vector portion sizes.

Category	Number of sensors	Vector size
Cardinality	25	25
RelativeFrequencies	2	117
HashedItemCounter	10	920
HashedTree	2	200
Signature	4	4
Total	43	1466