Huijaaminen verkkopeleissä

Oulun Yliopisto
Tietojenkäsittelytieteiden laitos
LuK tutkielma
Matti Ollila
24.5.2016

Tiivistelmä

Tutkielman aiheena on huijaaminen verkkopeleissä ja siinä keskitytään huijaajien käyttämiin huijauskeinoihin sekä miten niitä voitaisiin ehkäistä. Lisäksi lyhyesti tutustutaan huijaamisen psykologisiin piirteisiin. Tutkimus toteutetaan tutkimalla jo olemassa olevaa kirjallisuutta aiheesta.

Huijaajat ovat sosiaalisia pelaajia ja omasta epäsosiaalisesta käyttäytymisestä huolimatta verkostoituvat muiden pelaajien kanssa. Huijaamiselle voidaan löytää niin taloudellisia kuin henkilökohtaisempia syitä. Osa huijaajista huijaa kerätäkseen virtuaalista omaisuutta ja myydäkseen sitä eteenpäin. Osa taas nauttii saadessaan tuottaa ärsykkeitä muille pelaajille. Näille huijaajille syyt voivat liittyä pelaamiseen, muihin pelaajiin, ryhmäytymiseen tai omaan persoonallisuuteen.

Verkkopeleissä huijaaminen voi tapahtua tietoteknisiä ja sosiaalisia keinoja käyttäen. Huijaajat voivat hyökätä muita pelaajia tai pelin järjestelmiä vastaan, kalastella autentikointitietoja, hyväksikäyttää pelin heikkouksia sekä soveltaa tekoälypohjaista huijausta. Näiden lisäksi huijaajat voivat toimia yhdessä muiden huijaajien kanssa, manipuloida muita käyttäjiä sekä väärinkäyttää virtuaalista omaisuutta. Huijaamisen ehkäisy tapahtuu yleensä parantamalla olemassa olevia järjestelmiä. Hyökkäyksiä voidaan ehkäistä luomalla työkaluja, jotka havaitsevat ja ehkäisevät vahinkoja. Tiedon kalastelua voidaan vähentää muuttamalla sekä salasanakäytänteitä että tiedon siirron menetelmiä. Pelin heikkouksiin liittyvät datan muuntelut, kolmannen osapuolen ohjelmistot sekä pelin kehittäjien ylenkatsomat virheet. Datan käsittelyä voidaan estää sen paremmalla suojaamisella sekä asentamalla ohjelmistoja, jotka havaitsevat ja ilmoittavat datan muuntelusta. Kolmannen osapuolen ohjelmistojen kanssa voidaan käyttää yleistettyjä toteuttamismuotoja sekä hiekkalaatikko- ja hyväksymismenetelmiä. Pelissä esiintyviä suunnitteluvirheitä voidaan paikata päivityksillä. Tekoälyhuijaamiseen estoon voidaan joko implementoida autentikointimenetelmiä peleihin tai käyttää passiivisia havaitsemiskeinoja. Myös sosiaalisia huijausmetodeja estäessä voidaan käyttää tietotekniikkaa. Huijaajien yhteistyötä voidaan valvoa pelaajien käyttäytymistä analysoivilla ohjelmistoilla tai peleissä tapahtuvaa kommunikointia voidaan rajoittaa. Sisäisessä väärinkäytössä voidaan asettaa rangaistuksia ylläpitäjille ja vähentää heidän oikeuksiaan. Pelaajia voidaan myös kouluttaa turvallisuus käytänteistä, jolloin heidän manipulointi hankaloituu. Virtuaalisen omaisuuden suhteen voidaan soveltaa rajoitteita omaisuuden siirtelyssä sekä käyttää kolmansia osapuolia kaupankäynnin turvaamisessa.

Saatujen tulosten perusteella voidaan sanoa, että vaikka huijaaminen onkin jakautunut tietoteknisiin ja sosiaalisiin metodeihin, tietotekniset ehkäisykeinot ovat kuitenkin laajemmin käytettyjä ja niillä voidaankin ehkäistä monia eri huijaamisen muotoja. Pelin kehittäjien tulisi ottaa huijaaminen huomioon jo aikaisessa kehitysvaiheessa, jolloin peliin voidaan sujuvammin luoda järjestelmiä estämään huijaamista. Lisäksi niin pelaajien kuin pelin ylläpitäjien kouluttaminen turvallisuuskäytänteistä ja huijaamisesta sekä vuorovaikutteinen kommunikaatio on hyödyllinen keino turvallisen pelaamisen edesauttamisessa

Sisällysluettelo

1. Johdanto

Verkkopelit ovat tällä hetkellä erittäin suosittuja ja niillä on sekä taloudellisia että teknologisia vaikutuksia nykypäivän maailmaan (Randell & Yan, 2009). Pelkästään vuoden 2014 aikana maailman kymmenen suosituinta verkkopeliä tuottivat lähes miljardi dollaria tuottoa kehittäjilleen, joten kyseessä on taloudellisesti valtava markkina. Verkkopelit ovat tästä johtuen yhä suurempia työllistäjiä. Teknologiselta kannalta verkkopeleihin liittyvät turvallisuusriskit ovat odotettavissa esiintyväksi myös niiden ulkopuolella, kun peleissä esiintyvää teknologiaa aletaan käyttää laajemmin (Hoglund & McGraw, 2007). Verkkopelien turvallisuuden tutkiminen luo näin ollen pohjaa peleihin liittymättömien ohjelmistojen turvallisuuskehitykselle ja on täten tärkeä tutkimuskohde. Tutkimuksessa tarkastellaan aihetta seuraavan tutkimuskysymyksen kautta: **Minkälaisia huijaustapoja huijaajat käyttävät verkkopeleissä ja miten niitä voitaisiin estää?** Kysymystä tutkittiin analysoimalla jo olemassa olevaa kirjallisuutta.

Randell ja Yan (2009) määrittelevät huijaamisen "sellaiseksi käyttäytymiseksi, joita pelaaja käyttää verkkopelissä saadakseen edun muihin pelaajiin nähden tai saavuttaakseen jonkin tavoitteen, joita pelaajan ei pelin ylläpitäjän mukaan tulisi saavuttaa." Verkkopeleissä huijaaminen voi tapahtua sekä tietoteknisiä että sosiaalisia keinoja käyttäen. Huijaajat voivat hyväksikäyttää suunnitteluvirheitä, pinnan alta löytyviä infrastruktuuriongelmia, viattomia pelaajia tai korruptoituneita sisäpiiriläisiä. (Randell & Yan, 2009.) Huijaamista estäessä voidaan paneutua sekä pelaajiin että pelijärjestelmiin. Pelaajia voidaan kouluttaa huijaamisesta ja asettaa rangaistuksia. Tämän lisäksi peleissä käytetty data tulisi pyrkiä suojaamaan erilaisilla protokollilla. (Lan, Zhang & Pin, 2009.) Huijaamisen konkreettisia keinoja ja niiden estämismetodeja on siis tutkittu jo varsin laajasti. Sen sijaan huijaamisen psykologisia vaikutuksia ei tunneta vielä yhtä hyvin, vaikka aiheesta on jonkin verran tutkimusta myös tehty.

Tutkimuksen tavoitteena on saada kuvaa verkkopeleissä huijaamisessa käytetyistä menetelmistä sekä niiden ehkäisykeinoista. Tutkimuksella voidaan mahdollistaa pelien tehokkaampi kehitys niiden turvallisuuden kannalta, eli tutkimuksen kautta pelien kehittäjillä on parempi käsitys siitä, minkälaisia huijaamisesta johtuvia ongelmia on olemassa ja mitä heidän tulisi tehdä välttyäkseen näiltä.

Luvussa kaksi käsitellään erilaisia huijaamisessa käytettyjä menetelmiä. Nämä on jaettu tietoteknisiin ja sosiaalisiin metodeihin ja nämä edelleen pienempiin osiin. Kolmannessa luvussa käydään läpi, miten näitä menetelmiä voidaan estää ja ehkäistä. Neljännessä luvussa tutustutaan huijaamisen psykologisiin piirteisiin ja pyritään luomaan käsitystä siitä, minkälaisia pelaajia huijaajat ovat ja miksi he huijaavat. Luvussa viisi keskustellaan tutkimusongelmasta ja pohditaan esille tullutta tietoa. Lopuksi luvussa kuusi kootaan tulokset yhteen.

2. Huijausmetodeja

Vaikka pelaajilla itsellään olisikin käsitys siitä, mikä on huijaamista, monet silti huijaavat ja pyrkivät oikeuttamaan huijaamisensa. Jos pelaaja on jäänyt jumiin jossain pelin vaiheessa, he voivat usein päätyä huijaamaan. Tällöin he voivat oikeuttaa huijaamistaan käyttämällä apukeinoja vain vaikeuksia aiheuttavan tilanteen läpäisemiseksi, mutta eivät enempää. Tällöin huijaamisen syynä on palauttaa pelaamisesta saatu nautinto. Toisille taas huijaaminen itsessään tuottaa nautintoa; heistä on ”hauska leikkiä Jumalaa”. Pelaajat kuitenkin tähdensivät käyttävänsä huijauksia vain yksin pelattavissa peleissä, pelattuaan pelin jo kertaalleen. Tällöin huijaamisen syynä oli saada pelistä enemmän irti. Lisäksi varsinkin pitkissä peleissä, pelaajat voivat huijata nopeuttaakseen peliä, päästäkseen joistain osista nopeammin ohi. Osille huijaajista huijaaminen ja muiden pelaajien ärsyttäminen tuottavat nautintoa. Tällöin voidaan ajatella, suurimman osan huijaajista ollessa miehiä, että huijaamisella on jonkinlainen valta-aseman ja aggressiivisen maskuliinisuuden osoittamisen ominaisuuksia ja pyrkivät esittämään omaa paremmuuttaan huijaamalla. Toiset oikeuttavat huijaamistaan sillä, että kyseisessä pelissä on muitakin huijaajia, joten oma huijaaminen ainoastaan saattaa itsensä muiden tasolle. Lisäksi joillakin huijaajilla on näkemys, että koska he ovat jo erittäin taitavia kyseisessä pelissä, he ovat ”ansainneet oikeuden huijata”. Tällaiset huijaajat myös painottavat, että vähäisemmän taitotason omaavilla pelaajilla ei olisi oikeutta huijata, koska he voivat vielä kehittää itseään paremmaksi, kun taas taitavimmilla ainoa pelissä oleva haaste on itse pelin muokkaaminen. (Consalvo, 2007.)

Foo ja Koivisto (2003) tutkivat muiden pelaajien kustannuksella tehtyä huijaamista ja laativat sille neliosaisen luokittelun. Huijaajilla voi olla esimerkiksi peliin ja pelaamiseen liittyviä syitä. Koska pelaajat ovat anonyymeja, huijaajat voivat huijata sen takia. He voivat olla tylsistyneitä normaaliin pelaamiseen ja huijaamalla luovat uutta jännitystä. Huijaajat voivat myös huijata ahneuttaan, haluten itselleen muiden pelaajien virtuaalista omaisuutta. Pelissä voi olla myös tapahtunut jotain, jota huijaajat haluavat protestoida huijaamalla. Lisäksi jotkin huijaavat testatakseen itse huijaamista. Huijaamisen aiheuttajana voivat olla muihin pelaajiin liittyvät syyt, jolloin kyse voi olla kostonhalusta tai halusta hyväksikäyttää muiden pelaajien heikkoutta. Huijaajilla voi myös olla muiden huijaajien kanssa ryhmäytymiseen liittyviä haluja, joten huijaamalla yritetään saada nimeä ja mainetta näissä piireissä. Lisäksi huijaajalla voi olla persoonallisia syitä huijaamiselleen. Huijaaminen voi olla tapa purkaa omaa pahaa oloa, saada nautintoa ja tuntea olevansa voimakkaampi kuin yleensä. Myös peliin uppoutuminen voi helpottua joillakin huijaajilla, kun huijaamalla pelihahmon rooliin samaistuminen on entistä mahdollisempaa. Chen, Duh ja Ng (2009) hyödynsivät tätä tutkimusta ja saivat selville, että anonyymisyys ja peleihin uppoutuminen ovat tärkeässä osassa huijaamista. Jos pelaajalla on mahdollisuus pysyä tuntemattomana ja huijaaminen auttaa peliin uppoutumista, huijaaminen kasvaa.

Verkkopeleissä huijaaminen voi tapahtua sekä tietoteknisiä, että sosiaalisia keinoja käyttäen. Huijaajat voivat hyväksikäyttää suunnitteluvirheitä, pinnan alta löytyviä infrastruktuuriongelmia, viattomia pelaajia tai korruptoituneita sisäpiiriläisiä. (Randell & Yan, 2009.)

2.1 Tietotekniset huijausmetodit

Tietotekniset huijausmetodit voivat sisältää hyökkäyksiä, autentikointitietojen kalastelua, pelin heikkouksien hyväksikäyttöä sekä tekoälypohjaista huijausta.

2.1.1 Hyökkäykset

Yksi tapa jakaa tietoteknisiä huijausmetodeja ovat erilaiset hyökkäykset (eng. *attack*) Palvelunestohyökkäykset (eng. *denial of service attack*) voivat kohdistua joko peliä ylläpitävään serveriin tai yksittäisen pelaajan palvelimeen. Serveriin kohdistuva hyökkäys ylikuormittaa sen ja saattaa serverin saavuttamattomaan tilaan. Palvelinhyökkäykset puolestaan estävät yksittäistä pelaajaa yhdistämästä serveriin. (Prashar, Shao, Jiwani, & Malekzadeh, 2009.) Esimerkiksi pelattaessa shakkia, jossa aikarajan ylittävä pelaaja häviää pelin, huijaaja voi palvelunestohyökkäyksen avulla hidastaa vastapelaajan verkkoyhteyttä, jolloin tämä ei ehdi tehdä siirtoaan aikarajan sisällä. (Jeff Yan & Choi, 2002.) Peliserveriin kohdistuvaa palvelunestohyökkäystä kutsutaan yleensä nimellä kohdistettu palvelunestohyökkäys (eng. *distributed denial of service attack*). Tällöin hyökkääjä käyttää suurta määrää hyökkäyksen toteuttavia resursseja, kuten botteja, ja kohdistaa ne peliserveriin. (Loukas & Öke, 2009.) Tällöin hyökkäyksen estäminen on haastavampaa pelin ylläpitäjälle, koska bottien aikaansaama tietoliikenne on valtava ja hyökkäyksen onnistuessa suurempi, kuin mitä peliserveri voi hallita (Jeff Yan & Choi, 2002).

Vaikka serverin saattaminen saavuttamattomaan tilaan hoidetaankin yleensä palvelunestohyökkäyksillä, voidaan näitä edesauttaa muunlaisilla hyökkäyksillä. Puskurin ylivuotovirhe on yksi tällainen metodi. Huijaaja antaa serverille syötteen, joka ylittää serverin puskurin koon. Kun serveri sitten siirtää annetun syötteen puskuriin, saadaan aikaan ylivuotovirhe (Prashar ja muut, 2009). Puskurin ylivuotovirheellä voidaan saada aikaan lukuisia toimintaongelmia pelkän kaatumisen lisäksi. Ohjelmiston muistin käsittelyyn voi tulla virheitä, sen toiminnot voivat johtaa vääriin lopputuloksiin ja järjestelmän suojaus voi murtua täysin. Ylivuotovirheen aiheuttaminen on suosittu hyökkäysmuoto, koska ylivuotoheikkoudet ovat erittäin yleisiä. (Gupta, 2012.) Ylivuotovirheiden yleisyys johtuu myös niiden täydellisen poistamisen vaikeudessa. Vaikka yksittäiset ylivuotovirheet voidaankin helposti korjata, pinnan alla kytevät heikkoudet jäävät yleensä eloon. Ohjelmiston luonnissa käytetty koodi voi olla vanhentunut ja sisältää lukuisia virheitä. Vaikka uusia ohjelmistoja, kuten pelejä, kehitetäänkin turvallisemmin, käytetty koodikieli voi olla turvaton, jolloin yksinkertaiset virheet johtavat vakaviin heikkouksiin. Lisäksi tavat, joilla ylivuotovirheitä pyritään ehkäisemään, eivät ole välttämättä riittäviä. Tarpeeksi usein ei perehdytä hyökkäyksen määränpäähän, eli itse puskuriin. (Cowan, Pu, Maier, Walpole, Bakke, Beattie & Hinton, 1998.)

Toisenlainen hyökkäyksen muoto pyrkii antamaan huijaajalle kyvyn muuttaa joko pelin tietoja, pelin toimimiseen tarvittavia tiedostoja tai saamaan tietoja, joita tämän ei normaalisti tulisi saada ja mahdollisesti myös aiheuttamaan palveluneston. Paketti-injektio hyväksikäyttää kommunikaatiota asiakkaan ja peliserverin välillä. Huijaaja tällöin naamioituu asiakkaaksi ja antaa syötteen (paketin) serverille, joka palauttaa huijaajalle tämän haluaman tiedon (Prashar ja muut, 2009). Serveri ei siis tiedä, onko paketin lähettänyt taho odotettu asiakas vai huijaaja. Paketti-injektioita käytetään, koska joissain tapauksissa ne eivät ole ainoastaan helpompia toteuttaa kuin perinteiset palvelunestohyökkäykset, vaan niitä vastaan puolustautuminen on myös vaikeampaa. (Gu, Liu, Zhu, & Chu, 2005). SQL-injektiossa huijaaja puolestaan syöttää SQL-komentoja peliin, jonka tietokantamanageri prosessoi annetut komennot. Jos peli ei

kuitenkaan validoi komentoja, voi huijaajan antama syöte toimia sellaisenaan ja antaa tälle tietoa pelin tietokannan rakenteesta. Tämän jälkeen huijaaja voi kirjoittaa tarkkoja komentoja, joilla tämä saa haluamiaan tietoja. (Prashar ja muut, 2009.) Syötteiden avulla huijaaja voi esimerkiksi saada muiden pelaajien pelitilien tietoja, kuten salasanoja, tai ohittaa muita olemassa olevia turvallisuusmekanismeja (Boyd & Keromytis, 2004). Wittmannin (2009) mukaan SQL-injektioiden päivittäinen määrä on kasvanut muutamista tuhansista satoihin tuhansiin vuosien 2008 ja 2009 välillä. Hänen mukaan näiden hyökkäysten ongelmana on, ettei niitä vastaan ole yhtä varmaa puolustautumistapaa, sillä ne hyväksikäyttävät huolimatonta ohjelmointia ja heikkoja suunnitteluperiaatteita. Boyd ja Keromytis (2004) myös mainitsevat, että SQL-koodin syöttäminen verkkosovellukseen ei ole erityisen haastavaa henkilölle, joka on perehtynyt asiaan riittävällä tasolla. SQL-injektioiden tapaan huijaaja voi hyväksikäyttää myös DLL-syötteitä (eng. *dynamic linked libraries*). DLL-kirjastot sisältävät dataa, joita ohjelmat käyttävät toimiakseen. Huijaaja ensin selvittää, mitä DLL-kirjastoa peli käyttää ja vaihtaa tämän sellaiseen kirjastoon, joka kaataa järjestelmän suoritettaessa. (Prashar ja muut, 2009)

2.1.2 Autentikointitietojen kalastelu

Pelin tietojen lisäksi huijaajat voivat myös keskittyä saamaan muiden pelaajien autentikointitietoja. Tämä voidaan toteuttaa esimerkiksi pakotushyökkäyksillä (eng. *brute force attack)*. Nämä hyökkäykset perustuvat salasanan kokeiluun, usein jollain ohjelmistolla, kun tiedetään, kuinka monta merkkiä salasana voi sisältää. (Prashar ja muut, 2009.) Pakotushyökkäyksillä on kaksi päämuotoa, kytkeytynyt (eng. *online*) ja yhteydetön (eng. *offline*). Kytkeytyneessä muodossa hyökkääjä joutuu käyttämään samaa käyttöliittymää kirjautumiseen kuin mikä itse ohjelmistossa on olemassa, esimerkiksi pelialustan kirjautumisikkunaa. Yhteydetön hyökkäys vaatii hyökkääjää ensin varastamaan salasanatiedoston, mutta tämän jälkeen hän voi arvata salasanoja rajattomasti, ilman ohjelmiston tai verkon asettamia rajoitteita. (Be'ery, 2014.) Yksinkertaisin kytkeytynyt pakotushyökkäys kohdistuu yksittäiseen käyttäjätiliin, jonka salasanan hyökkääjä pyrkii saamaan. Hyökkäys voidaan myös kohdistaa useampaan tiliin kerralla, jolloin hyökkääjä jakaa salasanan "arvailun" useaan käyttäjätiliin samaan aikaan. Miljoonan yhteen tiliin kohdistuvan kokeilun sijaan hyökkääjä lähettää yhden kokeilun kerralla miljoonaan eri tiliin. Tällainen hyökkäys on vaikeampi havaita, koska mihinkään yhteen tiliin ei kohdistu epäluonnollista määrää liikennettä, mutta se myös vaatii hyökkääjää tuntemaan ison määrän käyttäjätilejä. Hyökkääjä ei myöskään voi tietää, minkä tilin salasanan hän saa selville. Yhteydettömät hyökkäykset puolestaan ovat vaarallisia, koska salatun salasanatiedoston saatuaan hyökkääjä voi kohdistaa hyökkäyksensä kaikkia käyttäjätilejä vastaan. (Florêncio & Herley, 2010.)

Salasanat ovat myös alttiita haistelulle (eng. *sniffing*) ja urkkimiselle (eng. *snooping*). Haistelulla tarkoitetaan autentikointitietojen keräämistä jotain kommunikointimediaa tarkkailemalla. Haistelu on tehokasta, koska jos autentikointitietoja lähetetään salaamatta, on niiden hyväksikäyttäminen erittäin helppoa. Vaikka tiedot olisikin salattu, voi huijaaja hyödyntää pakotushyökkäyksiä niiden murtamiseen. (Pipkin, 2003.) Urkkiminen puolestaan on toisen käyttäjän toimintojen tarkkailua. Huijaaja voi esimerkiksi vakoilla sähköisesti toisen pelaajan näppäimistön toimintoja, saaden täten selville minkä salasanan kyseinen pelaaja syötti. Tämä voi tapahtua jonkinlaisen huijausohjelmiston avulla, joka sieppaa ja esittää haluttua dataa, tai käyttämällä elektronisia laitteita, jotka lukevat ja tulkitsevat väylien ja porttien välityksellä kulkevaa dataa. (Heinrich, Le, Waldorf & Angelo, 2001.). Vaikka nämä tiedot lähetettäisiinkin edelleen salattuina, ei salaamisesta ole hyötyä, koska hyökkääjä saa tiedot ennen niiden lähettämistä (Pipkin, 2003).

Näiden hyökkäysten lisäksi huijaaja voi hyödyntää pelissä jo olemassa olevaa heikkoa autentikointia. Pelaajien tulee yleensä identifioida itsensä käyttääkseen verkkoon yhteydessä olevia peliservereitä, joten ainutlaatuinen tunniste on tärkeä. Huijaaja voi tätä hyväksikäyttääkseen luoda huijausserverin, jonka avulla huijaaja voi saada käsiinsä toisen pelaajan tietoja. Normaalien autentikointitietojen lisäksi pelin ostamisen takaava avainkoodi on tällöin vaarassa, jonka saatuaan huijaaja voi esimerkiksi myydä sen edelleen. Lisäksi yksitasoista autentikointia voidaan myös hyväksikäyttää. Jos esimerkiksi huijaaja pääsee toisen pelaajan pelissä käytettävään tiliin käsiksi, hän voi vaihtaa tämän salasanan ilman että peli vaatisi toisenlaista autentikointia. (Mørch, 2003.)

2.1.3 Pelin heikkouksien hyväksikäyttäminen

Heikko autentikointi on kuitenkin vain yksi esimerkki tapauksesta, jossa huijaaja hyväksikäyttää pelin suunnitteluvirheitä. Pelin kehittäjä voi luottaa liialti omaan kehittämäänsä asiakassovellukseen (eng. *game client*) ja täten tietämättään luoda tilanteen, jossa huijaaja voi muokata pelin sisäisiä tietoja. Chen ja Duh (2003) kutsuvat tätä ”väärinsijoitetuksi luottamukseksi”. Pelissä käytettyä tietoa voidaan muuttaa staattisesti tai dynaamisesti. Staattinen metodi muuttaa pelin tietoja huijauksen ajaksi ja tämän jälkeen muuttaa tiedot ennalleen. Huijaaja voi esimerkiksi muuttaa pelissä voimassa olevaa painovoima-asetusta ja mahdollistaa epäluonnollisen liikkumisen pelimaailmassa, luoden epäreilun kilpailutilanteen. Staattinen muokkaus on kuitenkin helpompi huomata, koska se hyödyntää selvästi vääriä tietoja pelin normaaleihin asetuksiin verrattuna. Dynaamisessa metodissa huijaaja käyttää hyväksyttyjä tietoja, mutta vaihtelee niiden arvoja. Huijaaja voi esimerkiksi pelata joukkuepohjaista toimintapeliä ja vaihdella pelin tietoja, jotka määräävät mihin joukkueeseen huijaaja kuuluu. Huijaaja voi hyväksikäyttää tätä muun muassa saamalla tarkkaa tietoa muiden pelaajien olinpaikasta kullakin ajanhetkellä. Kyseiset tietojen muokkaukset ovat suosittuja, koska niiden toteuttaminen on erittäin yksinkertaista. (Feng ja muut, 2008.) Asiakassovellukseen ei täten yleensä kannata luottaa liikaa, koska huijaajilla on sovelluksen ladattuaan täysi vapaus muokata sen tietoja. (Randell & Yan, 2009). Asiakassovelluksiin liittyy myös verkkopeleissä esiintyvä asiakas-serveri-asiakas vuorovaikutus (eng. *client-server-client interaction*). Tällöin käyttäjän lähettämä viesti kulkee asiakassovelluksen kautta peliserveriin ja lopulta toisen käyttäjän sovellukseen. Vaikka tämä kommunikaatioväylä onkin mahdollista suunnitella turvalliseksi, on sen kautta silti mahdollista hyökätä toista käyttäjää vastaan. (Bono ja muut, 2009).

Jotkin verkkopelit käyttävät vertaisverkkoa tiedonsiirrossa. Tällöin tieto liikkuu suoraan pelaajan koneelta toiselle, sivuuttaen peliserverin täysin. Vertaisverkkoa voidaan käyttää lievittämään stressiä varsinaisilta peliservereiltä ja vähentämään datan siirrossa tapahtuvaa viivettä, mikä edesauttaa joidenkin ominaisuuksien toimintaa. Joissain peleissä esimeriksi ääniviestintä pelaajien välillä suoritetaan vertaisverkolla edellä mainituista syistä. Tämä kuitenkin tarkoittaa, että huijaaja voi siirtää haitallisia datapaketteja ilman, että peliserveri on niistä tietoinen. Usein kyseessä on vielä tilanne, jossa vertaisverkkoyhteys on automaattinen, eikä pelaajilla ole mahdollisuutta kieltäytyä sen käytöstä. Tällöin pelkästään huijaajan ulottuville joutuminen voi riittää viattoman pelaajan hyväksikäyttämisen. (Bethea, Cochran, & Reiter, 2011.) Vertaisverkko on verkkopeleissä huijaamisen kannalta siis hyödyllinen, koska huijaaja on suoraan yhteydessä muihin pelaajiin ja voi täten helpommin aiheuttaa vahinkoa haitallisella datalla. Myös Bono ja muut (2009) mainitsevat, kuinka verkkopeleissä usein esiintyvä valtava määrä tietoa siirretään usein erikseen hyväksytyillä vertaisverkkoyhteyksillä.

Mitä enemmän erilaisia tiedostoformaatteja peli sallii, sitä enemmän hyväksikäytettäviä osia pelissä esiintyy. Eri tiedostoformaatteja käytetään käsittelemään erilaista tietoa

pelin sisällä, kuten ääntä, videota ja skriptejä. Vaikka näillä onkin mahdollista rikastaa pelikokemusta, ne kuitenkin luovat uusia mahdollisuuksia huijaajille löytää heikkouksia. Näitä tiedostoja suoritetaan erillisillä kirjastoilla, mutta vaikka nämä ovat yleensä standardoituja ja täten turvallisempia kuin puhtaalta pöydältä luodut kirjastot, niistäkin voi löytää heikkouksia. Lisäksi heikkouksia löydettäessä on usein mahdollista, että tarvittavia korjaustoimenpiteitä ei tehdä pitkään aikaan vikojen löytymisestä. Huijaajat täten löytävät ohjelmistoja, jotka käyttävät vanhentuneita ja haavoittuvia kirjastoja, ja hyväksikäyttävät niitä. Näennäisesti pienienkin tiedostoformaattien korjaamisella voi olla suuria vaikutuksia. (Bono, Caselden, Landau, & Miller, 2009.)

Pelin mekaniikkojen hyväksikäyttäminen voidaan myös katsoa huijaamiseksi, varsinkin kun kyseessä on pelin ylläpitäjän ylitsekatsoma ominaisuus. Yan (2009) esittelee tilanteen verkossa pelattavasta Bridge-korttipelistä. Kun huijaaja pelin kuluessa huomaa, että hän on häviämässä pelin, hän katkaisee yhteyden peliin. Jos pelissä on jonkinlainen sijoitusjärjestelmä, jossa pelejä voittamalla kerätään pisteitä, voi huijaaja pelistä poistumalla estää menettämästä pisteitä tai antamasta pisteitä vastustajalleen. Koska huijaaja hyödyntää pelin sisäisiä suunnitteluvirheitä, ei häneltä vaadita omaa teknistä osaamista (Randell & Yan, 2009). Vastaavia tapauksia voidaan löytää monenlaisista verkkopeleistä; toimintapeleissä peliin voi esimerkiksi tulla tekstuuriongelma, jolloin pelaaja voi nähdä niiden läpi. Heikkouksia voi nousta esiin myös varsinaisen pelin ominaisuuksien ulkopuolella, kolmannen osapuolen tekemien ohjelmistojen kautta. Jos pelissä ei esimerkiksi ole ääniviestintämahdollisuutta, pelaajat saattavat toteuttaa ohjelmiston, joka toimii pelin sisällä ja mahdollistaa tämän ominaisuuden. Kun nämä sovellukset sitten kasvattavat suosiotaan, huijaajat voivat pyrkiä saamaan käsiinsä muiden pelaajien informaatiota ujuttamalla haittaohjelmia sovelluksen asennustiedostoihin Koska pelin kehittäjä ei ole yleensä millään tavalla liitoksissa kolmannen osapuolten sovelluksiin, on kehittäjien vaikea korjata niiden aiheuttamia turvallisuusaukkoja; monesti ainoa mahdollisuus on sovelluksen käytön kieltäminen kokonaan, mikä voi johtaa epämiellyttävään reaktioon pelaajien keskuudessa, varsinkin jos kyseinen sovellus on laajalti käytössä (Bono ja muut, 2009.)

2.1.4 Tekoälypohjainen huijaus

Tekoälypohjainen huijaus ns. bottien avulla on kasvanut valtavasti viime vuosina ja se on sekä yleisin että vaikeimmin taltutettava huijauksen muoto. Botteja voidaan käyttää joko tilanteissa, joissa botti suorittaa jonkin pelissä tapahtuvan toiminnon ihmispelaajaa paremmin tai tapauksissa, joissa bottien käytöllä huijaaja voi saada pelinsisäisiä resursseja joutumatta itse tekemään vaadittua työtä. (Gianvecchio, Wu, Xie & Lang, 2009.) Toimintapeleissä botti voidaan ohjelmoida tähtäämään automaattisesti vastapelaajaan, jolloin huijaajan ei tarvitse kehittää omia pelinsisäisiä taitojaan, kuten käden ja silmän välistä koordinaatiota. Roolipeleissä puolestaan pelaajia vaaditaan panostamaan paljon aikaa ja suorittamaan itseään toistavia tehtäviä kehittääkseen pelihahmoaan ja kerätäkseen virtuaalista omaisuutta. Huijaaja voi bottien avulla ohittaa nämä toimet ja parantaa pelihahmoaan käyttämättä siihen vaadittua aikaa. (Golle & Duchenaut, 2005.) Tietokoneshakkiin liittyvässä tutkimuksessa on onnistuttu kehittämään ohjelmia, jotka pystyvät kilpailemaan jopa maailman parhaimpia shakin pelaajia vastaan (Randell & Yan, 2009). Botteja käyttämällä huijaaja voi siis saada aikaan epätasapainoa pelissä käytettyjen hahmojen välillä, jos botin avulla saavutetaan epäreilu kilpailuasetelma. Lisäksi virtuaalisen omaisuuden epänormaali hankkiminen voi johtaa inflaatioon varsinkin peleissä, joissa käydään kauppaa jollain pelinsisäisellä rahayksiköllä. (Gianvecchio ja muut, 2009.)

Boteilla ja ihmisillä on monenlaisia eroja pelien sisällä. Verkossa pelattavissa massiivisissa roolipeleissä pelaajille on tarjolla paljon erilaisia aktiviteetteja, kuten

tarinaa kertovia tehtäviä, hirviöitä vastaan taistelua ja muiden pelaajien kanssa kommunikointia. Koska jokainen pelinsisäinen aktiviteetti vaatii pelaajalta toisistaan erovia toimintoja, voidaan olettaa, että ihmisen käyttäytymisessä ja saaduissa syötteissä havaitaan selkeitä eroja erilaisten toimintojen välillä. Botit toimivat ilman ihmisen väliintuloa, toistaen yksinkertaisia tehtäviä, kuten hirviöiden tappamista, ja niiden käyttäytymisen voidaan ajatella olevan säännönmukaisempaa, paljastaen toistuvia malleja ja vähemmän variaatioita. Käyttäytymisen lisäksi vuorovaikutus pelin kanssa eroaa selvästi bottien ja ihmisten välillä, siitä huolimatta, että molemmat käyttävät näppäimistöä ja hiirtä. Ihmiset havaitsevat pelin graafisen tulosteen optisesti ja syöttävät vaaditut komennot peliin käyttämällä fyysisesti tähän tarkoitukseen laadittuja laitteita, kuten hiirtä ja näppäimistöä. Boteilla ei luonnollisesti ole tietokoneohjelmina mitään käsitystä näöstä ja niitä eivät sido mekaaniset fysiikat. Vaikka boteilla onkin mahdollista analysoida pelin grafiikoita, on se kuitenkin laskennallisesti kallista. Välttääkseen laskennasta johtuvia kuluja, botti pyrkii saamaan tarvitsemansa tiedon, kuten pelihahmojen sijainnit tai niiden ominaisuudet, lukemalla peliohjelman muistia. Botit ohjaavat pelihahmoja simuloimalla erilaisten ohjaimien syötteitä, kuten hiiren kursorin sijaintia tai näppäimistön painikkeen painallusta. Bottien käyttämät metodit ovat usein karkeita, mutta toimivia. Esimerkiksi siirtääkseen pelihahmoaan paikasta toiseen, botti tarvitsee vain kaksi koordinaattia, pelihahmon ja määränpään sijainnin. Botti pyrkii saavuttamaan kohteen liikuttamalla hahmoa eri suuntiin ja tarkistaa etenemisensä vertaamalla koordinaatteja. Jos pelihahmon paikka ei muutu tietyn ajan sisällä, botti olettaa tiellä olevan jonkinlainen este, jonka se pyrkii kiertämään eri liikekomennoilla. Ihminen toteuttaisi vastaavan toiminnon luonnollisesti tulkitsemalla näytöllä näkyvää peligrafiikkaa ja havainnoimalla pelihahmoa ympäröivää maastoa. Botilla on myös mahdollisuus selviytyä tilanteista, joissa grafiikan havainnoinnista olisi hyötyä, esimerkiksi pelihahmon poimiessa maasta jonkin objektin. Yleensä vastaavissa roolipeleissä peli kuvastaa jonkinlaisen informaatioikkunan, kun hiiren kursori asetetaan poimittavan objektin päälle. Botti voi täten liikutella kursoria ja havaita eron pikselien värissä siinä paikassa, johon pelin käyttöliittymä sijoittaa kyseisen ikkunan. (Gianvecchio ja muut, 2009.)

Pelien ja bottien kehittäjille on vuosien saatossa muodostunut eräänlainen kilpajuoksuasetelma. Aluksi bottien valmistavat muokkaavat pelialustaansa niin, että se tukee bottien käyttöä. Tämän jälkeen bottien valmistajat kehittävät kyseisellä ajanhetkellä toimivia botteja, joita pelien kehittäjät taltuttavat päivittämällä pelialustaa. Tämän jälkeen botteja uudistetaan uuden alustan mukaiseksi ja niin edelleen. Tämä on kuitenkin työläämpää bottien kehittäjille, koska pelialustojen jatkuva monimutkaistuminen hankaloittaa erikseen räätälöidyn pelialustan ylläpitoa. Tämä on kuitenkin johtanut uudenlaisen bottityypin syntyyn, joka ihmisen tavoin lukee näytöllä tapahtuvia asioita ja käyttää hiirtä ja näppäimistöä. Toisin sanoen, sen sijaan että huijaaja käyttäisi omanlaistaan pelialustaa, he toimivat kaikilla pelaajilla käytössä olevan alustan kautta, jonka välityksellä nämä kehittyneet botit lähettävät hiiren ja näppäimistön dataa ja lukevat sekä ruudun pikseleitä että mahdollisesti peliohjelmiston muistiosoitteen tiettyjä alueita. Lisäksi näillä boteilla on yleensä ominaisuuksia, jotka mahdollistavat niiden nopean päivittämisen ja sopeutumisen uuteen peliympäristöön. (Gianvecchio ja muut, 2009.) Tekoälypohjaisella huijauksella on kuitenkin rajoitteita. Jos pelissä käytettävät toiminnot ovat mahdotonta tai vaikeaa kuvata tietokoneohjelmalla tai kyseisiin peleihin liittyvä tekoälytutkimus ei ole riittävällä tasolla, voi tarvittavien huijausohjelmien luominen olla vaikeaa (Randell & Yan, 2009).

2.2 Sosiaaliset huijausmetodit

Huijaajat ovat sosiaalisia pelaajia ja he käyttävät enemmän aikaa peleissä, joissa on mahdollisuus pelata muiden pelaajien kanssa. Tämä näkyy niin huijaajien ostamissa

kuin pelaamissakin peleissä. Lisäksi tutkimus havainnollistaa huijaajien sosiaalisia kytköksiä osoittamalla, että huijaajilla on suurempi mahdollisuus ystävystyä muiden huijaajien kanssa. Noin 70 % tutkimuksen kohteena olleilla rehellisillä pelaajilla ei ollut yhtään huijaavaa ystävää, vastaavalla määrällä huijaajia ainakin 10 % heidän ystävistään myös huijaavat. Lisäksi huijaajat ovat todennäköisempiä verkostoitumaan maantieteellisesti läheisten huijaajien kanssa. Voidaan siis ajatella, että vaikka huijaaminen onkin epäsosiaalista toimintaa, huijaajat itse osallistuvat kuitenkin aktiivisesti sosialisointiin (Blackburn, Kourtellis, Skvoretz, Ripeanu, & Iamnitchi, 2014). Tietyissä peleissä huijaamiselle omistetuilla keskustelupalstoilla huijaajat vaihtavat kokemuksiaan ja tarpeen tullessa, esimerkiksi pelin ylläpitäjän alkaessa löytää huijaajia, toimivat yhdessä löytääkseen uusia keinoja kiertää ylläpitäjän vastatoimet. Tibia-peliin liittyvällä huijausfoorumilla huijaajat pyrkivät ratkaisemaan, miten pelin kehittäjän implementoima automaattinen huijauksenesto-ohjelma toimi ja miten parantaa huijauksessa käytettäviä ohjelmistoja. Verkkopeleissä huijaaminen on usein oppimistilanne, jossa huijaajat oppivat yhdessä kuinka käyttää eri metodeja. (De Paoli & Kerr, 2009.)

Blackburnin ja muiden (2014) mukaan huijaaminen voi levitä sosiaalisten suhteiden kautta jo kauan ennen itse huijaamisen alkamista. Itse huijaaminen voi tuottaa samanlaisia reaktioita kuin tartuntataudit. Kun joku pelaaja merkitään huijariksi, muut pyrkivät katkaisemaan siteitään tämän kanssa. Tämän takia huijaajat myös usein pyrkivät naamioitumaan, jos heidät on käräytetty huijaamisesta, esimerkiksi luomalla uusia käyttäjätilejä. Kuten tietotekninen huijaaminen, myös sosiaalisessa huijaamisessa on löydettävissä erilaisia yleisesti käytettyjä metodeja. Sosiaalinen huijaaminen perustuu joko useiden huijaajien yhteistyöhön tai muiden pelaajien hyväksikäyttöön. Lisäksi huijaajat voivat ostaa etuja itselleen.

2.2.1 Kolluusio

Kolluusiolla tarkoitetaan kahden tai useamman huijaajan yhteistyötä, tarkoituksenaan saada epäreilu etu muihin pelaajiin nähden. Kolluusio voidaan luokitella huijaajien yhteistyösopimuksen muodon avulla, jolloin tarkastellaan kahta kategoriaa. Ensimmäinen kategoria perustuu kolluusion tasoon. Huijaajilla voi olla selkeä salainen sopimus yhteistyöstä, joka on sovittu jonkin kommunikointikanavan kautta. Heillä voi olla myös sanaton sopimus, eli vaikka yhteistyöstä ei olisikaan sovittu, he pyrkivät saavuttamaan kumpaakin hyödyntävän tilanteen, esimerkiksi eliminoimaan parhaan pelaajan pelistä. Lisäksi kolluusio voi olla rajoitettu tiettyihin päätöksiin, mutta niiden ulkopuolella yhteistyön osapuolet pelaavat toisiaan vastaan normaalisti. Toinen kategoria käsittelee yhteistyön sisältöä. Huijaajat voivat muuttaa pelityyliään riippuen vastustajastaan, eli pelata normaalisti muita pelaajia vastaan ja välttää vahinkoa yhteistyökumppaneitaan kohtaan. Turnaustilanteessa kolluusiossa olevat huijaajat voivat tarkoituksella eliminoida itsensä, jos se parantaa toisen huijaajan sijoitusta. (Smed, Knuutila & Hakonen, 2006.) Peleissä, joissa käytetään sijoitusjärjestelmää ja pelin voittaja etenee tilastoissa, kaksi huijaaja voivat pelata toisiaan vastaan useaan kertaan ja vuorotella pelin voittajan suhteen (eng. *win-trading*), pelaamatta peliä kuten se on tarkoitettu. Näin molemmat vuoron perää nousevat tilastoissa pelaamatta ainuttakaan oikeaa peliä. Tästä seuraa, että käytettävät tilastot muuttuvat epätarkoiksi, eivätkä pelaajat voi tietää, onko listan kärkipäässä oleva henkilö ansainnut paikkansa vai onko kyseessä huijaamalla saavutettu taso. (Yan, 2003.) Huijaajat voivat yrittää saattaa viattoman pelaajan käyttökieltoon lähettämällä pelin ylläpitäjälle näennäisesti itsenäisiä valituksia kyseisestä pelaajasta. Huijaaja voi myös saada apua asiantuntijalta, joka ei välttämättä ole pelin osanottaja. Shakkia pelatessa huijaaja voi kääntyä joko kokeneen ihmispelaajan tai tietokoneohjelman puoleen, joka tekee siirrot huijaajan puolesta. Vastaavan tietämyksen jakamisen lisäksi huijaajat voivat vaihtaa

pelitilanteeseen liittyvää informaatiota. Toimintapelissä huijaajalla voi olla peliä seuraava apuri, joka kertoo vihollisjoukkojen sijainnit. (Smed ja muut, 2006.) Bridge-korttipelissä kaksi pelaajaa voivat kertoa toisilleen tietoja omista korteistaan ja näin saada selvemmän käsityksen pelin tilanteesta. Kaikissa epätäydelliseen tietoon perustuvassa pelissä, eli kun pelaajilla ei ole kaikkea tietoa saatavilla, tämän tyyppinen huijaaminen on erittäin tuhoisaa, koska se saattaa pelitilanteen epätasapainoon. Normaalisti kyseisiä pelejä pelaavat pelaajat joutuisivat toteuttamaan siirtonsa ja liikkeensä epävarmemmin, mutta tiedon jakaminen luo suuren etulyöntiaseman huijaajille. (Yan, 2003.) Asiantuntija-apuun liittyvä kolluusio voi tapahtua myös sisäisen väärinkäytön avulla. Sisäiseen väärinkäyttöön liittyy joku pelin tarjoaja, jolla on pääsy pelin hallintaan liittyviin ominaisuuksiin. Tässä tapauksessa huijaaja voi liittoutua kyseisen sisäpiiriläisen kanssa, joka voi muokata pelin ominaisuuksia huijaajan haluamalla tavalla, esimerkiksi luomalla tehokkaampia aseita tai hahmoja. (Randell & Yan, 2009.) Huijaajat voivat myös jakaa pelinsisäisiä resursseja keskenään, kuten virtuaalista omaisuutta, sekä manipuloida kyseisten resurssien hintoja, jolloin normaalit pelaajat eivät pääse niihin käsiksi. (Smed ja muut, 2006).

Kolluusion luokittelussa voidaan myös perehtyä siinä esiintyviin rooleihin, huijauksen päämärän sijaan. Tällöin on huomioitava pelin pelaajien ja osanottajien eroavaisuus ja kuinka näiden lukumäärä ei välttämättä täsmää. Kaksi pelaajaa voivat esimerkiksi vuorotella pelihahmon ohjaamisessa, jolloin pelin osanottajia on useampi kuin pelissä ilmenevien hahmojen määrä. Henkilö voi myös osallistua katsojana, jos pelissä on tähän mahdollisuus. Lisäksi mahdollisuus tietokoneohjelmaan, joka imitoi oikeaa pelaajaa, on olemassa. Peliä pelaavat voivat siis itsenäisesti vaikuttaa pelin tilaan, kun taas osanottajat vaativat pelaajan toimintoja. Ensimmäinen rooleihin pohjautuva kategoria liittyy pelaajan identiteettiin. Pelaajaa voidaan ohjata useamman kuin yhden henkilön toimesta tai yksi henkilö voi ohjata useampaa kuin yhtä pelaajaa. Toinen kategoria liittyy erilaisiin osanottajien muotoihin. Pelin katsoja voi antaa informaatiota pelaajalle, pelaajalla voi olla asiantuntija-avustaja tai pelaajat voivat toimia samoja intressejä kohti. Viimeinen kategoria perustuu pelitilanteisiin. Verkkoroolipeleissä useaa itsenäistä peliään pelaavat henkilöt voivat jakaa tietoa, esimerkiksi pelimaailman eri alueista. Myös sisäinen väärinkäyttö voidaan ajatella kuuluvaksi tähän kategoriaan, varsinkin kun väärinkäyttö on tahatonta. Pelaaja voi esimerkiksi ottaa yhteyttä pelin tukihenkilöön, joka vahingossa paljastaa pelaajaa hyödyntävää informaatiota. (Smed ja muut, 2006.)

2.2.2 Käyttäjän manipulaatio ja virtuaalinen omaisuus

Käyttäjän manipulaatiolla (eng. *social engineering*) tietotekniikassa tarkoitetaan ”ei-teknistä, ihmisten väliseen vuorovaikutukseen pohjautuvaa tungettelua, jonka tarkoitus yleensä on saada toinen henkilö rikkomaan normaaleja turvallisuuskäytäntöjä” (Chen, 2006). Toisin sanoen, huijaaja pyrkii saamaan toiselta pelaajalta tietoja, joita tämän ei normaalissa tilanteessa tulisi saada, ilman, että hän käyttää siihen tietoteknisiä metodeja. Vaikka kokeneille pelaajille pelkkä salasanan pyytäminen voikin kuulostaa triviaalilta, on tämä kuitenkin yllättävän tehokas tapa saada autentikointitietoja (Pipkin, 2003). Huijaaja voi uskotella, että toiselle pelaajalle on tapahtunut jotain ikävää, kuten pelaajan käyttämässä tilissä on jonkinlainen ongelma, ja ongelman ratkaisuun vaaditaan pelaajan kirjautumistietoja. Huijaajat pyrkivät yleensä tähän esiintymällä pelin kehittäjänä, lähettämällä viralliselta vaikuttavia sähköposteja tai pelinsisäisiä viestejä. (Yan, 2005.) Huijaaja voi myös manipuloida toista pelaajaa muuttamaan salasanansa huijaajan viestissään määrittelemäksi tai ajamaan haittaohjelman, joka kysyy käyttäjän kirjautumistietoja. Tämän jälkeen huijaaja voi kerätä saadut tiedot ja saada pääsyn toisen pelaajan pelitiliin. (Pipkin, 2003.)

Varsinkin roolipeleissä pelaajilla on mahdollisuus edistyä pelatessaan erilaisilla tavoilla. Pelaajan hahmo kehittyy pelatessa ja hahmolle voi kerätä omaisuutta, kuten aseita. Mitä pidempään pelaaja pelaa ja hahmo kehittyy, sitä vahvempi hahmosta myös tulee. Sen sijaan, että pelaaja käyttäisi aikaa pelin pelaamiseen ja opetteluun, hän voi huijata. Kuten Chen (2009) mainitsee, erilaisilta huutokauppasivuilta, kuten eBay, pelaaja voi yleensä löytää haluamansa pelinsisäiset tavarat tai jo valmiiksi kehitetyt hahmot. Hän voi myös halutessaan myydä nämä tavarat eteenpäin ja näin ollen hyötyä rahallisesti. Vastaavissa roolipeleissä huijaamisen syyt ovat täten usein virtuaaliseen omaisuuteen kytkettäviä ja niillä tehtävä kaupankäynti on suuri houkutin uusille huijaajille. Huijaajat myös kokevat, että näissä peleissä olevat turvallisuuspuutteet edesauttavat huijaamista ja siitä hyötymistä (Chen ja muut, 2005). Täten huijaamisesta voidaan myös luoda bisnes virtuaalisten maailmoiden ulkopuolella. Kuten De Paoli ja Kerr (2009) kirjoittavat, tekoälypohjaisessa huijauksessa käytettäviä botteja voi ostaa itselleen useista eri niitä myyvistä yhtiöistä. Samalla tavoin kuin perinteiset ohjelmistoyritykset myyvät lisenssejä tuotteisiinsa, myös nämä yhtiöt myyvät bottejaan pelaajien käyttöön. Kun pelin ylläpitäjät kehittävät vastatoimia, pyrkivät huijausohjelmistojen valmistavat päivittämään omia tuotteitaan. Näiden yhtiöiden foorumeilla käydyissä keskusteluissa huomataan, kuinka vastavuoroinen kokemus huijaaminen voi olla. Huijausfirmat pyytävät palautetta käyttäjiltä ja keräävät tietoa, joilla luoda entistä parempia ohjelmistoja. Huijaajat luonnollisesti haluavat ohjelmistojen parantuvan, jotta heidän havittelemansa etulyöntiasema pelissä säilyisi, kun taas huijausfirmalle viallinen tuote tarkoittaa taloudellista tappiota. Käytännössä kyseiset firmat ovat olemassa, koska pelikin on.

Kuten nähdään, useimmat tietotekniset huijausmetodit perustuvat pelin eri heikkouksien hyväksikäyttämiseen, niiden luokittelusta huolimatta. Hyökkäykset ja autentikointitietojen murtamiset pakottamalla hyödyntävät pelin kyvyttömyyttä vastaanottaa ja käsitellä suuria datamääriä tai heikkoa suunnittelua, mikä mahdollistaa haitallisen datan ja komentojen syöttämisen. Suunnitteluvirheet ovat havaittavissa myös esimerkiksi erilaisissa kommunikointitilanteissa, kuten asiakas-serveri tai vertaisverkko, jolloin pelin kehittäjä ei ole pystynyt luomaan turvallista asetelmaa. Asiakassovelluksen, pelin ominaisuuksien ja tiedostoformaattien hyväksikäyttäminen pohjautuvat niin ikään pelin kehittäjän ylenkatsomiin tapauksiin, joiden korjaaminen ei usein vaadi suurta työmäärää. Tekoälypohjainen huijaus on puolestaan vaikeampi korjata, joten se onkin suosittu huijausmuoto. Sen luomat mahdollisuudet, kuten omien kykyjen keinotekoinen parantaminen ja nopeampi edistyminen, ovat varsin uniikkeja muihin huijausmetodeihin verrattuna. Sosiaaliset huijausmetodit puolestaan pyrkivät hyväksikäyttämään sekä kanssapelaajia että käytettyjen metodien havaitsemisen vaikeutta, kuten kolluusiotapauksissa. Lisäksi peleihin luotu virtuaalinen omaisuus on kehittynyt itsenäiseksi huijaamisen osa-alueeksi, jolloin huijaajat voivat konkreettisesti ostaa etuja itselleen.

3. Huijaamisen ehkäisy

Kuten Randell ja Yan (2009) kirjoittavat, verkkopelit ovat merkittäviä niin taloudellisesti kuin teknologisesti. Taloudellisesti suosituimmat verkkopelit tuottavat miljardeja dollareita ja työllistävät entistä enemmän ihmisiä. Myös Gianvecchio ja muut (2009) mainitsevat, että verkkopeleillä on suuri taloudellinen ja sosiaalinen asema nykymaailmassa. Lisäksi verkkopeleillä on vaikutusta muiden ohjelmistojen turvallisuuteen (Hoglund & McGraw, 2007). Koska huijaaminen vaikuttaa negatiivisesti kaikkiin edellä mainittuihin asioihin, on huijaamisen ehkäisy tärkeää. Huijaamista estäessä voidaan paneutua sekä pelaajiin, että pelijärjestelmiin. Pelaajia voidaan kouluttaa huijaamisesta ja asettaa rangaistuksia. Tämän lisäksi peleissä käytetty data tulisi pyrkiä suojaamaan erilaisilla protokollilla. (Lan, Zhang & Pin, 2009)

3.1 Hyökkäysten ehkäisy

Koska palvelunestohyökkäyksiä on monenlaisia, myös niitä vastaan puolustautumiseen on esitetty monia erilaisia ehdotuksia. Täysvaltaiseen ratkaisuun kuuluu kuitenkin yleensä kolme pääosa-aluetta. Hyökkäyksen olemassaolo tulee ensin havaita. Järjestelmä voi joko huomata poikkeuksellisen käyttäytymisen joissakin sen asiakkaista tai tunnistaa tunnettujen hyökkäysten luonteenomaiset piirteet. Havainnointikeinoina voidaan käyttää esimerkiksi verkkoliikenteen tilastollisten ominaisuuksien analysointia tai sähköisiä agentteja, jotka valvovat verkkoliikenteessä ilmeneviä IP-osoitteita. Järjestelmän tulisi myös luokitella saapuvat datapaketit päteviin ja epäkelpoihin; tässä luokittelussa voidaan käyttää hyökkäysten havainnoinnissa käytettyjä periaatteita. Lopuksi tarvitaan vastatoimi, joka voi tarkoittaa epäkelpojen datapakettien hylkäämistä tai niiden ohjaamista ansaan, jossa niitä voidaan tutkia ja analysoida. (Loukas & Öke, 2009.)

Prasharin, Shaon, Jiwanin ja Malekzadehin (2009) toteuttamassa tutkimuksessa Terra-peliin suunnatut palvelunestohyökkäykset johtivat käyttökieltoihin. He kuitenkin arvioivat, että pelkkä käyttökieltojen jako ei ole tehokkain ratkaisu, sillä IP-osotteisiin suunnatut kiellot eivät ole välttämättä riittävän tarkkoja ja viattomia pelaajia voi joutua kieltojen alle. Sen sijaan he suosittelevat parantamaan sekä pelitilien suojausta, jolloin kaapattuja tilejä ei voida käyttää palvelunestohyökkäysten toteuttajina, että itse järjestelmän suojauksen parantamista, jolloin pelin ylläpitäjien ei tarvitse huolehtia viattomien pelaajien saattamisesta käyttökieltoon.

Hyökkäysten ehkäisyyn voidaan soveltaa niitä varten spesifioituja työkaluja. Yan, Early ja Anderson (2000) ehdottavat XenoService-palvelua palvelunestohyökkäysten estämiseksi, joka perustuu kohdistettuun serverien verkostoon, joka isännöi pelipalveluita. Kun yksi palvelu joutuu hyökkäyksen kohteeksi, XenoService luo siitä useita kopioita, jolloin peli pystyy jatkamaan toimintaansa. Yan ja Chen (2001) myös lisäävät, että jos peliserveri on suunniteltu tunnistamaan ja hylkäämään peliin liittymättömät datapaketit, on palvelunestohyökkäyksien lieventäminen helpompaa. Puskurin ylivuotohyökkäyksiin voidaan soveltaa esimerkiksi StackGuardia, joka rajoittaa ylivuotohyökkäyksen vahingon määrää. Työkalun etuna on sen monikäyttöisyys, sillä se ei ole suunniteltu vain tietynlaisia ylivuotohyökkäyksiä vastaan. Täten se voi suojata järjestelmää ennestään tuntemattomilta hyökkäyksiltä, mikä vähentää tarvetta tehdä jatkuvia ohjelmistopäivityksiä. Lisäksi StackGuardia

voidaan säätää omien tarpeiden mukaan, jolloin se voi tarvittaessa vähentää järjestelmän tehokkuutta turvallisuuden takaamiseksi. (Cowan ja muut, 1998.) Myös paketti-injektiot ja SQL-hyökkäykset voidaan estää niitä varten kehitetyillä välineillä. Hsu, Zhu ja Hurson (2007) esittelevät autentikointi-protokollan, joka estää hyväksymättömien pakettien lisäämisen verkkoliikenteeseen. Sekä Wittmann (2009) että Boyd ja Keromytis (2004) mainitsevat, että SQL-hyökkäyksiä voidaan ehkäistä huolellisemmalla ohjelmoinnilla ja suunnittelulla. Boyd ja Keromytis (2004) ehdottavat lisäksi SQLRand-järjestelmää, joka perustuu satunnaistettuun SQL-kyselykieleen. Tämän järjestelmän avulla on mahdollista havaita ja hylätä mahdolliset haitalliset injektiot, ilman, että joudutaan kärsimään vähentyneestä tehokkuudesta. Näiden spesifioitujen työkalujen ongelmana voi olla niiden saatavuus, varsinkin jos ne ovat hintavia.

3.2 Pelin heikkouksien hyväksikäytön ehkäisy

Pelisovellukseen voidaan sisällyttää erillinen moottori, joka havaitsee huijaajia. Tällä voidaan korvata joidenkin pelien tarjoajien ehdotus käyttää kokeneita kehittäjiä havainnoimaan pelaajien käyttäytymistä. Sovellukseen sisällytetty ominaisuus on paitsi halvempi ratkaisu, sitä voidaan myös mahdollisesti hyödyntää useammassa pelissä. Kyseinen ominaisuus voidaan sijoittaa hyvin suojattuna peliserveriin, jos pelin kehittäjä ei voi taata pelialustan turvallisuutta. (Yan & Chen, 2001.) Peliserveriin voidaan myös implementoida ominaisuus, joka havainnoi asiakassovelluksen toimintaa ja ilmoittaa, jos kyseinen asiakas toimii epätavallisesti. Tällöin voidaan perehtyä asiakassovelluksen toimintaan, eli miten se hyväksyy syötteitä ja päivittää tilaansa ja serveriä tarvittavilla tiedoilla. (Bethea ja muut, 2011.) Tällä voidaan estää tapauksia, joissa huijaajat muokkaavat asiakassovellusta, saaden epäreilun etulyöntiaseman.

Tietoja ja ohjelmistoja voidaan suojella datan salaamisella, jolloin sekä muistissa oleva että siirretyissä paketeissa käsitelty tieto ei paljasta huijaajalle mitään, mitä tämä voisi muokata omaksi edukseen. TRM (eng. *tamper resistant module*) ja binäärisuojaus ovat kaksi datan suojauksen metodia. TRM voi varmistaa ohjelmiston eheyden, kun siihen tehdään muutoksia. Binäärisessä suojauksessa ladattava koodi uusitaan nopealla tahdilla, joten huijaajan on vaikeampaa tehdä siihen muutoksia. Tähän suojaustyyppiin liittyy myös erillinen vartija, joka varmistaa itse suojausmekanismin eheyden. Vartijan käyttöä puolestaan toteuttaa erillinen RCAA-algoritmi. (Lan ja muut, 2009.) Myös Sethi ja Allen (2014) mainitsevat vastaavan binäärivartijan mahdollisena ratkaisuna. Tämäkin metodi on kuitenkin rajoitettu, sillä vaikka se pidentää suojausaikaa, suojaus ei kuitenkaan tapahdu kaiken aikaa (Lan ja muut, 2009). Sethi ja Allen (2014) kirjoittavat sovelluksen kovettamis (eng. *application hardening*) tekniikoista, jotka hyödyntävät logiikkaa, joka puolustaa sovellusta sekä koodin manipuloinnilta että hyökkäyksiltä. Pelin tiedostoja suorittavia kirjastoja tulisi myös valvoa mahdollisimman aktiivisesti ja päivittää niitä yhdessä pelin mukana. Muuten huijaajat yksinkertaisesti löytävät vanhentuneet kirjastot ja hyväksikäyttävät niitä (Bono ja muut, 2009).

Kolmannen osapuolen ohjelmissa voidaan hyödyntää samoja ominaisuuksia kuin selaimissa, eli niiden käyttöön voidaan varata erillinen ”hiekkalaatikko” (eng. sandbox), jossa käyttäjien ei ole mahdollista käyttää hienostuneita järjestelmäkomentoja. Lisäksi lisäosille voidaan vaatia tietty skriptauskieli ja suorittaa niille erillinen valvottu testaustilaisuus, ennen kuin niiden käytölle annetaan virallinen lupa (attack surface). Kolmannen osapuolen ohjelmien havainnointiin voidaan hyödyntää pelaajan tietokoneen muistin skannaamista serverin toimesta. Löytäessään jotain epätavallista lähetetään tiedote pelin ylläpitäjälle. (Lan ja muut 2009.) Sethin ja Allenin (2014) mukaan tämä voi kuitenkin johtaa yksityisyysongelmiin pelaajille ja olla liian myöhäinen ehkäisykeino. Valvonnan lisäksi serverin puolella voidaan analysoida

pelaajien pelin sisäisiä toimintoja ja löytämällä poikkeuksia voidaan todeta, onko kyseessä huijaus (Lan ja muut, 2009). Sethi ja Allen (2014) ovat samaa mieltä ja lisäävät, että serverin puolella tapahtuva analysointi tulisi implementoida jo varhaisessa vaiheessa, koska sen lisääminen jälkikäteen voi olla haastavaa. Tietoverkkotason huijauksen ehkäisyssä suosittu tapa on käyttää huijauksen vastaisia protokollia. Protokollien tarkoituksena on rajoittaa tiedon ja viestien siirrossa käytettyä aikaa tietoverkossa. (Lan ja muut, 2009.) Vertaisverkkotilanteissa pelin ylläpitäjät joutuvat yleensä luomaan hätäisesti päivityksiä, koska vertaisverkon välityksellä tapahtuvia väärinkäyttöjä on mahdotonta estää, sillä data ei kulje peliserverien kautta. Vertaisverkkoja käyttäessä pelaajilta tulisi pyytää hyväksyntä ennen yhteyden luomista ja mahdollistaa pelaajien luoda lista luotetuista vertaisista, jolloin yhteys voidaan muodostaa automaattisesti. Lisäksi kaikkeen pelaajien välillä siirrettyyn dataan tulisi suhtautua erittäin epäilevästi. Asiakas-serveri vuorovaikutuksessa paras keino on turvallisuuspainotteinen ajattelu suunnittelu- ja testausvaiheissa, jolloin mahdolliset heikkoudet voidaan havaita ja korjata. (Bono ja muut, 2009.)

Vaikka kaikkia heikkouksia, eli bugeja, ei voidakaan korjata ennen julkaisua, voidaan silti soveltaa niiden jälkikäteistä korjaamista. Tällöin peliä päivitetään ja päivityksissä korjataan tunnettuja heikkouksia. Lewis, Whitehead ja Wardrip-Fruin (2010) laativat tätä varten luokittelun erilaisista mahdollisesti esiintyvistä bugeista. Koska videopelit ovat yleisesti monimutkaisempia kuin perinteiset ohjelmistot, he kehottavat jakamaan pelin osiin ja käymään tämän jälkeen luokittelun avulla kaikki osat läpi. Esimerkiksi ensimmäisenä voidaan perehtyä pelaajan ohjaamaan hahmoon ja luoda tälle luokittelun perusteella testit, joilla löydetään ongelmia. Tämän jälkeen siirrytään muihin pelin objekteihin, kunnes kaikki tärkeä on käyty läpi. Heidän mukaan luokittelu auttaa pelien testaajia ohjaamalla heitä oikeaan suuntaan ja näin luomaan uusia ja järjestelmän paremmin kattavia testejä. Lisäksi vastaavan luokittelun avulla voidaan todentaa suunniteltujen testien hyödyllisyys ja opastaa uusia testaajia ongelmakohtien ymmärtämisessä. Kirjoittajat myös mainitsevat luokittelun implementointia Zenet-työkaluun, johon syötetään luokittelun määrittelevät olosuhteet, joiden mukaan peliä tarkastellaan. Tämän jälkeen työkalu käy läpi pelin osia ja ilmoittaa, jos luokittelun mukaisia virheitä löytyy. Koska pelien testaukseen joudutaan palkkamaan yleensä jopa satoja testaajia, voidaan automaattisilla työkaluilla vähentää kuluja. Työkalujen ohjelmointiin puolestaan voidaan soveltaa bugien luokittelua, jolloin jälleen hyödytään tarkemmasta virheiden löytämisestä. Pelinsisäinen tila tulisi pystyä palauttamaan siihen tilanteeseen, jossa se oli ennen huijaamisen tapahtumista. Apuna voidaan käyttää palautuspistemenetelmiä. Lokitiedostoja voidaan myös hyväksikäyttää ehkäisemään tietynlaisia huijauksia, kuten peleissä, joissa korkeimman pistemäärän saanut pelaaja voittaa pelin. Jos huijaaja on saanut pääsyn pelin tietoihin ja pystyy muuttamaan pelisession pistetilastoja, voidaan lokitiedostosta havaita väärinkäyttö. (Yan & Chen, 2001.)

3.3 Tekoälypohjaisen huijaamisen ehkäisy

Bottien havaitseminen ei ole helppoa, koska vaikka niiden käyttäytymisessä onkin eroja ihmisiin verrattuna, ne noudattavat pelinsisäisiä sääntöjä toiminnassaan. Botteja vastaan kamppailu perustuu yhä pääasiassa ihmisten väliseen vuorovaikutukseen, eli pelaajilla on mahdollista raportoida pelihahmo, jonka he uskovat olevan botin ohjaama. Raportin saatuaan pelin ylläpitäjä voi perehtyä kyseiseen pelihahmoon ja todeta onko kyseessä botti, esimerkiksi yrittämällä keskustella tämän kanssa. Tämän metodin ongelmana on sen epäkäytännöllisyys, varsinkin peleissä, joissa pelaajien määrä yltää kymmeniin tai satoihin tuhansiin. Joidenkin pelien kehittäjät turvautuvat samaan strategiaan kuin muitakin kolmannen osapuolen ohjelmistoja etsiessä, eli pelaajan tietokonetta skannaaviin automaattisiin ohjelmistoihin. Nämä ohjelmistot voivat kuitenkin havaita

vain tunnettuja ohjelmia, joten bottien kehittäjät ovat aina askeleen edellä. Lisäksi koska kyseinen skanneri toimii asiakkaan koneella, joka on täysin pelin kehittäjän vaikutusvallan ulkopuolella, ei skannauksen tuloksiin voida välttämättä luottaa. (Mitterhofer, Platzer, Kruegel & Kirda, 2009.)

Gollen ja Duchenaut (2005) ehdottavat pelin suunnittelussa hyödynnettäviä elementtejä, jotka voivat lannistaa bottien käyttöä. Koska botteja käytetään usein resurssien keräämisessä, jotka voidaan myöhemmin myydä eteenpäin, voidaan resurssien siirtämiseen pelitileiltä toiselle asettaa pitempi aikaraja. Tällöin varsinkin suurissa siirroissa voidaan helpommin havaita bottien käyttäjä. Jos bottien havaitseminen helpottuu, kasvaa botteja käyttävien huijaajien mahdollisuus menettää kaikki siihen asti keräämänsä resurssit. Tämän lisäksi peleistä voidaan tehdä entistä sosiaalisimpia, rohkaista tiimityöskentelyä ja samalla asettaa kovempia rangaistuksia bottien käytöstä. Esimerkiksi koko tiimi voidaan asettaa käyttökieltoon, jos kukaan tiimin pelaajista käyttää botteja. Chen, Liao, Pao ja Chu (2008) puolestaan ehdottavat passiivisia havaitsemiskeinoja, joissa keskitytään pelihahmojen liikkeiden seurantaan. Koska botit toimivat tietyllä tavalla, myös heidän liikkumisensa pelimaailmassa eroaa oikeista ihmisistä. Analysoimalla saatua dataa voidaan erotella oikeat pelaajat boteista. Myös Mitterhofer ja muut (2009) ehdottavat vastaavaa metodia, jossa havainnoidaan ja analysoidaan pelihahmon liikkumista. Löytämällä usein toistuvia liikesarjoja voidaan erotella botit oikeista pelaajista. He myös mainitsevat, että vastaavan järjestelmän etuna on sen näkymättömyys käyttäjälle, koska se ei sisällä peliä häiritseviä elementtejä.

Gollen ja Ducheneautin (2005) kuitenkin huomauttavat, ettei botteja voida eliminoida vain suunnittelemalla entistä monimutkaisempia pelejä, koska tällöin vaarana on vähentynyt pelaajakunta. Sen sijaan tarvitaan tekniikoita, joilla botit voidaan erottaa ihmispelaajista. Tätä varten he esittelevät kaksi näkökantaa, jotka pohjautuvat CAPTCHA-testeihin (eng. *Completely Automatic Public Turing Test to Tell Computers and Humans Apart*). Kyseiset testit ovat sellaisia, jotka useimmat ihmiset pystyvät läpäisemään, mutta tietokoneohjelmat eivät. Useimmin pelien ulkopuolellakin käytetty CAPTCHA-testi perustuu ihmisen kykyyn saada selvää vääristetystä tekstistä tai kuvasta. Ensimmäinen näkökanta perustuu pelin sisälle sijoitettuihin testeihin, joissa pelaajan täytyy suorittaa toimia, joita vain ihminen kykenee tekemään. Halutessaan luoda pelinsisäisen objektin, kuten uuden aseen, pelaajan täytyy todistaa olevansa ihminen suorittamalla tekstipohjainen CAPTCHA-testi. Jos pelaaja haluaa metsästää pelimaailman eläimiä, voidaan soveltaa kuvatestiä, josta pelaajan tulee tunnistaa jokin eläin. Vastaavat testit eivät loisi pelitilanteeseen kuin yhden lisäaskeleen, jonka oikea pelaaja voi läpäistä erittäin nopeasti. Tietynlaisiin peleihin, kuten massiivisiin verkkoroolipeleihin, tällaisten testien implementointi olisi sopivinta, koska ne ovat hidastempoisia, niiden peliympäristöissä vallitsee korkea entropia ja niissä käytettävät säännöt eivät ole liian tarkkoja. Täten CAPTCHA-testien läpäisyyn on riittävästi aikaa ja ne voidaan suunnitella siten, että ne sulautuvat pelimaailmaan huomaamattomasti. Toinen Gollen ja Ducheneautin (2005) esittämä näkökanta puolestaan pohjautuu fyysisiin laitteisiin, kuten ohjaimet ja näppäimistöt, joihin rakennetaan CAPTCHA-ominaisuuksia. Ihmiset siis erotellaan boteista heidän kyvyillään toimia fyysisessä maailmassa; ihmisen on helppo painaa näppäimistön näppäintä tai koskettaa kosketusnäyttöä, mutta vastaavien laitteiden kehittäminen, jotka emuloivat haluttuja toimia automaattisesti, on haastavaa ja kallista. Koska fyysisen CAPTCHA-laitteen ideana on tuottaa digitaalinen tuloste fyysisen toiminnon kautta, näiden laitteiden tulee autentikoida saatu tuloste ja olla suojattuja muokkauksilta. Autentikointi, esimerkiksi peliserverin kautta, varmistaa, että syöte tulee CAPTCHA-laitteelta kullakin ajanhetkellä. Jos käyttäjä pyrkii muokkaamaan laitetta, se menettää kykynsä autentikoida, eikä enää toimi.

Näilläkin ratkaisuilla on kuitenkin ongelmansa. Kuten Lan ja muut (2009) huomauttavat, tehokkuudestaan huolimatta pelinsisäiset CAPTCHA-testit voivat rikkoa pelin jatkuvuuden. Golle ja Duchenaut (2005) myös mainitsevat, että vuorovaikutteisuutensa takia CAPTCHA-testit ovat usein liian häiritseviä, varsinkin koska vastaavia testejä tulisi suorittaa useita saman pelisession aikana bottien sisäänkirjautumisen välttämiseksi. Jos testejä on vain yksi, huijaaja voi suorittaa sen ja tämän jälkeen yhdistää botin peliin. Pelattavuuden ylläpitämisen takia he esittävät, että testit tulisi suunnitella mukailemaan pelissä jo esiintyviä aktiviteetteja mahdollisimman tarkasti. Lisäksi Lanin ja muiden (2009) mukaan niin CAPTCHA-testit kuin passiiviset liikeseurannat ovat liian yksikantaisia, olettaen joko pelaajan ohjaavan botteja suoraan tai bottien toimivan itsenäisinä asiakkaina, muttei molempia yhtä aikaa. Golle ja Duchenaut (2005) myös myöntävät, että fyysisiin laitteisiin rakennetut CAPTCHA-ominaisuudet voivat tulla hyvin kalliiksi toteuttaa.

3.4 Kolluusion ehkäisy

Kolluusion ehkäisyn ongelmana on tahallisen kolluusion havaitseminen virheistä, tuurista ja taidosta. Kolluusiota tunnistaessa on täten useita seikkoja, joihin on perehdyttävä. Pelaajan parantaessa taitojaan hänen käyttämänsä taktiikat ja strategiat kehittyvät. Kehittynyt pelaaja pyrkii toimintoihin, jotka luovat ennustettavuutta. Toisin sanoen, hän pyrkii vähentämään mahdollisia lopputuloksia. Tähän päästäkseen kokenut pelaaja voi yrittää luoda vakaan peliasetelman, mikä toisaalta luo myös tietyissä tilanteissa esiintyviä toistuvia manöövereitä. Samanlainen ennustettavuus voidaan saavuttaa luonnollisesti myös kolluusiolla. Pelaajan tulokset ovat myös yleensä parempia tämän pelatessa hänelle tuttujen pelaajien kanssa. Pelaaja pyrkii myös pelaamaan järkevästi, eli minimoimaan itselle huonoimman lopputuloksen. Järkevä pelaaja yleensä pyrkii välttämään hyödyttömiä konflikteja ja on täten valmis luomaan sopimuksia säästääkseen resursseja. Pelin voittamisella on yleensä hintansa, jota pidentyneet konfliktit kasvattavat. Lisäksi pelissä voidaan päätyä tilanteisiin, joissa pelaajia koskevat päätökset ja toiminnat ovat samanaikaisesti sekä itseä että vastapelaajaa hyödyntäviä tai haittaavia. Jos kaksi yhtä vahvaa pelaajaa kohtaavat ja he ovat tietoisia taitotasoistaan, he voivat pelata varovaisemmin. Jos taas kyseinen vahvuus selviää vasta konfliktin jälkeen, he saattavat päätyä kolluusioon, välttääkseen kohtaamasta tulevaisuudessa. Kahden pelaajan ollessa eritasoisia ja silti välttävät konfliktia, voidaan olettaa heidän syyllistyneen kolluusioon. (Smed ja muut, 2006.)

Mitä vähemmän valintoja pelaajalla on, sitä vaikeampaa kolluusion salaaminen on. Korttipeleissä pelaajalla on rajallinen määrä toimintoja, joita tämä voi vuorollaan suorittaa, joten käyttäytymisessä ilmeneviä kaavamaisuuksia on helpompi havaita. Koska kolluusio on yksi tällainen kaavamaisuus, on sen havainnointiin mahdollista kehittää metodeja. Havainnointi muuttuu kuitenkin hankalaksi, kun pelissä esiintyvä vapaus kasvaa. Tämä on totta varsinkin verkkoroolipeleissä. Jotkin roolipelit, kuten *World of Warcraft*, ei salli kilpailevien kiltojen yhteistyötä ja pyrkii havaitsemaan tätä seuraamalla epäilyttävää käyttäytymistä. Luotettava havainnointi vaatii kuitenkin tiettyjen seikkojen analysointia. Tietääkseen miten pelaajat voivat hyötyä, joko laillisesti tai huijaamalla, pelissä ilmenevät mahdollisuudet tulee tuntea. Pelaajalle voidaan myös luoda erillinen profiili, josta ilmenee hänelle ominaisia käyttäytymisen tapoja. Lisäksi pelaajalle voidaan luoda tämän käyttäytymistä analysoimalla maine pelin sisälle. Kyseinen ominaisuus voidaan myös avata pelaajille, jolloin he voivat arvostella muita pelaajia. Pelaajan pelaamisen historiaa tarkkailemalla saadaan selville muutoksia tämän käyttäytymisessä, jotka voivat viitata kolluusioon. Edellä kuvailtua analyysia varten tarvitaan lisäksi joitain tarkistuspisteitä. Tekoälyjen avulla voidaan mallintaa kehittynyttä ja sattumanvaraista pelaamista. Tällöin saadaan tietoa toimintojen arvokkuudesta, kuinka hyvään lopputulokseen voidaan päätyä rehellisellä pelaamisella

ja mikä on päämäärättömin tekoälyn tuottama tulos. Pelaajien tutkimista varten tarvitaan lista tyypillisimmistä pelaajan toteuttamista toiminnoista sekä vertailua pelaajien välillä, jolloin nähdään, onko käyttäytymisissä korrelaatiota. Nämä tarkistuspisteet auttavat löytämään epäilyttävää käyttäytymistä yksittäisitä tai ryhmittyneistä pelaajista, sekä arvioimaan pelaajien tekemiä toimintoja ja niiden järkevyyttä. (Smed ja muut, 2006.)

Koska manuaalinen kolluusion valvominen on kallista ja lähes mahdotonta toteuttaa (Yan, 2003), pelisovellukseen voidaan implementoida automaattinen ominaisuus, joka havaitsee huijaajia, jolloin pelaajat voivat pelata kenen kanssa haluavat ja mahdollisen kolluusion tapahtuessa huijaus havaitaan sovelluksen toimesta (Yan & Chen, 2001). Ominaisuuteen voidaan ennalta ohjelmoida, minkälaisia pelaajan toimia sen tulee tarkkailla ja asettaa nämä toimet tärkeysjärjestykseen, erittäin epäilyttävistä vähemmän epäilyttäviin. Kun pelaajalle rekisteröidään tarpeeksi korkean tason epäilyttäviä toimintoja, tämä todetaan huijaajaksi. Lisäksi niille, joita ei todeta huijaajiksi, lasketaan erillinen pistetaso, joka kertoo, kuinka epäilyttävä pelaaja on. Tällöin pelaajilla on mahdollisuus päättää, näiden pistetasojen pohjalta, kuinka suuri on huijaamiseksi tulemisen riski pelatessaan kenenkin pelaajan kanssa. Lisäksi tämä metodi mahdollistaa pelaajien rajaamisen, jolloin vain erittäin epäilyttävien tapausten tutkimiseen voidaan keskittää enemmän aikaa. (Yan, 2003.) Automaattinenkaan järjestelmä ei kuitenkaan ole ongelmaton. Jos vastaava ominaisuus toteutetaan, voidaan kyseessä oleva peli tai sen osia joutua suunnittelemaan kokonaan uudelleen. Tutkittava data on esimerkiksi ennen voitu poistaa heti pelin jälkeen, mutta nyt se joudutaan säilömään. Säilöminen ja datan siirtäminen lisäävät vaadittuja resursseja. Pelaajien vahingossa tekemiä kolluusioon viittaavia toimintoja voidaan myös käyttää heitä vastaan. (Yan, 2003.) Varsinkin verkkoroolipelien kehittäjät ovat myös maininneet, kuinka automaattisen havainnointijärjestelmän luotettava toteuttaminen on vaikeaa (Smed ja muut, 2006).

Pelaajien sijoittamisessa peliin voidaan hyödyntää satunnaistamista, jolloin jokainen pelaaja asetetaan sattumanvaraisesti pelisessioon. Tämä tosin vähentää mahdollista sosiaalista puolta, koska ne pelaajat, jotka haluavat pelata ystäviensä kanssa, erotellaan toisistaan (Lan ja muut, 2009). Vaikka tämä voikin olla hyödyllinen menetelmä, se ei kuitenkaan täysin estä kolluusiota. Jos peliserverillä on paljon huijaajia, satunnaistaminen voi yksinkertaisesti saattaa heitä yhteen uusien huijaajien kanssa (Yan, 2003). Pelaajien välistä kommunikointia pelin sisällä voidaan myös rajoittaa. Jos pelissä on tekstin välityksellä käytävä keskusteluominaisuus, se voidaan yksinkertaisesti kytkeä pois päältä. Tekstin muotoilua voidaan myös rajoittaa, jolloin huijaajien on vaikeampaa sisällyttää salaisia viestejä kirjoittamaansa tekstiin. Tällöin voidaan hyödyntää oikeinkirjoituksen tarkistajaa sekä käyttöliittymään implementoituja automaattisia viestejä. Jos kyseessä on sosiaalinen peli, tällainen kommunikoinnin rajoittaminen ei kuitenkaan ole välttämättä suositu ratkaisu pelaajien keskuudessa, eikä sillä luonnollisesti voida estää pelin ulkopuolisten kommunikointivälineiden avulla tapahtuvaa kolluusiota. (Yan, 2003.)

Koska kolluusiota tapahtuu usein peleissä, joissa pelaajat listataan taitotasonsa mukaisesti, voidaan näitä tilastoja tarkkailla. Jos näyttää siltä, että kaksi pelaajaa nousevat tilastoissa epäilyttävän nopeasti, voidaan heidän pelihistoriansa tarkistaa lokitiedostojen kautta ja todentaa, onko kyseessä mahdollinen kolluusiotapaus. Tämä on tehokas tapa havaita kolluusio, joissa pelaajat vuorottelevat pelin voittajan välillä. Joissakin peleissä tämä pelkästään ei kuitenkaan riitä, koska pelaajien on mahdollista pelata yhdessä ja nousta tilastoissa sääntöjen mukaisesti, kuten Bridge-korttipelissä. On myös mahdollista, että kolluusiota tapahtuu, mutta huijaajat ovat aloittelijoita tai taitotasoltaan heikompia kuin vastustajansa, jolloin epäilyttävää tilastoissa nousemista ei välttämättä tapahdu. (Yan, 2003.) Tapauksissa, joissa kolluusio tapahtuu sisäisen väärinkäytön avulla, pelin ylläpitäjän oikeuksia ja valtuuksia voidaan vähentää. Lisäksi

väärinkäytösten kovempi rankaiseminen ja lokitiedostojen käyttö ylläpitoon käytettyjen toimien kirjaamisessa voi vähentää huijaamista. (Lan ja muut, 2009.)

3.5 Autentikointi, turvallisuuskoulutus ja virtuaalinen omaisuus

Lan ja muut (2009) esittävät, että pelkästään pelaajan pelitunnuksen ja salasanan tietäminen ei tulisi riittää pelaajan pelitiliin pääsemiseksi. Näiden lisäksi voidaan toteuttaa digitaalinen varmennus, joka lisää tilin suojausta. Heidän mukaan peliyrityksen tulisi perustaa erillinen osasto, joka käsittelee huijaamista kohdanneiden pelaajien raportteja. Myös Yan ja Chen (2001) mainitsevat, että pelaajilla tulisi olla kommunikaatiokanava, jota kautta he voivat ilmoittaa heikkouksista, väärinkäytöksistä ja huijaamisesta. Tätä kanavaa voidaan myös hyväksikäyttää pelaajien kouluttamisessa, ilmoittamalla uusista turvallisuusriskeistä tai uutisista huijaamiseen liittyen. Autentikointitietojen siirtämisessä tulisi myös käyttää mahdollisimman turvallista ratkaisua. Terra-pelissä http-ratkaisun sijaan olisi voitu käyttää SSL-menetelmää, jolloin tietojen urkkiminen olisi vaikeutunut huomattavasti. Lisäksi salasanan murtamisessa käytettyjä kytkeytyneitä pakotushyökkäyksiä voidaan ehkäistä asettamalla rajoitteita kirjautumisyrityksille. Jos vääriä yrityksiä on liikaa, voidaan käyttäjä lukita ulos järjestelmästä. (Prashar ja muut, 2009.) Yhteydettömiä pakotushyökkäyksiä voidaan ehkäistä salasanatiedoston tarkalla salaamisella, jolloin yksinkertaisimpienkin salasanojen murtaminen hidastuu huomattavasti. Lisäksi salasanatiedostoihin pääsyä tulee valvoa, vartioida ja rajoittaa niin paljon kuin mahdollista. (Florêncio & Herley, 2010.)

Salasanan valintaan liittyvä prosessi on myös tärkeässä roolissa huijausta ehkäistäessä. Yan ja Chen (2001) ehdottavat proaktiivista salasanan tarkistamista, jolloin salasanaa valittaessa pelaaja saa viestin, onko valittu salasana hyväksyttävä vai ei. Yanin (2001) mukaan entropia-pohjainen proaktiivinen tarkistaminen on hyödyllinen väline, vaikka se voikin hyväksyä joitakin heikkoja salasanoja. Yan (2004) myös tähdentää, että pelkkä käyttäjän kouluttaminen salasanan valinnassa ei ole riittävä metodi turvallisuuden lisäämiseksi. Vaikka kouluttaminen parantaakin salasanojen turvallisuutta jonkin verran, ei parannus ole riittävä olemaan tarpeeksi merkittävä, varsinkin sillä peleissä toisen pelaajan salasanan selvittäminen voi johtaa vakaviin ongelmiin. Täten koulutuksen lisäksi salasanan valinnan valvominen ja hyväksyminen ovat erittäin tärkeitä. Lisäksi Pipkin (2003) mainitsee, että varsinkin salasanojen vaihtamiseen tai autentikointitietoja vaativiin ohjelmiin liittyen tulisi olla selkeästi kommunikoidut periaatteet, jolloin käyttäjien manipulointi vähentyy ja he tietävät, milloin kyseessä on aito pelin ylläpitäjältä tuleva viesti.

Pelin tarjoajat voivat kouluttaa pelaajiaan huijaamisesta. Koulutuksessa voidaan paneutua moraalisiin ja ekonomisiin ongelmiin, joita huijaamisesta aiheutuu. Lisäksi huijaamisesta seuraavien rangaistusten selvä esittely voi vähentää huijaamisen määrää (Lan ja muut, 2009). Myös Yan ja Chen (2001) painottavat, että pelin tarjoajien tulisi kouluttaa pelaajia siitä, minkälaisia turvallisuusriskejä on olemassa ja miten näitä voidaan ehkäistä. Esimerkiksi joissain verkkopeleissä käytetään pelin ulkopuolisia applikaatioita, kuten verkkoselaimia, täydentämään joitain pelin ominaisuuksia; pelaaja suorittaa jonkin pelinsisäisen toimen, joka toteutetaan verkkoselaimen kautta. Samalla tavoin kuin sähköpostin välityksellä toimivat haittaohjelmat toimivat, huijaajat voivat käyttää näitä pelinsisäisiä ominaisuuksia levittämään omia haittaohjelmiaan, naamioimalla haitallisia linkkejä tai ominaisuuksia virallisten kaltaisiksi. Ellei pelin kehittäjä halua täysin luopua näistä ominaisuuksista, tarvitaan pelaajien koulutusta. Tällöin voidaan pelaajia varoittaa, kun he ovat suorittamassa ulkoisia sovelluksia (Bono ja muut, 2009). Pelaajat voivat myös henkilökohtaisesti vähentää mahdollisuutta

joutumasta huijauksen kohteeksi. Kuten Lan ja muut (2009) kirjoittavat, palomuurin asentaminen omalle tietokoneelle lisää omien tietojen turvallisuutta.

Pelin sisäistä huijauksen havainnointia voidaan hyödyntää myös virtuaalisen omaisuuden väärinkäytössä. Suosittu virtuaaliseen omaisuuteen liittyvä huijaus on tavaroiden duplikointi. Tavaroita duplikoitaessa sovellus havaitsee, että olemassa olevia tavaroita on liikaa ja voi toteuttaa halutut toimenpiteet, kuten asettaa huijaajan käyttökieltoon. Lisäksi peleissä, joissa pelaajat myyvät omaisuutta kauppiaille, sovellus voi asettaa rajoitteita resurssien myymiseen ja havaita, jos nämä rajat ylitetään (Yan ja Chen, 2001). Pelin tarjoajat voivat myös hyödyntää luotettavaa kolmatta osapuolta pelaajien välisissä kaupoissa. Tällöin pelaajat sopivat kaupan ehdoista, minkä jälkeen he siirtävät virtuaalisen omaisuuden kolmannen osapuolen haltuun, joka auttaa kaupan turvallisessa ratkaisussa. Tämän metodin ongelmana on sen mahdollinen hintavuus, koska se vaatii jatkuvaa ihmisten välistä toimintaa (Yan & Chen, 2001). Ben-or, Goldreich, Micali ja Rivest (1990) esittelevät automaattisen ratkaisun, jolloin väliintuloa vaaditaan vain pelaajien välisissä kiistoissa

Huijaamisen estämiseen on siis kehitetty lukuisia erilaisia käytänteitä. Hyökkäykset tulisi ottaa huomioon pelin suunnittelussa, jolloin niiden toteuttaminen on vaikeampaa. Lisäksi niitä vastaan on ehdotettu spesifioituja työkaluja sekä käyttökieltoja rankaisuiksi. Pelin heikkouksia varten yleisimmät metodit liittyvät olemassa olevan tiedon parempaan suojaukseen, pelin päivittämiseen ja valvontaan, esimerkiksi peliserverin tai pelaajan koneen osalta. Tekoälypohjaisen estäminen on haastavampaa, mutta peliä voidaan suunnitella siten, että bottien käyttäminen vaikeutuu. Lisäksi erilaiset CAPTCHA-testit ja CAPTCHA-pohjaiset fyysiset laitteet ovat ehdotuksia, joilla botteja voidaan ehkäistä. Myös sosiaalista huijaamista voidaan ehkäistä tietoteknisillä keinoilla. Kolluusiota varten voidaan soveltaa pelaajien satunnaistamista pelitilanteissa sekä vähentää pelaajien välisiä kommunikointimahdollisuuksia. Tuntemalla pelin ominaisuudet ja pelaajien tavoitteet, sekä näiden yleisimmät toiminnot, voidaan analysoida peleistä saatua dataa ja havaita epäilyttäviä tapauksia. Pelaajien havainnointi voidaan toteuttaa automaattisesti tai manuaalisesti, riippuen pelin koosta ja käytettävissä olevista resursseista. Lopuksi voidaan paneutua pelaajien kouluttamiseen turvallisuudesta, jolloin esimerkiksi manipulointi vähentyy. Pelien ylläpitäjillä tulisi myös olla aktiivinen kommunikointikanava pelaajien kanssa, jolloin pelaajat voivat raportoida huijaajia ja ylläpitäjät voivat ilmoittaa mahdollisista turvallisuuteen liittyvistä uutisista. Salasanojen valintaan tulisi kuitenkin hyödyntää tietoteknisiä metodeja pelkän kouluttamisen sijaan, kuten automaattisia viestejä, jotka kertovat onko valittu salasana tarpeeksi vahva.

4. Pohdinta

Tutkimuksessa käsitelty tutkimusongelma selvittää, minkälaisia huijaustapoja huijaajat käyttävät verkkopeleissä ja miten niitä voitaisiin estää. Yleisesti huijaajat voivat turvautua erilaisiin tietoteknisiin keinoihin, kuten hyökkäyksiin, autentikointitietojen kalasteluun, pelin heikkouksien hyväksikäyttöön sekä tekoälypohjaiseen huijaukseen. Lisäksi huijaajat voivat käyttää sosiaalisia menetelmiä, kuten muiden huijaajien kanssa tehtyä yhteistyötä, toisten käyttäjien manipulointia sekä virtuaalisen omaisuuden epärehellistä hankintaa ja väärinkäyttöä. Eri huijaamismetodeille löydettiin monia erilaisia ehkäisykeinoja. Näistä useimmat liittyvät tietoteknisesti pelin järjestelmien parantamiseen, kuten tiedon salaamiseen. Lisäksi pelaajia tulisi kouluttaa turvallisuuteen liittyvistä seikoista, jolloin heidän joutumistaan huijaajan uhreiksi voidaan vähentää. Tässä luvussa käydään läpi löydettyjä ongelmia ja ratkaisukeinoja, sekä miten niiden soveltaminen toimisi käytännössä.

Palvelunestohyökkäykset ovat hyökkäyksistä selvästi suosituin huijausmetodi. Yanin ja Chenin (2001) mukaan palvelunestohyökkäyksien estämiseen yleisesti ehdotetut menetelmät eivät kuitenkaan ole erityisen tehokkaita, koska huijaajilla on yleensä suurempi motivaatio huijata, kuin rehellisillä pelaajilla on suojella turvallisuutta. Tällä viitataan siihen, että palvelunestohyökkäyksissä usein käytetään kaapattuja palvelimia, joilla hyökkäys toteutetaan. Voidaan siis ajatella, että vaikka rehelliset käyttäjät pyrkisivätkin parantamaan omia turvallisuuskäytänteitä, huijaajilla on enemmän halua toteuttaa onnistunut palvelunestohyökkäys, joten he käyttävät enemmän aikaa ja resursseja murtaakseen parantuneetkin turvallisuusmetodit. Tällöin pelin ylläpitäjiltä vaaditaan entistä enemmän omia käytänteitä, joilla voidaan havaita ja estää hyökkäyksiä.

Pelin heikkouksien hyväksikäytön ehkäisyssä voidaan hyödyntää vuoropohjaista turvallisuuden parannusta. Kuten esimerkiksi Yan ja Chen (2001) ehdottavat, luomalla kommunikaatiokanava pelaajien ja ylläpitäjien välille mahdollistaa tämän. Tällöin pelaajat voivat havaitessaan heikkouksia raportoida niitä suoraan pelin kehittäjille, jotka puolestaan voivat päivittää peliä ja tukkia löytyneitä turvallisuusaukkoja. Pelin ongelmista suurimmat keskittyvät kuitenkin asiakasohjelmiin perustuvissa tapauksissa, joissa pelin kehittäjät luottavat liikaa luomaansa ohjelmaan, jota huijaajien on todellisuudessa helppo muokata. Tähän todennäköisesti yksinkertaisin ratkaisu olisi sijoittaa mahdollisimman paljon tärkeästä tiedosta peliserverin puolelle, jolloin siihen käsiksi pääseminen vaatisi huomattavasti monimutkaisempia toimenpiteitä. Asiakasohjelmiin tehdyt muutokset eivät vaadi huijaajalta välttämättä erityisen isoa teknistä asiantuntemusta.

Tekoälypohjainen huijaus on nykyään erittäin suosittua. Silti niitä vastaan löydetyistä puolustusmekanismeista monet eivät välttämättä ole erityisen käytännöllisiä. Gollen ja Ducheneautin (2005) esittämät CAPTCHA-järjestelmät, vaikkakin mahdollisesti tehokkaita, vaikuttavat varsin kankeilta ratkaisuilta, joiden implementointi ilman peliin uppoutumisen rikkomista voi osoittautua mahdottomaksi. Botteja analysoimalla ja tarkkailemalla niiden ominaisuuksista saadaan lisää tietoa, ja bottien käyttäjiä voidaan asettaa käyttökieltoon yhä tehokkaammin, mutta tällöin kyseessä on kenties ikuinen kilpajuoksu, jossa bottien kehittäjät pyrkivät luomaan entistä vaikeammin löydettäviä botteja ja pelien kehittäjät joutuvat luomaan entistä parempia havainnointikeinoja. Mikään löydetyistä tekoälypohjaisen huijauksen ehkäisykeinoista ei kuitenkaan

varsinaisesti luo kuvaa, jossa boteista päästäisiin täysin eroon, ilman pelaamiseen tehtyjä suuria uhrauksia.

Samanlaisia vajavuuksia voidaan havaita myös sosiaalista huijaamista estettäessä. Kolluusion estäminen varsinkin sosiaalisiksi suunnitelluissa peleissä voi olla erittäin haastavaa ilman, että mahdollisuuksia sosialisointiin rajoitetaan liikaa. Joissain tapauksissa kolluusion estäminen on kuitenkin selvästi helpompaa. Yanin (2003) esimerkissä, jossa huijaajat vaihtelevat voittajan kesken, voidaan yksinkertaisesti muuttaa pelissä käytettävää tilastointitapaa, jolloin pelin hävinneen pelaajan taso tilastoissa laskee. Tällöin voittojen vaihtelu ei tuota huijaajille mitään etua. Ainoa ongelma joka tästä voi mahdollisesti ilmentyä, vanhojen pelaajien hyväksynnän saaminen muutokselle, varsinkin jos pelissä käytetty järjestelmä on ollut käytössä jo pitkään. Tämä on kuitenkin varsin pieni hinta siitä, että tilastojen oikeellisuus voidaan taata. Myös automaattiset huijauksen havainnointi järjestelmät voivat olla hyödyllinen keino havaita kolluusiota.

Autentikointiin liittyvät huijausmetodit perustuvat pääosin joko heikkoon tiedon salaamiseen tai käyttäjän manipulointiin. Näille löydetyt ratkaisut ovat varsinkin moniin muihin ongelmakohtiin verrattuna varsin yksinkertaisia toteuttaa. Kuten Yan ja muut (2000) mainitsevat, yksinkertaisella tietojen siirrossa käytettävän menetelmän muuttamisella voidaan helposti vähentää autentikointitietojen päätymistä vääriin käsiin. Lisäksi esitetyt salasanojen luontiin käytetyt menetelmät eivät ole monimutkaisia toteuttaa ja ne lisäävät turvallisuutta huomattavasti.

Kuten Geigner (2014) kirjoittaa, ollaan tilanteessa, jossa verkkopelissä huijaamisesta on vangittu pelaajia, kun pelin ylläpitäjä on vienyt asian viranomaisille. Vaikka kyseessä onkin vielä tässä vaiheessa yksittäinen tapaus, osoittaa se kuitenkin, että tulevaisuudessa on mahdollista joutua vastaamaan teoistaan viattomaltakin tuntuvasta huijauksesta. Tämä antaa toisenlaisen näkökannan siihen, miksi pelaajien kouluttaminen huijaamisen suhteen on suotavaa. Luonnollisesti pelaajia voidaan opastaa varsin normaaleissa turvallisuuskäytänteissä, kuten salasanojen ja verkkoyhteyksien suojaamisessa, mutta voidaan myös paneutua etiikkaan ja moraalisiin asioihin. Lisäksi pelin ylläpitäjien ja pelaajien välillä tapahtuva kommunikointi mahdollistaa pelaajien pitämisen ajan tasalla uusista uhkatilanteista. Pelaajien rankaisuun voidaan käyttää myös pelinsisäisiä keinoja. Näistä helpoin ja lopullisin on käyttökielto, jolloin kieltoon asetettu tili ei voi enää kirjautua peliin. Tämä on tehokas keino, jos yhden tilin estämisellä saadaan aikaan suuri menetys huijaajalle. Toisaalta jos huijaaja luo uuden tilin ja jatkaa huijaamista, esimerkiksi kerätäkseen ja myydäkseen virtuaalista omaisuutta, käyttökiellolla ei välttämättä ole suurta hyötyä. Vaikka kiinni jääneiden rankaisu onkin tärkeää, tällöin vahinko on jo tapahtunut, joten suurempi painoarvo tulisi antaa ennaltaehkäisyyn.

5. Johtopäätökset

Tutkimuksessa tarkasteltiin aihetta seuraavan tutkimuskysymyksen kautta: **Minkälaisia huijaustapoja verkkopeleissä huijaavat käyttävät ja miten näitä voitaisiin estää?** Huijaajat käyttävät sekä tietoteknisiä tapoja että sosiaalisia huijaustapoja. Tietoteknisiin metodeihin kuuluvat peliservereihin ja muihin pelaajiin kohdistuvat hyökkäykset, muiden pelaajien autentikointitietojen kalastelu tietoja urkkimalla, pelin heikkouksien, kuten suunnitteluvirheiden, hyväksikäyttäminen sekä tekoälypohjainen huijaus, jolloin käytetään botteja suorittamaan pelissä esiintyviä tehtäviä huijaajan puolesta. Sosiaalisia menetelmiä ovat muiden huijaajien kanssa tehtävä yhteistyö, käyttäjien manipulointi sekä virtuaalisen omaisuuden väärinkäyttö. Näiden estämisessä menetelmät ovat pääosin pelin järjestelmien parantaminen. Voidaan luoda erillisiä ohjelmistoja, jotka havaitsevat väärinkäyttöä, parantaa olemassa olevan tiedon suojaamista ja muokata pelin ominaisuuksia niin, että käytetyt menetelmät eivät ole enää mielekkäitä käyttää. Lisäksi pelaajia voidaan kouluttaa turvallisuuskäytänteistä, jolloin heillä on parempi käsitys, miten parantaa omaa turvallisuutta ja välttää huijaamiseksi joutuminen.

Tehty tutkimus tiivistää jo olemassa olevaa tutkimusta ja luo selvemmän yleiskuvan siitä, miten huijaaminen tapahtuu ja miten sitä voidaan estää. Käytännön toimijoille tutkimusta voidaan soveltaa esimerkiksi pelien turvallisuutta suunnitellessa, jolloin pelin kehittäjät voivat tutkimuksen avulla varautua ja toteuttaa tarvittavia turvallisuustoimia. Tutkimus on kuitenkin rajallinen, koska siinä suoritettiin vain kirjallisuuskatsaus, eikä varsinaista empiiristä tutkimusta.

Mahdollisia jatkotutkimusongelmat voisivat liittyä varsinkin tekoälypohjaiseen huijaamiseen, koska se on alati kehittyvä ja suosittu huijausmuoto ja nykyiset ehkäisymenetelmät eivät varsinaiset tuoneet esille yhtä ja varmaa toimivaa ratkaisua.

Lähdeluettelo

Be'ery, T. (2014). *Brute-force Attacks: Crossing the Online-Offline Password Chasm.* Lainattu 13.2.2016, saatavilla: http://www.securityweek.com/brute-force-attacks-crossing-online-offline-password-chasm

Ben-Or, M., Goldreich, O., Micali, S., & Rivest, R. L. (1990). A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, *36*(1), 40-46.

Bethea, D., Cochran, R. A., & Reiter, M. K. (2011). Server-side verification of client behavior in online games. *ACM Transactions on Information and System Security (TISSEC)*, *14*(4), 32.

Blackburn, J., Kourtellis, N., Skvoretz, J., Ripeanu, M., & Iamnitchi, A. (2014). Cheating in online games: a social network perspective. *ACM Transactions on Internet Technology (TOIT)*, *13*(3), 9.

Bono, S., Caselden, D., Landau, G., & Miller, C. (2009). Reducing the attack surface in massively multiplayer online role-playing games. *IEEE Security & Privacy*, (3), 13-19.

Boyd, S. W., & Keromytis, A. D. (2004, January). SQLrand: Preventing SQL injection attacks. In *Applied Cryptography and Network Security* (pp. 292-302). Springer Berlin Heidelberg.

Chen, K. T., Liao, A., Pao, H. K. K., & Chu, H. H. (2008). Game bot detection based on avatar trajectory. *Proceedings of the 7th International Conference on Entertainment Computing,* 94-105

Chen, V. H. H., Duh, H. B. L., & Ng, C. W. (2009). Players who play to make others cry: The influence of anonymity and immersion. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology*, 341-344

Chow, M., McGraw, G. (2009). Securing Online Games: Safeguarding the Future of Software Security. *IEEE Security & Privacy, 7(3),* 11-12.

Consalvo, M. (2007). *Cheating: Gaining Advantage in Videogames*. Cambridge (MA): MIT Press.

Cowan, C., Pu, C., Maier, D., Walpole, J., Bakke, P., Beattie, S., & Hinton, H. (1998). StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks. *Proceedings of the 7th Usenix Security Symposium*, 63-78.

De Paoli, S., & Kerr, A. (2009). The Cheating Assemblage in MMORPGs: Toward a sociotechnical description of cheating. *Digra '09 - Proceedings of the 2009 Digital Games Research Association International Conference: Breaking New Ground: Innovations in Games, Play, Practice and Theory*, 1-12.

De Paoli, S., & Kerr, A. (2009). " We Will Always Be One Step Ahead of Them" A Case Study on the Economy of Cheating in MMORPGs. *Journal For Virtual Worlds Research*, *2*(4).

Duh, H., & Chen, V. (2009). Cheating behaviors in online gaming. *Proceedings of the 4th International Conference of Online Communities and Social Computing*, 567-573.

Feng, W. C., Kaiser, E., & Schluessler, T. (2008). Stealth measurements for cheat detection in on-line games. In *Proceedings of the 7th Association for Computing Machinery's Special Interest Group on Data Communications Workshop on Network and System Support for Games*, 15-20.

Florêncio, D., & Herley, C. (2010). Where do security policies come from? *In Proceedings of the Sixth Symposium on Usable Privacy and Security*, 10.

Geigner, T. (2014). *The Future Is Now: Cheating In Online Games Leads To Arrests In Japan.* Lainattu 31.1.2016, saatavilla https://www.techdirt.com/articles/20140625/08443327682/future-is-now-cheating-online-games-leads-to-arrests-japan.shtml

Gianvecchio, S., Wu, Z., Xie, M., & Wang, H. (2009). Battle of botcraft: fighting bots in online games with human observational proofs. *Proceedings of the 16th ACM conference on Computer and Communications Security*, 256-268.

Golle, P., & Ducheneaut, N. (2005). Preventing bots from playing online games. *ACM Computers in Entertainment (CiE)*, *3*(3), 3-3.

Gu, Q., Liu, P., Zhu, S., & Chu, C. H. (2005). Defending against packet injection attacks unreliable ad hoc networks. In *Global Telecommunications Conference, 2005. GLOBECOM'05, 3,* 5.

Gupta, S. (2012). Buffer Overflow Attack. *IOSR Journal of Computer Engineering*, *1(1),* 10-23.

Heinrich, D. F., Le, H. Q., Waldorf, R. O., & Angelo, M. F. (2001). *U.S. Patent No. 6,199,167*. Washington, DC: U.S. Patent and Trademark Office.

Hoglund, G., McGraw, G. (2007) *Exploiting online games: cheating massively distributed systems*. Boston, MA: Addison-Wesley Professional.

Hsu, H. Y., Zhu, S., & Hurson, A. R. (2007). LIP: a lightweight interlayer protocol for preventing packet injection attacks in mobile ad hoc network. *International Journal of Security and Networks*, *2*(3-4), 202-215.

Jeff Yan, J., & Choi, H. J. (2002). Security issues in online games. *The Electronic Library*, *20*(2), 125-133.

Lan, X., Zhang, Y., Pin, X., (2009) An Overview on Game Cheating and Its Counter-measures. *Proceedings of the Second Symposium International Computer Science and Computational Technology*, 195-200.

Lewis, C., Whitehead, J., & Wardrip-Fruin, N. (2010, June). What went wrong: a taxonomy of video game bugs. In *Proceedings of the fifth international conference on the foundations of digital games*, 108-115.

Loukas, G., & Öke, G. (2009). *Protection against denial of service attacks: a survey*. Lainattu 12.5.2016, saatavilla: https://pdfs.semanticscholar.org/cdd9/1609bb48cb0a43d56c51c87e293564d747c9.pdf

Mitterhofer, S., Kruegel, C., Kirda, E., & Platzer, C. (2009). Server-side bot detection in massively multiplayer online games. *IEEE Security & Privacy*, *7(3),* 29-36.

Mørch, K. H. T. (2003). *Cheating in Online Games–Threats and Solutions*. Lainattu 12.5.2016, saatavilla: http://publications.nr.no/directdownload/publications.nr.no/Cheating_in_Online_Games.pdf

Pipkin, D. L. (2003). *Halting the hacker: A practical guide to computer security*. Upper-Saddle River (NJ): Prentice Hall Professional.

Prashar, N., Shao, C., Jiwani, A., & Malekzadeh, A. (2009) *Security Analysis of Terra: Battle for the Outlands.* Lainattu 31.1.2016, saatavilla: http://courses.ece.ubc.ca/cpen442/previous_years/2009/term_project/reports/2009/Terra_analysis.pdf

Pritchard, M. (2000). *How to Hurt the Hackers: The Scoop on Internet Cheating and How You Can Combat It*. Lainattu 31.1.2016, saatavilla: http://www.gamasutra.com/view/feature/3149/how_to_hurt_the_hackers_the_scoop_.php

Randell, B., Yan, J. (2009). An Investigation of Cheating in Online Games. *IEEE Security & Privacy. 7(3),* 37-44.

Sethi, A., Allen, R. (2014). *Defending online games from piracy, cheating and fraud*. Lainattu 31.1.2016, saatavilla: http://www.develop-online.net/analysis/defending-online-games-from-piracy-cheating-and-fraud/0198678

Smed, J., Knuutila, T., & Hakonen, H. (2006). Can we prevent collusion in multiplayer online games. In *Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006),* 168-175.

Wittman, A. (2009). *SQL Injection: The Fastest Growing Security Threat*. Lainattu 20.2.2016, saatavilla: http://www.networkcomputing.com/careers/sql-injection-fastest-growing-security-threat/262888996

Yan, J. (2003). Security design in online games. In *Proceedings of the 19th Annual Computer Security Applications Conference,* 286-295.

Yan, J. (2004). Password memorability and security: Empirical results. *IEEE Security & privacy*, *2(5),* 25-31.

Yan, J. J. (2001). A note on proactive password checking. In *Proceedings of the 2001 workshop on New security paradigms*, 127-135.

Yan, J., Early, S., & Anderson, R. (2000). The xenoservice-a distributed defeat for distributed denial of service. In *Proceedings of the Information Survivability Workshop.*