Preventing Cache Based Side-Channel Attacks for Security in Cloud over Virtual Environment

S. Vengadesan, B. Muthulakshmi

[1]PG Student, Department of Computer Science and Engineering, A.V.C. College of Engineering, Mayiladuthurai, Tamil Nadu, India

[2]Assistant Professor, Department of Computer Science and Engineering, A.V.C. College of Engineering, Mayiladuthurai, Tamil Nadu, India

Abstract: ***Cloud computing is a unique technique for outsourcing and aggregating computational hardware needs. By abstracting the underlying machines Cloud computing is able to share resources among multiple mutually distrusting clients. While there are numerous practical benefits to this system, this kind of resource sharing enables new forms of information leakage such as hardware side-channels. The usage of CPU-cache based side-channels in the Cloud and how they compare to traditional side-channel attacks. New techniques are necessary to mitigate these sorts of attacks in a Cloud environment, and specify the requirement for such solutions was developed. The security issues faced by cloud service providers and the service issues faced by the cloud customers are analysed. In the existing system, providing security in cloud requires a huge amount of money based on the service on demand in cloud environment. The extensive use of virtualization in implementing cloud environment brings unique security providence for the cloud customers and all other resellers &subscribers of a public cloud service access. In the proposed system, an effective firewall security has been implemented for blocking and filtering the unwanted requests from the clients before the request reaches the virtual machine destination. During the request processing, if the user requests the high level data from the cloud, the data can be provided against payment.***

Keywords: Cloud security, Security architecture, Security and privacy, Cloud Computing, CPU Cache, Parallel Side-Channel, Performance, Side-Channel, Security, and Sequential Side-Channel

1. Introduction

The cloud system can be hacked via the caching concept of fetching Relative data and access the same via union hacking techniques. I don't have proposed system to overcome or avoid hacking the data in the cloud Server due to union caching technique. This technique provides the error output to the user immediately. An effective firewall security has been implemented for blocking and filtering the unwanted requests coming from the clients before the request approach to the virtual machine**.** Security issues in cloud concerns and mainly associated with security issue faced by cloud service providers and the service issues faced by customers**.** Security issues in cloud concerns and mainly

Associated with security issues faced by cloud service providers and the service issues faced by the cloud customers. In the proposed system, an effective firewall security has been implemented for blocking and filtering the unwanted requests coming from the clients before the request approach the virtual machine.

2. Background

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

The five essential characteristics are as follows:
- On-demand self-service
- Ubiquitous network access
- Resource pooling
- Location independence
- Rapid elasticity
- Measured service

A. The Cloud service model

1. **Cloud Software as a Service (SAAS)**—use provider's applications over a network.
2. **Cloud Platform as a Service (PAAS)**—deploy customer-created applications to a cloud.
3. **Cloud Infrastructure as a Service (IAAS)**—rent processing, storage, networkcapacityand other fundamental computing resources.

B. The Deployment Model

It can be either internally or externally implemented, are summarized in the NIST presentation as follows:
- **Private Cloud**—enterprise owned or leased
- **Community Cloud**—shared infrastructure for specific community
- **Public Cloud**—sold to the public, mega-scale infrastructure
- **Hybrid Cloud**—composition of two or more clouds.

3. Basics of Virtualization

Virtualization is the idea of partitioning or dividing the resources of a single server into multiple segregated VMs. Virtualization technology has been proposed and developed over a relatively long period. The earliest use of VMs was by IBM in1960, intended to leverage investments in expensive mainframe computers.

Virtualization can be defined as the abstraction of physical resources into logical units, such that a single physical resource can appear as many logical units and multiple physical resources can appear as a single logical unit. The primary motivation behind virtualization is to hide the physical characteristics and irrelevant details of these resources from their end users.

Thus, each user gets the illusion of being the lone user of that physical resource (one-to-many virtualization), or multiple physical resources appear as a single virtual resource to the user (many-to-one virtualization). In computing, virtualization means to create a virtual version of a device or resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments.

Virtualization is the running of several operating systems on a single host se1rver. The host server runs special software that sits between the server and the guest operating systems and parcels out resources as needed. The guest operating systems are unaware that they are virtualized.

Virtualization enables today's X86 computers to run multiple operating systems and applications, making your infrastructure simpler and more efficient. Applications get deployed faster, performance and availability increase and operations become automated, resulting in IT that's easier to implement and less costly to own and manage. Virtualization is a key technology used in datacenters to optimize resources. Even something as simple as partitioning a hard drive is considered virtualization because you take one drive and partition it to create two separate hard drives. Devices, applications and human users are able to interact with the virtual resource as if it were a real single logical resource.

A. One-to-many virtualization

Consider the familiar example of virtualizing an x86 servers in which software, called a virtual machine monitor or hypervisor, allows multiple virtual machines (VM) to run on the same physical server. Each VM emulates a physical computer by creating a separate operating system environment.

The ability to run multiple VMs means that we can now simultaneously run multiple operating systems on the same underlying physical machine. The operating system running inside the VM gets the illusion that it is the only operating system running on that host server; one physical machine has effectively been divided into many logical one.

Storage virtualization: Integration of multiple network storage devices into what appears to be a single storage unit. Server virtualization is of partitioning a physical server into smaller virtual servers.

Operating system-level virtualization: a type of server virtualization technology which works at the operating system (kernel) layer. Operating System (OS) virtualization loads a specialized layer of software on top of a base operating system running on a physical server. This software layer divides the system into containers each with its own file system, process tables, authorized users and networking.

Network virtualization: using network resources through a logical segmentation of a single physical network. Network virtualization presents logical networking devices and services logical ports, switches, routers, firewalls, load balancers, VPNs and more connected workloads. Applications run on the virtual network exactly the same as if on a physical network.

B. Desktop Virtualization

Deploying desktops as a managed service gives you the opportunity to respond quicker to changing needs and opportunities. We can reduce costs and increase service by quickly and easily delivering virtualized desktops and applications to branch offices.

4. The Xen Hypervisor

Xen is a specific, open-source implementation of a bare-metal hypervisor. At the time of writing, Xen is known to be quite developed and is being used as the backbone for many established Cloud enterprises, including Amazon's EC2. The combined nature of its canonical use and its open source access make it a prime candidate for experimentation relating to hypervisors in the Cloud. Xen functions as a combination of a bare-metal hypervisor (the Xen hypervisor), a single host domain (Dom0), and any number of guest domains (DomUs, or guests). Dom0 is a customized Linux OS capable of interacting with the hypervisor. It acts as the administration tool for the system. The DomUs represent the guest machines that clients will be using. The hypervisor controls basic operations, such as the CPU scheduling and memory management of the system. Complex functionality such as networking and IO are handled by the Dom0 after being redirected from the guests. The guest operating systems are, of course, unaware of the context in which they are running. This type of software isolation has a bonus of acting as a security mechanism to separate one VM from another, working under much the same lines as sandboxing. Unfortunately, side-channels have been demonstrated to bypass this sort of isolation. A graphical interpretation of the Xen hypervisor and its chief components can be seen in Figure

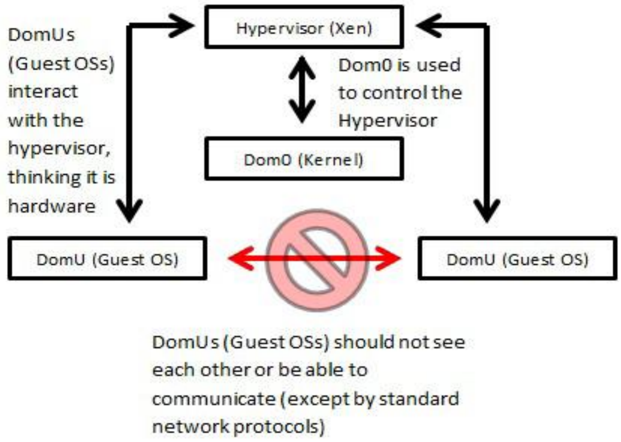
DomUs (Guest OSs) interact with the hypervisor, thinking it is hardware
Hypervisor (Xen)
Dom0 is used to control the Hypervisor
Dom0 (Kernel)
DomU (Guest OS)
DomU (Guest OS)
DomUs (Guest OSs) should not see each other or be able to communicate (except by standard network protocols)

Figure 1: Architecture of the Xen Hypervisor

5. CPU Cache

A CPU cache is a small section of memory built into the CPU. This memory acts as a medium through which any request for data must go. It serves to increase the speed of memory access for more commonly accessed data. Essentially the cache is a section of memory that can be accessed much more quickly, but is much smaller than main memory. Dedicating a small amount of memory to a cache can lead to massive speed increases, as CPUs often require frequent access to the same memory addresses. Keeping these frequently accessed data in the cache reduces the time needed to access these data, and therefore increases the speed of the program. A modern-day CPU contains multiple levels of cache dedicated to different purposes. The most common organization is depicted in Figure. In this figure, thedata storage areas are organized from the top-down by decreasing speed of access and by increasing size. This puts the smallest, and fastest storage mechanisms on the top, and the slowest/biggest on the bottom. When data is required by the CPU (for the registers), it is first requested from the L1 cache. The L1 cache is unique in that it is often divided into separate instruction and data caches. The speed requirements for this cache, as it will be accessed most often, mean that it is usually virtually indexed. This means that the mapping of a memory address to the cache position is determined by its virtual address (as viewed by a process) as opposed to its physical address (as viewed by the operating system).

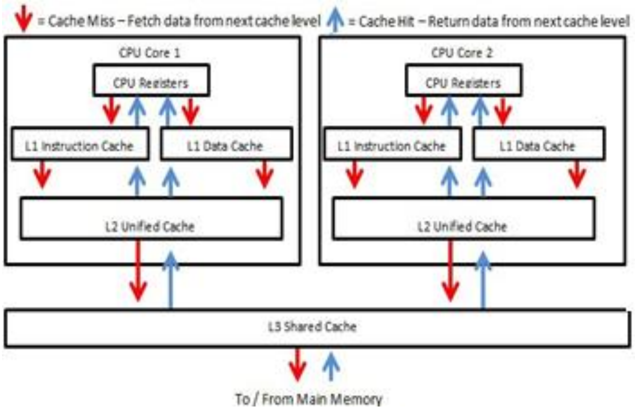

Figure 2: The Cache Hierarchy

6. Side-Channel Attacks

A side-channel (or covert-channel) in a software program is a means of communication via a medium not intended for information transfer. It typically involves correlating the higher level functionality of the software with the underlying hardware phenomena. With an established correlation, these phenomena can be measured and analyzed to infer what is occurring within the software program at a given time. While the measured phenomena vary with the specific properties of the hardware in question, any phenomenon that can be reliably correlated to the software's function can be used as a side-channel. Examples may include the timing of specific hardware functions, or the acoustic properties of a hardware device. Typically, the higher-rate hardware functions are more interesting to explore as side-channels because they can communicate information more quickly, and therefore can yield more details about the state of the program in execution. To this end, CPU cache-based side-channels typically receive the most attention, as they are one of the highest-rate measurable resources shared between processes. Traditionally, cache-based side-channels have been used to glean the functionality of closed source systems and functions. Examples include the breaching of cross-VMsystems, and the breaking of the Advanced Encryption Standard (AES) and Data Encryption Standard (DES) encryption algorithms. An experiment in 2003 used a timing-based measurement system on the CPU cache-channel to determine cache hits and misses in a DES encryption algorithm. By correlating the timing measurements with the execution of the algorithm, the authors were able to break the DES cypher in more than 90% of their attempts.

7. Cache Channel Attacks

In order for the CPU cache to be used in a side-channel attack, the cache must insome way be shared by the attacker and their target. A Cloud environment makes this condition particularly easy to achieve as both the attacker and target can get access to the same physical machine. Typically, a CPU cache can be shared in two ways, either the cache is exclusive to one CPU core, in which case two processes must access the cache sequentially; or cache is shared between CPU cores, in which case two processes can access the cache concurrently. We refer to the first type of attack as a sequential, cache channel, and to the second as a concurrent, or parallel, cache channel. Research has been done into attacks for both classes of channel with the former typically seen as more portable, as only some systems will allow for parallel access to a cache. While the techniques for attacking these two types of cache are quite similar, the hardware differences require dramatically different solutions. In order to addressboth types of channels we have devised two solutions. For the sequential channel we apply a technique called Selective Cache Flushing. For parallel channels we apply a technique called Cache Partitioning, based on cache coloring, which can be found in Chapter 6. These two solutions can be implemented in conjunction to insulate a hypervisor against both types of side-channel.

8. Cache Colouring

Cache coloring has been used in the past primarily as an optimization scheme tomaximize the cache hit rate. Previous work by Tam et al. explored the use of cache coloring in environments where multiple processes shared a common cache. In these experiments, they used a cache coloring technique to increase the cache hit rate by minimizing the number of cache data evictions occurring across processes. In our experiments, we use a similar technique to partition the cache on a virtual machine level granularity. While some similar results are observed in the reduction of cache misses in certain cases, our priority was the security of the system. As a result, the technique was implemented with some variations to guarantee the security features, as opposed to the performance features, of the system. Two recent papers attempt to use a form of cache partitioning to prevent cache-based side-channels. Both focus on reserving a small portion of the cache on a per-VM or per-core basis using cache-coloring that can dynamically be accessed by VMs when they need to execute a side-channel proof" process. The upside to these attempts is that they attempt to incur very minor overhead by maximizing each VM's access to the cache and by securing portions of the VM dynamically. The downside is that both of these solutions require the client programs to understand that they are running in a modified environment to take advantage ofthe security features. While minor, these alterations violate the Cloud Model.

9. Implementing Diagram

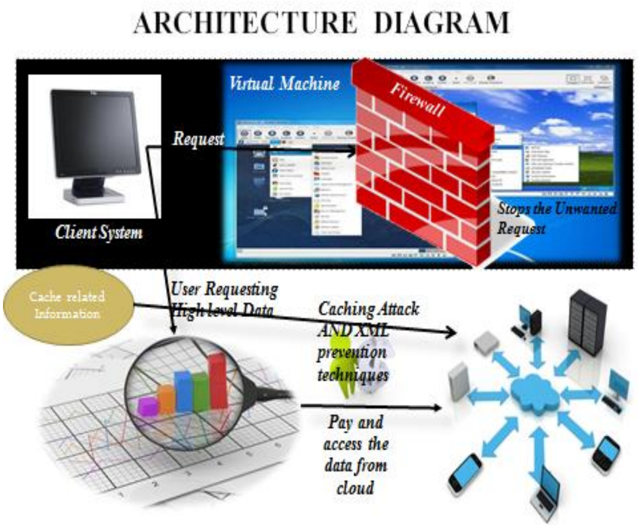
ARCHITECTURE DIAGRAM
Virtual Machine
Firewall
Request
Client System
Stops the Unwanted Request
Cache related Information
User Requesting High level Data
Caching Attack AND XML prevention techniques
Pay and access the data from cloud

Figure 3: Architectural Diagram

10. Existing System Work

- In the existing system, providing security in cloud option is a huge amount of pay, based on the service of usage by the customers in cloud environment.
- The extensive use of virtualization in implementing cloud environment brings unique security providence for the cloud customers and all other reseller's &subscribers of a public cloud service access.
- The request raised by client to the cloud server by stopping unwanted request by firewall.
- The unwanted request will be stored in virtual machine not raised to cloud server.
- The cloud system can be hacked via the caching concept of fetching Relative data and access the same via union hacking techniques.
- I don't have proposed system to overcome or avoid hacking the data in the cloud Server due to union caching technique. This technique provides the error output to the user immediately.

11. Proposed System Work

- In the proposed system, an effective firewall security has been implemented for blocking andfiltering the unwanted requests coming from the clients before the request approach the virtual machine.
- During the request processing, if the user requests the high level of data from the cloud, then based on the payment made by the cloud user, they can use and access the data's from the cloud server.
- The MAC (Media Access Control) address, IP address and system information will be blogged if an unauthorized or unsolicited person trying to access.
- Fast computing
- Highly authenticated user only can access the information.
- The users have to pay if the user wants high level data.To overcome the caching union hacking issue, we provide an innovativexsd validation technique which validates the database Dynamically and try to check out the incoming caching request and,
- In case of hack queries, an automatic process of filtering mechanism will be implemented on the system to restrict the queries to be passed.

12. Modules Description

A. Firewall Creation Module

A Firewall is a system designed to prevent unauthorized access to or from a private network (especially Intranets).Create a firewall rule that permits the ping command first and customize the implementations type.Using this rule to deploy all windows server and create a specific filter.Using this rule to verify the remote servers and work stations along with ping configuration.

B. Virtualized Firewall Creation Module

A firewall product is required to support virtual devices in most of its firewall features.In network configured zones, not necessary to configure security policy for each interface in a firewall network.Create resource based packet filtering within same virtual device to remove zones in a network. RBPF in different virtual devices are also accepted.

C. Data Access Module

If the IP address of request is within one of the ranges specified in server level firewall rules, the connection is granted to SQL Database server has a matching database-level rule.If the IP address request is not within the ranges specified in server level firewall rules mean, connection failed otherwise database firewall rules are checked.The connection established only when the client passes through firewall in SQL database.

D. Cost Computation Module

Flexible cloud hosting services, reliable and secure information all those involved in cost computation. It produces very low rate for the compute capacity is actually consuming and produce high performance over data.Having route access to each one and interact with machine, retainingdata based on boot partition also added an advantage.

E. Blocked User Module

Firewall that allows to blocking programs from being accessed by other people on the internet or network. It helps to keep computer secure.Testing a blocking rule, this rule used to test the website and block the website by network administrator.To create a content filter to block user access in group of websites in a network. Troubleshooting the block page to avoid unauthorized person using a network.

F. Mac Privilege Module

Mac address is a unique address assigned to almost all networking hardware's (ex: Mobile phones).Creating firewall rules based on Mac address this also very effective while accessing system from cloud server.It Address filters to prevent devices from sending outgoing TCP/UDP traffic to the WAN.

G. System Information Module

Mostly to check whether the person is authenticated user or unauthenticated user in a database while access the information in cloud server.Authenticated user information is stored in database this helps to make a user to access the cloud server.And, system information (IP address, Mac address) are also checked in a database to allow the user to utilize the system.

H. Performance Evaluation Module

Adoption of cloud, virtualization and mobility providing more vulnerabilities than ever for hackers to exploit.Now a day, Firewall performance based on shares and information about applications, attack signatures and address is increased.Firewall needs to manage flows between tiers of virtualized servers to increase the performance in a line-server.

I. Cache Hacking And Filtering Module

This module provides the functionality of hacking the data due to the output data cached from the system or the output retrieved from the system.To overcome the issue of caching information to retrieve the data, we proposed an innovative model of xml validation technique in which the data or query will be filtered through an xsd technique and the input query will be formed as xml and it's checked for strongly typed basis.

13. Conclusion and Future Work

A Cloud service discovery system specially designed for users for finding Cloud service over the internet is provided.Cloud ontology is also introduced for enhancing performance of the CSDS. The contributions of this work includes: 1) building of the Cloud service discovery system and 2) constructing the Cloud ontology. It is the first attempt in building an agent-based discovery system that consults ontology when retrieving information about Cloud services. In future when the Cloud computing is more commonly and widely used, it can be helpful for Cloud users to find a Cloud service under their specific preference. By consulting a Cloud ontology to reason about the relations among Cloud services, the CSDS is more successful in locating Cloud services and more likely to discover Cloud services that meets consumer requirements. Two unique security attributes of the Cloud motivated this research. First, the Cloud's architecture is particularly susceptible to cache-based side-channel attacks. Second, such attacks in the Cloud cannot be solved by conventional means without interfering with the Cloud model. To address these problems, two new techniques were designed to prevent cache-based side-channels. One is for dealing with sequential side-channels, and the other for parallel side-channels.

In future work, the overhead imposed by the solutions can be reduced and a practical solution deployed in an enterprise Cloud Environment was developed. The Cloud service discovery system is enhanced bymaking more depth of the Cloud ontology so that it can make more difference between two services in terms of service utility, completing functionalities of query processing, filtering and rating.

References

[1] Michael Godfrey and Mohammad Zulkernine. A server-side solution to cache-based side-channels in the cloud. In Proceedings of IEEE Cloud 2013, pages 163–170, 2013.

[2] Jicheng Shi, Xiang Song, Haibo Chen, and BinyuZang. Limiting cache based side-channel in multi-tenant cloud using dynamic page coloring. In Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops, DSNW '11, pages 194–199, Washington, DC, USA, 2011. IEEE Computer Society.

[3] Takabi, H., Joshi, J. B., &Ahn, G. J. (2010). Security and Privacy Challenges in Cloud Computing Environments. IEEE Security & Privacy, *8*(6), 24-31.

[4] Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., &Shenker, S. (2002). Controlling high bandwidth aggregates in the network. ACM SIGCOMM Computer Communication Review, 32(3), 62-73.

[5] S. Butt, et al., "Self-service cloud computing," in Proc. 2012 ACM ComputerCommunicationSecurity Conf.

[6] T. C. Chieu, et al., "Dynamic scaling of web applications in a virtualized cloud computing environment," in Proc. 2009 IEEE Int. Conf. e-Business Eng.

[7] S. T. Jones, et al., "VMM-based hidden process detection and identification are usingclosed," in Proc. 2008 ACM Virtual Execution Environments.

[8] K. Beaty, et al., "Network-level access control management for the cloud," in Proc. 2013 IEEE Int. Conf. Cloud Eng.

[9] J. Somorovsky, et al., "All your clouds belong to us security analysis ofcloud management interfaces," in 2011 ACM Compute. CommunicationSecurity Conf.

[10] P. Barham, et al., "Xen and the art of virtualization," in Proc. 2003 ACM Symp. Operating Syst. Principles.

[11] R. Beverly, R. Koga, and K. C Claffy, "Initial longitudinal analysis of IP source spoofing capability on the Internet," July 2013.