Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

TUGAS AKHIR

MUH. AHSAN
41518010169

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2022

Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:
MUH. AHSAN
41518010169

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2022

LEMBAR PERNYATAAN ORISINALITAS

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM	: 41518010169
Nama	: MUH. AHSAN
Judul Tugas Akhir	: Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 28 Juli 2022

MUH. AHSAN

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini:

Nama Mahasiswa	:	MUH. AHSAN
NIM	:	41518010169
Judul Tugas Akhir	:	Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 28 Juli 2022

MUH. AHSAN

SURAT PERNYATAAN LUARAN TUGAS AKHIR

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini:

Nama Mahasiswa	:	MUH. AHSAN
NIM	:	41518010169
Judul Tugas Akhir	:	Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Menyatakan bahwa:

a. Luaran Tugas Akhir saya adalah sebagai berikut:

No	Luaran	Jenis			Status	
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi			Diajukan	✓
		Jurnal Nasional Terakreditasi	✓			
		Jurnal International Tidak Bereputasi			Diterima	
		Jurnal International Bereputasi				
	Disubmit/dipublikasikan di:	Nama Jurnal	:	Jurnal INTECH		
		ISSN	:	2722-7367		
		Link Jurnal	:	http://journal.unbara.ac.id/index.php/INTECH/index		
		Link File Jurnal Jika Sudah di Publish	:			

b. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.

c. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan.

Demikian pernyataan ini saya buat dengan sebenarnya.

Mengetahui
Dosen Pembimbing TA

Dwi Anindyani Rocmah, ST, MTI

Jakarta, 28 Juli 2022

METERAI TEMPEL 10000 612B0AJX487496520

MUH. AHSAN

LEMBAR PERSETUJUAN PENGUJI

LEMBAR PERSETUJUAN PENGUJI

NIM	:	41518010169
Nama	:	MUH. AHSAN
Judul Tugas Akhir	:	Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022

(Anis Cherid, SE, MTI)

LEMBAR PERSETUJUAN PENGUJI

LEMBAR PERSETUJUAN PENGUJI

NIM	:	41518010169
Nama	:	MUH. AHSAN
Judul Tugas Akhir	:	Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022

(Eugenius Kau Suni, ST, MT)

LEMBAR PERSETUJUAN PENGUJI

LEMBAR PERSETUJUAN PENGUJI

NIM	:	41518010169
Nama	:	MUH. AHSAN
Judul Tugas Akhir	:	Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022

(Dhanny Permatasari Putri, S.Kom, MT)

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

NIM	:	41518010169
Nama	:	MUH. AHSAN
Judul Tugas Akhir	:	Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022

Menyetujui,

(Dwi Anindyani Rocmah, ST, MTI)
Dosen Pembimbing

Mengetahui,

(Wawan Gunawan, S.Kom, MT)	(Ir. Emil R. Kaburuan, Ph.D., IPM.)
Koord. Tugas Akhir Teknik Informatika	Ka. Prodi Teknik Informatika

KATA PENGANTAR

Alhamdulillahi Rabbil'alamin.

Puji syukur kita panjatkan kepada Allah SWT yang melimpahkan rahmat, nikmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Analisa Kerentanan Sistem Dengan Menerapkan *Open Vulnerability Assessment System* Menggunakan Greenbone Vulnerability Management (GVM)". Tidak lupa shalawat serta salam tercurahkan kepada Nabi agung Muhammad SAW yang syafa'atnya kita nantikan kelak.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, penulis tidak akan mampu menyelesaikan laporan ini dengan baik. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Ibunda dan Ayahanda (Almarhum) semasa hidupnya yang selalu mendoakan dan tiada hentinya mendidik, menyayangi, serta memberikan dukungan, semoga Allah ta'ala membalas amal budinya dengan kasih sayang berlimpah.
2. Ibu Dwi Anindyani Rohmach, ST, MTI selaku Dosen Pembimbing Tugas Akhir yang telah banyak membantu dan bersedia meluangkan waktunya untuk memberikan bimbingan, kritik serta saran yang berguna dalam penyusunan Tugas Akhir ini.
3. Bapak Wawan Gunawan, S.Kom, MT selaku Koordinator Tugas Akhir pada Jurusan Informatika Universitas Mercu Buana.
4. Bapak Achmad Kodar, Drs. MT selaku Dosen Pembimbing Akademik
5. Seluruh dosen Fakultas Ilmu Komputer UMB yang telah memberikan bekal ilmu selama masa kuliah di UMB.
6. Teman-teman Informatika angkatan 2018 yang berbagi pengalaman dan ilmu kepada penulis.

Akhir kata, penulis berharap penelitian ini dapat bermanfaat bagi diri penulis, Universitas, Organisasi, maupun bagi rekan-rekan mahasiswa sekalian.

Jakarta, 28 Juli 2022
Penulis

DAFTAR ISI

NASKAH JURNAL

JURNAL INTECH, VOL 3, NO 2, NOVEMBER 2022, PP. 1 - 7

IT

Terbit *online* pada laman web jurnal : http://journal.unbara.ac.id/index.php/INTECH

INFORMATIKA DAN TEKNOLOGI (INTECH)

ISSN (Online) : 2722-7367

Analisa Kerentanan Sistem Dengan Menerapkan Open Vulnerability Assessment System Menggunakan Greenbone Vulnerability Management (GVM)

MUH. AHSAN [1] *, Dwi Anindyani Rochmah, ST.,MTI* [2]

Universitas Mercu Buana, Jl Meruya Selatan No. 1, Kembangan, Jakarta Barat 11650, Indonesia

[1]41518010169@student.mercubuana.ac.id, [2]dwi.anindya@mercubuana.ac.id

INFORMASI ARTIKEL

Diterima Redaksi:

Revisi Akhir:

Diterbitkan *Online*:

KATA KUNCI

Network Scanning, Vulnerability Assessment, GVM

ABSTRACT

A computer network is a network telecommunication that connects one or more computers to exchange data and information with each other. Such huge benefit will certainly be reduced by the presence of interference that arises in the network, when the network only involves local devices or in other words, is not connected to the internet network then interference may be less calculated. However, when the local network is connected to the internet network, a security system will be something that must be considered. Every system and network will undoubtedly have vulnerabilities and can cause damage to the system and even data to cause losses. Tracing activities and identifying system vulnerabilities are effective ways to minimize the risk of continuous vulnerability, therefore this study aims to conduct network analysis to determine vulnerabilities system by applying the open vulnerability assessment system method using Greenbone Vulnerability Management (GVM) as a platform of network scanning and vulnerability management system. The results are achieved by doing the vulnerability analysis to evaluate the results of vulnerability findings and then categorized according to the level of risk, namely high, medium and low. Conducting a vulnerability assessment will improve information security and avoid bad risks that can cause losses. The results achieved doing the vulnerability assessment activity are to provide an evaluation of the risk of vulnerability found and the impact that can be generated and as a preventive measure to increase security system awareness.

1. PENDAHULUAN

Pada setiap sistem dan jaringan tentu akan mempunyai kerentanan dan dapat mengakibatkan kerusakan bahkan kehilangan data sehingga menimbulkan kerugian. Sangat diperlukan aktivitas penelusuran dan identifikasi kerentanan sistem untuk menanggulangi dampak kerusakan karena akibat adanya serangan pihak yang tidak bertanggung jawab. Hal ini menjadi dasar untuk meningkatkan kesadaran dan melakukan langkah awal untuk mendeteksi, mengidentifikasi dan mempelajari kelemahan yang dimiliki dari suatu sistem. Berdasarkan kasus tersebut maka sangat penting untuk menerapkan *vulnerability assessment* yang Dilakukan dengan menggunakan Greenbone Vulnerability Management (GVM). GVM sebagai wadah *network scanning* dan *vulnerability management system* yang mampu menjadi salah satu solusi untuk memberikan gambaran dari sebuah penelusuran celah keamanan. Hasil yang dicapai dalam analisa *vulnerability* adalah mengevaluasi hasil temuan kerentanan kemudian dikategorikan sesuai dengan tingkat resikonya yaitu tingkat tinggi, tingkat sedang dan tingkat rendah. Melakukan *vulnerability assessment* akan meningkatkan keamanan informasi serta terhindar dari resiko buruk yang dapat menimbulkan kerugian. Hasil yang dicapai dalam kegiatan *vulnerability assessment* adalah

JURNAL INTECH, VOL 3, NO 2, NOVEMBER 2022, PP. 2 - 7

memberikan evaluasi terhadap resiko kerentanan yang ditemukan dan dampak yang dapat ditimbulkan serta sebagai langkah preventif untuk meningkatkan kesadaran keamanan sistem.

2. TINJAUAN PUSTAKA

2.1 Virtual Machine

Virtual Machine adalah rekayasa perangkat lunak yang memiliki fungsi hampir sama seperti komputer fisik yang dapat digunakan untuk menambah sistem operasi didalam sistem operasi utama atau dapat dikatakan sebagai tempat untuk melakukan simulasi sistem operasi.

2.2 Nmap

Network Mapper merupakan *open source tool* untuk eksplorasi dan audit keamanan jaringan. Dirancang untuk memeriksa jaringan besar secara cepat, meskipun dapat pula bekerja terhadap single host. Nmap menggunakan paket IP raw untuk menentukan host mana saja yang tersedia pada jaringan, layanan nama aplikasi dan versi apa yang diberikan, sistem operasi dan versi apa yang digunakan, apa jenis *firewall* atau *filter* paket yang digunakan, dan sejumlah karakteristik lainnya [1]. Meskipun Nmap umumnya digunakan untuk audit keamanan, namun banyak administrator sistem dan jaringan menganggapnya berguna untuk tugas rutin seperti inventori jaringan, mengelola jadwal upgrade layanan, dan melakukan *monitoring up time host* atau layanan.

2.3 Greenbone Vulnerability Management (GVM)

Greenbone Vulnerability Management awalnya dibangun sebagai proyek komunitas yang bernama OpenVAS. GVM diteruskan dan dikembangkan oleh Greenbone Network [2] [3] [4]. Gambar berikut merupakan proses OpenVAS 9 menjadi GVM 11.

Gambar 1. OpenVAS 9 menjadi GVM 11

Gambar berikut merupakan arsitektur *patch level* GVM 21.04 secara umum.

Gambar 2. Arsitektur GVM 21.04

GMV dikelompokkan menjadi tiga bagian utama:

a. Aplikasi *scan* dapat melakukan eksekusi dengan menjalankan Vulnerability Test (VT) terhadap sistem target.
b. Greenbone Vulnerability Manager Daemon (gvmd)
c. Greenbone Security Assistant (GSA) dengan the Greenbone Security Assistant Daemon (gsad)

GVM memiliki komponen yang saling terkait satu dengan yang lain [5]. Komponen tersebut antara lain:

a. Greenbone Vulnerability Manager Daemon (gvmd)

Greenbone Vulnerability Manager Daemon (gvmd) adalah pusat layanan yang mengkonsolidasikan pemindaian kerentanan menjadi solusi manajemen kerentanan penuh. gvmd mengontrol OpenVAS Scanner melalui Open Scanner Protocol (OSP). Layanan itu sendiri menawarkan Greenbone Management Protocol (GMP) berbasis XML. gvmd juga mengontrol database SQL (PostgreSQL) tempat semua konfigurasi dan data hasil pemindaian disimpan secara terpusat. Selanjutnya, gvmd juga menangani manajemen pengguna termasuk kontrol izin dengan grup dan peran. Dan terakhir, layanan memiliki sistem runtime internal untuk tugas terjadwal dan acara lainnya.

b. Greenbone Security Assistant (GSA)

Greenbone Security Assistant (GSA) adalah web interface dari GVM yang mana pengguna mengatur scan dan mengakses informasi kerentanan secara bersama. Ini adalah titik kontak utama bagi pengguna dengan GVM. Terhubung ke gvmd melalui server web Greenbone Security Assistant Daemon (gsad) untuk menyediakan aplikasi web berfitur lengkap untuk manajemen kerentanan. Komunikasi terjadi menggunakan Greenbone Management Protocol (GMP) dimana pengguna juga dapat berkomunikasi secara langsung dengan menggunakan alat yang berbeda.

c. OpenVAS Scanner

OpenVAS Scanner adalah mesin pemindai berfitur lengkap yang menjalankan uji kerentanan (VT) terhadap sistem target. Untuk ini, ia menggunakan feed harian yang diperbarui dan komprehensif: Greenbone Security Feed (GSF) komersial berfitur lengkap, ekstensif, atau Greenbone Community Feed (GCF) yang tersedia gratis. OpenVAS Scanner terdiri dari komponen ospd-openvas dan openvas-scanner Pemindai OpenVAS dikendalikan melalui OSP. OSP Daemon untuk OpenVAS Scanner (ospd-openvas) berkomunikasi dengan gvmd melalui OSP: Data VT dikumpulkan, pemindaian dimulai dan dihentikan, dan hasil pemindaian ditransfer ke gvmd melalui ospd.

JURNAL INTECH, VOL 3, NO 2, NOVEMBER 2022, PP. 3 - 7

2.4 *Vulnerability Assessment*

Vulnerability Assessment melakukan identifikasi kerentanan dari suatu aplikasi, sistem operasi dan infrastruktur jaringan. *Vulnerability Assessment* tidak melakukan eksploitasi celah atau kelemahan dari suatu sistem lebih fokus untuk menemukan beragam public vulnerability pada seluruh sistem komputer dalam jaringan target dan tidak menuju ke proses eksploitasi namun memiliki potensi untuk di eksploitasi sehingga harus ditutup kerentanan yang ditemukan. [6].

2.5 *Vulnerability Management*

1. *Vulnerability Identification*

Tujuan dari langkah ini adalah untuk menyusun daftar lengkap kerentanan aplikasi. Analis keamanan menguji kesehatan keamanan aplikasi, server, atau sistem lain dengan memindainya dengan alat otomatis, atau menguji dan mengevaluasinya secara manual. Analisa ini juga mengandalkan database kerentanan, pengumuman kerentanan vendor, sistem manajemen aset, dan umpan intelijen ancaman untuk mengidentifikasi kelemahan keamanan.

2. *Vulnerability Analysis* (*Vulnerability Scanning*)

Tujuan dari langkah ini adalah untuk mengidentifikasi sumber dan akar penyebab kerentanan yang diidentifikasi pada langkah pertama.

Ini melibatkan identifikasi komponen sistem yang bertanggung jawab untuk setiap kerentanan, dan akar penyebab kerentanan.

3. *Risk Assessment* (*Vulnerability Assessment*)

Tujuan dari langkah ini adalah memprioritaskan kerentanan. Ini melibatkan analis keamanan yang menetapkan peringkat atau skor keparahan untuk setiap kerentanan, berdasarkan faktor-faktor seperti:

a. Sistem mana yang terpengaruh.
b. Data apa yang berisiko.
c. Fungsi bisnis mana yang berisiko.
d. Kemudahan menyerang atau kompromi.
e. Tingkat keparahan serangan.
f. Potensi kerusakan sebagai akibat dari kerentanan.

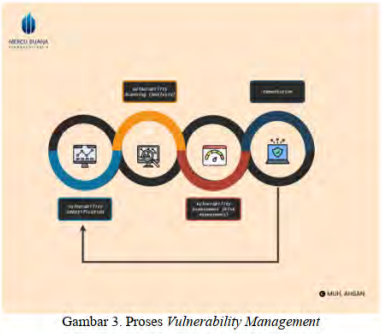
4. *Remediation*

Tujuan dari langkah ini adalah untuk menutup celah keamanan. Ini biasanya merupakan upaya bersama oleh staf keamanan, tim pengembangan dan operasi, yang menentukan jalur paling efektif untuk remediasi atau mitigasi setiap kerentanan.

Langkah-langkah perbaikan khusus mungkin termasuk:

a. Pengenalan prosedur, tindakan, atau alat keamanan baru.
b. Pembaruan perubahan operasional atau konfigurasi.
c. Pengembangan dan implementasi patch kerentanan.

Penilaian kerentanan tidak bisa menjadi kegiatan satu kali saja. Agar efektif, organisasi harus mengoperasionalkan proses ini dan mengulanginya secara berkala. Penting juga untuk mendorong kerja sama antara tim keamanan, operasi, dan pengembangan-sebuah proses yang dikenal sebagai DevSecOps



Gambar 3. Proses *Vulnerability Management*

3. METODE PENELITIAN

3.1 *Jenis Penelitian*

Metode yang digunakan penulis dalam penelitian ini adalah metode penelitian terapan yang berfokus pada analisis hasil evaluasi sehingga diharapkan dapat menghasilkan berupa informasi yang dijadikan masukan atau pengambilan keputusan tertentu sesuai urgensi sasaran.

3.2 *Metode Pengumpulan Data*

Metode pengumpulan data yang digunakan pada penelitian ini adalah observasi yaitu melakukan analisa terhadap kerentanan sistem dan penelitian tindakan yang mengimplementasi *Open Vulnerability Assessment System* menggunakan GVM berdasarkan topologi yang sedang berjalan.

3.3 *Tahap Penelitian*

Secara teknis penelitian ini akan dilaksanakan menggunakan 3 tahapan inti dari proses analisa *vulnerability*. Tahapan tersebut seperti yang terlihat pada Gambar 4.

Gambar 4. *Flowchart* Analisa *Vulnerability*

Dapat dilihat pada Gambar 4 diagram alir / *flowchart* dari penelitian yang dilaksanakan.

Pertama, Batasan proyek diperlukan agar analisa vulnerability tidak terlalu luas, sehingga melibatkan hal-hal lain yang tidak perlu dan tidak terlampau sempit sehingga

JURNAL INTECH, VOL 3, NO 2, NOVEMBER 2022, PP. 4 - 7

melewatkan hal-hal yang penting. Untuk menentukan batasan sistem, ada 2 hal yang perlu dijadikan sebagai pertimbangan yaitu pemahaman terhadap proses sistem yang akan diuji dan pemahaman kompleksitas sistem.

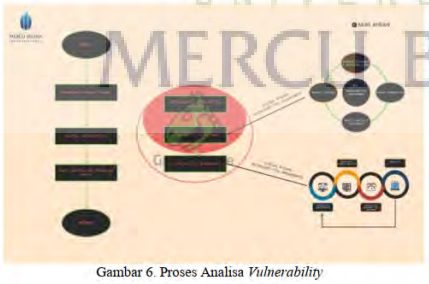
Kedua, Analisa vulnerability yang mengacu pada utilisasi GVM

Ketiga, Hasil analisa dan pelaporan akhir. Dalam tahap ini adalah tahap yang terakhir dari proses mekanisme analisa vulnerability. Tahap pelaporan adalah tahap yang paling penting, fase ini adalah memberikan rekomendasi tentang temuan hasil identifikasi kerentanan. Fase reporting merupakan kegiatan memetakan hasil identifikasi sehingga kerentanan yang ditemukan dapat dikategorikan dengan baik serta dapat dilakukan tindakan mitigasi untuk memberikan rekomendasi kepada pihak target tentang kerentanan sistem yang dimilikinya.

Gambar 5. Siklus Analisa *Vulnerability*

Pada gambar 5 penulis menggambarkan skema analisa yang terjadi dibagian urutan ke-2 gambar 7 terkait analisa vulnerability yang memiliki cakupan 3 unsur yaitu pelaksanaan *Vulnerability Scanning*, *Vulnerability Assessment*, *Vulnerability Management*.

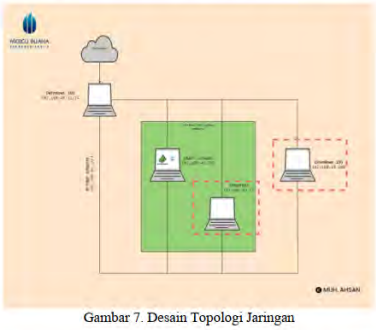
1. *Vulnerability Scanning*: Melakukan scanning pada sistem.
2. *Vulnerability Assessment*: Identifikasi vulnerability pada sistem.
3. *Vulnerability Management*: Pengolahan analisa vulnerability pada sistem.



Gambar 6. Proses Analisa *Vulnerability*

Skema umum kompleks yang terjadi pada proses analisa *vulnerability* yang penulis lakukan dapat dilihat pada gambar 6.

Berikut gambaran umum sistem yang didesain menjadi topologi jaringan yang digunakan untuk analisa penggunaan GVM sebagai mekanisme *vulnerability assessment*.



Gambar 7. Desain Topologi Jaringan

Jaringan yang digunakan adalah jaringan lokal. Topologi yang digunakan untuk pengujian ini menggunakan satu buah server dan dua client yang terhubung dalam jaringan virtual yang dipasang pada VMware Workstation. Di sisi linux terdapat sistem GVM yang dipasang sebagai alat bantu pengujian, kemudian sebagai sisi target terdapat windows 10 dan Ubuntu Linux yang memiliki masing-masing ip *address* dhcp. Pada sisi client yaitu kali linux yang sudah terpasang sistem GVM sebagai alat bantu pengujian. Didalam topologi terdapat satu buah server GVM yang terpasang dalam sistem kali linux dan dua buah client yang terhubung melalui jaringan virtual menggunakan Vmwork Workstation dengan network bridge mode. Maksudnya ketika virtual machine ini salah satu adapter network-nya menggunakan bridge maka virtual machine tersebut akan bisa terhubung dengan jaringan Wi-Fi, PC lain maupun LAN. IP *Address* yang diterima VM akan satu segment dengan *real network* dan bisa saling berkomunikasi antar keduanya.

Berikut alokasi Alamat IP pada tabel 2

Tabel 1. Alokasi Alamat IP

Nama Perangkat	Alamat IP
Bridge Network	192.168.43.1/24
Main OS	192.168.43.11/24
Kali Linux	192.168.43.250/24
Ubuntu Linux	192.168.43.27/24
Windows 10	192.168.43.209/24

Pada gambar 8 penulis mencari dan memastikan bawah target ip *address* tersedia dengan melakukan explorasi ip *address* menggunakan nmap *tool*.

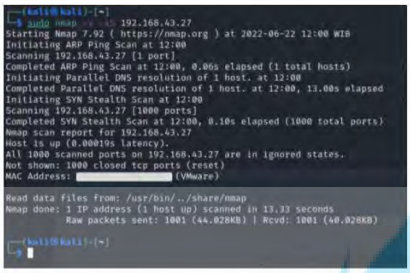
Gambar 8. Eksplorasi IP *Address*

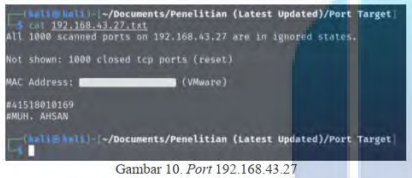
JURNAL INTECH, VOL 3, NO 2, NOVEMBER 2022, PP. 5 - 7

Hasil dari explorasi ip *address* pada gambar 8 terlihat bahwa terdapat target ip *address* 192.168.43.27 dan 192.168.43.209 yang penulis inginkan.

Gambar 9. IP *Address* Target

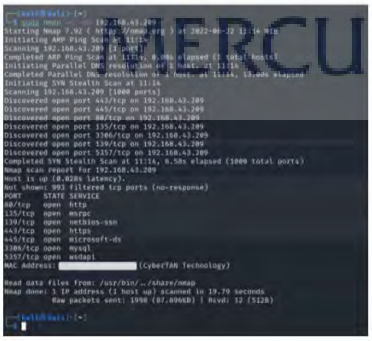
Pada gambar 9 penulis membuat direktori serta *file* baru untuk menyimpan ip *address* target yang akan digunakan untuk melakukan analisa *vulnerability*.

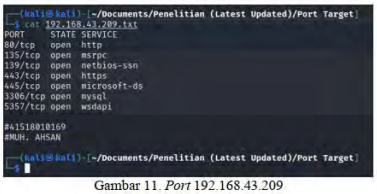




Gambar 10. *Port* 192.168.43.27

Hasil dari eksplorasi *port* ip *address* 192.168.43.27 pada gambar 10 terlihat bahwa tidak ada *port* yang terbuka dari 1000 tcp *ports* yang dieksplorasi oleh npm *tool* sehingga ip *address* 192.168.43.27 tidak digunakan oleh penulis untuk melakukan analisa *vulnerability*.





Gambar 11. *Port* 192.168.43.209

Hasil dari eksplorasi *port* ip *address* 192.168.43.209 pada gambar 11 terlihat bahwa adanya *port* yang terbuka dari 1000 tcp *ports* yang dieksplorasi oleh npm *tool* sehingga ip *address* 192.168.43.27 dapat digunakan oleh penulis untuk melakukan analisa *vulnerability*.

Gambar 12. GVM *Check Setup*

Pada gambar 12 penulis melakukan pengecekan GVM *tool* secara berkala yang sudah penulis instalasi di server kali linux. Hal ini sangat penting untuk mengetahui apakah *tool* dapat digunakan tanpa adanya *error* sebelum dijalankan dengan memanggil perintah gvm-start.

Gambar 13. GVM *Check Setup*

Pada gambar 13 terdapat bahwa ada beberapa daftar *port* secara bawaan pada *portlists menu*. Penulis menggunakan All TCP an Nmap top 100 UDP yang memiliki *Port Counts* untuk TCP 65535.

Gambar 14. GVM *Targets Menu*

Pada gambar 14 terdapat target yang dibuat oleh si penulis dengan rincian sebagai berikut:

a. *Name*: Penelitian MPTI - TA 2021
b. *Host*: 192.168.43.209
c. *Port Type List*: All TCP and Nmap top 100 UDP

JURNAL INTECH, VOL 3, NO 2, NOVEMBER 2022, PP. 6 - 7

Gambar 15. GVM *Feed Status Menu*

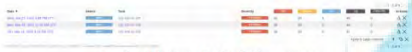
Pada gambar 15 terdapat daftar feed status NVT, SCAP, CERT, GVMD_DATA. Pembaruan *feed* status secara berkala sangat diperlukan untuk melakukan analisa vulnerability yang mana nantinya secara otomatis sumber konten komponen utama dari GVM akan *up to date*.

4. HASIL DAN PEMBAHASAN

Proses identifikasi kerentanan diwadahi dengan GVM dan secara umum menggunakan *open vulnerability assessment tool* dengan menargetkan 1 *host* yang sebelumnya dihasilkan dari proses tahapan eksperimen. Berikut hasil eksperimen analisa vulnerability.

Gambar 16. Hasil Eksperimen Akhir

Tanggal 22 Juni 2022 merupakan hasil akhir dari analisa kerentanan yang menghasilkan *severity* atau tingkat kerentanan 9.9 (*High*) dengan diikuti *trend level* yaitu *up*.



Gambar 17. Periode Hasil Eksperimen

Terdapat 3 priode hasil eksperimen pada gambar 17 sebagai berikut:

Pertama, pada Kamis, 24 Maret 2022 menghasilkan severity 9.9 (*High*). Dengan 5 parameter yaitu *high* sebanyak 36, *medium* 30, *low* 0, *log* 51, *False Pos*. 0.

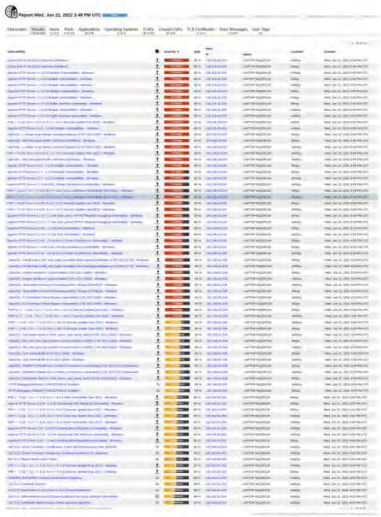
Kedua, pada Kamis, 29 Maret 2022 menghasilkan severity 9.9 (*High*). Dengan 5 parameter yaitu *high* sebanyak 36, *medium* 30, *low* 0, *log* 44, *False Pos*. 0.

Ketiga, pada Rabu, 22 Juni 2022 menghasilkan severity 9.9 (*High*). Dengan 5 parameter yaitu *high* 42, *medium* 29, *low* 0, *log* 49, *False Pos.* 0.

Gambar 18. Hasil Eksperimen pada *Teks Section*

Gambar 19. Hasil *Main Report Section*

Gambar 20. Hasil informasi



Gambar 21. Hasil Kerentanan

Gambar 22. Hasil *Host*

Gambar 23. Hasil *Port*

Gambar 24. Hasil Aplikasi

Gambar 25. Hasil Sistem Operasi

JURNAL INTECH, VOL 3, NO 2, NOVEMBER 2022, PP. 7 - 7

Gambar 26. Hasil CVEs

Gambar 27. Hasil *Closed* CVEs

Gambar 28. Hasil TLS *Certification*

Tanggal 24 maret - 22 juni 2022 adanya kenaikan *trend* kerentanan selama melakukan analisa *vulnerability* dengan target *single host* 192.168.43.209 seperti yang terlihat pada gambar 17.

Untuk hasil akhir penelitian pada tanggal 22 juni terdapat 42 kerentananan tingkat *high* dan 29 *medium* seperti yang terlihat pada gambar 16.

Terdapat 3 ports dengan *severity* kerentanan yaitu 443/tcp dan 80/tcp memiliki nilai 9.9 (*high*) serta 135/tcp memiliki nilai 5.0 (*medium*). Dapat dilihat pada gambar 23.

Hasil 42 kerentan dengan tingkat *high* memiliki tipe solusi vendor fix. Tipe solusi vendor fix yaitu dari setiap kerentan yang didapatkan ada tersedia informasi tentang perbaikan resmi yang dikeluarkan oleh pembuat asli produk. Dapat dilihat pada gambar 21.

Hasil 32 kerentanan dengan tingkat *medium* memiliki tipe solusi *workaround* dan *mitigation*. Tipe solusi *workaround* terdapat informasi tentang skenario spesifik konfigurasi atau pengembangan untuk menghindari adanya serangan ancaman dan ini merupakan langkah awal untuk *defence* terhadap *vulnerability*. Kemudian, untuk tipe solusi mitigation terdapat informasi sebuah skenario pengembangan dan konfigurasi untuk membantu dan meminimalisir tingkat resiko ancaman dari kerentanan akan tetapi solusi ini tidak menyelesaikan dampak kerentanan pada produk tersebut. Dapat dilihat pada gambar 21.

5. KESIMPULAN

Berdasarkan hasil pengamatan pada penggunaan Greenbone Vulnerability Management yang mana telah berhasil mengidentifikasi kerentanan sistem dengan menemukan beragam variasi dan tingkat kerentanan.

Hal tersebut merupakan suatu bentuk tindakan preventif dan meningkatkan kesadaran akan pentingnya keamanan sistem yang dilakukan oleh penulis melalui penerapan *vulnerability assessment* yang di wadai Greenbone Vulnerability Management (GVM).

6. DAFTAR PUSTAKA

[1] R. S. Lumbu, "PRAKTIKUM KEAMANAN JARINGAN," 2018.

[2] A. Analysis and U. Manual, "User Manual Greenbone Security Manager," *Data Base*, vol. 3304, no. January, pp. 1–148, 2012.

[3] B. Schneider and P. M. Asprion, "Testing a Vulnerability Scanner Greenbone Community Edition," no. August, 2020.

[4] "LIFECYCLE, GREENBONE OPERATING SYSTEM: ROADMAP &," *greenbone.net*, 2021, [Online]. Available: https://www.greenbone.net/en/roadmap-lifecycle/.

[5] "GVM Architecture," *greenbone.github.io*, 2021, [Online]. Available: https://greenbone.github.io/docs/architecture.html.

[6] "VULNERABILITY ASSESSMENT," *solvit.rs*. https://solvit.rs/vulnerability-assessment/.

KERTAS KERJA

1. Jaringan Komputer

Jaringan komputer yaitu dua atau lebih komputer yang saling berhubungan melalui media perantara sehingga dapat berbagi sumber daya atau resource [1]. Media perantara ini dapat berupa media kabel atau *wired* dan media tanpa kabel atau nirkabel. Berdasarkan fungsinya jaringan komputer dapat dibagi menjadi dua jenis antara lain:

a. *Client Server*

Client server adalah jaringan komputer yang salah satu atau boleh lebih komputer difungsikan sebagai *server* komputer lain. *Server* melayani komputer lain yang disebut klien. *Client server* banyak dipakai pada internet, namun jaringan lokal juga dapat diimplementasikan *client server*. Hal ini sangat bergantung pada kebutuhan masing-masing.

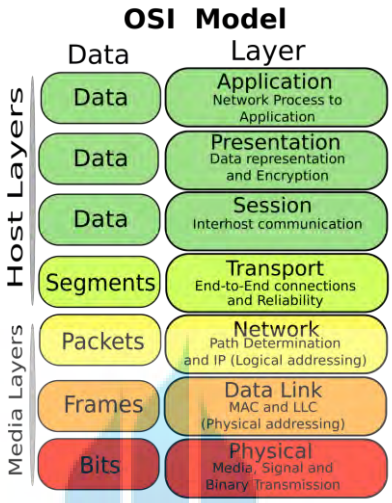
b. *Peer to Peer*

Jaringan *peer to peer* atau juga disebut P2P yaitu satu atau lebih komputer yang saling terhubung baik media atau nirkabel dan tiap komputer dapat berkomunikasi secara langsung. Pada jaringan P2P, sebuah komputer dapat menjadi *client* sekaligus *server* pada saat yang bersamaan dan tidak ada autentikasi secara terpusat. Autentikasi dapat diatur di setiap *node* yang memberikan layanan.

2. Model OSI

Model OSI dibuat untuk mengatasi kendala *internetworking* akibat perbedaan arsitektur dan protokol jaringan. Dahulu, komunikasi antar komputer dari vendor yang berbeda sangat sulit dilakukan. Masing-masing vendor menggunakan protokol dan format data yang berbeda sehingga *International for Standardization* (ISO) membuat sebuah arsitektur komunikasi yang dikenal sebagai *Open System Interconnection* (OSI) model yang mendefinisikan standar untuk menghubungkan komputer-komputer dari vendor yang berbeda. OSI model terdiri dari tujuh *layer* dan secara umum dibagi menjadi dua kelompok yaitu *upper layer* (*application layer*) dan *lower layer* (*data transport layer*). *Layer* yang tergolong *upper layer*

mendefinisikan bagaimana aplikasi sebuah *host* akan berkomunikasi dengan *user* dan *host* lainnya. Sedangkan *lower layer* mendefinisikan bagaimana data terkirim dari *host* satu ke *host* lainnya. Proses komunikasi dan pertukaran informasi dari tiap-tiap layer menggunakan sebuah protokol yang disebut *Protocol Data Unit* (PDU).



Gambar 1. OSI Layer

Secara umum, fungsi dari tujuh bagian OSI ini adalah:

a. *Layer* 7 (*Application Layer*)

Berfungsi sebagai antar muka (penghubung) aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan dan kemudian membuat pesan kesalahan. Pada *layer* ini user berinteraksi dengan jaringan. Contoh protokol yang berada pada *layer* ini adalah FTP, SMTP, HTTP, POP3.

b. *Layer* 6 (*Presentation Layer*)

Berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi dalam format yang dapat ditransmisikan melalui jaringan.

c. *Layer* 5 (*Session Layer*)

Berfungsi untuk mendefinisikan bagaimana koneksi dimulai, dipelihara dan diakhiri.

d. *Layer* 4 (*Transport Layer*)

Berfungsi untuk memecah data menjadi paket-paket data serta memberikan nomor urut setiap paket sehingga dapat disusun kembali setelah

diterima paket. paket yang diterima dengan sukses akan diberi tanda (*acknowledgement*). Sedangkan paket yang rusak atau hilang ditengah jalan akan dikirim ulang. Contoh protokol yang digunakan TCP dan UDP.

e. *Layer* 3 (*Network*)

Berfungsi mendefinisikan alamat-alamat IP, membuat header untuk paket-paket dan melakukan *routing* melalui *internetworking* dengan menggunakan *router* dan *switch layer* 3.

f. Layer 2 (Data Link)

Berfungsi untuk menentukan bagaimana bit-bit data dikelompokkan menjadi frame. Pada level ini terjadi *error correction*, *flow control*, pengalamatan perangkat keras (*MAC Address*) dan menentukan bagaimana perangkat-perangkat jaringan seperti bridge dan *switch layer* 2 beroperasi.

g. *Layer* 1 (*Physical*)

Berfungsi mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan, topologi jaringan dan pengkabelan. Selain itu level ini juga mendefinisikan bagaimana *network interface card* (NIC) berinteraksi dengan media *wire* atau *wireless*.

3. Keamanan Sistem Informasi

Keamanan informasi adalah perlindungan informasi dan sistem informasi dari akses pihak yang tidak berwenang, penggunaan, gangguan, modifikasi atau bahkan perusakan. Keamanan informasi mengacu pada setiap kegiatan yang dirancang untuk melindungi sistem informasi. Kegiatan ini terdiri dari teknologi dan proses yang dilakukan untuk melindungi sistem komputer dari ancaman internal atau ancaman eksternal. Sistem keamanan informasi melibatkan semua organisasi, perusahaan dan lembaga untuk melindungi aset demi kelangsungan perusahaan [2].

4. *Virtual Machine*

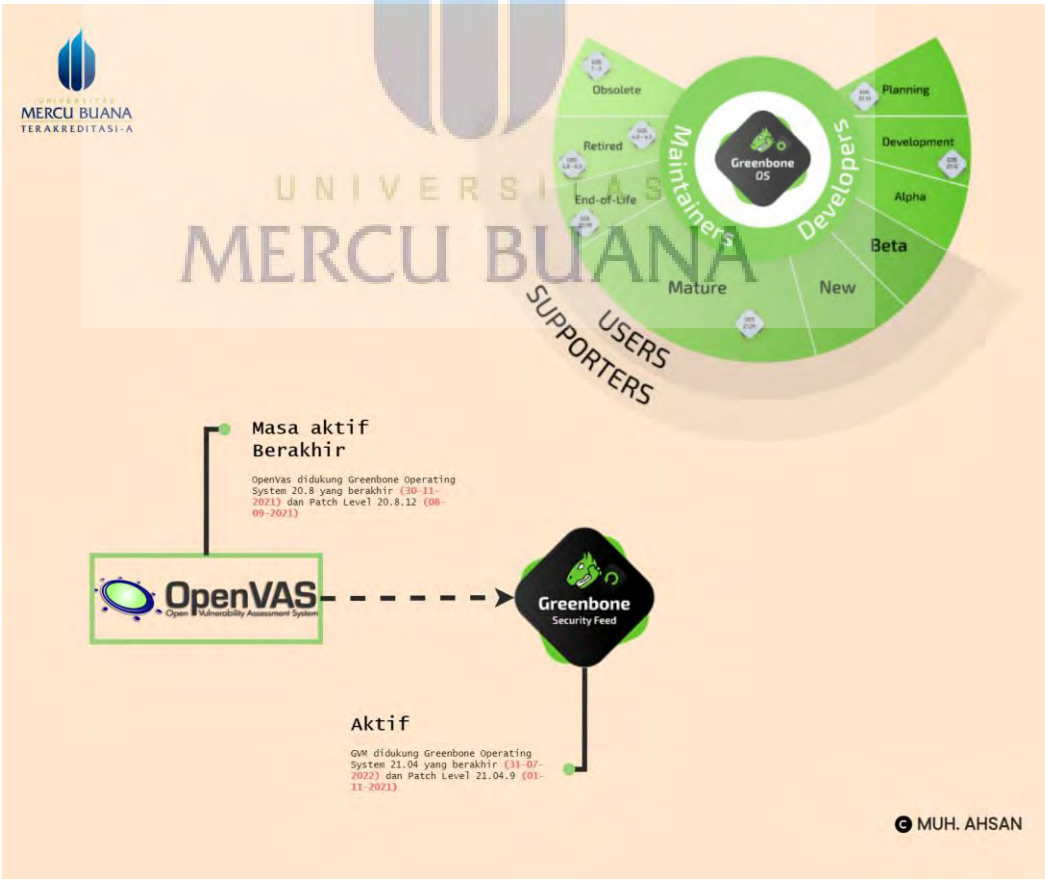
Virtual Machine adalah rekayasa perangkat lunak yang memiliki fungsi hampir sama seperti komputer fisik yang dapat digunakan untuk menambah sistem operasi didalam sistem operasi utama atau dapat dikatakan sebagai tempat untuk melakukan simulasi sistem operasi. [3].

5. Nmap

Network Mapper merupakan *open source tool* untuk eksplorasi dan audit keamanan jaringan. Dirancang untuk memeriksa jaringan besar secara cepat, meskipun dapat pula bekerja terhadap single host. Nmap menggunakan paket IP raw untuk menentukan host mana saja yang tersedia pada jaringan, layanan nama aplikasi dan versi apa yang diberikan, sistem operasi dan versi apa yang digunakan, apa jenis *firewall* atau *filter* paket yang digunakan, dan sejumlah karakteristik lainnya [1]. Meskipun Nmap umumnya digunakan untuk audit keamanan, namun banyak *administrator* sistem dan jaringan menganggapnya berguna untuk tugas rutin seperti inventori jaringan, mengelola jadwal upgrade layanan, dan melakukan *monitoring up time host* atau layanan.

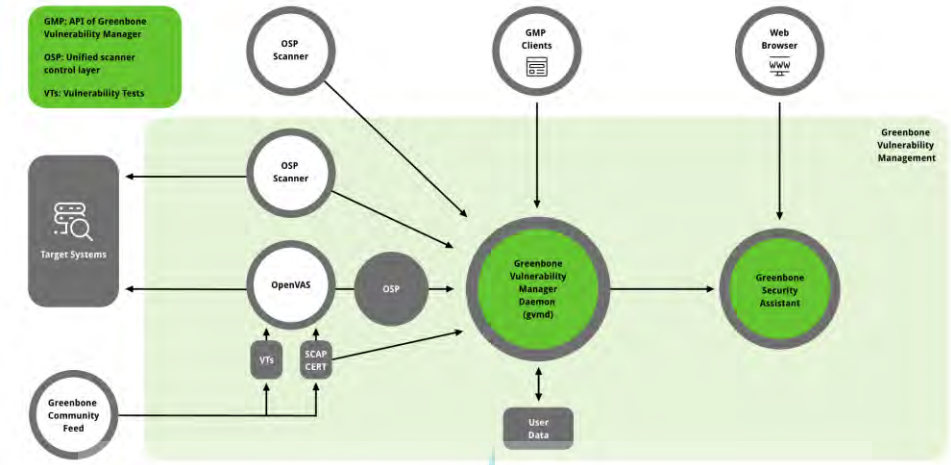
6. Greenbone Vulnerability Management (GVM)

Greenbone Vulnerability Management awalnya dibangun sebagai proyek komunitas yang bernama OpenVAS. GVM diteruskan dan dikembangkan oleh Greenbone Network [4] [5] [6]. Gambar berikut merupakan proses OpenVAS 9 menjadi GVM 11.



Gambar 2. OpenVAS 9 menjadi GVM 11

Gambar berikut merupakan arsitektur *patch level* GVM 21.04 secara umum.



Gambar 3. Arsitektur GVM 21.04

GMV dikelompokkan menjadi tiga bagian utama:

1. Aplikasi *scan* dapat melakukan eksekusi dengan menjalankan Vulnerability Test (VT) terhadap sistem target.
2. Greenbone Vulnerability Manager Daemon (gvmd)
3. Greenbone Security Assistant (GSA) dengan the Greenbone Security Assistant Daemon (gsad)

GVM memiliki komponen yang saling terkait satu dengan yang lain [7]. Komponen tersebut antara lain:

a. *Greenbone Vulnerability Manager Daemon* (gvmd)

Greenbone Vulnerability Manager Daemon (gvmd) adalah pusat layanan yang mengkonsolidasikan pemindaian kerentanan menjadi solusi manajemen kerentanan penuh. gvmd mengontrol OpenVAS Scanner melalui *Open Scanner Protocol* (OSP). Layanan itu sendiri menawarkan *Greenbone Management Protocol* (GMP) berbasis XML. gvmd juga mengontrol database SQL (PostgreSQL) tempat semua konfigurasi dan data hasil pemindaian disimpan secara terpusat. Selanjutnya, gvmd juga menangani manajemen pengguna termasuk kontrol izin dengan grup dan peran. Dan terakhir, layanan memiliki sistem runtime internal untuk tugas terjadwal dan acara lainnya.

b. *Greenbone Security Assistant* (GSA)

Greenbone Security Assistant (GSA) adalah web *interface* dari GVM yang mana pengguna mengatur *scan* dan mengakses informasi kerentanan secara bersama. Ini adalah titik kontak utama bagi pengguna dengan GVM. Terhubung ke gvmd melalui server web *Greenbone Security Assistant Daemon* (gsad) untuk menyediakan aplikasi web berfitur lengkap untuk manajemen kerentanan. Komunikasi terjadi menggunakan *Greenbone Management Protocol* (GMP) dimana pengguna juga dapat berkomunikasi secara langsung dengan menggunakan alat yang berbeda.

c. *OpenVAS Scanner*

OpenVAS Scanner adalah mesin pemindai berfitur lengkap yang menjalankan uji kerentanan (VT) terhadap sistem target. Untuk ini, ia menggunakan *feed* harian yang diperbarui dan komprehensif: *Greenbone Security Feed* (GSF) komersial berfitur lengkap, ekstensif, atau *Greenbone Community Feed* (GCF) yang tersedia gratis. *OpenVAS Scanner* terdiri dari komponen ospd-openvas dan openvas-scanner Pemindai OpenVAS dikendalikan melalui OSP. OSP Daemon untuk *OpenVAS Scanner* (ospd-openvas) berkomunikasi dengan gvmd melalui OSP: Data VT dikumpulkan, pemindaian dimulai dan dihentikan, dan hasil pemindaian ditransfer ke gvmd melalui ospd.

7. Istilah Dalam Greenbone

1. OSP *Scanner*

Pengguna dapat mengembangkan dan menghubungkan pemindai OSP mereka sendiri menggunakan kerangka kerja pemindai ospd generik. Contoh pemindai OSP (umum) yang dapat digunakan sebagai *template* pemindai OSP dapat ditemukan di sini

2. GMP *Clients*

Greenbone Vulnerability Management Tools (gvm-tools) adalah kumpulan alat yang membantu dengan *remote* mengendalikan *Greenbone Security Manager* (GSM) alat dan yang mendasari *Greenbone Vulnerability Manager*

Daemon (gvmd). Alat bantu dalam mengakses protokol komunikasi GMP (*Greenbone Management Protocol*) dan OSP (*Open Scanner Protocol*).
Modul ini terdiri dari klien interaktif dan noninteraktif. Bahasa pemrograman Python didukung langsung untuk skrip interaktif. Tetapi juga dimungkinkan untuk mengeluarkan perintah GMP/OSP jarak jauh tanpa pemrograman dengan Python.

3. VT

Vulnerability Tests (VTs), juga dikenal sebagai *Network Vulnerability Tests* (NVTs), adalah skrip yang ditulis dalam bahasa pemrograman NASL untuk mendeteksi kerentanan pada host jarak jauh.

4. OSPd

Sebuah framework untuk beberapa *scanner daemon*.

5. ospd-openvas

OSP scanner daemon mengelola *OpenVAS executable* untuk melaporkan hasil *scan* kepada manajemen daemon gvmd. Digunakan di GVM 11 dan yang lebih baru.

6. GOS

Greenbone Operating System adalah sistem operasi dari Greenbone Security Management (GSM).

7. GMP

Greenbone Management Protocol, protokol komunikasi berbasis XML yang disediakan oleh gvmd.

8. GPE

Greenbone Professional Edition (GPE) adalah lini produk Greenbone untuk solusi ditempat.

9. GSF

Greenbone Security Feed (GSF) adalah umpan komersial yang disediakan oleh Greenbone Networks yang berisi fitur perusahaan tambahan seperti pemeriksaan kebijakan dan kepatuhan, format laporan ekstensif, dan

konfigurasi pemindaian khusus. Umpan dilengkapi dengan perjanjian tingkat layanan yang memastikan dukungan, jaminan kualitas, dan ketersediaan.

10. GCF

Greenbone Community Feed (GCF) adalah *feed* yang tersedia secara gratis untuk informasi kerentanan yang dilisensikan sebagai *Open Source*. Ini berisi konfigurasi pemindaian dasar, format laporan, daftar *port* dan tes kerentanan yang paling penting. Data yang diberikan diperbarui setiap hari tanpa jaminan atau janji untuk perbaikan atau kelengkapan.

8. Istilah Pada Fitur GVM

1. *Alert*

Alert adalah tindakan yang dapat dipicu oleh peristiwa tertentu. Dalam kebanyakan kasus, ini berarti pemberitahuan, misalnya, email jika ditemukan kerentanan baru.

2. *Asset*

Aset ditemukan di jaringan selama pemindaian kerentanan atau dimasukkan secara manual oleh pengguna. Saat ini, aset termasuk host dan sistem operasi.

3. CERT-Bund Advisory

Sebuah *advisory* yang diterbitkan oleh CERT-Bund.

4. *Compliance Audit*

Compliance Audit adalah tugas pemindaian dengan audit bendera dan digunakan untuk memeriksa pemenuhan kepatuhan.

5. *Compliance Policy*

Compliance Policy adalah konfigurasi pemindaian dengan *flag policy* dan digunakan untuk memeriksa pemenuhan kepatuhan.

6. CPE

Common Platform Enumeration (CPE) adalah skema penamaan terstruktur untuk sistem, platform, dan paket teknologi informasi. Berdasarkan sintaks generik untuk *Uniform Resource Identifiers* (URI), CPE mencakup format nama formal, bahasa untuk menggambarkan platform yang kompleks, metode

untuk memeriksa nama terhadap sistem dan format deskripsi untuk mengikat teks dan tes ke nama.

7. CVE

Common Vulnerabilities and Exposures (CVE) adalah kamus kerentanan dan eksposur keamanan informasi yang diketahui publik.

8. CVSS

Common Vulnerability Scoring System (CVSS) adalah kerangka kerja terbuka untuk mengkarakterisasi kerentanan.

9. DFN-CERT Advisory

Sebuah *advisory* yang diterbitkan oleh DFN-CERT.

10. *Filter*

Filter menjelaskan cara memilih subset tertentu dari sekelompok sumber daya.

11. *Group*

Grup adalah sekumpulan pengguna.

12. *Host*

Host adalah sistem tunggal yang terhubung ke jaringan komputer dan dapat dipindai. Satu atau banyak *host* membentuk dasar dari target pemindaian. *Host* juga merupakan jenis aset. Setiap host yang dipindai atau ditemukan dapat direkam dalam aset *database*. *Host* dalam target pemindaian dan dalam laporan pemindaian diidentifikasi berdasarkan alamat jaringannya, baik alamat IP atau nama *host*. Dalam database aset, identifikasi tidak tergantung pada alamat jaringan yang sebenarnya, yang bagaimanapun juga digunakan sebagai identifikasi *default*.

13. *Note*

Note adalah komentar tekstual yang terkait dengan VT. Catatan muncul di laporan, di bawah hasil yang dihasilkan oleh VT. Catatan dapat diterapkan ke hasil, tugas, tingkat keparahan, *port* dan/atau *host* tertentu, sehingga catatan hanya muncul di laporan tertentu.

14. *Oval Definition*

Definisi OVAL adalah definisi yang ditentukan oleh OVAL (*Open Vulnerability and Assessment Language*), versi 5.10.1. Ini dapat digunakan untuk berbagai kelas data keamanan seperti kerentanan, patch, atau kebijakan kepatuhan.

15. *Override*

Override adalah aturan untuk mengubah tingkat keparahan item dalam satu atau banyak laporan.

16. *Permission*

Izin memberi pengguna, peran, atau grup hak untuk melakukan tindakan tertentu.

17. *Port List*

Port List adalah daftar *port*. Setiap target dikaitkan dengan daftar *port*. Ini menentukan *port* mana yang dipindai selama pemindaian target.

18. *Quality of Detection* (QoD)

Quality of Detection (QoD) adalah nilai antara 0% dan 100% yang menggambarkan keandalan deteksi kerentanan atau deteksi produk yang dijalankan. Nilai 70% adalah nilai minimum *default* yang digunakan untuk memfilter hasil yang ditampilkan dalam laporan.

19. *Remediation Ticket*

Remediation Ticket digunakan untuk menyelesaikan temuan kerentanan. Tiket dapat diberikan ke pengguna saat ini atau pengguna lain. Semua informasi berharga untuk memahami dan menyelesaikan masalah secara langsung saling terkait dan tersedia untuk pengguna yang ditugaskan.

20. *Report*

Report adalah hasil pemindaian dan berisi ringkasan tentang apa yang terdeteksi oleh VT yang dipilih untuk setiap *host* target.

21. *Report Format*

Format di mana laporan dapat diunduh. Contohnya adalah TXT yang memiliki tipe konten "*text/plain*", artinya laporan tersebut merupakan dokumen teks biasa.

22. *Result*

Hasil tunggal yang dihasilkan oleh pemindai sebagai bagian dari laporan, misalnya peringatan kerentanan atau pesan *log*.

23. *Role*

Role menentukan sekumpulan izin yang dapat diterapkan ke pengguna atau grup.

24. *Scan*

Scan adalah *task* yang sedang berlangsung. Untuk setiap *task*, hanya satu pemindaian yang dapat aktif. Hasil scan adalah laporan.

25. *Scanner*

Scanner adalah OpenVAS Scanner daemon atau OSP daemon yang kompatibel di mana pemindaian akan dijalankan.

26. *Scan Configuration*

Scan Configuration mencakup pemilihan VT serta parameter umum dan sangat spesifik (ahli) untuk server pemindaian dan untuk beberapa VT.

27. *Schedule*

Jadwal menetapkan waktu kapan tugas harus dimulai secara otomatis, periode setelah tugas harus dijalankan lagi, dan durasi maksimum tugas yang diizinkan.

28. *Severity*

Severity adalah nilai antara 0,0 (*no severity*) dan 10,0 (*highest severity*) dan juga menyatakan kelas keparahan (*Log*, Low, *Medium*, dan *High*).

Konsep ini didasarkan pada CVSS tetapi diterapkan jika tidak ada CVSS Base Vector lengkap yang tersedia juga. Misalnya, nilai arbitrer dalam rentang tersebut diterapkan untuk penggantian dan digunakan oleh pemindai OSP bahkan tanpa definisi vektor.

Perbandingan, pembobotan, dan prioritas hasil pemindaian atau VT apa pun dimungkinkan karena konsep keparahan diterapkan secara ketat di seluruh

sistem. Setiap VT baru ditetapkan dengan vektor CVSS penuh bahkan jika CVE tidak menawarkannya dan setiap hasil pemindai OSP diberi nilai keparahan yang memadai bahkan jika pemindai masing-masing menggunakan skema keparahan yang berbeda.

Kelas keparahan *Log*, *Low*, *Medium*, dan *High* ditentukan oleh sub-rentang dari kisaran utama 0,0-10,0. Pengguna dapat memilih untuk menggunakan klasifikasi yang berbeda. Standarnya adalah klasifikasi NVD yang paling umum digunakan.

Hasil pemindaian diberi tingkat keparahan saat dicapai. Tingkat keparahan VT terkait dapat berubah seiring waktu. Jika tingkat keparahan dinamis dipilih dalam pengaturan pengguna, sistem selalu menggunakan tingkat keparahan VT terbaru untuk hasilnya [8].

29. *Solution Type*

Informasi ini menunjukkan kemungkinan solusi untuk perbaikan kerentanan.

a. Solusi: Informasi tentang konfigurasi atau skenario penerapan tertentu yang dapat digunakan untuk menghindari paparan kerentanan tersedia. Tidak ada satu pun, satu atau lebih solusi yang tersedia. Ini biasanya merupakan "garis pertahanan pertama" terhadap kerentanan baru sebelum mitigasi atau perbaikan vendor dikeluarkan atau bahkan ditemukan.
b. Mitigasi: Informasi tentang konfigurasi atau skenario penyebaran yang membantu mengurangi risiko kerentanan tersedia tetapi tidak menyelesaikan kerentanan pada produk yang terpengaruh. Mitigasi dapat mencakup penggunaan perangkat atau kontrol akses eksternal ke produk yang terpengaruh. Mitigasi mungkin atau mungkin tidak dikeluarkan oleh penulis asli dari produk yang terpengaruh dan mereka mungkin atau mungkin tidak secara resmi disetujui oleh produsen dokumen.
c. Perbaikan vendor: Tersedia informasi tentang perbaikan resmi yang dikeluarkan oleh pembuat asli produk yang terpengaruh. Kecuali

dinyatakan lain, diasumsikan bahwa perbaikan ini sepenuhnya menyelesaikan kerentanan.

d. Tidak ada perbaikan yang tersedia: Saat ini tidak ada perbaikan yang tersedia. Informasi harus berisi detail tentang mengapa tidak ada perbaikan.
e. Tidak akan diperbaiki: Tidak ada perbaikan untuk kerentanan dan tidak akan pernah ada. Ini sering terjadi ketika suatu produk telah menjadi yatim piatu, tidak lagi dipertahankan atau tidak digunakan lagi. Informasi harus berisi detail tentang mengapa tidak ada perbaikan yang dikeluarkan.

30. *Tag*

Tag adalah paket data pendek yang terdiri dari nama dan nilai yang dilampirkan ke sumber daya apa pun dan berisi informasi yang ditentukan pengguna pada sumber daya ini.

31. *Target*

Target mendefinisikan satu *set* sistem (*host*) yang dipindai. Sistem diidentifikasi baik dengan alamat IP mereka, dengan nama host mereka atau dengan notasi jaringan CIDR.

32. *Task*

Tugas awalnya dibentuk oleh target dan konfigurasi pemindaian. Menjalankan tugas memulai pemindaian. Setiap pemindaian menghasilkan laporan. Akibatnya, tugas mengumpulkan serangkaian laporan.

33. TSL *Certificate*

Sertifikat TLS (*Transport Layer Security*) adalah sertifikat yang digunakan untuk otentikasi saat membuat koneksi yang diamankan oleh TLS.

Laporan pemindaian berisi semua sertifikat TLS yang dikumpulkan selama pemindaian kerentanan.

9. *Vulnerability Scanning*

Ada beberapa langkah berbeda yang terlibat dalam proses *scanning* kerentanan dan beberapa di antaranya tercantum di bawah ini:

a. Titik akhir dalam sistem dipindai dengan mengirimkan paket TCP atau UDP antara titik sumber dan tujuan dengan melakukan ping ke alamat IP.
b. Pemindaian dilakukan untuk menemukan port terbuka dan layanan yang berjalan pada sistem
c. Proses pemindaian menjalankan program yang berinteraksi dengan aplikasi web untuk menemukan kemungkinan kerentanan yang terletak pada arsitektur jaringan.
d. Proses mencari program yang tidak diinginkan yang diinstal di sistem, patch yang hilang, dan validasi konfigurasi yang dilakukan di sistem.

Gambar 4. Proses Scanning Vulnerability

Namun, melakukan proses pemindaian kerentanan juga memiliki beberapa risiko, karena terkadang saat menjalankan pemindaian di komputer, sistem akan *reboot* berulang kali dan bahkan ada kemungkinan beberapa sistem juga mengalami *crash* [9].

10. *Vulnerability Assessment*

Vulnerability Assessment melakukan identifikasi kerentanan dari suatu aplikasi, sistem operasi dan infrastruktur jaringan. *Vulnerability Assessment* tidak

melakukan eksploitasi celah atau kelemahan dari suatu sistem lebih fokus untuk menemukan beragam public vulnerability pada seluruh sistem komputer dalam jaringan target dan tidak menuju ke proses eksploitasi namun memiliki potensi untuk dieksploitasi sehingga harus ditutup kerentanan yang ditemukan. [10].

Tahapan yang dilakukan dalam kegiatan *vulnerability assessment* antara lain:

1. *Adjusting Scope*

Adjusting scope adalah proses untuk menentukan batas apa saja yang termasuk dalam proses uji penetrasi. Seperti contoh batas jaringan, alamat IP, server dan sebagainya.

2. *Target Enumeration*

Target enumeration merupakan aktivitas selanjutnya untuk mengidentifikasi topologi jaringan yang tepat dan sistem operasi serta versi aplikasi dan juga *port* yang terbuka pada sistem.

3. *Result Analysis Assessment*

Tahap ini untuk menentukan komponen jaringan baik *router*, *firewall*, *server*, IDS/IPS untuk mengenali kerentanan dan menetapkan level resiko untuk setiap kerentanan. Level kerentanan dibagi menjadi tiga bagian yaitu *low*, *medium*, dan *high* serta melakukan analisa solusi kebijakan mitigasi yang sesuai dan tepat. Hal yang dilakukan antara lain hardening sistem, menerapkan, patch serta membuat kebijakan keamanan.

4. *Report Finding*

Laporkan kerentanan yang teridentifikasi termasuk peringkat dampak dan tindakan yang direkomendasikan untuk memitigasinya.

Gambar 5. Proses Vulnerability Assessment

11. *Vulnerability Management*

    1. *Vulnerability Identification*

    Tujuan dari langkah ini adalah untuk menyusun daftar lengkap kerentanan aplikasi. Analis keamanan menguji kesehatan keamanan aplikasi, server, atau sistem lain dengan memindainya dengan alat otomatis, atau menguji dan mengevaluasinya secara manual. Analisa ini juga mengandalkan *database* kerentanan, pengumuman kerentanan vendor, sistem manajemen aset, dan umpan intelijen ancaman untuk mengidentifikasi kelemahan keamanan.

    2. *Vulnerability Analysis* (*Vulnerability Scanning*)

    Tujuan dari langkah ini adalah untuk mengidentifikasi sumber dan akar penyebab kerentanan yang diidentifikasi pada langkah pertama.

    Ini melibatkan identifikasi komponen sistem yang bertanggung jawab untuk setiap kerentanan, dan akar penyebab kerentanan.

    3. *Risk Assessment* (*Vulnerability Assessment*)

Tujuan dari langkah ini adalah memprioritaskan kerentanan. Ini melibatkan analis keamanan yang menetapkan peringkat atau skor keparahan untuk setiap kerentanan, berdasarkan faktor-faktor seperti:

1. Sistem mana yang terpengaruh.
2. Data apa yang berisiko.
3. Fungsi bisnis mana yang berisiko.
4. Kemudahan menyerang atau kompromi.
5. Tingkat keparahan serangan.
6. Potensi kerusakan sebagai akibat dari kerentanan.

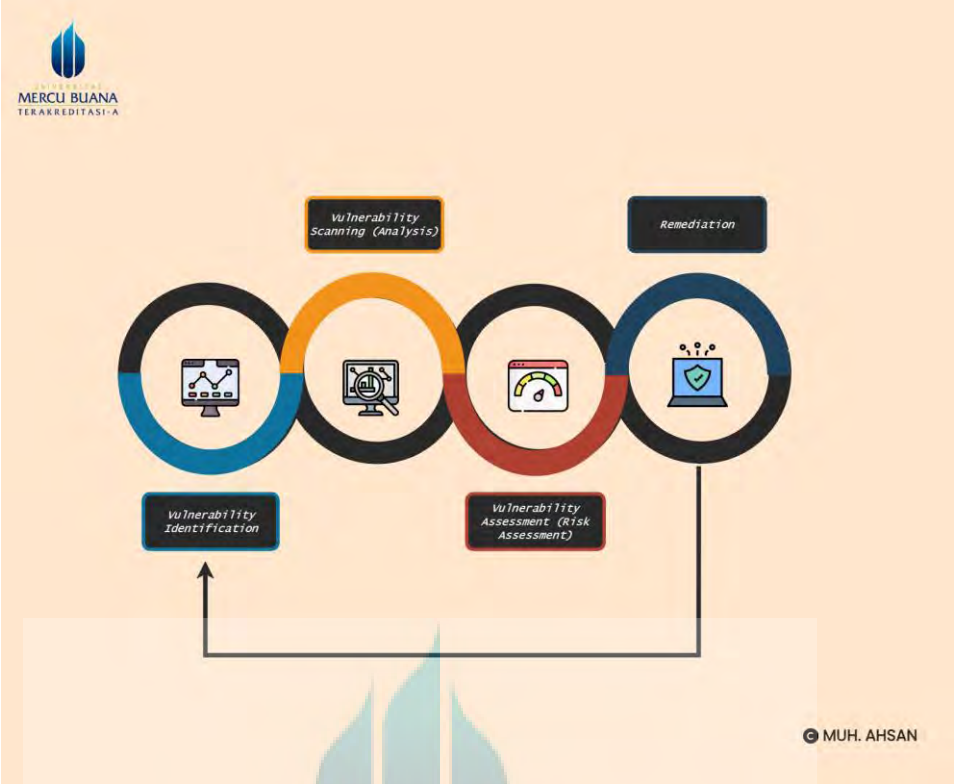
4. *Remediation*

Tujuan dari langkah ini adalah untuk menutup celah keamanan. Ini biasanya merupakan upaya bersama oleh staf keamanan, tim pengembangan dan operasi, yang menentukan jalur paling efektif untuk remediasi atau mitigasi setiap kerentanan.

Langkah-langkah perbaikan khusus mungkin termasuk:

1. Pengenalan prosedur, tindakan, atau alat keamanan baru.
2. Pembaruan perubahan operasional atau konfigurasi.
3. Pengembangan dan implementasi patch kerentanan.

Penilaian kerentanan tidak bisa menjadi kegiatan satu kali saja. Agar efektif, organisasi harus mengoperasionalkan proses ini dan mengulanginya secara berkala. Penting juga untuk mendorong kerja sama antara tim keamanan, operasi, dan pengembangan-sebuah proses yang dikenal sebagai DevSecOps [9] [11].



Gambar 6. Proses Vulnerability Management