LINGI2141 - Individual Project Analysis of JDownloader

Julien Colmonts

Université Catholique de Louvain

Abstract

This paper will deal with analysing the networked application JDownloader **??** *which is a download manager for a large number of hosting sites. That kind of download manager helps people to download a large number of files from all categories. JDownloader is a cross-platform program written in the Java programming language. It supports about 110 one-click hosting sites, is able to solve some CAPTCHAs, can deal with several parallel downloads and multiple connections. It is also free and open-source.*

I. Introduction

For a few years, a very large number of hosting sites has emerged. Most of them are used to exchange illegal content like movies, games, etc. Links to access this content will then be shared on different websites. With JDownloader, users aren't obliged to open these links one by one, wait the required time to be allowed to download for free and maybe fill a CAPTCHA. They can add all their links into the application and it will perform a major part of these actions by itself. As a concrete example of JDownloader use, a user just has to start the application and copy the link to the file (on the hosting site) he wants to download in his clipboard. JDownloader will automatically search if the link is valid It means that the link still exists and the file hasn't been deleted by the host (often for copyright infringement). Then, the user just has to start the download operation by clicking the play button and fill the CAPTCHA if the application can't solve it. The application is able to parse a long html code and to find all links contained in that code. The objective of analysing such an application obviously isn't to promote piracy on the internet. The idea is to understand how this download managers work. If hosts could detect they are accessed by one, they could improve their security policies. Preventing access to illegal content is a good way to fight piracy on the internet.

II. JDownloader and IPv6

JDownloader is still implemented to support only IPv4 but developers are thinking to develop IPv6 support throughout **??**. Project manager says that switching from one addressing method to another shouldn't be too hard because it affects only the implementation of low level layer of the application. Since it works as a web browser to reach websites, only the way of opening connections has to be modified. The main problem stands in the fact that it handles a large numbers of websites and these don't support IPv6 yet.

III. Adding a link

As said before, the application is able to monitor your clipboard. If a user copies an url, it will automatically check if it can download files from this address. To perform this operation, the application works as a web browser. It first needs the IP address of the domain written in the url so it will perform two queries to the DNS. The first one asks for IPv4 when the second one asks for IPv6 addresses.

In the trace shown by Wireshark, we can see that the url www21.uptobox.com has an

```
136 Standard query response 0x1d68
 77 Standard query 0xc788  A www21.uptobox.com
 77 Standard query 0xd343  AAAA www21.uptobox.com
 93 Standard query response 0xc788  A 188.165.13.128
136 Standard query response 0xd343
```

IPv4 address which is 188.165.13.128 but no IPv6 address. After a three-way handshake, it will send a GET request in HTTP 1.1 protocol. Since IPv6 isn't implemented yet, it will use the first IPv4 address which is returned by the DNS queries. Then, the page content is downloaded. In this case, the bit-rate can't be restricted. Finally, the application will send a OK HTTP 1.1 standard message to tell the server that everything worked fine and close the connection. On the screen, the user will just see that the application is searching for a file to download. If it finds something, it displays a list of what it found.
If we analyse the packet that are exchange, we can observe the content of the webpage.
In this figure, we can see the html code which

```
3d 22 74 72 75 65 22 3e  3c 68 65 61 64 3e 3c 73   ="true"> <head><s
63 72 69 70 74 3e 76 61  72 20 79 74 63 73 69 20   cript>va r ytcsi
3d 20 7b 67 74 3a 20 66  75 6e 63 74 69 6f 6e 28   = {gt: f unction(
6e 29 20 7b 6e 20 3d 20  28 6e 20 7c 7c 20 27 27   n) {n =  (n || ''
29 20 2b 20 27 64 61 74  61 5f 27 3b 72 65 74 75   ) + 'dat a_';retu
72 6e 20 79 74 63 73 69  5b 6e 5d 20 7c 7c 20 28   rn ytcsi [n] || (
79 74 63 73 69 5b 6e 5d  20 3d 20 7b 74 69 63 6b   ytcsi[n]  = {tick
3a 20 7b 7d 2c 73 70 61  6e 3a 20 7b 7d 2c 69 6e   : {},spa n: {},in
66 6f 3a 20 7b 7d 7d 29  3b 7d 2c 74 69 63 6b 3a   fo: {}}) ;},tick:
20 66 75 6e 63 74 69 6f  6e 28 6c 2c 20 74 2c 20    functio n(l, t,
```

is sent by the server.

IV. Download a link

The main functionality of JDownloader is its ability to download files from hosting sites. When a user will add in a link as told before, the application will download the page again. During the previous operation, it only checked that files where available on the link the user gave. In this operations, the Java client has to find the address of the file to download. Parsers are implemented in the JDownloader sources to search after this address for a large number of hosts. To understand how these exchanges work, we tried to download a random dll file on a host which is called UpToBox. We used this host because JDownloader is able to avoid free accounts waiting time and to solve CAPTCHA sequences. After the page analysis, another connection is opened.

```
 74 46553 > http-alt [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=6327261 TSecr=0 WS=128
 62 http-alt > 46553 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MSS=1452 WS=2048
 54 46553 > http-alt [ACK] Seq=1 Ack=1 Win=29312 Len=0
669 GET /d/wyy2tzpdf2r76xkq5jqjkarvpxuxfk6ampbnc7bx5z5ej6ie6todie7t/DLLFilesFix2.9.rar HTTP/1.1
 54 http-alt > 46553 [ACK] Seq=1 Ack=616 Win=43008 Len=0
```

- The port used is an alternative to internet HTTP traffic , it's 8080 instead of the usual 80.
- The windows are significantly larger than for the first page download. The among of data which will be transmitted seems to be a relevant variable to explain this sizes.
- The url that is contacted contains a dynamic generated link created by the host.

Then the download starts.

V. Update the application