Resisting Web Proxy-based HTTP Attacks by Temporal and Spatial Locality Behavior

Yi Xie, S. Tang, Y. Xiang and J. Hu

Abstract—A novel server-side defense scheme is proposed to resist the Web proxy-based distributed denial of service attack. The approach utilizes the temporal and spatial locality to extract the behavior features of the proxy-to-server traffic, which makes the scheme independent of the traffic intensity and frequently-varying Web contents. A nonlinear mapping function is introduced to protect weak signals from the interference of infrequent large values. Then a new hidden semi-Markov model parameterized by Gaussian-mixture and Gamma distributions is proposed to describe the time-varying traffic behavior of Web proxies. The new method reduces the number of parameters to be estimated, and can characterize the dynamic evolution of the proxy-to-server traffic rather than the static statistics. Two diagnosis approaches at different scales are introduced to meet the requirement of both fine-grained and coarse-grained detection. Soft-control is a novel attack response method proposed in this work. It converts a suspicious traffic into a relatively normal one by behavior reshaping rather than rudely discarding. This measure can protect the quality of services of legitimate users. The experiments confirm the effectiveness of the proposed scheme.

Index Terms—Traffic analysis, traffic modeling, distributed denial of service attack, attack detection, attack response

1 INTRODUCTION

DISTRIBUTED denial of service (DDoS) attacks have been a continuous critical threat to the Internet since ten years ago. Their implementation keeps on evolving and becomes more subtle. In [1], a new attack pattern that utilizes the edge servers of content delivery networks (CDNs) to launch DDoS attacks to the Web servers was introduced. Soon after this work, an actual variant of such an attack was reported in [2] where the attack was implemented by the widely deployed Web proxies instead of the edge servers of CDNs mentioned in [1].

The Web proxy-based HTTP attack is more flexible and covert than most of existing DDoS attacks. The difficulty of detection lies in three aspects: (i) real attacking hosts are unobservable to the origin server since they are shielded by the hierarchical Web proxies; (ii) a Web proxy may be passively involved in an attack event and may unconsciously act as an attacker; (iii) observed from the victim server, both legal and illegal traffic comes from the same sources (i.e., Web proxies). Although most of the large-scale official attacks are usually configured to have high security, they cannot avoid being abused for the proxy-based attacks. Our literature survey showed that currently few studies have paid attention to this issue due to its concealment. Thus, this type of attacks may bring new challenges to existing network security systems. Motivated by these issues, a novel resisting scheme is proposed to protect the origin server from Web proxy-based HTTP attacks in this work.

The proposed scheme is based on network behavior analysis [3]. It maps a Web proxy’s access behavior to a hidden semi-Markov model (HsMM) [4][5] which is a typical double stochastic processes model. The output process of an HsMM profiles the observable varying-process of a proxy-to-server traffic. The hidden semi-Markov chain of an HsMM describes the transformation of a proxy’s internal behavior states, which can be considered as the intrinsic driving mechanism of a proxy-to-server traffic. Based on this behavior model, detecting the abnormality of a Web proxy can be achieved by measuring the deviation between an observed behavior and the Web proxy’s historical behavior profile. Long-term and short-term behavior assessment methods are proposed. Long-term behavior assessment issues warnings on a large scale, while short-term behavior assessment locates abnormal request sequences embedded in the proxy-to-server traffic. A new “soft-control” scheme is proposed for attack response. The scheme reshapes the suspicious sequences according to the profile of a proxy’s historical behavior. It converts a suspicious sequence into a relatively normal one by partly discarding its most likely malicious requests instead of denying the entire sequence. Thus, it can protect the HTTP requests of legitimate users to the greatest extent possible from being discarded.

In summary, compared with most of the existing
and our previous works [6][7], the novelty of this work lies in: (i) it is focused on resisting Web proxy-based HTTP attacks and realizes the early detection without any cooperations of mid Web proxies; (ii) the approach is independent of traffic intensity and frequently-varying Web contents. It has good stability and need not frequently update model’s parameters; (iii) long- and short-term behavior assessment methods enable the multi-granularity diagnosis, while the “soft-control” scheme can improve the quality of services of normal users.

The remainder of the paper is organized as follows. Section 2 briefly reviews the Web proxy-based HTTP attacks. Section 3 summarizes the existing related work. Section 4 introduces the proposed scheme. In Section 5, the proposed method is applied to the proxy behavior modeling and attack detection. In Section 6, experiment results are shown, and then some key issues of the scheme are discussed in Section 7. Finally, conclusion is given in Section 8.

2 WEB PROXY-BASED HTTP ATTACK

A Web proxy may be turned into an attacker by two steps: (i) attacker sends attack requests to a Web proxy and forces it to forward the attack requests to the origin server; (ii) attacker disconnects connections between itself and the proxy. In step 1, two methods can be used to penetrate through the Web proxies: requesting dynamic documents or setting “Cache-Control: no-cache” in the headers of HTTP requests. Repeating these steps, a single host can simultaneously trigger a lot of Web proxies to attack a Web server without the need of invading them.

The attraction of such an attack lies in three aspects: (i) it enables the attacking host to break through the client-side restrictions by connecting different Web proxies via HTTP protocols; (ii) resisting such an attack by the mid Web proxies is not a practical approach, due to lack of cooperation mechanisms between server and proxies, in particular those uncontrollable private proxies; (iii) such an attack may confuse most of the existing detection systems designed for the traditional DDoS attacks due to two reasons: first, the origin server cannot directly observe and diagnose the terminal hosts shielded by the hierarchical proxy system; second, the attack traffic is mixed with the regular client-to-proxy traffic by each proxy which forwards the traffic. In the final aggregated proxy-to-server traffic, there is no obvious difference between the normal traffic and the attack traffic except their underlying purposes. Thus, the victim server is hard to accurately identify and filter the attack requests.

3 RELATED WORK

Countering DDoS attacks has attracted much attention during the past ten years. Traditional defense techniques, e.g., [8][7], focus on the network-layer DDoS attacks and use TCP and IP properties to discover attack signals. The summaries of these methods can be found in [9]. Since the HTTP-based DDoS attacks work on the application-layer and employ a new attack mechanism, the classical methods designed for the network-layer attack are no longer applicable.

Recently, HTTP-based DDoS attacks have been receiving more attention. In [10], clients are evaluated by trust management mechanism, and then the application layer DDoS is mitigated by giving priority to good users. In [11], the zombies is identified by automatically-changing puzzle, and then the HTTP requests of suspected hosts are blocked. In [12], a model is proposed to profile the normal access behavior based on four attributes of web page request sequences. The reconstruction error of a given request sequence is used as a criterion for detecting DDoS attacks. In [13], the flow correlation coefficient was used to measure the similarity amongst suspicious flows, and then the HTTP-based DDoS attacks from normal flash crowds were discriminated by the results of measurement. A traceback method was explored for the DDoS attacks based on entropy variations in [14]. In our previous work [15], user browsing behavior is applied to distinguish the anomalous HTTP requests from those of normal users. In [16], a multidimensional access matrix is defined to capture the traffic behavior of flash crowds and detect HTTP attacks which mimic or occur during the flash crowd event of a popular Web site.

However, the potential assumption of all these schemes is that the attacking hosts are directly connected to the victim server. Thus, the victim server can distinguish the traffic launched by different hosts. Once a host’s traffic does not fit to the predefined criterion of a given model, the defensive system will treat it as a suspect node and block its HTTP traffic. However, most of terminal hosts are shielded by the hierarchical Web-proxy system in the actual Internet scenario. Thus, the source of an incoming HTTP request observed by the victim server is usually the last Web-proxy which connects to the victim server directly and can be verified by the victim server. Since it is difficult for the origin server to infer the actual source of each HTTP request, resisting the attacks by traditional methods may decrease the quality of services of normal users.

In [1], the authors conceived a way that attackers can utilize the edge servers of CDNs to launch HTTP-based DDoS attacks to origin servers. Their solutions include enhancing the communication policies between CDNs and content providers, and improving the forwarding process of edge servers of CDNs. However, these solutions are not suitable to the issue of this paper. Since almost all CDNs are commercial systems, content providers can consult with each CDN and setup secure communication policies. However, it is...
impossible for a server to consult with all Internet Web proxies (including official and unofficial). Proxy behavior and its anomaly detection were first investigated in our previous work [6]. However, its behavior model is not fully parametric and is not independent of traffic intensity.

In term of the mathematical model used to describe the proxy behavior, the Hidden Markov Model (HMM) is a classical approach for modeling time series with the assumption that the hidden state process is a Markov chain. However, the state sojourn in the HMM is implicitly assumed to be constant or exponentially distributed, which limits its practical application. Compared with the HMM, the discrete HsMM allows for more general sojourn distributions, which make it widely applied in many areas, e.g., mobility tracking, activity recognition, and inference for structured video sequences [5]. However, the classical algorithm [4] designed for the discrete HsMM is computationally too expensive to be of practical use in many applications [5]. In [17], state durations of the HsMM was first parameterized by Gamma distribution. However, its main drawback is that the classical Newton’s method with second-order convergence is applied to solve the parameters of Gamma-distribution, whose iterative convergence speed is slow for large-scale real-time applications. Considering the real-time requirement of anomaly detection, a new iterative method based on Forward-backward algorithm [18] and eighth-order convergence [19] is proposed in this work.

4 THE PROPOSED SCHEME

4.1 Model Definition

The primary aim of the proposed scheme is to protect the origin server from the Web proxy-based HTTP attacks. To simplify the problem, attack traffic is assumed to begin from Web proxies instead of its real sources. This assumption is reasonable since the victim can only observe the proxies and our goal is to filter malicious traffic instead of traceback.

Like most of other physical processes in nature, a Web proxy’s access behavior can be regarded as a combination of external manifestations (e.g., temporal and spatial locality) and intrinsic driving mechanisms (e.g., normality or abnormality). The external manifestations are observable and usually controlled by the intrinsic driving mechanisms which cannot be accurately obtained by the origin server but can be estimated by the observable features of proxy-to-server traffic. According to the basic structure of the HsMM illustrated in Fig. 1, a Web proxy’s access behavior can be directly mapped to an HsMM.

The basic HsMM consists of a pair of stochastic processes: the observed process \( \{ o_t \} \) and the hidden semi-Markov state process \( \{ X_t \} \), where \( t \in \{ 1, 2, \ldots \} \) is the number of observation (also called event). \( \{ o_t \} \) is associated with \( \{ X_t \} \) by the conditional distribution depending on the state process which is a finite-state semi-Markov chain. The conditional distributions usually overlap and so, in general, a specific observation can arise from more than one state. Thus \( \{ X_t \} \) is not observable directly through \( \{ o_t \} \) but can be estimated. \( \{ o_t \} \) itself may be either discrete or continuous, univariate or multivariate.

To Model a Web proxy’s access behavior by an HsMM, each hidden semi-Markov state represents a driving mechanism of a type of proxy-to-server traffic. Transition between two different Markov states represents the changes of driving mechanism. Duration of a particular semi-Markov state represents the dwell time of its corresponding driving mechanism. Conditional on the hidden semi-Markov chain the dynamic process of the external manifestation of a Web proxy’s access behavior is modeled by the output process of a HsMM.

In this paper, three driving mechanisms are defined: Normality, Transition and Abnormality. The problem of resisting the proxy-based HTTP attack is equivalent to searching the Abnormality state of a Web proxy’s access process and filtering those suspicious requests caused by the Abnormality state. Given a behavior model (i.e., HsMM), this objective can be achieved by seeking the optimal underlying semi-Markov chain for an observed proxy-to-server traffic.

The model’s output process can be constructed by many common features of network traffic, e.g., arrival rate and packet size. However, for the issue concerned in this work, the observable parameters should meet the following requirements: (i) being independent of traffic intensity and constantly-varying Web contents and URLs; (ii) being able to realize the early detection. Considering these requirements, Temporal and Spatial Locality (TSL) are exploited to extract the proxy-to-server behavior.

4.2 Stack distance model for temporal locality

Temporal locality [20] of reference has been widely applied in many fields, e.g., program behavior [21], reference pattern of Web access [22], and Web proxy cache replacement strategy[23]. Temporal locality refers to the property that a referencing behavior in the recent past is a good predictor of the referencing
behavior to be seen in the near future, whereas the resource popularity metric only represents the frequency of the requests without indicating the correlation between a reference to a document and the time since it was last accessed.

Here we resort to the concept of stack distance [24][22]. The files are assumed to be placed on a stack such that, whenever a file f is requested, it is either pulled from its position in the stack and placed on the top, or simply added to the stack if the file is not yet in the stack. The stack distance for the request is then the distance of f from the top in the former case or undefined (or \( \infty \)) in the latter case.

Starting with an empty stack, the reference stream is \( \mathcal{F}_i = \{f_1, f_2, \ldots, f_i\} \), where \( f_i \) denotes the name of the \( i^{th} \) requested file. Index \( i \) indicates that \( i \) requests have already arrived at a server. Thus, the unit of time is one request, i.e., an incoming request represents a new event occurring. We define the least recently used (LRU) stack \( \mathcal{L}_i \), which is an ordering of all files of a server by the recency of usage. Thus, at index \( i \), the LRU stack is given by \( \mathcal{L}_i = \{u_1, u_2, \ldots, u_N\} \), where \( u_1, u_2, \ldots, u_N \) are the files of the server and \( u_1 \) is the most recently accessed file, \( u_2 \) the next most recently referenced, etc. In other words, \( u_1 \) is just accessed at index \( i \), i.e., \( f_i = u_1 \). Whenever a reference is made to a file, the stack must be updated. Considering that \( f_{i+1} = u_j \), then the stack becomes \( \mathcal{L}_{i+1} = \{u_j, u_1, u_2, \ldots, u_N\} \). Suppose now that \( \mathcal{L}_{i-1} = \{u_1, u_2, \ldots, u_N\} \) and \( f_i = u_j \), i.e., the request \( f_i \) is at distance \( j \) in stack \( \mathcal{L}_{i-1} \). Let \( d_i^T \) denote the stack depth (i.e., temporal locality) of a document referenced at index \( i \). Then, a new relation can be obtained by the following equation “if \( f_i = u_j \) then \( d_i^T = j \)”, where \( j \) denotes the stack depth of the requested document at index \( i \). Thus, the reference trace \( \{f_1, f_2, \ldots, f_i\} \) defines a numerical distance sequence \( \{d_1^T, d_2^T, \ldots, d_i^T\} \). In particular, \( d_1 = +\infty \) since the first request cannot be for a file in the empty stack.

It has been pointed out that about 10% of the web contents of a web-site may draw 90% of the access [25]. This finding indicates that the statistical properties of stack distance are unrelated to the frequently changing names, URLs or contents of documents.

4.3 Joint entropy for spatial locality

Spatial locality [22] refers to the property that objects neighboring an object frequently accessed in the past are likely to be accessed in the future. For example, when a home page is requested, all its embedded objects are likely to be accessed at the same time. Because spatial locality indicates correlation among a cluster of HTTP requests, capturing spatial locality can help mine the access behavior of Web proxies. Different from [22], here a new method is used to quantify the spatial locality. Let \( \Omega \) denote a set of addressable Web objects, \( (a, b) \in \Omega \), \( p_{ab}^w \) be the joint probability mass function when \( a \) and \( b \) are simultaneously accessed within the \( w^{th} \) time window, then \( e_{w}^{ab} = -p_{ab}^w \log(p_{ab}^w) \) denotes the joint entropy of \( a \) and \( b \) of the \( w^{th} \) time window. Assuming the reference stream within the \( w^{th} \) time window is denoted by \( \mathcal{F}_w = \{f_1^w, f_2^w, \ldots, f_n^w\} \), where \( f_i^w \in \Omega \), then the spatial locality \( d_{(w,i)}^S \) of the \( i^{th} \) reference of the \( w^{th} \) time window is calculated by:

\[
d_{(w,i)}^S = \frac{1}{|\mathcal{F}_w|} \sum_{f_j^w, f_j^w} e_{w}^{u_j^w, u_j^w}, \quad (f_j^w, f_j^w) \in \mathcal{F}_w
\]

Hence the spatial locality stream \( \{d_{(1,1)}^S, \ldots, d_{(1,|\mathcal{F}_1|)}^S, \ldots, d_{(|\mathcal{W}|,1)}^S, \ldots, d_{(|\mathcal{W}|,|\mathcal{F}|)}^S\} \) of the entire reference stream \( \{\mathcal{F}_1, \ldots, \mathcal{F}_W\} \) can be obtained by combining \( d_{(w,i)}^S \) of all time windows and concatenating them together, where \( W \) is the number of time windows.

4.4 Nonlinear algorithm for data mapping

Locality is analogous to heavy-tail distribution [22]. Thus, the infrequent strong signals located at the “tail” of the distribution may seriously interfere with the analysis of the front weak signals. In order to protect the information embedded in the small observed data which occur relatively frequently, we define a nonlinear mapping function \( \psi(x) \):

\[
\psi(x) = \begin{cases} 
Ax/\left[1 + \ln(A)\right], & 0 \leq x \leq 1/A \\
\left[1 + \ln(Ax)\right]/\left[1 + \ln(A)\right], & 1/A \leq x
\end{cases}
\]

where \( x \) is the normalized variable. Equation (2) protects the weak input signals by the linear mapping and penalizes the infrequent large input signals by the logarithmic compression function. Similar to the well-known A-law of Pulse Code Modulation, the constant \( A \) is a compression parameter which affects compression characteristic of the raw data. In practice, \( A \) can be decided by the distribution of training data. For example in our experiments, roughly 80% values of the temporal locality are smaller than 100. Thus, we can utilize the linear part of the mapping function to protect the first 80% values and apply the logarithmic part to compress the remaining 20% large values.

4.5 Parametric GGHsMM

In order to reduce the number of parameters, here a new parametric HsMM is introduced to describe the double stochastic processes of a proxy’s behavior: observable TSL process and the underlying state process. The parametrization includes two parts: the output distributions and the state duration distributions of the HsMM are parameterized by the mixture Gaussian distributions and Gamma distributions, respectively. We use GGHsMM to denote this new model.

The reasons for using the GGHsMM are: (i) it has been proved that a finite mixture of Gaussian components can model/approximate any continuous distribution with arbitrary precision if a sufficient number of components is provided and the parameters of the
model are chosen correctly [26]. This approach can be applied to a wide range of problems without any assumption with respect to distributional properties of the raw data analyzed; (ii) Gamma distribution is a flexible distribution to express different distributions (e.g., exponential/right-skewed/Gaussian distribution) of practical signals by adjusting its two parameters. Moreover, our preliminary experiment has showed that the Gamma distribution is more flexible than other single-form distributions to fit the various duration distributions of the hidden states; (iii) the GGHSMM has fewer parameters to be estimated than the discrete HsMM, which greatly reduces the computational complexity.

Let \( M \) denote the hidden state space. The output process at the \( t \)th event is assumed to depend only on the \( t \)th state of the underlying semi-Markov chain, i.e., \( P_r[\tilde{O}_t = \tilde{o}_t | \tilde{O}_{t-1} = \tilde{o}_{t-1}, X_t = x_t] = P_r[\tilde{O}_t = \tilde{o}_t | X_t = x_t] \). Thus, the output probability function \( b_m(\tilde{o}_t) \) is given by:

\[
b_m(\tilde{o}_t) = \frac{\sum_{k=1}^{K} c_{mk} b_{mk}(\tilde{o}_t)}{\sum_{k=1}^{K} c_{mk} N(\tilde{o}_t, \tilde{\mu}_{mk}, \Sigma_{mk})}\]

where \( K \) is the number of Gaussian components in state \( m \), \( N(\tilde{o}_t, \tilde{\mu}_{mk}, \Sigma_{mk}) \) denotes the multidimensional normal density function with mean \( \tilde{\mu}_{mk} \) and covariance matrix \( \Sigma_{mk} \) for the \( k \)th component in state \( m \). \( c_{mk} \) satisfies the following stochastic constraints: \( c_{mk} \geq 0 \) and \( \sum_{k=1}^{K} c_{mk} = 1 \) for \( m \in M \) and \( k \in [1, K] \). Thus, the \( b_m(\tilde{o}_t) \) is properly normalized, i.e., \( \int_{-\infty}^{\infty} b_m(\omega)d\omega = 1 \).

Let \( t_i^r \) denote the residing time of state \( X_t \), where \( \mathbb{D} \) is the possible state duration. Then the state duration probability function \( p_m(d) \) is given by:

\[
p_m(d) = \frac{\sum_{k=1}^{K} c_{mk} d^d e^{-(d^d - \omega_m^d) / \nu_m} / \nu_m^{d} \omega_m^d}{\sum_{i=1}^{\nu_m} \nu_m^{d} \omega_m^d} \quad \nu_m, \omega_m > 0
\]

where \( \Gamma(\nu) \) is the Gamma function and can be calculated by \( \Gamma(\nu) = (\nu + 1) !, \nu \in \mathbb{Z}^+ \). The mean value of \( d \) is \( \nu_m / \omega_m \) and its variance is \( \nu_m / \omega_m^2 \).

To overcome the shortcomings of the existing algorithms discussed in Section 3, a new iterative method based on the Forward-backward algorithm [18] and an eighth-order convergence [19] is proposed. The derivation is given in the Appendix.

5 IMPLEMENTATION OF THE SCHEME

As Fig. 2 shows, the scheme includes 3 phases: data extraction, model training, and detection and control.

5.1 Data extraction and model training

The detection system extracts a proxy’s TSLs from its reference string and generates a TSL string

\[((d^T_1, d^T_2), (d^T_1, d^T_2), ...), \] which is further handled by nonlinear mapping function and forms the final observed process \( (d^T_1, d^T_2, (d^T_1, d^T_2), ...), \)

Let the two-dimension vector series \( \{\tilde{o}_1, \tilde{o}_2, ...\} \) denote the observed process, where \( \tilde{o}_i = (d^T_i, d^T_i^\prime) \). Let \( W \) denote the total number of time windows (batch arrivals) in the observed process. Each time window (usually 1 second) includes an observation sequence. Then, two parameters, i.e., Behavior Index (BI) and Structure Factor (SF), are defined to measure the normality of proxy’s behavior. BI of the \( w \)th time window is defined by:

\[
BI_w = \frac{P[\tilde{\omega}_1^w T_w, \tilde{\omega}_2^w | T_w, \lambda]}
\]

where \( T_w \) and \( \tilde{\omega}_2^w \) denote the total number of requests and observation sequence of the \( w \)th time window, respectively. \( \tilde{\omega}_1^w T_w \) is the optimal hidden state sequence corresponding to \( \tilde{\omega}_1^w \). Let \( \mathcal{L}_w = P[\tilde{\omega}_1^w | T_w, \lambda] \) denote the likelihood of \( \tilde{\omega}_1^w \) fitting to the parameter set \( \lambda \). Given \( \lambda, \tilde{\omega}_1^w T_w \) can be derived by the Viterbi algorithm [5] or the maximum a posteriori (MAP) estimate introduced in the Appendix. The total BIs of training data are denoted by \( BI = \{BI_1, ..., BI_W\} \), each of which is independent and identically distributed (iid) and follows the Gaussian distribution, i.e., \( BI_w \sim N(\mu_{BI}, \sigma_{BI}) \). Given a width of confidence interval, the parameters \( (\mu_{BI}, \sigma_{BI}) \) can be estimated by maximum likelihood estimation (MLE) [27].

SF of the \( w \)th observed sequence is defined by

\[
SF^w = \frac{\sum_{i=1}^{\nu_m} \nu_m \omega_m^d}{\sum_{i=1}^{\nu_m} \nu_m \omega_m^d} \quad \text{where} \quad Num(i, w) \text{ denotes the number of requests generated by state } i \text{ of the } w \text{th time window, } \sum_{i=1}^{\nu_m} \nu_m \omega_m^d \text{ is the number of requests generated by state } i \text{ of the } w \text{th time window, } i \in M. \text{ Then, we obtain the expectations of SFs of all training data by: } SF^w = \frac{1}{W} \sum_{w=1}^{W} SF^w \text{ } i \in M.
\]

5.2 Detection and control

Two diagnosis methods are proposed. The long-term behavior diagnosis works in the large time scales. Its purpose is to provide early warning signal. Long-term behavior is made up of a string of consecutive observation processes. Its normality can be evaluated by comparing the distribution of consecutive BIs with \( N(\mu_{BI}, \sigma_{BI}) \) derived from training data in Phase 2. Many fit-test methods can be used, e.g., Chi-square test and Kolmogorov-Smirnov (K-S) test.
Most existing defense schemes use “hard-control” to deal with attack behavior, i.e., they reject the entire incoming suspicious request sequence. Since both the attack traffic and the normal traffic are mixed together in the Web proxy-based attacks, recklessly denying the whole sequence may seriously affect the quality of services of legitimate users. Here, we introduce a “soft-control” scheme for the attack response based on the underlying state process. The scheme reshapes the suspicious sequences according to the profile of normal behavior, i.e., converting the suspicious sequence into a relatively normal one by partly discarding its most likely malicious requests instead of denying the entire sequences. Given a suspicious reference string $f^w = \{f^w_1, \ldots, f^w_T\}$ of the $w^{th}$ time window, we define two auxiliary variables: discard number ($DN_i$) of requests generated by state $i$; global survival rate $SR_E = \frac{\varphi(BI_w, \mu_{BI}, \sigma_{BI})}{\varphi(\mu_{BI} + 2\sigma_{BI}, \mu_{BI}, \sigma_{BI})}$. Then, the reshaping algorithm for the “soft control” scheme is shown in Algorithm 1.

6 Experiment and analysis

The proposed scheme is implemented and evaluated in a simulation environment. The experiments include the following aspects: (i) the detection performance of conventional static statistical methods; (ii) the performance of the hidden state process when the attack behavior appears; (iii) the performance of both the long- and short-term behavior detection; (iv) the performance of the scheme in different attack scenarios; (v) the performance comparison between the proposed scheme and other methods. The simulation with real scenario data from the references is carried out to collect the results. The reason of using NS2 is two-fold: (i) it is a mature simulation technology which has been widely used in the network research; (ii) it can build different attack scenarios to test the detection performance.

6.1 Experiment configuration

Data: The proxy-to-server traffic is collected from the logs of a real campus server. It lasts 5 hours and includes twenty large-scale proxies.

Attack simulation: (i) in order to simulate a sophisticated attack, we deploy a computer to detect the structure of the Web site, e.g., the hyperlinks between the Web pages, the embedded objects of each Web page. This information will be used to construct a sophisticated attack traffic which seems more like a normal traffic; (ii) the attack network is assumed to include 20,000 compromised computers, which is the average size of current botnet reported by [29]. Considering the time zone and diurnal variations [29], we can roughly estimate the number ($N_A \approx 3000$) of online attack nodes based on the average bot infection rates reported in [30], and a simplified diurnal model of [29]; (iii) the attack routine can vary the intensity

Algorithm 1: Reshape suspicious reference string

Require:
The abnormal reference string: $f^w$;
The hidden state process of $f^w$: $x^w$;
The $BI_w$ of $f^w$ and its PDF $\varphi(x, \mu_{BI}, \sigma_{BI})$;
The Structure Factor: $SF_i$.

Ensure: Reshape $f^w$ and output: $f^w = \{\hat{f}^w_1, \ldots, \hat{f}^w_T\}$;

1: Calculate global survival rate $SR_E$;
2: Calculate final length of $f^w$ by $T_w = [T_w \cdot SR_E]$;
3: for $i = 1$ to $M$ do
4: $DN_i = 0$;
5: if $[T_w \cdot SF_i] \leq Num(i, w)$ then
6: $DN_i = Num(i, w) - [T_w \cdot SF_i]$;
7: end if
8: randomly mark $DN_i$ requests of state $i$ of $f^w$;
9: end for
10: Discard all marked requests of $f^w$;
11: Let $f^w$ = $f^w$ and output $f^w$;

Fig. 3. Simulation topology in NS2

[28]. Here, we take the latter as an example. Let $BI^F = \{BI_1, \ldots, BI_W\}$ denote the $BI$ sequence, each of which comes from a continuous empirical cumulative distribution function (CDF) $\Phi_W(x)$. Let $\Phi_0(x)$ denote the CDF of $N(\mu_{BI}, \sigma^2_{BI})$. The hypotheses to be tested are: $H_0 : \Phi_W(x) = \Phi_0(x)$ versus $H_a : \Phi_W(x) \neq \Phi_0(x)$.

If $H_0$ is accepted, it means that the examined long-term behavior approximates the profile of normal proxy behavior and is considered normal.

The short-term behavior diagnosis works in the small time scales. Its purpose is to locate those abnormal request sequences embedded in the proxy-to-server traffic. Since $Bls \sim N(\mu_{BI}, \sigma^2_{BI})$, intuitively, an incoming sequence’s behavior should be considered more normal if its $BI$ is much closer to the $\mu_{BI}$. Let $\varphi(x, \mu_{BI}, \sigma_{BI})$ denote the PDF of $N(\mu_{BI}, \sigma^2_{BI})$. Based on the normal distribution theory [27], we can infer that roughly $95\%$ of the $Bls$ of normal behavior sequences fall into $\mathbb{I}_{BI} = [\mu_{BI} - 2\sigma_{BI}, \mu_{BI} + 2\sigma_{BI}]$, and their probability density values fall into $\mathbb{I}_{pdf} = [\varphi(\mu_{BI} \pm 2\sigma_{BI}, \mu_{BI}, \sigma_{BI})]$. Thus, the detection threshold can be defined by the coordinates ($\mu_{BI} \pm 2\sigma_{BI}, \varphi(\mu_{BI} \pm 2\sigma_{BI}, \mu_{BI}, \sigma_{BI})$). Given an observed sequence, if $BI \in \mathbb{I}_{BI}$ or $\varphi(\mu_{BI}, \sigma_{BI}) \in \mathbb{I}_{pdf}$, its corresponding short-term behavior is regarded as normal; otherwise, it is considered abnormal.

The time scale of both long- and short-term can be adjusted according to the actual scenario, e.g., the time precision of logs and the computational performance of the detection system.
mode of its attack traffic by \( c_i \mathcal{I}_N \) and \( c_f \mathcal{F}_N \), where \( c_i > 0 \) is the intensity coefficient, \( c_f > 0 \) is the frequency coefficient, \( \mathcal{I}_N \) is the average arrival rate of normal traffic and \( \mathcal{F}_N \) is a ratio of the total seconds of normal traffic to the whole observation time; (iv) the Web proxy-based HTTP attack is simulated by NS2. The simulation topology is a three-level hierarchy, shown in Fig.3. It has 1 transit domain which has 11 transit nodes. One of the transit nodes connects to a stub domain which simulates the real campus network. Each of the remaining transit nodes has 3 stub domains which include 2 attack domains and 1 proxy domain. Each stub domain only connects to one transit node. There are no stub-stub edges. Each attack domain has (on average) 150 attack nodes while each proxy domain has 2 proxy nodes. The total number of attack nodes is about 3000. Each proxy node represents one of the proxies and replays its real proxy-to-server traffic collected from the real server. The attack routine is developed based on the well known test tool DoSHTTP [31]. Each attack node can select/change its Web proxy randomly.

6.2 Numerical results

In the following experiments, we use “normal” to denotes a proxy-to-server traffic without malicious behavior, use “polluted” to denote the mixed traffic of normal and attack requests when attack appears. Fig.4(a)-4(c) show 3 independent time processes of arrival rate, temporal locality and spatial locality, respectively. Since attacks do not always depend on heavy traffic but can be achieved by consuming the server’s computational resources (e.g., database query or cryptographic check), the intensity attributes do not present significant difference between the normal and the polluted.

Fig.4(d)-4(f) try to implement the detection by probability distribution. Each inset of Fig.4(d)-4(f) shows the relation between the false positive rate (FPR) and the detection rate (DR). As the figures show, given \( FPR = 0.2 \), the DRs of these statistical variables approximate to 0.3, 0.7 and 0.6, respectively. These results indicate that a simple statistical distribution is not enough to detect the proxy-based attacks. The main reason comes from the incompleteness of detection information. The observed data itself is a time-series which includes both the static numerical characteristics and the dynamic process information. General statistical methods assume that the observations are made up of lots of isolated and unordered data instead of the time-series data. Hence, they can only provide the static numerical characteristics of the observed data but cannot utilize the underlying dynamic process information, which
results in the degradation of detection performance.

In this experiment the proposed scheme is based on a 3-state GGHsSMM introduced in Section 4. Fig.5(a) shows the underlying state processes of the normal proxy-to-server traffic and the polluted one. When the proxy traffic is normal, state 1 and state 2 are the most visited states and there are few transitions among hidden states 2 and 3. Furthermore, the average duration of state 1 is longer than other states. These phenomena indicate that the proxy’s underlying behavior is quite stable in the normal access mode. During the attack period, transitions between states 2 and 3 are very frequently and the average duration of each state also becomes visibly shorter. These imply that the proxy behavior is deviating from its Normal State and becomes unstable. Thus, the hidden state process can show the normality of the proxy behavior. Fig.5(b) shows that the statistical distribution of the hidden states can provide a numerical method to measure the normality of the proxy behavior. For example, if the probability of occurrence of state 1 is lower than a given threshold or empirical value (e.g., 0.3), the system will deem the proxy behavior unusual and issue warnings.

Fig.6 shows the detection for the long-term proxy behavior. The horizontal axis denotes the length of the detection window. The vertical axis denotes the goodness of fitting where the smaller values represent higher normality. The solid-line box and the dotted-line box stand for the distribution of the goodness-of-fit of the normal behavior and the polluted behavior, respectively. The central mark of a box is the median of the goodness-of-fit, the edges of a box are the 25th and 75th percentiles. Fig.6 shows, (i) the goodness-of-fit of the normal behavior becomes better as the length of window gets longer, while the polluted has no significant changes; (ii) the gap between the normal and the polluted becomes wider as the length of window gets longer. These results show that the long-term behavior can be evaluated by the goodness-of-fit and the recognition performance becomes better as the size of observation data increases. Furthermore, the box plot provides a way to determine the minimal detection windows for the long-term behavior.

Fig.7 shows the results of the short-term behavior detection based on BI s. Fig.7(a) plots two logarithmic BI processes of the proxy-to-server traffic. Each logarithmic BI value is calculated by (5) per second. The logarithmic BI processes intuitively display the difference between the normal and the abnormal, i.e., they can be easily distinguished by the critical value −14. In order to derive a general theoretical method to obtain this threshold and realize the automatic numerical detection, three curves are analyzed in Fig.7(b), i.e., the logarithmic BI distribution (LBD) of the regular access behavior, a curve of Gaussian distribution fitting of the LBD and the LBD of the attack traffic. As the figure shows, given the width of the confidence interval (α = 0.05), the LBD of the proxies’ regular access behavior can be well fitted to the Gaussian distribution with \((\mu, \sigma) = (-11.442, 1.383)\), where \(\mu\) is the mean and \(\sigma^2\) is the variance. The confidence intervals of 95% for the parameter estimates on \(\mu\) and \(\sigma\) are respectively given by \(\mu_c, \sigma_c\) shown in Fig.7(b). Based on the approach of deriving threshold introduced in Section 5.2, the critical value can be theoretically calculated by \(\mu - 2\sigma \approx -14\), given \(FPR = 0.05\). Comparing the results with the distribution of the logarithmic BI s of the attack traffic fitting to the model shown in Fig.7(b), we can obviously see that this decision method works. It shows that we can apply BI s to implement the automated numerical diagnosis for the short-term proxy access behavior.

The main figure of Fig.8 illustrates the receiver operating characteristic (ROC) of the proposed scheme. The relationship between the threshold, the FPR and the DR is shown in the inset of Fig.8. The square-line is FPR vs. logarithmic BI while the circle-line is DR vs. logarithmic BI. It shows, \((FPR, DR) = (0.053, 93.59\%)\) if the threshold is set to −14.0. These results indicate that the detection threshold of BI can be chosen according to the actual performance requirements (i.e., FPR) in practical applications.

6.3 Performance comparison

In order to test the proposed scheme’s performance for different attack scenarios, we generate various attack traffic by modifying intensity coefficient \(c_i\) and frequency coefficient \(c_f\). Both \(c_i\) and \(c_f\) vary from 1/4 to 2 with step of 1/4. Thus, there are 64 attack scenarios generated for performance comparison.

Fig.9 plots the DRs of these attack scenarios under different FPRs. It shows that the proposed scheme is almost traffic volume independent. Even in the worst
Fig. 9. detection rate in various attack scenarios

Fig. 10. ROC curves of various detection methods

Fig. 11. box plot of hidden state distribution

cases, its DR is still over 0.88 (FPR = 0.05). This benefit comes from the behavior-based detection scheme. The proposed scheme is based on proxy behavior instead of traffic volume. It does not depend on the traffic intensity, but only compares the proxy’s current behavior with the normal behavior profile. Although traffic volume are not suitable for detection when attacks are based on low-traffic, the existence of attack traffic intensity, but only compares the proxy’s current behavior with its historical behavior profile. This enables the proposed scheme to achieve detection.

In Fig.9 also shows that the DRs increase with the growth of $c_i$ and $c_f$. This is because the detection becomes easier when the proxy’s access behavior is distorted seriously by the heavy attack traffic.

In Fig.10, the proposed scheme is compared with three commonly used methods: (i) HMM-based behavior detection; (ii) detection based on arrival rate; (iii) detection based on statistical distributions of TSL. Among these curves, we can see that the performance of both GGHsMM-based and HMM-based schemes are better than the others. In the low-FPR area FPR $\in (0, 0.1]$, the GGHsMM is better than the HMM.

7 Discussion

In Web proxy-based attacks, although Web proxies act on behalf of those real attack sources, the distributed nature of the attack still exists because the abused middle proxy-architecture is distributed in most of real network scenarios. Since the proposed scheme detects a proxy’s anomaly behavior by comparing its current behavior with its historical behavior profile without considering the other proxies participating in the same attack, detecting the distributed Web proxy-based HTTP attacks can be transformed into a series of relatively independent tasks. Thus, in a real network scenario the detection system can work in parallel to improve the efficiency of detection.

Based on the above introduction, the proposed scheme includes the following parameters (i) the compression parameter $A$; (ii) the parameters of GGHsMM; (iii) the confidence level $\alpha$ for long-term behavior diagnosis; (iv) a threshold of BI for short-term behavior diagnosis; (v) the $SF$ for soft-control. Since all these parameters can be obtained by the model training without too much manual intervention, the entire scheme has the advantages of self adaptability and can be applied to different scenarios.

Moreover, experiment results have shown that the proposed scheme is stable. For examples, Fig.4(e) and Fig.12 show that the observed features do not vary with the number of users and the constantly updated Web contents. Fig.11 implies that the normal underlying state process is a stable stochastic process, since the distribution of each state is very compact and stable. Fig.9 shows that the proposed scheme is almost traffic volume independent. The main reason for these phenomena is that real users can automatically follow the Web site’s updates. However, attack routines do not have this ability. Hence the polluted traffic will deviate from the normal behavior model when attack appears, which enables our scheme.

The computational complexity mainly comes from the GGHsMM. Since the model training can be implemented in off-line style, its computational complexity does not affect the real-time applications. In the detection phase, we only need the forward process to compute the BI. Based on a computer that is configured with Intel Core 2 Duo CPU 2.8GHz, 4G-RAM and 64 bits operating system, the processing rate...
of the detection approximates to 7250 reqs/s which can meet the needs of most of current Web servers.

8 CONCLUSION

In this paper, we tried to filter the attack traffic from the aggregated proxy-to-server traffic, which is a new problem for the DDoS detection. A novel resisting scheme was proposed based on TSL. GGHsMM, multi-precision diagnostic method and soft-control were proposed to improve the detection performance. Experiments confirmed the effectiveness and robustness of the proposed scheme. The main advantages of our approach shown in the experiments include: (i) its detection performance is better than the pure statistical methods; (ii) it is independent of the traffic intensity and the frequently-varying Web contents; (iii) it can realize the early detection. Moreover, the DR, FPR and the computational overhead can meet the needs of most of practical applications.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The authors also thank Network and Information Technology Center of Sun Yat-sen university for assisting them to collect the real experiment data reported here. Xie’s work was partially supported by the National Natural Science Foundation of China (Grant No. 60971046, U0735002 and 61173018); Doctoral Fund of Ministry of Education of China (Grant No.20090171120001); the Fundamental Research Funds for the Central Universities (Grant No.11lgpy38). Hu’s work was partially supported by ARC LP110100602, ARC LP100100404, ARCP100200538, and UNSW@ADFA Defence-Related Research Fund 2012.

REFERENCES

Yi Xie received a B.S. degree, M.S. degree and PhD. degree from Sun Yat-Sen University, Guangzhou, China. From July 1996 to July 2004, he worked at Civil Aviation Administration of China. He was a visiting scholar at George Mason University during 2007 and 2008. He is currently a lecturer at the School of Information Science and Technology in Sun Yat-Sen University, China. His research interests focus on computer networks, network security and network behavior model

S. Tang (SM07) received his B.S. degree from Tianjin University, China, and M.S. degree from China Academy of Telecommunications Technology, China. He received his PhD from University of Toledo, Ohio. He has over 8 years of industrial R&D experience in the field of information technology. He was a research fellow at George Mason University from 2006 to 2009. Currently, he is an assistant professor of electrical engineering at Missouri Western State University, USA. His research interests focus on network security, wireless networking, modeling and performance evaluation.

Y. Xiang received his PhD in Computer Science from Deakin University, Australia. He is currently with School of Information Technology, Deakin University. His research interests include network and system security, distributed systems, and networking. In particular, he is currently leading in a research group developing active defense systems against large-scale distributed network attacks. He is the Chief Investigator of several projects in network and system security, funded by the Australian Research Council (ARC). He has published more than 100 research papers in many international journals and conferences, such as IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Information Security and Forensics, and IEEE Journal on Selected Areas in Communications. He has served as the Program/General Chair for many international conferences such as ICA3PP 12/11, IEEE/IFIP EUC 11, IEEE TrustCom 11, IEEE HPCC 10/09, IEEE ICPADS 08, NSS 11/10/09/08/07. He has been the PC member for more than 50 international conferences in distributed systems, networking, and security. He serves as the Associate Editor of IEEE Transactions on Parallel and Distributed Systems and the Editor of Journal of Network and Computer Applications. He is a member of the IEEE.

J. Hu is a full professor of Cyber Security at the School of Engineering and Information Technology, the University of New South Wales at the Australian Defence Force Academy (UNSW@ADFA), Australia. He is also the Research Director of Cyber Security Centre at UNSW@ADFA. Prof. Hu is a panel member of Mathematics, Information and Computing Sciences (M10) in ERA (Excellence in Research for Australia) 2012 Research Evaluation Committee. Prof. Hu received his Bachelor’s degree in Industrial Automation in 1983 from Hunan University, PR China; a PhD degree in engineering in 1993 from the Harbin Institute of Technology, PR China; and a Master’s degree by research in the School of Computer Science and Software Engineering from Monash University, Australia, in 2000. In 1995 he completed his postdoctoral fellow work in the Department of Electrical and Electronic Engineering, Harbin Shipbuilding College, PR China. He was a research fellow of the Alexander von Humboldt Foundation in the Department of Electrical and Electronic Engineering, Ruhr University, Germany, during 1995-1997. He worked as a research fellow at the EE Department, Delft University of Technology the Netherlands in 1997 and research fellow at the EE Dept. of the University of Melbourne Australia in 1999. Prof. Hu’s major research interest is in computer security, especially biometric security. He has been awarded seven Australia Research Council (ARC) Grants. He served as Security Symposium Co-Chair for IEEE GLOBECOM ’08 and IEEE ICC ’09.