ANALISIS KEAMANAN SISTEM LOGIN

Dyna Marisa Khairina

Program Studi Ilmu Komputer, FMIPA Universitas Mulawarman
Email : dyna_mkh0707@yahoo.co.id

ABSTRAK

Sistem *login* merupakan suatu hal yang pasti ditemukan didalam dunia *internet*. Saat seseorang melakukan *login* pastinya akan memasukkan *password* dimana *password* tersebut bersifat privasi dan rahasia. Oleh karena itu, masalah keamanan menjadi masalah yang sangat penting mengingat *internet* merupakan jaringan publik yang saling terhubung dalam suatu jaringan dan akan sangat berbahaya jika *password* yang dimasukkan *user* tersebut tidak dienkripsi sebelum dikirim ke *server* melalui jaringan. Disitulah celah kesempatan bagi para *sniffer* atau pengendus dapat melacak *password* atau data *user*. Sistem *login* dibuat dengan pemrograman PHP kemudian dilakukan pengamanan dengan enkripsi menggunakan MD5 yang dikombinasikan dengan pengacak atau menggabungkan *password* asli dengan suatu *string* tertentu lalu dienkripsi. Isi pengacak serta format untuk enkripsi hanya yang membuat aplikasi yang mengetahuinya. Setelah dilakukan pengamanan pada sistem *login* kemudian dilakukan analisis keamanannya dengan menggunakan sebuah *software* yaitu *wireshark* dan dapat dideteksi mana *password* yang dienkripsi dan yang tidak dienkripsi.

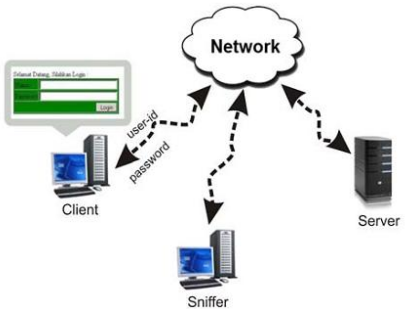
Kata Kunci: *Keamanan Sistem Login, MD5, Wireshark*

PENDAHULUAN

Masalah keamanan merupakan masalah yang penting dan utama dalam sistem komputer yang terhubung dalam suatu jaringan. Data maupun informasi menjadi target serangan oleh pihak-pihak yang tidak bertanggungjawab sehingga perlu untuk menjaga integritas data dan informasi. Dalam aplikasi web dibutuhkan mekanisme yang dapat melindungi data dari pengguna yang tidak berhak. Mekanisme ini dapat diimplementasikan dalam bentuk sebuah proses *login* yang biasanya terdiri dari tiga buah tahapan yaitu identifikasi, otentikasi dan otorisasi. Seiring banyaknya fasilitas *internet* yang membutuhkan akses masuk (*login*) seperti *email*, akses *web server* maupun *account* lainnya, maka *user* perlu lebih berhati-hati terutama jika *account* tersebut sangat rahasia dan berharga mengingat internet merupakan jaringan publik.

IDENTIFIKASI MASALAH

Sarana untuk melindungi data pada sistem serta untuk menentukan bahwa seseorang yang mengakses sistem adalah autentik atau asli adalah dengan autentikasi, yaitu proses memverifikasi identitas dari seorang anggota yang memberikan suatu data dan integritas dari data tersebut. Autentikasi adalah program pengamanan untuk mencegah pihak-pihak yang tidak memiliki otoritas dalam mengakses sistem. Salah satu metode untuk melakukan autentikasi adalah dengan menggunakan *password*. Untuk menjaga agar *password* tidak mudah dibaca oleh *sniffer* atau pengendus diperlukan proses pengamanan dengan melakukan enkripsi di sisi *client* sebelum data dikirimkan ke *server* melalui *internet* sehingga *password* yang dikirim melalui jaringan *internet* berbentuk *ciphertext*, setelah itu pada sisi *server* dilakukan dekripsi data kembali sehingga didapatkan data asli. Proses dekripsi *form login* digambarkan dalam gambar berikut:



Gambar 1. Proses Dekripsi Form *Login*

Program autentikasi ini memeriksa *client* dari dua parameter yaitu berdasarkan *userid* dan *password client*. *Client* hanya dapat mengakses sistem jika memenuhi dua pengamanan tersebut.

TUJUAN PENELITIAN

Tujuan dilakukannya penelitian adalah menganalisis kelemahan dari suatu sistem *login* dan memberikan solusi atas permasalahan yang ditemukan sehingga akan didapat aplikasi yang lebih baik dan aman.

TINJAUAN PUSTAKA

Sistem Login

Sistem login (*login*, juga biasa disebut *log in*, *log on*, *signon*, *sign on*, *signin*, *sign in*) adalah proses untuk mengakses komputer dengan memasukkan identitas dari akun pengguna dan kata sandi untuk mendapatkan hak akses menggunakan sumber daya komputer tujuan [2].

Pada saat melakukan *login* untuk masuk kedalam sistem, *user* akan diminta untuk memasukkan identitas *user* seperti *userid* dan *password* sebagai antisipasi dalam hal pengamanan sistem. *Password* dapat diubah sesuai dengan kebutuhan sedangkan *userid* tidak pernah diubah karena berupa identitas unik yang merujuk ke *user* tertentu. Jika kedua pengamanan tersebut berhasil atau memenuhi maka *user* memiliki hak untuk mengakses sistem.

Proses *login* memiliki mekanisme yang terdiri dari tiga tahap, yaitu:

1. Identifikasi. Tahap dimana *user* memberitahukan identitas dirinya.
2. Otentikasi. Tahap dimana *user* memverifikasi klaimnya *user* yaitu sesuatu yang diketahui, seperti kode PIN atau *password*; sesuatu yang dimiliki, seperti kartu magnetik; dan sesuatu yang menjadi jati diri, seperti sidik jari.
3. Otorisasi. Tahap terakhir dimana jika identifikasi *user* telah sukses atau benar, sistem menyelesaikan proses *login*nya dan mengasosiasikan identitas *user* dan informasi kontrol akses dengan sesi *user*.

Autentikasi

Autentikasi merupakan proses validasi user saat masuk kedalam sistem. Pada saat memasuki sistem, password dari user dicek melalui proses yang mengecek langsung ke daftar yang diberikan hak untuk masuk kedalam sistem tersebut. Autorisasi ini di *set up* oleh administrator, webmaster atau pemilik situs. Untuk proses tersebut, masing-masing user akan dicek dari data yang diberikannya seperti *userid* dan *password* serta hal-hal lain yang tidak tertutup kemungkinan. Melakukan autentikasi terhadap sebuah objek adalah melakukan konfirmasi terhadap kebenarannya. Sedangkan melakukan autentikasi terhadap seseorang adalah untuk memverifikasi identitasnya.

Proses autentikasi pada prinsipnya berfungsi sebagai kesempatan *user* dan pemberi layanan dalam proses pengaksesan *resource*. *User* harus mampu memberikan informasi yang dibutuhkan pemberi layanan untuk berhak mendapatkan *resource*nya. Sedangkan pihak pemberi layanan harus mampu menjamin bahwa pihak yang tidak berhak tidak akan dapat mengakses *resource* ini.

Autentikasi bertujuan untuk membuktikan identitas *user* sebenarnya. Ada banyak cara untuk membuktikan *user* sebenarnya. Ada empat kategori metode autentikasi:

1. *Something You Know*
   Merupakan metode autentikasi yang paling umum. Metode ini mengandalkan kerahasiaan informasi dan biasanya identik dengan *password*, *userid* atau PIN. Metode ini berasumsi bahwa tidak ada seorangpun yang mengetahui rahasia itu kecuali dirinya.
2. *Something You Have*
   Merupakan faktor tambahan untuk membuat autentikasi menjadi lebih aman. Metode ini mengandalkan barang yang sifatnya unik, seperti kartu magnetic/smartcard, hardware token, USB token dan sebagainya. Metode ini berasumsi bahwa tidak ada seorangpun yang memiliki barang tersebut kecuali dirinya.
3. *Something You Are*
   Merupakan metode yang paling jarang dipakai karena faktor teknologi dan manusia juga. Metode ini mengandalkan keunikan bagian-bagian tubuh yang tidak mungkin ada pada orang lain seperti sidik jari, suara atau sidik retina mata. Metode ini berasumsi
   bahwa bagian tubuh seseorang seperti sidik jari dan sidik retina tidak mungkin sama dengan orang lain.
4. *Something You Do*
   Merupakan metode yang melibatkan bahwa setiap user dalam melakukan sesuatu dengan cara berbeda, contohnya penggunaan analisis suara (*voice recognation*) dan analisis tulisan tangan.

Kriptografi MD5

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti keabsahan, integritas data serta autentikasi data.

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga [4].

Terminologi dalam kriptografi diantaranya enkripsi yang merupakan mekanisme untuk merubah *plaintext* menjadi *chipertext*. Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga

tidak dapat dibaca oleh orang yang tidak berhak. Fungsi hash merupakan fungsi yang secara efisien mengubah string masukan dengan panjang berhingga menjadi string keluararn dengan panjang tetap yang disebut nilai hash.

MD5 adalah salah satu dari serangkaian algoritma *message-digest* yang dirancang oleh Profesor Ronald Rivest dari Massachusetts Institute of Technology (MIT). Ketika kerja analitis menunjukkan bahwa pendahulu MD5 yaitu MD4 mulai tidak aman, maka MD5 kemudian dirancang pada tahun 1991 sebagai pengganti dari MD4. *Hash* MD5 sepanjang 128-*bit* (16-*byte*), yang dikenal juga sebagai intisari pesan, *message digest* secara tipikal ditampilkan dalam bilangan heksadesimal 32-digit. MD5 telah dimanfaatkan secara bermacam-macam pada aplikasi keamanan dan MD5 juga umum digunakan untuk melakukan pengujian integritas data.

PERANCANGAN APLIKASI

Rancang *Database*

Untuk *database* aplikasi ini hanya dibuat satu tabel yaitu tabel *user* dengan struktur sebagai berikut:

Tabel 1. Struktur Tabel User

No.	Field	Type
1	Userid	Varchar
2	Password	Varchar
3	Nama	Varchar
4	Email	Varchar

Tabel ini digunakan untuk proses *register user* sebelum dapat melakukan *login* untuk memiliki akses kedalam sistem.

Desain *Interface* Sistem

Rancangan *form* untuk aplikasi ini adalah sebagai berikut:

Gambar 2. Form untuk melakukan *register*

Register dilakukan dengan mengisi *password* lebih dari satu kali sehingga *user* dapat memastikan bahwa *password* yang dimasukkan adalah benar dari sisi ejaan.

Dari form *register* selanjutnya akan dilakukan pemrosesan *register user* dengan melakukan enkripsi *password* sebelum enkripsi tersebut disimpan dalam *database* sistem.

```
// mengenkripsi password dengan md5()
$password1 = md5($password1);
```

Bagian *script* diatas adalah proses enkripsi *password* hanya dengan menggunakan MD5. Penggunaan MD5 hanya untuk menghindari pengiriman *password* secara apa adanya tanpa adanya perlindungan atau pengamanan ke *webserver*. Pada masa sekarang sudah banyak *tool* yang dapat mendekripsi hasil enkripsi MD5. Untuk pengamanan lebih lanjut maka dibuatlah kombinasi MD5 dengan pengacak atau menggabungkan *password* asli dengan suatu *string* tertentu lalu dienkripsi.

```
// mengenkripsi password dengan md5() dan pengacak
$password1 = md5($pengacak . md5($password1) . $pengacak);
```

Isi pengacak serta format untuk enkripsi hanya pembuat aplikasi yang mengetahuinya.

```
// membuat isi pengacak
$pengacak  = "DMK225H1MAZ221S5";
```

Jika *user* telah berhasil melakukan register, selanjutnya ke proses *login*.

Gambar 3. Form untuk melakukan *login*

Password yang dimasukkan *user* ke dalam *form login* akan dicek kesesuaiannya dari *form* dan dari *database*, harus menggunakan pengacak dan *rule* yang sama dengan proses mengenkripsi *password* sebelum disimpan ke *database*.

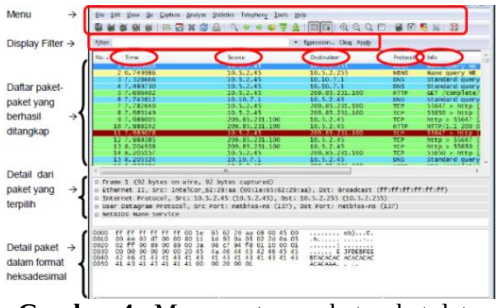
Proses Validasi dengan *Session*

Validasi berfungsi untuk mencegah *by pass* yang dilakukan oleh *user* yang tidak bertanggungjawab yang ingin masuk ke *resource*. Validasi dapat mengecek keberadaan *session* untuk *userid*. Jika *user* tidak melakukan *login*, maka session *userid* tidak pernah dibuat. Perintah isset() digunakan untuk mengecek keberadaan suatu variabel (dalam hal ini variabel *session* untuk *userid*). Perintah ini akan menghasilkan nilai TRUE jika variabel yang dicek ada, dan FALSE jika variabel tidak ada. Script untuk proses validasi disisipkan disetiap halaman yang sifatnya *private* atau yang membutuhkan autentifikasi *user*.

ANALISIS WIRESHARK

Wireshark

Wireshark adalah sebuah *Network Packet Analyzer*. *Network Packet Analyzer* akan mencoba menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi dipaket tersebut sedetail mungkin. *Network Packet Analyzer* diumpamakan sebagai alat untuk memeriksa apa yang sebenarnya sedang terjadi di dalam kabel jaringan. Wireshark juga merupakan salah satu *tool* gratis terbaik untuk menganalisa paket jaringan. Salah satu penggunaan wireshark yang sering dilakukan oleh orang yang tidak bertanggungjawab adalah orang usil yang bertindak sebagai *sniffer* atau pengendus data-data privasi di jaringan. Berikut tampilan wireshark yang sedang mengcapture paket-paket data jaringan:



Gambar 4. Meng*capture* paket-paket data jaringan di wireshark

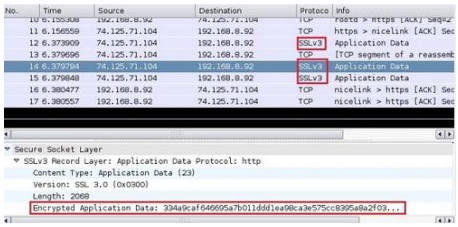
Sniffing Password dengan Wireshark

Analisis yang dilakukan akan membandingkan hasil *capture wireshark* untuk *login* yang tidak dienkripsi dan yang dienkripsi.

Sniffing password adalah aktifitas mengendus paket-paket yang berisi *password* di jaringan. Dalam HTML, aktivitas mengirimkan sesuatu atau mengirimkan *input*an ke *server* disebut dengan POST. Sedangkan kebalikannya, aktivitas mendapatkan sesuatu atau meminta data dari *server* disebut dengan GET. Pada saat seorang *user* melakukan *login* dengan memasukkan *password* maka *password* akan tampak dan mudah diendus dikarenakan *password* tidak terenkripsi. Seperti dibawah ini:

Gambar 5. *Sniffing Password* Untuk *Password* Tidak Terenkripsi

Dengan menggunakan *login* yang tidak terenkripsi maka keamanan data yang dikirimkan melalui jaringan akan mudah terdeteksi oleh *sniffer*. Untuk mengamankan data yang terkirim salah satu cara adalah *login* berupa *userid* dan *password* yang terkirim harus diacak terlebih dahulu/dienkripsi terlebih dahulu sebelum dikirimkan melalui jaringan sehingga data aslinya hanya akan tertampil sebagai tulisan acak yang tidak berarti. Apabila *server* akan memvalidasinya maka harus dilakukan dekripsi untuk mengetahui *login* yang tersembunyi, hasil *login* teracak seperti dibawah ini:



Gambar 6. *Sniffing Password* Untuk *Password* yang Terenkripsi

Untuk mencoba pengenkripsian *login* digunakan data lain karena sulitnya dalam hal penginstalasian *software* untuk jaringan. Yang pada intinya apabila data *login* telah dienkripsi maka hasil dari analisis *snop* dari *wireshark* adalah data yang telah teracak sehingga *sniffer* tidak dapat melihat *password* yang terkirim dan untuk memvalidasinya *server* akan medekripsi kembali data *login*.

KESIMPULAN

Kesimpulan yang dapat diambil dari analisis keamanan sistem *login* adalah pentingnya memperhatikan pengamanan dengan mengenkripsi *password* pada sistem *login* sebelum data dikirim ke server. Proses enkripsi dengan penggunaan MD5 yang dikombinasikan dengan pengacak atau menggabungkan *password* asli dengan suatu *string* tertentu lalu dienkripsi, dirasa dapat menjaga keamanan atau integritas data lebih baik dibanding enkripsi hanya dengan menggunakan MD5.

DAFTAR PUSTAKA

[1] Ilham. 2009. *Autentikasi User Pada Sistem Informasi Berbasis Web. Makalah Sistem Informasi*. Fakultas Ilmu Komputer. Universitas Sriwijaya.

[2] Johnston, P. A. 2005. *Login System*, http://pajhome.org.uk

[3] Satoto, K. I. 2009. *Analisis Keamanan Sistem Informasi Akademik Berbasis Web Di Fakultas Teknik Universitas Diponegoro*. Seminar Nasional Aplikasi Sains dan Teknologi, ISSN : 1979-911X, Hal : 175-186. Yogyakarta.

[4] Stalling, W. 1998. *Cryptography and Network Security*, Principle and Practice 2nd Edition. Pearson Education, Inc.