MODUL CLOUD COMPUTING DAN HL7 DALAM PELAYANAN KESEHATAN

SECURITY 1

PERTEMUAN 11 (ONLINE)

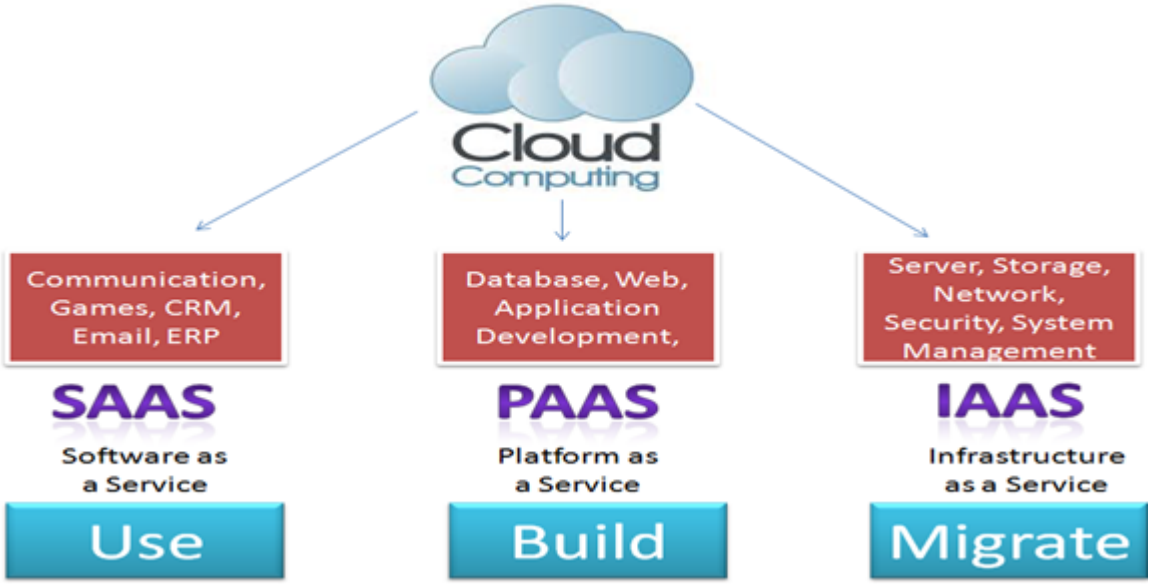
Disusun Oleh

Syefira Salsabila

Cloud computing telah menjadi tren teknologi yang signifikan, dan banyak ahli berharap bahwa komputasi awan akan membentuk kembali teknologi informasi (IT) proses dan pasar IT. Bahkan popularitas penggunaan komputasi awan tanpa SQL ini berkembang khususnya dalam domain data yang besar.

Pada dunia digital seperti sekarang *cloud computing* sudah ada dimana-mana. *Cloud computing* tidak hanya solusi teknis yang dapat mengurangi biaya infrastruktur, tetapi juga model bisnis yang dapat dijual dan disewakan. Dalam berbagai kasus pengguna menggunakan *cloud* tanpa mengetahui bagaimana menggunakannya. Hal ini menimbulkan sejumlah ancaman keamanan yang terkait dengan *cloud computing*. *Cloud computing* didefinisikan oleh *US National Institute of Standards and Technology* (NIST). Mereka mendefinisikan *cloud computing* sebagai model untuk dapat akses jaringan *on-demand* yang ada dimana-mana ke kumpulan sumber daya *computing* yang dapat dikonfigurasi (misalnya jaringan, *server,* penyimpanan, aplikasi, dan layanan).

Cloud computing terbagi menjadi 3 jenis, pada artikel ini pembahasan akan fokus kepada *cloud computing platform as a service* yaitu *cloud database*. Berdasarkan jenisnya *cloud computing* terbagi ke dalam beberapa jenis seperti pada Gambar 1. Jenis *Cloud Computing*.



Gambar 1. Jenis Cloud Computing

Cloud computing tidak hanya mempengaruhi teknologi tetapi juga masuk ke dalam lingkup *database*. Semakin berkembangnya teknologi *cloud computing,* maka permintaan layanan *database* juga bisa semakin banyak. *Cloud database* merupakan *database* yang dapat diakses oleh *client* dari *cloud service* yang didistribusikan ke *user* melalui internet oleh *cloud provider*. *cloud database* membantu penyimpanan data yang semula menggunakan *hard driver,* CD ataupun *hardware* lainnya. Dengan menggunakan *cloud* data hanya perlu disimpan pada *remote database* yang telah disediakan oleh pihak ketiga. Terdapat dua cara *cloud database*, yang pertama adalah menjalankan *database* pada *server* bersama di layanan *cloud*. Kedua adalah *cloud database* yang disediakan

oleh penyedia *cloud* yang menghosting database dan menyediakan akses ke pengguna. Pengguna dapat memilih *database* sesuai dengan kebutuhan setiap penyedia layanan.

1. Untuk mode *virtual machine image*, pengguna *cloud database* dapat membeli ruang server dari penyedia, karena pengguna dapat menjalankan *database* pilihan pengguna dan mengunggah *virtual machine image* dengan salinan database yang optimal. Hal ini dapat dilakukan dengan membuat *cloud database* dengan mudahnya karena vendor sudah membuatnya dengan sederhana dan mudah untuk dipahami. Edisi oracle 11 g *enterprise* bisa untuk *go image* di layanan web Amazon EC2. Seseorang dapat membuat *cloud database* dengan oracle untuk microsoft azure, dengan cara yang sama.
2. Penyedia layanan *cloud database* juga memiliki layanan DBaaS. Salah satu keuntungan dari mode *cloud database* ini yaitu semuanya menjadi tanggung jawab dari penyedia layanan. Pengguna hanya membayar berdasarkan penggunaan saja.
3. Penyedia *hosting cloud database* bekerjasama dengan pihak ketiga. Misalnya MongoDB tersedia di layanan web Amazon dan juga Azure. Konsol yang disediakan oleh penyedia layanan *cloud* membantu mengakses dan menggunakan database. penguna dapat membuat cloud database, backup dan memantau operasi.

Hal yang berkaitan dengan keamanan sistem informasi adalah yang berkaitan dengan jaringan komputer dan data yang ditransfer melalui jaringan komputer tersebut. Beberapa ancaman bisa terjadi dan dapat menyebabkan kerugian yang cukup besar, hal ini harus disadari oleh setiap individu yang terlibat dalam pemakaian *cloud database*. Oleh karena itu, setiap individu sudah seharusnya mengetahui ancaman apa saja yang mungkin terjadi dan pernah terjadi, dan bagaimana hal itu dapat terjadi.

Keberadaan aplikasi yang memiliki tingkat keamanan rendah dapat membahayakan banyak hal penting. Misalnya, data finansial, layanan kesehatan, pertahanan, energi, bahkan infrastruktur krusial. Apalagi, saat ini infrastruktur digital semakin kompleks dan semakin terhubung, menjadikan keamanan aplikasi berbasis web sebagai perhatian utama.

Keamanan Jaringan Informasi

Keamanan jaringan informasi pada *cloud computing* adalah topik yang sangat luas. Keamanan jaringan informasi pada *cloud computing*, khususnya dari segi komunikasi datanya (*secure communication*). Faktor-faktor Keamanan jaringan informasi pada cloud computing (komunikasinya) :

a. Struktur,
b. Metode transmisi,
c. Transport formats,
d. Perhitungan keamanan yang mendukung : integrity, availability, dan authentication (untuk *private* dan *public* jaringan komunikasi).

Diketahui juga komunikasi pada *cloud computing* dikatakan aman jika telah memastikan beberapa hal yaitu:

1. ***Confidentiality***
   Kepastian bahwa hanya orang/bagian yang berhak atau yang seharusnya, yang boleh mengakses data dan menerima data. Beberapa hal yang menjadi bagian dari kebutuhan telekomunikasi dalam menjamin *confidentiality* :
   a. *Network security protocols*
   b. *Network authentication services*
   c. *Data encription services*

2. ***Integrity***
   Kepastian bahwa data tidak berubah karena suatu yang tidak direncanakan atau tidak diinginkan. *Integrity* berarti menjamin pesan telah terkirim dan diterima. Dan pesan tersebut tidak berubah. Beberapa bagian dari *integrity* yaitu:
   a. *Firewall services*
   b. *Communications Security Management*
   c. *Intrusion detection services*

3. ***Availability***
   Kepastian bahwa data atau informasi pada jaringan dapat diakses di waktu dan dimana data/informasi itu dibutuhkan. User yang terotorisasi dapat diijinkan mengakses jaringan atau sistem saat dibutuhkan. Beberapa bagian yang harus diperhatikan untuk menjamin *availability* yaitu:
   a. *Fault tolerance* untuk *availability* data, seperti *backups*, *redundant disk system*
   b. *Acceptable logins and operating process performances*
   c. *Reliable and interoperable security processes and network security mechanisms*

Keamanan Teknologi *Cloud Computing*

Cloud Computing menyajikan banyak tantangan organisasi. Bila organisasi berpindah ke layanan komputasi awan publik tentu infrastruktur sistem komputasi dikendalikan oleh pihak ketiga yaitu *Cloud Service Provider* (CSP) dan tantangan ini harus ditangani melalui inisiatif manajemen. Inisiatif manajemen ini akan memerlukan gambaran jelas peran kepemilikan dan tanggung jawab dari CSP dan organisasi yang berperan sebagai pelanggan. Dalam Presentasi yang dilakukan oleh *Security Issues in Cloud Computing,* Saurabh K Prashar menyatakan bahwa masalah *security* merupakan masalah utama yang timbul dengan adanya teknologi *Cloud Computing*. Dengan adanya teknologi ini, keamanan data dari setiap *user* tidak dapat terjamin, karena setiap data dan informasi yang dimiliki terdapat di *Cloud* atau di internet tepatnya. Hal ini menjadi isu utama dari teknologi *Cloud Computing .*

Cloud Computing merupakan teknologi yang sekarang sedang banyak diadopsi dan menjadi trend dalam proyek-proyek teknologi informasi. Kemanan jaringan informasi

pada *cloud computing* adalah topik yang sangat luas. Ada banyak Aspek yang dapat dilihat dalam mengkaji celah keamanan pada *cloud computing*. Misalnya berdasarkan model layanan-layanan pada *cloud computing* dapat dilihat, apakah celah keamanan jaringan informasi tersebut berada pada model layanan *Software as a Service,* dan atau *Platform as a Service,* dan atau apakah pada *Infrastructure as a Service.*

Bahaya pada Teknologi Cloud Computing

Dengan adanya aspek keamanan, dapat mencegah *danger* atau bahaya dan *vulnerabilities* atau aspek kerentanan terhadap suatu aplikasi yang mengadaptasi teknologi *Cloud Computing*. Untuk aspek *danger* yang dapat timbul dari penggunaan teknologi *Cloud Computing* antara lain:

a. *Disrupts Services*
   Maksudnya adalah layanan terganggu, biasanya hal ini terjadi karena faktor alam, karena cuaca yang kurang baik sehingga koneksi tidak dapat berjalan dengan baik atau adanya bencana alam yang membuat server penyedia layanan bermasalah dan tidak dapat berjalan sebagaimana semestinya.

b. *Theft of Information*
   Hal inilah yang akan dibahas secara lebih mendalam di dalam makalah ini. Pencurian data menjadi isu yang cukup menarik, karena banyaknya cara-cara pencurian data seperti DoS (*Denial of Service*) maupun tipe pencurian data yang lain. Aplikasi dengan teknologi *Cloud Computing* merupakan aplikasi yang sangat rentan dengan pencurian data. Hal ini karena data disimpan di server yang berada di internet, sedangkan jaringan di internet sangat rentan untuk disadap atau dicuri.

c. *Loss of Privacy*
   Bahaya ini adalah dengan hilangnya *Privacy* dari User atau pengguna karena menyerahkan dokumen yang dianggap penting dan rahasia kepada pihak penyedia pelayanan. Hal ini cukup membahayakan bila terjadi kebocoran data. Selain itu hal – hal pribadi milik pengguna sudah tidak dapat terjamin lagi kerahasiannya.

d. *Damage information*
   Data yang dimasukkan melalui jaringan internet dapat rusak, hal ini karena koneksi jaringa yang kurang baik, sehingga data menjadi *corrupt* dan juga tidak digunakan kembali. Hal ini cukup mengganggu bila data yang rusak cukup banyak dan tidak memiliki *Backup*.

Keamanan Data dan Layanan

Pencurian data dalam teknologi *Cloud Computing* merupakan salah satu isu keamanan yang cukup besar. Hal ini karena setiap *hacker* dapat menggunakan berbagai cara untuk mendapatkan informasi yang dibutuhkan dari suatu perusahaan tertentu. Ada

beberapa cara untuk dapat mencegah hal ini dapat terjadi. Beberapa cara pencurian data dapat dilakukan dengan cara sebagai berikut:

a. *Denial of Service*
b. *QoS Violation*
c. *IP Spoofing*
d. *Port Scanning*
e. *ARP Cache Attack*

Keamanan untuk *Cloud Computing* dilakukan pada level – level di bawah ini :

a. *Server access security*
b. *Internet access security*
c. *Database / Datacenter access security*
d. *Data privacy security*
e. *Program access Security*

Setiap level di atas, harus diberikan keamanan yang baik. Misal untuk *server acces* akan diberikan *firewall* yang baik, agar tidak dengan mudah server ditembus oleh *hacker*. Secara khusus akan dibahas mengenai keamanan di dalam *datacenter access security*. Data dapat dicuri secara fisik yaitu mengambil data langsung ke pusat pata/*data* center maupun dapat mencuri dengan cara *hacking* langsung ke dalam basis data. Untuk keamanan di dalam Sebuah data center diperlukan beberapa hal untuk mencegah terjadinya pencurian informasi, hal ini lebih kearah fisik untuk pengamanan data center. Pengamanan ini dilakukan oleh pihak penyedia layanan.

Adapun prosedur keamanan yang dapat dilakukan adalah sebagai berikut :

a. Penggunaan petugas keamanan yang profesional yang dilengkapi dengan kamera pengawas dan berbagai sistem keamanan yang lainnya.
b. Untuk setiap petugas yang sudah tidak bertugas di dalam pusat data harus dihapus hak aksesnya untuk dapat masuk ke dalam pusat data. Bila hal ini tidak dilakukan, maka akan sangat dimungkin bila pencurian data dapat dilakukan.
c. Setiap akses secara elektronik dan akses secara fisik ke dalam pusat data yang dilakukan oleh pegawai harus dilakukan audit secara rutin. Hal ini dimaksudkan agar perusahaan dapat mengetahui *track record* dari setiap pegawai.
d. Digunakan aplikasi untuk melakukan proses audit,hal ini dilakukan agar dapat mengetahui bagaimana data disimpan, dijaga, digunakan dan data tersebut akan diverifikasi dengan peraturan yang sudah ada.

Selain itu untuk keamanan sebuah pusat data diperlukan tempat penyimpanan yang mudah dijangkau tetapi dengan tingkat keamanan yang tinggi dan juga diperlukan sebuah *Backup Storage*. Sedangkan untuk pengamanan dari segi digital, dapat digunakan beberapa cara sebagai berikut:

a. Dapat dibuat 1 buah server yang berada di *Front-End*. Server ini berfungsi untuk menjadi server palsu, yang di dalamnya bukan berisi data asli milik Perusahaan Penyedia Pelayanan, dapat dibuat juga beberapa *server storage* seperti ini agar dapat mengelabui para *hacker* yang akan melakukan pencurian data.

b. Untuk keamanan juga dapat digunakan authentifikasi yang berlapis. Hal ini dimaksudkan agar keamanan dapat berlapis dan juga hanya beberapa user saja yang memiliki *Privilledge* khusus yang dapat mengakses Data Center utama.
c. Dapat menggunakan koneksi VPN (*Virtual Private Network*), dimana antara Server dan User dapat saling berhubungan di dalam satu jalur saja. Jalur Khusus ini dapat membantu keamanan jaringan.
d. Diperlukan juga satu layer khusus untuk *Anti-Virus*, hal ini juga dapat mencegah bila ada penyusup yang akan masuk ke dalam aplikasi

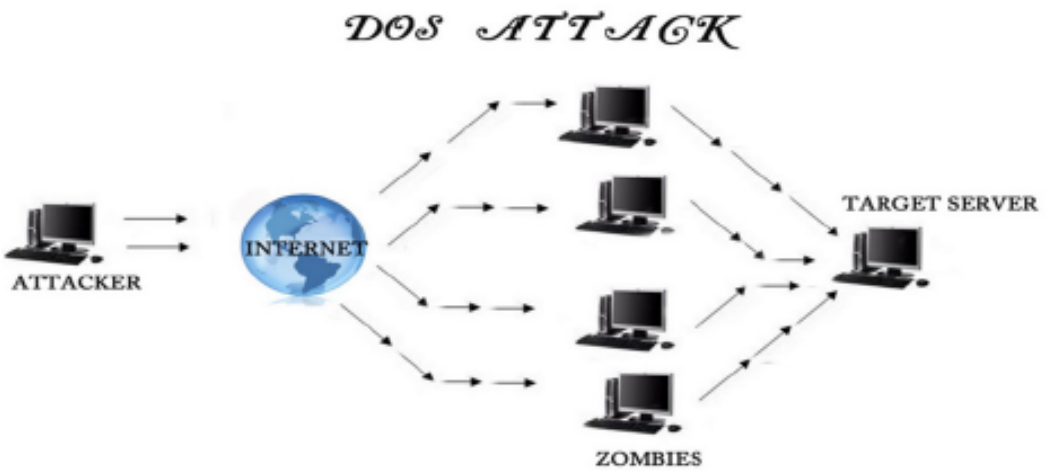
Serangan-serangan yang pernah terjadi pada *cloud database*

Ancaman serangan dan contoh kasus serangan yang mungkin dan pernah terjadi pada *cloud database* adalah sebagai berikut:

1. ***Denial Of Services Attack***

   DOS merupakan serangan yang paling mudah dipasang dan yang paling merusak, namun seiring berjalannya waktu serangan tersebut telah dapat diatasi dengan efisien, beberapa penyedia cloud sudah mengatasi infrastuktur *cloud* guna mencegah ataupun mengurangi serangan ini. namun beberapa solusi belum dapat mendeteksi secara sempurna semua kemungkinan serangan. Tujuan dari serangan ini adalah mencegah pengguna menikmati layanan yang diberikan oleh server. Server melayani permintaan pengguna selama 24 jam, namun apabila terkena serangan ini pengguna tidak bisa menikmati layanan server. Pada kasus di bidang keamanan, tidak ada solusi yang sempurna, akhirnya biasanya bermuara pada kompromi yang ditentukan oleh administrator sistem. Serangan ini adalah serangan favorit para attacker untuk melumpuhkan layanan dari sebuah *host yang* terhubung ke internet sementara. Hampir setiap situs besar menjadi korban serangan ini.

   DoS attack adalah serangan yang bertujuan untuk membuat sebuah server atau website tidak dapat diakses oleh *user* lain. Salah satu contoh serangan yang dapat dilakukan adalah dengan membanjiri jaringan dengan paket-paket sampah. Dengan menerapakan serangan ini, server penyedia *cloud* akan menjadi *down*, sehingga dapat dengan mudah dimasuki oleh seorang penyerang. Gambar 2 berikut mengilustrasikan *DoS Attack*.



Gambar 2. Ilustrasi *DoS* Attack

Serangan ini pernah terjadi pada *server* Dyn DNS. Pada Jumat (21/10/2016) Jam 07.10 UTC menyebabkan sejumlah layanan online besar menjadi lambat bahkan tidak dapat diakses. serangan ini terbilang cukup fatal karena server Dyn DNS digunakan oleh banyak nama besar seperti *GitHub, Twitter, Spotify,* dan lain-lain. Serangan ini merupakan serangan paket data yang menyerang ke *server* membuat server semakin melambat, bahkan jika terlalu besar mengakibatkan server tumbang. Menurut laporan *Ars Technica* serangan ini terjadi dua kali, namun pihak Dyn dapat mengatasi masalah tersebut.

Pada tahun 2000, serangan DDOS terjadi pada beberapa situs web terkenal seperti Amazon mengalami “downtime” selama beberapa jam. Ada lagi serangan yang pernah dilancarkan pada tahun 2002 ketika 9 dari 13 root DNS server diserang dengan menggunakan DDoS yang sangat besar yang disebut dengan “Ping Flood”. Beberapa server pada tiap detiknya mendapatkan lebih dari 150.000 request paket ICMP. Tetapi serangan hanya berlangsung selama setengah jam sehingga lalu lintas internet tidak terlalu terpengaruh oleh serangan tersebut. Setidaknya tidak membuat kerusakan yang fatal.

2. ***Ransomware***

Ransomware merupakan jenis *malware* yang dapat menyandra sistem , paling sering dengan mengenkripsi atau mencuri data, dan melakukan pemerasan. Serangan *ransomware* yang diamati menargetkan kerentanan pada database MongoDB. Peneliti keamanan menyebut serangan ini sebagai *‘ransack’,* dan memperkirakan lebih dari 40.000 database terkena dampak dalam dua minggu pertama. Penelitian menunjukkan bahwa *server Elasticsearch,* yang telah dianggap rentan sehingga bisa diakses melalui internet publik, terkena serangan *ransomware WannaCRYptO.* Terdapat 2.515 *server Elasticsearch* yang harus ditebus, sebanyak 34.298 kasus Elasticsearch yang rentan masih terbuka. Pada hari selanjutnya, jumlah *server* yang terkena meningkat lebih dari 5.000. sebagian

besar *adalah server ElasticSearch* yang rentan terbuka di *Amazon Web* Services (AWS).

Selain itu menurut *website*, Toni Casala menemukan keadaan *ransomware*. Seluruh operasi perusahaan dijalankan dari layanan aplikasi *hosting* di perusahaan *cloud* yang terkelola di California, dari *QuickBooks* sampai *Microsoft Office* dan *Outlook*. Karyawannya menggunakan Citrix untuk terhubung ke *cloud*, dan aplikasi *hosting* perusahaan memetakan *cloud* sebagai *hard drive* pengguna. Menurutnya bekerja pada *cloud* sangat menyenangkan karena dapat menjaga agar komputernya tetap kosong, dengan layanan yang sangat murah, jika dibandingkan menggunakan biaya dengan lebih banyak staf IT. Selain itu, jika membutuhkan *support,* mereka responsif. Tetapi sebelum malam tahun baru, seorang karyawan membuka lampiran email yang tampaknya merupakan faktur. Tiga puluh menit kemudian, tidak ada seorangpun di perusahaan casala yang dapat mengakses 4000 file milik perusahaan yang tersimpan di *cloud drive.* Setiap satu folder memiliki tulisan *"help decrypt"* yang pada dasarnya penyerang meminta tebusan. Penyedia *cloud* mengatakan masih membutuhkan waktu hampir seminggu untuk memulihkan semua file yang disandera. *Malware* tersebut juga mengganggu operasi bagi pelanggan lain di server yang sama. Meskipun perusahaannya memiliki *back up* harian.

Tetapi *malware* yang mengacak-acak file mereka sejenis *ransomware* yang disebut *TeslaCrypt*, berisi kelemahan pengkodean yang memungkinkan perusahaan keamanan dan antivirus membantu korban mendekripsi file tanpa membayar uang tebusan. Terdapat forum bantuan *Bleeping Computer* yeng telah membuat *Tesla Decoder*, yang memungkinkan korban untuk mendekripsi file yang dikunci *TeslaCrypt.*

3. ***SQL Injection***

   Injeksi SQL merupakan salah satu teknik dalam melakukan *web hacking* untuk menggapai akses pada sistem *database* yang berbasis SQL. Teknik ini memanfaatkan kelemahan dalam bahasa pemrograman *scripting* pada SQL dalam mengolah suatu sistem *database.* Hasil yang ditimbulkan dari teknik ini adalah membawa masalah baru yang cukup serius. Salah satu penyebab terjadinya serangan ini adalah tidak adanya penanganan terhadap karakter-karakter tanda petik satu (') dan *double minus* (–) yang menyebabkan suatu aplikasi dapat disisipi dengan perintah SQL. Sehingga seorang hacker dapat menyisipkan perintah kedalam suatu parameter maupun *form.* Serangan ini memungkinkan seseorang dapat *login* ke sistem tanpa memiliki *account.* Memungkinkan juga seseorang dapat merubah, menghapus, ataupun menambahkan data yang berada dalam *database.* Bahkan dapat mematikan *database* tersebut, sehingga tidak dapat memberi layanan pada *web server.*

4. ***Exploit***

Merupakan sebuah kode yang menyerang keamanan komputer secara spesifik. Exploit banyak digunakan untuk penentrasi baik secara legal ataupun ilegal untuk mencari kelemahan (*vulnerability*) pada komputer tujuan. Bisa juga dikatakan sebuah perangkat lunak yang menyerang kerapuhan keamanan (*security vulnerability*) yang spesifik namun tidak selalu bertujuan untuk melancarkan aksi yang tidak diinginkan. Banyak peneliti keamanan komputer menggunakan *exploit* untuk mendemonstrasikan bahwa suatu sistem memiliki kerapuhan.

Memang ada badan peneliti yang bekerja sama dengan produsen perangkat lunak. Peneliti itu bertugas mencari kerapuhan dari sebuah perangkat lunak dan kalau mereka menemukannya, mereka melaporkan hasil temuan ke produsen agar produsen dapat mengambil tindakan. Meskipun demikian, *exploit* kadang menjadi bagian dari *suatu malware* yang bertugas menyerang kerapuhan keamanan.

Ada beberapa metode untuk mengklasifikasi *exploit*. Yang paling umum adalah dengan melihat cara *exploit* membuat kontak dengan perangkat lunak yang rentan. *Remote exploit* (eksploit jarak jauh) bekerja melalui jaringan dan mengeksploitasi celah keamanan tanpa adanya akses terlebih dahulu ke sistem korban. *Local exploit* (eksploit lokal) mengharuskan adanya akses terlebih dahulu ke sistem yang rentan dan biasanya meningkatkan keleluasaan orang yang menjalankan *exploit* melebihi yang diberikan oleh administrator sistem. *Exploit* yang menyerang aplikasi klien juga ada, biasanya terdiri dari server-server yang dimodifikasi yang mengirimkan *exploit* jika diakses dengan aplikasi klien. *Exploit* yang menyerang aplikasi klien juga mungkin memerlukan beberapa interaksi dengan pengguna, dengan demikian dapat digunakan dalam kombinasi dengan metode social engineering. Ini adalah cara hacker masuk ke komputer dan situs web untuk mencuri data.

Klasifikasi lain adalah dengan tindakan terhadap sistem korban: unauthorized akses data, eksekusi kode sewenang-wenang, penolakan layanan. Banyak *exploit* dirancang untuk memberikan akses tingkat-"superuser" ke sistem komputer. Namun, namun mungkin juga menggunakan beberapa *exploit*, untuk mendapatkan akses tingkat rendah terlebih dahulu, kemudian meningkatkan hak akses berulang kali sampai mencapai *root*.

Biasanya *exploit* tunggal hanya dapat mengambil keuntungan dari satu celah keamanan software tertentu. Sering kali, setelah *exploit* diterbitkan, celah keamanan sistem diperbaiki melalui tambalan sehingga *exploit* tak berlaku lagi untuk perangkat lunak versi terbaru. Hal ini menjadi alasan mengapa beberapa blackhat hacker tidak mempublikasikan *exploit* mereka tetapi merahasiakannya untuk diri sendiri atau hacker lainnya. *Exploit* tersebut disebut sebagai '*exploit* zero day' dan untuk mendapatkan akses ke *exploit* tersebut adalah keinginan utama dari penyerang-penyerang amatir, yang sering dijuluki script kiddie

Exploit pada umumnya dikategorikan dan dinamai berdasarkan kriteria berikut:

a. Jenis celah keamanan yang mereka eksploitasi
b. Apakah mereka perlu dijalankan pada mesin yang sama dengan program yang memiliki celah (lokal) atau dapat dijalankan pada satu mesin berbeda untuk menyerang program yang berjalan pada komputer lain (remote).
c. Hasil dari menjalankan exploit (EoP, DoS, Spoofing, dll)

5. Pivoting

Pivoting mengacu pada metode yang digunakan oleh penguji penetrasi yang menggunakan sistem yang telah dikuasai untuk menyerang sistem lain pada jaringan yang sama untuk menghindari larangan seperti konfigurasi firewall, yang dapat melarang akses langsung ke semua mesin. Sebagai contoh, seorang penyerang menguasai web server pada jaringan perusahaan, penyerang kemudian dapat menggunakan web server yang telah dikuasai untuk menyerang sistem lain pada jaringan. Jenis serangan ini sering disebut serangan multi-lapis. Pivoting juga dikenal sebagai *island hopping*.

Pivoting dapat dibedakan menjadi *proxy pivoting* dan *VPN pivoting*:

a. **Proxy pivoting** umumnya digambarkan sebagai tindakan menelusuri jalur melalui target yang dikuasai menggunakan proxy payload pada mesin dan meluncurkan serangan dari komputer ini. Jenis *pivoting* ini terbatas pada TCP dan UDP tertentu yang didukung oleh proxy.
b. **VPN pivoting** memungkinkan penyerang untuk membuat terowongan lapisan ke-2 yang terenkripsi ke dalam mesin yang dikuasai untuk mengarahkan lalu lintas jaringan apapun melalui mesin target itu, misalnya untuk menjalankan scan kerentanan pada jaringan internal melalui mesin yang dikuasai, secara efektif memberikan akses jaringan secara penuh kepada penyerang seolah-olah mereka berada di belakang firewall.

Biasanya, aplikasi proxy atau VPN yang memungkinkan *pivoting*, dijalankan pada komputer target sebagai payload (perangkat lunak) dari *exploit*.

6. Brute Force Attack

Brute Force Attack adalah metode untuk meretas *password* (**password cracking**) dengan cara mencoba semua kemungkinan kombinasi yang ada pada "***wordlist***". Metode ini dijamin akan berhasil menemukan *password* yang ingin diretas. Namun, proses untuk meretas *password* dengan menggunakan metode ini akan memakan banyak waktu. Lamanya waktu akan ditentukan oleh panjang dan kombinasi karakter *password* yang akan diretas.

Brute Force Attack menggunakan formula sebagai berikut:

$$KS = L^{(m)} + L^{(m+1)} + L^{(m+2)} + \ldots + L^{(M)}$$

Keterangan:

L = Jumlah karakter yang kita ingin definisikan
M =Panjang maksimum kata kunci
m = Panjang minimum kata kunci

Contoh jika kita ingin meretas *password* dengan panjang password 5 karakter dan hanya menggunakan kombinasi huruf kecil (‘a’ – ‘z’ = 26), maka **Brute Force Cracker** harus mencoba KS = $26^1 + 26^2 + 26^3 + \ldots + 26^5$ = 12356721 kata yang berbeda. Istilah **Brute Force** sendiri di populerkan oleh **Kenneth Thomson**, dengan mottonya “**When in doubt, use brute-force**” (jika rugu, gunakan brute-force).

Brute Force dapat digunakan untuk meretas password secara offline maupun online, namun kombinasi karakter password yang panjang terkadang membuat seorang attacker putus asa lalu menghentikan serangannya sehingga membuat metode ini menjadi sia-sia. Seperti kata **Kenneth Thomson**, lakukanlah **Brute Force** jika anda merasa ragu dengan kombinasi karakter yang anda miliki.

Apa yang perlu dilakukan?

Untuk mencegah dan mengatasi permasalahan yang terjadi adalah dengan cara meningkatkan *awareness* mengenai keamanan informasi dan data yang dimiliki. Jika data yang dimiliki adalah berharga maka perlu dilakukan antisipasi dan keamanan yang baik. Beberapa diantaranya yang bisa dilakukan dalah dimulai dengan memilih penyedia *cloud* yang benar-benar terpercaya dan memiliki keamanan informasi yang cukup baik. Misalnya dalam kasus *SQL injection* maka bisa dengan menjadikan variabel *get* menjadi *absolute integer*. Dengan menambahkan variabel get berisi enkripsi md5 yang divariasi dengan url. Dengan melakukan enkripsi *password* ataupun merubah algoritma autentikasi *login* khusus untuk *form login*, atau juga dengan memfilter inputan yang masuk. Begitu pula dengan *ransomware*, dengan memiliki *backup* dan meng*update* sistem secara berkala. Berhati-hati terhadap *fraud* yang mencurigakan. Kasus ini hanya beberapa yang terjadi, dan mungkin banyak kejadian yang lain yang menimbulkan akibat yang lebih parah. Maka sangat penting menumbuhkan *awareness* terhadap keamanan informasi di era serba digital seperti saat ini.

OWASP

OWASP adalah organisasi non-profit yang berfokus pada keamanan berbasis web. Orang-orang yang terlibat di dalamnya merupakan voluntir. Di perjalanannya, semua piranti, dokumen, forum, serta cabang organisasi OWASP bersifat gratis dan terbuka bagi setiap orang yang tertarik dalam hal keamanan aplikasi. Hal ini dimaksudkan sebab mayoritas pengembangan keamanan aplikasi membutuhkan partisipasi aktif dari banyak orang di seluruh belahan dunia. Berikut Gamatechno beberkan penjelasan tentang OWASP Top 10.

1. ***Injection***

Dalam praktik penggunaan SQL, OS, dan LDAP, injeksi adalah hal yang sangat riskan untuk terjadi. Injeksi biasanya dilakukan dengan memasukkan data yang tidak terpercaya ke dalam interpreter sebagai bagian dari command atau query. Data yang dimasukkan oleh injektor dapat menipu interpreter untuk mengeksekusi perintah tertentu atau mengakses data rahasia tanpa izin.

2. ***Broken Authentication and Session Management***

   Fungsi pada aplikasi berbasis web yang berkaitan dengan autentifikasi dan manajemen sesi seringkali tidak terimplementasikan dengan baik. Apabila hal ini terjadi di level parah, penyerang sistem akan dengan mudah mencuri dan memanfaatkan password serta data pribadi lainnya yang akan merugikan pengguna.

3. ***Cross-Site Scripting* (XSS)**

   Kelemahan dalam XSS terjadi ketika sebuah aplikasi mengakses data yang tidak terpercaya dan mengirimkannya lewat web tanpa ada konfirmasi validasi sebagaimana mestinya. Kejadian XSS akan memberikan keleluasaan bagi penyerang sistem untuk menggunakan script dari browser guna mengakses web tanpa izin. Misalnya mengarahkan ke website palsu atau bahkan melakukan redirect ke situs berbahaya.

4. ***Insecure Direct Object References***

   Objek langsung di sini berkaitan ketika developer mengekspos referensi ke dalam implementasi objek internal. Misalnya ke file, direktori, atau database key. Tanpa memiliki accsess control check dan perlindungan lain, penyerang dapat memanipulasi referensi ini untuk mengakses data rahasia.

5. ***Security Misconfiguration***

   Selama ini, sistem keamanan yang bagus membutuhkan konfigurasi yang terjamin guna mengakses aplikasi, framework, web server, aplikasi server, database server, hingga platform. Sebab, setingan default seringkali tidak aman. Selain itu, pembaruan rutin terhadap software pun menjadi sebuah keharusan.

6. ***Sensitive Data Exposure***

   Banyak aplikasi berbasis web yang belum melindungi data sensitif secara layak. Misalnya data kartu kredit hingga data autentifikasi. Penyerang sistem sangat mungkin mencuri atau memodifikasi data bersistem pengamanan lemah tersebut untuk melakukan tindakan penipuan, pencurian identitas, atau kriminalitas lain.

7. ***Missing Function Level Access Control***

Mayoritas aplikasi berbasis web akan memverifikasi fungsi akses sebelum membuat fungsi tersebut ada di user interface. Faktanya, aplikasi juga perlu melakukan kontrol akses yang sama ke server tiap kali fungsi itu dijalankan. Apabila permintaan tidak terverifikasi, maka penyerang bisa dengan mudah mengakses fungsi privat tanpa izin.

8. ***Cross-Site Request Forgery* (CSRF)**

   Cara kerja CSRF adalah dengan memaksa masuk ke browser pengguna yang kemudian mengirimkan permintaan HTTP, termasuk cookies, serta berbagai informasi rahasia yang tersimpan di browser, ke aplikasi web gadungan. Hal ini akan membuat pengguna seolah-olah mengakses aplikasi tersebut secara langsung, padahal tidak.

9. ***Using Known Vulnerable Components***

   Komponen dasar seperti database, famework, dan berbagai modul software kebanyakan dijalankan dengan hak penuh. Apabila komponen yang riskan dieksploitasi, bisa menyebabkan kehilangan data dan pengambil-alihan server.

10. ***Unvalidated Redirects and Forwards***

    Aplikasi berbasis web yang digunakan user seringkali melakukan redirect dan forward ke halaman lain atau bahkan website lain. Tindakan semacam ini, tanpa validasi yang benar, dapat mengarahkan user ke laman phishing, malware, maupun menggunakannya untuk mengakses laman berbahaya lain.

Tujuan dari OWASP Top 10 di atas sebenarnya sangat sederhana. Yaitu meningkatkan kesadaran publik tentang keamanan aplikasi dengan cara mengidentifikasi hal-hal apa saja yang paling krusial dan sering tidak disadari. Dengan semakin tingginya tingkat keamanan, maka potensi terjadinya tindakan yang bisa merugikan hingga kriminal melalui web bisa semakin ditekan. (anas)

Keamanan pada sebuah jaringan Internet merupakan hal yang sangat penting untuk menjaga jaringan tersebut dari aktifitas yang bertujuan untuk menyerang dan menyusup ke dalam jaringan tersebut serta menjamin ketersediaan layanan untuk para penggunanya. Serangan yang masuk pada sebuah jaringan bisa menyebabkan jaringan tersebut lumpuh, atau menurunnya performa jaringan tersebut jika tidak ditangani dengan cepat dan tepat. Untuk itu sebuah jaringan memerlukan sebuah sistem yang bertujuan untuk memantau secara *real-time* pada sebuah jaringan guna mendeteksi aktifitas yang diindikasi sebagai serangan sehingga administrator bisa menaggulangi ancaman yang mungkin akan terjadi secara cepat sehingga mengurangi dampak buruk dari serangan tersebut. Salah satu solusi yang dapat digunakan adalah IDS (*Intrusion Detection System*). IDS adalah sebuah sistem yang dapat secara otomatis memonitor kejadian pada jaringan komputer dan dapat menganalisa masalah keamanan jaringan. IDS mampu mendeteksi penyusup dan memberikan respon secara *real time*. Pada umumnya

IDS terbagi menjadi dua jenis yaitu *rule based system* yang mendeteksi serangan berdasarkan aturan yang telah didefinisikan sebelumnya dan *adaptive system* yang mampu mengenali jenis serangan baru.

Intrusion Detection System (disingkat **IDS**) adalah sebuah metode yang dapat digunakan untuk mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IDS dapat melakukan inspeksi terhadap lalu lintas *inbound* dan *outbound* dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti dari percobaan intrusi (penyusupan).

Jenis-jenis IDS

Ada dua jenis IDS, yakni:

a. *Network-based Intrusion Detection System* (NIDS): Semua lalu lintas yang mengalir ke sebuah jaringan akan dianalisis untuk mencari apakah ada percobaan serangan atau penyusupan ke dalam sistem jaringan. NIDS umumnya terletak di dalam segmen jaringan penting di mana server berada atau terdapat pada "pintu masuk" jaringan. Kelemahan NIDS adalah bahwa NIDS agak rumit diimplementasikan dalam sebuah jaringan yang menggunakan *switch* Ethernet, meskipun beberapa *vendor* switch Ethernet sekarang telah menerapkan fungsi IDS di dalam *switch* buatannya untuk memonitor port atau koneksi.

b. *Host-based Intrusion Detection System* (HIDS): Aktivitas sebuah *host* jaringan individual akan dipantau apakah terjadi sebuah percobaan serangan atau penyusupan ke dalamnya atau tidak. HIDS seringnya diletakkan pada server-server kritis di jaringan, seperti halnya *firewall*, *web server*, atau server yang terkoneksi ke Internet.

Kebanyakan produk IDS merupakan sistem yang bersifat pasif, mengingat tugasnya hanyalah mendeteksi intrusi yang terjadi dan memberikan peringatan kepada administrator jaringan bahwa mungkin ada serangan atau gangguan terhadap jaringan. Akhir-akhir ini, beberapa *vendor* juga mengembangkan IDS yang bersifat aktif yang dapat melakukan beberapa tugas untuk melindungi host atau jaringan dari serangan ketika terdeteksi, seperti halnya menutup beberapa port atau memblokir beberapa alamat IP. Produk seperti ini umumnya disebut sebagai *Intrusion Prevention System* (IPS). Beberapa produk IDS juga menggabungkan kemampuan yang dimiliki oleh HIDS dan NIDS, yang kemudian disebut sebagai sistem hibrid (*hybrid intrusion detection system*).

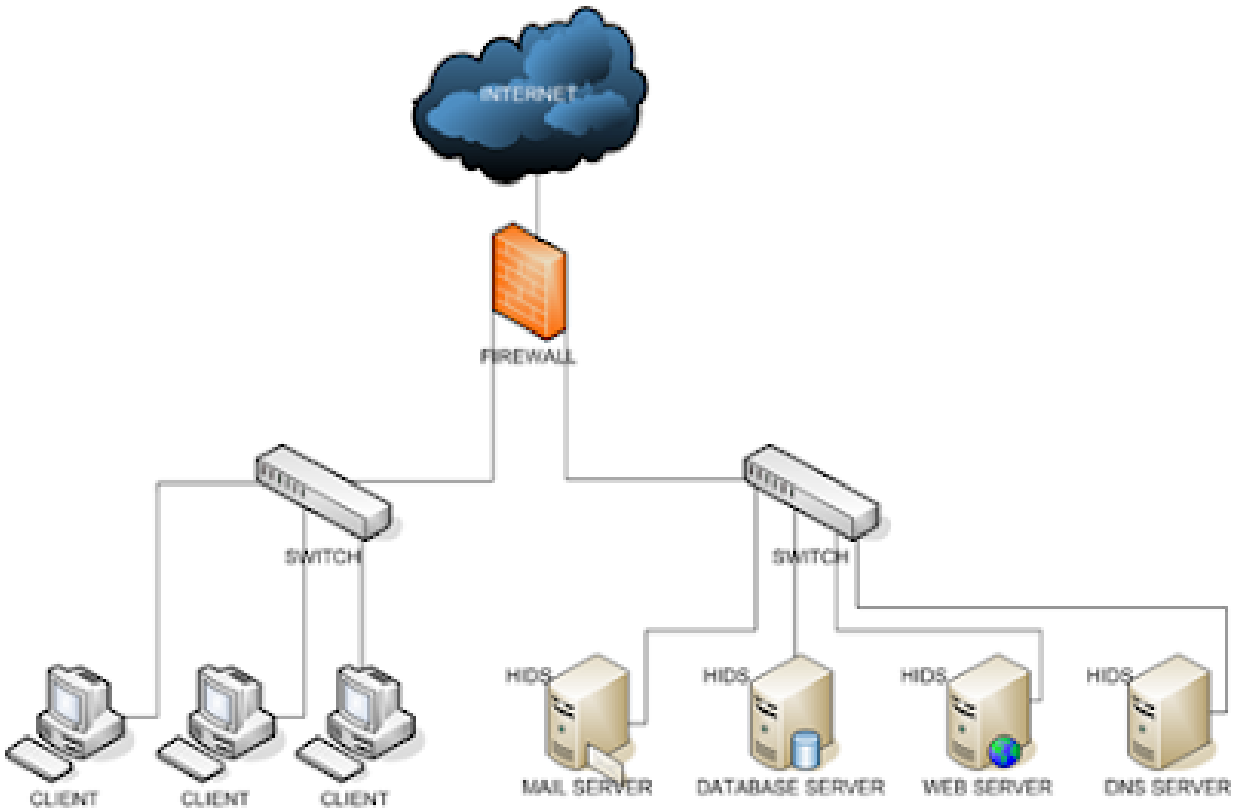
Ada beberapa cara bagaimana IDS bekerja. Cara yang paling populer adalah dengan menggunakan pendeteksian berbasis *signature* (seperti halnya yang dilakukan oleh beberapa antivirus), yang melibatkan pencocokan lalu lintas jaringan dengan basis data yang berisi cara-cara serangan dan penyusupan yang sering dilakukan oleh penyerang. Sama seperti halnya antivirus, jenis ini membutuhkan pembaruan terhadap basis data *signature* IDS yang bersangkutan.

Metode selanjutnya adalah dengan mendeteksi adanya anomali, yang disebut sebagai **Anomaly-based IDS**. Jenis ini melibatkan pola lalu lintas yang mungkin merupakan sebuah serangan yang sedang dilakukan oleh penyerang. Umumnya, dilakukan dengan menggunakan teknik statistik untuk membandingkan lalu lintas yang sedang dipantau dengan lalu lintas normal yang biasa terjadi. Metode ini menawarkan kelebihan dibandingkan *signature-based IDS*, yakni ia dapat mendeteksi bentuk serangan yang baru dan belum terdapat di dalam basis data *signature* IDS. Kelemahannya, adalah jenis ini sering mengeluarkan pesan *false positive*. Sehingga tugas administrator menjadi lebih rumit, dengan harus memilah-milah mana yang merupakan serangan yang sebenarnya dari banyaknya laporan *false positive* yang muncul.

Teknik lainnya yang digunakan adalah dengan memantau berkas-berkas sistem operasi, yakni dengan cara melihat apakah ada percobaan untuk mengubah beberapa berkas sistem operasi, utamanya berkas log. Teknik ini seringnya diimplementasikan di dalam HIDS, selain tentunya melakukan pemindaian terhadap log sistem untuk memantau apakah terjadi kejadian yang tidak biasa.

Host Intrusion Detection System

Host Intrusion Detection System dapat didefinisikan sebagai suatu sistem yang mampu mendeteksi aktifitas yang mencurigakan dalam sebuah jaringan yang menuju ke perangkat komputer tersebut, HIDS mampu melakukan pedeteksian dengan cara melakukan pemantauan terhadap lalu lintas *(traffic)* yang keluar maupun masuk dalam sebuah sistem atau jaringan ataupun mendeteksi berdasarkam perbandingan pola lalu lintas jaringan normal yang ada dan kemudian membandingkannya dengan lalu lintas yang ada pada jaringan komputer tersebut.



Gambar 3: *Host Based Intrusion Detection System (HIDS)*

Komponen-komponen HIDS:

a. IDS *Rule.* Merupakan database yang berisi pola-pola serangan berupa signature jenis serangan. *Rule* IDS ini, harus di *update* secara rutin sehingga IDS mampu mendeteksi jenis serangan baru.
b. IDS *Engine.* Merupakan program yang berjalan sebagai proses yang selalu bekerja untuk membaca paket data dan kemudian membandingkan dengan *rule* IDS.
c. IDS *Alert.* Merupakan catatan serangan pada deteksi penyusupan, jika IDS *engine* mengukumi paket daya yang lewat sebagai serangan, maka IDS *engine* akan mengirimkan *alert* berupa *log file.* Untuk kebutuhan analisa, *alert* dapat disimpan di dalam *database,* sebagai contoh BASE *(Basic Analysis and Securtiy Engine)* yang berfungsi untuk mencari dan mengolah *database* dari *alert network security* yang dibangkitkan oleh perangkat lunak pendeteksi intrusi (IDS).

DAFTAR PUSTAKA

Aurora, Indu dan Gupta, Anu, “Cloud Database: A Paradigm Shift In Databases”. (2012)

Krutz, Ronald L. And Vines, Russell Dean., 2010, *CLOUD SECURITY*, a comprehensive guide to secure cloud computing. Wiley Publishing Inc. Kanada, USA.

Setiawan, Deris, 2010, *Teknologi Cloud Computing*, Fasilkom, Universitas Sriwijaya.

Sinambela, Josua M, “Cloud Computing Security”. (2013)