ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

Enhancing Data Security and Privacy on WebOS with Desktop-as-a-Service

Gurjot Singh Sodhi
Department of Computer Engineering
Punjab Technical University, Jallandhar
India
er.gurjotsinghsodhi@gmail.com

Gurjit Singh Bhathal
Department of Computer Engineering
Punjabi University, Patiala
India
gurjit.bhathal@gmail.com

Abstract - Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. There are a number of online operating systems available today, and the number is steadily growing. EyeOS is a web-based open source platform designed to hold a wide variety of web applications over it. It looks like a regular PC operating system, but the "trick" is that it can be accessed from anywhere.The major issues in cloud computing is the security of data that is being stored on the provider's cloud and privacy while the data is being transmitted. This paper deals with the major security concerns that needs to be taken into consideration on WebOS while ensuring that unauthorized intruder can't access your file or data on WebOS and giving some solutions.

Keywords - API – Application Programming Interface, CSP – Cloud Service Provider, Cloud Customer or Client, VM – Virtual Machine, SaaS , PaaS , IaaS, Virtualization

I. INTRODUCTION

Service providers build their own datacentres to serve their customers. A data centre is a place where companies keep their physical hardware. It is usually composed of many servers, routers and switches interconnected together. Usually datacentre requires a lot of space, consuming resources and energy. Redundant links exist between those allowing the companies to use their services via the Internet or private IP-based networks at anytime and anywhere. Thus customers do not need to know where their data are being kept. However adding more and more equipment to their rack system saturates the data centre and consumes energy. Looking ahead there will be an increase in the management costs and resource utilization in the near future. To overcome those limitations, the virtualization technology was introduced.

A. What is Cloud Computing?

Cloud computing provides a model for enabling on-demand network access to a shared pool of computing resources (for example: networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or cloud provider interaction.[2]

Fig. 1 Cloud Computing [3]

B. The Lifecycle of Data

The typical lifecycle of data can be describe as the following stages:

Stage 1: Data Creation/Transmission -- this is the initial stage, data is created by the user and then pushed to the Cloud for consumption.

Stage 2: Data Reception -- data is received in the Cloud before being written to storage and logs taken of activity.

Stage 3: Output Preparation -- data is prepared to be returned to the consumer, this involves any transformations that needs to be performed on the data prior to its return i.e. serialization.

Stage 4: Data Retrieval -- data is received by the consumer from the cloud and has now within the domain of the user.

Stage 5: Data Backup -- the CSP will replicate data for archival purposes. This may involve the transferal of a copy of the data to an external store.

Stage 6: Data Deletion -- data is permanently deleted from the cloud.

Stage 7: Data Migration -- data is migrated from a resource inside the cloud to another for availability or scalability purposes.

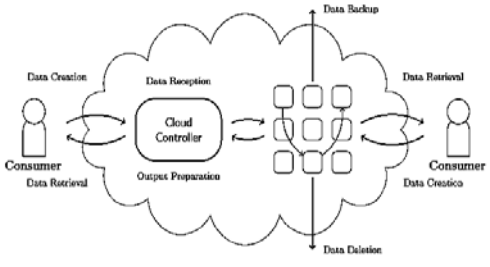

Fig. 2 Data Lifecycle Threat Model for Cloud

ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

C. Web based Operating Systems(WebOS)

A web desktop or WebTop is a network application system for integrating web applications into a web based work space. It is a virtual desktop on the web, running in a web browser as software. Web desktops often are characterized by an environment similar to that of Windows, Mac, or Linux, but are now considered to have much more functionality being dependent on the internet. Typical benefits include the ability to save work and settings over the internet rather than to the local desktop.

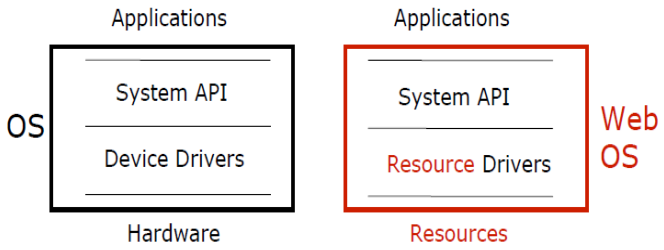

Fig. 3 Drivers in WebOS and Standard OS

II. SECURITY CONSIDERATIONS

Security is a principal concern when entrusting an organization's critical information to geographically dispersed cloud platforms not under the direct control of that organization. In addition to the conventional IT information system security procedures, designing security into cloud software during the software development life cycle can greatly reduce the cloud attack surface.

The security posture of a cloud system is based on its security architecture. While there is no standard definition for security architecture, the Open Security Alliance (OSA) defines security architecture as "the design artifacts that describe how the security controls (= security counter measures) are positioned, and how they relate to the overall IT Architecture.

Secure cloud communications involves the structures, transmission methods, transport formats, and security measures that provide confidentiality, integrity, availability, and authentication for transmissions over private and public communications networks.

A. Service Level Agreements (SLAs)

Typically, cloud-hosting agreements are concerned with "up-time" and high availability, with little or no mention or assurance of security. However, the client is ultimately responsible for ensuring the service they're using meets their security requirements and compliance obligations. Thus, SLAs and other written agreements between CSP and the Client should clearly identify their responsibilities. Failure to develop and agree upon appropriate SLAs may result in issues for the client if the cloud service does not meet the needs and demands of their business.

B. Data Acquisition

The client will ultimately determine how and when the cardholder data is acquired in the cloud environment. End-to-end processes and data flows must be documented across both client and cloud provider networks, so that it is clearly understood where cardholder data is located and how it is traversing the infrastructure.

C. Data Life-Cycle

For all Cloud models, it must be made a compulsion for CSP to provide clear requirements for Data retention, storage and disposal, in order, to ensure that sensitive data is-

- Retained for as per SLAs.
- Stored in Secured location.
- Accessible only to Authorized user.

D. Data Integrity

When a data is on a cloud anyone from any location can access those data's from the cloud. Cloud does not differentiate between a sensitive data from a common data thus enabling anyone to access those sensitive data's. Thus there is a lack of data integrity in cloud computing.

E. Data Encryption and Key Management[4]

In a public-cloud environment, one client's data is typically stored with data belonging to multiple other clients. This makes a public cloud an attractive target for attackers. Strong data-level encryption should be enforced on all sensitive or potentially sensitive data stored in a public cloud. At a minimum, key-management servers should be located in a separate network segment and protected with separate access credentials from the VMs that are using the keys and the data encrypted with them.

F. Identity and Access Management

Individual user identification and authentication for both CSP and client personnel is essential for access control and accountability.

- Shared credentials (such as user accounts and passwords) should not be used in the CSP environment.
- Client accounts and passwords should be unique for each service

III. EXPERIMENTATION AND RESULTS

The implementation of Security in a Cloud Environment requires specialized technical knowledge and skills. So, a secured web based operating system is being created keeping in view the security concerns. Here the web based operating system we are discussing is EyeOS which is being modified.

A. Requirements for Web based operating system

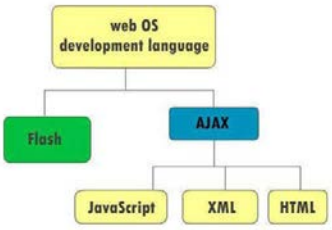

Fig. 4 EyeOS Programming Language Requirements

ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

EyeOS is a platform for web applications, created with the idea to make easy the application development. EyeOS is built on PHP and uses a combination of web standard technologies such as HTML, JavaScript and XML to create the UI. This means that it will run on any standards compliant browser. By itself, it doesn't require any extra plug-in in order to work. All it needs is a PHP 5 capable server, which is the common denominator for just about any hosting package – it doesn't even require a database.

B. Global File System

The Web File System architecture has two parts, a user-level daemon and a loadable vnode module. Consider the example of a read to a file in the WebFS namespace. Initially (step 0), the user-level daemon spawns a thread which makes a special WebFS system call intercepted by the WebFS vnode layer. The layer then puts that process to sleep until work becomes available for it. When an application makes the read system call requesting a WebFS file, the operating system translates the call into a vnode read operation (step 1). The vnode operation (step 2) checks to see if the read can be satisfied in the kernel (i.e., if the page being requested is cached in the kernel). If the operation cannot be satisfied in the kernel, the vnode operation calls in a structure requesting a read of the file page in question and wakes up one of the sleeping threads in the work queue(step3). The user level daemon is then responsible for retrieving the page, by contacting a remote HTTP or WebFS daemon (step4).

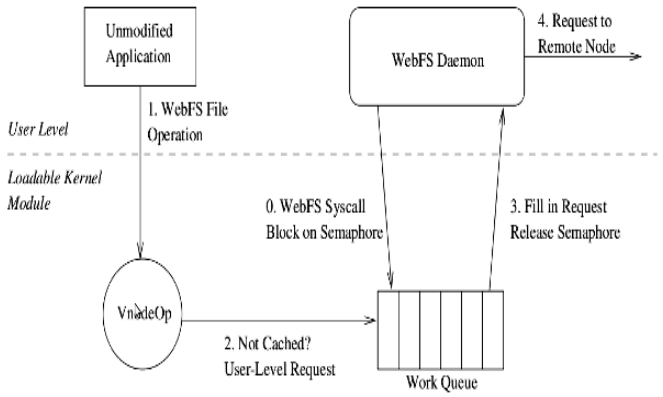

Fig. 5 EyeOS Global File System

C. Security Analyzer

All components of EyeOS that conduct operations on data or system information or access resources check the privileges of the user who executed the process before any operation takes place. The privileges are checked against a set of rules, ACL style, which determine whether the user can perform that action or not. The EyeOS component responsible for conducting these checks is the Security Analyzer. The rules of the Security Analyzer are in XML format.

These are a set of Policies and within are a set of handlers. A policy is a rule applied to an object type, i.e. a class. Normally a developer uses classes to access system resources, and the security manager is used to specify rules about these classes - for example the objects that represent a file, or the objects that represent a process. The best way to understand a policy is to first study what appears in file...

<policy objectClass="EyeUserFile">

This policy applies when somebody tries to manipulate an object of the type eyeUserFile.

This object is used when accessing a file belonging to a user through FSI::getFile. It is important to realise that handlers are processed from top to bottom, starting with the first then executing though to the last. The 'flag' parameter of each handler specifies whether to continue evaluating handlers or not. The first handler (in the policy shown above) is...

<handler class="SystemGroupSecurityHandler"flag= "sufficient">
<param name="groups" value="admin" />
</handler>

This handler specifies that if the user in the group 'admin' (note the <param> there within the handler), then this is already 'Sufficient' and it is no longer necessary check anything else, the user can access the file. The next handler of this policy is...

<handler class="SystemGroupSecurityHandler" flag="requisite">
<param name="groups" value="vfs" />
</handler>

D. General File and Directory Permissions

The server file system may provide some level of protection from abuse by non-authorized users. The EyeOS system requires write access to certain directories and files, on some systems that access may need to be provide outside the EyeOS system.

1. Using unique ID for each file instead of its real name (for security reasons).

Other than the etc/, usr/ and etc/ directories (and the usr_apps/ directory if enabled) EyeOS does not require write access to the other files and directories which may be write disabled for an additional level of security. Note, however, that those files and directories will need to be write re-enabled in order to perform system updates (for future).

2. Added remove protection in public notes: Can be removed only by the user who created the document and ROOTUSR.

3. Added protection to avoid existent files in Publicdir to being overwritten.

E. File Browsing Protection

The entire EyeOS files system for all users is by necessity a part of the web accessible file system. This means that a user may gain access to any file by the fact of knowing its name and location and typing in that URL. Most EyeOS configuration information is stored in .xml format which is readily browsable. In order to hide the configuration data, which may include encrypted passwords, you can change various system parameters in the installation advanced security parameters. A suitable change would be to change all .xml files to .php which cannot be browsed from a simple URL.

ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

F. Application Loading

The default EyeOS configuration allows any user to upload into the EyeOS space applications of the appropriate format. This is clearly a potential security vulnerability. It is relatively easy to write an EyeOS application which will allow a user to browse the file system. An option in the sysdefs.php file allows two levels of restriction on application loading. The first is to limit application uploading to only the root user, the second disables all application uploads. Added security can be provided by removing the eyeApps.eyeapp application in the /apps directory if its other features are not necessary.

Files like system/archive.php, system/stats.php, system/syscall.php, system/funcions.php,login/index.php ,login/browlang.php does not allowing the files from being executed from outside the system

The root user may limit any particular user as to the applications they are allowed to run by editing the usrinfo.xml file in that users /usr directory. The apps element is a comma separated list of applications which will appear on the users desktop icon bar. If the eyeApps. eyeapp app is not listed than that user will be unable to change that list of apps.

G. User Accounts

Normally, the user can register him/her by visiting the registration page. At server side, a validation link is being automatically generated for the email-id used by the user. In EyeOS, we had restricted the users to create User Accounts of their own. The user can only request the administrator by sending an email. The administrator within 7 days will validate the user. The reason behind is to restrict fake/invalid email-id's validation.

IV. CONCLUSIONS

Cloud Computing embodies the as-a-Service paradigm and allows for services to be provided enmasses to consumers. The problems associated with the use of cloud based services can be summarized by the unknown risk profile and unknown expectation of privacy. When service users push data to the cloud they need to rely upon Cloud Service Providers (CSPs) adhering to their remit, and doing so dutifully. However, when looking to build solutions to protect data in the cloud it is important to remember that for the service user the CSP can be trusted, albeit at arm's length.

The following conclusions are drawn on the basis of experimental observations and analysis:

1. Consumers and businesses to use applications without installation and access their personal files at any computer with internet access. Users can access applications from anywhere and anytime .
2. It could bring hardware cost down. No need to buy a set of software or software license for every employee. Save money on it support.
3. Speed up the calculations and processes.
4. With a private server, EyeOS can provide city councils, public library networks, free Internet points and other public environments the perfect system for their users to have a web place to work and communicate with the network managers, registering once and using it from every point.
5. One of the main triggers and great acceptance of people with regard to this service is its availability online, which has no dependencies and has a strong security system, achieving thus be an ideal application for storing content.

V. FUTURE SCOPE

BY 2020, most people won't do their work with software running on a general-purpose PC. Instead, they will work on Internet-based applications such as Google Docs, and in applications run from smart phones. Aspiring application developers will develop for Smartphone vendors and companies that provide Internet-based applications, because most innovative work will be done in that domain, instead of designing applications that run on a PC operating system.

1. Possible lines of research are the development of reliable and efficient virtual network securities to monitor the communications between virtual machines in the same physical host. To achieve secure virtualized environments, isolation between the different tenants is needed.
2. Another option is to go for RADIUS/LDAP implementation allowing 3rd party User Authentication.

VI. IMPACT ON ENVIRONMENT

With EyeOS, there will be a fall in demand for individual desktops/laptops. With EyeOS, users can save their data without worrying about their privacy. Moreover, it will reduce the number of servers in datacentres, thus decreasing electricity consumption and consequently the necessity of high-end cooling equipment's. In other words, we can say that EyeOS leads us to Green Computing. However, some initial investment in software's and hardware devices need to be made, but this is usually modest as compared to the savings we achieve in long run.

REFRENCES

[1] NIST Guidelines on Security and Privacy in Public Cloud Computing (SP SP800-144).
[2] The NIST Definition of Cloud Computing (SP 800-145).
[3] Tejas P. Bhatt, Ashish Maheta, "Security in Cloud Computing using File Encryption", International Journal of Engineering Research and Technology (IJERT), Vol. 1 Issue 9, November 2012.
[4] PCI Data Security Standard v2.0 Guidelines, February 2013.
[5] Assessing Cloud Node Security Whitepaper, March 2011.

