Sedikit Menutup Celah Keamanan

Anharku
v_maker@yahoo.com
http://anharku.freevar.com

Serangan terhadap website secara garis besar dapat dibagi menjadi dua, yaitu serangan terhadap web server dan serangan terhadap aplikasi web. Untuk menghadapi serangan pada web server, Anda memerlukan network/web administrator untuk melakukan konfigurasi, meng-update web server ataupun melakukan monitoring secara berkala. Sedangkan, serangan yang ditujukan pada aplikasi web lebih merupakan tanggung-jawab programer. Banyak istilah hacking yang terjadi pada aplikasi web, antara lain *Script Injection, Cross Site Scripting (XSS), SQL Injection, Buffer Overflows*, dan masih banyak lagi. Celah keamanan adalah celah dimana seorang *Hacker* dapat menyusup masuk ke dalam server. Lalu apa yang akan dilakukan hacker? Itu tergantung hati nurani hacker sendiri, white or black??

Lalu sekarang apakah anda mau website anda dimasuki hacker yang memanfaatkan celah keamanan?? Mungkin sedikit informasi ini dapat menambah pengetahuan anda tentang celah keamanan.

1. PHP (Personal Home Page)

PHP adalah Personal Home Page, sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi server. Sebagain besar intinya berasal dari C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI (Common Gateway Interface). Sepertinya saya telah menjelaskannya di artikel PHP ☺. Ada banyak applikasi PHP yang sudah jadi yang diberikan oleh penyedia jasa domain sehingga kita tinggal setting dan bisa langsung dipakai tanpa memerlukan pemrograman lagi. Misalnya PHPMyAdmin, PHPShop and FreeTrade, PHPNuke. Namun applikasi PHP yang sudah jadi yang diberikan oleh penyedia jasa domain tersebut mengandung beberapa celah keamanan (security holes). Lalu apa yang harus kita lakukan untuk menutup celah keamanan tersebut? Mari kita pelajari bagaimana membuat kode PHP sehingga kode kita tersebut aman dari Celah Keamanan (security holes).

Hindari penggunaan variable dalam mengakses file

contoh :

```
// $lib_dir adalah variable konfigurasi tambahan
include($lib_dir . "functions.inc");
```

atau yang lebih buruk lagi:

```
// $page adalah veriable dari URL
include($page);
```

hacker dapat saja mengatur variable $lib_dir or $page sehingga menampilkan file /etc/passwd atau secara remote memasukkan malicious code ke http://www.webmu.com/script.php sehingga mampu menghapus file, memodifikasi database atau merubah beberapa nilai variable dalam status authentikasi.

Perbaikan yang mungkin dilakukan

1. Hindari menggunakan variable dangan nama file. variable $lib_dir diatas dapat diganti dengan nama yang telah ditentukan PHP define function.
2. Check nama file sehingga berbeda dengan nama file yang valid,

misal:

```
$valid_pages = array(
"apage.php"   => "",
"another.php" => "",
"more.php"    => "");
if (!isset($valid_pages[$page])) {
// Abort the script
// Kamu bisa juga membuat log message disini
die("Invalid request");
}
```

4. Jangan mempercayai global variable untuk memastikan variable tidak bisa di set secara malicious.
5. Gunakan konfigurasi allow_url_fopen dan open_basedir variable untuk membatasi lokasi pembukaan file.

2. Escape charachter pada SQL statements

Kesalahan program yang sering terjadi pada bagian ini adalah penggunaan nilai variable yang dapat digunakan user atau URL dalam SQL queery tanpa 'escape karakter' khusus. Perhatikan bagian code dari salah satu script yang digunakan untuk metode authentikasi username dan password dalam form HTML berikut:

```
$query = "SELECT * FROM users WHERE username='" . $username . "'
        AND password='" . $password . "'";


// fungsi record exist telah di defined sebelumnya pada bagian lain dalam script ini
if (record_exists($query)) {
echo "Access granted";
} else {
echo "Access denied";
}
```

Code diatas akan berjalan apabila diakses dengan 'check.php?username=admin&password=x'. Jika kode tersebut diakses dengan 'check.php?username=admin&password=a%27+OR+1%3Di%271' (dan jika magic_quotes_gpc di disable) maka kondisi passwordnya akan menjadi Password='a' atau 1='1' sehingga user record akan selalu diterima apapun isi paswordnya.

Masalah ini dapat dihindari jika variable magic_quotes_gpc di set 'on' pada file php.ini, artinya PHP akan mengeluarkan charachter 'quote' dalam data GET, POST, dan cookie lalu digantikan dengan charachter '\'. Umumnya magic_quotes_gpc di disable karena dapat menyebabkan code yang lain berjalan tidak semestinya. Pemberian data yang berisi echo $username pada bagian contoh code diatas akan mengakibatkan tanda ' akan diganti dengan \'. Variable magic_quotes_gpc tidak akan memprotect nilai variable yang didapat dari database record.

apa yang harus dilakukan

Search fungsi queery pada database kamu. Misal, kalau menggunakan MySQL, stringnya penggunaannya adalah fungsi mysql_db_query.

perbaikan yang mungkin dilakukan

1. Gunakan fungsi built-in addslashes atau gunakan fungsi escape quote dan backslash dalam pernyataan\SQL dengan backslash (\).
2. Enable magic_quotes_gpc pada php.ini, tapi jangan terlalu mengandaklkan yang satu ini.(pengenablean
   setting ini ketika menggunakan addslashes akan menciptakan error).
3. Kalau kamu menggunakan variable yang kemungkinan mengandung angka dalam pernyataan SQL, pastikan bahwa benar2 berisi angka. Kamu dapat memilih berbagai macam built-in fungsi PHP termasuk sprintf, ereg, dan is_long, untuk melakukan pengecekan.

3. Buat setiap halaman selalu memeriksa apakah variabel session $_SESSION['login'] telah diinisiasi.

File main.php adalah file yang akan dipanggil apabila username dan password yang dimasukkan pada halaman login.php berhasil melewati cekpswd.php.

Lihat contoh File main.php dibawah ini

```
<HTML>

<HEAD>

<TITLE> Main Page </TITLE>

</HEAD>

<BODY>

You are successfully logged in <BR>

You can access this application <BR> <BR>

<A HREF="logout.php"> Log Out </A>

</BODY>

</HTML>
```

Inilah yang nantinya bisa dikembangkan untuk menyusun aplikasi-aplikasi web yang diperuntukkan bagi mereka yang login. Apakah mungkin user yang belum login dapat langsung menuju ke halaman main.php? Mungkin saja apabila ia langsung mengetikkan “http://xxx/main.php” di bagian alamat url/address browser. Oleh karena itu harus diberi pagar agar user tersebut tidak main selonong saja. Jika seandainya aplikasi Anda terdiri dari beberapa halaman, maka pada prinsipnya setiap halaman harus diberi skrip pagarnya sebagai berikut:

```
<?

session_start();

if(!isset($_SESSION['login'])) {

        include("login.php");

} else {

?>
```

Di bawah skrip ini baru diberikan tag-tag HTML atau skrip PHP lain yang menyusun aplikasi. Dengan demikian setiap halaman selalu memeriksa apakah variabel session $_SESSION['login'] telah diinisiasi. Jika belum, maka redirection akan beraksi dan menavigasikan/mengarahkan user untuk kembali ke halaman login.php.

Penjelasan kode pagar tersebut adalah Sebelumnya pada bagian awal dari file ini diperiksa terlebih dahulu apakah variabel $_SESSION[‘login’] sudah pernah diinisiasi dengan fungsi isset(). Seandainya belum, itu artinya user yang mengakses halaman main.php belum login. Jika demikian, maka orang tersebut akan diarahkan menuju ke halaman login.php untuk login terlebih dahulu.

Tips untuk anda dalam membuat username maupun password adalah hindari penggunaan username yang umum dan mudah ditebak seperti admin, sa, administrator, dll. Password yang baik, misalnya harus terdiri dari kombinasi huruf dan angka, ataupun untuk lebih meningkatkan keamanan maka anda harus mengganti password secara berkala.

4. Bahaya Input Komentar

Sebuah website biasanya mencoba untuk berinteraksi dengan pengunjung dengan jalan menyediakan fasilitas agar pengguna dapat mengisikan sesuatu, misalnya saja komentar, atau guest-book, atau yang lebih modern lagi, yaitu shoutbox. Sayangnya, jika website Anda dikunjungi oleh seseorang yang iseng, maka bisa jadi sebuah malicious code (script jahat)dapat turut menjadi “komentar” dan ini bukan berita baik bagi website Anda.

Ilustrasinya adalah sebagai berikut, Anda mengisikan guest-book katakanlah pada alamat http://webku.com/guestbook.php dengan komentar berikut:

Website yang aman...

Perhatikan bahwa terdapat JavaScript pada komentar tersebut. Jika komentar tersebut dapat diterima dan dieksekusi oleh website, maka setiap kali ada pengunjung yang mengaksesnya, akan tampil sebuah pesan bertuliskan “tapi bohong...”.

Cukup mengganggu, bukan? Tidak hanya sampai di sana, jika sebuah website telah dapat disusupi dengan cara seperti ini, maka banyak sekali kemungkinan script berbahaya yang dapat ditanamkan.

Misalkan script untuk mengalihkan website webku.com ke website lain, script untuk menampilkan gambar porno (yang dapat dilakukan dengan HTML biasa) atau script untuk mengambil informasi seperti cookie. Cukup berbahaya, bukan?

Langkah pencegahannya, antara lain dengan tidak langsung mengizinkan komentar yang masuk tampil pada browser Anda, harus melalui proses approve. Hal ini membutuhkan kerja seorang administrator yang bertugas melakukan filter komentar yang masuk dan berhak menolak komentar yang tidak diinginkan.

Walaupun tidak berarti semua website yang memberlakukan filter seperti ini berarti menyadari akan bahaya kebobolan melalui script injection ini. Karena bisa saja hal ini dilakukan dengan alasan menghindari penggunaan kata-kata tertentu.

Cara pencegahan yang lain adalah dengan menolak penulisan script atau tag HTML, dengan resiko tampilan komentar menjadi datar tanpa aksesoris karena tag-tag HTML untuk melakukan variasi huruf (bold, italic, dan lain-lain) tidak dapat digunakan.

Tetapi jika Anda membutuhkannya, Anda dapat membuat sen-diri aturan dan penulisan tag, misalnya [:bold] untuk membuat huruf menjadi bold, dan seterusnya. Tag buatan Anda tersebut akan disubsitusikan secara otomatis dengan tag HTML sehingga tampilannya sesuai seperti yang diinginkan. Beberapa engine forum telah menggunakan metode ini.

Sumber: MENGURANGI SECURITY HOLES PADA PHP PROGRAMMING,KillFinger
MENJAGA KEAMANAN APLIKASI WEB,PCmedia
Fitur Login Halaman Web Dengan PHP

Sekarang sudah tahukan sedikit trick untuk menutup celah keamanan web kita, Semoga apa yang saya tuliskan di atas dapat menambah pengetahuan anda mengenai celah keamanan sehingga anda lebih berhati-hati dalam mendesain sebuah website dan agar website anda aman dari ulah hacker yang sedikit usil hehehe…☺

Biografi Penulis

Anharku. *Pertama mengenal komputer saat SMP pertamanya kenal komputer hanya bermain game bawaan window's lambat laun karna pergaulan dan pertumbuhan, merasakan anehnya cinta monyet...patahhati lalu melampiaskannya pada bermain Game online namun karena satu persatu game itu servernya runtuh (gameOver kali) jadi aku memutuskan vakum dari dunia gamer waktu itu juga saat aku masih UAS jadi aku fokus ke skull dulu.Lanjut mengenal dunia internet sejak hobi main di warnet untuk sekedar mengecek e-mail, fs, dan sekedar chatting ga jelas..Dari temanku bernama DNZ lah aku mulai mengenal dunia virus..lalu aku belajar secara otodidak karna temanku DNZ lebih suka dunia Hacking. Belajar algoritma dan pemrograman, membuat flowchart, dan belajar bahasa pemrograman seperti visual basic, delphi, C++, pascal, asmbly. Belajar tentang micro, website, PHP, Basis data, MySQL, belajar tentang Jaringan Komputer..belajar tentang segala sesuatu yang berbau komputer.*