ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 6, June 2014

An Analysis on the Possibilities of Covert Transfers between Virtual Machines Clustered In Cloud Computing: Survey

Wilson Bakasa[1], Kudakwashe Zvarevashe[2], Nicholas N. Karekwaivanane[3]

M Tech Student, Dept of CSE, Jawaharlal Nehru Technological University, Hyderabad, India[1]

M Tech Student, Dept of CSE, Jawaharlal Nehru Technological University, Hyderabad, India [2]

M Tech Student, Dept of CSE, Jawaharlal Nehru Technological University, Hyderabad, India [3]

ABSTRACT: Cloud Computing has emerged to be one of the fast growing technology with a lot of devices being part of the cloud. A lot of applications have been deployed into the cloud for use by different users ranging from business applications, gaming applications, educational applications, health applications and other scientific applications. As different hardware, software and services are being deployed to provide cloud computing also a great number of vulnerabilities, attacks and threats arise that becomes an issue to the integrity, confidentiality and availability of cloud computing. The paper Covert Transfer between Virtual Machines in Cloud Computing looks at how covert transfers may compromise the security measure placed for cloud computing. Covert Transfers cause information leaks that affect the integrity, availability and confidentiality of cloud computing resources. Although achieving 100% security is unrealistic we aim to propose countermeasures that would give a favorable cost/benefit analysis for the consumers and providers of the services. We shall look into some of the issues that may cause the covert transfers.

KEYWORDS: cloud computing, covert transfers, virtual machines and security countermeasures.

I. INTRODUCTION

Cloud computing is an emerging technology with shared resources and lower cost that relies on pay per use according to the user demand. Due to its characteristics, it may face lots of threats and problems in the scopes of security. In this paper the issue of covert transfers between virtual machines in a cloud [1] is explained and discussed. There are many scholarly researches, articles and periodicals on cloud computing security concern out there. Security researchers and professionals are working on security vulnerabilities, attacks, potential threats and possible countermeasure in cloud computing constantly [2] [3] [4].

II. RELATED WORK

In [1] Hypervisor's primary shortfall is that it does not completely protect against unauthorized transfer of information between two virtual machines that are not allowed to share information. Cloud computing servers use the same OS, enterprise and web applications as localized VMs and physical servers. The ability for an attacker or malware to remotely exploit vulnerabilities in these systems and applications is a significant threat to virtualized cloud computing environments. In addition, co-location of multiple VMs increases the attack surface and risk of VM-to-VM compromise [7].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 6, June 2014

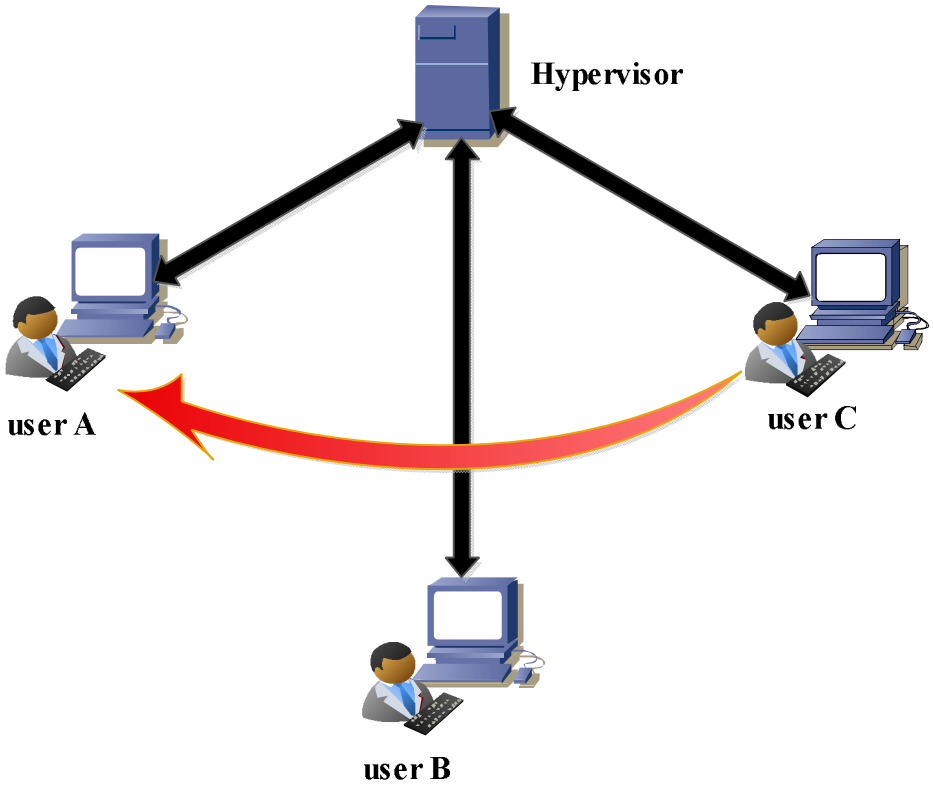

Figure 1: Covert Transfer between VM user A and VM user C through VM user B

Figure 1 shows an example of covert transfers that can result as a result of some of the issues to be discussed. VM user C transfers information to VM user A covertly. This is mainly as a result of settings that allows the sharing of information between the pair user A and B, or user B and C but not between pair user A and C. These can be compromised resulting in the leaking of information between user A and user C [1] [3].

Covert Transfer is a vulnerability that enables a guest-level VM to attack its host. Under this vulnerability an attacker runs code on a VM that allows an OS running within it to break out and interact directly with the hypervisor. It allows the attacker to access the host OS and all other VMs running on that particular host [5] [6] [8].

III. SCOPE OF RESEARCH

The paper will look at some cause of covert transfers and suggest some countermeasures to restrict leak of information due to covert transfers. The main cause of covert transfer comes into play when virtual machines are migrated from one cluster to another. As shown in figure 2 the migration of virtual machine user F from cluster B to cluster A may result into a number of security issues if not checked.

In cloud computing virtual machines are migrated [5] from one host to another host. This can be in the same cluster or different clusters. In the paper we give an example of migration between clusters which result in covert transfers being formed. Virtual machine user F has formed a covert transfer in Cluster A which will cause leak of information to attackers. This will compromise the security of the whole cluster. So the paper looks into these issues and how they can be overcome.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 6, June 2014

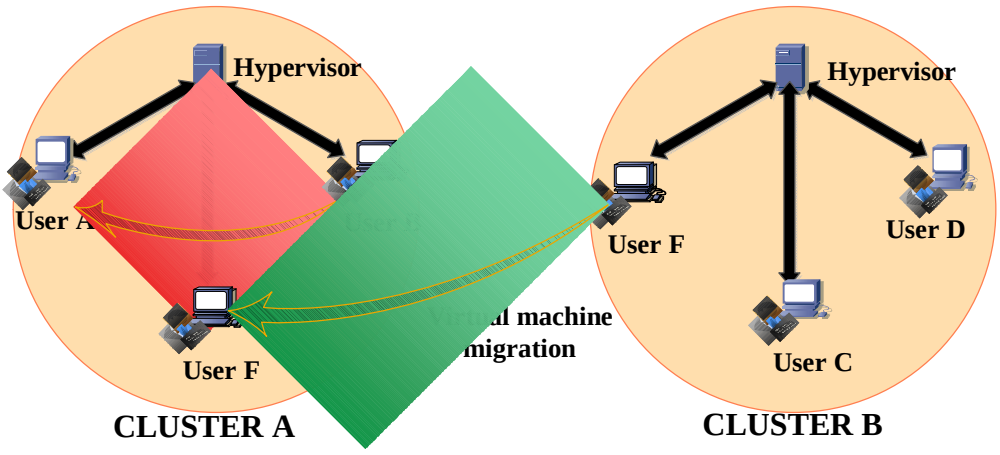

Figure 2: VM user F migrated from cluster B to Cluster A

IV. PROPOSED METHODOLOGY AND DISCUSSION

VMware workstation 9 was installed on a host that has windows 8 operating system. A virtual machine running Windows server 2008 R2 created on VMware workstation 9. 2 Hypervisors are created on VMware workstation 9 running the hypervisor VMware ESX/ESXi 4. vSphere client was used as the principal interface for administering vCenter Server and ESXi. vCenter Server used for the administering of all the server and virtual machines.

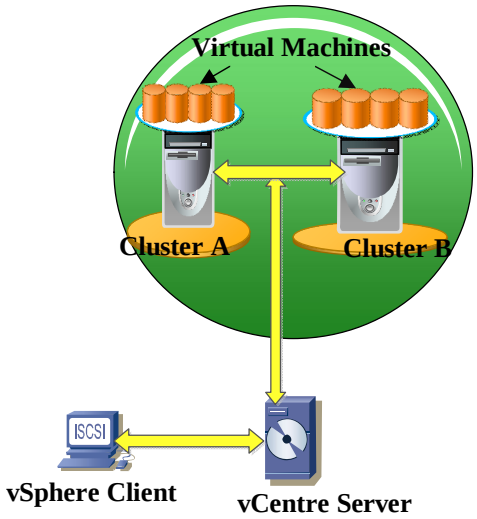

Figure 2: Setup of the vSpere and vCentre connected to VM on cluster A to Cluster B

Applications, like web browsers, games and Scilab were deployed to marked and known folders on all the virtual machines for easy checks on any changes to the installation folders. Applications and the virtual machines were tested for different aspects of security which could result in covert transfers. These include:

- Use of none, weak, medium and strong passwords on some applications and virtual machines to test the effects of unsecure authorisation that can be used to gain access to create covert channels..

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 6, June 2014

- Deleting of data and use data discovery software tools to search for some incompletely deleted data.
- Poor design and coding practices being tested on applications like web browsers given the dangers of other applications like malwares that could be used to create covert transfers.
- The sharing of hardware, software and other resources is also tested to if techniques used in data storage like data combination and blending can cause security issue in the cloud.

V. EXPERIMENTAL RESULTS

From the above experiments we came out with the following causes of covert transfers. We also suggested some of the countermeasures that can be used to reduce the effects of covert transfers.

A. CAUSES OF COVERT TRANSFER.

1. Unsecure Authorization:

Failure to lock down system resources against application identities and failing to limit database access to specified stored procedures can be vulnerabilities to the service provider. Inadequate separation of privileges and permitting over privileged accounts when using connection pooling to access information within the cloud storage can be used as a way of entry by attackers.

Users have a habit of using weak passwords that are ease to cram and may store clear text credentials in configuration files. Some pass clear text credentials over the network and permit prolonged session lifetime. These issues including the use of weak authentication mechanisms are a big issue of concern to cause covert transfers in cloud computing.

Service providers should do away with relying on a single gatekeeper, for example, relying on client-side validation only which they may be not sure of. Administrators use insecure custom administration interfaces and fail to secure configuration files on the server or store sensitive information in the clear text. This may be aggravated by having too many administrators, using over privileged process accounts and service accounts.

2. Poor design and coding practices:

Applications from different customers are deployed into the cloud, there is a high risk of some of them having bugs. Poor application design methods like the use structured exception handling (try/catch) or global exception handlers can be vulnerabilities revealing too much information to the client. This is also fueled by failure to specify fault contracts with the client. Some programmers leave non-validated input used to generate SQL queries relying only on client-side validation. They do not validate input from all sources including cookies, headers, parameters, databases, and network resources. These are vulnerabilities to cloud users. Unsecure application programming interfaces is also an issue that result in information leaks through covert transfers. As software interfaces or APIs are what customers use to interact with cloud services, those must have extremely secure authentication, access control, encryption and activity monitoring mechanisms.

3. Unrestricted Shared Technology:

Sharing infrastructure is a way of life for IaaS providers. Unfortunately, the components on which this infrastructure is based were not designed for that. To ensure that customers don't thread on each other's "territory", monitoring and strong compartmentalization is required. This calls for great need for the monitoring of potential treats and vulnerabilities between shared resources.

Each shared resource should be secured from illegal access that may cause covert transfers compromise the security measures that are in place and cause information leaks thereby affecting the availability, confidentiality and integrity of some sensitive information.

4. Data combination and Blending:

The Cloud Computing client needs to ensure that its private data whether its private data is stored separately from others or not. If they are combined or blended with those of other clients' data, then it is much more vulnerable or dangerous. For example, viruses might be transmitted from one client to others. If another client is the victim of a hack attack, the attack might affect the availability or integrity of the data of other companies located in the same environment.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 6, June 2014

5. **Malware:**
   A malware program attacks first a single computer as its host. If the computer is connected to a network, the malware outbreak takes place by spreading the attack to the other computers and makes the whole network down for maintenance. Some of the well-engineered program codes can do a very stealth way of penetrating the computer system.

   Data loss or data theft caused by covert transfers that are created by malware attacks. The damage caused by a successful attack that erases a user's data can be measured in terms of the value of the erased information to the user. If the attack targeted a home computer used for entertainment, the damage is probably minimal. The theft of important information can result in the loss of many years work, a valued photo archive or some other type of coveted correspondence. If data is stolen as the result of a targeted attack on a specific individual, the damage can be tremendous, particularly if the data belonged to a company or even the state – client databases, financial and technical documentation or even banking details can end up in the wrong hands – the possibilities are endlessly. We live in the information age and its loss or leakage can sometimes have disastrous consequences.

6. **Memory dumping:**
   An attacker is able to read sensitive data out of memory or from local files. Configuration file sniffing can be done by an attacker to steal sensitive information, such as connection strings, out of configuration files.

   Incomplete data deletion is too much risky in cloud computing, it does not remove completed data because the replication data is placed in other servers for example when a client request to remove a cloud resource then with most operating systems this will not remove accurately. Accurate data deletion is not possible because copies of data are stored in the nearest replica but are not available.

B. SECURITY COUNTERMEASURES

Countermeasure are safeguard that addresses a threat and mitigates risk [4] [5] [7] [8] [11]. As we have noticed the risks include information leaking to unauthorized parties covertly. In this section we try to look at some of the countermeasures that can be taken to address the risks of covert transfers

1. **Enforce applications separation.**
   Application should be controlled in the way they access shared resources. The settings to access shared resources should not allow leaking information to other virtual machines that should not have access to the sensitive information. Separation of applications including their files and settings is required so that they are well monitored how they are making changes to the resources.
   Avoid building generic roles with privileges to perform a wide range of actions. Roles should be designed for specific tasks and provided the minimum privileges required for those tasks. Good management of Administrative privileges is also required to a great extent**.** The ability to reactively defend an attack by shutting out a user should be supported through the ability to disable the connectivity of virtual machines. Maintain separate administration privileges. Consider granularity of authorization in the administrative interfaces as well. Avoid combining administrator roles with distinctly different roles such as development, test or deployment.
2. **Must catch exceptions:**
   Unhandled exceptions are at risk of passing too much information to the client. Handle exceptions when possible. Must not base on assumptions when it comes to security**.** Assuming all input is malicious means designing your application to validate all input. User input should never be accepted without being filtered and/or sanitized. Input and data validation should be performed using a common set of code such as a validation library.
   Never rely on client-side validation**.** Any code delivered to a client is at risk of being compromised. Because of this, it should always be assumed that input validation on the client might have been bypassed.
   Use structured exception handling in application development. A structured approach to exception handling lowers the risk of unexpected exceptions from going unhandled.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 6, June 2014

3. **Encrypt communication channel:**
   The channels should be encrypted and intrusion detection systems used. This will assist in identifying any hidden channel that may be used to leak information. Sensitive data should only be passed in encrypted form. This can be accomplished by encrypting the individual items that are sent over the wire, or encrypting the entire channel as with SSL. Keep unencrypted data close to the algorithm. Use decrypted data as soon as it is decrypted, and then dispose of it promptly. Unencrypted data should not be held in memory in code as this could result in some memory leaks.
   Provide strong access controls on sensitive data stores. Access to secret stores should but authorized. Protect the secret store as you would other secure resources by requiring authentication and authorization as appropriate.
4. **Good administration of user access to system-level resources:**
   Users should not be touching system resources directly. This should be accomplished through an intermediary such as the application. System resources should be restricted to application access. Sensitive data stored in application memory provides attackers another location they can attempt to access the data. Often this data is used in unencrypted form also. To minimize risk of sensitive data theft, sensitive data should be used immediately and then cleared from memory.
   Secure all the configuration store**.** The configuration store should require authenticated access and should store sensitive settings or information in an encrypted format.
5. **Separate public and restricted areas:**
   Applications that contain public front-ends as well as content that requires authentication to access should be partitioned in the same manner. Public facing pages should be hosted in a separate file structure, directory or domain from private content.
6. **Use application instrumentation.**
   Instrumentation refers to an ability to monitor or measure the level of a product's performance, to diagnose errors and to write trace information. Programmers implement instrumentation in the form of code instructions that monitor specific components in a system. When an application contains instrumentation code, it can be managed using a management tool. Instrumentation is necessary to review the performance of the application. Application transactions that are more likely to be targeted by malicious interactions should be logged or monitored. Examples of this might be adding logging code to an exception handler, or logging individual API calls. By providing a means to watch these transactions you have a higher likelihood of being able to identify malicious behavior quickly.
7. **Use multiple gatekeepers:**
   Passing the authentication system should not provide a golden ticket to any/all functionality. System and/or application resources should have restricted levels of access depending on the authenticated party. Some design patterns might also enforce multiple authentications, sometimes distributed through application tiers.

Use SSL to protect session authentication cookies. Session authentication cookies contain data that can be used in a number of different attacks such as replay, Cross-Site Scripting or Cross-Site Request Forgery. Protecting these cookies helps mitigate these risks

VI. CONCLUSION

From the survey we can conclude that both the service provider and the service consumer should look into security issues like exception monitoring systems. Seek an independent security audit of the host and details on third parties and whether they are able to access your data.

The hypervisor may fail to detect the covert channels which can cause illegal transfers. This requires frequent checks on all and every channel to see if it is legal or not. Remote access should be well managed and controlled so that only virtual machines at authorized levels can access data from other virtual machines.

Application threat modeling should be carried out [4] [8]. Threat modelling is an approach for analysing the security of an application. It is a structured approach that enables you to identify, quantify, and address the security risks associated with an application. Threat modelling is not an approach to reviewing code, but it does complement the security code review process. The inclusion of threat modelling in the SDLC can help to ensure that applications are being developed with security built-in from the very beginning.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 6, June 2014

Application separation methods are also helpful in reducing the risks of covert transfers. One of the methods that can be used is the use of sandboxes. A sandbox is a security mechanism for separating running programs. The sandbox typically provides a tightly controlled set of resources for guest programs to run in, such as scratch space on disk and memory. Network access, the ability to inspect the host system or read from input devices are usually disallowed or heavily restricted

REFERENCES.

1. Denz, Robert, and Stephen Taylor. "A survey on securing the virtual cloud."*Journal of Cloud Computing* 2.1 (2013): 1-9.
2. Greene, T. (2009). New attacks on cloud services call for due diligence. Network World. Southborough: Sep 14, 2009. Vol. 26, Iss. 28; pg. 8, 1 pgs. Retrieved from http://www.networkworld.com/newsletters/vpn/2009/090709cloudsec2.html
3. Edwards, J. (2009). Cutting through the fog of cloud security. Computerworld. Framingham: Feb 23, 2009. Vol. 43, Iss. 8; pg. 26, 3 pgs.
4. Dahbur K, Mohammad B, Tarakji AB (2011) A survey of risks, threats and vulnerabilities in Cloud Computing. In: Proceedings of the 2011 International conference on intelligent semantic Web-services and applications. Jordan: Amman. pp 1-6
5. http://searchservervirtualization.techtarget.com/definition/live-migration
6. Ertaul L, Singhal S, Gökay S (2010) Security challenges in Cloud Computing. In: Proceedings of the 2010 International conference on Security and Management SAM'10. Las Vegas, US: CSREA Press. pp 36-42
7. Jansen, W.A.; (2010), " Cloud Hooks: Security and Privacy Issues in Cloud Computing5719001 IEEE 2011 44th Hawaii International Conference on System Sciences (HICSS), pp1, 4-7 Jan. 2011.
8. Kresimir P; Zeljko H; (2010), "Cloud computing security issues and challenges", MIPRO 2010, May 24-28, 2010, Opatija, Croatia.
9. Subashini S, Kavitha V (2011) A survey on Security issues in service delivery models of Cloud Computing. J Netw Comput Appl 34(1):1-11
10. Minqi Z; Rong Z; Wei X; Weining Q; Aoying Z; (2010),"Security and Privacy in Cloud Computing: A Survey", Sixth international conference on Semantics Knowledge and Grid (SKG), pp 105, 1-3 Nov. 2010.
11. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for data storage security in cloud computing. In: Proceedings of the 29th IEEE International Conference on Computer Communications, INFOCOM'10, pp. 525–533. IEEE Press, Piscataway, NJ, USA (2010)

BIOGRAPHY

Wilson Bakasa born in 1983, received his BSc degree in Computer Science at Bindura University, Zimbabwe in 2008. He is currently doing M Tech CS final year at JNTUH, India. He is a HIT staff development research fellow. His research interests are in the area of Information security, wireless and sensor networks and cloud computing

Kudakwashe Zvarevashe born in 1986, received his BSc degree in Information Systems at MSU, Zimbabwe in 2010. He is currently doing M Tech IT final year at JNTUH, India. He is a HIT staff development research fellow. His research interests are in the area of big data, information security, cloud computing and web services.

Nicholas N. Karekwaivanane born in 1985, received his B Tech degree in Computer Science at MSU, Zimbabwe in 2010. He is currently doing M Tech CS final year at JNTUH, India. He is a HIT staff development research fellow. His research interests are in the area of computer networks, information security, cloud computing and adhoc and sensor networks.