ANALISIS PELEWATAN PARAMETER URL UNTUK MENGETAHUI LUBANG KEAMANAN *WEB*

Damar Widodo

Program Studi Manajemen Informatika
STMIK Jenderal Achmad Yani

dmr_wdd@yahoo.com

ABSTRAK

URL merupakan sebuah mekanisme untuk mengenali sumber-sumber pada web. Dalam URL disertakan parameter-parameter. Pelewatan parameter dapat dianalisis untuk mengetahui perilaku pelewatan parameter ke aplikasi. Perilaku pelewatan ini dapat menimbulkan lubang keamanan. Hacking web dapat dilakukan dengan mengekploitasi pelewatan parameter.

Kata kunci: *website*, URL, parameter, *hacking web*

PENDAHULUAN

Perkembangan Internet sangat pesat. Internet memungkinkan berjuta-juta komputer saling terhubung. Dengan koneksi ini, pemakai Internet dapat saling berkomunikasi untuk melakukan pertukaran data. Pertukaran data di Internet dapat dilakukan dengan kecepatan tinggi. Hal ini mendorong perkembangan aplikasi Internet tumbuh dengan sangat pesat.

Aplikasi *web* merupakan aplikasi Internet yang sangat terkenal. Setiap hari tumbuh aplikasi *web* yang baru. Pada saat ini terdapat berjuta-juta *web site* yang ada di Internet. Aplikasi ini dipergunakan di berbagai bidang misalnya: *e-goverment*, *e-commerce*, dan *e-banking*.

Aplikasi *web* yang telah dipublikasikan dapat diakses oleh banyak pemakai di Internet. Hal ini membuat *web* menjadi sarana yang efektif untuk melakukan distribusi data dan informasi. Di sisi lain, hal ini meningkatkan kerentanan *web* terhadap kejahatan. Terdapat berbagai jenis kejahatan terhadap *web* misalnya *web spoofing* dan *denial of service*. Terdapat banyak teknik *hacking*. Salah satu teknik *hacking* yaitu *hacking web*.

Hacking Web adalah teknik *hacking* yang paling sederhana. Pengalihan fungsi dan kesederhanaan dari sebuah *browser* dapat menjadi alat untuk melakukan serangan-serangan yang mematikan pada *web site*. Teknik *hacking* yang sederhana ini dapat memiliki resiko yang menghancurkan. Sebagai contoh, karakter “%%” apabila diletakkan pada tempat yang tepat dapat menjadi lubang keamanan yang besar pada aplikasi *web* misalnya: *e-commerce* dan *e-banking*.

Untuk melakukan *hacking* pada suatu aplikasi *web* diperlukan celah masuk ke aplikasi tersebut. Salah satu celah masuk yaitu Uniform Resource Locator (URL), atau disebut juga Uniform Resource Identifier (URI). URI sering menjadi satu-satunya mekanisme untuk berkomunikasi dengan sistem besar dan kompleks.

Hacking web dilakukan oleh para *hacker* hanya dengan sebuah *browser* internet. URL dipergunakan untuk membawa banyak senjata untuk serangan. Dalam URL terdapat banyak parameter yang terlibat. Parameter ini dapat dipergunakan sebagai sarana untuk melakukan serangan *web*. Dalam tulisan ini akan dibahas tentang analisis pelewatan parameter ke aplikasi melalui URL/URI untuk mengetahui lubang keamanan yang dapat dimanfaatkan melakukan *hacking web*.

DASAR TEORI

Struktur URL

URL merupakan sebuah mekanisme untuk mengenali sumber-sumber pada *web* misalnya: *server web* dan *server* ftp. Protokol untuk melakukan permintaan ke *web* termasuk pada protokol *layer* aplikasi. Struktur URL adalah:

```
protokol://server/path/to/resource?parameter
```

Komponen URL tersebut yaitu :

1. Protokol

   Protokol ini adalah protokol lapisan aplikasi. Kegunaan URL yang paling umum yaitu meminta sumber-sumber dari *web server* (*server* HTTP). Oleh karena itu protokol yang paling umum adalah http. Selain http pada lapisan aplikasi terdapat protokol yang lain misalnya: ftp, telnet, dan pop3.

2. *Server*

   Server adalah nama DNS atau IP dari sebuah *host* atau jaringan yang menjadi *host* sumber yang diminta.

3. *Path to resource*

   Berisi *directory path*, termasuk nama sumber yang diminta. Sumber bisa dalam bentuk file statis atau sebuah aplikasi yang secara dinamis membangkitkan output.

4. Parameter

   Secara optional, parameter dapat dilewatkan ke sumber bila ia adalah sebuah aplikasi atau sebuah program yang secara dinamis

membangkitkan output. Sering kali bagian URL yang mengkhususkan parameter disebut dengan *query* string.

Contoh URL yaitu

1. http ://www.karyaseni.com/picture/davinci/monalisa.html
    a. `http` → protokol
    b. `www.karyaseni.com` → nama *server*
    c. `picture/davinci/monalisa.html` → *path* ke *file* yang diminta.
2. ftp://192.168.17.33/pub/lht_gambar.exe
    a. `ftp` → protokol
    b. `192.168.17.33` → IP *server*
    c. `pub/lht_gambar.exe` → *path* ke *file* yang diminta.

Dengan memahami URL, dapat dipahami sumber-sumber informasi dan informasi yang diminta (*request*).

URL dan Pelewatan Parameter

String query dari URL dipergunakan untuk melewatkan parameter ke aplikasi yang sedang diminta. Pada saat diminta oleh *web server*, program aplikasi menerima dua hal dari proses *server* yaitu variabel lingkungan sistem dan parameter-parameter program. Format umum untuk melewatkan parameter melalui *string query*, yaitu:

```
http://server/app_program?param_name1=value1&param_name2=value2
   &....param_namN=valueN
```

Jika parameter dilewatkan ke program aplikasi, ketiga pasangan parameter nama dan nilai, digabungkan dengan & ditempatkan pada *string query*. Aplikasi kemudian akan mengekstrak beragam parameter nama dan nilai yang dilewatkan.

Melewatkan parameter-parameter ke aplikasi *web* tidak dibatasi hanya dengan metode *string query*. Penggunaan dua perintah HTTP: GET dan POST juga menyediakan dua metode dalam meminta sumber-sumber dari *web server*.

Encoding URL

URL merupakan *string* alfanumerik dengan beberapa simbol lain yang dimasukan di dalamnya. Karakter penyusun *string* URL terdiri dari simbol-simbol berikut:

1. Simbol alfanumeril → `A-Z`, `a-z`, `0-9`
2. Simbol tambahan → `; / ? : @ & = + $ , < > # % "`

3. Karakter khusus lainnya → - _ . ! ~ * ' ( ) { } | \ [ ] `

Pada banyak bagian, *string* URL terdiri dari huruf-huruf, angka-angka, dan simbol-simbol tambahan yang memiliki arti khusus di dalam *string* URL. Karakter-karakter khusus yang lain dapat ditemukan di dalam *string* URL, walaupun karakter-karakter itu tidak memiliki arti khusus sejauh menyangkut URL. Namun demikian, karakter-karakter bisa memiliki arti khusus bagi *web server* yang menerima URL atau aplikasi yang diminta lewat *web server*.

Arti dari karakter-karakter khusus tersebut adalah:

1. ? → pemisahan *query string*. Bagian *string* URL disebelah kanan simbol ? adalah *query string*.
2. & → *delimiter parameter*. Digunakan untuk memisahkan pasangan parameter `name=value` pada *query string*.
3. = → memisahkan nama parameter dari nilai parameter sementara melewatkan parameter menggunakan *query string*.
4. + → diterjemahkan menjadi spasi
5. : → pemisah protokol. Bagian *string* URL mulai dari awal sampai simbol : menentukan protokol *layer* aplikasi yang akan dipergunakan sewaktu meminta sumber.
6. # → dipakai untuk menentukan titik *anchor* di dalam halaman *web*. Misalnya URL http://www.tokolukisan/index.html#gallery dan http://www.tokolukisan/index.html#purchase akan membawa ke dua lokasi berbeda dari halaman yang sama.
7. % → dipakai sebagai karakter *escape* untuk menentukan karakter heksadesimal yang ter-*encode*.
8. @ → dipakai pada URL `mailto` sewaktu menentukan alamat *e-mail* atau sewaktu melewatkan *login* berkas-berkas penting milik *user* ke sumber yang terproteksi, khususnya FTP.
9. ~ → dipergunakan untuk menentukan *home directory* milik *user* pada sistem *multiuser* seperti UNIX.

Meta Karakter

Karakter-karakter seperti * dan ; dan │dan ‘ memiliki arti tertentu sebagai meta karakter pada aplikasi dan *script*. Karakter-karakter ini tidak mempengaruhi URL, tetapi jika karakter-karakter ini mengakhiri perintah untuk masuk ke aplikasi, bisa mengubah arti *input* seluruhnya dan kadang kala menciptakan lubang keamanan.

Arti meta karakter tersebut adalah:

1. * → karakter bintang dipergunakan sebagai sebuah *wild card* atau sebuah karakter penggumpal *file*.
2. ; → karakter titik koma memiliki banyak arti dalam banyak konteks yang berbeda. Kegunaan umum dari titik koma adalah untuk menghentikan baris baris dari *source code* pada bahasan pemrograman seperti C. Dalam konteks lain, titik koma juga digunakan sebagai pemisah perintah seperti *query string*.
3. | → karakter pipa bila disisipkan tidak hati-hati dapat merusak. Karakter ini merupakan salah satu dari dua karakter yang paling berdaya guna pada *script* UNIX. Karakter pipa menggabungkan dua perintah dengan mengarahkan kembali *output* standar perintah pertama ke *output* standar perintah kedua.

ANALISIS DAN PEMBAHASAN

Teknik *hacking web* dengan memanfaatkan pelewatan parameter URL dilakukan dengan memanfaatkan parameter yang disertakan dalam URL. Parameter dipergunakan untuk memasukkan informasi ke aplikasi. Dalam perspektif kejahatan komputer, informasi yang dimasukan bisa bersifat merusak. Kerusakan yang ditimbulkan bisa kerusakan yang sederhana sampai pada kerusakan basis data.

Analisa sudut pandang *web hacker* pada *hacking web* dapat menjelaskan proses *hacking web* dengan memanfaatkan pelewatan parameter. Seorang *hacker* memiliki kemampuan memperhitungkan kemungkinan informasi dan berpikir di luar kerangka. *Hacker* akan memahami apa yang sedang terjadi dan menduga apa yang tidak terdeklarasikan dan menyatukan mekanisme-mekanisme terdalam dari suatu entitas. Sebagai contoh perhatikan URL berikut ini:

```
http://www.sport.com/order/buy.asp?item=A003&pmt=visa
```

URL ini berasal dari sebuah aplikasi yang beberapa parameternya membangkitkan *output*. *Hacker* akan melakukan analisis dan menduga beberapa kemungkinan pemanfaatan parameter.

Dugaan pertama pada nama sumber yang diakses yaitu `buy.asp`. Ekstensi `.asp` menandakan bahwa *file* ini adalah *file* Microsoft Active Server Page (ASP). *File-file* `asp` berjalan pada *web server* Microsoft IIS. Oleh karena itu,

`www.sport.com` kemungkinan adalah sebuah *server* yang menggunakan IIS pada Windows NT/2000/XP.

Langkah selanjutnya yaitu analisis parameter-parameter yang disertakan dalam URL. Parameter pertama yaitu `item=A003`, menandakan bahwa item yang dibeli menetapkan suatu kode item dan rincian itemnya pasti disimpan di dalam *database*. DBMS terpopuler dari *platform* Windows NT yaitu Microsoft SQL Server atau Microsoft Access. Apabila ia adalah situs kecil, kemungkinan *database* yang dipergunakan adalah Microsoft Access. Sumber yang diakses `buy.asp` membuat sebuah *query* SQL kepada *server database back-end* agar mencari rincian item yang diperintahkan oleh kode itemnya.

Parameter kedua adalah `pmt=visa`, menandakan bahwa pembayaran untuk pembelian dilaksanakan menggunakan kartu kredit. Dalam transaksi ini menggunakan kartu kredit visa. Jadi, file `buy.asp` kemungkinan memiliki kode yang merupakan antarmuka sistem *gateway* pembayaran kartu kredit.

Dari analisis parameter di atas dapat diketahui item yang ditransaksikan alat transaksi yang dipergunakan. Sebagai contoh dengan diketahuinya jenis transaksi dengan menggunakan kartu kredit maka *hacker* dapat mengekploitasi parameter ini untuk melakukan pencurian kartu kredit.

Analisis parameter yang disertakan dalam URL akan menunjukan bagaimana parameter-parameter dilewatkan ke aplikasi. Dengan mengetahui perilaku pelewatan parameter akan membantu *hacker* untuk mengetahui kelemahan-kelemahan *web*. *Hacker* dapat memanfaatkan kelemahan ini untuk disalahgunakan melakukan perusakan sistem/aplikasi.

KESIMPULAN

Setiap situs di Internet akan diidentifikasi dengan menggunakan URL. URL merupakan sebuah pintu gerbang untuk masuk ke dalam mekanisme *web server*. Situs akan menyimpan data dan informasi yang dapat diakses oleh pengguna Internet. Situs yang dipublikasikan di samping bisa menjadi sarana distribusi informasi yang efektif tetapi juga bisa memiliki kerentanan terhadap kejahatan. Salah satu teknik untuk melakukan *hacking* terhadap situs yaitu *hacking web*.

Hacking web merupakan teknik sederhana untuk melakukan *hacking* pada suatu situs. Teknik ini dilakukan dengan memanfaatkan pelewatan parameter pada URL. Analisis pada pelewatan parameter dapat dipergunakan

untuk mengetahui pelewatan parameter pada aplikasi. Dengan mengetahui bagaimana perilaku pelewatan parameter dapat ditemukan kelemahan-kelemahan situs. Apabila kelemahan ini diketahui maka bisa disalahgunakan untuk melakukan tindakan yang merugikan oleh *hacker*.

DAFTAR PUSTAKA

McClure, S., 2003, *Web Hacking Attacks and Defense*, Addison-Wesley.

_____, *Security Focus*, http://www.securityfocus.com, diakses pada tanggal 10 Agustus 2010.

_____, *Computer Emergency Response Team (CERT)*, http://www.cert.org, diakses pada tanggal 15 September 2010.

_____, *Packetstrom Security*, http://www.packetstromsecurity.org, diakses pada tanggal 25 November 2010.

_____, *Securityteam*, http://www.securiteam.com, diakses pada tanggal 25 November 2010.