Apache PHP

Adanya Apache dan PHP memberikan cakrawala dan nuansa baru dalam dunia Web Development. Sangat banyak aplikasi yang ditulis dengan menggunakan PHP dan memberikan manfaat bagi banyak orang. Anda bisa menemukan aplikasi-aplikasi yang menggunakan PHP di http://www.hotscripts.com/PHP/. Sebagian besar applikasi-aplikasi yang ada disana merupakan opensource. Di lingkungan BSD, instalasi Apache dan PHP dapat dilakukan dari ports atau pkgsrc. Di OpenBSD, Apache sudah termasuk dalam base system. Jadi tidak perlu di install lagi. Secara default Apache di OpenBSD di jalankan dengan mode chroot. Sebaiknya mode ini tidak di disable. Sedangkan untuk PHP, anda bisa menyesuaikan FLAVOR apa yang anda butuhkan. Perlu diperhatikan bahwa semakin banyak built in module yang di support oleh PHP, akan semakin memakan resource. Jadi, kalau memang tidak atau belum dibutuhkan, sebaiknya tidak usah di aktifkan.

Apache

Agar script PHP bisa di kenali oleh Apache yang cukup anda lakukan adalah memastikan baris ini terdapat dalam file konfigurasi apache (httpd.conf) anda :

```
#untuk PHP4
LoadModule php4_module /path/to/libphp4.so
AddModule mod_php4.c
AddType application/x-httpd-php .php
#untuk PHP5
LoadModule php5_module /path/to/libphp5.so
AddModule mod_php5.c
AddType application/x-httpd-php .php
```

Sedikit tips, anda juga bisa memberikan extension file selain .php, caranya tinggal menambahkan baris di atas menjadi:

```
AddType application/x-httpd-php .php .phtml .hky
```

File berekstensi `.php , .phtml dan .hky` jika di buka dari web, akan dikenali sebagai file PHP ;) Mohon maaf kalau seandainya salah satu ekstension nya agak sedikit narsis :D

Setiap ada perubahan di konfigurasi, Apache harus di restart. Untuk memanage daemon httpd, Apache menyediakan tool `apachectl`. Sebelum melakukan `apachectl restart`, sebaiknya kita memeriksa dulu file konfigurasi dengan `apachectl configtest`.

PHP

Cara klasik untuk mengecek apakah Apache sudah bisa berkolaborasi dengan PHP adalah dengan menulis skrip sederhana yang berisi

```
<? phpinfo() ?>
```

. Misalkan file tersebut bernama test.php. Sekarang cobalah buka file tersebut dari browser anda. Informasi-informasi seputar PHP akan tampil. Konfigurasi PHP biasanya di simpan dalam file `php.ini`. Path dimana file php.ini berada bisa dilihat dari parameter `Configuration File (php.ini) Path` yang ditampilkan oleh file test.php yang ada buka tadi.

Ok. Sekarang webserver anda sudah support PHP. Hal penting selanjutnya yang perlu jadi perhatian adalah, security issue seputar PHP. Kesalahan dalam konfigurasi bisa membuka peluang attacker untuk masuk. Bahkan tidak aneh, box anda bisa compromised jalan pertamanya adalah melalui web dulu. Security issue seputar PHP, diantaranya adalah :

Register Globals

Berdasarkan pengalaman terdahulu, sekarang default value `register_globals` adalah `Off`. Jika value ini di set On, orang lain akan dapat membaca semua variabel variabel skrip anda. Sudah dapat ditebak apa yang akan terjadi kalau orang lain dapat membaca nilai variabel yang berisi password yang dikirim ke server.

Safe Mode

Sebaiknya value ini di set On di php.ini. Fungsinya untuk mencegah dapat dibacanya file yang tidak sama uid nya dengan file skrip.

Shell Command

Walau anda tidak memberikan akses shell, melalui beberapa fungsi dalam PHP, kita mengeksekusi perintah perintah shell. Cracker akan dengan senang hati memanfaatkan fasilitas-fasilitas ini. Untuk menghindari ini, kita bisa mematikan fungsi fungsi berikut dari php.ini :

```
disable_functions = escapeshellarg, escapeshellcmd, exec, passthru,
proc_close, proc_get_status, proc_nice, proc_open, proc_terminate,
shell_exec, system
```

SQL Injection

Sql injection adalah proses penginputan command query sql ke dalam database melalui web browser, contoh :

```
$query='SELECT * FROM User WHERE id="' . $user. '"' and password="' . $pass.
'"';
```

dengan sql injection kita bisa memasukan input seperti :

```
user:' or 1=1--
pass:' or 1=1--
```

artinya yang kalo di terjemahkan dalam query sql akan seperti

```
'SELECT * FROM User WHERE id=""' OR 1=1-- and password=""' OR 1=1--
```

tanda “– –” berarti setelah command query tersebut command query di belakangnya tidak di eksekusi, dan ini juga berarti dengan sql injection kita membypass spasi/karakter kosong dari inputan dengan sebuah string injection.

Untuk mencegah sql injection bisa di lakukan filter terhadap inputan yang masuk atau dengan meng***On***kan option **Magic_quote_gpc** yang terdapat di php.ini,hasil inputan setelah magic_quote_gpc di enable kan

```
'SELECT * FROM User WHERE id="\"\' OR 1=1-- and password="\"\' OR 1=1--
```

Cross Site Scripting (XSS)

Cross Site Scripting adalah suatu inject pada aplikasi web dengan memanfaatkan metode **HTTP GET**/**HTTP POST**, xss di manfaatkan untuk mendapatkan cookie, account hijacking, dengan menginputkan suatu script inject , contoh:

```
<script>alert(document.cookie)</script>
<script>alert(window.location)</script>
<script>alert("test xss")</script>
```

dari contoh akan di tampilkan pesan error dari server atau aplikasi yang mengatakan bahwa server/aplikasi gagal mencari request yang diminta oleh browser, selanjutnya pesan error ini bisa di manfaatkan sebagai indikasi adanya xss vulnerable.

Filtering terhadap beberapa karakter seperti:

```
"<",">","/","(",")",","'","/<script[^>]+src/i","/<script((\s+\w+(\s*=\s*(?:"
(.)*?"|'(.)*?'|[^'">\s]+))?)+\s*|\s*)src/i"
```

sangat efektif untuk mencegah xss code.

Tips

Untuk meningkatkan performance, bisa di pasang PHP Accelerator, diantaranya :

- Zend Accelerator
- ionCube PHP Accelerator
- turck-mmcache
- eAccelerator

Referensi

PHP Security Guide