International Journal of Computer Science and Information Technology Research ISSN 2348-1196 (print)
ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (613-618), Month: April - June 2015, Available at: www.researchpublish.com

Survey on Different Methods for Attacking Virtual Machine

[1]Radha Korimani, [2]Geetha S, [3]Mahesh Kaluti

[1, 2, 3] Visvesvaraya Technological University, Alva's Institute of Engineering and Technology, Managlore, India

Abstract: **One of the benefits of IaaS to the shopper is that the rapid property of their provision. This property can would like relocation of a VM from one physical machine and / or one hyper-visor to a different. Whereas such migration is obvious and likely seamless, it's about to in addition introduce vulnerability. We explore here the potential for a malicious user to exploit vulnerabilities associated with mobile VMs to urge huge volumes of cloud-user information. In this paper ,we are presenting totally different attacks on virtual machines throughout migration that don't seem to be detectable.**

Keywords: **VM, sniffer.**

1. INTRODUCTION

Virtual machine migration is method of moving a running virtual machine or application between totally different physical machines while not disconnecting the client or application. Insiders in Cloud Computing environments probably have access to no equal amount of various organisations knowledge. With Infrastructure as a Service (IaaS), client knowledge is housed in their virtual machine (VM), that is within the possession of the cloud provider. The provider can lawfully prove copies and backups of a VM, delete VMs and, with service-level-agreement acceptance, login to a customer's VM for body functions. As per Gartner [3], the number of companies pattern IaaS is prepared to continue rising with a compound annual rate of growth expected to be 41.7% over the four years to 2016. This emphasises the sheer volume of client information being settled with cloud suppliers. To a malicious government, there's most likely worth in obtaining copies of a VM or extracting information from a given VM, this might be analogous to intruder making an entire copy of one of your personal computer devices, for his or her own use and without your knowledge. This information harvested maliciously in different ways, but to the current purpose no methods have been released that leave no theorthical proof. Therefore it can often prompt insider attack aimed toward obtaining copies of virtual machines is always detectable.

The ideal state of model would be to detect attacks as early within the kill-chain as possible, either predicting attack or detection it inside the first stages, since this minimises costs and actual exposure. Before we going to achieve this, we'd like to grasp what quite attacks are performed, however we are able to locate them. The identification of the methods to compromise a VM without any current methodology of detection wouldn't be terminal for IaaS as a technology, but the use of IaaS in environments where sensitivity is terribly high would instantly become inappropriate without any development. The remainder of this paper is unionised as follows: related work discusses existing analysis throughout this and similar areas; in proposed system we tend to mentioned regarding 2 technologies to intake copy and/or data of vitual machines and totally different attack vectors by analysing capture file created throughout virtual migration. Packet sniffing and Packet Sniffing detection methods are discussed.

2. RELATED WORK

According to Carnegie Mellon's CERT [2], this paper focusses the cloud related insider threat has three different perspectives: the rogue cloud-provider administrator, the employee in the victim organisation that exploits cloud weaknesses for unauthorised access, and the insider who uses cloud resources to carry out attacks against the company's local IT infrastructure. a fourth actor is Acid Cloud, where the cloud provider, itself, is the malicious actor.

In 2008, the topic of passive and active snooping of VM migration was discussed in a paper by Oberheide et al [4]. The paper focuses on active snooping using VMWare version 3 and Xen 3.1 and alters the contents of the VM as it passes the sniffer. The paper demonstrates a successful ssh attack on the compromised VM. This attack vector would be very difficult to detect on a live migration as a hash is not possible while the machine state is still changing. If the VM is powered off or suspended, a hash can be taken of the disk image file before and after the migration. If the hashes are not the same, then the contents of the VM were altered in some way during migration. If the contents were altered on a migration of a running VM, detection would be much more complex.

Shetty et al expand on this work in [5], with five attack vectors identified: denial of service (disabling a hypervisor), internal attacks (compromising the hypervisor or other guest VMs), guest VM attack (gaining control of a VM), false resource sharing (attracting migration to a compromised or malicious hypervisor), and inter-VM attack (attacking a sibling VM). They suggest four ways to secure the VM in transit.The first is to use a VLAN for each VM, second is a special hypervisor known as the Network Security Engine Hypervisor (NSE-H), third is role-based migration and The last option discussed is a Virtual-TPM-based migration protocol. They point out that this protocol doesn't support live migration, which makes it incomplete for current Cloud environments.

3. PROPOSED SYSTEM

Virtual machine migration is the process of moving virtual machine or application from one server/system to another system/server. Figure 1 shows a situation were virtual machine migration takes place. Prior to identifying migration as a key source of exposure to attack, we examined both low-technology and hightechnology methods of attacking a VM to obtain customerdata without being detected. We examined ways that a variety of attackers could use to compromise a VM with a relatively low level of technical ability or knowledge. We then progressed on to more sophisticated approaches that would be within the ability of support staff with a greater technological knowledge or within the capacity of an insider to perform using a set of detailed instructions.

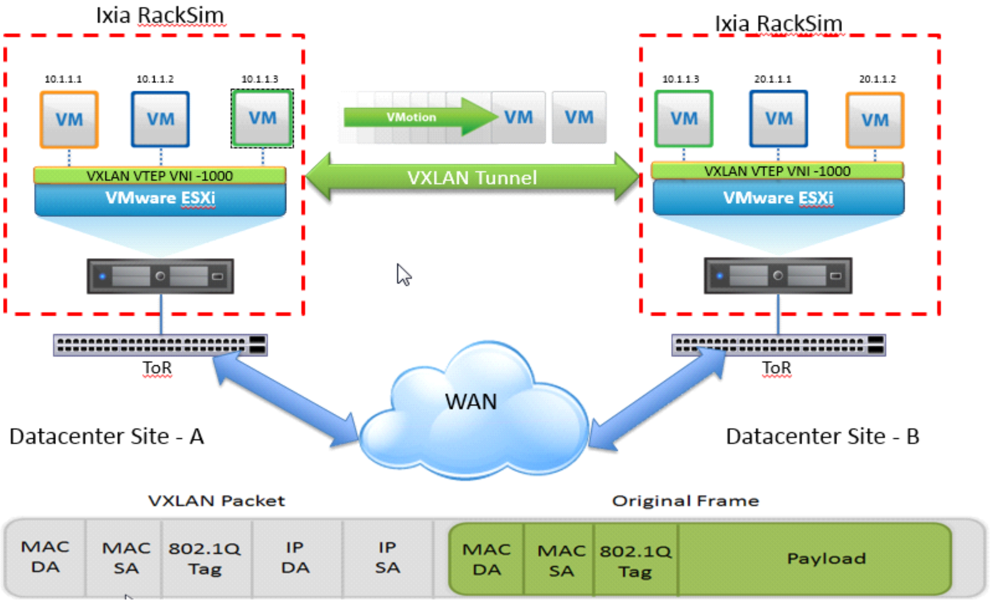

Figure 1: Virtual Machine Migration

Low-Technology Attacks:

Initially we examined some of the more simple methods of compromising a VM, which are achievable by a variety of insiders. An obvious method of obtaining a virtual machine's data would be to make a duplicate, then boot that duplicate either on the same hypervisor or on a hypervisor in a remote location. Making duplicates using the hypervisor however, will leave audit trail entries on the hypervisor making it easy to discover that a VM has been compromised. One could copy the datastore of a VM using the Datastore Browser in VMWare. The actions carried out using the datastore browser are all logged and this log can be monitored in real time using scripts to detect such events. This form of attack would be detectable as it happens making it highly overt.

One can access the datastore server directly, If one were to copy a Virtual Machine's configuration files and datastore on a Windows machine, *link* (.lnk) files are made detailing the duplication with its source and destination [6]. In addition to the link files, Windows records serial numbers of external memory attached to the operating system. With these two techniques, a file that is copied can be traced back to the originating machine, user account and destination media together with the date and time that it was copied.

The malicious insider could also FTP or use a cloud storage account to transfer the upload the files to another server, from where they can be downloaded and accessed at a later date. The use of outgoing FTP or cloud server access can be monitored by the systems administrators who can terminate transfers of certain file types, or lengths. Linux can also easily record system events, such as file copy and FTP commands. One of way of achieving this is the central recording of shell history entries. This makes it extremely difficult for an insider to copy individual files or a VM without leaving evidence, which is ideal for the owner of the VM and the security auditors of the provider. Alerts can be created to detect these commands as they are entered to allow system administrators to intercept these attacks as they happen.

Higher-Technology Attacks:

By their very nature, Virtual Machines require flexible storage and adaptive physical locations. A VM may require additional disk space or memory, which it may not be possible to be facilitate on the physical machine on which it currently resides. To accommodate this, the VM's datastore or host files must be moved to a new physical machine. An attacker with a higher level of technological knowledge may use the network to attack a VM while it is in transit, thus removing the possibility of host-based detection.

The migration of virtual machines has three options. One can migrate the host files, the vmdk file (the datastore), or the two together. This last option is simply a combination of the former two, but requires the OS to be powered off: the former two can be done while the machine is powered up and being accessed. Live-migration facilitates the moving of the guest OS without it being powered down.

The user of the VM can continue to use the VM as if migration is not taking place. The hypervisor creates a cache of events(capture file) that occur, which is then used to update the VM in its new location. The benefit to this type of migration is that the user does not experience any down-time. Although most of the data of interest to an attacker resides in the hard-drive image in the datastore or in memory.

The following information is stored in capture file during migrations:

- The contents of the clipboard were seen in the captured file. Although the contents were separated by a null (0x00) byte between each character.

- The contents of the VM's log files were also sent and visible in the capture files with no padding between characters present.

- Guest OS configuration files are present in plain text.

- Installed programs on Ubuntu were listed together with their version numbers.

- Magic number fields were seen in the capture files.

- The entire contents of the VM's configuration files were seen in plain text (vmdk, vmx and vmxf.)

- The SSL certificate settings and whether the default certificate was still being used.

- BIOS details, in this case Phoenix Bios, can be seen.

- Website URLs that were visited on the VM were seen.

The contents of capture file will become attack vectors for malicious user. Not all of these information have been translated into attack vectors as yet. Several attack vectors can be identified however, which was sufficient for us to determine that the monitoring of network traffic is a very real risk for the security of VMs.

Some of the information in capture file presented clear opportunities for a malicious insider to harvest customer data. These are set out below.

A. Clipboard:

With the contents of the clipboard obtainable machines and capture the contents of the clipboard. It is in clear text, a malicious insider can monitor the movements of virtual also not uncommon for people logging in to many systems to copy their password to the clipboard for rapid re-use, indeed several password programs offer this functionality. Whole images or documents may also be copied to the clipboard, which will now be accessible to the malicious insider. With each additional migration, more information can be obtained by the attacker. The potential for cascading attacks from the contents of the clipboard is very real.

B. VM Cloning:

By capturing the configuration files during migration, an attacker can bypass command logging systems and forensic systems for detecting file access. The attacker have will now access without any current method of detection. The detection of the VM's configuration files suggested a possible attack vector via making an unknown copy of the VM and accessing the target's datastore.

- The configuration files were saved out to new files and a new virtual machine was created on the same datastore.

- The vmx file had several entries updated: displayName and extendedConfigFile were changed to the new VM's name and ide0:0.fileName was changed to the full path of the target vmdk file.

- The vmxf file had one entry updated: vmxPathName was changed to point to the above file name.

- The captured configuration files were uploaded to the datastore to override the newly created ones.

- An attempt is made to power-up the newly created VM. VMWare detects that there is an issue with the vmdk file not being in the same directory. The user is prompted to say whether they moved it or copied it: the *Copied* option was selected. The boot terminates.

- The new VM now uses the original VM's vmdk file, which is detectable, so a clone of the new VM is made, which copies the new configuration files and the original vmdk file to a new location. This process creates a complete clone of the original. There is some access needed to the original vmdk file, which is the only link from the new VM to the original.

As the log files are recorded for each machine, the only log trace of the copy is on the new VM. A check of the main log file (viclient) logs after the above process reveals no trace of the cross-VM link. Once the insider has perfected this attack, the viclient log file will show no trace of the cross-VM link. This attack vector does require a new VM to be created in order to access the target VM datastore, which by its very nature leaves evidence. The process of creating and cloning a new VM in a highly populated hypervisor may be a highly common event and therefore may go completely undetected or unquestioned. The stealth element of this attack is only possible by the capturing of configuration files during migration.

Once the malicious insider has a complete, and unknown, copy of the virtual machine they can recover data in a variety of ways. The most obvious method being to use software tools to compromise the administrator account, affording access to the virtual machine in its entirety. By disconnecting the network, the malicious insider will circumvent any client-installed software that could report the computer as active. Once compromised, the malicious insider can use brute force to recover encrypted data, or obtain user credentials for use in cascading attacks on other systems.

C. File Carving:

Although the VM had a relatively small datastore size, we were not able to completely capture the 4-gigabyte files during their migration without packet loss. With migration over longer distances, one will expect slower speeds and lower packet loss in the captures making a full capture possible. We surmise that a malicious insider may also find it difficult to capture, complete datastore files on a local network. While extracting the configuration files, we noticed magic numbers [1] in the capture files, which reduced the impact of this problem. This suggested that other complete files inside the capture file could be detected using file carving techniques. This attack vector would be significantly easier than attempting to capture a large disk images reliably without packet loss, but would still yield a significant number of files from the guest OS. This attack vector is undetectable, the attack vector will only negate the hypothesis if the packet sniffing is undetectable.

D. Packet Sniffing:

One way to avoid this is for the packet sniffer[7] to not be connected to the network in any way and eavesdrop on an existing wire. we split an ethernet cable and exposed the inner wires. While the cable was still connected to a live machine and the switch, the Transmit (green, and white with green stripe) and the Receive (orange, and white with orange stripe) wires were punched down into separate RJ45 terminator sockets, creating a t-piece for each, thus creating a passive ethernet tap as shown in figure 2. This can also be achieved using a patch panel.

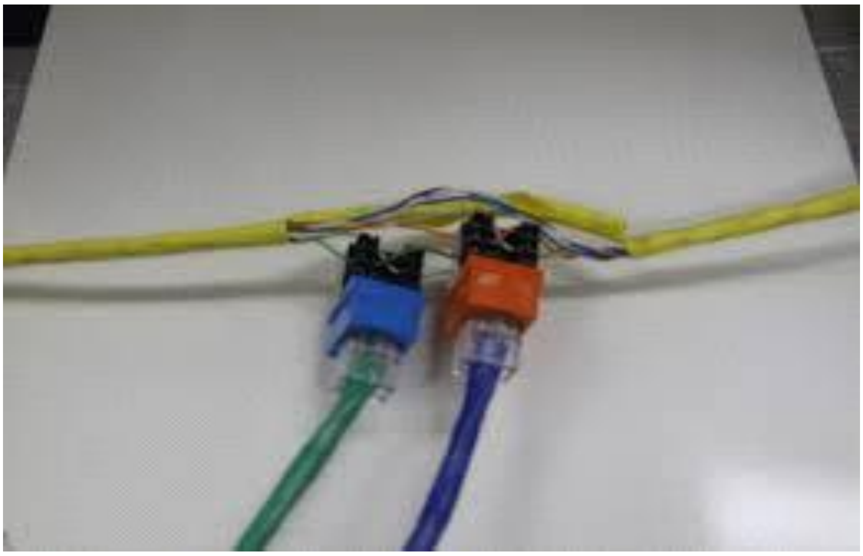

Figure 2: Passive Ethernet Tap

The switch was being monitored during this process and showed no connection loss. Although this process is visible at the time of attack, the ethernet cable can replaced afterwards and taped back together with minimal visual clues that an attack took place. An insider is likely to be able to obtain the physical access needed with relative ease. The switch did not display any error messages. This confirms that the packet sniffing can take place with the switch not being able to detect the presence of the machine attacked to the t-piece. We were then able to monitor all traffic on the chosen pair using the passive ethernet tap.

Current Packet-Sniffing Detection Methods:

1. Detection Using Decoy Method:

The decoy method is to detect the use of stolen credetials, which were set up as a honeypot. Although this would be an effective method of determining a compromised system, it won't necessarily indicate the presence of a packet sniffer and more importantly, won't tell one where the sniffer is located.

2. Detection Using ARP:

Using the setup described at the beginning of this section, we monitored both tapped pairs of cables, i.e. the Tx and Rx pairs of the machine generating the ping requests. The monitoring machine was able to see all of the ping request traffic on one ethernet card and the reply traffic on the other. A ping request was sent to the IP address of the sniffing machine, however, it did not respond on either network card. A special ARP packet can be created that deceives an OS running a packet sniffer in promiscuous mode into responding to the ARP request even though it shouldn't.

3. Detection Using TDR:

The passive ethernet tap was not visible using conventional hardware or software detection methods. The only weakness in its method is the physical interference of the tap itself. In the communications industry, breaks and failures in underground wire cables have been a longstanding and expensive problem. This is due to the fact that the failure could occur anywhere along a cable that can often be many kilometres in length, and underground. A common method of detecting these faults is time-domain reflectometry (TDR). Time-domain reflectometry involves sending very short electrical signal pulses with very fast rise times down the faulty line (or light pulses in the case of optical fibre cables), and if the pulse reaches a point where any impedance mismatch occurs, then a ghost ’echo’ signal is reflected back towards the source (with a signal strength that depends upon the extent of impedance mismatch, thus open and short circuits provide the largest echo signals). It is, of course, for this reason that network cables and devices need to be impedance matched to the network to prevent loss of data integrity. Through timing how long it takes for the pulse’s reflection to return, multiplying that time by the speed with which the pulse travels down the line, and then dividing by two (there and back again) we can calculate the distance to the point of failure. A passive tap on any cable in the network will provide an open circuit (and thus a point for impedance mismatch) and can then, in principle, be detected by TDR.

4. CONCLUSION

In this paper, we have illustrated a variety of methods of attacking virtual machines, with the low-technology methods detectable by current network administration and computer forensics methods. We empirically demonstrated that packet sniffing can be used to extract a significant amount of valuable information from a virtual machine and highlighted several methods of achieving this. This is significant enough to flag migration attacks as a very real threat to the security and integrity of customers’ data.

A malicious insider attaching a packet sniffer via a passive tap is a serious issue for a cloud provider due to the sheer volume of data that can be harvested and the difficulty of detecting it. The only way to detect such a device would be a visual inspection of the cabling, which in many cases will be impractical, and also may not be perfectly reliable. From a network-technological perspective, this attack is undetectable using current tools and confirms the null hypothesis.

REFERENCES

[1] L. Aronson and J. van den Bos. Towards an engineering approach to file carver construction. In Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual, pages 368 –373, July 2011.

[2] William R Claycomb and Alex Nicoll. Insider threats to cloud computing: Directions for new research challenges. In Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual, pages 387–394. IEEE, 2012.

[3] Gartner. High-tech Tuesday webinar: Gartner worldwide IT spending forecast, 2q12 update: Cloud is the silver lining. Online, 1 August 2011.

[4] J. Oberheide, E. Cooke, and F. Jahanian. Empirical exploitation Of live virtual machine migration. In Proc. of BlackHat DC convention, 2008.

[5] J. Shetty, M.R. Anala, and G. Shobha. A survey on techniques of secure live migration of virtual machine. International Journal of Computer Applications, 39(12), 2012.

[6] A. Svensson. Computer forensics applied to windows NTFS computers. Stockholm’s University, Royal Institute of Technology, 2005.

[7] T. King. Packet sniffing in a switched environment. SANS Institute. August 4th, 2002.

[8] http://www.gartner.com/resources/