Intrusion Tolerance of Stealth DoS Attacks to Web Services

Massimo Ficco, Massimiliano Rak

Intrusion Tolerance of Stealth DoS Attacks to Web Services

Massimo Ficco and Massimiliano Rak

Department of Information Engineering, Second University of Naples (SUN)
{massimo.ficco,massimiliano.rak}@unina2.it

Abstract. This paper focuses on one of the most harmful categories of Denial of Service attacks, commonly known in the literature as "stealth" attacks. They are performed avoiding to send significant volumes of data, by injecting into the network a low-rate flow of packets in order to evade rate-controlling detection mechanisms. This work presents an intrusion tolerance solution, which aims at providing minimal level of services, even when the system has been partially compromised by such attacks. It describes all protection phases, from monitoring to diagnosis and recovery. Preliminary experimental results show that the proposed approach results in a better performance of Intrusion Prevention Systems, in terms of reducing service unavailability during stealth attacks.

Keywords: stealth attacks; intrusion tolerance; web services.

1 Introduction

Denial of Service (DoS) attacks are serious threats to the Internet causing billions of dollars in economic loss. In particular, brute force and flooding attacks against application-layer services, like the Web Services (WSs), pose a huge risk to several business-critical services. The recent tide of DoS attacks against high-profile WSs, including PayPal, MasterCard and Amazon, demonstrate how devastating DoS attacks are. In general, attackers are aware of the presence of protection mechanisms: they thus attempt to perform their activities in a stealthy fashion in order to elude local security mechanisms. From an attacker point of view, one of the most effective way of circumventing these security countermeasures consists in distributing the attacks both in 'form' and/or 'time' executing. Thus, a particular categories of DoS attacks are low-rate DoS attacks, commonly known in the literature as "stealth" attacks [1]. Unlike high-rate attacks, they minimize the visibility of the attack, at the same time they can be as harmful as common brute force attacks [2]. The attack is carried out by directing a flow of packets to a particular system at such a low-rate that would evade DoS detection mechanisms protecting it. Finer detection and prevention mechanisms have to be defined to to reduce the intrusion on the target system [3].
In this paper, we focus on low-rate DoS attacks to WSs that exploit application-level vulnerabilities. In general, network defenses, including firewalls and network Intrusion Detection/Prevention Systems (IDS/IPS) are useless against such attacks to software systems, since they do not analyzes packet contents. Such

attacks are based on application-level exploits, which in most cases are indistinguishable from normal use. We propose an intrusion tolerant approach that aims at mitigating application-level low-rate DoS attacks from generating a service unavailability. The approach consists of monitoring anomalous resources consumption of the target machine and correlating this information with some attack symptom. If an attack is detected, *i.e.*, both the system resources consumption exceeds a critical level and some malicious requests to the WS are observed, an "adaptive" filtering is applied. It consists of filtering application requests on the base of a threshold that is dynamically adapted during the attack. Our experiments consist to inject low-rate of malicious requests that exhaust the computational resources of the host system. In particular, we focus on an example of stealth DoS attacks that exploit XML vulnerabilities [4]. Our preliminary results shown that the proposed approach results in a better performance of the IPSs that adopt "static" threshold-based prevention mechanisms, in terms of reducing the service unavailability during a stealth attack.

2 Building Stealth Attacks

Denial-of-Service attacks aim at reducing services availability by exhausting the resources of the services host system, like memory, processing resources and network bandwidth. Some classic way to perform such attacks to WSs, consists of: (*i*) querying a service using a very large request message, in order to exploits the high memory consumption of the request processing; (*ii*) forcing to decrypt a large number of encrypted messages in order to lead to high CPU load; (*iii*) creating a new process instance each time a message arrives; and (*iv*) formulating well-formed messages that require complex processing on the target system [5]. Stealth attacks have been defined in [7] as those aiming at keeping the attackers virtually invisible to network-based defenses. These attacks can be significantly harder to detect compared with more traditional brute-force and flooding style attacks [2]. It is an interesting open issue to detect and react to such attacks that exhibit "variable" and "polymorphic" behaviors [12]. It is hard to specify quantitative time constraints in stealth attacks. Attackers can perform polymorphic attacks that change their time constraints.
Therefore, since such kind of dynamic changes are very hard to predict, intrusion tolerant mechanisms that use "adaptive", instead of "static", thresholds could overcome such limitations (*i.e.*, thresholds that do not depend from an initial training phase). Obviously, the kind of detection and reaction mechanisms to be used cannot be absolute, but must be defined with respect to an intrusion model, that is, the specification of effects that the successful attack has on the target system (*e.g.*, the degree of success of the intruder in terms of resource consumption). Therefore, in order to define an adaptive mechanism to detect and react to a low-rate attack, it is necessary to identify good attack examples and analyze/model their effects of the target system. We considered flooding attack that exploits the XML vulnerability, which is one of main factor that makes WSs vulnerable to DoS attacks. In particular, we consider the *Coercive Parsing* attack, also called *Deeply-Nested XML DoS (XDoS)*. It is a resource exhaustion attack, which exploits the XML message format by inserting of a large number of nested XML tags in the message body. The goal is to force the XML parser

within the server application, to exhaust the computational resources of the host system by processing numerous deeply-nested tags. In a previous work [8], we analyze the CPU consumption depending on the number of nested XML tags and the frequency with which the malicious messages are injected. The experiment shows that a message of 500 nested tags is sufficient to produce a peak of CPU load of about 97%, whereas with 1000 tags the CPU is fully committed to process the malicious message for about 3 seconds. A flooding attack example that exhaust fully the system's CPU resource, consists to inject malicious sequences of XML messages with 150 nested tags every 200 ms.
In this work, in order to identify good candidate stealth attacks, we analyze the CPU consumption in presence of sustained XDoS attack. We perform different attack scenarios. Each scenario takes about 2 hours and consists of a sequence of messages injected with a fixed frequency and a fixed number of nested tags. The experiments show that it is sufficient to inject messages with about 5 nested XML tags every 5 ms, to make unavailable the target machine (*i.e.*, to exhaust the CPU resource). On the base of such analysis, we define a distributed low-rate XDoS attack. It consists in distributing the attacks both in 'form' and 'time' executing, *i.e.*, injecting messages with a different number of nested XML tags, in order to circumvent fixed thresholds (attack distribution in form), and delays single messages so as to bypass time window and rate controls (attack distribution in time). Such an attack flow keeps the computational capacity of target host constantly busy, hence affecting the WS availability. The sequence of injected messages is generated through an algorithm that randomly varies the frequency and the number of nested tags in order to ensure a constant CPU overloading.

3 The Intrusion Tolerant Approach

In order to detect low-rate DoS attacks, which exhaust the computational recourses of the target machine for extended periods, an active resource monitoring approach is adopted. It consists in observing anomalous resource usage load (monitoring), and analyzing the possible causes of such resource usage overloading, in order to decide if such anomalous behavior is due to either an attack or a normal operation (diagnosis). When a stealth DoS attack is detected, a threshold-based filtering reaction is triggered on the base of the target resource usage load (recovery).
The CPU monitoring mechanism has to deal with application scenarios highly variable, which alternate periods of heavy workloads, which involve high number of software components and heterogeneous type of service requests, with periods of low computational activities. We assume that during 'normal' operation, the monitored CPU behavior can be modeled as a random walk, which alternates stable periods, during which the CPU load has some 'stable' behavior (*i.e.*, it is within a specific range), with transient periods (smaller compared with the stable), during which significant variation of CPU consumption occurs. During the transient period, changes in the monitored behavior consists in continuous increments or decrements with respect to the 'stable' behavior due to a workload variation (*e.g.*, due to a burst of requests). Thus, a count-and-threshold over-time monitoring mechanism using heuristic approaches is adopted to de-

tect anomalous CPU consumption. It consists to monitor extended excessive CPU consumption and detect when a threshold is reached. The diagnosis activity consists to verify the presence of some stealth attack symptom in the application requests. It combines several information collected by using different anomaly-based detection models that estimate the anomaly degree of monitored features, including: (*i*) the type of sequence of XML nested tags included in the message, (*ii*) the actual distribution of nested XML tags in the message sequence, and (*iii*) the number of nested XML tags of each message. Monitored symptoms are correlated (by means of a simple weighted sum) and rearranged based on a confidence level, which indicates the likelihood that they are symptomatic of an ongoing attack on the system. Finally, if the confidence exceeds a fixed threshold, an recovery action is performed [9, 10]. The objective of the reaction is to reduce the CPU load on the target system, in order to reduce the period in which the service is unavailable. In particular, it filters each XML message that contains a number of nested tags greater than a given threshold T_A. In order to face stealth attacks, we have to adapt the threshold, so that it cuts even low-rate attack messages (*i.e.*, messages with a low number of nested tags). The 'adaptive' threshold T_A is decreased until the CPU consumption falls below a severity level. The drawback of such an approach is the presence of false positive results (*i.e.*, the filter cuts valid messages). We adopt the following exponential function to decrease the filtering threshold T_A, which is equal to $B\lambda e^{-\lambda k}$ when $k > 0$, and equal to 0 when $K < 0$. k is a discrete time variable, and B is the maximum number of nesting tags that is initially admitted.

4 Experiments and Results

In this section, we present preliminary results achieved by applying the proposed intrusion tolerant approach in our experimental testbed. The experiment consists of the following steps: (1) launch attacks from attacker machine, while maintaining a simulated stress load on the Web server; (2) trigger an alert only when an intrusion is detected (*i.e.*, whether both an excessive CPU consumption is observed and the attack symptoms are diagnosed); (3) perform the intrusion reaction. In order to assess the effectiveness of the proposed solution, TPC Benchmark W (TPC-W) is adopted [11]. It is a transactional Web benchmark. The benchmark simulates the activities of a business oriented transactional Web server. The performance metric reported by TPC-W is the number of Web interactions processed per second (WIPS), which is used to evaluate the reaction mechanism. During the experimental campaign, in order to evaluate the proposed solution, a workload based on real traffic is adopted. It is collected from production Web server at the university, in which the considered application runs, during an interval time of 12 hours. It consists of about 768.154 messages containing not more than 17 nested XML tags. Using results described in Section 2, during the detection phase, we performed several experiments, in which we injected low-rate of malicious SOAP messages. It is characterized by 215.531 messages with a number of nested XML tags including within the range [5-4500]. Finally, we overlapped the stealth XDoS attack traffic on top of the collected workload and TPC-W background traffic.

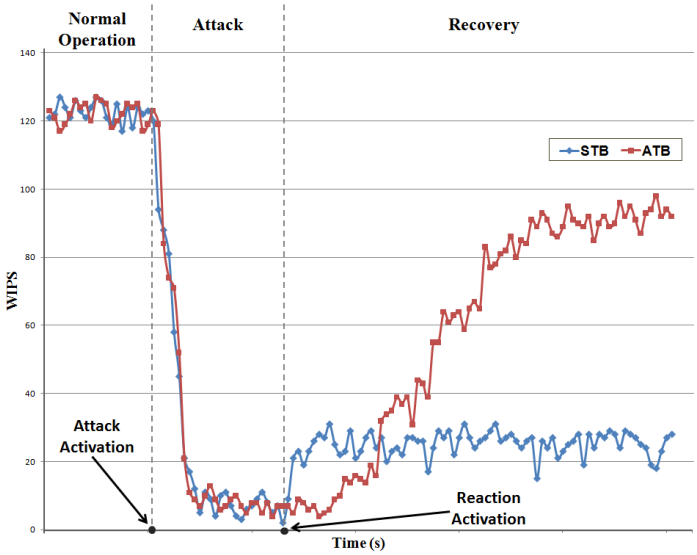

Fig. 1. WIPS evaluation during a recovery process to a stealth XDoS attack

In Figure 1 compares the results archived by using the proposed *adaptive-threshold* based recovery approach (ATB), with an intrusion prevention approach that adopts a *static-threshold* (STB). The STB approach exploits a worst-case threshold T_S, which is estimated during a training phase. In particular, the threshold T_S id fixed to the greater number of nested tags contained in the used workload. Thus, all the received messages that contains more than $T_S = 17$ nested tags, are preventively filtered. Figure 1 represents the WIPS variations with respect to the time, during the three temporal windows. During the first two windows the recovery mechanism is disabled. In particular, the first window shows the values of the WIPS in absence of the attack. The second one shows the intrusion effects. As Figure 1 shows, during the attack, the number of TPC-W interactions processed is very low, about 7% respect to the normal operation. Finally, during the last window, the recovery mechanism is enabled. Experimental results confirm the limitations of the STB approach with polymorphic behavior of stealth attacks. Although, the reaction latency of STB approach is shorter than ATB approach (*i.e.*, the system filters immediately all messages with more that T_S nested tags), the number of TPC-W interactions processed increases by only about 12% respect to the system under the attack. Such approach allows to filter only a part of malicious messages. Moreover, the attacker could use this information to reduce progressively the number of nested tags in order to fully evade the fixed threshold. As Figure 1 shows, by using the ATB approach, the number of TPC-W interactions processed increases by about 69% respect to the system under the attack. On the other hand, the excessive reduction of the threshold can produce 'false positives', *i.e.*, correct messages are filtered. Results show that, in order to reduce the CPU load under the 85%, about 0,009% of correct messages are filtered.

5 Conclusion and Future Work

The proposed approach emphasizes the relation among intrusion monitoring, diagnosis and recovery. It consists to monitor anomalous behaviors of resources consumption on the target system, and then, verify/diagnosis if they are caused by an low-rate attack. The experiment results show that the proposed approach is able to cope with the polymorphism of such attacks as well as improve the availability/performance of the WS during the intrusion. The objectives of our future work will be define a more robust model to correlate the symptoms of the attack with its intrusion effects, and extend the proposed approach to a larger set of stealth DoS attacks.

6 Acknowledgment

This research is partially supported by FP7-ICT-2009-5-256910 (mOSAIC) project and the MIUR-PRIN 2008 project "Cloud@Home".

References

1. A. Kuzmanovic. Low-rate tcp-targeted denial of service attacks and counter strategies. In *IEEE/ACM Trans. on Networking*, vol. 14, no. 4, 2006, pp. 683-696.
2. Y. Zhang, Z. M. Mao, and J. Wang. Low-Rate TCP-Targeted DoS Attack Disrupts Internet Routing. In Proc. of the *14th Network and Distributed System Security Symposium (NDSS'07)*, Feb. 2007.
3. N. Boggs, S. Hiremagalore, A. Stavrou, and S.J. Stolfo. Experimental Results of Cross-Site Exchange of Web Content Anomaly Detector Alerts. In Proc. of the *IEEE Int. Conf. on Technologies for Homeland Security*, Nov. 2010, pp. 8-14.
4. M. Jensen, N. Gruschka, and R. Herkenh. A survey of attacks on web services. In *Computer Science*, vol. 24, no. 4, 2009, pp. 185-197.
5. M. Jensen, N. Gruschka, R. Herkenh, and N. Luttenberger. SOA and Web Services: New Technologies, New Standards - New Attacks. In Proc. of *the Fifth European Conference on Web Services*, 2007, pp. 35-44, IEEE CS.
6. A. Kuzmanovic and E. W. Knightly. Low-Rate TCP Targeted Denial of Service Attacks: the shrew vs. the mice and elephants. In Proc. of the *International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2003. ACM Press.
7. S. Antonatos, M. Locasto, S. Sidiroglou, A. D. Keromytis, and E. Markatos. Defending against next generation through network/endpoint collaboration and interaction. In Proc. of the *3rd International Conference on Computer Network Defense*, LNCS, vol. 30, 2008, pp. 131-141. Springer US.
8. M. Ficco and M. Rak. Intrusion Tolerant Approach for Denial of Service Attacks to Web Services. In Proc. of the *1st International Conference on Data Compression, Communications and Processing (CCP'11)*, Jun. 2011. IEEE CS Press.
9. M. Ficco. Achieving Security by Intrusion-Tolerance Based on Event Correlation. In *Journal of Network Protocols and Algorithms (NPA)*, vol. 2, no. 3, 2010, pp. 70-84.
10. M. Ficco and L. Romano. A Correlation Approach to Intrusion Detection. In *Mobile Lightweight Wireless Systems*, vol. 45, 2010, pp. 203-215. Springer LNICST.
11. TPC Benchmark W (TPC-W), Available at: at http://www.tpc.org/tpcw/
12. Zhichun Li, Lanjia Wangt, Yan Chen and Zhi Fut. Network-based and Attack-resilient Length Signature Generation for Zero-day Polymorphic Worms. In Proc. of the *IEEE Int. Conf. on Network Protocol*, Oct. 2007, pp. 164-173. IEEE CS Press.