ANALISA KERENTANAN PADA *VULNERABLE DOCKER* MENGGUNAKAN ALIENVAULT DAN DOCKER BENCH FOR SECURITY DENGAN ACUAN *FRAMEWORK* CIS CONTROL

Fatin Hanifah[1], Avon Budiyono[1], Adityas Widjajarto[2]

[1,2,3] Universitas Telkom, Bandung
fatinhanifah@student.telkomuniversity.ac.id[1], **avonbudi@telkomuniversity.ac.id[2]**,
adtwjrt@telkomuniversity.ac.id[3]

Abstrak

Docker merupakan sebuah *project open-source* yang menyediakan *platform* terbuka untuk *developer* maupun *sysadmin* untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun sebagai sebuah kontainer yang ringan. Penelitian ini menggunakan *Vulnerable Docker* yang merupakan sebuah Virtual Machine berisi Docker yang rentan yang dibuat oleh perusahaan *NotSoSecure*, perusahaan yang berfokus pada keamanan komputer. Penelitian ini bertujuan untuk menguji secara empiris mengenai analisis kerentanan pada *Vulnerable Docker* menggunakan *vulnerability scanner* dengan mengacu pada *framework* CIS Control. Tujuan penelitian tersebut dapat dicapai dengan melakukan eksperimen berupa mengumpulkan data yang diperlukan, menganalisis hasil eksploitasi yang telah dilakukan pada *Vulnerable Docker* dan yang terakhir yaitu membuat kesimpulan berupa laporan yang disesuaikan dengan *framework* CIS Control. *Vulnerability Scanner* yang digunakan pada penelitian kali ini yaitu AlienVault OSSIM dan Docker Bench for Security. Pada penelitian kali ini ditemukan risiko tertinggi pada level aplikasi yaitu risiko yang diakibatkan oleh *vulnerability* jenis *WordPress User IDs and User Names Disclosure* melalui *scanning* AlienVault. Dan pada level sistem diakibatkan oleh *vulnerability* jenis *Enable User Namespace Support*. Terdapat 6 kontrol pada CIS Control V8 yang digunakan untuk mengurangi risiko yang terjadi pada penelitian ini.

Kata Kunci: ***docker, vulnerability, alienvault, cis control***

Abstract

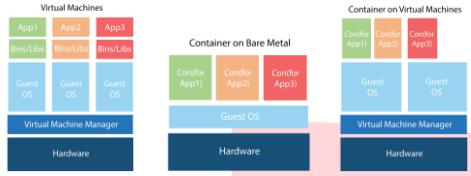
Docker is an open-source project that provides an open platform for developers and sysadmins to be able to build, package, and run applications anywhere as a lightweight container. This study uses Vulnerable Docker, a Virtual Machine containing vulnerable Docker created by NotSoSecure company, a company that focuses on computer security. This study aims to empirically test the vulnerability analysis in Vulnerable Docker using a vulnerability scanner concerning the CIS Control framework. The purpose of this research can be achieved by conducting experiments to collect the necessary data, analyzing the results of exploits that have been carried out on Vulnerable Docker, and finally making conclusions in the form of reports that are adapted to the CIS Control framework. The Vulnerability Scanner used in this study are AlienVault OSSIM and Docker Bench for Security. In this study, the highest risk was found at the application level, namely the risk caused by the WordPress User IDs and User Names Disclosure type vulnerability through AlienVault scanning. And at the system level caused by a vulnerability type Enable User Namespace Support. There are 6 controls in the CIS Control V8 that are used to reduce the risk that occurs in this study.

Keywords: docker, vulnerability, alienvault, cis control

1. Pendahuluan

Kontainer sering disebut virtualisasi tingkat sistem operasi. Penggunaan kontainer terus meningkat setiap tahunnya seperti pada diagram gambar 1.

Diagram tersebut didapatkan melalui laporan Datadog. Laporan tersebut menjelaskan bahwa terdapat lebih dari 1,5 miliar kontainer yang dijalankan oleh puluhan ribu pelanggan Datadog untuk memahami bagaimana registri *image*, jaringan, dan teknologi lain digunakan di lingkungan kontainer dunia nyata [11].



Gambar 1. Teknis kerja VM, kontainer pada *Bare Metal* dan kontainer pada VM

Dengan hadirnya kontainer Docker, dan segala kemudahan didalamnya, tentu juga perlu diperhatikan mengenai keamanan dan risiko dari penggunaan Docker tersebut. Risiko bisa diakibatkan karena adanya kerentanan (*vulnerability*) pada suatu sistem. Untuk mengukur *vulnerability* pada aplikasi, dapat dilakukan dengan melakukan pemindaian (*scanning*). Pada pengukuran *scanning* akan menemukan risiko *vulnerability* sehingga dapat menemukan evaluasi dan strategi atau rekomendasi yang tepat untuk mengurangi atau menghilangkan risiko. *Vulnerability scanning* tentu dilakukan melalui pemindai (*scanner*). Pada penelitian ini akan dilakukan *scanning* melalui AlienVault dan Docker Bench for Security.

Selain harus memerhatikan keamanan suatu aplikasi, maka yang harus diperhatikan kembali adalah standardisasi secara global untuk menyamaratakan faktor-faktor prasyarat apakah yang harus terdapat pada suatu aplikasi agar aplikasi tersebut dapat dikatakan aman. Salah satu *framework* yang dapat dijadikan sebagai acuan atau standardisasi keamanan aplikasi yaitu *CIS Control*. Justin Gratto (2020) pada unggahahnnya di laman *Securicy.com* mengatakan bahwa *CIS Control* merupakan serangkaian 20 praktik terbaik yang dapat memandu melalui proses pembuatan strategi keamanan *siber*. Penelitian menunjukkan bahwa menerapkan *CIS Control* dapat mengurangi risiko serangan dunia maya yang berhasil di perusahaan sebanyak 85%.

Maka dari itu berdasarkan permasalahan di atas, pada penelitian ini akan dijelaskan bagaimana hasil analisis mengenai kerentanan pada *Vulnerable Docker* menggunakan *vulnerability scanner* yakni AlienVault dan Docker Bench for Security dengan mengacu pada *framework* CIS Control. Hasil analisis tersebut nantinya dapat membantu memperkuat sistem Docker untuk kedepannya

2. Metode Penelitian

2.1 Kontainerisasi

Kontainer umumnya dianggap sebagai teknologi *lean* karena memerlukan *overhead* terbatas, tidak memerlukan lapisan emulasi atau lapisan *hypervisor* untuk dijalankan, melainkan menggunakan antarmuka panggilan sistem normal sistem operasi. Hal ini mengurangi *overhead* yang diperlukan untuk menjalankan kontainer dan memungkinkan kepadatan kontainer yang lebih besar untuk dijalankan pada host.

2.2 Docker

Docker merupakan salah satu *platform* yang dibangun berdasarkan teknologi kontainer. Menurut Saleh Dwiyatno et al., (2020), Docker merupakan sebuah *project open-source* yang menyediakan *platform* terbuka untuk *developer* maupun *sysadmin* untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun sebagai sebuah kontainer yang ringan. Pada penelitian ini digunakan *Vulnerable Docker*. Yakni sebuah *Virtual Machine* yang berisi Docker yang rentan. *Vulnerable Docker* ini dibuat oleh perusahaan bernama *NotSoSecure*.

2.3 *Vulnnerability*

Menurut Rudy Agus (2021), *vulnerability* merupakan suatu karakteristik atau kelemahan spesifik yang membuat informasi atau sistem informasi menjadi terbuka untuk dieksploitasi oleh para peretas.

2.4 *Threat*

Menurut Rudy Agus (2021), ancaman (*threat*) berkaitan erat dengan risiko. *Threat* merupakan suatu keadaan yang berpotensi mengeksploitasi sistem sehingga berdampak buruk pada kelangsungan operasional organisasi, aset, informasi dan sistem informasi, individu, organisasi lain dan masyarakat.

2.5 Risk

Risiko merupakan akibat dari ancaman sistem berupa kehilangan akses data dan informasi penting. Risiko sendiri tidak dapat sepenuhnya dihilangkan, tetapi risiko dapat diperkecil dampaknya terhadap hasil suatu kegiatan. Perhitungan risiko dalam buku *CISSP Study Guide* karya Eric Conrad et al., (2010) dapat dirumuskan seperti berikut.

Risiko = *Threat* x *Vulnerability*

2.6 CIS Control

Framework yang digunakan pada tugas akhir ini adalah CIS Control. Justin Gratto (2020) pada unggahannya di laman *Securicy.com* mengatakan bahwa CIS Control V7.1 merupakan serangkaian 20 praktik terbaik yang dapat memandu melalui proses pembuatan strategi keamanan *siber* serta dapat mengurangi risiko serangan dunia maya yang berhasil di perusahaan sebanyak 85%. CIS Control memiliki 18 poin yang dapat ditindaklanjuti yang dirancang untuk mengelola risiko terkhusus pada Docker.

2.7 *Walkthrough*

Walkthrough merupakan dokumentasi yang menjelaskan mengenai langkah-langkah aksi serangan terhadap suatu sistem. *Walkthrough* juga dapat melakukan eksploitasi terhadap *vulnerability operating system* yang telah banyak tersebar di internet melalui situs komunitas, personal blog, atau Medium.

2.8 AlienVault OSSIM

Normis Sisilya (2019) dalam *Cyber Defense Bulletin Sixth Edition*, AlienVault *Open Source Security Information Management* (OSSIM) merupakan *tools* untuk memantau *event-event* keamanan informasi yang masuk ke dalam kategori SIEM (*Security Information Event Management*).

2.9 Docker Bench for Security

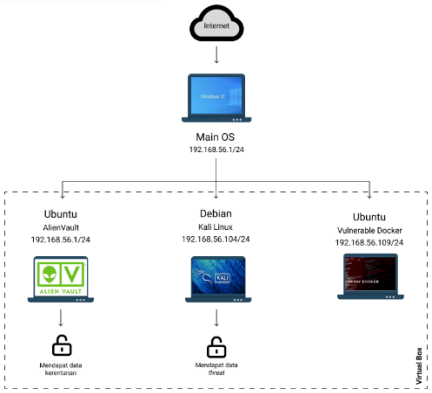
Menurut Konstruktoid (2021), Docker Bench for Security adalah skrip praktik terbaik seputar penerapan kontainer Docker. Semua pengujian dilakukan secara otomatis, dan didasarkan pada CIS Docker Benchmark v1.3.1. Docker Bench for Security ini bermanfaat untuk menilai *host* itu sendiri dan Docker *Container* terhadap tolok ukur yang telah tersedia.

2.10 *Common Vulnerability Scoring System (CVSS)*

CVSS adalah standar industri yang *open source* untuk menilai tingkat keparahan dari *vulnerability* keamanan sistem komputer sehingga nantinya dapat menetapkan solusi yang tepat untuk menangani kerentanan tersebut.

3. Hasil dan Pembahasan

Terdapat rancangan *platform* eksperimen yang merupakan platform yang akan digunakan pada penelitian ini. Bisa dilihat pada gambar 2.

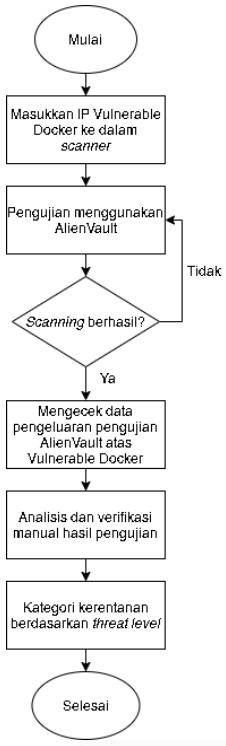


Gambar 2. *Platform* eksperimen

Dapat dilihat bahwa *platform* eksperimen pada penelitian ini terdiri dari *main OS* yang terhubung dengan Virtual Box yang didalamnya memiliki dua *Virtual Machine*, yaitu Kali Linux yang digunakan untuk melakukan *scanning* dan juga digunakan untuk melakukan *attacking*. Sedangkan pada *Virtual Machine* yang kedua, terdapat sistem operasi Ubuntu yang merupakan

tempat untuk meletakkan *Vulnerable Docker* yang berfungsi sebagai target atau *asset IT*.

Selain itu diperlukan juga skenario pengujian pada penelitian ini. Berikut merupakan simulasi langkah pada eksperimen dalam mencari kerentanan yang terdapat pada *Vulnerable Docker*, bisa dilihat di Gambar 3 di bawah ini:



Gambar 3. Skenario pengujian *vulnerability scanning*

Berdasarkan Gambar 3 di atas diketahui bahwa tahapan pertama dengan memasukkan IP dari *Vulnerable Docker*. Setelah ditentukan target, kemudian melakukan pengujian scanning. Kemudian setelah dilakukan pengujian, kemudian lakukan pengecekan *report* hasil scanning pada *Vulnerable Docker*. Jika berhasil maka bisa dilanjutkan dengan menganalisis hasil pengujian dan verifikasi secara manual untuk memastikan apakah *Vulnerable Docker* terdapat kerentanan. Jika tidak maka lakukan *scanning* ulang sampai mendapatkan hasilnya. Setelah di analisis, kategorikan kerentanan yang ada berdasarkan *threat* level.

Maka dari itu, berdasarkan pengujian yang dilakukan pada Docker menggunakan *tool* AlienVault dan Docker Bench for Security, didapatkan hasil seperti pada tabel 1sebagai berikut:

Tabel 1. Hasil *scanning* menggunakan AlienVault

Host	***Threat Level***	***Total***
Vulnerable Docker	*High*	0
	Medium	6
	Low	3
	Info	19

Tabel 2 Hasil *scanning* menggunakan Docker Bench for Security

Host	***Threat Level***	***Total***
Vulnerable Docker	*High*	4
	Medium	6
	Low	1
	Info	0

Pada Tabel 1 diketahui bahwa hasil pengujian dengan AlienVault pada Docker ditemukan kerentanan dengan risiko level *high* sebanyak nol, level *medium* sebanyak enam, level *low* sebanyak tiga, dan level info sebanyak sembilan belas.

Pada Tabel 2 diketahui bahwa hasil pengujian dengan Docker Bench for Security pada *Vulnerable Docker* ditemukan kerentanan dengan risiko level *high* sebanyak empat, level *medium* sebanyak enam, level *low* sebanyak satu, dan tidak ditemukan info.

Pada penelitian ini, terdapat delapan buah *attack threat* yang terjadi, yaitu:

- *Port scanning*
- *Brute force*
- *SSH enumeration*
- *WPScan*
- *SQL Injection*
- *Meterpreter*
- *Reverse shell*

- *Privilage escalation*

Perhitungan risiko menggunakan skor CVSS V2 dari masing-masing *vulnerability* dan dikalikan dengan *threat frequency* yang didapatkan dari perhitungan jumlah *attack threat* yang dijalankan pada satu *walkthrough*. Maka risiko tertinggi didapatkan oleh *vulnerability* jenis:

- *WordPress User IDs and User Names Disclosure* yang merupakan hasil *scanning* AlienVault pada level aplikasi
- *Enable User Namspace Support* yang merupakan hasil *scanning* Docker Bench for Security pada level sistem

Setelah dilakukan analisis risiko maka selanjutnya yaitu menganalisis hasil *scanning Vulnerable Docker* dengan CIS Control V8. Dimana CIS Control V8 bisa menjadi rekomendasi bagi tiap *vulnerability* yang terdeteksi belum memenuhi standar. Berikut penjelasan terkait kontrol:

1. Kontrol yang tepat untuk *vulnerability* jenis *Enable user namespace support* yaitu *Access Control Management*. Dimana disarankan untuk menggunakan alat untuk membuat, menetapkan, mengelola, dan mencabut kredensial akses dan hak istimewa pada akun pengguna, administrator, dan layanan aset serta perangkat lunak perusahaan.

2. Kontrol yang tepat untuk *vulnerability* jenis *Ensure HEALTHCHECK instructions have been added to the container image* yaitu *Implement Code-Level Security Checks*. Yakni menerapkan alat analisis statis dan dinamis dalam aplikasi untuk memverifikasi bahwa praktik yang aman telah dilakukan.

3. Kontrol yang tepat untuk *vulnerability* jenis *Ensure live restore is Enabled* yaitu *Apply Secure Design Principles in Application Architectures*. Yaitu menerapkan prinsip-prinsip desain yang aman dalam arsitektur aplikasi. Prinsip-prinsip desain yang aman yakni mencakup konsep hak istimewa, mempromosikan konsep "jangan pernah mempercayai masukan *user*." Desain yang aman berguna untuk meminimalkan serangan infrastruktur aplikasi.

4. Kontrol yang tepat untuk *vulnerability* jenis *Ensure auditing is configured for the Docker daemon* yaitu *Collect Audit Logs*. Yaitu memastikan bahwa *logging* sudah sesuai dengan proses manajemen *log* audit perusahaan, dan telah diaktifkan di seluruh aset perusahaan.

5. Kontrol yang tepat untuk vulnerability jenis *Ensure network traffic is restricted between containers on the default bridge* yaitu *Securely Manage Network Infrastructure*. Dimana mengelola infrastruktur jaringan agar aman seperti penggunaan protokol jaringan SSH dan HTTPS.

6. Kontrol yang tepat untuk *vulnerability* jenis *Ensure a separate partition for containers has been created* yaitu *Segment Data Processing and Storage Based on Sensitivity*. Yaitu memproses dan menyimpan data berdasarkan sensitivitas data.

7. Kontrol yang tepat untuk *vulnerability* jenis *Ensure Userland Proxy is Disabled* yaitu *Uninstall or Disable Unnecessary Services on Enterprise Assets and Software*. Yaitu mencopot pemasangan atau menonaktifkan layanan yang tidak perlu pada aset dan perangkat lunak perusahaan, seperti modul aplikasi web atau fungsi layanan.

8. Kontrol yang tepat untuk *vulnerability* jenis *Ensure containers are restricted from acquiring new privileges* yaitu *Establish an Access Granting Process*. Yakni menetapkan dan mengikuti proses secara otomatis untuk memberikan akses ke aset perusahaan setelah perekrutan baru atau hak hibah.

9. Kontrol yang tepat untuk *vulnerability* jenis *Enable Ensure Content trust for Docker is Enabled* yaitu *Data Protection*. Yakni mengembangkan proses dan kontrol teknis untuk mengidentifikasi, mengklasifikasikan, menangani, menyimpan, dan membuang data dengan aman.

10. Kontrol yang tepat untuk *vulnerability* jenis *Ensure auditing is configured for Docker files and directories - /var/lib/docker* yaitu

Collect Detailed Audit Logs. Yaitu mengonfigurasikan *logging* audit mendetail untuk aset perusahaan yang berisi data sensitif. Dengan menyertakan sumber acara, tanggal, nama pengguna, stempel waktu, sumber alamat, alamat tujuan, dan elemen berguna lainnya yang dapat membantu dalam penyelidikan forensik.

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka kesimpulan yang dapat diambil yaitu, risiko tertinggi didapatkan oleh *vulnerability* jenis *WordPress User IDs and User Names Disclosure* yang merupakan hasil *scanning* AlienVault pada level aplikasi dan *Enable User Namspace Support* yang merupakan hasil *scanning* Docker Bench for Security pada level sistem. Serta *framework* yang digunakan dalam penelitian ini adalah CIS Control V8. Dari 18 kontrol yang diberikan oleh CIS Control V8, terdapat 6 kontrol yang digunakan untuk mengurangi risiko yang terjadi pada penelitian ini. Diantaranya yaitu *Data Protection*, *Secure Configuration of Enterprise Assets and Software*, *Access Control Management*, *Audit Log Management*, *Network Infrastructure Management*, dan *Applications Software Security*.

Referensi

- Books

[1] Eric Conrad, Sith Misenar, Joshua Feldman. (2010). CISSP Study Guide.

[2] Sahinoglu, Mehmet. (2016). Cyber-Risk Informatics Engineering Evaluation With Data Science.

[3] Sisilya, Normis. (2019). Cyber Defense Bulletin. (6th edition).

[4] Security, Center of Internet. (2021). CIS Control Version 8.

[5] Security, Center of Internet. (2021). Implementation Groups.

[6] Benchmark, CIS. (2021). CIS Docker Benchmark.

- Journal

[7] Dika Priska Prastika, Joko Triyono, Uning Lestari. (2018). "Audit dan Implementasi CIS Benchmark pada Sistem Operasi Linux Debian Server (Studi Kasus: Server Laboratorium Jaringan dan Komputer 6, Institut Sains dan Teknologi AKPRIND Yogyakarta)." Jurnal JARKOM. Vol (5), pp. 182.

[8] Makino, Yuma, Klyuev, Vitaly. (2015). "Evaluation of Web Vulnerability Scanners." The 8th IEEE International Conference on Intelligence Data Acquisition and Advanced Computing Systems: Technology. Pp. 399.

[9] Purwantoro. (2017). "Implementasi Metode Online untuk Mencari Kerentanan (Vulnerability) Server." Jurnal Rekayasa Informasi. Pp. 30.

[10] Saleh Dwiyatno. (2020). "Implementasi Vistualisasi Server Berbasis Docker Container". Jurnal PROSISKO 1.

- World Wide Web

[11] Datadog. "11 Facts About Reak Container Use." Internet: https://www.datadoghq.com/container-report/, Nov. 2020 [Jan. 2021].

[12] Prevasio. "Analysis of Vulnerable Image Container." Internet: https://prevasio.com/dashboard, 2020 [Jan. 2020].

[13] Cobalt. "Understanding the CVSS Base Score: An Essential Guide." Internet: https://cobalt.io/blog/understanding-the-cvss-base-score-an-essential-guide, Apr. 2, 2021 [Jun, 2021].

[14] NotSoSecure." Vulnerable Docker VM." Internet: https://notsosecure.com/vulnerable-docker-vm/, 2017 [May, 2021].

[15] Docker. "Github-Docker Bench Security." Internet: https://github.com/docker/docker-bench-security, Nov. 5, 2019 [Jun, 2021]

[16] Gratto, Justin. "How To Use the CIS Controls Framework for Your Business." Internet: https://www.securicy.com/blog/how-to-use-

cis-security-framework/, Jan. 15, 2020 [Jun, 2021]
[17] NIST. “National Vulnerability Database: Vulnerability Metrics.” Internet: https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator. [Jun, 2021].
[18] Smith, Beth. “IT Security Guru.” Internet: https://www.itsecurityguru.org/2020/05/22/securing-docker-with-cis-controls/, Jun. 5, 2020 [Jun, 2021].
[19] Team Dewaweb. “OWASP: Standar Keamanan Web App Dunia.” Internet: https://www.dewaweb.com/blog/owasp-standar-keamanan-web-app-dunia/, Aug. 13, 2018 [Feb, 2021]