



UNIVERZITET SINGIDUNUM

**DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE I
MEĐUNARODNU SARADNJU**

STUDIJSKI PROGRAM

SAVREMENE INFORMACIONE TEHNOLOGIJE

**Resursi, sigurnost i autentifikacija kod online video
igara**

- MASTER RAD -

Mentor:

prof. dr Mladen Veinović

Student:

Mihajlo Jovanović

410274/2012

Beograd, Novembar 2013.

SADRŽAJ

1. UVOD	2
2. PROBLEMI RESURSA	4
2.1 SPECIFIČNOSTI PO PITANJU ZAHTEVNOSTI RESURSA	4
2.2 POJAM „MASSIVE MULTIPLAYER ONLINE GAME“ – MMOG	7
2.3 STANDARDNE ARHITEKTURE MMOG SISTEMA.....	8
2.3.1 <i>Klijent – server arhitektura</i>	8
2.3.2 <i>Peer – to – peer (P2P) arhitektura</i>	9
2.4 PREDLOZI I RAZVOJ NOVIH REŠENJA.....	11
2.4.1 <i>Virtualizacija i Cloud computing</i>	11
2.4.2 <i>Poboljšanja kod P2P arhitekture</i>	15
2.4.3 <i>Praćenje i predviđanje opterećenosti servera</i>	17
2.4.4 <i>Hibridna arhitektura</i>	18
3. CHEATS – VARANJE.....	20
3.1 POJAM I ISTORIJAT.....	20
3.2 UZROCI VARANJA	21
3.3 TIPOVI VARANJA	23
3.3.1 <i>Varanje putem neosnovanog poverenja</i>	23
3.3.2 <i>Varanje u dosluhu sa drugima</i>	24
3.3.3 <i>Varanje putem iskorišćavanja procedura u igri</i>	25
3.3.4 <i>Varanje putem iskorišćavanja veštačke inteligencije</i>	29
3.3.5 <i>Varanje putem iskorišćavanja grešaka (bugs) u programiranju</i>	31
3.3.6 <i>Varanje putem unutrašnje zloupotrebe</i>	32
4. OSTALE SIGURNOSNE PRETNJE.....	34
4.1 NEISPUNJAVANJE DOGOVORA OKO RAZMENE RESURSA	34
4.2 NAPADI MODIFIKOVANJEM IZLAZNIH INFORMACIJA NA KLIJENT/SERVER PROTOKOL.....	36
4.3 (D)DOS NAPADI	37
5. AUTENTIFIKACIJA	40
5.1 AUTENTIFIKACIJA KOD ONLINE IGARA	41
5.2 REŠENJA ZA POBOLJŠAVANJE PROCESA AUTENTIFIKACIJE	46
5.2.1 <i>Multifaktorska autentifikacija</i>	46
5.2.2 <i>Autentifikacija kao igra ili njen deo</i>	48
6. SLUČAJ DIABLO III.....	53
7. ZAKLJUČAK	57
LITERATURA.....	58

1. Uvod

Umrežavanje u informacionim tehnologijama predstavlja veoma značajnu tehnologiju. Pojavom Interneta, kao svetske mreže, moguće je komunicirati sa osobom koja se nalazi na drugoj strani planete, što otvara dosta novih mogućnosti. Takođe, to dovodi i do napretka postojećih tehnologija i vrsta aplikacija. Video igre takođe spadaju u jednu od tih vrsta.

Video igre se mogu podeliti po različitim kriterijumima, od kojih je jedan i to koliko igrača podržavaju, tj. da li je igra pravičana za igranje od strane jedne osobe u datom periodu (*single - player*) ili je pravičana sa idejom da je grupa ljudi igra zajedno u istom vremenskom periodu (*multi - player*). Danas je kod većine naslova čest slučaj da su obe opcije prisutne.

U periodu pre masovne ekspanzije umrežavanja i Interneta, princip igranja *single - player* i *multi - player* igara se nije preterano razlikovao. Jedna ili više osoba bi sela pored uređaja sa igrom, i u zavisnosti od tipa igre, vršila određene akcije u okviru iste, radi dostizanja određenog cilja. Problem kod *multi - player* igara u to vreme je bio sledeće prirode: nedostatak igrača na datom mestu u datom periodu. Recimo da neko želi da igra određeni naslov ali nije u mogućnosti jer nema sa kime da igra, bilo zato što je osoba sama na nekoj lokaciji, ili zato što niko u njenom okruženju nije zainteresovan za taj naslov. Pored toga, neki od žanrova video igara, gde se koristi kompjuterski miš i gde uz pomoć kursora miša upravljate likom, isključivo su mogli da funkcionišu kao *single - player* naslovi.

Umrežavanje je rešilo jedan od tih problema. Naime, povezivanjem više računara putem lokalne mreže, omogućeno je, umesto da na jednom računaru budu svi igrači, da svaki igrač ima svoj računar. Pored povećanog komfora, to je takođe omogućilo i da pomenute igre koje koriste miš mogu da se prave i sa opcijom za više igrača. Ali, lokalna mreža je i pored svoje veličine i dalje lokalna, i zahteva da igrači budu ipak relativno blizu jedan drugome. Kao što je već rečeno, razvoj Interneta je učinio mogućim komunikaciju sa ljudima širom sveta, i samim time, otvorio je i nove mogućnosti kada su u pitanju *multi - player* igre. Sada više nije bilo ograničenja prostorom, tako da su time višestruko povećane mogućnosti uživanja u naslovima, kao i izbor igrača sa kojima se može igrati. Takav pristup je postao toliko popularan, da je doveo do stvaranja nove vrste takozvanih online igara, koji predstavljaju tip igara koji isključivo može da se igra preko Interneta, zajedno sa hiljadama i hiljadama drugih igrača istovremeno.

Idealno bi bilo da se sve na tome završi, ali nažalost to nije slučaj ovde. Korišćenje Interneta kod *multi - player* igara je dovelo do mnoštva novih problema, za čije rešavanje je potrebno uložiti puno truda, vremena, i resursa. Navedeni problemi su dodatno eskalirali kod online igara, gde bukvalno ne postoji sistem ni organizacija koja može da reši sve iznenadne poteškoće, koje maltene predstavljaju svakodnevnicu u funkcionisanju jednog informacionog sistema vezanog za određeni online naslov.

Ovde bi trebalo spomenuti, da takođe postoje razni kriterijumi od kojih barem neki moraju biti ispunjeni da bi video igre bile uspešne. Zabavnost, inovativnost, priča i likovi, kao i razni drugi elementi umeju biti presudni za uživanje u određenom naslovu. Sve to naravno zavisi od tipa naslova, i kranjeg cilja koji razvojni tim želi postići.

Kada su online igre u pitanju, imamo posebnu grupu kriterijuma specifičnu za pomenuti tip. Koliko god određeni online naslov bio dobro odrađen, to ništa ne znači ako korisnici nisu u stanju da ga igraju kako treba. Uzmimo na primer online simulaciju šaha. Koliko bi igra bila zabavna da recimo jedan igrač u toku partije dobija mogućnost izvođenja dva poteza za redom, dok drugi igrač može izvršiti samo jedan? Ako izuzmemo neke ekstremne slučajeve, velika većina ljudi ne bi bila zainteresovana da tako nešto isproba. Ekvivalentno tome, u online igrama gde su potrebni brzi refleksi, teško da će bilo ko tolerisati kašnjenje u odzivu komandi, i pored svih drugih dobro odrađenih i originalnih elemenata i ideja prisutnih u konkretnom naslovu. Takođe, kao bitan problem sa povećanjem broja korisnika nameće se i autentifikacija istih. Što je više ljudi prisutno, postoji veća mogućnost zloupotrebe i prevare korišćenjem tuđeg identiteta. Samim time, potrebno je sam proces autentifikacije detaljno razraditi, i sprečiti bilo kakvu mogućnost iskorišćavanja procesa od strane zlonamernih lica. Sve ovo takođe spada u probleme i izazove sa kojima se razvojni timovi suočavaju danas.

U današnje vreme, mnogi ljudi često postavljaju pitanje zašto se uopšte baviti bilo kojim problemima vezanim za video igre generalno, jer i dalje postoji određeno verovanje da su video igre laka zabava. Međutim, dovoljno je pogledati kolika su ulaganja potrebna za stvaranje uspešnog naslova, kakvi uslove je potrebno ispuniti, kao i sam profit koji se može ostvariti, da svako može shvatiti da se tu više ne radi o naivnim izazovima i dobicima.

Kod velikog dela online igara, posebno kada su u pitanju popularniji naslovi, prisutan je određeni sistem naplate za igranje. Jedan od najpopularnijih je sistem mesečne pretplate – korisnici plaćaju određenu sumu novca za vremenski period od jednog meseca, tokom kojeg imaju pravo da igraju igru. Čim prođe mesec dana, pristup igri biva blokiran za igrača, dok ne uplati za sledeći mesec.

Uzmimo sada za primer neku online igru, čija je mesečna pretplata 15 američkih dolara, što je recimo neka prosečna cena pretplate. Ako je ta igra veoma popularna, ona verovatno ima par miliona igrača svakog meseca. Dakle, 15 dolara na svaku od tih par miliona osoba, na mesečnom nivou. Već se tu vidi koliki je obrt novca u pitanju, i koliko je to zapravo unosan biznis. Samim time, može se pretpostaviti i koliki uticaj ima (ne)zadovoljstvo korisnika u svemu tome. Poznati su primeri velikog broja online igara koje su propale zato što nisu primenjivale dobre metode zaštite, čime su automatski gubile veliki broj korisnika, koji nisu bili voljni da budu žrtve hakerskih napada i ostalih malverzacija.

Cilj ovog rada je prikaz sigurnosnih problema kod online igara, uključujući i autentifikaciju, kao i zahtevnost po pitanju resursa, sa preporukama za njihovo rešavanje.

2. Problemi resursa

2.1 Specifičnosti po pitanju zahtevnosti resursa

Uvek je potrebno imati u vidu da, u zavisnosti od tipa igre tj. žanra, resursi potrebni za efikasno online funkcionisanje istih različiti, i često umeju biti veoma specifični. Uzećemo za primer poređenja dva tipa video igara, simulacije šaha sa jedne strane, i sa druge video igre koje spadaju u žanr borilačkih igara (fighting games, skraćeno FG). Takođe pretpostavljamo da obe igre poseduju online funkcionalnosti (igranje online protiv drugih igrača), kao i mogućnosti igranja naslova na jednom uređaju, tj. igranja offline (bilo protiv AI-a, ili protiv drugih igrača).

Ako uzmemo primer simulacije šaha, online meč bi se zasnivao na povezivanju 2 igrača, A i B, i nakon određivanja strana, igrači bi se smenjivali po potezima. Server na koji su igrači povezani prati njihovu partiju, beleži broj poteza, beleži koja je figura pomerena i u koje vreme itd. Redosled događaja je generalno sledeći: igrač A (za koga pretpostavljamo da je prvi na potezu, tj. da koristi bele figure) povlači svoj potez. Server onda prihvata informaciju, i beleži koji je potez po redu, koju je figuru igrač A pomerio, odakle ju je pomerio i gde ju je pomerio. Server može te podatke dodatno analizirati (da li je pomenuti potez validan, kada je vremenski izvršen itd.), i nakon utvrđene validnosti, poslati ih igraču B. Kada klijent na strani igrača B primi podatke, oni se automatski izvršavaju, igrač B dobija informaciju šta je to igrač A odigrao, i da je red sada na B da donese svoju odluku, nakon čega se postupak ponavlja, samo ovaj put u suprotnome smeru. Održati ovakav sistem zahteva određene resurse, koji eksponencijalno rastu sa povećanjem broja igrača i partija koje se paralelno odigravaju, primenom različitih sistema zaštite radi sprečavanja zloupotreba, kao i uvođenjem vremenskih ograničenja po potezu.

Kada su borilačke igre u pitanju, imamo dodatni problem: potezi izvođeni od strane igrača su paralelni tj. nema utvrđenog redosleda. Samim time, kašnjenje u odzivu komandi (eng. lag) je mnogo bitnije pitanje u odnosu na šah. Uzećemo za primer igru *The King of Fighters XIII*, koja radi u 60 FPS-a (Frames per Second, tj. broj slika u sekundi). Igranje određenog naslova online uvek podrazumeva i određen lag, zbog činjenice da je uvek potrebno vreme da informacija stigne od jedne strane drugoj, makar sami uređaji bili maltene na par metara udaljenosti.

Recimo da su igrači A i B uzeli da igraju *The King of Fighters XIII* , ali, pošto se nalaze na određenoj udaljenosti, potrebno je vreme da informacija dodje od igrača A do B i obrnuto, pritom, to vreme se meri u milisekundama (ms), i to se zove mrežno kašnjenje (eng. network delay). Uzmimo da je vrednost 100 ms, što predstavlja deseti deo sekunde. Već smo rekli da igra radi u 60 FPS-a, što konkretno znači da, ako recimo igrač A izvede neki potez na 10. frejmu, igrač B će to kod njega videti tek na 16. frejmu (pošto je kašnjenje $60 / 10 = 6$). Naravno, sada se postavlja pitanje, zašto je to toliko bitno kada je u

pitanju svega delić sekunde? Nažalost, to je veoma velika razlika kada su u pitanju borilačke igre. Same igre su često veoma brzog tempa, kao i potezi koji izvede likovi u okviru istih. Opet, recimo da potez koji je igrač A izveo u 10. frejmu traje 3 frejma, i nakon toga, u 13. frejmu ponovo izvede isti potez. Sa napomenutom razlikom od 6 frejma, igrač B u svom 16. frejmu vidi samo da je izgubio energiju, a ni sam ne razume zašto. Iz navedenog se da videti da je ovo jedan veoma veliki problem.

Kako onda rešiti navedeni problem? Najčešće upotrebljavano rešenje je korišćenje input-delay tehnike. Šta to podrazumeva? Kada u navedenom slučaju igrač A izvede potez na 10. frejmu, sistem uzimajući u obzir lag, igru za igrača A usporava toliko, da i kod njega tek na 16. frejmu potez biva izvršen. Samim time, igrači su na taj način izjednačeni. Ovo je zapravo slučaj sinhronizacionog kašnjenja (eng. synchronization delay), koji se svodi na to da server čeka sve klijente, pa onda upoređuje njihove rezultate, i onda šalje povratne informacije. Čini se da je time rešen problem, međutim, nije tako. Borilačke igre se često igraju i offline, tj. uživo protiv drugih igrača. U takvim slučajevima, usporavanja nema, samim time dolazi do problema izvođenja poteza, gde je način ili osećaj kako se i kojom brzinom potezi izvedu dosta drugačiji online i offline. Samim time, igrači često razvijaju loše navike, i njihov način igranja online se pokaže neefikasnim u offline mečevima, što izaziva frustracije i odustajanje od dalje igre, čime opet dolazi do problema za developere.

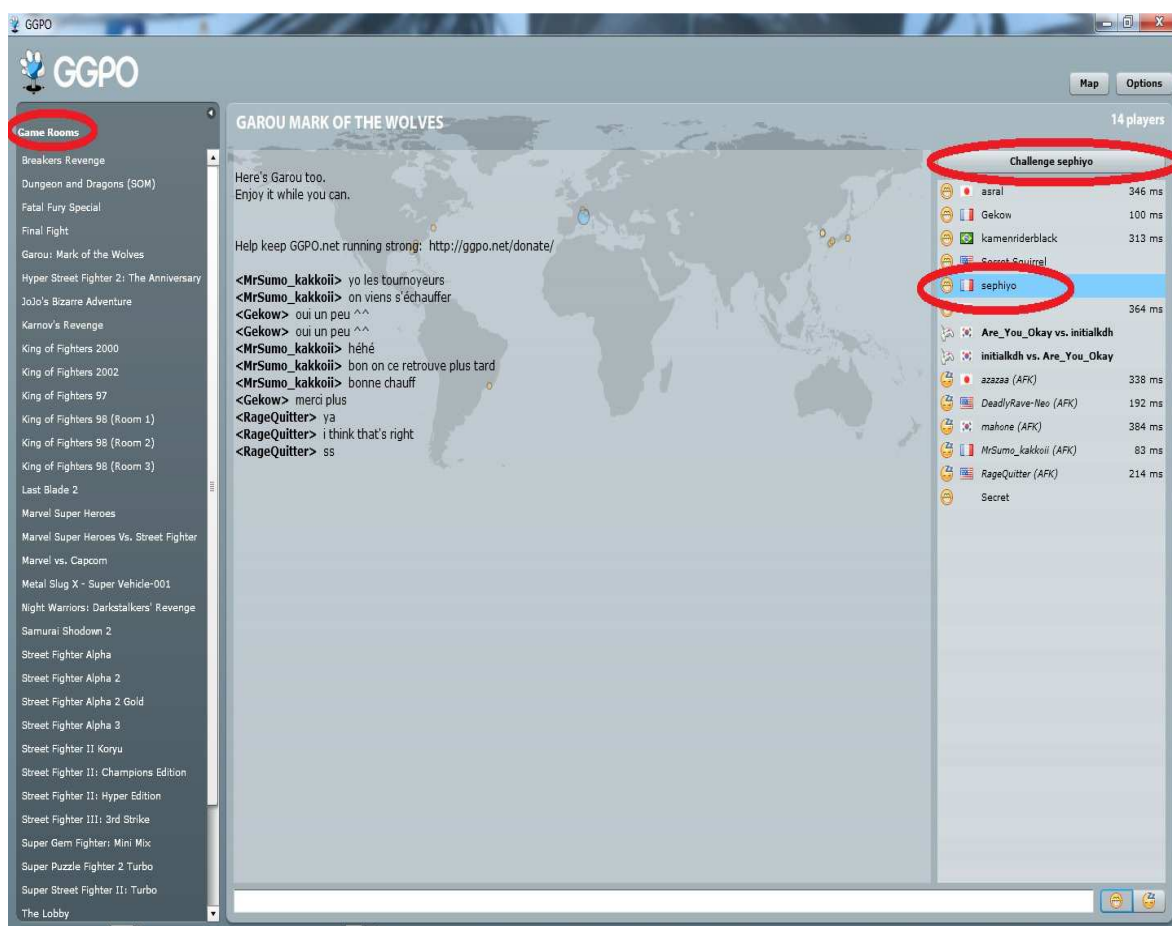
Ima li onda načina da se igranje borilačkih igara offline i online izjednače? Odgovor na to nudi middleware GGPO (Good Games, Peace Out)¹. Kako funkcioniše GGPO? Ideja je da se omogući igračima da igraju bez prisustva laga, tj. da ako želi, igrač A može da podesi vrednost laga na 0, čime se svaka komanda koju on izvrši dešava trenutno, bez kašnjenja. Tako recimo igrač A može uneti komandu na 10. frejmu, i nakon toga on u 13. frejmu vidi kako njegov lik udara lika kojim upravlja B. Ali, vreme za prenos informacija je i dalje prisutno, i igra ne zna kako je B reagovao na napade od A, već koristi algoritam predviđanja, koji na osnovu prethodnih inputa od B računa koje je inpute B verovatno uneo. Konkretno, na strani A, igra može da predvidi da je lik od B pogođen napadom tj udaren. U međuvremenu, kada se dođe do 16. frejma, i B uspe da blokira napad A (što je u ovome slučaju realno stanje), igra registruje da je predviđanje pogrešno, zatim vraća 3 frejma unazad i prikazuje lika kojim upravlja B u stanju blokiranja, a ne u stanju kao da je udaren, i cilj je da se to vraćanje unazad (eng. rollback) uradi što je brže moguće. Nažalost, te same ispravke umeju biti primetne kao „seckanje“, a što je veza između A i B lošija, ispravke su češće. Ali, u tom slučaju, ako to smeta B, on može izaberati vrednost laga od npr. 6, čime se kod njega igra usporava, i odziv komandi je manji, ali zato „seckanja“ nema. Dakle, glavna prednost korišćenja GGPO sistema je mogućnost izbora između trenutnog odziva ili eliminisanja ispravljanja, tako da sve zavisi od samih igrača.

Sam sistem je nastao radi omogućavanja igranja raznih popularnih borilačkih igara sa drugih sistema na PC platformi. Igrači bi se konektovali na GGPO server, i onda bi putem korišćenja emulatora nFBA i posedovanjem odgovarajućih kopija igara mogli da se

¹ <http://ggpo.net/>

međusobno takmiče u datoj igri. U konkretnom slučaju igrači A i B bi imali pokrenute kopije igre kod njih, ali bi i server na koji su konektovani takođe pokrenuo svoju kopiju. Zatim, uzevši u obzir podešavanja i kod A i kod B, server bi primao informacije od klijenata, i proveravao sa stanjem svoje kopije. Na osnovu svega toga, server bi ustanovio da li je sve u redu, i u slučaju da nije, poslao bi nazad nazad klijentima zahtev da se vrate na prethodno ispravno stanje. Sve se to dešava toliko brzo, da u dobrim uslovima ljudsko oko ne može ni da primeti rollback.

Nažalost GGPO je, iako popularan i efikasan, po pitanju resursa zahtevan. Ako uzmemo konkretan slučaj sa nFBA i nekom igrom za njega, dovoljno je imati računar sa procesorom tipa Intel Pentium III brzine 700 MHz radi ugodnog igranja, bilo offline, bilo putem interneta. Ali, ako na to dodamo GGPO, zahtevnost se povećava na računar sa Intel Pentium 4 procesorom brzine 1,4 GHz, dakle maltene duplo veća brzina, i to su zahtevi samo sa klijentske strane. Server sa druge strane, omogućava da veliki broj parova igrača igra istovremeno, i za svaki par ima posebnu kopiju igre upaljenu, tako da se može zamisliti kolika je tu tek zahtevnost po pitanju resursa. Upravo iz datih razloga, GGPO nije još zaživeo kod novijih naslova.

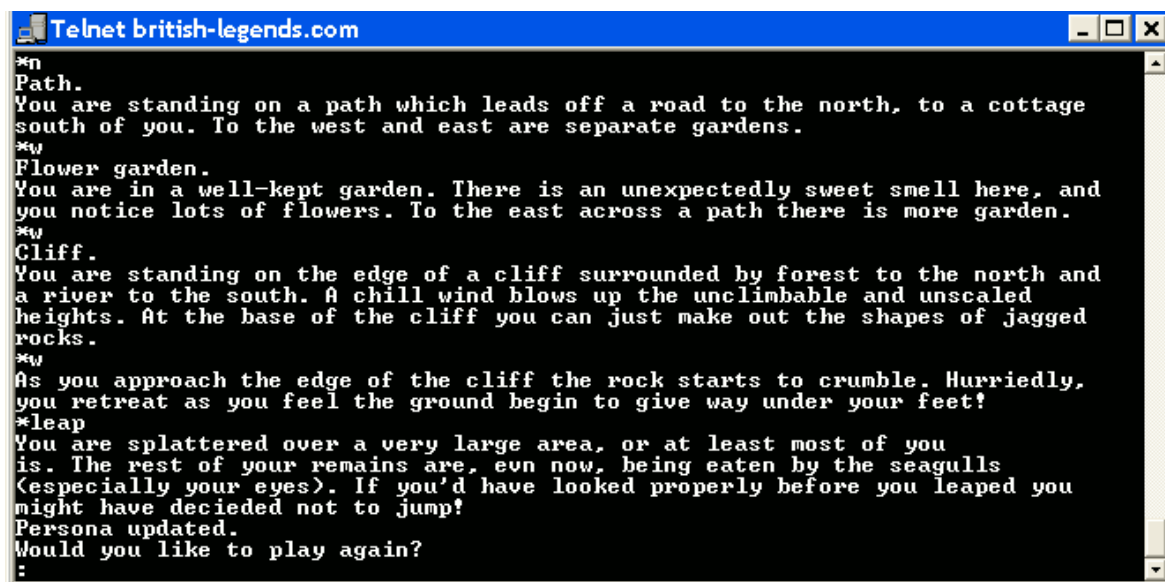


Slika 1: Prikaz GGPO aplikacije za PC

2.2 Pojam „Massive Multiplayer Online Game“ – MMOG

U prethodnom primeru smo govorili o tipovima igara koje poseduju offline i online opcije igranja. Međutim, postoji još jedan oblik video igara, poznat kao „Massive Multiplayer Online Game“ (skraćeno MMOG). Ovde je reč o igrama koje zahtevaju vezu putem Interneta sa serverima, i to je jedini način da se date igre igraju.

Pozabavimo se malo istorijatom MMOG igara. Počeci MMOG igara se zasnivaju na pojavi „Multi User Dungeon“ (skraćeno MUD) igara sredinom '70-ih godina prošlog veka. MUD igre zapravo predstavljaju online tekstualne avanture, čiji se svet sastoji iz velikog broja „soba“ kojima igrači pristupaju. U okviru svake sobe nalaze se drugi igrači, kompjuterski vođeni likovi, kao i predmeti, i igrač putem tekstualnih komandi vrši interakciju sa pomenutim elementima.



```
Telnet british-legends.com
*
Path.
You are standing on a path which leads off a road to the north, to a cottage
south of you. To the west and east are separate gardens.
*
Flower garden.
You are in a well-kept garden. There is an unexpectedly sweet smell here, and
you notice lots of flowers. To the east across a path there is more garden.
*
Cliff.
You are standing on the edge of a cliff surrounded by forest to the north and
a river to the south. A chill wind blows up the unclimbable and unscaled
heights. At the base of the cliff you can just make out the shapes of jagged
rocks.
*
As you approach the edge of the cliff the rock starts to crumble. Hurriedly,
you retreat as you feel the ground begin to give way under your feet!
*leap
You are splattered over a very large area, or at least most of you
is. The rest of your remains are, evn now, being eaten by the seagulls
(especially your eyes). If you'd have looked properly before you leaped you
might have decieded not to jump!
Persona updated.
Would you like to play again?
:
```

Slika 2: Primer MUD igre

MMOG igre su uzele osnovu MUD igara, i nadogradile je. Glavna nadogradnja predstavlja upotreba grafičkih elemenata umesto teksta za prikaz virtuelnog sveta, i svega što se u njemu nalazi. Druga značajna promena predstavlja podršku za veći broj igrača u odnosu na pre. Ovde možemo govoriti o ciframa reda veličine 10 miliona, sa tendencijom rasta, u naslovima kao što su npr. „World of Warcraft“. Vidi se da koliki je razvoj u pitanju.

Veliki broj igrača je jedan faktor, ali takođe, treba imati u vidu i velike mogućnosti interakcije, kako između igrača i elemenata u okviru igre, tako i između njih samih. Samim time, ako uzmemo u obzir i prethodne primere, možemo zaključiti da je zahtevnost po pitanju resursa kada su MMOG igre u pitanju veoma velika, posebno ako govorimo o popularnim naslovima. Neke od potrebnih uslova koji omogućavaju da naslov

funkcioniraju kako treba podrazumevaju: mogućnost pravovremenog reagovanja uprkos kašnjenju prouzrokovanoj samom konekcijom, pouzdan rad sistema, pouzdani i efikasni sistemi zaštite, veštačka inteligencija za likove kontrolisane od strane servera, brz odziv na naredbe od strane klijenata, itd.

U skladu sa svim navedenim, potrebno je naći efikasna i ekonomski isplativa rešenja za funkcionisanje modernih MMOG igara, što predstavlja veoma izazovan problem.

2.3 Standardne arhitekture MMOG sistema

2.3.1 Klijent – server arhitektura

Klijent – server arhitektura predstavlja najpopularnije rešenje kod dizajniranja MMO igara. Ideja je da mašina koju koristi igrač poseduje aplikaciju (klijent), uz pomoću koje se mašina povezuje sa serverom, i time se omogućava igranje. Komunikacija je dvosmerna, svaka klijent aplikacija šalje određene podatke serveru, koji onda menja stanje u skladu sa datim podacima, i onda ukupne rezultate šalje nazad klijentima, na osnovu čega igrači vrše određene akcije, generišući pritom podatke koji se opet šalju serveru, i time se postupak ponavlja.

Ovakvo rešenje se smatralo dobrim u periodu dok MMO igre nisu podržavale više od nekoliko stotina igrača. Danas, kada se broj igrača meri desetinama hiljada pa naviše, jedan server nije u stanju, koliko god bio brz i efikasan, da opsluži sve igrače bez problema. Samim time, razvijen je model koji umesto jednog, koristi više servera (server clusters), između kojih se deli opterećenje. Ali, potrebno je imati dobre mehanizme sinhronizacije između servera, radi održavanje konzistentnosti MMO igre.

Prednosti klijent – server arhitekture su centralizovanost, i relativna jednostavnost modela. Sve potrebne informacije se nalaze na serveru, gde ih je lako proveriti i nadgledati, pa i modifikovati, ako za tako nešto ima potrebe. Takođe, relativno lako se uočavaju nepravilnosti u radu, posebno kada je u pitanju varanje sa strane klijenta.

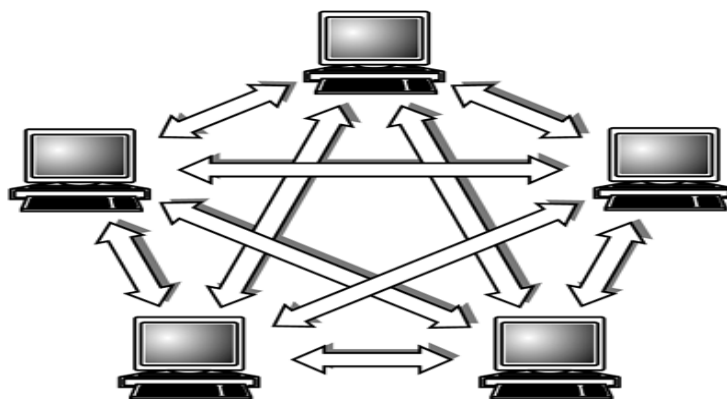
Nažalost, pomenuti model ima i svoje mane. Upravo pošto sve zavisi od servera, u slučaju lošeg rada istoga, velika je verovatnoća da igrači ne budu u mogućnosti da igraju igru kako treba, ili uopšte da igraju. Ako uzmemo u obzir da se u trenutku prekida rada servera automatski prekida i sve što su igrači radili, može se odatle videti koliki je to izvor frustracija, posebno ako su u pitanju neki zadaci (eng. „ quests“), za čije je uspešno završavanje potrebno utrošiti sate i sate.

Drugi problem predstavlja skalabilnost. Pretpostavimo da imamo više servera koji funkcioniraju kao cluster. Takođe, pretpostavimo veoma čest slučaj da imamo

fluktuirajući broj aktivnih igrača, gledano sa strane određenog vremenskog perioda (dan, mesec, godina, itd.). Recimo da u toku dana, u periodu od 8h – 18h imamo oko 2 miliona igrača aktivnih u okviru igre, da bi se taj broj okvirno udvostručio u periodu od 18h – 2h, i zatim smanjio na 80 hiljada u periodu od 2h - 8h. Možemo ići na varijantu da imamo toliko servera u clusteru, da možemo bez problema da obezbedimo uslove za 4 miliona igrača. Međutim, u periodima kada imamo osetno manji broj igrača, pojavljuje se određeni broj servera koji ima malo opterećenje, ili čak uopšte nema opterećenja, ali i dalje su aktivni, i samim time koštaju isto kao da su maksimalno popunjeni, čime dolazi do gubljenja novca. Sa druge strane, smanjivanje broja aktivnih servera može lako dovesti do toga da postane nemoguće opslužiti sve igrače, što opet dovodi do nezadovoljstva istih, i njihovog odustajanja od igranja naslova, čime na kraju opet dolazi do gubljenja para. Naravno, idealno bi bilo da tačno znamo koliko će igrača kada biti aktivno, kao u primeru, ali nažalost, u realnosti to ume biti jedan veoma veliki problem. U suštini, pravilo je da pravila nema, samim time, veoma je teško u okviru klijent-server rešenja naći dobar balans između utrošenih resursa, efikasnog funkcionisanja i ostvarenog profita. Ali, postoje određene ideje koje mogu u tome pomoći, i o njima ćemo govoriti kasnije.

2.3.2 Peer – to – peer (P2P) arhitektura

Peer – to – peer (P2P)² arhitektura predstavlja decentralizovani tip arhitekture. Konkretno, to znači da umesto komunikacije klijent računara sa serverom, radi dobijanja i razmene informacija putem istog, klijent računari međusobno komuniciraju svaki sa svakim, bez potrebe za nekim centralnom tačkom, koja kontroliše komunikaciju. Radi se o tome da u takvoj mreži, klijenti su istovremeno i serveri, i to predstavlja čist P2P model. Pored te opcije, postoji i mogućnost korišćenja centralizovanih servera, čija je svrha da olakšaju međusobno nalaženje klijenata, sa mogućnošću indeksiranja klijenata i informacija / sadržaja koji oni imaju, što dodatno smanjuje opterećenje samih klijent računara.

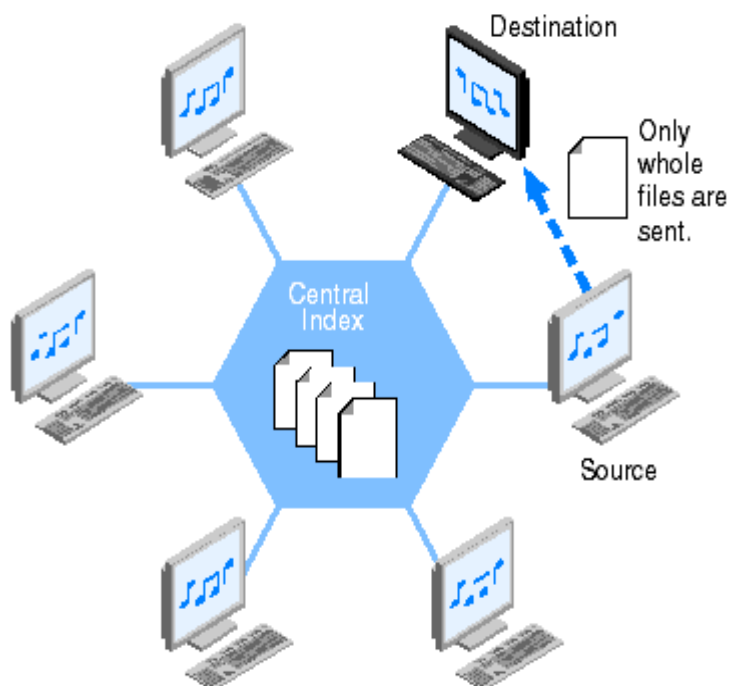


Slika 3: Peer – to – Peer mreža

² <http://en.wikipedia.org/wiki/Peer-to-peer>

THE ORIGINAL NAPSTER

Napster provided a central directory of users who had files to share.



Slika 4: Peer – to –Peer mreža sa centralizovanim serverom – Napster primer

Koje su to prednosti koje P2P arhitektura ima u odnosu na klijent – server arhitekturu? Kao prvo, tu je pitanje pouzdanosti. Već smo napomenuli koliki je problem kod klijent – server arhitekture prestanak rada servera. Kod P2P arhitekture, u slučajevima ako neki klijent prestane da funkcioniše, to ne sprečava ostale igrače da uživaju u igri, pošto se informacije nalaze kod svih klijenata, tako da se oni mogu preusmeriti na druge članove P2P mreže.

Drugi element predstavlja skalabilnost. Opet se ovde vraćamo na probleme kod klijent – server arhitekture: sa svakim novim igračem koji se priključi igri, server biva dodatno opterećen. P2P arhitektura po svojoj prirodi rešava taj problem. Naime, pošto svi klijenti međusobno imaju podjednako podeljeno opterećenje po pitanju potrebnog rada, svaki novi klijent takođe dobija svoj deo posla, samim time eliminišući dodatno opterećenje kod ostalih klijenata. Drugim rečima, svaka nova klijent mašina je po pitanju resursa odgovorna sama za sebe, pod uslovom da je pređen minimum potrebnih klijenata.

Tesno povezana sa istim je i cena održavanja. Kao što smo već kazali, kod P2P arhitekture svaka klijent mašina je sama za sebe odgovorna po pitanju resursa, čime se automatski smanjuju troškovi rada, zatim troškovi održavanja servera, kao i potrebe za

balansiranjem opterećenja istih. Sve to ima veliki značaj, posebno za online naslove koji su novi, i pravljeni od strane developera koji nemaju jaku finansijsku potporu.

Naravno, nijedno rešenje nije savršeno. Mane P2P arhitekture su nedostatak garancije da sve funkcioniše kako treba. Šta to konkretno znači? Recimo da se igri priključio neki igrač čija je klijent mašina relativno slaba, a uz to, njegova internet konekcija je takođe loša. Rezultat toga je mogućnost da svim igračima igra postane neigriva, zbog sporog odziva pomenutog klijenta. Ekvivalentan primer tome bi bio slučaj da podelimo određenim ljudima posao, tako da je svaka osoba podjednako opterećena. U idealnim uslovima, svi bi završili svoj deo na vreme. Ali, u realnosti, često se dešava da neko u okviru grupe nije dorastao zadatku, i da mu je potrebno više vremena da završi dati posao, čime se automatski usporava rad cele grupe. Upravo ovakao nešto se dešava u navedenom slučaju u P2P mreži. Posledice toga su nemogućnost uživanja igrača u igri, čime se opet gube klijenti i novac.

Drugi problem predstavlja činjenica, da je za P2P online igre potreban određeni minimalni broj klijenata radi obrade određene količine informacija, da bi igra mogla da funkcioniše kako treba. Takođe, tu je i pitanje sigurnosti, jer upravo zbog činjenice da su sve informacije dostupne svim klijentima, dosta je teže locirati i sprečiti varanje u okviru igre.

2.4 Predlozi i razvoj novih rešenja

Na osnovu svega do sada, možemo videti da nije lako naći univerzalno rešenje za sve slučajeve. Navešćemo nekoliko primera i tehnologija uz pomoću kojih se pokušava naći najbolji balans između potreba korisnika, efikasnog funkcionisanja sistema i ekonomičnosti.

2.4.1 Virtualizacija i Cloud computing

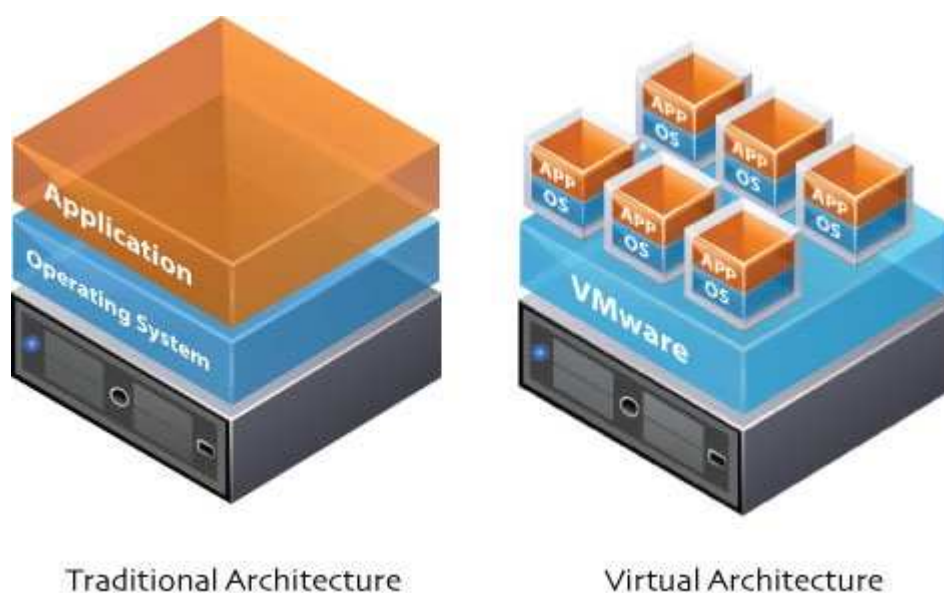
Virtualizacija (eng. virtualisation³) podrazumeva kreiranje virtuelne verzije određenog elementa, poput memorijske jedinice, operativnog sistema, kompletnog računarskog hardvera, itd.

Cloud computing⁴ predstavlja izraz koji ima nekoliko definicija, ali ovde ćemo se fokusirati na onu koja govori o cloud computing-u kao servisima i aplikacijama koje se izvršavaju na virtuelnom hardveru, simuliranom od strane softvera koji se nalazi na pravim mašinama.

³ <http://en.wikipedia.org/wiki/Virtualisation>

⁴ http://en.wikipedia.org/wiki/Cloud_computing

Šta nama ovi pojmovi znače po pitanju menadžmenta resursa za MMOG?



Slika 5: Primer virtualizacije - VMware

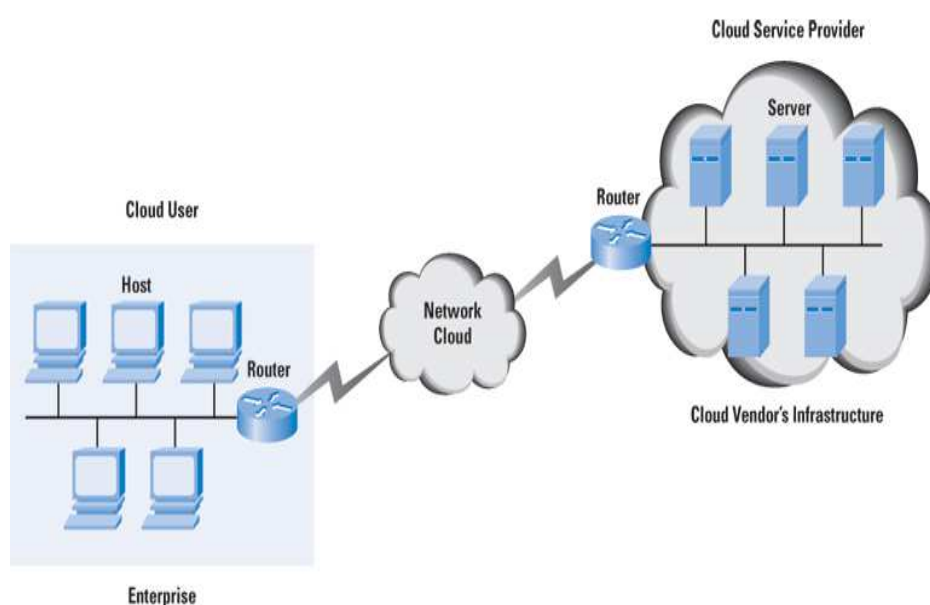
Kao što vidimo na slici 3, virtualizacija nam omogućava da imamo nekoliko virtuelnih mašina na jednoj fizičkoj. Prednosti takvog pristupa se svode na činjenicu da možemo recimo držati paralelno upaljenih nekoliko aplikacija, od kojih svaka radi na drugom operativnom sistemu. Bez virtualizacije, morali bi za svaki OS da imamo posebnu mašinu, što značajno povećava troškove. Cloud computing proširuje ovu ideju: na jednom serveru može se nalaziti nekoliko virtuelnih servera, svaki zadužen za poseban servis.

Ovde se postavlja pitanje zašto ne zakupiti dovoljno fizičkih servera, tako da svaki server služi za jedan servis? Poslužićemo se primerom: recimo da smo kupili dve server mašine, A i B, obe jednake po ceni i performansama. A je zadužen za servis video razgovora, i radi na jednom operativnom sistemu, dok je B zadužen za neku MMO igru, i koristi drugi OS. Opterećenje na serveru A u nekim periodima je sredinom dana ogromno, toliko da čak dolazi do usporavanja video razgovora, pa i do nemogućnosti korišćenja usluge usled preopterećenja servera. U međuvremenu, na server B je povezan relativno mali broj igrača, i dobar deo njegovih resursa stoji neiskorišćen, a upravo ti resursi bi mogli da stabilizuju rad video servisa. Kako onda maksimalno uposliti servere i izbeći preopterećenje jednog od njih?

Naravno, uvek imamo mogućnost da na A i B instaliramo oba operativna sistema, i onda po potrebi prebacimo recimo A na drugi OS, i njega da koristimo takođe kao game server, i time izbegnemo zagušenje na B u periodima kada se preveliki broj igrača konektuje. Rešen problem, zagušenja u igri nema, međutim, šta je sa klijentima koji su želeli da koriste video razgovor? Pošto je veoma mala verovatnoća da postoje periodi u toku dana gde baš niko ne koristi servis, sigurno će biti klijenata kojima će biti uskraćena usluga, što opet dovodi do velikog i opravdanog nezadovoljstva, i gubitka para na kraju.

Upravo ovde možemo primeniti virtualizaciju. Na A napravimo virtuelne servere C1 i G1, dok kod B napravimo C2 i G2. C serveri su zaduženi za servis video razgovora, dok su G serveri zaduženi za MMO, i svi su istovremeno aktivni na A i B. Vraćamo se na prethodni primer, gde je A aktivan za video razgovore, tj. C1 server je primaran, a G1 radi u pozadini i zauzima relativno malo resursa, dok je kod B situacija obrnuta, tj. G2 je primaran, a C2 je u pozadini. Odjednom, toliko klijenata se konektuje na C1 da A više nema resursa da ih opsluži. Upravo u tom momentu, može se aktivirati C2 na B serveru, i preuzeti deo klijenata. Samim time, ljudi koji igraju igru na G2 i dalje mogu da nastave sa time, a pritom se neiskorišćeni resursi na B upotrebljavaju za funkcionisanje C2 virtuelnog servera, uz čiju pomoć servis za video razgovore i dalje funkcioniše kako treba. Kasnije u toku dana, kada se dosta klijenata konektuje u želji da igra igru, G1 je taj koji preuzima višak resursa na serveru A, i sprečava da dodje do preopterećenja na G2, i samim time na serveru B.

Ekvivalentan primer ovome, koji je vezan za MMO igre, mogao bi biti ako uzmemo neku MMORPG igru, koja ima veliki broj igrača aktivnih u isto vreme, i sastoji se iz jednog velikog sveta, koji je podeljen u dva dela. Server A je zadužen za jedan deo, dok je za drugi zadužen server B. Sve deluje idealno, ali problem nastaje zbog činjenice da igrači, kao društvena bića, često idu zajedno u grupi sa prijateljima, i samim time, lako može da se desi, da na jednom delu sveta bude mnogo više likova nego na drugom. Samim time, mnogo je više klijenata povezano na A, dok B „zvrji prazan“, što opet dovodi do problema adekvatnog funkcionisanja igre. Upravo primenom virtualizacije kao u prethodnome primeru možemo omogućiti da svi resursi budu maksimalno iskorišćeni, i time poboljšati rad igre, i sprečiti probleme.



Slika 6: Primer cloud computinga

Danas postoje firme koje nude servis cloud computing usluge (eng. cloud service provider, skraćeno CSP), koje poseduju veliki broj fizičkih servera, na kojima se nalazi veliki broj virtuelnih servera preko kojih funkcionišu razni servisi. Samim time, ovo zvuči kao veoma dobro rešenje za mnoge developere, koji žele da pokrenu svoj MMO, ali nisu u mogućnosti da odvoje novac za nabavku sopstvenih servera.

Gde su mane pomenutih pristupa? Ako uzmemo u obzir virtualizaciju, prvo što može da se desi je mogućnost hardverske greške kod servera na kome se nalaze virtuelni serveri. To dovodi do gubitka performansi, pa i velikog rizika da servisi prestanu da rade kako treba, što ne bi bio toliki problem da je za svaki servis odvojen poseban fizički server.

Dalje, tu je i mogućnost, da upravo jer je virtuelan, server, iako tehnički ima određene resurse na raspolaganju, ipak nije na nivou fizičkog servera koji ima isti hardver i softver. Zatim, postoji više vrsta softvera za virtualizaciju servera, i svi se međusobno osetno razlikuju, tako da je svaki potrebno dobro proučiti. Takođe, postoji mogućnost da neke aplikacije same po sebi ne funkcionišu dobro, ili ne funkcionišu uopšte upravo zbog toga što se izvršavaju u okviru virtuelnog servera.

Kod cloud computinga, kao prvo se nameće pitanje sigurnosti. Podaci koji se nalaze na serverima u okviru cloud-a su van kontrole developera MMO-a, tj. developer ne mogu znati ko ima pristup njihovim podacima, pošto sve to potpada pod kontrolu CSP-a. Takođe, mi nemamo pristup konkretnom hardveru i OS-u na kome se nalaze podaci koje smo poslali na cloud. Sve to dovodi do toga da se veoma lako može desiti da izgubimo podatke na cloud-u, usled raznih rizika poput hardverskih problema na serverima, ili greške u okviru OS-a, itd.

Na ovo se direktno nadovezuje problem prevelike zavisnosti od CSP-a. CSP određuje koje će sigurnosne mere primenjivati na svojim serverima, kako će sve funkcionisati, da li ima ili nema dovoljno servera itd. Uz to, sve mere protiv eventualne havarije i problema sa radom takođe su u rukama cloud providera. Ako se uzme u obzir i činjenica da prelaženje sa jednog na drugog CSP-a nije lako ni jednostavno, može se videti da je tu u pitanju jedan veliki nivo zavisnosti developera od strane provider-a. Toliки nivo zavisnosti od jedne firme je problematično, bez obzira na to koliko je sama firma uspešna i funkcioniše efikasno.

Pored toga, tu su i problemi cene (mogućnost povećanja novčane nadoknade za korišćenje usluga), nemogućnost dodatnog prilagođavanja (uzimanje u obzir specijalnih potreba klijenata), kao i otežana integracija drugih autonomnih sistema (servera i sl) sa cloud serverima.

Sveukupno, koristiti mogućnosti virtualizacije se može pokazati kao veoma dobar potez, posebno sa finansijske strane, ali treba imati u vidu da je to alternativna opcija, koja ima neke od problema kao i korišćenje fizičkih servera. Korišćenje cloud computinga je, sa druge strane, relativno rizična alternativa, zbog navedenih mana, ali može biti dobro rešenje za razvojne timove koji nemaju finansijskih sredstava za sopstvenu infrastrukturu.

2.4.2 Poboljšanja kod P2P arhitekture

Videli smo da P2P arhitektura ima mogućnost primene kod online igara. Međutim, postoje tu i određene mane, koje nisu zanemarljive. Ovde ćemo govoriti o nekim idejama i principima koje možemo primeniti, radi smanjivanja uticaja pomenutih nedostataka arhitekture.

Kod jedne MMO igre, koja se sastoji iz jednog velikog sveta kojim se kreću likovi kontrolisani od strane igrača, veoma bitno je globalno stanje, koje zapravo predstavlja rezultat svih događaja koji su se desili u okviru igre. Na primer, događaj pobeđe grupe igrača nad jakim protivnikom kontrolisanim od strane kompjutera, i samim time njegova „smrt“, čini da se protiv tog protivnika niko od igrača ne može boriti dok on ponovo ne „oživi“ (eng. respawn), i to predstavlja deo globalnog stanja igre. Ono što je bitno je da to stanje mora biti ekvivalentno kod svih igrača.

Pored globalnog stanja, postoji nešto što se zove lokalno stanje. Lokalno stanje predstavljaju informacije vezane za: stanje predmeta koji igrač poseduje (npr. snaga mača koga u ruci drži lik kojim igrač upravlja), zatim informacije vezane za komunikaciju između više igrača, kao i dešavanja u igri koja su vezana za neki deo sveta (npr. padanje kiše u jednom delu sveta). Lokalna stanja nisu bitna za igrače koji nemaju veze sa tim konkretnim stanjem (npr. ako se u pomenutoj regiji sa kišom igrač trenutno ne nalazi, sama informacija nema značaja za njega), i samim time, nije toliko bitno da ta stanja budu međusobno ekvivalentna kod svih igrača.

Kod P2P arhitekture, uzevši u obzir navedena stanja, može se govoriti o podeli opterećenja među klijentima, i to na struktuiranu i nestruktuiranu podelu. Šta to konkretno znači? Već smo napomenuli, da bi MMO igra funkcionisala putem P2P arhitekture, sav posao obrade podataka mora da se podeli između klijenata, i rezultati obrade se međusobno upoređuju.

Ali po kome kriterijumu izvršiti podelu? Kod struktuirane podele, svim povezanim klijentima se dodeljuju brojevi, i svakom broju se dodeljuju određena zaduženja, tj podaci koje treba da obrade. Sami podaci nemaju pravila po kome se dodeljuju, tako da npr., podaci vezani za jednog igrača mogu biti obrađivani od strane klijent mašine, čiji se igrač nalazi na drugoj strani virtuelnog sveta. Nestruktuirana podela, sa druge strane, dodeljuje podatke u skladu sa time gde se igrači nalaze u okviru sveta, tako da ako se neki igrač nalazi na lokaciji X, podaci koji su mu dodeljeni biće vezani za njegovog lika, kao i za likove i stvari koje se nalaze u okviru lokacije X. Oba pristupa imaju prednosti i mana (struktuirana podela omogućava bolje usklađivanje globalnog stanja, dok nestruktuirana podela omogućava bržu obradu podataka vezanu za lokalno stanje), tako da se oba pristupa koriste u kombinaciji.

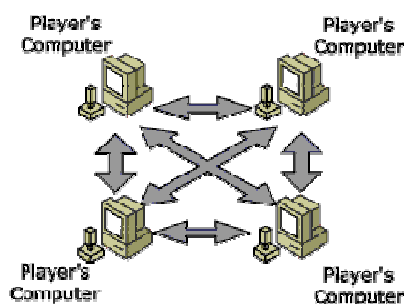
Takođe, ono što predstavlja veliki problem je činjenica da je teško održavati konzistentnost informacija i stanja zbog čestog priljučivanja i isključivanja klijenata. Način kojim se to može rešiti je multipliciranjem vrednosti stanja kod nekoliko klijenata, tako da

ako jedan klijent napusti iznenada igru, određene bitne informacije i dalje ostaju u istoj. U tome slučaju, potrebno je naći balans između performansi i dostupnih resursa, i sprečavanja gubitka bitnih informacija.

Vezano za problem usporavanja igre usled određenih klijenata, moguće je primeniti algoritme predviđanja, kao u GGPO slučaju. Dakle, čim nemamo pravovremeni prijem podataka od nekog klijenta, aktiviraju se algoritmi koji na osnovu prethodnih poteza igrača predviđaju najverovatniji sled akcija, i u slučaju greške, vraćaju stanje unazad. Efikasnost pomenutog rešenja varira od slučaja do slučaja, kao i od tipa igre. Drugi pristup bi bilo ravnomerno usporavanje igre za sve klijente, ali opet, već smo prikazali zašto to ume biti loše, i u mnogim slučajevima neprihvatljivo rešenje. Tu je i opcija periodičnog sinhronizovanja klijenata, gde se usklađivanje stanja ne vrši konstantno, već u određenim vremenskim razmacima, čime se izbegava usporavanje igre za sve igrače, ali je veća mogućnost da dođe do greške.

Sprečavanje variranja u okviru igre predstavlja veliki izazov kod P2P orijentisanih sistema. Neko ko vara može recimo ići na varijantu da modifikuje informacije koje pristižu ili odlaze sa njegove mašine. Konkretno, moguće je da poruke u okviru igre od jednog igrača drugome budu modifikovane, zato što je ta poruka morala da prođe kroz nekoliko drugih klijenata, gde je dovoljno da je samo jedan igrač voljan i sposoban da vrši modifikaciju informacija da dođe do problema. Zatim, postoji mogućnost pregleda i modifikacije informacija koje se odnose na stanja u igri, gde na primer, igrač može imati u vidu gde se koji igrač ili kompjuterski vođeni lik nalazi, kao i uticanje na njihovu poziciju, itd. Tu je naravno i mogućnost uticanja na stabilnost igre direktno. Sve ovo su problemi koje je lakše rešiti u klijent – server arhitekturi, upravo zato što se obrada informacija radi pretežno na serveru, koji je pod kontrolom developera.

Predlozi za rešenje su razni, spomenućemo samo neke: šifrovanje informacija (pomaže u sprečavanju modifikacija unutar fajla, ali ne sprečava zamenu celog fajla), korišćenje nekoliko nasumice izabranih klijenata kao kontrolne tačke (svaka kontrolna tačka vrši dodatni pregled ispravnosti stanja i informacija kod ostalih klijenata, i nakon određenog perioda, kontrolna tačka se menja), korišćenje struktuirane podele informacija (tako da klijent mašina bude zadužena za podatke regije gde se lik trenutno ne nalazi) itd. Nijedno rešenje nije 100% sigurno, tako da primena zavisi od konkretnog slučaja.



Slika 7: Primer osnovne P2P arhitekture kod online igara

2.4.3 Praćenje i predviđanje opterećenosti servera

Nakon navedenih primera, ovde bi smo mogli da spomenemo određene ideje i pristupe radi povećanja efikasnosti menadžmenta resursa. Konkretno, radi se o idejama praćenja opterećenja servera i predviđanja opterećenosti. Sve ove metode imaju za cilj obaveštavanje o mogućim dešavanjima, radi pravovremenog reagovanja na iste.

Praćenje opterećenja servera ima smisla ako dodatno uzmemo u obzir i promenljivost broja aktivnih igrača. Već smo spomenuli kako se broj igrača menja u određenim vremenskim periodima, i koliko to ume biti problem kod određivanja potrebnog broja aktivnih resursa. Postoji dodatni rizik kod sprovođenja promena u okviru igre, zbog kojih su određeni MMO naslovi (npr. *Runescape* kompanije Jages Games Studio) gubili i po 30% korisnika usled značajnijih promena. Pored svih drugih negativnih posledica, to takođe dovodi do još manje iskorišćenosti servera, u slučaju kada ih imamo fiksni broj.

Treba spomenuti još jedan elemenat koji utiče na opterećenost servera: interakcija između igrača. Potreba za resursima će rasti u odnosu na to da li je igra bržeg tempa ili sporijeg tempa, zato što kod bržih igara igrači dosta lakše primete prisustvo laga, i samim time da bi se to izbeglo, potrebno je brzo razmeniti velike količine informacija, bez obzira na sam broj igrača. Sa druge strane, ako je igra sporijeg tempa, zahtevnost po pitanju broja i brzine prenosa informacija se direktno povećava sa brojem igrača, posebno ako su pomenuti igrači udruženi u grupu, jer se interakcija između tih igrača u tom slučaju značajno povećava.

Upravo zbog toga potrebno je koristiti dinamičku dodelu resursa, ali onda se postavlja pitanje koliko resursa odvojiti da sve funkcioniše kako treba, a da pritom ne dođemo u situaciju da imamo neiskorišćeni višak. Metode analize predstavljaju odgovor na to. Kod MMO igara, najefikasniji pristup bi bio korišćenje podataka iz prethodnog perioda, radi predviđanja budućeg stanja. Problem je što kod metoda predviđanja jeftina rešenja nisu efikasna u slučajevima kompleksnih igara koje pritom imaju veliki broj igrača, dok rešenja koja daju zadovoljavajuće rezultate imaju relativno visoku cenu.

Kao alternativu, možemo koristiti neuronske mreže za predviđanje opterećenosti. Međutim, kada su u pitanju MMOG naslovi, teško je utvrditi sve interakcije između igrača u okviru igre. Način na koji možemo to rešiti je da podelimo virtuelni svet u veliki broj malih zona, gde u okviru svih tih zona imamo toliko entiteta, da možemo na osnovu njihovog broja da predvidimo broj interakcija u okviru zone, i samim time predvidimo sveukupno opterećenje zone, dok sabiranjem svih zona dobijamo sveukupno opterećenje. Sistem funkcioniše na principu perioda, gde na osnovu broja entiteta, samim time opterećenosti u okviru određenih perioda u prošlosti kao ulaznog podatka, dobijamo na izlazu predviđenu opterećenost u narednom periodu iste dužine.

No, da bi neuronska mreža mogla da funkcioniše kako treba, potrebno je da se prođe kroz dve faze. Prva je faza prikupljanja podataka, koja vremenski može dugo da potraje, gde se analizira broj entiteta i opterećenost zona u određenim periodima. Druga

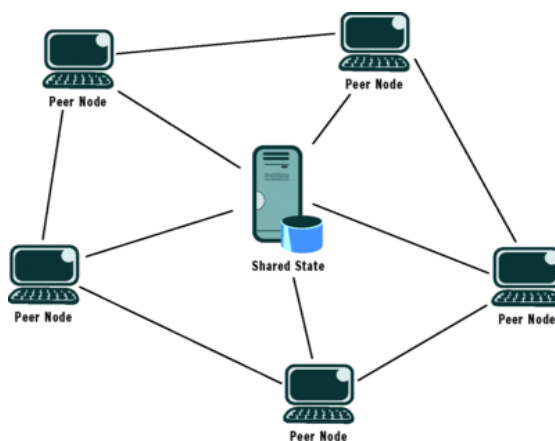
faza je faza učenja, u okviru koje se prvo vrši podela parova entitet (opterećenje) - period na trening parove, koji čine većinu, i test parove. Trening parovi se onda u određenom redosledu analiziraju u okviru neuronske mreže, da bi se zatim test parovi upotrebili radi provere efikasnosti predviđanja. Ovakav pristup zahteva više vremena od većine drugih pristupa vezanih za predviđanje, i takođe ume biti malo sporiji, ali je tačnost predviđanja dosta veća kod neuronskih mreža u odnosu na ostale pristupe, tako da treba imati u vidu i tu mogućnost.

2.4.4 Hibridna arhitektura

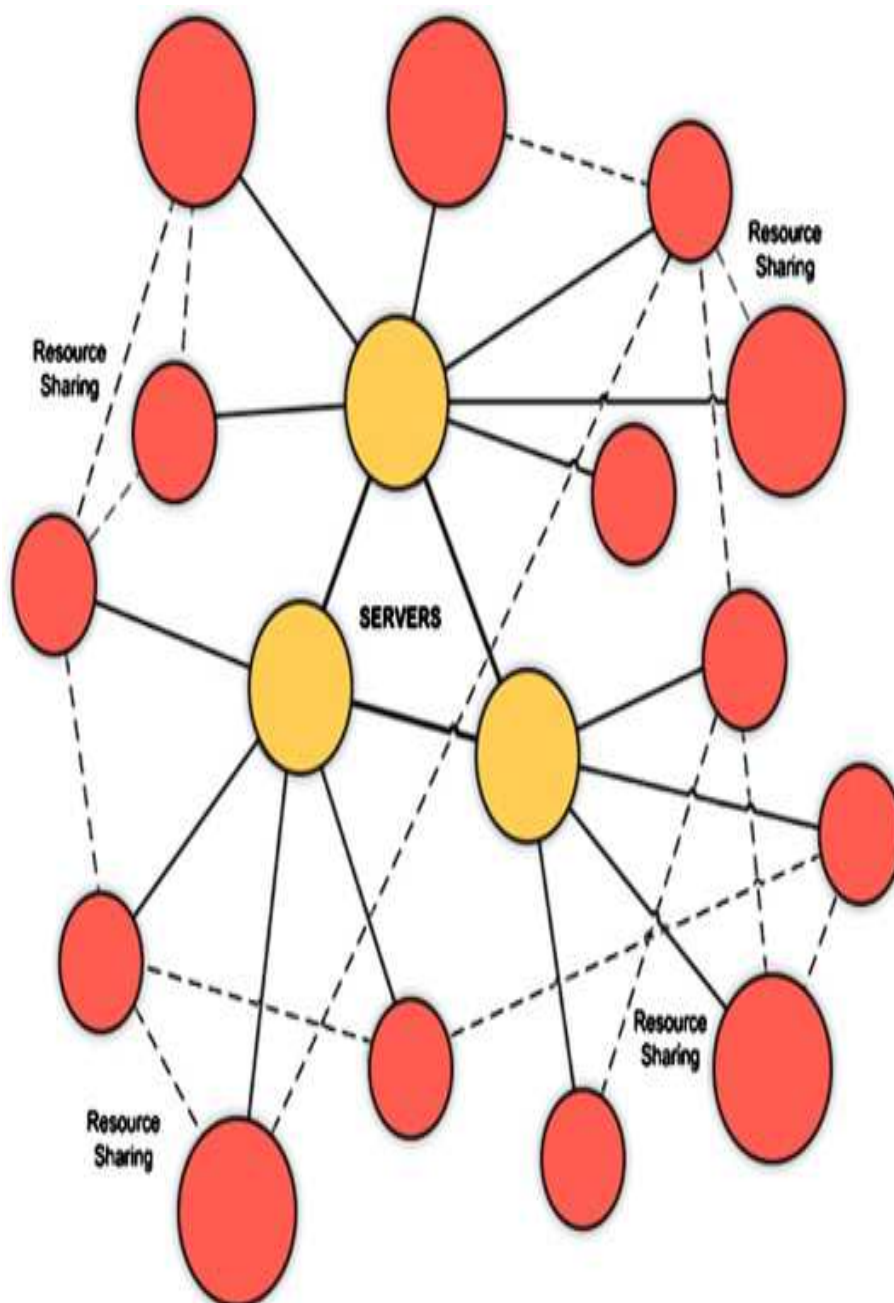
Korišćenje hibridne arhitekture predstavlja noviju ideju kada su u pitanju MMO igre. Princip se zasniva na kombinovanju određenih ideja prisutnih kod klijent – server i P2P arhitekture, da bi se dobio sistem koji je sigurniji i lakše kontrolisan, a opet efikasniji po pitanju utroška resursa.

Kako zapravo ovakav sistem funkcioniše? Ideja je da se klijenti koji se priključuju igri prvo povežu sa master serverom. Uloga master servera je da odradi autentifikaciju igrača, i da se brine o sigurnosnim aspektima igre. Nakon toga, sve ostalo se zasniva na P2P modelu. Kada igrač počne sa igrom, on se nalazi u određenom delu virtuelnog sveta. U okviru tog dela, nalazi se još igrača, i oni se često udružuju u grupe, bilo radi takmičarskog, ili kooperativnog cilja. Ovde je pristup taj, da je za obradu podataka vezanih za stanje grupe i elemenata u njihovoj relativnoj blizini zadužena sama grupa, tj. njihove klijent mašine.

Samim time, ovde se smanjuje opterećenje na strani servera, jer, umesto da server konstantno vrši obradu informacija, to se prepušta samoj grupi, dok se u okviru iste bira jedan klijent koji predstavlja kontrolnu tačku. Kontrolna tačka rezultate obrade grupe periodično prosleđuje serveru, koji onda dalje obrađuje date podatke, i šalje rezultate nazad. Prosto rečeno, ovakav pristup omogućava da jedna grupa igrača bude viđena kao jedan entitet od strane servera, čime se olakšava kontrola, i smanjuje potreba za resursima.



Slika 8: Hibridna arhitektura – primer 1



Slika 9: Hibridna arhitektura – primer 2

Međutim, kod ovog pristupa takođe imamo problema. U suštini, radi se o manama koje se nasleđuju od P2P arhitekture: sigurnosni problemi, gde ipak i pored upošljenog servera po pitanju sigurnosti, postoji mogućnost varanja od strane klijenata, kao i problem pravilnog funkcionisanja, usled nedostatka resursa na klijent mašinama. Zatim, tu je i zavisnost od servera, što je mana nasleđena od klijent – server arhitekture. I na kraju, tu je problem kod same činjenice, da je ovakva arhitektura dosta složenija od prethodnika, samim time je potrebno više vremena da se kompletan sistem aktivira.

3. Cheats – varanje

3.1 Pojam i istorijat

Među najspecifičnijim sigurnosnim problemima kod online video igara nalaze se oni koji potpadaju pod kategoriju varanja (eng. cheats). Definicija varanja podrazumeva upotrebu određenih metoda, u suprotnosti sa utvrđenim pravilima, radi sticanja prednosti u odnosu na ostale igrače, koju sam igrač inače ne bi mogao i / ili trebao da ima⁵.

Pojam varanja u video igrama je star koliko i same video igre. U početku su bila najčešća dva tipa varanja: iskorišćavanje programskih grešaka i korišćenje šifara za varanje (cheat codes). Koliko god sam kôd bio jednostavan, i koliko god programer bio iskusan, često se dešava da dođe do grešaka u kodu i time se omogući igračima da te greške nađu i iskoriste radi lakšeg ostvarivanja cilja u igri. Ovakav način varanja je prisutan i dan danas, i predstavlja jedan od velikih i veoma rasprostranjenih sigurnosnih problema.

Vratimo se sada opet kroz vreme, i uzmimo kao primer period napretka video igara po pitanju popularnosti kod šire populacije. Tada su do izražaja došli neki problemi vezani za sam dizajn. Naime, primećeno je da su igre iz tog perioda veoma kratke i brzo se završe, npr. od strane prosečno sposobnog igrača za pola sata i manje. To se pokazalo kao veliki problem kojim su igrači bili nezadovoljni, a samim time, to je bio i problem za ljude koji su razvijali video igre (developers). Da bi pridobili igrače, developer-i su počeli praviti igre u koje je bilo potrebno uložiti dosta više vremena da bi se došlo do njihovog kraja. No, time je nastao jedan drugi izazov: kako istestirati datu igru? Ako je njena dužina npr. deset sati igranja, i pored toga imate u okviru iste više opcija kako ćete igrati, da bi se stiglo do njenoga kraja i isprobale sve opcije i njihove kombinacije, potrebno je prelaziti igru nekoliko puta, što uopšte nije mali vremenski period.

Upravo zbog toga su nastale šifre za varanje. Korišćenjem pomenutih šifara, developeri su mogli sebi da daju u okviru igre određene attribute, sa kojima bi prelaženje datih igara bilo mnogo lakše i brže, pritom olakšavajući i ubrzavajući i samo testiranje. Naravno, sve bi bilo dobro, da samo developeri imaju pristupa tim šiframa. U praksi, do istih šifri često dolaze i sami igrači. Samo po sebi, to ne bi bio problem kod single – player igara, ali to postaje itekako nezgodno ako igra poseduje i multi – player opcije, jer u velikoj većini slučajeva date šifre će i tada raditi. Sa daljim razvojem igara, razvijale su se i nove metode varanja.

⁵Yan J, Rendell B. A systematic classification of cheating inonline games.NetGames 05, October 10–11, 2005, Hawthorne,New York, USA, ACM, pp.1–9

3.2 Uzroci varanja

Kada govorimo o varanju kod online igara, uzimamo u obzir da se radi o primenama metoda koje utičuili se vrše na samoj klijent mašini, bez direktnih napada na druge klijente, server, protokol itd. Problem varanja u online igrama je veliki zbog toga što predstavlja jedan od glavnih razloga nezadovoljstva korisnika. Niko od nas ne voli da bude izigran, posebno u slučaju kada se sami držimo pravila. Naravno, postavlja se onda i pitanje zašto ljudi uopšte varaju u online igrama, ako to mnogima kvari užitak? Psihološki faktori tu igraju presudnu ulogu.

Online igre predstavljaju i jedan vid socijalne interakcije između ljudi, gde je prisutan širok spektar različitih tipova ponašanja. Imate ljude koji su prijatni i ponašaju se fer, ljude koji nisu prijatni, ljudi koji vređaju druge itd, dakle raznovrsnost kao u stvarnom svetu. I u najuređenijim i najstabilnijim državama sveta, neprijatne situacije su veoma moguće, upravo zbog kompleksnosti ljudske psihe i međuljudskih odnosa. Potpuno ista situacija je i u virtuelnim svetovima online igara. Koliko god se developeri trudili da sankcionišu loše ponašanje, skoro uvek će se desiti da se neki igrač ponaša prema drugima kako ne treba. Ako je uz to sama igra takmičarskog tipa, gde se igrači direktno sučeljavaju jedni protiv drugih, velika je verovatnoća da će tu biti prisutno i dosta negativne energije između igrača, kao što je to slučaj kod sportskih ili bilo kojih drugih takmičenja.

Već ovde vidimo odakle neko može dobiti motivaciju da primeni neki oblik varanja. U suštini, može se tu grubo govoriti o nekoliko kategorija motiva. Kao prvo, postoje ljudi kojima je zabavno da igraju protiv pravila, da kvare drugima igru, i mnogi čak uživaju u reakcijama drugih dok dolaze do cilja lakše, brže i efikasnije, uz manje uloženog truda. Zatim, imamo tu i ljude koji sebe vide kao jednu vrstu žrtve nefer ponašanja, bilo zato što smatraju da su oštećeni u igri jer nisu pobedili, bilo zbog toga što su trpeli loše ponašanje drugih. Samim tim, u njima se budi mali Kalimero, i u cilju “ispravljanja nepravde”, oni pribegavaju upotrebi varanja, kako bi “stvari doveli na svoje mesto”. Tu su i ljudi kojima je cilj da budu pobednici po svaku cenu, da budu “neko i nešto” u igri, da im svi ostali igrači zavide. Postoji još puno drugih primera, ali navedeni predstavljaju najčešće tipove ljudi koji koriste cheat-ove, radi postizanja određene moralne/emotivne/psihološke satisfakcije.

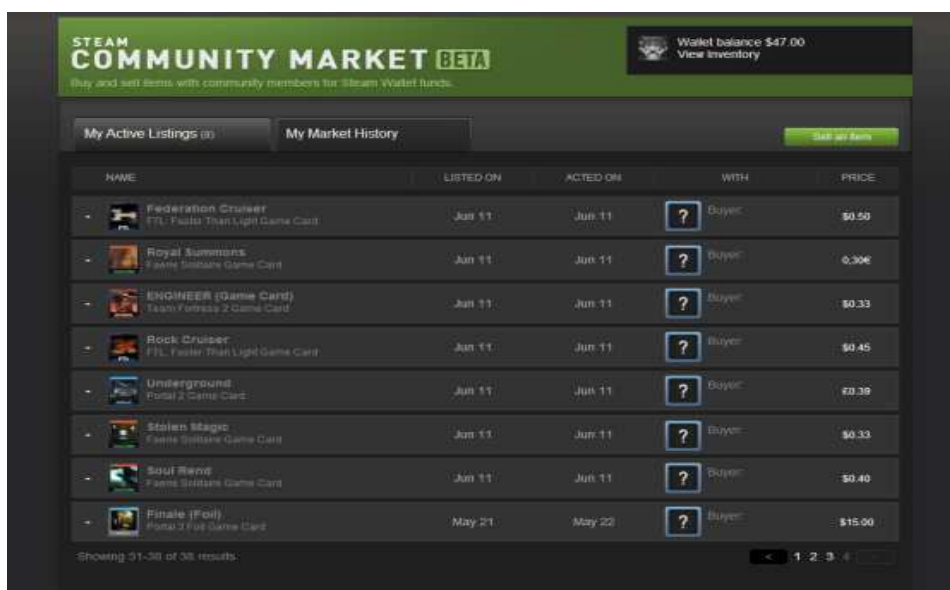
Međutim, stvari se dodatno komplikuju kada uzmemo u obzir Real Money Trading (skraćeno RMT). RMT podrazumeva zamenu virtuelnih resursa stečenih u okviru igre za pravi novac⁶. Šta je to što navodi ljude da daju novac za nešto što nema neku upotrebnu vrednost u realnom svetu? Najčešće su u pitanju ušteda vremena i već spomenuta želja za dokazivanjem i pobedom, pored drugih mogućih faktora. Naveli smo već da je jedan od ciljeva developera taj da igra što duže traje, kako bi igrači bili što zadovoljniji i proveli što duže vremena uz naslov. Kod online igara, to je postao još bitniji

⁶<http://tobolds.blogspot.com/2005/05/real-money-trade-rmt.html>

kriterijum upravo zbog pojma pretplate, jer što više vremena igrači igraju, to sve više para moraju dati da bi stigli do kraja.

Uzmimo sada primer da igrate online igru, gde ste u ulozi ratnika koji borbom stiče iskustvo i postaje sve jači i jači. Nakon uloženog vremena, koje se može meriti mesecima, došli ste do trenutka kada vas pomenuti naslov više ne zanima. Videli ste sve što ima u ponudi, prošli ste i preživeli razne avanture, i smatrate da je vreme da “okačite kopačke o klin”. Dok tako razmišljate, vama u okviru igre prilazi drugi igrač, početnik, koji se divi tome koliko ste uspeli da svog lika ojačate. Početnik se nažalost ne snalazi najbolje, nema puno vremena za igranje, veruje da bi mu igra bila zabavnija, ali ako bi imao bolju poziciju, i ako ne bi morao da utroši puno vremena da do te pozicije dođe. U takvim situacijama može doći do “nepristojnih ponuda”, gde vam početnik nudi određenu finansijsku nadoknadu, ako bi ste njemu ustupili vašeg lika. Naravno, pošto ste vi ionako već rešili da prekinete sa igranjem, vama ova ponuda deluje savršeno, I vi na nju pristanete. Sve prođe dobro, i svi zadovoljni. Ovakve situacije su kod online igara česte, toliko da čak u državama poput Kine imate igraonice gde zaposleni za određenu nadoknadu igraju online igre, radi sticanja retkih resursa koje posle prodaju drugima, i time poslodavcu donose solidan profit. Naravno, čim su pare prisutne, prisutne su i prevare radi sticanja istih, i to takođe predstavlja veoma jaku motivaciju onima koji varaju.

Ovde su navedeni samo neki primeri i ideje oko toga zašto postoje ljudi koji varaju u igrama. Razlog navođenja svega ovoga je potreba da se ukaže na motivisanost i upornost takvih ljudi, koje nikako nisu za potcenjivanje. Upravo zbog toga, potrebno je prilikom primena metoda zaštita imati u vidu da je protivnik neko ko ima visok nivo energije i volje da iskoristi sve naše greške, koliko god one male bile, radi ispunjavanja svoga cilja, i da je brojnost takvih protivnika možda i veća od brojnosti ljudi koji vrše sigurnosne napade na druge tipove aplikacija i informacionih sistema.



The screenshot shows the Steam Community Market interface. At the top, it says "STEAM COMMUNITY MARKET BETA" and "Buy and sell items with community members for Steam Wallet funds." On the right, it shows a "Wallet balance \$47.00" and a "View inventory" link. Below this, there are tabs for "My Active Listings (8)" and "My Market History". A green button "Add an Item" is also visible. The main part of the screen is a table of active listings. The table has columns: NAME, LISTED ON, ACTED ON, WITH, and PRICE. The listings are for various game cards, including Federation Cruiser, Royal Summons, ENGINEER (Game Card), Rock Cruiser, Underground, Stolen Magic, Soul Wield, and Finale (Foil). Each listing shows the item name, the date it was listed (Jun 11), the date it was acted on (Jun 11), the user who acted on it (Buyer), and the price in USD. The prices range from \$0.306 to \$15.00. At the bottom, it says "Showing 1-8 of 35 results." and there are pagination controls.

NAME	LISTED ON	ACTED ON	WITH	PRICE
Federation Cruiser FTL: Faster Than Light Game Card	Jun 11	Jun 11	Buyer	\$0.50
Royal Summons Fable: The Journey Game Card	Jun 11	Jun 11	Buyer	\$0.306
ENGINEER (Game Card) Team Fortress 2 Game Card	Jun 11	Jun 11	Buyer	\$0.33
Rock Cruiser FTL: Faster Than Light Game Card	Jun 11	Jun 11	Buyer	\$0.45
Underground Portal 2 Game Card	Jun 11	Jun 11	Buyer	\$0.39
Stolen Magic Fable: The Journey Game Card	Jun 11	Jun 11	Buyer	\$0.33
Soul Wield Fable: The Journey Game Card	Jun 11	Jun 11	Buyer	\$0.40
Finale (Foil) Portal 2 Game Card	May 21	May 22	Buyer	\$15.00

Slika 10: Steam marketplace – primer regulisanog RMT-a

3.3 Tipovi varanja

3.3.1 Varanje putem neosnovanog poverenja

Kod većine online igara, da bi igrač mogao da ih igra, potrebno je da nakon registracije skine klijent aplikaciju. Klijent aplikacija se onda povezuje sa serverom na kome je igra aktivna, i zabava može da počne. Međutim, tu se javlja jedan veliki problem, a to je da se prevelika količina kritičnih podataka nalazi na strani klijenta. Kao primer bi mogli uzeti podatke vezane za položaj igrača u igri, ili količinu životne energije koju lik u okviru igre ima.

U međuvremenu, nisu prisutni načini sprečavanja zloupotrebe klijent aplikacije, izuzev što postoji nada od strane developera da igrači neće pokušavati ništa protiv pravila. Drugim rečima, previše se poverenja ukazuje drugoj strani. Samim time, veoma su česti slučajevi gde su igrači modifikovanjem klijent aplikacije uspevali da dobiju veliku prednost u odnosu na druge igrače. Mogućnosti su razne: od “kozmetičkih” promena (npr. promena boje likova i okoline), do kritičnih (npr. promena količine resursa koje igrač ima u datom trenutku u igri).

Prvo što bi se neko zapitao u datoj situaciji je: “Zašto uopšte dati toliko podataka na uvid klijent strani? Bolje je sve podatke prebaciti na server.” Logički gledano, takvo rešenje ima smisla. Kada se svi podaci nalaze na serveru, klijent aplikacija služi isključivo za povezivanje, a oni koji žele da bilo šta modifikuju, moraće da napadaju sam server, što je mnogo veći i rizičniji poduhvat. Nažalost, tu se javlja jedan veoma veliki problem: utrošak resursa. Ako uzmemo za primer popularne online igre, čak i u slučajevima kada imamo dosta podataka na klijent strani, serveri umeju biti veoma opterećeni, toliko da može doći i do nemogućnosti igranja igre. Možemo onda zamisliti koliko bi opterećenje bilo ako bi se svi podaci obrađivalina serveru, i koliki bi bili troškovi održavanja takvog sistema.

Drugi pristup predstavlja upotrebu arhitekture javnih servera (public server architecture). Takva arhitektura se zasniva na principu da svako može da dobije fajlove kojima može aktivirati server za igru. U tom slučaju, developer se jedino brine za klijent aplikaciju, kao i nalaženje servera, dok druge osobe pokreću servere i održavaju ih. Zvuči kao sjajno rešenje: umesto nekoliko centralizovanih servera, ovde imamo na stotine manjih, samim time opterećenje po serveru je značajno manje, a time se opet može otvoriti i mogućnost smanjenja kritičnih podataka na strani klijenta. Međutim, i tu su prisutni problemi.

Kao prvo, postoji mogućnost da se dođe u situaciju da niko ne želi da drži servere za igru, što je veliki problem. Tu je i mogućnost ljudi da u svakom trenutku ugase svoj server, čime se prekida igra ljudi konektovanih na isti, što opet dovodi do njihovog

nezadovoljstva. Ali čak i kad sve funkcioniše naizgled kako treba, koja garancija postoji da sam server nije bezbednosno ugrožen, i da se primenjuju adekvatne metode zaštite?

I pored svega toga, ovakav pristup se generalno pokazao dobrim za igre koje se ne sastoje iz jednog velikog sveta, već se igraju po principu partija koje traju određeni period.

Treći pristup čini upotreba peer-to-peer arhitekture. U toj varijanti, svi klijent računari su istovremeno i serveri, i svi proračunavaju sve, i zatim međusobno porede rezultate, na osnovu kojih se određuje stanje svih učesnika. U takvim uslovima, modifikacija klijenta se može dosta lako otkriti, zbog činjenice da će se njegovi rezultati razlikovati od ostalih. Nažalost i ovde postoji problem vezan za resurse. Naime, dovoljno je da samo jedan klijent ima problem sa obradom podataka, da dodje do zastoja u proračunima, čime se sama igra usporava za sve učesnike. To je veliki problem u slučajevima ako igra zahteva brzu reakciju od strane igrača. Jedini način da se takav problem reši je da se sama igra uprosti do te mere, da čak i stari uređaji mogu da obrade sve potrebne informacije. Naravno, to neizbežno dovodi i do problema da takva igra može brzo da dosadi, i opet time dolazi do gubljenja klijenata.

Možemo takođe govoriti i o enkripciji kao mogućem rešenju. Prosto, zaštitimo šifrovanjem sve fajlove na klijent aplikaciji, i time onemogućimo modifikovanje podataka u fajlovima. Opet, to ne sprečava igrača da zameni neki određene fajlove u okviru aplikacije drugim fajlovima, kojima npr. "otključava" pristup lokacijama kojima inače ne bi trebao da ima pristupa. Rešenje u tom slučaju predstavlja proveru podataka klijenta od strane servera. Klijent šalje checksum učitaneog fajla, koju server poredi sa očekivanom vrednošću. U slučaju da ne dođe do poklapanja, server automatski registruje pokušaj varanja.

Kao mogući kompromis, mogla bi se primeniti podela podataka po kritičnosti, tako da bi pre pomenuti podaci vezani za boju bili na klijentu, dok oni koji su bitniji, kao položaj igrača, budu obrađivani na serveru. Time se smanjuje opterećenje servera, što opet otvara i mogućnost iskorišćavanja dodatnih resursa na dodatne sisteme zaštite.

3.3.2 Varanje u dosluhu sa drugima

Veoma čest oblik varanja predstavlja dogovor između igrača, koji umesto da se međusobno takmiče, po dogovoru puštaju jedan drugog da pobeđuje. Uzmimo na primer online igru koja simulira poker, gde se za svaku pobeđuju bodovi. Recimo da od četiri igrača, tri se međusobno dogovaraju. Neko ko je tu u ulozi četvrtog igrača teško da će dobro proći u takvoj situaciji. I tako se tri drugara međusobno dogovaraju, prepuštaju pobeđuje jedno drugome, i svako od njih napreduje na rang listi.

Način na koji se ovako nešto može sprečiti je putem dodele negativnih bodova za svaku izgubljenu partiju. Jedan od sistema rangiranja, napravljen i primenjen za potrebe šaha, koji funkcioniše po tom principu je i ELO sistem rangiranja⁷. ELO sistem funkcioniše tako što svaki igrač ima određeni broj poena, u zavisnosti od rezultata partija odigranih protiv drugih igrača. Nakon partije procenjuje se razlika u bodovima između dva takmičara, i na osnovu te razlike dobija se broj, koji se dodaje pobedniku, a oduzima gubitniku. Ovde je pretpostavka da u igri koju igramo ne može doći do nerešenog rezultata. Recimo da igrač A ima 1000 bodova, a igrač B 500 bodova. Ukoliko pobedi A, broj bodova koji se dodaje njemu je npr. 25, dok se od B isto toliko oduzima. Međutim, ukoliko pobedi B, broj bodova koji se njemu dodaje je npr. 250, i isto toliko se oduzima od A. Ovako koncipiran sistem onemogućava primenu podele pobeđa (win-trade).

Iako u teoriji dobar sistem, u praksi se pokazalo da nije sve savršeno. Prvi problem je vezan za činjenicu da su ELO bodovi često vidljivi, tako da igrači u cilju čuvanja svog rezultata mogu namerno da izbegavaju partije sa ljudima koji imaju veliki broj bodova, i umesto toga se fokusiraju na igrače koji imaju malo bodova, omogućavajući sebi spor, ali konstantan napredak. Uz to, kao nadogradnja te ideje, postoji mogućnost da ista osoba napravi nekoliko alternativnih naloga, protiv kojih uvek može ostvariti pobeđu sa svojim “glavnim” nalogom.

Kao moguće rešenje može se ići na ideju da protivnike ne može izabrati igrač, već da sistem to čini za njega. Mada je takav pristup pošteniji, pod uslovom da je odrađeno sve kako treba, isto tako ume biti i uzrok nezadovoljstva igrača. Naime, većina ljudi koja ne primenjuje varanja takođe ima želju da igra protiv određenih igrača. Neki pod tim igračima podrazumevaju prijatelje, dok drugi podrazumevaju ljude koji se nalaze na vrhu lestvice, jer žele izazov i mogućnost da uče o igri igrajući protiv najboljih. Ukidanjem opcije izbora protivnika, mnogim igračima se smanjuje i zabavnost same igre, što dovodi do njihovog odustajanja od iste.

3.3.3 Varanje putem iskorišćavanja procedura u igri

Neke tipove varanja je moguće sprovesti čak i bez velikog tehničkog znanja. Među takva varanja spada i varanje putem prekidanja veze (eng. disconnect), često prisutan kod online igara kod kojih je takmičarski duh veoma izražen.

Kao primer bi nam mogla poslužiti igra *The King of Fighters XIII* kompanije SNK – Playmore, koja takođe poseduje i opciju za online mečeve. Pravila su sledeća: igrač od 36 boraca bira 3, koji čine njegov tim. Nakon izbora, prelazi se na izbor redosleda kojim će se borci boriti protiv tima drugog igrača. Same borbe idu po principu da pobednik runde prelazi u sledeću i bori se protiv sledećeg člana protivničkog tima. Pobednik je onaj igrač,

⁷http://en.wikipedia.org/wiki/Elo_rating_system

čiji treći član tima ostane neporažen. Igra koristi princip ELO bodova (u igri nazvani battle points, skraćeno BP) kod online igranja, radi rangiranja svih igrača na globalnoj listi.



Slika 11: The King of Fighters XIII – izbor boraca



Slika 12: The King of Fighters XIII – izbor redosleda

Zamislamo sada igrača X kome je cilj da dođe do vrha lestvice, i bude broj jedan na listi. Da bi ispunio cilj, služi se lukavom taktikom: izbegava mečeve protiv igrača sa velikim brojem BP-a, i koncentriše se na početnike, koji nemaju puno BP-a. Samim time, uspeva svaki odigrani meč da dobije, mada za svaku pobjedu dobija malo bodova. Uspen je spor, i zahteva dugo vremena, ali nakon par meseci, X je pred ciljem, još par pobjeda, i dospeće na prvu poziciju. Upravo tada, X primećuje igrača Y, koji po bodovima deluje kao početnik. “ Super, još jedna žrtva”, misli u sebi X dok izaziva Y-na borbu. Oba igrača su izabrala likove, odredila redosled, i meč je počeo. Međutim, nešto nije kako treba. Y pokazuje zavidan nivo veštine, toliki da u svakom pogledu prevazilazi X-a. Ispostavlja se da je Y veoma dobar igrač, koji je dosta mečeva odigrao offline protiv par drugih igrača koji su takođe veoma dobri u igri, i samim time je i on uspeo da dosta toga nauči i da napreduje. Razlog niskog broja BP-a je taj što je Y tek pre koji dan počeo da igra online. Šokiran razvojem situacije, X sve to shvata tek kada je za njega već prekasno – treći lik njegovog tima je pri kraju sa energijom. Svestan toga da bi veoma lako sav njegov višemesečni trud mogao biti uzaludan, X počinje u krajnjem očajanju da traži bilo kakav izlaz iz ove dramatične situacije. I uspeva da ga nađe – taman kada njegov poslednji borac treba da padne, X brzinom svetlosti skače ka ruteru, u letu hvata UTP kabl koga bukvalno čupa iz rutera, i time prekida vezu između konzole i interneta. Zahvaljujući tome, meč ne može da se završi kako treba, i zbog toga se računa kao da do njega nije ni došlo. Što se X-a tiče, misija obavljena, bodovi su odbrani.



Slika 13: The King of Fighters XIII – trenutak pre prekida konekcije

Već je rečeno da ovakav tip varanja predstavlja čest problem kod kompetitivnih online igara, tako da je potrebno sprovesti i mere koje će obeshrabriti takve pokušaje. Najčešće rešenje predstavlja oduzimanje poena onoj strani koja je prekinula vezu. Upotrebom te metode nastaju dva problema. Prvi problem vezan je za činjenicu da broj

izgubljenih bodova zbog prekidanja veze za jednu stranu može biti manji nego broj bodova bez koje bi ta ista strana ostala da izgubi meč. Rešenje bi mogli da tražimo u proračunu bodova pre početka meča, gde bi mogli videti i koliko koja strana može dobiti i izgubiti, i u skladu sa time izračunati ko koliko bodova gubi ako se diskonektuje. Drugi problem je činjenica da do prekida veze može doći zbog uticaja drugih faktora, na koje sam igrač ne može da utiče (npr. nestanak struje). Na žalost, tu pomoći nema, tako da je to jedna od mana koja predstavlja manje zlo, u odnosu na ne kažnjavanje diskonektovanja.

U želji da umanje problem, developeri često omogućavaju igračima da se ponovo konektuju u samu partiju, posebno ako je u pitanju igra gde se više ljudi, podeljenih u dva tima, bore jedni protiv drugih, i cilj je da tim pobedi. Jedna od takvih igara je i *League of Legends*. Dva tima se sučeljavaju, svaki od po pet igrača, i svaki igrač bira po jednog heroja. Cilj je uništiti protivničku bazu. U takvom okruženju dešava se da se neko ne snađe najbolje.

Recimo da P, koji je član tima A loše igra, i to iskoriste članovi tima B, tako što stalno ubijaju heroja kojim upravlja navedeni igrač. Time članovi tima B postaju sve jači i jači, što opet dovodi do toga da su sve manje šanse da tim A pobedi. Na kraju, ljut i razočaran, prekida igru i izlazi iz iste. Da se partija ne bi prekinula za sve igrače, data je mogućnost da se P ponovo konektuje, dok tim A dobija bonus u vidu bržeg jačanja njihovih heroja, jer se smatra da je veliki hendikep ako tim nema svih pet igrača, pa se na taj način gleda taj hendikep anulirati ili bar smanjiti. Međutim, u ovom konkretnom primeru, da li je baš tako? Kao prvo, P više nije u igri, što automatski sprečava tim B da iskoriste njegovo loše igranje, što je samim time prednost za tim A. Pored toga, tim A dobija dodatni bonus, tako da mogu ravnopravno da se bore protiv tima B, iako imaju manjak u ljudstvu. Uzevši sve to u obzir, postoji solidna mogućnost da tim A dobije partiju. Ako uzmemo u obzir taj scenario, gde je potrebno još samo malo da tim A pobedi, moguće je da se desi i to da se P ponovo priključi igri, maltene u momentu kada je sve gotovo. Šta onda to znači, ako uzmemo u obzir da *League of Legends* takođe koristi ELO sistem bodovanja za svakog igrača posebno? To znači da P nije registrovan kao igrač koji je napustio partiju, i isto tako, da je P ni kriv ni dužan pobedio, iako bi tim A izgubio, da je P igrao svo vreme. Upravo iz tog razloga je davanje vremena za rekonekciju mač sa dve oštrice.

Još jedna od čestih metoda varanja ovoga tipa predstavlja zloupotreba laga. Lag predstavlja kašnjenje odziva usled nekoliko faktora: same konekcije prema serveru, gubitak paketa, loših performansi same klijent mašine, opterećenosti servera itd. Već smo spominjali kako je kašnjenje veliki problem kod peer-to-peer arhitekture, ali čak i online igre sa centralizovanim serverima nisu pošteđene toga. Usled laga, mnoge igre prestaju da funkcionišu kako treba. Efekti su takođe različiti, negde je to generalno usporavanje za sve učesnike, no u većini slučajeva je u pitanju ponašanje lika igrača koji laguje, u poređenju sa ostalim učesnicima igre.

Recimo, heroj koji laguje, ostalima deluje kao da stoji u mestu duži period, i iako je napadan, njegova životna energija (eng. health) se takođe ne menja. Odjednom, taj

isti lik nestaje sa svoje pozicije i pojavljuje se na drugoj lokaciji, sa smanjenim health-om, kao da se teleportovao. Iz ovoga se vidi koliki to problem može da bude i kada svi igraju pošteno. Zamislite sada nekoga ko je rešio da namerno kvari igru drugima, i zato namerno usporava svoj kompjuter i opterećuje svoju internet konekciju, najčešće koristeći druge uređaje koji su povezani na lokalnu mrežu (eng. lag-switch). U najgorem slučaju, to omogućava onome ko vara, i pritom je gubio da preokrene rezultat u svoju korist, jer njegovi protivnici nisu u mogućnosti da se snađu u datim uslovima, koji ionako nisu u skladu sa pravilima igre. Kao moguće rešenje, server može izdati challenge klijentu, i ako ne dobije odgovor u određenom periodu, prekida se veza i klijent se automatski diskonektuje sa servera. Naravno, i u ovom slučaju treba voditi računa o mogućnosti da lag može nastati usled faktora na koje klijent ne može uticati.

Varanja putem iskorišćavanja procedura u igri su često specifična za svaku igru pojedinačno. Naravno, primenom metoda kojim se takvo ponašanje detektuje i kažnjava smanjuju se takvi slučajevi, ali da bi same metode bile odgovarajuće i efikasne, potrebno je konkretnu igru detaljno analizirati, i tu leži najveći izazov kod borbe protiv ovakvog tipa varanja.

3.3.4 Varanje putem iskorišćavanja veštačke inteligencije

Razvoj oblasti veštačke inteligencije takođe predstavlja značajan korak u informacionim tehnologijama. Naravno, to je imalo i veliki uticaj na video igre. Zahvaljujući primeni veštačke inteligencije, single – player igre su dobile dodatnu dinamiku. Odjednom, ponašanje neprijatelja u igri nije bilo toliko predvidljivo, i imalo je više smisla sa logičke strane. Kod multi – player i online igara, to se pokazalo kao veoma dobro za učenje pravila igre, i poboljšanje veština igrača, naročito kada su u pitanju takmičarski orijentisani naslovi.

Recimo da ste kupili neku igru i željni akcije, odmah nakon instalacije priključujete se online partiji. Naravno, nemate prethodnog iskustva, i zato provodite vreme pokušavajući da naučite osnovna pravila. Nažalost, vaši protivnici nemaju razumevanja za vašu situaciju, i napadaju vas odmah bez imalo milosti. Ekvivalentna situacija bi bila da ste početnik u šahu, ne znate čak ni kako se koje figure kreću po tabli, a igrate protiv Garija Kasparova. U situaciji ste onda da stalno gubite, možda naučite ponešto, ali u većini slučajeva samo služite kao “glineni golub”. Često u takvim situacijama dolazi i do frustracije, kao i do gubitka želje za daljom igrom. Developeri su, u nameri da takav problem reše, omogućili igračima da simuliraju online mečeve, ali da pritom umesto da se suočavaju sa drugim igračima, za protivnike imaju kompjuterski kontrolisane likove, kojima je uz to moguće odrediti nivo efikasnosti. Sjajan način za uhodavanje, uz znatno manje stresa.

Ono što je loša strana kod primene veštačke inteligencije je mogućnost njene zloupotrebe od strane igrača. Kao potvrdu efikasnosti veštačke inteligencije možemo uzeti

čuveni meč između pomenutog gospodina Kasparova i računara Deep Blue⁸. Igre bilo koje vrste mogu imati puno opcija i kombinacija istih, ali u suštini je broj i jednih i drugih ograničen, što uz odgovarajuće programiranje i logiku, može dovesti do toga da mašina bude efikasnija čak i od ljudi koji su veoma dobri u pomenutoj igri. Recimo da uzmemo neku online verziju šaha koja takođe koristi Elo sistem bodovanja. Može da se desi da igrač dođe u posed programa koji sadrži algoritam za igranje šaha, kao i bazu podataka sa mnoštvom partija odigranih kroz istorijat igre. Nakon logovanja, igrač pokreće aplikaciju, koja umesto njega vuče poteze. Ako imamo u obzir da je jedan sistem uspeo da porazi jednog šahovskog velemajistora, kolike onda šanse ima jedan prosečan igrač protiv tako nečega, makar u ovom slučaju sam program bio manje efikasan od Deep Blue?

Veća efikasnost kod igranja i prednost u analizi u odnosu na čoveka nisu jedini razlozi zbog čega se primenjuje veštačka inteligencija u cilju varanja. Uzmimo primer online igara koje se sastoje od velikih konzistentnih online svetova sa puno igrača aktivnih na njima (eng. Massively Multiplayer Online Role-Playing Games, skraćeno MMORPG). Već smo napomenuli, da je cilj developera i to da igrači što duže igraju njihovu igru. Jedan od načina kako je to rešeno kod velikog dela MMORPG igara je taj da, kada igrač koji je početnik krene sa igrom, on nema pristup svim lokacijama virtuelnog sveta. Često kao uslov da to promeni, igrač mora da ispuni određene zadatke (eng. quests).

Šta je problem sa time? Najčešće su ti zadaci pravljani tako da oduzmu što više vremena. Samo po sebi to ne bilo toliko strašno, ali nažalost, takvi zadaci često umeju biti tipa: “Idite i pobedite 1000 kompjuterski vođenih protivnika”. Mi time zapravo dobijamo jedan period dosade, kroz koji moramo proći, da bi dalje napredovali. Upravo zbog toga, igrači su došli na ideju, da koriste programe koji umesto njih upravljaju njihovim likom i završavaju im takve zadatke. U pitanju je značajna ušteda vremena i živaca. Takve aplikacije se na engleskom nazivaju “bots”, i uz pomoć njih je moguće doći do velike prednosti u odnosu na ostale igrače.

Jedna od mogućih metoda zaštite predstavlja upotreba CAPTCHA kodova⁹. CAPTCHA (skraćeno od Completely Automated Public Turing test to tell Computers and Humans Apart - Potpuno automatizirani javni Turingov test za razlikovanje čoveka od računara) predstavlja test koji se lako generiše od strane računara, ali se veoma teško ili uopšte ne može rešiti od strane istog, dok rešavanje nije problem za čoveka. Najčešći primeri CAPTCHA kodova su prepoznavanje slova na slici. Sačestim promenama sistema i pravilima generisanja CAPTCHA kodova se takav tip zaštite pokazao maltene 100% efikasnim, posebno od kako je kompanija Google preuzela na sebe generisanje testova za druge sajtove. Ipak, mnogi smatraju da rešavanje takvih testova ometa igrače u toku igre. Tu se postavlja i mogućnost unajmljivanja ljudi kojima bi se automatski prosledio svaki CAPTCHA zahtev, koga bi onda oni mogli da reše umesto programa, i time omogućući botu da nastavi sa svojim poslom.

⁸[http://en.wikipedia.org/wiki/Deep_Blue_\(chess_computer\)](http://en.wikipedia.org/wiki/Deep_Blue_(chess_computer))

⁹<http://en.wikipedia.org/wiki/CAPTCHA>

Drugi pristup predstavlja analizu saobraćaja između klijenata koji koriste i ne koriste botove sa jedne strane, i servera sa druge. Analizom se došlo do nekoliko kriterijuma po kojima možemo utvrditi razlike: brzina unosa komandi (botovi generalno unose komande ravnomerno, po određenom pravilu), razlika u generisanom saobraćaju između klijenata sa i bez botova (veće varijacije kada igraju ljudi u odnosu na botove), kao i osetljivost na stabilnost same konekcije (prisustvo laga tj. kašnjenja u odzivu mnogo više pravi problema ljudima nego botovima, tako da u uslovima gde postoji lag, botovi i dalje izvršavaju sve akcije besprekorno bez pauze, za razliku od ljudi).¹⁰ Mana ovakvog pristupa predstavlja utrošak resursa koji su potrebni za nadgledanje.

3.3.5 Varanje putem iskorišćavanja grešaka (bugs) u programiranju

Svi znamo da ko radi, taj i greši. Koliko god se neko trudio da sve uradi savršeno, uvek negde nastane neka greška. Programiranje se po tome ne razlikuje od bilo kog drugog posla. Greške koje nastaju u radu aplikacije nazivaju se bagovi (eng. bugs). Danas je maltene nemoguće naći aplikacije koje rade 100% savršeno, bez ijedne mane i бага. Stvari se dodatno komplikuju sa povećavanjem kompleksnosti i broja opcija u okviru samih aplikacija, među koje spadaju i video igre. Stotine i stotine linija koda, koje programer mora da proveri, što postaje ogroman problem, posebno ako je pritom potrebno ispuniti određeni rok za izdavanje igre.

Koliko je situacija po pitanju testiranja ozbiljna pokazuje činjenica da se javila potreba za zapošljavanjem ljudi koji testiraju igru (eng. playtesters), čiji je zadatak da igraju igre koje su u fazi razvoja, u cilju otkrivanja bagova i drugih problema. Ali, nažalost, čak i sa velikim timom playtestera, moguće je izdati naslov koji ima u okviru sebe dosta bagova i grešaka koji pritom utiču na iskustvo samih igrača, opet najviše usled rokova. Uvek će važiti da su igrači ujedno u najbolji tester, najviše zbog vremena koje provedu uz igru, i mogućnosti da sve pregledaju detaljno. Nisu retki slučajevi da su igrači mesecima ili čak i godinama nakon izlaska otkrivali velike bagove u naslovima za koje se dotle verovalo da nemaju bitnih problema.

Naravno, ako uzmemo u obzir online igre, stvari postaju još dramatičnije, posebno kada se uzme u obzir broj igrača, i konstantne promene u okviru igre. Događa se i da bagova ima toliko, da igrači konstantno nailaze na njih, čak iako ne žele da varaju. Ako na to dodate i korisnike koji su skloni varanju, imate onda veoma neugodnu situaciju. Kao i kod prethodnih kategorija, varanje putem iskorišćavanja bagova može imati raznorazne efekte, počevši od benignih (iskorišćavanjem бага, igrač dobija mogućnost da malo brže završi neki quest), do veoma problematičnih (iskorišćavanjem бага, igrač dobija ogromnu količinu resursa, koje ne bi trebao da poseduje).

¹⁰ Chen KT, Jiang JW, Huang P, Chu HH, Lei CL, Chen WC. Identifying MMORPG bots: a traffic analysis approach. ACE06, June 14–16, 2006, Hollywood, California, USA.

Sve navedeno govori o tome koliko je bitno testirati kod igre, što je moguće više i češće. Nažalost, kada je u pitanju borba protiv ovakvog tipa varanja, tu ne može puno da se uradi u momentu kada je igra aktivna. Potrebno je stalno nadgledanje i proveravanje igrača. Takođe, čim se primeti bag, potrebno je to odmah objaviti i sprečiti dalju zloupotrebu, najčešće uvođenjem zabrane i kažnjavanja zloupotrebe pomenutog бага.

Na prvi pogled, varanja koja se sprovode iskorišćavanjem procedura i varanja putem iskorišćavanja bagova deluju identično. No, postoji razlika: kod iskorišćavanja procedura, varanje se vrši u skladu sa određenim idejama, pravilima i ponašanjem u igri koje su developeri očekivali. Sa druge strane, kod zloupotrebe bagova, varanje se vrši protivno očekivanjima developera, tj. igrač je u stanju da uradi nešto što developeri nisu ni pomislili da je moguće u okviru same igre.

Ovde je bitno napomenuti jednu stvar: bagovi sami po sebi nekada mogu biti i dobra stvar za igru, posebno ako je u pitanju neki takmičarski orijentisan naslov. Ovo zvuči paradoksalno, međutim kao primer bih uzeo igre iz žanra VS tuča, u koje spada i *The King of Fighters XIII*. U igrama ovog žanra, čest je slučaj da određeni naslovi među ljubiteljima budu popularni godinama. Samim time, dešava se da često budu otkrivene neke nove taktike i osobine koje mnogi smatraju bagovima, a često su nepoznanica čak i za same developere.

Recimo da je u okviru *The King of Fighters XIII* otkrivena taktika A, za koju niko drugi ne zna, izuzev igrača koji ju je otkrio. Taj igrač odlazi onda na turnir, gde prikazuje tu svoju taktiku, i ostvaruje zavidan rezultat. U većini slučajeva, drugi igrači takvu taktiku proglašavaju ne fer i posledicom бага, tražeći pritom zabranu primene taktike A, smatrajući da su njome smanjene šanse drugih igrača da pobeđu, osim ako i oni ne počnu da je primenjuju. Nekad su u pitanju stvarno dosta problematične stvari, koje budu ispravljene sledećom zakrpom, ali nekad budu takve taktike i ostavljene. Zašto? Zato što je, u ovome slučaju, nakon nekog vremena, drugi igrač otkrio taktiku B koja efikasno kontrira tehniku A. No, iako efikasna protiv A, taktika B se pokazuje slaba na primenu taktike C koja predstavlja “normalno” igranje, koje je poznato bilo igračima pre otkrivanja A. Šta je time dobijeno na kraju? Igra je odjednom postala još zanimljivija, jer imamo više opcija i načina igranja, što dalje produžava vek same igre. Upravo iz tog razloga, treba nekada imati i u vidu da nekada nešto što se zove bag, može doneti i poboljšanje same igre, čak i kada developeri uopšte nisu planirali takav rezultat.

3.3.6 Varanje putem unutrašnje zloupotrebe

Uprkos svim sigurnosnim metodama koje mogu biti primenjene, čak i najsavršeniji sistemi odbrane lako bivaju probijeni iako neko iznutra radi protiv istog. Svi znamo kako je Troja pala u ruke neprijateljima, iako je pre toga godinama odolevala napadima. Isti tako i online igre mogu imati velikih problema u slučajevima kada su na

bitnim pozicijama ljudi koji nisu pouzdani, i u još goreм случају, zloupotrebljavaju svoj položaj radi sopstvene koristi.

Uzmimo na primeri neku popularnu MMORPG igru. Održavanje sistema za takvu igru predstavlja veoma opsežan i zahtevan posao po pitanju resursa. Kao jedan od elemenata održavanja MMORPG-a javlja se potreba za proverom da sve funkcioniše kako treba. Često je najbolji, najefikasniji, a ponekada i jedini način da se primete problemi u okviru igre aktivno uključivanje i igranje iste. U tu svrhu, kompanije najčešće unajmljuju administratore, čiji je zadatak da sa svojim likom prelaze sa jedne na drugu stranu virtuelnog sveta, u potrazi za igračima koji vrše zloupotrebe. Samim time, administratorima je omogućena velika sloboda po pitanju mogućnosti koje imaju. Naime, likovi kojima oni upravljaju mogu imati osobine kao što su neuništivost, neograničena snaga, ogromna brzina itd. , od koje neke su čak i nemoguće za likove običnih igrača.

I tako možemo sada zamisliti nekog administratora, koji radi svoj posao naizgled korektno, ali, kada ga niko ne nadgleda, on iskoristi priliku da uradi i ponešto više od onoga što bi trebalo. Šta bi na primer mogla jedna osoba na toj poziciji da uradi?

Prvi primer je najbenigniji: recimo da je neki rođak / prijatelj administratora počeo da igra pomenutu igru. Naravno, kao početnik, snalaženje mu nije baš jača strana. Administrator, u želji da pomogne, reši da npr. umesto početnika završi za njega neki zadatak, i time ubrza njegov razvoj unutar igre. Samome administratoru, s obzirom na mogućnosti koje ima, samo po sebi to nije nikakav problem. I takva priča može dugo trajati.

Kao još goru varijantu, administrator može početniku dati predmete koji njegovog lika čine mnogo jačim, čime se početniku opet može dosta pomoći. Stvar time postaje još problematičnija, ako igra podržava međusobnu borbu igrača, gde svaka, čak i najmanja prednost puno znači. Već smo rekli koliko igračima smeta kada jasno vide da neko dolazi dosta lakše do cilja, zbog nepravedne prednosti.

No, zamislimo sada da administrator sve to radi za određenu nadoknadu, najčešće finansijskog tipa. Dakle, pored plate koju dobija za to što radi, administrator može, prodajom predmeta u igri može zaraditi solidnu količinu para, a inače te predmete sam može lako da stvori, unosom odgovarajućih komandi. Zastrašujuća pomisao, zar ne?

Zaštita protiv unutrašnje zloupotrebe se generalno sprovodi putem određenih pravila i poštovanja istih. Potrebno je definisati šta zaposleni smeju da rade na računaru, gde pritom svako ima određeni nivo privilegija i odgovornosti u skladu sa time. Takođe, potrebno je izvršiti i obuku zaposlenih po pitanju sigurnosnih problema i rizika. Pored toga, postoji i mogućnost nadzora i pravljenja zapisa korišćenja (eng. logs), gde detaljno možemo videti šta je ko i kada radio, u cilju sprečavanja zloupotrebe. Najefikasniji način sprečavanja zloupotrebe predstavljaju razni stimulansi i druge beneficije koje zaposlenima bude jaku želju da svoj posao, i samim time čine da im bilo kakva ideja iskorišćavanja svoje pozicije bude veoma rizična, pa i neisplativa u skladu sa tim rizikom.

4. Ostale sigurnosne pretnje

Pored varanja, kod online igara su prisutni i drugi tipovi sigurnosnih problema. Šta je specifično kod njih? U suštini, napadač se ne ograničava samo na svoju mašinu, već pokušava da direktno uvuče u svoje planove što je moguće više drugih elemenata nad kojima može steći kontrolu, i iskoristiti ih za ostvarivanje svojih ciljeva. Dakle, dok onaj ko vara gleda da sve što treba uradi preko svoje klijent mašine, napadač će gledati npr. da iskoristi i server igre, druge klijente koji igraju datu igru, protokol itd.

Naravno, rezultati takvih napada mogu da budu identični kao i kod primene varanja. Česti su slučajevi da neki od napadača uspeju da dobiju kontrolu nad serverom, samo da bi dobili mogućnost da svog lika u igri ojačaju i nakon toga prekinu vezu sa serverom, i uopšte ne promene niti poremete igru na bilo koji drugi način. Ovo se ipak klasifikuje kao napad, jer je u ovome slučaju pored klijent mašine, iskorišćen i server igre.

Nakon svega ovoga može se postaviti pitanje, zašto onda i varanja putem dogovora ne svrstati u ostale pretnje, pošto i tu važi činjenica da je više klijent mašina umešano u priču? Mada ta primedba stoji, treba imati u vidu da kod dogovora, više igrača dobrovoljno učestvuju u poduhvatu, dakle nikakav napad i / ili iskorišćavanje direktno umešanih nije prisutno, samim tim, ovde je reč o varanju. Sa druge strane, da je jedan napadač preuzeo kontrolu nad ostalim klijent računarima, i na taj način došao do neke prednosti, onda već nije reč o varanju, nego o napadu. U ovome delu navešćemo neke tipove sigurnosnih pretnji.

4.1 Neispunjavanje dogovora oko razmene resursa

Vratimo se na jedan već pomenuti slučaj: dakle, igrali ste igru gde ste u ulozi ratnika prošli kroz puno avantura, i onda ste dobili ponudu da za određenu količinu novca, vaš nalog date na korišćenje drugom igraču, koji ne želi da provede puno vremena uz igru da bi došao na taj nivo na kome ste vi trenutno. Međutim, zamislite sada sledeću situaciju: nakon što ste dali korisničko ime i password, niste dobili nikakvu uplatu. Moguća je i suprotna situacija, ako ste vi kupac, uplatili ste novac, ali niste dobili navedene informacije za korišćenje naloga. Ako uzmemo u obzir da neke od tih transakcija umeju biti reda veličine stotinu i više evra, možete onda zamisliti koliko je privlačno prevarantima da uđu u tu priču.

Vezano za primerprodaje naloga, većina online igara zabranjuje bilokakvu prodaju, upravo zbog navedenih mogućnosti prevare. Jedna od najpopularnijih MMORPG igara, *World of Warcraft*, u okviru klauzule zabranjuje bilo koji tip kupoprodaje naloga, po cenu ukidanja naloga (eng. "banned account"), u slučaju da do toga dođe.

Uzmimo sada primer razmene resursa (najčešće predmeta) u okviru igre, što važi za dozvoljenu radnju u okviru igre. Kako se recimo obavljala razmena u MMORPG igrama? Često, pošto su to ipak bili počeci, igre same po sebi nisu imale razvijen sistem za direktnu razmenu između igrača. Recimo da želite da nekome date određeni predmet, koji vama ne znači puno, a da zauzvrat dobijete predmet koji bi vama puno pomogao u igri. Jedini način da izvršite razmenu bio bi da vi vaš predmet uzmete iz vašeg inventara, i pustite da on padne na zemlju. Igrač sa kojim vi trebate da odradite razmenu takođe to odradi sa svojim predmetom. Onda, vi pokupite predmet drugog igrača sa zemlje, a drugi igrač pokupi vaš predmet. Naravno, sve je u redu ako su obe strane korektne, ali šta ako to nije slučaj? Najčešće jedna strane pokupi tuđ predmet, ali ne ostavi svoj radi razmene. Nažalost, oštećenoj strani ne preostaje ništa drugo, nego da se pomiri sa tužnom sudbinom.

Uvođenjem interfejsa za razmenu predmeta, problem je barem delimično rešen. U suštini, kada se pokrene razmena između igrača, dobija se poseban inventar podeljen u dva dela, gde jedan deo predstavlja predmete koje nudi jedan igrač, a drugi deo predmete koje nudi drugi igrač. U takvoj situaciji, razmena se obavlja kada obe strane imaju nešto dato, tako da ne može da se desi da bilo ko ostane bez ičega. Međutim, i tu imamo problema. Veliki broj online igara u okviru sebe sadrži hiljade i hiljade različitih predmeta. Naravno, pored balansiranja igre, svrha postojanja tolikog broja predmeta je i raznovrsnost, kojom se opet zadržava dalja želja za igranjem.

Ali, modelovanje takvih predmeta, tako da svaki bude vizuelno jedinstven, zahtevalo bi dosta vremena i finansijskih resursa. Iz tog razloga, developeri često idu na varijantu, gde predmete podele na klase, i predmeti u okviru svake klase vizuelno izgledaju identično, iako mogu imati velike razlike po pitanju bonusa koji se dobijaju posedovanjem istih. Najčešći problemi su bili vezani za činjenicu da ste kod razmene vi samo videli vizuelno koje je klase sam predmet, ali niste u mogućnosti da proverite koje je on snage. Samim time, uvek bi postojao rizik da vas neko prevvari, time što vam obeća predmet određene snage, a onda vam da predmet iste klase, ali mnogo slabiji. Naravno, opet se priča svodi na to, da oštećena strana nema kome da se žali.

Kako onda rešiti ovakav problem? U slučaju razmene, čim se ona pokrene, pokreće se i zahtev prema serveru, koji nadgleda šta se dešava. Kao prvo, oba igrača koji učestvuju u razmeni su u prilici da vide sve predmete koje suprotna strana nudi, i takođe kompletan opis njihove snage i osobina. Kao drugo, oba igrača, kada stave predmete za razmenu, moraju da potvrde u datom roku da se slažu sa datom razmenom. Kao treće, server može da uradi proveru da li su sami predmeti u redu, za dodatnu potvrdu. Čim se svi ti uslovi poklope, dolazi do razmene, i time se smanjuje mogućnost prevare i greške. Ali, šta ako razmenu obavlja neko ko je početnik, i biva uveren da menja svoj vredan predmet za još vredniji, a zapravo to nije slučaj? Najbolji način rešavanja ovoga je, u okviru prikaza osobina predmeta, i prikazivanje cene predmeta u monetarnim jedinicama koje se koriste unutar igre, radi mogućeg upoređivanja vrednosti.

4.2 Napadi modifikovanjem izlaznih informacija na klijent/server protokol

Recimo da smo, kao developeri jedne online igre, uradili nešto što su mnogi oduvek maštali: uspeli smo da napravimo klijent aplikaciju, kojom igrač, koliko god manipulisaao, ne može da promeni ništa bitno, što bi mu dalo šansu da lakše pobedi u igri. Srećni i zadovoljni, ulogujemo se i mi u igru, da sve još jednom proverimo. Međutim, odjenom, naćemo na protivnika koji, koliko god ga mi puta napadali, ne gubi životnu energiju. Kako je to moguće, kada je klijent aplikacija sigurna, a na samom serveru ne registrujemo bilo kakav upad? Postoji mogućnost da nismo predvideli put kojim informacija ide od klijent mašine do servera.

Napadi modifikovanjem izlaznih informacija na klijent/server protokol su mnogobrojni. Na primer, napadač može koristiti packet snifer, program koji omogućava pregled svih informacija koje se prenose putem mreže. Samim time, napadača može “u letu” da promeni izlazne vrednosti, gde rezultati mogu biti identični kao i kod modifikovanja klijent aplikacije. Naravno, šifrovanjem izlaznih podataka možemo sprečiti tako nešto.

I ovde imamo još problema: recimo da igramo neku online igru gde upravljamo avionima, i vodimo borbe sa drugim avionima. Pri tome, jedino o čemu treba da brinemo je upravljanje i brzina paljbe. Pošto smo solidni igrači, ne zaostajemo ni za najboljima puno po bilo kom kriterijumu. Samim time, deluje nam neverovatno kada odjednom naletimo na protivnički avion, čija brzina paljbe je okvirno deset puta veća od naše. Koliko god čovek brzo stiskao taster, postoji određena pauza između ispaljivanja, jer postoje limiti kod ljudskih refleksa. Ali, ovde pauze nema, rafal za rafalom bez ijednog momenta predaha, dok na kraju ne izgubimo svu energiju. Kako je sad moguće da se ovako nešto desi, ako smo primenili šifrovanje kod informacija koje razmenjuju klijent i server?

To je moguće zahvaljujući napadu ponavljanja paketa (eng. replaying packets). Ovde napadač putem packet snifera može utvrditi koji odlazni paketi sadrže podatke koji govore da je pritisnut taster za paljbu, i samim time te iste pakete kopirati i ponovo poslati jedne za drugim, praveći tako jedan veliki lanac bez ikakve pauze između. Način na koji se ovo može rešiti je dodela određene sekvence brojeva, koja se menja sa svakim sledećim paketom. Čim sekvenca nije u skladu sa očekivanom od strane servera, paket se odbacuje.

Još jedan tip situacije bi bio u slučaju da opet, na klijent strani posedujemo neke kritične informacije, kao npr. informacije vezane za količinu životne energije koju ima lik koga igrač vodi. I sada, iako smo klijent aplikaciju obezbedili, igrač i dalje ide kroz igru, pritom pobeđujući sve moguće neprijatelje, a da ne izgubi nimalo energije. Sve to uspeva zahvaljujući tome, što pomoću packet snifera, igrač blokira sve poruke sa servera koje mogu imati loše posledice po njegovog lika, konkretno u ovom slučaju, sve poruke koje kažu da je lik izgubio određenu količinu životne energije. Rešenje za ovakav tip napada je

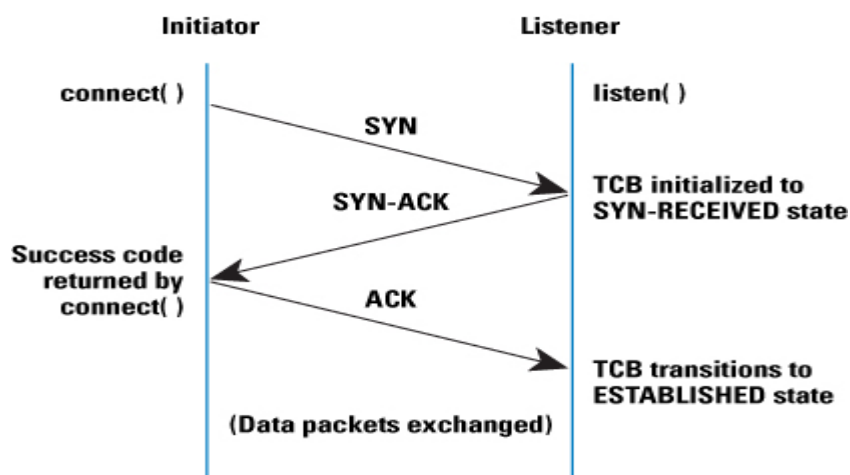
već spominjano, prebacivanje svih kritičnih informacija na server. No, čak i u tom slučaju, potrebno je obratiti pažnju na ovakve napade, jer umeju biti dobro skriveni.

4.3 (D)DoS napadi

Napadi uskraćivanja usluga (eng. Denial of Service, skraćeno DoS) su krajem devedesetih predstavljali više kratku i prolaznu neugodnost, nego što su se smatrali ozbiljnim sigurnosnim problemima. Najčešće žrtve su bili pojedinačni računari u okviru sistema, iskorišćavajući razne propuste operativnih sistema. Neki od napada koji su u to vreme bili efikasni su: preveliki paketi (program šalje ciljnom računaru paket veći od 65.535 bajta, čime je bilo moguće “srušiti” ga), prekomerno fragmentiranje (kada se poslati podaci fragmentiraju u pakete, moguće je da pošiljalac zada da se podaci maksimalno usitne, tako da prijemnik, usled velikog broja paketa provede previše vremena spajajući iste), itd. Srećom, većina takvih propusta je bila relativno brzo zakrpljena, tako da danas sa ovakvim slučajevima nemamo problema.

Nažalost, razvoj DoS napada se može uporediti sa kompjuterskim virusima: u početku pretežno bezazleni, da bi kasnije evoluirali u jedan od najvećih sigurnosnih problema, ne samo za online igre, već i za bilo koje druge informacione sisteme povezane na Internet. Razlog sve češće upotrebe ovakvih tipova napada je u činjenici da je vremenom postalo osetno lakše poremetiti rad sistema nego upasti i preuzeti kontrolu nad istim, a kod online igara, veliki broj napadača upravo cilja na to da sistem onesposobi za normalno funkcionisanje.

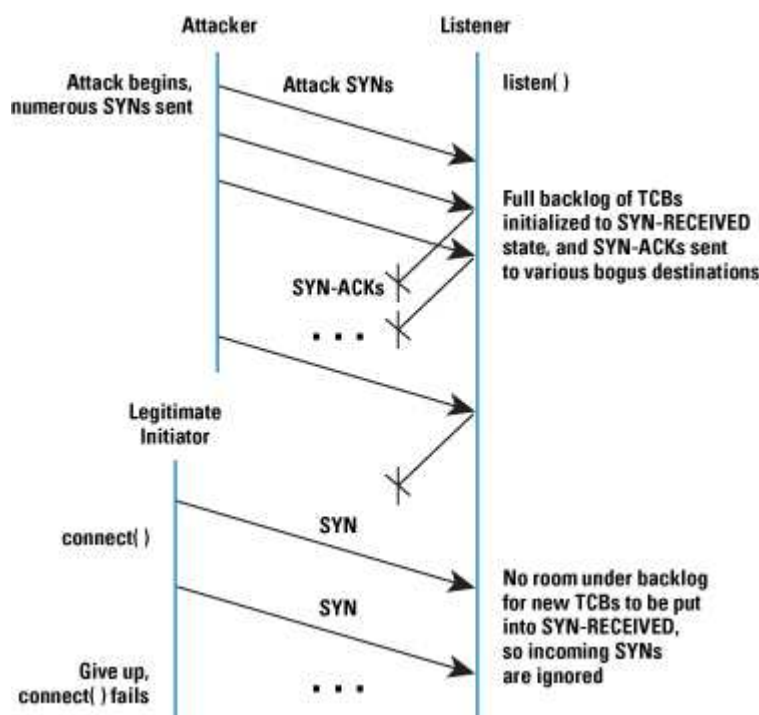
Jedan od najčešćih DoS napada predstavlja plavljenje SYN paketima putem TCP protokola. O čemu je ovde reč? Da bi se mogla ostvariti komunikacija između dva računara putem TCP / IP protokola, potrebno je da se prvo obavi proces trostepenog usaglašavanja (eng. three-way shaking), kojim se zapravo uspostavlja veza između dva računara, i potvrđuje sposobnost slanja i primanja paketa.



Slika 14: Three-way handshaking

Dakle, kao što se vidi na slici, uspostavljanje komunikacije se sastoji iz tri koraka. Prvi korak predstavlja slanje SYN paketa od računara A, koji želi da uspostavi komunikaciju, računaru B. Računar B prihvata paket SYN, zatim potencijalnu TCP konekciju prebacuje u stanje SYN-RECEIVED, nakon čega računar B šalje računaru A SYN-ACK paket. Nakon što primi SYN-ACK paket, računar A kao odgovor šalje računaru B ACK paket. Kada ACK paket stigne računaru B, veza između A i B se uspostavlja, i komunikacija može započeti.

Kako sada ovu proceduru može da iskoristi napadač? Ključna stvar koju ovde moramo imati u vidu je ta da kada se konekcija nalazi u SYN-RECEIVED stanju, ona i dalje zahteva određene resurse da se održi. Samim time, zamislimo da je trenutno aktivno puno veza ovakvoga tipa na jednoj lokaciji. Iako se komunikacija u tom periodu ne ostvaruje, i dalje je prisutno zakrčenje puteva ka ciljnom računaru, tako da i računari koji hoće da uspostave vezu sa ciljnim računarom to ne mogu. Jedan primer može se videti na sledećoj slici.



Slika 15: Korišćenje SYN paketa za DoS napad

U ovom slučaju, napadač preko računara A kontaktira računar B, međutim on pritom koristi lažnu, neaktivnu IP adresu koju navodi kao adresu računara A, i šalje zahtev za uspostavljanje konekcije. To onda dovodi do situacije da, kada računar B pošalje SYN-ACK paket na datu lažnu adresu, odgovor ne stiže, i usled toga veza je i dalje u SYN-RECEIVED stanju. Pored zauzetih resursa, sistem B se podređuje uspostavljanju veze sa računarom A, sve dok ne prođe određeno vreme, nakon čega se veza odbacuje. To vreme

može varirati od slučaja do slučaja, ali to ne sprečava napadača da konstatno šalje SYN pakete kojima neprekidno zahteva uspostavljanje veze. Time efektivno može da se spreči komunikacija računara B sa bilo kojim drugim tačkom na mreži. Čak i sa relativno slabim protokom na svojoj strani (reda veličine 14 kb/s), napadač je u stanju da pošalje toliko zahteva da ciljna strana ne može da stigne sve da obradi, makar imala i mnogo veći protok i dosta jače mašine, kao što je to slučaj sa serverima na kojima se nalaze online igre.

DoS napadi su sposobni da pričine značajnu štetu. Nažalost, njihov napredak kroz vreme ih je učinio dosta opasnim. Vratimo se sada na priču vezanu za resurse napadača. Rekli smo da jedna prosečna mašina, sa relativno niskim protokom, može prouzrokovati solidne probleme. Ekvivalentno tome bi bilo kada želite da radite nešto što je dosta zahtevno, odgovorno, i za šta vam je potrebna puna koncentracija, a neko vas konstantno ometa, i to na veoma lak način i uz malo truda, i to toliko čini, da na kraju ne možete da izdržite sve to, i rešite da prekinete sa radom. Samo po sebi, to već zvuči previše naporno, ali, zamislite da situacija odjednom postane još gora, tako što sve više i više ljudi počinje da vas ometa. Zvuči kao noćna mora, zar ne?

Distribuirani napadi uskraćivanja usluga (eng. Distributed Denial of Service, skraćeno DDoS) se upravo baziraju na ovoj logici. U prvoj fazi napada pokušava se steći kontrola nad što više računara, tako što se koriste razni načini inficiranja istih putem najrazličitijeg softvera. Pomenuti računari koji su uspešno zaraženi, popularno nazvani zombiji, automatski stoje na raspolaganju napadaču, koji ih kontroliše putem programa za daljinsko upravljanje, popularno zvani botovi (eng. bots). Sami botovi za kontrolu koriste tehnike i skripte primenljivane u okviru usluge za interaktivnu komunikaciju putem Interneta (eng. Internet Relay Chat, skraćeno IRC). Kao što smo već videli, DoS napad samo sa jedne lokacije sa relativno malim protokom može biti problem, tako da može se zamisliti koliki je problem kada stotine, pa čak i hiljade zaraženih računara organizovano šalju SYN pakete na jednu lokaciju. U pitanju je toliko zagušenje protoka, da čak i najveći sajtovi na svetu mogu veoma brzo postati nepristupačni i nedostupni.

Na osnovu svega ovoga, može se zaključiti da, nažalost, ne postoje načini da se DoS napadi potpuno izbegnu. Sa druge strane, postoji dosta metoda koje se mogu primeniti da se šteta umanja. Jedna od metoda bi bila primena gotovih tehnologija za borbu protiv DoS napada, kao što su Cisco Guard i Top Layer. Zatim možemo primeniti i filtriranje ulaznog i izlaznog saobraćaja, gde odbijamo komunikaciju sa svim sumnjivim ip adresama. Same servere je potrebno konstantno ažurirati. Kod sistema koji koriste Linux, postoji opcija korišćenja SYN kolačića (eng. SYN cookie). Aktiviranjem ove opcije, sistem otkriva i zapisuje sve moguće adrese sa kojih se proceni da stižu SYN napadi, čime se omogućava legitimnim korisnicima da se konektuju čak i u slučajevima jakih napada. Kod Windows sistema, postoji mehanizam dinamičke podrške, gde kada dođe do popunjavanja reda za povezivanje, sistem automatski dodeljuje nove resurse.

5. Autentifikacija

Korišćenje Interneta, kao što većina ljudi danas zna, ima veliki broj mogućnosti i prednosti : audio i video komunikacija sa udaljenim prijateljima, brzo i efikasno nalaženje potrebnih informacija, mogućnost kupovine i obavljanja finansijskih transakcija na daljinu itd. U nekim slučajevima je upotreba Interneta za određene poslove postala poželjniji način, što je takođe dovelo do velikog skoka po pitanju broja ljudi koji se služe Internetom.

Imajući sve to u vidu, postavlja se sledeće pitanje: Ako je, recimo, naručen određeni predmet preko Interneta, kako znati ko ga je naručio? Ako je preko Interneta izvršena uplata za članarinu, kako znati koji je član izvršio uplatu? Zapravo, kako uopšte, u tom silnom broju korisnika, znati ko je šta tačno uradio? Proces autentifikacije predstavlja odgovor na to pitanje.

Autentifikacija (prevod grčke reči αὐθεντικός, što znači pravi tj. originalni) predstavlja proces utvrđivanja istinitosti nekog atributa ili entiteta.¹¹ U informatici to najčešće podrazumeva proces u okviru koga korisnik ili izvor informacija dokazuju da su to za šta se predstavljaju - drugim rečima, proces utvrđivanja identiteta korisnika koji pokušava da pristupi sistemu.¹² Na koji način se može utvrditi identitet onoga ko želi pristupiti sistemu? Postoje tri pristupa: na osnovu nečega što onaj koji pristupa zna, korišćenjem onoga što onaj koji pristupa ima, i na osnovu onoga što onaj koji pristupa jeste.

Korišćenje lozinki (eng. password) predstavlja primer autentifikacije na osnovu onoga što neko ko pristupa zna. Lozinka predstavlja određeni niz karaktera, koji povezan sa određenim korisničkim imenom (eng. username) čini par kojim se identifikuje korisnik. Sama lozinka može na primer biti datum rođenja, nadimak, PIN broj itd. Šta je prednost ovakvog pristupa? Lozinke su generalno lake za pamćenje, i najčešće logične. Ako je recimo neko kao lozinku stavio svoj datum rođenja, to je jedan podatak koji za samog korisnika od velikog značaja, i pritom je šansa da se taj podatak zaboravi veoma mala. Sledeća prednost je cena, pošto je izdavanje i definisanje lozinki besplatno, za razliku od drugih pristupa kod autentifikacije. Takođe, tu je i činjenica da je lozinke moguće po potrebi brzo resetovati i izdati nove.

Ono što je prednost za lozinke sa stanovišta korisnika je takođe mana istih po pitanju sigurnosti. Činjenica da su lozinke uglavnom logične ih istovremeno čini i riskantnim za upotrebu. Konkretno, neka pomenuti korisnik upotrebi datum svog rođenja kao lozinku, npr. 13121980, čime je obezbedio svoje podatke vezane za određeni online servis. Međutim, u isto vreme, neko iz njegovog okruženja, npr. njegov kolega sa posla, želi da pristupi njegovom nalogu da bi mu naneo štetu. Ovde vidimo problem, da napadač može lako da sazna dosta informacija o samom korisniku. Samim time, postoji velika

¹¹ <http://en.wikipedia.org/wiki/Authentication>

¹² <http://www.kombib.rs/recnik.php?slovo=A&rec=Authentication++autentifikacija>

verovatnoća da prilikom pokušaja da otkrije password, napadač upotrebi i neke podatke o žrtvi, među kojima je i pomenuti datum.

Naravno, sada može da se kaže da umesto očiglednih stvari, za lozinke treba koristiti složene izraze, koji nisu očigledni. I u praksi, takve lozinke su se pokazale kao dosta bolje po pitanju sigurnosti, ali loša strana je ta što je za korisnike teško da ih zapamte. To je još izraženije ako uzmemo u obzir da jedna osoba može biti korisnik velikog broja online usluga, i pritom za svaki servis ima drugu lozinku.

Mogu li se onda naći lozinke koje nisu toliko komplikovane za pamćenje, a opet relativno teške za razbijanje? Rezultati testiranja i ispitivanja su pokazali da su najbolje one lozinke koje se zasnivaju na frazama, jer se lako pamte, a opet su dovoljno duge da ih nije lako razbiti. Ali, iako su obavešteni o tome, veliki broj korisnika ne poštuje pravila, i samim time pravi dodatne sigurnosne probleme. Tu se onda pojavljuje mogućnost da lozinke ne budu kreirane od strane korisnika, već dodeljene od strane administratora, ili da periodično budu zamenjene.

Biometrika predstavlja drugi pristup kod autentifikacije, koristeći za identifikovanje ono što mi jesmo, tj. naš otisak prsta, oblik lica, prepoznavanje našeg govora, itd. Primeljivost biometrike nije velika kada su u pitanju online igre, upravo zbog problema privatnosti.

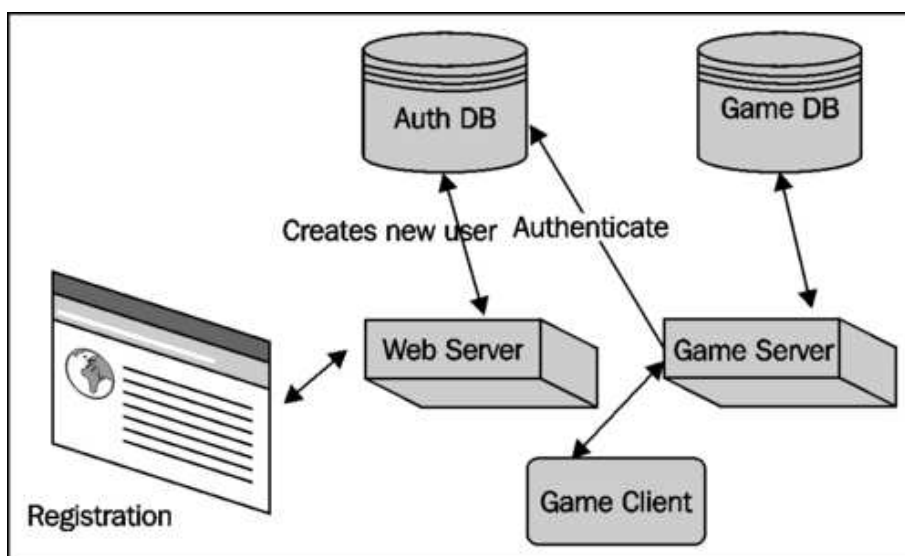
Treći pristup kod autentifikacije predstavlja upotreba nečega što posedujemo za potrebe identifikacije, npr. identifikaciona kartica, čijim skeniranjem dobijamo pristup objektu firme gde smo zaposleni. Ovde je rizik mogućnost da izgubimo predmet koji nam služi za autentifikaciju, čime nastaju ogromni problemi, zbog nemogućnosti autentifikacije, kao i zbog velike mogućnosti zloupotrebe, ako predmet bude nađen od osoba voljnih da se lažno predstavljaju u naše ime, i to iskoriste u druge svrhe.

Takođe, tu je i mogućnost dvofaktorske autentifikacije, koja kombinuje 2 od navedena 3 pristupa. Primer za to bi bila kreditna kartica (predstavlja nešto što posedujemo), koju koristimo da bi preko bankomata podigli određenu sumu novca sa našeg računa. Nakon ubacivanja kartice, potrebno je da ukucamo naš PIN kod (nešto što znamo), da bi autentifikacija bila uspešno obavljena.

5.1 Autentifikacija kod online igara

Proces autentifikacije kod online igara jer generalno sličan kao i kod drugih online servisa: budući korisnik se registruje, i nakon verifikovanja naloga, ima mogućnost logovanja na sistem sa svojim korisničkim imenom i lozinkom.

Uprošćeni primer kako izgleda i funkcioniše informacioni sistem online igre sa registracijom i autentifikacijom možemo videti na sledećoj slici.



Slika 16: Primer informacionog sistema online igre

Dakle, proces registracije i autentifikacije teče u nekoliko koraka. Prvo, budući korisnik se putem sajta online igre, koji se nalazi na web serveru registruje, tj. kreira svoj nalog, gde unosi željeno korisničko ime i lozinku, i takođe još nekoliko traženih podataka, npr. e-mail adresu. Unešeni podaci, pod uslovom da prođu validaciju, preko web servera bivaju prosleđeni bazi podataka korisnika, gde postaju novi unos u bazi, i time je registracija završena.

Nakon toga sledi autentifikacija, koja počinje tako što se korisnik putem klijent aplikacije online igre, sa podacima koji čine nalog loguje na server online igre. Server nakon primanja podataka od klijenta prosleđuje iste bazi korisnika, gde se vrši provera da li su uneti podaci identični sa podacima u bazi. Ako sve prođe kako treba, klijent aplikacija biva povezana sa serverom, i igrač može početi da igra igru.

Naravno, realnost je dosta komplikovanija, i teško da je moguće naći ovakav model, koji pritom funkcioniše bez problema i sigurnosnih rizika. Koji su najčešći problemi u ovakvim slučajevima? Za početak možemo uzeti u obzir loše navike korisnika, što podrazumeva i korišćenje lozinka koje je lako probiti. Napadač uvek može upotrebiti brute-force pristup, tj. pokušavanje svih mogućih kombinacija znakova za lozinku sa poznatim korisničkim imenom, sve dok ne dođe do dobitne kombinacije. Takav pristup inače zahteva puno vremena, ali ako je korisnik izabrao kratku i lošu lozinku, potrebno vreme za njeno probijanje je značajno manje.

No čak i ako je lozinka relativno duga, to opet ne znači da je dobra. Uzmimo za primer da je lozinka „automobil“. Brute-force napad bi ovde trajao neko vreme. Međutim, napadač je svestan jedne činjenice: mnogi korisnici koriste lozinke koje je lako zapamtiti, i koje su pritom reči koje se nalaze u rečniku. Upravo to omogućava „napad rečnikom“, koji predstavlja varijaciju brute-force napada, gde se umesto isprobavanja svih mogućih kombinacija, isprobavaju sve reči u okviru liste pojmova, preuzetih iz rečnika.

Kako onda izaći na kraj sa ovakvim tipovima napada? Prvi pristup bi bila upotreba raznih mehanizama provere lozinke prilikom kreiranja iste, gde ako lozinka ne ispunjava određene uslove (ne sastoji se iz kombinacije slova i brojeva, suviše je kratka, nema kombinacije malih i velikih slova, itd.), ona može biti odbijena i samim time, ceo proces registrovanja stopiran.

Druga opcija je korišćenje blokiranja logovanja nakon nekoliko neuspešnih pokušaja. Recimo, nakon 10 pokušaja, korisnik i dalje nije uneo pravi password, i samim time ne može u narednih 10 minuta da se loguje. S obzirom na to da brute-force metodi zahtevaju veliki broj pokušaja da bi dostigli cilj, i samim time puno vremena, svaka dodatna pauza još više ometa postupak.

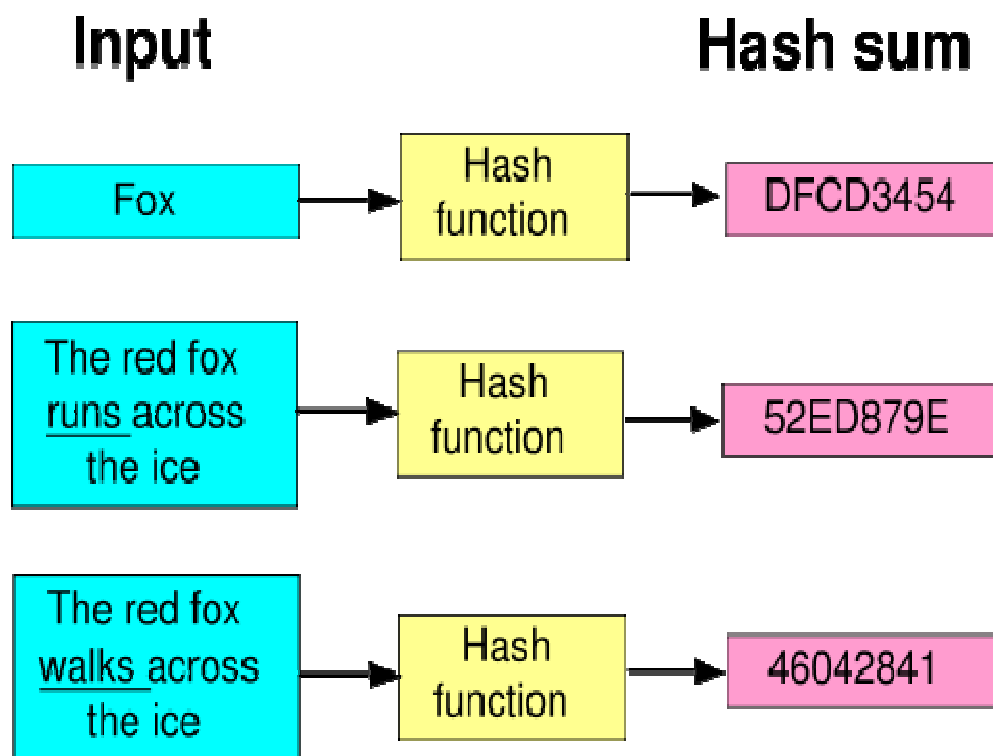
Problem kod primene ove metode nastaje prilikom izbora trajanja vremena blokiranja. Ako je vreme relativno kratko, tipa pola minuta, napad će se i dalje nastaviti, iako usporeno. Ako je vreme relativno dugo, kao u našem primeru 10 minuta, onda napad postaje neefikasan sa strane otkrivanja dobitne kombinacije. Međutim, napad onda postaje efikasan sa strane DOS (skraćeno od „Denial Of Service“, tj. uskraćivanje usluge), gde usled čestih pokušaja napadača, sam korisnik ne može da se loguje na sistem. Kao alternativa tome, može se upotrebiti dodatno sigurnosno pitanje, gde se korisniku nakon nekoliko neuspešnih pokušaja postavlja neko od pitanja, koje je on izabrao, i gde je potrebno da on ukuca odgovor koji je naveo prilikom registracije.

Sledeće što trebamo imati u vidu jeste činjenica da su korisničko ime i lozinka, kao autentifikacioni par, često zapisani jedno uz drugo. Konkretno, to znači da recimo svi igrači i njihovi podaci, među kojima i lozinke, mogu da se nađu u okviru jednog fajla, ili u našem slučaju, u okviru jedne baze. Već možemo zamisliti koliki bi bio problem ako bi neko neovlašćen uspeo da dođe do takvih fajlova ili da ima pristup takvoj bazi. Šta možemo uraditi u tom slučaju? Korišćenje heš funkcija (eng. „hash functions“) nam može ovde pomoći.

Heš funkcija¹³ predstavlja algoritam kojim se podaci smanjuju na fiksnu veličinu koja je osetno manja. Ideja je da, ako je recimo kod jednog korisnika lozinka „automobil“, a kod drugog „mačka“, i u jednom i u drugom slučaju heš vrednost bude veličine reči od 4 slova, za „automobil“ npr. „AFGH“, dok bi za lozinku „mačka“ heš bila „AABD“. Samim time, umesto da zapisujemo bilo gde same lozinke, možemo zapisati heš vrednosti, a nakon unosa lozinke od strane korisnika, računati heš vrednost iste, i onda upoređivati sa vrednošću u fajlu / bazi.

Heš funkcije bi trebalo da ispunjavaju određene uslove da bi se smatrale savršenim. Prvo, dobiti heš vrednost ($h(„a“)$) određene poruke „a“ bi trebalo biti brzo i lako. Drugo, obrnuto, dobiti poruku „a“ na osnovu $h(„a“)$ bi trebalo biti maltene nemoguće. Treće, ako imamo poruku „a“ i njenu heš vrednost $h(„a“)$, prilikom promene poruke „a“, mora se promeniti i $h(„a“)$. I na kraju, ako imamo poruke „a“ i „b“, nemoguće je da postoji $h(„a“) = h(„b“)$.

¹³ http://en.wikipedia.org/wiki/Hash_function



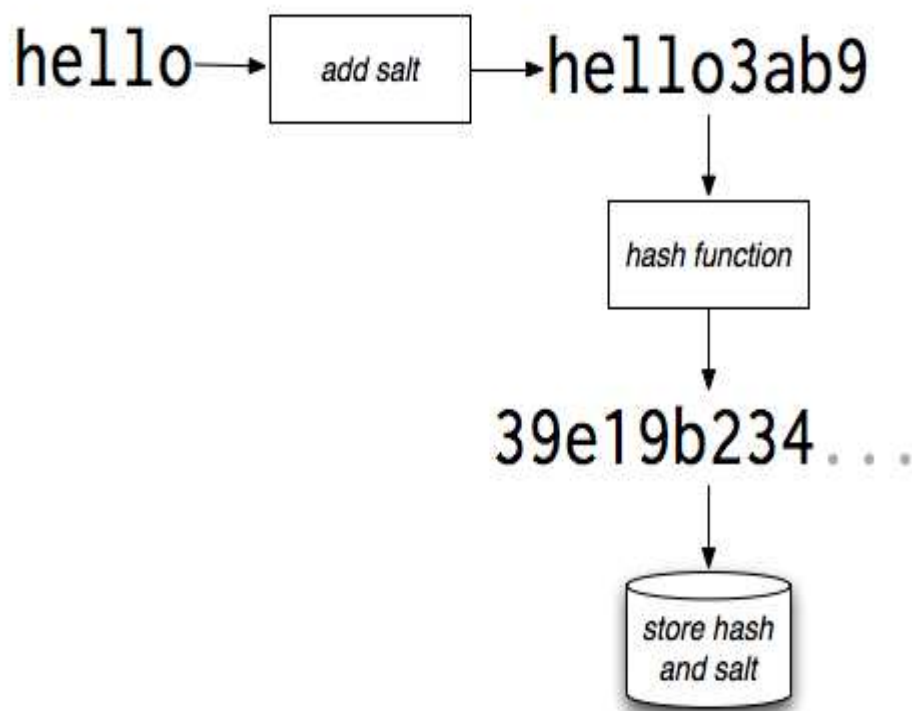
Slika 17: Primer funkcionisanja heš funkcije

Ovde deluje kao da smo rešili problem. Napadač, ako i dođe u posed fajla, ili ima pristup bazi, i dalje ne može da ono što se u njima nalazi direktno primeni. Ipak, to ne znači da je sve time rešeno. Napadač, recimo, može da isproba sve moguće heš algoritme na heš vrednostima, i ima solidne šanse da nađe pravi algoritam, i samim time otkrije lozinke. Kao dodatnu mogućnost, napadač može izračunati sve moguće heš vrednosti za sve reči u rečniku, i onda samo izvršiti direktno poređenje. Postoji li odgovor na ovako nešto?

Ovde je moguće koristiti salt¹⁴. Salt predstavlja slučajno izabranu vrednost, koja se dodaje lozinci, i zatim se od rezultata spajanja računa heš vrednost, koja se na kraju, zajedno sa salt-om zapisuje umesto lozinke. Na primer, korisnik je prilikom registracije kao lozinku izabrao „hello“. Već vidimo da je samo po sebi to loša lozinka, koju bi bilo relativno lako probiti. Nažalost, kao što smo već napomenuli, veliki broj korisnika ima tendenciju da bira ovakve lozinke.

Da ne bi svakog korisnika opominjali i menjali njegovu lozinku ručno, uzećemo da se automatski, za svakog pojedinačnog korisnika uzme odabrana lozinka, i njoj doda slučajno izabran salt. Ovde je to u pitanju niz „3ab9“, koga dodajemo nizu „hello“, i kao rezultat dobijamo „hello3ab9“. Nakon toga, dati rezultat heširamo, i na kraju dobijenu heš vrednost upisujemo zajedno sa salt-om umesto lozinke.

¹⁴ [http://en.wikipedia.org/wiki/Salt_\(cryptography\)](http://en.wikipedia.org/wiki/Salt_(cryptography))



Slika 18: Primer funkcionisanja heš funkcije sa salt-om

Kod ovakvog pristupa imamo takođe problema. Prvi je vezan za samu prirodu heš funkcija. Naime, naveli smo da je jedan od uslova za savršenu heš funkciju to da ne postoje dve poruke čiji je heš isti, tj. da ne postoje dva heša koji su isti za jednu poruku. Neispunjavanje tih uslova dovodi do kolizije, i u realnosti, ona je prisutna kod mnogih heš algoritama, samo je pitanje koliko je lako te kolizije naći. Drugi problem je činjenica da u ovakvoj organizaciji, zapisujemo salt uz heš vrednost, tj. sam salt nije tajan. Takođe, i sa svim ovim primenjenim merama, one koriste prvenstveno da se uspori i spreči probijanje svih lozinka, dok ako je napadač fokusiran na probijanje samo jedne, onda je samo donekle usporen.

Recimo, napadač želi da preuzme kontrolu nad nalogom nekog igrača. Kako on to može učiniti? Jedan od načina je taj, da se napadač igraču predstavi kao ovlašćeno lice (najčešće slanjem e-maila), koje je zaduženo za razna pitanja, i u cilju poboljšavanja usluge i same igre, mora da proveriti neke stvari, i za to mu je potrebna lozinka igrača. Iako sve to zvuči sumnjivo, solidan broj ljudi je u stanju da poveruje i ostavi svoje podatke, na prvom mestu password, čime direktno ugrožava svoj nalog. U cilju sprečavanja takvih situacija, u okviru samih igara često među obaveštenjima stoje informacije tipa, da administratori igre nikada, ni u kakvim situacijama, neće od igrača tražiti informacije o njihovom nalogu.

Programi koji snimaju redosled pritisnutih tastera (eng. keyloggers) takođe predstavljaju značajnu opasnost. Samo zamislite da svaki ukucani password, ili bilo koji

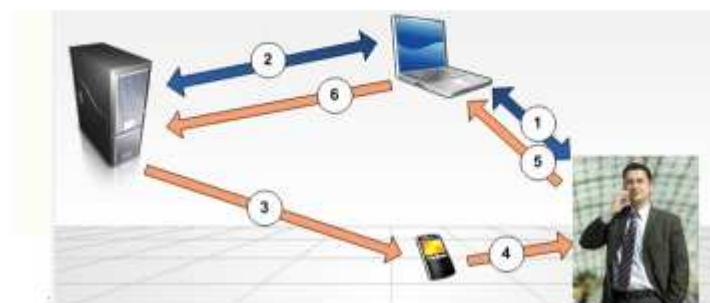
poverljivi podatak koji ste uneli, bude dostupan “na izvol’te” onoj strani koja je keylogger uspela da ubaci na vaš računar. Kompromitovanje naloga nikad lakše. Kao protivmera takvim programima, mogu se koristiti razni antivirus programi, ali uvek je pitanje kolika je njihova efikasnost. Takođe, kao preventiva, pomaže i “pametno” korišćenje Interneta (izbegavanje sajtova koji su rizični, ili sumnjivi). No, i pored svega, korišćenje passworda samo po sebi i dalje predstavlja jedan veliki sigurnosni rizik, zbog mana koji takav sistem zaštite ima po svojoj prirodi.

5.2 Rešenja za poboljšavanje procesa autentifikacije

5.2.1 Multifaktorska autentifikacija

Primena multifaktorske autentifikacije predstavlja korak koji se smatra logičnom evolucijom po pitanju sigurnosti. Glavni razlog za to predstavlja činjenica da su se 3 kategorije autentifikacije (autentifikacija na osnovu onoga što znamo, na osnovu onoga što jesmo, i na osnovu onoga što posedujemo) samostalno pokazale kao nedovoljno efikasne. Ako bi koristili grubi primer, to bi bilo kao poređenje vrata koje imaju jednu ili više tačaka zaključavanja, gde je, teoretski, za ona vrata sa više tačaka zaključavanja potrebno više vremena napadaču da ih otvori.

Multi Factor Authentication



Slika 19: Primer funkcionisanja multifaktorske autentifikacije

U suštini, ovakav pristup teoretski zvuči veoma dobro. Recimo da je napadač uspeo da dođe do podataka vezanih za korisnički nalog nekog igrača. Samim time, kod klasične autentifikacije, dovoljno bi mu bilo da ukuca otkrivene podatke, i time je završio veliki deo posla. Međutim, kod dvofaktorske autentifikacije, napadač nakon unošenih podataka, mora da uzme i da unese dodatne podatke, kao npr. jedinstveni PIN broj koji se

nalazi na kartici dobijenoj uz primerak igre, koja je u posedu igrača. Samim time, pošto napadač ne zna taj PIN broj, nalog i dalje nije kompromitovan. Imamo i nekoliko primena takve vrste autentifikacije.

Kompanija *Blizzard Entertainment*, čija igra *World of Warcraft* predstavlja najpopularniju online igru na svetu, je takođe veoma bila zainteresovana za razvoj rešenja koji poboljšavaju autentifikaciju, radi bolje sigurnosti. S obzirom na veliki mesečni obrt zahvaljujući pretplatama od velikog broja igrača, česti su slučajevi krađe naloga. U skladu sa time, smišljeno je nekoliko rešenja od kojih ćemo ovde predstaviti dva, koja u suštini dodaju još jedan sloj zaštite na postojeći proces autentifikacije.

Prvo rešenje je Blizzard autentifikator. Ovde se radi o ideji korišćenja dvofaktorske autentifikacije, gde se kombinuje ono što korisnik zna, i ono što poseduje. Konkretno, ovde igrač nabavlja uređaj (autentifikator), ili aplikaciju za mobilni, koju onda povezuje sa svojim nalogom. U procesu autentifikacije, pored korisničkog imena i lozinke, potrebno je uneti i odgovarajući autentifikacioni kod. Taj kod ima 8 cifara, i biva generisan od strane uređaja ili aplikacije, i sa njime je moguće u naredna 2 minuta logovati se na server, nakon čega se dobija novi kod. Ukoliko su svi potrebni podaci tačno uneti, autentifikacija je uspešna, u suprotnom, pristup se odbija.

Ovakav pristup je funkcionisao neko vreme dobro, ali onda su se desili problemi, i igrači koji su koristili autentifikator su prijavili kako su im nalozi kompromitovani. Kako se to desilo? Napadači su uspeli da prvo kompromituju klijent računare, i na njima instaliraju program koji im je omogućio da izvedu varijaciju „Man-in-the-Middle“ napada. Konkretno, to znači da je program preuzimao podatke koje su korisnici unosili, i umesto njih slao pogrešne podatke. To biouzrokovalo da korisnici vide poruku kako autentifikacija nije uspela. U međuvremenu, podaci koje su korisnici uneli bi bili prosleđeni računarima koji su kontrolisani od strane napadača. Čim prime podatke, napadači su onda u stanju da se prijave na sistem putem kompromitovanog naloga, i time dobiju pristup osetljivim informacijama.

Druga sigurnosna mera je uvođenje Dial-In autentifikatora. Ovde, umesto da se koristi poseban softver ili uređaj, postoji deo koda u okviru login dela aplikacije, koji u slučaju nekih nepravilnosti ili nelogičnosti zahteva dodatnu verifikaciju. Primer bi bio da je recimo igrač pokušao da se uloguje sa druge lokacije u odnosu na onu odakle se inače loguje. Zbog toga, login aplikacija šalje zahtev za unošenjem određenog koda koji igrač zna, i nakon uspešnog unosa, aplikacija šalje igraču određeni sigurnosni kod, koji on unosi da bi pristupio svom nalogu. U suštini, ovakav pristup je više zamena za one koji nisu u mogućnosti da koriste klasičan autentifikator, i manje je siguran od prethodnog rešenja.

Dakle, kao što ovde možemo videti, ovakav pristup je bolji u odnosu na klasična rešenja, ali i dalje ima slabosti koje je moguće iskoristiti. Takođe, za primenu multifaktorske autentifikacije, potrebna su značajna finansijska ulaganja, koja umeju biti prevelika za manje developere. Isto tako, moramo uzeti u obzir i hardversku zahtevnost. Na primer, ako koristimo određeni uređaj poput Blizzard autentifikatora, mogući je

scenario da pomenuti uređaj zahteva određene drajvere, koji ne rade na verziji operativnog sistema koji poseduje korisnik. Ako uz to korisnikov računar nije dovoljno jak za noviju verziju OS-a, dolazimo u situaciju da zbog jednog perifernog uređaja, igrač nije u mogućnosti da igra igru. Na kraju, tu je i činjenica da je za same korisnike često veoma iritirajuće kada moraju da provedu određeno vreme na autentifikaciji, što ume da dovede i do odustajanja od igre. Sve u svemu, iz navedenog možemo zaključiti da primena multifaktorske autentifikacije takođe nosi sa sobom i određene rizike, koje je potrebno imati u vidu.

5.2.2 Autentifikacija kao igra ili njen deo

Na par mesta smo već spomenuli jedan problem koji se pojavio još od početka primene sigurnosnih mera kod računara. U pitanju su korisnici i njihov stav prema sigurnosti. Često su upravo stav i viđenja nekoga zaduženog za očuvanje sigurnosti informacionog sistema u suprotnosti sa stavovima korisnika. Igrači žele što jednostavniji pristup, brz ulazak u igru i početak akcije, sa što manje sporednih stvari. U želji da to postignu, u stanju su, kao što smo već spomenuli, da izaberu proste i lake lozinke, koje se brzo iskucaju i lako pamte, jer to su kvaliteti koji su samim korisnicima bitni kod lozinki, pošto oni, u većini slučajeva ne shvataju šta se dešava u pozadini.

Generalno, korisnici će, u cilju bržeg i lakšeg izvršavanja zadataka koji su njima bitni, biti itekako voljni da čak i preskoče određene sigurnosne korake. Ako uzmemo u obzir da se neki od mehanizama sigurnosti kriju od korisnika, radi smanjivanja rizika od napada, to često dovodi do njihovog nerazumevanja, i čak i do verovanja da ako im neko kompromituje nalog, oni sami ne smatraju to nekim posebnim rizikom i problemom.

Iz ovoga možemo videti, da je veoma bitno, ako ne i kritično, da se korisnici motivišu da promene svoje navike i način razmišljanja. To je moguće postići raznim načinima, kao što je dodatna edukacija korisnika. Jedan od načina je i primena autentifikacije kao igre. Ali kako je to moguće uraditi? Da li je i koliko tako nešto korisno? Zašto bi to bilo uopšte efikasno? Kao odgovor na to, prvo bi trebalo da objasnimo šta to motiviše ljude da uopšte igraju igre.

Igranje igara u velikom broju slučajeva ume da deluje kao rad na poslu. U suštini, igrači provode vreme u okviru igre, ulažu određeni trud, i dobijaju odgovarajuće nagrade kao rezultat za sve to. Isto tako, zaposleni idu na posao, ispunjavaju svoje radne zadatke, sve do kraja radnog vremena. Ali, šta je to što razlikuje pomenute slučajeve? Kada pogledamo rad na poslu, uglavnom je motivacija za rad materijalna sigurnost, želja za napredovanjem, itd. Međutim, tu imamo i određene rizike i probleme, kao što su mogućnost da ostanemo bez posla uprkos svom trudu, da nam plata bude smanjena, da ne dobijemo unapređenje, i dosta drugih mogućnosti. Sve to dovodi do toga, da nam sam rad na poslu često ume biti jedno dosta nezahvalno i nemotivišuće iskustvo, u odnosu na uloženo vreme i trud.

Autentifikacija za većinu korisnika takođe predstavlja dodatni rad vezan za posao, dok je kod igrača to kao jedna dodatna obaveza u moru drugih. Najčešća pomisao kod jednog korisnika prilikom autentifikacije je: „Već imam puno toga da radim, a ovde su sada našli još i sa ovim da me maltretiraju, kao da ja svo vreme ne radim ništa i nemam čime drugim da se bavim.“. Porazna je činjenica da zahvaljujući ovakvom razmišljanju, većina korisnika vidi proces autentifikacije kao problem koji treba što pre završiti, pa čak i izbeći, radi smanjenja sopstvenog opterećenja. Ako je uz to proces autentifikacije složeniji za samog korisnika, kao korišćenje dvofaktorske autentifikacije, vidimo koliko nezadovoljstvo može nastati zbog korišćenja takvog sistema.

Kod video igara je situacija drugačija. Iako imaju dosta zadataka u okviru sebe koje treba ispuniti, iako je za sve to potrebno puno vremena, video igre su često dizajnirane tako da motivišu korisnike koji ih igraju na najrazličitije moguće načine. Nastavak razvoja priče, mogućnost osvajanja virtuelnih trofeja, dobijanje raznih bonusa u okviru igre, to su sve samo neke od varijanti, zahvaljujući kojim igrači nastavljaju igranje određenog naslova. Dizajn igara podrazumeva i što češće slanje povratnih informacija, tako da je sam igrač svakog momenta svestan kako i koliko napreduje, za razliku od realnosti. Zahvaljujući svemu tome, igrači često imaju osećaj produktivnosti i napretka dok igraju igre, i to postaje jedno veoma ispunjavajuće iskustvo i jedan veoma veliki motivator, veći čak i od zarađivanja para. Ovo posebno važi ako na to dodamo da su pozitivna osećanja kod igrača (sreća, zadovoljstvo, ponos, oduševljenje, itd.) uglavnom mnogo češća nego negativna (tuga, krivica, osramoćenost, nezadovoljstvo, itd.).

Uvidevši potencijal, mnoge firme su se isprobale u razvoju takozvanih „edutainment“ igara. Ideja je bila iskombinovati sve prednosti koje igranje igara pruža zajedno sa različitim edukativnim elementima, kao što su učenje matematike, jezika i sl. Kao rezultat imamo naslove gde je recimo potrebno naučiti i rešiti nekoliko matematičkih problema određenog tipa, da bi dobili mogućnost rešavanja drugih tipova zadataka. Dalji nastavak takvih ideja predstavlja pojam „gamification“, koji podrazumeva korišćenje ideja i mehanika prisutnih u video igrama na drugim procesima.

Primena autentifikacije kod video igara je prisutna i pre online igara. Ideja da se proces autentifikacije integriše kao deo igre, nastala je još u periodu 80-ih godina, kada su korišćene kasete i diskete za skladištenje podataka. Usled veoma lakog kopiranja podataka sa takvih skladišnih jedinica, potrebno je bilo naći način da se utvrdi da li je igrač neko ko je legitimno nabavio igru. Kao rešenje za taj problem, došlo se na ideju da se koriste priručnici koji su se dobijali uz naslov. Pre početka same igre, igraču bi se postavljala pitanja tj. „challenge“ kao što su: „Koja je 5. reč u 4. rečenici 3. pasusa na drugoj stranici priručnika?“. Kasnije, developeri su došli na ideju da umesto pitanja na početku, igrač kasnije u okviru igre mora da konsultuje priručnik, radi nalaženja odgovarajuće šifre sa kojom može da nastavi dalje da igra.

SECTION 1									
	A	B	C	D	E	F	G	H	I
1	==	+	==	==	+	==	==	==	+
2	==	==	+	==	+	==	==	==	==
3	==	==	==	+	==	==	==	==	==
4	+	==	==	==	==	==	==	==	+
5	==	+	==	==	==	==	==	==	==
6	==	==	==	==	==	==	==	==	==
7	==	==	==	==	==	==	==	==	==
8	==	==	==	==	==	==	==	==	==
9	==	==	==	==	==	==	==	==	==
10	==	==	==	==	==	==	==	==	==
11	==	==	==	==	==	==	==	==	==
12	==	==	==	==	==	==	==	==	==
13	==	==	==	==	==	==	==	==	==
14	==	==	==	==	==	==	==	==	==
15	==	==	==	==	==	==	==	==	==
16	==	==	==	==	==	==	==	==	==
17	==	==	==	==	==	==	==	==	==
18	==	==	==	==	==	==	==	==	==
19	==	==	==	==	==	==	==	==	==
20	==	==	==	==	==	==	==	==	==
21	==	==	==	==	==	==	==	==	==
22	==	==	==	==	==	==	==	==	==
23	==	==	==	==	==	==	==	==	==
24	==	==	==	==	==	==	==	==	==
25	==	==	==	==	==	==	==	==	==
26	==	==	==	==	==	==	==	==	==
27	==	==	==	==	==	==	==	==	==
28	==	==	==	==	==	==	==	==	==
29	==	==	==	==	==	==	==	==	==
30	==	==	==	==	==	==	==	==	==
31	==	==	==	==	==	==	==	==	==
32	==	==	==	==	==	==	==	==	==
33	==	==	==	==	==	==	==	==	==
34	==	==	==	==	==	==	==	==	==
35	==	==	==	==	==	==	==	==	==
36	==	==	==	==	==	==	==	==	==
37	==	==	==	==	==	==	==	==	==

Slika 20: Stranica sa sigurnosnim kodovima iz priručnika igre „Maniac Mansion”

U početku se takav pristup pokazao dobrim za suzbijanje piraterije, i autentifikaciju vlasnika originala, ali kasnije su iskrslili problemi. Prvi problem je taj, što je sam pristup bio naporan za igrače. Drugi problem je bila činjenica, da su pitanja i priručnici bili identični za sve kopije igara, tako da je neko mogao da nađe kopiju knjige, i da prepíše odgovore za sva pitanja na papir.

Naprednija varijanta ovog pristupa je primenjena kod igre Ni No Kuni, za Nintendo DS džepnu konzolu. Jedan od bitnih elemenata u okviru same igre je korišćenje magija, koje se aktiviraju crtanjem odgovarajućih simbola na DS-ovom donjem ekranu,

koji je osjetljiv na dodir (touch screen). Uz svaki primerak igre, dobijala se i knjiga koja sadrži prikaz svih simbola koje je moguće koristiti u igri. Ono što je interesantno, to je da su za određene magije, koje se moraju obavezno upotrebiti, simboli koje je potrebno iscrtati drugačiji za svaku kopiju igre, i samim time, i svaka knjiga je drugačija. Na ovaj način su igrači i naterani da kupe original verzije igre.



Slika 21: Primer crtanja znakova u igri Ni No Kuni

Upravo na ovom principu, i kod online igara možemo napraviti i autentifikaciju kao igru, tj. deo igre. Na primer, u okviru igre, igrači kontrolišu svog avatara, koga su kreirali nakon registracije na server. Avatar predstavlja grafički prikaz igrača, ili njegov alter-ego, u virtuelnom svetu.¹⁵ Proces kreiranja avatara je dosta značajan za samog igrača, jer je u pitanju lik koga će najčešće gledati, i koji je njegov/njen predstavnik, pomoću koga mogu još lakše da se užive u igru. Sve to dovodi do toga da se stvara i jedna vrsta emotivne povezanosti.

Upravo sve to možemo iskoristiti kod autentifikacije. Recimo da neke podatke kod avatara možemo putem sistema da slučajno odredimo, kao npr. u okviru fiktivnog istorijata avatara, može da se navede ime psa koga je imao u detinjstvu, i onda se ime tog psa koristi kao odgovor na jedno od pitanja prilikom autentifikacije. Sa stanovišta napadača, podatak vezan sa fiktivni istorijat avatara nema nikakvog značaja niti smisla, ali ima za samog igrača, upravo zbog pre pomenute emotivne povezanosti. Sam napadač će pre pokušati da kao lozinku isproba ime igračevog psa iz detinjstva, ali kako je ovde ime slučajno generisano od strane sistema, teško da će se ta dva podatka poklopiti.

Loša strana ovakvog pristupa je ta što svi ti sami podaci mogu lako biti zaboravljeni od samog korisnika. Način na koji to možemo sprečiti je češće spominjanje tih podataka u okviru same igre, recimo kao elemenata priče, čime postižemo da se poveća povezanost između igrača i njegovog avatara, i samim time, olakša pamćenje istih. Ipak, čak i onda, moguće je recimo da challenge pitanja budu vezana za nešto što korisnik jednostavno ne može da upamti. Kao rešenje, možemo ponuditi mogućnost da korisnik izabere nekoliko kategorija vezanih za avatara (istorijat, karakter, itd) na osnovu kojih će mu biti postavljana pitanja. No u svakom slučaju, cilj je da autentifikacija bude integralni deo same igre, čime igrači neće imati toliko problema sa procesom, za razliku od situacija kada je autentifikacija predstavljena kao odvojeni proces.

Ovakav pristup može pomoći čak i kod bilo kojih drugih informacionih sistema, gde bi npr. mogli da koristimo pravila neke društvene igre poput šaha. Lozinka bi se onda, umesto kucanjem, unosila pomeranjem figura po šahovskim pravilima, tako da svako polje gde korisnik postavi određenu figuru predstavlja jedan simbol u okviru lozinke. Postoje razne druge mogućnosti, ali uvek treba imati u vidu da dati proces autentifikacije treba da bude istovremeno efikasan i siguran, ali i zanimljiv i jednostavan za same korisnike.

Samim time, nakon ostvarenog pozitivnog utiska sa procesom autentifikacije, korisnike je lakše edukovati po pitanju drugih principa zaštite, koji podrazumevaju čuvanje lozinke na siguran način, izbegavanje korišćenja istih login podataka na drugim mestima, itd. Sve to predstavlja još jedan korak napred ka sigurnijem i boljem iskustvu.

¹⁵ [http://en.wikipedia.org/wiki/Avatar\(computing\)](http://en.wikipedia.org/wiki/Avatar(computing))

6. Slučaj Diablo III

Jedna od igara koja je unela veliku pometnju među igračima širom sveta je *Diablo III*. Da bi bolje razumeli koliki je uticaj u pitanju, trebalo bi malo ispričati istorijat kompanije i serijala.

Kompanija Blizzard Entertainment je osnovana 1991 godine pod imenom Silicon & Synapse. Razvojem video igara počeli su se baviti 1993, a među prvim igrama koje su napravili bili su *Rock n' Roll Racing* i *The Lost Vikings*. Godine 1994., kompanija je promenila ime u Blizzard Entertainment. Igra kojom je kompanija stekla svetsku popularnost je *Warcraft: Orcs & Humans*. Nastavljajući sa uspesima, Blizzard je napravio nekoliko veoma popularnih igara, među kojima su i nastavci *Warcraft-a*, zatim *StarCraft*, *Diablo* serijal, kao i MMORPG *World of Warcraft*. U pitanju su naslovi koji su postavili određene standarde, zatim imali prodaju merenu milionima primeraka, kao i ogroman broj igrača širom sveta.

Igre iz *Diablo* serijala su takođe potpadale pod datu kategoriju. Prvi deo je izašao 1996. godine. U okviru igre, vi birate jednu od tri klase likova, i onda sa datim likom se borite protiv velikog broja kompjuterski vođenih neprijatelja, dok na kraju ne dođete do borbe protiv završnog protivnika, demona po imenu Diablo. U toku igre nailazite na razne predmete koje možete iskoristiti u borbi. Uprkos relativnoj jednostavnosti, *Diablo* je stekao veliku popularnost, toliku da je godinu dana kasnije izašao dodatak *Hellfire*, koji je omogućio izbor dodatne klase, kao i nove zadatke i predele koji su se mogli istraživati.

Reklo bi se da je prvi deo ostvario veliki uspeh koji je teško nadmašiti, međutim, izlazak *Diablo II* 2000. godine je pokazao suprotno. Formula je ostala ista, ali, dodato je još opcija: više klasa likova, dosta novih veština koji su likovi mogli upotrebiti, veliki broj novih predmeta, kao i dosta veći broj tipova neprijatelja. No, glavna stvar koja je drugi deo, i njegovu nadogradnju *Diablo II: Lord of Destruction* učinila veoma popularnim je prisustvo multiplayer opcija, zahvaljujući kojima zabavi nije bilo kraja. Zamislite samo mogućnosti gde vi počnete igru sa izabranim likom u single - player varijanti. Nakon nekog vremena, čujete se sa svojim prijateljem, koji takođe igra *Diablo II* sa svojim likom, i onda on donese kod vas svoj računar koji putem lokalne mreže poveže sa vašim, samim time omogućavajući da igrate zajedno putem LAN multiplayer opcije. No, drugi put, vaš prijatelj nije u mogućnosti da dođe kod vas, tako da onda imate opciju da putem TCP / IP konekcije ostvarite vezu i nastavite da igrate zajedno. I naravno, imate i opciju konektovanja na battle.net servis, putem kojeg možete igrati sa dosta drugih igrača širom sveta, sve ovo sa jednim istim likom koga ste kreirali na početku. Pokazalo se da je ovakav pristup bio pun pogodak, i da je zahvaljujući tome popularnost igre bila nemerljivo veća od očekivane.

Naravno, ovakav pristup je imao i određenih sigurnosnih mana, koje nisu baš bile zanemarljive, no i pored njih, fanovi su željno iščekivali sledeći nastavak. Delovalo je kao da ništa ne može poći po zlu.



Slika 22: Diablo II meni

Spomenuli smo da je *Diablo II* imao određene sigurnosne probleme. Glavni problem je predstavljala činjenica da je bilo potrebno sve bitne podatke smestiti na korisničkoj strani, a već smo spominjali koliko je to opasno. Naravno, igrači su tu okolnost obilato koristili, battle.net je bio preplavljen likovima čije su osobine bile blago rečeno neverovatne. To se nije svidelo čelnicima Blizzard-a, jer su smatrali da se time smanjuje užitak ostalim igračima, a takođe je tu bio i problem piraterije.

Kako se bližio datum izlaska *Diabla III*, polako je otkrivano sve više informacija vezanih za igru. Slike iz beta verzija igre, hardverska zahtevnost, kao i dodatne informacije, među kojima je bila i ta, da će igra zahtevati konstantnu internet konekciju. Komentari su bili razni, od odobravajućih do kritičkih, ali to i dalje nije pomutilo optimizam ljudi uverenih da će sve proći u najboljem mogućem redu.

Diablo III je izašao 15. maja 2012. godine. Komercijalno igra je bila ogroman uspeh, gde je u toku prva 24 sata prodato 3,5 miliona kopija, a do kraja prve nedelje se ta cifra popela na 6,3 miliona. Recenzije su takođe bile veoma pozitivne, i generalno, sve je delovalo savršeno. No, to nije trajalo dugo, jer su problemi počeli da se pojavljuju.

Developeri *Diablo III* su imali pred sobom imali određene ciljeve, među kojima su prioriteta bili i na koji način sprečiti pirateriju i varanje u okviru igre, dva najveća problema koje su imali sa prethodnikom. Iako je *Diablo II* komercijalno i finansijski prošao odlično, čelnici Blizzard-a su bili uvereni da bi sve bilo još bolje da igrači nisu bili u prilici da lako koriste ilegalne kopije igre. Takođe, problem varanja je bio veoma izražen, upravo zahvaljujući tome da su se skoro sve bitne informacije vezane za igru nalazile kod igrača. Samim tim, to je borbu protiv takve pojave učinilo maltene sizifovskim poslom, jer su sa svakom sigurnosnom zakrpom, igrači našli neki novi način da modifikuju igru i nastave sa varanjem.

Na kraju, developeri su došli do jednog rešenja za oba problema: prebacivanje igre u format koji bi zahtevao konstantnu internet konekciju (eng. online - only). To bi onda podrazumevalo prelazak svih kritičnih podataka na stranu servera. Time se automatski drastično otežava mogućnost varanja, a takođe je potrebno da igrači kupe original da bi mogli da igraju, jer ne postoji mogućnost offline igranja. Dakle, primenjen je sistem zaštite o kome smo govorili kod varanja putem neosnovanog poverenja.

Naravno, mane takvog pristupa su osetne za same korisnike. Kao prvo, stoji činjenica da i u današnje vreme, postoji veliki broj mesta na svetu gde pristup Internetu nije moguć. Još je češći slučaj da, iako ima tehničkih mogućnosti, igrač ne može da se konektuje na Internet, usled npr. problema vezanih za njegovog provajdera, ili bilo kojih drugih problema. Sve to stvara dodatne frustracije kod samih igrača, što naravno nije dobro. Ekvivalentno tome bi bilo recimo da imate novi Office 20XX paket, i na računaru vam se nalazi fajl u Word 20XX formatu koga želite da otvorite, jer vam je hitno potrebno da nešto odatle pročitate. Nažalost, pošto se u tom trenutku nalazite na lokaciji gde nemate pristup Internetu, vama Word 20XX ne daje da otvorite fajl, jer ne može da uradi validaciju i potvrdi vaš identitet.

Drugi problem predstavlja poređenje sa prethodnikom. *Diablo II* je, kako smo već napomenuli, davao igračima veliki broj opcija po pitanju načina igranja. Čak i u slučaju kada niste konektovani na Internet, možete barem da igrate u single-player varijanti, tako da vam igra i dalje može biti zabavna. *Diablo III*, sa druge strane, zahteva da se konektujete na server, makar vi želeli da isključivo igrate sami. Takođe, igranje putem lokalne mreže (LAN) ne postoji, iako je takav način igranja takođe bio popularan među igračima. Generalno, u očima nekih ljudi, nastavak je delovao kao osiromašena verzija prethodnika, i kao korak unazad po pitanju mogućnosti.

Uprkos svemu navedenom, vidimo da je popularnost igre i dalje bila na veoma zavidnom nivou, kao i da je velika većina igrača bila voljna da prihvati sve promene. Nažalost, onda je nastupio problem koji je imao mnogo veće posledice, koji igrači nisu mogli da ignorišu, niti da se pomire sa time. Zamislite ovako, da ste recimo nakon dugog napornog rada došli kući, i željni opuštanja, pokrenuli ste *Diablo III*, igru koju ste (ako ste uzeli novu) platili oko 60 dolara, koju ste čekali dugi niz godina da izadje, i zbog koje ste čak uzeli i brži internet paket. Sve kod vas radi kako treba, na login ekranu ste ukucali vaše podatke, ali onda vam se pojavilo jedno obaveštenje.



Slika 23: Error 37 – noćna mora svakog Diablo III igrača

Ovde bih opet upotrebio primer sa imaginarnim Office 20XX paketom, gde vi iako imate i internet konekciju i original Office, opet ne možete da otvorite fajl, jer ne može sistem da vas identifikuje, usled velikog broja zahteva za validaciju, a sam fajl vam hitno treba. Frustriranost, bes, očajanje, sva ta osećanja bi neko imao u takvoj situaciji. Isto tako su se osećali i svi *Diablo III* igrači pri susretu sa ovom porukom, koja se nažalost često pojavljivala kod velikog broja korisnika. Situacija je bila toliko loša da su se u zemljama poput Južne Koreje, Nemačke i Francuske igrači organizovano žalili kompaniji Blizzard, i tražili povraćaj novca.¹⁶

Ceo ovaj slučaj predstavlja primer sprovođenja metoda i tipa zaštite koji je doveo do toga da korisnici ispaštaju. Sistem se sa aspekta zaštite generalno pokazao dobrim, i definitivno je bio efikasniji u odnosu na *Diablo II*, međutim, problemi koji su nastali njegovim sprovođenjem doveli su do gubitka poverenja kod klijenata, u toj meri da su mnogi igrači koji su do juče bili ljubitelji kompanije rešili da više nikada ne kupe igre od kompanije Blizzard. Upravo iz tih razloga je potrebno uvek, prilikom uvođenja metoda i sistema zaštite uzeti u obzir same korisnike i njihove potrebe.

¹⁶http://en.wikipedia.org/wiki/Diablo_III#Controversies

7. Zaključak

Izazovi vezani za sigurnost informacionih sistema su veliki i brojni. Uz napredak tehnologije napreduju i načini na koji se sistemi mogu kompromitovati, i svako rešenje povlači sa sobom nove probleme i pitanja. Kod online igara rizici se povećavaju upravo sa činjenicom da veliki broj napadača često ima dodatne motive izuzev finansijskog, što treba uvek imati u vidu.

Greške u radu su takođe uvek prisutne, tako da je teško govoriti o savršenom sistemu i savršenoj odbrani, posebno kada se radi o online igrama. Propusta će uvek biti, kao i novih i inovativnih načina napada. Korišćenjem metoda koje su ovde navedene, šteta se može smanjiti na neki prihvatljivi nivo.

Autentifikaciju smo predstavili kao posebnu celinu zbog njene specifičnosti kao sigurnosne celine gde sami korisnici imaju aktivnu ulogu. Upravo ta činjenica je ono što sam proces čini veoma interesantnim, kao i posebno izazovnim za rešavanje, i u ovome radu je predložen jedan pristup kako se sami korisnici mogu motivisati da postupak autentifikacije odrade sa svoje strane na način koji je potreban.

Menadžment resursa takođe predstavlja jedan veliki izazov koji stoji pred onima koji razvijaju online igre. Ovde takođe imamo veliki broj faktora na koje trebamo obratiti pažnju, kao i na njihov međusobni odnos i povezanost.

Naravno, treba voditi računa i o korisnicima i njihovom zadovoljstvu. Ako uvođenje nekih sigurnosnih metoda i ideja čini igru manje privlačnom, da li onda вреди uopšte uvoditi ih? Takođe, postoji i pitanje resursa. Da li stvarno treba prelaziti na drugačiju arhitekturu, ako će takva implementacija dovesti nezadovoljstva igrača, uprkos svojoj efikasnosti?

Ovakva pitanja su česta kod zaštite online igara, i nemoguće je naći odgovor koji je uvek tačan. Samim time, preostaje da se nađe balans između svih mogućnosti, i doneti rešenja u skladu sa svakim slučajem pojedinačno. Upravo to čini oblast resursa i zaštite online igara izazovnom i dinamičnom, kao i interesantnom za proučavanje.

Literatura

- [1].Hu, J., Zambetta, F., “*Security issues in massive online games*”, SECURITY AND COMMUNICATION NETWORKS, Security Comm. Networks. 2008; 1:83–92, Published online in Wiley InterScience(www.interscience.wiley.com) DOI: 10.1002/sec.5, http://goanna.cs.rmit.edu.au/~jiankun/Sample_Publication/Gaming.pdf
- [2].Van Summeren, R., “*Security in online gaming*”, Radboud University Nijmegen, January 26, 2011,
- [3].Hayes, J.E., “*Playing it Safe: Avoiding Online Gaming Risks*”, <http://www.us-cert.gov/sites/default/files/publications/gaming.pdf>
- [4].Yan, J.J., Hyun-Jin, C., “*Security Issues in Online Games*”, <http://homepages.cs.ncl.ac.uk/jeff.yan/TEL.pdf>
- [5].Yan, J., “*Security Design in Online Games*”, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, http://homepages.cs.ncl.ac.uk/jeff.yan/yan_acsac03.pdf
- [6].Kirmse, A., Kirmse, C. “*Security in online games*”, July 7, 1997, http://www.gamasutra.com/view/feature/3212/security_in_online_games.php
- [7].McGraw, G., Hoglund, G., “*Online Games and Security*”, <http://www.cigital.com/papers/download/attack-trends-EOG.pdf>
- [8].Wesley, M.E., “*Defenses Against TCP SYN Flooding Attacks*”, The Internet Protocol Journal - Volume 9, Number 4, http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-4/syn_flooding_attacks.html
- [9].McClure, S., Scambray, J., Kurtz, G., **Hakerske tajne – zaštita mrežnih sistema**, Mikro knjiga, Beograd, 2006
- [10].Lu, F., “*Solving Key Design Issues for Massively Multiplayer Online Games on Peer-to-Peer Architectures*”, May 2009, School of Mathematical and Computer Sciences, Heriot-Watt University
- [11].Nae, V. *, Iosup, A. †, Podlipnig, S. *, Prodan, R. *, Epema, D. †, Fahringer, T. *, “*Efficient Management of Data Center Resources for Massively Multiplayer Online Games*”, * Institute for Computer Science, University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria., †Parallel and Distributed Systems Group, Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands.
- [12].Yahyavi, A., Kemme, B., “*Peer-to-Peer Architectures for Massively Multiplayer Online Games: A Survey*”, McGill University
- [13].Neumann, C. —————, Prigent, N. † , Varvello, M. ‡ , Suh, K. § , “*Challenges in Peer-to-Peer Gaming*”, —————Thomson Research Lab, Paris, France, †Thomson Security Research Lab, Rennes, France, ‡Eurecom, SophiaAntipolis, France, §University of Massachusetts, Amherst, US

-
- [14].Zaimont, M., “Skullgirls, GGPO... and You”,
<http://skullgirls.com/2011/09/skullgirls-ggpo-and-you/>
- [15].Yamane, R.S., “Secure online game play with Token: A case study in the design of multi-factor authentication device”, Aoyama Gakuin University eLPCO, Shibuya 4-4-25, Tokyo, 1508366 JAPAN
- [16].Sandeep, K.P., “Chess Game As A Tool For Authentication Scheme”, Department of Information Technology, Institute of Technology and Management, Gorakhpur (U.P), India
- [17].Micallef, N. *, Just, M. †, “Using Avatars for Improved Authentication with Challenge Questions”, * School of Informatics, University of Edinburgh Edinburgh, UK, † School of Engineering & Computing, Glasgow Caledonian University, Glasgow, UK