Design and analysis security architecture virtualization OpenVz

Indrawat Wardhana*, Musi Ariawijaya, Riska Hasnur, Rita Syafitri, and Akhmad Nasuha
UIN Sulthan Thaha Saifuddin Jambi, Indonesia

*indrawataw@uinjambi.ac.id

Abstract. OpenVZ is a capsule-based virtualization technology for OS Linux which allows administrators to deploy multiple Operating Systems with different virtual hardware specifications, called containers, virtual environments, or Virtual Private Servers. In this paper, we propose new security architecture for OpenVZ depends on the type of attacks that commonly happen in servers. This security technic called OpenVzSec. The server is attacked by the client using OS windows and Ubuntu OS which is equipped with an attacker code program based on python language. Type of attacks used in this research: SSH vulnerability, SYN Flood attacks, Attack on the Rootkit vulnerability, and checksum spoofing. Some server attacks on a server and containers able to detected and anticipate by OpenVZSec. The OpenVZSec security model does not decrease performance of the server.

1. Introduction
Server virtualization has many advantages, which are reduce administrative costs, flexibility, simplicity, server consolidation, and fast recovery processes [1]. Using virtual server, the costs that are supposed to buy hardware can be reduced costs by 74% [2]. Even with low and cheap hardware, the server performance is still good and has high performance [3].

OpenVZ is open-source virtualization technology. Built on the Linux operating system. It can perform its function perfectly [4], the flow of packet data traffic in a virtualized network is also in accordance with the network architecture created. Open VZ performance based on test tests shows that OpenVZ near the original performance of the base system [5]. OpenVz is also used for cloud storage applications [6].

The results of [7] observation on a virtual private server which is based on Centos (Linux) and Windows found that Linux has a higher resistance than Windows. The test was carried out by flooding the two VPSs with maximum flooding, where CentOS was better than Windows server in web server response time. Virtualization security is one of the problems. Based on research [9] found that from 210 accounts tested, 14.7% were successfully cracked with a dictionary attack technique and 13.8% were cracked by brute force. The dictionary attack technique is better than the brute force technique regarding the effectiveness and reliability of the penetration test [9]. It is known that 90% of DoS attacks use TCP [10]. OpenVZ security is very important because it is virtualized better than KVM [11] and is widely used by users where users want security and system comfort. OpenVZ was recommended by developers for stability, features and security [12].
Virtual Machine security services have been provided by several providers. The main problem is the high cost of this service. Virtual HSM \cite{13} provides a virtualization security solution based on sensitive data encapsulation techniques and cryptographic operations in an isolated virtual environment. Flavio Lombardi with ACPS and KVMSec architectures that have been tested on Eucalyptus, openECP and KVM can run optimally on the host server \cite{14,15}.

2. Experimental
The server that was used in this experiment: Intel Pentium Core i5 with 8 Gb RAM, 1 IP public. OpenVZ was installed in Linux Debian 7. Virtual Machine was deployed based on ID, storage capacity, private IP, IP public, and nameserver. Moreover, this research tries to build new security architecture for OpenVZ based on the most common server. The simulation attacker used operating system windows with VirtualBox OS. The framework system used python language, bash script, SQLite database with Linux OS. Attacker attack that was used in this research: SSH vulnerability, SYN Flood, Rootkit vulnerability, and checksum file.

The research divided into 3 steps:
- Installation: configuration, deploy VM and save checksum file
- Build attacker script
- Detection by OpenVZSec framework

![Flow chart of OpenVZ Security](image)

**Figure 1.** Flow chart of OpenVZ Security
3. Results and Discussion
The OpenVZ Security framework is built on the main Operating System. This framework can access all layers contained in the main Operating System, namely debian 7. Three vps are deploy in OpenVZ. Access is given to OpenVZSec to detect all activities from the VPS that has been deployed.

![Figure 2. OpenVZSec Architecture on Server](image)

3.1. SSH Attack
Port 22 as default port SSH was used in this simulation. Furthermore, simulation starts with script sshattack.py. This script attacks IP VM Host and it was used continuously until login success. User and password were taken from file CSV using a random combination. The attacker exploits the variability of VM clients that not supported by the firewall system. Framework OpenVZ security will discover every request from ssh port, after that running some action to analyze continuously login access from one IP. The IP address that requests login will be added to the log system. If fault requesting more than 10 times, the framework automatically sends the command to Iptables to block IP addresses.

![Figure 3. Framework OpenVZSec block SSH attacker](image)
The actuator will recognize attack after attacker running SSH brute-force. Then, it identifies IP address attacker then added that address into IP blacklist storage. Furthermore, actuator requesting Iptables blocking that address. In this simulation, there are four times login attempts via port 22. Iptables will drop IP that tries entering.

```bash
/usr/sbin/iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --set
/usr/sbin/iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --update -seconds 60 --hitcount 4 -j DROP
```

### 3.2. SYNC Flooding

SyN packet makes a new connection to the target server. The server will allocate several memories for handling the next connection, sending back an acknowledgment, then waiting for the client to finish the connection then start sending data. On the other hand, the attacker sends a large SYN request that makes memory on the server full and need waiting several times until data is coming. When memory is loaded full, the server can not handle requests from the real client.

In this scenario, the scapy module was used for running a syn flood attack. The attacker performed on the web using port 80 (HTTP) and port 363(MySQL). Client / computer attacker with IP 115.xxx.xxx.xxx sending a data packet via port 1234. Then server with IP 180.23x.xxx.xxx sending SYN+ACK via port 80. Format syn packet:

```python
pkt = IP ( dst="180.23x.xxx.xxx", id=1111, ttl=99)/ TCP ( sport=1234, dport=80, seq=12345, ack=1000, window=1000, flags="S")
```

The second attack was performed by sending a large syn request from a different port, using RandShort() for generating a unique port number. Moreover, the port to attack: 22,80,336. The attacker carried out over interval 0.2 seconds and timeout 6.

```python
pkt = IP ( dst="180.x.x.x", id=1111, ttl=99)/ TCP ( sport=RandShort(), dport=[22,80,3636], seq=12345, ack=1000, window=1000, flags="S")
```

The attack that occurs in the server is detected by OpenVZSec, which is blocking the IP address of the suspected attacker. The connection will be cut automatically. OpenVZSec will deny all suspicious connections from TCP with TCP RSP packets, as well as prevent any potential Distribution Denial of Service. All connections in the SYN_RECV state will be closed with the RST packet.

In this study, we tested 25 attacks using one IP address. If the incoming attempts are continuous from the same IP address, the SYN packet will be rejected.
3.3. Rootkit Attack

This type of attack applies to the server or virtual machine clients that have been compromised with various login techniques. The attacker has put some files which can remotely take over the client-server / VM using the rootkit techniques. In this research, a modified file is placed in the VM client folder to run the rootkit command.

The server has been installed by OpenVZSec will automatically give orders to the actuator to detect the entire file and system for possible rootkits. Rootkits that have been planted in the modified files will be detected by the actuator as suspicious files. The actuator will then perform a checksum comparison of the previous checksum. Rootkit detection was carried out on more than 60 types of rootkits from the data in the database. The detected rootkits will be deleted by the actuator. The detected files will be included in the deletion log. Moreover, OpenVZSec restarts the server to make sure that the system VM client running well.

It can be seen that, maximum server load when running rootkit detector on OpenVZSec only 13 percent of processors works. Working time takes 5 seconds for finishing detection and delete the suspicious file.

Figure 4. Attacker has been detected and blocked by framework OpenVZSec
3.4. **Checksum file changed**

The attack is detected by viewing the hash file, auth log. The first method looks at attacks by looking at authentication logs: Logs are viewed in logs var/log/auth.log for the first 100 lists. The attack usually brutes repeatedly with the same IP. Through IP detection, the IP address can be blocked then it is added into the IP blacklist.

The second method is looking at the kernel hash image file: The kernel image file is rechecked where the hash file which will then be compared with the kernel hash image file stored in the SQLite database. The comparison will check the checksum match, date created, including the size of the kernel image file.

The attack is carried out after entering the server and VM client via SSH brute-force. Then, the OpenVZ Image OS template file is changed by the intruder, recreated. The image has consequences switch over to new checksum. This is mean the new file is not identical anymore with the old one.

### 4. Conclusion

The design and analysis of the OpenVZ security architecture have been successfully created. It is found that attacks using the ssh technique via ports 22 and 3306 on the server can be detected and blocked by OpenVZSec. Syn flood transmissions are successfully rejected and the suspicious IP address is included in the IP blacklist. Infiltration of rootkit files on VPS has been detected and deleted from the main system. Some server attacks on a server and containers able to be detected and anticipate by OpenVZSec. The OpenVZSec security model does not decrease the performance of the server.
References

[1] E. Wamy, F. Machfudz, A. Heriady, and A. A. Ilham, “Perancangan Aplikasi Server Berbasis Virtualisasi,” *Semin. Nas. Tek. Inform.*, vol. 2006, no. 2, pp. 34–36, 2013.

[2] S. M. Isa, “Penerapan Konsolidasi Server Menggunakan Teknologi Virtualisasi,” *Semin. Nas. Apl. Teknol. Inf.*, vol. 2006, no. Snati, 2006.

[3] T. Prasandy and W. Adhiwibowo, “Virtualisasi Server Sederhana Menggunakan Proxmox,” *J. Transform.*, vol. 12, no. 2, p. 37, 2015.

[4] T. I. Bayu, I. R. Widiasari, D. W. Chandra, and S. Kom, “Penerapan Teknologi Virtual isasi Tingkat Sistem Operasi Pada Server Linux Ubuntu 8.04 Menggunakan OpenVZ JURNAL Oleh: Teguh Indra Bayu NIM: 672004054 Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salat,” no. September, 2013.

[5] J. P. Walters, V. Chaudhary, C. Minsuk, S. Guercio, and S. Gallo, “A comparison of virtualization technologies for HPC,” *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, no. January, pp. 861–868, 2008.

[6] S. A. Indrawata Wardhana, “Perancangan dan Penerapan Arsitektur Cloud Storage Pada Iain STS Jambi,” *Manaj. Sist. Inf.*, vol. 2, no. 1, pp. 244–259, 2017.

[7] R. S. Wirawan and Teknik, “Analisa Kinerja Dan Keamanan VPS Pada Operating System Centos Dan Windows Server,” *Univ. Muhammadiyah Surakarta.,* 2013.

[8] E. Hakim, L, Affandi, A., & Setijadi, “Perbandingan Teknik Penetration Tes: Brute Force Dan Dictionary Attack – Publikasi MMT-ITS,” *MMT ITS*, 2015. [Online]. Available: https://mmt.its.ac.id/publikasi/perbandingan-teknik-penetration-tes-brute-force-dan-dictionary-attack/. [Accessed: 12-Sep-2020].

[9] L. Bosnjak, J. Sres, and B. Brumen, “Brute-force and dictionary attack on hashed real-world passwords,” *2018 41st Int. Conf. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc.*, no. February, pp. 1161–1166, 2018.

[10] H. Wang, D. Zhang, and K. G. Shin, “Detecting SYN flooding attacks,” *Proc. - IEEE INFOCOM*, vol. 3, no. March, pp. 1530–1539, 2002.

[11] P. R. M. Vasconcelos, G. Azvedo A. Freitas, and T. G. Marques, “KVM, OpenVZ and Linux Containers: Performance Comparison of Virtualization for Web Conferencing Systems,” *Int. J. Multimed. Image Process.*, vol. 6, no. 1/2, 2016.

[12] K. Bernsmed and S. Fischer-Hübner, “Security of OS-Level Virtualization Technologies,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8788, no. July, 2014.

[13] D. Kartashov, “Virtual HSM implementation in OpenVZ containers,” vol. 2014, pp. 1–15, 2014.

[14] F. Lombardi and R. Di Pietro, “KvmSec: A security extension for Linux kernel virtual machines,” *Proc. ACM Symp. Appl. Comput.*, no. May 2014, pp. 2029–2034, 2009.

[15] F. Lombardi and R. Di Pietro, “Secure virtualization for cloud computing,” *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1113–1122, 2011.