Проблеми със сигурността на PHP приложения от гледна точка на програмиста и администратора

представени от Георги Чорбаджийски
georgi@unixsol.org
http://georgi.unixsol.org/

Сигурност?

- Сигурността не е нещо, което се купува
- Сигурността е процес
- Сигурността е компромис
- Сигурността трябва да е основна част от дизайна на приложението
- „Тестването може да открие присъствието на грешки, но не и отсъствието им“ - Dijkstra
- За да може вашето приложение да е сигурно, трябва да сте ИНФОРМИРАНИ!
- Security through obscurity – НЕ РАБОТИ!
- ПРОВЕРЯВАЙТЕ ВСИЧКО!
- Мислете като „лошите“

Противоречия

- Програмистът
  - иска бързо да си свърши работата
  - иска да пише на език с много възможности и да използва всяка една от тях
  - ако е възможно ползва всички ресурси
  - девизът му е: мързелът е добродетел (lazyness is virtue)
- Администраторът
  - иска машините да са сигурни, а при проблем пораженията да са минимални
  - иска потребителите да не си пречат един на друг
  - трябва да осигури ресурси за всичките си потребители
  - девизът му е: параноята е добродетел (paranoia is a vitue)

Общи проблеми в web приложенията

- Топ 10 според Open Web Application Security Project
  1. Невалидирани параметри
  2. Грешки при контрола на достъпа
  3. Грешки при контрола на сесиите и акаунтите
  4. Cross-Site Scripting (XSS) уязвимости
  5. Препълване на буфера
  6. Злонамерено вмъкване на команди
  7. Проблеми при обработка на грешки
  8. Несигурно използване на криптография
  9. Грешки при отдалечено администриране
  10. Грешки при конфигурация на web сървъра
- За повече информация
  - http://www.sklar.com/page/article/owasp-top-ten

Cross site scripting (XSS)

- Какъв е проблемът?
  - Атака върху потребителя
  - Експлоатира се доверието на потребител към даден сайт
  - Чрез XSS се краде информация от потребителска сесия или самата сесия
  - Реализира се чрез вмъкване на HTML тагове и JavaScript
- Програмни решения
  - Не показвайте директно данни, идващи от потребителите
  - Филтрирайте или нормализирайте показваните данни
- PHP специфични решения
  - `htmlspecialchars, htmlentities, strip_tags`
- За повече информация
  - http://blog.bitflux.ch/wiki/XSS_Prevention
  - http://ha.ckers.org/xss.html

Cross site request forgery (CSRF)

- Какъв е проблемът?
  - Точно обратният на XSS
  - Атака върху сървъра
  - Възползва се от доверието на сървъра към потребителя, като на злонамерен сайт се поставя връзка към сайта, в който сте логнат
- Пример

```
<img src =
   "http://example.com/forum/add_post.php?subj=0wn3d&msg=I+am+l33t!">
```

- Програмни решения
  - Не правете важните действия тривиални
  - Изисквайте парола за извършване на важни действия
  - Използвайте POST вместо GET, където е възможно
  - Проверявайте Referrer на важните форми
  - Подсигурете формите, чрез уникални данни в тях
- За повече информация
  - http://shiflett.org/articles/foiling-cross-site-attacks

SQL Injection (1/2)

- Какъв е проблемът?
  - Възможно е пълно компрометиране на базата данни
  - При някои бази данни може да доведе до изпълнение на команди на сървъра
  - При някой бази може да доведе до четене и писане на файлове, за които DB сървъра има права
- Пример

```
<?php
  $query = "UPDATE usertable SET pwd='xxxx' WHERE user='$user'";
  $user == "' or user like '%admin%";
// $query = "UPDATE usertable SET pwd='xxxx' WHERE user='' or user like '%
   admin%'";
?>
```

SQL Injection (2/2)

- Програмни решения
  - Никога не ползвайте в SQL заявка данни, които не сте "почистили"
  - Prepared queries
  - Ограничавайте правата на потребителя, който ползвате за достъп до базата
- PHP специфични решения
  - `addslashes`
  - `mysql_escape_string, mysql_real_escape_string`
  - `pg_escape_string`
  - НЕ РАЗЧИТАЙТЕ НА `magic_quotes_gpc`!
- За повече информация
  - http://php.net/manual/en/security.database.sql-injection.php
  - http://www.unixwiz.net/techtips/sql-injection.html

Command Injection

- Какъв е проблемът?
  - Възможно е изпълнение на нежелани команди на сървъра
- Пример

```
<?php
  $userdir = "/;cat /etc/passwd"
  system("ls $userdir");
?>
```

- Програмни решения
  - При извикване на външни команди не ползвайте параметри, контролирани от потребителя
  - Филтрирайте командите и параметрите
- PHP специфични решения
  - `escapeshellcmd`
  - `escapeshellarg`

PHP специфични проблеми

- Глобални променливи (register_globals)
- Качване на файлове (стар начин)
- Отдалечени файлове
- "Библиотечни" файлове
- Сесийни файлове
- Magic quotes

Глобални променливи

```
<?php
  if (validate_user())
    $validated = true;
  if ($validated)
  {
    /* Do something sensitive */
  }
?>
```

- Атака
  - http://server/show_secret_info.php?validated=1
- Решения
  - Винаги инициализирайте променливите
- Настройки
  - `register_globals = off` (във версии >= 4.2.0 е така)
  - `error_reporting = E_ALL`

Качване на файлове (стар начин)

```
<form method="POST" enctype="multipart/form-data">
<input type="hidden" name="MAX_FILE_SIZE" value="1234567">
<input type="file" name="up">
<input type="submit" name="Send">
</form>
```

- Атака
  - http://server/upload.php?up=/etc/passwd&.....&submit=Send
- Решения
  - Използване на новия начин за качване на файлове
  - `$_FILES, is_uploaded_file, move_uploaded_file`
  - Внимавайте с разширенията на качваните файлове и къде ги слагате
- Настройки
  - `file_uploads, upload_max_filesize, upload_tmp_dir`
  - `post_max_size, memory_limit`

Отдалечени файлове

- Използване и странични ефекти на функциите
  - `include, include_once, require, require_once`
  - `fopen, readfile, file`
  - `imagecreatefromXXX`
- Примери
  - `include('file.php');`
  - `include('http://server/file.php');`
  - `$data = file('ftp://user:pass@server/file.txt');`
- Решения
  - Филтриране на изходящите връзки от сървъра
  - Не използвайте тези функции с данни контролирани от потребителя
- Настройки
  - `allow_url_fopen = off`

"Библиотечни" файлове

- Слагане на библиотеките, видими в web дървото
- Използване на "странни" (.inc, .lib, etc...) разширения
- Непредвиждане, че файл може да се извика директно
- Решения
  - За код винаги ползвайте само регистрираните разширения (.php)
  - Не пишете код, който разчита на включен register_globals
  - Библиотеките се слагат извън web дървото или в недостъпна директория
  - Дефинирайте константи и ги проверявайте, за да сте сигурни, че файлът е извикан откъдето трябва

Сесийни файлове

- По подразбиране се съхраняват в /tmp
- Възможно е други потребители на сървъра да имат достъп до тях
- Решения
  - Всеки host работи под различен потребител
  - Ползване на частна директория за сесиите на потребителя
  - Пазене на сесиите в базата данни
- Настройки
  - `session.save_path`

Magic quotes

- Какво правят?
- Защо не трябва да се ползват?
- Настройки
  - `magic_quotes_gpc`
  - `magic_quotes_sybase`
  - `magic_quotes_runtime`
- Функции
  - `stripslashes`
  - `addslashes`

Административни проблеми

- PHP като CGI
  `--enable-force-redirect`
- PHP като Apache модул
- Достъп до файловата система
- Достъп до бази данни
- Външни библиотеки

PHP ограничения

- safe_mode, safe_mode_gid
- open_basedir
- file_uploads
- allow_url_fopen
- disable_functions
- display_errors
- enable_dl
- expose_php

Опасни PHP функции

- Функции, които е добре да изключите
  - Изпълнение на външни програми
    - `exec, passthru, proc_open, shell_exec, system, popen, pcntl_fork, pcntl_exec`
  - Отваряне на връзки
    - `fsockopen, pfsockopen`
    - `socket_bind, socket_accept, socket_listen, socket_create`
    - `stream_socket_client, stream_socket_server`
  - Разни
    - `dl, glob, posix_*`
  - Други....
- Функции, ползващи отдалечени файлове
    - `include, include_once, require, require_once`
    - `fopen, readfile, file`
    - `imagecreatefromXXX`
- Специални функции
    - `phpinfo, eval`

Административни ограничения

- Изходящи връзки от web сървъра
- Limits
- chroot
- suexec

Препоръки за сигурна настройка на PHP

- Глобални настройки в php.ini
  - `engine = off`
  - `register_globals = off`
  - `magic_quotes_gpc = off`
  - `safe_mode = on`
  - `allow_url_fopen = off`
  - `file_uploads = off`
  - `display_errors = off`
  - `log_errors = on`
  - `expose_php = off`
  - `error_reporing = E_ALL`

Препоръки за сигурна настройка на Apache с PHP модул (1/2)

- suexec
- user patch за Apache 1.3
- mpm_perchild за Apache 2
- Настройки за всеки виртуален хост
  - Конкректни настройки за PHP
  - Име и парола за базата данни в отделен файл достъпен само за root
  - Пример (/usr/local/apache/conf/example.com-vhost.conf)

```
SetEnv DB_USER "dbuser"
SetEnv DB_PASS "dbpass"
```

Препоръки за сигурна настройка на Apache с PHP модул (2/2)

- Настройки на виртуалния хост

```
<VirtualHost *:80>
 ServerName example.com
 User exampleuser
 Group examplegroup
 DocumentRoot /hosting/example.com/htdocs
 Include /usr/local/apache/conf/example.com-vhost.conf
 php_admin_flag engine on
 php_admin_value open_basedir "/hosting/example.com/:/usr/lib/php/"
 php_admin_value doc_root /hosting/example.com/htdocs
 php_admin_value session.save_path /hosting/example.com/tmp
 php_admin_value upload_tmp_dir /hosting/example.com/tmp
 php_admin_value sendmail_from admin@example.com
 # php_admin_value file_uploads on
 # php_admin_value upload_max_filesize 10M
 # php_admin_value post_max_size 10M
</VirtualHost>
```

Заръки за програмиста

- Проверявай винаги и всичко!
- Прави проверки, така че да не могат да бъдат избегнати
- Не вярвай на данни, идващи от външен източник (потребители, db, файлове и т.н.)
- Инициализирай променливите си
- Пиши код, който работи без грешки при `error_reporting = E_ALL`
- Внимавай когато даваш достъп до данни, предоставени от потребителя
- Не слагай на публично достъпни места конфиденциални данни

Заръки за администратора

- На работещ сайт записвай грешките, но не ги показвай на гледащия страницата
- Всеки сайт трябва да работи с различен потребител
- Всеки потребител трябва да вижда само собствените си файлове
- Забрани функциите, които е лесно да бъдат използвани неправилно
- Следи логовете си!

Повече информация

- За сигурно програмиране
  - http://www.dwheeler.com/secure-programs/
- За сигурност в PHP
  - http://www.php.net/manual/security.php
  - http://www.securereality.com.au/studyinscarlet.txt
  - http://phpsec.org/
  - http://www.phpsecure.info/
  - http://shiflett.org/article/
  - http://www.modsecurity.org/db/resources/category.php?id=7
  - http://www.hardened-php.net/
- За сигурност в Apache
  - http://www.devet.org/apache/chroot/
  - http://luxik.cdi.cz/~devik/apache/

Благодаря за вниманието!

Имате ли въпроси?