Comparison between security majors in virtual machine and linux containers

Udit Gupta
Information Networking Institute
Carnegie Mellon University, Pittsburgh – Pennsylvania, USA
uditg@andrew.cmu.edu

Abstract:

Virtualization started to gain traction in the domain of information technology in the early 2000's when managing resource distribution was becoming an uphill task for developers. As a result, tools like VMWare, Hyper-V (hypervisor) started making inroads into the software repository on different operating systems. VMWare and Hyper-V could support multiple virtual machines running on them with each having their own isolated environment. Due to this isolation, the security aspects of virtual machines (VMs) did not differ much from that of physical machines (having a dedicated operating system on hardware). The advancement made in the domain of linux containers (LXC) has taken virtualization to an altogether different level where resource utilization by various applications has been further optimized. But the container security has assumed primary importance amongst the researchers today and this paper is inclined towards providing a brief overview about comparisons between security of container and VMs.

1. Introduction:

The concept of virtualization [1 and 2] was created with the sole intent of managing resource distribution efficiently and hence system resources started getting divided logically instead of physical division. Ever since the introduction of mainframe computers in 1960s by IBM, this technique of logical division has been of great utility. Apart from efficient resource distribution, virtualization has another advantage: it is easier to deal with systems at software level than working on hardware. Hardware virtualization has enabled users to deal with hardware more efficiently via logical connections where a hypervisor emulates a piece of computer hardware.

With the advent of cloud computing [3 and 4], virtualization has assumed primary importance for enterprises as well. It becomes easier to manage data centers across the world and thus it helps in working remotely. As functionality tends to increase for enterprise applications, there has been a disproportionate rise in the number of virtual machines (VMs) on each data center. Each hypervisor had an upper limit on the number of VMs that can run on it. Moreover, most of these applications did not require even half the resources allocated to VM by CPU. Thus, it was mandated with growing number of applications to run two to three times more server instances on a given server as compared to VMs. This lead to the development of linux containers (LXCs) [15 and 16] wherein only those resources will be used which are required by applications.

Although LXCs has been used efficiently in successfully executing various enterprise level applications, the security features of LXCs are yet to evolve when compared with VMs. Due to the isolation provided in VMs the threat of intrusions has been neutralized to a great extent via intrusion detection algorithms [5, 6, 7 and

8]. Furthermore, all the security aspects pertaining to detection of spam [9] and malware [10 and 11] which have been addressed in VMs are yet to remain addressed in containers. Although each LXC receives its own network stack and process space as well as its instance of file system, it doesn't have its own user namespace. As we proceed further we'll look at various container based technologies how security features varies across LXCs and VMs.

2. Container based technology:

Linux containers (LXCs) provide operating system-level virtualization and they have their own process and network space. Although work is still going on in creating a separate user namespace, the isolation provided in other aspects has been at par with virtual machines. Keeping in view the demand of growing number of instances required along with sandboxing, various container based technologies were designed to address this feature. The two primary container based technologies which we'll look at: docker and openVZ.

- **2.1 OpenVZ**: OpenVZ [17 and 18] is a container based technology which allows a physical server to run multiple instances of an operating system called containers or virtual private servers (VPS). Each container has its own network stack, serial ports, process tree and file system. In later versions of openVZ, work is being done to create container's own user space. It uses single patched linux kernel and can run only linux. It might be disadvantageous to use openVZ in case containers need different kernel version since all openVZ containers share the same architecture and kernel version. Also since it doesn't carry the overhead associated with operating system, it is much more efficient, scalable. Furthermore, there is dynamic memory allocation i.e. the memory allocated to 1 linux container can be used for other container as well without the requirement of rebooting the entire system.
- **2.2 Docker:** Docker [14] is different from openVZ in the sense that former sees a container as an application/service while latter sees container as a VPS. Since container is a single application in docker's terminology, it is important that interface between various containers needs to be robust since multiple containers might be required to run an application. For instance, in order to run a particular application, we might need configuration files from 1 container and database from another container. Thus, it is important to have secure and robust communication between two containers. Since each container has its own network stack, the secure communication between 2 containers can be achieved via any of the following protocols: SFTP [19], SSH [20], FTPS [21 and 22] and SCP.

Docker also has the capability of importing/exporting containers via access to the public registry. Moreover, docker defines an API for automating and customizing the creation and deployment of containers. It also has the capability of tracking and managing successive versions of a container, inspecting diff between versions, committing new versions, etc. Apart from this it has similar features compared to openVZ where it provides namespace, file system, network, resource isolation. Furthermore, docker also provides logging feature where the standard streams of each process container is collected and logged for real-time or batch retrieval. Lastly, docker provides an abstraction layer for containers so that they can run on different operating systems without any compatibility issues.

3. Comparison between containers based technology and virtual machines

Linux containers were designed with a single view of managing CPU resources distribution more efficiently. On any instance of VMWare [12] or Hyper-V, it is difficult to run more than 10 VMs due to the overhead incurred. Containers have resolved this problem to a great extent where they only make use of resources which are required by the application or service. Thus, more than 50 instances of container can run on single quad-core machine. Let's consider the example of any enterprise email security product: its main functionality would be to scan emails for spam/virus/malware, manage logs [13], manage message transfer agent (MTA) and report any datacenter outage in case the product is deployed on cloud network. In most cases, these functionalities described will not make any use of kernel data structures or operating system libraries or any related dependencies. Thus, rather than having VMs for each aspect of the product, it's better to containerize each feature by sandboxing them using docker/openVZ. In many of the organizations, VMs are emulated so as to perform feature testing which consumes hordes of memory space and CPU utilization. Having containers would ensure that redundancy would not have much impact on the resource consumption. Scalability would be easier since the time required for container installation is much less as compared to VMs.

On the other hand security aspect of docker/openVZ has been of concern lately. As isolation reduces, security is bound to decrease exponentially. Since containers share the same operating system and kernel, it is easier to gain access to containers especially as root user on linux. Although docker isolates many aspects of the underlying host from an application running in a container from an underlying host, the separation is not as strong as that of VM. Moreover, some applications might need to run on different operating system which is not possible in containerized technology.

Keeping in view the advantages and disadvantages of LXCs/docker, we need to arrive at some kind of trade off. The ideal situation would be to have few VMs installed on physical machine and then have many instances of container running on VM. This would ensure that security features of VM along with optimized features of LXCs would give maximum performance for a system. In case of cloud networks having data centers around the world, few VMs can be installed on a base machine on every datacenter followed by running multiple instances of LXCs on those VMs. Since resource consumption would reduce drastically, this solution would be cost effective for many organizations.

4. Conclusion:

The manner in which virtualization has dominated over the past decade goes on to show how important the role of linux based containers will be in future. This is evident from the fact that many organizations like Amazon, Microsoft have already adopted this technology in their cloud based products. Furthermore, the scalability which is being offered by the container technologies would ensure that cost savings would grow manifold in the near future. In this paper, a brief overview was provided about docker, openVZ: two container based technologies which are often seen as potential replacement for VM. In addition, a comparison was provided between LXCs and VMs from general as well security point of view.

References

- [1] Uhlig R, Neiger G, Rodgers D, Santoni AL, Martins FCM, Anderson AV, Bennett SM, Kagi A, Leung FH, Smith L, "Intel virtualization technology", IEEE Computer Society, Pages 48-56, May 2005
- [2] NM Mosharaf Kabir Chowdhurya, Raouf Boutaba, "A survey of network virtualization", Science direct magazine, Computer Networks, Volume 54, Issue 5, Pages 862-876, April 2010
- [3] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "A view of cloud computing", Communications of the ACM, Volume 53 Issue 4, Pages 50-58, April 2010
- [4] Simon Ostermann, Alexandria Iosup, Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, Dick Epema, "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing", pp 115-131, DOI 10.1007/978-3-642-12636-9 9, 2010
- [5] Animesh K Trivedi, Rajan Arora, Rishi Kapoor, Sudip Sanyal and Sugata Sanyal, "A Semi-distributed Reputation-based Intrusion Detection System for Mobile Ad hoc Networks", arXiv preprint arXiv: 1006.1956, 2010/6/10
- [6] Manoj Rameshchandra Thakur, Sugata Sanyal, "A Multi-Dimensional approach towards Intrusion Detection System", arXiv: 1205.2340v1 [cs.CR], 2012/5/10
- [7] Ajith Abraham, Ravi Jain, Sugata Sanyal, Sang Yong Han, "SCIDS: a soft computing intrusion detection system", Pages 252-257, Book Distributed Computing-IWDC 2004, 2005/1/1
- [8] Animesh Kr. Trivedi, Rishi Kapoor, Rajan Arora, Sudip Sanyal, Sugata Sanyal, "RISM Reputation Based Intrusion Detection System for Mobile Ad hoc Networks", Third International Conference on Computers and Devices for Communications CODEC-06, pages 234-237, 2006
- [9] Zoltán Gyöngyi, Hector Garcia-Molina, Jan Pedersen, "Combating web spam with trustrank", VLDB '04 Proceedings of the Thirtieth international conference on Very large data bases Volume 30 , Pages 576-587, 2004
- [10] Christodorescu M, Jha, S, Seshia SA, Song D, Bryant RE, "Semantics-aware malware detection", Security and Privacy, IEEE Symposium, Pages 32 46, IEEE, May 2005
- [11] Manoj Rameshchandra Thakur, Divye Raj Khilnani, Kushagra Gupta, Sandeep Jain, Vineet Agarwal, Suneeta Sane, Sugata Sanyal, Prabhakar S. Dhekne, "Detection and Prevention of Botnets and malware in an enterprise network", International Journal of Mobile and Wireless Computing, Inderscience, http://arxiv.org/pdf/1312.1629, Volume 5, Issue 2, 2012
- [12] CA Waldspurger, "Memory resource management in VMware ESX server", ACM SIGOPS Operating Systems Review OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, Volume 36, Pages 181-194, 2002

- [13] Van der Aalst W, Weijters T, Maruster L, "Workflow mining: discovering process models from event logs", IEEE Transactions on Knowledge and Data Engineering (TKDE), Pages 1128 1142, Volume 16, Issue 9, IEEE, Sept. 2004
- [14] Dirk Merkel, "Docker: lightweight Linux containers for consistent development and deployment", Linux journal, Volume 2014, Issue 239, ACM, March 2014
- [15] Xavier MG, Neves MV, Rossi FD, Ferreto TC, Lange T, De Rose CAF, "Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments", Pages 233-240, ISSN 1066-6192, IEEE, March 2013
- [16] Bardac M, Deaconescu R, Florea AM, "Scaling Peer-to-Peer testing using Linux Containers", Roedunet International Conference (RoEduNet), Pages 287-292, ISSN 2068-1038, IEEE, June 2010
- [17] Yuhao Zheng, David M Nicol, "A Virtual Time System for OpenVZ-Based Network Emulations", Proceeding PADS '11 Proceedings of the 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation, Pages 1-10, IEEE, 2011
- [18] Jianhua Che, Yong Yu, Congcong Shi, Weimin Lin, "A Synthetical Performance Evaluation of OpenVZ, Xen and KVM", Services Computing Conference (APSCC), Pages 587-594, IEEE, Dec. 2010
- [19] Liu Xia, Feng Chao-sheng, Yuan Ding, Wang Can, "Design of secure FTP system", Communications, Circuits and Systems (ICCCAS), Pages 270-273, IEEE, July 2010
- [20] T Ylonen, "SSH secure login connections over the internet", Proceedings of the 6th USENIX Security Symposium, July 22-25, 1996, San Jose, California, USA https://www.usenix.org/legacy/publications/library/proceedings/sec96/full_papers/ylonen/
- [21] Anish Bhimani, "Securing the commercial Internet", Communications of the ACM, Volume 39, Issue 6, Pages 29-35, ACM, June 1996
- [22] Apostolopoulos G, Peris V, Pradhan P, Saha D, "Securing electronic commerce: reducing the SSL overhead", Network, Pages 8-16, ISSN 0890-8044, IEEE, August 2002