Volume 14 No 1 2024, 61-68
http://dx.doi.org/10.20473/jpua.v14i1.2024.61-68

JPUA
Media Informasi dan Komunikasi Kepustakawanan

Jurnal Perpustakaan Universitas Airlangga: Media Informasi dan Komunikasi Kepustakawanan
https://e-journal.unair.ac.id/JPERPUS

STUDY AND ANALYSIS OF DOCKER INTERNAL SYSTEM SECURITY FROM DOCKER DAEMON ATTACK AND DDOS ON THE OPEN JOURNAL SYSTEM

STUDI DAN ANALISIS KEAMANAN SISTEM INTERNAL DOCKER DARI DOCKER DAEMON ATTACK DAN DDOS PADA SISTEM OPEN JOURNAL SYSTEM

Literature Studi
Studi Literatur

Choirul Rio Prabowo**, Dodo Irmanto*, Erfan Rohadi**
* Politeknik Angkatan Darat, Malang, Indonesia
**Politeknik Negeri Malang, Malang, Indonesia

ABSTRACT

Introduction: The growing use of data centers encourages virtualization technology to become an alternative solution in virtualization to provide a dense environment that can be adjusted according to needs. In hypervisor-based virtualization technology in managing servers in the Open Journal System data center, network administrators must allocate resources that are large enough so that when developing web or mobile systems takes a long time, and in hypervisor-based virtualization techniques, they must have access to the host kernel.
Purpose: Research on the attack model and vulnerabilities of the Docker internal security system on the Open Journal System from the Docker daemon attack and DDoS attacks when building and managing the Docker internal system.
Method: This research begins with a literature study, network scope, system design, and system implementation based on plans that have been made, as well as testing, analysis, and concluding the tests that have been carried out.
Finding: At the testing stage, the results obtained were the success of the Docker system in handling DDoS attacks and the success of securing the Docker Daemon from Docker Daemon attacks.
Conclusion: The Docker Daemon Attack can occur by misconfiguring containers. This flaw allows an unauthorised party to take control of a container that has already been created. By gaining root access, attackers can perform various malicious activities within the container. Therefore, it is important to have an understanding and implementation of proper security practices in the management and configuration of Docker containers to reduce the risk of these types of attacks.
Keyword: Docker, Container, DDoS, Attack.

INFO ARTICLE

Received: 1 June 2024
Accepted: 26 June 2024
Published: 28 June 2024

Correspondence:
Name: Choirul Rio Prabowo
Email: choirul.rio.p@gmail.co

How to cite this article:
Prabowo, C. R., Rohadi, E. ., & Irmanto, D. . (2024). Study and Analysis of Docker Internal System Security From Docker Daemon Attack and DDOS on The Open Journal System. JPUA: Jurnal Perpustakaan Universitas Airlangga: Media Informasi Dan Komunikasi Kepustakawanan, 14(1), 61–67. https://doi.org/10.20473/jpua.v14i1.2024.61-68

A B S T R A K

Pendahuluan: Berkembangnya penggunaan data center ini mendorong teknologi virtualisasi menjadi salah satu alternatif solusi dalam virtualisasi untuk menyediakan lingkungan yang padat agar dapat disesuaikan sesuai kebutuhan. Pada teknologi virtualisasi berbasis hypervisor dalam pengelolaan server pada data Open Journal System, center administrator jaringan harus mengalokasikan resources yang cukup besar, sehingga ketika dilakukan development sistem web atau mobile membutuhkan waktu yang lama, serta pada teknik virtualisasi berbasis hypervisor harus memiliki akses ke kernel host.
Tujuan: Meneliti model serangan dan kerentanan sistem keamanan internal Docker pada *Open Journal System* dari ancaman *Docker Daemon Attack* dan serangan DDoS pada saat membangun dan mengelola sistem internal Docker.
Metode Penelitian: Penelitian ini dimulai dengan studi literatur, ruang lingkup jaringan, perancangan sistem, implementasi sistem berdasarkan rancangan yang telah dibuat, serta pengujian, analisis dan penarikan kesimpulan dari pengujian yang telah dilakukan.
Hasil Penelitian: Pada tahap pengujian dilakukan hasil yang didapatkan yaitu keberhasilan sistem docker dalam menangani serangan DdoS serta keberhasilan pengamanan docker daemon dari serangan docker daemon attack.
Kesimpulan: Serangan Docker Daemon dapat terjadi karena kesalahan konfigurasi container. Kelemahan ini memungkinkan pihak yang tidak berwenang untuk mengambil kendali atas container yang telah dibuat. Dengan mendapatkan akses root, penyerang dapat melakukan berbagai aktivitas berbahaya di dalam container. Oleh karena itu, penting untuk memiliki pemahaman dan penerapan praktik keamanan yang tepat dalam pengelolaan dan konfigurasi container Docker untuk mengurangi risiko jenis serangan ini.
Kata Kunci: Docker, Container, DDoS, Attack.

PENDAHULUAN

Semakin berkembangnya teknologi informasi saat ini mempengaruhi aspek dalam lingkungan data center untuk melakukan development dan production system web maupun mobile. Berkembangnya penggunaan data center ini mendorong teknologi virtualisasi menjadi salah satu alternatif solusi dalam virtualisasi untuk menyediakan lingkungan yang padat agar dapat disesuaikan sesuai kebutuhan (Shameem Ahamed et al., 2021). Dalam teknik virtualisasi dibagi menjadi dua yaitu berbasis hypervisor dan berbasis container. Hypervisor merupakan sebuah teknologi virtualisasi yang menjadi landasan agar berbagai sistem dapat berjalan secara bersamaan pada sebuah mesin. Sedangkan container merupakan sebuah teknologi virtualisasi yang melakukan isolasi pada tingkat sistem operasi (Alauddin et al., 2017).

Pada teknologi virtualisasi berbasis hypervisor dalam pengelolaan server pada data Open Journal System center administrator jaringan harus mengalokasikan resources yang cukup besar sehingga ketika dilakukan development system web atau mobile membutuhkan waktu yang lama, serta pada teknik virtualisasi berbasis hypervisor harus memiliki akses ke kernel host. Sedangkan pada virtualisasi berbasis container dalam melakukan alokasi resource pada sistem operasi host tidak membutuhkan banyak resource. Hal ini dikarenakan virtualisasi berbasis container memiliki sifat fleksibel dan scalable (Fiddin et al., 2018).

TINJAUAN PUSTAKA

Salah satu platform yang menawarkan teknologi berbasis container adalah Docker. Secara teori Docker yang menawarkan teknologi virtualisasi berbasis container dianggap lebih bagus dari segi performa, keamanan, dan kehandalan dibandingkan virtualisasi berbasis hypervisor seperti virtual box, VM Ware yang akan digunakan pada Open Journal System (Sulastri Apridayanti et al., 2018).

Berdasarkan latar belakang dan permasalahan diatas, maka Penulis bermaksud melakukan penelitian terhadap model serangan dan kerentanan sistem keamanan internal Docker pada Open Journal System dari ancaman Docker Daemon Attack dan serangan DDoS pada saat membangun dan mengelolah sistem internal Docker.

METODE PENELITIAN

Pada bagian ini menjelaskan metodologi penelitian yang bersifat implementatif. Peneliti akan melakukan implementasi sistem Docker dengan sistem OJS. Penelitian ini dimulai dengan studi literatur, ruang lingkup jaringan, perancangan sistem, implementasi sistem berdasarkan rancangan yang telah dibuat, serta pengujian, analisis dan penarikan kesimpulan dari pengujian yang telah dilakukan.

A. Identifikasi Masalah

Perancangan Identifikasi masalah dalam penelitian ini adalah bagaimana melakukan implementasi sistem Docker dan OJS. Setelah sistem Docker dan OJS telah selesai diimplementasikan, selanjutnya bagaimana cara untuk mengamankan sistem Docker Daemon dari *Docker Daemon Attack* dan serangan DDoS *Units.*

B. Studi Literatur

Studi literatur diperlukan untuk mendalami permasalahan yang akan diteliti dan menambah pengetahuan yang diperlukan dalam melakukan penelitian dan penulisan tesis. Dalam hal ini maka peneliti memerlukan informasi dan referensi mengenai dasar Docker, Docker Bench, dan lain-lain.

C. Analisis Kebutuhan Sistem

Kebutuhan Fungsional

Kebutuhan fungsional yang dibutuhkan pada penelitian ini adalah:

1. Docker dapat menjalankan *container* berisi sistem OJS
2. Docker Bench sebagai alat pengujian keamanan untuk serangan *Docker Daemon Attack*
3. *Web Server Stress Tool* untuk melakukan Pengujian QoS

Kebutuhan Non Fungsional.

Dalam penelitian ini, Docker dan Sistem OJS akan dibangun menggunakan *Rack Server* dengan spesifikasi sebagai berikut ini:

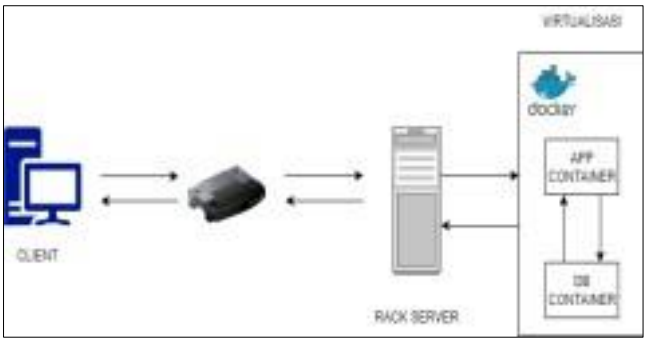
Tabel 1. Spesifikasi PC *Server*

Processor	Intel Xeon Silver 4214, 2.20 GHz
Memory	32 GB
Disk	500GB
OS	Ubuntu 20.04

D. Perancangan Arsitektur Sistem

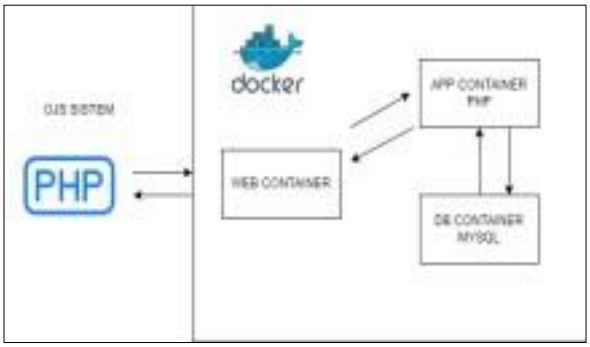
Sistem akan diterapkan pada jaringan dengan bandwith unlimited. Sistem dibuat dengan arsitektur client-server. Dimana Rack Server bertindak sebagai server terpusat yang menyimpan data dan Docker bertindak sebagai virtual sistem(Bik & Asmunin, 2017).

Langkah selanjutnya client akan mengakses virtual sistem Docker melalui Rack Server yang sudah terinstall Docker.



Gambar 1.Perancangan Arsitektur Sistem

E. Perancangan Topologi Sistem



Gambar 2.Perancangan Topologi Sistem

Merupakan detail perancangan arsitektur pada mesin virtual yang berjalan sebagai Docker. Docker disini akan menjalankan aplikasi OJS untuk membangun sebuah layanan *web repository* jurnal, yang dapat diakses *user* untuk melakukan *upload* jurnal dan menyimpan jurnal. Sistem OJS dibangun dengan bahasa pemrograman PHP dan Basis data MySQL. Mesin virtual pada *web container* akan menjadi pusat *container* yang akan melakukan semua manajemen pada sistem Docker dan *app container* berisi PHP dan MySQL yang akan menjadi *worker* (Hung et al., 2016).

HASIL DAN PEMBAHASAN

Setelah melakukan perancangan sistem, implementasi dilakukan sesuai dengan perancangan sistem yang dibuat. Implementasi ini berupa langkah-langkah dalam membangun sistem OJS pada Docker serta melakukan pengamanan pada Docker Daemon dan pengamanan terhadap serangan DDoS. Tahap pengujian sistem guna mengetahui permasalahan-permasalahan yang mungkin muncul diantara lain pengujian:

A. Keberhasilan Sistem Docker dalam menangani Serangan DDoS.

Langkah langkah proses Simulasi DDoS pada Docker:

1. Masuk ke Ubuntu-Docker pada virtual box
2. Masukkan *Username*: Ubuntu dan *Password*: xxxx
3. Setelah berhasil *Log In*, Lakukan instalasi Host sFlow dengan cara ketik:

```
# docker pull sflow/host-sflow
```

4. Host sFlow merupakan aplikasi yang digunakan untuk melakukan *monitoring* proses CPU, RAM, dan lain-lain untuk melihat apakah proses DDoS berhasil atau tidak. Untuk menjalankan aplikasi ini ketik perintah berikut ini:

```
Docker run --rm -d -e
"COLLECTOR=host.Docker.internal" -e
                "SAMPLING=10" \
        --net=host -v
/var/run/Docker.sock:/var/run/Docker.soc k:ro \
    --name=host-sflow sflow/host sflow
```

5. Menjalankan ExaBGP, disini ExaBGP akan terkoneksi dengan sflow-RT untuk melakukan kontrol *traffic* serangan DDoS yang berjalan. Untuk menjalankan ExaBGP ketik perintah:

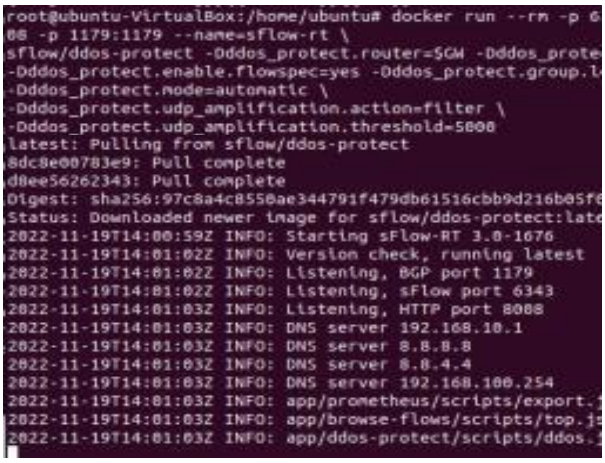
```
Docker run –rm sflow/exabgp
```

6. Untuk setting jaringan menjalankan sFlow Monitoring tool

```
GW=`Docker network inspect bridge -f
'{{range
.IPAM.Config}}{{.Gateway}}{{end}}'` SUBNET=`Docker network inspect bridge -f
'{{range
.IPAM.Config}}{{.Subnet}}{{end}}'`
```

7. Aplikasi sFlow *Monitoring* dengan mengetik perintah:

```
Docker run --rm -p 6343:6343/udp -p
8008:8008 -p 1179:1179 --name=sflow-rt \
sflow/DDoS-protect-
DDDoS_protect.router=$GW
- DDDoS_protect.as=65001 \
-DDDoS_protect.enable.flowspec=yes
- DDDoS_protect.group.local=$SUBNET \
-DDDoS_protect.mode=automatic \
-DDDoS_protect.udp_amplification.action=filter \
- DDDoS_protect.udp_amplification.threshold=5000
```



Gambar 3. Aplikasi sFlow Monitoring

Jika muncul informasi seperti diatas, maka menandakan aplikasi telah berjalan.

8. Selanjutnya untuk masuk ke *dashboard monitoring* ketik perintah: http://localhost:8008 , jika berhasil akan muncul seperti gambar di bawah ini:

Gambar 4. Dashboard Monitoring

9. Setelah proses instalasi sFlow *monitoring* berhasil, lanjutkan untuk melakukan instalasi hping3. Disini aplikasi hping3 digunakan untuk melakukan *generate* simulasi DDoS *Attack* (Baset et al., 2016). Sebelum instalasi dan menjalankan aplikasi hping3. Silahkan lakukan *setting* jaringan pada Docker dengan mengetik perintah dibawah lalu tekan *enter*:

   GW=`Docker network inspect bridge -f

   '{{range

   .IPAM.Config}}{{.Gateway}}{{end}}'`

10. Selanjutnya lakukan instalasi dan menjalankan hping3 dan *generate* simulasi DDoS dengan mengetik perintah:

    Docker run --rm sflow/hping3 --flood -- udp - k -a 198.51.100.1 -s 53 $GW

Proses DDoS Berjalan:

HASIL PENGUJIAN

Gambar 5 . Simulasi DDos sukses

B. Keberhasilan Pengamanan Docker Daemon dari serangan Docker Daemon Attack.

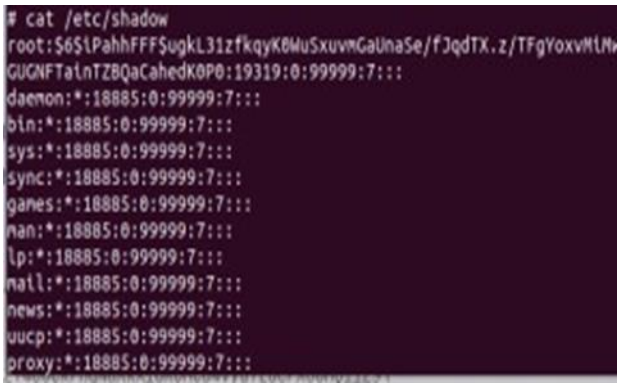
1. Pengujian dilakukan menggunakan Docker Bench menggunakan tools makemeroot yang dibuat oleh hysnsec (Krochmalski, 2017). Untuk menjalankan container tersebut ketik perintah:

Docker run – v /:/host –it hysnsec/makemeroot

2. Berikut hasil ketika *container* telah berhasil dijalankan

3. Untuk dapat melihat layanan apa yang mengalami kerentanan ketika dilakukan penyerangan *Docker Daemon Attack* ketik:

cat /etc/shadow



4. Pada percobaan kali ini penulis mencoba kerentanan pada *Docker Daemon Attack* untuk mengubah *password root* pada Docker *container.*

Maka pengamanan *Docker Daemon Attack* berhasil dijalankan.

KESIMPULAN

Serangan Docker Daemon Attack terjadi karena misconfigurasi pada container. Hal ini menyebabkan orang yang tidak bertanggung jawab dapat mengambil alih container yang sebelumnya kita buat. Ketika penyerang mendapatkan akses root ini, penyerang dapat melakukan apa saja di container mulai dari mengubah password, memasang malware, dan lain-lain yang dapat membahayakan container maupun memberatkan kinerja container. Bukti Ketika Docker Container mengalami penyerangan Docker Daemon Attack, untuk melakukan pengecekan apakah Docker mengalami kerentanan pada Docker Daemon.

DAFTAR PUSTAKA

Alauddin, M. F., Ijtihadie, R. M., & Husni, M. (2017). Implementasi Virtual Data Center Menggungakan Linux Container Berbasis Docker dan SDN. *Jurnal Teknik ITS*, *6*(2), 6–8. https://doi.org/10.12962/j23373539.v6i2.23755

Baset, S., Berger, S., Bottomley, J., Isci, C., Nagaratnam, N., Pendarakis, D., Rao, J. R., Steinder, G., & Ramanatham, J. (2016). *IBM Research Report Docker and Container Security White Paper Docker and Container Security White Paper*. https://dominoweb.draco.res.ibm.com/reports/rc25625.pdf

Bik, F. R., & Asmunin. (2017). IMPLEMENTASI DOCKER UNTUK PENGELOLAAN BANYAK APLIKASI WEB (Studi Kasus : Jurusan Teknik Informatika UNESA). *Jurnal Manajemen Informatika*, *7*(2), 46–50.

Fiddin, C., Munadi, R., & Mayasari, R. (2018). Analisis Performansi Virtualisasi Container Menggunakan Docker Dibawah Serangan Networked Denial of Service. *E-Proceeding of Engineering*, *5*(1), 281–290.

Hung, L. H., Kristiyanto, D., Lee, S. B., & Yeung, K. Y. (2016). GUIdock: Using Docker containers with a common graphics user interface to address the reproducibility of research. *PLoS ONE*, *11*(4), 1–14. https://doi.org/10.1371/journal.pone.0152686

Krochmalski, J. (2017). Docker and Kubernetes for Java Developers. In *Jurnal Sains dan Seni ITS*. Packt Publishing Ltd. http://repositorio.unan.edu.ni/2986/1/5624.pdf%0Ahttp://fiskal.kemenkeu.go.id/ejournal%0Ahttp://dx.doi.org/10.1016/j.cirp.2016.06.001%0Ahttp://dx.doi.org/10.1016/j.powtec.2016.12.055%0Ahttps://doi.org/10.1016/j.ijfatigue.2019.02.006%0Ahttps://doi.org/10.1

Shameem Ahamed, W. S., Zavarsky, P., & Swar, B. (2021). Security Audit of Docker Container Images in Cloud Architecture. *ICSCCC 2021 - International Conference on Secure Cyber Computing and Communications*, *April*, 202–207. https://doi.org/10.1109/ICSCCC51823.2021.9478100

Sulastri Apridayanti, Isnawaty, & Rizal Adi Saputra. (2018). Desain Dan Implementasi Virtualisasi Berbasis Docker Untuk Deployment Aplkasi Web. *SemanTIK*, *4*(2), 37–46.

How to cite this article:
Prabowo, C. R., Rohadi, E. & Irmanto, D. (2024). Study and Analysis of Docker Internal System Security From Docker Daemon Attack and DDOS on The Open Journal System. *JPUA: Jurnal Perpustakaan Universitas Airlangga: Media Informasi dan Komunikasi Kepustakawanan*, 14(1), 61–67. https://doi.org/10.20473/jpua.v14i1.2024.61-67