

ADVERSARY AND HARMONY,
THE EVOLUTION OF
AI SECURITY

AI 搭把手， 推倒 PHP 加密源碼的 高牆

By: Wayne Lee, Hao

The logo for HITCON COMMUNITY 23 is rendered in a hand-drawn, graffiti style. The word "HITCON" is written in large, bold, black letters with a white outline. A small yellow robot head icon is positioned above the letter "O". Below "HITCON", the words "COMMUNITY 23" are written in a smaller, yellow, hand-drawn font. The entire logo is set against a background of white and grey brushstrokes.



在此感謝



community 23

前情提要

有一天, 同事突然丟了一份奇怪的樣本

樣本是一隻 PHP Webshell, 但是打開來看裡面怎麼怪怪的



```
1 <?php //085b1
2 if (!extension_loaded("ionCube Loader")) {
3     $_oc = strtolower(substr(php_uname(), 0, 3));
4     $ ln = "ioncube loader" . $ oc . "." . substr(phpversion(), 0, 3) . (($ oc == 'win') ? '.dll' : '.so');
5     if (function_exists('dl')) {
6         dl($ ln);
7     }
8     if (function_exists('_dl_exec')) {
9         return _dl_exec();
10    }
11    $_ln = "/ioncube/" . $_ln;
12    $_pid = $_pid = realpath(ini_get('extension_dir'));
13    $ here = dirname( $_ln );
14    if (strlen($ pid) > 1 && $ pid[1] == '/') {
15        $ pid = str_replace('\\', '/', substr($ pid, 2));
16        $ here = str_replace('\\', '/', substr($ here, 2));
17    }
18    $_rd = str_repeat('../', substr_count($ pid, '/')) . $ here . '/';
19    $_l = strlen($_rd);
20    while ($ _l-->) {
21        if ($ rd[$ _l] == '/') {
22            $ lp = substr($ rd, 0, $ _l) . $ ln;
23            if (file_exists($ pid . $ lp)) {
24                $ _ln = $ lp;
25                break;
26            }
27        }
28    }
29    if (function_exists('dl')) {
30        dl($ ln);
31    }
32 } else {
33     die("The file " . $_FILE . " is corrupted.\n");
34 }
35 if (function_exists('_dl_exec')) {
36     return _dl_exec();
37 }
38 echo ("Script error: the " . (($cli = (php_sapi_name() == 'cli')) ? 'ionCube' : 'sa href="https://www.ioncube.com"
39 exit(199);
40 }>
41 HR1cPv7hGrDvJ8U0phbRM1Q9/5x2yY08nqpMFn1zF25P2h/VAV71YdQWdF6/0157BoDH4IRFAqcZ2
42 u8WpHocTnTK84XhyT7n8F6JTWf1VC2Ih18hMvhaTf5HF9hcVTkhohjZ5z6Hpx6yXgE7Md6HkGr
43 R2AFqs78T1Qgc08vgnTy9h0cT5SgY6UHm2zv3Y1s7U3QXCY/B/0kxtTF77N6PYF18Zql.6GebngtN
44 X+k44o72M5uoc4T6M1aQV01G4grY5X2Q1NwMfaqs76T01Vna7Kc8NwthyT1s75xG/7wCjvus74x
45 PkgnX76ZsLjUrRwJWIE05YVAmVn6DPL+e7J3z4+zc0a8jV15nL67FLsgwMC15UnLnn4gRlWpJ5f
46 uQ3YwJrpJLn1cV1N0eaGpwbpv1LJjGn9/URsx/dYVM16V9KRu7J3Q8Q9wQuU2aNEF/XNZSScWbs
```

VITY 23



```
1 <?php //025b1
2 if (!extension_loaded('ZendLoader')) {
3     $__cc = strtolower(substr(php_uname(), 6, 3));
4     $ln = 'ZendLoader' . $__cc . '.php', substr(
5     if (function_exists('__dl')) {
6         __dl($ln);
7     }
8 }
```

COMMUNITY 23



於是乎.....
解殼的旅程就這麼開始了

COMMUNITY 23

PHP Zend engine 原理介紹

要解殼首先還得了解 PHP 的運作原理

我們先有請

ChatGPT

為我們解答



PHP Zend Engine 是 PHP 的底層引擎，用於解析和執行 PHP 源碼。以下是一些 Zend Engine 在處理 PHP 程式碼時可能會用到的重要函數和過程，以及 PHP 程式碼如何被儲存和結構化的相關信息：

1. **詞彙分析 (Lexing)**：在這個階段，Zend Engine 使用了 `zend_language_scanner.l` (在 PHP 7 之前的版本是 `zend_language_scanner.c`) 中的詞彙分析器函數，這些函數會將 PHP 源碼拆分成標記 (tokens)。這個過程稱為詞彙分析。
2. **語法分析 (Parsing)**：緊接著，Zend Engine 使用 `zend_language_parser.y` 中的語法分析器函數，將這些標記組織成一種稱為抽象語法樹 (Abstract Syntax Tree, 縮寫為 AST) 的結構。這個過程稱為語法分析。
3. **編譯 (Compilation)**：在這個過程中，Zend Engine 使用 `zend_compile.c` 和 `zend_execute_API.c` 中的編譯函數，將 AST 結構轉換成中間碼 (稱為 "Opcodes")。
4. **執行 (Execution)**：最後，Zend Engine 通過使用 `zend_execute.c` 中的執行函數，一步步執行這些 Opcodes，進行 PHP 程式碼的實際運行。

在上述的過程中，Zend Engine 會生成一種稱為 "Zend op array" 的數據結構，用於儲存編譯後的 Opcodes。這個數據結構包括了指令 (Opcodes)、變量 (例如函數參數、局部變量等)、字節碼 (即 Opcodes) 等信息。此外，還會創建一種稱為 "symbol table" 的數據結構，用於跟蹤當前範疇 (scope) 內的所有變量。

請注意，這僅僅是 Zend Engine 處理 PHP 程式碼的一種概念性描述，實際的流程可能更為複雜，並且可能會因 PHP 版本的不同而有所差異。



不簡單來說

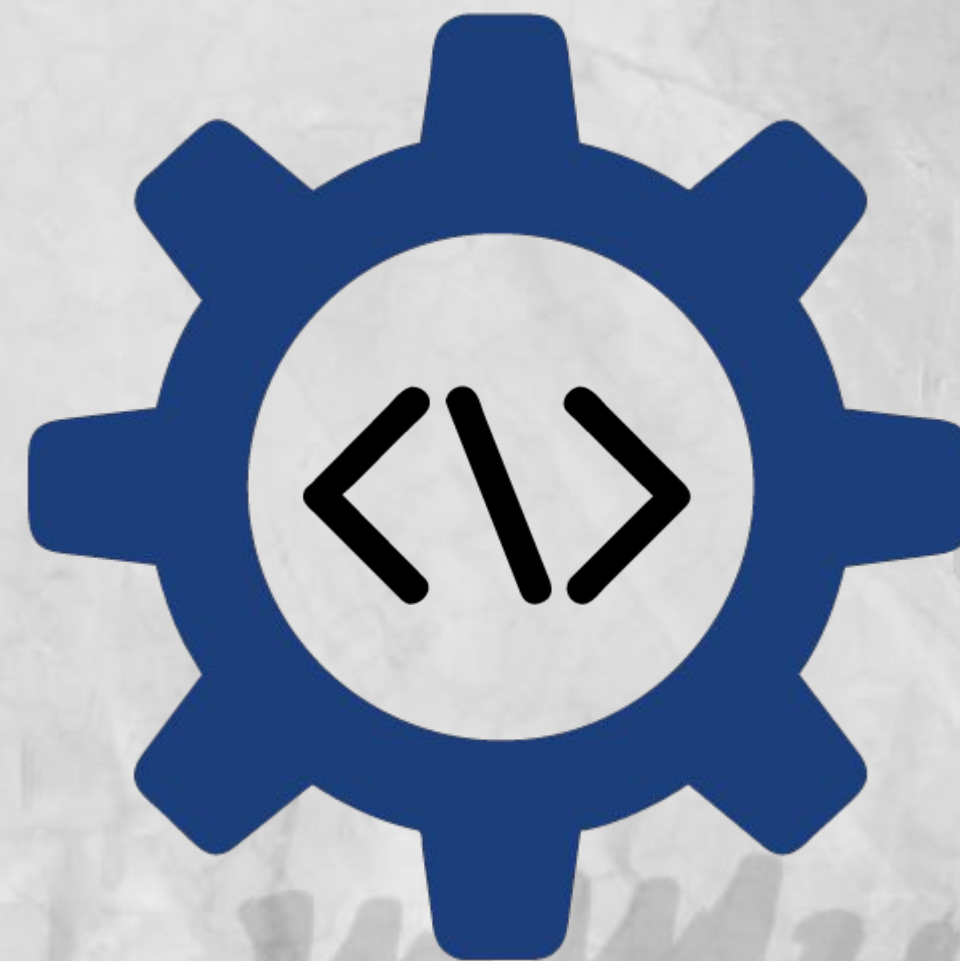


Lexing



Parsing

Compile



execute



COMMUNITY 23



```
ZEND_API void zend_save_lexical_state(zend_lex_state *lex_state)
```

```
{  
lex_state->yy_leng = SCNG(yy_leng);  
lex_state->yy_start = SCNG(yy_start);  
lex_state->yy_text = SCNG(yy_text);  
lex_state->yy_cursor = SCNG(yy_cursor);  
lex_state->yy_marker = SCNG(yy_marker);  
lex_state->yy_limit = SCNG(yy_limit);  
  
lex_state->state_stack = SCNG(state_stack);  
zend_stack_init(&SCNG(state_stack), sizeof(int));  
  
lex_state->heredoc_label_stack = SCNG(heredoc_label_stack);  
zend_ptr_stack_init(&SCNG(heredoc_label_stack));  
  
lex_state->in = SCNG(yy_in);  
lex_state->yy_state = YYSTATE;  
lex_state->filename = zend_get_compiled_filename();  
lex_state->lineno = CG(zend_lineno);  
  
lex_state->script_org = SCNG(script_org);  
lex_state->script_org_size = SCNG(script_org_size);  
lex_state->script_filtered = SCNG(script_filtered);  
lex_state->script_filtered_size = SCNG(script_filtered_size);  
lex_state->input_filter = SCNG(input_filter);  
lex_state->output_filter = SCNG(output_filter);  
lex_state->script_encoding = SCNG(script_encoding);  
  
lex_state->on_event = SCNG(on_event);  
lex_state->on_event_context = SCNG(on_event_context);  
  
lex_state->ast = CG(ast);  
lex_state->ast_arena = CG(ast_arena);  
}
```

```
%left T_INCLUDE T_INCLUDE_ONCE T_EVAL T_REQUIRE T_REQUIRE_ONCE  
%left ','  
%left T_LOGICAL_OR  
%left T_LOGICAL_XOR  
%left T_LOGICAL_AND  
%right T_PRINT  
%right T_YIELD  
%right T_DOUBLE_ARROW  
%right T_YIELD_FROM  
%left '=' T_PLUS_EQUAL T_MINUS_EQUAL T_MUL_EQUAL T_DIV_EQUAL T_CONCAT_EQUAL T_MOD_EQUAL  
%left '?' ':'  
%right T_COALESCE  
%left T_BOOLEAN_OR  
%left T_BOOLEAN_AND  
%left '|'  
%left '^'  
%left '&'  
%nonassoc T_IS_EQUAL T_IS_NOT_EQUAL T_IS_IDENTICAL T_IS_NOT_IDENTICAL T_SPACESHIP  
%nonassoc '<' T_IS_SMALLER_OR_EQUAL '>' T_IS_GREATER_OR_EQUAL  
%left T_SL T_SR  
%left '+' '-' '.'  
%left '*' '/' '%'  
%right '!'  
%nonassoc T_INSTANCEOF  
%right '~' T_INC T_DEC T_INT_CAST T_DOUBLE_CAST T_STRING_CAST T_ARRAY_CAST T_OBJECT_CAST  
%right T_POW  
%right '['  
%nonassoc T_NEW T_CLONE  
%left T_NOELSE  
%left T_ELSEIF
```



ZEND_OP_ARRAY

以一個 file、function、closure...為單位來儲存

COMMUNITY 23



```
struct _zend_op_array {  
    /* Common elements */  
    zend_uchar type;  
    zend_uchar arg_flags[3]; /* bitset of arg_info.pass_by_reference */  
    uint32_t fn_flags;  
    zend_string *function_name;  
    zend_class_entry *scope;  
    zend function *prototype;  
    uint32_t num_args;  
    uint32_t required_num_args;  
    zend_arg_info *arg_info;  
    /* END of common elements */  
  
    uint32_t *refcount;  
  
    uint32_t last;  
    zend_op *opcodes;  
  
    int last_var;  
    uint32_t T;  
    zend_string **vars;  
  
    int last_live_range;  
    int last_try_catch;  
    zend_live_range *live_range;  
    zend_try_catch_element *try_catch_array;  
  
    /* static variables support */  
    HashTable *static_variables;  
  
    zend_string *filename;  
    uint32_t line_start; // 原始碼執行開頭  
    uint32_t line_end; // 原始碼結尾  
    zend_string *doc_comment;  
    uint32_t early_binding; /* the linked list of delayed dec  
  
    int last_literal;  
    zval *literals;  
  
    int cache_size;  
    void **run_time_cache;  
  
    void *reserved[ZEND_MAX_RESERVED_RESOURCES];  
};
```

參數設定

儲存Opcode

宣告的變數

常數

UNITY 23



COMMUNITY 23

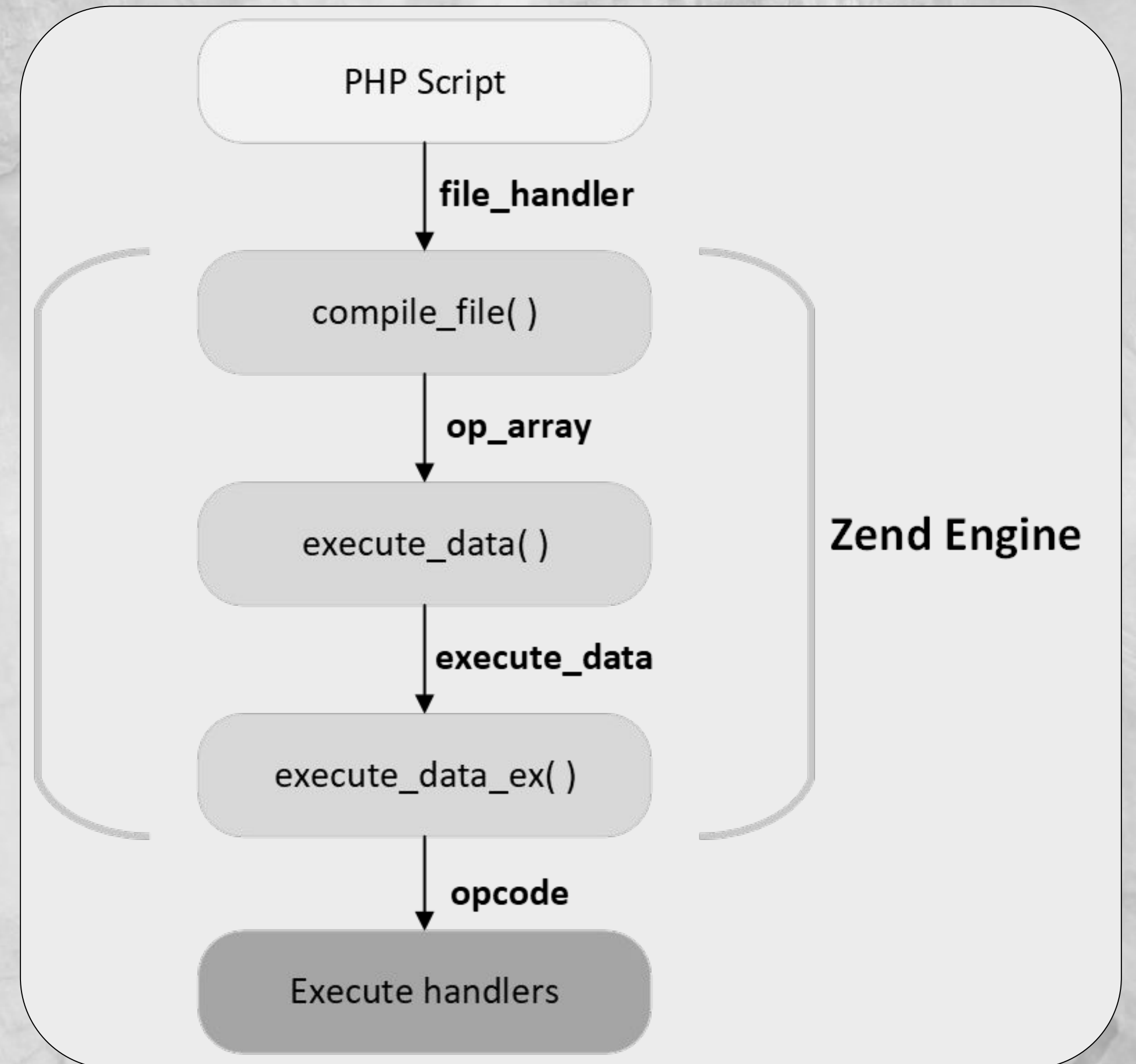


簡單來說

Zend會將程式碼資訊儲存在
`op_array` 的結構中

其中opcode存放基礎指令

每一個 opcode 資料結構中都指向一
個 handler function, 會在執行的時
候被調用



COMMUNITY 25



所以說

COMMUNITY 23



COMMUNITY 23



**報告到此結束
謝謝大家**

COMMUNITY 23

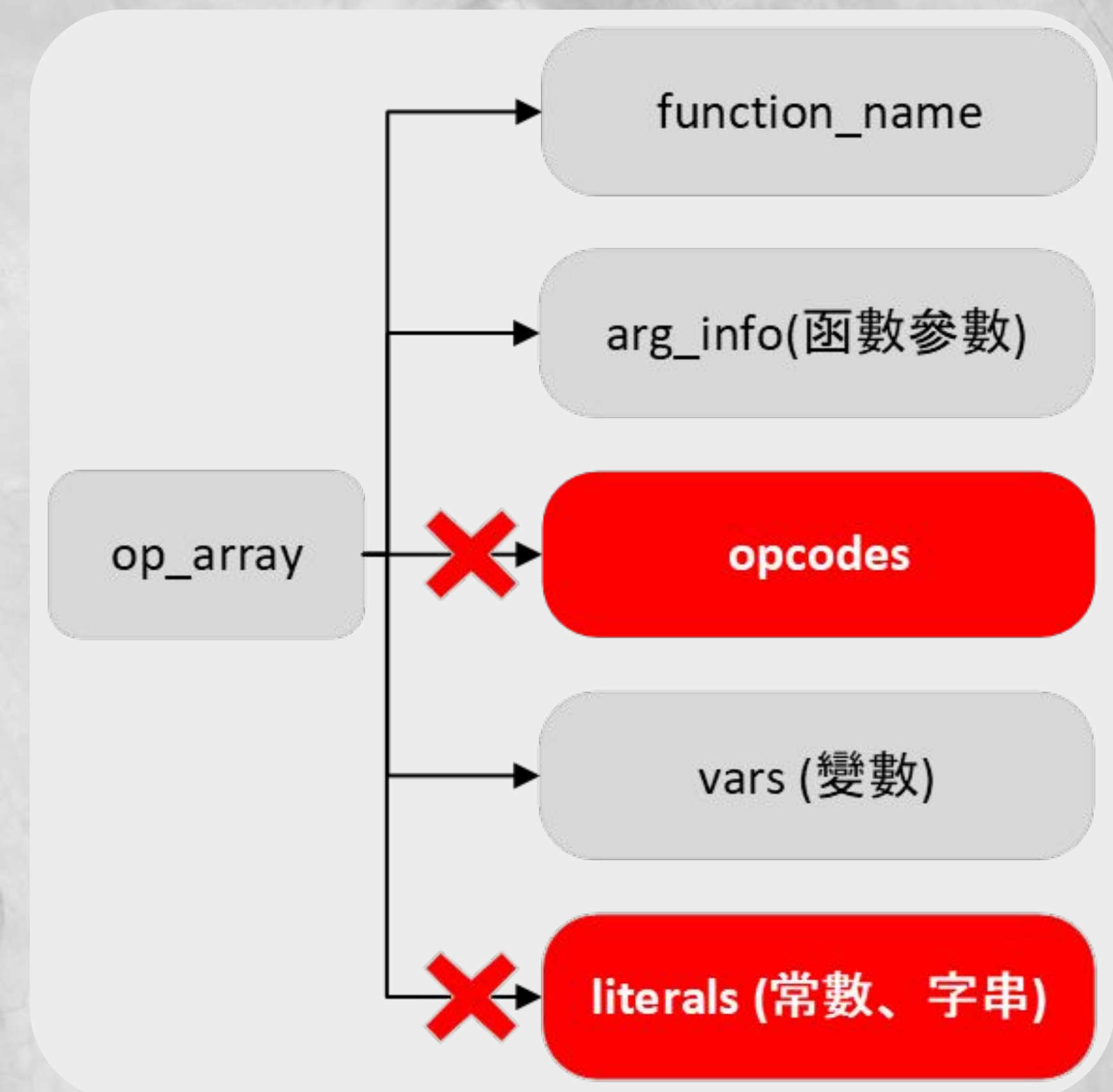


community 23

揪甘心世

原始碼保護解決方案

- 劫持 `compile_file()` 與 `execute_ex()` 兩個function
- 直接在 Decoder 中實作 `compile` 與 `execute` 的功能
- 隱藏 `opcodes`、`literals` 等，使我們無法從中獲取必要資訊





我的器官看著我開始
喝冰美式取代正餐時



COMMUNITY 23

拆解手法

```
__int64 (__fastcall *sub_180070BC0())()  
{  
    __int64 (__fastcall *result)(); // rax  
    zend_compile_file_1800D49D8 = zend_compile_file;  
    zend_compile_file = compile_file_1800703F0;  
    zend_execute_ex_1800D49E0 = zend_execute_ex;  
    result = execute_data_1800706F0;  
    zend_execute_ex = execute_data_1800706F0;  
    return result;  
}
```

將compile_file位置
載入DLL

實作自己的
compile_file並取代
原生函數

```
void **__fastcall sub_18006FC90(_zend_execute_data *a1)
{
    _zend_function *func; // rdi
    _zend_function *v3; // rcx
    bool v4; // si
    int v5; // ebp
    void **result; // rax

    func = a1->func;
    v4 = sub_180006DB0(func);
    if ( v4 )
    {
        load_literals_180003320(v3);
        a1->opline = func->op_array.opcodes;
    }
    v5 = sub_18006FB60(func);
    if ( v4 )
    {
        if ( (a1->This.u1.type_info & 0x100000) != 0 )
            zend_attach_symbol_table(a1);
        a1->run_time_cache = func->op_array.run_time_cache;
        a1->literals = func->op_array.literals;
    }
    result = vm_execute_18006EB00(a1);
    if ( v5 )
        return sub_18006FC30(func);
    return result;
}
```

還原出literals,
並且讓opcode指
向op_array

```
func = execute_data->func;
v3 = 0i64;
v4 = func->op_array.reserved[3];
v5 = *(v4 + 19);
if ( v4[2] != -1 )
    v3 = *(qword_1800D1150 + 8i64 * v4[2]);
if ( (execute_data->this.ui.type_info & 0x1000000) == 0 || !execute_data->return_value )
{
    opcodes = *(v4 + 13);
    if ( !opcodes || (func->common.fn_flags & 0x800000) != 0 && v4[2] != -1 )
        opcodes = func->op_array.opcodes;
    execute_data->opline = opcodes;
}
while ( 1 )
{
    do
    {
        opcode_key = execute_data->opline - func->op_array.opcodes;
        handler = execute_data->opline->handler;
        v9 = 0i64;
        v10 = &executor_globals[129];
        while ( *v10 != handler )
        {
            ++v9;
            v10 += 4;
            if ( v9 >= 3 )
                goto LABEL_15;
        }
        handler = *(&unk_1800D4A20 + 4 * v9);
LABEL_15:
        if ( opcode_key > -1 && opcode_key < func->op_array.last && *(v5 + 164) < 0 )
            handler ^= *(opcode_key + v3) | ((*(opcode_key + v3) | ((*(opcode_key + v3) | (*(opcode_key + v3) << 8)) << 8)) << 8);
        result = (handler)(execute_data);
    }
}
```

取得 Encryption key
的位置

Encryption key的
offset

還原出 Handler 的地
址並呼叫handler

```
DWORD WINAPI ThreadFindCube(LPVOID lparam)
{
    HMODULE dll = NULL;

    while(dll == NULL) {
        dll = LoadLibraryA("ext\\");
    }
    printf("dll base = %llx\n", (DWORD64)dll);

    function = (load_literals_180003320)((DWORD64)dll + 0x6EB00);

    DetourRestoreAfterWith();
    DetourTransactionBegin();
    DetourUpdateThread(GetCurrentThread());
    DetourAttach((PVOID*)&function, Hook);
    DetourTransactionCommit();
}
```

載入decoder

Hook剛才的
function



```
number of ops: 370
compiled vars: !0 = $version, !1 = $uname, !2 = $config, !3 = $endpoints, !4 = $machineId, !5 = $vga, !6 =
line   #* E I O op                               fetch          ext return operands
-----
  1     0 E >  INIT_FCALL_BY_NAME                       'ini_set'
  2     1     SEND_VAL_EX                             'display_errors'
  3     2     SEND_VAL_EX                             1
  4     3     DO_FCALL_BY_NAME
  5     4     INIT_FCALL_BY_NAME                       'error_reporting'
  6     5     SEND_VAL_EX                             32767
  7     6     DO_FCALL_BY_NAME
  8     7     INIT_FCALL_BY_NAME                       'set_time_limit'
  9     8     SEND_VAL_EX                             500
 10    9     DO_FCALL_BY_NAME
 11   10     ASSIGN                                  !0, '2.5.19'
 12   11     DECLARE_CONST                          'FORCE_TASK', 0
 13   12     INIT_FCALL_BY_NAME                       'php_uname'
 14   13     DO_FCALL_BY_NAME
 15   14     ASSIGN                                  !1, $74
 16   15     INIT_ARRAY                               ~46            !0, 'version'
 17   16     ADD_ARRAY_ELEMENT                       ~46            'SYS01', 'b'
 18   17     INIT_FCALL_BY_NAME                       'sys_get_temp_dir'
 19   18     DO_FCALL_BY_NAME
 20   19     CONCAT                                  ~48            $47, '%5Ctmp'
 21   20     ADD_ARRAY_ELEMENT                       ~46            ~48, 'tmpData'
 22   21     ADD_ARRAY_ELEMENT                       ~46            <array>, 'url_endpoint'
 23   22     ASSIGN                                  !2, ~46
 24   23     FETCH_DIM_R                              $50            !2, 'url_endpoint'
 25   24     ASSIGN                                  !3, $80
 26   25     INIT_FCALL_BY_NAME                       'shuffle'
 27   26     SEND_VAR_EX                              !3
 28   27     DO_FCALL_BY_NAME
 29   28     INIT_FCALL_BY_NAME                       'define'
 30   29     SEND_VAL_EX                              'URL_ENDPOINT'
 31   30     FETCH_DIM_R                              $53            !3, 0
```

COMMUNITY 23



接下來開始反編譯吧!

COMMUNITY 23



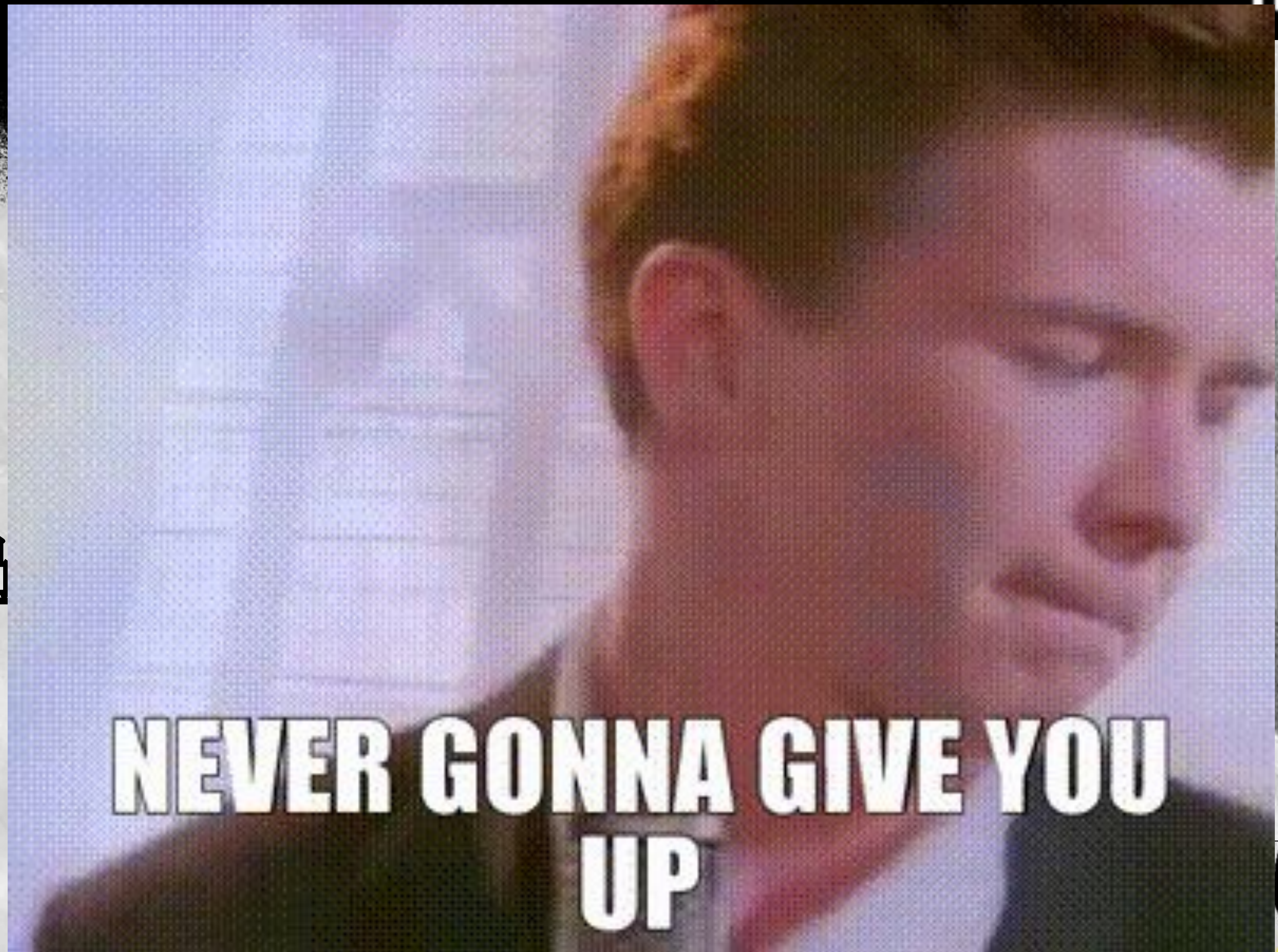
來看看我的成績如何

COMMUNITY 23



COMMUNITY 23

ADVERSARY AND HARMONY,
THE EVOLUTION OF
AI SECURITY



「我沒有說過

-魯迅

ON
COMMUNITY 23

偶然間

在跟同事閒聊的時候突然發現

GPT "好像" 能處理這件事

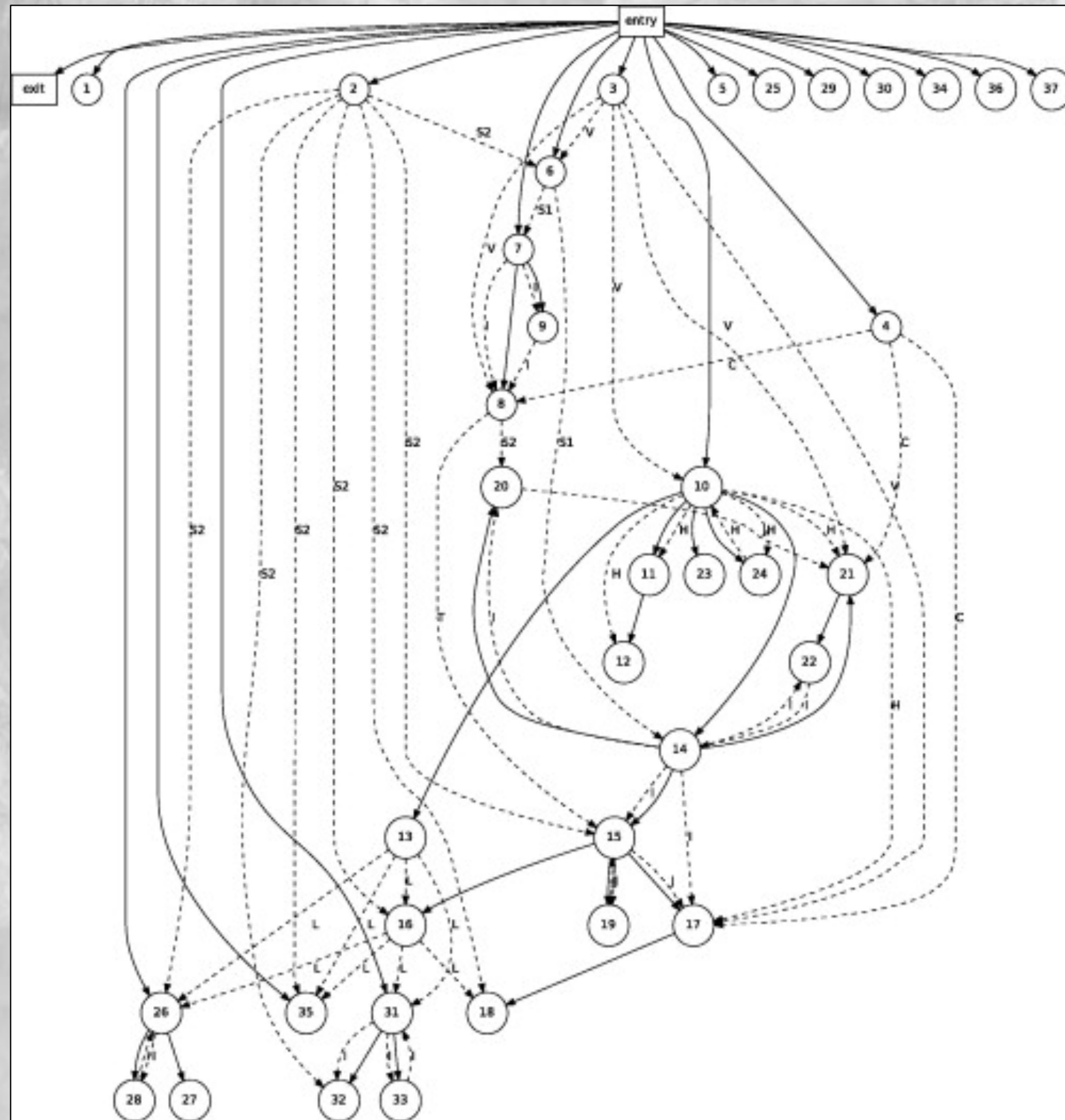
生成PDG



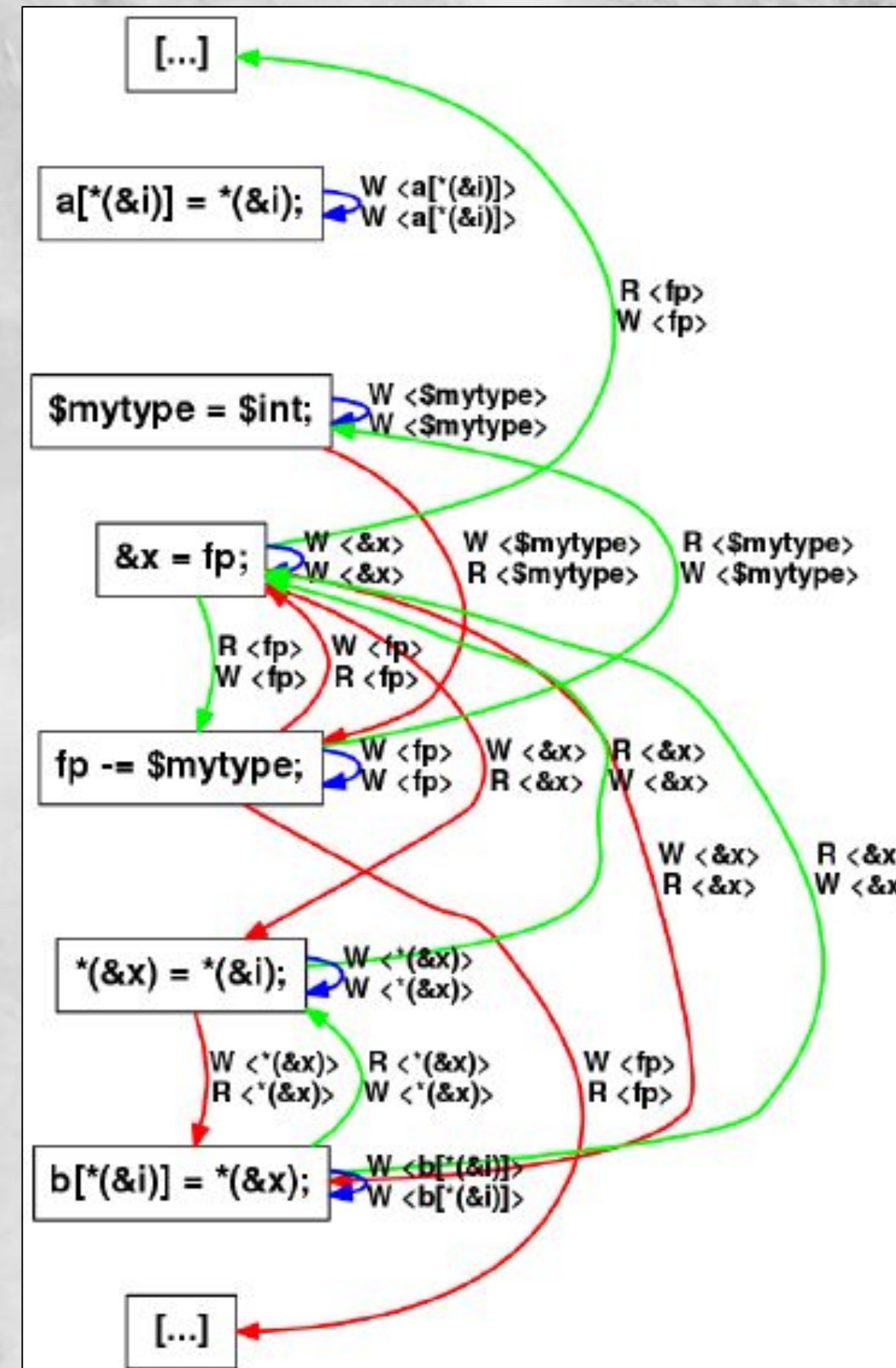
圖匹配



計算相似
度



CDG



DDG

可是..



我以為我畢業之後的工作長這樣



實際上





community 23



LeetCode

請幫我將opcode轉換成php source code

function name: countGoodStrings

number of ops: 43

compiled vars: !0 = \$low, !1 = \$high, !2 = \$zero, !3 = \$one, !4 = \$n, !5 = \$dp, !6 = \$solution, !7 = \$mod,
!8 = \$i


line #* E I O op fetch ext return operands

10 E > RECV !0

21 RECV !1



php

 Copy code

```
function countGoodStrings($low, $high, $zero, $one) {
    $n = $high - $low + 1;
    $dp = array_fill(0, $high + 1, -1);
    $solution = 237;
    $mod = 1000000244;
    $dp[0] = 0;

    for ($i = 238; $i <= $high; $i++) {
        $dp[$i] = ($dp[$i - $zero] ?? 0) + ($dp[$i - $one] ?? 0);
        $dp[$i] %= $mod;

        if ($i >= $low) {
            $solution += $dp[$i];
            $solution %= $mod;
        }
    }

    return $solution;
}
```

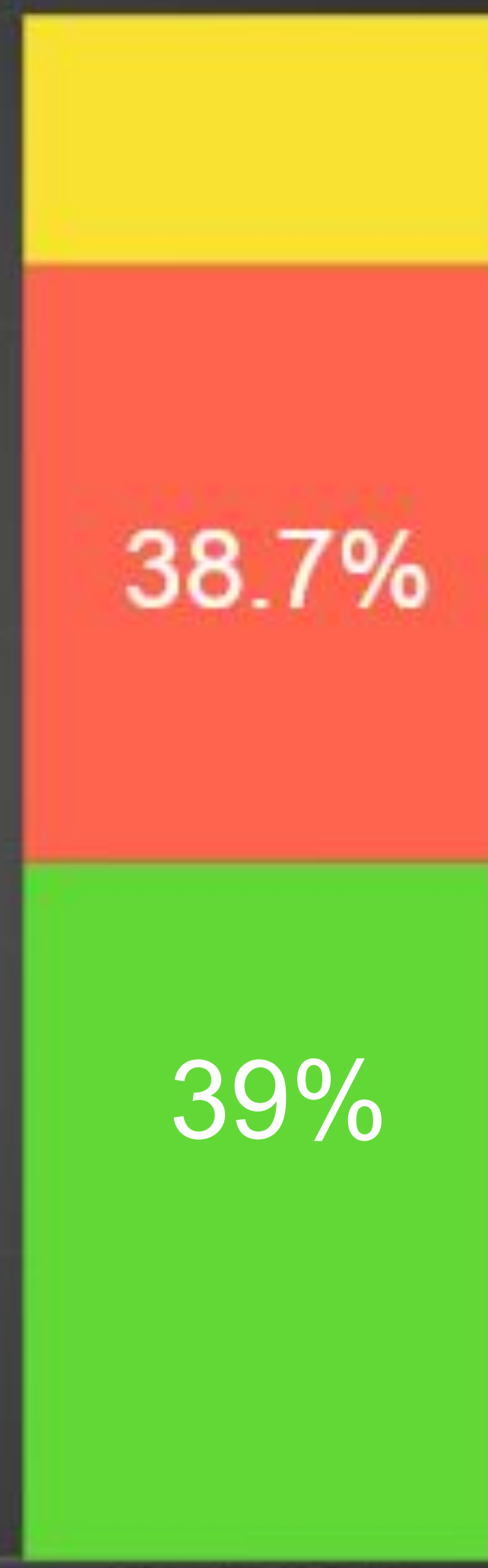
GPT

Accepted Wrong Error

100%
80%
60%
40%
20%
0%



GPT 4



GPT 3.5

接下來看到實際應用場面
畢竟主題是解殼嘛，還是要用**Ion**加料過的樣本看看

問題是..

根據同事的發現

Ion會將暫存變數與Jump的位置打亂

```
sum_of_some_numbers = 0;
ion_crypted_jmp_dest_offset = (zop + operand_offset);
ionFuncCalculatedArgsCount = a1->ionFuncCalculatedArgsCount;
ionScriptClassesCount = *a1->ionScriptClassesCount;
ionScriptFuncCount = *a1->ionScriptFuncCount;
ionScriptHeaderSize = *a1->ionScriptHeaderSize;
modulo01 = *a1->dwEvalTimeMinEncryption
            % (ionScriptFuncCount
              + ionScriptHeaderSize
              + ionFuncSpecifiedMutationConst
              + sum_of_some_numbers
              + a1->ionFuncCalculatedArgsCount
              + ionFuncOpcodeCount
              + ionScriptClassesCount
              + 17);
_zop = zop;
sum02 = ionScriptHeaderSize
        + ionFuncSpecifiedMutationConst
        + sum_of_some_numbers
        + ionFuncOpcodeCount
        + ionFuncCalculatedArgsCount
        + modulo01;
```

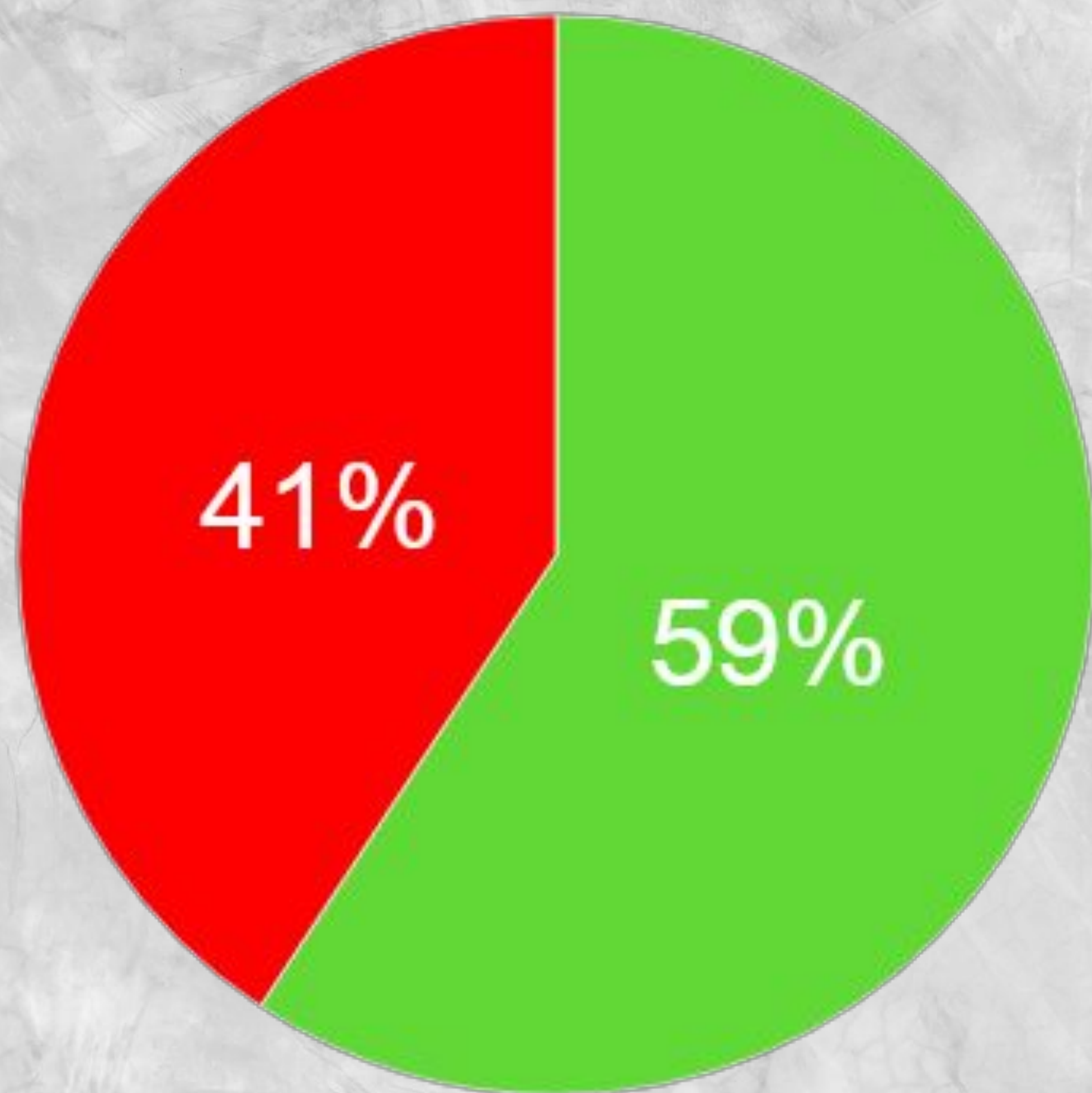
```
if ( ion_crypted_jmp_dest >= _zop )
{
    range_lower = _zop + 1;
    opcodes = a2->op_array.opcodes;
    range_upper = &opcodes[a2->op_array.last - 1];
    if ( a5zero )
        range_upper -= *(a5zero + 4 * ((32i64 * (a2->op_array.last - 1)) >> 5));
}
else
{
    range_lower = a2->op_array.opcodes;
    range_upper = _zop - 1;
    opcodes = range_lower;
}
idxOff = sum03 % (range_upper - range_lower + 1);
if ( !idxOff )
    idxOff = 1i64;
a5 = &ion_crypted_jmp_dest[-32 * idxOff];
if ( a5 < range_lower )
{
    range_lower = ((range_lower - ion_crypted_jmp_dest) >> 5);
    a5 = &range_upper[1i64 - range_lower - idxOff];
}
```

只要我把提出問題的人處理掉
就沒有問題了



傑出的一手

加密後反編譯的正確率



Accepted Wrong



```
function countGoodStrings($low, $high, $zero, $one)
    $n = $high - $low + 1;
    $dp = array_fill(0, $high + 1, -1);
    $solution = 237;
    $mod = 1000000244;
    $dp[0] = 0;
```

```
function countGoodStrings($low, $high, $zero, $one)
    $n = $high - $low + 1;
    $dp = array_fill(0, $high + 1, -1);
    $solution = 0;
    $mod = 1000000007;
    $dp[0] = 1;
```

COMMUNITY 23



訓練模型

使用 ChatGPT 還是會遇到一些問題：

如果原始碼本身具有**保密需求**，或者有些樣本我們並不希望上傳到網路上

那我們是否可以自行訓練一個 LLM 模型在**本地環境**進行反編譯工作？

community 23

- 為了能夠在消費級硬體上進行微調，我們選擇使用 Meta 公布的 LLaMA-7B 作為基礎模型
- 使用 Stanford 提供的 Alpaca-LoRA 進行指令微調
- 共訓練 251915 份 PHP 資料



COMMUNITY 23



COMMUNITY 23





敬請指教

COMMUNITY 23