ANALISA CELAH KELAH KEAMANAN TERHADAP WEB SERVER MENGGUNAKAN METODE ATTACK SURFACE DAN KEPADATAN KERENTANAN

Arnold Nasir
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar
Alamat e-mail:arnold_nasir@outlook.com

ABSTRACT

The use of software security metrics is one of the methods in measuring quantitatively the reliability of a software. These measures can be used in assessing resource allocation, program planning, risk assessment, and product or service selection. One of the commonly used measurements is the attack surface and vulnerability density. Both of these measurement methods have been widely used by several large technology companies, but determining the validity of software security measures remains a challenge in itself. A safety measure cannot measure all aspects of security and therefore the use of various measurement methods is required in some cases. This study aims to see the relationship between the measurement based on attack surface and vulnerability density by applying to several web servers that are placed in the demilitarized zone (DMZ) area, which is an area flanked by two or more firewalls.

.

Keywords: *Attack Surface,* Celah Keamanan Komputer, *Demiliterized Zone (DMZ), Firewall,* Kepadatan Kerentanan

1. PENDAHULUAN

Teknologi, khususnya perangkat lunak (software); baik berupa aplikasi atau servis, telah menjadi bagian yang penting dalam setiap bagian dari kehidupan masyarakat global saat ini. Salah satu contoh nyata adalah penggunaan Internet yang telah mampu menggeser penggunaan telefon sebagai media untuk memberikan atau menukar informasi antara satu pihak dengan pihak lain. Hal ini dapat dilihat dari berbagai macam aplikasi dan layanan yang berada pada platform Internet, misalnya menelusuri website untuk mendapatkan informasi, hingga melakukan proses jual beli melalui media e-commerce. Dari data yang diperoleh[1] jumlah pengguna Internet di Indonesia pada tahun 2014 mencapai 83,7 juta orang dan diprediksi pengguna Internet pada tahun 2016 mencapai 102,8 juta orang.

Perkembangan aplikasi dan layanan berbasis Internet pun saat ini makin meningkat, hanya saja tidak diimbangi dengan kualitas aplikasi dan layanan yang masih membutuhkan peningkatan kualitas. Banyak aplikasi dan layanan saat ini memiliki kecacatan yang tingkatannya beragam; mulai dari rendah hingga membutuhkan penanganan serius yang tentunya dapat merugikan pengguna aplikasi dan layanan tersebut. Tak jarang, kecacatan tersebut terkait dengan masalah keamanan yang biasa disebut kerentanan (vulnerability). Apabila kerentanan tersebut dieksploitasi oleh pengguna yang tidak bertanggung jawab, maka tentu memiliki dampak yang sangat besar, seperti yang terjadi pada kasus Aadhar pada tahun 2018 di India dimana sekitar 1,1 Milyar database penduduk India yang menyimpan identitas masyarakat serta informasi biometric dicuri oleh para peretas[2]. Untuk mengurangi maupun mencegah kejadian tersebut terjadi, maka keamanan perangkat lunak harus menjadi prioritas yang utama bagi vendor maupun pengguna dengan mengembangkan metode-metode yang dapat memberikan informasi terkait kondisi keamanan untuk selanjutnya dilakukan pembaharuan sehingga perangkat lunak tersebut menjadi lebih aman.

Pengukuran keamanan adalah sebuah pengukuran kuantitatif yang menunjukkan tingkatan keamanan pada sebuah atribut dari sistem[3]. Beberapa metode pengukuran keamanan perangkat lunak telah diusulkan oleh beberapa peneliti; baik dari kalangan akademisi, industri, maupun pemerintahan. Metode-metode pengukuran tersebut diantaranya kerentanan relatif, Attack

Surface, Vulnerability Density, CVSS score, dan lain-lain [4]. Tiap-tiap metode pengukuran tersebut memiliki basis penilaian terhadap pandangan, target, serta asumsi yang berbeda dalam mengukur atribut keamanan perangkat lunak.

Menurut Verendel, kurangnya validasi dan perbandingan diantara setiap pengukuran merupakan tantangan yang dalam pemutusan penggunaannya dapat menjadi resiko apabila tidak dicermati dengan baik penggunaannya. Disamping itu, tidak ada satu metode pengukuran keamanan yang digunakan tunggal, melainkan diperlukan sebuah framework yang dapat mengkombinasikan pengukuran yang berbeda untuk mengukur keamanan sebuah sistem [5].

Tujuan dari penelitian ini adalah untuk melakukan proses analisa kerentanan keamanan dari beberapa web server dengan platform Operating System yang berbeda yang terhubung pada jaringan Wide Area Network (WAN) menggunakan dua metode pengukuran keamanan, yaitu Attack Surface, dan Vulnerability Density. Disamping itu, diharapkan dari penelitian ini dapat terlihat keterkaitan antara kedua metode pengukuran tersebut.

2. TINJAUAN PUSTAKA

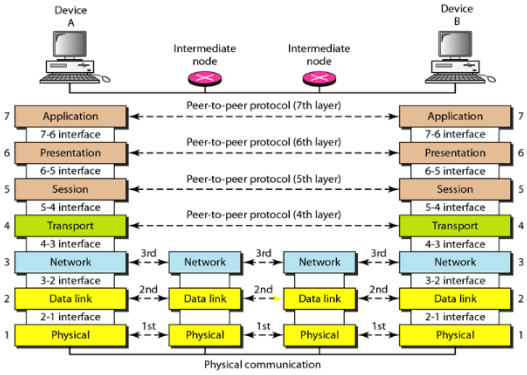
2.1 Jaringan Komputer

Menurut Tanenbaum, jaringan komputer merupakan penggabungan teknologi komputer dan komunikasi dimana sekumpulan komputer dengan jumlah yang banyak dan ditempatkan pada area yang berbeda, akan tetapi saling berhubungan dalam melaksanakan tugasnya.

Berdasarkan dari pengertian tadi, maka secara umum dapat dikatakan bahwa jaringan komputer memiliki beberapa tujuan, yaitu membagi sumber daya (seperti printer, CPU, memori, dan kapasistas penyimpanan), komunikasi (e-mail), dan akses informasi melalui Internet.

2.2 Open System Interconnection (OSI)

Open System Interconnect Layer atau yang sering dikenal dengan istilah OSI Layer merupakan sebuah referensi tentang bagaimana sebuah aplikasi dapat berjalan pada sebuah jaringan.



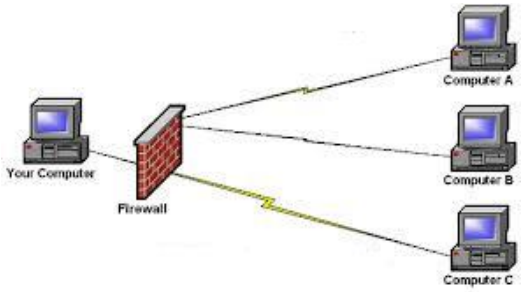
Gambar 1. OSI Layer

Secara singkat tujuh layer OSI memiliki fungsi sebagai berikut:

a. *Physical layer*: merupakan layer pertama dari OSI layer dan berperan dalam transmisi dan penerimaan raw bit stream yang belum terstruktur yang berasal dari medium fisik seperti kabel Ethernet.
b. *Data link*: berperan dalam menyediakan transfer error-free dari frame data dari 1 node ke node lainnya melalui physical layer.
c. *Network layer*: mengontrol operasi dari subnet, memutuskan jalur fisik data yang harus diambil berdasarkan kondisi jaringan, prioritas pelayanan, dan faktor-faktor lainnya.
d. *Transport layer*: memastikan bahwa pesan-pesan yang disampaikan bebas dari kesalahan, telah berurutan, dan dengan tidak ada yang hilang atau terduplikasi.
e. *Session layer*: memungkinkan pembentukan sesi antara proses yang berjalan pada node yang berbeda.
f. *Presentation layer*: berperan dalam memformat data sebelum ditampilkan pada Application layer.
g. *Application layer*: berperan sebagai tampilan untuk pengguna dan proses aplikasi dalam mengakses layanan jaringan.

2.3 Firewall

Firewall merupakan sebuah sistem yang berfungsi untuk membatasi akses terhadap sebagian atau keseluruhan jaringan dari sebuah perusahaan dari“Internet buruk”yang berasal dari luar dan dalam jaringan perusahaan. Firewall sendiri secara umum dapat berupa software seperti Windows Firewall maupun dapat berupa hardware seperti Comodo Firewall.



Gambar 2. Firewall

Dalam penerapannya, firewall memiliki sejumlah aturan yang sering disebut dengan policy yang berisikan aturan-aturan terhadap arus informasi yang masuk dan keluar dari sebuah jaringan perusahaan. Aturan-aturan tersebut nantinya akan menentukan apakah sebuah paket data dapat diteruskan ke tujuannya atau ditolak. Selain itu, firewall juga tergolong sangat fleksibel. Hal ini dikarenakan karena bukan hanya dapat diterapkan sebagai filter pada layer Transport dalam OSI Layer, tetapi dapat pula bekerja pada layer yang lebih tinggi.

Dalam pengoperasiannya, firewall memiliki empat teknik untuk mengontrol akses dan memaksakan security policy pada jaringan:

a. *Service control*: menentukan jenis layanan Internet yang dapat diakses, baik informasi yang menuju keluar maupun menuju kedalam perusahaan.
b. *Direction control*: menentukan arah dimana permintaan layanan tertentu dapat dimulai dan diizinkan melewati firewall.
c. *User control*: mengontrol akses sebuah layanan menurut pengguna yang berusaha untuk mengakses layanan tersebut.
d. *Behavior control*: mengontrol bagaimana sebuah layanan yang spesifik digunakan. Contohnya, firewall dapat melakukan proses filter terhadap e-mail yang masuk guna mencegat spam.

2.4 Attack Surface

Attack surface merupakan jumlah total kerentanan yang dapat dieksploitasi oleh pengguna yang tidak bertanggung jawab dalam melakukan serangan terhadap keamanan sebuah aplikasi maupun sistem. Attack surface dapat berupa serangan bersifat fisik (penerobosan kantor, merusak fisik server) maupun bersifat digital. Pengertian Attack surface sering disalahartikan dengan pengertian attack vector, dimana pada attack surface yang ditekankan adalah apa yang diserang, sementara attack vector lebih menekankan kepada sarana yang digunakan oleh penyusup untuk mengakses[6].

Dalam melakukan pengukuran menggunakan metode Attack surface, sumber daya yang terkait dengan aplikasi atau sistem seperti API, socket menjadi hal yang diperhitungkan dalam melakukan proses analisa kerentanan keamanan. Semakin banyak sumber daya tersebut dimiliki oleh aplikasi atau sistem, maka semakin besar pula permukaan serangan yang mengakibatkan aplikasi atau sistem tersebut menjadi kurang aman.

Potensi kerusakan dan rasio upaya merupakan rerata yang sifatnya informal yang digunakan untuk mengestimasikan potensi kerusakan terhadap atribut sumber daya yang dimiliki oleh aplikasi atau sistem. Potensi kerusakan bergantung pada hak istimewa metode, tipe channel, dan tipe data item, sementara rasio upaya bergantung pada hak sumber daya yang perlu dimiliki penyerang untuk menggunakan sumber daya dalam melakukan serangan. Secara singkat dapat dirumuskan sebagai berikut:

$$daya\ serang\ (metode) = \frac{hak\ istimewa}{hak\ akses} \quad (1)$$

$$daya\ serang\ (channel) = \frac{tipe}{hak\ akses} \quad (2)$$

$$daya\ serang\ (data\ item) = \frac{tipe}{hak\ akses} \quad (3)$$

Perlu diperhatikan dengan seksama bahwa semakin tinggi hak istimewa, tipe channel, atau data item, semakin tinggi potensi kerusakan, sementara semakin tinggi hak akses maka semakin tinggi rasio upaya.

Kerentanan perangkat lunak didefinisikan sebagai sebuah kecacatan pada perangkat lunak yang mengakibatkan resiko keamanan. Setiap kerentanan pada sebuah aplikasi maupun sistem harus ditemukan dan dipublikasikan pada publik, kemudian dikelola. Basis data terkait kerentanan dikelola oleh *National Vulnerability Database* (NVD), *Open Source Vulnerability Database* (OSVDB), serta vendor dari perangkat lunak. Kerentanan diberikan kode

identifikasi unik menggunakan MITRE *Common Vulnerability and Exposure* (CVE).

3. METODOLOGI PENELITIAN

Penelitian ini dikembangkan dengan menggunakan metode eksperimental. Secara umum kegiatan penelitian ini dibagi menjadi 2 tahapan utama, yakni tahap perancangan dan tahap evaluasi. Pada tahapan perancangan, pengusul akan merancang simulasi kondisi jaringan Wide Area Network (WAN) untuk kemudian ditempatkan beberapa web server didalamnya. Lamanya kegiatan dijadwalkan selama kurang lebih dua (2) minggu dan dapat diperpanjang apabila jumlah layanan aplikasi yang berjalan bertambah. Setelah diperoleh data yang terkait dengan penggunaan aplikasi, maka selanjutnya dilakukan proses analisa terhadap web server dengan framework yang meintegrasikan metode pengukuran berbasis Attack Surface dan Vulnerability Density. Proses analisa berlangsung selama 2-3 bulan, tetapi dapat berlangsung lebih lama jika didapati aplikasi baru yang berjalan pada jaringan.

3.1 Lokasi Penelitian

Kegiatan penelitian ini dilakukan pada Laboratorium Jaringan Universitas Atma Jaya Makassar, dengan membuat sebuah kondisi simulasi jaringan berbasis WAN yang memiliki beberapa web server didalamnya.

3.2 Analisis Data

Proses pengumpulan data dilakukan dengan metode observasi dan studi literatur. Observasi dilakukan terhadap proses analisa yang dijalankan dengan menggunakan beberapa perangkat lunak yang dibuat khusus untuk melakukan proses audit keamanan sistem dengan melakukan simulasi serangan terhadap web server yang ditempatkan pada jaringan WAN. Sementara itu, studi literatur dilakukan terkait metode-metode penilaian terhadap tingkat parahnya sebuah kerentanan pada sebuah aplikasi atau sistem..

4. HASIL DAN PEMBAHASAN

4.1 Analisa Kebutuhan

4.1.1 Observasi

Kegiatan observasi sendiri dijadwalkan untuk dilakukan selama dua (2) minggu, kemudian ditambah seminggu untuk mendapatkan data traffic jaringan selama satu (1) bulan.

Adapun hasil observasi yang diperoleh adalah sebagai berikut:

a. Jumlah pengguna jaringan komputer di lingkungan UAJM adalah sebesar 1.600 orang yang terdiri atas dosen, karyawan, dan mahasiswa. Dari ketiga kategori pengguna tersebut, dosen dan karyawan merupakan pengguna yang diprioritaskan sehingga diberikan bandwidth jaringan yang lebih besar bila dibandingkan dengan mahasiswa. Alasan mengapa dosen dan karyawan diberikan bandwidth yang lebih besar ialah untuk mendukung kegiatan pekerjaan mereka, seperti pembayaran pajak via online, mengunggah materi kuliah, mengunggah hasil penelitian pada jurnal online, dsb.
b. Adapun aktivitas penggunaan yang diamati lebih difokuskan pada penggunaan layanan Internet dimana layanan tersebut sering dikeluhkan oleh para pengguna di lingkungan universitas.

4.1.2 Sumber Data

Stack trace atau disebut juga jejak yang digunakan sebagai data untuk penelitian ini berasal tiga sumber pelengkap: *fuzzing, user mod*, dan *kernel mod*. Penulis melacak sumbernya dari setiap kerusakan saat Penulis mengurai setiap jejak. Penulis sekarang menjelaskan setiap sumber kerusakan.

A. Fuzzing

Tim pengujian keamanan setiap menghasilkan crash yang tidak jelas. Fuzzing adalah strategi pengujian yang berputar di sekitar pengiriman acak atau sengaja merusak / data berbahaya ke titik input aktif sebuah sistem. Tujuan fuzzing adalah untuk mensimulasikan serangan dan untuk mendapatkan sistem untuk berperilaku secara tak terduga. Biasanya, apa saja respon dari sistem yang berbeda dari kesalahan standar pesan ditandai dan diselidiki. Kemungkinan hasil dari fuzz pengujian termasuk crash, kebocoran memori, dan bug keamanan seperti itu sebagai kehilangan data atau akses yang tidak tepat. Kecelakaan fuzzing sangat berguna sumber data sebagai input apa pun yang dihasilkan dan dimasukkan ke dalam

sistem sedang diuji juga bisa berasal dari pengguna dan dengan demikian dari peretas potensial. Fuzzing bertujuan menemukan keamanan kerentanan, dan setiap hasil fuzzing tidak teratur relevan untuk menentukan area kode apa yang terlibat dalam penanganan input salah bentuk.

B. User Mod

Pengguna Distro Linux menghasilkan crash mode pengguna. Mode pengguna crash adalah kerusakan pelanggan yang bukan karena perangkat keras kegagalan. Gangguan pada aplikasi yang berjalan tanpa administrator hak memicu pengumpulan informasi macet yang mungkin dikirim ke Seluruh OS. Sistem yang bertanggung jawab untuk mengumpulkan data, Pelaporan Kesalahan Seluruh OS, dijalankan di bawah yang sama hak pengguna sebagai pengguna yang menjalankan aplikasi yang macet. Namun, menjalankan dalam mode pengguna membatasi jejak tumpukan kemampuan generator untuk mengakses dan menyelesaikan sumber daya mana terlibat dalam kecelakaan itu. Dengan demikian, mode pengguna hanya crash mengidentifikasi area kode yang dapat diakses oleh pengguna dalam mode itu.

C. Kernel Mod

Kernel crash terjadi pada mesin, tetapi tidak seperti itu aplikasi crash, mereka terjadi di dalam kernel Distro Linux. Secara umum, crash kernel mengindikasikan kegagalan yang lebih parah dan biasanya menyertakan resolusi penuh artefak kode, selama mereka adalah bagian dari sistem Distro Linux. Kernel berjalan di bawah hak administratif yang memungkinkan generator penelusuran jejak kumpulkan lebih detail tentang detail granular sistem.

D. Kerentanan yang Diketahui

Untuk mengukur efektivitas pendekatan Penulis, Penulis membutuhkan seperangkat kerentanan untuk membandingkan pendekatan Penulis terhadap. Untuk tujuan ini, Penulis menggunakan set kerentanan yang terlihat di Distro Linux 8 baik sebelum dan sesudah rilis produk. Penulis kemudian memeriksa lihat apakah kerentanan ini muncul di permukaan serangan.

4.2 Hasil Penelitian

Dari hasil pengujian didapatkan data sebagai berikut:

Tabel 1. Hasil pengujian berdasarkan Attack Surface dan Celah Kerentanan pada tiap OS

Jenis OS	OSVDB & CVE	Attack Surface & Celah Kerentanan
Fedora	3	13.5
FreeBSD	8	48.4
ClearOS	10	62.1
Mageia	5	27.2
CentOS	8	47.5
Solaris	8	44.2
Ubuntu	6	31.6

5. KESIMPULAN

Kesimpulan yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Web server berbasis Apache versi 2.4.x memiliki nilai kerentanan yang cukup besar.
2. Tidak semua OS memiliki kecocokan untuk menggunakan Apache versi 2.4.39 (versi terbaru saat pengujian). Adapun hal ini dikarenakan tidak semua OS memiliki arsitektur yang sama.
3. Hampir semua sistem yang diujikan memiliki tingkat kerentanan mulai dari sedang hingga tinggi. Hal ini dapat dilihat dari potensi bahaya seperti XSS (*Cross Site Scripting*), OSVDB-3092 (terkait administrator page).

6. DAFTAR PUSTAKA

[1] P. KOMINFO, "Pengguna Internet Indonesia Nomor Enam Dunia", Website Resmi Kementerian Komunikasi dan Informatika RI, 2014. [Online]. Available: https://kominfo.go.id/content/detail/4286/pengguna-internet-indonesia-nomor-enam-dunia/0/sorotan_media. [Accessed: 01- Feb - 2019].

[2] L. Paige, "The 21 Scariest Data Breaches of 2018", SearchNetworking, 2016. [Online]. Available https://www.businessinsider.sg/data-hacks-breaches-biggest-of-2018-2018-12/?r=US&IR=T [Accessed: 02 – Feb - 2019].

[3] W. Jansen, “Direction in Security Metrics Research,” NIST, NISTIR 7564, p.21, April, 2009.

[4] K. Goertzel, et al. “Software security assurance: A State-of-art report (soar), Tech.rep”, Information Assurance Technology Analysis Center (IATAC), 2007.

[5] V. Verendel, “Quantified security is a weak hypothesis: a critical survey of results and assumptions”. NSPW ’09: Proceedings of the 2009 workshop on new security paradigms workshop, pp 37–50. ACM, New York, NY, USA.

[6] R. Margaret, “Definition Attack Surface” [Online]. Available: https://whatis.techtarget.com/definition/attack-surface

[7] P. Manadhata dan J. M. Wing. “An attack surface metric”.Technical Report CMU-CS-05-155, 2005.

[8] Mitre.org, “Common Vulnerabilities and Exposures”[online], Available: https://cve.mitre.org/. [Accessed: 03 – Feb – 2019].