ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

DETECTION AND CONTROL SCHEME ON BOTNET ATTACKS UNDER CLOUD SERVICES

Mr. P. Selvakumar, IInd ME., (CSE)
Mr. S. Thiruvenkateasamy, Asstistant Professor/CSE
Nandha College of Technology, Erode, Tamilnadu, India
selvakumardave@gmail.com

Abstract

Cloud resources are provided to the consumers to access their applications. Service requests are received from various cloud users and responses are redirected to the cloud users. Distributed Denial of Service (DDoS) attacks are raised by the botnet members. Botnet is constructed to attack a service provider. Intrusion detection schemes are designed to detect the normal and attackers. Attack pattern discovery is performed under the application environment. Slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS) are initiated to raise attacks on cloud service providers. Cloud service performances are reduced by the SIPDAS. Signature identification is affected with polymorphic behavior of the users. Dynamic behavioral changes increases the load I the cloud servers.

Detection and Controlling scheme is constructed to discover the Slowly-Increasing-Polymorphic DDoS Attack Strategy (SIPDAS) based attacks. Cost factor analysis operations are attached with the system. Cloud consumer behavior changes are identified with the support of the flow analysis mechanism. Application server operations are enhanced to discover the attacks raised by the cloud users.

1. Introduction

Cloud computing aims to provide convenient, on-demand, network access to a shared pool of configurable computing resources can be rapidly provisioned and released with minimal management effort or service provider interactions [1]. Cloud provides services in various forms: Software as a Service-SaaS, Platform as a Service-PaaS and Infrastructure as Service-IaaS. As Cloud services are provisioned through the Internet; security and privacy of Cloud services are key issues to be looked upon. International Data Corporation (IDC) survey [8] showed that security is the greatest challenge of Cloud computing. The recent cloud computing security white paper by Lockheed Martin Cyber Security division [9] shows that the major security concern after data security is intrusion detection and prevention in cloud infrastructures. Cloud infrastructure makes use of virtualization techniques, integrated technologies and runs through standard Internet protocols. These may attract intruders due to much vulnerability involved in it.

Cloud computing also suffers from various traditional attacks such as IP spoofing, Address Resolution Protocol spoofing, Routing information Protocol attack, DNS poisoning, Flooding, Denial of Service (DoS), Distributed Denial of Service (DDoS) etc. E.g. DoS attack on the underlying Amazon Cloud infrastructure caused BitBucket.org, a site hosted on AWS to remain unavailable for few hours [10]. In [2], the computing-cost using current cryptographic techniques cannot be overlooked for Cloud. Firewall can be a good option to prevent outside attacks but does not work for insider attacks. Efficient intrusion detection systems (IDS) and intrusion prevention systems (IPS) should be incorporated in Cloud infrastructure to mitigate these attacks.

2. Related work

In this section we briefly cover some of the attacks that are currently used within cloud computing. We also cover previous research on SOTA, which is based on service-oriented architecture and service-oriented grid architecture. To conclude this section, we briefly cover the research done on X-DoS which is a DDoS attack that could affect cloud computing.

2.1. Cloud Computing Attacks

In the current research on cloud computing [7] most think of cloud computing as the virtualization of on-demand, elastic, scalable, resource that is service. But as Balding pointed out in his Rivest, Shamir and Adelman (RSA) conference presentation, cloud computing is actually much more and that it really is the abstraction of services [6].

Since cloud computing security follows the idea of cloud computing, there are two main areas that security experts look at securing in a cloud system: These are VM vulnerabilities and message integrity between cloud systems. Some of the attacks that encompass both are: Rafal’s Heap Overflow in I/O, Rafal’s against Xen, Rafal’s against Microsoft Virtual Server and Greg McManus Shared Folders vulnerability in VMware.

2.2. Service-Oriented Trace Back Architecture

SOTA is a web security service application that is product neutral [3],[4],[5]. Its main objective is to apply a SOA approach to trace back methodology. This is in order to identify a forged message identity, since one of the main objectives of X-DoS and DX-DoS is to hide the attacker’s true identity. The basis of SOTA is founded upon the Deterministic Packet Marking (DPM) algorithm. DPM marks the ID field and reserved flag within the IP header.

As each incoming packet enters the edge ingress router it is marked. The marked packets will remain unchanged as they traverse the network. Outgoing packets are ignored. DPM methodology is applied to our SOTA framework, by placing the Service-Oriented Traceback Mark (SOTM) within web service messages. If any other web security services are already employed, SOTM would replace the ‘token’ that contains the client identification. Real source message identification is stored within SOTM and placed inside the SOAP message. SOTM, as in DPM tag, will not change as it traverses through the network. The composition of SOTM is made up of one XML tag, so not to weigh down the message, It is then stored within a SOAP header. Upon discovery of an X-DoS or DX-DoS attack, SOTM can be used to identify the true source of forged messages. SOTA does not directly eliminate an X-DoS or DX-DoS attack message. This is left for the filter section of a defense system called Cloud Protector. This leaves SOTA with the important task of dealing with the main objectives of X-DoS and DX-DoS, which are:

- Exploit a known vulnerability or to flood the system with useless messages to exhaust the web server’s resources to the point of collapse. These vulnerabilities could be found in communication channels or known exploits within the services provided.

- Attackers who try to hide their identities. The reasons for this vary, depending on the type of attack, but usually it is to cover their crime or to bypass a known defense that is in place to prevent it. It is with this second objective that SOTA attempts to cover, as other trace back methods do, items like Probability Packet Marking (PPM) and DPM.

There are a number of reasons why cloud computing should employ a SOTA type framework:

- Current web security is not up to handling an X-DoS or DX-DoS attack. In fact, how WS-Security can be used in an X-DoS attack.
- With IPv6 coming into use, current IP traceback methods will no longer be viable. This is due to the changes IPv6 introduces IPSec and the packet header format which no longer holds support for the fields that are required for IP traceback.
- SOTA does not violate IP protocols due to storing information in the IP packet.
- Using the SOA model, SOTA can be employed on any ubiquitous grid system.

2.3. XML-based Denial Of Service (X-DoS) Attacks

A Denial of Service (DoS) is where an attacker attempts to deprive legitimate users of their resources [11]. An X-DoS attack, according is where a network is flooded with XML messages instead of packets in order to prevent legitimate users to access network communications. Further, if the attacker floods the web server with XML requests, it will affect the availability of these web services. Attackers can also manipulate the message content, in order to cause the web server to crash. To adapt X-DoS into a Distributed Denial of Service paradigm, called Distributed XML based Denial of Service (DX-DoS), the attacker uses multiple hosts to attack the victim with X-DoS attacks. Though none of these attacks have been reported as yet, this type of attacks could be a very serious threat facing cloud computing in the future.

3. Intrusion Detection Systems

Intrusion detection systems are the ‘burglar alarms’ of the computer security field. The aim is to defend a system by using a combination of an alarm that sounds whenever the site’s security has been compromised and an entity most often a site security officer (SSO) that can respond to the alarm and take the appropriate action, for instance by ousting the intruder, calling on the proper external authorities and so on. This method should be contrasted with those that aim to strengthen the perimeter surrounding the computer system. Both methods should be used, along with others, to increase the chances of mounting a successful defense, relying on the age-old principle of defense in depth. It should be noted that the intrusion can be one of a number of different types. For example, a user might steal a password and hence the means by which to prove his identity to the computer. The user is called masquerader and the detection of such intruders is an important problem for the field. Other important classes of intruders are people who are legitimate users of the system but who abuse their privileges and people who use pre-packed exploit scripts, often found on the Internet, to attack the system through a network.

Early in the research into such systems two major principles known as anomaly detection and signature detection were arrived at, the former relying on flagging all behaviour that is abnormal for an entity, the latter flagging behavior that is close to some previously defined pattern signature of a known intrusion. The problems with the first approach rest in the fact that it does not necessarily detect undesirable behavior and that the false alarm rates can be high. The problems with the latter approach include its reliance on a well defined security policy, which may be absent and its inability to detect intrusions that have not yet been made known to the intrusion detection system. It should be noted that to try to bring more stringency to these terms, user use it in a slightly different fashion than previous researchers in the field. An intrusion detection system consists of an audit data collection agent that collects information about the system being observed. This data is then either stored or processed directly by the detector proper, the output of which is presented to the SSO, then take further action, normally beginning with further investigation into the causes of the alarm.

4. Botnet Attacks In Clouds

Over the past decade, many efforts have been devoted to the detection of Botnet attacks in distributed systems. Security prevention mechanisms usually use approaches based on rate-controlling, time-window, worst-case threshold and pattern-matching methods to discriminate between the nominal system operation and malicious behaviors. The attackers are aware of the presence of such protection mechanisms. They attempt to perform their activities in a “stealthy” fashion in order to elude the security mechanisms, by orchestrating and timing attack patterns that leverage specific weaknesses of target systems. They are carried out by directing flows of legitimate service requests against a specific system at such a low-rate that would evade the DDoS detection mechanisms and prolong the attack latency, i.e., the amount of time that the ongoing attack to the system has been undetected.

A sophisticated strategy is applied to orchestrate stealthy attack patterns against applications running in the cloud. Instead of aiming at making the service unavailable, the proposed strategy aims at exploiting the cloud flexibility, forcing the application to consume more resources than needed, affecting the cloud customer more on financial aspects than on the service availability. The attack pattern is orchestrated in order to evade, greatly delay the techniques proposed in the literature to detect low-rate attacks. It does not exhibit a periodic waveform typical of low-rate exhausting attacks. In contrast with them, it is an iterative and incremental process. In particular, the attack potency is slowly enhanced by a patient attacker, in order to inflict significant financial losses, even if the attack pattern is performed in accordance to the maximum job size and arrival rate of the service requests allowed in the system. Using a simplified model empirically designed, we derive an expression for gradually increasing the potency of the attack, as a function of the reached service degradation. We show that the features offered by the cloud provider, to ensure the SLA negotiated with the customer can be maliciously exploited by the proposed. Stealthy attack, which slowly exhausts the resources provided by the cloud provider and increases the costs incurred by the customer.

The attack strategy, namely Slowly-Increasing-Polymorphic DDoS Attack Strategy (SIPDAS) can be applied to several kind of attacks, that leverage known application vulnerabilities, in order to degrade the service provided by the target application server running in the cloud. The term polymorphic is inspired to polymorphic attacks which change message sequence at every successive infection in order to evade signature detection mechanisms. Even if the victim detects the SIPDAS attack, the attack strategy can be reinitiate by using a different application vulnerability, or a different timing. Brute-force attacks are raised against through specific periodic, pulsing and low-rate traffic patterns. Rate-controlling, time-window, worst-case threshold and pattern-matching are adapted to discriminate the legitimate and attacker activities. Stealthy attack patterns are raised against applications running in the cloud. Slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS) can be applied to initiate application vulnerabilities. SIPDAS degrades the service provided by the target application server running in the cloud. Polymorphic attacks changes the message sequence at every successive infection to avoid signature detection process. Slowly-increasing polymorphic behavior induces enough overloads on the target system. XML-based DoS (XDoS) attacks to the web-based systems are applied as the testing environment for the attack detection process. The following problems are identified from the existing system. SIPDAS based attack detection is not supported. Polymorphic behavior identification is not adapted. Application level vulnerability detection is low. Service degradation and resource consumption cost analysis is not performed.

5. Detection and Control Scheme on Botnet Attacks

In this section are presented several attack examples, which can be leveraged to implement the proposed SIPDAS attack pattern against a cloud application. In particular, we consider DDoS attacks that exploit application vulnerabilities, including: the Oversize Payload attack that exploits the high memory consumption of XML processing; the Oversized Cryptography that exploits the flexible usability of the security elements defined by the WS-Security specification the Resource Exhaustion attacks use flows of messages that are correct regarding their message structure, but that are not properly correlated to any existing process instance on the target server and attacks that exploit the worst-case performance of the system, for example by achieving the worst case complexity of Hash table data structure, or by using complex queries that force to spend much CPU time or disk access time. In this paper, we use a Coercive Parsing attack as a case study, which represents one of the most serious threat for the cloud applications. It exploits the XML verbosity and the complex parsing process. In particular, the Deeply-Nested XML is a resource exhaustion attack, which exploits the XML message format by inserting a large number of nested XML tags in the message body. The goal is to force the XML parser within the application server, to exhaust the computational resources by processing a large number of deeply-nested XML tags.

The system is aimed to defining the objectives that a sophisticated attacker would like to achieve and the requirements the attack pattern has to satisfy to be stealth. Recall that, the purpose of the attack against cloud applications is not to necessarily deny the service, but rather to inflict significant degradation in some aspect of the service, namely attack profit PA, in order to maximize the cloud resource consumption CA to process malicious requests. In order to elude the attack detection, different attacks that use low-rate traffic have been presented in the literature. Several works have proposed techniques to detect low-rate DDoS attacks, which monitor anomalies in the fluctuation of the incoming traffic through either a time or frequency-domain analysis. They assume that, the main anomaly can be incurred during a low-rate attack is that, the incoming service requests fluctuate in a more extreme manner during an attack. The abnormal fluctuation is a combined result of two different kinds of behaviors: (i) a periodic and impulse trend in the attack pattern and (ii) the fast decline in the incoming traffic volume. In order to perform the attack in stealthy fashion with respect to the proposed detection techniques, an attacker has to inject low-rate message flows $\varphi_{Aj} = [\varphi_{j,1}, \; . \; . \; . \; , \varphi_{j,m}]$, that satisfy the following optimization problem:

In order to implement SIPDAS-based attacks, the following components are involved:

- a Master that coordinates the attack;
- π Agents that perform the attack; and
- a Meter that evaluates the attack effects.

The approach implemented by each Agent to perform a stealthy service degradation in the cloud computing. It has been specialized for an X-DoS attack. Specifically, the attack is performed by injecting polymorphic bursts of length T with an increasing intensity until the attack is either successful or detected. Each burst is formatted in such a way as to inflict a certain average level of load C_R. In particular, we assume that C_R is proportional to the attack intensity of the flow Φ_{Aj} during the period T.

Denote I_0 as the initial intensity of the attack and assuming $\Delta C_R = \Delta I$ as the increment of the attack intensity. For each attack period, fixed the maximum number of nested tags (tagThreshold), the routine pickRandomTags(. . .) randomly returns the number of nested tags nT for each message. Based on nT , the routine compute Inter arrival Time uses a specific algorithm to compute the inter-arrival time for injecting the next message. At the end of the period T, if the condition ‘attack Successful’ is false, the attack intensity is increased. If the condition ‘attack Successful’ is true, the attack intensity is maintained constant until either the attack is detected or the auto-scaling mechanism enabled in the cloud adds new cloud resources. The attack is performed until it is either detected, or the average message rate of the next burst to be injected is greater than dT. In this last case, the Agent notifies to the Master that the maximum average message rate is reached and continues to inject messages formatted according to the last level of load CR reached.

6. Conclusion

Cloud resources are shared with mutual and commercial models. Slowly-Increasing-Polymorphic DDoS Attack Strategy (SIPDAS) is adapted to initiate DDoS attacks on the clouds. Cloud Intrusion Detection System (CIDS) is constructed to discover the SIPDAS attacks with flow correlation analysis. Polymorphic behavior identification and cost analysis methods are integrated with the CIDS. Cloud Intrusion Detection System (CIDS) is build to discover slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS). The CIDS controls the resource consumption and cost factors. The system minimizes the application level vulnerabilities. Attack behavioral changes are automatically detected by the system.

REFERENCES

[1] P. Mell and T. Grance. (2011). The NIST Definition of Cloud Computing (Draft). *NIST* [Online]. Available: http://csrc.nist.gov/publications/drafts/800 145/Draft-SP-800- 145_cloud-definition.pdf

[2] Y. Chen and R. Sion, “On securing untrusted clouds with cryptography,” *In WPES ’10*, pp. 109–114, 2010.

[3] Chonka A, Zhou W, Xiang Y. Protecting web services with service oriented traceback architecture. In: Proceedings of the IEEE eighth international conference on computer and information technology, IEEE, 2008.

[4] Chonka A, Zhou W, Xiang Y. Protecting web services from DDoS attacks by SOTA. In: Proceedings of the IEEE fifth international conference on information technology and applications, IEEE, 2008b.

[5] Chonka A, Zhou W, Xiang Y. Defending grid web services from X-DoS Attacks by SOTA. In: Proceedings of the third IEEE international workshop on web and pervasive security (WPS 2009), IEEE, 2009.

[6] Balding G. What everyone ought to know about cloud security. Cloudsecurity.com, /http://www.slideshare.net/craigbalding/what-everyone-ought to-know-a bout-cloud-securityS, 2009.

[7] Laplante P, Zhang J, Voas J. What's in a name? Distinguishing between saas and soa IT Professional 2008;10(3):46–50.

[8] International Data Corporation. 2009. [Online]. Available: http://blogs.idc.com/ie/wpcontent/uploads/2009/12/idc_cloud_challenge s_2009.jpg, 2009.

[9] Lockheed Martin White Paper: Available: http://www.lockheedmartin.com/data/assets/isgs/documents/CloudComp utingWhitePaper.pdf

[10] C. Brooks. Amazon EC2 Attack Prompts Customer Support Changes. *Tech Target*. [Online]. Available: http://searchcloudcomputing.techtarget.com/news/article/0,289142,sid201_gci1371090,00.html

[11] Rogers L. What is a Distributed Denial of Service (DDoS) attack and what can I do about it? Computer Emergency Response Team, /http://www.cert.org/ homeusers/ddos.htmlS, last accessed 19 November 2009.