Data Security in the Cloud

Matt Lauer
University of Minnesota, Morris
600 East 4th Street
Morris, MN 56267
laue0095@morris.umn.edu

ABSTRACT

Cloud computing is an emerging trend in the provision of computing resources. For economical reasons users are outsourcing applications and data storage to the cloud, a managed hardware infrastructure providing various services. As the cloud grows it becomes necessary to secure the data and applications from unwanted attackers. This paper examines a combination of traditional methods for virtualization security and data transfer along with new methods of securing data and virtual machines. With these techniques, a user's cloud applications and data may be secured from unwanted intrusions.

Categories and Subject Descriptors

C.2.m [**Computer-Communication Networks**]: Miscellaneous; D.4.6 [**Security and Protection**]: Access Controls

General Terms

Security

Keywords

Cloud Computing, Virtual Machines, Amazon EC2, Data Security

1. INTRODUCTION

In 1961, computing pioneer John McCarthy predicted that "computation may someday be organized as a public utility" [3]. Cloud computing brings that prediction one step closer to reality. Today, with just a credit card, anyone can obtain access to a vast array of computational resources.

Cloud computing represents a recent paradigm shift for the provision of computing infrastructure which outsources computation and storage requirements of applications and services to a managed infrastructure. The University of Minnesota's recent shift to Gmail and the Google Apps system is a real-world example of outsourcing applications and data to the cloud. It is often more economical, easier and faster for companies and universities to transfer storage and computation requirements to managed systems with a larger resource set [3].

However, when private data is stored on public infrastructure, many extra steps are necessary to ensure the data remains private.

In the following sections we will examine modern methods of virtual machine security and take a look at Amazon's Elastic Cloud Compute (EC2) infrastructure. Then we will look at securing data flow between applications and methods of enforcing cloud accountability.

1.1 Defining the Cloud

Cloud computing is a very young concept and there is no consensus on a formal definition at the time of writing. Most experts agree that cloud computing is a buzz which encompasses a variety of services [3, 5]. Some definitions claim immediate scalability and usage optimization are key ingredients. Others focus on the business model which is typically a pay-as-you-go service. Others claim data centers and/or virtualization is the primary basis for cloud computing. Consequences for ambiguity may turn cloud computing into a generalized and overused concept of outsourced computing.

The following definition approaches cloud computing from a broad conceptual level:

> [Cloud computing represents] a broad array of web-based services aimed at allowing users to obtain a wide range of functional capabilities on a "pay-as-you-go" basis that previously required tremendous hardware/software investments and professional skills to acquire. Cloud computing is the realization of the earlier ideals of utility computing without the technical complexities or complicated deployment worries. [5]

Although most definitions do not use such generalized concepts, these generalizations are often implied as a base for other definitions. This makes the definition above highly applicable. As an addendum to the definition above, these key technical concepts are often associated with (but not required of) cloud computing: instantaneous and on-demand resource scalability, parallel and distributed computing, and virtualization [5].

UMM CSci Senior Seminar Conference, April 2011 Morris, MN.

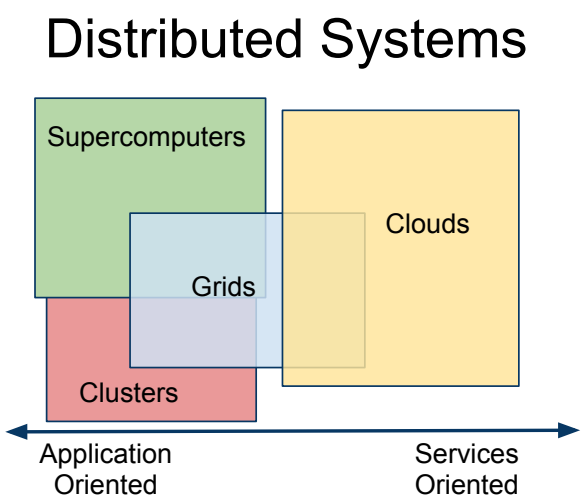

Figure 1: Representation of clusters and supercomputers (raw computing power and storage) in relation to grid and cloud infrastructures. Adapted from [3].

1.2 Cloud vs Grid

The driving concepts behind cloud computing are not new. Grid computing was introduced in the mid-90s with a similar vision of the more recent cloud: reduced costs, flexible and distributed computing power [5]. Grid computing still exists today and there is often confusion about the difference between cloud and grid computing.

Grid computing aims to "enable resource sharing and coordinated problem solving in dynamic, multi-institutional virtual organizations" [3]. The goal of grid computing is to combine distributed resources using common protocols to produce a large resource pool. Technical details are where cloud and grid computing differ. [3] argues that cloud computing is not a new technology; rather it is the evolution of grid computing and relies on grid computing as its backbone. Grid computing provides computational resources and storage while cloud computing, built upon grid technologies, utilizes virtualization and hardware sharing to deliver economical packages of abstract resources and services. Figure 1 expresses the relationship between grid and cloud computing alongside other types of distributed systems.

1.3 Uses of Cloud Computing

Various authors have proposed three different tiers of systems employed by cloud service providers [4, 5]. These tiers make up the different levels of technologies used in cloud computing.

1.3.1 Infrastructure as a Service (IaaS)

This level represents the most basic form of cloud service. Infrastructure providers (e.g., Microsoft, Google, Amazon) manage a vast set of computational and storage resources. Depending on the provider, end users may have direct access to the hardware resources or access to a set of virtual resources. Clouds typically utilize virtual resources and grid applications typically have direct access to hardware. Application and services built upon virtual resource sets are not hardware dependent and can be deployed seamlessly across different cloud platforms. This service is best represented by services like Amazon EC2, a virtual machine platform. Amazon provides hardware infrastructure for users to manage virtual machines, software implementation of normal operating systems, which may run any service or application the user desires.

1.3.2 Platform as a Service (PaaS)

At the next level services are presented to users as a software/application platform instead of hardware. Typically this layer consists of application frameworks that make up the basis of the SaaS layer described next. The Google App Engine and Microsoft Azure both offer a large set of programming tools at this level. The Google App Engine allows users to develop web applications that run on Google's infrastructure. The PaaS layer is encompassed within the SaaS layer as these programming frameworks are considered software as well.

1.3.3 Software as a Service (SaaS)

This is the highest level of services provided by cloud platforms. This level provides applications that end users interact with. Examples include Google Docs, Microsoft Office Live, Google Maps and Facebook [4].

2. VIRTUAL MACHINE SECURITY

2.1 Virtual Machine Monitoring

Cloud computing provides standardized resources by efficiently connecting and sharing large pools of hardware resources. Virtual machines operate on top of these resources and allow users to customize their environment. This allows the cloud to be a versatile resource and accommodate many different user interests.

The economical and performance benefits of cloud computing may be enticing but security is a major concern of businesses and organizations migrating to the cloud. Typically users upload code and data to a cloud virtual machine (VM). In this context, cloud computing offers IaaS; the user must obtain a VM image and configure it to his needs. The VM runs in a shared execution environment with other VMs. Other users may infiltrate the VM if it is not configured properly. Historically research has focused on securing the virtual machine itself through firewalls, user access restrictions, and other software provisions. While those methods are not obsolete, they do not provide the necessary level of security in a cloud environment because of a mismatch in security requirements and threat models [2]; more on this in Section 2.2. A more modern approach looks at building a virtualization-aware security mechanism.

There are two spots where security issues arise during software execution. First, users' code must be isolated to prevent unwanted intrusion. Second, the data being processed must be secured (more in Section 3). Virtualization is the common solution to the first issue. In the cloud, a hardware provider (e.g. Amazon) issues an end user a virtual machine. Since two parties are involved, the hardware provider also includes a virtual machine monitor (VMM) tool. Since the contents of the VM's operating system (OS) are unknown to the hardware provider, the VMM allows for detection of the OS and the monitoring of its operation. It also may attempt to detect and correct any system anomalies. Unfortunately this is not a perfect approach. The VMM assumes that the

OS has the original system files intact so it can identify the OS. Malware that modifies system files could trick the VMM to wrongly identify the OS.

Monitoring a VM "from the outside" is possible due to the nature of virtualization. But there are still challenges when monitoring VMs: they can be paused and restarted (and cloned) arbitrarily. This makes monitoring during the OS boot sequence unrealistic. Also, the source code of all operating systems is not publicly available making identification by code comparison difficult.

VMM methods typically employ secure introspection to validate the integrity of the guest OS on a VM. This process can be viewed as self-observation: the VMM tools monitor the virtual machine's memory state to identify the OS. Once the OS has been identified, security measures may be enforced based on common known weaknesses such as software and firewall limitations. Since secure introspection provides information about the OS code integrity by validating each memory fragment, this method also provides a secondary feature of detecting system intrusions.

2.2 Security Audit of Amazon Elastic Cloud Compute

Amazon provides IaaS via their Elastic Cloud Compute (EC2) service. EC2 provides powerful virtual machines for end users. Previously we mentioned that cloud computing suffers from new security holes in unexpected ways. In this section [1] explores the vulnerability of a multi-tier Amazon EC2 configuration.

2.2.1 *Background*

Amazon's EC2 allows users to configure and deploy virtual machines on Amazon servers. VMs are initialized from predefined images, which Amazon calls Amazon Machine Images (AMIs). Once they have been initialized they are referred to as VM instances. These instances are connected directly to the internet and can interact with each other. Amazon provides security groups that act as a firewall between the VM and the Internet, blocking undesired traffic. Each VM instance is a member of a security group that the user defines. It restricts inbound traffic using a set of user defined rules. Outbound traffic is not limited. Rules can allow traffic based on port, protocol (TCP), or source (IP address). TCP, or Transmission Control Protocol, is the core communication protocol behind the Internet Protocol (IP) and provides a communication layer for web applications and services. IP addresses are a unique numeric label assigned to each device on a network in the form *x.x.x.x* where *x* is a number between 0 and 255. Each time two computers establish a communication link, a port number associated with it in the range 1 - 65535. The port number depends on the application.

Members of the same security group can communicate if explicitly allowed to do so [1]. VMs and security groups are managed by a web-interface. Since the security groups are user-defined, there are frequent security concerns with the configuration. Security groups provide isolation and robustness to the overall system. However, they also leave the door open for potential vulnerabilities in a multi-tier system.

[1] uses reachability graphs to determine vulnerable areas of VM instances, see Figure 2. For example, if a VM is a member of the *web* security group the graph would consist of a source IP node, the associated security group *web*, and

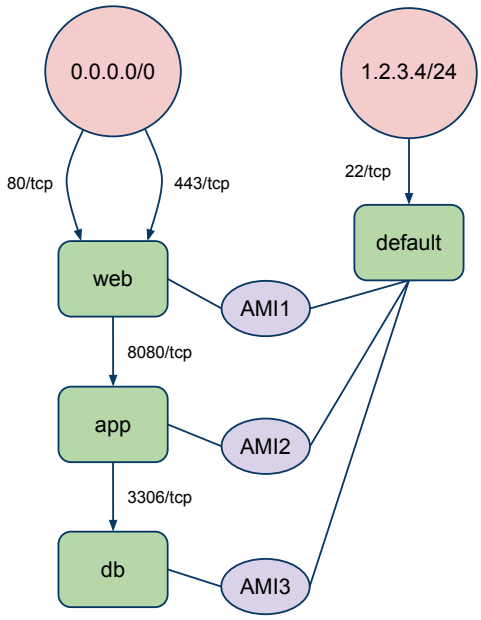

Figure 2: Relationship of Amazon Security Groups and AMIs. 0.0.0.0/0 represents any IP address as a source and 1.2.3.4/24 represents a computer connected to the same network as the web, application, and database servers (typically the internal corporate network). Adapted from [1].

the virtual machine that is accessible through the security group. Consider a widely used, multi-tier configuration of web servers, application servers, and database servers: the web servers are reachable on TCP port 80 (http) or TCP port 443 (https) from any source. The application servers are reachable on port 8080 but only accept requests from the web servers. The database servers are reachable on port 3306 but only accept requests from the application servers. We will also assume that all servers are internally accessible via SSH on TCP port 22 through the default security group. This multi-tiered architecture is highly scalable and robust and provides speed and security for web applications.

2.2.2 *Vulnerability of Cloud Multi-Tier Service*

Figure 3 presents a attack graph created in two steps [1]. The nodes in the graph represent a server and the edges (arrows) are directional communication links. The first step is to establish the relationship between AMIs and security groups. Amazon offers a command line API tool for this and much of this information is readily available in the web-based control panel as well. This creates the edges on the graph. The second step is to weight the edge vulnerability by running a program, Nessus, which is a popular network vulnerability scanner. With the AMI information and the security group it is a member of, Nessus is able to scan the AMI and evaluate its weaknesses. The range of *low, medium, high* has been extrapolated from the results to simplify the graph. In Figure 3 the edges are weighted based upon port weaknesses Nessus has discovered.

To prevent unwanted intrusions into web services three remedies are proposed: split up security groups, close unnecessary ports, and extract common ports. All three methods

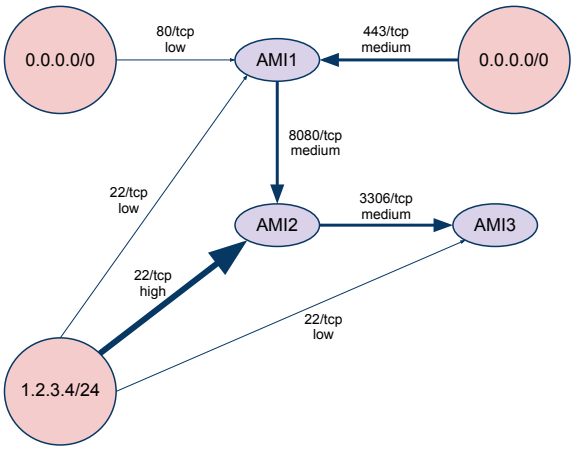

Figure 3: A typical multi-tiered web application environment. In the current configuration, an attacker in the corporate network (1.2.3.4/24) could compromise the application server (AMI2) via SSH. From there he also has access to the database server (AMI3). Adapted from [1].

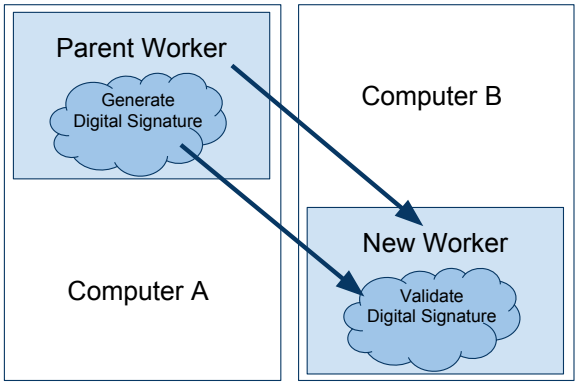

Figure 4: Representation of authentication between MapReduce workers. Parent workers generate a signature and create new workers. The new worker authenticate its workload by validating the signature was sent by a legitimate worker.

may be implemented together or individually. For example, isolating each AMI and putting it in its own security group would provide the highest level of security. However it may hinder future connection attempts because a needed port is closed. Closing unused ports in the security group is a simple and effective tactic.

Since an AMI can be a member of multiple security groups, extracting common ports among AMIs into their own group then adding AMIs to that group may be beneficial. This is similar to refactoring code but instead of code it is the accessibility of AMIs that is being refactored via Amazon security groups.

3. DATA CENTRIC SECURITY

A majority of work dedicated to cloud security focuses on the security of the underlying virtual machine and/or operating system encapsulating the services and applications. In this section the focus is on securing data transfer as it flows between cloud applications and services. As cloud applications become more decentralized and interdependent, as expressed with the multi-tier web application, potentially dishonest infrastructure and content providers (e.g. Amazon employees) have access to vast amounts of private user data.

The remedy is to secure the transfer of the data itself. Queries and data processing must be secured and participating parties should be authenticated. In [7] a framework is presented, Declarative Secure Distributed Systems (DS2), which aims to give developers tools to secure communication links. The DS2 framework allows developers to create applications and services that verify the source and authenticity of the information it sends and receives. It provides secure network protocols and security policies using *Secure Network Datalog* (SeNDlog) [6]. SeNDlog is a language based on Datalog, a query and rule language for deductive databases, which utilizes declarative networking techniques to maintain state across a network. Using these techniques, the authors of [7] propose using authentication via digital signatures which allow the receiver to verify the source of the request. Digital signatures are cryptographic schemes used to verify the authenticity of messages.

3.1 Where Data Security Matters

Consider online marketplaces such as Amazon, eBay, and Yahoo!. These services operate as online storefronts for merchants across the globe. The services collect product and inventory information from sellers and present that information to potential buyers in a unified storefront. Cloud computing offers economic incentives for both merchants and the marketplace portal applications to move to the cloud. The data exchange between merchants (product inventory and information) and the portal applications happens efficiently in the cloud. Cloud computing offers services for exchanging information between virtual nodes belonging to the same user but does not ensure secure query execution between different cloud users. The DS2 framework provides an authentication layer for query execution between cloud users. This ensures that merchants cannot tamper with each others' product inventory or hijack payments.

Another motivating example is data privatization in social networks. Most existing social network services (such as Facebook and LinkedIn) store user information and content on local databases. Due to the nature of frequently changing privacy policies [7] there may be desire to build a cloud-based social network that stores user content in the cloud instead of on social network servers. DS2 provides a secure authentication layer to allow the social network to access remote user data.

3.2 Secure Data Processing

[7] demonstrates a potential use of the DS2 framework with an authenticated implementation of a MapReduce algorithm which counts the words in a set of webpages. Typically a MapReduce algorithm consists of two steps: map and reduce. During the map step, a master node splits up the workload and hands sections to worker nodes. These workers may exist on parallel resources. The subworkers may also split their data and create another subset of workers. The process of splitting up the work may happen many

times. Once workers can no longer split data up, they process their workload. Then the MapReduce algorithm enters the reduce phase where the results of the finished workers are collected, combined and returned to the parent worker and eventually the master worker.

In typical MapReduce environments, workers may be created on unauthenticated resources. Since this implementation is cloud-based, new workers may be initialized in different locations (depending on what resources are available to the algorithm). Digital signatures are used in the following implementation to verify the authenticity of the workload. The reduce workers verify that a valid digital signature has been included by legitimate map worker when they process the data. Figure 4 represents a worker creating a new worker in a different location. The parent worker sends a digital signature to the child worker. If the child validates the digital signature, it may proceed with its workload.

3.2.1 Authenticated WordCount Results

The MapReduce WordCount algorithm demonstration was tested with three different authentication types: i) no authentication ii) RSA-1024 iii) SHA-1 HMAC.

Both RSA-1024 and SHA-1 HMAC are common methods of data encryption. RSA is a widely used algorithm for public-key encryption. It encrypts the entire workload before passing it to a new worker. The new worker must decrypt and validate the workload before processing. RSA-1024 provides digital signatures as well.

SHA-1 is a Secure Hash Algorithm developed by National Security Agency. It is a common cryptographic hash function that may encode or decode data (if the key is accessible to both sides). RSA-1024 encrypts the entire workload, while SHA-1 HMAC attaches a a hash to the message (which is visible) to verify its authenticity and integrity.

As shown in Figure 5, the word count iteration with no authentication finished in 350 seconds, RSA-1024 finished in 620 seconds and HMAC finished in 410 seconds. The time increase is overhead due to processing the signature generation and verification. Since RSA-1024 encrypts/decrypts the entire message and SHA-1 HMAC only attaches a digital signature, the 210 second discrepancy in encryption types is based mostly on processing overhead. The authors of the algorithm hypothesize that the authentication method could be further optimized to reduce the overhead in signature generation and verification.

3.3 Distributed Provenance

Provenance refers to the derivation history of a data product, including all the data sources, intermediate data, and the procedures that were applied to produce the data product [3]. This information is useful for debugging and error detection. The ability to backtrace applications on a network-level may yield the source of malicious activity or the cause of a bug.

In an environment where cloud applications are highly integrated and distributed, it is important for services to retain a log for accountability. If a merchant is trusted at time T and then commits a deceitful action at time X then all actions occurring between time T and X should be reevaluated and implications of those actions on other users should be reversed. DS2 provides a capture mechanism for identifying the source of a communication, the derived data and parties affected [7].

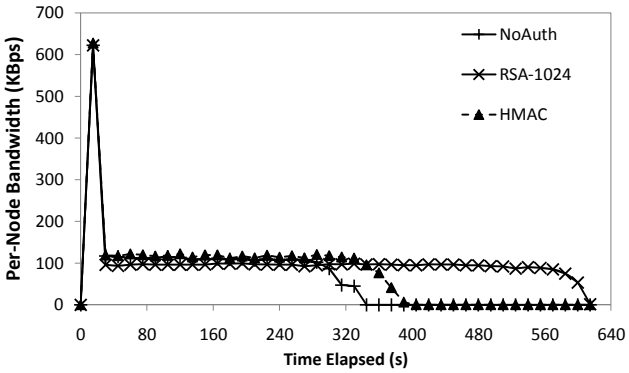

Figure 5: Per-node bandwidth (KBps) utilization. Completeness of algorithm is determined when bandwidth reaches 0 KBps as there is no further communication between nodes [7].

In the context of the cloud, provenance information is highly distributed. Distributed provenance data can be constructed and maintained over a distributed network automatically. Distributed provenance information may be stored as a directed graph representing the information workflow. It can also be stored in relational database tables. With this information, an automatic rewrite function can restore previous states. It is also possible to query the distributed provenance data and restore previous states given certain conditions [7]. To provide another layer of security, the distributed provenance data is often encrypted, and using self-detection methods mean that unauthorized tampering with the data is eventually noticed.

4. CONCLUSION

The use of cloud computing is growing everyday. The performance and economical gains when services and applications are outsourced to cloud infrastructure may be significant. When sensitive data leaves private systems and networks, more layers of security must be added. Virtual machines must be accessible by only a small amount of people, firewalls must be configured correctly, and applications must encrypt the data transfer. None of these are necessarily new concepts. But due to the expanding and evolving environment for cloud applications and services, extra security measures must be taken to ensure private data remains private.

5. ACKNOWLEDGMENTS

Thanks to Nic McPhee and Elena Machkasova for guidance and support throughout the development of this paper.

6. REFERENCES

[1] S. Bleikertz, M. Schunter, C. W. Probst, D. Pendarakis, and K. Eriksson. Security audits of multi-tier virtual infrastructures in public infrastructure clouds. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, CCSW '10, pages 93–102, New York, NY, USA, 2010. ACM.

[2] M. Christodorescu, R. Sailer, D. L. Schales, D. Sgandurra, and D. Zamboni. Cloud security is not

(just) virtualization security: a short paper. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pages 97–102, New York, NY, USA, 2009. ACM.

[3] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1 –10, 2008.

[4] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm. What's inside the cloud? an architectural map of the cloud landscape. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, CLOUD '09, pages 23–31, Washington, DC, USA, 2009. IEEE Computer Society.

[5] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39:50–55, December 2008.

[6] W. Zhou, Y. Mao, B. T. Loo, and M. Abadi. Unified declarative platform for secure netwoked information systems. In *Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on*, 29 2009.

[7] W. Zhou, M. Sherr, W. R. Marczak, Z. Zhang, T. Tao, B. T. Loo, and I. Lee. Towards a data-centric view of cloud security. In *Proceedings of the second international workshop on Cloud data management*, CloudDB '10, pages 25–32, New York, NY, USA, 2010. ACM.