Cyber Forensics in Cloud Computing

Arijit Paul[1*] Mayuri Kiran Anvekar[1**] Rishil Jacob[1***] K. Chandra Sekaran[1]

1. Department of Computer Science and Engineering, NITK, Surathkal, India

* Email: arijitpaul90@gmail.com

** Email: mayuri.anvekar@gmail.com

*** Email: rjpunk@gmail.com

Abstract

Cloud computing is a broad and diverse phenomenon; much of the growth represents a transfer of traditional IT services to a new cloud model. Cloud computing is anticipated to be one of the most transformative technologies in the history of computing. Cloud organizations, including the providers and customers of cloud services, have yet to establish a well-defined forensic capability. Without this they are unable to ensure the robustness and suitability of their services to support investigations of criminal activity. In this paper, we take the first steps towards defining the new area of cloud forensics, and analyze its challenges and opportunities.

Keywords: Cloud Computing, Software as a Service, Platform as a Service, Infrastructure as a Service, Signature-based Analysis, Behavior-based Analysis, Cloud Forensics.

1. Introduction to Cloud Computing Conceptual Model

Cloud computing model supports convenient, on-demand software access via network access to a shared group of configurable computing devices (e.g., servers, networks, applications, services and storage) that can be released and fast provisioned with less management work and service provider interaction.

There is no standard or single architectural method in cloud computing. This stands as the biggest challenge in this aspect. Therefore, it's best to view cloud architectures as a set of approaches, each with its own examples and capabilities.

A cloud computing system is a set of IT resources designed to be allocated ad-hoc to run applications, rather than be assigned a static set of applications as is the case in client/server computing. In a cloud computing environment, a user (via a virtual desktop, for example) requests information from an application. The cloud computing environment must then broker resources to run that application.

1.1 Virtualization

Virtualization is the key element in implementing cloud computing. Cloud Computing is defined as a pool of virtualized computer resources. Based on this Virtualization the Cloud Computing paradigm allows workloads to be deployed and scaled-out quickly through the rapid provisioning of virtual machines or physical machines. A Cloud Computing platform supports redundant, self-recovering, highly scalable programming models that allow workloads to recover from many inevitable hardware/software failures. A Cloud Computing platform is more than a collection of computer resources because it provides a mechanism to manage those resources. In a Cloud Computing platform software is migrating from the desktop into the "clouds" of the Internet, promising users anytime, anywhere access to their programs and data. The concept of cloud computing and how virtualization enables it offers many innovative opportunities to make the cloud environment more dynamic and versatile.

VMware solutions are engineered and integrated to equip the cloud with a unique combination of benefits. Virtualization is the essential catalyst for cloud computing. We can see that by the following process: Firstly the user requests an application resource in a symbolic form (via URL). Secondly the cloud computing environment fields the request and assigns resources to the task. Thirdly resources are loaded with the required software. Finally the address of the resources is returned to the user and the application interaction proceeds.

As this sequence shows, the most critical requirement for cloud computing is that users have a virtual view of their applications and should never refer to an application resource with a static address. Doing so would prevent the cloud from allocating resources flexibly. Since all cloud computing models must support a virtualized "front-end" interface to users, the management style of their virtual resources may be very different from one implementation to another.

1.2 Server-Virtualization

The cloud computing system is divided into two sections as shown in Figure 1: the front end and the back end. They connect to each other through a network, usually the Internet. The front end is the client side (or the user end). The back end is the "cloud" section of the system.

The front end includes the user’s computer and the application required to access the cloud computing system. All cloud computing systems do not have the same user interface. Services like Web-based e-mail programs leverage existing Web browsers like Internet Explorer or Firefox. Other systems have their own unique applications that provide network access to clients.

On the back end of the system are the various servers and data storage systems that create the "cloud" of computing services. Normally, every application will have its own dedicated server to execute or run its applications.

A central server administers the system, monitoring traffic and client demands to ensure everything runs efficiently and smoothly. It follows a set of rules called protocols and uses a special kind of software called middleware. Middleware allows the computer systems connected in the network to communicate with each other and exchange data. Usually, the servers do not run at full capacity which means that there is unused processing power going to waste. It is possible to overcome this problem by making a physical server act as if it is actually multiple servers, each running with its own independent operating system. The technique is called server virtualization. By maximizing the output of individual servers, server virtualization reduces the need for more physical machines. This helps in handling large amounts of loads which facilitates in scaling up and down of resources provided by the cloud.

If a cloud computing company has a lot of clients, there's likely to be a high demand for a lot of storage space. Some companies require hundreds of digital storage devices. Cloud computing systems need at least twice the number of storage devices it requires to keep all its clients' information stored. That's because these devices, like all computers, occasionally break down. A cloud computing system must make a copy of all its clients' information and store it on other devices. The copies enable the central server to access backup machines to retrieve data that otherwise would be unreachable.

2. Proposed Cloud Computing Service Architecture

Cloud computing delivers software, platform, and infrastructure as services, which are made available as services in a pay-per-use model to consumers. These services in industry are respectively referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These three fundamental classifications are often referred to as the “SPI Model”.

The diagram describes the cloud architecture based on the SaaS, PaaS and IaaS deployment models. Understanding the relationships and dependencies between Cloud Computing models is critical to understanding Cloud Computing security risks. IaaS is the foundation of all cloud services, with PaaS building upon IaaS, and SaaS in turn building upon PaaS as described in the Cloud Reference Model diagram. In this way, just as capabilities are inherited, so are information security issues and risk.

IaaS includes the entire infrastructure resource stack from the facilities to the hardware platforms that reside in them. It incorporates the capability to abstract resources, as well as deliver physical and logical connectivity to those resources. Ultimately, IaaS provides a set of APIs which allow management and other forms of interaction with the infrastructure by consumers.

PaaS sits atop IaaS and adds an additional layer of integration with application development frameworks, middleware capabilities, and functions such as database, messaging, and queuing, which allow developers to build applications upon to the platform; and whose programming languages and tools are supported by the stack.

SaaS in turn is built upon the underlying IaaS and PaaS stacks and provides a self-contained operating environment used to deliver the entire user experience including the content, its presentation, the applications, and management capabilities.

PaaS is similar to IaaS, except that the service includes a specific set of programming languages and tools (the platform). Generally aimed at the developer community, PaaS is analogous to an on-premises application server, only with elasticity and other cloud-computing features. Since platform software is fixed with the service, PaaS places restrictions on the applications that can be built.

SaaS is essentially the delivery of conventional IT applications to end users over the Internet. SaaS is analogous to a client/server model, except that the server is replaced by the SaaS provider's data center, the clients are web browsers on desktops, and the service offers cloud computing benefits such as elasticity and pay-as-you-consume metering. SaaS gained a foothold with universal applications such as email and Customer Relationship Management (CRM) applications like Salesforce.com.

3. Comparison of the Three Service Layers of Cloud Architecture

The three cloud computing service models can be viewed as a stack as depicted in Figure 3, with each layer increasing in specificity, while decreasing control of the underlying resources. The three layers sit above a virtualization layer, which itself sits above the physical servers, storages, and network hardware.

It should therefore be clear that there are significant trade-offs to each model in terms of integrated features, complexity and security. Trade-offs between the three cloud-deployment models includes:

- SaaS provides the most integrated functionality built directly into the offering, with the least consumer extensibility, and a relatively high level of integrated security.

- PaaS is intended to enable developers to build their own applications on top of the platform. As a result it tends to be more extensible than SaaS, at the expense of customer ready features. This tradeoff extends to security features and capabilities, where the built-in capabilities are less complete, but there is more flexibility to layer on additional security.

- IaaS provides few if any application-like features, but enormous extensibility. This generally means less integrated security capabilities and functionality beyond protecting the infrastructure itself. This model requires that operating systems, applications, and content be managed and secured by the cloud consumer.

The key point for security architecture is that the lower down the stack the cloud service provider stops, the more security capabilities and management consumers are responsible for implementing and managing themselves.

In the case of SaaS, this means that service levels, security, compliance, and expectations of the service and provider are predetermined, managed, and enforced. In the case of PaaS or IaaS it is the responsibility of the consumer's system administrators to effectively manage the same, with some compensation by the provider for securing the underlying platform and infrastructure components to ensure basic service availability and security.

4. Proposed Cloud Forensics Steps

The growth in networking connectivity, complexity and activity has been accompanied by an increase in the number of crimes committed within networks, forcing both enterprises and law enforcement to undertake highly specialized investigations. Forensic analysis, the methodical investigation of a crime scene, presents special difficulties in the virtual world. What is problematic for an investigator to do within a computer, making sense out of fragile digital data arranged in obscure and complex ways, can be very difficult within the significantly larger digital context of the network. Network forensics is the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents. It helps in identifying unauthorized access to computer system, and searches for evidence in case of such an occurrence. Network forensics is the ability to investigate, at a network level, things taking place or that have taken place across an IT system. The ultimate goal of network forensics is to provide sufficient evidence to allow the criminal perpetrator to be successfully prosecuted. The practical application of Network Forensics could be in areas such as hacking, fraud, insurance companies, data theft—industrial espionage, defamation, narcotics trafficking, credit card cloning, software piracy, electoral law, obscene publication, and discrimination.

One of the challenges in forensics is establishing of snapshots of the system in operation. But in this case one can question if this is good enough for such a "vast" and possibly globally distributed ecosystem. Let’s take the instance of malware injected into the kernel space of a system- it is possible that it may be programmed to modify data or functionality or both. We propose that since data may be present or available in a given configuration for a limited time or be staged through different levels of storage hierarchies, we can place bounds on events in question so as to be able to capture events of interest completely and hence it will be easier to keep track of events in the cloud. The steps to be followed in forensic investigation are depicted in the proposed design described in Figure – 4.

4.1 Collection and Storage of Data

Initially, to protect the data in the database, we would collect the data and store it in encrypted form. The data collection step collects all sorts of data like login, IP address etc. It mainly finishes collecting and storing network data. All the activities done in the network via a system can be stored in the database for future investigation. Forensics analysis can be initiated on a time-basis, i.e. after some time period. During this time, whatever changes have been made in the system can be collected and saved in the database, which will be taken into consideration while performing the analysis. A time-stamp can also be maintained to keep only recent modifications in the database, while removing the old ones to save memory. A checklist of malicious activities needs to be maintained as well, which will help us identify the restricted activities. This information needs to be monitored over for constant update due to the rapid growth of crime in the world.

To detect an intrusion, we need examine data describing the environment’s state. The event auditor can monitor the data that the analyzers are accessing.

The first component monitors message exchange between nodes. Although audit information about the communication between nodes is being captured, no network data is taken into account—only node information.

The second component monitors the middleware logging system. For each action occurring in a node, a log entry is created containing the action’s type (such as error, alert, or warning), the event that generated it, and the message. With this kind of data, it’s possible to identify an ongoing intrusion.

4.1.1 Signature-Based Method

Also called as Knowledge-based intrusion detection, is the most often applied technique in the field because it results in a low false-alarm rate and high positive rates, although it can’t detect unknown attack patterns. It uses rules (also called signatures) and monitors a stream of events to find malicious characteristics.

4.1.2. Behavior-Based Method

Numerous methods exist for behavior-based intrusion detection, such as data mining, artificial neural networks, and artificial immunological systems. We will use a feed-forward artificial neural network, because—in contrast to traditional methods—this type of network can quickly process information, has self-learning capabilities, and can tolerate small behavior deviations. These features help overcome some limitations which are there with traditional attacks.

Using this method, we need to recognize expected behavior (legitimate use) or a severe behavior deviation. Training plays a key role in the pattern recognition that feed-forward networks perform. The network must be correctly trained to efficiently detect intrusions. For a given intrusion sample set, the network learns to identify the intrusions using its retro propagation algorithm.

However, we focus on identifying user behavioral patterns and deviations from such patterns. With this strategy, we can cover a wider range of unknown attacks.

Behavior Analysis dictates how to compare recent user actions to the usual behavior. System needs to recognize expected behavior or deviation from regular behavior. With this strategy, we can cover a wider range of unknown attacks. This is performed on learned behavior that can't be modified without losing the previous learning.

Generating rules is the key element in this technique—it helps the expert system recognize newly discovered attacks. Creating a rule consists of defining the set of conditions that represent the attack.

4.2 Forensics Analysis Using Network Tools

The forensics analysis module includes working with various tools to gather information on the present status of the network. This information will be used to check for misbehavior later on. These tools allow us to know the current scenario like the activities carried out by a user at a particular IP address, or the sites accessed by him/her, or recovery of passwords using methods such as network packet sniffing, cracking various password hashes etc. This information collected is then either used directly for invasion detection or are saved in the database to be used for detection in future.

4.3 Invasion Detection

Invasion Detection module is the core of system, where the stored information is send for signature based analysis. In this method, all the data which has been collected so far is evaluated against the restricted checklist currently available in the database. This will allow us to detect whether an intrusion has occurred in the system. This is done by using pattern matching techniques where the occurrence of a sequence of tokens from the available checklist is matched with the inputted string (collected data). If match is successful, then alarm is generated and the detected network data is saved and logged in the result database, so that not only the original data is saved in the database, but also the data of invasion analysis.

4.4 Logging the Result in the Database

The result of the analysis is gathered together and depicted in a presentable and acceptable manner. Using the result obtained and conducting flow statistical analysis on data, invasion data restoration, and inquiry analysis, a detailed invasion report can be generated to be presented to the higher authorities. Some new virus, new invasion methods of hacker and tools which can't be detected during invasion detecting and module analysis can be explored and worked upon to be saved in the checklist.

References

http://www.cloudcomputingmodel.com/

http://www.vmware.com/solutions/cloud-computing/index.html

http://www.cloudcomputingarchitecture.net/

http://www.tatacommunications.com/downloads/whitepapers/Tata_Communications_IaaS_WhitePaper_v2.0-web

"Oracle Platform as a Service (PaaS) FAQ"; platform-as-a-service-faq-v4-444924

Platform-as-a-Service Private Cloud with Oracle Fusion Middleware, An Oracle White Paper, October 2009

Keyun Ruan, Prof. Joe Carthy, Prof. Tahar Kechadi, Mark Crosbie, Cloud forensics: An overview, Centre for Cybercrime Ivestigation, University College Dublin.

http://www.sei.cmu.edu/library/assets/presentations/Cloud%20Computing%20Architecture%20-%20Gerald%20Kaefer.pdf

Computer and Network Forensics Evidence Investigation Tools, http://www.edecision4u.com/
Network Forensics, http://en.wikipedia.org/wiki/Network_forensics
Resource Centre for Cyber Forensics India,
http://www.cyberforensics.in/(A(cos8NMWQywEkAAAAODMwODM4YWMtNWFmZC00ZWNhLThkNDEtNTlhMWM3MGE5MzA5hkCziwldj9ts_CCtkjYQI68akds1))/Research/NetworkForensics.aspx

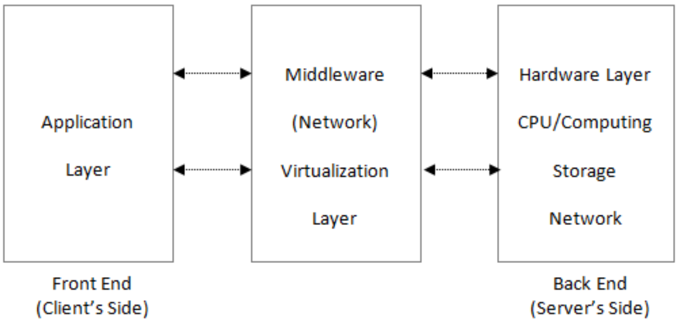

Figure 1. Software Server - Virtualization Layer

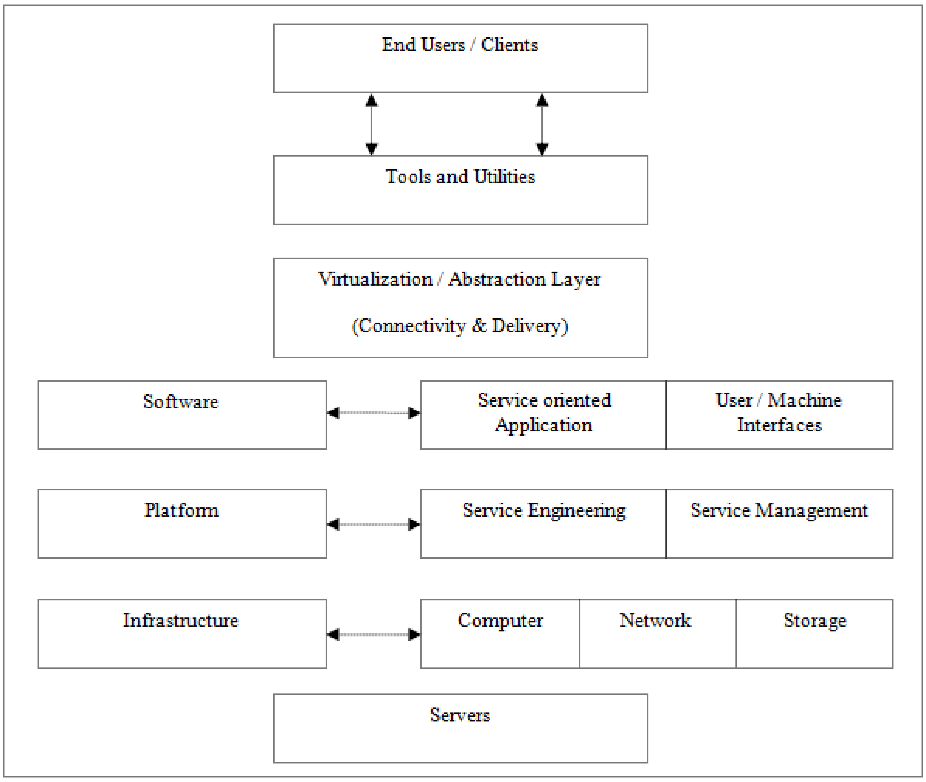

Figure 2. Proposed Cloud Architecture

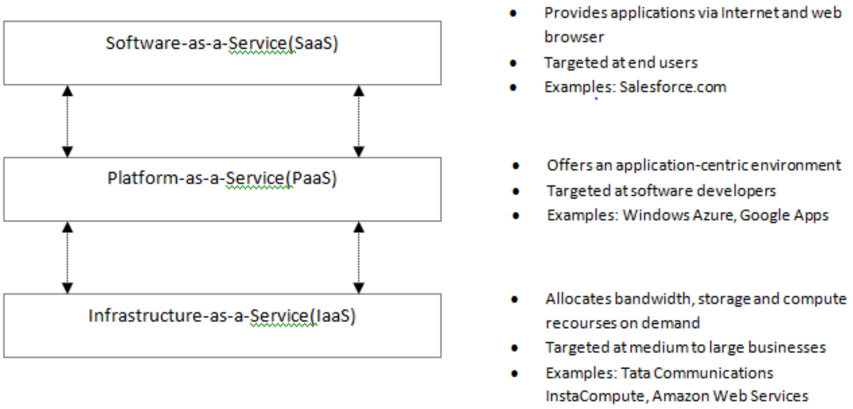
Software-as-a-Service(SaaS)
Platform-as-a-Service(PaaS)
Infrastructure-as-a-Service(IaaS)
Provides applications via Internet and web browser
Targeted at end users
Examples: Salesforce.com
Offers an application-centric environment
Targeted at software developers
Examples: Windows Azure, Google Apps
Allocates bandwidth, storage and compute recourses on demand
Targeted at medium to large businesses
Examples: Tata Communications InstaCompute, Amazon Web Services

Figure 3. The SPI Layers of Cloud Computing Service Model

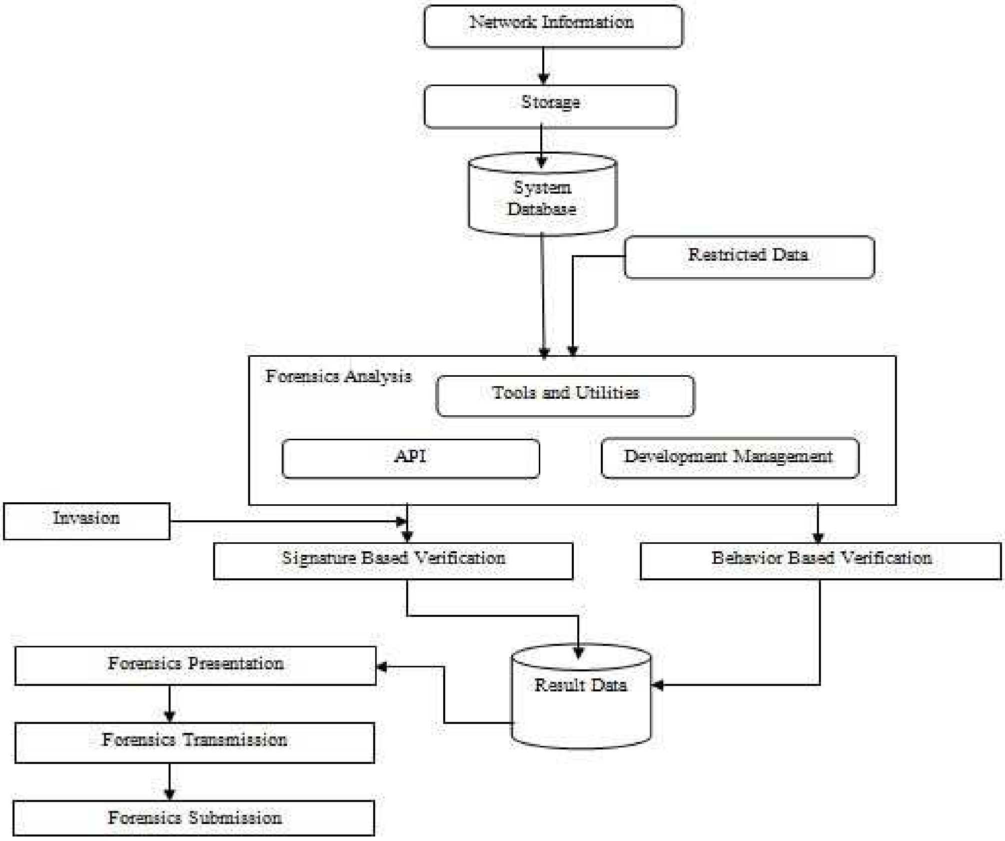

Figure 4. Proposed design and Implementation Steps