Opinnäytetyö (AMK / YAMK)

Tietojenkäsittelyn koulutusohjelma

Yrityksen tietoliikenne ja tietoturva

2016

Justus Jokinen

RTMP-MEDIAPALVELIMEN JA -SUORATOISTON TIETOTURVA

TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

OPINNÄYTETYÖ (AMK / YAMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma | Yrityksen tietoliikenne ja tietoturva

2016 | 36

Justus Jokinen

RTMP-MEDIAPALVELIMEN JA -SUORATOISTON TIETOTURVA

Opinnäytetyö käsittelee suoratoistoon tarkoitettua Real-Time Messaging protokollaa (RTMP) keskittyen tietoturvapuoleen niin suoratoistomedian tekijän kuin dataa käsittelevän mediapalvelimen kohdalla. Työn tavoitteena on havainnollistaa suoratoistossa käytettäviä tietoliikenneprotokollia ja mediapalvelinten toimintaa sekä tuoda ilmi suoratoistettaessa välittyvän informaation tietoturvariskit ja -uhat. Tutkimus keskittyy enimmäkseen sosiaalisessa mediassa esiintyvään suoratoistoilmiöön, mutta samoja tutkimustuloksia voidaan hyödyntää myös yrityselämässä.

Tutkimustyö toteutettiin kahdessa osassa. Ensin selvitetään suoratoiston teoreettista puolta näkökulmien ja teorioiden avaamisessa tutkimustehtävän toteuttamiseksi. Empiirinen osio käsittää RTMP-mediapalvelimen tietoturvan tutkimuskehyksen Kali Linuxin murtautumistestaustyökaluja sekä protokolla-analysaattori Wiresharkia apuna käyttäen. Tutkimusta varten luodaan suljettu virtuaalikoneympäristö, jossa suoratoiston tietoturvallisuutta tutkitaan tutkimuksillisin motiivein.

Opinnäytetyössä puhutaan myös erilaisten suurten suoratoiston palveluntarjoajien roolista ja niiden toimitavoista. Palveluntarjoajiin liittyen spekuloidaan myös suoratoistoa tuottavan tahon eli käyttäjän roolista ja vuorovaikutuksesta katsojakuntien edessä. Tutkimuksessa myös käsitellään ilmiötä, jossa suoratoiston tuottajaan kohdistetaan internetissä ilkivaltaa sekä miten ja miksi sitä mahdollisesti tehdään.

Suoritettujen tutkimusten tuloksena käy selväksi suoratoiston tuottajan käyttäjätietojen (IP-osoite, käyttäjätunnus ja salasana) olevan erittäin helposti kaapattavissa, mikäli verkkoliikennettä kuunteleva, hyökkäävä osapuoli on oikeassa paikassa oikeaan aikaan kahden suoratoistolaitteen välillä suoratoistolähetyksen aloitushetkellä. Tutkimustuloksissa käy kuitenkin ilmi, että tietoturvalliset käytännöt saattavat olla tarpeeksi suojaamaan suoratoistoa tuottavan osapuolen herkkäluonteiset käyttäjätiedot. Tekstissä käydään myös läpi ennaltaehkäiseviä toimenpiteitä tietomurtojen välttämiseksi sekä syitä sille miten ja miksi tietomurtoja suoratoistossa voi tapahtua.

ASIASANAT:

RTMP, suoratoisto, tietoturva, mediapalvelin

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Business Data Communications and Information Security

2016 | 36

Justus Jokinen

INFORMATION SECURITY IN A RTMP-BASED MEDIA SERVER AND LIVE STREAM

The focus of this thesis is to research the information security of live streaming in the instance of Real-Time Messaging Protocol (RTMP) siding on both the live streamer and the media server to which the live stream is broadcast to. The objective of the thesis is to discuss different information transfer protocols and the activities of a media server as well as bring to light different threats in live streaming. Although the results that are achieved in this thesis mainly deal with the social media aspect of the live streaming phenomenon, it is possible utilize this information within a business environment. The research of this thesis was conducted in a theoretical and a practical segment. Firstly, the theoretical segment focuses on what live streaming is and how it works by discussing the different roles and operative policies of live streaming service providers as well as the position of the live streamer itself.

Secondly, in the practical portion, a virtual environment was created to observe an active RTMP live stream in which a server receives a multimedia data stream from a sender. This data stream was then listened to and possibly captured or interfered with different penetration testing tools via a networked machine running a Kali Linux operating system.

In conclusion, it is made clear that there are major information security issues with RTMP. By examining the data obtained by the practical section of this thesis, it is possible not only to steal user live streaming credentials but to also capture the live stream media straight from its source or media server. However, it is also possible with cautious information security practices to deny an attacker access to vital information, such as a password, and being able to hijack a live stream. Finally the thesis also reflects on why and how an attacker would attempt to steal secured information and how such behavior can be defied.

KEYWORDS:

RTMP, live streaming, media server, information security

SISÄLTÖ

LIITTEET

KUVAT

KÄYTETYT LYHENTEET TAI SANASTO

CDN	Sisällönjakoverkostot (Content Delivery Network) ovat ympäri maailmaa sijoitettuja datakeskuksia ja välimuistipalvelimia, joita ylläpitää niiden palveluita myyvät tahot.
DDoS	Palvelunestohyökkäykset (Distributed Denial of Service) ovat tietoliikenteessä ne hyökkäykset, jotka hidastavat tai estävät palvelinten toiminnan jatkumon tulvimalla kohdepalvelinta jatkuvalla verkkoliikenteellä. DDoS-hyökkäyksiä on erilaisia.
FMLE	Adobe Flash Media Live Encoder on pakkauksenhallintasovellus, joka on tarkoitettu pakkaamattoman multimedia datan liikennöimiseen paikalliselle tai ulkoiselle kovalevylle. Sitä voidaan käyttää suoratoiston luontiin vastaanottavalle mediapalvelimelle.
Full HD	Teräväpiirtokuva (Full High Definition) on markkinointitermi korkealaatuiselle ja tarkalle videokuvalle. Termiä voidaan käyttää kuvastamaan videodataa, joka on tiedostokokonsa kannalta vaativaa.
HLS	HTTP Live Streaming on Applen luoma HTTP-pohjainen suoratoistoprotokolla. Kuten RTMP, HLS:ää voidaan käyttää multimedian suoratoistoon vastaanottavalle palvelimelle.
HTML	Hypertext Markup Language eli hypertekstin merkintäkieli on avoimesti standardoitu kuvauskieli. Se on kieli, joka yhdistetään internetsivujen kirjoittamiseen ja esittämiseen.
HTTP	Hypertext Transfer Protocol on hypertekstin siirtoprotokolla, jota selaimet ja palvelimet voivat käyttää tiedonsiirtoon.
IE11	IE11 on Internet Explorer -verkkoselaimen 11. versio, joka ennen toimitettiin muun muassa Microsoft Windows -käyttöjärjestelmän mukana. Sitä voidaan käyttää myös muissa käyttöjärjestelmissä.
IP-osoite	Internet Protocol on verkkoliikennettä käyttävien laitteiden yksilöllinen tunnus. Se mahdollistaa tiedonlähetyksen laitteiden välillä, sillä se auttaa lähteen ja määränpään tunnistuksessa.
MMS	Microsoft Media Server on Microsoftin yksinoikeudella omistama RTMP:n kaltainen verkkoliikenteen suoratoistoprotokolla, jota käytetään Windowsin mediapalveluissa.
OBS	Open Broadcast Software on avoimeen lähdekoodiin perustuva pakkauksenhallintasovellus, jota käytetään useiden lähteiden yhdistämiseen multimediaksi. Sitä voidaan käyttää paikalliseen median tallennukseen tai suoratoistoon.

QoS	Quality of Service eli laadunvalvonta on keskimääräisen suorituskyvyn varmistaminen missä tahansa verkossa. Se varmistaa tietyn tasoisen suorituskyvyn dataliikenteelle priorisoinnin avulla. QoS usein yhdistetään telekommunikointiprotokolliin.
RTC	Real-time Communication viittaa integroituun kommunikaatiomedia, jota käytetään esimerkiksi HTML5:ssä. Se on mahdollistaa verkkoselaimen kautta käyttäjille kyvyn selata multimediaa reaaliajassa.
RTCP	Real-time Transport Protocol Control Protocol on Real-time Transport Protocollan sisarprotokolla. Se ei itsessään ole tarkoitettu datan lähettämiselle vaan laadunvalvontaan RTP:n ohella.
RTMP	Real-Time Messaging Protocol on Adoben omistama Flash-pohjainen suoratoistoprotokolla, joka on julkistettu avoimeen käyttöön. Sitä voidaan käyttää reaaliaikaiseen suoratoiston esittämiseen verkossa.
RTMPE	Real-Time Messaging Protocol Encryption viittaa RTMP-suoratoiston kevyeen salaukseen. Se on oma versio RTMP:stä, joka käyttää Adoben omaa salausmetodia, ja jonka Adobe omistaa yksinoikeudella.
RTMPS	RTMPS on versio RTMP:stä, joka käyttää TLS/SSL -tasoista salausta liikenteessään.
RTP	Real-Time Transport Protocol on RTCP:n kanssa samanaikaisesti toimiva kommunikaatioprotokolla. RTP toimittaa dataa laitteiden välisessä kommunikaatiossa, mutta se ei itsessään kykene laadunvalvontaan. RTP:tä käytetään usein RTSP:n kanssa datalähetykseen ja sen hallintaan.
RTSP	Real-Time Streaming Protocol on reaaliaikainen suoratoistoon tarkoitettu protokolla. Se toimii RTMP:n lailla, kommunikaatiossa mediapalvelimen ja lähettäjän välillä. Se ei itsessään kykene lähettämään mediaa vaan se toimii kaukosäätimenä multimediapalvelimille.
TLS/SSL	Transport Layer Security ja sitä edeltävä Secure Socket Layer ovat kryptograafisia protokollia, joiden tarkoitus on tarjota tietoturvallisuutta tietoliikenteessä. Sen keskeisenä teemana on avainparivaihdos, joka luo kahden kommunikaatiolaitteen välille luotettavan datayhteyden.
TCP	Transmission Control Protocol on verkkoliikenneprotokolla, joka on tarkoitettu toimimaan kahden verkossa kommunikoivan laitteen välillä. Se takaa laadunvalvonnan, sillä rikkinäiset ja vajaat datapaketit lähetetään uudelleen.
UDP	User Datagram Protocol on yhteydetön protokolla, jota käytetään kahden verkossa kommunikoivan laitteen välillä. Se

	ei takaa laadunvalvontaa kuten TCP, sillä rikkinäisiä datapaketteja ei lähetetä koskaan uudelleen.
URL	Uniform Resource Locator tai verkko-osoite on merkkijono, jolla merkitään WWW-sivustoja verkkoselaimessa.
VCR	Video Casette Recorder eli kasettisoitin on vanhanajan tallennuslaite, jolla pystytään tallentamaan ja soittamaan mediaa analogisesti.
VOD	Tilausvideoilla (video-on-demand) tarkoitetaan multimediaa, joka on saatavilla verkossa sitä pyytävälle taholle. Mediasisältö lähetetään esim. TCP:lla katsojalle tämän pyynnöstä tai tilauksesta.
WSE	Wowza Streaming Engine on Wowza Media Systemssin rakentama multimediapalvelinratkaisu. WSE-palvelinta voidaan käyttää tilausvideoiden ja suoratoistojen tuottamiseen ja jakamiseen IP-verkoissa katsojakunnille.

1 JOHDANTO

Kuten kuvien levittämisellä Instagramissa, viestien lähettämisellä Facebookissa ja videoiden lataamisella YouTubeen, on suoratoistollakin sijansa nykyajan sosiaalisessa mediassa. Suoratoistomediaa (live streaming media) voi kuka tahansa tuottaa ja jakaa sisällöstä riippuen sille tarkoitetulla verkkosivulla. Yksittäisten henkilöiden ja ryhmien keskeisimmiksi suoratoiston lajityypeiksi ovat osoittautuneet live videoblogit (tai arkityylisesti kutsuttuna vlogit), videopelisisältö ja opetusvideot. Tarkoituksena suurimmassa osassa näistä suoratoistolajityypeistä on tuottaa sisältöä niistä kiinnostuneille kohderyhmille ja mahdollisesti antaa myös katsojalle mahdollisuus olla yhteydessä itse suoratoiston tuottajaan ainakin amatööritasolla.

Yrityksille suoratoistettava media tuo mahdollisuuden mainosten esittämiselle. Esimerkiksi suoratoistoa tukevat kaupalliset sivustot, kuten YouTube, Twitch.tv ja Justin.tv, esittävät mainoksia jokaiselle, joka suoratoistoa tulee katsomaan. Mainoksia esitetään sekä mainospalkkeina www-sivun laidoissa että videon sisällä ja sen yhteydessä. Syy mainosten näyttämiselle on, että se tuo palveluntarjoajille mainostuloa. Kuten televisiossa, on suoratoistossakin mahdollista pitää mainoskatkoja, mutta mainoskatkot tulevat usein suoratoistoa tuottavan tahon aloitteesta. Esimerkiksi Twitch.tv-sivustolla videopelin pelaamista suoratoistavan henkilön on mahdollista painaa itse mainos näytettäväksi saaden näin jokaista katsojaa kohden tietyn määrän rahaa mainoksen esittämisestä sopimuksesta riippuen.

Tässä tutkielmassa keskitytään suoratoistoon tarkoitettuun Real-Time Messaging Protokollaan (RTMP) ja käsitellään tietoturvauhkia ja -riskejä liittyen sekä suoratoiston tuottajaan että suoratoistoon tarkoitettuun mediapalvelimeen. Työssä selvitetään suoratoistoon liittyvän teknologian perusteet ja edetään kehittävällä tutkimusmenettelyllä keskittyen käsitteiden selvittämiseen ja ratkaisujen löytämiseen. Empiirisessä osassa tutkimusta suoritetaan RTMP:n tietoturvan läpileikkaus käyttäen ”Kali Linux”-murtautumistestaustyökaluja sekä protokolla-analysaattori Wiresharkia.

2 SUORATOISTOPROTOKOLLA RTMP

2.1 Perustiedot

RTMP eli Real-Time Messaging Protocol on sovellustason protokolla, joka on suunniteltu audion, videon ja interaktiivisen sisällön kanavointiin, paketointiin ja toimittamiseen sille soveltuvan tietoliikenneprotokollan avulla (Parmar & Thornburgh 2012, 1). Se on alun perin Macromedian yksinoikeudella kehittämä protokolla, mutta vuonna 2005 Macromedia myytiin kokonaisuudessaan Adobe Systemsille (Adobe Systems Inc, 2005), mistä syystä protokolla ja kaikki siihen liittyvä on nyt Adoben omistuksessa. Joulukuussa 2012, RTMP:sta julkaistiin spesifikaatio avoimeen käyttöön. Tähän julkaistuun versioon Adobe ei ole sisällyttänyt palvelinpuolen turvallisuusmenetelmiä jättäen datasiirron ja tietoturvan ylläpitäjän oman teknologian ja käytäntöjen nojaan.

RTMP on pääsääntöisesti suoratoistoon suunnattu protokolla, kuten Real-Time Streaming Protocol (RTSP), Microsoft Media Server (MMS) tai HTTP Live Streaming (HLS). Sanalla ”suoratoisto” viitataan jatkuvaan multimedian lähettämiseen verkossa lähes reaaliajassa. Huomattavana erona RTMP:ssa muihin aiemmin mainittuihin protokolliin on se, että RTMP on Flash-pohjainen protokolla. Sen avulla pystytään lähettämään multimediasisältöä verkossa Flash-yhteensopivasta pakkauksenhallintasovelluksesta Flash-yhteensopivaan mediapalvelimeen (RTMP-mediapalvelin), josta dataa pystytään lähetysmäisesti jakamaan katsojakunnalle (Adobe Systems Inc. 2016). Adoben mukaan multimediasisältö voidaan palvelimen puolella muuntaa HTTP:ssa katsottavaksi riippuen palvelimella käytetyistä sovelluksista.

2.2 Ominaisuudet

RTMP tukee sekä TCP- että UDP-tietoliikenneprotokollia (Seel 2014, 3200). Yhteys palvelimen ja käyttäjän välillä on jatkuva ja vuorovaikutteinen molempiin suuntiin. Käyttäjä pystyy lähettämään komentoja palvelimelle ja toisinpäin luoden täten vahvan kaksisuuntaisen kommunikaation. RTMP mahdollistaa videoiden tilaamisen (VOD, video-on-demand) ja suorien lähetysten seurannan. Yksi maininnan arvoinen ominaisuus, joka saa RTMP:n erottumaan muista sen kaltaisista protokollista, on sen dynaaminen suoratoisto. Tämä tarkoittaa käytännössä sitä, että videon laatua pystytään

vaihtamaan automaattisesti sopivaksi verkon siirtonopeuteen live-esityksenkin aikana. Tämän lisäksi suoratoiston aikana katsojien on mahdollista katsoa videolla aiemmin tapahtuneita kohtia samanaikaisesti videota nauhoitettaessa ilman, että heidän tarvitsisi ladata koko videota. RTMP kuitenkin tukee vain muutamaa tiedostotyyppiä, jotka ovat MP4 ja FLV videodatalle sekä AAC ja MP3 äänidatalle (JWPlayer 2015).

2.3 RTMP:n käyttöönotto

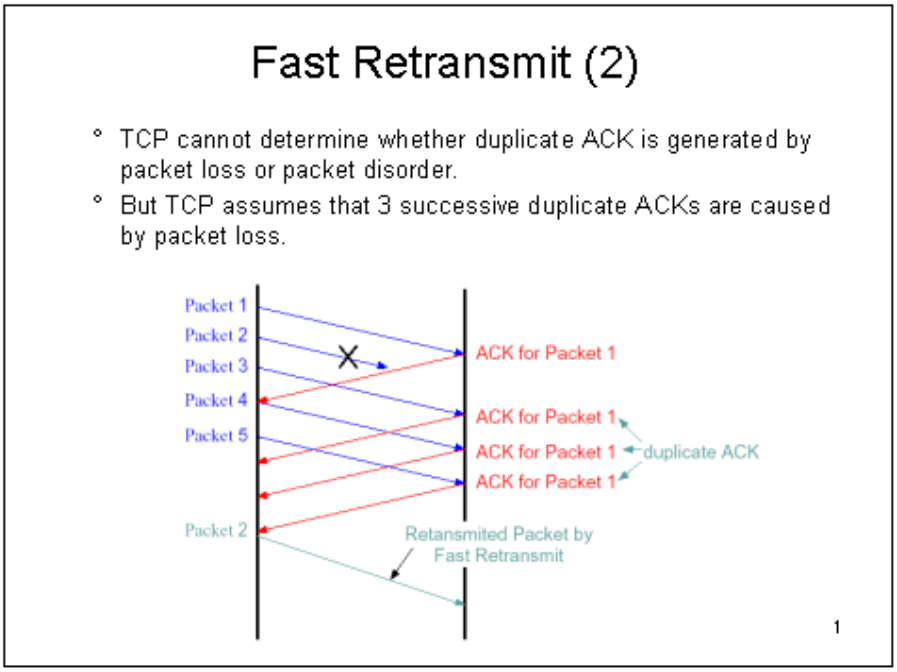
RTMP-protokolla on otettu käyttöön mediapalvelimilla, joilta halutaan jakaa sisältöä yksittäiselle tai usealle käyttäjälle. RTMP-mediapalvelimen avainominaisuus on suoratoiston mahdollisuus. Suoratoistoon kuuluvat muun muassa opetusmateriaalin, konserttien, uutislähetysten ja viihteen tarjoaminen katsojakunnalle suorana lähetyksenä sekä mahdollisesti tilausvideona. Suoratoistoon ja videomateriaalin tallentamiseen tarvitaan RTMP-yhteensopiva pakkauksenhallintasovellus (live video encoder), jonka avulla lähetys taltioidaan ja lähetetään eteenpäin. Tallentavan sovelluksen sekä TCP- tai UDP-protokollien avulla dataa lähetetään verkossa RTMP-mediapalvelimelle eli ingest-palvelimelle, jossa lähetetty data vastaantotetaan ja käsitellään suoraa lähetystä ja arkistointia varten. Pakkaamattoman video- ja äänidatan käsittely ja pakkauksenhallinta tapahtuvat lähettäjäpuolessa (Wankel 2011, 2–4). Julkisella sektorilla lähetetyn datan pakkauksenhallinta-asetukset ja -vaatimukset tulevat usein RTMP-mediapalvelimen ylläpitäjän ohjeista. Tämä johtuu siitä syystä, että esimerkiksi liian suurilaatuinen (Full HD, teräväpiirto) data voi rasittaa mediapalvelinta ja se saattaa aiheuttaa palvelinpuolen ongelmia. Vääränlaatuisen median lähettäminen ingest-palvelimelle voidaan hylätä palvelinpuolessa kokonaan (Twitch Help Center 2016).

2.4 TCP- ja UDP-lähetysprotokollat

Yhteyden muodostamisen jälkeen RTMP-yhteensopivalla sovelluksella mediapalvelimelle lähetetään dataa joko TCP- (Transmission Control Protocol) tai UDP- (User Datagram Protocol) lähetysprotokollien avulla. Lukijan on tärkeä ymmärtää näiden kahden protokollan eroavaisuus suoratoistetussa mediassa.

2.4.1 TCP suoratoistossa

TCP-lähetysprotokolla on suunniteltu informaation lähettämiseen verkossa yhdeltä järjestelmältä toiselle virheettömästi käyttäen IP- (Internet Protocol) osoitteita. Se on optimoitu tarkkuuteen enemmän kuin välittömyyteen, sillä rikkinäiset paketit lähetetään vastaanottajalle uudelleen. Tämä aiheuttaa teoriassa katkoksia ja odottamista suoratoistettavassa mediassa, kun virheellisiä paketteja ilmenee (Wankel 2011, 3). Julkaisussaan The Case for Streaming Multimedia with TCP (osana teosta Interactive Distributed Multimedia Systems 2001, 213–218) Charles Krasic, Kang Li ja Jonathan Walpole pohtivat TCP-protokollan ominaispiirteitä suoratoistossa sanoen, että reaaliaikaisesti lähetettävän videon luonteen vuoksi ei ole hyväksyttävää uudelleenlähettää kadonneita paketteja. Krasicin, Lin ja Walpoleen sanojen mukaan siitä syntyy lähetykseen odotusta ja katkoksia, sillä alkuperäinen data ”ei ehdi näytettäväksi ajoissa”. Vaikka kyseisen teoksen julkaisusta on jo kulunut noin 15 vuotta, on silti huomattavissa, että muutosta TCP:n kohdalla ei ole vielä tapahtunut. Sen soveltuvuutta suoratoiston tiedonlähetyksessä voidaan edelleen kyseenalaistaa. TCP-protokollassa katkoksen tapainen ongelmatilanne voi syntyä esimerkiksi kaksinkertaisesta ACK (Acknowledge) -viestistä. Kaksinkertainen ACK-viesti voi syntyä siitä, että jotain datapakettia ei ole vastaanottajalle saapunut (Kuva 1) tai se on saapunut vajaanaisesti tietoliikennepolulla tapahtuneen hetkittäisen katkoksen vuoksi (Karanjit 2000, 108–109). Tästä huolimatta TCP-protokollaa käytetään suoratoiston tuottamiseen, koska se takaa suoratoistossa tuotetun multimedian eheyden tallennettaessa mediapalvelimelle.



Kuva 1. Esimerkki uudelleenlähetetystä ACK-viestistä TCP-protokollassa (Nishida, J. 2003).

2.4.2 UDP suoratoistossa

UDP on tietoliikenteen lähetysprotokolla, joka on suunniteltu TCP:n lailla, datan lähettämiseen verkossa laitteelta toiselle käyttäen IP-osoitteita. Sen eroavaisuus TCP:n kanssa on sen yhteyttömyydessä, sillä virheellisiä datapaketteja ei pyritä korjaamaan tai lähettämään uudelleen. Tämä tarkoittaa sitä, että painoarvo lepää enemmän täsmällisyyden kuin tarkkuuden varassa tietoliikenteen toiminnassa. UDP on siis nopeampi vaihtoehto TCP:n rinnalle ja on soveltuvampi median liikennöimiseen. Kyseenalaistettava haittapuoli UDP:ssa on sen tietoturvassa, sillä se ei ole yhtä turvallinen kuin TCP laadunvalvonnan puutteellisuuden vuoksi (Wankel 2011, 3)

3 SUORATOISTON MEDIAPALVELIN

Mediapalvelimella tarkoitetaan laitetta tai sovellusta, jonka tehtäväksi on allokoitu multimedian jakaminen. Tässä luvussa keskitytään tarkemmin RTMP-pohjaiseen mediapalvelimeen ja sen toimintaan.

RTMP-mediapalvelin toimii avoimella suorituksella (open socket). Se tarkoittaa sitä, että normaaliin web-palvelimeen verrattuna yhteyttä ei katkaista kun data on palvelimelta käyttäjän puolesta noudettu. Yhteys pidetään auki kunnes vastaanottaja tai palvelimen ylläpitäjä itse sen katkaisevat. Tämä toimintaperiaate on optimoitu suoratoistoa varten, missä dataa voidaan lähettää jatkuvalla syötöllä (Sanders 2008, 2).

Suoratoiston lähettäjä voi olla mikä tahansa laite tai sovellus, joka RTMP:n tapauksessa tukee Flash-pohjaisia tiedostotyyppejä ja suoratoistoa. RTMP-palvelin kaappaa suoratoiston lähettäjältä ja jakaa sen eteenpäin sekä mahdollisesti arkistoi sen. RTMP-mediapalvelimella on kolme pääfunktiota median lähettämiselle verkossa: suora lähetys, tilausvideot ja simuloitu suora lähetys (Seel 2014, 3200).

3.1 Lähetysmetodit

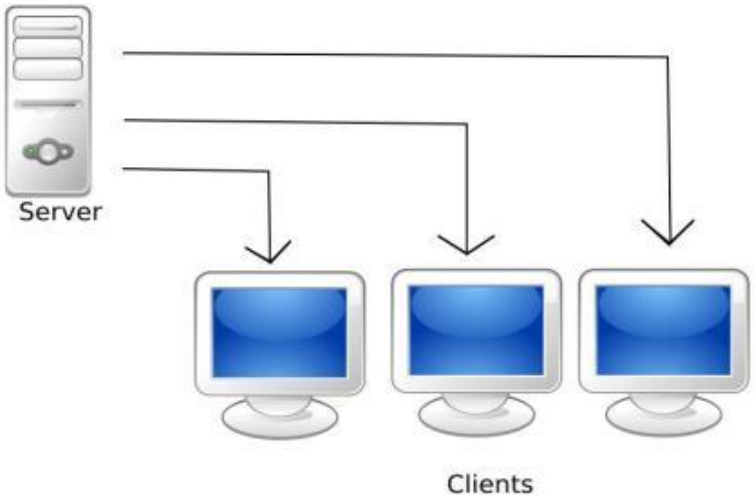
RTMP-mediapalvelimen suoratoiston lähetys voidaan jakaa kolmeen alakategoriaan. Suoratoistossa reaaliajassa esitettävää materiaalia pystytään toistamaan verkossa pienellä viivellä käyttämällä RTMP-palvelinta. Erilaiset tapahtumat kuten luennot, seminaarit, konferenssit sekä opetus- ja viihdevideot pystytään nauhoittamaan paikan päällä ja lähettämään suoratoistopalvelimelle verkossa, josta se jaetaan katsojakunnalle esitettäväksi (Seel 2014, 3200). Tilausvideot eli videos-on-demand mahdollistavat pääsyn esitallennettuun sisältöön. Yksittäinen käyttäjä kykenee pyytämään taltioidun materiaalin katseltavaksi RTMP-mediapalvelimelta ja käyttämään videonauhuri (VCR)-tyylisiä komentoja, kuten toista, tauota, kelaa eteenpäin ja taaksepäin sen katsomiseen.

”Simulated live broadcast” on termi mitä käytetään kuvastamaan, kun RTMP-mediapalvelimelle luodaan soittolistoja esitallennetusta materiaalista. Tämä eroaa aiemmin mainituista on-demand- ja suora lähetys -metodeista siten, että erillistä suoratoistosovellusta ei lähetykselle tarvita. Multimediadata lähetetään RTMP-palvelimelle erillisenä tiedostona tai vaihtoehtoisesti soittolista kootaan valmiista

materiaalista, joka löytyy jo tietokannasta. Suoratoistettaessa tätä valmista materiaalia suoraan palvelimelta, jokainen käyttäjä yhdistäessään lähetykseen näkee ja kuulee saman kohdan sisältöä. Täten ”simulated live” naamioituu suoraksi lähetykseksi, mutta onkin esinauhoitettua materiaalia. Näiden edellä mainittujen metodien avulla voidaan verkossa lähettää multimediadataa kahdella eri tavalla. Nämä lähetystavat ovat yksittäislähetys (unicast) ja ryhmälähetys (multicast).

3.2 Unicast-suoratoisto

Kun katsojan ja palvelimen välille luodaan yhteys kutsutaan sitä suoratoistoyhteyttä unicastiksi. Esimerkki tällaisesta yhteystyypistä näkyy kuvassa 2. Unicast on ennalta-asetettava asetus palvelinpuolella, mikä mahdollistaa suoratoiston jakamisen nille käyttäjille, jotka sitä erikseen pyytävät. Tämä lähetystyyppi kuvastaa VOD:tä (videos-on-demand), sillä tilausvideoita voidaan lähettää vain yksittäisesti niitä pyytäville tahoille. Unicast-suoratoiston hyviä puolia ovat muun muassa sen interaktiivisuus mediasoittimen ja palvelimen välillä ja kyky usean siirtonopeuden pakkauksenhallinnalle (multi-bit-rate streaming). Varsinaisen suoratoiston esittäminen suurelle katsojakunnalle unicast-tyypillä ei ole suvaittavaa, sillä tämä lähetystyyppi on hyvin riippuvainen verkon lähetyskapasiteetista ja palvelimen kykeneväisyydestä lähettää dataa. Mikäli verkossa lähetettävää suoratoistoa tulee katsomaan usea käyttäjä, interaktiivisuudella ei ole väliä ja/tai verkon reitittimestä löytyy multicast-ominaisuus, on multicast-suoratoisto silloin optimaalisempi vaihtoehto suoratoistettavalle medialle (Wankel 2011, 5).

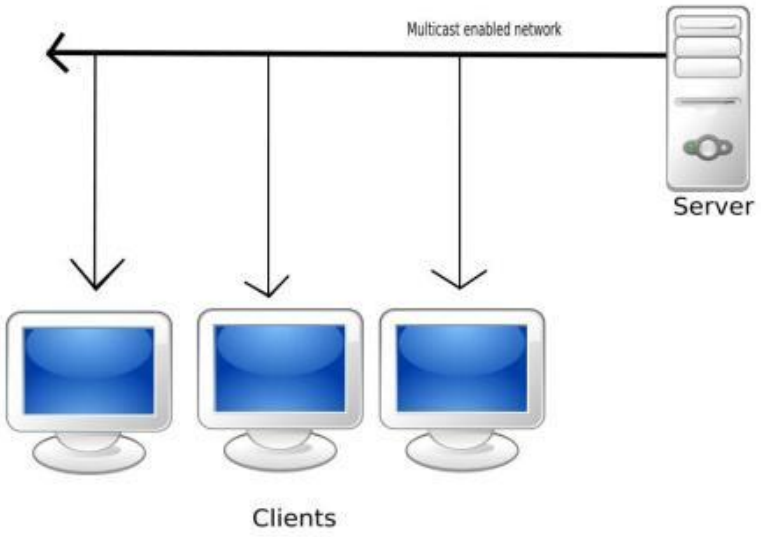


Kuva 2. Esimerkki useasta unicast-suoratoistolähetyksestä verkossa (Catalin 2009).

3.3 Multicast-suoratoisto

Toisin kuin unicast-suoratoistossa, multicast-vaihtoehto ei mahdollista interktiivista suhdetta mediasoittimen ja palvelimen välillä. Multicast esittää verkossa lähetettävän suoratoiston kaikille samanlaisena käyttäen multicast IP-osoitetta. IPv4-pohjaisessa verkossa multicast-osoitteet luokitellaan D-luokan IP-osoitteiksi. Näiden IP-osoitteiden avaruus on välillä 224.0.0.0 ja 239.255.255.255 (Microsoft 2007).

Maininnan arvoisena asiana multicastissä on se, että se vaatii vain yhden unicast-lähetyksen verran kaistaa verkolta vaikka katsojia olisi useita tuhansia (Kuva 3). Unicast-lähetyksessä puolestaan vaaditaan jokaista katsojaa varten oma osa kaistaa, mikä lisää verkkoliikennettä ja palvelimen suorituskykyyn kohdistuvaa stressiä jokaista katsojaa kohden. Multicast on nimensä mukaan optimoitu suoratoiston lähetykseen usealle katsojalle. Multicast-ominaisuutta rajoittavana tekijänä saattaa olla ”legacy”-laitteet eli vanhemman sukupolven laitteet (esim. reitittimet), jotka eivät tue multicast-ominaisuutta (Wankel 2011, 5–6).



Kuva 3. Esimerkki multicast-suoratoistosta verkossa (Catalin 2009).

4 MUITA SUORATOISTOPROTOKOLLIA

Tässä osiossa verrataan mediapalvelimelle mahdollisten protokollien ominaisuuksia. Kappaleen tarkoitus ei ole kattavasti tutkia muita datalähetysprotokollia sen tarkemmin, kuin mitä RTMP:n vertailtavuuden vuoksi todetaan oleelliseksi.

4.1 Hallinnointiprotokolla RTSP

RTSP (Real Time Streaming Protocol) on hallinnointiprotokolla, joka mahdollistaa yhden tai usean audiovisuaalisesti synkronoidun median toiston. RTSP ei itsessään kykene jatkuvaan suoratoistoon kuten RTMP, vaan se on sen sijaan tarkoitettu usean datalähetyksen hallinnoimiseen kaukosäätimen lailla verkossa käyttäjän ja mediapalvelimen välillä. Verrattuna RTMP:n toimintoihin, RTSP kykenee suoratoiston sijaan useasta lähteestä noudettavan datan esittämiseen lähes reaaliajassa. Tähän lähetykseen voi myös sisältyä esitallennettu data. RTSP:n toimintaa ei voida luokitella varsinaisesti suoratoistoksi, sillä protokolla itsessään ei kykene sitä tuottamaan. RTSP-palvelin toimii yhdessä RTP:n (Real-time Transport Protocol) ja RTCP:n (RTP Control Protocol) kanssa tuottaakseen suoratoistettavaa mediaa (Javvin Technologies Inc. 2005, 129). RTP:n tehtävä on mahdollistaa suoratoiston tyylinen datalähetys mediapalvelimelle, mutta protokolla ei omaa laadunvalvontaa (Quality of Service) tai lähetettyjen epäkunnollisten datapakettien korjausta. Tästä syystä RTP ja RTSP ovat riippuvaisia RTCP:sta. RTCP:n tehtävät ovat laadunvalvonta ja kulunvalvonta. Kulunvalvonnalla tässä yhteydessä tarkoitetaan verkkoa käyttävien osapuolien määrän valvontaa, jotta protokolla pystyy laskemaan parhaan mahdollisen lähetysnopeuden (Javvin Technologies Inc. 2005, 144–145).

4.2 Microsoft Media Server

MMS (Microsoft Media Server) on Microsoftin yksinomistuksessa oleva suoratoistoon käytettävä verkkoprotokolla. Sitä käytetään multimediadatan reaaliaikaiseen lähetykseen. MMS on otettu käyttöön instansseissa, joissa dataa täytyy käsitellä (renderöidä) ja lähettää verkossa samanaikaisesti. Toisin kuin RTMP, MMS lähettää dataa tietyllä ennaltamäärätyllä nopeudella sitä pyytäneelle lähteelle eli katsojalle.

Katsoja pystyy kuitenkin itse määrittämään kuinka korkealaatuista multimediaa hän haluaa katsoa. MMS ei ole kykeneväinen tuottamaan simuloitua suoraa lähetystä, sillä sen avulla ei voida luoda palvelinpuolen soittolistoja.

4.3 HTTP Live Streaming

Hypertext Transfer Protocol Live Streaming (joskus lyhennettynä HLS) on mekanismi datan lähetykselle web-palvelimelta web-selaimelle pyynnöstä. Tyypillisessä HTTP:n tiedonsiirron syklissä pyyntö katkaistaan kun toiminto on suoritettu loppuun, mikä ei ole suoratoiston kannalta optimaalista (Wankel 2011, 4). Applen kehittämä HLS mahdollistaa jatkuvan tiedonsyötön suoratoiston katsojalle HTTP:n kautta. Pyyntö jätetään palvelimen puolelta auki eli normaalia HTTP-sykliä ei saateta loppuun, ellei mediapalvelin tai palvelua käyttävä katsoja sitä katkaise. Multimediadata lähetetään verkossa palvelimelta katsojalle lohkoina, mutta sitä ennen katsojan täytyy ladata oikean soittolistan tunniste. Tunnisteen lukemalla kohdekone pystyy lataamaan oikean segmentin oikealla hetkellä ja täten pystyy seuraamaan suoratoistoa reaaliajassa. HLS tukee myös tallennetun (on-demand) sisällön esittämistä (Apple Inc 2015). Tällä hetkellä HLS on kilpailukykyinen suoratoistoprotokolla RTMP:n rinnalle, sillä sekin kykenee tiedonsiirtonopeuden adaptoimiseen verkon kykeneväisyyden mukaan. Huomattavana erona on protokollien käyttämät oletusportit. RTMP:n käyttämä portti 1395 vaatii erillistä huomiota esimerkiksi VPN:ssä (Virtual Private Network), kuten yrityksen sisäverkossa. Tuon portin tulee olla RTMP:aa käsittelevässä reitittimessä avoinna, jotta dataa voidaan edes vastaanottaa, kun taas HLS:n käyttämä portti 80 liikuttaa multimediaa vapaasti muun selainliikenteen kanssa. Maininnan arvoisena asiana on myös se, ettei HTTP Live Streaming (eivätkä Apple-tuotteet kuten iOS, iPhone, jne.) tue Flash-pohjaista multimediaa.

4.4 Flashin tulevaisuus ja HTML5

HTML5:stä luvataan tulevaisuudessa todella paljon. Sen mahdollisuudet toimia natiivisti monelle tiedostotyypille käyttäjän selaimessa ovat uhkana Flash-pohjaiselle medialle. HTML5 mahdollistaa tällä hetkellä yhteyden käyttäjän ja palvelimen välillä HTML- ja ei-HTML-datan vaihtamiselle, palvelinpuolesta lähtöisen (server-to-client push event) datasiirron ja RTC:n (Real Time Communication), jonka ansioista useat käyttäjät voivat

yhdistää esimerkiksi videokonferenssiin ilman ulkopuolisten pluginien tarvetta. Kaikki tämä toimii hyvänä perustana suoratoistolle (Mozilla Developer Network 2016).

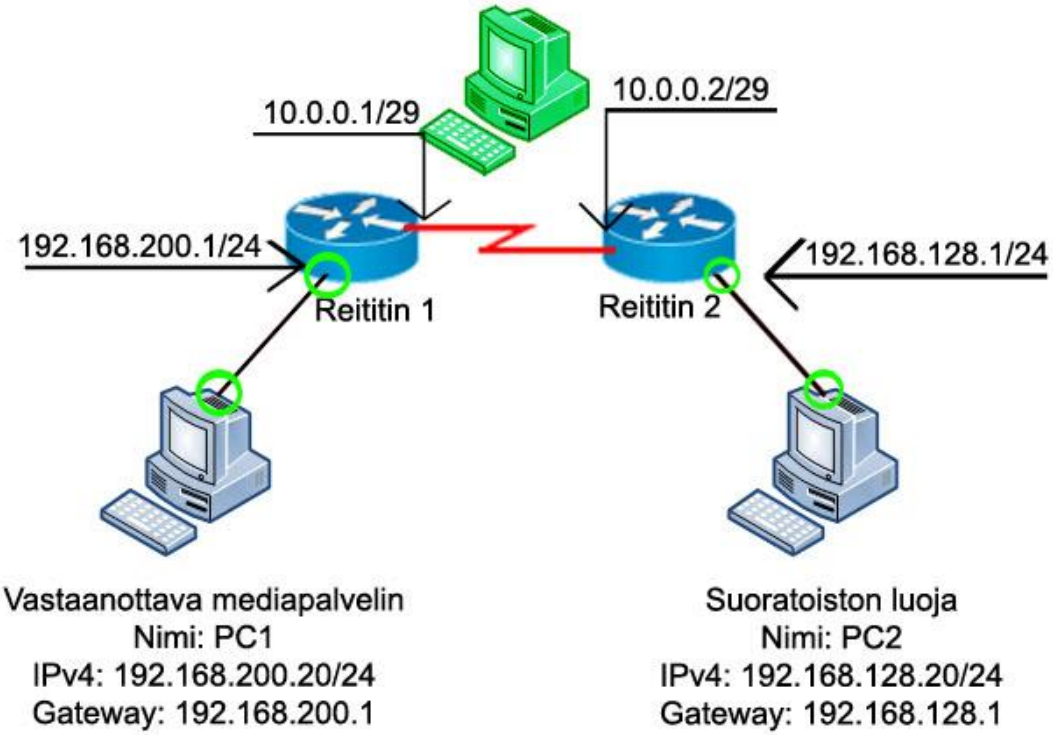
Web-selaimet eivät kykene renderöimään Flash-mediaa ilman ulkopuolista tekijää kuten Adobe Flash Player. Tämä tarkoittaa sitä, että jokaiselle selaimelle on löydyttävä yhteensopiva plugini, joka pystyy toistamaan Flash-pohjaista sisältöä. Mainittavana seikkana on myös, että tämän selain-pluginin suorituskyky on riippuvainen muun muassa sen toimittajasta ja kykyisyydestä toimia tietyllä selaimella. Tätä ongelmaa ei HTML5:ssä teoriassa olisi, sillä kaikki selaimet pystyvät käsittelemään HTML:a. HTML5 tulee vahvistamaan HLS:n kilpailukykyisyyttä RTMP:n rinnalla.

Tämä muutos HTML5:een näkyy jo erilaisten palveluntarjoajien verkkosivuilla, joista yksi suurimmista on YouTube. Tammikuussa 2015 YouTube ilmoitti pudottavansa Flash-tarpeet sen sivuilla toistettavista videoista siirtyen HTML5:een tietyillä web-selaimilla. Näihin selaimiin kuuluvat Chrome, IE 11, Safari 8 ja Firefoxin beta versiot (Richard Leider 2015). Selainpuolella tämä tarkoittaa kevyempää videontoistoa, mutta se ei välttämättä mitätöi RTMP:aa palvelintasolla. Muutos HTML5:een tulee näkymään enimmäkseen suoratoistovideoiden katsojille enemmän kuin sen tuottajille, sillä jopa YouTube vaikuttaa kyseenalaistavan RTMP:sta pois siirtymistä sisällön tuottamisessa. Näin voidaan päätellä siitä, että YouTuben yksittäisen henkilön kanavasivuilla ja luomistyökaluissa on edelleen yhdistämisohjeet YouTuben RTMP ingest -palvelimelle.

5 SUORATOISTON TIETOTURVATESTAUSYMPÄRISTÖ

Empiirisenä osana tätä työtä on VMWare Workstation 10.0:llä rakennettu virtuaalikoneympäristö, jossa suoratoistoa voidaan tuottaa sitä vastaanottavalle palvelimelle. Tuotettavaa mediaa vastaanottava kone on varustettu Wowza Streaming Engine -mediapalvelinohjelmistolla. Suoratoistoa lähettävässä koneessa on otettu käyttöön OBS- (Open Broadcast Software) sekä FMLE (Adobe Flash Media Live Encoder) -suoratoiston pakkauksenhallintaohjelmat. Näiden kahden koneen datalähetyksen väliin asetetaan kolmas osapuoli kuuntelemaan verkkoliikennettä ja mahdollisesti aiheuttamaan haittaa yhteydelle, kaappaamaan tiedonsiirtoa tai pahimmassa tapauksessa varastamaan avaintietoja kuten salasanoja. Tämä ”hyökkäävä” ulkopuolinen kone toimii Kali Linux -käyttöjärjestelmällä ja on varustettu erilaisilla murtautumistestaustyökaluilla.

Kuvassa 4 on esitetty tätä tutkielmaa varten rakennettu virtuaaliverkko, jonka sisällä suoratoistoon ja mediapalvelimeen kohdistuvia verkkohyökkäyksiä ja -uhkia pyritään simuloimaan tutkimuksellisin motiivein. Tunkeilijan rooliin astuvan vihreän ”KaliKoneen” tarkoituksena on toimia verkon sisäisenä ja mahdollisesti ulkopuolisena uhkana molemmille kohdekoneille. Kalikoneen hyökkäysväylät on merkitty kuvassa vihreillä ympyröillä. Kohdekoneet tässä ympäristössä ovat mediapalvelimen kone ”PC1” ja suoratoistoa tuottava kone ”PC2”. Tutkimuksen tarkoituksena ei ole tutkia sitä miten tunkeilija pääsee murtautumaan koneiden käyttämiin sisäverkkoihin, vaan syventävästi tutkia RTMP:aan kohdistuvia tietoturvanäkökulmia.



Kuva 4. Tutkielmaa varten luodun virtuaalikoneverkon pohjapiirrustus.

5.1 Wireshark

Wireshark on tietoliikenneverkon protokolla-analysaattori. Sitä käytetään datapakettien reaaliaikaiseen taltiointiin ja esittämiseen. Wiresharkilla on useita käyttötarkoituksia: sitä voidaan käyttää esimerkiksi yhteysongelmien vianmääritykseen, verkkoprotokollien paloitteluun, verkkoliikenteen tarkastukseen sekä valitettavasti myös ilkivaltaan. Wireshark on avoimen lähdekoodin ohjelma, mikä tarkoittaa sitä, että kuka tahansa lisenssinhaltija pystyy ohjelmaa lukemaan, muokkaamaan ja kehittämään (Lamping ym. 2014, 1–2). Tässä tutkimuksessa Wiresharkin avulla pyritään haravoimaan informaatiota verkossa kulkevasta mediasta mediapalvelimen ja suoratoistokoneen välillä.

5.2 Pakkauksenhallintasovellukset

Pakkauksenhallintasovelluksella tarkoitetaan sellaista sovellusta, joka pystyy yhdistämään eri lähteistä taltioitua mediaa, kuten audiota ja videota, yhdeksi tiedostoksi. Sovellus pakkaa mediat ennalta-asetettuun muotoon erillaisilla koodekeilla käyttämällä

sovellusta hallitsevan laitteen prosessointiresursseja. OBS eli Open Broadcast Software (https://obsproject.com/) on ilmainen avoimen lähdekoodin pakkauksenhallintasovellus, joka on tarkoitettu videon tallentamiseen ja muun muassa RTMP-suoratoistoon. Se toimii median pakkauksenhallintaohjelmana ja -lähettäjänä verkossa mediapalvelimelle. Tutkimuksen kannalta merkittävää OBS:ssä on, että se ei sisällä minkäänlaisia salausmetodeja. Sovellus on kuitenkin helppokäyttöinen sekä yksinkertainen ja se on saanut suoratoistomaailmassa hyvin suuren suosion. Sen avulla voidaan suoratoistaa mediaa palvelun tarjoajien kuten YouTuben, Twitch.tv:n ja DailyMotionin mediapalvelimille. OBS:een ei kuitenkaan ole mahdollista saada RTMP-autentikaatiomahdollisuutta, jossa käyttäjä kirjoittaa ennen jokaista suoratoiston yhdistämistä käyttäjänimensä ja salasanansa. Tietoturvallisen tunnistautumisen testausta varten OBS:n lisäksi tutkimuksessa käytetään myös Adoben omaa pakkauksenhallintasovellusta Flash Media Live Encoderia (FMLE, http://www.adobe.com/products/flash-media-encoder.html). Toisin kuin OBS, FMLE ei ole avoimeen lähdekoodiin perustuva ja se on maksullinen. Siitä on kuitenkin olemassa kokeiluversio, jota tässä tutkimuksessa käytetään.

5.3 Wowza Streaming Engine

Suoratoiston lähettämiselle tarvitaan vastaanottava taho eli mediapalvelin. RTMP-suoratoiston vastaanottoa varten valittiin tähän tutkimukseen Wowza Streaming Engine -ohjelmisto, josta on saatavilla kokeiluversio. Se on monipuolinen live- ja tilausvideoiden käsittelysovellus sekä helppokäyttöinen sen graafisen käyttöliittymän ansiosta. Siihen pystytään myös implementoimaan suoratoistojen kannalta tärkeitä tietoturvallisia ominaisuuksia, kuten käyttäjäkohtaisia suoratoistoavaimia.

6 SUORATOISTON TIETOTURVA

Kuten muullakin verkkoliikenteellä, on suoratoistolla myös omat tietoturvariskinsä. Jatkuva tiedonsiirto verkossa luo tarpeen tietoturvallisille datalähetyksille. Suoratoiston tietoturvassa on hyvä ajatella kaiken lähetettävän tiedon olevan vastaanotettavissa missä tahansa. Oli tiedon kaappaus tahallista tai tahatonta, kaikki suoratoistettaessa välittyvä informaatio voi olla altista ulkopuolisille tahoille ja tätä kautta väärinkäytölle. Miten tätä informaatiota suojataan ja miten se voidaan kaapata, on riippuvaista sen lähetystavasta. Tämän kappaleen tarkoituksena on käsitellä suoratoiston tietoturvaa keskittyen RTMP:aan. Tämän tutkimuksen tietoturvalliset toimenpiteet ovat rajattu RTMP:n ympärille.

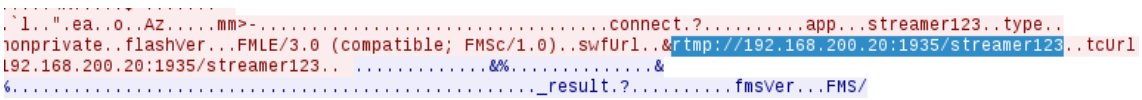
6.1 Suoratoistajan tunnistautuminen palvelimelle

Suoratoistojen vastaanottamiseen tarkoitettujen palvelinten on tunnistettava niille pyrkivät yhteydet. Käyttäjänä suoratoiston tuottajan on tavalla tai toisella rekisteröidyttävä, jotta palvelimet voivat vastaanottaa tuotettua mediasisältöä tunnistettavalta taholta. Esimerkiksi YouTuben ja Twitch.tv:n palveluiden käyttöön on vaatimus rekisteröityä palveluntarjoajan verkkosivuilla käyttäjänimellä, salasanalla ja sähköpostiosoitteella. Rekisteröidyttyä näillä verkkosivuilla käyttäjillä on mahdollisuus yhdistää palveluntarjoajan ingest-palvelimille pakkauksenhallintaohjelmalla ja täten julkaista suoratoistomateriaalia. Nämä ingest-palvelimet tunnistetaan niille ainutlaatuisella Uniform Resource Locatorilla (URL). Nämä URL:it eli verkko-osoitteet ovat yleensä julkisia. Suoratoiston tuottajan tunnistusmetodi on ”stream key” eli suoratoistoavain, jota voidaan käyttää palvelimelle tunnistauduttaessa. Tämä mahdollistaa suoratoiston näkyvyyden oikeassa osoitteessa. Suoratoistoavaimet ovat käyttäjäkohtaisia, ja niiden kanssa on muistettava kolme tärkeää asiaa palveluntarjoajana:

1. Suoratoistoavaimen luomista ei kannata jättää käyttäjäpuolella asetettavaksi. Tietoturvallisista syistä sen asettaminen on tultava palveluntarjoajalta. Mikäli mahdollisuus avaimen luontiin jätetään käyttäjälle, saattavat joidenkin käyttäjien avaimet olla pahimmassa tapauksessa identtiset tai niiden yksinkertaisuuden vuoksi helposti varastettavissa.

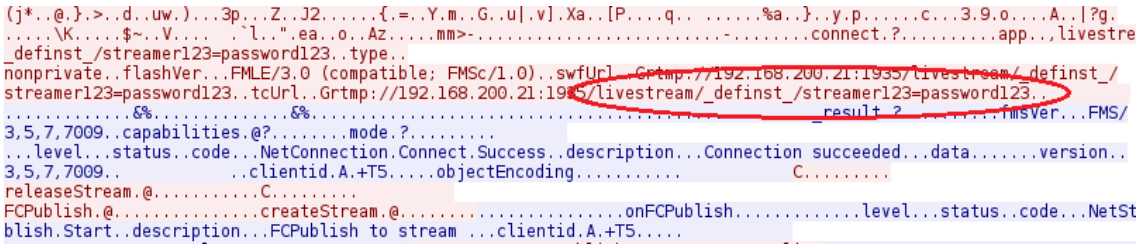
2. Käyttäjälle on tehtävä ilmiselväksi suoratoistoavaimen herkkäluonteisuus. Hänelle on kerrottava uhat ja riskit sen levittämisessä.
3. Suoratoiston väärinkäytön estämiseksi on tärkeää antaa käyttäjälle mahdollisuus resetoida tai uudelleengeneroida hänen suoratoistoavaimensa. Mikäli käyttäjä epäilee, että se on saatettu varastaa tai se on vuotanut julkisuuteen, on käyttäjälle hyvä tarjota tällainen mahdollisuus.

Suoratoisto ilman tunnistusavainta on kuin ilman lukkoa oleva ovi, jonka läpi kuka tahansa voi kulkea. Virtuaalikoneistoa ja -ympäristöä apuna käyttäen on mahdollista tutkia, miten hyökkääjä voi saada selville suoratoistoa lähettävän tietokoneen IP-osoiteen ja suoratoistopolun Wireshark-ohjelmalla. Tätä toimenpidettä kutsutaan URL-sniffingiksi. Verkkoliikennettä tässä tapauksessa haravoidaan mediapalvelimen verkkoympäristössä, mutta sama informaatio saadaan selville sekä lähettäjän että vastaanottajan lähiverkossa, sillä paketit ovat vastaanottajan ja lähettäjän päädyssä identtiset. Huomiokohtana yleisestä tietoturvasta on se, että hyökkääjä on simuloidussa tapauksessamme jo onnistunut tunkeutumaan mediapalvelimen lähiverkkoon. Esimerkissämme mediapalvelimen ja suoratoiston tuottajan yhteyttä ei ole suojattu minkäänlaisilla funktioilla. Kokonaan suojaamaton suoratoisto RTMP:lla tarkoittaa altistumista IP-osoitteen paljastumiselle ja suoratoiston kaappaamiselle. Wireshark-pakettitarkastelutyökalulla on mahdollista saada suoratoiston URL-osoite ja siihen liittyvät avaintiedot talteen suoratoiston aloitus- eli ”handshake”-hetkellä. Käytännössä tämä on se hetki, jolloin suoratoiston tuottaja aloittaa mediasisällön lähettämisen ingest-palvelimelle. Suoratoiston kaappaamisella tarkoitetaan tässä yhteydessä sitä, kun hyökkääjä on tietoinen kuvassa 5 näkyvästä suojaamattomasta suoratoisto-URL:sta ja on täysin kykeneväinen nyt itse avaamaan suoratoiston omalta koneeltaan käyttäen haluamaansa pakkauksenhallintaohjelmaa ja sisältöä. Kuvassa 5 näkyvä ”rtmp”-alkuinen URL koostuu mediapalvelimen osoiteesta ja käyttäjätiedoista, joilla suoratoistaja ottaa yhteyden mediapalvelimeen ja täten aloittaen suoratoiston. Käyttäjätiedot näkyvät TCP-virrassa ”clear text” -muodossa ilman minkäänlaista salausta.



Kuva 5. Wiresharkilla kaapatussa informaatiossa näkyvä RTMP URL.

Tässä tilanteessa lisätään kyseiseen suoratoistoon mediapalvelimen puolelta kevyt yhteyksien tunnistusmetodi. Käyttäjänimi- ja salasanayhdistelmän tarkistuksen avulla voidaan yksittäisiä käyttäjiä jäljittää ja tunnistaa täten lisäten tietoturvallisuutta. Lisätään URL-parametrikäyttäjä ”streamer123”, joka tässä esimerkissä on PC2:n (kuva 4) käyttäjätili ja asetetaan hänelle salasana ”password123”. Tämä vaihtoehto on tarkoitettu sellaisille pakkauksenhallintaohjelmille, joissa ei ole omaa käyttäjä- ja salasananasetusmahdollisuutta. Valitettavaa on, että tämä ei ole riittävä toimenpide täyden tietoturvan varmistamiseksi, kuten kuvassa 6 näkyy. Wiresharkilla kaapatun verkkoliikenteen paketeissa edelleen näkyy ”clear text”-muodossa palvelimen ja PC2:n IP:t sekä toimenpiteidemme vuoksi käyttäjänimi ja salasana. Hyökkääjä voi käyttää hyväksi tätä URL-kokonaisuutta ja tehdä suoratoistolle ilkivaltaa.



Kuva 6. Wiresharkilla TCP-virrasta kaapattu käyttäjänimi ja salasana.

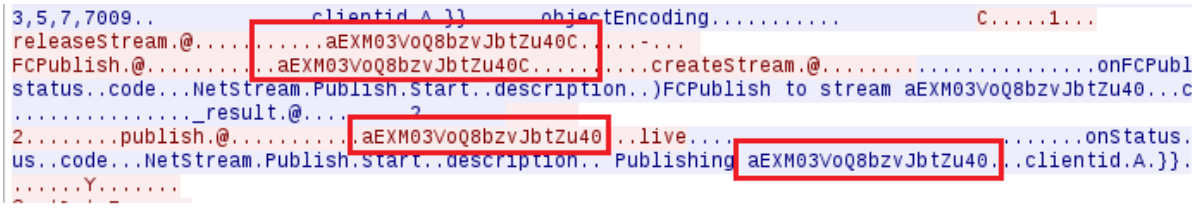
6.2 Mainittavaa IP-osoitteista

IP-osoite on jokaisen käyttäjän henkilökohtainen tunnus informaation valtaväylällä. Tavalliselle käyttäjälle IP-osoitteen välittyminen tai julkistuminen ei ole todennäköisesti suuri asia, mutta suoratoistossa tämä saattaa olla toisin. Suoratoiston tuottaja voi olla kuka tahansa. Se voi olla yksittäinen henkilö, ryhmä, tekoäly tai jopa kokonainen kollektiivi ryhmiä. Kun kyseessä on yksittäinen henkilö tai ryhmä, jotka viihdyttävät katsojakuntaa omilla persoonillaan, voivat aiemmin tässä tutkimuksessa mainitut tietoturvariskit olla suuria. Mainittavana tietoturvauhkana on tällaisten viihdemediaa tuottavien tahojen IP-osoitteiden leviäminen. Kun kyseessä on suoratoistettava reaaliaikainen media, missä muutos asioihin ja tapahtumiin näkyy lähes välittömästi, saa nettikiusaaminen uuden puolen. Suoratoistajan IP-osoitteen julkistuminen avaa suoratoistajan kohteeksi ilkivallalle, jota lietsoo ilkivallantekijän mahdollisuus nähdä reaaliajassa suoratoiston tuottajalle aiheutuva haitta. Esimerkiksi suoratoiston sabotointi on mahdollista IP-osoitteen vuotamisen myötä. DDoS (Distributed Denial of Service) eli palvelunestohyökkäykset ovat todella yleisiä suoratoiston maailmassa. Tällaisessa

tapauksessa hyökkääjä käyttää useaa murrettua järjestelmää tukkimaan verkkoyhteyden yhteen tiedettyyn kohteeseen eli tässä tapauksessa suoratoistajaan. Palveluntarjoajana on tärkeää varmistaa suoratoiston tuottajan IP-osoitteen salattavuus.

6.3 Tunnistautuminen suoratoistoavaimella

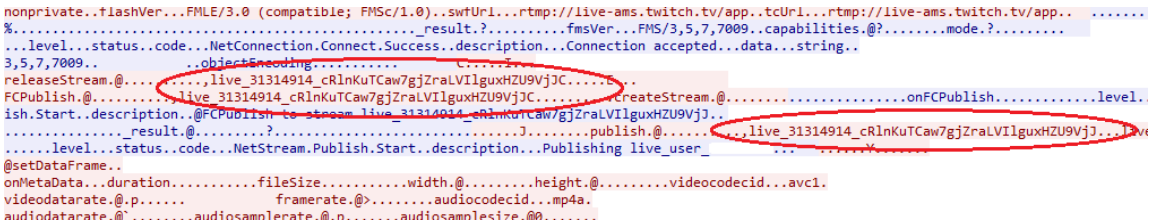
Yleisimmät palveluntarjoajat vaativat jokaista suoratoistajaa kohtaan tunnistautumista suoratoistoavaimella. Se on pieni tietoturva-askel eteenpäin aikaisemmin käytetystä käyttäjänimi- ja salasanayhdistelmästä. Suoratoistoavain mahdollistaa kahden osapuolen kanssakäymisen ilman, että heidän tulisi tuntea tai tunnistautua toisilleen millään tavalla. Suoratoistoavain on usein saatavilla palveluntarjoajan verkkosivuilta käyttäjänimeä ja salasanaa vastaan. Valitettavasti näin kevyt salausmetodi RTMP:ssa ei myöskään ole tarpeeksi tietoturvallinen ratkaisu itsessään. Vaikka suoratoiston yhteydessä käytettäisiin monimutkaista ja ennalta arvaamatonta avainta, esimerkiksi ”aEXM03VoQ8bzvJbtZu40”, mitätöityy avaimen monimutkaisuus, jos hyökkääjä tietää mitä etsiä. Lähtökohtaisesti on turvallisempaa aina olettaa, että hyökkääjä tietää mitä hän tekee. Kuvassa 7 näkyy esimerkki edellä mainitusta suoratoistoavaimesta, joka toistuu TCP-virran sisällä useaan kertaan.



Kuva 7. Wiresharkilla kaapattu suoratoiston suoratoistoavain TCP-virrassa.

Ehkä pahinta nykypäivänä on, että lähes isoimmat palveluntarjoajat kuten YouTube ja Twitch.tv käyttävät kuvassa 7 esitettävää suoratoistoavaimen välitysmetodia. Voiko tälle tietoturvallisen käytännön laiminlyönnille olla syynä silkka välinpitämättömyys? Ainakin palveluntarjoajien tietoturvallinen lähestymistapa asian suhteen on melko tekopyhää. Palveluntarjoajat kertovat kyllä verkkosivuillaan suoratoistoavaimen salaamisen tärkeyden, mutta eivät vaadi suoratoistoyhteyksien tietoturvallisuudesta juuri mitään. Kuvassa 8 näkyy pakettikaappaus suoratoiston tuottajan ja Twitch.tv:n ingest-palvelimen välisestä kommunkaatiosta. Yhteyttä ja sillä välitettävää informaatiota ei salata juuri mitenkään. Vaatimattomuus tietoturvassa saattaa tosin johtua siitä syystä, että

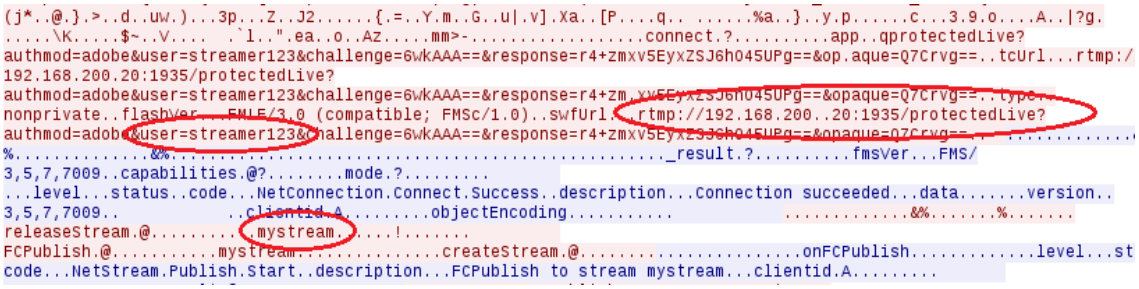
suoratoistettavan median alusta (platform) kärsisi ”ease-of-access”:in etulyöntiaseman menetyksen, jos RTMP:n tilalle implementoitaisiin esimerkiksi RTMPE (RTMP Encryption) tai RTMPS (RTMP over Secure Socket Layer). RTMP:lla lähetettävää mediaa pidetään niin helposti tuotettavana, että mikäli YouTube tai Twitch.tv vaihtaisivat protokollakohtaisia tietoturvakäytäntöjään nyt, saattaisivat heidän tarjoamansa palvelut pudota suosiosta. Vaikeuttamalla tietoturvakäytäntöjään palveluntarjoajien tilalle saattaisi nousta suosioon jokin muu helpompaan suoratoistokäytäntöön keskittyvä palveluntarjoaja. Voi olla, että YouTube ja Twitch.tv pelkäävät juuri tätä, eivätkä siksi tiukenna tietoturvakäytäntöjään tai uskalla ottaa ensimmäistä askelta tietoturvallisempaan lähestymistapaan suoratoistossa.



Kuva 8. Wiresharkilla kaapattu Twitch.tv:n ingest-palvelimelle lähtevä suoratoistoavain.

6.4 Suoratoiston salaus

Tutkimuksen edetessä kävi ilmi, että suoratoiston tuottajan ja mediapalvelimen välinen yhteys on lopulta suhteellisen helposti ja pienellä vaivalla salattavissa. Kuvassa 9 on esitetty mediapalvelimen ja suoratoistoa tuottavan käyttäjän välinen kommunikaatio. Tässä tapauksessa salasanaa ei näy pakettianalysaattorissa ainakaan puhtaan tekstin muodossa. Tämä salaus voidaan suorittaa mikäli mediapalvelimessa on mahdollisuus asettaa vaatimus turvattuun käyttäjätietojen tunnistamiseen ennen varsinaista suoratoiston aloittamista. Tämän käytännön tavoin voidaan välttyä suoratoiston kaappaukselta ainakin URL-sniffingin kohdalla. Käyttäjätietojen vaihdossa pystytään siis pitämään salaisena mediapalvelimen ja käyttäjän pakkauksenhallintaohjelman kesken välittyvä salasana. Käyttäjänimi, suoratoiston nimi, suoratoiston tuottajan IP-osoite ja kohde-URL välittyvät silti vielä RTMP-suoratoiston aloitushetkellä. Ne ollaan merkitty kuvassa 9 punaisilla ympyröillä.



Kuva 9. Salauksen jälkeen herkkäluonteista tietoa välittyy salasanaa lukuunottamatta.

Ainoa esto pakkauksenhallintaohjelman käyttäjätunnuskäytännön yleistymiselle on, että se on riippuvainen mediapalvelimen asetuksista salasanan vaatimisessa ja pakkauksenhallintaohjelman kyvystä lähettää RTMPauth-komento mediapalvelimelle. Jos edes toista aikaisemmista ei ole implementoitu käytettäviin sovelluksiin, ei suoratoistoa voida tällä tavoin salata. RTMPauth-komennon yleistyminen pakkauksenhallintasovelluksissa parantaisi herkkäluonteisen tiedon turvaamista varastamiselta ja väärinkäytöltä.

6.5 Yhteenveto suoratoiston tietoturvasta

Käyttäjäkohtainen suoratoistoavain on siis mahdollista varastaa hyvinkin helposti hyökkäävänä osapuolena. Tutkimuksessa kävi kuitenkin ilmi, että suoratoisto ja sen herkkäluonteinen informaatio on myös suhteellisen helppo suojata. Tietomurtoon tarvitaan oikea paikka ja oikea aika. Oikea paikka tässä tapauksessa tarkoittaa asetelmaa, jossa hyökkääjä on kykeneväinen kuuntelemaan verkkoliikennettä kahden laitteen välillä ilman häirintää (kuva 4) ja oikea aika viittaa suoratoiston aloitushetkeen RTMP-handshaken tapahtuessa laitteiden välillä tunnistautumiseen.

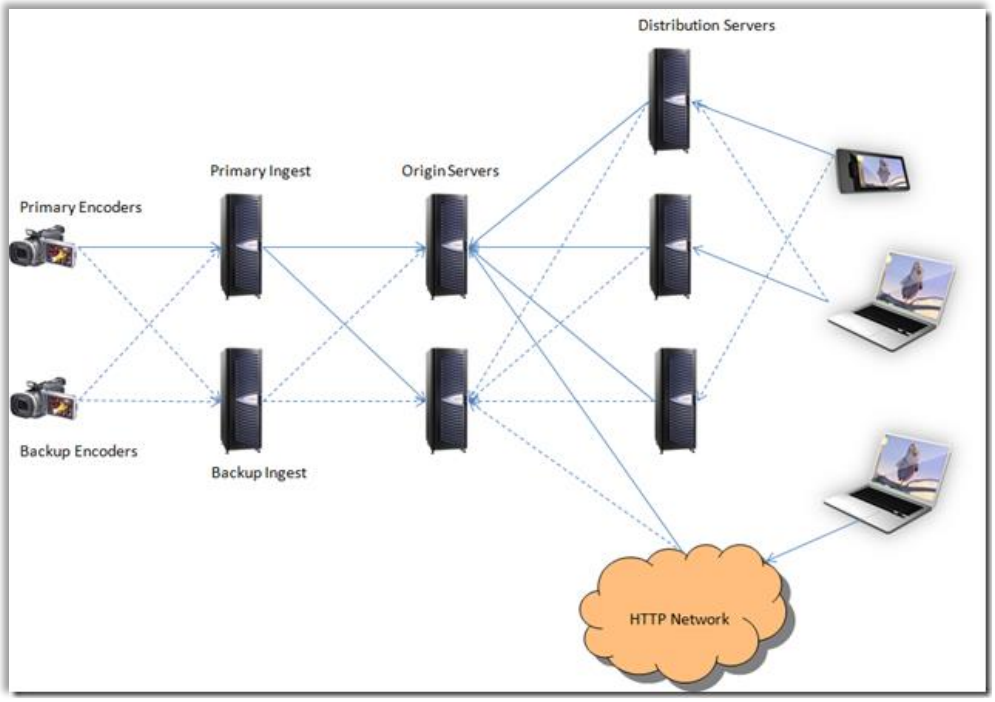
Saattaa kuitenkin olla, että pelkkä suoratoistoavain on tarpeeksi suojaamaan suoratoistoa kaappaukselta, sillä vastaus löytyy siihen liittyvistä tietoturvallisista käytännöistä. Aiemmin tässä tutkielmassa mainittiin, miten suoratoistoavaimen resetointi tai uudelleengenerointi on hyvä tapa välttyä väärinkäytöltä. Mitä jos olisi mahdollista täten luoda jokaiselle suoratoistolle tai suoratoistokerralle oma ainutlaatuinen kertakäyttöavain? Jokaisella kerralla, kun suoratoisto oltaisiin käyttäjän puolelta aloittamassa, voitaisiin tuo uusi suoratoistoavain hakea palveluntarjoajan verkkosivuilta käyttäjänimeä ja salasanaa vastaan HTTP Secure -yhteydellä. Tähänkin käytäntöön olisi vielä lisäksi helppo implementoida verkkosivuilla käytettäväksi niin kutsuttu ”two-factor

authentication” eli kaksivaiheinen tunnistautuminen esimerkiksi mobiililaitteella. Mediapalvelimen asetuksista tulisi myös löytyä vaihtoehto muiden suoratoistojen aloittamisen estämiselle samalla tunnuksella siltä varalta, että suoratoistoavainta yritetään väärinkäyttää, kun aito ja jo tunnistautunut käyttäjä on luomassa suoratoistosisältöä. Välittömästi käyttäjän lopetettua suoratoiston hänen suoratoistoavaimensa uudistettaisiin palveluntarjoajan verkkosivuilla, mistä sen voisi seuraavaa kertaa varten noutaa. Oletuksena samalla suoratoistoavaimella alkava suoratoisto aikakatkaiseen aikaisemman suoratoiston ja aloittaa uuden. Tämä käytäntö on suunniteltu ennaltaehkäisemään huonoa yhteyttä mediapalvelimen ja suoratoiston tuottajan välillä varmistaen, että mahdollisimman vähän reaaliaikaista mediaa hukataan tietoliikenteessä. Kuten tässä tapauksessa käy ilmi, käytäntöä voidaan käyttää myös ilkivaltaan.

7 MEDIAPALVELIMEN TIETOTURVA

Isojen palveluntarjoajien kuten Twitch.tv:n ja YouTuben mediapalvelimet ovat usein ne julkiset palvelimet, joista suoratoistoa ja tilausvideoita voidaan hakea ja seurata. Multimedia noudetaan CDN (Content Delivery Network) -datapankkipalvelimilta suurille yleisöille katsottavaksi (Kuva 10). Katsojat luovat yhteyden suoratoistettavaan mediaan esimerkiksi verkkosivulla olevasta linkistä jakelupalvelimella (Distribution Server), johon media noudetaan lähdepalvelimelta (Origin Server). Kuvassa 10 vasemmalta lähetetään suoratoistoa tuottavan tahon puolesta pakkauksenhallintasovelluksella multimediaa ensisijaisille ingest-palvelimille, joissa se käsitellään lähetettäväksi yleisölle. Tämän käytännön tarkoitus on helpottaa katsomiskokemusta muuntamalla tämän tutkimuksen tapauksessa vastaanotettava Flash-pohjainen RTMP-suoratoisto käyttäjille katsottavaksi suoraan HTML5:ssä tai yksinkertaisen selaimen lisäosan avulla. Katsojalla ei ole tarvetta avata yksittäisiä URL:eja monimutkaisessa sovelluksessa katsoakseen julkisia multicast-suoratoistoja, vaan palveluntarjoaja automatisoi lähes kaiken ja tarjoaa katsojalle yksinkertaistetun kanavalistan.

Ylläpitämällä suoratoistosivustoa, jossa katsojat voivat koska tahansa käydä seuraamassa ilmaista viihdettä, on CDN:ille mahdollista sisällyttää mainoksia jokaiseen sivustolla näytettävään suoratoistoon. CDN:ien palveluita käyttävät suoratoistomediaa jakavat palveluntarjoajat kuten YouTube ja Twitch.tv. CDN:t toimivat välimuistipalvelimina niiden asiakkaille ja suoratoiston tapauksessa ne toimivat myös yleisöjen kohdistajina. CDN:n tarkoitus on toimia välikätenä auttamalla katsojia löytämään heille mieluinen suoratoisto ja mainostajia löytämään heidän tuotteillensa kohdistuva yleisö riippuen maantieteellisestä sijainnista. Tämä tarkoittaa käytännössä sitä, että esimerkiksi jokaisesta suoratoiston avaamisesta näytetään yksi mainos. Palveluntarjoajat haalivat isoja summia tällaisista mainostussopimuksista. Twitch.tv on hyvä esimerkki siitä, kuinka suuria katsojamääriä CDN:illä tuettulla suoratoistosivustolla voi vuodessa olla. Twitch.tv -sivusto keskittyy videopelien ja luovien taiteiden suoratoistojen esittämiseen suurille katsojamäärille. Vuonna 2015 Twitch.tv:n arvioitu suoratoistokäyttäjien eli yksittäisten suoratoistoa tuottavien käyttäjien määrä oli 1,7 miljoonaa ja keskimääräinen kävijämäärä oli 550 000. Kokonaismäärä suoratoistetulle medialle oli 241 441 823 059 minuuttia. Mikäli tämä minuuttimäärä pyöristettäisiin helpommin luettavaan muotoon, olisi se noin 459 000 vuotta videomateriaalia yhtäjaksoisesti (Twitch 2016, 1. luku).



Kuva 10. Esimerkki Microsoft Smooth Streaming skaalautuvasta palvelinratkaisusta (Zhang, S. 2009).

7.1 Mainostulot ja -esto

Mainoksia löytyy internetin jokaisesta kolkasta. Ne ovat käyttäjien ja palveluntarjoajie yksi mainittavimmista tulomuodoista. Web-selaimille kuten Google Chrome ja Mozilla Firefox on saatavilla erilaisia lisäosia, joiden tehtävänä on estää mainosten esiintyminen www-sivulla. Nämä lisäosat ja niiden toiminta ovat usein hyvin suuren kiistelyn kohteena siitä syystä, että hyvin monelle käyttäjälle mainoksista muodostuva tulo on tärkeää. Suoratoistetussa mediassa on monella sivustolla palveluntarjoajan mahdollistamana kyky näyttää mainoksia katsojakunnalle. Se kuinka usein mainoksia näytetään suoratoiston aikana ja kuinka paljon siitä käyttäjälle maksetaan riippuu palveluntarjoajasta ja itse käyttäjästä. Mikäli nämä suoratoiston yhteydessä esitettävät mainokset poistettaisiin, siitä kärsisi suurimmaksi osin yksittäinen käyttäjä eli suoratoiston tuottaja. Jos kuvitellaan, että käytännön esimerkissä A suoratoiston tuottajalla on reaaliajassa 2000 katsojaa kun hän aloittaa mainoksen näyttämisen.

Mainos näkyy jokaikiselle kahdelletuhannelle katsojalle, sillä yksikään näistä katsojista ei käytä minkäänlaista mainosesto- tai mainosväistelytekniikkaa. Sovitaan, että tässä esimerkissä käyttäjälle muodostuu jokaista mainoksenkatsojaa kohden 0,015€ tuottaen täten käyttäjälle lopullisen summan, joka on 30€ kaikista katsojista tätä kyseistä mainosta kohden. Seuraavaksi esimerkissä B suoratoistajalla on 3000 katsojaa, mutta tällä kertaa puolilla katsojista on käytössä jotain, joka estää mainokset näkymästä www-sivulla. Täten kun mainos näytetään 1500 henkilölle ja jokaista kohden maksetaan 0,015€ kuten aiemmin tulee loppusummaksi 22,50€. Vaikka nämä summat eivät ole suuria, näkyy niissä selvästi ero ja jälki usean mainoksen jälkeen.

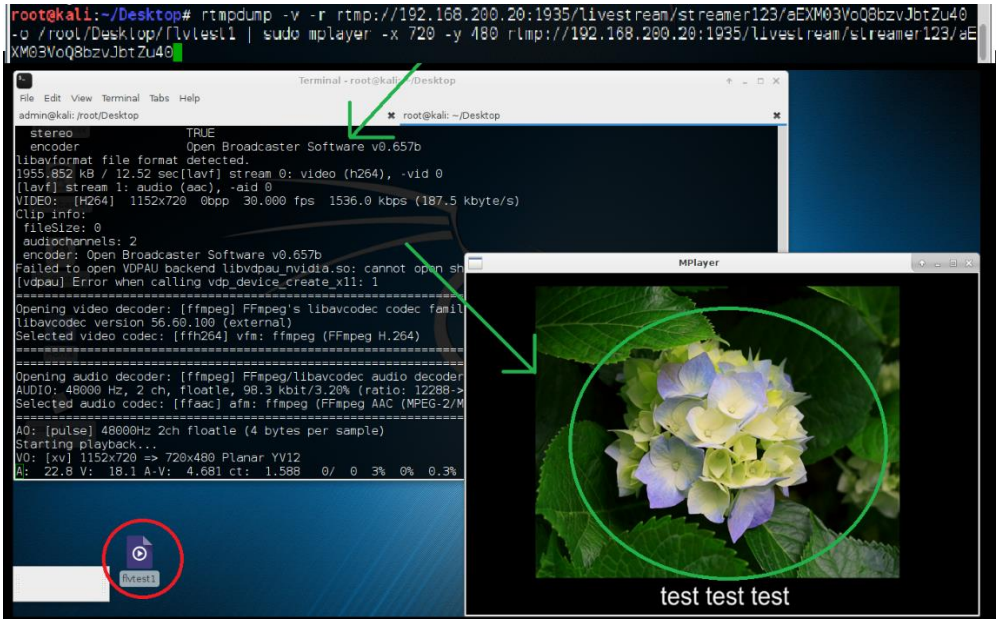
7.2 Mainosväistely ja RTMPDump

Ilkikurinen katsoja pystyy väistämään mainoksia myös muullakin tavalla kuin selainten lisäosilla. RTMP-suoratoisto on mahdollista kaapata kesken sen matkan sovellustasolle RTMPDump-sovelluksen avulla. Verkkoliikennettä tarkastelemalla etsitään katsojan selaimessa olevan mediasoittimen ja palvelimen välillä tehdyn ”handshaken” aikana esiintyvä URL suoratoistoon (Kuva 5). Hyökkääjä kykenee kaappaamaan ja jopa tallentamaan suoratoiston kesken sen matkan selaimelle suoraa lähdedatasta. Tällä toimenpiteellä väistetään kokonaan www-sivujen tietoliikenne ja esimerkiksi aiemmin mainitut mainokset.

RTMPDump:iin keskittyvässä osassa tutkimusta kuunneltiin jälleen kahden koneen vuorovaikutusta aktiivisen suoratoiston aikana. Suoratoiston tuottaja PC2 lähettää tässä esimerkissä dataa verkossa mediapalvelimelle PC1. Hyökkääjä, joka on jälleen sijoitettu palvelimen päätyyn, kuuntelee aktiivista verkkoliikennettä ja nappaa Wireshark-ohjelmalla ”handshake”-tapahtuman suoratoiston alkaessa. Tästä tapahtumasta käy ilmi muun muassa käyttäjän suoratoistoavain (Kuva 7). RTMPDump-sovellusta ja haravoitua informaatiota käyttämällä hyökkääjä voi nyt aloittaa nauhoituksen ja suoratoiston seuraamisen seuraavanlaisella komennolla:

rtmpdump -v -r
rtmp://192.168.200.20:1935/livestream/streamer123/aEXM03VoQ8bzJbtZu40
-o /root/Desktop/flvtest1 | sudo mplayer -x 720 -y 480
rtmp://192.168.200.20:1935/livestream/streamer123/aEXM03VoQ8bzJbtZu40

Tämän komennon jälkeen hyökkääjän eteen aukeaa kuvassa 11 esiintyvä näkymä, jossa suoratoistettavaa mediaa voidaan seurata mediasoittimella samalla kun kaikki data tallennetaan paikalliselle kovalevylle. Tätä mediadataa voidaan jälkeenpäin hyökkääjän osalta käyttää esimerkiksi kyseisen käyttäjän imitoimiseen, jossain muussa sosiaalisessa mediassa.



Kuva 11. Hyökkäävän koneen näkymä, kun RTMPDump-komento on suoritettu.

Mainittavaa RTMPDumpin ja URL-sniffingin kannalta on se, että näissä ei välity esimerkiksi mainoksia katsojalle. Suoratoistoa voidaan siis seurata reaaliajassa ilman HTML-liitäntöjä. Tämä tarkoittaa siis käytännössä sitä, että kaikkia suoratoistoja, joita toistetaan suojaamatta www-sivulla, voidaan altistaa URL-sniffingille ja RTMPDumpin toiminnoille. Mikäli palveluntarjoajien sivustoilla on käytössä Flash-pohjainen mediasoitin, jonka välityksellä katsojalle lähetetään data RTMP:na, on hän täysin kykeneväinen tallentamaan lähdemateriaalin omalle koneelleen.

8 YHTEENVETO

Sosiaalisen median ja viihdeteknologian kehittymisen myötä internetin välityksellä tarjottavaa mediasisältöä nähdään jatkossakin todella paljon. Suoratoistojen luominen ja seuraaminen ovat kasvava trendi, joka herättää todella paljon huomiota. Yksittäisenä henkilönä ja etenkin palveluntarjoajana on tärkeä ymmärtää, kuten muussakin sosiaalisessa mediassa, suoratoistoon liittyvät riskit ja vaarat.

Tämän tutkimuksen tarkoitus oli käsitellä Flash-pohjaiseen RTMP-suoratoistoon liittyviä uhkia, huomiokohtia ja mahdollisia korjaustoimintoja ja -käytäntöjä. Tutkimus käsitteli suoratoiston informaationvälityksessä lähetettävää ja vastaanotettavaa dataa ja analysoi sitä. Tutkimustuloksena saatiin selville usea tietoturvauhka sekä suoratoiston tuottajalle että suoratoiston mahdollistavalle taholle eli mediapalvelimelle. Mediapalvelimiin kohdistuvien uhkien lisäksi tutkimuksessa käytiin läpi huomiokohtia liittyen suoratoiston esittämiseen ja siihen, kuinka suoratoiston avoimuuden hyväksikäyttö on otettava mainostajana huomioon. Tämän opinnäytetyön tutkimustulokset ovat esitetty tuloksensaannin pohjalta. Tietoturvalliset käsitteet ja menettelytavat ovat kuitenkin rajoitettu RTMP:n toimikykyisyyteen, eikä tutkielmassa keskitytty esimerkiksi erilaisten mediapalvelinsovellusten toimitapoihin juuri lainkaan kyseisen protokollan ulkopuolella.

Totuus on, että palveluntarjoajan kannalta ei olisi käytännöllistä suojata joka ikistä yksilöllistä suoratoistoa mahdollisimman turvallisella tavalla, sillä siihen vaaditaan aikaa ja resursseja. Tämä on nykypäivänä huomattavissa muissakin yrityksissä suoratoiston ulkopuolella. Yritykset pyrkivät säästämään rahaa keskittymällä enemmän muuhun kuin tietoturvaan. Suoratoistoon liittyvien tahojen vaihtama informaatio on kuitenkin vain mediaa, joka on tarkoitettu muutenkin julkiseen käyttöön. Tosin tällaisella ajattelutavalla usein unohdetaan yksilöihin kohdistuvat tietoturvauhat. Tässä yksilöllä tarkoitetaan suoratoiston tuottajaa, joka vaikuttaa jääneen oman onnensa nojaan suoratoiston tietoturvassa.

LÄHTEET

Adobe Systems Inc. 2016. Adobe Media Server Help. Viitattu 16.02.2016. https://helpx.adobe.com/adobe-media-server/tech-overview/topics.html > Common uses for the server & > Streaming media

Adobe Systems Inc. 2005. Adobe Completes Acquisition of Macromedia. Viitattu 11.02.2016 https://www.adobe.com/aboutadobe/pressroom/pressreleases/pdfs/200512/120505AdobeAcquiresMacromedia.pdf

Apple Inc. 2015. HTTP Live Streaming draft-pantos-http-live-streaming-18. IETF Versio 18. Viitattu 16.02.2016 https://tools.ietf.org/html/draft-pantos-http-live-streaming-18

Catalin 2009. Unicast and multicast streaming. Viitattu 18.2.2016. http://www.thehdstandard.com/hd-streaming/unicast-and-multicast-streaming/

Javvin Technologies Inc. 2005. Network Protocols Handbook. 2nd Edition. Verkkopainos: https://books.google.fi/books?id=D_GrQa2ZcLwC&dq=RTSP&source=gbs_navlinks_s

JWPlayer 2015. About RTMP Streaming. Viitattu 11.02.2016 https://support.jwplayer.com/ > TOPICS > Streaming > About RTMP Streaming

Karanjit, S.S. 2000. Windows® 2000 TCP/IP. Second Edition. USA: New Riders Publishing

Krasic, C.; Li, K. & Walpole, J. 2001. The Case for Streaming Multimedia with TCP. Verkkojulkaisu: http://cobweb.cs.uga.edu/~kangli/publications.html > "The Case for Streaming Multimedia with TCP". Julkaisu on osana teosta Shepherd, D.; Finney, J.; Mathy, L. & Race, N. 2001. Interactive Distributed Multimedia Systems: 8th International Workshop.

Lamping, U.; Sharpe, R & Warnicke, E. 2014. Wireshark User's Guide: For Wireshark 2.1. Verkkojulkaisu: https://www.wireshark.org/download/docs/user-guide-us.pdf

Microsoft 2007. Windows Server: Selecting unicast vs. multicast distribution. Viitattu 01.03.2016 https://technet.microsoft.com/en-us/library/cc731550(v=ws.10).aspx > About multicast streaming > About multicast IP addresses and ports

Microsoft 2016. 8.2.3 Microsoft Media Server Protocol. Viitattu 16.02.2016. https://msdn.microsoft.com/en-us/library/cc239490.aspx

Mozilla Developer Network 2016. HTML5. Viitattu 17.02.2016. https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5

Nishida, J. 2003. Fast Retransmit (2). http://www.soi.wide.ad.jp/class/20020032/slides/15/33.html

Parmar, H. & Thornburgh, M. 2012. Real-Time Messaging Protocol (RTMP) specification (Version 1.0). http://www.adobe.com/devnet/rtmp.html: Adobe

Richard Leider 2015. YouTube Engineering and Developers Blog: YouTube now defaults to HTML5 <video>. Viitattu 17.02.2016 http://youtube-eng.blogspot.fi/2015/01/youtube-now-defaults-to-html5_27.html

Sanders, W. 2008. Learning Flash Media Server 3. USA: O'Reilly Media, Inc.

Seel, N. M. 2014. Encyclopedia of the Sciences of Learning. RTMP. Verkkopainos: https://play.google.com/books/reader?id=u7kue5YV4lkC&printsec=frontcover&output=reader&hl=en_GB&pg=GBS.PP1: Springer Science & Business Media

Twitch Support Center 2016. Broadcast Requirements. Viitattu 12.02.2016 http://help.twitch.tv/ > Broadcast Hardware and Software > Broadcast Requirements

Twitch, 2016. Retrospective: 2015. Verkkojulkaisuun viitattu: 15.03.2016. https://www.twitch.tv/year/2015

Wankel, C. 2011. Streaming Media Delivery in Higher Education: Methods and Outcomes. Verkkopainos: https://books.google.fi/books?id=hXN7pSpD48cC&lpg=PR1&hl=fi&pg=PR1#v=onepage&q&f=false: Information Science Reference

Zhang, S. 2009. How to Build Scalable and Robust Live Smooth Streaming Server Solutions. Kuva. http://blogs.iis.net/samzhang/how-to-build-scalable-and-robust-live-smooth-streaming-server-solutions > Figure 9

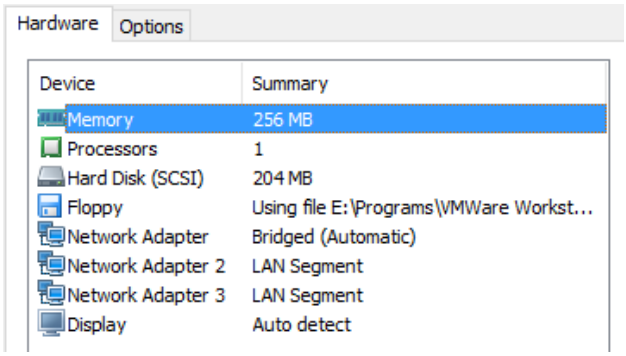
Tutkimuksessa käytetyn virtuaaliympäristön asentaminen

Tässä dokumentissa käydään läpi tutkimuksessa asennetut ja luodut virtuaalikoneet ja niiden ympäristö.

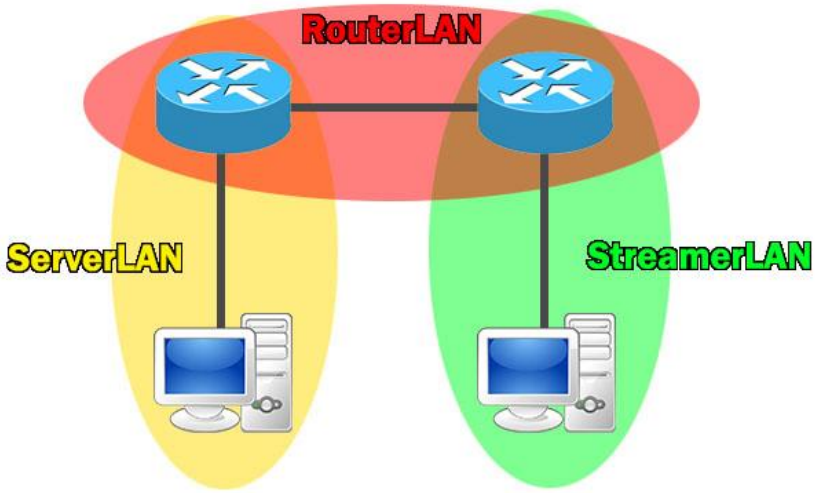
Verkkoympäristö

Tämän tutkimuksen kuvan 4 mukaisesti luotu verkkoympäristö sisältää 3 lähiverkkoa, jotka on yhdistetty reitittimien avulla. Reitittimien asettamiseen on käytetty ilmaista ja kevyttä Linux-pohjaista sovellusta Freesco (latauslinkki: http://freesco.sourceforge.net/). Sovelluksen imagetiedosto täytyy muuntaa .flp-muotoon asennusta varten. Asennamme ensimmäisenä vastaanottavan mediapalvelimen lähiverkon.

Kun olemme lisänneet Linux-pohjaisen virtuaalikoneen Wmwareen tulisi Hardware-asetusten näyttää tältä.

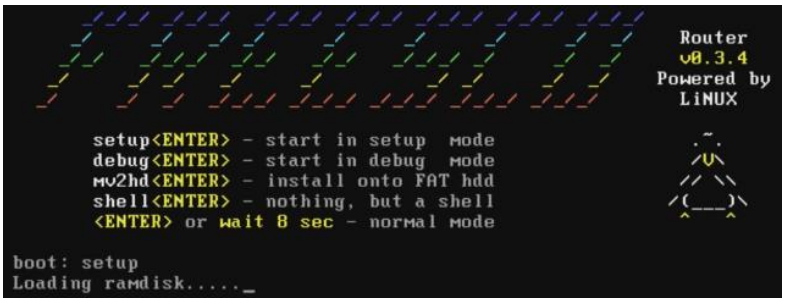


Ylimääräisiä komponentteja, kuten tulostin ja äänikortti, ei tarvita. Vastaanottavan mediapalvelimen virtuaalikoneeseen lisätään Floppy-asema, Network Adapter 2 ja Network Adapter 3. Adapterit ovat lähtökohtaisesti eri LAN Segmenteissä. Tämän virtuaalikoneen Network Adapter 2 on asetettu LAN segmentille ”ServerLAN” ja Network Adapter 3 on asetettu LAN segmentille ”RouterLAN”. Sama tullaan tekemään suoratoistoa tuottavan koneen lähiverkossa. Lopullisen verkkosegmentoinnin tulisi näyttää tältä.



Verkkoympäristöt eivät kuitenkaan toimi automaattisesti niin kuin haluamme vaan meidän täytyy asettaa reitittimet jakamaan IP-osoitteita.

Virtuaalikoneen käynnistämisen jälkeen Freesco kysyy mitä haluamme bootata. Tähän kirjaamme ”setup” ja painamme Enter-näppäintä.



Haluamme asentaa Freesco-reitittimemme ”Ethernet router”-asetuksilla. Tässä tapauksessa kirjaamme ”e” ja painamme Enter-näppäintä. Etenemme asennuksessa haluammallamme tavalla ohjeiden mukaisesti

Lopullisen verkoituksen tulisi näyttää ServerLAN-reitittimessä tältä:

```
[ Network #  ]        0.                1.                2.
[ Interface  ]        eth0              eth1              eth2
[ IP address ]        192.168.2.2       192.168.200.1     10.0.0.1

[ Network #  ]        3.                4.                5.
[ Interface  ]
[ IP address ]

[ Network #  ]        6.                7.                8.
[ Interface  ]
[ IP address ]

[ Network #  ]        9.
[ Interface  ]
[ IP address ]
[ ---------------------- Network #0 specific settings -------------------
620. Interface Name    = eth0              626. Use PPP ethernet = n
621. IP address        = 192.168.2.2       627. Use DHCP client  = n
622. Network mask      = 255.255.255.0     628. Set DNS via DHCP = y
623. Network addr      = 192.168.2.0       629. MAC addr =
624. Broadcast addr    = 192.168.2.255
625. DHCP server pool =
631. Gateway  =
Choose network 0-9 or its parameter to change (x - exit) []? _
```

```
[ ---------------------- Network #1 specific settings --------
620. Interface Name    = eth1              a. (A)uto configure
621. IP address        = 192.168.200.1     c. (C)lear network
622. Network mask      = 255.255.255.0
623. Network addr      = 192.168.200.0
624. Broadcast addr    = 192.168.200.255
625. DHCP server pool = 192.168.200.20 192.168.200.40
```

```
[ ---------------------- Network #2 specific settings ----------
620. Interface Name    = eth2              a. (A)uto configure ne
621. IP address        = 10.0.0.1          c. (C)lear network #2
622. Network mask      = 255.255.255.248
623. Network addr      = 10.0.0.0
624. Broadcast addr    = 10.0.0.7
625. DHCP server pool =
```

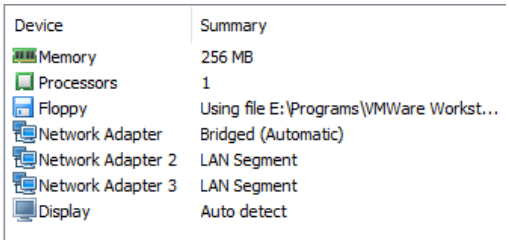
Erityisenä huomiokohtana asennuksessa on, että reitittimestä asetetaan lähiverkot luotetuiksi kun asennus kysyy seuraavaa asiaa:

```
251 Trust local network 1 (y/n) [y]? y
251 Trust local network 2 (y/n) [n]? y_
```

Asetusten jälkeen kirjaamme päävalikossa ”s” eli ”Save current config and exit”.

```
s) Save current config and exit

Your choice []? _
```

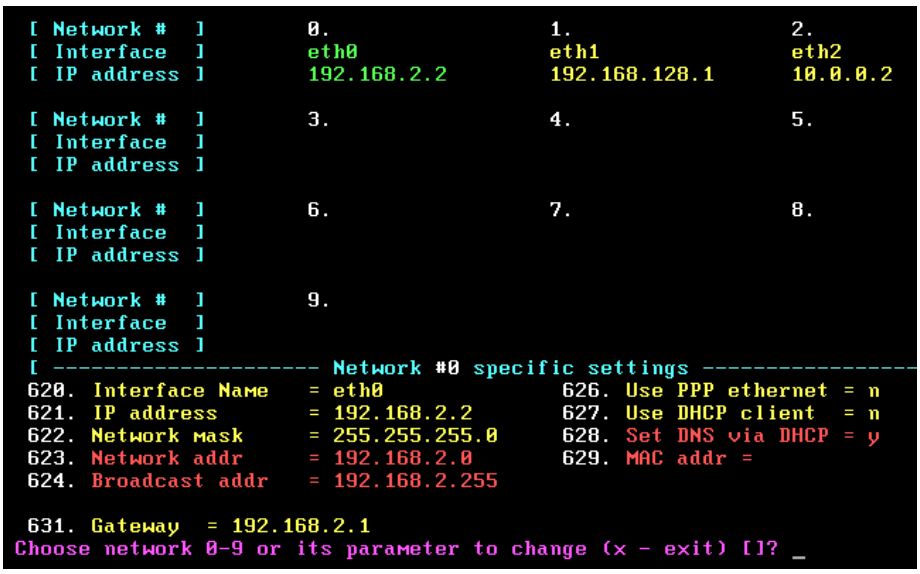
Samat toimenpiteet teemme toiselle reitittimelle suoratoiston tuottajan päädyssä. Reitittimen hardware näyttää tältä:



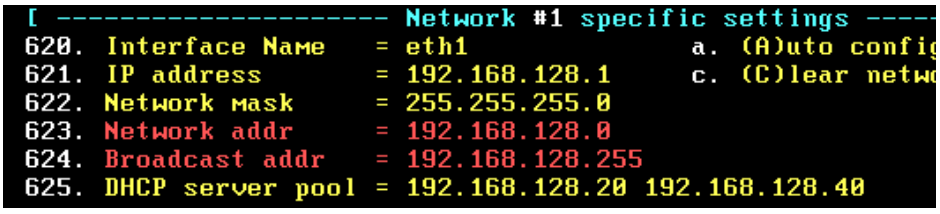
Device	Summary
Memory	256 MB
Processors	1
Floppy	Using file E:\Programs\VMWare Workst...
Network Adapter	Bridged (Automatic)
Network Adapter 2	LAN Segment
Network Adapter 3	LAN Segment
Display	Auto detect

Network Adapter 2 on asetettu LAN segmenttiin ”StreamerLAN” ja Network Adapter 3 on sama kuin mediapalvelimen reitittimellä, eli se on segmentissä ”RouterLAN”. Tämä on siksi, että yhdyskäytävä reitittmien välillä olisi samassa lähiverkossa.

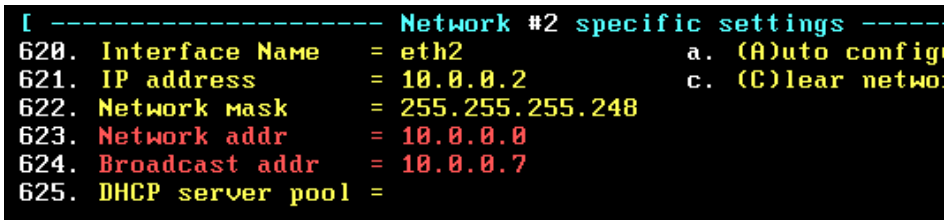
Verkoitus näyttää suoratoiston tuottajan lähiverkossa olevassa reitittimessä tältä:



```
 [ Network #   ]        0.                1.                2.
 [ Interface   ]        eth0              eth1              eth2
 [ IP address  ]        192.168.2.2       192.168.128.1     10.0.0.2

 [ Network #   ]        3.                4.                5.
 [ Interface   ]
 [ IP address  ]

 [ Network #   ]        6.                7.                8.
 [ Interface   ]
 [ IP address  ]

 [ Network #   ]        9.
 [ Interface   ]
 [ IP address  ]
 [ ---------------------- Network #0 specific settings ----------------
 620. Interface Name    = eth0              626. Use PPP ethernet = n
 621. IP address        = 192.168.2.2       627. Use DHCP client  = n
 622. Network mask      = 255.255.255.0     628. Set DNS via DHCP = y
 623. Network addr      = 192.168.2.0       629. MAC addr =
 624. Broadcast addr    = 192.168.2.255

 631. Gateway   = 192.168.2.1
Choose network 0-9 or its parameter to change (x - exit) []? _
```



```
 [ ---------------------- Network #1 specific settings -----
 620. Interface Name    = eth1              a. (A)uto confi
 621. IP address        = 192.168.128.1     c. (C)lear netw
 622. Network mask      = 255.255.255.0
 623. Network addr      = 192.168.128.0
 624. Broadcast addr    = 192.168.128.255
 625. DHCP server pool  = 192.168.128.20 192.168.128.40
```



```
 [ ---------------------- Network #2 specific settings -----
 620. Interface Name    = eth2              a. (A)uto confi
 621. IP address        = 10.0.0.2          c. (C)lear netwo
 622. Network mask      = 255.255.255.248
 623. Network addr      = 10.0.0.0
 624. Broadcast addr    = 10.0.0.7
 625. DHCP server pool  =
```

Verkkojen asettamisen jälkeen meidän täytyy asettaa oletusyhdyskäytävät molemmista reitittimistä. Mediapalvelimen reitittimessä kirjaamme ”route”-komennolla näin:

route add -net 192.168.128.0 netmask 255.255.255.0 gateway 10.0.0.2

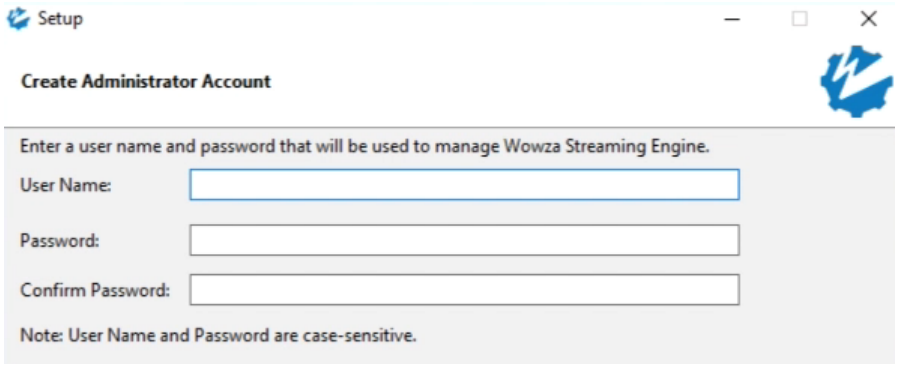
Suoratoiston tuottajan päädyssä reitittimeen kirjataan näin:

route add -net 192.168.200.0 netmask 255.255.255.0 gateway 10.0.0.1

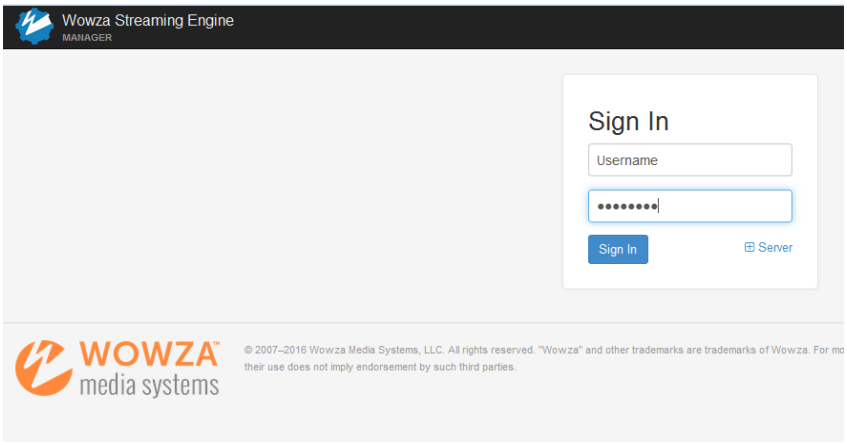
Reitityksen pitäisivät tämän jälkeen olla toiminnassa ja esimerkiksi ”ping”-komennot mennä laitteelta laitteelle läpi. Mikäli ”pingit” eivät jostain syystä löydä perille, kannattaa varmistaa, ettei kummassakaan koneessa ole palomuuria estämässä verkkoliikennettä.

Vastaanottava mediapalvelin

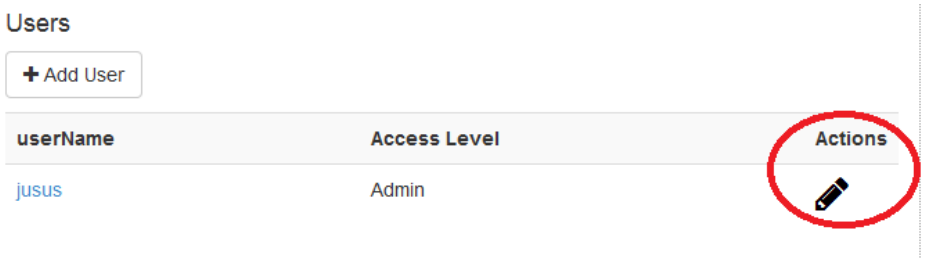
Suoratoistoa vastaanottava mediapalvelin toimii Windows 7 -käyttöjärjestelmällä ja Wowza Streaming Engine 4.1 -sovelluksella. WSE-sovellus on saatavilla verkko-osoitteesta https://www.wowza.com/pricing/installer testikäyttöön 180 päiväksi rekisteröitymällä. Tätä tutkimusta varten ilmainen kokeiluversio on huomattu riittäväksi. Asentaessamme Windows 7 -virtuaalikoneelle WSE (Wowza Streaming Engine) pyytää asettamaan käyttäjänimen ja salasanan, joita käytämme manager-ohjelman käynnistyksessä. Näitä käyttäjätietoja tarvitaan myöhemmin.



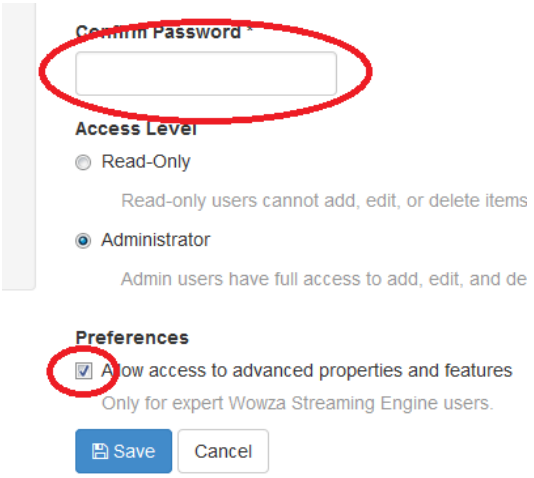
Asennuksen jälkeen käynnistämme virtuaalikoneestamme ohjelman ”Wowza Streaming Engine Manager.exe”. Tämä avaa oletusselaimella ohjelman hallintapaneelin, joka pyytää aiemmin asettamia käyttäjätunnuksia.



Sisäänkirjautumisen jälkeen eteemme aukeaa kojelauta ja ”Home”-ruutu. Ensimmäisenä navigoimme ylhäältä välilehdelle ”Server” ja ”Users”. Editoimme omaa profiilia painamalla kynän kuvaa keskisivun oikesta laidasta käyttäjänimemme kohdalta.

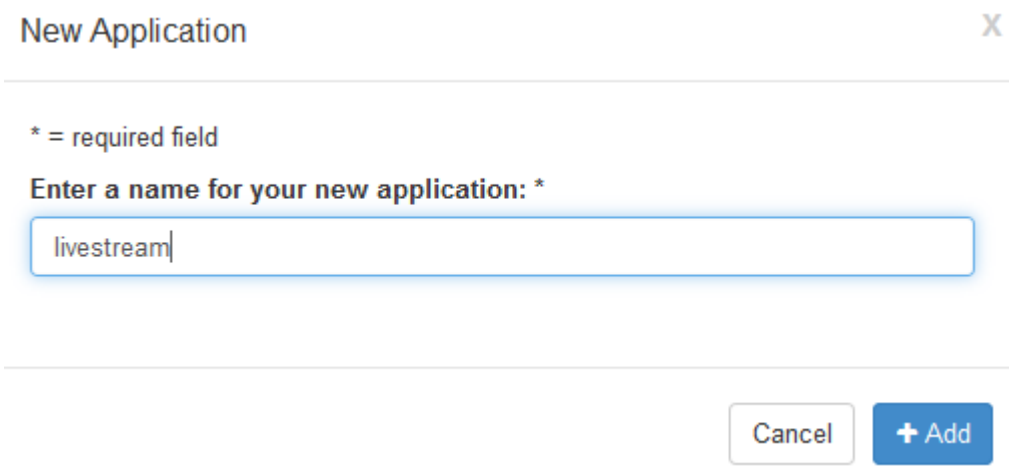


Ennen kuin voimme tehdä muutoksia, kuten lisätä moduuleja ja ominaisuuksia palvelimellemme, meidän on ensin asetettava tämän sivun ”Preferences”-asetuksista täppä ”Allow access to advanced properties and features” -kohtaan. Tämän jälkeen kirjoitamme salasanamme tyhjään laatikkoon ”Confirm Password *” ja painamme sinistä ”Save”-nappia sivun alalaidassa. HUOM! Tämän jälkeen saatamme joutua kirjautumaan uudelleen sisään ennen kuin muutoksemme tulevat voimaan.



Nyt kun olemme ”Advanced”-käyttäjä, voimme navigoida ylälaidasta ”Applications”-välilehdelle ja käydä luomassa ”Live”-profiilimme ”+Add Application” -kohdasta vasemmasta ylälaidasta. Tämä profiili sisältää asetukset vastaanotettavia suoratoistoja varten, kuten PC2:n suoratoisto. PC2:n asetukset käymme läpi myöhemmin.

Painamme ”Live” ja nimeämme profiilimme ”livestream”.

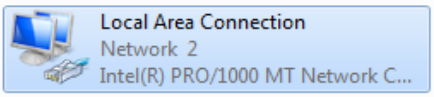


Painamme ”+Add” ja siirrymme profiilimme asetuksiin.

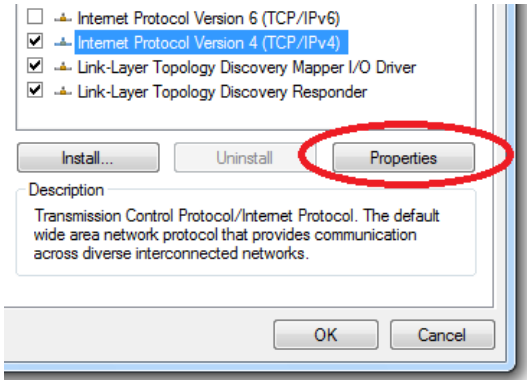
Seuraavaksi siirrymme Windowsin verkkoadapterien hallintapaneeliin asettamaan oikean IP-osoitteen. Painamme näppäimistöstämme Windows-nappia tai avaamme vasemmasta alalaidasta käynnistysvalikon.

Kirjoitamme hakupalkkiin ”Network and sharing center” ja painamme Enter-näppäintä.

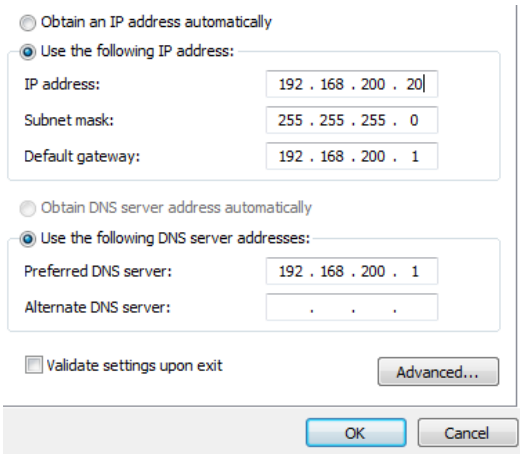
Klikkaamme ikkunan vasemmasta laidasta ”Change adapter settings” ja tuplaklikkaamme käytössämme olevan verkon kuvaketta.



Tästä aukeaa status-ikkuna, jonka alalaidasta painamme ”Properties”. Klikkaamme hiiren kohtaan ”Internet Protocol Version 4 (TCP/IPv4)” ja painamme jälleen ”Properties”.



Asetamme IP-osoitteet seuraavasti:



Tämän jälkeen painamme ”OK”.

Mikäli nyt haluamme yhdistää pakkauksenhallintaohjelmallamme suoratoistoa vastaanottavaan mediapalvelimeen suoraa lähetystä varten, meidän on vain yhdistettävä URL:iin:

rtmp://192.168.200.20:1935/livestream/

Tutkimuksen mediapalvelimessa käytettiin kahta erilaista salausmetodia suoratoistolle. Ne olivat käyttäjänimi- ja salasanayhdistelmä yhdistysURL:iin sekä suoratoistoavain. Molemmat vaativat erityisasennuksia mediapalvelimelle.

Suoratoistokoneen asetukset

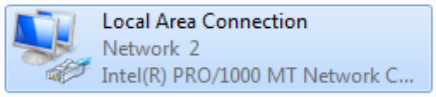
Tässä välissä säädämme suoratoistoa tuottavan koneen kuntoon multimedian lähetystä varten.

Asetamme koneelle aluksi oikean IP-osoitteen samalla tavalla kuin mediapalvelimellekin.

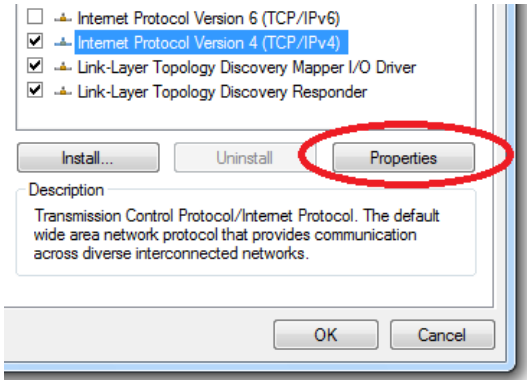
Painamme näppäimistöstämme Windows-nappia tai avaamme vasemmasta alalaidasta käynnistysvalikon.

Kirjoitamme hakupalkkiin ”Network and sharing center” ja painamme Enter-näppäintä.

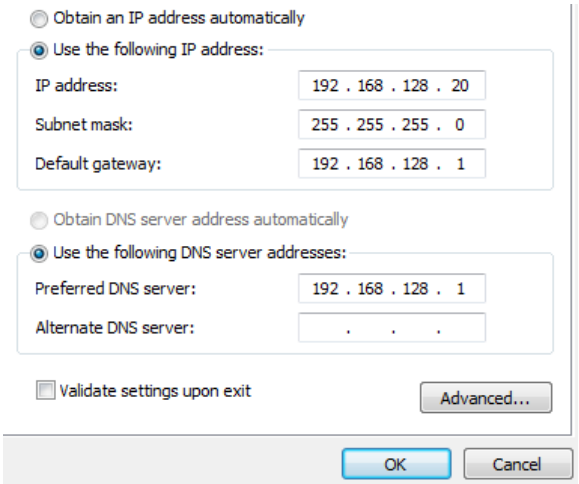
Klikkaamme ikkunan vasemmasta laidasta ”Change adapter settings” ja tuplaklikkaamme käytössämme olevan verkon kuvaketta.



Tästä aukeaa status-ikkuna, jonka alalaidasta painamme ”Properties”. Klikkaamme hiiren kohtaan ”Internet Protocol Version 4 (TCP/IPv4)” ja painamme jälleen ”Properties”.

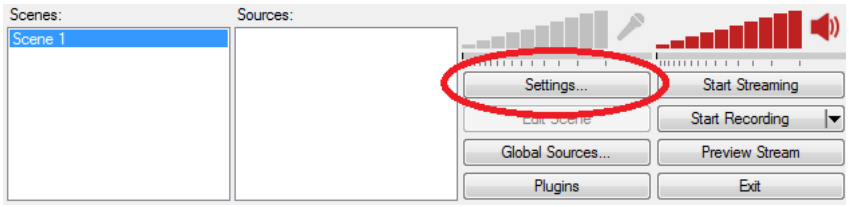


Asetamme IP-osoitteet seuraavasti:

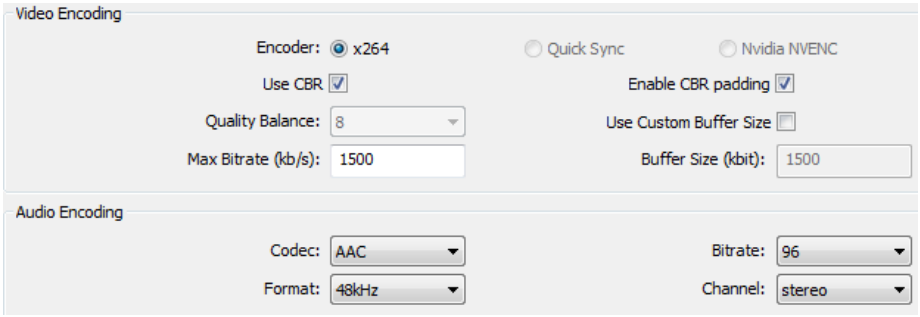


Tämän jälkeen painamme ”OK”.

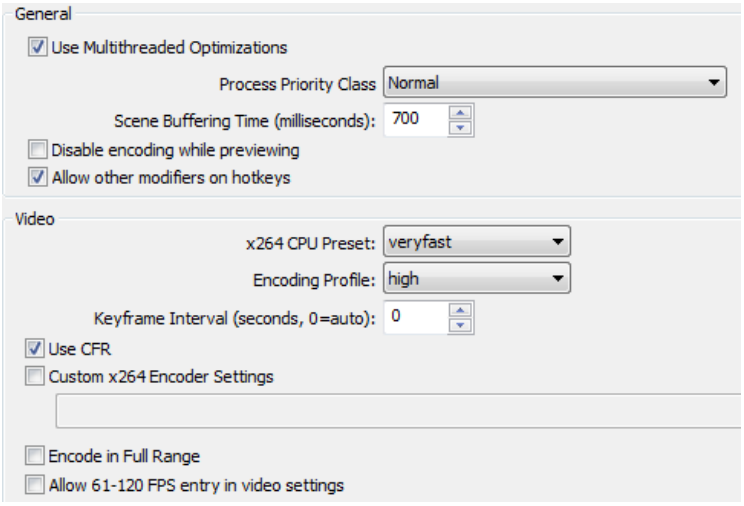
OBS-sovelluksen asentamisen ja käynnistämisen jälkeen painamme sovelluksesta ”Settings”-painiketta.



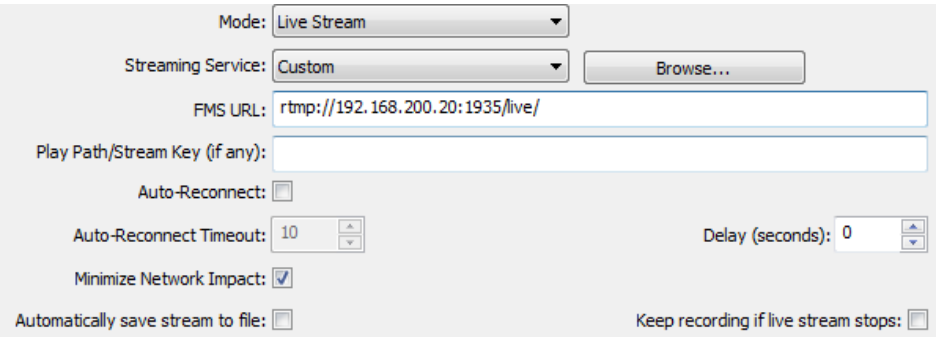
Välilehdeltä ”Encoding” voimme tämän tutkimuksen kannalta käyttää seuraavia asetuksia.



Tämän tutkimuksen kannalta suoratoiston ei tarvitse olla kovinkaan resursseja vaativa, joten välilehdellä ”Advanced” voimme käyttää seuraavia asetuksia. HUOM! Näitä samoja asetuksia voi käyttää Adoben Flash Media Live Encoderissa.

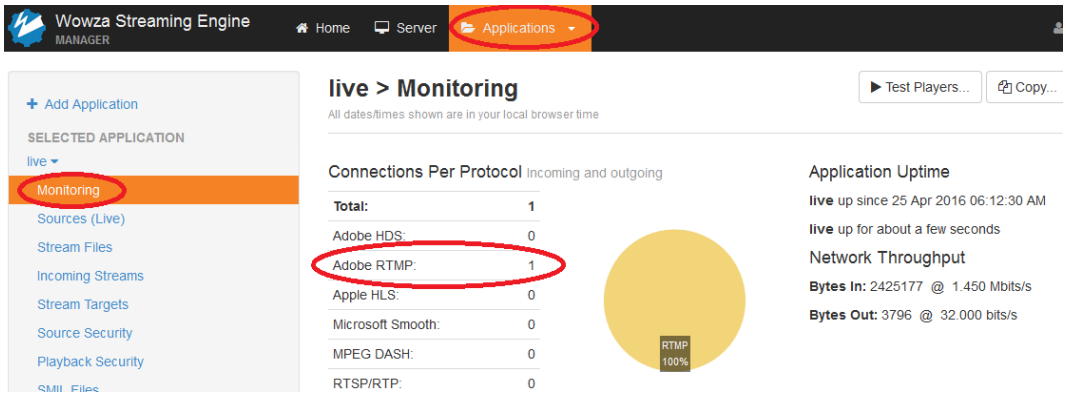


Tämän jälkeen ainut asia, jota mediapalvelimen vaatimuksien mukaisesti tulee muuttaa on välilehdellä ”Broadcast Settings”. Asetukset voidaan tässä vaiheessa jättää seuraaviksi.

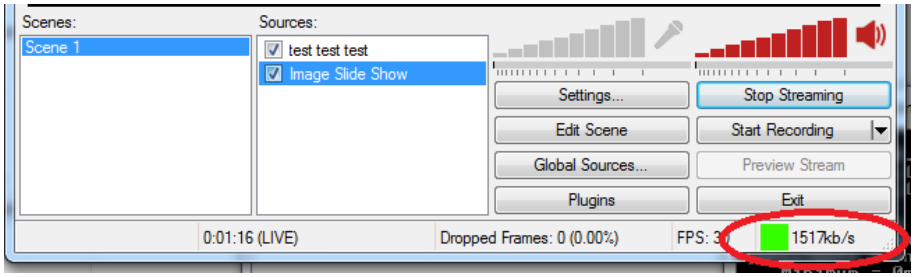


Voimme nyt tallentaa ja poistua asetuksista painamalla alalaidasta ”Apply” ja ”OK”.

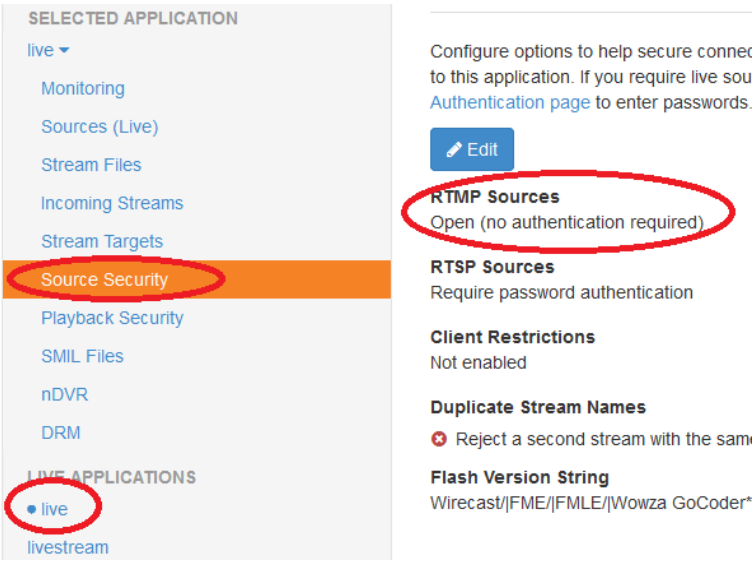
Tässä vaiheessa voimme testata rtmp://-yhteyttä painamalla ”Start Streaming”-painiketta OBS-sovelluksesta. Yhteys on muodostettu mikäli mediapalvelimella näkyy rtmp-liikennettä:



Ja OBS-ilmoittaa alalaidassa verkkoliikenteen määrän:



Mikäli yhteyttä ei voitu muodostaa, kannattaa mediapalvelimen asetuksista ”Applications”-kohdasta tarkistaa ”live”-applikaation ”Source Security”-välilehti. RTMP Sources tulisi olla ”Open”, sillä tässä vaiheessa ei suuria muutoksia olla vielä WSE:lle tehty.

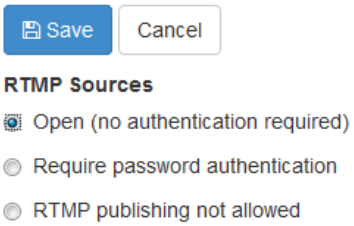


URLParams-käyttäjänimi ja -salasana

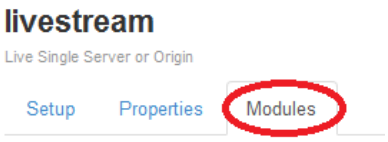
Kaikissa ilmaisissa pakkauksenhallintasovelluksissa (kuten OBS) ei löydy mahdollisuutta RTMPauthille, joka kysyy käyttäjänimeä ja salasanaa yhteyden luomisessa mediapalvelimelle. Selitämme tässä osiossa tyylin käyttäjänimi ja salasana tunnistukselle ilman RTMPauthia.

Ensimmäisenä siirrymme sivun ylälaidasta löytyvälle ”Applications”-välilehdelle ja varmistamme, että olemme luomassamme ”livestream”-applikaatiossa klikkaamalla sitä sivun vasemmasta laidasta.

Seuraavaksi painamme ”Source Security” vasemmasta laidasta ja sivun latauduttua sinistä ”Edit”-nappia. Asetamme kohdassa ”RTMP Sources” täpän kohtaan ”Open (no authentication required)” ja painamme sinistä ”Save”-nappia.



Nyt menemme luomaan moduulin ja lisäämään ominaisuuden. Avaamme ”Modules” -välilehdelle sivun ylälaidassa.

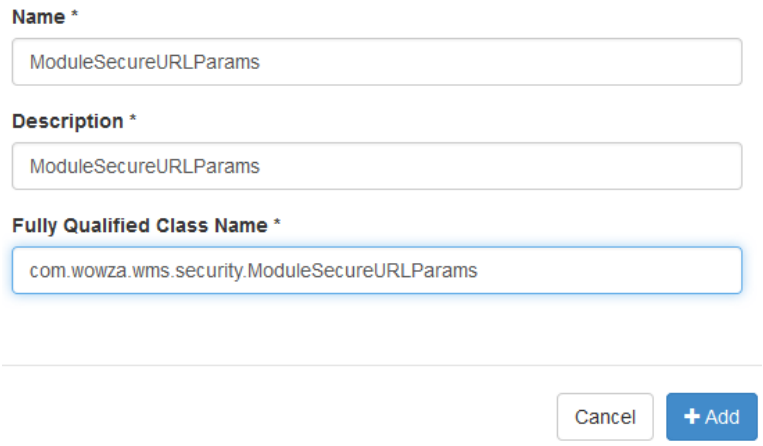


Valitsemme ”Edit” ja ”+Add Module:”. Lisäämme palvelimeemme moduulin, joka vaatii käyttäjäpuolen autentikaatiota ennen suoratoiston hyväksymistä. Nimen, kuvauksen ja ”Fully Qualified Class Name” -kohdan merkitsemme seuraavalla tavalla ja painamme lopuksi ”+Add”.

ModuleSecureURLParams

ModuleSecureURLParams.

com.wowza.wms.security.ModuleSecureURLParams



Ennen kuin jatkamme, painamme sinistä ”Save”-nappia joko ruudun ylä- tai alalaidasta, jonka jälkeen sivu pyytää uudelleenkäynnistämään applikaation. Painamme oranssia ”Restart Now”-nappia.

Navigoimme nyt itsemme välilehdelle ”Properties”, joka sijaitsee ”Modules”-välilehden vieressä. Siirrymme sivun alalaitaan painamalla pikalinkkiä: ”Custom” tai rullaamalla hiiren rullaa ja painamme jälleen ”Edit”-nappia ”Custom”-paneelin vieressä. Mikäli kyseisiä ominaisuuksia ei vielä löydy lisätään ne painamalla ”+Add Custom Property:”.

/Root/Application

secureurlparams.publish

String

password123.streamer123

Listan tulisi näyttää lopuksi tältä:

Path	Name	Type	Value
/Root/Application	pushPublishMapPath	String	${com.wowza.wms.context.VHostConfigHome}/conf/${com.wowza.wms.context.Application}/PushPublishMap.txt
/Root/Application	secureurlparams.publish	String	password123.streamer123

Painamme sinistä ”Save”-nappia ja sivun kehotteesta käynnistämme applikaation jälleen uudelleen painamalla oranssia ”Restart Now”-nappia.

Tämän moduulin ansiosta mediapalvelin hyväksyy vain yhteyksiä, joissa käyttäjänimi ja salasana ovat yhteyden muodostukseen käytettävässä URL:ssa. Pakkauksenhallintaohjelman yhdistettävän palvelimen osoiterivi tulisi näyttää tässä tapauksessa tältä:

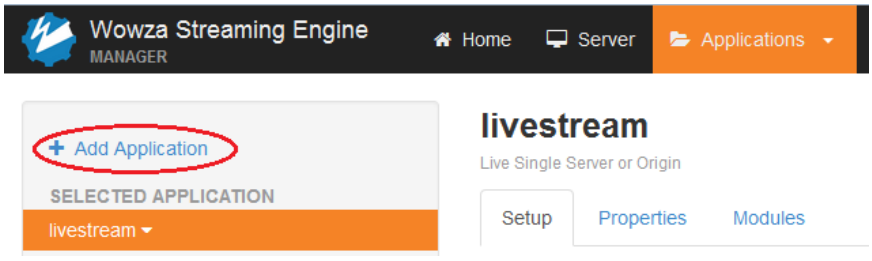
rtmp://192.168.200.20:1935/livestream/_definst_/streamer123=password123

Mikäli jatkossa luotaisiin lisää käyttäjiä, tulee heidän tiedot sisällyttää erillisille applikaatioille kuten ”livestream”. Esimerkiksi ”livestream2”, ”livestream3”, jne.

HUOM!: URL-parametri EI SAA sisältää käyttäjänimeä tai salasanaa mahdolliselle käyttäjäportaalille administraatiokäyttöön mediapalvelimella. Tämä on vain suoratoiston tunnustukseen. Mikäli suoratoiston tunnukset ovat samat kuin käyttäjän pääsytiedot mediapalvelimen asetuksiin, syntyy suuri tietoturvariski.

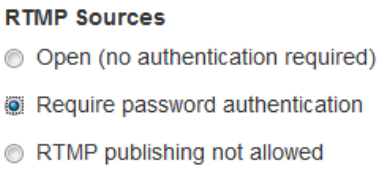
RTMPauth-käyttäjänimi ja -salasana

Käyttäjänimen ja salasanan vaatiminen pystytään toteuttamaan vain RTMPauth-mahdollisuutta tukevien pakkauksenhallinta- ja mediapalvelinsovellusten välillä. Tässä tutkielmassa Wowza Streaming Engineen otettiin yhteyttä Adoben oman Flash Media Live Encoderin avulla, jossa RTMPauth on mahdollista. Tätä varten luotiin WSE:ssä uusi applikaatio painamalla ”**+**Add Application”-nappia ”Applications”-välilehden vasemmasta ylälaidasta.

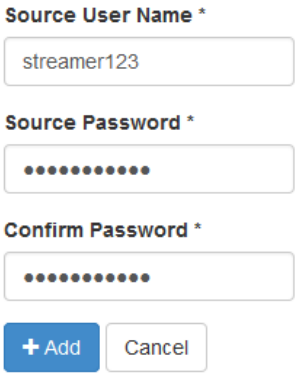


Valitsemme ”Live” ja nimeämme applikaatiomme ”protectedLive”.

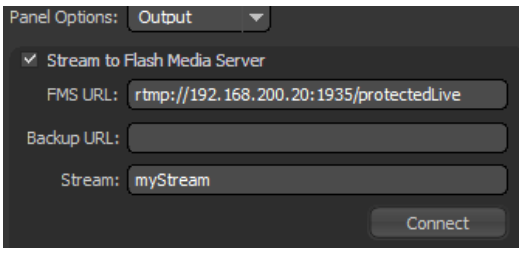
Tarkistamme vasemmalta ”Source Security”-välilehden asetuksista kohdan ”RTMP Sources”. Tällä kertaa asetuksen tulisi olla ”Require password authentication”. Tämä asetus pitäisi olla oletuksena.



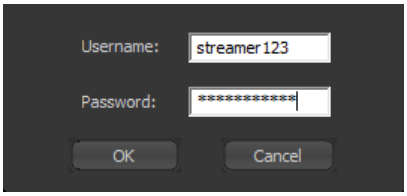
Tämän jälkeen painamme sivun ylälaidasta ”Server”-välilehdelle ja vasemmalta ”Source Authentication”. Lisäämme ”+Add Source” painikkeesta uuden käyttäjätilin mediapalvelimelle. Tutkimuksemme esimerkissä käytettiin ”streamer123” käyttäjänimeä ja ”password123” salasanaa. Kun olemme valmiit painamme sinistä ”+Add”-nappia.



Nyt kun suoratoistoon, jonka tunnus on ”protectedLive”, yhdistetään, tulee RTMPauth-yhteensopivassa pakkauksenhallintasovelluksessa asettaa käyttäjänimi ja salasana. Käyttäjätietojen on oltava samat kuin mediapalvelimen asetuksissa. FMLE:stä yhdistäminen näyttää tältä:



Painetaan ”Connect”, jonka jälkeen ponnahdusikkunaan kirjataan käyttäjätiedot.



Valmis.

Tietomurtotestaus

Tietomurtotestauksessa käytettiin linux-pohjaista Kali Linux -käyttöjärjestelmää, jonka valmiit työkalut erikoistuvat murtautumistestaukselle verkkoympäristössä. Image ladattiin verkko-osoitteesta https://www.kali.org/downloads/.

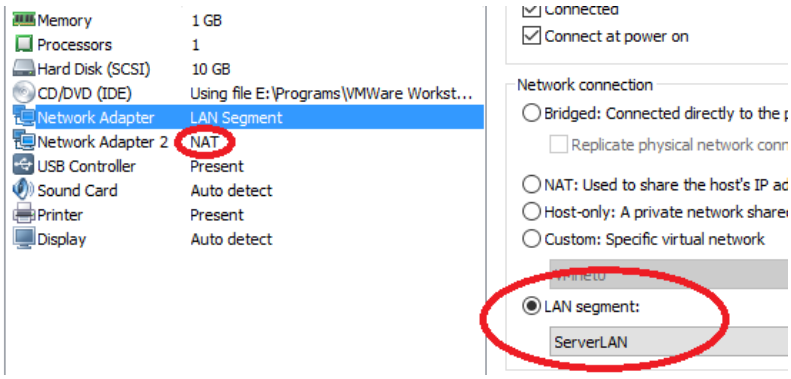
Kali Linux -koneessa jouduimme asentamaan erikseen sovellukset ”rtmpdump” ja ”mplayer”. Tämä tapahtui seuraavilla terminaali-komennoilla lähes automaattisesti.

apt-get install rtmpdump ja:

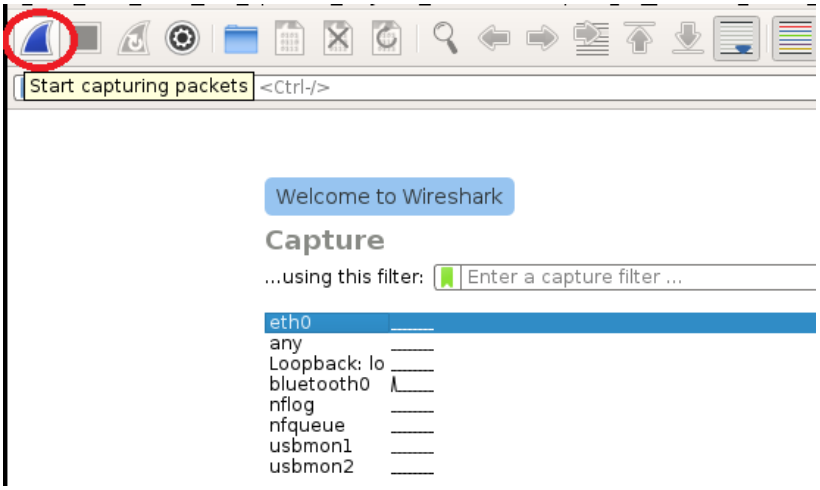
apt-get install mplayer

(Wireshark-ohjelman pitäisi löytyä Kali Linuxista valmiina, mutta mikäli ei löydy, käytetään seuraavaa komentoa: *apt-get install wireshark*)

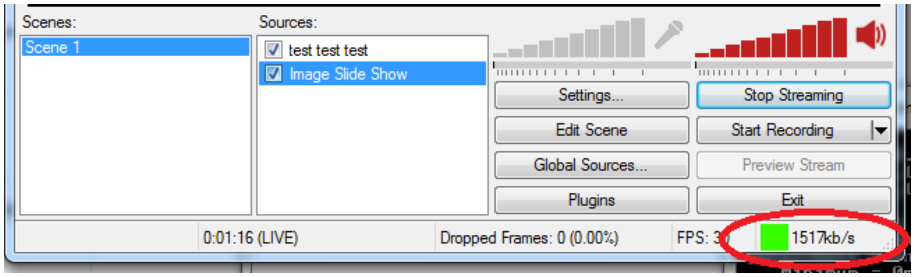
Mikäli asennuksissa tulee verkko-ongelmia täytyy virtuaalikoneen asetuksista varmistaa, että Kali Linux on tällä hetkellä samassa verkossa kuin ”Host”-pöytäkone. Itse olen tutkimusta varten laittanut Kali Linux -koneelle 2 verkkoadapteria, joista toinen jakaa oman pääkoneeni IP-osoitteen ja toinen on samassa lähiverkossa kuin mediapalvelimen virtuaalikone:



Ensin kaapataan verkkoliikenteestä Wiresharkilla suoratoistoon liittyvät tiedot. Käynnistetään Wireshark konsolista komennolla *wireshark*.

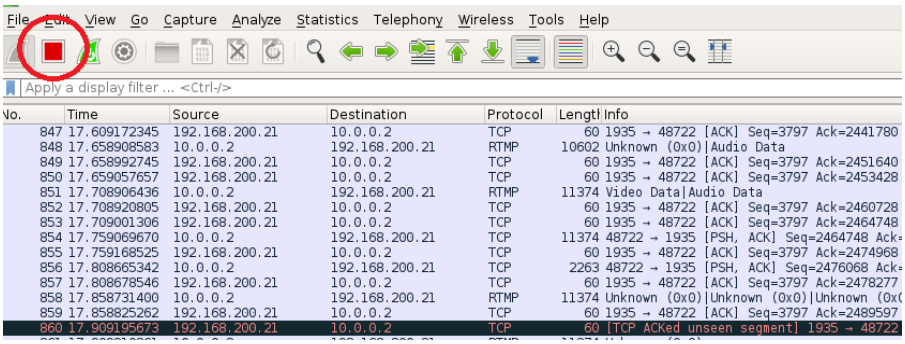


Valitaan pakettikaappausta varten oikean lähiverkon verkkosovitin tuplaklikkaamalla. Tämän jälkeen suoratoistokoneelta käymme painamassa ”Start Streaming” pakkauksenhallintasovelluksesta (OBS/FLME).

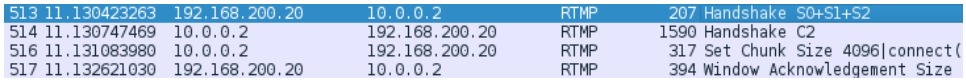


Annamme Wiresharkin kaapata muutaman sekunnin ajan suoratoiston käynnistyksestä, jonka jälkeen painamme suoratoistajan koneesta ”Stop Streaming”.

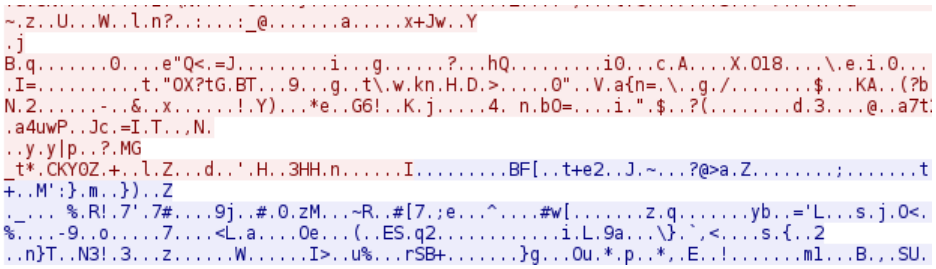
Pysäytämme myös Wiresharkin.



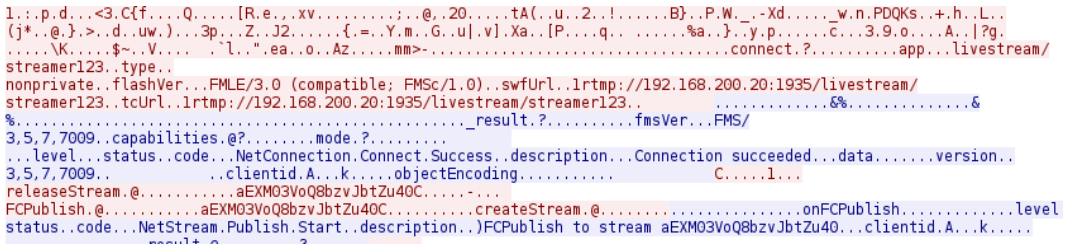
Tämän jälkeen Kali Linux -koneella etsimme Wiresharkin pakettivirrasta hakufunktiolla ”RTMP”-protokollan paketteja. Tarkemmin sanottuna etsimme pakettia, jonka info-osiossa lukee ”Handshake”.



Painamme hiiren oikealla painikkeella Handshake-kohtaan ja valitsemme Follow > TCP Stream. Tästä aukeaa erillinen ikkuna, jossa on datavirtaa kuvastavaa ascii-symboliikkaa. Käytännössä tämä on multimediaa tekstitiedostona.



Selaamme datavirtaa kunnes löydämme yhteydenluonnin kohdan. Tässä välittyy kaikki suoratoiston kannalta tärkeä informaatio.

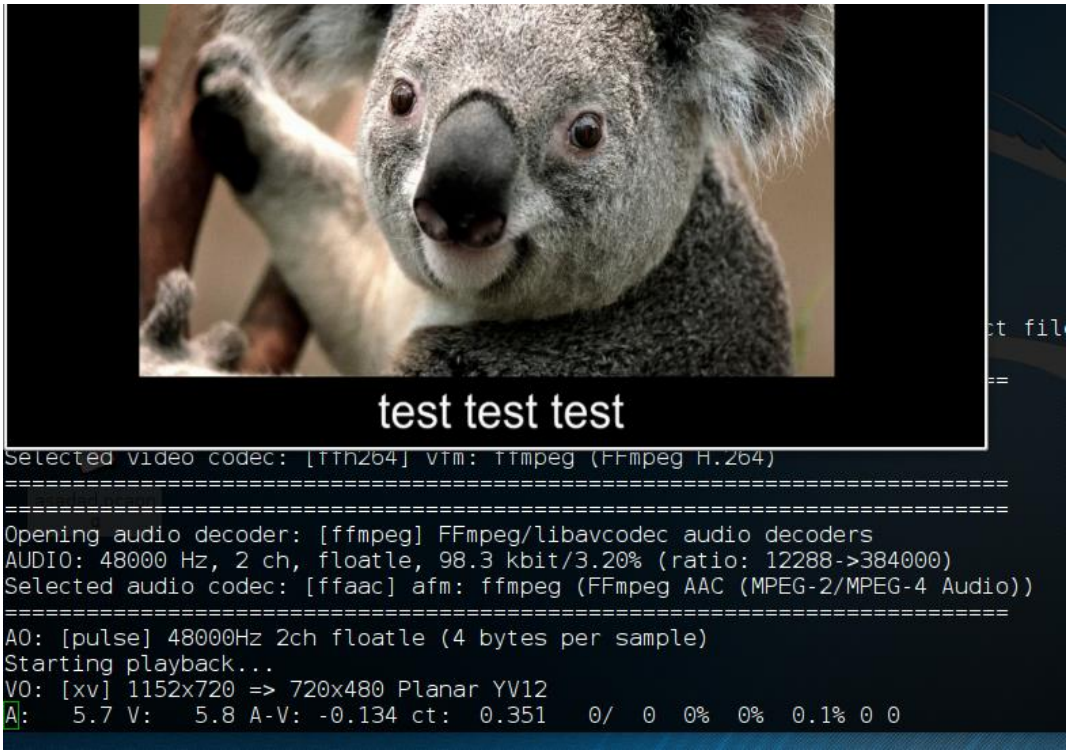


RTMPDump

RTMP-suoratoiston kaappaukseen tarvitaan rtmpdump-sovellus. Sovelluksen komento tulee suorittaa käynnissä olevan suoratoiston aikana. Mikäli suoratoisto tapahtuu URL:lla ”*rtmp://192.168.200.21:1935/live/*” tulee rtmpdump-komento olemaan seuraava:

rtmpdump -v -r rtmp://192.168.200.20:1935/live/ -o /root/Desktop/flvtest1 | sudo mplayer -x 720 -y 480 rtmp://192.168.200.20:1935/live/

Mikäli kaikki on tehty oikein mplayer-sovellus avaa erillisen ikkunan josta suoratoistoa voidaan seurata samalla kun se tallentuu tiedostoon nimeltä ”flvtest1”.


test test test
Selected video codec: [ffh264] vfm: ffmpeg (FFmpeg H.264)
Opening audio decoder: [ffmpeg] FFmpeg/libavcodec audio decoders
AUDIO: 48000 Hz, 2 ch, floatle, 98.3 kbit/3.20% (ratio: 12288->384000)
Selected audio codec: [ffaac] afm: ffmpeg (FFmpeg AAC (MPEG-2/MPEG-4 Audio))
AO: [pulse] 48000Hz 2ch floatle (4 bytes per sample)
Starting playback...
VO: [xv] 1152x720 => 720x480 Planar YV12
A: 5.7 V: 5.8 A-V: -0.134 ct: 0.351 0/ 0 0% 0% 0.1% 0 0