BAB 2
LANDASAN TEORI

2.1 Studi Literatur

Studi literatur pada penelitian ini menggunakan tujuh jurnal dan tiga skripsi yang diambil dari berbagai sumber di internet. Tabel 2.1 dibawah ini merupakan hasil dari studi literatur yang telah dilakukan.

Tabel 2.1: Studi Literatur

No	Peneliti	Judul	Tahun	Hasil
1	I Dewa Gede Govindha Dharmawangsa, Gusti Made Arya Sasmita, dan I Putu Agus Eka Pratama	*Penetration testing* Berbasis OWASP *Guide* Versi 4.2 (Studi Kasus: X *Website*)	2023	Peneliti berhasil melakukan *penetration testing* pada *website* X dengan menggunakan OWASP *Testing Guide* v4.2. Dari penemuan kerentanan tersebut, dilakukan pula penilaian menggunakan CVSS v3.1 dan berhasil ditemukan bahwa terdapat total 11 kerentanan yang bersifat aktif.

Continued on next page

Tabel 2.1: Studi Literatur (Continued)

2	Addi Amalana Arafat	*Penetration testing* pada *Website Registrar* Pengelola Nama Internet Indonesia (PANDI)	2020	Peneliti berhasil melakukan *penetration testing* menggunakan metode *black box testing* untuk menemukan celah keamanan pada *website registrar* PANDI. Dari hasil *penetration testing* yang dilakukan, terdapat sebuah kerentanan yang dapat berpengaruh kepada 7 dari 10 *website* yaitu adalah *x-header frame not set* yang dapat mengarah kepada serangan *clickjacking.*
3	Syarif Hidayatulloh dan Desky Saptadiaji	*Testing* pada *Website* Universitas ARS Menggunakan *Open Web Application Security Project* (OWASP)	2021	Keamanan *website* Universitas ARS secara keseluruhan sudah baik, tetapi masih terdapat beberapa kerentanan yang perlu diperbaiki. Dari hasil *penetration testing* yang dilakukan, ditemukan 13 kerentanan pada *website* Universitas ARS. Dari jumlah tersebut, 12 kerentanan berada pada tingkat ancaman yang rendah dan 1 kerentanan berada pada tingkat ancaman yang sedang.

Continued on next page

Tabel 2.1: Studi Literatur (Continued)

4	Muhammad Yaqi	*Vulnerability Assessment* dan *Penetration Testing* (Vapt) Menggunakan Metode *Zero Entry Hacking* (Zeh) Terhadap Studi Kasus: Dinas Penanaman Modal dan PTSP Kota Tangerang Selatan	2023	Peneliti telah berhasil melakukan *vulnerability assessment* dan *penetration testing* menggunakan metode *zero entry hacking* kepada *website* milik Dinas Penanaman Modal dan PTSP Kota Tangerang Selatan. Melalui tahapan pada metode *Zero Entry Hacking* (ZEH) berhasil ditemukan 7 celah kerentanan, hanya saja peneliti tidak berhasil melakukan tahap terakhir dalam ZEH yaitu *post exploitation* dikarenakan tidak terdapat fitur untuk menaruh *file* pada *website* tersebut.

Continued on next page

Tabel 2.1: Studi Literatur (Continued)

5	Fathurrachman	Pengujian Kerentanan Log4shell Pada *Website E-Commerce* Menggunakan Metode *Vulnerability Assessment And Penetration Testing (Vapt) Life Cycle*	2023	Peneliti berhasil melakukan VAPT *Life cycle* pada *website e-commerce*. Dari proses tersebut terdapat kerentanan yang memiliki tingkat *severity low* berjumlah 13 dengan *level confidence* yang berbeda diantaranya *certain, firm*, dan *tentative*. Pada kerentanan yang memiliki tingkat *severity high* berjumlah 23 dengan *level confidence* yang berbeda seperti *certain, firm* dan *tentative*.
6	Bagus Wicaksono, Rr. Yuliana Rachmawati Kusumaningsih, dan Catur Iswahyudi	Pengujian Celah Keamanan Aplikasi Berbasis *Web* menggunakan Teknik *Penetration Testing* dan Dast (*Dynamic Application Security Testing*)	2021	Penelitian ini menggunakan metode DAST untuk menguji celah keamanan pada *website bagusw.win*. Hasil penelitian menunjukkan bahwa *website* tersebut rentan terhadap celah keamanan *Cross Site Scripting, Broken Access Control, dan SQL Injection*.

Continued on next page

Tabel 2.1: Studi Literatur (Continued)

7	Widi Linggih Jaelani, Yanto, dan Fitri Khoirunnisa	*Penetration Testing Website* Dengan Metode *Black Box Testing* Untuk Meningkatkan Keamanan *Website* Pada Instansi (*Redacted*)	2023	Untuk hasil dari proses *information gathering* menggunakan Nmap, ditemukan bahwa terdapat 3 *port* yang terbuka yaitu *port* 22, 80, dan 8000. Kemudian peneliti menemukan kerentanan *clickjacking* menggunakan *tools clickjacker* yang dimana ketika dihitung menggunakan CVSS v3.1, kerentanan ini masuk ke dalam kategori *low* dengan skor 3.1. Selain itu, peneliti juga menemukan kerentanan *brute force directory* dengan menggunakan *tools* Dirb. Kerentanan *brute force directory* ini ketika dihitung menggunakan CVSS v3.1 mendapatkan skor 3.7 yang mana termasuk ke dalam kategori *low*.

Continued on next page

Tabel 2.1: Studi Literatur (Continued)

8	Tara Rizkayanti, dan Yunanri. W	Analisis Keamanan *Website* Sistem Informasi Administrasi Kependudukan Menggunakan Metode *Vulnerability Assesment*	2023	Peneliti berhasil menggunakan metode *vulnerability assessment* untuk melakukan pengujian keamanan pada *website* SIAK. Dari pengetesan yang dilakukan, terdapat 14 kerentanan yang ditemukan. Dari 14 kerentanan tersebut, 3 masuk ke dalam kategori *medium*, dan 11 masuk kedalam kategori *low*.
9	Marzuki Hasibuan	*Penetration Testing* Sistem Jaringan Komputer Menggunakan Kali Linux Untuk Mengetahui Kerentanan Keamanan *Server* Dengan Metode *Black Box* Studi Kasus *Web Server Diva Karaoke.co.id*	2022	Peneliti berhasil menggunakan metode *black box penetration testing* untuk menguji kerentanan pada *website* milik diva karaoke. Dari pengujian yang dilakukan ditemukan terdapat 6 *port* yang masih terbuka, dan juga dari proses *scanning vulnerability* ditemukan bahwa terdapat 3 kerentanan yang dapat mengarah kepada serangan *clickjacking* dan *sniffing*.

Continued on next page

Tabel 2.1: Studi Literatur (Continued)

10	Setyo Utoro, Bayu Andi Nugroho, Meinawati, dan Septian Rheno Widianto	Analisis Keamanan *Website E-Learning* SMKN 1 Cibatu Menggunakan Metode *Penetration Testing Execution Standard*	2020	Berdasarkan hasil penelitian ini. dapat disimpulkan bahwa analisis kerentanan aplikasi *website* milik SMKN 1 Cibatu dengan menggunakan metode PTES (*Penetration Testing Execution Standard*) mampu mengetahui tingkat kerentanan sistem informasi dengan risiko serangan yang paling tinggi seperti *Cross Site Scripting, Cross Site Request Forgery* dan *Eavesdropping* yang sangat berpotensi mengakibatkan kebocoran data penting. Melalui tahapan pengujian keamanan yang telah dilakukan, maka metode PTES dinilai dapat dijadikan sebagai standar penilaian keamanan aplikasi berbasis *web*pada *website-learning* di alamat *belajar.smkn1cibatu.sch.id* yaitu mulai dari tahap *pre-engagement interactions* hingga *reporting*

2.2 Vulnerability Assesment

Menurut Engebretson (2013), *vulnerability assesment* atau penilaian celah kerentanan merupakan suatu rangkaian proses yang dilakukan untuk meninjau *services* dan *system* yang memiliki potensi celah kerentanan [10].

Menurut pendapat lain oleh Baloch (2014), *vulnerability assesment* yaitu mencari tahu semua kerentanan dalam aset dan dokumentasikan sesuai dengan itu. Definisi lainnya, *vulnerability assesment* merupakan tahap dalam menemukan analisis uji kerentanan [11]. Jadi dapat disimpulkan bahwa *vulnerability assesment* merupakan tahapan yang terdiri dari serangkaian kegiatan untuk mencari tahu semua kerentanan dalam asset baik berupa *services* dan *system* dan di dokumentasikan.

Dalam melakukan *vulnerability assesment* dapat dipermudah salah satunya dengan menggunakan *tools* yang tersedia untuk melakukan pengujian kerentananan. Dalam penelitian ini menggunakan beragam *tools* yang siap digunakan dalam pengujian kerentanan. Adapun *tools* yang digunakan dalam *vulnerability assesment* penelitian ini sebagai berikut :

1. Owasp Zap

   Owasp Zap (*Zed Attack Proxy*) adalah alat pengujian penetrasi *web* gratis dan *open-source* yang membantu pengembang dan profesional keamanan menemukan dan mendeteksi kerentanan dalam aplikasi *web*. Alat ini bekerja dengan memindai permintaan *web* secara pasif, menggunakan daftar kamus untuk mencari *file* dan *folder* di server *web*, dan melakukan berbagai teknik pengujian lainnya. ZAP dapat digunakan untuk menguji berbagai jenis aplikasi *web*, termasuk aplikasi *web* yang kompleks dan modern [12].

2. Pentest-tools.com.

   Pentest-Tools.com adalah sebuah *platform* daring yang menyediakan sejumlah besar alat dan layanan yang dibutuhkan untuk melakukan pengujian penetrasi dalam lingkungan sistem dan jaringan komputer. Pentest-tools memiliki banyak alat yang membantu dalam mengidentifikasi, mengevaluasi, dan mengatasi celah keamanan dalam infrastruktur IT sebuah *website* [13].

2.3 Penetration Testing

Menurut Engebretson (2013), *penetration testing* merupakan sebuah percobaan yang legal dan diijinkan untuk melakukan eksploitasi terhadap sebuah sistem dengan tujuan meningkatkan kualitas keamanan dari sistem tersebut [10]. Dengan kata lain, *penetration testing* merupakan sebuah aktivitas pengujian keamanan dari sebuah sistem. Dari hasil pengujian tersebut, didapatkan sejumlah celah keamanan pada sistem yang kemudian menjadi bahan rekomendasi kepada organisasi yang memiliki sistem tersebut untuk dibenahi. Argumen lain menurut Weidman (2014), *penetration testing* meliputi simulasi serangan nyata untuk menilai risiko yang terkait dengan penerobosan keamanan yang sifatnya potensial [14].

Dengan uraian di atas, dapat disimpulkan bahwa *penetration testing* merupakan serangkaian kegiatan berupa simulasi yang dilakukan oleh pihak yang telah mendapatkan izin untuk melakukan eksploitasi suatu sistem berdasarkan *vulnerability assessment* dan berbeda dengan *illegal hacking* yang merusak sistem, karena *penetration testing* sudah memiliki izin untuk melakukan pengujian kerentanan kemudian dilakukan eksploitasi, selanjutnya dilakukan analisis terhadap hasil pengujian dan kemudian diberikan suatu rekomendasi mengenai bagaimana cara untuk membenahi celah kerentanan yang ada [15].

Menurut Jayasuryapal (2021), *penetration testing* dibagi menjadi 3 jenis yaitu [16]:

1. *White box*

   White box testing adalah kondisi dimana penguji mensimulasikan serangan dengan informasi lengkap tentang infrastruktur, detail sistem operasi, alamat IP, dan beberapa kata sandi. Ini dirancang untuk memungkinkan penguji melakukan serangan menggunakan pengetahuan yang akrab tentang sistem target organisasi, seperti detail pribadi seorang karyawan internal. Hal ini menjaga integritas infrastruktur jaringan organisasi dan mengurangi risiko penyerang internal, seperti karyawan yang tidak puas.

2. *Black box*

   Dalam *black box testing*, penguji mensimulasikan serangan tanpa informasi apapun tentang infrastruktur. Dengan cara ini, penguji menemukan semua kerentanan menggunakan metode dan alat mereka. Ini berarti bahwa penguji menggunakan berbagai teknik serangan nyata seperti rekayasa sosial dan

akses jarak jauh. Misalnya, penguji mendapatkan alamat IP jaringan tanpa informasi lain. Kemudian, penguji mensimulasikan semua teknik serangan untuk menemukan semua kerentanan yang diketahui dan tidak diketahui dalam jaringan.

3. *Gray box*

   Pendekatan *gray box* dilakukan ketika *white box* dan *black box* digabungkan dan digunakan bersama untuk menangkap informasi keamanan internal dan eksternal. Dengan cara ini, penguji memiliki beberapa informasi terbatas tentang infrastruktur jaringan.

Pada penelitian ini, penulis menggunakan metode *black box testing*, hal ini dikarenakan penulis ingin mensimulasikan serangan tanpa mengetahui infrastruktur dari *website UMN*. Menurut hukum di Indonesia yaitu UU ITE pasal 30 dan 31, ketika seseorang melakukan *penetration testing* tanpa meminta izin, merupakan suatu pelanggaran hukum [17]. Hanya saja menurut standar internasional seperti *Computer Fraud and Abuse Act* (CFAA) dan *General Data Protection Regulation* (GDPR), *penetration testing* dapat dilakukan dengan tujuan untuk penelitian keamanan. Hal ini tentunya dilakukan dengan cara yang proporsional dengan mempertimbangkan hak-hak individu dan dengan itikad baik[18].

2.4 Penetration testing tools

Alat pengujian penetrasi adalah instrumen yang digunakan untuk mensimulasikan serangan *cyber* pada sistem untuk mengidentifikasi kerentanan yang dapat dieksploitasi oleh aktor jahat [19]. Alat-alat ini membantu para profesional keamanan menilai postur keamanan jaringan, aplikasi, dan perangkat dengan mereplikasi taktik penyerang potensial, sehingga memungkinkan mereka untuk secara proaktif memperkuat pertahanan dan mengurangi risiko [20]. Pada penelitian ini, terdapat empat buah *pentesting tools* yang digunakan pada tahap pemindaian dengan tujuan untuk melakukan pemindaian jaringan dan juga *port* dari kedua *website* yang diujikan. Berikut ini adalah penjelasan untuk keempat *tools* tersebut :

1. Dnschecker.org

   DNS *Checker* adalah alat yang digunakan untuk memeriksa status catatan sistem nama *domain* (DNS) dari sebuah *website*. Dnschecker.org adalah

layanan *online* yang memungkinkan anda memasukkan nama *domain* dan memeriksa catatan DNS saat ini. Alat ini berguna bagi pemilik situs *web*, administrator sistem, dan insinyur jaringan yang perlu memastikan bahwa catatan DNS mereka benar dan terbaru. Salah satu fungsi yang dapat dilakukan dari *tools* ini adalah untuk melakukan DNS *propagation checker* [21].

DNS *propagation checker* adalah alat *online* yang membantu menentukan status propagasi DNS *domain* setelah dilakukannya perubahan pada pengaturan DNS. DNS (*Domain Name System*) adalah sistem yang menerjemahkan nama *domain* menjadi alamat IP, yang memungkinkan situs *web* dapat diakses oleh pengguna di internet. Ketika dibuat perubahan pada pengaturan *domain* DNS, diperlukan waktu agar perubahan ini disebarkan ke semua *server* DNS di internet. Hal ini karena *server* DNS menyimpan informasi tentang nama *domain* dalam *cache* mereka, dan memerlukan waktu agar *cache* ini diperbarui dengan informasi baru [21].

DNS *propagation checker* dapat membantu memeriksa status propagasi *domain* DNS dengan melakukan *query* ke beberapa *server* DNS dari berbagai lokasi di seluruh dunia. Alat ini dapat memberitahu apakah pengaturan *domain* DNS telah diperbarui dan disebarkan dengan benar atau jika masih ada *server* yang mengembalikan informasi lama. Menggunakan DNS *propagation checker* dapat membantu memastikan bahwa sebuah situs *web* dapat diakses oleh pengguna sesegera mungkin setelah melakukan perubahan pada pengaturan DNS [21].

2. Nmap

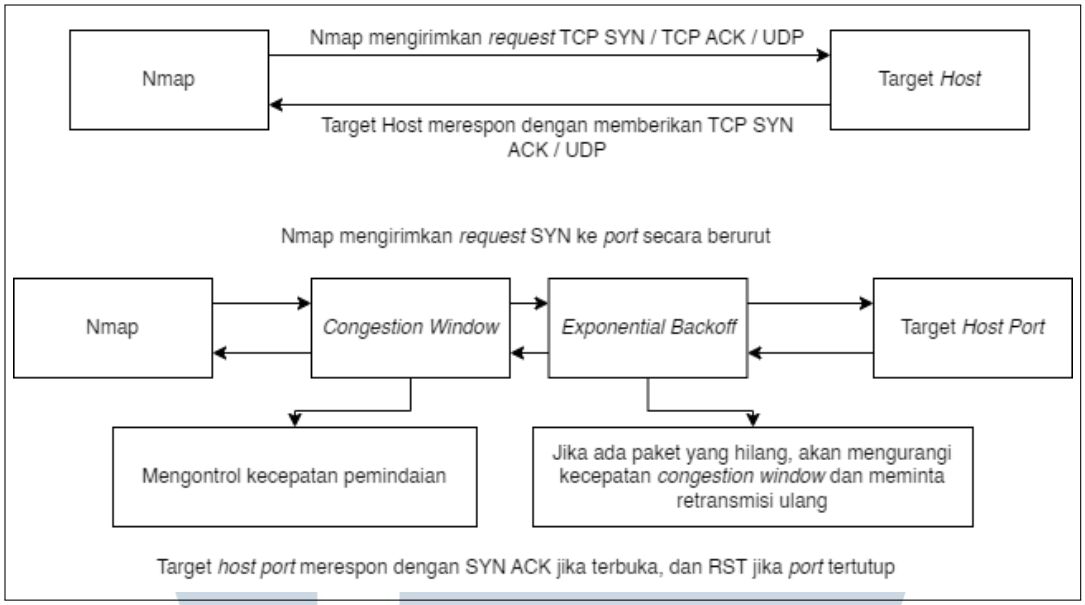
Nmap (singkatan dari *Network Mapper*) merupakan suatu *open source tools* yang biasa digunakan untuk eksplorasi, *information gathering*, dan *vulnerability scanning* sebuah jaringan. *Tools* ini merupakan ciptaan seorang ahli *cyber security* bernama Gordon Lyon pada tahun 1997 dan menjadi salah satu *tools* yang paling sering digunakan dalam *hacking* dan *penetration testing* karena kemampuannya memindai jaringan yang terhubung ke komputer atau *machine* target dan fitur-fitur lainnya yang membantu pengguna memahami detail pada sebuah jaringan [22]. Secara teori, *port* yang terdapat dalam sebuah jaringan berjumlah sebanyak enam puluh lima ribu lima ratus tiga puluh lima buah *port*. Dengan banyaknya *port* yang ada, maka dari itu dibutuhkan kecepatan dan juga ketepatan dalam

mendeteksi *port* apa saja yang terbuka dalam sebuah sistem.

Cara kerja dari *tools* Nmap adalah dengan menandai setiap pencarian dengan nomor urut, *port* sumber atau tujuan, bidang ID, atau aspek lainnya (tergantung pada jenis pencarian) yang memungkinkannya mengenali respons (dan dengan demikian paket yang hilang). Kemudian Nmap menyesuaikan kecepatannya secara tepat untuk tetap secepat yang diizinkan jaringan (dan opsi baris perintah yang diberikan) tanpa melampaui batas dan mengalami ketidakakuratan atau membebani jaringan bersama secara tidak adil [23].

Nmap menggunakan tiga algoritma yang dimodelkan untuk TCP dengan tujuan mengontrol seberapa agresif pemindaian tersebut: *congestion window*, *exponential backoff*, dan *slow start*. *Congestion window* mengontrol berapa banyak pencarian yang Nmap boleh kirimkan secara bersamaan. Jika *window* sudah penuh, Nmap tidak akan mengirimkan lebih banyak lagi sampai sebuah tanggapan diterima atau sebuah pencarian habis waktu. *Exponential backoff* membuat Nmap melambat secara dramatis ketika mendeteksi ada paket yang hilang. *Congestion window* biasanya dikurangi menjadi satu setiap kali terdeteksi adanya paket yang hilang. *Slow start* adalah algoritma yang cukup cepat untuk secara bertahap meningkatkan kecepatan pemindaian untuk menentukan batas kinerja jaringan [23].

Ketika Nmap memindai sekelompok target, ia menyimpan dalam memori sebuah *congestion window* dan *threshold* untuk setiap target, serta sebuah *window* dan *threshold* untuk kelompok secara keseluruhan. *Congestion window* adalah jumlah pencarian yang dapat dikirimkan secara bersamaan. *threshold congestion* menentukan batas antara *slow start* dan mode *congestion avoidance*. Selama *slow start*, *congestion window* tumbuh dengan cepat sebagai respons terhadap tanggapan. Setelah *congestion window* melebihi *threshold congestion*, mode *congestion avoidance* dimulai, di mana *congestion window* bertambah lebih lambat. Setelah sebuah *drop*, baik *congestion window* maupun *threshold* dikurangi menjadi sebagian dari nilai sebelumnya [23]. Gambar 2.1 dibawah ini akan memvisualisasikan cara Nmap bekerja dalam memindai targetnya.



Gambar 2.1. Cara kerja Nmap

3. Zmap

ZMap adalah pemindai jaringan paket tunggal cepat yang dirancang untuk survei jaringan luas Internet. Pada komputer desktop biasa dengan koneksi gigabit Ethernet, ZMap mampu memindai seluruh ruang alamat IPv4 publik pada satu *port* dalam waktu kurang dari empat puluh lima menit. Dengan koneksi 10gigE dan netmap, ZMap dapat memindai ruang alamat IPv4 dalam waktu kurang dari lima menit. Zmap pada dasarnya menggunakan algoritma sebagai berikut [24] :

(a) *Optimized probing*

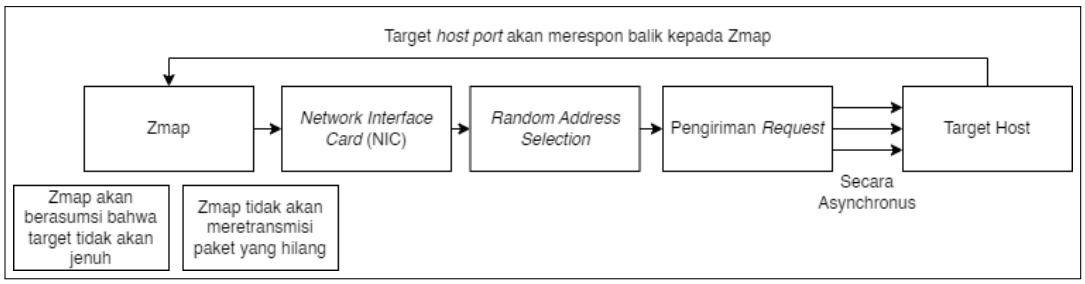
Sementara Nmap menyesuaikan laju transmisinya untuk menghindari jaringan sumber atau target yang jenuh, Zmap mengasumsikan bahwa jaringan sumber telah dipersiapkan dengan baik (tidak dapat jenuh oleh *host* sumber), dan bahwa target-target tersebut diurutkan secara acak dan tersebar luas (sehingga tidak mungkin ada jaringan atau jalur yang jauh menjadi jenuh oleh pemindaian). Oleh karena itu, zmap berusaha untuk mengirimkan pencarian secepat mungkin sesuai dengan dukungan NIC sumber, dengan melewatkan stack TCP/IP dan langsung menghasilkan *frame* Ethernet. ZMap dapat mengirimkan *probe* pada kecepatan garis gigabit dari perangkat keras komoditas dan sepenuhnya di ruang pengguna [24].

(b) *No per-connection state*

Sementara Nmap menjaga keadaan untuk setiap koneksi untuk melacak *host* yang telah dipindai dan untuk menangani *timeout* dan *retransmission*, ZMap mengabaikan setiap keadaan per-koneksi. Karena ditujukan untuk menargetkan sampel acak dari ruang alamat, ZMap dapat menghindari menyimpan alamat yang telah dipindai atau perlu dipindai dan malah memilih alamat berdasarkan permutasi acak yang dihasilkan oleh *cyclic multiplicative group*. Alih-alih melacak waktu habis koneksi, ZMap menerima paket respons dengan bidang keadaan yang benar selama pemindaian, memungkinkannya untuk mengekstraksi sebanyak mungkin data dari tanggapan yang diterimanya. Untuk membedakan respons pencarian yang valid dari lalu lintas latar belakang, ZMap membebani nilai yang tidak digunakan dalam setiap paket yang dikirim, dengan cara yang mirip dengan SYN *cookies* [24].

(c) *No retransmission*

Sementara Nmap mendeteksi *timeout* koneksi dan secara adaptif mengirimkan ulang pencarian yang hilang karena kehilangan paket, ZMap (untuk menghindari menyimpan keadaan) selalu mengirimkan jumlah pencarian yang tetap per target dan secara *default* hanya mengirimkan satu. Dalam pengaturan eksperimental, diperkirakan bahwa ZMap mencapai cakupan jaringan 98% hanya dengan satu pencarian per *host*, bahkan pada kecepatan pemindaian maksimumnya. Dipercaya jumlah kehilangan ini tidak akan signifikan untuk aplikasi penelitian yang tipikal [24]. Gambar 2.2 dibawah ini akan memvisualisasikan cara Zmap bekerja memindai targetnya.



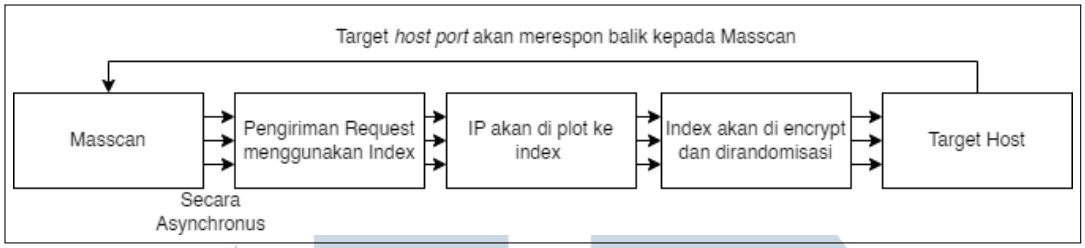
Gambar 2.2. Cara kerja Zmap

4. Masscan

Masscan adalah sebuah alat pemindaian jaringan yang sangat cepat dan ringan yang dirancang untuk melakukan pemindaian besar-besaran jaringan dalam waktu yang sangat singkat. Masscan mendukung pemindaian masal dari banyak mesin. *Scanner* ini cepat dan berpotensi untuk memindai seluruh internet dalam waktu kurang dari lima menit, mengirimkan sepuluh juta paket per detik, dari satu mesin saja. Secara internal, Masscan menggunakan transmisi *asynchronous*, dan fleksibel karena memungkinkan rentang *port* dan alamat yang ditentukan sendiri [25].

Perbedaan utama antara Masscan dan pemindai lain adalah cara pengacakan targetnya. Prinsip dasarnya menggunakan variabel indeks tunggal yang dimulai dari nol dan ditambah satu untuk setiap pemeriksaan. Namun terdapat masalah yaitu diperlukan untuk menerjemahkan indeks tersebut menjadi alamat IP. Misalnya, untuk memindai semua alamat IP ”pribadi”. Proses ini menjadi lambat karena pembagian internet menjadi ratusan rentang yang lebih kecil. Ini mengharuskan konversi variabel indeks menjadi alamat IP untuk setiap pemeriksaan. Masscan mengatasi hal ini dengan menggunakan ”pencarian biner” dalam memori yang kecil. Fungsi yang menerjemahkan dari indeks ke alamat IP disebut *pick* [25].

Masscan juga mendukung rentang *port*, yang berarti kita perlu memilih alamat IP dan *port* dari variabel indeks. Instruksi pembagian / *modulo* membutuhkan waktu yang cukup lama. Namun, untungnya, dua operasi tersebut dapat dijalankan secara bersamaan. Meskipun demikian, pengacakan diperlukan untuk menyebarkan lalu lintas secara merata ke target dan menghindari pemboman jaringan yang tidak siap menangani kecepatan pemindaian Masscan. Masscan mengacakkan target dengan mengenkripsi variabel indeks. Ini menghasilkan pemetaan 1-ke-1 antara variabel indeks asli dan *output*, sehingga urutan alamat IP menjadi acak sepenuhnya. Pengacakan ini memiliki biaya karena rentangnya tidak selalu berukuran sama. Masscan harus menggunakan operasi *modulus* (%) yang lebih mahal dibandingkan teknik biner sederhana [25]. Gambar 2.3 dibawah ini akan memvisualisasikan cara Zmap bekerja memindai targetnya.



Gambar 2.3. Cara kerja Masscan

Pada penelitian ini digunakan 3 *tools* untuk menjalankan *port scanning*. Hal ini dikarenakan *tools* tersebut memiliki perbedaan antar satu sama lain. Tabel 2.2 dibawah ini melampirkan perbedaan dari ketiga *tools* tersebut.

Tabel 2.2: Perbedaan *tools port scanning*

Fitur	**Nmap**	**Zmap**	**Masscan**
Kecepatan	Lebih lambat	Lebih cepat	Sangat cepat
Penyesuaian laju transmisi	Ya	Tidak	Tidak
Pilihan alamat target	Berurutan	Acak	Acak
Pelacakan *timeout*	Ya	Tidak	Tidak
Target *port*	Tidak harus spesifik	Harus spesifik	Tidak harus spesifik
Penanganan kehilangan paket	Kirim ulang pencarian yang hilang	Tidak mengirim ulang	Tidak mengirim ulang

Continued on next page

Tabel 2.2: Perbedaan *tools port scanning* (Continued)

Kelebihan	Baik dalam mendeteksi *host* yang aktif, mengidentifikasi *port* terbuka dan tertutup, dan menangani berbagai jenis respons	Menggunakan teknik pemindaian *asynchronous* untuk mencapai kecepatan tinggi.	Menggunakan *randomization* untuk menghindari kejenuhan jaringan, meningkatkan cakupan pemindaian, dan mengidentifikasi layanan yang rentan.
Aplikasi ideal	Pemindaian umum yang membutuhkan detail dan keandalan	Pemindaian jaringan skala besar yang membutuhkan kecepatan tinggi	Pemindaian jaringan skala besar yang membutuhkan kecepatan tinggi

2.5 Zero Entry Hacking (ZEH)

Dalam *Penetration Testing*, terdapat banyak metode yang dapat digunakan, termasuk OSSTM (*Open Source Security Testing Methodology Manual*), OWASP (*Open Web Application Security Project*), NIST (*National Institute of Standards and Technology*), PTES (*Penetration Testing Methodologies and Standards*), ISSAF (*Information System Security Assessment Framework*), dan ZEH (*Zero Entry Hacking*). Metode ini memiliki kelebihan dan kekurangannya masing - masing. Tabel 2.3 dibawah ini menunjukkan kelebihan dan kekurangan dari masing - masing metode.

Tabel 2.3: Perbandingan Metode *Penetration Testing*

No	Metode	Kelebihan	Kekurangan
1	OSSTM (*Open Source Security Testing Methodology Manual*)	- *Open Source* - Mendukung tim pengembangan jaringan - Memiliki 6 tahapan	- Kompleks dan membutuhkan pemahaman yang lebih mendalam tentang keamanan sistem - Tergantung pengalaman penguji / *pentester*
2	OWASP (*Open Web Application Security Project*)	- Memiliki Standar keamanan *website* sendiri (OWASP ASVS) OWASP Top 10 (Checklist Standar Keamanan *Website*)	- Hanya cocok untuk pengujian berbasis *website*
3	NIST (*National Institute of Standards and Technology*)	- Memiliki panduan yang disediakan di NIST SP 800-53 - Memiliki 3 kelas kontrol keamanan - Terdiri dari 6 fase pengukuran keamanan	- Terdiri dari 19 bidang pengukuran - Ruang lingkup dan sumber daya yang besar
4	PTES (*Penetration Testing Methodologies and Standards*)	- Memiliki langkah uji penetrasi, komunikasi awal, pengumpulan informasi, dan pemodelan ancaman - Memiliki tujuh fase untuk menjamin uji penetrasi	- Melibatkan keahlian dan penilaian dari seorang *pentester* - Melalui banyak langkah

Continued on next page

Tabel 2.3: Perbandingan Metode *Penetration Testing* (Continued)

5	ISSAF (*Information System Security Assessment Framework)*	- Terstruktur dan khusus untuk pengujian penetration testing - Kerangka kerja yang komprehensif - Terperinci dengan sembilan tahapan	- Sangat kompleks - Keterbatasan pembaruan
6	WAPT (*Web Application Penetration Testing*)	- Memiliki 5 tahapan pengujian - Lebih sederhana dibandingkan beberapa metode lainnya	- Memerlukan sumber daya yang cukup banyak - Memakan waktu yang lebih lama - Dapat mengganggu operasional *website*
7	BSIMM (*Building Security in Maturity Mode*l)	- Kerangka kerja yang terstruktur - Memiliki 112 kegiatan BSIMM dalam 4 *domain*	- Sangat kompleks untuk pemula - Kurang fleksibel
8	ZEH (*Zero Entry Hacking*)	- Memiliki 4 Tahapan sederhana - Lebih mudah - Tetap mampu memberikan hasil pengujian secara baik	- Tidak termasuk ke dalam metode umum yang diketahui oleh banyak orang

Dari tabel diatas, penulis memilih untuk menggunakan metode ZEH karena memiliki empat tahapan sederhana yang sangat cocok untuk pemula namun tetap dapat memberikan hasil yang baik dalam pengujian kerentanan yang dapat diketahui berdasarkan beberapa penelitian sebelumnya. Selain itu, kompleksitas, waktu serta sumber daya yang dibutuhkan masih dapat dipertimbangkan. ZEH sudah pernah digunakan pada beberapa penelitian sebelumnya. Tabel 2.4 dibawah ini merupakan penelitian sebelumnya yang menggunakan metode yang sama.

Tabel 2.4: Perbandingan penelitian terdahulu

No	Peneliti	Judul	Tahun	Perbedaan
1	Rama Sahtyawan	Penerapan *Zero Entry Hacking* Didalam *Security Misconfiguration* Pada VAPT (*Vulnerability Assesment And Penetration Testing*)	2019	Hanya menguji kerentanan SMB (*server Message block*) dan RDP (*remote desktop*) di sebuah *server*, tidak melakukan pengujian pada kerentanan yang ditemukan menggunakan *vulnerability scanner*
2	Muhammad Yaqi	*Vulnerability Assessment* dan *Penetration Testing* (Vapt) Menggunakan Metode *Zero Entry Hacking* (Zeh) Terhadap Studi Kasus: Dinas Penanaman Modal dan PTSP Kota Tangerang Selatan	2023	Peneliti menggunakan metode *gray box* testing, yang dimana peneliti mengetahui sebagian informasi mengenai *server* melalui wawancara dengan pihak terkait
3	Leonardo Pandapotan	*Penetration Testing* Terhadap *Damn Vulnerable Web Application* (DVWA) Menggunakan Metode *Zero Entry Hacking* (ZEH) Dan *Open Web Application Security Project* (OWASP)	2021	*Website* yang diuji merupakan *website* dvwa, yang dimana *website* tersebut biasanya digunakan orang sebagai alat belajar dalam melakukan *penetration testing*, sehingga objek dari penelitiannya bukan kepada pihak nyata.

Continued on next page

Tabel 2.4: Perbandingan penelitian terdahulu (Continued)

4	Oman Gunawan	*Penetration Testing* Terhadap *Website* Universitas Pasundan Dengan Metode *Zero Entry Hacking* (STUDI KASUS: *http://www.unpas.ac.id*)	2022	Hanya menguji kerentanan *sensitive data exposure* dan *security misconfiguration* pada *website* tersebut, tidak melakukan pengujian pada kerentanan yang ditemukan menggunakan *vulnerability scanner*
5	Achmad Nur Sholeh	Analisis dan Pengujian Kerentanan Sistem Informasi Perpustakaan	2019	Kerentanan yang ditemukan tidak diukur menggunakan alat pengukur kerentanan, dan juga tidak adanya pemberian solusi.

Dapat dilihat pada tabel diatas bahwa sudah terdapat beberapa penelitian yang menggunakan metode ZEH, maka dari itu, pada penelitian ini akan menggabungkan dua metode *penetration testing* yaitu ZEH dan OWASP. Selain itu, pada penelitian ini pula, akan dilakukan *vulnerability scanning* guna mencari kerentanan yang ada pada *website* milik UMN. Setelah kerentanan ditemukan, kerentanan akan diukur menggunakan CVSS.

Menurut Engebretson (2013), ZEH (*Zero Entry Hacking*) merupakan salah satu metodologi yang digunakan untuk melakukan *Penetration Testing* [10]. ZEH merupakan salah satu metodologi yang cocok digunakan untuk pemula dalam melakukan *Penetration Testing* karena menggunakan empat tahapan sederhana saja.

Terdapat empat tahapan dalam ZEH yang digunakan dalam pengujian sistem. Adapun empat tahapannya menurut yaitu . Pengintaian Sistem (*Reconnaisance*), Pemindaian (*Scanning*), Eksploitasi Celah Keamanan (*Exploitation*) dan Pasca Eksploitasi (*Post Exploitation*).

Pengintaian sistem (*Reconnaisance*) atau biasa disebut pengumpulan informasi merupakan tahapan di mana seorang penguji mengumpulkan informasi sebanyak-banyaknya mengenai objek yang diteliti . Adapun *tools* yang digunakan

sebagai berikut :

1. WHO IS

   Layanan Whois memungkinkan kita untuk mengakses informasi spesifik tentang target termasuk alamat IP atau nama *host* dari target *Server*, DNS dan informasi kontak yang biasanya berisi alamat dan nomor telepon. Who is tersedia dalam OS Linux.

2. The Harvester

   The Harvester adalah skrip Python sederhana namun sangat efektif yang ditulis oleh Christian Martorella di Edge Security. Alat ini memungkinkan peneliti untuk membuat katalog alamat email dan *subdomain* dengan cepat dan akurat yang terkait langsung dengan target. Harvester dapat digunakan untuk mencari email, *host*, dan *subdomain* di *server* Google, Bing, Yahoo, dan lainnya. *Tool* ini juga dapat mencari LinkedIn untuk nama pengguna [10].

3. What Web

   WhatWeb digunakan untuk mengidentifikasi situs *web*. What Web berguna untuk mengenali teknologi *web* termasuk sistem manajemen konten (CMS), platform *blogging*, paket statistik / analitik, perpustakaan JavaScript, *server web*, dan perangkat yang disematkan [26].

Setelah tahap pengintaian selesai dilakukan, dilanjutkan dengan tahap pemindaian. Dalam tahapan ini, dibagi lagi menjadi lima tahapan :

1. Menentukan apakah objek tersebut dapat di *ping*.
2. *Network mapping* subnet menggunakan NMAP.
3. *Port scanning* subnet menggunakan NMAP, Zmap, dan Masscan.
4. *Vulnerability mapping* port yang ditemukan terbuka.
5. Pemindaian sistem menggunakan *vulnerability scanner*.

Eksploitasi merupakan proses untuk mendapatkan kontrol atas sistem. Jika ada celah, maka penguji dapat mencoba untuk mengeksploitasi atau mencoba celah tersebut. Eksploitasi berbasis *web* di sini merupakan upaya-upaya yang dilakukan untuk mengeksploitasi aplikasi berbasis *web* berdasarkan celah keamanan yang

diekspos oleh *vulnerability scanner*. Selain menggunakan *web*, eksploitasi juga akan dibantu menggunakan dua *tools* lainnya yaitu Burp Suite dan WireShark

Dalam tahapan pasca eksploitasi (*post exploitation*), Penulis akan merangkum hasil dari uji penetrasi dan juga menghitung skala prioritas kerentanan untuk mengetahui kerentanan mana yang perlu untuk diperbaiki terlebih dahulu. Selain itu, penulis juga akan membuat laporan berupa catatan hasil dari *penetration testing* mencakup kerentanan yang ada dan solusi atas kerentanan tersebut.

2.6 Common Vulnerability Scoring System (CVSS)

Common Vulnerability Scoring System (CVSS) adalah metode yang digunakan untuk menyediakan ukuran kualitatif dari tingkat *severity*. CVSS bukanlah untuk mengukur risiko. CVSS terdiri dari tiga kelompok matriks : *Base*, *Temporal*, dan *Environmental*. Matriks *Base* menghasilkan skor mulai dari 0 hingga 10, yang kemudian dapat dimodifikasi dengan melakukan penilai pada matriks *Temporal* dan *Enviromental*. CVSS sangat cocok digunakan sebagai sistem pengukuran standar untuk industri, organisasi, dan pemerintah yang membutuhkan skor *vulnerability severity* yang akurat dan konsisten. Dua penggunaan umum CVSS adalah untuk menghitung *vulnerability severity* yang ditemukan pada sebuah sistem dan sebagai faktor dalam memprioritaskan aktivitas perbaikan kerentanan [27].

Menurut Wardaya (2019), adapun benefit yang akan didapat bila menerapkan CVSS sebagai pengukur tingkat kerentanan yaitu [15] :

1. Skor kerentanan yang terstandarisasi

   Ketika suatu organisasi menormalkan skor kerentanan di semua *platform* perangkat lunak dan perangkat kerasnya, ia dapat memanfaatkan kebijakan manajemen kerentanan tunggal. Kebijakan ini mungkin mirip dengan *Service Level Agreement* (SLA) yang menyatakan seberapa cepat kerentanan tertentu harus divalidasi dan diatasi.

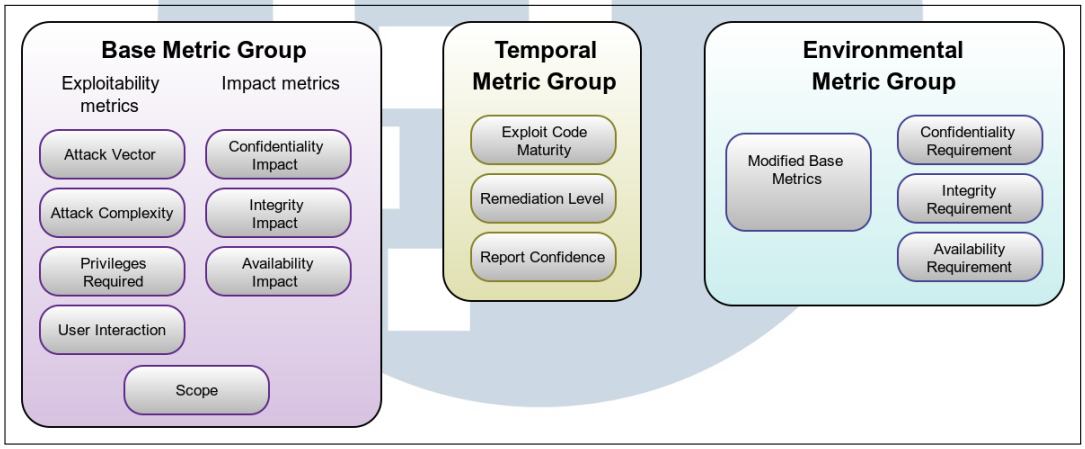
2. Open framework

   Pengguna dapat bingung ketika kerentanan diberikan skor arbitrer. “Properti mana yang memberikan skor itu? Apa bedanya dengan yang dirilis kemarin? ”Dengan CVSS, siapa pun dapat melihat karakteristik individu yang digunakan untuk memperoleh skor.

3. Risiko yang diprioritaskan

   Ketika *enviromental score* dihitung, kerentanan sekarang menjadi kontekstual. Artinya, skor kerentanan sekarang mewakili risiko aktual bagi suatu organisasi. Pengguna tahu betapa pentingnya kerentanan yang diberikan dalam kaitannya dengan kerentanan lainnya.

CVSS terdiri dari tiga *group metrics*: *Base*, *Temporal*, dan *Environmental* masing-masing terdiri satu set *metrics*. CVSS *group metrics* dapat dilihat pada Gambar 2.4 sebagai berikut [28]:



Gambar 2.4. CVSS *Group Metrics*