

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**COMPARATIVE ANALYSIS AND BEST
PRACTICES FOR
TIVIBU WEB & DIGITURK PLAY**

Master Thesis

YAVUZ SERT

ISTANBUL, 2016

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES
COMPUTER ENGINEERING**

**COMPARATIVE ANALYSIS AND BEST
PRACTICES FOR
TIVIBU WEB & DIGITURK PLAY**

Master Thesis

YAVUZ SERT

Supervisor: ASST. PROF. DR. SELÇUK BAKTIR

ISTANBUL, 2016

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING**

Name of the thesis: COMPARATIVE ANALYSIS AND BEST PRACTICES FOR
TIVIBU WEB & DIGITURK PLAY

Name/Last Name of the Student: Yavuz SERT

Date of the Defense Thesis:

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc.Prof. Nafiz ARICA
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of
Science.

Asst.Prof.Dr. Tarkan AYDIN
Program Coordinator

This is to certify that we have read this thesis and that we find it fully adequate in scope,
quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members

Signature

Thesis Supervisor

Asst.Prof.Dr.Selçuk BAKTIR

Member

Assoc.Prof.Dr.M.Alper TUNGA

Member

Asst.Prof.Dr.Alptekin KÜPÇÜ

ACKNOWLEDGEMENTS

This thesis is dedicated to **my wife Kıymet SERT** who has been my biggest support to do this master's program many years after my graduation.

I also wish to thank my supervisor **Asst. Prof. Dr. Selçuk BAKTIR** for his guidance and support.

Istanbul, 2016

Yavuz SERT

ABSTRACT

COMPARATIVE ANALYSIS AND BEST PRACTICES FOR TIVIBU WEB & DIGITURK PLAY

Yavuz SERT

Computer Engineering

Thesis Supervisor Asst.Prof.Dr.Selçuk BAKTIR

January 2016, 61 pages

In today's world, voice-based revenue for telecom operators is decreasing because of low-cost or free voice based solutions. Operators have to provide new valuable services beside voice based services to keep their subscribers and revenue.

Another type of service that is preferred by operators is television and entertainment service. Subscribers can reach good-quality content through infrastructure of the operator online with all types of supported devices such as, tablets or smart phones. Operators profit from renting exclusive contents and periodic subscription fees for this type of service.

There are two major operators that provide such services in our country, TTNET's Tivibu and Digiturk. The purpose of this study is to offer best practices for Web TV solutions. This is done by examining and comparing topics such as streaming technologies and levels of security; revealing positive and negative aspects of the solutions, and suggesting solutions for negative aspects of both solutions. In this context, these two services were tested with real service subscribers, assessment was done based on the results of the tests and an evaluation of the services according to the assessments was done.

Keywords: Tivibu, Digiturk, WebTV, Streaming, DRM

ÖZET

TIVIBU WEB & DIGITURK PLAY İÇİN KARŞILAŞTIRMALI ANALİZ VE EN İYİ UYGULAMA YÖNTEMLERİ

Yavuz SERT

Bilgisayar Mühendisliği

Tez Yöneticisi: Yrd. Doç. Dr. Selçuk BAKTIR

Ocak 2016, 61 sayfa

Günümüz Telekom dünyasında ses üzerinden gelir imkanları daha az maliyetli hatta maliyetsiz çözümler nedeni ile gün geçtikçe daralmaktadır. Operatörler abonelerini tutmak ve gelirlerini korumak için ses iletimi dışında hizmetler sunmak zorundadırlar.

Büyük operatörlerin ses iletimi dışında tercih ettikleri diğer bir hizmet türü televizyon / eğlence servisleridir. Bu servis türünde aboneler operatör altyapısını kullanarak kaliteli içeriğe internetin olduğu her yerden ve desteklenen tüm cihazlardan erişebilirler. Operatörler bu servis türünde hem periyodik abonelik ücreti hem de özel içeriklerin tek başına satılması ile gelir sağlarlar.

Ülkemizde bu tür servis sağlayan iki büyük operatör vardır, bunlar TTNET'in sahip olduğu Tivibu ve Digiturk'tür. Bu çalışmanın amacı Web TV çözümleri için en üstün yöntemleri ortaya koymaktır. Bunu yaparken bu iki Web TV çözümü teknolojileri, güvenlik seviyeleri gibi konulara göre karşılaştırılmış, olumsuz ve olumlu yanları değerlendirilerek olumsuz yanları için çözüm önerileri sunulmuştur. Bu kapsamda her iki servis gerçek aboneler ile test edilmiş, bu testlerin sonucuna göre tespitler yapılmış, son olarak bu tespitlere göre iki servis arasında bir değerlendirme yapılmıştır.

Anahtar Kelimeler: Tivibu, Digiturk, WebTV, Streaming, DRM

TABLE OF CONTENTS

ABSTRACT	vi
TABLES	x
FIGURES	xi
ABBREVIATIONS	xiii
1. INTRODUCTION	1
1.1 IMPORTANCE OF CONTENT	2
1.2 INTERNET PROTOCOL TELEVISION (IPTV) AND OVER THE TOP TV (OTT)	2
1.3 OTT IN TURKEY	2
1.4 PREVIOUS WORK AND THESIS ROADMAP	3
2. TIVIBU WEB	5
2.1 TECHNOLOGY	6
2.2 ACCOUNT CREATION	7
2.3 USER AUTHENTICATION AND SESSION MANAGEMENT	9
2.4 LOGOUT PROCESS	13
2.5 STREAMS	15
2.5.1 Unencrypted Streams	18
2.5.2 Encrypted Streams	20
2.6 GEOLOCATION	23
2.7 OUTPUT PROTECTIONS	25
2.8 WATERMARKING	26
2.9 TIME DIFFERENCE PERFORMANCE BETWEEN SATELLITE AND TIVIBU WEB STREAM	27
3. DIGITURK	28
3.1 TECHNOLOGY	29
3.2 ACCOUNT CREATION	29
3.3 USER AUTHENTICATION AND SESSION MANAGEMENT	30
3.4 LOGOUT PROCESS	32
3.5 STREAMS	33
3.5.1 Octoshape’s Technology	35
3.5.2 Unencrypted Streams	38

3.5.3 Encrypted Streams.....	40
3.6 GEOLOCATION	43
3.7 OUTPUT PROTECTIONS	44
3.8 WATERMARKING.....	45
3.9 TIME DIFFERENCE PERFORMANCE BETWEEN SATELLITE AND DIGITURK PLAY STREAMS	45
4. HACKING FLASH PLAYER OF DIGITURK PLAY.....	47
5. COMPARATIVE ANALYSIS AND BEST PRACTICES	57
5.1 CLASSIFICATION BASED ON SOLUTION ARCHITECTURE	57
5.2 STREAM PROTECTION	58
5.3 LOAD BALANCING	59
5.4 WEB SITE SECURITY	59
5.5 DRM	59
5.6 GEOLOCATION	60
5.7 TIME DIFFERENCE FROM SATELLITE	60
6. CONCLUSION	61
BIBLIOGRAPHY	62
VITAE	68

TABLES

Table 2.1: Time difference between satellite broadcasting and Tivibu Web streams	27
Table 3.1: Time difference between satellite broadcasting and Digiturk Play streams.....	46

FIGURES

Figure 2.1: Request details for account creation process	8
Figure 2.2: Account creation parameters	9
Figure 2.3: Account creation request details	9
Figure 2.4: Fiddler output for login process	10
Figure 2.5: Service configuration response	11
Figure 2.6: Request Headers, tivibus cookie	12
Figure 2.7: Sample tivibua cookie	12
Figure 2.8: HTTPS requests.....	13
Figure 2.9: Logout Request and Response	13
Figure 2.10: Active session warning	14
Figure 2.11: Active Session termination warning	14
Figure 2.12: Client request for login operation	16
Figure 2.13: Server response for login.....	16
Figure 2.14: Tivibu Web HTTP Streams	17
Figure 2.15: Tivibu Web CDN Session	18
Figure 2.16: Manifest Response	18
Figure 2.17: Microsoft Smooth Stream Chunks	19
Figure 2.18: Manifest file of the encrypted stream.....	20
Figure 2.19: Decoded Protection Header tag	21
Figure 2.20: individualization process for PlayReady.....	22
Figure 2.21: License control request	23
Figure 2.22: Warning message for geolocation control.....	24
Figure 2.23: Invalid IP response	24
Figure 2.24: "Modify Headers" add-on configuration page	25
Figure 2.25: Screenshot taken from Tivibu Desktop client.....	26
Figure 3.1: Browser warns as the connection is not secure.....	30
Figure 3.2: HTTPS based communication	31
Figure 3.3: Sample request for login operation.....	31
Figure 3.4: Sample request for TV link	32
Figure 3.5: Logout Operation.....	32
Figure 3.6: Warning message for multiple session.....	33
Figure 3.7: Octoshape Player download link	34

Figure 3.8: Digiturk Play and Tivibu Web Clients on same machine with one internet connection, same stream.	34
Figure 3.9: Octoshape's grid technology	36
Figure 3.10: Octoshape's UDP packages captured in Wireshark	36
Figure 3.11: OctoshapeClient.exe	37
Figure 3.12: One of the clients gets stream from another client not server	37
Figure 3.13: Manifest file for a catchup TV video in Digiturk Play service	38
Figure 3.14: Smooth Streaming Chunks for Catch UP TV videos	39
Figure 3.15: Microsoft Smooth Streaming Test Player with Digiturk Play Catchup TV video Manifest File	40
Figure 3.16: Digiturk Play video renting page	41
Figure 3.17: Request details for rented video stream start	41
Figure 3.18: Manifest for encrypted video in Digiturk Play	42
Figure 3.19: Decoded ProtectionHeader value	42
Figure 3.20: Bit rate values for encrypted videos	43
Figure 3.21: Warning message of Digiturk Play's geolocation control	44
Figure 4.1: Source code of Digiturk Play player web page	47
Figure 4.2: Apache Web Server configuration	48
Figure 4.3: Hosts file entries	48
Figure 4.4: JPEXS Free Flash Decompiler	49
Figure 4.5: Search results for onOctoshapeShowUnique term	50
Figure 4.6: Code for onNewNetStream function	50
Figure 4.7: Code of onOctoshapeShowUnique function	51
Figure 4.8: Class code for our call	51
Figure 4.9: Assembler code for this_view function	52
Figure 4.10: Calling this._view.showUnique function	52
Figure 4.11: Digiturk Play's player with watermark	53
Figure 4.12: Response for "/AjaxRequest/GetOctoshapeTicket" request	53
Figure 4.13: Digiturk Play's player with our watermark	54
Figure 4.14: Code for showUnique function	54
Figure 4.15: Code for newUnique function	55
Figure 4.16: Watermark with the alpha value 0.2	56
Figure 4.17: Obfuscated code example	56

ABBREVIATIONS

CDN	:	Content Delivery Network
DRM	:	Digital Rights Management
HLS	:	Http Live Streaming
IPTV	:	Internet Protocol Television
NPAPI:		Netscape Plugin Application Programming Interface
OTT	:	Over the Top Television
RC4	:	Rivest Cipher 4
RSA	:	Rivest, Shamirand, Adleman
RTMP	:	Real Time Messaging Protocol
SHA1	:	Secure Hash Algorithm 1
SS	:	Smooth Streaming
SSO	:	Single Sign On
STB	:	Set Top Box
SVOD	:	Subscription Video on Demand
SWF	:	Small Web Format
TVOD	:	Transactional Video on Demand
VOD	:	Video on Demand
VPN	:	Virtual Private Network

1. INTRODUCTION

As technological improvements changed the devices that we use, so has the way of using them. With the growing speed of Internet access and capability of using this speed for devices, such as smart phones, tablets, or smart watches, lifestyle and entertainment habits of human beings have changed as well.

In the light of these developments, mobile and fixed telecom operators started to gravitate towards a more data-services-centric mode [1] to protect their revenue and subscribers since their revenue was almost based on the voice transmission and short message service before.

One of these services is entertainment. As Internet access speed increased, accessing contents (such as videos, music clips, games, etc.) has become easy on the Internet with High-definition¹ quality. Today, both mobile and fixed operators are trying to increase the number of their subscribers and hence their revenue by providing Internet TV² services.

1.1 IMPORTANCE OF CONTENT

For Internet TV services, one of the most important parts of the service is the content. Contents, such as TV films, videos or games are provided by content providers such as Sony Pictures, Fox or SineTivi. Operators sell these contents at a certain price, so the content is valuable. Because of this value, service providers have to provide security for these contents. For example, Digiturk pays 450 million dollars annually for the broadcasting rights for the Turkish Super League.³ Similarly, film studios spend a lot of money to make films, so they lay down security as a critical condition to provide contents to service providers.

¹ https://en.wikipedia.org/wiki/High-definition_video

² https://en.wikipedia.org/wiki/Internet_television

³ <http://www.ligtv.com.tr/haber/super-lig-maclari-2016-2017ye-kadar-digiturkte>

1.2 INTERNET PROTOCOL TELEVISION (IPTV) AND OVER THE TOP TV (OTT)

There are two main services for Internet TV; IPTV [2] and OTT [3]. IPTV service is provided in a closed network of the operator. Subscribers need a set-top-box [4] to obtain this service. OTT service is provided online. This service doesn't require a special client like STB; subscribers can reach the content via a PC, mobile phone, tablet or a Smart TV client.

IPTV service is provided on a closed network but still, there are important security issues on flows such as STB authentication and authorization to prevent unauthorized access to the system. OTT service is an online service and OTT subscribers can use any internet connection to access the service, so security is crucial.

As for content providers, the content is very valuable and service provider has to protect the content. Protection has to be done in two ways by the content providers; first, some contents might be produced for specific geographic locations, service providers have to provide access to that content only from authorized locations. Second, the content has to be served only to the subscribers who have rights to access that content. Service providers have to provide a Digital Rights Management [5] solution to solve this problem. Other problem that has to be solved by a digital rights management server is the protection of the content from copying and recording. An OTT solution has to solve these kinds of security issues to provide a successful service.

1.3 OTT IN TURKEY

The first deployment of these kinds of services was done by TTNET [6] in Turkey. TTNET uses the internet infrastructure of Türk Telekom. TTNET was the first to deploy an OTT service, then an IPTV service was launched as the first IPTV solution in Turkey with the brand of "Tivibu EV". Another company in this sector is Digiturk. Digiturk has a satellite solution, but it is not in our scope as it is not an IPTV or OTT solution. Digiturk Play was launched by Digiturk as an OTT service in 2012.

1.4 PREVIOUS WORK AND THESIS ROADMAP

OTT services have some sub-services such as Web TV, Mobile TV and Smart TV. In this thesis, two Turkish Web TV services, Tivibu Web and Digiturk Play, will be compared. Issues of streaming infrastructure, content protection applications, web site security, DRM solutions, output protections, geographical restrictions and client capabilities will be compared. As a conclusion, deficiency points of said services will be mentioned and best practices will be revealed.

Issues that will be discussed for this comparison are: accessing the service, session management, authorization and authentication, login to the service with same username from different locations, streaming infrastructure, content security and rights management, service security and location control.

Valuable-content-oriented Internet TV services are very popular in the world. Especially in the United States, service providers such as Hulu and Netflix are the leading service providers. These operators make their content available online. There is one study on the OTT service security. That thesis was studied at the University of Lisbon by Carlos Filipe Zambujo Lopes Pereira [7]. In his study, Pereira compared Netflix, Hulux and Comcast Xfinity TV services.

To examine Tivibu Web and Digiturk Play, we captured and monitored the Internet traffic of the clients of these services. Tools such as Wireshark⁴, Fiddler⁵ and capabilities of browsers were used to capture and examine the Internet traffic. Location control tests were done with a Chrome⁶ browser extension called ZEN Mate⁷, and with this extension, traffic could be simulated as if it is coming from outside of Turkey.

Our aim in this study is to examine two popular Web TV solutions in Turkey and reveal the best practices about Web TV solutions. In this study, Tivibu Web solution will be examined in the first part, then Digiturk Play solution will be examined through the

⁴ <https://www.wireshark.org/>

⁵ <http://www.telerik.com/fiddler>

⁶ <https://www.google.com/chrome/browser/desktop/index.html>

⁷ <https://zenmate.com/>

issues mentioned before. In Section 4, a hacking study will be done on Digiturk Play's flash player. In the last section, a conclusion will be made and the best practices will be mentioned.

2. TIVIBU WEB

Tivibu is the brand of the Internet TV service of TTNET. TTNET uses this brand for both IPTV and OTT services. For IPTV services the brand is “Tivibu EV”⁸ and for OTT services the brand is “Tivibu Go”⁹. There are 290.932 subscribers for Tivibu EV service [7].

Tivibu Web, officially “Tivibu Go”, is the first service TTNET launched as an Internet TV service. Subscribers can access “Tivibu Go” service via PC, browser, mobile/tablet and Smart TV clients. Tivibu Go service has 1.5 million subscribers [8]. In this report, only the number of total Tivibu subscribers is mentioned; number of Tivibu Web subscribers is calculated by subtracting the number of Tivibu EV subscribers from the total number of Tivibu subscribers.

Tivibu WEB service has five essential functions [9]:

1. **TV Broadcasting:** There are over 140 national and international TV channels in Tivibu WEB service¹⁰. In addition, TTNET provides its own thematic channels in this service such as Tivibu Spor.
2. **Subscription Video on Demand (SVOD)** [10]: Subscribers can access this type of videos based on their subscription packages; there is no additional fee for SVODs.
3. **Transactional Video on Demand (TVOD):** Subscribers can watch transactional videos by renting them. They have to pay a fee to rent the video for a certain period of time. After that time expires, subscriber cannot access the content. Popular videos are served as TVOD in general.

⁸ <http://www.tivibu.com.tr/sikca-sorulan-sorular/iptv>

⁹ <http://www.tivibu.com.tr/sikca-sorulan-sorular/tivibu-go>

¹⁰ <http://www.tivibu.com.tr/paketler/super-paket-go>

4. **Catchup TV:** This service is based on recording broadcasts of the contracted channels and serving these records as SVOD videos. These videos are recorded by the server of Tivibu Web. Subscribers cannot choose the programs that will be recorded. Since catchup TV videos are served as SVOD videos, there is no additional fee to access this type of content.

5. **Pause-Watch / Rewind:** Subscribers can pause live content with this option. After a while, subscribers can continue to watch the stream from the paused point. This option is managed by the server. The duration of the pause is limited to 1 hour [11]. Subscribers can also rewind live stream up to 1 hour in Tivibu Web service. Pause-watch and Rewind options for Tivibu EV and Tivibu Web services work differently. For Tivibu Web, the stream is recorded by the server so the subscribers can rewind the stream as they shift to that channel. For Tivibu EV, stream is recorded by the STB, so the subscribers can rewind the stream only if they watch the stream.

TTNET OTT service consists of three types of sub-services. These sub-services are web service, mobile service and Smart TV service. All these services use the same Internet TV infrastructure of TTNET. Tivibu Mobile service (with the brand name Tivibu Cep) is providing service via Android Smart Phone, Android tablet, IPAD, iPhone and Windows 8. Tivibu Smart TV service is providing service via Next OTT box, Vestel OTT box, Arçelik, Samsung, LG, Philips, Beko and Vestel Smart TVs.

2.1 TECHNOLOGY

Tivibu services use different technologies for infrastructure because of various number of clients using specific services.

Tivibu Web service is a Microsoft Silverlight [13] based solution for both desktop and browser clients. Silverlight uses [14] Microsoft Smooth Streaming [15] technology to play live streams, so Tivibu Web uses Microsoft Smooth Streaming technology to play live broadcast / video streams for desktop / browser clients. In the same way, Tivibu

uses Microsoft PlayReady [16] application as a DRM solution, as it is also a Microsoft product.

Subscribers should have a Silverlight supported operating system and browser to watch Microsoft Smooth Streaming based streams. If they don't not have a Silverlight supported browser, clients are prompted to download and install Silverlight at the first attempt to access the service. In 2013, Google Chrome announced [17] they will not support NPAPI as of 2014. Now, Google Chrome cannot be used as a Tivibu Browser client. Likewise, Microsoft Edge browser does not support Silverlight as they removed ActiveX support from the Edge browser [18].

2.2 ACCOUNT CREATION

It is enough to create a single account to use all TTNET services. This technology is called "Single Sign On (SSO)". Tivibu Web service also uses SSO authentication and authorization service. Account set up can be done at a TTNET dealer or from <https://uyelik.tivibu.com.tr/tivibugo> web site. Tivibu services can be accessed through SSO like any other TTNET services such as "TTNET Muzik" or "TTNET Wifi". During account set up, the web site requests Turkish Republic National Identity Number and a credit card number to create the account and to withdraw the subscription fee. To protect these data, account creation page is served under HTTPS [19] protocol. Data transmission is done via HTTP POST method.

Figure 2.1: Request details for account creation process

```
POST /kisisel-bilgiler HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Referer: https://uyelik.tivibu.com.tr/kisisel-bilgiler
Accept-Language: tr,en-US;q=0.7,en;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: uyelik.tivibu.com.tr
Content-Length: 39
DNT: 1
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: _ga=GAI.3.114120846.1419430713; _gat=1; ASP.NET_SessionId=rz5btuvikejtzrnwigatjty5
citizenshipno=2[REDACTED]&captcha=18141
```

Web pages that take these kinds of parameters could be targeted by hackers. One of the most used methods to attack the web submit pages is to run an automated script that sends parameters to that web server. Captcha [20] is a protection method that is used against this kind of attack. With captcha protection method, the web page checks whether the requester is a computer or a human. Tivibu Go account set up page uses captcha protection method. Figure 2.1 shows HTTP POST details for a request. It is shown that “captcha” parameter is used to control whether the requester is a computer or not.

The value that the user submits for the captcha parameter can be controlled by the server or by the client. Client control should not be preferred because with a proxy¹¹ system like Burp Suite¹², requests can be manipulated by an attacker. Results of tests show that captcha controls for Tivibu account creation pages are done at by the server. It is not possible to bypass captcha control as the control is done by the server.

¹¹ https://en.wikipedia.org/wiki/Proxy_server

¹² <https://portswigger.net/burp/>

Figure 2.2: Account creation parameters

We used Burp Suite proxy program to catch requests, after the user submits parameters as it is shown in Figure 2.2. Burp Suite catches the details of the request. During the test “invalid captcha” message appeared as server controls the process.

Figure 2.3: Account creation request details

```
POST /kisisel-bilgiler HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Referer: https://uyelik.tivibu.com.tr/kisisel-bilgiler
Accept-Language: tr,en-US;q=0.7,en;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: uyelik.tivibu.com.tr
Content-Length: 39
DNT: 1
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: _ga=GA1.3.114120846.1419430713; _gat=1; ASP.NET_SessionId=rz5btwikejtzrnwlgatjty5
citizenshipno=28[REDACTED]&captcha=10300
```

2.3 USER AUTHENTICATION AND SESSION MANAGEMENT

Subscribers can use Tivibu PC client or Tivibu Browser client to access the Tivibu Web service. Subscribers should have a TTNET SSO user to login to the system.

Tivibu Browser client is an embedded version of Tivibu PC client (also known as native client). When you open the Tivibu Web Browser client for the first time, client download is initiated as a XAP [21] file. These two clients are identical in terms of functionality.

We used Fiddler application to capture the traffic between the client and the server to examine the service. This program shows all incoming and outgoing traffic for the client. When subscriber logs in to Tivibu client, it first makes an authentication request to the server. Server address of Tivibu system is <https://mw.webtv.ttnet.com.tr>. This authentication request is done via HTTPS protocol.

Figure 2.4: Fiddler output for login process

19	200	HTTP	Tunnel to	mw.webtv.ttnet.com.tr:443
20	304	HTTPS	mw.webtv.ttnet.com.tr	/clientaccesspolicy.xml
21	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
22	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc

Cross Site Request Forgery is a type of attack to the web pages. It is known as CSRF. It is also known as a one-click attack or session riding and it is a type of malicious exploit of a website whereby unauthorized commands are transmitted from a user that the website trusts [22]. To prevent cross-site request forgery, Silverlight allows only site-of-origin communication by default for all requests other than images and media [23]. For example, a Silverlight application hosted at <https://mw.webtv.ttnet.com.tr/LoginService> can access only services on the same domain by default, for example <https://mw.webtv.ttnet.com.tr/SecureClientServices.svc>, but an application served at <http://dummypage.com/service.svc> cannot access the site because of this protection. Because of this rule, if we want to use <https://mw.webtv.ttnet.com.tr/SecureClientServices.svc> service, our client should be located under mw.webtv.ttnet.com.tr domain. To make exceptions for this rule, clientaccesspolicy.xml file should be used. This file consists of domain names that can use the service from different domains. In Figure 2.4 request of the client to that file is shown.

User authentication and getting service related information (service configuration) is done through “SecureClientServices” service. Service configuration response, as shown in Figure 2.5, is a response for service configuration request that is done before login request, since clients get necessary information from server to make the client usable. Configuration parameters for rating and advertisements or VOD categories are examples of service configuration.

Figure 2.5: Service configuration response

```

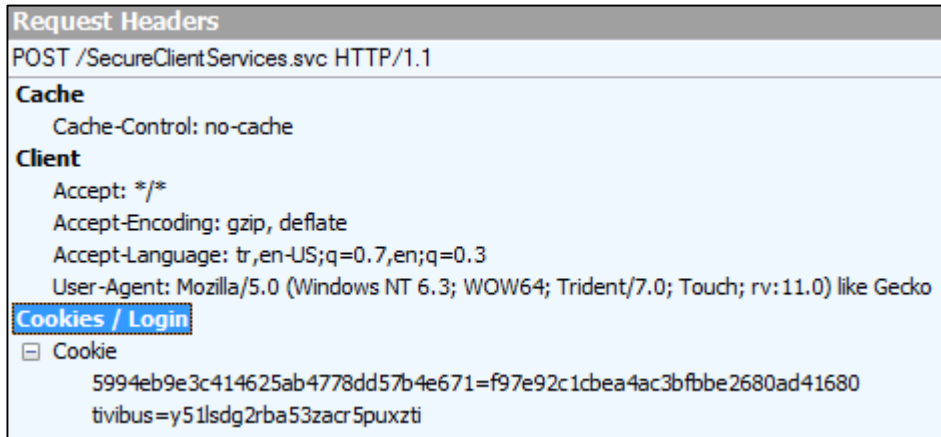
s:Envelope [ xmlns:s=http://schemas.xmlsoap.org/soap/envelope/ ]
├── s:Body
│   └── GetAPIConfigurationResponse [ xmlns=http://tempuri.org/ ]
│       └── GetAPIConfigurationResult [ xmlns:a=http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.CommonsSec xmlns:i=http://www.w3.org/2001/XMLSchema-instance ]
│           ├── ConfigurationResult [ xmlns:b=http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Commons ]
│           │   └── ErrorMessage [ xmlns:i="true" xmlns:s="http://www.w3.org/2001/XMLSchema-instance" xmlns:b="http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Commons" /> ]
│           │       └── HasError
│           │           └── false
│           └── Configuration
│               ├── LanguageInfoList [ xmlns:c=http://schemas.datacontract.org/2004/07/Argela.WebTV.MwCore.Services.CommonOperationsService ]
│               │   └── LanguageInfo
│               │       ├── ClientDefaultField
│               │       │   └── 1
│               │       ├── ClientDefaultFieldSpecified
│               │       │   └── true
│               │       ├── LanguageIdField
│               │       │   └── 1
│               │       ├── LanguageIdFieldSpecified
│               │       │   └── true
│               │       ├── LanguageLocaleField
│               │       │   └── tr
│               │       └── LanguageNameField
│               │           └── tur
│               └── RatingServerConfig [ xmlns:c=http://schemas.datacontract.org/2004/07/Argela.WebTV.MwCore.Services.CommonOperationsService ]
│                   ├── RatingPostPeriodInMinsField
│                   │   └── 1
│                   ├── CodeRatingPostPeriodInMinsField
│                   │   └── 1
│                   ├── ServerRatingLogPeriodInMinsField
│                   │   └── 5
│                   ├── ServerRatingPostPeriodInMinsField
│                   │   └── 10
│                   ├── RandomDiffInSecsField
│                   │   └── 1
│                   ├── RatingRetryCountField
│                   │   └── 3
│                   ├── RatingServerURLField
│                   │   └── https://clienttivibuplay.tivibu.com.tr/TivibuDataServer/LDataServlet
│                   └── UseOldRatingServerField
│                       └── true

```

After the login request is processed by the server, if the user data is correct, a session is created by the server and information related to the subscription package of the subscriber is given as the server’s response. This information consists of the user’s private data such as channel list, favorite list and rented video list.

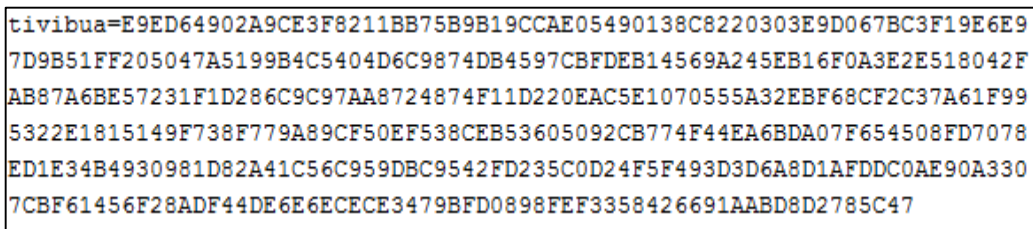
When a user logs in to Tivibu Web service, a session cookie [24] called “tivibus” is created. An example value for tivibus cookie is shown in Figure 2.6.

Figure 2.6: Request Headers, tivibus cookie



During user authorization process another cookie called tivibua is created. Here is an example value for tivibua cookie:

Figure 2.7: Sample tivibua cookie



Sessions for users created by the server are tracked by these cookies. As these cookies have no expiration time, when a user logs out or the browser is closed, the session expires. If the user wants to use Tivibu Web again, they have to login again.

As shown in Figure 2.8, all requests sent to mw.webtv.ttnet.com.tr domain are secure requests as the connection protocol is HTTPS.

Figure 2.8: HTTPS requests

23	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
24	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
25	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
26	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
28	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
60	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
61	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc

SSL connection is encrypted using 128 bit RC4 which is a famous key stream generator due to its simple algorithm and fast speed and it is widely used in some popular protocols such as SSL (Secure Socket Layer) and TLS (Transport Layer Security) to protect internet traffic [25]. Message authentication is done with SHA1 [26] and key exchange mechanism is RSA [26].

2.4 LOGOUT PROCESS

When a user logs out of Tivibu Web service, the client sends a logout request to the server. The server ends the session of the user upon this request.

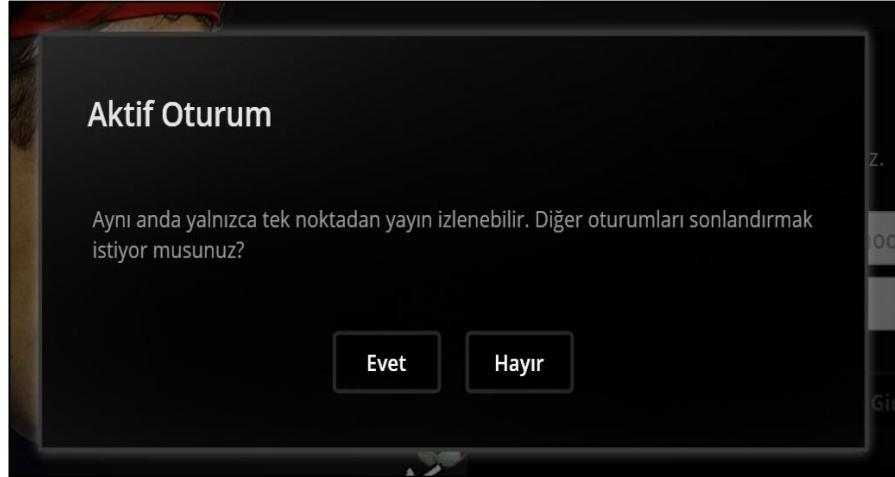
Figure 2.9: Logout Request and Response

The screenshot displays a SOAP message viewer interface. The top section shows the request message, which is an `s:Envelope` containing an `s:Body` with a `<Logout xmlns="http://tempuri.org/" />` element. Below this, there are tabs for 'Expand All' and 'Collapse'. The bottom section shows the response message, which is an `s:Envelope` containing an `s:Body` with a `LogoutResponse` element. This response includes a `LogoutResult` element with an `ErrorMessage` (i:nil="true"), a `HasError` element (i:nil="true"), and an `ReturnResult` element with the value 'SUCCESS'.

Tivibu subscribers can login to the service only once at the same time. If there is an active session, the subscriber cannot login to the service with the same username. Because of this limit, the logout process is very important. If there is an active session

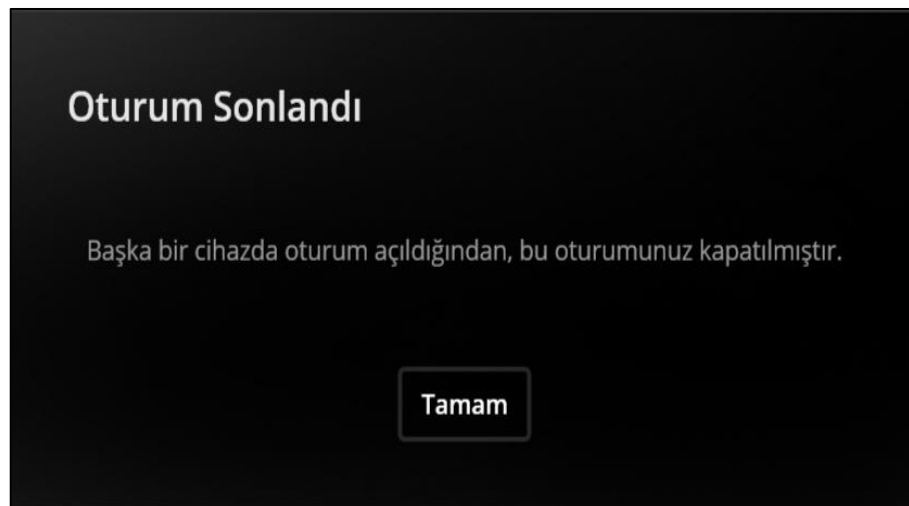
for a subscriber, and if that subscriber tries to login again with the same username, a warning message appears as shown in Figure. 2.10.

Figure 2.10: Active session warning



If the subscriber clicks “Evet” [Yes] button in this warning box, then the active session will be terminated. After this operation, the user for the other active session will get the message shown in Figure 2.11.

Figure 2.11: Active Session termination warning



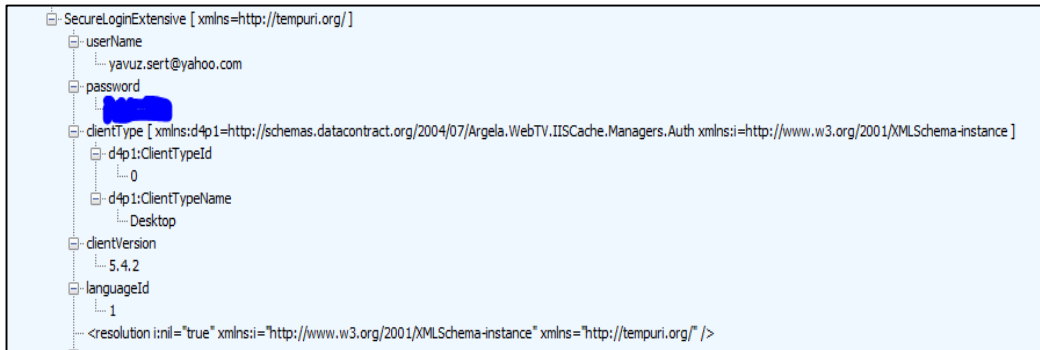
2.5 STREAMS

There are two main stream categories of Tivibu Web service: TV streams and video streams. Some of the TV channels and some of the video streams are encrypted. There are also clean streams for some channels and videos.

When a subscriber logs in to the Tivibu Web service with a client, the client gets some configuration parameters from the server application. Two of these parameters show the Content Delivery Network (CDN) address values to get the streams. This configuration consists of two values, the first is a Primary CDN address and the second is a Secondary CDN address. CDNs improve network performance and offer fast and reliable applications and services by distributing content to cache servers located close to users [28]. The primary goal of a CDN is to provide to end users such content as webpages, videos, and applications with high availability and performance. The key component that ensures availability and performance is the CDN's load balancing system that assigns each incoming request to a server that can serve that request. To this end, a CDN's load balancing system routes each user's request to a server that is active and not overloaded [29].

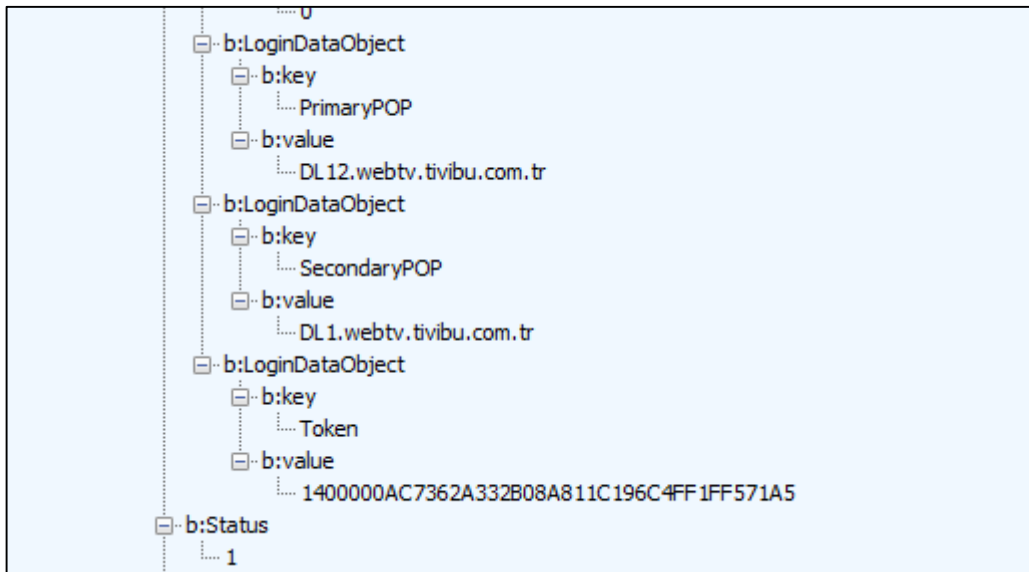
This configuration shows that, Tivibu Web service does not have a load balancing solution to access CDN streams. There is only one alternative (Second CDN address) for CDN access and if two CDN addresses are both unavailable, then the user cannot get a stream from the Tivibu Web service. If there was a load balancing solution to access the CDN, then clients could get only one CDN address from the server which is the domain address of the CDN load balancer. In such a solution, when a user tries to watch, the stream load balancer receives a request and then forwards it to the most suitable CDN address. Figure 2.12 shows login parameters that the client sends.

Figure 2.12: Client request for login operation



Upon this request, the server application processes the request and gives a response which consists of some configuration values required for the service as shown in Figure 2.13.

Figure 2.13: Server response for login



As shown in Figure 2.13, the subscriber gets “DL12.webtv.tivibu.com.tr” address as the primary CDN address, so the client tries to get streams from that address first. If there is a problem regarding getting stream from the primary CDN, then the client will try to use a second CDN address. As shown in Figure 2.13, the second CDN address is “DL1.webtv.tivibu.com.tr”.

Figure 2.14: Tivibu Web HTTP Streams

56	200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc
57	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/Manifest
58	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/Manifest
59	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio102_orj=3
60	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(500000)/Fragments(video=334023
61	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3
62	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3
63	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3
64	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3
65	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3
66	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(500000)/Fragments(video=334024
67	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3
68	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(500000)/Fragments(video=334025
69	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3
70	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(1000000)/Fragments(video=33402
71	200	HTTP	dl3.webtv.tivibu.com.tr	/Live/80001.isml/QualityLevels(64000)/Fragments(audio101_tur=3

Tivibu streams are served via HTTP protocol as shown in Figure 2.14. Some of the streams are encrypted, and some are clear. When an attacker records a capture for a clear stream, he still will not be able to watch the stream because Tivibu Web Service is setting a CDN session cookie called SID for each session and this SID value is sent to the CDN for all the chunk requests. This cookie value is generated by Tivibu clients. So even if the stream is clear, the attacker cannot watch the captured stream as he does not have the SID value from a Silverlight supported player. This technique is called “URL Signing” and ensures that only authorized clients can receive streams from the Tivibu CDN network. The URL Signing system generates a unique key for each user and this key is transmitted to streamers using cookies. Server application intercepts this request and checks the validity of the key received from the client. If the key is valid, then stream is allowed, otherwise 403 HTTP Error comes back on. For encrypted streams, there are both DRM protection and SID control.

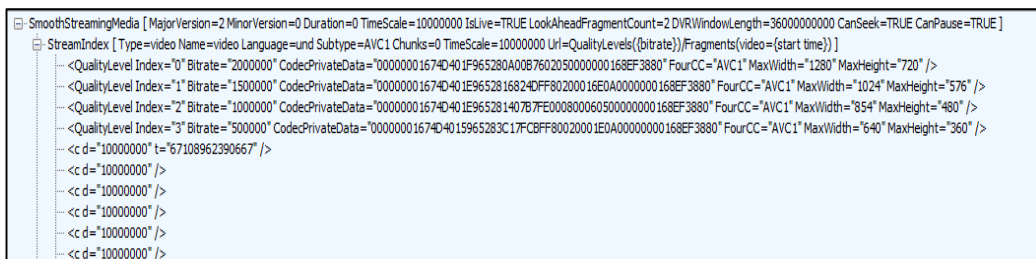
Figure 2.15: Tivibu Web CDN Session



2.5.1 Unencrypted Streams

When a subscriber switches to a TV channel in a Tivibu Web client, the client requests a Manifest [30] file first, then the server gives a response to that request which is to find out how many profiles there are and what the bitrate values are for each profile. We captured this kind of request via the Fiddler capture application as shown in Figure 2.16.

Figure 2.16: Manifest Response



The Manifest response shows us that the Tivibu Web service uses the Microsoft Smooth Streaming infrastructure. We understand this as there is a tag named “SmoothStreamingMedia” [31] in the Manifest response. Another piece of information that we got from the Manifest file is that there are four stream profiles and bitrates; these profiles are 2 mbit, 1,5 mbit, 1 mbit, and 500kbit.

Microsoft Smooth Streaming is an Adaptive Bitrate [32] based solution. Other popular adaptive bitrate based solutions are Adobe Flash¹³ and Apple HTTP Adaptive Streaming¹⁴. Adaptive bitrate streaming is a technique used in streaming multimedia.

¹³ <http://www.adobe.com/tr/products/adobe-media-server-standard.html>

¹⁴ <https://developer.apple.com/streaming/>

Adaptive streaming technologies are almost based on HTTP and designed to work efficiently over large distributed HTTP networks such as the Internet. It works by detecting a user's bandwidth and CPU capacity in real time and adjusting the quality of a video stream accordingly. It requires the use of an encoder which can encode a single source video at multiple bit rates. The client switches between these bit rates depending on available resources. In our case, according to the streaming profiles in the manifest file, if the machine sources and the Internet speed is sufficient, a stream can be watched with a 2 mbit bitrate. This manifest file does not belong to an encrypted channel stream because the manifest file doesn't consist of any encryption-related tags. Clear streams are played directly in the player according to the data read from the manifest file. Microsoft Smooth Streaming streams are received as chunks [33] by the player client. Figure 2.17 shows Microsoft Smooth Streaming chunks.

Figure 2.17: Microsoft Smooth Stream Chunks

200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/Manifest	122,657	max-age=2	text/xml
200	HTTP	Tunnel to	mw.webtv.ttnet.com.tr:443	0		
200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc	411	private	text/xml; c...
200	HTTPS	mw.webtv.ttnet.com.tr	/SecureClientServices.svc	533	private	text/xml; c...
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178097943334)	8,275	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178102390667)	72,227	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178107543334)	8,432	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178117143334)	8,143	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178126743334)	8,276	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178136343334)	8,106	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178145943334)	8,417	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178112390667)	46,923	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178155543334)	8,183	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178122390667)	50,679	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178165143334)	8,181	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178174743334)	8,270	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178132390667)	66,692	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178142390667)	72,872	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178184343334)	8,215	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178193943334)	8,211	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178152390667)	68,174	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178203543334)	8,194	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178162390667)	61,220	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178172390667)	44,250	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=6717813143334)	8,401	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178182390667)	47,988	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178222743334)	8,193	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178192390667)	65,583	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178232343334)	8,094	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178202390667)	64,756	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178241943334)	8,354	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178212390667)	59,897	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178251543334)	8,232	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(500000)/Fragments(video=67178222390667)	60,374	max-age=3600	video/mp4
200	HTTP	d12.webtv.tvibu.com.tr	/Live/spot/prt60001.isml/QualityLevels(64000)/Fragments(audio101_tur=67178261143334)	8,268	max-age=3600	video/mp4

As shown in Figure 2.17, some of the chunks are video chunks, others are audio chunks. Silverlight player receives these chunk packages and shows them as a video. The value (500000) seen in the video package shows that the stream received by player was 500kbit at that time. This manifest file and chunks are captured from a clear channel stream. In Tivibu, catchup TV VODs are not encrypted as clear channels.

2.5.2 Encrypted Streams

The security of content is very important for content owners like film studios, owners of thematic TV channels, and content providers who have rights for broadcasting of football leagues. To ensure the contents are secure, a service provider should send the stream of such contents to the player client in an encrypted form. If the content is encrypted, the path of the stream from the encoder to the player client will be secure and protected against attacks. Encrypted streams can only be decrypted with a key. As mentioned, the player reads a manifest file to get details about the stream. Encryption related data is also located in the manifest file. A manifest file for an encrypted channel is shown in Figure 2.18.

Figure 2.18: Manifest file of the encrypted stream

```
<?xml version="1.0" ?>
<SmoothStreamingMedia MajorVersion="2" MinorVersion="0" Duration="0" TimeScale="10000000" IsLive="TRUE"
LookAheadFragmentCount="2" DVRWindowLength="36000000000" CanSeek="TRUE" CanPause="TRUE">
<ProtectionHeader SystemID="9a04f079-9840-4286-ab92-e65be0885f95">
DAMAAEAQAQACAzwAVvBSAE0ASABFAEERABFAPFIALAB4AG0AbABnAHMAPQAIAGgAGAB0AHAAOgAvAC8AcwBjAGgAZQBtAGEAcwAuAG0AaQBjAHIAbwBzA
g8AZgB0AC4AYwBvAG0ALwBEAFIATQAvAD IAMAaWADcALwAwADMALwBQAGwAYQ55AFIAZQBhAQQAeQBIAGUAYQBkAGUAcgAiACAAAgBjAHIAbwBzA
A9ACIANAAuADAALgAwAC4AMAAIAD4APABEAEEAVABBAD4APABQAFIATwBUAEUAQwBUAEKATgBGAE8APgA8AEsARQBZAEwARQBOAD4AMQA2ADwALwBLAEU
AWQBMAEUATgA+ADwAQQBMAEeASQBEAD4AQQBFAFMAQwBUAFIAPAAvAEEATABHAEKARAA+ADwALwBQAFIATwBUAEUAQwBUAEKATgBGAE8APgA8AEsASQBE
AD4AUwBvAFQASQA2AGsAdABPAEEAawBXAGQAQOBiAHQAcbABsAFYAMwAzAFQAQQA9AD0APAAvAEsASQBEAD4APABMAEEAXwBVAFIATAA+AGgAdAB0AHAAO
gAvAC8AbQB3AC4AdwBLAGIAdAB2AC4AdAB0AG4AZQB0AC4AYwBvAG0ALgB0AHIAPAAvAEwAQQAeQBFAPUAgBMAD4APABMAFUASQBFAFUAgBMAD4AaAB0AH
QAcaAA6AC8ALwBtAHcALgB3AGUAYgB0AHYALgB0AHQAbgB1AHQALgBjAG8AbQAUAHQAQcA8AC8ATABVAEKAXwBVAFIATAA+ADwARABTAF8ASQBEAD4AUgB
GAGIAawAzADQAEAB0ADYAMAB1ADIARQB1AEKAUgB6AG4AaQBIAEsAdwA9AD0APAAvAEQAUwBFAEKARAA+ADwAQwBIAEUAQwBLAFMAVQBNAD4AVwAwAHcA
NQBWAFMAcgBqAHKAZQBjAD0APAAvAEMASABFAEMASwBTAFUATQA+ADwALwBEAEAEVABBAD4APAAvAFcAUgBNAEgARQBBAEQRBQBSAD4A
</ProtectionHeader><Protection><StreamIndex Type="video" Name="video" Language="und" Subtype="AVC1" Chunks="0"
TimeScale="10000000" Url="QualityLevels{(bitrate)}/Fragments(video={start time})"><QualityLevel Index="0" Bitrate=
"20000000" CodecPrivateData="00000001674D01F965280A00B7602050000000168EF3880" FourCC="AVC1" MaxWidth="1280"
MaxHeight="720"/><QualityLevel Index="1" Bitrate="10000000" CodecPrivateData=
"00000001674D01E965281407B7FE000800060500000000168EF3880" FourCC="AVC1" MaxWidth="854" MaxHeight="480"/>
<QualityLevel Index="2" Bitrate="15000000" CodecPrivateData=
"00000001674D01E9652816824DFF80200016E0A0000000168EF3880" FourCC="AVC1" MaxWidth="1024" MaxHeight="576"/>
<QualityLevel Index="3" Bitrate="5000000" CodecPrivateData=
"00000001674D015965283C17FCBFF80020001E0A00000000168EF3880" FourCC="AVC1" MaxWidth="640" MaxHeight="360"/><c d=
"10000000" t="27614505804861"/><c d="10000000"/><c d="10000000"/><c d="10000000"/><c d="10000000"/>
<c d="10000000"/><c d="10000000"/><c d="10000000"/><c d="10000000"/><c d="10000000"/><c d="10000000"/><c d=
"10000000"/>
```

As it can be seen from the manifest file, this manifest file has a <Protection> tag different from the manifest file of a clear stream. This tag shows that this stream is an encrypted stream. In this tag, there is another tag named <ProtectionHeader>. ProtectionHeader tag is base64 hashed value and it consists of data about encryption. Base64 encoding is mainly used when there is necessity to encode binary data as ASCII text that needs to be stored or transferred in environments that, perhaps for legal reasons, are restricted to US-ASCII data [34]. Decoded version of ProtectionHeader for our example is shown in Figure 2.19.

Figure 2.19: Decoded Protection Header tag

```
<WRMHEADER xmlns="http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader" version="4.0.0.0">
<DATA><PROTECTINFO>
<KEYLEN>16</KEYLEN>
<ALGID>AESCTR</ALGID></PROTECTINFO>
<KID>SoTI6ktOAKWd9btp1V33TA==</KID>
<LA_URL>http://mw.webtv.ttnet.com.tr</LA_URL>
<LUI_URL>http://mw.webtv.ttnet.com.tr</LUI_URL>
<DS_ID>RFbk34xt60u2EuIRznibKw==</DS_ID>
<CHECKSUM>W0w5pSrjyec=</CHECKSUM>
</DATA></WRMHEADER>
```

As we can see in the ProtectionHeader data, Tivibu Web system uses Microsoft PlayReady 4.0 version as DRM application. Data in ProtectionHeader shows us:

- i. The stream is encrypted because there is a <Protection> tag in ProtectionHeader.
- ii. Stream is encrypted by Microsoft PlayReady application. (from the tag <WRMHEADER xmlns="http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader" version="4.0.0.0">
- iii. KEYLEN tag reveals that KEY parameter has a 16 byte length (128 bit)
- iv. Encryption is done with AES Counter Mode. (from the tag ALGID) AES was introduced by the National Institute of Standards and Technology (NIST) in 2001. AES supports five secure modes of operation approved by the Federal Information Processing Standard (FIPS). They are Electronic Code Book Mode (ECB), Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), and Counter Mode (CTR) [35].

Explanations of those parameters are given below:

1. "KID= SoTI6ktOAKWd9btp1V33TA==": Contains a base64-encoded key ID value.
2. "LA_URL": Contains the URL for the license acquisition Web service.
3. "LA_URI": Contains the URL for a non-silent license acquisition Web page.
4. "DSID: RFbk34xt60u2EuIRznibKw==": Service ID for the domain service.

If it is the first time that a subscriber tries to watch a Microsoft PlayReady encrypted stream, client player sends a request to Microsoft server for individualization [36]

process [37]. This process is done to control whether client machine has a DRM application called “Individualized Black Box (IBX)”. If the appropriate individualized component software is not already on the client, the client automatically requests the component from the Microsoft Individualization Service [38]. The process of obtaining the individualized component software is called individualization. A request for individualization process is shown in Figure 2.20.

Figure 2.20: individualization process for PlayReady

```
POST /PlayReady/FW-OSAKA/default.freeway?Individualize HTTP/1.1
Host: services.silverlight.microsoft.com
User-Agent: Mozilla/5.0 (X11; Linux i686; rv: 35.0) Gecko/20100101 Firefox/35.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US, en; q=0.5
Accept-Encoding: gzip, deflate
Cookie: WT_FPC=id=1751b390-548c-4a94-af2b-46e80bec960f:lv=1429419163684:ss=1429419163684;
MSFPC=ID=7eecb9d415396f44b6dc83b06186ce85&CS=1&LV=201504&V=1;
MCI=GUID=432ef026e0aee1438ad5299e7062ff46&HASH=26f0&LV=201504&V=4&LU=1429433573409; A=I&I=
AxUFAAAAABqBwAAxj+1QaFEivmldO+Gz1zUyA!!&V=4; MUID=0CEAA694F73661FA16C2A068F33662F9
Connection: keep-alive
Referer: http://www.tivibu.com.tr/ClientBin/Argela.WebTV.Desktop.UI.xap
Content-Type: application/x-www-form-urlencoded
Content-Length: 178
x-playready-info: OSVersion=5.1.2600 p=2 Service Pack 3;
ClientDllVersion=slmsprbootstrap.dll/2.0.1446.0000; ClientMisc=wininet.dll/8.00.7601.17601;
Session=f751ce16f4e4d5b4a5ce7341a0af3a9e; Last-Session=8ddebacadb2595b4371bfe391afa303;
msprdrm_server_exception_compat: false

PostType=DrmsIndivAcquire&ProtocolVersion=1.1.0.0&SecurityVersion=5.0.0.0&Platform=0
&Architecture=0&ClientSdkType=1&ClientId=1o67MYh60AEeAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQak1AAAZDQ*3D
```

After the individualization process, the client player sends a request to DRM application to obtain a valid license. The response from DRM application is shown in Figure 2.21. The client player decrypts the stream with this key and the subscriber watches encrypted stream.

Figure 2.21: License control request

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body>
<GetLicenceResponse xmlns="http://tempuri.org/"><GetLicenceResult>
<?xml version="1.0" encoding="
"utf-8" ?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
><soap:Body><AcquireLicenseResponse xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols"
><AcquireLicenseResult><Response xmlns=
"http://schemas.microsoft.com/DRM/2007/03/protocols/messages"><LicenseResponse xmlns=
"http://schemas.microsoft.com/DRM/2007/03/protocols"><Version>1</Version><Licenses><License>
WE1SAAAAAOfdrGLO61apNDGjTyb+EBAAAMAAQAAASQAAGAEAAAAKAACADYAAAAGAAAAOQAAABjYJ2Z4psK+RI+ICK41WwGnAA
MAAGAAADIAAANAAAAACgABAAEAMgAAAaWAAAAANAAEANAaaaaoAlgAAADMAAAAKAAEAAwAJAAApgABAAoAAAAceSoTI6ktOAKWd
9btp1V33TAABAAMAgLr+9zM6kAcmhvsvkVUfm3gq7FNdkBrDTkpGkWFhLrNUNqXl42dhMfNUj1d8Yuo4zddwSSqGvnOY/A/1os
KjoovvYz4cuKlqPni jQV0CnPhfra9LKR0sVH32ii0IFcwFtF7bpBBQ7IA8BpGtRJULvXksJf+nDREDJs50T9czQQbUAAEACwAA
ABwAAQAQ1P1DUKCYGHRsIX+9KTY+Vw==
</License></Licenses></LicenseResponse></Response></AcquireLicenseResult></AcquireLicenseResponse>
</soap:Body></soap:Envelope></GetLicenceResult></GetLicenceResponse></s:Body></s:Envelope>
```

For the Tivibu Web service, SVOD and TVOD videos are encrypted and then streamed. SVOD is the abbreviation of *subscription based video on demand* and a subscriber can watch this type of videos free of charge, based on their subscription package. TVOD is the abbreviation of *transactional video on demand* and the subscriber has to pay a fee to watch this type of videos. Encrypted videos and encrypted TV streams use the same encryption method and manifest file structure.

For the Tivibu Web service, there are two main functions of watching a stream, the first is rewind and the second is pause live TV. For both functions, the maximum time for rewinding and pausing is 1 hour [12]. Video streams also have “continue from where you have paused” and “continue with other client types” function.

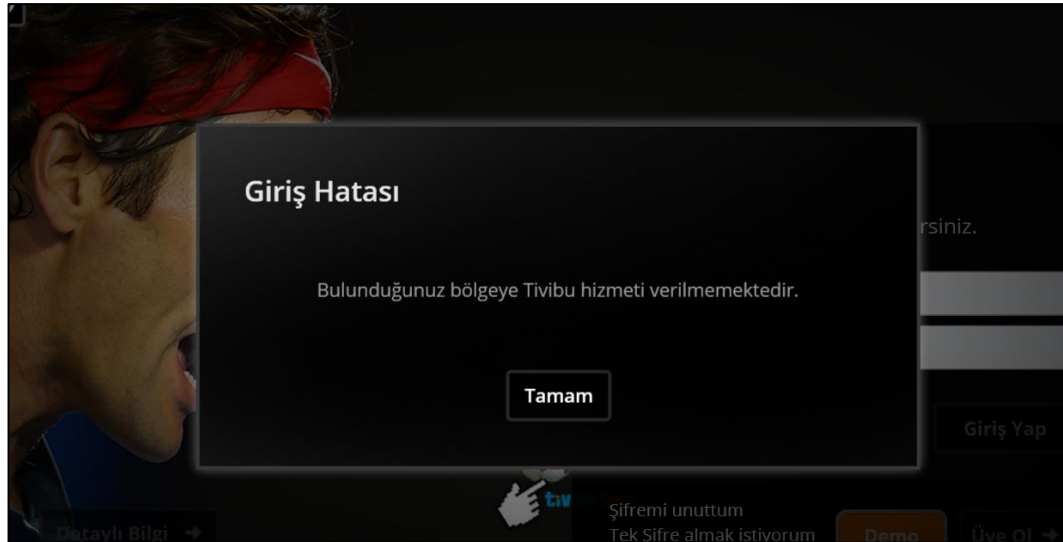
2.6 GEOLOCATION

Tivibu provides service only in Turkey [12]. This is because the content rights are signed to be provided only in Turkey. Tivibu has content rights of UEFA Champions League and UEFA League and matches from these leagues can be watched on Tivibu Spor channel which is provided by Tivibu service.

To simulate our login location as if we were in Turkey, we used Zen Mate add-on developed for Mozilla based browsers. Zen Mate is a proxy/VPN [39] application you can use to simulate your connection as if you were connecting from any other country.

If you login to Tivibu service from outside Turkey, a warning message appears as shown in Figure 2.22.

Figure 2.22: Warning message for geolocation control



If you try to login to the Tivibu Web service from outside Turkey, the server application responds with an error message- “invalid IP”, as shown in Figure 2.23.

Figure 2.23: Invalid IP response

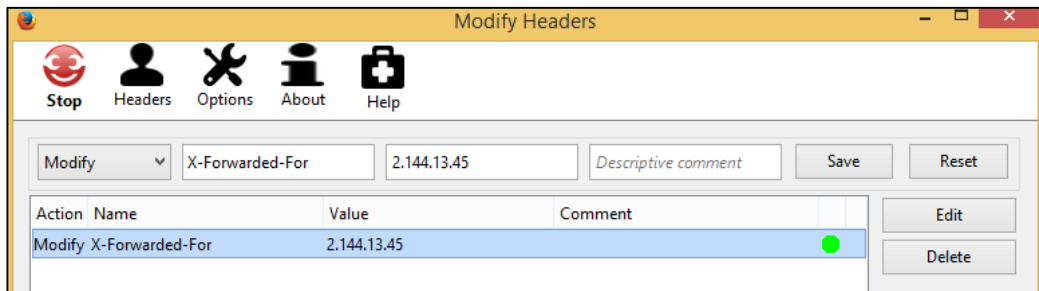
```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body><SecureLoginExtensiveResponse xmlns="http://tempuri.org/">
    <SecureLoginExtensiveResult xmlns:a=
      "http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.AuthSec" xmlns:i=
      "http://www.w3.org/2001/XMLSchema-instance"><a:ReturnDigest>
        62806B6579C9996C5AD0F7CEC5760185C903B6494</a:ReturnDigest><a:ReturnLoginResult xmlns:b=
          "http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Auth"><ErrorMessage xmlns=
            "http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Commons">INVALID_IP
          </ErrorMessage><HasError xmlns=
            "http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Commons">true
          </HasError><b:AccountID>201109275</b:AccountID><b:AccountServiceID>9201109275
          </b:AccountServiceID><b:ChannelInfos i:nil="true" xmlns:c=
            "http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Btv"/>
          <b:DeactivationTime>0001-01-02T00:00:00</b:DeactivationTime><b:Location xmlns:c=
            "http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Location.Model"><c:ID>0
          </c:ID><c:Name>Everywhere</c:Name></b:Location><b:Result>INVALID_IP</b:Result><b:ServerTime>
            2015-04-19T12:35:28.920005Z</b:ServerTime><b:Status>1</b:Status><b:ThematicChannels xmlns:c=
              "http://schemas.datacontract.org/2004/07/Argela.WebTV.IISCache.Managers.Tap.Model"><b:VnoID>1
            </b:VnoID></a:ReturnLoginResult></SecureLoginExtensiveResult></SecureLoginExtensiveResponse></s:Body>
</s:Envelope>
```

According to this test scenario, this control could be bypassed with VPN add-ons or VPN applications, so even if you are at a location outside of Turkey, you can still use

the Tivibu Web service at that location with these add-ons. To solve this problem, major VPN / proxy IPs can be banned from the system.

We can try the same test without a VPN application. To do this, first we have to add another add-on named “Modify Headers” [40] to Mozilla Firefox browser. This add-on helps us change header values for the request sent from the browser. Our aim is to change X-Forwarded-For [41] parameter and see whether the Tivibu Web service is affected by this change. To do this test, we changed X-Forwarder-For header value with an IP from outside of Turkey at the configuration page for “Modify Headers” add-on as shown in Figure 2.24.

Figure 2.24: "Modify Headers" add-on configuration page



When this configuration is activated, we got “Bulduğunuz bölgeye Tivibu hizmeti verilememektedir [Tivibu does not provide service for your region]” error message as the server sends and “Invalid IP” message in response. This test shows that the IP control for the Tivibu Web service is done via the X-Forwarder-To header value. When we did the same test with Netflix service, even when we changed the X-Forwarder-To header value to an IP value from United States, Netflix gave an error message because of our location.

2.7 OUTPUT PROTECTIONS

There are some output protection properties provided by Microsoft PlayReady DRM [42]. These properties are used to prevent illegal movements, such as recording and

copying of the stream. It is also possible with these properties to prevent taking a screenshot of the stream or to not allow video recording.

The Tivibu Web service can be used with a PC client and a browser client. Microsoft PlayReady output protections can only be used for a PC client. Browsers don't have these capabilities. When you try to take a screenshot or do record while watching a stream with Tivibu Desktop Client, screenshot or record are saved as a black screen because of these protections as shown in Figure 2.25.

Figure 2.25: Screenshot taken from Tivibu Desktop client



2.8 WATERMARKING

Even if the service providers take precautions to protect their content by the server or client, these protections cannot prevent users from recording streams from a screen that the stream is being played on. There are two main methods to solve this issue. The easier and cheaper way is the showing of an ID by the client that reveals the identity of the subscriber viewing this screen and this method is called watermarking [43].

The second way is harder and more expensive than the first method. In this method, a unique ID that tells service provider who is using the screen is placed into the stream package. As ID is in the stream package, when you watch the stream on the screen you

cannot see any ID on the screen. Special cameras are needed to read this ID. This method is complicated because stream packages have to be changed for every unique subscriber. In Tivibu Web service, neither of these methods are used as we noticed from our tests. As there is no watermark solution in Tivibu Web, subscribers can easily record streams and service provider is not be able to find who the recorder is.

2.9 TIME DIFFERENCE PERFORMANCE BETWEEN SATELLITE AND TIVIBU WEB STREAM

Time differences between satellite broadcasting and digital broadcasting are an important performance issue for the world of digital TV. Broadcasts from satellites are always ahead of digital streams. This is because, for digital TV, broadcasts taken from satellite are transcoded in transcoders, encoded in encoders and, if it is required, an encryption process is done. All these steps take some time, so there is a time difference between satellite broadcasts and digital streams.

Think about a live football match, you have a digital Web TV subscription and you are watching the football match from your Web TV account, your neighbor is a satellite subscriber and he is also watching the same match. After a while your neighbor screams “goaaaalllll” but you are watching the same match and no goal has been scored yet. After about 30 seconds you see the goal. So, for digital TV solutions this time difference is important. We tested this performance for the Tivibu Web and Digiturk Play services. We compared the time difference for three national channels. The results are shown in Table 2.1. According to a study [45] about digital TV delays, these time differences are not short for live football matches.

Table 2.1: Time difference between satellite broadcasting and Tivibu Web streams

Channel	Time Difference between Tivibu Web Stream and Satellite
TRT 1	+16 seconds
Fox TV	+12 seconds
ATV	+16 seconds

3. DIGITURK

Digiturk is the first digital television platform in Turkey and was founded in 1999. It initiated digital broadcasting services in 2000 [46]. The main service for Digiturk digital platform is Pay TV service which consists of Turkish Super League rights and this service is provided via satellite. The number of subscribers of Digiturk is 2.891.305 [8] according to the second quarter report of Information and Communication Technologies Authority (BTK).

Digiturk Play is another service of Digiturk Platform which serves live TV broadcasts and video streams over the Internet. In this thesis, a comparison is made between the Tivibu Web service and the Digiturk Play service.

The Digiturk Play service provides these functionalities:

1. **TV Broadcasting:** There are a lot of national and international TV channels in the Digiturk Play service. Besides these channels, there are also thematic channels like Turkmax, and sport channels to broadcast Turkish Super League such as LigTV 1, LigTV 2.
2. **Subscription Based Videos (SVOD):** Subscribers can access this type of videos based on their subscription package, there is no additional fee for subscription VODs.
3. **Transactional Videos (TVOD):** Subscribers can watch transactional videos by renting them. They have to pay a fee to rent the video for a certain period of time. After the period expires, the subscriber cannot access the content. Purchasing is operated via service client and fee is withdrawn online from a credit card.
4. **Catchup TV:** This service is based on recording broadcasts of the contracted channels and serving these records as SVOD content. These videos are recorded

by the server. Subscribers cannot choose the programs that will be recorded. Since catchup TV videos are served as SVOD, there is no additional fee to access this type of content.

5. **Pause-Watch / Rewind:** Subscribers can pause live content with this option. After a while, they can continue to watch the stream from the paused point. This option is managed by the server and it has a limit of 12 hours for Digiturk Play.

3.1 TECHNOLOGY

Digiturk Play service uses different solutions for different services such as SVOD, Catchup TV and live broadcasting. For live broadcasting, Digiturk Play uses Octoshape Infinite HD-M [47] solution, but for video streaming, it uses different Third Party CDN solutions based on Microsoft Smooth Streaming.

Digiturk Play doesn't have its own CDN solution. It uses Third Party CDN solutions such as Octoshape or Unified Streaming. The client machine needs Silverlight and Octoshape modules to play Digiturk Play TV or stream videos.

3.2 ACCOUNT CREATION

Membership is required to log in to the Digiturk Play service. There are three types of subscription and only one of them is free with only 4 TV channels in the package [48]. Subscription is done via Digiturk Play Subscription page server at <http://www.digiturkplay.com.tr/club/webtvuyelik/> URL. This page is served under HTTP protocol so the information written on this page is sent to the server as clear text. Since private data, such as Turkish Republic National Identity Number and username are sent to the server from this page, it should be SSL protected. However, the login page is protected by HTTPS protocol. Subscribers can switch to paid packages from the free package using Digiturk Play internet site and all these pages are protected by HTTPS protocol.

As mentioned, HTTPS protocol protects sites from the man-in-the-middle attack types. We conducted a test to check this protection. We used a Burp Suite Proxy server and located it between the browser and Digiturk Play servers. Browsers give a warning message in such situation to warn the user, so the user can understand whether the connection is secure or not. For our test, Firefox browser warned us as the connection was not secure as shown in Figure 3.1, because there was a proxy between the browser and Digiturk Play servers.

Figure 3.1: Browser warns as the connection is not secure



3.3 USER AUTHENTICATION AND SESSION MANAGEMENT

The Digiturk Play service does not have a desktop client like Tivibu Web does. Subscribers login and get service only from the browser client. The internet address of this service is <http://www.digiturkplay.com.tr/Anasayfa/>. Digiturk Play also provides service from mobile clients such as IOS or Android clients. Subscribers use their email addresses as a username to login to the service.

When subscribers enter the Digiturk Play service page, the browser client communicates with the server for user authentication and authorization processes from <https://www.digiturkplay.com.tr/club/login/> URL. This communication is done via HTTPS protocol as shown in Figure 3.2.

Figure 3.2: HTTPS based communication

25	302	HTTPS	www.digiturkplay.com.tr	/dub/login/
26	302	HTTPS	www.digiturkplay.com.tr	/Anasayfa/
27	200	HTTP	www.digiturkplay.com.tr	/Anasayfa/
css{28	200	HTTP	www.digiturkplay.com.tr	/Files/Styles/Style.css?version%20=%20V_2.33

Digiturk Play does not have an XAP file like Tivibu Web, requests are processed and response pages are produced by the server and then sent to the client machine. A request sent from the browser during login operation is shown in Figure 3.3. Digiturk Play uses “ASP.NET_SessionId =zmjwoq55plda2ajf0oh5n345” session to identify the subscriber.

Figure 3.3: Sample request for login operation

```
GET http://www.digiturkplay.com.tr/Anasayfa/ HTTP/1.1
Accept: text/html,application/xhtml+xml,*/*
DNT: 1
Accept-Language: tr,en-US;q=0.7,en;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Pragma: no-cache
Cookie: ASP.NET_SessionId=zmjwoq55plda2ajf0oh5n345; VL_LastPageViewTime=2015-08-08 19:50:10;
VL_LastPVTimeForTD=2015-08-08 19:50:10; VL_TotalDuration=354; VL_FirstVisitTime=2015-08-08 19:44:17;
VL_TotalPV=2; VL_PVCountInVisit=2; VL_VisitStartTime=2015-08-08 19:44:17; VL_TotalVisit=1;
OfferMiner_ID=VPBIDTCQYVJRNEHC20150808194417; OM_INW=1; OMB_New=1;
VL_FirstReferrer=http://www.bing.com/search?q=digiturkplay.com.tr&form=IE10TR&src=IE10TR&pc=LNJB;
OM_q=digiturkplay.com.tr; OM_rDomain=http%3A%2F%2Fwww.digiturkplay.com.tr%2FAnasayfa%2F;
__utma=144279456.1206027910.1439052258.1439052258.1439052258.1; __utmb=144279456.2.10.1439052258;
__utmc=144279456;
__utmz=144279456.1439052258.1.1.utmcsr=bing|utmccn=(organic)|utmcmd=organic|utmctr=digiturkplay.com.tr;
__utmt=1
Host: www.digiturkplay.com.tr
Cache-Control: no-cache
```

When a subscriber clicks a TV link, a request is sent to the server as shown in Figure 3.4. As seen in this request, there is no information like username and the subscriber is followed by an ASP.NET_SessionId= zmjwoq55plda2ajf0oh5n345 session cookie. Tivibu Web service uses HTTPS protocol for all its processes, but for some pages of Digiturk Play service, communication is done via HTTP protocol.

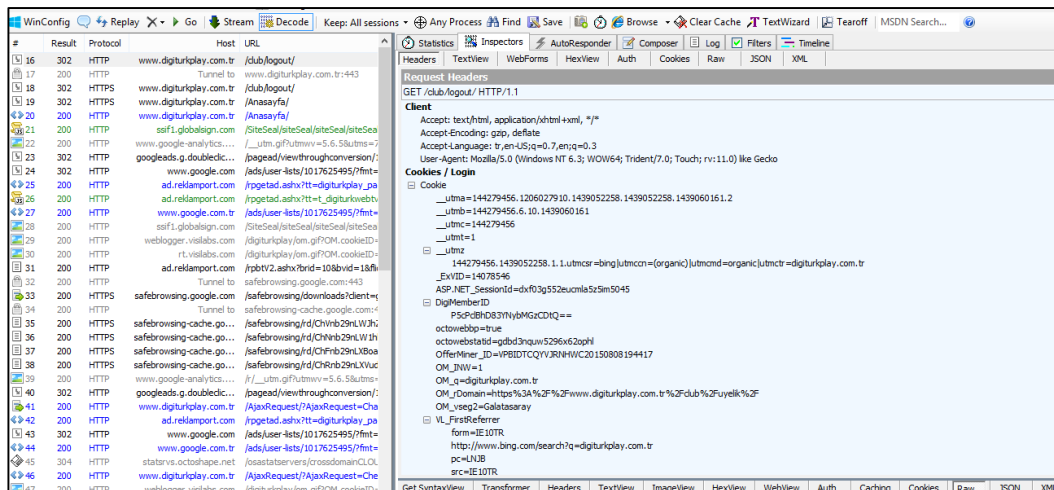
Figure 3.4: Sample request for TV link

```
GET http://www.digiturkplay.com.tr/Canli-TV/Tv/ HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Referer: http://www.digiturkplay.com.tr/Anasayfa/
Accept-Language: tr,en-US;q=0.7,en;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
DNT: 1
Host: www.digiturkplay.com.tr
Cookie: ASP.NET_SessionId=zmjwoq55plda2ajf0oh5n345; VL_LastPageViewTime=2015-08-08 19:51:03;
VL_LastPVTTimeForTD=2015-08-08 19:51:03; VL_TotalDuration=407; VL_FirstVisitTime=2015-08-08 19:44:17;
VL_TotalPV=3; VL_PVCountInVisit=3; VL_VisitStartTime=2015-08-08 19:44:17; VL_TotalVisit=1;
OfferMiner_ID=VPBIDTCQYVJRNHWC20150808194417; OM_INW=1; OMB_New=1;
VL_FirstReferrer=http://www.bing.com/search?q=digiturkplay.com.tr&form=IE10TR&src=IE10TR&pc=LNJB;
OM_q=digiturkplay.com.tr; OM_rDomain=https%3A%2F%2Fwww.digiturkplay.com.tr%2Fclub%2Fuyelik%2F;
EXVID=14078546; OM_vseg2=Galatasaray; __utma=144279456.1206027910.1439052258.1439052258.1439052258.1;
__utmb=144279456.3.10.1439052258; __utmc=144279456;
__utmz=144279456.1439052258.1.1.utmcsr=bing|utmccn=(organic)|utmcmd=organic|utmctr=digiturkplay.com.tr
```

3.4 LOGOUT PROCESS

“Log Out” link is used to logout from Digiturk Play service. When the user clicks this link, the browser sends a request to <http://www.digiturkplay.com.tr/club/logout/> URL. This process is done via HTTPS protocol. Fiddler output of logout operation is shown in Figure 3.5.

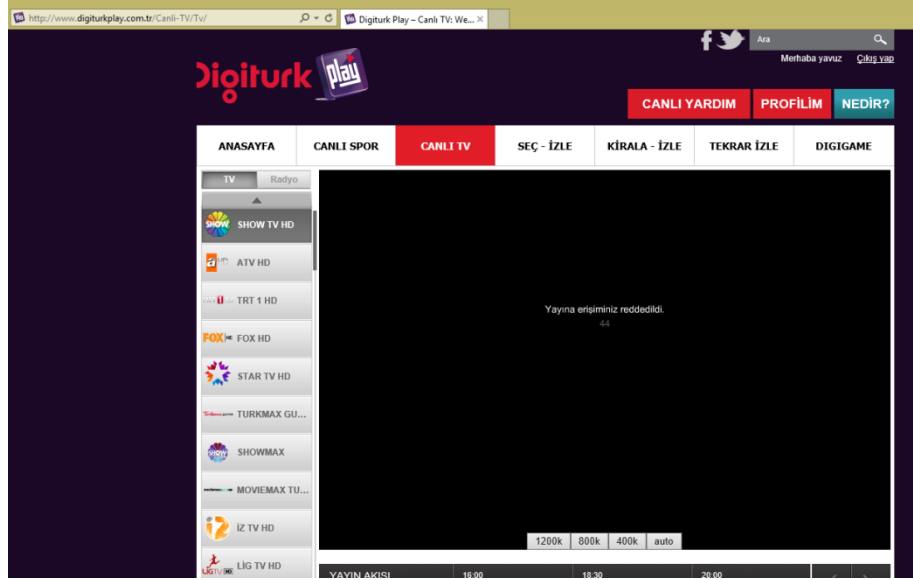
Figure 3.5: Logout Operation



Just like Tivibu Web, Digiturk Play controls if there are multiple sessions for a subscriber. If there is a multiple login with the same username, Tivibu warns the user with a message that includes an explanation as to why the session has been terminated. In the same situation, Digiturk Play service is terminated with a warning message but

message does not include an explanation. It simply states “Yayına erişiminiz reddedildi [You have been denied access to the broadcast]” as shown in Figure 3.6.

Figure 3.6: Warning message for multiple session



3.5 STREAMS

There are two main service categories in Digiturk Play like there are in Tivibu Web: TV channels and videos. Some of these contents are served as clear content and some of them are served as encrypted content. Digiturk does not use its own CDN solution. For live TV broadcasting, Digiturk uses Octoshape’s solution. If you log in to Digiturk Play service for the first time, your browser downloads an Octoshape player from the Octoshape CDN. As shown in Figure 3.7, the Octoshape player is downloaded from <http://cdn.octoshape.net/resources/player/infinitehd3/player.swf> URL.

Figure 3.7: Octoshape Player download link

1	200	HTTP	www.digiturkplay.com.tr	/Canli-TV/Tv/
2	200	HTTP	cdn.octoshape.net	/resources/player/infinitehd2/swfobject.js
5	200	HTTP	ssif1.globalsign.com	/SiteSeal/siteSeal/siteSeal/siteSeal.do?p1=www.digiturkplay.com.tr&p2=S2100-50&p3=image...
6	200	HTTP	www.google-analytics.com	/__utm.gif?utmwv=5.6.6&utms=5&utmh=479006206&utmhn=www.digiturkplay.com.tr&utmc...
7	502	HTTP	Tunnel to	googleads.g.doubleclick.net:443
8	200	HTTP	www.digiturkplay.com.tr	/AjaxRequest?AjaxRequest=ChannelGuide.Json&ChannelID=22&Protocol=http:
9	200	HTTP	ad.reklamport.com	/rpgetad.ashx?tt=digiturkplay_pageskin&rpc=rp_d_digiturkplay_pageskin&async=1&rnd=493...
10	200	HTTP	cdn.octoshape.net	/resources/player/infinitehd3/player.swf
11	200	HTTP	ad.reklamport.com	/rpbTV2.ashx?brid=10&brid=1&fid=9&osid=1&srld=0&dom=digiturkplay.com.tr&url=CanliTV/...
12	200	HTTP	cdn.octoshape.net	/resources/player/infinitehd3/assets/digiturk.xml
13	200	HTTP	d2iwbhgdzdwil.cloudf...	/crossdomain.xml
14	200	HTTP	status.octoshape.net	/assets/crossdomain/CROSSDOMAIN.xml

Digiturk Play uses Octoshape solution for its live TV streams as a CDN solution. Octoshape has its own protocol and architecture [49] to stream live TV broadcasts. The main difference between this solution and an Http Live Stream is that Octoshape uses a UDP [50] protocol and not TCP [51]. Http Live Streaming and Smooth Streaming are TCP based services. Octoshape claims that their solution solves problems of HTTP Live Streaming such as chunk loss or fluctuations of the Internet connection. They claim that TCP is optimized for accurate delivery rather than timely delivery. However, the important issue for digital TV services is quality. We conducted a test to see if their claim is true and in this test we used one laptop, one internet connection and opened two clients at the same time and watched the same TV channels. We saw that this claim is true as Tivibu waited for chunks to be downloaded but Digiturk Play stream continued without delay.

Figure 3.8: Digiturk Play and Tivibu Web Clients on same machine with one internet connection, same stream.

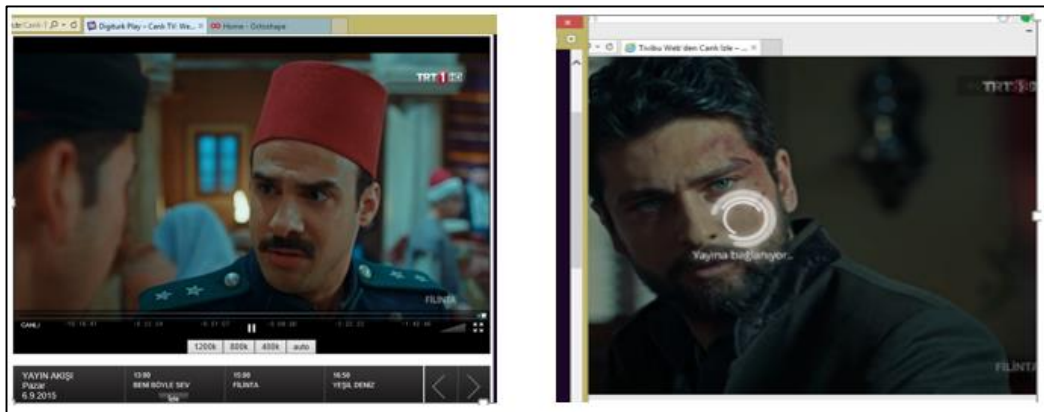


Figure 3.8 shows two clients that are running on the same client machine and using the same internet connection. The Digiturk Play client on the left continues to play the channel but the Tivibu Web client on the right is waiting for TCP chunks to continue.

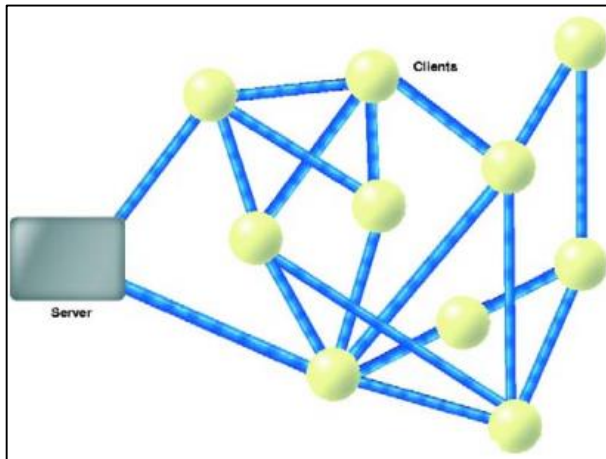
3.5.1 Octoshape's Technology

The main difference between Octoshape's technology and HLS or Smooth Streaming is that Octoshape uses UDP as its data transfer protocol [52]. They claim that their solution is better than TCP based HLS or Smooth Streaming, because, by design, TCP has aggressive throughput back off mechanisms in place to ensure the end user receives data reliably and it sacrifices video quality for reliability. The job of TCP is to make sure every piece of data is received, but it is not designed to sustain a given bit rate (which directly translates to video quality). In this way it is not well suited to deliver video over the Internet.

Actually, HLS is developed to solve these disadvantages of TCP but Octoshape argues that these solutions cause lots of switch operations between bitrates. To replace the reliability characteristics of TCP, Octoshape has implemented patented resilient coding schemes in the data that ensure the video is received at the end user [53].

Other important functionality that Octoshape provides is grid casting [54], Octoshape uses grid casting, which is a stream-sharing system to minimize the load on bandwidth for the broadcaster, the content delivery network, the ISP or the mobile operator. The intention is that each listener relays either a part or all of the stream they download to several other nodes in the grid as shown in Figure 3.9. The system is able to handle peers' leaving the network immediately after switching to another peer that is serving the same portion of multimedia stream.

Figure 3.9: Octoshape's grid technology



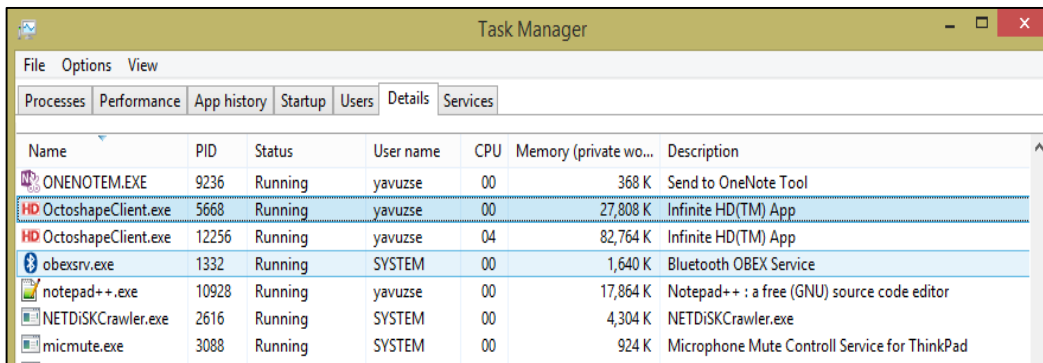
Since Octoshape uses the UDP protocol, its streams cannot be captured like HLS with the Fiddler application. Packages can be seen in Wireshark but as it is a UDP package, packages are not readable as shown in Figure 3.10. When a client machine gets Octoshape UDP packages, local Octoshape client gets them and converts them to RTMP packages and then the client plays the stream [55].

Figure 3.10: Octoshape's UDP packages captured in Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
25829	75.4008150	81.214.130.30	192.168.1.37	UDP	1085	Source port: 8247 Destination port: 8247
25830	75.4060920	88.251.140.17	192.168.1.37	UDP	317	Source port: 8247 Destination port: 8247
25831	75.4173790	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25832	75.4218760	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25833	75.4229300	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25834	75.4249580	176.42.127.107	192.168.1.37	UDP	1085	Source port: 8247 Destination port: 8247
25835	75.4275230	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25836	75.4280290	209.95.50.254	192.168.1.37	UDP	317	Source port: 21082 Destination port: 8247
25837	75.4335880	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25838	75.4360710	176.40.145.251	192.168.1.37	UDP	317	Source port: 8247 Destination port: 8247
25839	75.4363860	192.168.1.37	176.40.145.251	UDP	57	Source port: 8247 Destination port: 8247
25840	75.4398830	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25841	75.4398840	78.165.243.32	192.168.1.37	UDP	317	Source port: 16459 Destination port: 8247
25842	75.4459110	176.240.144.17	192.168.1.37	UDP	317	Source port: 51937 Destination port: 8247
25843	75.4567820	78.175.253.60	192.168.1.37	UDP	60	Source port: 17771 Destination port: 8247
25844	75.4632900	85.110.96.33	192.168.1.37	UDP	317	Source port: 8247 Destination port: 8247
25845	75.4635740	192.168.1.37	85.110.96.33	UDP	57	Source port: 8247 Destination port: 8247
25846	75.4773460	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25847	75.4773500	88.251.140.17	192.168.1.37	UDP	317	Source port: 8247 Destination port: 8247
25848	75.4895720	209.95.50.254	192.168.1.37	UDP	317	Source port: 21082 Destination port: 8247
25849	75.4912660	192.168.1.37	212.252.184.34	UDP	1085	Source port: 8247 Destination port: 8247
25850	75.4925250	176.240.144.17	192.168.1.37	UDP	317	Source port: 51937 Destination port: 8247
25851	75.4970790	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247
25852	75.4970810	209.95.50.254	192.168.1.37	UDP	1085	Source port: 21082 Destination port: 8247

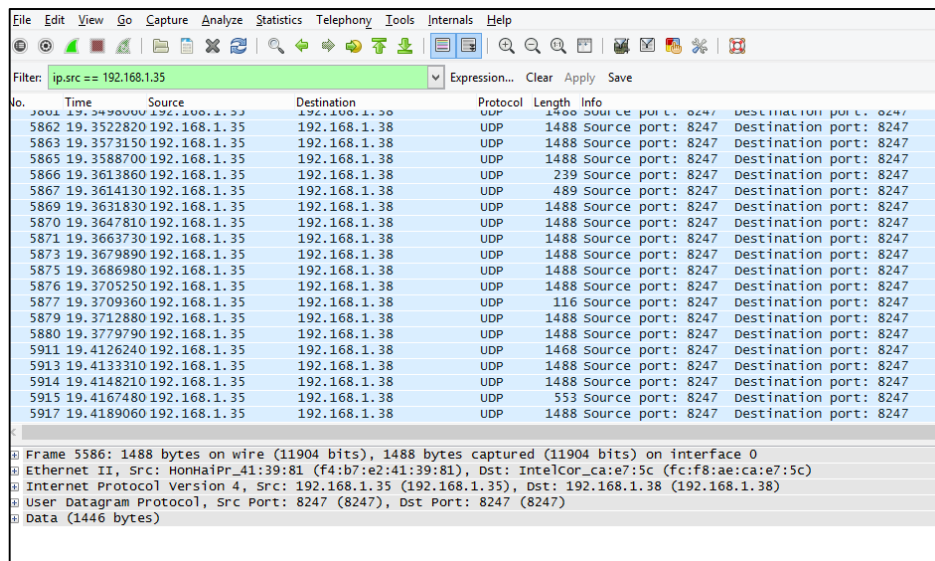
The OctoshapeClient.exe client program, as shown in Figure 3.11, gets these UDP packages and converts them to RTMP format as the player requests the stream in RTMP format.

Figure 3.11: OctoshapeClient.exe



Octoshape's grid casting technology makes it easier for the service provider to reduce bandwidth costs. In this solution, if there is more than one client in the same network, one of the clients gets stream packages from the other client, not from the server. This operation reduces bandwidth usage for the service provider. We tested this scenario with two laptops in the same network and saw that one of the laptops gets the stream from the other laptop as shown in Figure 3.12. IP addresses of our laptops are 192.168.1.35 and 192.168.1.38. The client, whose IP is 192.168.1.38, gets the stream from the laptop with the IP number 192.168.1.35, as shown in Figure 3.12.

Figure 3.12: One of the clients gets stream from another client not server

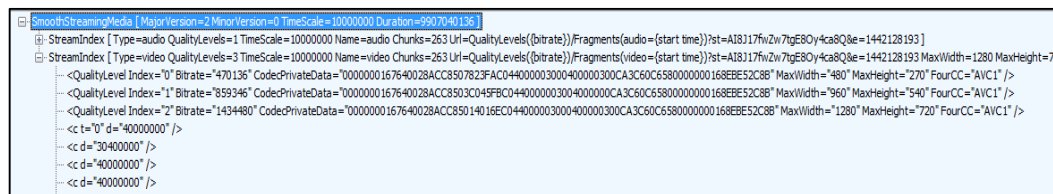


3.5.2 Unencrypted Streams

Digiturk Play uses a different stream infrastructure for TV and video streams. For TV streams, Octoshape applications and CDN solution are used, for video streaming Silverlight-based Microsoft Smooth streaming is used as in the Tivibu Web service.

When we capture a catchup TV video served in Digiturk Play on Fiddler, we see that the first, browser sends a Manifest [30] request to the server.

Figure 3.13: Manifest file for a catchup TV video in Digiturk Play service



```
SmoothStreamingMedia { MajorVersion=2 MinorVersion=0 TimeScale=10000000 Duration=9907040136 }
  StreamIndex [ Type=audio QualityLevels=1 TimeScale=10000000 Name=audio Chunks=263 LH=QualityLevels(bitrate)/Fragments(audio=(start time))?st=A18J17fvZiw7tgE8Oy4ca8Q&e=1442128193 ]
  StreamIndex [ Type=video QualityLevels=3 TimeScale=10000000 Name=video Chunks=263 LH=QualityLevels(bitrate)/Fragments(video=(start time))?st=A18J17fvZiw7tgE8Oy4ca8Q&e=1442128193 MaxWidth=1280 MaxHeight=720 ]
    <QualityLevel Index="0" Bitrate="470136" CodecPrivateData="00000000167640028ACC8507823FAC0440000030004000000300CA3C60C65800000000168EBE52C88" MaxWidth="480" MaxHeight="270" FourCC="AVC1" />
    <QualityLevel Index="1" Bitrate="859346" CodecPrivateData="00000000167640028ACC8503CD49FBC0440000003004000000CA3C60C65800000000168EBE52C88" MaxWidth="960" MaxHeight="540" FourCC="AVC1" />
    <QualityLevel Index="2" Bitrate="1454480" CodecPrivateData="00000000167640028ACC85014016E0C04400000030004000000300CA3C60C65800000000168EBE52C88" MaxWidth="1280" MaxHeight="720" FourCC="AVC1" />
    <t="0" d="40000000" />
    <c d="30400000" />
    <c d="40000000" />
    <c d="40000000" />
```

As shown in Figure 3.13, there is a tag in the Manifest file named “SmoothStreamingMedia” [31]. This tag shows that Digiturk Play uses Microsoft Smooth Streaming format for Catchup TV video streams. There are three bitrate profiles in the Manifest file; 1,36 Mbit, 840Kbit and 460Kbit.

There is no tag about encryption and DRM in the Manifest file, which shows us that Catchup TV videos are not encrypted and are streamed in a clear format. Clear streams are played in the player directly according to the data in the Manifest file. Smooth Streaming packages are received as chunks by the player client. Captured chunks are shown in Figure 3.14.

Figure 3.14: Smooth Streaming Chunks for Catch UP TV videos

#	Result	Protocol	Host	URL	Body	Caching	Content-Ty
30.4		HTTP	dt.ercdn.com	/player/bin/ismf_player_xap	0		
11	200	HTTP	switch.dt.ercdn.com	/clientaccesspolicy.xml	337	max-age=2; Expires...	text/xml; c
12	200	HTTP	switch.dt.ercdn.com	/api/er/Get?ai=636&kak=null&switch=casup&customerid=1&ar=DUB_ERSSNODRM_20001063...	1,219	max-age=2; Expires...	video/x-ms
13	200	HTTP	s2.dt.ercdn.com	/clientaccesspolicy.xml	412	private	text/xml; c
14	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/Manifest?st=8U6E2mAt61FCaZbm03Gcyw&e...	13,960	private	text/xml
15	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/Manifest?st=8U6E2mAt61FCaZbm03Gcyw&e...	13,960	private	text/xml
16	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(470886)/Fragments(video=0)...	148,244	private	video/mp4
17	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=0)?...	49,082	private	video/mp4
18	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=400...	49,687	private	video/mp4
19	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(470886)/Fragments(video=400...	200,279	private	video/mp4
20	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=800...	49,603	private	video/mp4
21	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(470886)/Fragments(video=800...	236,466	private	video/mp4
22	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=120...	49,389	private	video/mp4
23	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(860720)/Fragments(video=12...	435,790	private	video/mp4
24	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=160...	49,367	private	video/mp4
28	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(860720)/Fragments(video=16...	457,831	private	video/mp4
29	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=200...	49,039	private	video/mp4
35	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(860720)/Fragments(video=20...	399,454	private	video/mp4
36	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=239...	49,460	private	video/mp4
37	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=279...	49,382	private	video/mp4
38	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(860720)/Fragments(video=23...	416,493	private	video/mp4
43	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(860720)/Fragments(video=27...	420,753	private	video/mp4
44	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=319...	49,528	private	video/mp4
45	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(860720)/Fragments(video=31...	480,011	private	video/mp4
46	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(95998)/Fragments(audio=359...	48,329	private	video/mp4
47	200	HTTP	s2.dt.ercdn.com	/s2/ss/M/ep/DUB_ERSSNODRM_2000106353.ism/QualityLevels(860720)/Fragments(video=35...	413,792	private	video/mp4

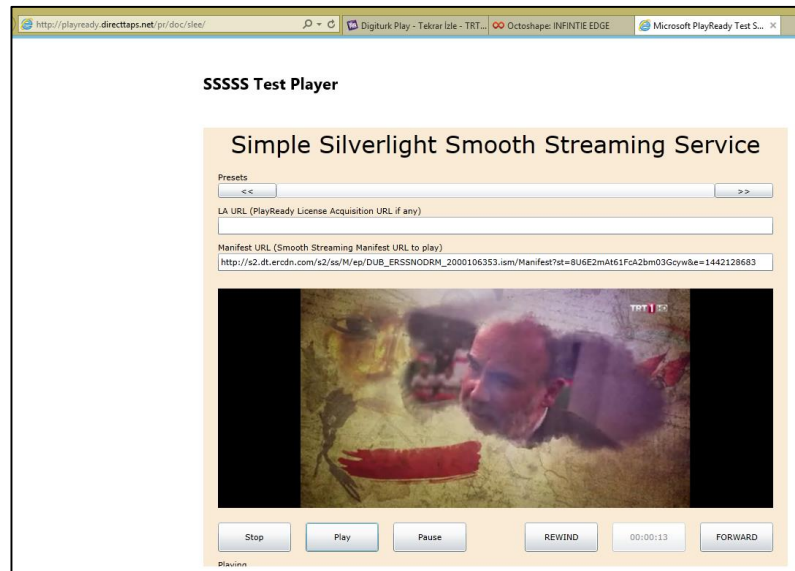
For smooth streaming, audio packages are also received as chunks like video packages. Silverlight client that is embedded to the client player gets these chunks and plays the stream. The value (470886) seen in the video package shows that the stream received by the player is 460kbit at that time.

The most important difference for unencrypted streams between Tivibu Web and Digiturk Play is that Digiturk Play does not use a protection method such as “URL Signing”. URL Signing is a technique used by Tivibu Web clients to authenticate their video stream requests so that unauthorized users cannot receive video streams from Tivibu Web streamers by directly accessing their publishing points using non-WebTV specific software as described in Section 2.5.

As shown in Figure 3.14 chunks are requested as an HTTP request. Anybody can capture these URLs, especially the manifest file URL and then play the stream if there is no URL protection. There are test player pages for Smooth Streaming on the Internet; if you paste the Manifest file onto that sample page, the test player plays the stream if there is no URL Signing protection. We tested this scenario for catchup TV video streamed in Digiturk Play. The video’s name was “Heredot Cevdet” and it was broadcast on the TRT 1 channel. We captured the Manifest request from Fiddler and pasted that link to Microsoft Smooth Streaming page located at

<http://playready.directtaps.net/pr/doc/slee/>. The test player played that stream successfully as shown in Figure 3.15.

Figure 3.15: Microsoft Smooth Streaming Test Player with Digiturk Play Catchup TV video Manifest File

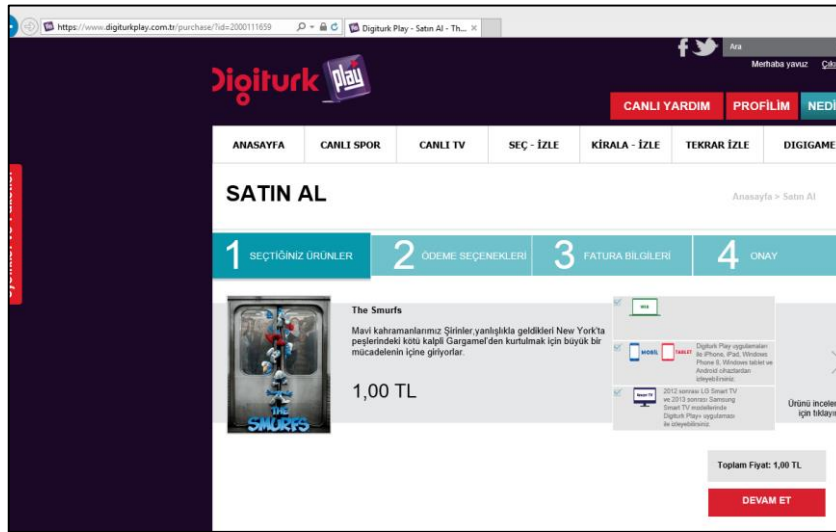


We also tested this scenario connecting from outside of Turkey through a VPN application. With this scenario we succeeded to play the stream. This shows that there is no geolocation protection for stream URLs. Other test that we did with the same Manifest URL is to open multiple pages of this test web site and call that Manifest URL multiple times, this scenario is also succeeded. This shows that there is no multiple session control for stream URLs for Digiturk Play.

3.5.3 Encrypted Streams

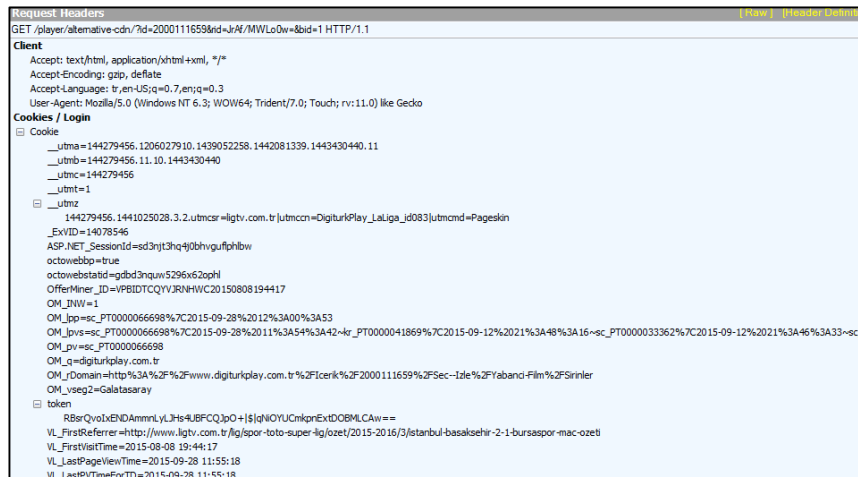
We rented a video to test Digiturk Play's encrypted video streams from its web site. Rent operations can be done with credit card via Digiturk Play web site. All traffic for rent operation is protected by the HTTPS protocol.

Figure 3.16: Digiturk Play video renting page



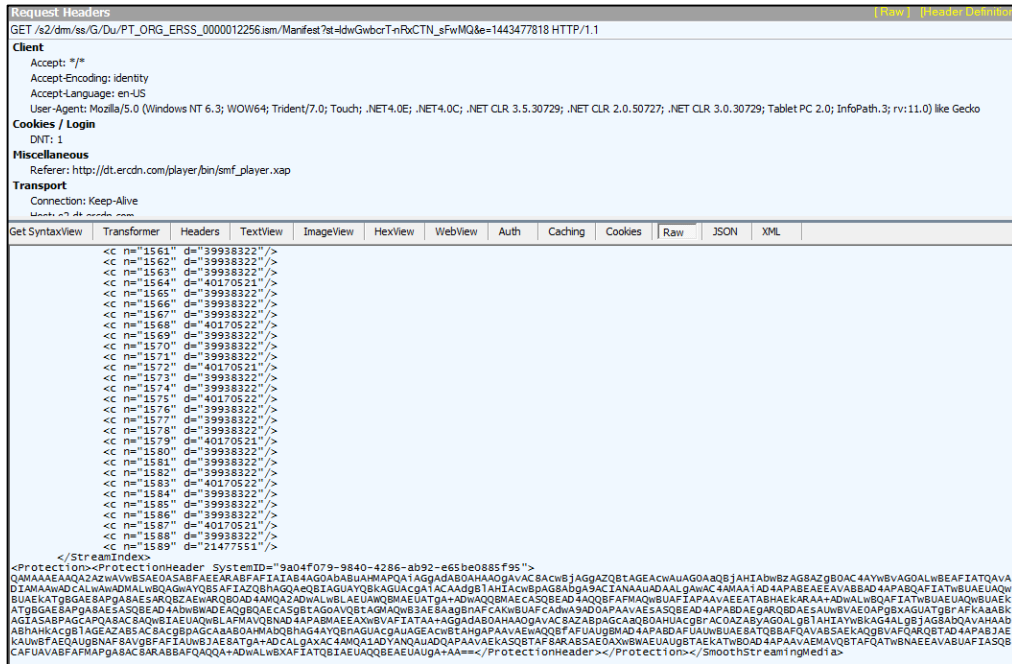
The expiry time for a rented video is 7 days for Digiturk Play. This time is 2 days for the Tivibu Web service. When you play your rented video, Digiturk Play opens a new browser window. The requested URL to stream the rented video is <http://www.digiturkplay.com.tr/player/alternative-cdn/?id=2000111659&rid=JrAf/MWL0w=&bid=1>. The stream does not play when a subscriber tries to connect to this URL from another browser page.

Figure 3.17: Request details for rented video stream start



As Digiturk Play uses Smooth Streaming technology, there should be a Silverlight support in the client machine. Transactional videos are streamed as encrypted for Digiturk Play. The manifest file is shown in Figure 3.18.

Figure 3.18: Manifest for encrypted video in Digiturk Play



As shown in Figure 3.18, the manifest file has a `<Protection>` tag. This tag shows that this stream is encrypted. In this tag, there is another tag named `<ProtectionHeader>`. `ProtectionHeader` tag is base64 hashed value; this tag consists of data about encryption. The decoded version of `ProtectionHeader` for our example is shown in Figure 3.19.

Figure 3.19: Decoded ProtectionHeader value



The `ProtectionHeader` data shows us that the Digiturk Play system uses Microsoft PlayReady version 4.0, similar to DRM applications like Tivibu Web.

Data in `ProtectionHeader` shows us:

- I. The stream is encrypted because there is a `<Protection>` tag in `ProtectionHeader`.

- II. The stream is encrypted by the Microsoft PlayReady application. (from the tag <WRMHEADER xmlns="http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader" version="4.0.0.0">
- III. KEYLEN tag reveals that KEY parameter has a 16-byte length (128 bit)
- IV. The encryption is done in the AES Counter Mode. (from the tag ALGID)

As it is seen from this header information, Digiturk Play and Tivibu Web services use the same encryption method for encrypted videos. However, there is a difference; the DRM system of Tivibu Web is managed by TTNET but the Digiturk DRM system is managed by a third party solution. As seen from LA_URL parameter (<http://digiturk-drm.ercdn.com/playready/rightsmanager.asmx>) the DRM system is managed by ercdn.com domain.

The bit rates for encrypted videos are a bit larger compared to the unencrypted videos. As shown in Figure 3.20, the bit rates for the encrypted videos are 1,38 Mbit, 820 Kbit, and 450 Kbit:

Figure 3.20: Bit rate values for encrypted videos

```
<?xml version="1.0" encoding="utf-8"?><SmoothStreamingMedia MajorVersion="2" MinorVersion="0" Duration="5907560000">
  <StreamIndex Type="video" QualityLevels="3" Chunks="1590" Url="QualityLevels({bitrate})/Fragments(video={start time})">
    <QualityLevel Index="0" Bitrate="1448781" FourCC="H264" MaxWidth="1280" MaxHeight="720"
      CodecPrivateData="00000000167640028ACC85014016EC044000003000400000300CA3C60C6580000000168EBE152C8B0"/>
    <QualityLevel Index="1" Bitrate="867742" FourCC="H264" MaxWidth="960" MaxHeight="540"
      CodecPrivateData="00000000167640028ACC8503C045FBC044000003004000000CA3C60C658000000000168EBE152C8B0"/>
    <QualityLevel Index="2" Bitrate="473210" FourCC="H264" MaxWidth="480" MaxHeight="270"
      CodecPrivateData="00000000167640028ACC8507823FAC044000003000400000300CA3C60C65800000000168EBE152C8B0"/>
  </StreamIndex>
</SmoothStreamingMedia>
```

3.6 GEOLOCATION

The Digiturk Play service is available only in Turkey [56]. Digiturk owns the content rights of the Turkish Super League, the British Premier League, and the Spanish La Liga. Therefore, football matches from these leagues can be watched on the Lig TV channel, which is provided by Digiturk services, but the same content can be served in other countries with different fees or subscription packages. Because of this, the location of the login place should be controlled for these kinds of services.

As we have done for Tivibu Web, to simulate our login location as one outside of Turkey, we used the Zen Mate add-on developed for Mozilla-based browsers. Zen Mate is a proxy/VPN application with which you can simulate your connection as if connecting from any other country. If you login to Digiturk Play service from outside Turkey, a warning message is given as shown in Figure 3.21. After this message, the site forwards you to Digiturk’s site that serves subscribers reaching it from outside Turkey.

Figure 3.21: Warning message of Digiturk Play’s location control



We repeated the same test with the “Modify Headers” add-on for Mozilla Firefox. If you install this add-on, you can modify the X-Forwarded-For value for your requests. When we activated this add-on and changed the X-Forwarded-For value as if it were an IP located outside of Turkey, the Digiturk Play site continued operating as if I were in Turkey. This shows that Digiturk Play does not use the X-Forwarded-For value to ascertain whether the subscriber is in Turkey or not.

3.7 OUTPUT PROTECTIONS

There are a number of output protection functions provided by Microsoft PlayReady DRM. These functions are used to prevent the illegal recording and copying of the stream. With these functions, it is also possible to prevent the taking of screenshot from

the stream or to not allow the recording of any videos. [42]. But these functions are not supported by browsers yet.

The Digiturk Play service does not have a native player for desktops. It has only browser and mobile clients, so Digiturk Play does not use these output protection functions. We could not get a screenshot from the Tivibu Web Desktop client, but we were able to get it in Digiturk Play as it uses a browser client.

3.8 WATERMARKING

The illegal broadcasting of Digiturk's Lig TV is a very critical issue for Digiturk. Digiturk tries to track illegal broadcasts in legal manners [57]. Digiturk uses the watermark method to track down illegal broadcasts.

As mentioned at 2.8 there are two methods to print a unique ID over the stream of the client. The easier and less costly way is to show an ID at the client side that reveals the subscriber using this screen, and the second way is harder and costlier than the first method. In this method, a unique ID that informs the service provider about who is currently viewing the screen is put in the stream package. As the ID is in the package, when you watch the stream on the screen, you cannot see any ID on the screen. Special cameras are needed to read this ID. This method is hard and costly because for every unique subscriber, stream packages have to be changed. Therefore, Digiturk uses the first method. We will try to bypass this protection method at Section 4.

3.9 TIME DIFFERENCE PERFORMANCE BETWEEN SATELLITE AND DIGITURK PLAY STREAMS

The time difference between satellite broadcasting and digital broadcasting is an important performance issue for Digiturk Play as it is a digital TV solution. The time difference was about 15 seconds for Tivibu Web as we tested in Section 2.10. When we calculated the time difference for Digiturk Play, it was seen that the performance of

Digiturk Play was far worse than Tivibu Web. We tested this performance for three national channels. The results are shown in Table 3.1.

Table 3.1: Time difference between satellite broadcasting and Digiturk Play streams

Channel	Time difference between satellite and Digiturk Play
TRT 1	+51,8 seconds
Fox TV	+51,6 seconds
ATV	+50 seconds

4. HACKING FLASH PLAYER OF DIGITURK PLAY

In this section, we will try to bypass the watermarking protection method of the Digiturk Play service. As Tivibu Web does not use the watermarking method, only the Digiturk Play service will be examined in this section.

As it is explained in Section 3.8, Digiturk uses the watermark protection method to follow the illegal copies of their broadcasts. In our tests, we bought a video film and a live match event but a watermark value did not appear on screen. But when we examined the decompiled codes for Digiturk Play's flash player, we saw that the player supports watermarking.

In this section we will try to manipulate the appearance of the watermark value on Digiturk Play's flash player screen. First of all, we have to find out where the flash player is downloading from. The download link can be seen in the source code of the page as shown in Figure 4.1.

Figure 4.1: Source code of Digiturk Play player web page

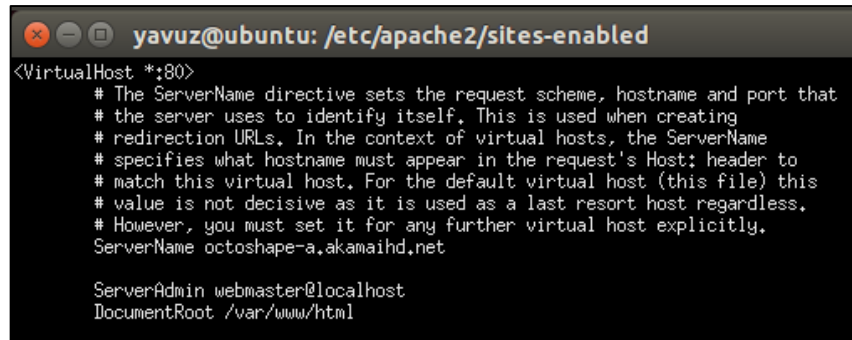
```
<div id="LivePlayerContent">
  <div id="LivePlayerContent_Cell_0_0" class="LivePlayerContent_Row_0">
    <div id="oplayer">
      <script src="http://cdn.octoshape.net/resources/player/infinitehd2/swfobject.js" type="text/javascript"></script>
      <script type="text/javascript">var player_id = "OctoShape";var player_width = "766"</script>
      <object id="OctoShape" width="768" height="485" type="application/x-shockwave-flash" name="OctoShape" data="http://octoshape-a.akamaihd.net/eps/players/infinitehd4/player.swf" style="visibility: visible;">
        <param name="allowFullScreen" value="true"></param>
        <param name="scale" value="noscale"></param>
        <param name="allowScriptAccess" value="always"></param>
      </object>
    </div>
  </div>
</div>
```

As seen from Figure 4.1, Digiturk Play's flash player is downloaded at <http://octoshape-a.akamaihd.net/eps/players/infinitehd4/player.swf>. Every time a user enters this website, their flash player is downloaded from the client machine. So, the first step of hacking is to control this download process.

We need a web server to control this download process. The web server will serve for the "octoshape-a.akamaihd.net" domain. We used the Apache Web Server as our web

server on a Linux virtual machine and configured it to serve for the “octoshape-a.akamaihd.net” domain as shown in Figure 4.2.

Figure 4.2: Apache Web Server configuration

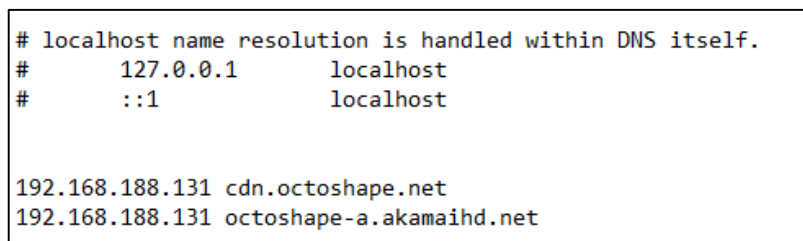


```
yavuz@ubuntu: /etc/apache2/sites-enabled
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName octoshape-a.akamaihd.net

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
```

The Apache Web Server is installed on a virtual Linux machine on our laptop. We have to make another change to download flash player from our system when we enter Digiturk Play’s website. We have to change “C:\Windows\System32\drivers\etc\hosts” file for our laptop to change the result of the DNS query. The DNS query result is overridden with this change. To do this, we type “192.168.188.131 octoshape-a.akamaihd.net” in our host’s file as shown in Figure 4.3. After this entry, our laptop sends the requests for the “octoshape-a.akamaihd.net” domain to our Linux machine as its IP address is 192.168.188.131. This is the first part of the hacking process. Now, when we enter Digiturk Play’s website, the player will be downloaded from our Apache Web Server.

Figure 4.3: Hosts file entries



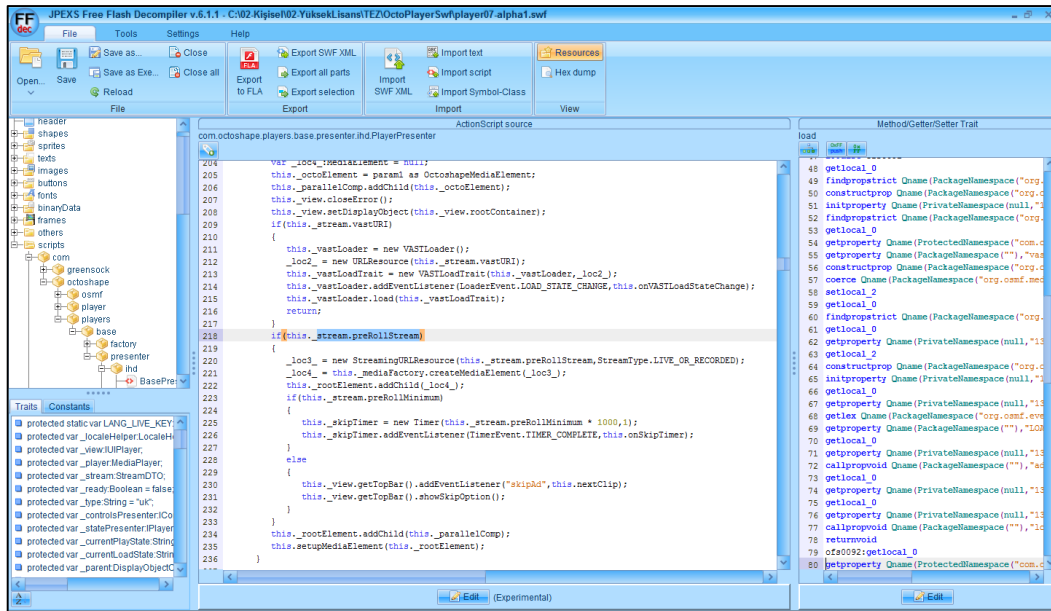
```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost

192.168.188.131 cdn.octoshape.net
192.168.188.131 octoshape-a.akamaihd.net
```

The second part of the hacking process is to change the code of the “player.swf” file. We need a decompiler to do this. We used “JPEXS Free Flash Decompiler” software to

decompile the flash player code. After the decompiling process, the assembler code of a swf file can be edited with this tool.

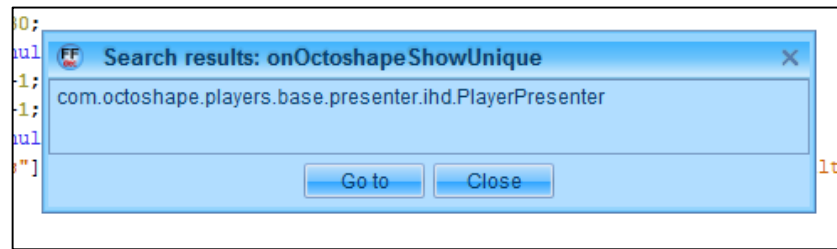
Figure 4.4: JPEXS Free Flash Decompiler



There are two main windows in “JPEXS Free Flash Decompiler” tool, the left one is for the class code, whereas the right one is for the assembler code. We can change only the assembler code; it is not possible to change and compile the class code with this tool. We will carry out two operations on this swf file. As we do not know when Digiturk will populate watermark for a stream, we will push our own watermark through first, then we will demonstrate that we can disable it.

In the Octoshape documentation, it is said that “If you find that someone is doing a screengrab of the stream, you can then inject the special cue point called *onOctoshapeShowUnique*, the player will then show the token unique value in the player viewport of every active viewer.”[58] So, we searched for the term “onOctoshapeShowUnique” in the code as shown in Figure 4.5.

Figure 4.5: Search results for onOctoshapeShowUnique term



This term is located in “com.octoshape.players.base.presenter.ihd.PlayerPresenter” class and used in “onNewNetStream” function as shown in Figure 4.6.

Figure 4.6: Code for onNewNetStream function

```
private function onNewNetStream(param1:OctoshapeNetStreamEvent) : void
{
    var _loc3_:String = null;
    var _loc4_:Array = null;
    this._ns = param1.netStream;
    NetClient(this._ns.client).addHandler("onOctoshapeTriggerAd", this.onOctoshapeTriggerAd);
    NetClient(this._ns.client).addHandler("onOctoshapeShowUnique", this.onOctoshapeShowUnique);
    NetClient(this._ns.client).addHandler("onOctoshapeDisconnectUnique", this.onOctoshapeDisconnectUnique);
    this._ns.addEventListener(NetStatusEvent.NET_STATUS, this.onNetStatus);
    var _loc2_:Dictionary = JSBridgeHelper(OctoshapeCache.config.getJSBridgeHelper()).cuepoints;
    for(_loc3_ in _loc2_)
    {
        _loc4_ = JSBridge.instance.addCuePointFunc(_loc3_, _loc2_[_loc3_]);
        NetClient(this._ns.client).addHandler(_loc3_, _loc4_[_loc3_]);
    }
}
```

As it is seen from Figure 4.6, the watermark event is triggered by a streamer. So, to respond to this event, we have to activate this function. To do this, first we have to find and examine the “onOctoshapeShowUnique” function.

Figure 4.7: Code of onOctoshapeShowUnique function

```
private function onOctoshapeShowUnique(param1:Object) : void
{
    var _loc2_:Number = 30;
    var _loc3_:String = null;
    var _loc4_:Number = -1;
    var _loc5_:Number = -1;
    var _loc6_:String = null;
    if(param1["parameters"]["filter"] && this._stream.stream.indexOf(param1["parameters"]["filter"]) < 0)
    {
        return;
    }
    _loc6_ = this.getOctoshapeMediaElement().getAuthUniqueValue();
    if(!_loc6_)
    {
        return;
    }
    if(param1["parameters"]["ttl"])
    {
        _loc2_ = param1["parameters"]["ttl"];
    }
    if(param1["parameters"]["position"])
    {
        _loc3_ = param1["parameters"]["position"];
    }
    if(param1["parameters"]["xper"])
    {
        _loc4_ = param1["parameters"]["xper"];
    }
    if(param1["parameters"]["yper"])
    {
        _loc5_ = param1["parameters"]["yper"];
    }
    this._view.showUnique(_loc6_,_loc2_,_loc3_,_loc4_,_loc5_);
}
```

As it is seen from the “onOctoshapeShowUnique” function, the player calls “this._view.showUnique(_loc6_,_loc2_,_loc3_,_loc4_,_loc5_);” function to show the watermark as shown in Figure 4.7, so we have to get this call to show the watermark manually. This function gets five parameters; three of them are important for us. The “_loc6_” parameter is for the text that will be shown as the watermark, the “_loc2_” parameter is for the ttl value (ttl shows how many seconds watermark will be shown on the screen) and the “_loc3_” parameter is for the place of watermark on the screen (“C” for center). To determine the value of “_loc6_” variable, “this.getOctoshapeMediaElement().getAuthUniqueValue();” function is used. This function generates a unique id for each user. According to these information, our call is shown in Figure 4.8.

Figure 4.8: Class code for our call

```
var _loc3_:String = null;
_loc3_ = this.getOctoshapeMediaElement().getAuthUniqueValue();
this._view.showUnique(_loc3_,15,"C",-1,-1);
```

As we mentioned before, we cannot change the class code for the swf file, so we have to write an assembler code for this call. Figure 4.9 shows the assembler code for this call.

Figure 4.9: Assembler code for this_view function

```
getlocal_0
pushnull
coerce_s
setlocal 3
getlocal_0
callproperty QName(PrivateNamespace(null,"13"),"getOctoshapeMediaElement") 0
callproperty QName(PackageNamespace(""),"getAuthUniqueValue") 0
setlocal 3
getlocal_0
getproperty QName(ProtectedNamespace("com.octoshape.players.base.presenter.ihd:PlayerPresenter"), "_view")
getlocal 3
pushbyte 15
pushstring "C"
pushbyte -1
pushbyte -1
callpropvoid QName(Namespace("com.octoshape.player.standard2.view.components:UIPlayer"),"showUnique") 5
getlocal_0
getproperty QName(ProtectedNamespace("com.octoshape.players.base.presenter.ihd:PlayerPresenter"), "_view")
callpropvoid QName(Namespace("com.octoshape.player.standard2.view.components:UIPlayer"),"layout") 0
```

The last process that we have to carry out is to find the right place to call this function. We called this function in the “onNewBw” function, so the player will show the watermark value for every bandwidth switch operation. Figure 4.10 shows the class code equivalence for the assembler code that we added.

Figure 4.10: Calling this._view.showUnique function

```
private function onNewBw(param1:OctoshapeStreamInfoEvent) : void
{
    this._view.streamData = {
        "bitrate":param1.newBitrate,
        "stream":param1.newStreamName,
        "index":param1.newPlayingIndex
    };
    JSBridge.instance.doCallback(JSBridge.ONBITRATESWITCH,this._stream.
var _loc2_:DisplayObjectTrait = DisplayObjectTrait(this.getOctoshap
var _loc3_:String = null;
_loc3_ = this.getOctoshapeMediaElement().getAuthUniqueValue();
this._view.showUnique(_loc3_,15,"C",-1,-1);
this._view.layout();
}
```

Now, we can save the swf file and serve it from our Apache Web Server. After doing this process, the result is shown in Figure 4.11. The value printed on the screen is used to know who the user is. In our test, the value of watermark printed on the screen is “ed21332046”. There should be a relevance between our user and this value as it will be used to know who the user is.

Figure 4.11: Digiturk Play's player with watermark



When we trace the traffic between the player and the server with Wireshark, it is seen that this value is a unique value for our test user, as shown in Figure 4.12. That value is sent to our client as the response of “/AjaxRequest/GetOctoshapeTicket” request with parameter “unique” as shown in Figure 4.12.

Figure 4.12: Response for “/AjaxRequest/GetOctoshapeTicket” request

```
GET /AjaxRequest/GetOctoshapeTicket/?rand=424163017728&octoshapeAuthid=%2D33308
%2FCanli%2DTV%2FTV%2F&octoshapestream=octoshape%3A%2F%2Fstreams%2Eoctoshape%2En
Host: www.digiturkplay.com.tr
User-Agent: Mozilla/5.0 (windows NT 6.3; wow64; rv:42.0) Gecko/20100101 Firefox
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: tr-TR,tr;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://octoshape-a.akamaihd.net/eps/players/infinitehd4/player.swf
Cookie: VL_LastPageviewTime=2015-12-05 19:47:00; VL_TotalDuration=18133; VL_Fir
OM_INW=1; __utma=144279456.965774587.1447656187.1449326386.1449334913.16; __utm
2Fwww.digiturkplay.com.tr%2FAnasayfa%2F; _ExVID=14078546; OM_vseg2=Galatasaray;
OM_lpv=cnl_0%7C2015-11-27%2017%3A37%3A55; ASP.NET_SessionId=2eqjvk45rchuqy45n4;
__utmb=144279456.17.10.1449334913; token=RBSrQvoIXENDAmnLyLJHs4UBFCQJpo+|$_qN1
Connection: keep-alive

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Sat, 05 Dec 2015 17:56:08 GMT
Content-Length: 50

715768be0b62b254ddc915a364e19d79
unique=ed21332046
```

It is also possible to print any string as watermark value by changing the “_loc3_” parameter. In Figure 4.13, it is shown that we can also change the watermark value.

Figure 4.13: Digiturk Play’s player with our watermark



We have succeeded in viewing the watermark manually. Now we can try to disable it. To do this, we have to examine “showUnique” function. As it is seen from Figure 4.14, the showUnique function calls another function named “newUnique”.

Figure 4.14: Code for showUnique function

```
public function showUnique(param1:String, param2:Number, param3:String, param4:Number, param5:Number) : void
{
    this._showUnique.newUnique(param1,param2,param3,param4,param5);
}
```

The code of the “newUnique” function is shown in Figure 4.15. The most important part of the code for us is the “txtField.alpha=1.0” line. In this line, the transparency of the watermark is set, so if we change this value to 0, the watermark will be disabled.

Figure 4.15: Code for newUnique function

```
public function newUnique(param1:String, param2:Number, param3:String, param4:Number = -1, param5:Number = -1) : void
{
    var obj:MovieClip = null;
    var txtField:TextField = null;
    var unique:String = param1;
    var ttl:Number = param2;
    var position:String = param3;
    var xper:Number = param4;
    var yper:Number = param5;
    obj = new MovieClip();
    obj.name = unique;
    if(position)
    {
        position = position.toUpperCase();
    }
    if(position == "RANDOM")
    {
        position = this.positions[Math.floor(Math.random() * (1 + (this.positions.length - 1)))];
    }
    obj["position"] = position;
    obj["xper"] = xper;
    obj["yper"] = yper;
    txtField = new TextField();
    txtField.name = "text";
    obj["text"] = txtField;
    obj.addChild(txtField);
    txtField.alpha = 1.0;
    txtField.selectable = false;
    txtField.text = unique;
    var timer:Timer = new Timer(ttl * 1000,1);
    obj["timer"] = timer;
    timer.addEventListener(TimerEvent.TIMER_COMPLETE, function():void
    {
        mClips.splice(mClips.indexOf(obj),1);
        obj.removeChild(txtField);
        txtField = null;
        removeChild(obj);
        obj = null;
        timer = null;
    });
    this.mClips.push(obj);
    this.updateLayout(null);
    addChild(obj);
    timer.start();
}
```

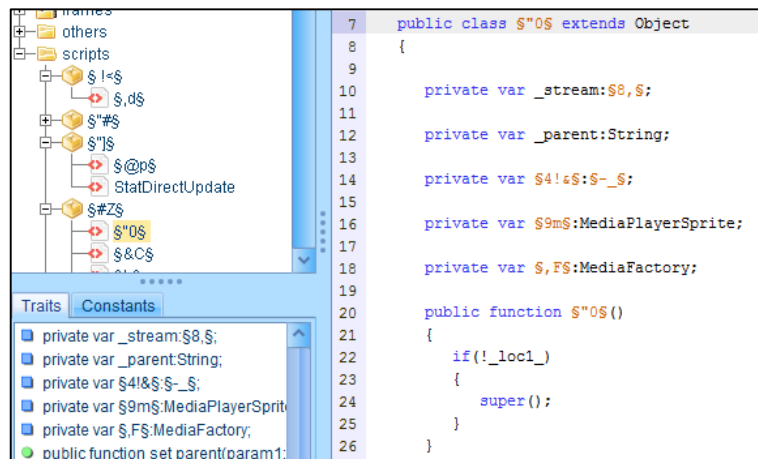
To see the affect of this change, we set alpha value to 0.2. Result of this process is shown in Figure 4.16. If we set alpha value to 0, watermark value will not be shown. Digiturk Play is showing advertisements on the stream using same way, it could be passed like watermark, too.

Figure 4.16: Watermark with the alpha value 0.2



We did this hacking by searching function names in decompiled code. If Digiturk Play's player file is obfuscated it would not be easy to find where watermark functions are located, because obfuscate process changes names of variables and functions. Obfuscation means to make code harder to understand or read, generally for privacy or security purposes. Figure 4.17 shows an example of an obfuscated code.

Figure 4.17: Obfuscated code example



5. COMPARATIVE ANALYSIS AND BEST PRACTICES

In the previous sections, we have analyzed the Tivibu Web and the Digiturk Play Internet TV services based on technology and security mechanisms that they use, and the differences between their solutions as well as the setbacks of these solutions. For this analysis, we have focused on the points described below:

1. Web site security for account creation, authentication and authorization flows
2. Protecting contents with DRM and granting authorization
3. DRM solution details
4. Output protection methods to protect contents from being copied or recorded, and how to hack these methods
5. Geographical restrictions
6. Client capabilities

We analyzed each of these aspects using browser-based clients and we also evaluated the Desktop client of Tivibu Web. According to the results of these analyses and examinations, our assessments and best practices are listed below:

5.1 CLASSIFICATION BASED ON SOLUTION ARCHITECTURE

When we assess these two services, it is seen that Tivibu Web is more holistic than the Digiturk Play service. Tivibu Web uses the same streaming format for all its sub-services like TV and video. Tivibu also has its own CDN installation located in Turkey and manages this infrastructure itself.

Digiturk Play has different sub-solutions for TV and video streams. For TV streams, the Octoshape solution is used, but for video streams they chose Microsoft Smooth Streaming. This difference makes me feel that the Digiturk Play solution is unsteady. As mentioned above in Section 3.5.1, Octoshape's protocol seems very successful but it is not preferred for video streaming by Digiturk. We do not know the exact reason of this

choice, but since Digiturk uses different third party CDN companies, the reason might be a financial one.

Digiturk Play does not have its own CDN. Using third party CDN solutions is common in this sector, but the interesting point is that, they use different CDN companies for catchup TV videos and other videos, and this does not result in an optimum situation for managing and supporting the service. This also supports our assessment about Digiturk Play's solution that it is unsteady.

The cost of operations for such services is very important for companies. If a company has different types of solutions, that company will need more engineers to operate those systems and a greater budget to buy support from each vendor. For easy management and less costs, companies may choose to work with only one vendor.

5.2 STREAM PROTECTION

In terms of security, both solutions seem to be stable. There is only one point that we found out about Digiturk Play; Digiturk Play does not use a URL Signing solution to protect clear or encrypted stream URLs against being used by people apart from the clients themselves. Catchup TV videos are streamed unencrypted by Digiturk Play and if any user obtains the manifest URL they can watch that stream on any Silverlight player. Tivibu Web has a URL Signing solution as described in Section 3.5.2 whether the stream is clear or encrypted.

TV Service providers should use a URL Signing protection method even when the stream is unencrypted. One of the biggest costs for TV service providers is data usage costs, so it is very important to ensure that only authorized clients are receiving the streams.

5.3 LOAD BALANCING

One shortcoming of Tivibu Web is that after the login process, two CDN IPs are returned to the subscriber. This shows that there is not a common Load Balancer in front of all CDNs. Even though there is little likelihood that two CDNs may be down at the same time, this is a point that needs to be improved. Digiturk Play uses one CDN URL for each CDN company.

5.4 WEB SITE SECURITY

Most of the processes such as account creation or login are done via the HTTPS protocol for both services. There is one exception, however, at the first step of account creation: Digiturk Play's website sends identity number and other data to the server via the HTTP protocol. This flow should be moved to HTTPS.

Both WebTV providers rely on clients deleting the cookies from the browser's cookie store to sign out the user. But if the subscriber does not click on "logoff", the browser stores the session cookie and next time the subscriber can use the service without having to login for Digiturk Play. For Tivibu Web, even if the subscriber does not click "logoff", the session cookie expires, so the subscriber should enter his/her login credentials to log in to the service every time.

5.5 DRM

For both services, Microsoft PlayReady DRM application is used to encrypt valuable contents. TNET manages its own DRM solution, whereas Digiturk uses third party CDN companies for the DRM solution. Both services use the same version of the Microsoft PlayReady DRM application.

5.6 GEOLOCATION

Regarding authorization mechanisms, both providers enforce geographical restrictions based on the source IP address of the request. If the user's location is outside of Turkey, clients do not allow users to log in to the services. This restriction can be easily circumvented through VPN applications. For Tivibu, it is also possible to bypass this control by modifying the X-Forwarder-header value. To ensure a stronger enforcement of geographical restrictions, services could require the users to provide a credit card number with a billing address in Turkey. This means that even with VPN connections that assign Turkey-based IP addresses, the user has a significant barrier to overcome if he/she does not possess a credit card with the required billing address.

5.7 TIME DIFFERENCE FROM SATELLITE

Time differences between the broadcasts of Tivibu Web and Digiturk Play, and satellite broadcasts are not negligible. For Tivibu Web, the average time difference is 14,6 seconds, whereas for Digiturk Play, the average is 51,1 seconds. The value of Tivibu Web is also not acceptable. The time difference value for Digiturk Play is far worse than that of Tivibu Web. The CDNs of Digiturk Play are not located inside Turkey but this could not be the single reason for this big difference. This test shows that Digiturk Play spends a plenty of time transcoding and encoding. Service providers should try and minimize the time they spend for operations such as encoding and encrypting.

6. CONCLUSION

In this thesis, we have analyzed and compared the Tivibu Web and Digiturk Play Internet TV services on the basis of technology and security mechanisms that they use, the differences between their solutions, and the down sides of these solutions. Both solutions have advantages and also have a number of points that should be improved.

When the results of the tests and technology used by these two services are analyzed, it is clear for both services that the Tivibu Web solution is a few steps ahead of the Digiturk Play solution. Tivibu Web makes it possible by utilizing a unique technology for all its services, by managing/operating its own services and by offering user-friendly clients.

In terms of security, there are critical issues for the Digiturk Play service as described above, the first of which is, Digiturk Play does not have an URL Signing protection for its clear streams. Secondly, when submitting some private data such as the Turkish ID number, the HTTPS protocol is not used. We have also showed that it is easy to crack into the watermark protection method of Digiturk Play's player.

In terms of DRM, both of the solutions are stable as they make use of the Microsoft PlayReady DRM solution.

BIBLIOGRAPHY

- [1] Sujata J. & Jayendran G.S. & Rohit D., 2015. Transforming Telecom Business: Scaling the Shift using Predictive Analytics. Indian Journal of Science and Technology. Vol 8(S4), pp. 34-43.
- [2] Yim J. & Lee G. & Lee T. & Jeon J., 2014. Review of IPTV System Architectures. Advanced Science and Technology Letters. Vol.46 (Multimedia 2014), pp.83-86
- [3] Kokaram A. & Crinon R. & Nicolas C. 2015. OTT (Over-The-Top) in 2015. Motion Imaging Journal, SMPTE. Vol.124 (6), pp.65-68
- [4] Available online:
http://web.calstatela.edu/faculty/cliu/EE446/Literature/STB_Architecture.pdf,
last accessed at Nov 10, 2015.
- [5] Coyle K. 2003. The Technology of Rights: Digital Rights Management, http://www.kcoyle.net/drm_basics.pdf, last accessed at Nov 10, 2015.
- [6] TTNET Hakkında,
<http://www.ttnet.com.tr/bireysel/Hakkinda/Sayfalar/Hakkimizda.aspx>, last
accessed at Nov 10, 2015.
- [7] Pereira C. F., (2011). Security On Over the Top TV Services. Thesis for MSc Degree. Lisbon: University of Lisbon
- [8] Bilgi Teknolojileri ve İletişim Kurumu, 2015, Üç Aylık Pazar Verileri Raporu, http://www.btk.gov.tr/File/?path=ROOT%2f1%2fDocuments%2fSayfalar%2fPazar_Verileri%2f2015-Q2.pdf, last accessed at Nov 10, 2015.
- [9] Türk Telekom, 2014, 2014 Faaliyet Raporu. http://www.ttinvestorrelations.com/_files/pdf/tr/2014-faaliyet-raporu.pdf, last
accessed at Nov 10, 2015
- [10] Tivibu Özellikleri, <http://www.tivibu.com.tr/tivibu-ozellikler>, last accessed at
Nov 10, 2015.
- [11] Svod Press, Subscription Video on Demand (SVoD), http://www.thevab.com/pdf/SVOD_Cable_Nation-Report.pdf, last accessed at
Nov 10, 2015.

- [12] TTNET Tivibu Sıkça Sorulan Sorular, <http://www.tivibu.com.tr/sikca-sorulan-sorular>, last accessed at Nov 10, 2015.
- [13] Microsoft Silverlight, <http://www.microsoft.com/silverlight/default.aspx>, last accessed at Nov 10, 2015.
- [14] Microsoft IIS Smooth Streaming, <http://www.microsoft.com/silverlight/iis-smooth-streaming/>, last accessed at Nov 10, 2015.
- [15] Microsoft, what is Smooth Streaming, <http://www.microsoft.com/silverlight/smoothstreaming/>, last accessed at Nov 10, 2015.
- [16] Microsoft PlayReady DRM Application, <http://www.microsoft.com/playready/>, last accessed at Nov 10, 2015.
- [17] Google Chromium Blog, Saying Goodbye to Our Old Friend NPAPI, <http://blog.chromium.org/2013/09/saying-goodbye-to-our-old-friend-npapi.html>, last accessed at Nov 10, 2015.
- [18] Microsoft, Windows Edge Dev Blog, <https://blogs.windows.com/msedgedev/2015/07/02/moving-to-html5-premium-media/>, last accessed at Nov 10, 2015.
- [19] E. Rescorla, (May. 2000), "HTTP Over TLS" IETF RFC 2813
- [20] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: using hard AI problems for security. In Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT'03, pages 294–311, 2003.
- [21] Pietschmann C., Silverlight: Anatomy of a .XAP file, <http://pietschsoft.com/post/2008/03/Silverlight-Anatomy-of-an-XAP-file>, last accessed at Nov 10, 2015.
- [22] Siddiqui M.S., Verma D., 2011. Cross site request forgery: A common web application weakness. IEEE 3rd International Conference, 27-29 May 2011, pp. 538-543
- [23] Microsoft. 2010. Making a Service Available Across Domain Boundaries, <https://msdn.microsoft.com/en-us/library/cc197955.aspx>, last accessed at Nov 10, 2015

- [24] Park J. S., Sandhu R, 2000. Secure Cookies on the Web, 3rd ed. IEEE Internet Computing, pp. 36-44
- [25] Xie J., Pam X., 2010. An improved RC4 stream cipher. 2010 International Conference. pp. V7-156 - V7-159
- [26] U.S. Department of Commerce, 2008. Secure Hash Standard (SHS). NIST FIPS PUB 180-3.
- [27] Boneh D. & DeMillo R. A. & and Lipton R. J., 2001. On the Importance of Eliminating Errors in Cryptographic Computations. J. Cryptology, 14(2), pp. 101-119
- [28] Vakali A. & Pallis G. 2003. Content Delivery Networks: Status and Trends. IEEE INTERNET COMPUTING. pp. 68-74
- [29] Mathew, V. & Sitaraman, R.K. & Shenoy, P., "Energy-efficient content delivery networks using cluster shutdown," in Green Computing Conference (IGCC), 2013 International , vol., no., pp.1-10, 27-29 June 2013
- [30] Microsoft. IIS Smooth Streaming Live Server Manifest Format. [https://msdn.microsoft.com/en-us/library/ee673443\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/ee673443(v=vs.90).aspx), last accessed at Nov 11, 2015
- [31] Microsoft Developer Network. SmoothStreamingMedia. <https://msdn.microsoft.com/en-us/library/ff469578.aspx>, last accessed at Nov 11, 2015
- [32] Hemalatha, K. & Yadav, P.K. & Ramasubramanian, N., 2015. "Adaptive bitrate transcoding for power efficient video streaming in mobile devices," in Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on , vol., no., pp.1-5
- [33] Microsoft IIS Forums. 2011. Smooth Streaming fragment length. <http://forums.iis.net/t/1177120.aspx?Smooth+Streaming+fragment+length>, last accessed at Nov 11, 2015
- [34] S. Josefsson, 2006. "The Base16, Base32, and Base64 Data Encodings", RFC 4648, <http://www.ietf.org/rfc/rfc4648.txt>
- [35] 2001. National Institute of Standards and Technology.

- [36] Microsoft Developer Network. Digital Rights Management (DRM). [https://msdn.microsoft.com/en-us/library/cc838192\(v=vs.95\).aspx](https://msdn.microsoft.com/en-us/library/cc838192(v=vs.95).aspx), last accessed at Nov 11, 2015
- [37] Microsoft. Using Silverlight™ DRM, Powered by PlayReady®, with Windows Media® DRM Content. http://download.microsoft.com/download/7/6/D/76D540F7-A008-427C-8AFC-BE9E0C0D8435/Using_Silverlight_with_Windows_Media_DRM-Whitepaper_FINAL.doc, last accessed at Nov 11, 2015
- [38] Windows Dev Center. Individualizing DRM Applications. [https://msdn.microsoft.com/en-us/library/windows/desktop/dd743231\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd743231(v=vs.85).aspx), last accessed at Nov 11, 2015
- [39] Mizusawa, J. & Shigematsu, N. & Itoh, H., 1998. "Virtual private network control system concept," in Private Switching Systems and Networks, 1988., International Conference on , vol., no., pp.137-141
- [40] Hunt G. Modify Headers, <http://www.garethhunt.com/modifyheaders/?v=0.7.1.1>, last accessed at Nov 11, 2015
- [41] W3C. Guidelines for Web Content Transformation Proxies 1.0. 2009. <http://www.w3.org/TR/2009/WD-ct-guidelines-20091006/#sec-additional-headers>, last accessed at Nov 11, 2015
- [42] Microsoft Developer Network. Output Protection Levels. <https://msdn.microsoft.com/en-us/library/dn468832.aspx>, last accessed at Nov 11, 2015
- [43] I. J. Cox & M. L. Miller & K. Tanaka & Y. Wakasu, 1998. "Digital data watermarking," Patent EP0 840 513, <http://www.freepatentsonline.com/EP0840513A2.html>
- [44] Callegati, F. & Cerroni, W. & Ramilli, M., 2009. "Man-in-the-Middle Attack to the HTTPS Protocol," in Security & Privacy, IEEE , vol.7, no.1, pp.78-81
- [45] Mekuria R. & Cesar P. & Bulterman D., Digital TV: The Effect of Delay when Watching Football. <http://oai.cwi.nl/oai/asset/20502/20502D.pdf>, last accessed at Nov 11, 2015

- [46] Digiturk Tarihçe. <http://www.digiturk.com.tr/kurumsal/tarihce/>, last accessed at Nov 11, 2015
- [47] Octoshape. Infinite HD-M the Federated Linear Broadband TV Platform. <http://www.octoshape.com/solutions/infinite-hd-m>, last accessed at Nov 11, 2015
- [48] Digiturk. Digiturk Uyelik Detayları. <http://www.digiturkplay.com.tr/uyelikdetaylari/sinema>, last accessed at Nov 11, 2015
- [49] Octoshape. Broadcasting Guide. <https://support.octoshape.com/entries/20655323-broadcasting-guide>. last accessed at Nov 11, 2015
- [50] J. Postel, (Aug. 1980), "User Datagram Protocol" IETF RFC 768
- [51] Darpa Internet Program Protocol Specification, (Sep. 1981), "Transmission Control Protocol" IETF RFC 793, <https://tools.ietf.org/html/rfc793>
- [52] Octoshape. TV Quality TV Quality. <http://www.octoshape.com/technology/tv-quality>, last accessed at Nov 11, 2015
- [53] Ax components. Audio/video streaming technologies overview. <http://ax-comp.com/video-streaming-technologies.html>, last accessed at Nov 11, 2015
- [54] Alstrup S. & Rauhe T., 2005. A new technology for large-scale streaming over the Internet. https://tech.ebu.ch/docs/techreview/trev_303-octoshape.pdf, last accessed at Nov 11, 2015
- [55] Brown S., 2012. Enabling the Quality, Scale, and Economics of Television today Over the Top with Multicast. <http://www.octoshape.com/wp-content/uploads/2012/05/octoshape-tvnext-10-2012v3.pdf>, last accessed at Nov 11, 2015
- [56] Digiturk. Digiturk Play Nedir. <http://www.digiturkplay.com.tr/digiturkplaynedir/> last accessed at Nov 11, 2015
- [57] Digiturk. Digiturk Haber - Korsan Yayın Faturası. <http://www.digiturk.com.tr/digihaber/korsan-yayinin-faturasi-26960-tl>, last accessed at Nov 11, 2015

[58] Octoshape Infinite HD Player 3 (IHDP3) - Advanced configuration, <http://webcache.googleusercontent.com/search?q=cache:Vv-ns2sK71MJ:https://support.octoshape.com/entries/26519037-Octoshape-Infinite-HD-Player-3-IHDP3-Advanced-configuration+&cd=2&hl=tr&ct=clnk&gl=tr>, last accessed at Nov 23, 2015

VITAE

Yavuz Sert was born in Bilecik, Turkey on December 16, 1978. He graduated from Bilecik Anatolian High School in 1996. He received his B.Sc. degree in Mathematics Engineering from Istanbul Technical University in 2000.

After his graduation, he worked as test engineer, system administrator, test manager, and product development manager at Mynet and Argela Technologies. Currently, he works at Argela as Engineering Manager.

His main areas of interest are operating systems, databases, and applications and test / quality. He is a certified Sun Solaris Administrator and a certified test engineer.