Cloud Computing Vulnerabilities to Denial of Service Attacks

ALEXANDER POWELL
ajpowell@email.wm.edu
College of William & Mary
Computer and Network Security
CSCI 454
Prof. Kun Sun
May 10, 2016

Abstract

Cloud computing provides an efficient and effective way to increase computing capabilities without the need to invest in new infrastructure. It can also be an extremely economical decision for businesses and their IT needs. In the past decade alone, cloud computing has become one of the fastest growing segments of the IT industry. However, there are still a lot of unanswered questions about how secure the system is and how big a threat these issues pose. One of the largest threats to the security of cloud computing comes in the form of Denial of Service (DoS) attacks. The purpose of this paper is to survey the major DoS threats to the system and look at possible solutions to these attacks.

Keywords: cloud-computing, software, security, deployment, business

Introduction

Cloud computing, otherwise known as on-demand computing is a type of computing that is based on the internet and set up in such a way that it can share processors and data between computers and other devices on demand. One of the biggest benefits of cloud computing is that it allows companies to avoid infrastructure costs up front, and therefore give more attentions to other projects. Additionally, cloud computing offers the ability to get applications up and running faster, and makes it easier to manage projects and spend less time maintaining them. In recent years, cloud computing has been in high demand due to its power, performance, price points, scalability, and availability. The most significant developments in this field have come about in the past decade. In 2008, NASA released the first open-source software package for developing private and hybrid clouds, called OpenNebula.[1]

Currently, there are three major service models in the cloud infrastructure. Together, these models form a stack: infrastructure, platform, and software. The most basic cloud service model is called infrastructure as a service (IaaS). IaaS refers to online services that abstract the user from the details of infrastructure like physical computing resources, location, security, etc.[2] To deploy their applications, cloud users install operating-system images and their software on the cloud infrastructure. In this way, the user patches and maintains the operating system. In platform as a service (PaaS), vendors offer a development environment to application developers. The provider then develops toolkits and standards for development as well as channels for distribution. It is important to note that in this platform, the provider givers a platform, usually an operating system and development environment on which the developers can develop and run their software solutions on a cloud platform. Examples of PaaS include frameworks like Microsoft Azure, the Google App Engine, etc.[3] There also exists specialized applications of PaaS. These include integration platform as a service (iPaaS) and data platform as a service (dPaaS). Furthermore, consumers of PaaS do not manage or control the cloud infrastructure of their network, only the deployed applications and configuration settings. The last service model is software as a service (SaaS). In this model, users gain access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS gives businesses the potential to reduce IT operational costs by outsourcing hardware as well as software maintenance directly to the cloud provider.[2]

Cloud computing is currently one of the fastest growing developments in the computing world. The economic case for it is extremely compelling, however there are a lot of concerns about the security challenges that it poses. Many businesses have accepted the economic case for cloud computing and more and more are moving towards similar solutions every day. Cloud computing providers can easily build large data warehouses for a relatively low cost to their customers. Also, as more individuals and businesses buy into these solutions, the resulting economies of scale increase revenue for cloud

providers at a lower cost for cloud users. However, while cloud computing provides a convenient option for businesses to utilize, the security of the system has become its greatest barrier to success.[4] Currently, there are too many risks to service availability, data confidentiality, etc. for cloud computing to be presented as a feasible solution for most people, but the demand is nonetheless increasing as the price goes down. For this reason, it's important to evaluate the problems in cloud security and their implications.

There are a number of concerns about the current state of cloud computing security. The biggest of these are access, compliance, location, recovery, viability, and availability. Access deals with who has administrator access to sensitive data as well as who makes decisions about hiring and supervision of such individuals. Compliance deals with potential issues like audits and security certifications from the outside. Unsurprisingly, the principle of location pertains to whether the cloud vendor allows for any control of the location of data. Recovery addresses the issue of whether data can easily be restored or recovered in the event of a disaster, and how much time would that take. Viability pertains to issues like what would happen if the client outlives the cloud vendor. Do they get their data back, and how is it returned to them. Finally, data availability means if there are periods when the vendor needs to change/update system requirements, is the data temporarily unavailable to the client.

As we have seen, there are three major delivery models in the cloud computing world, SaaS, PaaS, and IaaS. As these services gain more and more capabilities, the information security issues and risks grow as well. Recently, the SaaS model has emerged as the dominant service model in the cloud IT industry.[24] The biggest challenge for the adoption of SaaS applications in the cloud is addressing enterprise security concerns. Before security for this service model is addressed, it is necessary to define the terms that are used to talk about these issues. Words like vulnerability, threat and risk are often used to mean the same thing when they actually have different meanings. When people use the term vulnerability, they are talking about a hardware, software, or procedural weakness that may provide an attacker with an open door to enter a computer or network and have unauthorized access to specific resources.[5] In other words, a vulnerability allows an attacker to reduce a system's information assurance. Vulnerabilities can be defined as the combination of three elements: a system flaw, attacker access to the flaw, and attacker capability to exploit the flaw. A threat is any potential danger to information or systems. Threats occur when an attacker identifies a specific vulnerability and uses it against the system.[6] A risk is the likelihood of a threat agent taking advantage of vulnerability and its impact on the system. Risks tie all the vulnerabilities, threats and likelihood of an exploitation into a single variable. The most significant security elements to the SaaS model are data security, network security, data locality, data segregation, data access, and data authentication and authorization.[5]

Survey Details

Data Security

Data security in the cloud computing world is much different from more traditional application deployment models. Traditionally, an organization's data would reside on the premise and would be subject to whatever security policies were deemed necessary. However, using the SaaS service model, data is stored outside of the boundaries of the organization. As a result, the vendor of the cloud service must adopt additional security checks to provide adequate data security and prevent data breaches.[7] This is commonly done using strong encryption. In many cases, the administrators of the cloud vendors do not require access to customer accounts. However, when this behavior is needed, generally administrators gain access to a host using their individually encrypted cryptographically strong secure shell keys. When this happens, any account access is well logged and often audited. Some of the most common flaws in the data security component of the SaaS model are cross-site scripting, access control weaknesses, OS and SQL injection flaws, cross-site request forgery, and cookie manipulation.[8]

When a consumer and vendor of the SaaS model are communicating or exchanging data, it is important that data flow over the network be secured in order to prevent leakage of sensitive information. To ensure this, strong network traffic encryption techniques must be applied. Examples of these protocols are Secure Socket Layer (SSL) and Transport Layer Security (TLS).[8] Secure socket layer is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. Establishing an SSL connection first requires an SSL certificate. The web server will also create a public and a private key. An SSL certificate typically contains a domain name, company name, address, city, state and country. Additionally, it will contain the expiration date of the certificate as well as details of the certification authority that was responsible for the issuance of the certificate.[9] TLS, a successor to the secure socket layer, is another protocol that ensures privacy between communicating applications and their users on the internet.

The networking layer provides significant protection against more traditional attacks like the Man-In-The-Middle attack, IP spoofing, and port scanning.[8] However, a vulnerability in the SaaS model lies in packet sniffing. Packet sniffers, or packet analyzers, are computer programs that can intercept and log traffic that passes over a digital network or part of a network. Network penetration, packet analysis, and insecure SSL trust configuration pose the largest threats to the network security component of cloud computing.[10]

Denial of Service

As cloud computing is becoming more and more popular, large companies like IBM, Google, Amazon and Ebay are beginning to increase their investments. When the cloud architectures of these large institutions is threatened by attackers, often with some sort of Denial of Service attack, it's typical for the company to pay the ransom instead of having their systems go offline. Because DoS attacks are so difficult to prevent, it's a valid concern for cloud service customers. The two areas that are most important when it comes to cloud computing security are virtual machine vulnerabilities and message integrity between cloud systems.[22] Denial of service attacks, specifically, Distributed Denial of Service (DDoS) attacks, usually aim large numbers of unique IP packets at specific network entry points. Cloud computing is a logical target for DDoS attacks because of the nature of its shared infrastructure among many clients. This is unfortunate because DDoS attacks are inherently hard to protect against. There are some Intrusion Prevention Systems (IPSs) that are widely used. However, they are only effective when the attacks are identified and have pre-existing signatures to check against. Also, firewalls are ineffective at preventing such malicious attacks because the attacker designs the requests to appear like legitimate traffic.[22] DDoS attacks can be particularly threatening if launched from botnets with a large number of zombie machines.[23]

A particular variant of the DoS attack is XML based DoS, (XDoS). Web services based on XML are particularly vulnerable to XDoS and can be brought down by flooding the network with XML messages to halt legitimate network traffic, flooding the service with XML requests to disrupt the availability of the service, or by passing malicious content in XML form in a request to disrupt the service and delay legitimate traffic.[22] Some of the vulnerabilities of cloud computing that can lead to XDoS attacks are buffer overflow, entity expansion, entity reference, and oversize payload. In a buffer overflow attack, a variant of XDoS attacks, an attacker can execute hostile code on a web server that will cause the space on the allocated buffer to overflow.[11] An example of when this attack can take place is when a DOM parser is used.[24] Even following a simple request a large XML document can overflow the memory of the system due to the loading of the document prior to parsing. Additionally, code could be injected to upload the request message in the buffer in an infinite loop, thus causing a buffer overflow. The nesting capabilities of XML also create vulnerabilities for overflow attacks because there is no restriction on the level of nesting.[16] If there are enough nested clauses in an XML document this can exhaust the CPU and cause an XDoS attack. This behavior is known as XML bombing. One of the benefits of using XML can also sometimes be a drawback. Because XML allows users access to an international standard to follow we can always encode plain-text messages without any ambiguity.[12] However, this adds vulnerability to the XML system because an attacker could send the same message to a web service multiple times using different Unicode styles and the system will be unable to detect them as the same request.[22] Most firewalls have some internal logic to block repetitive requests, but in this case they will be ineffective.

Another common variant of the DoS style attack is the HTTP DDoS attack, or HDoS for short. HDoS attacks are particularly troublesome because they make it very difficult to distinguish attack traffic from legitimate HTTP requests.[23] Additionally, the attack steals resources from the web application and not the TCP/IP stack, so it can easily take a system offline. However, just as cloud computing faces security vulnerabilities, so do the attacks against it. HDoS attacks can be stopped using a system called "tarpitting". Tarpitting basically takes advantage of the TCP/IPs window size. Typically, during a normal tarpitting session, the server sets the TCP window size to just a few bytes and the stack does not send any more data than will fit into the TCP window. By following this protocol, we are able to handle packet loss in a convenient way.[23] However, when the host machine is attacked and doesn't respond, the attacker's CPU load is minimal and able to handle other requests normally. The downside is that the target CPU often faces a large CPU load and becomes unresponsive.[14] The takeaway from this is that there are possible ways to deal with denial of service type attacks on cloud computing architectures, but they may need to evolve a bit more before they are widely accepted.

Possible Solutions

We have already seen that there are a number of DoS and variant attacks that can pose a serious threat to the success of cloud computing. However, these threats have their own vulnerabilities. Many people are scrambling to devise solutions to these malicious attacks. Some of the most famous are the use of Service Oriented Traceback Architecture (SOTA), and PreSODoS.

SOTA

The first of the two potential solutions for DoS attacks we will examine is the SOTA traceback architecture. There are two main categories for IP traceback schemas: proactive and reactive.[19] In reactive traceback, the system simply does it's best to respond to an attack and not prevent it in the first place. One downside to this is that it must be active while the attack is happening. Reactive schemas typically rely on ISP cooperation for them to work properly and this is not always feasible. Proactive schemas record trace information as packets cross over the network. By recording this traffic, the cloud platform is able to reconstruct the path that the specific packets have taken and therefore identity who and where the attack came from. The goal of the service oriented traceback architecture is to apply a SOA approach to previously existing traceback methodology so as to identify the source of the DDoS attack.[25]

In general, SOTA is deployed at the edge routers so that it can be closer to the source end of the network. Then, all service requests are first sent to SOTA for marking. By placing the SOTA before the web server, the cloud service provider is able to remove their address and prevent a direct attack.[25] When an attack is discovered, or even when one succeeds, the victim of the attack can recover the marked

tag and determine where it came from for future knowledge. Also, when a client attacks a server, they request a web service from the SOTA and the service request is sent to the server. At this point the client creates the SOAP message request based on the proper protocols.[11] Once the SOAP request has been recieved by the server, a marked tag is placed in the header based on whether the message is legitimate or not. It's important to note that the SOTA does not directly work with any requests or outgoing messages. When talking about the SaaS platform of cloud computing, SOTA follows a number of protocols. This is rooted in the philosophy of the service oriented architecture, in that SOTA is loosely coupled, handles message based interaction, promotes dynamic discovery, practices late binding, and is a policy based behavior.[25]

There are also variants to the traditional SOTA system. Chonka and Xiang (2008) propose a mini SOTA approach that can be created with its own web service definition language (WSDL).[22] The WSDL is used to communicate to all clients about what servers are currently available and how they can be requested. Additionally, the mini SOTA method lets cloud service providers split their services into different categories.[13] By following this behavior, service resources can be made more efficient as well as effective. Also, searching for different web services inside a mini WSDL is often simpler than a full scale version.

PreSODoS

Preventing Service Oriented Denial of Service (PreSODoS for short) is a method first devised by Padmanabhuni, Singh, Kumar, and Chatterjee in 2006 to detect and prevent XML based denial of service attacks against cloud computing architectures.[24] The project was motivated because prior to its invention, XML validation was the only feasible defense against XDoS attacks. However, this procedure has a high cost in most hardware systems. Also, they are difficult to upgrade, which is vital as denial of service attacks are written to constantly evolve so that they stay effective.[17] PreSODoS was an attempt at creating a better prevention system. Because one of the shortcomings of XML processing is that it is computationally expensive, a better algorithm needed to be implemented to verify XML documents. The answer to this was to use a data structure that Padmanabhuni et al. refer to as a Patricia Trie, and is commonly referred to as a radix trie or tree.[24] A radix tree is a data structure that represents a space-optimized trie in which each node that is the only child is merged with its parent. This results in a trie such that the number of children of every internal node is at least the radix r of the radix trie, where r is a positive integer and a power x of 2, having $x \geq 1$.[20] Radix trees differ from more regular trees because edges can be labeled with sequences of elements as well as single elements. An example is if there are two strings *abcde* and *abccc*, then *abc* would be one node and *de* and *cc* would be the two different leaves. These tries were utilized in the PreSODoS framework as a mechanism to store all information available in a schema but provide fast traversing and quick comparisons between multiple

documents. This is a very convenient data structure to use because the worst case of a key lookup of length n is $O(n)$.[24] In contrast, a data structure like a binary search tree would take $O(\log n)$ time, because it depends on the depth of the tree.[16]

The first step to the PreSODoS proposal is to create a repository of radix tries of schema for all the services. Once the tries have been created, the framework can be deployed and will begin receiving request messages.[24] In addition to the tries themselves, two tables are used to store and element ID and name. Examples of these elements include things like XML headers, IP addresses, message size, body, subject, etc.[12] Once all of this is set up, the PreSODoS looks into the schema and searches for elements that are declared either without restrictions or restrictions that don't match up with the SOAP request. Examples of these kind of restrictions include max size of different fields and timeouts for external references to the system. There are also two different logging mechanisms attached to the system.[24] At this point, PreSODoS can be deployed to the cloud service provider's architecture and will intercept every request message following the processes described above. As we have seen, the PreSODoS framework provides a thought-out approach to preventing XDoS attacks at a time when the value for this sort of system is growing exponentially.

Evaluation

The first possible solution we examined was the service oriented traceback architecture (SOTA). Chonka and Xiang (2008) found that it took an average of 2 seconds more of processing time using the SOTA approach than the more traditional system.[22] They interpreted these results as meaning that the SOTA is much more effective and efficient than the original authentication procedure. They inferred that one of the reasons that SOTA is able to respond quicker to the messages than SOAP authentication is that the cloud service provider only has to access SOTA if they need the source of the message request. This behavior is desired because it will lessen the load on machine resources during some sort of denial of service attack.[22] However, many more research projects need to be devoted to this approach before it becomes widely used in the cloud computing environment. It's possible that the results from one paper would not be the same as results from another. Also, different cloud architectures may behave differently under different circumstances.

The second approach to solving the DoS vulnerability in cloud computing was the PreSODoS framework developed in Infosys Technologies Ltd. in Bangalore, India. The authors found that it provides an overarching agile approach to XML based DoS prevention due to its adoption of a more efficient knowledge data structure, a radix trie.[24] Also, by combining the new approach with already existing security mechanisms and content introspection, it can become very practical for cloud service providers and their customers. However, the authors emphasize that this is still a work in progress. There are a number of extensions to the framework that are currently in the planning phases but have

not yet been implemented. Once more time and research has been devoted to these add-ons, PreSODoS could have a good shot at rendering denial of service attacks against cloud structures ineffective. It would be in the best interests of cloud service providers and their customers to invest more in this kind of technology.

Summary of Trends

More work in the field of cloud computing has been done in the last decade than any previous amount of time. As we have already seen, cloud computing appears to be an attractive business solution to many. However, its greatest weakness is its security, which is a major concern depending on the customer. In providing a secure cloud computing solution, a big decision is to decide the type of cloud to be implemented. There are currently three main types of cloud deployment models: public, private, and hybrid clouds.[19] These all offer different advantages and disadvantages in terms of performance and security.

- Public Clouds

  The model of the public cloud allows user access to the cloud through an interface like an internet browser. In general, it's either free of maintained on a pay-per-use system, meaning the client will pay more during periods of higher demand.[15] In terms of what the user sees, there may appear to be no difference between public and private cloud deployment models. However, the implications for security can differ substantially depending on the service. Some of the more famous public cloud service providers like Amazon, Microsoft, and Google own and operate the entire infrastructure at their own private data warehouse. The data is access using an internet connection.

- Private Clouds

  The concept of a private cloud computing system is intended for a single organization, and is either managed internally or by some third party. The private cloud model raised security issues that must be addresses in order to prevent dangerous security loopholes.[18] Furthermore, private cloud centers often require a lot of space and resources, in addition to a significant level of engagement by the business. Because of this, many individuals choose the public cloud route because they don't end up saving money or resources. In this way, private cloud models aren't as economical as it's public or hybrid cousins.[21]

- Hybrid Clouds

  Unsurprisingly, a hybrid cloud is a mix between the public and private cloud models. Examples of scenarios when hybrid clouds may be beneficial include situations when private client data may be stored on a private cloud application, but then that application is connected to a business

intelligence application deployed on a public cloud. Another example would be when a public cloud is used to meet temporary capacity needs that cannot be met by the private cloud. This technique is known as cloud-bursting, and it is employed when an application normally runs on a private cloud but automatically transfers (or bursts) to a public cloud when the demand for computing capacity reaches a certain level.

Conclusion & Future Work

In conclusion, cloud computing provides clear advantages to individuals and businesses due to lowered price and convenience. The rate at which cloud solutions are utilized continues to grow exponentially and there is little doubt that trends will continue in the same direction. However, there remain a lot of practical problems that need to be solved. These problems come in several forms; some are more technical threats and vulnerabilities of the specific code behind the scenes, whereas others are more ethical or legal questions about how parties should behave in certain circumstances. Since this is still such a new field of research, there remains a lot of opportunity for researchers to make new discoveries in this field. We have a long way to go before this becomes a perfect solution for IT enterprises. Also, when these companies move their business to the cloud, they should be aware of the existing threats and instead of simply relying on the cloud service vendor to address the vulnerabilities, they should take an active role in the process as well.

References

1. Mervat Adib Bamiah, Sarfaz Nawaz Brohi, *Seven Deadly Threats and Vulnerabilities in Cloud Computing*, International Journal of Advanced Engineering Sciences and Technologies, 2011

2. Jianhua Che, Yamin Duan, Tao Zhang, Jie Fan, *Study on the security models and strategies of cloud computing*, Procedia Engineering, 2011

3. Wentao Liu, *Research on Cloud Computing Security Problem and Strategy*, Wuhan Polytechnic University, 2014

4. Uma Somani, Kanika Lakhani, Manish Mundra, *Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing*, International Conference on Parallel, Distributed and Grid Computing, 2010

5. Chunming Ronga, Son T. Nguyena, Martin Gilje Jaatun, *Beyond lightning: A survey on security challenges in cloud computing*, Computers & Electrical Engineering, 2013

6. Farhan Bashir Shaikh, Sajjad Haider, *Security Threats in Cloud Computing*, International Conference on Internet Technology, 2011

7. Farzad Sabahi, *Cloud Computing Security Threats and Responses*, Azad University, 2013

8. Kuyoro S O, Ibikunle F, Awodele O, *Cloud Computing Security Issues and Challenges*, ResearchGate, 2012

9. Ramgovind S, Eloff MM, Smith E, *The Management of Security in Cloud Computing*, School of Computing, University of South Africa, 2009

10. Yanpei Chen, Vern Paxson, Randy H. Katz, *What's New About Cloud Computing Security?*, UC Berkeley, 2010

11. Balachandra Reddy Kandukuri, Ramakrishna Paturi V, Dr. Atanu Rakshit, *Cloud Security Issues*, IEEE International Conference on Services Computing, 2009

12. Mohamed Al Morsy, John Grundy, Ingo Muller, *An Analysis of The Cloud Computing Security Problem*, APSEC Cloud Workshop, 2010

13. Meiko Jensen, Jorg Schwenk, Nils Gruschka, Luigi Lo Iacono, *On Technical Security Issues in Cloud Computing*, IEEE International Conference on Cloud Computing, 2009

14. Radut Carmen, Popa Ionela, Codreanu Diana, *Cloud Computing Security*, IEEE International Revista Economica, 2012

15. Cong Wang, Qian Wang, and Kui Ren, Wenjing Lou, *Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing*, IEEE Communications Society, 2010

16. S Subashini, V Kavitha, *A Survey on Security Issues in Service Delivery Models of Cloud Computing*, Journal of Network and Computer Applications, 2011

17. Peter Mell, Timothy Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology, 2011

18. Sean Carlin, Kevin Karlin, *Cloud Computing Security*, International Journal of Ambient Computing and Intelligence, 2011

19. Ankur Mishra, Ruchita Mathur, Shishir Jain, Jitendra Singh Rathore, *Cloud Computing Security*, International Journal on Recent and Innovation Trends in Computing and Communication, 2013

20. Jon Brodkin, *Seven Cloud-Computing Security Risks*, Network World, 2008

21. Dimitrios Zissis, Dimitrios Lekkas, *Addressing Cloud Computing Security Issues*, Future Generation Computer Systems, 2012

22. Ashley Chonka, Yang Xiang, Wanlei Zhou, Alessio Bonti, *Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks*, Journal of Network and Computer Applications, 2011

23. Joe Stewart, *HTTP DDoS Attack Mitigation Using Tarpitting*, SecureWorks, 2007

24. Padmanabhuni, Singh, Kumar, Chatterjee, *Preventing Service Oriented Denial of Service (PreSODoS): A Proposed Approach*, Software Engineering and Technology Labs, 2006

25. Chonka, Zhou, Xiang, *Protecting Web Services with Service Oriented Traceback Architecture*, School of Engineering & Information Technology, Deakin University, 2008