BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Menurut penelitian terdahulu yang telah dikerjakan oleh peneliti sebelumnya *tools* yang digunakan cukup bervariasi, sehingga hasil yang diberikan juga bervariasi. Oleh karena itu, berikut tabel kajian sebagai pembanding penelitian terdahulu dengan penelitian yang akan dilakukan.

Tabel 2. 1 Tabel Penelitian Terdahulu

No	Author	Judul	Tahun	Metode	Hasil
1.	Tika Astriani, Avon Budiyono, Adityas Widjajarto	Analisa Kerentanan Pada *Vulnerable Docker* Menggunakan Scanner Openvas Dan *Docker* Scan Dengan Acuan Standar NIST 800-115	2021	*Vulnerability* scanner dengan acuan standar NIST 800-115	• Tools OpenVAS memperoleh 7 Vulnerabilty, pada Docker Scan diperoleh 8 Vulnerability kemudian kategorikan dalam level High, Medium dan Low • Berdasarkan frekuensi penggunaan tiap walktrough, hasil vulnerability dengan nilai perhitungan resiko tertinggi sebesar terdapat pada Wordpress User IDs and User.
2.	Fatin Hanifah, Avon Budiyono, Adityas Widjajarto	Analisa Kerentanan Pada *Vulnerable Docker* Menggunakan Alienvault Dan *Docker* Bench for Security dengan acuan Framework Cis Control	2021	*Vulnerability* scanner dengan acuan standar Cis Control	• Hasil *scanning* dengan *tools* AlienVault pada aplikasi memperoleh risiko tertinggi pada *vulnerability Wordpress* user IDs and User Names Disclosure. Selanjutnya *vulnerability* system diperoleh Enable User Namspace Support hasil *scanning Docker* Bench. • CIS Control V8 terdapat 18 kontrol, 6 kontrol untuk mengurangi resiko
3.	Ryan Supriadi Ramadhan, Adityas Widjajarto, Ahmad Almaarif	*Vulnerability* Management Pada *Vulnerable Docker* Menggunakan Clair Scanner Dan	2022	*Vulnerability* Management dengan acuan standar GSA CIO-IT Security-17-80	• Versi pertama memerlukan lebih banyak waktu untuk melakukan pemindaian kerentanan daripada versi kedua. • Hasilnya menunjukkan bahwa versi kedua memiliki waktu pemindaian yang lebih cepat. Perbandingan *vulnerabilities* yang ditemukan menunjukkan bahwa pada *Docker Images* terdapat peningkatan sebesar 44,45%, sedangkan pada *Joomla* terdapat peningkatan sebesar 77,78%.

		Joomscan Berdasarkan Standar GSA CIO-IT Security-17-80			
4.	Milenia Oktaviana, Adityas Widjajarto, Ahmad Almaarif	Analisis *Vulnerability* Management Pada Container *Docker* Menggunakan *Opensource* Scanner Berdasarkan Standar Cyber Resilience Review (CRR	2022	*Vulnerability* Management dengan acuan standar Cyber Resilience Review (CRR)	• Keduanya vulneable sistem mempunyai versi yang berbeda-beda yaitu versi – 1 dan versi – 2. Elemen software pada versi – 2 mempunyai tingkat versi yang lebih tinggi dibandingkan versi – 1 • Data eksperimen berupa laporan kerentanan dianalisis berdasarkan Cyber Resilience (CRR) yang fokus pada empat tahap yaitu Menentukan Strategi, mengembangkan rencana, menerapkan kapabilitas, menilai dan meningkatkan kemampuan. Sehingga diperoleh hasil Category *Vulnerability* yaitu Closed *Vulnerability* 30, Open *Vulnerability* 10 , dan 13 New *Vulnerability*.

Pada Tabel 2.1, merupakan penelitian sebelumnya yang telah dilakukan, metode yang digunakan pun kurang lebih sama yaitu *Vulnerability Scanning tools* dan acuan yang digunakan berbeda sehingga menghasilkan kerentanan atau *vulnerability* yang bervariasi. Selanjutnya kerentanan yang sama muncul yaitu *Wordpress* ID user dan User dari hasil peneliti 1 dan 2 . Melalui saran penelitian dari Tika Astriani salah satunya menyarankan untuk menggunakan *tools* yang berbeda pada objek *vulnerable Docker*, oleh karena itu dalam penelitian ini berencana untuk menggembangkan penelitian sebelumnya dengan *tools* yang berbeda. *Tools* yang dipilih oleh peneliti yaitu Trivy dan Nessus, karena kedua *tools* mudah digunakan dan optimal untuk *container Docker* yang akan di uji. Selanjutnya untuk metode yang digunakan peneliti yaitu sama dengan peneliti sebelumnya *Vulnerability Scanning*, tetapi peneliti mencoba untuk menggabungkan dengan metode *Penetration Testing* sebagai pendukung metode *Vulnerability Scanning*, karena kedua metode tersebut saling berhubungan.

2.2 *Vulnerable Docker*

Vulnerable docker merupakan *Virtual machine* dengan *Docker* yang rentan atau *vulnerable* diproduksi oleh perusahaan *NotSoSecure* pada tahun 2017, perusahaan yang fokus pada keamanan komputer [3]. Dalam virtual machine tersebut, sistem operasi yangdigunakan yaitu Ubuntu yang merupakan *Vulnerable Docker* yang berjalan sebagaitarget [3]. *Vulnerable docker* tersebut menjadi objek

peneliti, dengan masalah *vulnerability* pada *image Docker* berasal dari fakta bahwa pengguna tidak melakukan pekerjaan verifikasi keamanan terpisah pada *image Docker* dan diunggah dalam repositori gambar *Docker* [9]. Sehingga *container* pada *vulnerable Docker* akan menjadi objek yang akan di serang atau *attack* menggunakan *tools* nantinya.

2.3 *Vulnerability Scanning*

Vulnerability scanning merupakan proses mendapatkan *vulnerability* atau kerentanan pada *network* menggunakan *tools scanning* [10]. Selanjutnya alur dari *vulnerability scanning* sebagai berikut:

1. *Identification*

   Diawali dengan mencari suatu permasalahan pada penelitian, setelah masalah diidentifikasi dan permasalahan tersebut dapat dirumuskan, Selanjutnya akan mengaudit sistem yang akan di-scan.
2. *Type Vulnerabilities scanning*

   Tahap ini menetentukan tepi *vulnerability scanning* yang akan digunakan dalam penelitian
3. *Scanning* Analisis

   Pada tahap ini melakukan *scanning* dan analisa data jaringan target sehingga hasil yang diperoleh menjadi perbaikan kedepannya untuk sistem yang diuji.
4. *Correct weaknesses*

   Tahapan ini akan melakukan perbaikan pada kelemahan atau kerentanan yang ditemukan

Dalam penelitian ini, untuk melakukan *vulnerability scanning* menggunakan *opensource tools* yaitu *Trivy* dan *Nessus*. Kedua *tools* tersebut akan melakukan *scanning* pada objek *Vulnerable docker*.

2.4 *Penetration Testing*

Pengujian *Penetration Testing* merupakan alur kegiatan yang dilakukan untuk mengidentifikasi dan mengeksploitasi kerentanan celah keamanan atau *vulnerability* yang ada dalam sebuah sistem [11] . Dalam banyak situasi, uji penetrasi dianggap sebagai jenis pengujian yang paling agresif yang dapat dilakukan terhadap suatu organisasi. Sementara itu, pengujian lain memberikan

informasi tentang keunggulan dan kelemahan sistem dalam sebuah organisasi [12]. Sebelum menemukan regulasi yang tepat untuk memastikan keamanan yang efektif, ada beberapa langkah yang harus dilewati selama proses uji coba. Berikut adalah langkah-langkah dari Pengujian Keamanan (*Pentest*) [13]:

1. Tahap Perencanaan

Langkah awal yang diperlukan sebelum memulai pengujian adalah memahami sistem keamanan server dan mengumpulkan informasi tentang sistem tersebut. Tujuannya adalah untuk memahami lingkungan sistem yang akan diuji dan merencanakan uji coba yang sesuai.

2.Tahap Pemindaian

Setelah perencanaan dilakukan dengan cermat, langkah berikutnya adalah melakukan pemindaian untuk mencari celah keamanan pada sistem yang dituju. Proses ini melibatkan penggunaan alat tambahan seperti enumerasi layanan, pemindaian port, dan pemindaian kerentanan.

3.Tahap Mendapatkan

Akses Setelah menemukan celah keamanan pada target, langkah selanjutnya adalah mencoba masuk ke dalam sistem tersebut. Penguji akan mencoba berperan sebagai penyerang yang berusaha untuk mendapatkan akses penuh ke dalam sistem yang diuji. Untuk melakukan hal ini, penguji akan menggunakan alat seperti injeksi SQL, *cross-site scripting*, dan *backdoor*.

4.Tahap Mempertahankan Akses

Setelah berhasil mendapatkan akses penuh, langkah selanjutnya adalah memastikan apakah celah keamanan tersebut bersifat sementara atau permanen. Jika celah tersebut bersifat permanen, hal ini bisa berdampak buruk bagi pengguna karena penyerang dapat terus masuk ke dalam sistem hingga ke inti sistem.

5.Tahap Pelaporan Hasil

Setelah mendapatkan akses, langkah berikutnya adalah menghasilkan laporan yang akan disampaikan kepada perusahaan yang memiliki sistem tersebut. Laporan ini akan berisi informasi tentang celah yang ditemukan, solusi terbaik untuk memperbaikinya, dan rekomendasi untuk meningkatkan keamanan sistem.

6.Tahap Perbaikan

Langkah terakhir adalah mengimplementasikan solusi untuk mengatasi kerentanan yang ditemukan pada sistem. Jika masih ada kerentanan yang tinggi, sistem akan kembali untuk diuji lagi guna memastikan bahwa keamanan telah ditingkatkan

2.5 *Opensource tools*

Open souce *tools* adalah jenis software yang kode *resource*-nya terbuka untuk dipelajari, diubah, ditingkatkan dan disebarluaskan. Berikut kelebihan dan kekurangan dari *opensource tools*. Penggunaan alat Open Source (*opensource tools*) memiliki sejumlah kelebihan dan kekurangan. Berikut adalah beberapa poin utama untuk masing-masing:

Kelebihan *Tools Open Source*:

1. Akses *Opensource*:

Pengguna memiliki akses penuh ke *opensource*, memungkinkan penggunaan, modifikasi, dan penyesuaian sesuai kebutuhan spesifik.

2. Biaya Rendah atau Gratis:

Banyak *tools Open Source* dapat diunduh dan digunakan tanpa biaya lisensi. Ini dapat membantu organisasi atau individu menghemat biaya.

3. Komunitas Pengembang yang Luas:

Banyak proyek *Open Source* didukung oleh komunitas pengembang yang besar. Ini berarti adanya dukungan, pembaruan reguler, dan berbagi pengetahuan.

4. Keamanan yang Dapat Diperiksa:

Dengan akses ke kode sumber, komunitas dapat secara terbuka memeriksa dan mengevaluasi keamanan alat. Keterbukaan ini dapat mengurangi risiko adanya celah keamanan yang tidak terdeteksi.

5. Kemudahan Penyesuaian dan Integrasi:

Kode *Open Source* dapat disesuaikan sesuai kebutuhan spesifik dan dengan mudah diintegrasikan dengan sistem lain.

Kekurangan *Tools* Open Source:

1. Kurangnya Dukungan Resmi:

Beberapa proyek *Open Source* mungkin kurang memiliki dukungan resmi atau *SLA (Service Level Agreement).* Ini bisa menjadi masalah jika organisasi memerlukan tingkat dukungan yang tinggi.

2. Tingkat Kesulitan Penggunaan:

Beberapa alat *Open Source* mungkin memiliki tingkat kesulitan penggunaan yang lebih tinggi atau kurangnya antarmuka pengguna yang ramah.

3. Ketidakpastian Pengembangan:

Beberapa proyek *Open Source* mungkin mengalami ketidakpastian dalam pengembangan jika tidak ada sumber daya yang memadai atau dukungan komunitas berkurang.

4. Kurangnya Fitur Secara *Default*:

Beberapa alat *Open Source* mungkin tidak memiliki semua fitur yang dibutuhkan secara default, memerlukan penyesuaian tambahan.

5. Isu Kompatibilitas:

Dalam beberapa kasus, alat *Open Source* mungkin tidak sepenuhnya kompatibel dengan beberapa teknologi tertentu atau sistem operasi. Keputusan untuk menggunakan alat *Open Source* atau *propietary* harus mempertimbangkan kebutuhan spesifik organisasi atau individu serta karakteristik dan kondisi proyek yang bersangkutan. *Open Source* dapat menjadi pilihan yang sangat baik, tetapi kelebihan dan kekurangan harus dievaluasi dengan cermat.

Trivy adalah *tools opensource scanning* pihak ketiga yang pertama-tama akan memeriksa apakah images tersebut benar-benar aman atau tidak [14]. *Trivy* memindai images kontainer lokal dan jarak jauh, mendukung beberapa mesin kontainer, serta *image* yang diarsipkan dan diekstraksi. Ia bekerja pada sistem file mentah dan repositori git jarak jauh.

Nessus adalah salah satu alat populer untuk melakukan pemindaian keamanan pada jaringan dan sistem komputer. *Nessus* memiliki *database* Kerentanan yang Luas, sehingga *tools* ini memiliki database kerentanan yang luas dan terus diperbarui secara berkala. Ini memungkinkan *Nessus* untuk mengenali kerentanan terbaru dan memberikan Informasi yang akurat.