International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 6, Issue 7 (April 2013), PP. 23-27

Deploying File Based Security on Dynamic Honeypot Enabled Infrastructure as a Service Data Centre

Priyanka Paliwal
[1]Mtech student cse (Fourth sem), Government Engineering College , Ajmer,India.

Abstract:- This paper is about deploying Distributed Honeypot System that captures and analyze the attacks on different operating system on Eucalyptus Iaas cloud. The file based security is proposed and implemented on data to make data access secure on virtual machines and physical system in cloud. This makes the private cloud deployment secure enough due to three layers of security provided by virtualization, encryption mechanism and Honeypot supported physical node and prevents system exploitation.

Keywords:- Honeypot, Cloud, Iaas cloud, Virtualization, DHS

I. INTRODUCTION

A Honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. Honeypots is an easy target for the attackers can simulate many vulnerable hosts in the network and provide us with valuable information of attackers. Honeypots are not the solution to the network security, they are tools which are implemented for discovering unwanted activities on a network. They are not intrusion detectors, but they teach us how to improve our network security or more importantly, teach us what to look for. According to the definition we can note that honeypot is a system which is built and set up in order to be hacked. We categorize types of honeypots based on the level of interaction they offer to the attackers as:

1) Low-interaction honeypots
2) Medium-interaction honeypots
3) High-interaction honeypots

Low-interaction honeypots simulate only some parts of the system as the network stack. Low-interaction honeypots simulate only services that can't be used by an attacker to get full access to the honeypot and thus are not able to control the system. Use of these honeypots includes identification of port scans, generation of attack signature and malware collection. Popular examples of this kind of honeypots are Honeyd and Nepenthes, Specter, and KFSensor. They are also used for analyzing spammers or detecting worms [1].

A high-interaction honeypot is a conventional computer system or a fully functional Virtual Machine, a router or a switch. Here attackers can interact with a real system where almost nothing is restricted and thus, it is more risky compared to low-interaction honeypots. Therefore this kind of honeypot is normally placed behind a firewall to lessen the risk. They are not easily deployable compared to low-interaction honeypots, but by using them we learn more about the attackers behavior and find new vulnerabilities. Argos, Potemkin and Honeybow are some high interaction honeypots.

Honeypots are also classified as:

1) Physical honeypots
2) Virtual Honeypots

A physical honeypot is a real machine on the network with its own IP address. This is often high-interactive so allow the system to be compromised completely if compromised.

A virtual honeypot is simulated by another machine that responds to network traffic sent to the virtual honeypot and responds to the traffic sent to it. A virtual Honeypot may simulate a lot of different virtual honeypots at the same time.They include Honeyd and Kfsensor . Niels Provos created the original Honeyd (honeypot daemon) in 2002 as an open-source UNIX tool. Michael A. Davis, released a ported version of Honeyd in March 2003. Recently open-source Honeyd is become available for Windows [2].

Cloud computing is the access to computers and their functionality via the Internet or a local area network. Cloud users request this access from a set of web services that manage a pool of computing resources as machines, network, storage, operating systems and application programs. When granted, a fraction of the resources in the pool is dedicated to the requesting user until he or she releases them.

We cannot actually see or specify the physical location and organization of the equipment hosting the resources we are ultimately allowed to use. That is, the resources are drawn from a "cloud" of resources when they are granted to a user and returned to the cloud when released. Cloud are categorised into:-

Infrastructure as a Service (IaaS)

IaaS clouds provide access to collections of virtualized computer hardware resources, including machines, network, and storage. With IaaS, we can assemble our own virtual cluster on which we do installations, maintenance, and execution of own software stack. We build individual virtual machines as execution environments for tasks to access the required software stack without having to deal with Cloud site administrators.

Platform as a Service (PaaS)

PaaS clouds provide access to a programming or runtime environment with scalable compute and data structures embedded in it. With PaaS, we develop and execute our own applications within an environment offered by the service provider.

Software as a Service (SaaS)

SaaS clouds deliver access to collections of software application programs. SaaS providers offer us access to specific application programs controlled and executed on the provider's infrastructure. SaaS is also termed Software on Demand.

A virtual machine (VM) is a software implementation of a machine (i.e., a computer) that executes programs like a physical machine. Each VM includes its own kernel, operating system, supporting libraries and applications. A hypervisor provides a uniform abstraction of the underlying physical machine. Virtualization is the ability to run virtual machines on top of a hypervisor.Multiple VMs can execute simultaneously on a single hypervisor. The decoupling of the VM from the underlying physical hardware allows the same VM to be started on different physical machines. Thus virtualization is seen as an enabler for cloud computing, allowing the cloud computing provider the necessary flexibility to move and allocate the computing resources requested by the user wherever the physical resources are available.[3]

The use of virtual machines is beneficial for SaaS and IaaS providers and users by lowering the costs and improving utilization and management capabilities. Since virtual machines are cheap and easy to create, we tend to create distinct virtual machines for different tasks. We can branch new virtual machines based on old ones, snapshot machines or even rollback machines to a previous state. These features provide great flexibility to us.

II. INSTALLATION OF EUCALYPTUS IMAGE

Architecture of Eucalyptus 2.0.3

A starter Eucalyptus Machine Image (EMI) is a pre-configured operating system and virtual application software that is used to create an instance in either Amazon's EC2 environment or a Eucalyptus environment. EMIs are the basic building blocks for deploying services in a Eucalyptus/EC2 environment.Eucalyptus 2.0.3 system consists of 3 parts:

- CLC-Cloud Controller to manage the whole installation.
- CC-Cluster Controller which divides the installation into several clusters. Cluster is group of machines connected to the same LAN.We have installed cloud controller and cluster controller on the same host.
- Eucalyptus uses a storage service called Walrus which is an open software using the same public API as Amazon S3.
- NC-Node controller on each physical machine that runs VMs. Since we have used VMware which does not require NC to manage VMs. We only interact with cloud controller, which then sends commands and reads replies from the nodes. All VM images are initially stored in Walrus in compressed and encrypted form. Images can be either private or public. A Eucalyptus private cloud is deployed across an enterprise’s on premise data centre infrastructure and is accessed by users over enterprise intranet. Thus, sensitive data remains entirely secure from external intrusion behind the enterprise firewall.

Eucalyptus can be deployed on all major Linux OS distributions including Ubuntu, RHEL, Centos, and Debian.

A Eucalyptus compatible self service management portal can manage XEN, KVM, VMware virtual environments from a single pane of glass leading to much greater efficiency and cost reduction. We can manage multiple machine Image formats as we run multiple versions of Windows and Linux virtual machine images on IaaS clouds. When we create account using dashboard and install all components using command line interface we receive the available library of Eucalyptus Machine Images (EMIs) to run them on Eucalyptus clouds. Amazon Machine Images are also compatible with Eucalyptus clouds. VMware Images and vApps can be converted to run on Eucalyptus clouds, and Amazon AWS and Amazon compatible public clouds.

Command Line Interface

Eucalyptus supports two command line interfaces (CLIs). The administration CLI is installed when we install Eucalyptus server-side components. The administration CLI is for maintaining and modifying Eucalyptus. The other user CLI, called Euca2ools, are downloaded and installed on clients. Euca2ools are for end users and can be used with both Eucalyptus and Amazon Web Services (AWS).The Eucalyptus Dashboard provides cloud administrators with a way to perform several management tasks in a webuser interface.

Credentials

Each user has a unique set of credentials. These credentials are used to authenticate access to resources. There are three types of credentials:
An X.509 certificate is used to authenticate requests to the SOAP API service.
A secret access key is used to authenticate requests to the REST API service.
A login password is used to authenticate the Eucalyptus Dashboard access.
We can manage credentials using the command line tools euare- commands or the Dashboard.
We can download the full set of credentials for a user or an account, including X509 certificate and secret access key, by: /usr/sbin/euca_conf -- get-credentials or euca-get-credentials[4]

The following are EMIs provided by Eucalyptus to help getting started with using Eucalyptus Fig1. We have versions of each EMI - "default" which has a 1.3 Gig root file system, and "large" which has a 4.5 Gig root file system. When using the large image type the correct vm type must be used when launching an instance using the image**.**

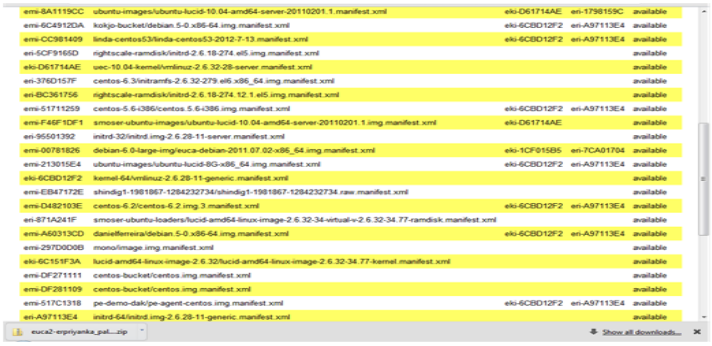

Fig.1: EMIs provided by Eucalyptus

Additionally, each image has a custom startup script that allows the utilization of bash scripts to be passed by the "user-data" option with the euca-run-instance command and executed on instance startup. Once we have downloaded an image, we can bundle, upload and register it for use in your Eucalyptus cloud as in Fig 2.

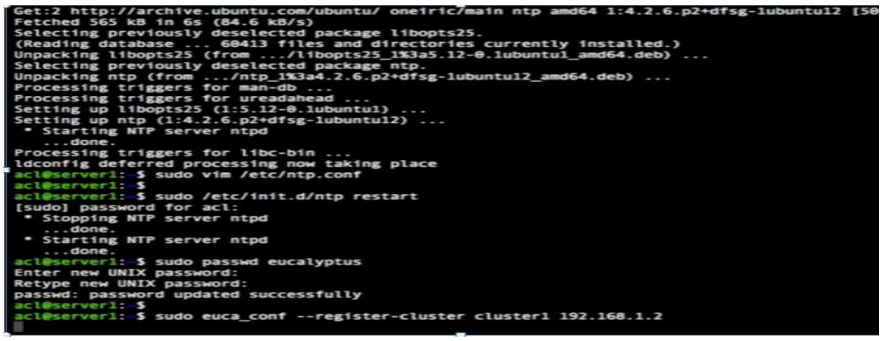

Fig.2: - Eucalyptus image configuration screenshot

III. DEP LOYMENT OF DISTRIBUTED HONEYPOT SYSTEM ON EUCALYPTUS IMAGE

Honeypot generates alerts by email or web based alerts to send notification about traffic going to or from the honeypot to the administrator to review intruder activity. It's Logging unit provides efficient storage for all the firewall and system logs and of the traffic going between the firewall and the honeypot system.

A distributed honeynet system is a collection of honeypot systems which are deployed at different geographical locations and communicates to a central server for analysis of captured data and logs. Here we have chosen to structure DHS on three tier architecture to fully decouple the data captured from data storage. This has twofold advantages as it is easier to build and scale the whole system and if a honeypot sensor is compromised there is no way it can harm the data store.

The first stage is the actual honeypot which is implemented using DCH framework and is a combination of low interaction and high interaction sensors. Malware samples acquired from this setup are sent to the central server which validates and pre processes them and then the samples are sent to the database for storage and analysis. This architecture is fully scalable adding more sensors is only matter of creating new virtual machines or adding other physical machines.

The main features of DCH are:

DCH system supports Dynamically Configurable Honeynet framework as remote node and follows client server based architecture. The use of SSl/Tls layer ensures data authenticity, integrity and confidentiality. In built automated data collection mechanism .It also has separate web based console for captured data display, administrative user right assignments, remote node management and monitoring.

DHS components consists of:-

1) DCH Server
2) DCH GUI
3) DCH Database
4) DCH Node

DCH Server Communicates with DCH Client through secure channel also register the new client requests. Sends configuration files to clients. Collects data from different nodes and reflect to Database . Sends virtual image to different remote nodes.

DCH GUI: Using this web based application user can configure and reconfigure the remote nodes also can add remove and edit the existing services, contents and operating system options.It also has provision of creating and managing users. With the display user can see the binary, Pcap data coming from different nodes in graphical form. User can see the daily ,monthly ,yearly distribution of data on each node, on each honeypot, and user can check profile wise distribution of data .user can check number of files created, edited deleted on each honeypot and also the details of them.

DCH Node: DCH Client is a remote node which is used to deploy honeynet system. It communicates with the DCH server and register itself to DCH server, Receive configuration file from DCH server, configure the honeypots, collects the incoming data on honeypots and honeywall and sends the collected data to DCH server .Requirements for DCH server is high speed processor with redhat enterprise linux 5 and DCH code with Public IP and Mysql database for DCH nodes.[2]

For viewing the content of each node details Fig.3 click on node details and then select node for which want to view content of selected node and click on submit button which displays content (honeypots, services, contents) of data.

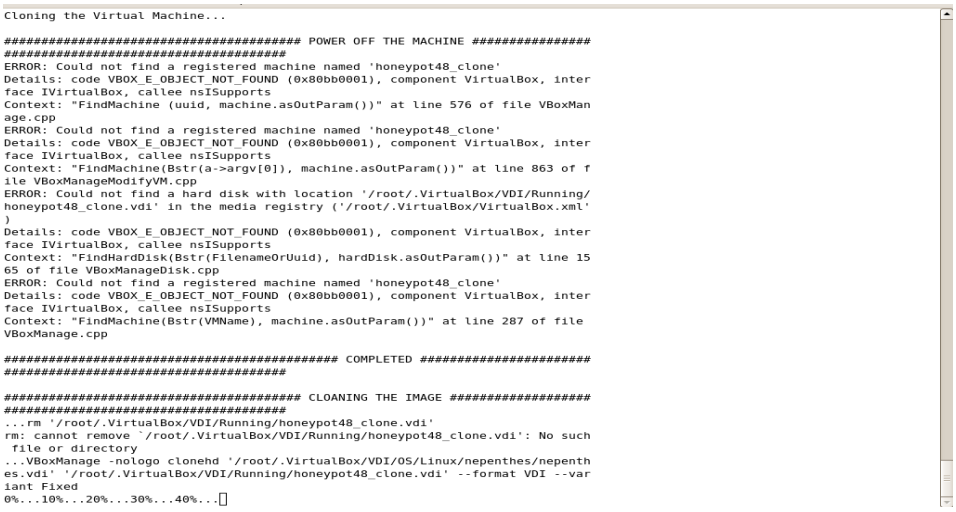

Fig.3: - DHS Node content

Traffic going to and from DCH provided us with a roughly 1,50,000 KB PCap file recorded in a week's time Fig. 4. This record has all our intruder's moves, from the time of the intrusion to the use of our node. We found that scans that occurred across our system on the Iaas data are most likely worm type.

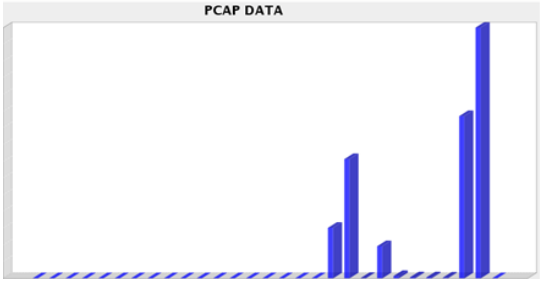

Fig.4: -Pcap binaries obtained based on date(x) 02/01/2013 08/01/2013 and size in kb (0 to 1,50,000) (y)

IV PROPOSED ENCRYPTION ALGORITHM AND RESULTS

Here the proposed encryption and decryption algorithm is File Level symmetric key Encryption In Cloud is an encryption system in which encrypted files are encrypted by features of the file system itself. With the use of this File Level Encryption some advantages are more easy way to granular control over the information that is encrypted easily and stored on the same location where the original file is stored.

It allows to integrate access level restriction in many file formats. Allows to mange when data is encrypted on file level and moved off the storage location it can then be decrypted only by the same password used for encryption. Confidentiality, is maintained. The table below Table1 shows the file types, encryption and decryption time taken with the proposed algorithm hence forth

Table I: Data from algorithm implementation

File Name	File Size	*Encryption Time*	*Decryption Time*
Enigma.flv	60928kb	1342 ms	703ms
Tinkerbell.jpg	132 kb	16ms	0 ms
Water.pdf	458 kb	15ms	15ms
Filesize-encryptiontime.fig	3kb	0ms	0ms
Guidelines_MTech_Disseratation.pdf	213kb	62ms	15ms
National_Finals.xls	27kb	15ms	16ms
02-mediaformats.ppt	933kb	31ms	15ms
Bear.wmv	3951kb	109ms	78ms
Final.rar	345kb	16ms	15ms
Despertar.wma	6069kb	156ms	109ms
AartiShriGanesh.flv	4778kb	172ms	78ms

The next figure Fig. 5 shows the Matlab simulation of the proposed algorithm

Fig.5: - Matlab simulation

V CONCLUSIONS

We installed on Eucalyptus Iaas cloud cluster and Distributed honeynet system and captured data about the exploits on the system. The proposed encryption algorithm is applied on files in cloud, using symmetric key cryptography concept. The data captured is analysed daily so that attacks are found and further access by the intruder be prevented. The levels of security provided by using Eucalyptus cloud and virtual machines along with deployment of honeynet along with the encrypted files using the proposed algorithm makes the data unassailable. Such a system works as intelligent system for network protection.

REFERENCES

[1]. http://www.projecthoneypot.org
[2]. www.honeynetindia.org
[3]. http://www.eucalyptus.com
[4]. http://www.eucalyptus.com/docs/3.2/ig/installing_euca2ools_standalone.html