IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-11 E-ISSN: 2347-2693

Performance Enhanced Live Migration of Virtual Machines in the Cloud

D. Ragupathi[1] and S.Sivaranjani[2*]

[1]*Asst. Professor, Department of Computer Science, A.V.V.M Sri Pushpam College, Thanjavur*
[1]*M.Phil Research Scholar, Department of Computer Science, A.V.V.M Sri Pushpam College, Thanjavur*
www.ijcseonline.org

Received: Oct/23/2015 Revised: Nov/06/2015 Accepted: Nov/18/2015 Published: Nov/27/2015

***Abstract*—**As virtualization proceeds to gotten to be increasingly well known in enterprise and organizational networks, operators and administrators are turning to live rearrange of virtual machines fat that point a pick up the purpose of work notice adjusting and management. However, the security of live virtual machine rearrange has yet to be analyzed. This paper looks at this poorly explored range and endeavors to empirically illustrate the significance of securing the migration process. We start by defining and investigating three classes of dangers to virtual machine migration: control plane, data plane, and rearrange module threats. We at that point show how a malignant party utilizing these assault frame meets expectations can exploit the latest versions of the well-known Xen and VMware virtual machine monitors and present an instrument to automate the control of a visitor working system's memory amid a live virtual machine migration. Utilizing this experience, we talk about frame meets expectations to ad-dress the deficiencies in virtualization programming and se-cure the live rearrange process.

***Keywords*—** Xen, VMWare, Virtual Machine, Live Migration

I. INTRODUCTION

Recent advances in virtualization have made virtual mama chines an increasingly vital relook and operational area. Successful business ventures counting VMware, Xen Source, and Parallels have accelerated the adoption of virtualization programming in numerous organizations. According to a recent IDC report, the number of virtualized servers will rise at a compound annual growth rate of over 40% from 2005-2010. Live rearrange of virtual machines (VMs), the process of transitioning a VM from one virtual machine monitor that point a pick up (VMM) to another without halting the visitor operating system, frequently between distinct physical machines, has opened new opportunities in figuring. Implemented by a few existing virtualization products, live rearrange can aid in perspectives such as high-avail limit services, transparent mobility, consolidated management, and work notice adjusting. While virtualization and live rearrange enable important new functionality, the combination introduces novel security challenges. A virtual machine monitor that point a pick up that in-corporates a vulnerable execution of live rearrange usefulness might expose both the visitor and host operating framework to assault and result in a tradeoff of integrity. Given the huge and expanding market fat that point a pick up virtualization technology, a comprehensive understanding of virtual machine rearrange security is essential. However, the security of virtual machine rearrange has yet to be analyzed. This paper presents a point by point investigation of the issue and investigates three classes of dangers to the rearrange process.

- **Control Plane:** The correspondence instruments utilized by the VMM to start and oversee live VM territories must be authenticated and resistant to tampering. An aggressor might be capable to manipulate the control plane of a VMM to impact live VM territories and pick up control of a visitor OS.

- **Information Plane:** The data plane over which VM territories happen must be secured and secured against snooping and altering of visitor OS state. Detached assaults against the data plane might result in leakage of delicate data from the visitor OS, while dynamic assaults might result in a complete com-promise of the visitor OS.

- **Relocation Module:** The VMM part that implements rearrange usefulness must be resilient against attacks. On the off chance that an aggressor is capable to subvert the VMM utilizing vulnerabilities in the rearrange module, the aggressor might pick up complete control over both the VMM and any visitor OSes.

This paper investigates assaults against live virtual ma-chine rearrange in the context of these three threats. We present a few viable assaults against the rearrange usefulness

of the latest versions1 of the Xen and VMware virtualization items and develop a instrument to automate the control of a visitor virtual machine's memory amid live migration. Utilizing this experience, we talk about frame meets expectations to address the deficiencies in virtualization programming and secure the live rearrange process.

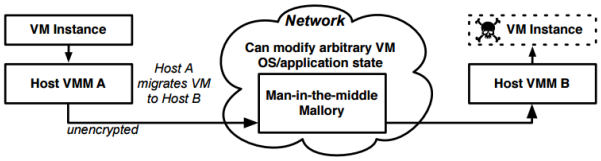

Figure 1: An illustration of a man-in-the-focus assault against a live VM migration

II. BACKGROUND

Virtual machines and virtualization innovation give numerous technical and taken a toll advantages. However, the utilization of virtualization too introduces a novel set of security challenges. In particular, there are novel concerns related with virtual environments such as se-curing huge numbers of virtual machines, securing a di-verse range of working frame meets expectations and applications over virtual images, and securing versatile virtual machines that might move between distinctive physical has and net-works. There are numerous ways in which a virtual machine can be moved from one VMM to another. Since virtual systems are ordinarily put away as regular records on disk, the records related with a halted framework can be duplicated to another VMM utilizing a framework at that point a pick up utilizing portable limit gadgets such as USB drives. In expansion to the rearrange of halted virtual systems, numerous well known VMMs support live migration, the process of transitioning a VM from one virtual machine monitor that point a pick up to another without halting the visitor working system. While different virtual machine monitors have distinctive wire conventions fat that point a pick up live migration, the hidden algorithms are similar. Live rearrange methods more often than not start by copying memory pages of the VM over the framework from the source VMM to the destination while the VM proceeds to run inside the source VMM. This process proceeds as pages are dirtied by the VM. At the point when the source VMM reaches a threshold and deems that no additional huge progress is being made in the transferring of dirty pages, it will halt the VM, send the remaining memory pages, and signal the destination VMM to resume the execution of the VM. The point at which the VMM decides to halt the source VM and duplicate the remaining pages is more often than not an implementation-particular heuristic that endeavors to balance and minimize both the length of time of rearrange and the downtime of the moving VM. Other variations include the destination VMM resuming the VM early and requesting pages from the source VMM on-demand.

While one might accept that frame meets expectations over which VM pictures are relocated are secure, this is not an entirely safe assumption anymore. As live VM migration gets to be more regular in numerous organizations, it is likely that the rearrange travel way might span multiple commodity frame meets expectations and huge geographic distances. Indeed, virtual machines have been success completely relocated over continents with application downtimes as low as 1 to 2 seconds. In addition, a traded off framework inside a framework employing live territories can facilitate untrusted access to moving VM images. The limit to view at that point a pick up change data related with live migrations at that point a pick up impact the rearrange administrations on source and destination VMMs raises a few vital security questions. In the next area we elaborate on the some of these threats.

III RELOCATION ATTACK CLASSES

In this section, we introduce three classes of dangers to live virtual machine rearrange and depict a few at-tacks applicable to each.

3.1 Control Plane

The correspondence instruments utilized by the VMM to start and oversee live virtual machine territories must be authenticated and resistant to tampering. In addition, the conventions utilized in the control plane must be secured against spoofing and replay attacks. A lack of fitting access control might permit an aggressor to arbitrarily start VM migrations.

Incoming Relocation Control: By initiating unauthorized approaching migrations, an aggressor might utilization visitor VMs to be live relocated to the attacker's machine and pick up full control over visitor VMs.

Outgoing Relocation Control: Similarly, by initiating outgoing migrations, an aggressor might relocate a huge number of visitor VMs to a legitimate victim VMM, overloading it and utilizing disruptions at that point a pick up a denial of service.

False Resource Advertising: In an environment where live territories are started automatically to disseminate notice over a number of servers, an attacker might be capable to falsely publicize available re-sources by implies of the control plane. By pretending to have a huge number of spare CPU

cycles, the aggressor might be capable to impact the control plane to mi-grate a VM to a traded off VMM.

As most existing VM items rely on manual intervention to start a migration, their access control mechanisms fat that point a pick up the control plane are simplistic. Fat that point a pick up example, Xen employs a whitelist of host addresses allowed to per-form migrations. However, as automatic territories fat that point a pick up load-adjusting between numerous machines gotten to be more common, possibly over different administrative do-mains and between unpredictable host addresses, mechanisms fat that point a pick up policing the control plane must be exhibited and maintained.

3.2 Information Plane

The data plane over which VM territories happen must too be secured and secured against snooping and tampering in demand to secure the VM's state. An aggressor might be capable to logically position himself in the rearrange travel way utilizing a number of methods such as ARP spoofing, DNS poisoning, and course hijacking. With such a position, an aggressor can conduct a man-in-the-focus assault as illustrated in Figure 1.

Detached Snooping: Detached assaults against the data plane might result in leakage of delicate information. By checking the rearrange travel way and associated framework stream, an aggressor can extract information from the memory of the moving VM such as passwords, keys, application data, and other protected resources.

Active Manipulation: One of the most severe at VMware tacks, an inline aggressor might control the memory of a VM as it is relocated over the network. Such a man-in-the-focus assault might result in a complete and covert trade off of the visitor OS

Even in the occasion that fitting encryption and identity administration is used, it still might be conceivable fat that point a pick up an aggressor to pick up valuable data from snooping on a rearrange stream. Fat that point apick up example, an aggressor might be capable to uniquely identify visitor VMs based on characteristics of the rearrange flow, such as size and duration, and identify the endpoint VMMs involved in the migration. This data might aid an aggressor in focusing on a later assault against a particular VM at that point a pick up critical foundation supporting that VM.

As we will illustrate in the next section, well known VMMs conveyed in production networks, such as Xen and VMware, fizzle to implement indeed basic data plane security to guarantee visitor OS honesty amid live migration and are vulnerable to attack.

3.3 Relocation Module

The VMM part that actualizes live rearrange usefulness must too be resilient to attacks. As the migration module gives a framework administration over which a VM is transferred, regular programming vulnerabilities such as stack, heap, and entirety number overflows can be exploited by a remote aggressor to subvert the VMM. Given that VM rearrange might not just be viewed as a publicly exposed service, the code of the rearrange module might not be scrutinized as thoroughly as other code. While such assaults are regular over all sorts of software, special attention should be utilized on the security of a VMM's rearrange module. As the VMM controls all the visitor working frame meets expectations running inside it, the severity of a VMM vulner limit is much more prominent than most normal software. On the off chance that an aggressor is capable to com-promise a VMM through its rearrange module, the integrity of any visitor VMs running inside the VMM, and any VMs that are relocated to that VMM in the future, might too gotten to be compromised.

As we will talk about in the next section, a brief audit of Xen's rearrange module resulted in different vulnerabilities that might trade off the VMM.

IV IMPLEMENTATION AND EVALUATION

We created a tool, *Xensploit*, to perform man-in-the-focus assaults on the live rearrange of virtual machines. The instrument operates by manipulating the memory of a VM as it traverses the framework amid a live migration. *Xensploit* is based on the frag route framework. While its name is influenced by the to start with VMM (Xen).

While its name is influenced by the to start with VMM (Xen) we joined it to, *Xensploit* is capable to control territories as well. In the taking after evaluations, we illustrate assaults against the data plane class of both the Xen and VMware VMMs. In addition, we explore assaults against the rearrange module of Xen, coming about from different vulnerabilities found through an audit of Xen's rearrange code.

4.1 Attack Evaluation

4.1.1 Simple Memory Manipulation

To evaluate *Xensploit*, we performed a basic proof-of-concept control amid the live rearrange of a Xen VM. In Xen terminology, a host VMM is known as a dom0 range while visitor VMs are known as domU domains. Our tested consisted of three machines: the source dom0, the

destination dom0, and a malignant hub running *Xensploit*. We started a new visitor domU, the range to be migrated, inside the source dom0. Inside domU, we executed a test process that basically prints a "Hi World" string to the terminal each second.

```
1180795919.260261: Hi World!
1180795920.270992: Hi World!
1180795921.281870: Hi World!
```

The live rearrange was at that point activated to move domU from the source dom0 to the destination dom0. As the memory pages of the running visitor OS are transmitted over the framework and pass through the malignant hub running *Xensploit*, the "Hi World" string is replaced with our custom value.

```
1180795921.920290: Xensploited!
1180795922.932574: Xensploited!
1180795923.942636: Xensploited!
```

In a matter of seconds, the visitor OS has been seamlessly relocated to the destination dom0. As expected, *Xensploit's* man-in-the-focus assault was fruitful and the memory of our test process has been manipulated, resulting in the new string being printed to the terminal of the visitor OS inside the destination dom0.

4.1.2 SSHD Authentication Manipulation

As a more advanced and viable illustration of our tool, we instrumented Xensploit to control the memory of the Secure Shell daemon (sshd) process of a visitor VM amid a live migration. Instead of performing the assault on Xen again, we switched our association to VMware Virtual Base to illustrate Xen-sploit's flexibility. Our testbed consisted of four ma-chines: the source and destination VMMs both running VMware ESX Server 3.0.1, a administration hub running VMware Virtual Base Client/Server 2.0.1 to oversee the VMMs and start the migration, and the malignant hub running Xensploit.

Sometime recently initiating the migration, we attempted to ssh to the visitor OS running inside the source VMM. The sshd process was arranged to just permit confirmation of the sort PubkeyAuthentication. As our public key was not in the root user's .ssh/authorized keys file, access was denied.

```
jonojono@apollo ~ $ ssh   root@testvm1
Permission   denied (publickey,keyboard-interactive).
```

We at that point started the live rearrange by implies of the Virtual Base Client and performed the man-in-the-focus assault utilizing Xensploit. Specifically, the in-memory object code of the sshd process, originating from client key allowed2 limit in auth2-pubkey.c, is controlled amid rearrange to successfully complete authenticate any approaching ssh logins. As seen below, after Xensploit's attack, our attempt to ssh to the VM succeeds due to the controlled sshd process.

```
jonojono@apollo ~ $ ssh   root@testvm1
Last  login: Tue Jun   5 19:25:19 2007 from
localhost   testvm1 ~ #
```

These examples of manipulating the memory of the visitor OS are just a little subset of the conceivable assaults outlined to evaluate our Xensploit tool. Much more insidious man-in-the-focus assaults can be performed such as transparently slipping a rootkit into kernel memory amid the live migration.

4.1.3 Xen Relocation Module

While exploring the Xen source code, we dis- covered different issues which fall into the migration module class of live rearrange threats. The vulnerabilities are present in Xen's VMM migration routines, specifically the code in xen-3.1.0- src/tools/libxc/xc range restore. c, which is utilized to restore an approaching rearrange to operational state.

As we previously mentioned, exploitable vulnerabilities in a VMM are especially serious as the honesty of all the presently hosted VMs, and any VMs relocated to the exploited VMM in the future, might be compromised. One vulnerlimit exploits an entirety number signedness issue coming about in a stack overflow, and yet another involves a malloc() entirety number flood coming about in a potential pile overflow. These two issues might permit a remote aggressor to accomplish privileged code execution and completely trade off the Xen VMM and host machine.

The vulnerabilities have been reported to the Xen-Source development team and will be resolved in an up-coming release. Further subtle elements regarding the particular routines affected can be found in Appendix A.

V. DISCUSSION

This paper has empirically demonstrated how two of the most well-known and broadly conveyed VMMs, Xen and VMware are vulnerable to viable assaults focusing on their live rearrange functionality. These dangers are utilization fat that point a pick up concern and require that suitable solutions be applied to each class of live rearrange threats.

In demand to support the secure rearrange of virtual ma-

chines, mutual confirmation of the source and destination VMMs, as well as any administration agents, must be performed to secure the control plane communications. Flexible access control policies should permit ad-ministrations to oversee rearrange privileges. The data plane over which the rearrange happens must be secured against snooping and control of the state of migrating VMs. Solutions incorporate protecting the travel way utilizing encryption at that point a pick up utilizing a separate physical at that point a pick up virtual framework to partition and isolate rearrange development from potential threats. While encrypting the rearrange might seem like a trivial arrangement to implement, effectively maintaining a public key foundation to guarantee mutual authentication will add huge complexity to VM management programming and might be infeasible fat that point a pick up certain deployments. Finally, robust secure coding methods such as in-put validation, privilege separation, type-safe languages, and continuous code audits can help lessen the chance of compromises of the rearrange module of a VMM.

Traditionally, a breach in framework security results in a tradeoff of data integrity. However, at the point when dealing with virtual machine rearrange of full working systems, a breach in the framework can result in not just a tradeoff of data, however too host integrity. This fundamental shift in the threat model of a framework might require re-thinking existing access control and isolation mechanisms. Fine-grained framework access control frame meets expectations such as SANE might give sufficiently flexible policies to address such a threat model. Be-yond VLANs, complete virtualization of framework re-sources throughout the stack might permit isolation fat that point a pick up secure territories and might give an inherent complement to host virtualization.

References

[1] Chanchio, K. ; Dept. of Comput. Sci., Thammasat Univ., Patumtani, Thailand ; Thaenkaew, P., "Time-Bound, Thread-Based Live Migration of Virtual Machines", Published in: Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on Date of Conference: 26-29 May 2014 Page(s): 364 – 373.

[2] Anala, M.R. ; Dept. of Comput. Sci. & Eng., RVCE, Bangalore, India ; Shetty, J. ; Shobha, G., "A framework for secure live migration of virtual machines", Published in: Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on Date of Conference: 22-25 Aug. 2013 Page(s): 243 – 248.

[3] Shah, S.A.R. ; Korea Univ. of Sci. & Technol., Daejeon, South Korea ; Jaikar, A.H. ; Seo-Young Noh, "A performance analysis of precopy, postcopy and hybrid live VM migration algorithms in scientific cloud computing environment", Published in: High Performance Computing & Simulation (HPCS), 2015 International Conference on Date of Conference: 20-24 July 2015 Page(s): 229 – 236.

[4] Deshpande, U. ; Binghamton Univ., Binghamton, NY, USA ; Keahey, K., "Traffic-Sensitive Live Migration of Virtual Machines", Published in: Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on Date of Conference: 4-7 May 2015 Page(s): 51 – 60.

[5] Kejiang Ye ; Coll. of Comput. Sci., Zhejiang Univ., Hangzhou, China ; Xiaohong Jiang ; Ran Ma ; Fengxi Yan, "VC-Migration: Live Migration of Virtual Clusters in the Cloud", Published in: Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on Date of Conference:20-23 Sept. 2012 Page(s): 209 – 218.

[6] Kejiang Ye ; Coll. of Comput. Sci., Zhejiang Univ., Hangzhou, China ; Xiaohong Jiang ; Dawei Huang ; Jianhai Chen, "Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments", Published in: Cloud Computing (CLOUD), 2011 IEEE International Conference on Date of Conference: 4-9 July 2011 Page(s): 267 – 274.

[7] Jiao Zhang ; Dept. of Comput. Sci. & Technol., Tsinghua Univ., Beijing, China ; Fengyuan Ren ; Chuang Lin, "Delay guaranteed live migration of Virtual Machines", Published in: INFOCOM, 2014 Proceedings IEEE Date of Conference: April 27 2014-May 2 2014 Page(s): 574 – 582.

[8] Strunk, A. ; Dept. of Comput. Networks, Tech. Univ. of Dresden, Dresden, Germany, "A Lightweight Model for Estimating Energy Cost of Live Migration of Virtual Machines", Published in: Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on Date of Conference: June 28 2013-July 3 2013 Page(s): 510 – 517.

[9] Sarker, T.K. ; Sch. of Electr. Eng. & Comput. Sci., Queensland Univ. of Technol., Brisbane, QLD, Australia ; Maolin Tang, "Performance-driven live migration of multiple virtual machines in datacenters", Published in: Granular Computing (GrC), 2013 IEEE International Conference on Date of Conference: 13-15 Dec. 2013 Page(s): 253 – 258.

[10] Koto, A. ; Keio Univ., Yokohama, Japan ; Kono, K. ; Yamada, H., "A Guideline for Selecting Live Migration Policies and Implementations in Clouds", Published in: Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on Date of Conference: 15-18 Dec. 2014 Page(s): 226 – 233.

[11] Sisu Xi ; Dept. of Comput. Sci. & Eng., Washington Univ. in St. Louis, St. Louis, MO, USA ; Wilson, J. ; Chenyang Lu ; Gill, C., "RT-Xen: Towards real-time hypervisor scheduling in Xen", Published in: Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on Date of Conference: 9-14 Oct. 2011 Page(s): 39 – 48.

[12] Yun Chan Cho ; SungKyunKwan Univ., Suwon ; Jae Wook Jeon, "Sharing data between processes running on different domains in para-virtualized xen", Published in: Control, Automation and Systems, 2007. ICCAS '07. International Conference on Date of Conference: 17-20 Oct. 2007 Page(s): 1255 – 1260.

[13] Liu Fagui ; Sch. of Comput. Sci. & Eng., South China Univ. of Technol., Guangzhou, China ; Zhang Hao ; Zhou Haiyan, "A Xen-Based Secure Virtual Disk Access-Control Method", Published in: Multimedia Information Networking and Security (MINES), 2010 International Conference on Date of Conference: 4-6 Nov. 2010 Page(s): 375 – 378.

[14] Li Sun ; Sch. of Math. & Geospatial Sci., RMIT Univ., Melbourne, VIC ; Ebringer, T. ; Boztas, S., "An automatic anti-anti-VMware technique applicable for multi-stage packed malware", Published in: Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on Date of Conference: 7-8 Oct. 2008 Page(s): 17 – 23.

[15] Nikzad, A. ; Eng. & Comput. Sci, Concordia Univ., Montreal, QC, Canada ; Khendek, F. ; Toeroe, M., "OpenSAF and VMware from the perspective of high availability", Published in: Network and Service Management (CNSM), 2013 9th International Conference on Date of Conference: 14-18 Oct. 2013 Page(s): 324 - 331