UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

# SECURITY ON OVER THE TOP TV SERVICES

**Carlos Filipe Zambujo Lopes Pereira**

MESTRADO EM SEGURANÇA INFORMÁTICA

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

# SECURITY ON OVER THE TOP TV SERVICES

**Carlos Filipe Zambujo Lopes Pereira**

**Orientador**
Nicolas Christin
**Co-Orientador**
Nuno Neves

MESTRADO EM SEGURANÇA INFORMÁTICA

# Resumo

A oferta abundante de serviços de acesso à Internet em banda larga, em redes fixas e redes móveis, e a popularização do uso de dispositivos móveis capazes de mostrar vídeo de alta qualidade descarregado da Internet, criaram um mercado para serviços que disponibilizam conteúdos de vídeo e televisão através na Internet para computadores pessoais, dispositivos móveis e aparelhos de televisão. Este mercado, com o paradigma de “3 ecrãs em qualquer sítio”, levou ao aparecimento de serviços de vídeo e televisão *over-the-top*. Os fornecedores OTT entregam conteúdos de vídeo e televisão através da Internet, usando redes de outros operadores. Os clientes acedem aos conteúdos OTT “por cima” dos serviços contratados ao fornecedor de Internet. A disponibilização de serviços de vídeo num ambiente aberto, como a Internet, requer que os seus operadores implementem mecanismos de segurança que protejam os seus valiosos conteúdos de acessos ilícitos, duplicação e distribuição não autorizada. Nesta tese, propomo-nos determinar as propriedades de segurança necessárias para fornecer serviços de vídeo OTT de forma segura. Com o objectivo de avaliar os mecanismos de segurança usados para assegurar autenticação, autorização, gestão de direitos digitais e restrições geográficas, estudamos três fornecedores OTT proeminentes. Pelo seu tamanho e pela tecnologia usada, escolhemos analisar o Netflix, Hulu e Comcast, três serviços de grande dimensão e popularidade nos Estados Unidos. Recorrendo a analisadores de protocolos de rede para inspecção do tráfego de mensagens, estudamos as interacções entre as aplicações cliente e os servidores. Para cada um dos mecanismos de segurança identificados e estudados, fizemos experiências com o objectivo de encontrar falhas e testar a sua eficácia. Descrevemos as experiências realizadas com clientes baseados em navegadores de Internet e clientes para dispositivos móveis com sistema operativo Android. Os resultados obtidos são apresentados e para cada fornecedor de serviço OTT é feita uma análise de segurança, onde são identificados os problemas de segurança encontrados. De entre estes, os mais significativos são problemas relacionados com o tratamento e a transmissão de cookies HTTP em claro pelos clientes baseados em navegadores de Internet. Estas vulnerabilidades são comuns aos três operadores OTT analisados e podem ser exploradas por adversários para roubar cookies de autenticação e personificar o cliente legítimo, permitindo acesso ilícito a conteúdos de vídeo e a informação privada do cliente. O ataque de roubo de cookies de autenticação é descrito e exemplificado com uma experiência feita numa rede Wi-Fi aberta. A simplicidade do ataque e a existência deste tipo de redes em espaços públicos, como escolas, universidades, centros comerciais, hotéis e aeroportos, tornam importante a implementação de medidas correctivas. São apresentadas estratégias de mitigação para fornecedores de serviços OTT, utilizadores e administradores de redes sem fios. Estas consistem, respectivamente, na utilização de SSL para proteger a informação de autenticação, a utilização somente de ligações HTTPS ou de acessos cifrados a redes privadas virtuais, e a utilização de protocolos do standard WPA2 para protecção de redes sem fios. Ao contrário do que é observado com os clientes baseados em navegadores de Internet, o cliente móvel da Netflix para dispositivos Android não é vulnerável ao ataque descrito. Este cliente usa SSL para proteger todas as ligações em que são transmitidos cookies de autenticação.

**Palavras-chave:** Internet, segurança, vídeo, OTT, IPTV

## Abstract

The widespread availability of high bandwidth Internet access on fixed and mobile networks, in conjunction with the availability of mobile devices powerful enough to play streamed high quality video, has created the demand for services that deliver television and video content over the Internet to television sets, personal computers and mobile devices. This demand has lead to the appearance of over-the-top TV and video service providers that deliver video over the Internet, using networks not operated by them. Video delivery in an open environment, like the Internet, requires operators to implement security mechanisms to protect their valuable content from illicit access and distribution. In this thesis, we investigate security properties needed to securely deliver OTT video services. In order to assess the security mechanisms employed to enforce authentication, authorization, digital rights management and geographical restrictions, we survey three prominent OTT service providers. Due to their size and choice of technologies, we selected Netflix, Hulu and Comcast. We studied the interactions between the client applications and the providers' servers by inspecting the traffic of messages exchanged. For each of the security mechanisms analyzed, experiments were designed to find flaws and test their effectiveness. The most important of the identified security issues are related to the handling and transmission of HTTP cookies when using web browser-based clients. These vulnerabilities are common to all surveyed providers and can be exploited by adversaries to steal authentication cookies and impersonate the customer, allowing illicit access to video assets and private information of the customer. A cookie stealing and session hijacking attack is described and mitigation strategies are presented for OTT service providers, users and wireless network access point administrators. These consist in the use of SSL to protect authentication tokens, the use HTTPS only or VPN services, and the use of WPA2 to protect wireless networks, respectively. An interesting result, observed with the analyzed mobile client for Android devices, is that it uses SSL to protect the transmission of HTTP cookies used for authentication. Thus, it is not vulnerable to the described attack.

**Keywords:** Internet, video, security, IPTV, OTT

## Acknowledgments

*I owe my gratitude to Doctor Nicolas Christin and Doctor Nuno Neves for their invaluable guidance as advisors of this thesis. I am grateful to my colleagues André, Rui and Paula for interesting and fruitful discussions about the research work and help with writing in LaTeX. It is my pleasure to thank Professors Paulo Veríssimo, Adrian Perrig, Miguel Correia, Alysson Bessani and António Casimiro for the teachings in security and dependability that provided me with the background necessary for this thesis. I am thankful to my wife Sílvia, my brother Frederico and my parents Francisco and Yolanda, for the encouragement and unconditional support in difficult moments. At Portugal Telecom, I would like to thank Ana Maltez and Cristina Veríssimo for encouraging and helping me to pursue the opportunity provided by the Carnegie Mellon|Portugal partnership; João Ferreira and Luís Gueifão for proposing and giving their support for this thesis. Finally, I would like to thank Portugal Telecom, my employer, for making my participation in this masters program in Information Security possible.*

Lisbon, November 2011

*Dedicated to my wife Sílvia.*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Over-the-top TV services

With the widespread adoption of high bandwidth Internet access in developed nations, new business opportunities have opened to TV content creators, owners and distributors. It is now possible to deliver high quality video over the Internet, providing customers with the flexibility to watch the content of their choice on a variety of Internet connected device categories. Many companies have been trying to succeed in the marketplace, experimenting with various business models. Most prominently, Hulu and Netflix became large players in the video distribution over the Internet space. This type of video services, which are distributed over a network but are not offered by the network operator, came to be referred to as OTT video services: they are served *on top* of the network service provided by an Internet service provider.

Telecommunications operators that have five years ago started to offer IPTV services, integrated with voice and Internet services, are now expanding their TV offerings to include the ability to watch television programming on personal computers, mobile smart phones, tablets, Internet connected TV sets, gaming consoles, Blu-ray players and other types of Internet connected devices. The demand for *3-screens anywhere services* – TV, personal computer, and mobile screens – is a growing trend. Although these services are offered by the network operator, generally the customer can also access them through other networks, over the top, as long as there is an Internet connection and access is not explicitly blocked due to policy enforcement.

## 1.2 Motivation

Television programming and premium video content distribution over fixed networks have traditionally been done in closed environments, where the operators have a large degree of control over the network and the customer premises equipment. Where fraud is possible, network operators have been working to thwart it, implementing encryption mechanisms and set-top box authentication schemes that prevent unauthorized devices from receiving and decoding the signal. Nevertheless, the problem of unauthorized access has been largely limited to the physical network of the provider.

With over-the-top TV services, distributed via the Internet, users do not need to be physically connected to the operator's network. Access can potentially be done from anywhere in the world. The ability to access content on mobile devices, personal computers and other Internet connected devices from anywhere within the allowed geographical regions is a critical feature of OTT services. Distributing premium content over open IP networks presents new challenges to fixed network TV operators:

- content distribution licenses impose geographical restrictions on where the content can be accessed from, even for existing legitimate customers;
- content distribution servers must not allow unauthorized access to premium content by non-paying users;
- the operator does not control end devices but unauthorized recording and copying of content bits should be prevented;
- Denial of Service attacks on the network, coming from thousands of machines located all over the world, may leave legitimate customers without service.

To be able to serve premium content, OTT service providers must properly monitor and address these security risks. Authentication, authorization, session management, digital rights management, auditing geolocation, quality assurance and video output protections are security requirements that operators must guarantee to be satisfied.

## 1.3 Our work

This thesis sets out to survey and explore the security properties needed to secure OTT service delivery. In order to assess the main security strategies and mechanisms employed by major OTT video service providers, we selected Netflix, Hulu and Comcast. By virtue of their size and the choice of technologies they use, we conjectured that these three large and prominent services would provide a representative sample of user experience.

We found that the three providers use SSL during the sign in process to authenticate the server to the user and encrypt connections. SSL protects the user's log in credentials, with which he/she authenticates to the server. After successfully signing in, SSL is no longer used for video catalog browsing and users are then redirected to HTTP connections. Session state and authentication is maintained via HTTP cookies sent in the clear, leaving the customer vulnerable to cookie stealing and session hijacking by network eavesdroppers. To assess the difficulty of this attack, we ran a proof of concept experiment in a controlled environment, which revealed that the attack is simple to do.

We analyzed how geographical location is done and in which phases of the customer interaction with the service are the geographical restrictions enforced. We found that the use of VPN software allows some of these restrictions to be bypassed. In the case of Netflix, this method can be applied in the initial authentication and authorization phases to obtain a license that allows the user to subsequently watch the video without the VPN connection. In the case of Hulu and Comcast, the VPN must be used during the entire time the user is watching. Hulu is even more restrictive in the case of premium content: by verifying the billing address of the credit card in the account creation process, Hulu enforces that its premium customers must have a billing address within the United States.

Our analysis proceeded with simultaneous streaming restrictions, video stream encryption and output protections. All three OTT providers we studied enforce simultaneous streaming restrictions, thus limiting the sharing of account credentials. They all encrypt the video stream the server sends to the client, requiring the client to obtain a valid license containing the decryption key. We learned that none of the providers are using output protections, making it possible to send the images to unprotected analog and digital outputs.

The experimental work was concluded with the analysis of the Netflix client application for mobile phones running Android. We observed that the Android client encrypts all communications that are related to authentication, session management and authorization. Thus, unlike the web browser-based client for personal computers, the Netflix mobile client for Android is not vulnerable to cookie stealing attacks by network eavesdroppers.

Reflecting our views on the subject of insecure web authentication and session cookies, we wrote a set of recommendations for OTT service providers, network administrators and end users. The recommended practices are not new, but would increase the security for everyone using these premium content services.

## 1.4 Contribution

During the preparation work for the thesis, we found an abundance of literature on the topic of IPTV security. These works focus predominantly on closed IPTV networks like those implemented by traditional telecommunications and cable operators. These networks use custom built set-top boxes, with the specific purpose of delivering the operator's TV service to the customer.

We found very few works on analysis of OTT services. The most relevant analysis, and the only one we found on the topic of security, is a report done by Pomelo, LLC[1] [27] that briefly describes the authentication and authorization interactions between the Netflix client and the service's servers. This report provided us with good starting point, but did not detail the interactions with the depth we were aiming for and did not mention aspects like how Netflix does geographical location, simultaneous streaming restrictions or output protections. Adams also briefly describes the interactions between the Netflix client, the Netflix Controller server and the CDNs [2]. The analysis is not focused on security aspects, but rather on aspects related to simulating the traffic load generated by Netflix. A company called ViaForensics has done a security analysis[2] of the Netflix mobile application for Android, but the study is not freely available.

[1] http://www.pomelollc.com/
[2] http://viaforensics.com/appwatchdog/netflix-android.html

In our work, we surveyed three OTT video service providers, with emphasis on the security mechanisms of authentication, authorization, simultaneous streaming, geographical restrictions, output protections and, in the case of Netflix, mobile client security. As far as we know, this is the first work that surveys multiple over-the-top video service providers.

We strived to describe the mechanisms implemented by the OTT providers in more detail than the mentioned previous works, as well as test them for effectiveness and correctness. In most cases, our search for bugs and vulnerabilities did not reveal any flaws and did not enable us to bypass the services' security controls. We did find, however, a pattern of vulnerability common to authentication schemes used by Netflix, Hulu and Comcast on their websites: they all use cookies without any encryption to authenticate the user after the sign in process, leaving their users vulnerable to cookie stealing and session hijacking attacks.

Another aspect not included in the previous works, is the effectiveness of enforcement of geographical restrictions early on in the account creation process, which revealed different strategies being followed by Hulu and Netflix: Netflix only relies on geographical location based on source IP address, while Hulu verifies the billing address associated with the credit card.

## 1.5 Organization of the thesis

The rest of this document is organized as follows:

**Chapter 2 – Objectives, methodology and limitations,** presents the goals set out for this thesis and the methods employed in the experiments. The last Section describes limitations encountered while doing the research.

**Chapter 3 – Netflix,** describes the experimental work developed to analyze Netflix's service and results obtained. The Chapter concludes with a security evaluation of the service.

**Chapter 4 – Hulu Plus,** presents the experimental work developed to analyze Hulu's service. The findings of our study are then presented and the Chapter concludes with a security evaluation of Hulu's service.

**Chapter 5 – Comcast Xfinity TV,** presents the study of Xfinity's online video service and its security evaluation.

**Chapter 6 – Recommendations,** contains a set of recommendations to mitigate the security vulnerabilities found in cookie handling and transmission.

**Chapter 7 – Conclusions,** summarizes the main results and conclusions of the thesis.

# Chapter 2

# Objectives, methodology and limitations

## 2.1 Objectives

We aimed to survey and evaluate the security mechanisms available to and implemented by some of the major OTT service providers. We could not attempt to analyze an exhaustive list of existing providers, as there are today many of them. The limited time available to us would make the task impossible. A complete security review of the services offered by any one provider would also be too lengthy to fit our schedule. Another difficulty arises when doing this kind of security evaluation work: service providers are not willing to grant inside access to their systems to be evaluated by outsiders.

For these reasons we chose to do a security evaluation of a few selected companies, using a *black box* approach, concentrating on very specific aspects. We selected three OTT providers to analyze:

- **Netflix** – offers a streaming service that offers its customers access to movies, TV shows and documentaries for a flat monthly fee. Users have unlimited access to the catalog from within the allowed geographical regions. Netflix has enjoyed large market share in the movie streaming business and has secured content deal with many content providers. According to a report [26] by networking equipment company Sandvine, in May of 2011, Netflix was the largest source of Internet Traffic in North America during peak hours. For content protection, Netflix uses a combination of in-house solutions and Microsoft PlayReady DRM technology. Customers can access content using personal computers, mobile smart phones, tablets, gaming consoles and other Internet connected devices. We chose to analyze Netflix for its size and technology.
- **Hulu** – offers streaming of TV shows, movies, movie trailers and web specific content. It has an ad-supported free service that allows users to watch content on personal computers and it also has a paid service called Hulu Plus. Hulu Plus also has ads, but customers have

access to additional premium content, can access HD content and have the ability to watch on Internet connected TVs and mobile devices. Content is streamed and DRM protected using Adobe Flash technology and is only available in the United States and Japan. Hulu is a joint venture of major traditional television content providers. We chose to evaluate Hulu because it is one the prominent players in the Internet video distribution arena, because of its technology and because it is owned by traditional television business players.

- **Comcast** – is the largest cable operator and Internet service provider in the United States [24]. It is also one of the largest telephone providers. Comcast offers Internet access to movies, TV series and programming from premium networks like HBO, Starz and others. Premium networks' content is only available to subscribers with access through Comcast's television service. Other content is available for free for anyone accessing over the Internet from within the United States. We chose to evaluate Comcast because of its size and because it is a traditional cable TV operator.

For each one of these providers, we concentrated our efforts on the mechanisms realizing the security properties that are particularly relevant to secure OTT service delivery:

1. **Authentication** – How do securely OTT service providers authenticate their users?
2. **Authorization** – After making sure of the identity of the user, how do OTT providers determine whether to grant access to content and how do they securely give permissions to the user?
3. **DRM** – How is content protected from unauthorized access and unauthorized use?
4. **Simultaneous streaming** – Do OTT allow content to be accessed with the same account from multiple devices?
5. **Geolocation** – How do OTT providers enforce geographical restrictions imposed by their content licensing deals?
6. **Output protections** – Are OTT providers enforcing output protections to prevent unauthorized copying?
7. **Mobile client security** – Are the previous points implemented differently on mobile clients than they are in web browser clients?

Our last goals were to identify vulnerabilities in the surveyed mechanisms and propose remedies and mitigating measures.

## 2.2 Methodology

The services from each of the selected providers expose interfaces to client applications. Our analysis was focused on the interactions that occur through these interfaces using a black box approach, on the client behavior and on the service responses to non-standard client requests. No information was asked to the OTT providers about their systems' internal architecture, implementation details and specific security mechanisms employed.

To carry out our work we used a number of freely available software tools on Windows machines and on Macintosh computers. The tools we used include:

- **HTTP debugging proxy servers** – are HTTP proxies that allow interception, modification, generation and analysis of HTTP requests and responses. With the installation of the appropriate TLS certificates [11], some HTTP debugging proxies allow the analysis, modification and generation of HTTPS traffic. In our work, we did use such techniques to analyze traffic protected by TLS connections.

- **Network protocol analyzers** – are more generic traffic analysis tools than HTTP proxies. They allow monitoring of traffic at various levels of protocol stacks and support the dissection of many known and standard protocols. On hardware with network interfaces that support *promiscuous mode*, these analyzers can be used to analyze traffic not addressed to, originated by, or being routed through the machine on which they are installed, as long as they have access to the network medium that is carrying the traffic. We used network protocol analyzers, instead of the simpler HTTP debugging proxies, in two specific situations:

  1. To monitor traffic to and from a mobile smart phone, connecting to the Internet via through a machine serving as a wireless access point. The protocol analyzer was installed on this machine.
  2. To demonstrate an attack on a wireless network in a controlled environment. This experiment was run in a network specifically set up for this demonstration, in an environment that did not pose any risks to non-participating entities or machines, and with user accounts created specifically for this purpose.

- **OpenSSL** – is an open implementation of the SSL/TLS protocols. It supports the generation of RSA public/private key pairs, generation and signing of X.509 certificates [17].

- **Android rooters** – are tools that can be used to *root* Android devices. Rooting is the process of modifying the operating system in order to allow users to run applications with root privileges. There are many applications available to root Android devices, with varying levels of support for device models and operating system versions.

- **VPN connections** – were used to run experiments in the United States with connections originating in Portugal and to run experiments in Portugal with connections originating in the United States.

- **Media metadata readers** – were used to run tests and determine whether a decrypted file had MP4 metadata or not.

- **Image capturing tools** – we used to capture video images displayed by the client software used by the OTT video service providers.

To test the mobile Netflix application, we used an Android based smart phone, running Android 2.2.2, which had to be *rooted* in order to install an SSL CA certificate in it.

## 2.3 Limitations

We had not completely accomplished our initial objectives when we decided to stop our experimental work and concentrate on writing the thesis. The main difficulties we faced are inherent to the security mechanisms and strategies the OTT service providers we chose to analyze:

- To allow access to premium content, Comcast's Xfinity online service requires the user to be a subscriber of the Xfinity TV cable service and a paying customer of premium channels. We did not have an Xfinity TV cable account dedicated to our tests. To study the services offered in Xfinity's website, we had to borrow access from a web account belonging to one of the advisors of this thesis. Although this web account was associated with a cable account, the physical installation did not have a cable box connected to network. Thus, we could not test the cable box management and remote control functions on the website.
- Most of the work for the thesis was done during a three months period, in the summer semester, in Pittsburgh. The time was limited and more in-depth analysis of protocols such as RTMP, used by Adobe Flash players, was not possible.
- Time was also a limiting factor on the mobile client application analysis. The necessary access point configuration on a machine connected to the Internet is not allowed by the Cisco AnyConnect VPN software, used to connect to CMU's network and run experiments from Portugal. As a result, analyzing the mobile client in Portugal took more time than expected.
- The lack of a device supported by the Hulu Plus Android application prevented us from doing the experiments with Hulu's mobile client.

Despite these limitations, we believe our work met most of the goals we set out in the beginning. The survey and evaluation of the selected security aspects implemented by Netflix, Hulu and Comcast are presented in the following chapters.

# Chapter 3

# Netflix

Netflix, Inc. is company that specializes in video rentals. Customers can subscribe to DVD by mail rentals and/or a video on demand streaming service over the Internet called *Watch Instantly*, which is only available in the United State and in Canada[1]. For a flat monthly fee, *Watch Instantly* subscribers have unlimited access to its entire movie catalog.

A report done by Pomelo, LLC[2] in April of 2009, helped us understand the basic architecture implemented by Netflix [27] to enforce authentication, authorization and Digital Rights Management. However, that report had been done more than two years prior to our work and Netflix had been evolving the service. We also wanted our study to be both broader and more detailed. In order to analyze the mechanisms employed by Netflix to enforce authentication, authorization, DRM and enforcement of geographical restrictions, we analyzed the interactions between the client and Netflix's servers. In the following Sections, we describe the interactions we observed using Netflix's web browser client, from the account creation process to browsing the catalog, selecting a movie, watching the movie and closing the player. As we learned the processes by which Netflix enforces its security policy, we tried to find flaws and work around imposed limits. After studying the web browser client, we analyzed the mobile client on an Android smart phone. The Chapter ends with a security evaluation of the described processes.

## 3.1 Technology

*Watch Instantly* content can be accessed through a wide range of consumer devices from a variety of manufacturers: gaming consoles, Blu-ray players, HDTVs, Home Theater systems, dedicated web TV set-top boxes, phones, tablets and personal computers[3]. On a personal computer running Microsoft Windows or Mac OS X, the service can be accessed via a web browser with the Microsoft

[1] http://www.netflix.com/Help?action=2&jsEnabled=false&faqtrkid=5&p_faqid=4042&lnkctr=yas_faq

[2] http://www.pomelollc.com/

[3] https://account.netflix.com/NetflixReadyDevices

Silverlight plug-in installed. To implement DRM protections on *Watch Instantly* content, Netflix is using Microsoft PlayReady DRM[4], which is not available for Linux users.

## 3.2 Account creation

To create and manage an account with Netflix, users must navigate to `https://www.netflix.com/` and provide their email address and a password, which will be their credentials to access the service.

An HTTP POST request [12] sends this information to the server, together with an opaque token designed to defend against Login CSRF (Cross Site Request Forgery) attacks, as described in [8]. As an example, one HTTP POST request includes the parameters in Table 3.1.

| **Parameter** | **Value** |
|---|---|
| nextpage | http://www.netflix.com/ |
| errorPage | https://www.netflix.com/Default?loms=abcd |
| authURL | nHEEK4DmHSTmbhQI++zvA.1310401128967.3XYr /12rM6GwQ/160Q/zajtN+Yw= |
| email | email@address.com |
| email2 | email@address.com |
| password1 | user_password |
| password2 | user_password |
| SubmitButton | Continue |
| RememberMe | True |
| REGISTRATION_LOCATION | /Default?loms=abcd |

Table 3.1: Parameters passed in the HTTP POST request to register a new user account.

The account is created in `Inactive` state and the user is then redirected to a page that requests the additional information required to activate the account: first and last name; credit card details; and year of birth.

If it were not for the authURL token, a Login CSRF attack could allow an adversary to cause a victim to login as the attacker, by doing the account creation POST request with the attacker's credentials, and then trick the victim into providing credit card details in the account activation page.

No valid street address is required in the registration process when the user is connecting with an IP address from within the United States. When connecting from an IP address from Portugal, the user's browser is redirected to a page with the message:

> Sorry, Netflix is not available in your country... yet[5].

When Netflix is unable to determine the geographical location of the IP address, or if a connection from a U.S. IP address is sending HTTP cookies [7] assigned to a browser previously connecting from outside the U.S., Netflix presents the message in Figure 3.1 to the user.

[4] `http://www.microsoft.com/PlayReady/Default.mspx`
[5] `https://signup.netflix.com/global`


Figure 3.1: Message displayed by Netflix.com when it is not able to determine the user's IP address geographical location.

We have not been successful in triggering the subsequent U.S. mailing address confirmation. Clicking the `I have a valid U.S. mailing address` button causes Netflix to proceed to the `Sign up` page or to the `Global` page, as shown in Figure 3.2, depending on whether we are connecting from the U.S. or not.


Figure 3.2: Message displayed by Netflix Global page when a user connects from a foreign IP address.

We have tried replaying the new user registration HTTP POST request with the parameters shown in Table 3.1 from a foreign IP address from Portugal, without going to the main Netflix.com front page first. The registration request was accepted and it was possible to register a new user this way. However, the `authURL` shown in Table 3.1 imposes some restrictions to those willing to bypass the geolocation restrictions:

- `authURL` must be valid and the method to generate it is not publicly available – otherwise it would not be a secure defense against CSRF[8].

- It contains a timestamp in Unix format. In the example in Table 3.1, 1310401128967 represents Mon, 11 Jul 2011 16:18:48.967 GMT. Replaying it a few minutes after it has been generated causes the browser to be redirected to a page that performs the geolocation validation again and the user is presented with the message in Figure 3.2.

The entire sign up process is done over HTTPS to protect the user credentials and credit card information from eavesdropping and man-in-the-middle attacks. These measures are sufficient as long as adversaries are not able to forge SSL certificates with a valid Certificate Authority signature [29], [30] and the user adopts secure behavior when the browser presents SSL errors and warnings.

## 3.3 Authentication and session management

### 3.3.1 Sign in process

To login to Netflix, subscribers must navigate to `https://signup.netflix.com/` where they can provide their email and password credentials. The `Member Sign In` is a secure page served over TLS, as shown in Figure 3.3.

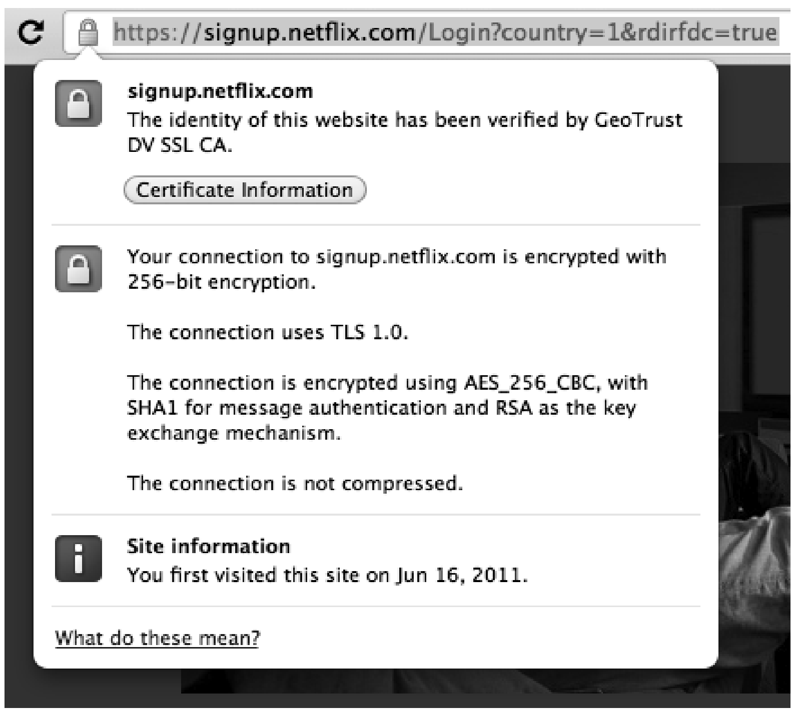


Figure 3.3: Secure submission of credentials in the Sign in process.

The `Member Sign In` does a number of cookie operations in the browser cookie store, one of which is to clear the `NetflixId` cookie, to deal with cases in which it was previously defined:

```
Set-Cookie: NetflixId=""; Domain=.netflix.com;
Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/
```

The sign in page contains in a hidden element the secret token `authURL`, shown in Table 3.2, and sets the cookie `VisitorID`:

```
VisitorId=002~a3181d57-84b3-49ac-b1d3-5dd095a020c3~1309555650935
~false~1309555650935~
```

When the user enters the sign in credentials and clicks `Continue`, the browser does a POST request that sends to the server the parameters shown in Table 3.2, including the user credentials and the secret token, and the cookie `VisitorID`.

| Parameter | Value |
|---|---|
| authURL | http://www.netflix.com/ |
| errorPage | https://www.netflix.com/Default?loms=abcd |
| authURL | 1310418460927.fZKrKcjveNWRDcOh2ZzHU%2BYl6tA%3D |
| nextpage | |
| SubmitButton | Continue |
| country | 1 |
| email | email@address.com |
| password | user_password |

Table 3.2: Parameters passed in the HTTP POST request to authenticate the user and sign in.

Tampering with, or removing, either the cookie `VisitorID` or the token `authURL` in the POST request causes the sign in process to fail. This scheme is designed to defend against Login CSRF attacks, as described in [8]. The cookie VisitorId establishes a *pre-session*, which the server uses to check the validity of the secret session-dependent nonce `authURL`.

If the credentials are correct, the server sets the authentication cookie `NetflixId` in the user's browser:

```
NetflixId=v%3D2%26ct%3DBQAOAAEBELo7W-YTy2bJNXcEl1TIeDCA4AnuGT1ifC5sP4hv
EI8UgFEdZC2k99H8Z-gf17PMV5BSlyKf7sK5QtbduIJe6uRLIGsLv8sXR2jEATDCXucUZJD
fCFnfkw8fPQac8-YRn_RiHozXyzEslK3Aht5V2xU1jfgm-RpK85pYH02xZzbD01Grbl48Y3
QJi_2glV83rfDRbroJAZ1y-YLyNvtTBIz3F0fB-EgXZjm4xkyhDKoCuZgTrEvek9mR0RZii
6GcfdtakzSuuR4GoXcXEHjCqPoUeljrMtlrvCXKyXdYpdZLWk2Eu9U51To15MdxJzWMOuvr
%26bt%3Dusr%26ch%3DAQEAEAABABRl67oGydn0Domopyk8KQjEqPFNWDOSXqo.%26mac
%3DAQEAEAABABTLNxa0RDLs6kSHXPY8Tz6v67Mn93RM1RE.;
Domain=.netflix.com; Expires=Tue, 10-Jul-2012 21:16:31 GMT; Path=/
```

This cookie is responsible for keeping the user authenticated in the Netflix.com website. Every time the user's browser makes a request to .netflix.com the cookie is sent in the request and is used by the servers to identify the user. The cookie is URL encoded and by decoding it we can identify the parameters shown in Table 3.3.

The cookie is a persistent cookie because there is an expiration date. It will not be lost when the user quits the browser, as a session cookie would. The Netflix site sets the cookie as persistent, even if the user does not select the `Remember me on this computer` checkbox in the Sign In page as shown in Figure 3.4. From a security perspective, this can constitute a vulnerability. User $A$ signing in to Netflix on a shared computer account, might think that not checking the checkbox will

| Parameter | Value |
|---|---|
| v | 2 |
| mac | AQEAEAABABTfeNqfpDBCIAQ2-eefKNHcsVm3FAlQRHI |
| ch | AQEAEAABABQetHlHVol4E02dVwbm315voHpO9i8sNhk |
| ct | BQAOAAEBEFCPZu557vjYzErOe0_XX3aBAElikp0KOI3489aUQcz1QciQXw 0F-UQX_F93-dzbNU5YPKsf1axyiXBHQidLnbMcTnXThNMPJLI1njTcgAWUpI cWt-pZ3JJ6JnepH7gc0d800gnayu_yyVELE8la9vQA3OtzT70unSVRXLkonxD xy_L2WCnqRtZMWVRFBVeDBMFmRGu5WjsSL6GRuFcLaGDOMS9PzQ_E ThtKJQAIrrV7926j5FxHMomlFRPT7EjzUaHYeQJt7KJMN9Zt4f3osHhjiqBUY_ 5bu7n5BlQgphpKbtY7sW1Ge1rVTeyuai7bju8-tjUbENW3hChoSmPysEqr6Jx2 bCz_Ls2HYzo36f4r3sA |
| bt | usr |
| Domain | .netflix.com |
| Expires | Tue, 10-Jul-2012 21:16:31 GMT |
| Path | / |

Table 3.3: Parameters in the `NetflixID` authentication cookie.

cause the browser to forget the session when it is closed without explicitly signing out from the website. This is not the case, and user $B$ using the same browser will be automatically signed in to $A$'s account when visiting `http://www.netflix.com`.

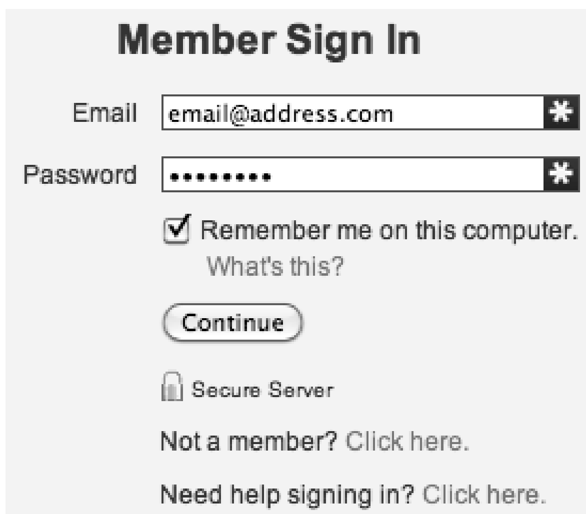


Figure 3.4: `Remember Me` option in the sign in page of Netflix.com.

The sign in process is done using HTTPS, all data being sent and received over SSL. On modern web browsers and operating systems, like Mozilla Firefox[6] running on Microsoft Windows 7 or Mac OS X, these SSL connections are encrypted with 256-bit AES and authenticated with SHA-1. Provided that users do not ignore security warnings displayed by the browser when server certificates are not properly signed by a Certificate Authority (CA), that the system does not contain a rogue CA certificate installed [11] and that adversaries have not been able to compromise a CA and produce forged certificates, this scheme provides a robust mechanism to protect the user credentials and authentication cookie from third parties.

[6] `http://www.mozilla.com/firefox/`

### 3.3.2 Session management

The HTTPS response that sets the `NetflixId` cookie, also redirects the user to `http://movies.netflix.com/` and then to `http://movies.netflix.com/WiHome`. This is the *Watch Instantly* home page that presents personalized movie suggestions. Although the sign in process was done over SSL, the user has now been redirected to plain HTTP, with no encryption. In order to keep track of the user's state, the website needs the browser to send the authentication cookie with every request it makes to domains with suffix `.netflix.com`, including `http://movies.netflix.com`. While the user is browsing the movie catalog and accessing movie information, every request sends the `NetflixId` cookie in the clear.

The fact that these communications are unencrypted allows adversaries with eavesdropping capabilities to see the cookie in clear text. This is a vulnerability that allows such attackers to steal the cookie and impersonate the victim to the Netflix website [10], in what are known as cookie stealing and session hijacking attacks.

### 3.3.3 Sign out process

When the user clicks on the `Sign Out` link, the server invalidates the authentication cookie on the browser by doing:

```
Set-Cookie: NetflixId=; Domain=.netflix.com;
Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/
```

The browser is then redirected to the `Signed Out` page. To regain access to the account, a customer must re-authenticate by entering the user credentials in the `Sign In` page. Despite the fact that the browser no longer has the authentication cookie, Netflix servers do nothing to invalidate it. To demonstrate this point, we have done the following sequence of actions:

1. While signed in, export the cookie `NetflixId` to a text file.
2. Sign out from the website by explicitly clicking `Sign Out`;
3. Import the cookie into the browser cookie store.
4. Go to `http://www.netflix.com` to confirm that we sign in without re-entering the user credentials.

Exporting the cookie and importing it to a browser in a different machine produces the same result: we are signed in to the user account without having to re-enter the authentication credentials. This means that signing out of the website only impacts the client side. From a security point of view, the consequence is that even if a user suspects that the cookie has been compromised, there is no quick way of explicitly invalidating it on the server to prevent continued abuse. The customer would have to contact Netflix through their Customer Service line, in what could potentially be a time consuming process. We have not tried calling Netflix's Customer Service and we don't know whether they would be able to solve this problem in an effective and satisfactory way.

### 3.3.4 User account details

In order to protect more sensitive user account information, such as the credit card update page, type of plan and billing history, Netflix also sets two secure cookies during the sign in process:

- `SecureNetflixId` – used during the sign in process;
- `NetflixShopperSecret` – used when the user accesses the account details page.

This provides considerable more security for these pages because the cookies are marked as Secure and are never transmitted in the clear. To capture these cookies and adversary would need to break SSL. These cookies are cleared during the sign out process using the same techniques employed for the `NetflixId` cookies. Exporting these cookies to a file and using them on another machine allows access to the user account details without having to provide the user credentials. This means that these cookies are not invalidated on the server as well.

Password and email address changes require the user to re-enter the current user credentials, even when the user is already logged in, increasing security even more to protect those critical data.

## 3.4 Authorization

### 3.4.1 Client start

When a user selects a movie to watch instantly, an HTTP GET request identifies the movie in the URL:

`http://movies.netflix.com/WiPlayer?trkid=1537777&movieid=70018715`

JavaScript contained in the server response downloads the Silverlight based Netflix player and launches it with the appropriate initialization settings. Notable parameters are listed in the Table 3.4.

| **Parameter** | **Value** |
|---|---|
| xsrf | 1309111643200.0dxkXOLYvz2nfnnzJ5KSBnGh/iA= |
| SupportedCountry | US |
| NccpControllerCloudEnabled | true |
| NccpControllerCloudShopperIdEnabled | true |
| UseNetflixId | true |
| xapUrl | http://movies.netflix.com/layout/silverlight/ SLPlayer3.xap?v=2.876.642.1 |

Table 3.4: Parameters passed to the Silverlight based Netflix player.

With the exception of the `UseNetflixId`, changing the value of these parameters does not produce any apparent differences in the player behavior. Changing `UseNetflixId=true` to `UseNetflixId=false` causes the player to display the error shown in Figure 3.5.

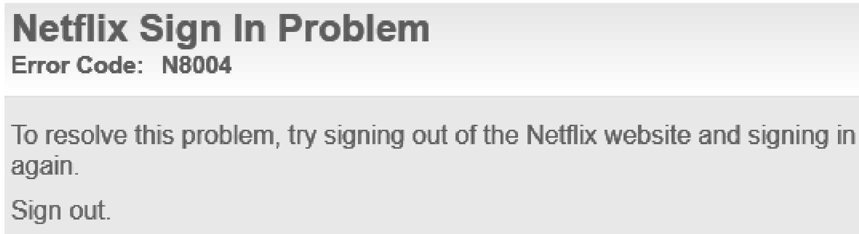


Figure 3.5: Error caused by setting `UseNetflixId=false`.

The `xapUrl` parameter in Table 3.4 contains the URL to the SLPlayer3.xap player, which is a Silverlight application that runs inside the browser. From this point onwards, the player begins requesting the necessary information to access and play the selected video asset.

The player requests the file `https://agmoviecontrol.netflix.com/clientaccesspolicy.xml` and it receives a list of valid domain URI's with which it can interact through NCCP, as is shown in Appendix A.1. NCCP is a protocol that the Netflix player uses to exchange data with Netflix's authentication, authorization and license servers. As far as we know, this is not a standard protocol and is a Netflix proprietary solution. The file `clientaccesspolicy.xml` also specifies in line 5 a number of integrity and authentication mechanisms the client should accept:

```
<allow-from http-request-headers="Content-Type,X-HMAC,X-CTicket,X-ESN,
X-ShopperID,X-AuthenticationType,X-FORCEIP,X-AllowCompression,
X-Netflix-ForceCountry">
```

Intercepting and then suppressing or modifying any of the specified integrity and authentication mechanisms from the server response causes the player to reissue the request, only this time it is done without any encryption over HTTP. Whether we allow the response to this second request to go unmodified or tamper with it to match the tampered response over HTTPS, result is always the same: an error message is displayed by the player as shown in Figure 3.6.

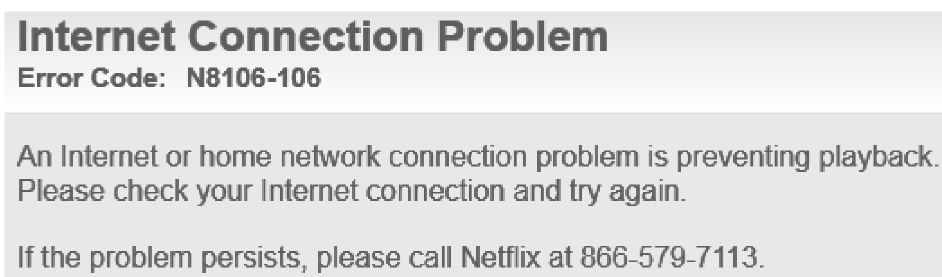


Figure 3.6: Error caused by changing or suppressing the authentication and integrity methods specified in `clientaccesspolicy.xml`.

### 3.4.2 Client registration

If this is the first time the Netflix player is running on this computer, the next step it does is to register itself with Netflix by doing an HTTP POST to `https://agmoviecontrol.netflix.com/`

nccp/controller/2.10/register. This HTTP post contains the NetflixId authentication cookie, as do all requests to the netflix.com domain, and some of the integrity and authentication mechanisms specified in the clientaccesspolicy.xml in the form of custom HTTP headers presented in Table 3.5.

The body of the POST request contains the NCCP message exchange of the player as shown in Listing 3.1 in abbreviated form. The complete Listing is presented in Listing A.1.

```xml
<?xml version="1.0" encoding="utf-8"?>
<nccp:request xmlns:nccp="http://www.netflix.com/eds/nccp/2.10">
  <nccp:header>
    <nccp:softwareversion>2.876.642.1</nccp:softwareversion>
    <nccp:certificationversion>1</nccp:certificationversion>
    <nccp:preferredlanguages>
      <nccp:appselectedlanguages>
        <nccp:language>
          <nccp:index>1</nccp:index>
          <nccp:bcp47>en</nccp:bcp47>
        </nccp:language>
      </nccp:appselectedlanguages>
    </nccp:preferredlanguages>
    <nccp:payload encrypted="true">
      AhC5NPMRretsOhxjyz9sYXShgJCyDn6qNX20+a4L2uRv0SiWlv8TNGq ...
      </nccp:payload>
  </nccp:header>
  <nccp:register>
    <nccp:idcookiereg>
      <nccp:payload encrypted="true">
        AhAB1lkyZLqjLbx3BAgRHyMSg3A++r7EsW5o1cZcuaK+r8W/KyIltELV ...
      </nccp:payload>
    </nccp:idcookiereg>
  </nccp:register>
</nccp:request>
```

Listing 3.1: NCCP message in the player registration request.

| **Parameter** | **Value** |
|---|---|
| X-AuthenticationType | ShopperID |
| X-ESN | SLW32-TXD06LNCZZE1QW7GFL3NENUWNM |
| X-HMAC | EM6I/6ziHEIBA3BSjTzN4GGaJAaN0P+gFCBATFnZUts= |
| X-ShopperID | (Identical to NetflixId cookie) |

Table 3.5: Custom HTTP header with authentication and integrity parameters in the registration request.

The payloads in the <nccp:header> and in the <nccp:idcookiereg> are base64 encoded and encrypted and, presumably, contain unique information to identify and register this particular client. Replaying the request with any modification to the data in the message body or to the X-HMAC header causes the server to reply with:

HTTP/1.1 500 Internal Server Error

Replaying the request unmodified but with a delay of a few minutes causes the server to send the reply shown in Listing 3.2.

```
<nccp:result method="register">
    <nccp:status>
        <nccp:success>false</nccp:success>
        <nccp:error>
            <nccp:code>4005</nccp:code>
            <nccp:description>Clock skew</nccp:description>
            <nccp:actionid>2</nccp:actionid>
        </nccp:error>
    </nccp:status>
</nccp:result>
```

Listing 3.2: Clock skew NCCP error message due to delayed request.

The response of a successful registration request contains an NCCP message with encrypted payloads and protected by the HMAC, in a way similar to the request. In Listing 3.3 we have omitted the encrypted payloads, and in line 17 we can see the result of the operation indicated in line 8.

```
<HMAC>q8jqRVFzJTaK1BxuCp4KYnaSbWl10Co2XWEe7h28NUk=</HMAC>
<?xml version="1.0" encoding="utf-8"?><nccp:response xmlns:nccp="http://www.
    netflix.com/eds/nccp/2.10">
    <nccp:responseheader>
        <nccp:payload>
            base64 encoded encrypted payload
        </nccp:payload>
    </nccp:responseheader>
    <nccp:result method="register">
        <nccp:registrationdata>
            <nccp:payload>
                base64 encoded encrypted payload
            </nccp:payload>
            <nccp:userid>Carlos P</nccp:userid>
            <nccp:userdescription/>
        </nccp:registrationdata>
        <nccp:status>
            <nccp:success>true</nccp:success>
        </nccp:status>
    </nccp:result>
...
</nccp:response>
```

Listing 3.3: Portion of the NCCP response to a successful registration request.

The registration request is important for Netflix to be able to restrict the number of players associated with each particular account. The limit of 6 active devices – an installation of the Silverlight Netflix player in a user account of a computer counts as one device – has recently been increased to 50 by Netflix. Every attempt we made of intercepting and blocking either the request or the response, replaying requests to the server, replaying responses to the client, in an attempt to bypass the registration of new client were detected by the client and resulted in an error message being displayed.

On the same machine and OS user account, signing out of the Netflix account and signing in with a different one causes the player to issue a new registration request, registering itself in the latter.

During the course of this analysis Netflix has changed their device management page[7] so that it no longer lists information about every individual device. We have successfully registered 50 devices, at which point Netflix's device management page shows the warning in Figure 3.7. Nevertheless, we were allowed to continue registering new devices and old devices continued to be allowed to play content. It is not clear whether Netflix has decided not to enforce this as a hard limit during an experimental period of the new limit or this is an implementation flaw.

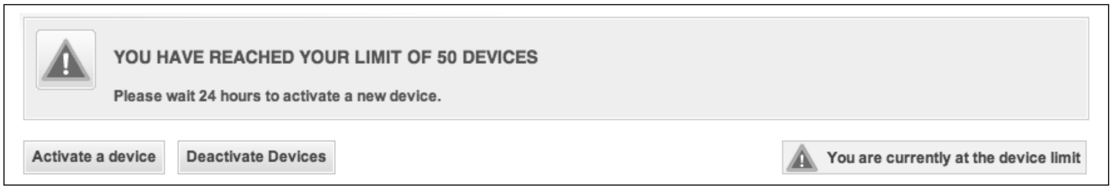


Figure 3.7: Warning displayed when the 50th device is registered with a single user account.

From what we have observed in the previous message exchanges, we can formulate a series of hypothesis that we believe to be true:

1. The X-HMAC custom header protects the integrity of the body of the POST request – modifying the X-HMAC value or the contents of the message produces the same error.
2. The message data contains clock information – replaying delayed messages results in a `Clock skew` error.
3. The player registration data contains identifiers that are unique to this particular Silverlight player installation – every registration request payload is different and uniquely identifies this client in the Netflix Authorized Devices list.

### 3.4.3 Client authorization

After registration, the client issues an authorization request to `https://agmoviecontrol.netflix.com/nccp/controller/2.11/authorization`, once again using NCCP. Unlike registration, which only occurs on the first run of the client, authorization is done every time the client runs to play a movie. The NCCP message in the request is protected in a similar way as the registration request. HTTP request now contains two new custom headers and these are protected by an HMAC, presented in Table 3.6. Any modifications to either of these headers results in a server error.

This is the first time we see the `CTicket` being used and it appears in a client originated request. Either:

1. it is generated by the client; or
2. it was given by the server in one of the previous encrypted interactions.

From numerous experiences we have done, we concluded that:

[7] `https://account.netflix.com/Player?manage_device=0`

| Parameter | Value |
|---|---|
| X-HMAC | WOa5iA6k+jLMqMeKERFjo0Nq3rfXCb6CapUs2nRqrhw= |
| X-AuthenticationType | Cticket |
| X-CTicket | AQAAAAEEEBpdm/ifDiLAHPEGRxosZRWA4PESyWw7byt2PznK Y7iVeBhA3RRh6hRtQAbrWOyFCDgmcgNr+CSA6daQMSq1XI/H7 15oZ6PWR+BsBtimgHaE2uqVmZExFa2ktLJGm1uajOFuK1VoehY hpGPkdYojeybamvdTh5Jciz+BjquQAfkOFnVdJd29+hap6EGGwrN AwgtCqfwdGo/WAcivlnj6aArWqowqsEKwx22edc+ozP92uROpOm RFAs3OGBPXGzythM8cTiDa7sUl68UkQLlfUM7DjBtcm7HTAUwL GTG+bqisVhwGt+Crs7IgEIsyVERaPNtU |

Table 3.6: Custom HTTP headers authenticating the authorization request.

- Signing in to a different Netflix account triggers authentication renewal, device registration and a new CTicket is used.
- Deleting the Isolated Storage folder, where the Silverlight based player locally stores data, causes the player to re-register (no authentication renewal) and use a new CTicket.
- After 12 hours the CTicket expires. The server responds with the message in Listing 3.4 and triggers an authentication renewal request, and then the CTicket is renewed.

CTicket is probably given to the player when it registers, it is stored in Isolated Storage and is renewed when it expires or is invalid – authentication renewal request to `https://agmoviecontrol.netflix.com/nccp/controller/2.10/authenticationrenewal`.

```
<nccp:error>
    <nccp:code>4003</nccp:code>
    <nccp:description>CTicket expired</nccp:description>
    <nccp:actionid>5</nccp:actionid>
</nccp:error>
```

Listing 3.4: Portion of the NCCP response to an authorization request with expired CTicket.

All attempts we made to block requests or responses, replay messages, with the objective of bypassing the authorization were detected by the client or the server and resulted in errors.

A successful authorization response contains instructions on how the client should access the movie, as can be seen in Appendix A.2. The content can be fetched from a number of Content Delivery Networks (CDN) and the instructions contain the URLs that can be used for each of the CDN. For Level3 for example, the URL is

`http://nflx.i.80ed7672.x.lcdn.nflximg.com/685/941879685.ismv?etime=20110625044831&movieHash=715&encoded=05a03c23763faca66e1c4`

The client then makes requests to the CDNs by issuing HTTP requests with the form

`http://nflx.i.80ed7672.x.lcdn.nflximg.com/685/941879685.ismv/range/0-48269?etime=20110625044831&movieHash=715&encoded=05a03c23763faca66e1c4&random=1650549099`

From our experiments, we have determined the following for each of the parameters in the URL:

- **462565273.ismv** – is a video file chunk in Microsoft IIS Smooth Streaming Media Video format, which is based on the ISO MP4 standard file format with some modifications to the organization schema [35].
- **range/0-48269** – is the byte range of the vide file in the chunk being fetched. Changes as the client requests for different parts of the movie.
- **etime=20110625044831** – is the expiration date of the URL and it is always the same as the client requests movie chunks. This URL, for example, will be valid until 25 Jun 2011 04:48:31 GMT (8 hours after the authorization request).
- **movieHash=715** – three last digits of the Movie ID and it is always the same as the client requests movie chunks. Movie ID was in the URL when the user selected the movie in the movie catalog.
- **encoded=05a03c23763faca66e1c4** – is a MAC that protects the other parameters in the URL except: **range** and **random**. It is always the same as the client requests movie chunks.
- **random=1650549099** – is different for every request made by the client. Its function is either to disambiguate movie chunk requests made by different clients that requested authorization at the same time; or, more probably, to avoid cached responses by caches in Internet Service Providers.

The client also fetches audio chunks with .isma extension, instead of .ismv. Modifying any of the parameters in the URL – except **range** and **random** – or making the request after the expiration time causes the CDN server to reply with an error:

HTTP/1.1 401 Unauthorized

To determine whether the **encoded** parameter is generated on the client side or the server side, we signed in to two different user accounts and we got the players in two separate machines to issue the authorization requests at the same time. They both got the instructions with the same **encoded** parameter. Apart from the **random**, both clients started using the same URLs to fetch the movie chunks:

**Client A**:

```
http://nflx.i.80ed91bb.x.lcdn.nflximg.com/592/925963592.ismv/range/
0-57857?etime=20110920244132&movieHash=640&encoded=0c7324abb5bac31c91d02
&random=1896822204
```

**Client B**:

```
http://nflx.i.80ed91bb.x.lcdn.nflximg.com/592/925963592.ismv/range/
0-57857?etime=20110920244132&movieHash=640&encoded=0c7324abb5bac31c91d02
&random=779206362
```

This result leads us to believe that the MAC parameter **encoded** is generated by the authorization server and is not dependent on any client side information.

No other type of authorization validation is done by the CDNs. Anyone with the valid URL can request movie chunks without any restrictions until the URL expires. This means that after the client is authorized and is given the correct URL to fetch the file, what prevents unauthorized clients from displaying the movie is the DRM encryption.

## 3.5 Stream encryption

When the client is playing a movie and stops, it sends HTTP POST requests with NCCP messages to `http://movies.netflix.com/nccp/controller/2.10/playdata` and `http://movies.netflix.com/nccp/controller/2.10/logblob`. These signal the server that the client has stopped streaming, the position in the movie at which it stopped and other statistics. The next time the user selects this movie it will resume from the point at which it previously stopped.

Nevertheless, the first chunks of video and audio the player requests are always in the beginning of the movie range. It does this because the first bytes of the movie contain headers that are needed to determine if the movie is encrypted and, if it is, to obtain the key required for decryption.

```
<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>AAAAADk67SMAAAAAAAAAAA==</KID>
        <CHECKSUM>6AmJA9oC3Xw=</CHECKSUM>
    </DATA>
</WRMHEADER>
```

Listing 3.5: Header in the first bytes of the video file.

In Listing 3.5 we can see an example of the header in a video file. The header informs the player that:

1. The video file is protected with Microsoft PlayReady DRM [21].
2. The movie is encrypted.
3. The key is 16 bytes long (128 bits).
4. The cipher used to encrypt the movie was AES in counter mode (CTR).
5. The Key ID of the key the player needs to fetch.

If this is the first time the client is playing PlayReady DRM protected content, it will contact Microsoft servers and perform what Microsoft calls the *Individualization* process. It will do an HTTP

POST request to `http://services.silverlight.microsoft.com/PlayReady/FW-I81/default.freeway?Individualize` with client data:

PostType=DrmIndivAcquire&SecurityVersion=5.0.0.0&Platform=0&
Architecture=0&ClientSdkType=1&ClientId=
t4LqFbAyzAEkAAAAAAAAAAEAAAAAAAEAAAABAAEAAQCcnwAAYgXxnileYFY%3D

Player individualization, according to [22] and [20] is the process of acquiring a software component, the Individualized Black Box (IBX), that is embedded into the Silverlight plug-in to handle requesting licenses and protecting sensitive data used in the decryption process.

Replaying the same individualization request multiple times always results in a different Individualized Black Box being downloaded. The IBX contains randomized portions that uniquely identify each particular instance, but as far as we know, Microsoft does not provide details of the component inner workings.

After the individualization process, the client sends an NCCP message requesting a license to `https://agmoviecontrol.netflix.com/nccp/controller/2.10/license`. The message is authenticated with the NetflixId cookie, the CTicket protected by the HMAC, and contains the NCCP encrypted payload. The NCCP response contains an encrypted payload of DRM data of more than 20kB, with the usual protection mechanisms. After the license has been successfully acquired, the player fetches the desired byte ranges from the CDN and starts displaying the movie.

The code for the Silverlight based client made by Netflix and the Individualized Black Box is obfuscated. Analyzing decompiled and disassembled code would take much more time than we had available. So we decided to pursue other options.

Movie chunks fetched from the CDN are always the same – for the same byte ranges – independently of what client fetches them, when it does it and from where it does it. We have verified that their SHA-1 hashes are the same when requesting from different countries, different machines and with a time difference of as much as two months. This means that the AES encryption key to decrypt a particular movie is always the same. If the key is known, it is simple to build a client in Silverlight, provide the key locally and fetch the movie chunks as long as the URL is valid.

We tried using `aeskeyfind`[8], which is a program that can be used to analyze memory dumps and retrieve candidate AES keys, based on the AES key scheduling data structures [13]:

1. We accessed Netflix.com and started watching a movie on a virtual machine.
2. We paused the virtual machine, causing its memory to be saved to a file.
3. We ran `aeskeyfind` on the memory file.

This process generated a number of candidate keys, as shown in Listing 3.6.

With each of these candidate keys, we tried to decrypt the first movie chunk and see if we got an mp4 file chunk, but we were not successful. We also tried to decrypt – after base64 decoding – the NCCP payloads but without success. In the case of NCCP payloads we do not even know what encryption algorithm is being used, but we gave it a try. In conclusion: we did not succeed in breaking the encryption.

[8] `http://citp.princeton.edu/research/memory/`

```
eb724c7e8b4c15d2d6c39785af527143
ace6e5ed4737ae4b9a0dd232658bdfce
e175c9b65bab6a9d7a514c09d792cca6
eb724c7e8b4c15d2d6c39785af527143
5808eb8b1f54dfeb4f6a3b1bd025b7f8
5808eb8b1f54dfeb4f6a3b1bd025b7f8
ace6e5ed4737ae4b9a0dd232658bdfce
e175c9b65bab6a9d7a514c09d792cca6
```

Listing 3.6: Output of the aeskeyfind program ran on the virtual machine memory file to find AES encryption keys.

## 3.6 Simultaneous streaming

To prevent widespread sharing of credentials by users, Netflix enforces restrictions on simultaneous streaming. Using the same account it is possible to watch movies on two computers simultaneously. Trying to watch on a third computer with that account causes the player to display the error message in Figure 3.8.

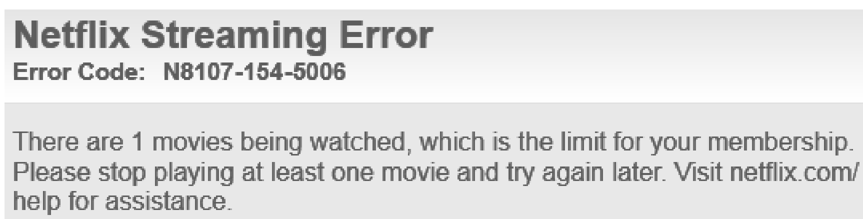


Figure 3.8: Error displayed by the Netflix player when attempting to play a movie on a third computer with the same user account.

The error occurs when the client issues the license request to `https://agmoviecontrol.netflix.com/nccp/controller/2.10/license`. Instead of replying with `success` as in Listing 3.7 and granting the license, the server replies with the error in Listing 3.8.

```
<nccp:status>
  <nccp:success>true</nccp:success>
</nccp:status>
```

Listing 3.7: Success code when license is granted by the license server.

We ran several experiments with simultaneous streaming and concluded that the license server grants at most two outstanding licenses for each account. To be able to obtain a new license for that account one of two things has to happen:

1. The server has to receive the HTTP POST request to `http://movies.netflix.com/nccp/controller/2.10/logblob`, mentioned in Section 3.5, with data signaling that the player has stopped displaying the movie.
2. That user must wait until the server side timeout expires, which we have measured to be 36 minutes.

```
<nccp:status>
          <nccp:success>false</nccp:success>
          <nccp:error>
              <nccp:code>5006</nccp:code>
              <nccp:description>Maximum number of concurrent streams</
                  nccp:description>
              <nccp:actionid>3</nccp:actionid>
              <nccp:reasoncode>102</nccp:reasoncode>
              <nccp:usertext>
                  <nccp:bcp47>en</nccp:bcp47>
                  <nccp:text>There are 1 movies being watched, which is the
                      limit for your membership. Please stop playing at least
                      one movie and try again later. Visit netflix.com/help for
                      assistance.</nccp:text>
              </nccp:usertext>
          </nccp:error>
</nccp:status>
```

Listing 3.8: Error code when license is denied by the license server due simultaneous streaming.

We assume that Netflix is allowing two outstanding licenses, instead of just one, to avoid too many problems with clients crashing and not sending the signal to the license server that they have stopped streaming. A user that experiences a client crash while watching a movie can simply re-launch the client and resume watching with the second outstanding license. If there could be only one license outstanding, that user would have to wait 36 minutes until a new license could be granted.

## 3.7 Geolocation

Netflix enforces geographical restrictions on who can access their content because it is only licensed to stream to customers inside the United States and Canada. As we have seem in Section 3.2, Netflix determines the location of the user based on origin IP address and prevents the user from accessing the main homepage and the account creation page (message presented in Figure 3.1). It also prevents users from reaching the customer sign-in page at `https://signup.netflix.com/` and the *Watch Instantly* catalog page at `http://movies.netflix.com/WiHome`. It redirects the user and displays the message depicted in Figure 3.9.

An authorization request to `https://agmoviecontrol.netflix.com/nccp/controller/2.11.2/authorization` is also unsuccessful and the server replies with the error presented in Listing 3.9.

We have also experimented with the `X-Forwarded-For` HTTP header, which is the de facto standard for identifying the origin IP address of the client when it connects via an HTTP proxy[9]. Connecting from Portugal with a forged `X-Forwarded-For` header with an IP address from the United States did not cause Netflix's servers to accept the connection as if it was originated in the U.S.

However, Netflix is not imposing geographical restrictions on the license granting process, and additionally the CDNs do not perform any kind of geographical validation. This means that a user

[9] `http://en.wikipedia.org/wiki/X-Forwarded-For`


Figure 3.9: Error displayed by the Netflix player when an authenticated user tries to access the *Watch Instantly* page from outside the United States.

```xml
<nccp:result method="authorization">
      <nccp:status>
          <nccp:success>false</nccp:success>
          <nccp:error>
              <nccp:code>5008</nccp:code>
              <nccp:description>Service not supported at this location</
                  nccp:description>
              <nccp:actionid>3</nccp:actionid>
              <nccp:reasoncode>104</nccp:reasoncode>
              <nccp:usertext>
                  <nccp:bcp47>en</nccp:bcp47>
                  <nccp:text>NO_STREAMING_FROM_LOCATION</nccp:text>
              </nccp:usertext>
          </nccp:error>
      </nccp:status>
</nccp:result>
```

Listing 3.9: Error code sent by the authorization server when the request is made with a foreign source IP address.

that is able to authorize the player and obtain the license can then watch the movie from anywhere, without limitations. It allows users with VPN services that assign IP addresses from within the United States, or users that use Tor[10] with an exit node in the United States, to be able to browse the movie catalog, launch the player, properly perform authorization and obtain the license. Then, when the player starts fetching the movie chunks from the CDN, which is very demanding in terms of bandwidth and amount of data, they can simply close the VPN or Tor connection and continue to watch the movie. This way it is possible to get around Tor's bandwidth limitations and most VPN services' data and bandwidth constraints.

## 3.8 Output protections

Silverlight with PlayReady DRM supports output protections on Microsoft Windows XP or newer and on Apple Macintosh [20]. These can be used to limit the quality or the ability to display movies on graphics cards and monitors that do not support various standards for content protection, such as HDCP, CGMS-A, ACP and SCMS. The goal of these technologies is to prevent unauthorized reproduction of the video content.

We conducted experiments with Windows and Macintosh computers and were successful in displaying movies from Netflix on digital and analog external monitors. This leads us to conclude that

[10] `http://www.torproject.org/`

Netflix is not enforcing output protections.

## 3.9 Mobile client application

Netflix has software clients available for Android and Apple iOS mobile devices. We used a smart phone running Android to conduct our experiments with the Netflix application available in the Android Market.

A company called ViaForensics has done a security analysis[11] of Netflix's Android client and found that the application stored passwords in clear text in the device. The application has since been updated and we have not been able to find passwords stored in clear text in version 1.3. We have rooted[12] the phone, which is the process that allows the user to run applications with root privileges, and searched the entire file system for the password we configured in the application, but the search did not find any.

We analyzed the traffic between the mobile application and the Netflix's servers. All traffic containing authentication credentials and tokens is protected with SSL during the sign in process, catalog browsing and authorization process. Unencrypted interactions do not contain authentication information and they occur in three phases:

- Downloading images containing movie art from CDNs;
- Downloading the movie chunks containing video and audio from CDNs.
- Final OAuth (Open Authentication) exchange [15].

User authentication in the sign in process is done via the OAuth protocol. After sign in, the client is given the `NetflixId` cookie, like we observed with the web browser client in Section 3.3. This authentication cookie is always transmitted protected by SSL, which is different from what was observed with the web browser client on Windows and Mac OS X machines. This means that as long as the SSL's assumptions mentioned in Section 3.3 hold true, Netflix's Android mobile client is not vulnerable to cookie stealing by network eavesdroppers.

The authorization and license request interactions are done via the NCCP protocol, as seen with the browser client. The difference is that instead of contacting an authorization server and then a license server, all NCCP interactions are done with a server located at `https://nccp-spyder.cloud.netflix.net`. Although were not able to decrypt the contents of the NCCP messages, we presume that this server acts as an application proxy that talks to the authorization and license servers. One significant difference we observed, when comparing to the browser-based client, it that the mobile client does not issue an Individualization request to Microsoft's servers.

The movie chunks are fetched via HTTP and using URLs with the same format and characteristics we described in Section 3.4.3. The requested files are smaller in size, due to the reduced image resolution, but they are similar in every other respect.

[11] `http://viaforensics.com/appwatchdog/netflix-android.html`
[12] `http://en.wikipedia.org/wiki/Rooting_(Android_OS)`

The restrictions we have observed using the web browser client are also imposed on the Android mobile client:

- Geographical location restrictions are also enforced on the mobile client:
  1. Users connecting from outside the United States can reach the sign in screen, but after entering their credentials the client is unable to authenticate with Netflix.
  2. Signing in with a U.S. IP address using a VPN, disconnecting the VPN and trying to begin streaming with a Portugal IP address, causes the application to display the error message:

     NO_STREAMING_FROM_LOCATION
  3. If we sign in with a U.S. IP address, launching the movie and then disconnecting the VPN, streaming continues with the application fetching movie chunks from the CDN with an IP address from Portugal.
- Simultaneous streaming restrictions are done in the same way as described in Section 3.6; the mobile application counts as a streaming device and consumes one license.

## 3.10 Security evaluation

### 3.10.1 Attacker model

To be able to assess what kinds of attacks are possible, we should first define the capabilities of the attacker. We assume that an adversary is able to observe the traffic flowing between the OTT provider and the customer's machine. There are several ways an adversary can achieve such observation powers, including several techniques to perform man-in-the-middle attacks, but the simplest attack is to just be on the same open Wi-Fi network as the victim and passively sniff all packets in the network. Open Wi-Fi environments are common nowadays in university campuses, airports, restaurants, hotels and other public places. On wired switched LANs it is considerably more complex to achieve these monitoring capabilities and may require tools to perform MAC flooding and ARP spoofing to trick switches into sending the traffic to the attacker. Man-in-the-middle attacks would allow the adversary to go beyond passive packet sniffing and engage in active attacks, tampering and forging data to cause mischief. While not trivial to perform, tools do exist to facilitate man-in-the-middle attacks on wired and wireless LANs.

The adversary may also be able to gain temporary physical access to the victim's machine. This can happen in many common scenarios: shared computers in a household, in work and education environments, demo machines in conferences and workshops, and many other situations. Non-security conscious users frequently do not take precautions to protect their machine accounts and share it with others. With the logged on user's privileges, an attacker with access to a machine and a logged in account can steal browser cookies without the owner of the machine noticing. For all modern versions of Microsoft Windows, it is possible to construct an `Autorun.inf` file that runs an executable or a batch file. This functionality can be exploited to copy the cookies stored in the user profile to a USB. Although Microsoft has updated the AutoRun functionality on removable drives, making it more difficult to exploit by disabling it on USB flash devices [23], it is possible for a USB flash drive to present itself to the operating system as two drives: one emulates CD media with AutoRun enabled; the other is a writable file system, like in a regular USB flash drive. One such implementation is the U3, a technology of SanDisk[13]. Tools exist to modify the CD file system of U3[14]. An adversary can use such tools to construct an `Autorun.inf` file in the CD file system that copies the cookies to the flash USB writable file system. This procedure would allow theft of all the persistent cookies in the cookie store of the targeted browsers.

On the side of the OTT provider, adversaries may be able to find Cross Site Scripting vulnerabilities [18] and leverage them to steal cookies from the customers' browsers. Considering the 2011 CWE/SANS Top 25 Most Dangerous Software Errors rank [32], we must assume this to be a realistic possibility.

In the case of video services, like the OTT providers we are analyzing, the user with full control of the machine can engage in malicious activities. We have to assume that users of the service can collude with others in unauthorized ways. A user with a legitimate account can share credentials with others, can send captured traffic to be replayed and can generally coordinate actions to defeat the provider's authorization mechanisms. With full control of the machine, a user may save the movie data, record the video signal that is available on analog or digital video outputs and distribute it to others.

With full control of the hardware and software used to access the server, the adversary can also decrypt SSL communications, save messages for later use, inject forged messages and perform any software modifications to the client.

Furthermore, the adversary may be located anywhere in the world, whether he/she is a customer of the service or not. VPN services available on the Internet, the Tor project network and any other tools that anonymize or assign IP addresses from different locations can also be used by an adversary to make it appear to be in a different location.

### 3.10.2 Vulnerabilities and weaknesses

In Section 3.2 we were able to bypass the geographical location restrictions for registering a new account. Nevertheless, this still did not give us access to play video content. As seen in Sections 3.7 and 3.9, geographical location is done and enforced in several places, including the *Instant Watch* homepage that gives access to the movie catalog and in the authorization phase. Using VPN connections that assign to the user's device IP addresses located in the United States, it is possible to bypass the geographical restrictions. The user can even user the VPN to login and authorize the player, and then disconnect the VPN, since the CDNs do not enforce geolocation restrictions.

Mobile smart phones, like the ones running Android, generally offer location capabilities that go beyond the IP range of the assigned address. They can determine location by using the mobile

[13] `http://u3.sandisk.com/`
[14] `http://u3-tool.sourceforge.net/`

network or the GPS system. Nevertheless, the Netflix mobile application did not take advantage of these capabilities. One reason for this behavior might be the fact that client side validations can be subverted by the attacker, who controls the hardware and can modify the software at will.

In the sign in, session management and sign out mechanisms we presented in Section 3.3, the following weaknesses and vulnerabilities were identified:

1. `NetflixId` and secure cookies are persistent even when the user does not select `Remember me on this computer` – allows user $B$ to sign in as user $A$ if user $A$ is convinced that the browser will not remember the session by simply quitting the browser.
2. `NetflixId` cookie is not Secure and is sent over HTTP in plain text – allows an eavesdropper on the network to steal the cookie and impersonate the user to the website.
3. `NetflixId` and secure cookies are not HttpOnly – allows an adversary to leverage Cross Site Scripting vulnerabilities [18], if they are found on the Netflix website, to steal the cookies and impersonate the user to the website.
4. `NetflixId` and secure cookies are not invalidated on the server side when the user explicitly signs out – allows an attacker that has stolen the cookie to continue to impersonate the user, even if the user explicitly signs out of the site. A user that suspects or knows for a fact that the authentication cookie has been compromised cannot force it to be invalidated by the server.

To assess the difficulty of performing the cookie stealing attack allowed by vulnerability number 2, we ran the experiment described in Section 3.10.3. An adversary only needs to be able to eavesdrop the network communications. Unencrypted Wi-Fi hotspots, or open Wi-Fi hotspots, are common in restaurants, parks, hotels, university campuses and other public places. It is easy to sniff traffic and steal cookies from other users by employing packet sniffing software. The cookie would also be sent to an attacker that performed DNS cache poisoning attacks on the DNS server used by the victim.

As described in Section 3.9, Netflix's mobile client for Android devices is not vulnerable to this attack because the authentication cookie is never sent in the clear. All traffic containing authentication credentials and tokens is protected by SSL, making the attack infeasible.

Because of vulnerability number 4, the user whose cookies have been stolen by an attacker will have no way of invalidating the authentication cookie for Netflix, whether it has been stolen via a USB flash drive or the network.

Although we were not able to successfully break the encryption used by Netflix to protect the movie assets and the NCCP protocol payloads, as described in Section 3.5, the keys must be residing in memory for the client application to work. An adversary with more time to analyze the client software, and possibly by reverse engineering it, may be able to develop a method for retrieving the keys. However, even if an attacker is able to recover the keys and distributes them to others, the necessary movie chunks are only available for download from the CDNs through URLs that are valid for 8 hours only, as seen in Section 3.4.3. There are plenty of examples on how to build Silverlight players to decrypt video streams on the Internet, provided the key is available, and therefore the attack would be doable.

Access to the cryptographic keys in the client application would also give the adversary the ability to break the integrity of the messages in the NCCP protocol. The adversary would be able to generate the correct HMAC for a forged message. Whether this could have serious consequences to the security of the service we cannot assess.

A much easier way to illegitimately share the movie with others is to capture it from unprotected outputs. As shown in Section 3.8, the Netflix Silverlight based client is not enforcing output protections, despite the capabilities offered by the technology. We can speculate that due to the multitude of hardware configurations in use that do not implement the standardized output protection mechanisms, Netflix and the content providers do not wish to impose a high barrier for legitimate use.

As of version 4, Silverlight does not have explicit support for client side watermarking of the images it outputs. As mentioned in Section 3.5, the movie chunks that are downloaded from the CDNs are identical for all users. This leads us to believe that Netflix would not be able to identify the malicious users that record the movie and distribute it: no user identifying information is present in the video output.

To limit unauthorized sharing of user credentials, Netflix relies on the restrictions it imposes on simultaneous streaming. As we've seen in Section 3.6, no more than two streams are allowed simultaneously for each account. Netflix used to impose a hard limit of 6 active devices for each account. As users began hitting the limit more often, due to the proliferation of Netflix capable devices like mobile phones, tablets, gaming consoles, Internet connected TVs and others, Netflix decided to increase the limit to 50 devices. As described in section 3.4.2, even after reaching the limit we were able to continue to register new devices and use them. We are lead to conclude that simultaneous streaming restrictions are the main mechanism used to prevent widespread sharing of credentials.

### 3.10.3 Cookie sniffing simulation

To demonstrate the cookie stealing attack on a network, we setup a Wi-Fi network with no encryption and used two portable computers to connect to it in order to access the Internet. This network was setup specifically for this demonstration and in an environment where other computers/users were not likely to connect (certainly not unintentionally). The Netflix account we used was specifically setup for the purpose of this study. On computer $A$ – the attacker – we ran Wireshark, configured to capture packets in promiscuous mode and in monitor mode.

On computer $B$, we used a web browser to navigate to the Netflix website and sign in. After signing in we browsed the movie catalog and accessed the movie queue.

HTTP traffic to and from computer $B$ was captured by Wireshark running on computer $A$. By inspecting some of the captured HTTP requests we found a request to `http://www.netflix.com/Beacon/Clear.gif?beacon=true&lnkce=wiz-lyrc&ts=1310093265824` containing the `NetflixId` cookie.

By importing this cookie to a web browser in computer $A$, we were able to sign in to the Netflix website without having to provide user credentials, thereby demonstrating the attack.

# Chapter 4

# Hulu Plus

Hulu[1] is an over-the-top video service provider that streams programming from several television networks in the United States. Hulu's web streaming service uses a custom player built with Adobe Flash technology and most of the movies and TV shows are available free of charge. Hulu also provides a paid subscription service called Hulu Plus that offers access to additional content, HD content and the ability to watch on smart-phones, tablets, gaming consoles, Internet connected smart TVs, Blu-ray players and set-top boxes[2].

Similarly to what has been done with Netflix, we set out to understand the security mechanisms implemented by Hulu to enforce the security properties presented in Section 2.1. We observed the interactions of the web browser client with Hulu's servers, from the account creation process, to browsing the content catalog, selecting and watching the video. As we learned about the implementation details, we tested their effectiveness and tried to find flaws. In the following sections we describe what we observed and learned as our analysis progressed. The Chapter ends with a security evaluation of the mechanisms observed.

## 4.1 Account creation

To create a basic free Hulu account, users must provide some personal information:

- Email address
- Password
- First and last name
- Date of birth
- City, State and Zip code

[1] `http://www.hulu.com/`
[2] `http://www.hulu.com/plus/devices`

- Gender

To defend against automated account creation by bots, users also have to complete a CAPTCHA [6] challenge. To create a Hulu Plus account, which allows access to paid content, the following additional information is needed:

- Billing Address
- Credit card details: type, number, security code and expiration date.

When Hulu.com is accessed from an IP address outside the United States, the message in Figure 4.1 is shown and the user is warned that videos can only be accessed from within the United States. Despite the warning, the user is allowed proceed to the main webpage, can create an account and can sign in. No restrictions are imposed at this stage based on the origin IP address. However, the credit card needed for the creation of a Hulu Plus account must be associated with an address in the United States. As shown in Figure 4.2, if in the provided information the billing address is not associated with the credit card, the account creation process is interrupted and is not allowed to proceed.

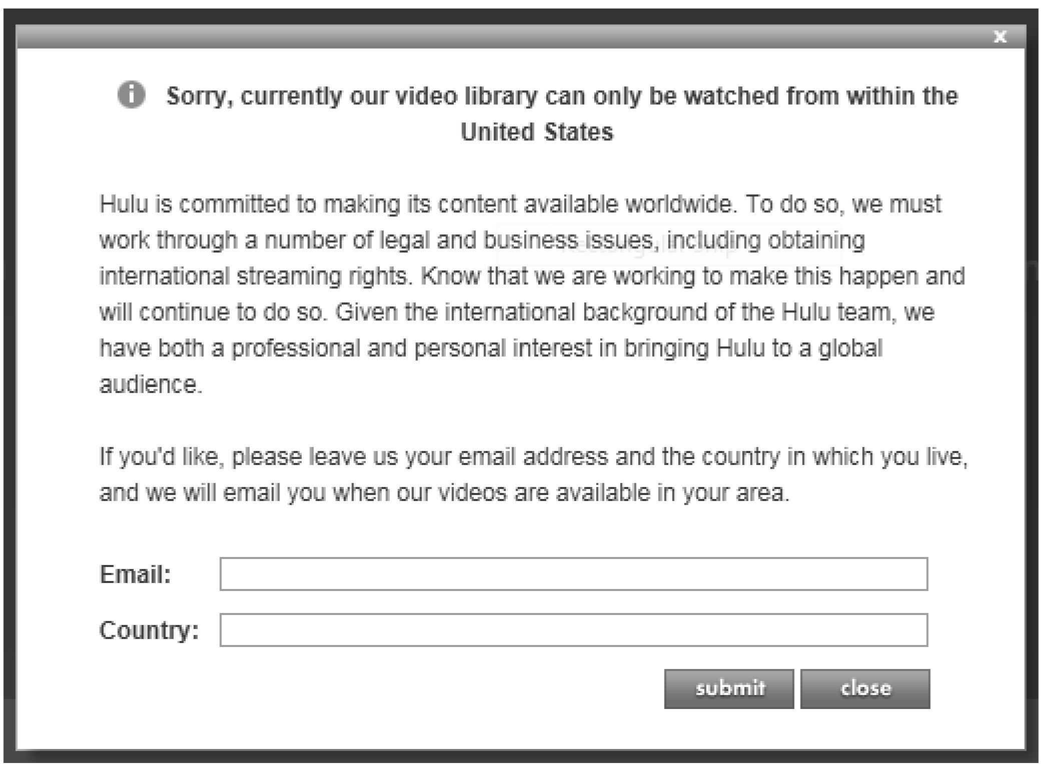


Figure 4.1: Message displayed by Hulu.com when accessed from an IP address outside the United States.

The address and credit card verification presents a significant barrier to keep users from outside the United States from being able to register for Hulu Plus accounts. Cards created through online

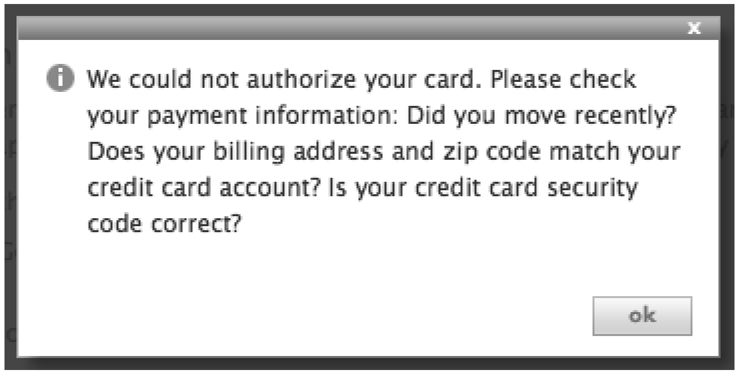


Figure 4.2: Error displayed when the billing address is not associated with the credit card provided.

services that specialize in virtual credit cards, like Cliffs Card[3], Instant Virtual Credit Cards[4], Entropay[5], UnblockUs[6], Shop Shield[7], allowing users to create credit cards on the fly for use on online commerce, are blocked by Hulu Plus.

## 4.2 Authentication and session management

### 4.2.1 Sign in process

To log in, Hulu customers need to navigate to `http://www.hulu.com/` and click the "Log In" link. This will display a form where the user can type in the email address and password associated with the user account. The user also has the opportunity to specify whether the logged in state should be remembered for 30 days or just for the current session.

JavaScript in the webpage creates cookies with the user email address and password. These cookies are sent in a GET request to `https://secure.hulu.com/`. The HTTPS connection protects the information against network sniffing attacks.

Example:

```
GET https://secure.hulu.com/account/authenticate?862519903 HTTP/1.1
Host: secure.hulu.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:5.0) Gecko/20100101 Firefox/5.0
Accept: */*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Referer: http://www.hulu.com/?src=topnav
```

[3] http://www.cliffscard.com
[4] http://www.instantvirtualcreditcards.com/
[5] https://www.entropay.com/
[6] http://www.unblock-us.com/
[7] http://www.shopshield.net/

Cookie: login=address%40email.com; password=user_password; sli=1; __qca=P0-1394193266-1311090907803; __utma=1.1385679209.1311090906.1311090906.1311090906.1; __utmb=1.13.9.1311695168213; __utmc=1; __utmv=1.anonymous; __utmz=1.1311090906.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); _fb_notify=1; km_ai=user_id%3A25173203; km_ni=user_id%3A25173203; km_uq=; km_vs=1; _hulu_lg=1

The sli=1 cookie signals the server that the user has selected the "stay logged in" option. As illustrated in Figure 4.3 the server response sets session and user ID cookies and clears the log in credentials cookies from the browser by doing:

- Set-Cookie: login=; domain=.hulu.com; path=/account/authenticate; expires=Mon, 26 Jul 2010 09:48:35 GMT
- Set-Cookie: password=; domain=.hulu.com; path=/account/authenticate; expires=Mon, 26 Jul 2010 09:48:35 GMT

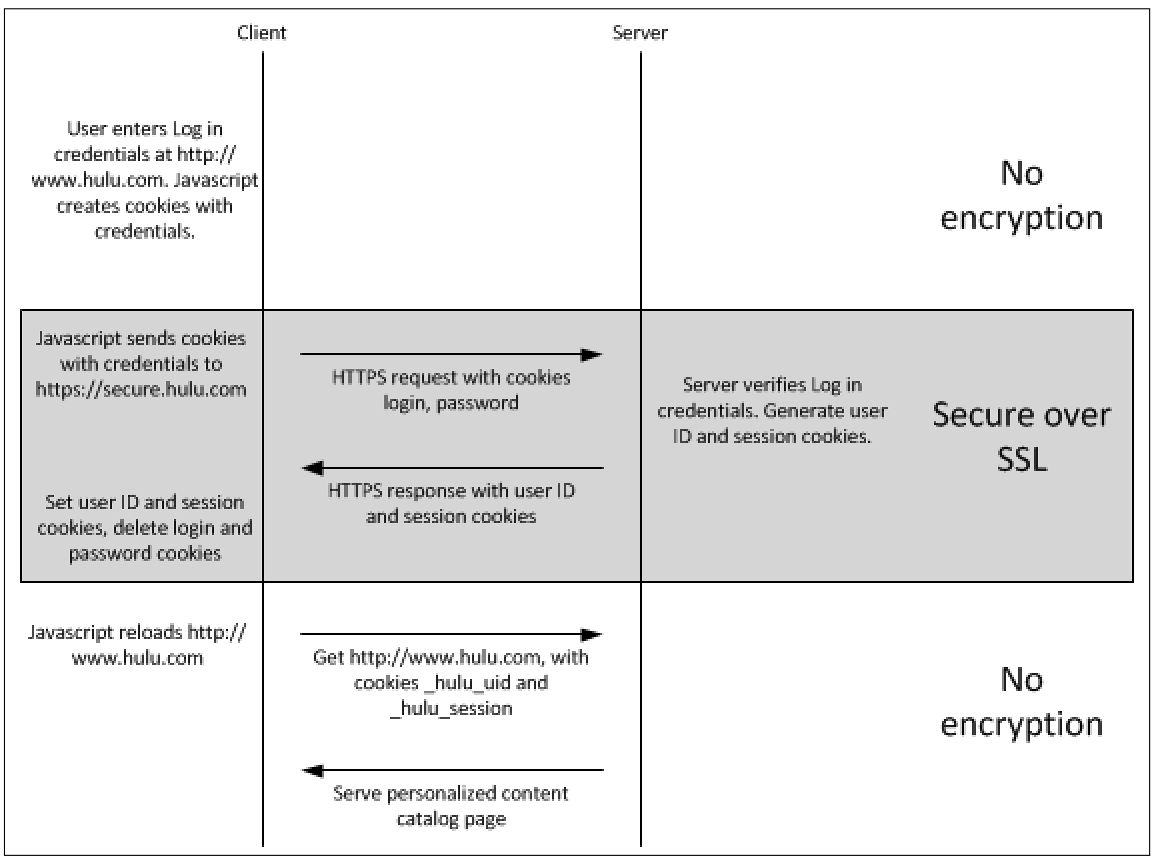


Figure 4.3: Hulu's Log in process using JavaScript to securely send credentials.

Alternatively, the user can log in using a Facebook[8] account. After associating the Hulu account with a Facebook account, whenever the user is logged in to Facebook, Hulu uses the Facebook

[8]http://www.facebook.com/

API to automatically authenticate the user. This is convenient, but because Facebook does not use HTTPS all the time by default, an adversary may be able to sniff Facebook cookies and use them to authenticate to both sites: Facebook and Hulu.

### 4.2.2 Session management

Different functions of the Hulu website depend on one or both of two authentication cookies set during the log in process: `_hulu_session` and `_hulu_uid`. Their typical content is shown in Table 4.1. These cookies are sent in the clear with every request the user makes while browsing the Hulu website for content, accessing Favorites or History of recently watched videos. Access to Favorites, for example, only requires the `_hulu_session` cookie. To access Hulu Plus content, both cookies are needed. As we saw with Netflix in Chapter 3, an adversary on the network may be able to capture these authentication cookies and impersonate the user. The cookies are not secure and are not HttpOnly, which also makes them vulnerable to XSS attacks, should such vulnerabilities be found in Hulu's website.

| Cookie name | Value |
|---|---|
| _hulu_uid | 25259243 |
| _hulu_session | T_MHeSdWK3ZYdZRLPLK7GA |

Table 4.1: User ID and session cookies used by Hulu to keep the user authenticated.

Some cookies are set to specific values to signal the website that the customer is a Hulu Plus customer, as shown in Table 4.2:

| Cookie name | Hulu | Hulu Plus |
|---|---|---|
| _hulu_p | – | 1 |
| _hulu_pgid | 1 | 3 |
| _hulu_plid | – | integer |

Table 4.2: Cookies are set to different values according to the type of customer, such as if he/she is a Hulu Plus or free Hulu customer.

Deleting all cookies from the browser cookie store except those listed in Tables 4.1 and 4.2 is sufficient for a Hulu Plus customer to continue to be authenticated and be able to watch video content.

For a user with basic Hulu access only, setting the cookies in Table 4.2 to the values in the right column causes the site to present pages and show content as if the user was a Hulu Plus customer. In Figure 4.4 we can see a screen requested with the cookies set similar to a Hulu Plus account, even though this user does not have this kind of account. The basic Hulu access screen looks like the one displayed on Figure 4.5.

### 4.2.3 Sign out process

When the user clicks on the “Log Out” link, the corresponding server response clears the authentication cookies from the browser cookie store:

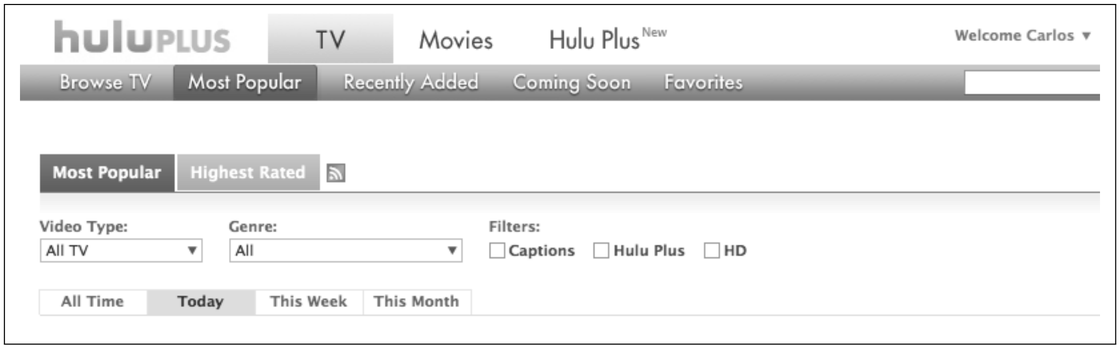


Figure 4.4: *Most Popular* content screen as seen on a Hulu Plus account.

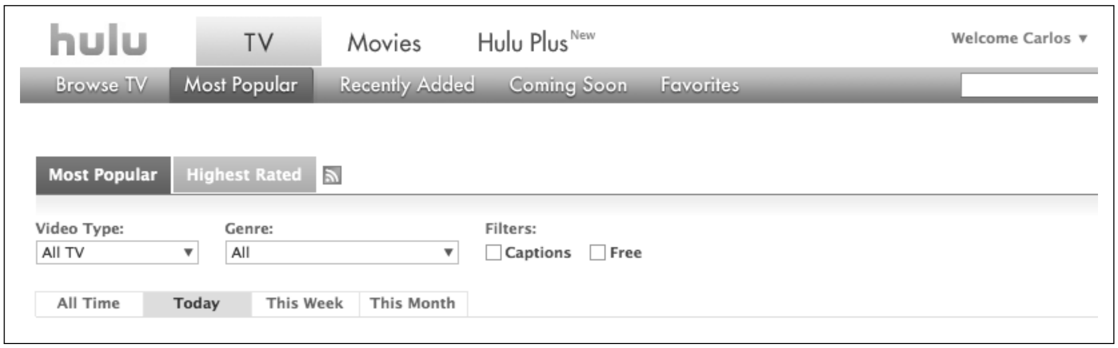


Figure 4.5: *Most Popular* content screen as seen on a basic Hulu account.

- `Set-Cookie: _hulu_uid=; domain=.hulu.com; path=/; expires=Wed, 28 Jul 2010 14:55:16 GMT`
- `Set-Cookie: _hulu_session=; domain=.hulu.com; path=/; expires=Wed, 28 Jul 2010 14:55:16 GMT`

These cookies are not invalidated on the server. As a consequence, if they are stolen by an attacker they continue to be valid for 30 days even after the user explicitly logs out. The user has no means to force invalidation of sessions on the server. Sessions continue to be valid even after the user password has been changed.

After 30 days, trying to use the authentication and session cookies to play a video causes the player to display the error message shown in Figure 4.6.

**Your user session has expired. Please reload this page to continue watching.**

Figure 4.6: Attempt to play Hulu Plus content with expired session cookie.

Reloading the page will ask the user to re-authenticate with email address and password.

## 4.3 Authorization

The cookies in Table 4.2 can be used to cause the website to display pages as if a user has a Hulu Plus account. It allows the user to select content available only to Hulu Plus customers, but when the Flash based Hulu player requests the video content from the servers, a stronger authorization is performed and the content is denied with message presented in Figure 4.7, suggesting other content.


Figure 4.7: Attempt to play Hulu Plus content on a basic Hulu account by setting the cookies shown in Table 4.2

.

When the user selects a Hulu Plus video to play, the browser loads the Flash based player, which then requests information about the CDNs where the content is located and requests the license to play the video before attempting to play it.

The request for the license is done over HTTP and carries the authentication cookies. A request looks as follows:

```
GET http://s.hulu.com/select?video_id=50066927&v=850037518&ts=1311966393
&np=1&vp=1&device_id=F6C960FD8614EDDAF5417DA6D26507C5&pp=hulu&
dp_id=hulu&region=US&language=en&bcs=87f8065c7d1faec28628a542848cde6c
HTTP/1.1
Host: s.hulu.com
```

If the user is allowed to play the video content, the response contains a blob of data approximately 25.5kB long. If the player is not allowed to play the content, the response will contain a much smaller blob. These blobs, shown in Appendix B, are encrypted and encoded, and should contain the key to decrypt the video and possibly other information like output protections and other usage rights. Without resorting to reverse-engineering techniques to analyze the Flash based player and how it decodes these data blobs, we were not able to determine what encodings and cryptographic techniques are being used.

Unlike Netflix, whose content is delivered via HTTP chunk transfer, in Hulu the encrypted content streams are delivered via Real Time Messaging Protocol (RTMP). RTMP is a protocol initially developed by Macromedia for streaming audio and video. It uses TCP port 1935 by default.

## 4.4 Simultaneous streaming

Another important access control function is limiting simultaneous streaming. A paying user could share access credentials with others and all users would access the content without limitation.

Limiting simultaneous streaming to one stream at a time for each user is one solution for this problem.

Hulu does not limit the number of simultaneous streams each user can request for free basic Hulu content. Logging in with the same credentials on multiple computers and requesting the same or different video streams is allowed. We have successfully verified this with up to 6 simultaneous streams on different machines. Trying to do the same with Hulu Plus content results in an error, as shown in Figure 4.8. Hulu allows at most two simultaneous Hulu Plus streams.

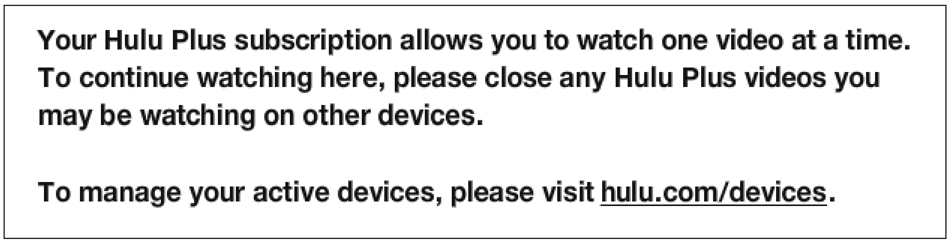


Figure 4.8: Attempt to play Hulu Plus content simultaneously on more than two computers
.

To restrict the number of simultaneous streams for each Hulu Plus account, the license server only grants two licenses every 30 minutes. By issuing two consecutive license requests by replaying the HTTP request shown in Section 4.3, we were able to receive two valid licenses – identical to the one shown in Appendice B.1, that the player receives prior to successfully playing the content. In response to a third request, the server does not return a valid license, sending instead a response similar to the one shown in Appendice B.2.

To grant another license, the server needs to receive a signal informing that the player has stopped displaying the video content in question. The player uses beacons to inform the server of various events, in particular that it is still receiving the stream and when it stops. Prior to requesting the license, as described in Section 4.3, the player requests instructions with the HTTP request:

```
GET http://t.hulu.com/config/v3/config?cb=1311966390001_775&
distro=hulu&distroplatform=hulu HTTP/1.1
```

The server responds with beacon instructions in XML format, as shown in Listing 4.1.

In lines 8, 11 and 25 it is possible to see that the player should send playback beacons to servers specified for each CDN and server indicated in lines 3 and 23. If we block these beacons or their responses, the player refuses to start playing the video.

If there are two players logged in to same account playing Hulu Plus content, a third player is only able to get a valid license if one of the other two stops playing and sends the corresponding beacon.

## 4.5 Geolocation

Hulu allows users to browse the Hulu.com website from IP addresses outside the United States. The first time they do so, the warning in Figure 4.1 is shown, but then users can log in, browse the content catalog and perform account management functions.

```xml
<?xml version="1.0" encoding="utf-8"?>
<beacons>
  <realtime host="t.hulu.com">
    <beacon type="error" send="always" cdn-specific="true">
      <event name="applicationerror.appname" send="never" />
      <event name="connectionerror.loadtimeout" send="never" />
    </beacon>
    <beacon type="session" send="always" cdn-specific="true">
    </beacon>
    <beacon type="playback" send="never" cdn-specific="true">
      <event name="start" send="always">
      </event>
    </beacon>
    <cdn-hosts>
      <cdn name="akamai" host="t-ak.hulu.com">
      </cdn>
      <cdn name="level3" host="t-l3.hulu.com">
      </cdn>
      <cdn name="limelight" host="t.hulu.com">
      </cdn>
    </cdn-hosts>
  </realtime>
  <standard host="t2.hulu.com">
    <beacon type="dataload" send="onerror" />
    <beacon type="playback" send="always">
      <event name="connectionerror" send="never" />
      <event name="connectionchange" send="never" />
      <event name="netstreamerror" send="never" />
      <event name="datastreamerror" send="never" />
      <event name="applicationerror" send="never" />
      <event name="prerollstart" send="never" />
      <event name="prerollposition" send="never" />
      <event name="prerollend" send="never" />
    </beacon>
    <beacon type="revenue" send="always">
      <event name="request" send="onerror" />
      <event name="response" send="onerror" />
      <event name="httpstreamerror" send="onerror" />
      <event name="request" send="onerror" />
      <event name="request" send="onerror" />
    </beacon>
    <beacon type="abortedsession" send="never" />
  </standard>
</beacons>
```

Listing 4.1: Beacon instructions in XML for the Hulu player.

Despite being able to browse Hulu.com from outside the United States, content streaming is restricted and the license server will not grant licenses in response to requests coming from IP addresses outside the United States. It instead replies with an error that causes the player to display the message shown in Figure 4.9.

It is possible with VPN software to bypass this geographical restriction: if a user has a VPN connection that terminates in the United States, and consequently assigns an IP address from within the United States, the license server grants the license and the video can be played. If, however,

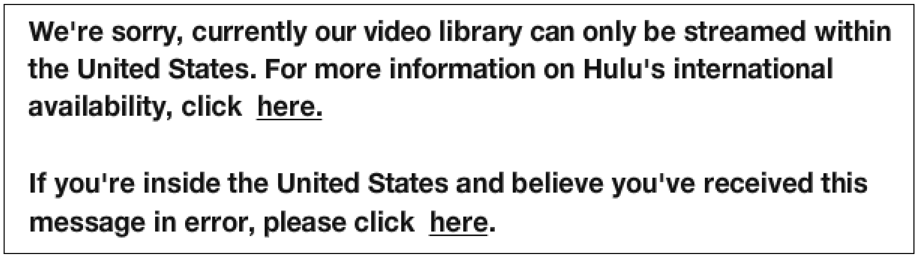


Figure 4.9: Attempt to play Hulu content with an IP address from outside the United States.

the VPN connection is closed after the video has started, the Hulu Player will try to make a new request for the license. This time the request comes from a foreign IP address and the license is denied, presenting the user with the message shown in Figure 4.9.

## 4.6 Output protections

Adobe Flash technology supports analog and digital content output protections on Microsoft Windows operating systems [4][3][5]. The protection technologies include HDCP, CGMS-A, and Rovi (formerly Macrovision) ACP, allowing content owners to specify requirements for analog and digital outputs to external displays in order to prevent unauthorized video recording.

Hulu is not enforcing these output protections and we were able to output video to both analog and digital displays using a machine running Microsoft Windows. On Macintosh computers running OS X 10.6 we were equally successful in displaying the video content on external monitors. Moreover, recording software can be successfully used to capture the screen and record Hulu content on both Windows and OS X.

## 4.7 Security evaluation

In this section we will assume that the adversary has the same capabilities as those described for the attacker model in Section 3.10.1.

In Hulu's cookie management practices, described in Section 4.2, we observed some of the vulnerabilities that were found with Netflix:

1. Authentication cookies – `_hulu_session` and `_hulu_uid` – are not secure and are transmitted in the clear;
2. Authentication cookies are not HttpOnly and are visible to potentially malicious scripts;
3. Authentication cookies are not invalidated on the server during the sign out process.

The option of using Facebook to authenticate to Hulu does not improve the security of any of the aspects we have focused on. For users who choose such a method, in addition to problems inherent to Hulu's cookie management, a stolen Facebook cookie would also allow an adversary to impersonate the user to Hulu's site. It is one additional point of failure. This is possible because HTTP is the default protocol to access Facebook. To use HTTPS users must explicitly enable that option, and therefore most users do not employ secure connections.

As described in Section 4.5, Hulu enforces geographical restrictions in order to make sure that its users are located in the United States when they request a video to watch. The geographical location is done based on IP address transmitted the license server, who refuses to grant the license if the request is coming from a foreign IP address. This verification can be defeated by the use of VPN services, but the mechanisms Hulu uses are more aggressive than those employed by Netflix: they will prevent a user from keep watching the content if the VPN connection is closed and the client tries to reconnect using a foreign IP address.

Hulu is also more restrictive in the account creation process than Netflix. As we have seen in section 4.1, the user is required to provide a credit card that is associated to a billing address located in the United States. Credit cards from abroad and those created by online virtual credit card services are blocked. This mechanism can still be circumvented by those who can find someone willing to collude and share their U.S. based credit card details. Adversaries using stolen credit card information can also circumvent these restrictions, although credit card stealing is a much more serious offense than accessing Hulu from outside the U.S. and Hulu may not be worried about this scenario. Overall, it is our opinion that the credit card billing address verification constitutes a rather effective barrier for those wishing to bypass the geographical restriction imposed by Hulu.

Like Silverlight, Adobe Flash technologies support output protection industry standards and may be used to restrict video output to image recording devices. Despite these capabilities, the experiments from Section 4.6 show that Hulu is not enforcing output protections. The consequences are similar to those presented in Section 3.10.2 for Netflix, although we were not able to verify whether Hulu includes any customer identifying watermarks in the video images.

As was the case with Netflix, our experiments with Hulu also determined that the license server enforces a limit of two simultaneous streams of Hulu Plus content per account. This mechanism prevents the widespread sharing of Hulu Plus account credentials, which would otherwise allow access to premium content by illegitimate users.

# Chapter 5

# Comcast Xfinity TV

Comcast Corporation is the largest cable, Internet and telephone service provider in the U.S. [24]. Comcast's Xfinity TV customers can go to `http://xfinity.comcast.net/` and click `"Sign In"` to log in and access their account details, manage DVR and cable box devices, access email and watch content online. Comcast allows online access to content from Premium Networks only to customers who have subscribed to such packages and can watch them on their regular TV service.

We studied the Comcast online video service by observing the interactions between the web browser and Comcast's servers. To understand how Comcast enforces the security aspects described in Section 2.1, we analyzed the interactions in the account creation process, user authentication process, during content catalog browsing while keeping the user authenticated and during content watching. We also performed several interactions to test simultaneous streaming, geographical location restrictions and output protections. In the following Sections, we describe what we have observed during these experiments. The Chapter ends with a security evaluation of the analyzed mechanisms.

## 5.1 Account creation

Customers need to create an account with the website and link it to their regular Xfinity TV service. In the account creation process, customers have to provide their Xfinity TV Account Number and the Phone Number associated with it, or they need to provide the last four digits of their Social Security Number, Date of Birth and Phone Number. Figure 5.1 shows the verification screen using the Account Number and Phone Number, which also features a CAPTCHA [6]. If the phone number does not match the Account Number the following error message is displayed to indicate that the registration is not allowed to proceed:

> The phone number entered does not match this account. Please try again.

The experiments we made with wrong telephone numbers did not lock our test account. Triggering any account locking mechanism would allow adversaries to perform DoS attacks on customers' web accounts.

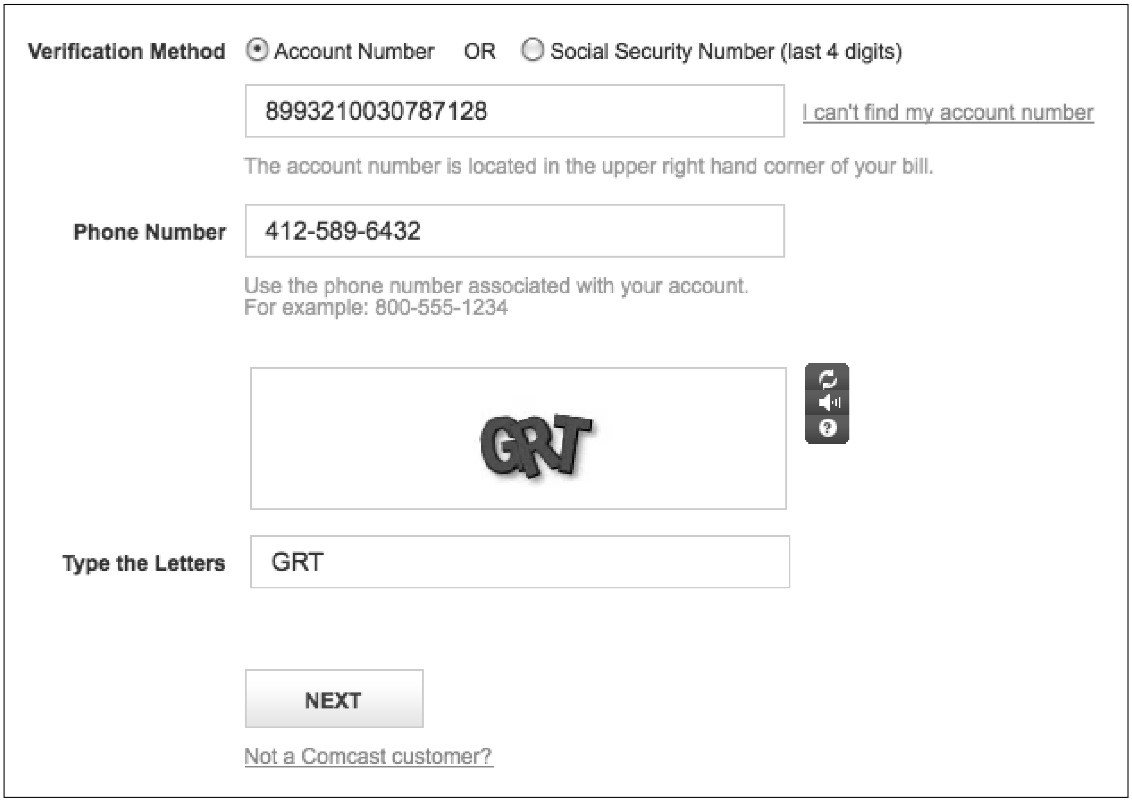


Figure 5.1: Xfinity website's Comcast Account Number Registration.

The CAPTCHA prevents automated submissions by robots, which might attempt to brute force the Account Number and Phone Number to find a valid match.

Users of the website also have the option of creating a *mySIGN-IN* account, which is not linked to an existing regular TV account. The requested information includes:

- First name
- Last name
- Email address
- Password
- Zip code
- Birthday
- Gender

Since users are not required to sign in to watch non-premium content, it is not clear what benefits a *mySIGN-IN* account brings to the user.

## 5.2 Authentication and Session Management

### 5.2.1 Sign in process

To sign in to Comcast's website, customers need to provide a username, which is either their Comcast ID or an email address, and a password. This sign in process is done at `https://login.comcast.net/login` where the POST request with user credentials is sent securely over HTTPS. As is the case with Netflix and Hulu, the response to this request sets in the user's browser the authentication and session cookies presented in Table 5.1 and redirects it to `http://xfinity.comcast.net/`, which is served over HTTP with no encryption.

| Cookie name | Persistent | Purpose | Secure |
|---|---|---|---|
| MYPORTAL | Yes | Authentication | No |
| s_ticket | No | Session | No |
| session@comcast.net | No | Session | No |
| rm_ticket | Yes | Authentication | Yes |
| tg_ticket | No | Session | Yes |
| tls_s_ticket | No | Session | Yes |

Table 5.1: Authentication and session cookies set by the Xfinity TV sign in process when user selects the "`Remember me...`" option.

Three cookies are responsible for keeping the user authenticated on HTTP pages:

- MYPORTAL
- s_ticket
- session@comcast.net

The two last cookies are not persistent and will be lost when the user quits the browser, as only persistent cookies remain present. If the user selected "`Remember me...`" on the Sign In page, `MYPORTAL` will be persistent and an additional secure cookie is set: `rm_ticket`.

When the customer returns to `http://xfinity.comcast.net/`, after closing and restarting the browser, the session cookies are not sent in the request and the browser is redirected to `https://login.comcast.net/login`. The GET request sends the persisted `MYPORTAL` and `rm_ticket` cookies and the user is re-authenticated. Session cookies are set and the browser is redirected back to `http://xfinity.comcast.net/`. Similar to the Netflix and Hulu, most content on the website and even some account management pages are served over HTTP, with `MYPORTAL` and the session cookies being used to keep the user authenticated. This process is illustrated in Figure 5.2.

`tg_ticket` and `tls_s_ticket` in Table 5.1 are transmitted to authenticate the user on the account management pages, which are served over HTTPS.

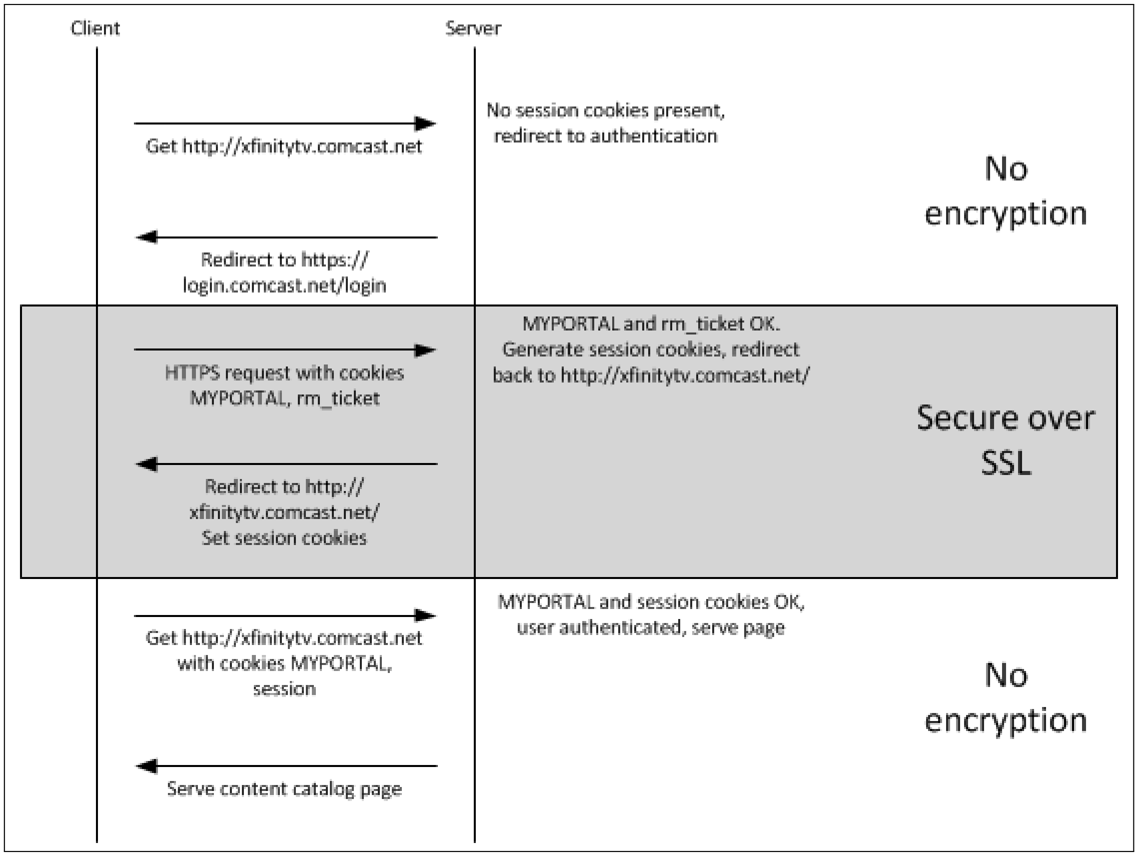


Figure 5.2: Comcast re-authentication process when the customer returns with persistent authentication cookies.

### 5.2.2 Session management

The non-Secure authentication cookies allow the user to watch content online, like movies and TV series, and also allow the user to go to the profile page. Here, the `Account Number` and `Zip Code` are visible. These are personally identifiable information. Whenever the customer browses these HTTP pages while logged in, the non-Secure authentication and session cookies in Table 5.1 are sent with every request in the clear. As we saw in section 3.10.3, the user is then vulnerable to cookie stealing and session hijacking attacks. To visit HTTPS pages, the Secure session cookies are required.

### 5.2.3 Sign out process

The sign out process of Comcast's Xfinity website occurs on an HTTPS page. After clicking `Sign Out` all cookies shown in Table 5.1 are deleted from the browser cookie store.

As was the case with Netflix and Hulu, the deletion only affects the browser side and the cookies are still accepted by the server: if we save the cookies to a file, sign out, and then import the cookies from the file, when we visit `http://xfinity.comcast.net/` we are signed in.

### 5.2.4 Unprotected user account details

After being signed in, Xfinity customers can access their user profile online in plain HTTP pages. At `http://xfinity.comcast.net/profile/` customers can view `Account Number`, `Primary User ID` and various DVR and cable box management functions. We were able to access the cable box management functions, to name devices and select them to perform remote channel changing. However, we were not able to confirm whether we could act on the device and if the channels were actually changed because the account we were using, despite being configured with a cable box, did not have the physical box connected. As we detailed in Section 2.3, this was the only account available to us and we could not properly evaluate these functions of the website.

More sensitive functions like bill payment do require the user to reenter the email address and password.

## 5.3 Simultaneous streaming and geolocation

Comcast restricts the number of simultaneous streams of premium content on Xfinity online and imposes geographical restrictions in the same way Hulu does. The license server will refuse to grant a license if it determines that there are two licenses already outstanding or if the source IP address of the request is located outside the United States. The error messages, however, are not informative and are the same in both cases. The displayed message is shown in Figure 5.3.

Similarly to what we described for Hulu, Comcast's Xfinity TV online player receives the encrypted content streams and sends control messages using RTMP.

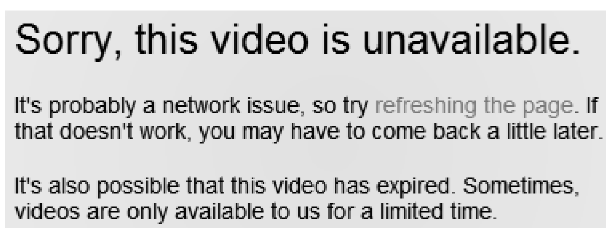


Figure 5.3: Xfinity TV online error when a user located outside the U.S. tries to watch a video or the number of simultaneous streams has been exceeded.

## 5.4 Output protections

Like Hulu, Comcast uses Adobe Flash technology, which supports various industry standards for output protections [4] [3] [5]. Nevertheless, in our experiments we were able to output video images from Xfinity TV online to external analog monitors and digital monitors not supporting output protections schemes. This shows that Comcast is not enforcing output protections.

## 5.5 Security evaluation

We will once again use the attacker model defined for Netflix in Section 3.10.1. The account creation process described in Section 5.1 does not require the user to be a Comcast customer. The user can create an account of type *mySIGN-IN*, without providing the Comcast account number or a Social Security number. However, this type of account does not allow access to content from premium networks. Additionally, the non-premium content is available to be watched on the site without having to sign in.

To watch premium content users must create an account that is linked to the Comcast account through which they pay for Internet access, cable box subscription and premium channels. To link the accounts, the user must provide the Comcast account number and the telephone number associated with it. Just requiring the account number would not properly authenticate the customer because an adversary could steal a Comcast bill delivered by mail and enter a valid account number. Verifying the phone number creates an authentication mechanism by requiring information that the adversary cannot retrieve from a stolen Comcast bill.

If the adversary is not a Comcast customer but is able to convince a customer to share their Xfinity TV website's credentials, the adversary is able to illegitimately access premium content. To avoid the widespread sharing of credentials, like Netflix and Hulu, the Xfinity TV website imposes simultaneous streaming restrictions, as described in Section 5.3. Sharing between two users will still allow both to watch simultaneously, but a third user requesting to watch a video would have the license server refusing to grant a license.

The authentication and session management processes we have described for the Comcast Xfinity website reveal flaws on the management of authentication cookies:

1. Authentication cookies are transmitted in the clear – after the initial sign in process that occurs securely over HTTPS, these cookies are transmitted in the clear with every page request while browsing content in the `http://xfinity.comcast.net/` website. `MYPORTAL`, `s_ticket` and `session@comcast.net` are available to adversaries to sniff off the network.

2. Authentication cookies are not HttpOnly – this would allow them to be sent to an adversary via script executed in the browser, should a Cross Site Scripting vulnerability be found and exploited.

3. Authentication cookies are not invalidated on the server on sign out – an adversary in possession of the authentication cookies continues to be able to fraudulently impersonate the customer on the website, even after an explicit sign out.

4. Personally identifiable information sent in the clear – as shown in Section 5.2.1, the cookie MYPORTAL that is transmitted over HTTP contains:

   - Email address
   - First name
   - Zip code

While vulnerabilities 1., 2. and 3. already allow an adversary to view personally identifiable information by impersonating the user to the website, vulnerability 4. would allow an adversary to learn private information even if for some reason (incomplete packets captured, for example) the session hijacking has been not successful. This information could be maliciously used in social engineering attacks.

As described in Section 5.4, Comcast does not implement output protections. As seen with Hulu in Section 4.7, which uses the same technology, this is not imposed by any Adobe Flash limitation, but rather due to a policy decision. The consequences are similar to those described for Hulu.

# Chapter 6

# Recommendations

Of the vulnerabilities that have been enumerated for Xfinity's, Netflix's and Hulu's websites, we consider the transmission of authentication cookies in the clear to be the most severe. The abundance of open Wi-Fi at universities, airports, restaurants, hotels and other public spaces has allowed computer and Internet use to become commonplace in open and shared wireless networks. Internet surfing, email, social networking and movie watching are common activities in such networks. A malicious party can setup a computer to sniff traffic from other users on the network and gather authentication cookies to hijack their sessions. In accordance with our views, we make some recommendations on mitigation techniques for this problem.

## 6.1 For OTT video service providers

One robust solution to mitigate this vulnerability is to encrypt all traffic to and from the affected websites. Setting the cookies as Secure and switching to HTTPS all the time, effectively protects the authentication information from network sniffers (as long as the assumptions for SSL security hold true [29] [30]) without causing any additional complexity to the users. Additionally, to defend against more complex SSL-stripping Man-in-the-Middle attacks, sites could adopt HTTP Strict Transport Security [16] (HSTS). Upon contacting the legitimate site securely for the first time, the browser would only connect using HTTPS, refusing any HTTP connection attempt. HSTS is not universally adopted by major browsers, but support is built into Google Chrome[1] and Firefox 4+. As was described in Section 3.9, the Netflix mobile application only transmits authentication cookies over SSL, effectively protecting them from network eavesdroppers.

To allow users to explicitly invalidate sessions when they sign out, the cookies should be invalidated on the server side. Even if an adversary gains access to the authentication cookies, a sign out enforced on the server would prevent the attacker from continuing to access the compromised user account. Websites should also provide users with an option to sign out all sessions, which would invalidate all sessions for that user on the website in the event of suspicious activity being detected.

[1] `http://www.google.com/chrome`

For example, the Gmail[2] service from Google communicates over HTTPS only, no cookies are sent in the clear; it implements server side sign out; and has the option of signing out all other sessions. These measures vastly improve security to defend against cookie theft and session hijacking and should be implemented by all websites that have sensitive user accounts and information.

Neither Netflix, Comcast Xfinity nor Hulu use HttpOnly cookies. This poses a problem if an adversary is able to find and exploit Cross-Site Scripting [18] (XSS) vulnerabilities on their sites. XSS errors come in fourth place in the 2011 CWE/SANS Top 25 Most Dangerous Software Errors rank [32] and have been in the top 10 for several years. Using HttpOnly cookies is a good defense against cookie theft by browser scripts. M. Johns presents alternative ways to defend against XSS attacks if HttpOnly cookies are not practical on the main content site [18]. Specifically, to defend against session identifier theft, every user action would trigger two parallel requests to different subdomains containing a request identifier to correlate the two on the server side. The cookie containing the session identifier would be a secure cookie not visible to scripts running on the context of the main content site.

To correct Netflix's vulnerability 1 mentioned in Section 3.10.2, authentication should be done with session cookies that the browser does not remember after it is shut down. Comcast's website does this correctly. On the other hand, Comcast sends the email address, first name and zip code of the customer in the clear as part of a cookie data. Making the cookies contain only opaque tokens or pointers to database entries in the server would eliminate this privacy problem identified for Comcast Xfinity's website.

The security vulnerabilities and the corrective mechanisms and techniques proposed in this report are not new. Gmail, as previously mentioned, is a service that has implemented encryption by default for all of communications. A letter from security experts and privacy advocates addressed to Google's CEO Eric Schmidt in June of 2009 [33], and the response on Google Online Security Blog [34], illustrate the debate that occurred more than two years ago. At the time, Gmail users already had the option of using HTTPS for all connections, but it was not the default behavior. In January of 2010 Google started rolling out HTTPS by default for Gmail [28].

Facebook appears to be going through a similar process. Following the release of Firesheep in October of 2010[3], which automated the process of capturing HTTP cookies and hijacking Facebook accounts (among others), Facebook was under pressure to move to HTTPS in all pages. This option became available in January of 2011[4]. It is still not the default option and many users are still using HTTP. Consequently, they are still vulnerable.

## 6.2 For Network Administrators

While most OTT service providers and other website operators have not fully adopted HTTPS and are still insecurely sending authentication cookies in the clear, Wi-Fi access points can adopt WPA 2 (Wi-Fi Protected Access 2) [19] [1] with CCMP encryption to avoid packet sniffing and capturing

[2] https://mail.google.com/
[3] http://codebutler.com/firesheep
[4] https://www.facebook.com/blog.php?post=486790652130

on the wireless networks. While this adds a layer of complexity to both network administrators and users, it is practical and effective. Some universities already require students and staff to authenticate using their student ID and password to join a secure wireless network; restaurants and airports could adopt simple strategies like setting the password equal do the access point SSID and inform their customers through various mechanisms. Customers would still be able to freely access the network and their security would be greatly improved.

It is worth noting that WEP (Wired Equivalent Privacy) does not provide privacy within the network. Clients with knowledge of the key are able to decrypt all traffic [19]. In recent years, weaknesses in WPA with TKIP have been found that allowed progressively more severe attacks to be developed [31] [25] [14] [9]. Therefore, WPA 2 with CCMP encryption should be preferred whenever both options are available.

## 6.3 For Network Users

Users joining open Wi-Fi networks should always prefer HTTPS versions of websites if they are available. If the website does not offer HTTPS everywhere as an option, users with VPN (Virtual Private Network) access to home or corporate networks or through VPN service providers can leverage that option to protect their traffic from nearby traffic sniffers.

# Chapter 7

# Conclusions

In the previous chapters we have analyzed security mechanisms implemented by three major OTT video service providers: Netflix, Hulu and Comcast. These companies have millions of customers each, who pay for services that allow them to watch video content online and have sensitive information in their user accounts. The purpose of the security mechanisms is to enforce security policies related to business models and content licensing deals these video distributors have with content providers. We did not attempt to provide an exhaustive security analysis and focused only on very specific aspects of security enforcement:

- Perform authentication in order to establish the correct customer identity.
- Perform authorization in order to determine whether to grant licenses that allow access to the protected assets.
- Ensure strong cryptographic protections preventing unauthorized access to the protected assets.
- Prevent unauthorized copying of video assets.
- Enforce geographical restrictions imposed by content license deals made with content owners and providers.

We analyzed each of these aspects using web browser based clients and we also evaluated the mobile Netflix application available for Android smart phones and tablets.

Regarding authentication, we found that the three providers use HTTP cookies to maintain state pertinent to user authentication and browser sessions. Although the initial interaction where the user sends the authentication credentials is protected by SSL, the authentication cookies that are then given by the server and stored by the browser are not marked as secure. Therefore, when the user is redirected to pages served over HTTP they are sent in the clear, with no encryption protecting them from network eavesdroppers. This is not the same as having the authentication credentials – user name and password – stolen, but the cookies are valid for 30 days and allow the attacker to impersonate the user and access the content in the OTT video provider website.

We have shown that the three analyzed OTT providers allow adversaries in open Wi-Fi networks to easily steal authentication cookies from legitimate users. The widespread use of computers to watch movies, access email and other Internet services in places like airports, hotels, restaurants, schools and university campuses provide adversaries ample opportunity to engage in cookie theft and session hijacking, or sidejacking. To assess the difficulty of the attack and seriousness of the risk, we ran the experiment described in Section 3.10.3 in a controlled environment, making sure that the experiment did not affect any other legitimate users. The experiment revealed that the attack is simple to do and should be a serious concern for users and for the companies.

Still on the topic of authentication cookies, we found that all three OTT providers analyzed rely on client side deletion of the cookies from the browser cookie store to sign out the user. While this effectively causes the browser not to remember the authentication cookies the next time the user tries to access the service, if those cookies have been stolen the adversary can continue to impersonate the user and access the site.

There were slight differences in the way each of the providers handled the authentication cookies. Netflix, for example, set the authentication cookie to be persistent even when the user did not select the option to be remembered of the next visit. This results in unexpected behavior that can lead to unauthorized access to the user account. Hulu allowed authentication using a Facebook account, which did not improve authentication security in any regards. Comcast unnecessarily sends personally identifiable information in cookies in the clear.

While the identified vulnerabilities are not new, the same is true for the mitigation techniques. The mechanisms that allow secure authentication and session management are known and have existed for many years. We provided Google Gmail as an example of a service that has adopted these practices. Companies like Netflix, Comcast and Hulu have still not fully adopted these secure mechanisms.

On the authorization mechanisms, we found that all three providers enforce geographical restrictions based on the source IP address of request for the license to play the movie. If the user is outside the United States, or also Canada in the case of Netflix, the license server will not grant the license. The license will not be granted also if there have already been granted two licenses and the server has not received indication that the clients have stopped displaying the video. Without the license, the player cannot get the key which is necessary to decrypt the video stream. Both Microsoft Silverlight with PlayReady DRM and Adobe Flash support strong AES encryption and provide robust solutions from protecting the video assets from unauthorized use. Despite the capabilities provided by these technologies, none of the services analyzed used the available output protection mechanisms to prevent unauthorized recording of the video signal by devices that connect to video outputs. Without even requiring external devices, we were also able to record the video images directly on the machine that was running the client with the aid of image capturing software.

To ensure stronger enforcement of geographical restrictions, Hulu Plus accounts require the user to provide a credit card number whose billing address is in the United States. This means that even with VPN connections that assign U.S. 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. Nevertheless, the VPN connections were effective to bypass the geographical restrictions on source IP of the client requests, and we were able to access Hulu content not exclusive to Hulu Plus accounts, Netflix movies and non-premium content from Comcast. Access to premium content on Comcast requires the account to be associated with a physical cable installation, which also provides a significant barrier to most adversaries.

The analysis we did to the Netflix mobile client for Android revealed that the mobile client interacts with Netflix's servers in a way that is very similar to the web browser based client. The main differences were that authentication is done via the OAuth protocol, the client does not contact with Microsoft's individualization servers and all communications transmitting authentication information are protected with SSL. This last finding represents a significant difference, compared to the browser-based authentication on personal computers, because it means that the mobile application is not vulnerable to the cookie stealing attack described in section 3.10.2. Regarding the enforcement of geographical location restrictions, we noted that the mobile application does not take advantage of the mobile smart phone's location capabilities: geographical location was done only by the servers based on IP address and the GPS system or mobile cellular network location information was not used.

Finally, we presented some recommendations which reflect our views and opinions on the authentication mechanisms used by Netflix, Hulu and Comcast: it is our opinion that they should adopt known techniques which have been implemented by others, like server side session invalidation and SSL connections to protect all traffic.

In the course of this work, the research on the technologies employed by OTT providers, the tools used in the analysis, the guidance of the supervisors, the results we observed, and even the tools used to produce this report, provided us with a rich and instructive experience.

As a last note, we remind the reader that the descriptions, results and conclusions presented in this report are valid for a limited time period, during which we conducted our research. The services we analyzed are constantly being evolved by their operators and the results presented here may soon be outdated. In fact, as we were analyzing the Netflix service, it's device management features changed in significant ways: where customers before had a page to individually manage each device – activate, deactivate, delete, and other operations – now users cannot access details of each device and can only activate devices.

# Bibliography


[1] ISO/IEC International Standard - Information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 6: Medium access control (MAC) security enhancements. *ISO/IEC 8802-11, Second edition: 2005/Amendment 6 2006: IEEE STD 802.11i-2004 (Amendment to IEEE Std 802.11-1999)*, pages c1 –178, 23 2004.

[2] C. Adams. Emulating Netflix: Delivering video through the cloud, January 2011. Available at `http://www.breakingpointsystems.com/community/blog/emulating-netflix-delivering-video-through-the-cloud/`.

[3] Adobe Systems Incorporated. Adobe Flash Access 2.0 whitepaper, 2009. `http://www.adobe.com/mena/products/flashmediaserver/pdfs/flashaccess2_0_whitepaper.pdf`.

[4] Adobe Systems Incorporated. Flash player 10.1 release notes, April 2010. `http://kb2.adobe.com/cps/838/cpsid_83808.html`.

[5] Adobe Systems Incorporated. HTTP dynamic streaming on the Adobe Flash platform, 2010. `http://www.adobe.com/products/httpdynamicstreaming/pdfs/httpdynamicstreaming_wp_ue.pdf`.

[6] 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.

[7] A. Barth. HTTP State Management Mechanism. RFC 6265 (Proposed Standard), Apr. 2011.

[8] A. Barth, C. Jackson, and J. C. Mitchell. Robust defenses for cross-site request forgery. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 75–88, 2008.

[9] M. Beck. Enhanced TKIP Michael Attacks, February 2010.

[10] N. Christin, S. Yanagihara, and K. Kamataki. Dissecting one click frauds. In *Proceedings of the Conference on Computer and Communications Security*, Oct. 2010.

[11] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878, 6176.

[12] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785, 6266.

[13] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: cold-boot attacks on encryption keys. *Communications of the ACM*, 52(5):91–98, May 2009.

[14] F. M. Halvorsen, O. Haugen, M. Eian, and S. F. Mjølsnes. An improved attack on TKIP. In *Proceedings of the 14th Nordic Conference on Secure IT Systems: Identity and Privacy in the Internet Age*, pages 120–132, 2009.

[15] E. Hammer-Lahav. The OAuth 1.0 Protocol. RFC 5849 (Informational), Apr. 2010.

[16] J. Hodges, C. Jackson, and A. Barth. HTTP Strict Transport Security (HSTS). draft-ietf-websec-strict-transport-sec-02, August 2011.

[17] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459 (Proposed Standard), Jan. 1999. Obsoleted by RFC 3280.

[18] M. Johns. Sessionsafe: Implementing XSS immune session handling. In *European Symposium on Research in Computer Security 2006*, pages 444–460, 2006.

[19] A. Lashkari, M. Danesh, and B. Samadi. A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 48 –52, August 2009.

[20] Microsoft Corporation. Digital rights management (DRM). Available at `http://msdn.microsoft.com/en-us/library/cc838192%28v=vs.95%29.aspx`.

[21] Microsoft Corporation. Microsoft PlayReady content access technology, July 2008. Available at `http://download.microsoft.com/download/b/8/3/b8316f44-e7a9-48ff-b03a-44fb92a73904/Microsoft%20PlayReady%20Content%20Access%20Technology-Whitepaper.docx`.

[22] Microsoft Corporation. Using Silverlight DRM, powered by PlayReady, with Windows media DRM content, November 2008. Available at `http://download.microsoft.com/download/7/6/D/76D540F7-A008-427C-8AFC-BE9E0C0D8435/Using_Silverlight_with_Windows_Media_DRM-Whitepaper_FINAL.doc`.

[23] Microsoft Corporation. Microsoft security advisory (967940) – update for Windows autorun, February 2009. Available at `http://technet.microsoft.com/en-us/security/advisory/967940`.

[24] National Cable & Telecommunications Association. Top 25 multichannel video programming distributors as of dec. 2010, December 2010. `http://www.ncta.com/Stats/TopMSOs.aspx`.

[25] T. Ohigashi and M. Morii. A practical message falsification attack on WPA. Available at `http://jwis2009.nsysu.edu.tw/location/paper/A%20Practical%20Message%20Falsification%20Attack%20on%20WPA.pdf`.

[26] Sandvine. Global internet phenomena spotlight – Netflix rising, May 2011. Available at http://www.sandvine.com/downloads/documents/05-17-2011_phenomena/Sandvine%20Global%20Internet%20Phenomena%20Spotlight%20-%20Netflix%20Rising.pdf.

[27] A. Schapira. Analysis of Netflix's security framework for 'Watch Instantly' service. Technical report, Pomello, LLC, April 2009.

[28] S. Schillace. Default HTTPS access for Gmail, January 2010. In The Official Gmail Blog. Available at http://gmailblog.blogspot.com/2010/01/default-https-access-for-gmail.html.

[29] C. Soghoian and S. Stamm. Certified lies: Detecting and defeating government interception attacks against SSL, 2010.

[30] M. Stevens, A. Lenstra, and B. de Weger. Target collisions for MD5 and colliding X.509 certificates for different identities. Cryptology ePrint Archive, Report 2006/360, 2006.

[31] E. Tews and M. Beck. Practical attacks against WEP and WPA. In *Proceedings of the second ACM conference on Wireless network security*, ACM Conference on Wireless Network Security 2009, pages 79–86, 2009.

[32] The MITRE Corporation. 2011 CWE/SANS top 25 most dangerous software errors, June 2011. Available at http://cwe.mitre.org/top25/.

[33] Various. Ensuring adequate security in Google's cloud based services, June 2009. Available at http://www.wired.com/images_blogs/threatlevel/2009/06/google-letter-final2.pdf.

[34] A. Whitten. HTTPS security for web applications, June 2009. In Google Online Security Blog. Available at http://googleonlinesecurity.blogspot.com/2009/06/https-security-for-web-applications.html.

[35] A. Zambelli. Smooth streaming architecture, February 2009. In Alex Zambelli's Microsoft Media Blog. Available at http://alexzambelli.com/blog/2009/02/10/smooth-streaming-architecture/.

# Appendices

# Appendix A

# Netflix listings

## A.1 clientaccesspolicy.xml


```
<?xml version="1.0" encoding="utf-8"?>
<nccp:request xmlns:nccp="http://www.netflix.com/eds/nccp/2.10">
  <nccp:header>
    <nccp:softwareversion>2.876.642.1</nccp:softwareversion>
    <nccp:certificationversion>1</nccp:certificationversion>
    <nccp:preferredlanguages>
      <nccp:appselectedlanguages>
        <nccp:language>
          <nccp:index>1</nccp:index>
          <nccp:bcp47>en</nccp:bcp47>
        </nccp:language>
      </nccp:appselectedlanguages>
    </nccp:preferredlanguages>
    <nccp:payload encrypted="true">AhC5NPMRretsOhxjyz9sYXShgJCyDn6qNX20+
        a4L2uRv0SiWlv8TNGqkmCWBTfDhIT3Zmr+ZdeZc9kaqDWt+KqND4cd6VgN9Ed
        oeExmBa8FMAu15UMqZySDnRPzdmAqkUiX0r4I73Jh6nGuKWH2QcOxf3VQDnh9gw/3t6Tn4WD/
        CAoEehrKN2t5h6inU2UkwnB0ShxNT4Fwb7qnAuuu31e4G1i4=</nccp:payload>
  </nccp:header>
  <nccp:register>
    <nccp:idcookiereg>
      <nccp:payload encrypted="true">AhAB1lkyZLqjLbx3BAgRHyMSg3A++r7EsW5o1cZcuaK+r8W/
          KyIltELV+GLsyAbdXYWK2Tu5FUOiRiWNX5dVe YpcpxDfbedusnZo0y+
          rYNC1G6xR94r1X9CmGUv0NhXfexzFYzjL4iQ7YSM2
          KXLnkIA2N8cWI1msU15eZb0qhkyf6OrM1wvNqdqLCrwpGdLDFma+Sdy7cNQAc
          iafbMLxray2O4A2ApMjpORVm8y6UiKaY155Z16EnmaVUcNCOOHgfHYW+Fcifn2X/Fm1+
          Iyh4IKXrfZKmN1n6hMet9w3FIOwj57WyDZWgztl6ltQG3W auTTunoqJkFTc3XGUXA/1cZHA87KKB/
          vjEsWIGgKNnixvH2DoEnuOU HgASCaGvj+DVp5RD6W9yRYDQEN9St+Wi117jVDDvVJUYCykc5LtII3mX
           vVH5N7TqNPvF7gc2e4 HNuh9GEhyUdjsFmf9OwhVKsv/jzibTCqiuKTD/hfmn9CDujFckA6G50KgRxZ
          /VE6wTvvXlWHk806ocURkQLCaDS8182dYp BN4pmxZJRm49rDhDC2IX/8MH8Xtv0+kVW1pqfwxaY7+3
          GRpTSyNxCa0EIdD+yJFuPiRBCqzdOfqhG4/noVGkTOFLI9eKLABFOHaKn4v4F/z6sDcKm6ID
          PPEbr22Dw+pGpgbCyehpFj8VnY91nxFGLZvWyGR6ara/smzIPwnXUIE
          Lcca3H0S8XxJwBzjYTObwjNkaIXITE8C4is9UOA3i5Miz528SYni8d8YXrF
          W6UDIUgRi7Qe1hKe0WBlZ/3TLAZpbiNatEA1sGTtRy7wSL+ipVrRDYLZ
          PqNgmiEm8jSjYzrZMvgGe5SSL0YfueekE464YmWNkwSaEs3lS57hveeuu9uEd pSsWmtmDN/
```

```
            rh5EHWpsnWomJV3QZgfZw8YSt7Q2bCrNDzLcc52usdU/XkxMCI6kDb0yAsr3rb41J0IrYd+
            dctZgL6h9uXQbmsGy1TR9JaZ27++PGk1e9migRHgGGc4F+t0k6PIkyyqmIHZGSi1CDE641CU80RhIcv+
            b+VadqCQ/V0i0I8H3XqCr3VjFVRhjSfvJEDYtD+mWHtTfLyxVv8eBL7IH+
            AroxE6wErkIFLT4cWEk3lO3dWljmKPjL72S9rsUs3tcNdgaXHT5XQUkD8d0P GzlitkGo7yHbTRZw+
            Gmcw2dIOlf/A7hGt</nccp:payload>
      </nccp:idcookiereg>
   </nccp:register>
</nccp:request>
```

Listing A.1: NCCP message in the player registration request.

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
   <cross-domain-access>
      <policy>
         <allow-from http-request-headers="Content-Type,X-HMAC,X-CTicket,X-ESN,X-ShopperID,X-
               AuthenticationType,X-FORCEIP,X-AllowCompression">
            <domain uri="http://netflix.ca"/>
            <domain uri="http://*.netflix.ca"/>
            <domain uri="https://netflix.ca"/>
            <domain uri="https://*.netflix.ca"/>

            <domain uri="http://netflix.co.uk"/>
            <domain uri="http://*.netflix.co.uk"/>
            <domain uri="https://netflix.co.uk"/>
            <domain uri="https://*.netflix.co.uk"/>

            <domain uri="http://netflix.com"/>
            <domain uri="http://*.netflix.com"/>
            <domain uri="https://netflix.com"/>
            <domain uri="https://*.netflix.com"/>

            <domain uri="http://netflix.com:7001"/>
            <domain uri="http://*.netflix.com:7001"/>
            <domain uri="https://netflix.com:7001"/>
            <domain uri="https://*.netflix.com:7001"/>

            <domain uri="http://netflix.fr"/>
            <domain uri="http://*.netflix.fr"/>
            <domain uri="https://netflix.fr"/>
            <domain uri="https://*.netflix.fr"/>

            <domain uri="http://netflix.jp"/>
            <domain uri="http://*.netflix.jp"/>
            <domain uri="https://netflix.jp"/>
            <domain uri="https://*.netflix.jp"/>

            <domain uri="http://netflix.tw"/>
            <domain uri="http://*.netflix.tw"/>
            <domain uri="https://netflix.tw"/>
            <domain uri="https://*.netflix.tw"/>
         </allow-from>
         <grant-to>
            <resource path="/nccp/controller" include-subpaths="true"/>
         </grant-to>
```

```
    </policy>
  </cross-domain-access>
</access-policy>
```


Listing A.2: `https://agmoviecontrol.netflix.com/clientaccesspolicy.xml`.

## A.2 Authorization response


```
<HMAC>TzMwnT5gtAcMtuIsArT8Lw3JDhBPgLfA1oW4Js1ADGc=</HMAC>
<?xml version="1.0" encoding="utf-8"?><nccp:response xmlns:nccp="http://www.netflix.com/
    eds/nccp/2.11">
    <nccp:responseheader>
        <nccp:payload encrypted="true">AhAtIRjfQne/IrTFTWxkhs0bgJCufVut3O9hC+PB3qO+
            hoONsUc3TylM9PeBkG6YaFB7JvkTsNkPj7RRHoAlJlnUYu/b+QD4WUCE7jn3jdBeFeCbRjD1708yc/
            LJtLIcBRJBDE4Yke1dVnncRUNtAIK/
            KpMsXZhAOhOKBBrHAxTXNWqiQxddXM109n1cgSxOhW8Gil81kK1ImeM75SA8VxzYBJc=</
            nccp:payload>
    </nccp:responseheader>
    <nccp:result method="authorization">
        <nccp:authorization movie_id="70018715">
            <nccp:payload encrypted="true">AhBngWVhilhwM4MTNeyFVs/FgXCt2Tgbb8jB/fIkQ/B5ON/
                MftxqYUC3diqz0NwTT1oVrXIqmn6kgFwXvil47dpAMMPZ/0LK7tc44ft+P//
                B7rLV5YMhifmxgwU/ZPj2VKTuFpG4bHJEeUmWPqUilr3EEFXiCKE7Tra/
                LJDdYrX8Gbyeosu3VMQYk1ZJVPekXIA0xjyqIZU2JiTqm+9ZBL+Bs8S5oXeC7+JKWP3z1Kr/
                nTsos2wWTU3alnfPmHvDQ7+IJd6cf8rS+
                JsYt7bEjc25DL6XmPRiemZFwGsKm9GjzqVKVaVYP12aixD50uAfxG2d85XDCoZKbRGoVJ4x1t
                JqwG0vbn/U6rCHmADuE5y+RV9dS4jwCx8KqolG4vSqGw7R3wFznze9gnmSUGHDCYZ6Gz 67
                bt2AIhy45U6X1LCRx8CTDuzUXRQbek9M1+SKLHZEt+19
                UuAq9Q3E4LUJk3SOAVVIPS6rsHkTXstXksaruXrM+dQSni4t/zZFqokV2JHMiA==</
                nccp:payload>
            <nccp:cdns>
                <nccp:cdn>
                    <nccp:name>level3</nccp:name>
                    <nccp:cdnid>6</nccp:cdnid>
                    <nccp:rank>1</nccp:rank>
                    <nccp:weight>140</nccp:weight>
                </nccp:cdn>
                <nccp:cdn>
                    <nccp:name>limelight</nccp:name>
                    <nccp:cdnid>4</nccp:cdnid>
                    <nccp:rank>2</nccp:rank>
                    <nccp:weight>120</nccp:weight>
                </nccp:cdn>
                <nccp:cdn>
                    <nccp:name>akamai</nccp:name>
                    <nccp:cdnid>9</nccp:cdnid>
                    <nccp:rank>3</nccp:rank>
                    <nccp:weight>100</nccp:weight>
                </nccp:cdn>
            </nccp:cdns>
            <nccp:trickplay>
                <nccp:resolution>
```

```
                <nccp:width>320</nccp:width>
                <nccp:height>180</nccp:height>
            </nccp:resolution>
            <nccp:pixelaspect>
                <nccp:width>1</nccp:width>
                <nccp:height>1</nccp:height>
            </nccp:pixelaspect>
            <nccp:trickplayinterval>10</nccp:trickplayinterval>
            <nccp:size>0</nccp:size>
            <nccp:trickplayid>231872966</nccp:trickplayid>
            <nccp:downloadurls>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>4</nccp:cdnid>
                    <nccp:url>http://netflix-715.vo.llnwd.net/s/s11/966/231872966.bif?
                        p=55&amp;e=1308977311&amp;h=150bdb71cbcec9ee74de8c8183446e31</
                        nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>9</nccp:cdnid>
                    <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa19
                        /966/231872966.bif?token=1308977311
                        _76e1ef3d4f6e8cc5529fb2e73e5ceda9</nccp:url>
                </nccp:downloadurl>
            </nccp:downloadurls>
        </nccp:trickplay>
        <nccp:trickplay>
            <nccp:resolution>
                <nccp:width>240</nccp:width>
                <nccp:height>121</nccp:height>
            </nccp:resolution>
            <nccp:pixelaspect>
                <nccp:width>8</nccp:width>
                <nccp:height>9</nccp:height>
            </nccp:pixelaspect>
            <nccp:trickplayinterval>10</nccp:trickplayinterval>
            <nccp:size>0</nccp:size>
            <nccp:trickplayid>231873150</nccp:trickplayid>
            <nccp:downloadurls>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>4</nccp:cdnid>
                    <nccp:url>http://netflix-715.vo.llnwd.net/s/s11/150/231873150.bif?
                        p=55&amp;e=1308977311&amp;h=a18a7303888977e3a795fcc32744f52c</
                        nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>9</nccp:cdnid>
                    <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa19
                        /150/231873150.bif?token=1308977311
                        _4540dbb32bdc33f363cd59c9bdf57d2a</nccp:url>
                </nccp:downloadurl>
```

```
      </nccp:downloadurls>
    </nccp:trickplay>
    <nccp:videotracks>
      <nccp:videotrack>
        <nccp:videodownloadables>
          <nccp:videodownloadable>
            <nccp:downloadableid>462565273</nccp:downloadableid>
            <nccp:size>1518592344</nccp:size>
            <nccp:bitrate>1750</nccp:bitrate>
            <nccp:videoprofile>playready-h264mpl30-dash</nccp:videoprofile>
            <nccp:resolution>
              <nccp:width>720</nccp:width>
              <nccp:height>480</nccp:height>
            </nccp:resolution>
            <nccp:pixelaspect>
              <nccp:width>32</nccp:width>
              <nccp:height>27</nccp:height>
            </nccp:pixelaspect>
            <nccp:downloadurls>
              <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>6</nccp:cdnid>
                <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/273/462565273.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=0dc2a913fcf93bdd69613</nccp:url>
              </nccp:downloadurl>
              <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>9</nccp:cdnid>
                <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/273/462565273.ismv?token=1308977311_e09ad7b8a4680028f44814c21fa48b29</nccp:url>
              </nccp:downloadurl>
              <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>4</nccp:cdnid>
                <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/273/462565273.ismv?p=58&amp;e=1308977311&amp;h=a64bf339804ced356b0e79991b96b34b</nccp:url>
              </nccp:downloadurl>
            </nccp:downloadurls>
          </nccp:videodownloadable>
          <nccp:videodownloadable>
            <nccp:downloadableid>462580300</nccp:downloadableid>
            <nccp:size>964425140</nccp:size>
            <nccp:bitrate>1050</nccp:bitrate>
            <nccp:videoprofile>playready-h264mpl30-dash</nccp:videoprofile>
            <nccp:resolution>
              <nccp:width>640</nccp:width>
              <nccp:height>480</nccp:height>
            </nccp:resolution>
```

```
                    <nccp:pixelaspect>
                        <nccp:width>4</nccp:width>
                        <nccp:height>3</nccp:height>
                    </nccp:pixelaspect>
                    <nccp:downloadurls>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>4</nccp:cdnid>
                            <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/300/462580300.ismv?p=58&amp;e=1308977311&amp;h=a3f1cfbdb95b62d08ffd02e2598f163b</nccp:url>
                        </nccp:downloadurl>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>6</nccp:cdnid>
                            <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/300/462580300.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=066deeff60e2d374e5faa</nccp:url>
                        </nccp:downloadurl>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>9</nccp:cdnid>
                            <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/300/462580300.ismv?token=1308977311_fa001018df7e8e954a48002c211b4b0b</nccp:url>
                        </nccp:downloadurl>
                    </nccp:downloadurls>
                </nccp:videodownloadable>
                <nccp:videodownloadable>
                    <nccp:downloadableid>462580866</nccp:downloadableid>
                    <nccp:size>480060924</nccp:size>
                    <nccp:bitrate>500</nccp:bitrate>
                    <nccp:videoprofile>playready-h264bpl30-dash</nccp:videoprofile>
                    <nccp:resolution>
                        <nccp:width>480</nccp:width>
                        <nccp:height>360</nccp:height>
                    </nccp:resolution>
                    <nccp:pixelaspect>
                        <nccp:width>4</nccp:width>
                        <nccp:height>3</nccp:height>
                    </nccp:pixelaspect>
                    <nccp:downloadurls>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>9</nccp:cdnid>
                            <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/866/462580866.ismv?token=1308977311_773219b8da14c38d55fbf21ac9676c00</nccp:url>
                        </nccp:downloadurl>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>4</nccp:cdnid>
```

```
                        <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/866/462580866.ismv?p=58&amp;e=1308977311&amp;h=5b77c295c578ea9592e11863b39880e6</nccp:url>
                    </nccp:downloadurl>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>6</nccp:cdnid>
                        <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/866/462580866.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=0ac44facc7777a3643ad0</nccp:url>
                    </nccp:downloadurl>
                </nccp:downloadurls>
            </nccp:videodownloadable>
            <nccp:videodownloadable>
                <nccp:downloadableid>462581717</nccp:downloadableid>
                <nccp:size>495683168</nccp:size>
                <nccp:bitrate>560</nccp:bitrate>
                <nccp:videoprofile>playready-h264mpl30-dash</nccp:videoprofile>
                <nccp:resolution>
                    <nccp:width>512</nccp:width>
                    <nccp:height>384</nccp:height>
                </nccp:resolution>
                <nccp:pixelaspect>
                    <nccp:width>4</nccp:width>
                    <nccp:height>3</nccp:height>
                </nccp:pixelaspect>
                <nccp:downloadurls>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>4</nccp:cdnid>
                        <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/717/462581717.ismv?p=58&amp;e=1308977311&amp;h=ad308be229805f1e656e825cf6b9a90c</nccp:url>
                    </nccp:downloadurl>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>9</nccp:cdnid>
                        <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/717/462581717.ismv?token=1308977311_70f29aaf9337949c5443c4228d1aff76</nccp:url>
                    </nccp:downloadurl>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>6</nccp:cdnid>
                        <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/717/462581717.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=01ca6bdc1861c95ad6fda</nccp:url>
                    </nccp:downloadurl>
                </nccp:downloadurls>
            </nccp:videodownloadable>
            <nccp:videodownloadable>
```

```
                    <nccp:downloadableid>943989291</nccp:downloadableid>
                    <nccp:size>656601423</nccp:size>
                    <nccp:bitrate>750</nccp:bitrate>
                    <nccp:videoprofile>playready-h264mpl30-dash</nccp:videoprofile
                        >
                    <nccp:resolution>
                        <nccp:width>512</nccp:width>
                        <nccp:height>384</nccp:height>
                    </nccp:resolution>
                    <nccp:pixelaspect>
                        <nccp:width>4</nccp:width>
                        <nccp:height>3</nccp:height>
                    </nccp:pixelaspect>
                    <nccp:downloadurls>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>4</nccp:cdnid>
                            <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3
                                /291/943989291.ismv?p=58&amp;e=1308977311&amp;h
                                =099f393c2fd1e4d26bf25b86d5b2aa20</nccp:url>
                        </nccp:downloadurl>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>9</nccp:cdnid>
                            <nccp:url>http://netflix715.as.nflximg.com.edgesuite.
                                net/sa3/291/943989291.ismv?token=1308977311
                                _8fc6c92d3f31d51d5b106a0912bde4b7</nccp:url>
                        </nccp:downloadurl>
                        <nccp:downloadurl>
                            <nccp:expiration>1308977311</nccp:expiration>
                            <nccp:cdnid>6</nccp:cdnid>
                            <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com
                                /291/943989291.ismv?etime=20110625044831&amp;
                                movieHash=715&amp;encoded=0388d0c855bb6974723bd</
                                nccp:url>
                        </nccp:downloadurl>
                    </nccp:downloadurls>
                </nccp:videodownloadable>
                <nccp:videodownloadable>
                    <nccp:downloadableid>942097318</nccp:downloadableid>
                    <nccp:size>1143921953</nccp:size>
                    <nccp:bitrate>1400</nccp:bitrate>
                    <nccp:videoprofile>playready-h264mpl30-dash</nccp:videoprofile
                        >
                    <nccp:resolution>
                        <nccp:width>640</nccp:width>
                        <nccp:height>480</nccp:height>
                    </nccp:resolution>
                    <nccp:pixelaspect>
                        <nccp:width>4</nccp:width>
                        <nccp:height>3</nccp:height>
                    </nccp:pixelaspect>
                    <nccp:downloadurls>
                        <nccp:downloadurl>
```

```
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>6</nccp:cdnid>
                <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/318/942097318.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=03a1b40b772dbbd73d30b</nccp:url>
            </nccp:downloadurl>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>4</nccp:cdnid>
                <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/318/942097318.ismv?p=58&amp;e=1308977311&amp;h=465bab3da96c7c7c9a1933a00454cb0a</nccp:url>
            </nccp:downloadurl>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>9</nccp:cdnid>
                <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa3/318/942097318.ismv?token=1308977311_390b79842434529c189e42765399b59d</nccp:url>
            </nccp:downloadurl>
        </nccp:downloadurls>
    </nccp:videodownloadable>
    <nccp:videodownloadable>
        <nccp:downloadableid>462542349</nccp:downloadableid>
        <nccp:size>172212588</nccp:size>
        <nccp:bitrate>175</nccp:bitrate>
        <nccp:videoprofile>playready-h264bpl30-dash</nccp:videoprofile>
        <nccp:resolution>
            <nccp:width>480</nccp:width>
            <nccp:height>360</nccp:height>
        </nccp:resolution>
        <nccp:pixelaspect>
            <nccp:width>4</nccp:width>
            <nccp:height>3</nccp:height>
        </nccp:pixelaspect>
        <nccp:downloadurls>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>6</nccp:cdnid>
                <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/349/462542349.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=0bed841ecca06249ca869</nccp:url>
            </nccp:downloadurl>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>9</nccp:cdnid>
                <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/349/462542349.ismv?token=1308977311_02c8dca02a1281adbc26bb639e23f0b8</nccp:url>
            </nccp:downloadurl>
            <nccp:downloadurl>
```

```
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>4</nccp:cdnid>
                        <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/349/462542349.ismv?p=58&amp;e=1308977311&amp;h=3f78502d3114ca2271d406ed8ca4898f</nccp:url>
                    </nccp:downloadurl>
                </nccp:downloadurls>
            </nccp:videodownloadable>
            <nccp:videodownloadable>
                <nccp:downloadableid>462601068</nccp:downloadableid>
                <nccp:size>1278945063</nccp:size>
                <nccp:bitrate>1350</nccp:bitrate>
                <nccp:videoprofile>playready-h264bpl30-dash</nccp:videoprofile>
                <nccp:resolution>
                    <nccp:width>640</nccp:width>
                    <nccp:height>480</nccp:height>
                </nccp:resolution>
                <nccp:pixelaspect>
                    <nccp:width>4</nccp:width>
                    <nccp:height>3</nccp:height>
                </nccp:pixelaspect>
                <nccp:downloadurls>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>6</nccp:cdnid>
                        <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/068/462601068.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=09be623ea93b613536507</nccp:url>
                    </nccp:downloadurl>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>9</nccp:cdnid>
                        <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/068/462601068.ismv?token=1308977311_8f2da9bf886b778fc8e53f5c73d7627d</nccp:url>
                    </nccp:downloadurl>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>4</nccp:cdnid>
                        <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/068/462601068.ismv?p=58&amp;e=1308977311&amp;h=6dd462b79e1dba9eb7064717233ffc7b</nccp:url>
                    </nccp:downloadurl>
                </nccp:downloadurls>
            </nccp:videodownloadable>
            <nccp:videodownloadable>
                <nccp:downloadableid>941802057</nccp:downloadableid>
                <nccp:size>360490136</nccp:size>
                <nccp:bitrate>375</nccp:bitrate>
                <nccp:videoprofile>playready-h264mpl30-dash</nccp:videoprofile>
                <nccp:resolution>
```

```
                <nccp:width>384</nccp:width>
                <nccp:height>288</nccp:height>
            </nccp:resolution>
            <nccp:pixelaspect>
                <nccp:width>4</nccp:width>
                <nccp:height>3</nccp:height>
            </nccp:pixelaspect>
            <nccp:downloadurls>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>9</nccp:cdnid>
                    <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa3/057/941802057.ismv?token=1308977311_4df88f910656181c54a2896dbcc43eb3</nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>4</nccp:cdnid>
                    <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/057/941802057.ismv?p=58&amp;e=1308977311&amp;h=e215e4db1cf71d7aa88ba8c5f940ac9b</nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>6</nccp:cdnid>
                    <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/057/941802057.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=07fe548dd567745a3fb8d</nccp:url>
                </nccp:downloadurl>
            </nccp:downloadurls>
        </nccp:videodownloadable>
        <nccp:videodownloadable>
            <nccp:downloadableid>941879685</nccp:downloadableid>
            <nccp:size>230214833</nccp:size>
            <nccp:bitrate>235</nccp:bitrate>
            <nccp:videoprofile>playready-h264mpl30-dash</nccp:videoprofile>
            <nccp:resolution>
                <nccp:width>320</nccp:width>
                <nccp:height>240</nccp:height>
            </nccp:resolution>
            <nccp:pixelaspect>
                <nccp:width>4</nccp:width>
                <nccp:height>3</nccp:height>
            </nccp:pixelaspect>
            <nccp:downloadurls>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>6</nccp:cdnid>
                    <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/685/941879685.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=05a03c23763faca66e1c4</nccp:url>
```

```
            </nccp:downloadurl>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>9</nccp:cdnid>
                <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa3/685/941879685.ismv?token=1308977311_14e3ec7121c8af5fb4825c6e8af8cbcf</nccp:url>
            </nccp:downloadurl>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>4</nccp:cdnid>
                <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/685/941879685.ismv?p=58&amp;e=1308977311&amp;h=4935ce8404ba49d3ce7edd92f7bebe95</nccp:url>
            </nccp:downloadurl>
        </nccp:downloadurls>
    </nccp:videodownloadable>
    <nccp:videodownloadable>
        <nccp:downloadableid>462584790</nccp:downloadableid>
        <nccp:size>243275243</nccp:size>
        <nccp:bitrate>250</nccp:bitrate>
        <nccp:videoprofile>playready-h264bpl30-dash</nccp:videoprofile>
        <nccp:resolution>
            <nccp:width>480</nccp:width>
            <nccp:height>360</nccp:height>
        </nccp:resolution>
        <nccp:pixelaspect>
            <nccp:width>4</nccp:width>
            <nccp:height>3</nccp:height>
        </nccp:pixelaspect>
        <nccp:downloadurls>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>6</nccp:cdnid>
                <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/790/462584790.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=02e52bce54b16aa0c59e3</nccp:url>
            </nccp:downloadurl>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>9</nccp:cdnid>
                <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/790/462584790.ismv?token=1308977311_de338b79df31c554a4005f58d5832828</nccp:url>
            </nccp:downloadurl>
            <nccp:downloadurl>
                <nccp:expiration>1308977311</nccp:expiration>
                <nccp:cdnid>4</nccp:cdnid>
                <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/790/462584790.ismv?p=58&amp;e=1308977311&amp;h=17eb53578709345207c19ee63f39b77d</nccp:url>
            </nccp:downloadurl>
```

```
              </nccp:downloadurls>
            </nccp:videodownloadable>
            <nccp:videodownloadable>
              <nccp:downloadableid>462554059</nccp:downloadableid>
              <nccp:size>101453701</nccp:size>
              <nccp:bitrate>100</nccp:bitrate>
              <nccp:videoprofile>playready-h264bpl30-dash</nccp:videoprofile>
              <nccp:resolution>
                <nccp:width>480</nccp:width>
                <nccp:height>360</nccp:height>
              </nccp:resolution>
              <nccp:pixelaspect>
                <nccp:width>4</nccp:width>
                <nccp:height>3</nccp:height>
              </nccp:pixelaspect>
              <nccp:downloadurls>
                <nccp:downloadurl>
                  <nccp:expiration>1308977311</nccp:expiration>
                  <nccp:cdnid>4</nccp:cdnid>
                  <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3/059/462554059.ismv?p=58&amp;e=1308977311&amp;h=23ee864925e9cadc6b8b498f404e0b5d</nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                  <nccp:expiration>1308977311</nccp:expiration>
                  <nccp:cdnid>6</nccp:cdnid>
                  <nccp:url>http://nflx.i.80ed7672.x.lcdn.nflximg.com/059/462554059.ismv?etime=20110625044831&amp;movieHash=715&amp;encoded=0b813b6fe1544b4040d0f</nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                  <nccp:expiration>1308977311</nccp:expiration>
                  <nccp:cdnid>9</nccp:cdnid>
                  <nccp:url>http://netflix715.as.nflximg.com.edgesuite.net/sa49/059/462554059.ismv?token=1308977311_55d13943d64419dc1edbaa850f12a78b</nccp:url>
                </nccp:downloadurl>
              </nccp:downloadurls>
            </nccp:videodownloadable>
          </nccp:videodownloadables>
        </nccp:videotrack>
      </nccp:videotracks>
      <nccp:audiogroups>
        <nccp:audiogroup>
          <nccp:audiotype>
            <nccp:audiotypeid>primary</nccp:audiotypeid>
            <nccp:displayname>
              <nccp:bcp47>en</nccp:bcp47>
              <nccp:text>Primary</nccp:text>
            </nccp:displayname>
          </nccp:audiotype>
          <nccp:audiotracks>
```

```
<nccp:audiotrack>
    <nccp:language>
        <nccp:bcp47>en-US</nccp:bcp47>
        <nccp:iso639-2>eng</nccp:iso639-2>
        <nccp:iso639-1>en</nccp:iso639-1>
        <nccp:displayname>
            <nccp:bcp47>en-US</nccp:bcp47>
            <nccp:text>English</nccp:text>
        </nccp:displayname>
    </nccp:language>
    <nccp:isnative/>
    <nccp:audiodownloadables>
        <nccp:audiodownloadable>
            <nccp:downloadableid>462578871</nccp:downloadableid>
            <nccp:size>95187320</nccp:size>
            <nccp:bitrate>96</nccp:bitrate>
            <nccp:audioprofile>playready-heaac-2-dash</
                nccp:audioprofile>
            <nccp:downloadurls>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>9</nccp:cdnid>
                    <nccp:url>http://netflix715.as.nflximg.com.
                        edgesuite.net/sa49/871/462578871.isma?
                        token=1308977311
                        _fef622de9e2446c8773b4a6b73bba6e9</
                        nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>4</nccp:cdnid>
                    <nccp:url>http://netflix-715.vo.llnwd.net/s/
                        stor3/871/462578871.isma?p=58&amp;e
                        =1308977311&amp;h=9950
                        cecc1b91b72a6afc02d9dd9f5d7c</nccp:url>
                </nccp:downloadurl>
                <nccp:downloadurl>
                    <nccp:expiration>1308977311</nccp:expiration>
                    <nccp:cdnid>6</nccp:cdnid>
                    <nccp:url>http://nflx.i.80ed7672.x.lcdn.
                        nflximg.com/871/462578871.isma?etime
                        =20110625044831&amp;movieHash=715&amp;
                        encoded=09ddc908ae94318dfbeb4</nccp:url>
                </nccp:downloadurl>
            </nccp:downloadurls>
        </nccp:audiodownloadable>
        <nccp:audiodownloadable>
            <nccp:downloadableid>462562230</nccp:downloadableid>
            <nccp:size>64551564</nccp:size>
            <nccp:bitrate>64</nccp:bitrate>
            <nccp:audioprofile>playready-heaac-2-dash</
                nccp:audioprofile>
            <nccp:downloadurls>
                <nccp:downloadurl>
```

```
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>6</nccp:cdnid>
                        <nccp:url>http://nflx.i.80ed7672.x.lcdn.
                            nflximg.com/230/462562230.isma?etime
                            =20110625044831&amp;movieHash=715&amp;
                            encoded=07079f8bcc74d6e4bc793</nccp:url>
                    </nccp:downloadurl>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>9</nccp:cdnid>
                        <nccp:url>http://netflix715.as.nflximg.com.
                            edgesuite.net/sa49/230/462562230.isma?
                            token=1308977311
                            _66a5ef1a973a26b65c44b482300997ce</
                            nccp:url>
                    </nccp:downloadurl>
                    <nccp:downloadurl>
                        <nccp:expiration>1308977311</nccp:expiration>
                        <nccp:cdnid>4</nccp:cdnid>
                        <nccp:url>http://netflix-715.vo.llnwd.net/s/
                            stor3/230/462562230.isma?p=58&amp;e
                            =1308977311&amp;h=82
                            f2d426e9c5dfcc728e885ea3c88caf</nccp:url>
                    </nccp:downloadurl>
                </nccp:downloadurls>
            </nccp:audiodownloadable>
        </nccp:audiodownloadables>
      </nccp:audiotrack>
    </nccp:audiotracks>
  </nccp:audiogroup>
</nccp:audiogroups>
<nccp:timedtexttracks>
  <nccp:timedtexttrack>
    <nccp:timedtexttype>subtitles</nccp:timedtexttype>
    <nccp:language>
      <nccp:bcp47>en-US</nccp:bcp47>
      <nccp:iso639-2>eng</nccp:iso639-2>
      <nccp:iso639-1>en</nccp:iso639-1>
      <nccp:displayname>
        <nccp:bcp47>en-US</nccp:bcp47>
        <nccp:text>English</nccp:text>
      </nccp:displayname>
    </nccp:language>
    <nccp:timedtextdownloadables>
      <nccp:timedtextdownloadable>
        <nccp:downloadableid>130093801</nccp:downloadableid>
        <nccp:size>200398</nccp:size>
        <nccp:timedtextprofile>simplesdh</nccp:timedtextprofile>
        <nccp:downloadurls>
          <nccp:downloadurl>
            <nccp:expiration>1308977311</nccp:expiration>
            <nccp:cdnid>6</nccp:cdnid>
            <nccp:url>http://netflix-715.vo.llnwd.net/s/stor3
                /801/130093801.dfxp?p=58&amp;e=1308977311&amp;h=
```

```
                                        bf532a18621e7e91bd3d77ba4e2e2b16</nccp:url>
                                </nccp:downloadurl>
                                <nccp:downloadurl>
                                    <nccp:expiration>1308977311</nccp:expiration>
                                    <nccp:cdnid>9</nccp:cdnid>
                                    <nccp:url>http://netflix715.as.nflximg.com.edgesuite.
                                        net/sa49/801/130093801.dfxp?token=1308977311
                                        _6a9aa26862a93b1be3678257f57422c6</nccp:url>
                                </nccp:downloadurl>
                            </nccp:downloadurls>
                        </nccp:timedtextdownloadable>
                    </nccp:timedtextdownloadables>
                </nccp:timedtexttrack>
            </nccp:timedtexttracks>
            <nccp:streamingparams/>
            <nccp:bookmark timestamp="1308948088">6</nccp:bookmark>
        </nccp:authorization>
        <nccp:status>
            <nccp:success>true</nccp:success>
        </nccp:status>
    </nccp:result>
    <nccp:parameters>
        <nccp:retrycontrol>10</nccp:retrycontrol>
        <nccp:mintimeout>60</nccp:mintimeout>
        <nccp:lowfrequencypollinterval>28800</nccp:lowfrequencypollinterval>
        <nccp:mediumfrequencypollinterval>60</nccp:mediumfrequencypollinterval>
        <nccp:highfrequencypollinterval>15</nccp:highfrequencypollinterval>
        <nccp:hightomediumpollswitchinterval>300</nccp:hightomediumpollswitchinterval>
        <nccp:mediumtolowpollswitchinterval>600</nccp:mediumtolowpollswitchinterval>
        <nccp:loginterval>60</nccp:loginterval>
        <nccp:maxlogsize>2000000</nccp:maxlogsize>
        <nccp:loglevel>error</nccp:loglevel>
        <nccp:supportphone>866-579-7113</nccp:supportphone>
        <nccp:playbackparameters>
            <nccp:sendheartbeats>true</nccp:sendheartbeats>
            <nccp:heartbeatinterval>300</nccp:heartbeatinterval>
        </nccp:playbackparameters>
        <nccp:certstatusparams>
            <nccp:enforce>true</nccp:enforce>
            <nccp:permissivewindow>2592000</nccp:permissivewindow>
            <nccp:vccflushcache>2592000</nccp:vccflushcache>
        </nccp:certstatusparams>
    </nccp:parameters>
</nccp:response>
```


Listing A.3: Netflix successful authorization response.

# Appendix B

# Hulu video license responses

## B.1 License granted response

```
HTTP/1.1 200 OK
Server: nginx/0.7.65
Content-Type: text/xml
X-Fuzz-Disable: True
ntCoent-Length: 23520
Expires: Fri, 29 Jul 2011 20:33:46 GMT
Cache-Control: no-cache
Vary: User-Agent, Accept-Encoding
Content-Length: 23520
Date: Fri, 29 Jul 2011 20:33:47 GMT
Connection: keep-alive
```


c84d7f721579e461f98688ea4ff8bb37a8302111b112226a4a32e06a1b4acb74a37e9
81f795a6b012b6690c1ea4256a82f24d76f214fdaa423383d5e1d6b5c94441e833e88
0b7aabba018d9a5e5531f59b9b42017a3802e487045d577bc38416ca23821f41db7
4f632c7b9645d7ef5a7d06b6a1877bfb3ce8be1b4b1ea6a37cf2d51226642248e70b9
a85f3673266fc174c03478d3631d5815e08034cfa243c398625d8f3e12d888f8745b2
80dd11c3c526d11866123a34067452a9bdfe995e9fe3b84bcc668914c8daadc80b14
026ce57661328f1f2deef8f0f3853a784f1e8071c9b1c41c48bac8086e6b1f5aacd7405
5bce9231d49a1fe60f8693bd0e200996427e7f0f7719bbf132c1117c716cb1a31d603
7d6f1cacdf306704581d474d85a7abf821a22bf4420f48a13097a378f5eff30633fc5bc3
ba8939eb95181fd65192652c2ffdcbe46f5d014f11961e04a55b8abcb31e1c39c53d95
c12300a3f7fc81d73a3e74e423ac0bfb1e7b72dd59346b9939a71c7bd0d09ab57046
e1cb0cb02f720bfa4084fd646324a342095aaa1c94c34938c74ad90ee4747fbcceb52
7012398491a3d0a29eae392701551e01be54f6431c43641cf4afe9b3994d96d0738b
373db9f55ebece641221333ae7e059ed11895354ae963cd136b3a6763b5a74a8d74
21a944aa86cae2330b03525a038bec334e3820d760742de5fcc939540db1bc233f43

a0152cb0ee0f39727361cc0eafbf02695b8904a8bfc1bd78f3a45037d4bd13fe57a6ea
313d66cbe636a130a9221b78cd314f0717d993336d9fbcd84679314f5e2df0160816b
33c8bde03a3959ab0d812dc45d4bc7c084e0b2904e85bf4287c1f40517f3ef2a2d1a8
2447142c3557481ba9d2a7bbb6733e83ea9535861a8c9b20775093f01fb1f2169180
a332f1142dd6f34bc59ab8f93596f60699b7dff5806d1ff20afe52c714f9e6007b5b8da9
2ccc976869f4ebc32e68b91fc858f9d78b03fb1c22875091cc0aa59473eaccdea061c8
4fdfd8ef47d3385ad617aa5726796c03761ebaab205141bd865292b2a5e4cf25df9cb
b0e8c3f191b4eaca9075c052c4c26cae0c5e1ff9bdf1deb32c5f444e6ba802beafcfc6d
9ef7f0234390c07d88b744d25e67fa1fc0020b89d5708f6ef13f76a42ccf4172abe90e4
6548afdd987214e9be6010f41bb9eddde2f72556fa57d1f35a4a05a9abef29cb3c5397
bd9ab855c81ab944570548356e999cef47e6a13d19752604a30ec0de00c3e33538fb
95fbf7501697ac5312c2823adfc74d61e1cb94bd7de3e51fecdd0e0474562740b7d2a
29eab23c6764c34c5e40d6aa79cf20c0605f4f22cd5f7447cf2ef1db04647091faca175
1d4a74d357e91e2db3cf27f69ef03ed514178ab40ce4b16cb6c1b8a525008e67774d
5177807708ba6a418294ab263f021de36f89526ea85c3edde160b546bf7593e32c94
5094bdab7f0727f18b9ac672f4fe2e58423d3cf3afc499773eee3999ae04d3ec18f6025
c6751dbc4748dbf2212a4601bdb831557f899bac01dfefb10e534491ffcee4b06878f37
74e6af08da8afd52a15fcfb64895687aacd587267c8f71edaf2b34fd018e60b9ba1818
3aac397df58c21599b9b28cf1c11de3403cb3109522ea6ce9546a65cfb96640356895
55146402bb9a6b302b2dcee564ed72e4ddeabe47e26e38ce28c47e593fa52af23ae1
d137e2768950b6aa36739af1b7303b0b289443129cfac1df02abf7e8a605ceb92e642
64239cb52bd595c096ee6f1d95a46e4e3a3b01f05fb351a5d1b34f4a4bd4bb4802bd9
6de0803d12d24e20b425b6c888b8fd8cac7557f39cad013bbb7be2ace8aea2136578
c9415ffefa7984ca3625859d0c17cc3b86e8829f36971ca41f20ecb1f247c4d140232e6
ccecaa1eee69d58219b5d10d4c4936533b7b05a029da68af2d7c3769d2ad7c60f63f1
3cef806a51ef3b71c46dcae032f2ca58b6aa1004283b57e58616358729b11f5adc890
73248a4e37c9d9dfe354fc51d9cc54a51bc887557df2b736d59e5e14dbef6b536d1c9
6b4db0947ec4f87c76a48e5f20d286134ad141984694e2b26103505f9e2ba5df65281
030be35fc2bac892acae045075e40e22ee81a723110de09308af15467c6e82026223
28c3c3da0d0ac40a44a8aeb192055ec60ba8ff3f3efcb4941485b4cd3f2420bfce4e73
73f4a861f64357f727c26a85087169b12890ee68cc86f740bae681b4c4919ab9bb75b
ebb64d2d50b1666f1e306f4f8f6fdb7f480d7b3f8142ac0ed3c8563c5419b4ab8f18654
00a75912fce9b0450cb03b8a3f9459c51532765917a6045f48fd462e79b00a7eeae7a
b9af4d2707acb4dd59dfebd472c422c8b18180a681efe806eaa1c5f7c7751efe903ad2
28b1aa8e48d8b745c47cb7c471d51e2b2c47d408df791b6ba2e73edb5113ee1d471f
684721086a6b13de607530cf3bc6af91a28982966b1bca945c6c05045c68b7980960
e8cfc87f112f0793550c87c73dcd8800537192bb750a06113d4046fd65204b956d45e
3cbaca58aa9f3c3ef688b0e88ff933a52635983bf2f14a663a800644138cdb6312b628
07e8a859f20343823dfceec88e5a3473b0a98ce2dccc67a39dc7b0cfd7ed95bc6ffcca
be435f617f746a83236a19f63e231c282a3b282e379c41ef00c47238d467c5ac61ae6
c5210b0bb67bd39ba2204744a2c054227a3ddba981e4e7ab1072fb491187d06611b
46083a7c3ad9bb997e16abbd94cbf78aa139024dd88d2418deff4c5f5b468bbe68f6e
d3c060440933390939efbdcac6eb7c7dd2c4481c67c479be52a1efb4c965e90bbd83

beda8b1a579573b07534ad517518c77b94d79f60942211fcca9881fa82fb1fd52d71c
df0fa9697bd55c37e60f5f03ffe711b135059ddae5c42852f7a98859307a94188990c3
684b02fa7d88d1d4b7150867828a188182861eb9cc2fd18168ee3f4a591c20a7d96c
1885962e73e56c65dfbaeb1386e81b92f05a6ca975c9ef3f2c0125b7313e1d1255b54
b43ccadbba7a81600c178aa853b2eef395d4b4e419fa9622297b95feddb6351d5439
9fca3c746798449b3c7549a7f419020be08d3a85c758875cd363b6574a86fdb6327e
ad517424f6d4b9574c42492dedd78cbe089991b569bd934a829b4716452e174dc47
bd9e6fdb0f0836a82a806acd7bc8638c57de1f08392d5f1937fb64cbf8343de42a761a
144e4a68637ca578fde3bfc4d6983219663550ea18f19e9106f88f5c2ac26763c9b101
a5ca0966a4f58be628cfc6d7f2121b7736516d07b55abbc0dc8d7895b8852e41cd410
534e12034f52edd1b92a8ec5667172db79b4e8d1fe4625bf21ea085675568967bb00
0420c791cec049e5705211893e08df4b6b9bc9f5b5326eebbffc7acf10dfde5e3d5f372
62bf257f2555cbe907487b4fba6fbcf6af2b0e69a5af4cf74c67b5684b7d8dae61ebf258
16dfb2575a845950443c43b6b550546eeb6a674f4a03783b60f3fa05cf40a6a26ff557
638146f46a180e30bd7b93a88b1767818fe0cd28df52f9248335ade859825b8f18f819
746980019258d2eaaac3f0d90f7f0e9a410955fa86001270b4b1d23dca2506101d30d
0aa81585856ec092fb39f1e7b9e8bf30a5fa56d5248d5c24b5b9148cf1e75a8e9a472
d76b4a0451d07b7f3d71d0cdff49a7332856456185c305fdee046593f41d0efd9cf54a
05a592c60499777f719a9cc771f664d00729434021630b159ac712e1169c6602827e
5975546f932de432f56584154a572512d8a0b3d2a99740e22991f3a9b280cdaf9bee
27989f11df04cdae2dd20f4bddc65af1bcf542750e021504491b77f39797c6c3bf7775b
d62ec14d879d2ecce7c4b960b1f26d6a6e152d3f3de6b5a043ba4d6641f7cd4f283e7
3440cbaebb594ee2e3d5fabe837e1ff3807e9931e7c69541c1d5974b3846f972117fe
2dffcbcc23b108bcfc495825ff5c1b1d4869544f09dcd829a5beabf2046d4b0a4109736
2598dfcca6ad29944510cb8daf9783d47cfdd66d53959be79b7ca1898358a9f233e8af
2043d458b426a7921a1bb3626ef0824c56e2c4964c617f2ad539cb4bdb0dceceea06f
168394e8edf7ab057c36102ef514d2fce94fd3321c75a18358f3ac8ee0b2a51b54268
75013975e9224b0dd9d85f1cd58b925c4a5ad4d87a27945fc5d96c6dfb6086de4fb5d
778fdccde3413ee8f59e3c1b097fad2f65b3fab3a03fbb2f06014c4f32b08771114095fd
5bb0ed1d5f427afdebfca5c6fc38503612cecbcff35ccaae50927a2c8f6a68c3028dbf17
ca65a04e80ead0b7dbb14250e8596bc94589e089ccab5e690e224b278a8d30aaa23
347a57480e1a65ad58d1cd67b7d7e7d9cbb24349a40a2f08c81fb7e811770916d3c1
757335c472187bd6751d1db75bdf80e641a9582c9e03a45905cde2b14d8bdd182f97
b15d050cb8e17a7e36169a0a18f239c5147d6da592aaa173e1a72658dfd90f060c70
639a791c84d087189fcd612e234b105173ecb5b0b9c9dada93cadc1edb416c763008
dc7475ad8a0d669dd6953bf244caffb46a9878526523ca4b4ffe4a1406099c229b4e3
2999ad3af34a40af5acee1a8e9e9453e54ee4b9c9673c959bc175e1588ad928cea5f
9b4ffd71566912735e7aa7317b6c6d56a12742576477c53acbca2b6126ed33d786dd
b909a3c1df37be8ab913a1011de7775f70126fb8a7ab4b4eac6a8e11e6d9c094d29fe
14cdef61bdd672c27b92bc0d64ae73d51fce1dde4325affdd100cec86ea4910322624
a0228b4f2c4cce31d00634a85204e35a1cbd92f23122b28c1ca3010abb1fe0d23a9b3
edd7a043d5b5ac01808b9541c48ab7c1a69aa304c666c511964c6340e7c4e490ecc
259bee39b2c56ae1ce962784479f0265798afa431537c9751aaf56760bfccdc74eb7e

bc5632a8b169d8d8d9a735f9042dcbd300c69ca1c758bc35789f1c1f4db616f8234bc
6e65114f10351d6d2245a7e401e5d2de4669663489da9606f1046bc148f7139303e6
6c712d4b99358087df7d63884ee7d71b4763a9f458831162344ac5ba8c791bc39ce0
b59e3f5a61ee74593d4ecb1e158d93bf29bb6bc3748ce76138ee1fd1c3b57f432cbfb4
3ab1ddf19692bc2e32b52f576d198dad386163d3da22444b8880939806bae869b8e
900b2f36c4b4543047f4c20ff01a712dce50012c2a37a0e183b2393574e277dc3de66
62c272278c78a3160269afe36eb69edb27b43fabdff8a6a3eaf5da02ebdd28bb848b2
dc1929ab9796dfd5d634c5e526f1953efdefaf9c247763dff34e245069401fec124f1fde
053b5fa7e2620f078c7dc26c1d02bc2dd7f27e0d99695540efdb0a7afb99424dfeac67
6ceed1905860fc96010d5d7234b3d0b6333221c5a817e2b1508dbbce27a89bac630
e655ff4f9e41d8740acf304f1f2a606ebdaebade49bd7e834383ff07e154bbc3150e2e8
f838c5ebd1768afb2066ec3f78ae32f8b127f87e7107cf77e02e7bd06ee332025c9e90
cb73ec323d5d0dd95d80457119bb23681a9059c4f0548fe9faf1d29586271481de419
f0313eb898fbba22501df0bad3a6c59237903f5c2d579c8dc1142c2023294e15a2037
39d34932f7e1b969ea8e319137ee56538528e181570ca02b0134dc44053341f01e1
a35f8aaa2f6be9790bf24b93ab5b1c4210c88bac7fc68ece0c66cdbeb9df40bd5046ba
cbe52b8bafeb710e2e0967ba90b3ce488242e30bcf6b9eda2188a5f90b4e2e46af0cc
c36ac541b81415c2439b10b67fb774b2ae8591b5a5ff7a3c4a51dc086bba4a3b4f97f5
72957c49e7c21f9c92e20297f07ad1ceb79cc157e6e87350c5e341d566fd0e18df9a9
803822c935ca8667cc40e3d9ca3af63a87394ebd139da915db7e3bf57723d5429774
9fe5284de3e10bfbc74d0d9f61cc3ec96fbb7f34657f6de40cd306e17722908b96dcc5
9e495e8a0ea2321ecc8dd7a7efe84308108bfb03f6be52fd2394729b6672c09bfd8d4
95bb6a98dc4be7381561bb2450b7f69385e192df49e5ffd58f8113158c0fb1aba0c2ba
21e0ca47753546237ad78cdae746f8a70c53efb377df0bf875891ce6e2ecfa05d9e8af
cc0587860eae52d6fa25b06f3a4d0395f9aaff7e79094c450b9ca92413d418746e944
b5f0085f13fec47ec9a64e2ba1c9366b161c7f35840ad63d1ec43ca6a5255249bb4c1
0b1054e5460b771c63721fae17fee0ac04b431c646c15c59548479c53849c6a3fe027
cd92d39e9f07fb34e17ac5ceeb5bec5bcc37f049069c1ae1d87c341941a9ed4a2b4cb
e3eddc445b610e3a40b378d712c60b67231f7d33f2f653dbddaa15d4e30c5ae230f3a
a9f0cc731cc1dc9e2fcfa80e24580f292f86cdbea4e62ef28efe98a27bb8dbaedb567ee
0e22185f7b2e8e2869202a95f79c6b1ff0919196b0f69cbbab4db40741e6789b6ef63e
09ab838fd7fcd747ec9739fa9128bb1418c760c8c7011c0dc017c096725ebc3d7c5c7
1e5262835538a1512e108bf82518b2164571aead90fcdc77cc864c2cf8c74190a7972
88752f5eb9930b91fb14e286ca1525c8e5dc88096533edbc4b235676e17249cbc725
eda497ef1d1c82049eab2daa62fabb792fb9a46ff814f1ec9aec0fb690122c7c0dd6280
c3a002bb35e94890a1a28e70baf52f3b4ca63b59e91ae0052b754bc373df34467d5fe
4d9f8d68deadad390a65e714998cb6ff7d9d99e6903f80302d3d228a462b319e34fec
cc6091abb54527e2ee3d9aaa8971f4240719bde5868e0e36c6c98527d791ee9f5611
c510532fd4d11064e45555fbc98ca62113f086317cdefed7f9898224c9c4400132a0bf
8d85d3d43295012992a507ede20902a7de6cfb932b300725afb577121602b894d34
1bda4640196da29ac134bf8443177cea9e7b5983b837d8bee5a729cbaf6c1d77b1a8
5a4ca3205768fe554af6670c7a10df32692a9b9acdbb90d89c6bf7c1e2af26940149a0
224fc748953dbe2f94fd166c4df041f2d0554f25b554af4058e61f45890ec73dee01aa3

217f118660875d164c800a9f54c4620d0d438f94a18b345a5a99f319c4633fa4a612d
39ac8117b05b3f125241d90f4bf40cf74fe6c5163b534013cec35067013c94b39d24aa
ea96ded8f02ce73c4757afb3b76eb956c65d494abf77e0efe63d50e43bc075640bdc1
9589e637612c4103086caf549063a90b54edd2fc45aeca62e094c43f5718b7b78ef0a
83e2ff8e8d522504fc87e1e0595a8dcf9bc1c90955e87f2052ab380314a746495f8292
0cdfe5ddc0f44f1addc36c1f68c08776ac85b9c3797989428db84b523edc0a5df351a6
6ff871a4dea5794ccea3bac49d1bf0b62e46654d1a4927456d25a3e20034de5029ea
c647d60a0eb6873e22c187667373b5ec0643e62fd261034d1c33060b0fa441bf6cb3
647aed726e999f597f8c5ce0a2cac1eecc05ffc83ef5575fcc833b04d14bb6e382a8a15
6f6d220f56591bdbbad0671d71608e96084b28971e25c643c425454247799508b8c8
bcc74ef4b63003145c722112c1d32137ab5573ca5b6e021f8a17e5e416bc3fcb33974
6bc8db3abcbd2cdf84a4a4d7aa17becfc3e0f27a275f9db6d5a3bccde5ee3cd189a7fcf
6a1c929b09eada338b20388495f966fcd7b46183a85a6584df73dbd1bc2278fa2e13d
b039884c546bd6618dc1545f9f40069e67ee4390d92743c27a951a75650edb6afece
4ed051f4a8a82cf19b29a3d3198b0343f783033c7c7b259ac1178241c5cf0ad88ba7c
3d9e04f946333e76418ebbc03e6ebf72a3cb5c85e0cf79c86ec17bc29cec546611594
ca50107555b5a8fbaf42932a42b96facc82c115be1f37eb08a45ab4de51e27dd7d72e
3e3892a890ac0ceae04e5ee5e4d5e61fe0004a986aa5c94d34a20647141fc544bf76
94f86591c139a13980586f9edf8e6c240e542515f4f0ca1e4aab8d188e356ec9e2a84
908adfbb57614c374786ac5ae8bc8de47b65dde999f2e50ba4e5152b629340dd6918
aea87b24bcbbd1c02413b2148477d46da4a29e3def6963379ddd62c439bca47118ff
83ee339c6183b8a105e05b5ce82e8ee86dad0343f73b1316fd349f7895b6feb023c98
12b5f6b87239bb351aeebe21ecfa57ccca30029264dd36d20d70d6b6fc9f6975c8e68
aaabdb1e2e0949fcdef61e08d918983f75eaad0d2553bcd89f2bb3b7ecdd0789387c6
c4584ca239f5f271259536d3fb07ce56568156a2edbaefdd58574b23ff07dd35944a0b
80fa1f9e2f1f4b2c0de6f459de16b0c286d5d56c3129d1997a4d054bdce8021988453
431e01097683d8327d598b549d09d8c153a12d835e855779e9c4c9a952662618a9c
625f013f6b973c5a11aa8c8f3289d7d852ebc9a6f74dff9471313f08642ca40631d73b3
6d564167a340b4a93b09977e28b6f8c3839d326b8db84586ee29c2d085d4f7462fba
675617b3bff833cedd68618f5209e67f9d097daa4dcae1db32a70a679aaa0335c600b
b9e4d9c1dfb3ae444fc481b3c45b982efd65aeb16996d6f129903fafc858090c8d5f766
d4d92cf3fda26c54660a3d54ed6c27f8cd0c1f21fe675fe6451f6b314391e505edf66aa
7b2b18d419b477ee91d175a6521fc14849d7abf3014d86017b14a587b72aa1dbc84c
51cc42109b96c256fa4177e3e10e546ed2c200ec1d33a8dac2623e50fc5bbc08a21f6
954ac7f411fc9b9d026cdc08c23b7de004e2952b13beab1b8dd53b155bd1445bf319
ad0eabfe13b7c2d5d353ca2abee0811717339dac13ecb48055cb967d1e66d7a3fc98
cc7be0366503e805edcf651e50ef391637a2e6061968a3cbc9a43ab50e231f9925c8e
d7d6bd728f7fe36f38a433cbc18fa9d5e7de6f924ef8cadadc28082a807c67b95c4c4e
e0c1f20c38fda77c1d3351190ef97a534867f0d4d160ad21f6faf5b6e20f246df72f5605
e0aac121e84d7a497e4c915847c2354b07060b9751fa5f801a938034c3892d98c9ee
5ee0d836323c5e1dfe87a081cde61b1c0574ace478bb5013c0705c2dc4d2e470e441
04a250fd87b8d7495b3ee26aae4ce68d1bbedf3fb1f291063983b9b715bc443755a6c
08f9d27e37f1c52a3202ab576dfb3a190ae663144cb26f1e26f832492c7c3a3f9a2ecd

6fe34e901f5edbb27239e796ff940c9ee8721de34c31b1de5593485aea25feef146ccb
e4ddbb8855686d0b96e52003c750062f909395e9bb8dfe8f1ffa561f817bf7be58ab6a
964217a058befc6274f738f6c0a105dac827002a5b4b73f136dcfce1a5e6361b2f0075
c1964dc908bb3028ce3ca05fedf039fb3450117b142c6423e4504ee5d78dd5708e8ec
1e4ed0861b9d0286fe4c61be9247119de7c16e064fac18f7f4b82befe87a2c6470516
19dbbd7800b73ac344909323cc034ba4ba6e2c3c46de2ca1997488d7bcb1c9dd532
8d7090eae81f1c92dc4c5a9503b8bb65c293a0fc15e478cd060fed4a7006562d55019
a64373ab2332abd2d2b593c54d3e75a2fcb99826e4896d6fb4ecf0720cb212df33e56
8253d354cc8a5bfc1dd27c0ed54837ca945208e7231c25e24548b33138cfce658600
7fc88db1aa392f62267b8101fd3b06b4ad1f1122a0eae31a243f72d12bb7612d70999
cc851860b057ec17e8eba3de0a0c93f8c0cfb19f717a3af2fcad4a8110b5ac863bc922
e459ff0a391b5fd2eb63027337bacc42fdcd509daa560fef17433214c2200066ade696
7bf9c9e6a5b9b645a997c4e5af44a40ecc205af607d3b7ecbd7ff277fa86950ef18e0b1
3f3021896733283577755fa629de3e28a4fa6bd01b66472d67f9a2c3bc360a04975d4a
f2b3aab3e589eb6bc0114acd737a2152d68fd276e6144d687a9863738c046530ba5b
44656b7bef42f180a78cd453afac974e9189378c8b4332244353a2dbc436f33e488f7
a61585b43cf3561df61c51bfbc16afda95de5e87b3ef118096705f18fa7495869d1227
0fbaacb759aed2278db100759405db0cb5dfe2b0047fe3a096302c5759889e49b35a
0144e9cc54334798fb2ecb164d05472d1911866698a3d4fefc75fde469eda43448d02
883280dcc2aade54f72df0ab258fa27e72e39f9c1a59cff98153d993746ff3a2b6b1a86
13e0340d6ff8cf171c1a15b2b8b7368ddfb5108d4464dac3de6612767b50b752ff340b
843875bd8814d5bef6ddef00fe6d160058fa45eddb751a0726736a78dfedb36a766c0
cc64c4c9858a3d2ac5d29c92bcef871c0cd0842ad0ec78a3512f5467d76379c2a40c2
cad71094ed83a29f7e36ba0a4d357bdc92b43daa3f02c0f826bd87213b8aa3ff62636
59b160ccc9e37ccbe612bf4707d3610ca90a44349ec76166cc7de373bec84022d75d
1f8a6ad7059e8ed6f6635fc0d41fc55f2bae1c12ff1cd2690bb60aeb65dfdec886c0cec6
d6c768c754be70aabf4a8206812730bcb5245c787e462972ce200a32194baebf56cb
5a996b2d178f5554868f0f9129f35b9fba3d418fd201bf996f04907415921e094b61a4a
39402adfd4f79a3a70da86bff2024c1173c55affde27033a669472ca1b405f0e51b4b5
031deb5509b5cd466c23efe0205ca6ae9d7af9f7c754193a88d61008e678ed17b728
09282aff77635b9cff512d45b953270d36614ff905acc0c4020a10998a3d84a46bc3ed
294075b8b71e5585f27625e9f35b75989766adee9e7d580d56ae051d28614b2b3ae
8396f1dba734aa0659be0f1bd36983850a7df3caa4faddcd577565f1b347a62ec4052
8616ee35c1edb908599b6e996194e11578154d3186dab7b348b9c83e2933f6da398
f09ecd03d689889583013c915c26453f294e333cd520770015c575e47e05596017e6
22ac4ddfeb92f0d947cd883244f81793423efb833d9ac923a0aca8ab40d3c4d8d3079
5068f9468ac10826bdb182146a04e6afdb038da5ec8ae4c6b076d1e86a9677e0a501
28632da187032cef3fb0945756bd7ecb3fca7ce55d6e461e814b43d8b31aa8327953
085d1c890b18077b9bf810e158188fbe9e1cb4a316fcb0699067fa39aed0dc40b5c1a
9cae41ecc5d7e01220ac23f2ff408a08a0d24900921f6abd2ede60a9521873493a777
1b2de1a1a73cfd891ae7c0c533d66d7fe6f575545d9e3037386ff34e63f7519b912863
d358c269e53d49ab63aabfe3aa413996c49d8f63e3a0a36a14943a4d488d65bcb815
c5d877ce9a4dfe2399627ece01d698ab87295206c48307dd0ece8545b2420241d04

d6ada315751d77f809b2910c61dee14c733517a5ea463996d6a4271e2de7a7f2120
30dfbf98d45816ad1daa2d5f97025e24abd3136bfae69a81f39533c7ce2f74ed4e09ef
a8ac680d5aca4bbe3fdf5a1e0494ce82cb3708798c0d61795350b6e1a06cd46e63bd
58a45dfe811f9afabd53b905aa8ebea94b8d5df1e5c903bb61ba9be09c1b90d11d21c
df508e759d30cc1c2dd31f0da450f41391ed2778c576976a9428ad25ad3bbaa76954
4f5c00e089d005c76c86ab2f0a2d69af57f9beddc929bf3485e2d646e73d6693fdad63
8335bb96186981f5024aaa4297032a5dbb50d5ebb9003ca05e9f7c9de6af37acea60
2049082c5027c7fa9b2969d36a9cbe725695394e37a65c3611c878efa7f99a6c27e40
39a408b61968c8bac4d7501872cb5f3ac89d4b0dfe29b41d80bdd6c7199585565992
6b6ef68b73bd713186786ae71fdacbe9b2bfee8546d08617030c8217713e545f6c78b
749ae9495d9c6537052be44c5d8332b1a8a6e2a2640795a72b55491781ea1afd2a5
1ad3a390157e847dfa9071db9e0d8c1808d947147134c275e95d324a1b177e788e7
2c0b4976e300d5586431999aabf5266cd305ba139f8d5ea5071603176dcd7578baf0
8ee1552d95cee5cd520d6be976620a6657f0c25a0c7e2158cf106396b78092adeb0f
740050713b45948d17f85a875460311be816f57cd8275e08baa5c5256894ac497c46
0d6d3ae4759c2784368951f249a8b15ef98273518a42b1299da9d27a06762f1a0194
8217df970ae6eefedac7b42930c1d9b394bbc35e81264d41c810b740b11e777145d0
26246ad5b0148344f822e3447f35d75bb7bcfd6f2f33b8fe998cd32d02defee0e99eb1
19979f77df05cd36a563181764b0e0142708db8525550f1325ccf79b11224aa328f6a
52c1c79435747414f3b8b59ab59ec739b980fa0a717598c029925025bdb7a7ab3ee2
b89aa306203329b884688897f0456a734fba279a41e3bd873a7f81bd510ae03216a3
6cf29f488d2b0e25462dd2bc3b4ba6e503889a7f54f11e3acc5bed8391ff84f94140c2e
738a1be854ef8076dea9c1216b93f3b6f3f3d295ea9141cc1dbe645550c69c11e4a8a
760c2a246aa81c5e59118a3acdf962b27f4989fd196b6170dc7c16f3d13a5ac01edce
2dbaec0a1320cc1d7205e54801fe6dfe970fd5b3d62392859a6008e9a4b74c79df80f
73575ecc6adb695ef2f557e097d4eb59e8e6080b0f02ddd2a35d1fe1f6cacdca9fa421
c024534176ea0d6354dd91576d7147f6abf9de3cacccfe19952cb401340975e6b3ec0
c463c1c5466df1cdba4520d0b20d3325bd4e1c289df9e3590a19634fc6236b103bde8
9cd20a4578b1320cacf591521240f7a4771d92e32f380b63cc8820f2b6c5b562da14d
a1bcb872971927ec6a97a1a49f6bb5e396eef0f681491150859c3bc72f2d8cc35fb335
ec87f0ae1831249eef69629d941b7f29c2e68a43c1838faf1cf2cf88e11bd38b9f0ba7f5
ef3bc94e7c39d92cec3f4e7d90ca8efb12df1071246da7c9939916360629e3523864a
9bfbda80af6d3c82d483daf9c118c5dddf68e04e668a7afcea36da317402187da567c3
4b229e419678dbb9fdb3a351491f8155b4b303ef9631f4a2868582d303ac446c36b42
98fe3d278c153ed6eec90fdee10ba5555f87ab8e61264bbfd1967ef77ef88c306d84f9
ac1d6a5e3cd58338c19ed1a42ab9241316a9b0cec44bd6b9e8d92c15d55430715d9
50cd9d65f591a1fadbade6c1ea2b31d77fa3f6a1c2ba9bad57942b12b5fc422bb7f496
0ca037f4198055212dcd11834b0c8a031605b77cbd8a89957a5d9b08499f8435e5a6
bfaf42c60ac47ffd80b77afa8e79779c6760b9047436a456c4b2d61c0b4ad92263017e
572413ba04552ec3221b3d22b958acf456464bc47b8c30bf3dcfb28e45f7ab142c5bb
4da1fb7d0b1e6f4852361532be8c78c6bd2d82900fc1c917bbccb31c5f03925b4bf10f
7d1b5f05687363f806cae54dd0c9de0a529520a38bcfb62d2523622d9c953bcdabc7f
1eb8c38b7655240f28997c570f191d0d3ec1ac1ac8e6d6f3ea6373a63dccafa0eaa99

44e799224a004c4c6243774836e821e002bdf2894367900f3e67edcbaeca3165cc2df
3c54d24ee3d53450eb9f9b97d5f0729a953ce3f094b89f5971956c5dab7be0c180daa
fa738481faf5b2175f9f0db6aa514b73200713bd41ab6ebfebdaaaf933f41773ed886bc
4bde2c5367cbd9bb03e6c685f475d03169e5ba1552a08416804f558493de94a15892
3f3de0ef9cf1d9384292bd9ebc61839ff617d29a838eb33792afd1db4c95fbf86d39d1af
d6fcd2a99abd7102d3fa076822ceb24051feda574be6510841f2c3e76031efca413f25
bb4bcce63cf285a44089d517a22f8bfa1fd72e52acb3358951e1ec466770110d21f56a
c3e6965a5b2f18cc0d59dbd33f86f2e6e51597bcd284a0da8ad8df9e3d1c8c3ae51f1e
9a21380e02f48889e5a0aeb1b49f8db477406859bf0f71f489655a4f106bdb8eb0118
0e2bb02acb703f9eab195acb6c7cc84e50be6d7ad31d7e25e8e06913e17605fc914d
1656175868d928c5cfc861e064d9e9aaa6641c581160e0aa82ec8d6e81856cd6ba9c
ba843901f1bf12b29ed17df60649a5c944112924c3d15b15e2085046e4c8614d549df
3590c788f092396c4daff4797c8672efc646073f8e46bf7abc9d0f5fc172a06cfc16e6b3
40db6762e28029c6baed1a6250f0ac76131bde6d91d5070b47e7cea5a437d3855f8c
5b4c4580b56521031355a06c89109f24b8c2985562567771137b5de45e2e4abeacf6
951a22c48a97ef8988df74a434aa3d3d1d7edaa98a5d5204afea0aab371285a78af0
e4461b8c869d596d7afd694334bc6b72d291f1886c5cf9b770ac49520ac40f3c8e72d
24598ef5b72f5a583ef92fcd1ee8c4ba9c1f5fce17e86dbb33ba8cfb51512d14e528606
67709e50f2bb370d97db4ada2f2f0c9a1ef0279e0b47e9cd187294cff405ef921838b7
870d76c0abe69fe1eb59625fb9d53bdd97ca57f7b16ea03c666d458db2afa61b80e51
b0439f0957886f499776b60e7d8fc2f71ec2a5b82732f8d28fa274ffde9b30ca1e17a9a
0574ead1a806379ce4feb54c142c50d422dd3be2168f1db7d064fbb388104cfd1cf39d
e08832d1278189eded12220b01d91d5341e8228cff2b4d2ff485eb0a6dcebedb0182e
3d60976744732d31b94b76652a35c58b26c6ddc22ee2c41b7be30611cb7cee1954d
7596fbdc4c16e4365cf703f6ef628adc0b6a297bdad289c33335d28ecac66458a3838
5fd27304c10c7a9238b00ea45ed43c5b637ea261646b0401bf1c36294b298cab9ddc
1f061a9e81cd5da7a409849d574688d11fd0e2b0f88ed2959f90b5339d6c837a930cc
ce0182f5902ba12e4b248ac763687222b84b7edd80c2a7fa420c2bcaa45fea0849b0
47aeac47ff5f4b097f2b37d1ccbc057d937466a986c6ceb09466b0823f8f3ef34539e24
5e7f3e4d9cd4b2f2ac11b534189fb33e1fe775f2e572f80971ba017ae3e4e204349865
43ed9366ceb1a24ec2fe6b87fb0464011c22defde76da8ef8b41046c1d87ec94129f7d
e9beae1ba91fbbf71fd31391f14af696fc69692dc243c89e8682a0cdfd5500fbe6a04a2
06d0141f2dc924b1eaf17f048ae33431961abb0f6aa2497aa9efacba605dcd83f4bb67
325de2aa6c51c467d49071798172464a528ce710e566f54a0ecd8f673610c8516751
c76ae690382b5814f4fead6b9c99239afc700fa184bd67e8ebdb10e1268b0618b9ffe7
b9b9a45f415b3f19a7ee53e6153dc9915b1a61378cdb58b94aa4168f573cf791d4271
2b794d0a657b3aa12334ba9e9f07f085916ff4a85e78af3d96a386ddc78d1c73a884e
27a0b5db2bdc3c86c57e69e14b17cc0f1e5778a995736e1e6e21479e12e5852e44d
54d0bfa267f8ab4b8baaed30b3e9e73754f9bb67f75be265f0cbe2a36e2340e9ac126
9bc2d41ea79c1e6d201085a010e580ffa2a0292679aa6c2fb82d74f0c49c7ed576bcb
8a2db98e0c0ecc116d1433021ece574f32c3e4ac373535976f32243566448847186e
cc945f1b7b25776ddc02eaa507df6bce885020bcccac22bf3a0bf4761e5656c5f15812
08c2689b0e0174795b55370ab271de1e5ac650fa29321122b42f19acdb29b3d8ebca

cd5b614c0412a2d3faf2f1bdec8684eb398ae4f820bf218b7c9cd88bf5126345bdde64
9dde32cc2b70d704b4e734dac4d254b6f95db8d13d77c22b497ef4870e47f5d5e3efa
6f4902b19d143d0b00656dc8b99d83e8001476c5a95b99049998664a56a686537a9
c25f79274f6f32fe83c742cb0cbc390556af6f8f0932f0db87dd08eb379c70fe8c5d614e
accaa6d0d8606f10db77636caab8432c2f8c76a1810a104fe099d80f9132a2174d394
b52c12bd4e00539a86faf3ccf871f712c5a0fc2a739eb374f070f22b3afd0cf610a3f947
ea8885dd0c35dcaa612c379a45b96430cda720f15270401b26197b42a5e36841eb2a
eb1d6ea9ed365879ac4bafdf14c8c878512801332158ef8d6e54ef284854314a7b313
0103213b4c684acd6de8ce49ec8e1441de87ff4153152acae95f758e402fd0a50dcee
c1b5d4f560c44330a208650fe8f7e0d46680830e6643e2f91a26346d2502804564a10
bf803ed9c8412b7484db01c50b2a807a4d6577f102a0dc2e0e2ab1334b572c658867
e0d759361622fa5336acdc6331b7d1cdb142a48491edeb6b9cd4cab7fc2f5131e4204
53f433abcf66074387102408b33d5099036d4fd6933dc70e291ed49dfb08993bf972af
9927f92c51f05f1321fa67df20366f4ec54efced8141e7d9162cd2f65d79161258bce9e
7a7ae96cbea6f78f1d96574dd0d63b7365a6de5ebf5806d287402900d9362b30bef8
962aeabf96b927ee00f60f718604816e9cc0a0ed785083e231712a2a1aed9b84e98b
98fc358f69e43b47c08d1dbf96d55331c5cafdc5bcc3dede6f97fb80b10ccb583c66638
5cb06fdcc7c0731a839a7b45b9268e90406189d010cb32ec3095b234407840b9160
0ba7e2804893143565031984780438fbd688cc8db7906cbe730d63b53eff058cb61b
ea1fbe8aaa436c19ac9c9763f9c987ecb9a153c0106c89ac3993ab59e6d7531975c2
bd2fa40b6cdabe9ce0e0097f65b19556e3c106e00341e60b29a787795acccaa49d16
efb37e4d2c9e14218bd2653c47babaa9c5c89954d434f965b376a00deece691fc002b
effe17d5d9329fb5d77644dd4df1e35787e7e9548268c0627b8e9b4765185e9fdcf6ed
104514965e750a5e3e21d236c890ccdc45946ffe2b398cd160b4a92311b270672463
d151af8887059c19ebfec5d87

## B.2 License not granted response

```
HTTP/1.1 200 OK
Server: nginx/0.7.65
Content-Type: text/xml
X-Fuzz-Disable: True
ntCoent-Length: 672
Expires: Fri, 29 Jul 2011 20:34:03 GMT
Cache-Control: no-cache
Vary: User-Agent, Accept-Encoding
Content-Length: 672
Date: Fri, 29 Jul 2011 20:34:04 GMT
Connection: keep-alive
```

b4d8d24cc943a2e2307855e50225dd418cf84630a1306a3a028150fb631f65a8fdd3
d364b3b8ba317089da55d4499dabc69ca8d4b61df6cfb443a03d6c65ab701fd530ac

e697222f3a8a4c217872ef8b25275fc27a34199688b24bdb8eebfb1f349761e1634ef
c7e170d7344581067b40d166de3108cbc6e5c39f81895ebc6eeaaba8ae2d5f5ded6f
1b6654fe70cfa877d08344d53ed7a5f3556ddae94c1fd78baaea5ff87c0ecb90fd7069
825da5a676419264af1d4985e86156c6a9814c8e44df093a70a413b74773f343ec2
1096622dcc5f1aebb00e981b534193d7fec8bb887f5e0318460bf476e747342dc83b
3914762a3faf02bcfd9196354b288f02f4e8705848dbb077fd48471a21596dc04dafa
e4d361a10a83f88f34262e110c1a085fa547d4debf00e64847a6c1d3c0a6b643b15a
797b9886f5566221598e28f0f8cff3ccc9244b4a5427aab59042108db

# Appendix C

# Comcast Xfinity cookies

## C.1 Cookies set by Xfinity online sign in process

- MYPORTAL=em=email%40address.com&gt=Carlos&zip=99999&guid=urn%3Auuid%3Ac0d29361-35d3-4e2a-a41e-2b06054a2945&auto=1&tid=4354cf501c1df6b38f07-99d97f2ded04c144f7c7f6380187809f2386091f;domain=comcast.net; expires=Thursday, 28-Jul-2011 20:18:49 GMT;path=/

- s_ticket=TFNUAQECAIAxSRG8qYaw0GLYPRoyl2hs7S3pRQoKF2gio_1lNLmNBO1_EFF2tJiiomFgXmh6g64x6LW3vHbMD5Vkov-mQAu-NyUc3BImidXO_VZFisLkcheOQrQIQFz4oenxUj65NhQ_AamxQr4QIrz6FJsg0-Xk_dbcgtX3j4tF3aGHNrPpuBA89IiDTjt_hM0uB04ZU0KhAAAA8EfEP-OJDNUBUsy2YV-byKJLKEuI50y9DWg3bf40tIlBVSslctM-HP5ZBEo6JZ3fdUCNIbx3t1m4GSaZa-aJ824G4QNTAxyTQ9ijhWbCT8-5OkVOeJAtFmE9IcC1KhRxH9hU2AmSNHllVS6cbXRyOsAV4Pg0_lqZZo8zqkLR_0gCGbsnRboJZrVV1gQaPovOlDdqpCVP69L4iQtgYtFPJ6HXX0-MAdWZSEvBA_fdgXdWCo6sswnDebf_LhT4XxsLYnVa1GSlPFc3DsOoSOsGJmKwA3guW7kA5L1v3U7Vmsju9_k6dmaCA1GRxSMUgBAw-A**; domain=comcast.net; path=/

- session@comcast.net=TFNUAQECAIAxSRG8qYaw0GLYPRoyl2hs7S3pRQoKF2gio_1lNLmNBO1_EFF2tJiiomFgXmh6g64x6LW3vHbMD5Vkov-mQAu-NyUc3BImidXO_VZFisLkcheOQrQIQFz4oenxUj65NhQ_AamxQr4QIrz6FJsg0-Xk_dbcgtX3j4tF3aGHNrPpuBA89IiDTjt_hM0uB04ZU0KhAAAA8EfEP-OJDNUBUsy2YV-byKJLKEuI50y9DWg3bf40tIlBVSslctM-HP5ZBEo6JZ3fdUCNIbx3t1m4GSaZa-aJ824G4QNTAxyTQ9ijhWbCT8-5OkVOeJAtFmE9IcC1KhRxH9hU2AmSNHllVS6cbXRyOsAV4Pg0_lqZZo8zqkLR_0gCGbsnRboJZrVV1gQaPovOlDdqpCVP69L4iQtgYtFPJ6HXX0-MAdWZSEvBA_fdgXdWCo6sswnDebf_LhT4XxsLYnVa1GSlPFc3DsOoSOsGJmKwA3guW7kA5L1v3U7Vmsju9_k6dmaCA1GRxSMUgBAw-A**; domain=comcast.net; path=/

- rm_ticket=Uk1UAQEAABS9E6O3EkXy7mCX70Q_v9q445UAFRARH6ll6msbuV4tt7ytQ4bAAAABQAlZUPASoL8FMDQ9yrR0Jo7G3vi1yYu0E1khHp1XnvX85GDoDFzIXgF3N9ubtbjOSB5Yn7b-NUFDGspnLitEJa8GW6M3wEvce2KiqHeewvXbPqM8FVhwu6kVq8R8I7q

BZUMSLjFJTQqUnycR00e0b7ay5lYFC762P2wa_zvmbjzf__KCjnYf8CGqpb6Z5pzFNH
wpr53kmpQ26kpn13MXUnltw2LmKO_C7uqSnL8_tz8eouVJJSbtuwAN2fYpGwi4qIxQR
oFpx_wguHZXgmK8k8H2kX06XH_mGOxcvs1V8_wyUY9x64Wod01GDkKB1RKZ48w-G_rT
byxu0B7lW8N56UJvu6tFJhlf8rA93tsOJmAZdDox2922TzmBP2RtGCCT9g4_XbFtzuC
CUcvnSSo719pYvpH_ctWjeFa6reNHzQ71; domain=login.comcast.net;
expires=Thursday, 28-Jul-2011 19:16:04 GMT; path=/; secure

- tg_ticket=VEdUAQEAABRQaKd7YRxkd9MLejeq183WYCvx-RDBCvux0FQSEut-uxeXde
3QAAAA4Bqj0t4Hd2N2gHv0Ym5kqmwg3CU99SmixC8tPx3vKXdT8RVCFI6Pwc0J_SwJ1b
BMLCHE6fLoFA_sNlFPlPzvNSbbGFWKZa01RtjmWi7XRj0U8SBVQmXKq-Eg4VYUDUc4_C
GWlDriT47M7uDz3pY90eDaULXgaHUJJx4zT5lsEWP_hqgHVuv3AS4XFBX0HRqIY0E9Op
xQdSh3hDxn2b8nFlft66Uy8oiKiSnnAc6CaaBftH_67-lTDTJjyz3LriQ8SYgbTAiL-v
jh4GVQ0LeO3aoXoMwDPDI7Ifl5qG8Oa-3L;domain=login.comcast.net; path=/;
secure

- tls_s_ticket=TFNUAQECAIB0vNff5sXT2lScClsb26VEWvW0k_UsAmqANHSw2Z0QtsE
YQn6_Ums58-U1w-aW66Ok6joFg5utOhG0ffbH48nGJv5q2HKAjjSDMw1niJspt8H8tTt
ZxXWgh3E4D2jfdGWEFoyRMwHtgRZ5lCc3aPYDzHW9Pz2un4D71-VKOQhXfhBqSr4eQ7G
zT4b7HuGbQz_hAAAA8DFQRbfamF5E8XQtjEVyPLpC2aSMLxsLThWPGReZRYMc4HeSKEn
FMg87j5RxiVBov4s4qw6kDSNTa-98GKWURWwJN4BOIpq0cJEHn6sSimfq-XDkFvGQVAx
EJhfk0HJihshsY4EdCVbfS7-MQqdRgtQ6T3gtdt3XM6c4bpTOSZII74QmGWo4Eo9Ykpa
_REOIVxHoXDHMqkXQK02F5QaaPhM4P3XJIJKVaa-KxAXOuRNUKfd5MQMrOe5fz0tj_hK
8vSsWkfvNu4XA6Qyvl6Fymtl48_-I7EiiHU4MInHQs8u-e8RiZdZM-XnAqDM0R1cSUA*
*; domain=comcast.net; path=/; secure