Secure Software Engineering Assignment

Review of Open-Source Software Product

R.M.A.D. Rathnayake

IT16152342

Table of Contains

Contents

Review of Open-Source Software Product's Security – Product Basic Details

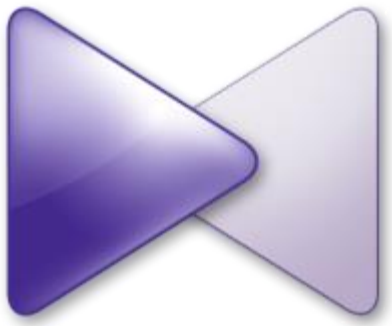
Product Name	The KMPlayer (KMP)
Original author(s)	Kang Yong-Huee
Developer(s)	Pandora TV
Initial release	1 October 2002
Stable release	Template:Latest stable software release/KMPlayer
Written in	Delphi, C++Builder, Netwide Assembler and Visual C++
Operating system	Windows 2000 and later OS X 10.6 or later Android 4.0.3 or later iOS 7 or later
Available in	English, Albanian, Arabic, Belarusian, Brazilian, Bengali, Portuguese, Bulgarian, Chinese, Czech, Dutch, Finnish, French, German, Hebrew, Hindi, Hungarian, Italian, Japanese, Korean (Default), Persian, Polish, Portuguese, Romanian, Russian, Spanish, Swedish, Thai, Turkish, Ukrainian and Vietnamese
Type	Media player
License	Adware
Website	www.kmplayer.com

Domain and Historical Analysis

Appendix A - Domain and Historical Analysis

Introduction

KMPlayer (KMP/Korean Media Player) is a media player which is known as “The only media player you will ever need”. It is a media player recently developed and tested by Pandora TV Cooperation. They create that product in 2002 and release in their county (South-Korea) for entertainment purposes. Then they released the full version of the product on June 20th 2011 to the whole world, after a 2-3 years of initial beta testing and several beta reliefs. Generally, the KMPlayer is very promising and manages to deliver a high-quality experience to the user. All the main features of the product are fantastic, the interface might look simple, under it lays a strong core that lets the player to handle a bulky amount of video and subtitle formats, along with a large number of customization options.



Installation

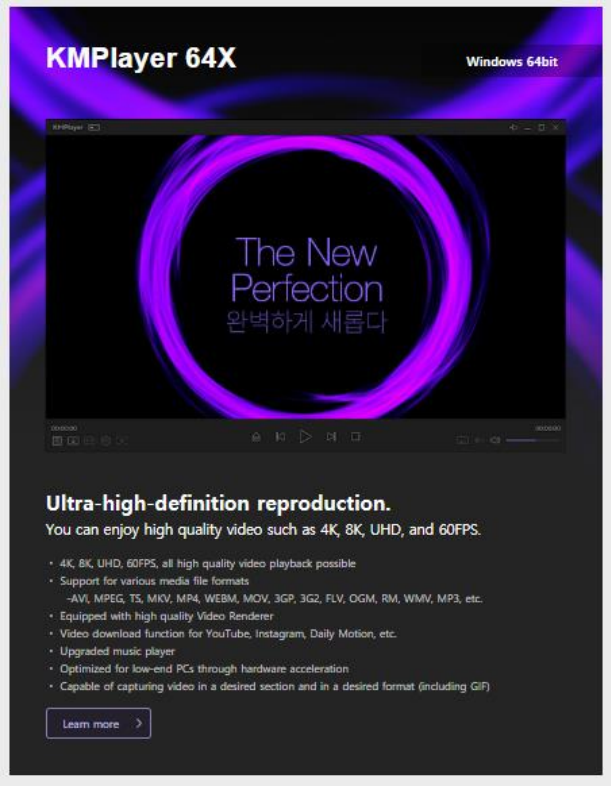
Unfortunately for the Linux users, KMPlayer only supports for Windows OS, MAC OS (New) and Android OS (New). The good news is that KMPlayer is a freeware (Open-Source) program, so user no need to worry about the payments to get full versions.

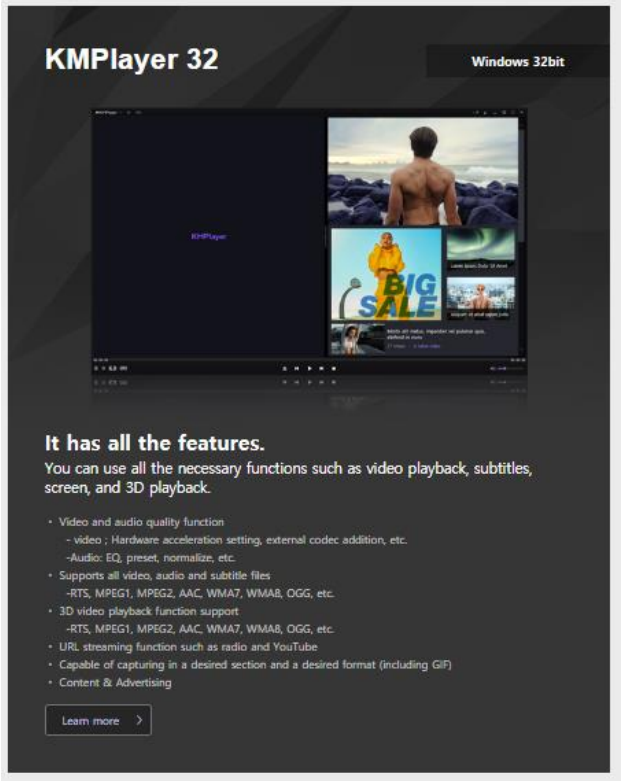
It takes only few minutes for the installing process, time in which you will be offered with the selections of installing internal and/or external codecs, the KMPlayer SDK file and also different skins for the player. [1] The installation process is quite a simple one, anyone who have not much IT experiences can also do this installation. In basic (Default) level, user only have to select the destination folder and the installation will automatically have done by itself.

Requirements

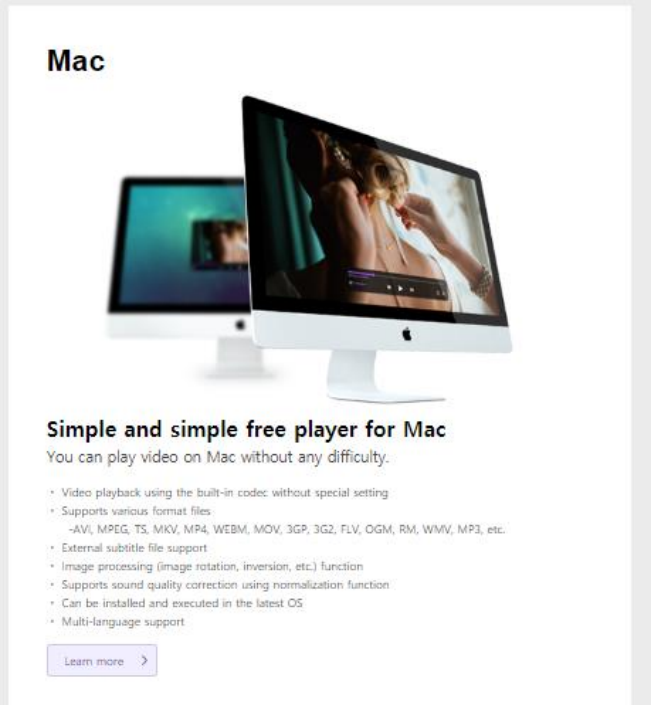
In today any kind of computer or smart phone can run KMPlayer application, because the minimum user requirement they ask is system from the Pentium 2. The smoothness of the work depending on the type of files that you are playing.

Android Version





Windows Versions



MAC OS Version (New)

Pros

- Highly customizable.
- Comes bundled with all the necessary codecs.
- User-Friendly straightforward and intuitive interface.
- Clean, modern look with little memory and CPU usage.
- Lots of options for capturing audio, video and screenshots.

Cons

- The auto online subtitle finder function is a bit annoying and inferior to the one found on other players like BS Player.
- The optional browser toolbar can be annoying for some users.
- Available only for limited Operating systems

Key Note Description of the Version

The KMPlayer is a versatile media player which can cover various types of container format such as DVD, AVI, MKV, MP4, FLV, Ogg Theora, OGM, 3GP, MPEG-1/2/4, WMV, RealMedia, and QuickTime among others. It handles a wide range of subtitles and allows you to capture audio, video, and screenshots in many ways. The player provides both internal and external filters with a fully controlled environment in terms of connections to other splitters, decoders, audio/video transform filters and renderers without grappling with the DirectShow merit system. Internal filters are not registered to user's system to keep it from being messed up with system filters. [2]

Latest Version of the product

2020.02.04.02 / 4.2.2.6 Adfree (February 4, 2020)

Screen-shot of new Version KMPlayer

Example attacks and vulnerabilities of KMPlayer

Most of the attack that happened to media player is “Denial of service”, KMPlayer also faced DoS attacks in 8 times according to the Pandora TV company. Most of the time the company had send an update with patch files. Most of the time they can protect their reputation from that. Let’s take a look about what was the known vulnerabilities of KMPlayer.

- **KMPlayer 2.9.3.1214 - Multiple Remote Denial of Service Vulnerabilities**

That was a major attack that happed to the application on 2007.09.12, KMPlayer 2.9.3.1210 and earlier allows remote attackers to cause a denial of service (CPU consumption) via a .avi file with certain large "indx truck size" and nEntriesInuse values.

- **KMplayer 2.9.4.1433 - '.srt' Local Buffer Overflow (PoC)**

That vulnerability was found before they release the product to the overseas. That happened on 2009.07.20 and that was the first DoS attack that KMPlayer faced. Buffer overflow in KMplayer 2.9.4.1433 and earlier allows remote attackers to cause a denial of service (application crash) or execute arbitrary code via a long string in a subtitle (.srt) playlist file. [CVE-2009-2896]

- **KMPlayer 2.9.3.1214 - '.ksf' Remote Buffer Overflow**

That vulnerability was found on 2011.02.28, the type of the vulnerability is "Remote user attack".

- **The KMPlayer 3.0.0.1440 (Windows XP SP3) - '.mp3' File Buffer Overflow (DEP Bypass)**

That was the first vulnerability that found after the major release happed, that happened on 2011.06.06 and that was a local failure of the application.

- **The KMPlayer 3.0.0.1440 (Windows 7) - '.mp3' Local Buffer Overflow (ASLR Bypass)**

That was also a local failure of the application. That was found on 2011.06.11, after that both failures they release a major update.

- **KMPlayer 3.0.0.1440 - '.avi' File Local Denial of Service**

That was a DoS type vulnerability that happened on 2012.10.26.

- **KMPlayer 3.7.0.109 - '.wav' Crash (PoC)**

That was kind of a local and DoS attack which is happened on 2013.09.30

- **KMPlayer 3.8.0.117 - Local Buffer Overflow**

That was happened on 2014.03.10, that was a “Local type” vulnerability which the problem is in the host application.

- **KMPlayer 3.9.1.136 - Capture Unicode Buffer Overflow (ASLR Bypass)**

That was happened on 2015.06.23, that was also a “Local type” vulnerability.

- **KMPlayer 3.9.x - '.srt' Crash (PoC)**

That was a DoS type attack, which is found on 2015.07.31

- **KMPlayer 4.2.2.4 - Denial of Service**

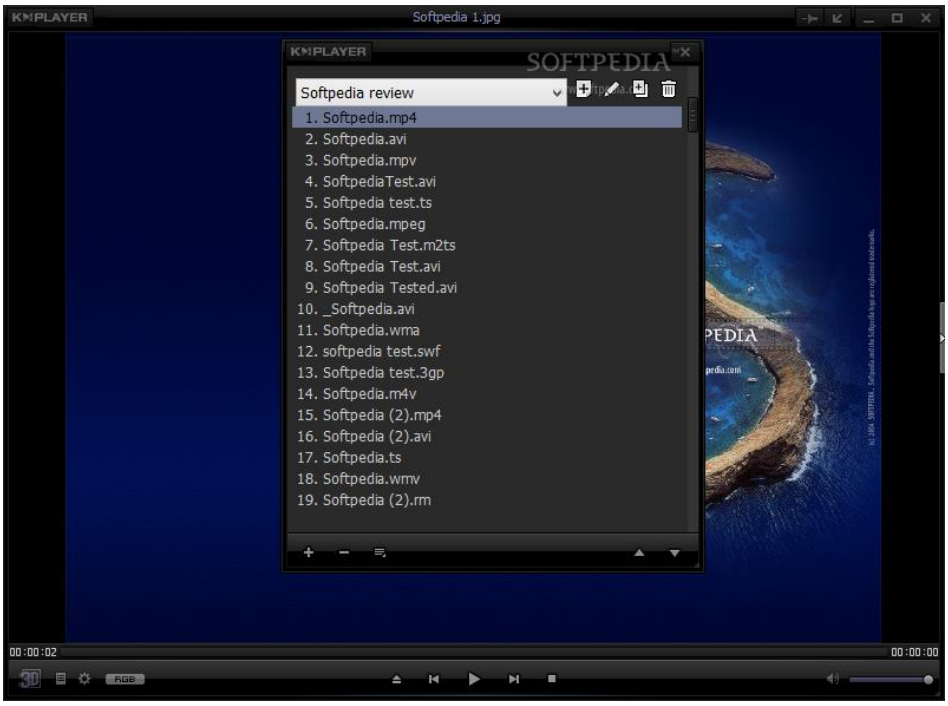
That was a Denial of service type vulnerability that happened on 2017.11.22. KMPlayer 4.2.2.4 allows remote attackers to cause a denial of service via a crafted NSV file.

- **KMPlayer 4.2.2.28 – Denial of Service**

That's a Denial of service attack which happened on 2018.12.24. When processing subtitles format media file, KMPlayer version 2018.12.24.14 or lower doesn't check object size correctly, which leads to integer underflow then to memory out-of-bound read/write. An attacker can exploit this issue by enticing an unsuspecting user to open a malicious file. [CVE-2019-9133]

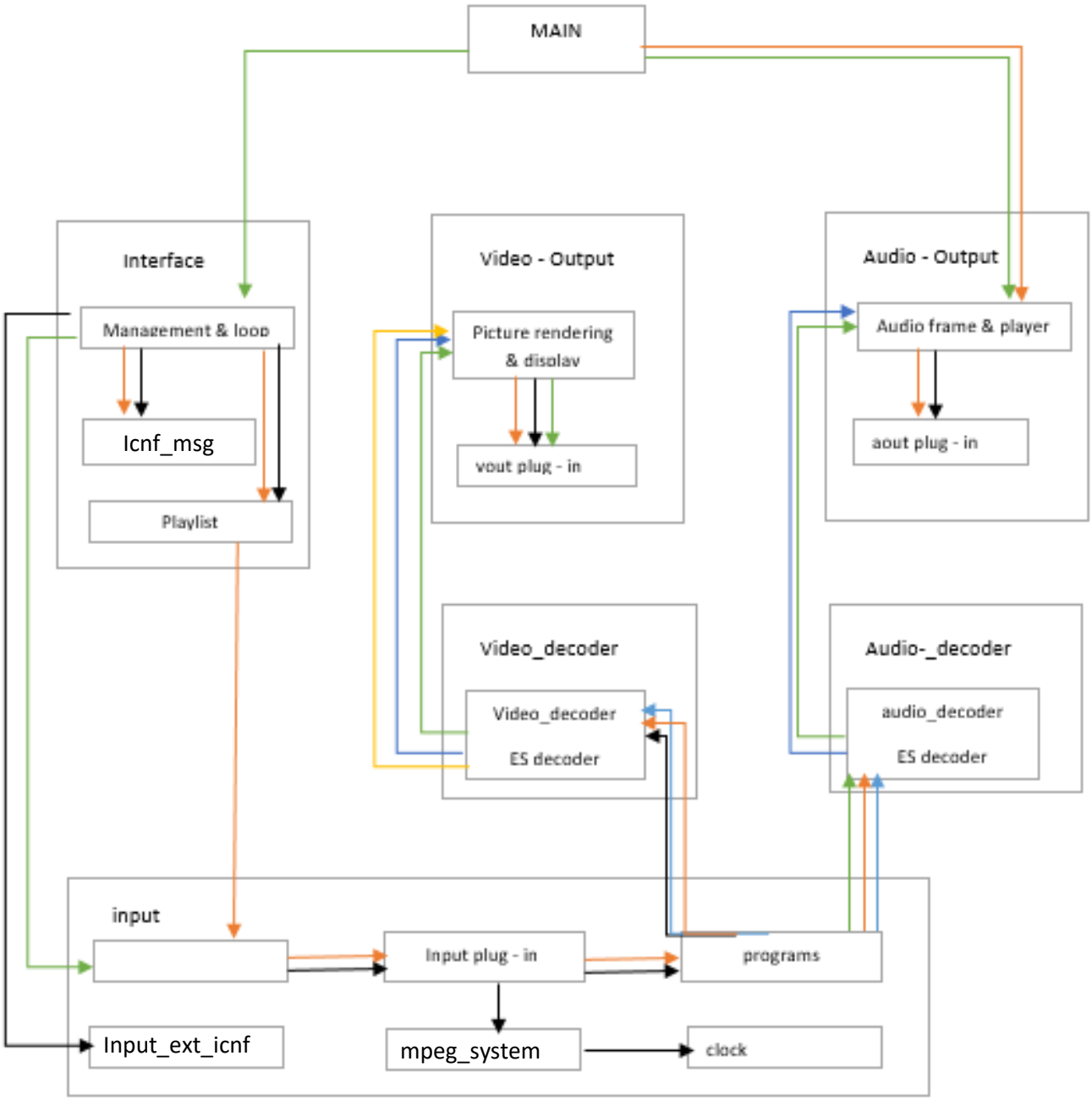
- **KMPlayer 4.2.2.31 – Denial of Service**

That happened on 2019.10.08 and also a denial of service type vulnerability. KMPlayer 4.2.2.31 allows a User Mode Write AV starting at utils!src_new+0x000000000014d6ee. [CVE-2019-17259]



Appendix B - Design Analysis

Architecture Review of KMPlayer - Introduction



In here explains how to isolated a KMPlayer in to a media controller (UI) and an actual media player (Media session). In here we show two media app architecture,

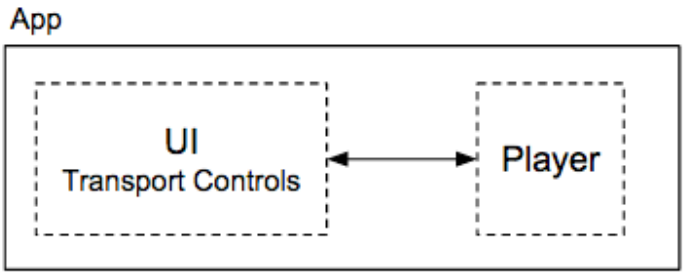
1. Client server design – That work well for audio apps
2. Single activity design – That work well for Video players

This shows how the media apps reply to hardware controls and collaborate with other apps that use the audio output stream.

Player and UI

KMPlayer which is playing audio and video usually has two parts,

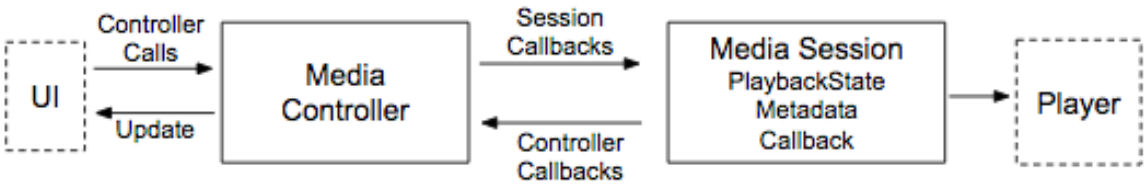
- Player get the signal of digital media and render itself to audio and/or video.
- User interface with transport controls to run the player and after that shows the player's status.



The KMPlayer provides the simple functionality on behalf of a bare-bones player that supports the most mutual audio/video formats and data sources.

Media session and media controller

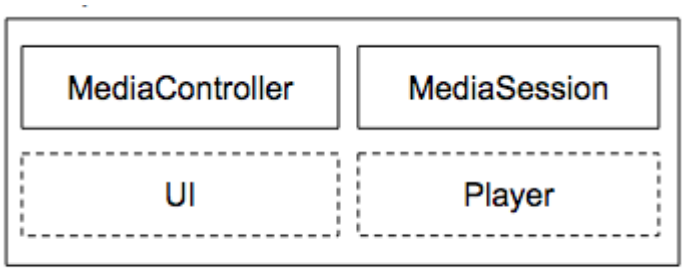
The media session of the KMPlayer and the media controller communicate by using pre-defined callbacks that reassemble for standard actions of the basic media player, as an example play callback, pause callback, stop callbacks etc. Apart from the other players KMPlayer have much more unique extensible custom calls that unique for the application.



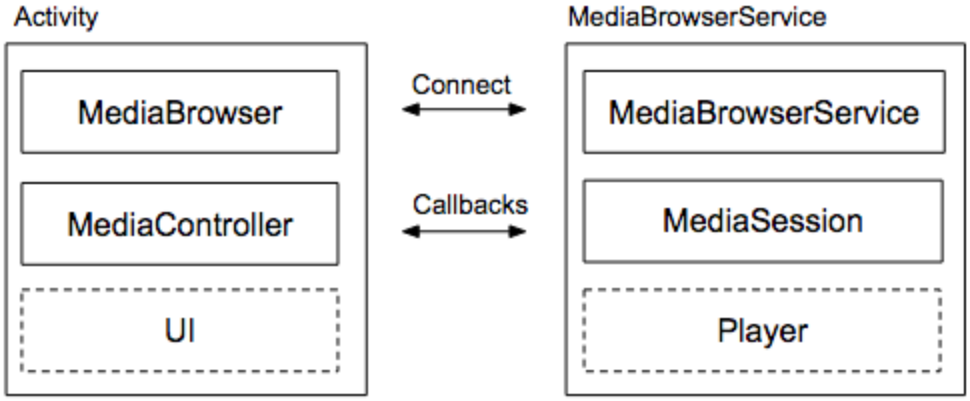
Video apps versus audio apps

When user playing a video, both eyes and ears are engaged for the process, but when user listening to music they use only ear and at that same time user will do some another activity parallel. So that KMPlayer designed a different design for each of this use cases.

When KMplayer is using for a video app, that should need a window for viewing contain.

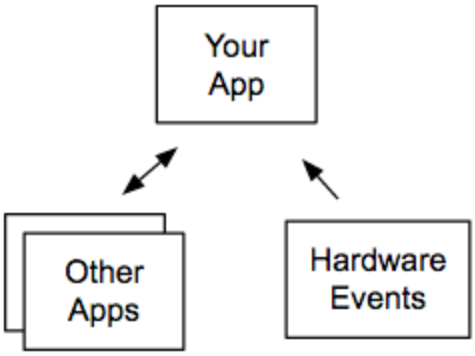


But an audio player doesn't always need to have UI visibility. Once it plays an audio, the player can have run as invisible (in background). User can do another task while continuing to listen.



Media apps and audio infrastructure

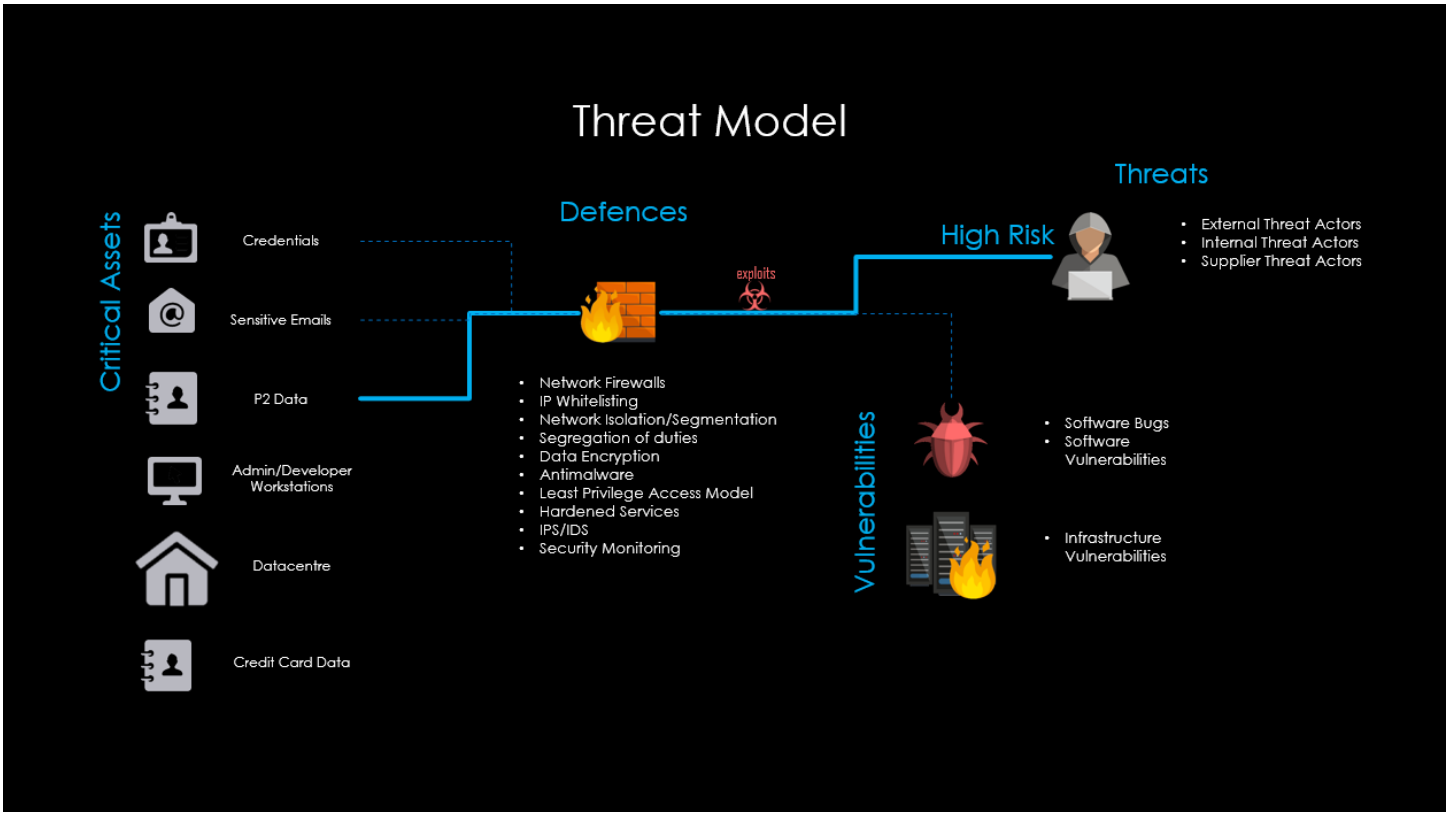
The KMPlayer app have “Play well together” with other apps that play audio. It prepared to switch video and audio formats very well. It also responds to hardware controls on the device



Threat Modeling – What is Threat Modeling

Threat modeling is a process by which potential threats, such as structural vulnerabilities or the absence of appropriate safeguards, can be identified, enumerated, and mitigations can be prioritized. The purpose of threat modeling is to provide defenders with a systematic analysis of what controls or defenses need to be included, given the nature of the system, the probable attacker's profile, the most likely attack vectors, and the assets most desired by an attacker. Threat modeling answers questions like “Where am I most vulnerable to attack?”, “What are the most relevant threats?”, and “What do I need to do to safeguard against these threats?”. [3]

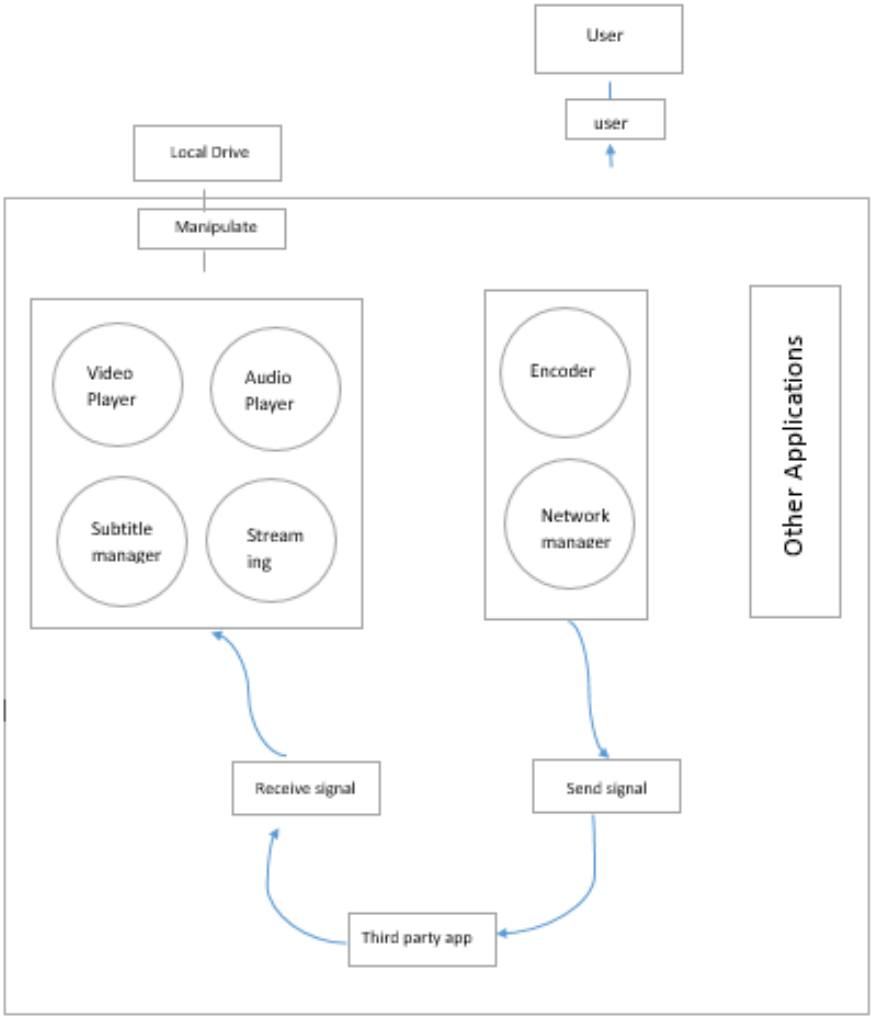
In here, when we consider about the big picture of threat modeling, the goal of the attackers is getting information from users.



In above figure shows what are the critical assets when user is using their personal PC/ device. Also that shows what kind of defenses user can be configured. But if there are much more vulnerabilities/ loopholes/ bugs in any application can break that any kind of defenses. So that we have to check the applications before installation process.

Assets to Threat Model Tracing

The KMPlayer is not affected by other programs when installing in the system. By looking at the threat model it can be seen that libKMPcore is a subsystem which is an integral part of KMPlayer and it is the cause of many functionalities which system provides as a whole. Below diagram shows the threat model of the KMPlayer. [4]



Threat Model of KMPlayer

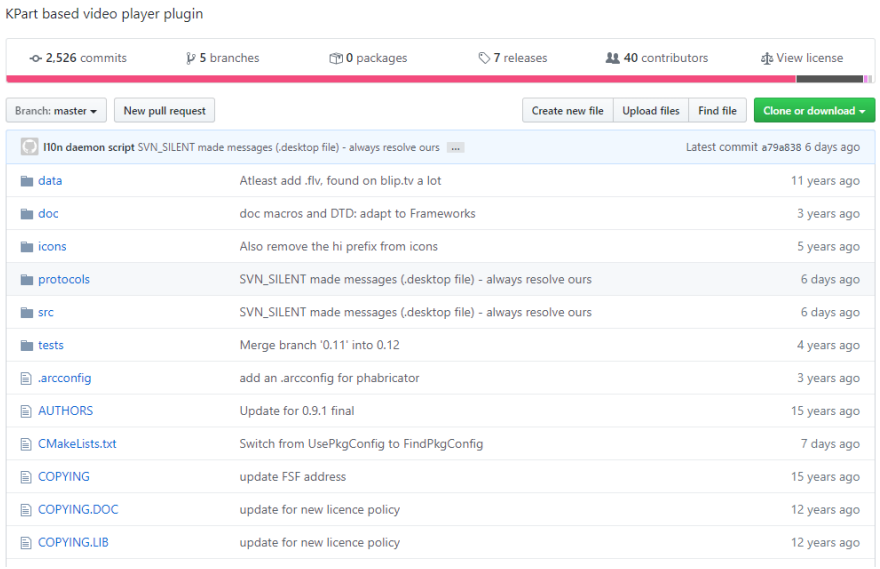
Code Inspection and Review

In order to find threats of KMPlayer, we used stride model. “Tampering”, “information disclosure” and “denial of service” are identified as threats to the KMPlayer. Security vulnerabilities lie in the streaming server and online features of KMPlayer. Three modules were inspected in KMPlayer which are MKV, MMS, and Codec. In MKV it is possible to execute code via crafter MKV files. The codec is vulnerable to execute arbitrary code by the attacker. KMPlayer is identified to be vulnerable to DOS attack. [4]

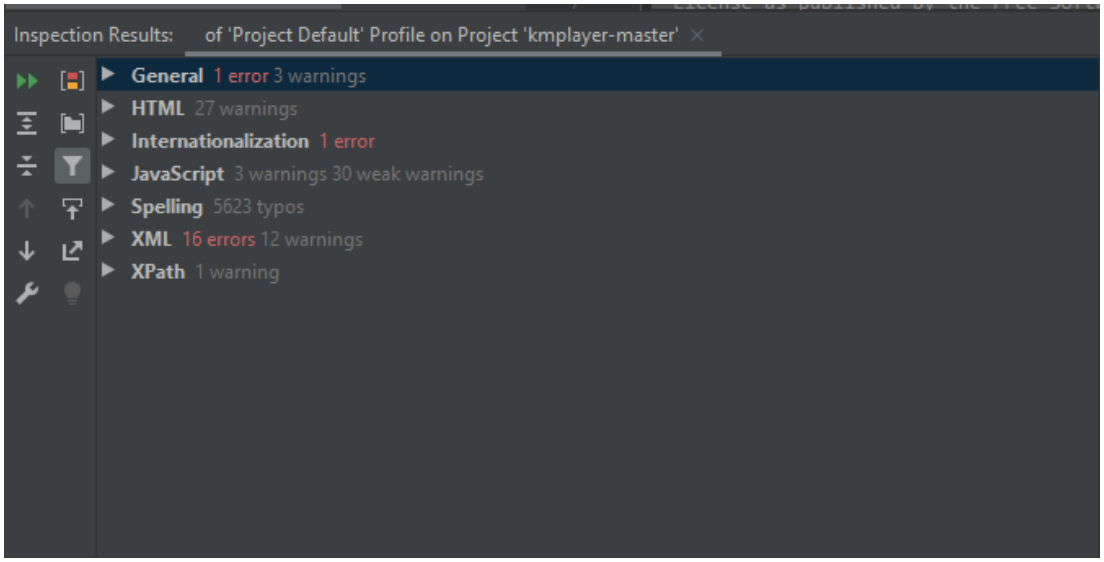
Code review

The KMPlayer source code is available on GitHub in

https://github.com/KDE/kmplayer link.



When after source code analysis, I had much issues, warnings and error message regarding this source file.



I'm not go through to the Spelling warnings (5623), Weak warnings and HTML warnings (27). When we consider about general errors and warnings,

1. There was a syntax error, in the code example_smil.html file, the problem is the element frameset is not closed. So you can understand the problem after looking about the below code screenshot.

```
        alert ("Error: " + ex);
    }
}
function play(id, url) {
    if (currentid > -1) {
        var td = menuframe.document.getElementById(currentid);
        td.style.backgroundColor = '#323232';
    }
    var td = menuframe.document.getElementById(id);
    td.style.backgroundColor = '#646464';
    currentid = id;
    var doc = playerframe.document;
    doc.open();
    doc.write("<html><body bgcolor='#161616'><embed type='video/x-ms-wmv' src='" + url
    doc.close();
}
function finished(id) {
    var td = menuframe.document.getElementById(id);
    td.style.backgroundColor = '#323232';
}
</script>
<frameset cols="200,*" onLoad="loadXML('file:/your-smil.xml')">
    <frame name="menuframe" src="about:blank">
    <frame name="playerframe" src="about:blank">
</html>
```

2. There are 3 warnings regarding the General category

```
Annotator 3 warnings
    blip-api.xsl 1 warning
        Unknown function '{http://exslt.org/strings}encode-uri'
    example_smil.html 2 warnings
        Duplicate declaration
        Duplicate declaration
```

Then When we consider about internationalization category, we can have a one error

3. The error is knowing as “File was loaded in the wrong encoding format”

```
Internationalization 1 error
    Lossy encoding 1 error
        npruntime.h 1 error
            File was loaded in the wrong encoding: 'UTF-8'
```

```
Reload in 'windows-1252'    Set project encoding to 'windows-1252'    Reload in another enc
                    const NPVariant *value);
bool NPN_RemoveProperty(NPP npp, NPObject *npobj, NPIdentifier propertyName);
bool NPN_HasProperty(NPP npp, NPObject *npobj, NPIdentifier propertyName);
bool NPN_HasMethod(NPP npp, NPObject *npobj, NPIdentifier methodName);
bool NPN_Enumerate(NPP npp, NPObject *npobj, NPIdentifier **identifier,
                   uint32_t *count);
bool NPN_Construct(NPP npp, NPObject *npobj, const NPVariant *args,
                   uint32_t argCount, NPVariant *result);

/*
    NPN_SetException may be called to trigger a script exception upon
    return from entry points into NPObjects.  Typical usage:

    NPN_SetException (npobj, message);
*/
void NPN_SetException(NPObject *npobj, const NPUTF8 *message);

#ifdef __cplusplus
}
#endif

#endif
```

Then, we are going to consider about the Javascript warning section, there are main 3 warnings and 30 weak warnings. So let's consider about warnings,

- JavaScript 3 warnings 30 weak warnings
  - ECMAScript 6 migration aids 15 weak warnings
    - 'var' used instead of 'let' or 'const' 15 weak warnings
  - General 1 warning 15 weak warnings
    - Signature mismatch 1 weak warning
    - Unresolved JavaScript function 6 weak warnings
    - Unresolved JavaScript variable 8 weak warnings
    - Unused global symbol 1 warning
      - example_smil.html 1 warning
  - Probable bugs 2 warnings
    - Equality operator may cause type coercion 2 warnings
      - example_smil.html 2 warnings
        - Comparison node.nodeName == "video" may cause unexpected type coercion
        - Comparison title == "" may cause unexpected type coercion

4. Unused global symbol warning in General category,

```
ms-wmv' src='" + url + "' width='100%' height='100%'><script>\nfunction onFinished(){t
```

5. Equality operator may cause type coercion in Probable bugs

```
function writeMenu(node, doc) {
    if (!node) return;
    if (node.nodeName == "video") {
        var src = node.getAttribute("src");
        var title = node.getAttribute("title");
        if (!title || title == "")
            title = "no title";
        doc.write("<tr><td id='" + entriecount + "'><a href=\"javascript:top.play(" +
```

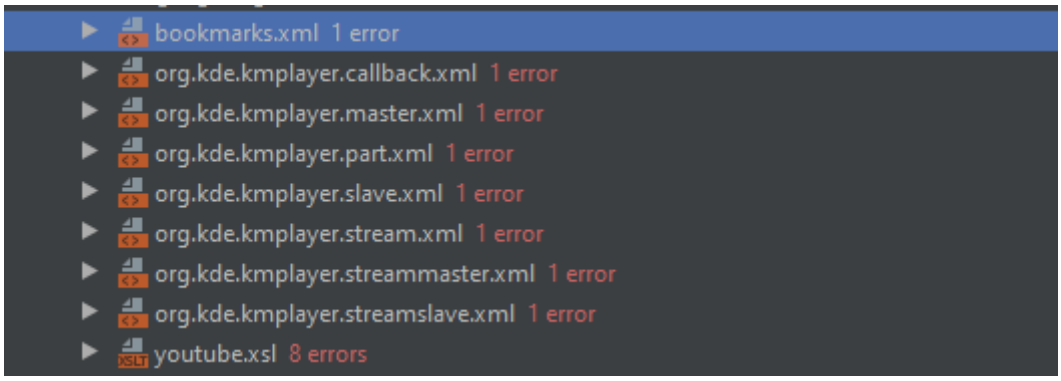
Let’s consider about XML errors and warnings,

```
XML 16 errors 12 warnings
  Unused XML schema declaration 5 warnings
    youtube.xsl 5 warnings
  XML highlighting 16 errors
    bookmarks.xml 1 error
    org.kde.kmplayer.callback.xml 1 error
    org.kde.kmplayer.master.xml 1 error
    org.kde.kmplayer.part.xml 1 error
    org.kde.kmplayer.slave.xml 1 error
    org.kde.kmplayer.stream.xml 1 error
    org.kde.kmplayer.streammaster.xml 1 error
    org.kde.kmplayer.streamslave.xml 1 error
    youtube.xsl 8 errors
  XML tag empty body 7 warnings
    controls.html 7 warnings
```

6. Unused XML schema declaration - In “youtube.xsl” file contains 5 warnings regarding schema declaration. Checks for unused namespace declarations and location hints in XML

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:n1="http://www.w3.org/2005/Atom"
xmlns:openSearch="http://a9.com/-/spec/opensearchrss/1.0/"
xmlns:gml="http://www.opengis.net/gml"
xmlns:georss="http://www.georss.org/georss"
xmlns:media="http://search.yahoo.com/mrss/"
xmlns:batch="http://schemas.google.com/gdata/batch"
xmlns:yt="http://gdata.youtube.com/schemas/2007"
xmlns:gd="http://schemas.google.com/g/2005">
```

7. XML highlighting – There are 16 errors regarding this category. Highlights XML validation problems in the results of batch code inspection.

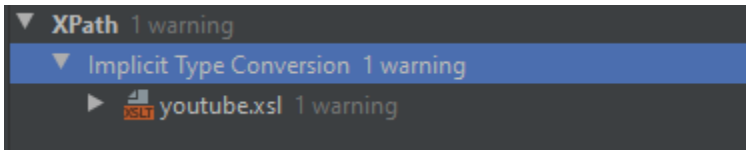


```
<bookmark icon="sound" href="http://radio.netbynet.ru:8000/di.fm.trance" >
 <title>Digitally Imported - Trance</title>
</bookmark>
<bookmark icon="sound" href="http://radio.netbynet.ru:8000/di.fm.vocal-trance" >
 <title>Digitally Imported - Vocal Trance</title>
</bookmark>
<bookmark icon="sound" href="http://www.groovefm.de/modules/mod_shoutcastextended/
 <title>GrooveFM</title>
 <info>
  <metadata owner="http://www.kde.org" />
 </info>
</bookmark>
<bookmark icon="sound" href="http://213.251.135.175:8040" >
 <title>Radio Blagon - Techno</title>
</bookmark>
<bookmark icon="sound" href="http://www.shoutcast.com/sbin/shoutcast-playlist.pls?
 <title>TechnoBase.FM</title>
</bookmark>
</folder>
<folder folded="yes" icon="bookmark_folder" >
 <title>Funk, Soul, Disco, R&amp;B</title>
 <bookmark icon="sound" href="http://64.62.252.130:8023" >
  <title>1.FM - Urban Adult Choice</title>
 </bookmark>
 <bookmark icon="sound" href="http://145.58.33.31:8056" >
  <title>AVRO Back to the Old School</title>
```

8. XML tag empty body - Reports empty tag body. The validation works in XML / JSP / JSPX / HTML/ XHTML file types. There are 7 warning regarding this problem.

```
<tr>
  <td COLSPAN=2>
    <embed SRC="excl_timings.smil" TYPE="audio/x-pn-realaudio" WIDTH=320 HEIGHT=
  </td>
  <td>
    <embed WIDTH=240 HEIGHT=240 TYPE="audio/x-pn-realaudio" CONTROLS=PlayList CO
  </td>
</tr>
<tr>
  <td>
    <embed WIDTH=160 HEIGHT=16 TYPE="audio/x-pn-realaudio" CONTROLS=PlayButton C
  </td>
  <td>
    <embed WIDTH=160 HEIGHT=16 TYPE="audio/x-pn-realaudio" CONTROLS=VolumeSlider
  </td>
  <td>
    <embed WIDTH=240 HEIGHT=55 TYPE="audio/x-pn-realaudio" CONTROLS=InfoPanel CO
  </td>
</tr>
<tr>
  <td COLSPAN=3>
    <embed WIDTH=560 HEIGHT=16 TYPE="audio/x-pn-realaudio" CONTROLS=StatusBar CO
</table>
```

Let’s consider about XPATH warning.



9. This inspection checks for any implicit conversions between the predefined XPath-types STRING, NUMBER, BOOLEAN and NODESET. While this is usually not a problem as the conversions are well-defined by the standard, this inspection can help to write XSLT scripts that are more expressive about types and can even help to avoid subtle bugs

```
        <xsl:otherwise>#C0C0C0</xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <path style="stroke:#A0A0A0;stroke-width:2px;stroke-opacity:1;fill:{$fill};" d="M
  </xsl:template>

  <xsl:template match="gd:rating">
    <xsl:variable name="avg">
      <xsl:value-of select="floor(@average) mod 6"/>
    </xsl:variable>
    <img region="rating">
      <svg width="200" height="40">
        <xsl:call-template name="svg_star">
          <xsl:with-param name="avg">
            <xsl:value-of select="$avg"/>
          </xsl:with-param>
```

References

[1] KMPlayer: The only media player you will ever need

https://www.bytesin.com/kmplayer-review-media-player/

(https://www.bytesin.com/kmplayer-review-media-player/)

[2] KMPlayer Version History - VideoHelp

https://www.videohelp.com/software/KMPlayer/version-history

[3] Threat model

https://en.wikipedia.org/wiki/Threat_model

[4] Security assessment of four open source software systems

Indonesian Journal of Electrical Engineering and Computer Science Vol. 16, No. 2,

November 2019, pp. 860~881 ISSN: 2502-4752, DOI: 10.11591/ijeecs.v16.i2.pp860-881

GITHUB Link --> https://github.com/DulRu/Exploit-Development-Project

Youtube Link --> https://www.youtube.com/watch?v=Nq2pTAv13V0

Google Drive → https://drive.google.com/open?id=1-Uu9VvM3EFMd6aFXO6gFQk9g0i7QJ2vG