



UNIVERZITET SINGIDUNUM

**-MASTER STUDIJSKI PROGRAM-
SAVREMENE INFORMACIONE TEHNOLOGIJE**

Implementacija i zaštita chat servisa baziranog na XMPP protokolu

- Master rad -

**Mentor:
Prof. dr. Mladen Veinović**

**Student:
Đorđe Došić
Br. indeksa: 410270/2014**



Beograd, 2015. godine

SADRŽAJ

1. UVOD.....	4
2. INSTANT MESSAGING.....	5
2.1. Osnovni pojam.....	5
2.2. Kratak istorijat razvoja.....	6
2.3. Osnovna IM komunikacija.....	7
2.4. IM komunikacioni modeli.....	8
3. XMPP (Extensible Messaging and Presence Protocol).....	10
3.1. Osnovni pojmovi.....	10
3.2. Kratak istorijat.....	10
3.3. Karakteristike protokola.....	11
3.3.1. Prednosti.....	11
3.3.2. Nedostaci.....	12
3.3.3. Podržani servisi.....	12
3.3.4. Aplikacije za implementiranje.....	14
3.4. Arhitektura.....	15
3.5. Funkcionisanje XMPP sa DNS protokolom.....	19
3.6. Adresiranje.....	21
3.6.1. Domeni.....	21
3.6.2. Korisnici.....	22
3.6.3. Resursi.....	22
3.6.4. Internacionalizacija.....	22
3.6.5. XMPP URI.....	23
3.7. Strimovanje XML-a.....	23
3.8. Primitive u komunikaciji.....	24
3.8.1. Poruka.....	24
3.8.2. Prisutnost.....	25
3.8.3. IQ.....	26
3.8.4. Proširivost.....	28
4. PRETNJE I RIZICI.....	29
4.1. Opšte pretnje.....	29
4.1.1. Nesigurne (otvorene) konekcije.....	29
4.1.2. Denial of Service (DoS) napad.....	30
4.1.3. Imitiranje.....	30
4.1.4. Lažiranje DNS i IM servera.....	30
4.2. Pretnje iz IM funkcionalnosti.....	30
4.2.1. Razmena fajlova.....	31
4.2.2. Registri u formi običnog teksta i arhiviranje poruka.....	31
4.2.3. Nesigurna osnovna (default) podešavanja.....	31
4.2.4. Deljenje IM funkcionalnosti sa drugim aplikacijama.....	32
4.2.5. Zlonamerni hiperlinkovi.....	32
4.2.6. URI (Universal Resource Identifier) moduli.....	32
4.3. Crvi.....	33

4.3.1. Ranjivosti klijentskih aplikacija	33
4.3.2. Primeri IM crva	34
4.4. Analiza potencijalnih XMPP ranjivosti.....	34
5. Mehanizmi zaštite XMPP protokola.....	36
5.1. Upostavljanje XMPP strima.....	36
5.2. Opcije autentifikacije (SASL).....	38
5.3. Enkripcija konekcije	39
5.4. BOSH	41
5.4.1. Funkcionisanje BOSH.....	42
5.5. Slanje poruka bez servera.....	47
6. OPENFIRE	50
6.1. Instalacija	50
6.2. Povezivanje eksterne baze podataka sa serverom	51
6.2.1. MySQL.....	51
6.2.2. Oracle	52
6.2.3. SQL Server.....	52
6.2.4. PostgreSQL.....	53
6.3. Integracija baze korisnika.....	53
6.3.1. Podešavanja konekcije sa bazom podataka	53
6.3.2. Integracija autentifikacije.....	54
6.3.3. Integracija korisnika.....	55
6.3.4. Grupna integracija	56
6.4. LDAP	57
6.4.1. Konfiguracija	57
6.4.2. Rad sa aktivnim direktorijumom	58
6.4.3. Manualna izmena konfiguracionog fajla.....	59
6.5. SSL instrukcije.....	62
7. ZAKLJUČAK	66
LITERATURA.....	67

1. UVOD

Ljudska potreba za povezivanjem i socijalizacijom sa drugim ljudima je rezultirala u vidu velike mreže koju danas znamo kao Internet.

Internet je svetski sistem umreženih računarskih mreža koji je transformisao način na koji funkcionišu komunikacioni sistemi. Počeci Interneta se vežu za stvaranje ARPANET-a, 1969. godine, mreže računara pod kontrolom Ministarstva odbrane SAD. Danas, Internet povezuje milijarde računara širom sveta na jedan nehijerarhijski način. Međutim, Internet nije samo proizvod tehnološkog napretka, nego takođe društvenih i političkih procesa. Od svojih korena kao jedno neindustrijsko i neposlovno okruženje vezano za naučnu zajednicu, Internet se vrlo brzo proširio u svet trgovine i poslovanja. Ipak, bilo je potrebno oko 30 godina da se nametne kao tehnološka inovacija koja konstantno menja društvo i ekonomiju.

Pojam Interneta predstavlja mrežu unutar mreže, ili interkonekciju između više računara. Strukturalno postoje male mreže koje se međusobno vezuju, i time čine ovu strukturu. Internet se sve više naziva globalnom mrežom informacija (velika internacionalna-globalna baza podataka). Broj uređaja na Internetu se trenutno procenjuje na nekoliko pa i desetina milijardi, s obzirom na to da je sve učestala upotreba bežičnih uređaja kao što su tableti i pametni mobilni telefoni. Količina informacija koja cirkuliše svetskom mrežom je ogromna, i teško je realno proceniti i prikazati kolika je ona zaista.

Razvoj svetske Internet mreže je u velikoj meri doprineo i evoluciji komuniciranja putem informacionih tehnologija. Elektronska pošta (e-mail) je do skorijeg vremena bio najrasprostranjeniji i najusvojeniji oblik komunikacije preko Interneta. Pre otprilike dve decenije, malo ljudi je znalo da uopšte i postoji. Danas mnogi koriste servis elektronske pošte, umesto dopisivanja papirnim pismima kao tradicionalnim načinom. Ljudi širom sveta razmene preko milijardu elektronskih poruka svakodnevno.

Nekada e-mail servis jednostavno nije dovoljno brz. Ne znamo ni to da li je osoba s kojom se dopisujemo trenutno prisutna, da odgovori u najkraćem mogućem vremenu. Takođe, razmena elektronske pošte u smislu običnog razgovora nije baš praktična zbog suvišnih koraka prilikom slanja i prijema poruka. Tu na scenu stupa **instant messaging** (IM) servis kao sredstvo brze i trenutne komunikacije preko računarske mreže i postaje veoma popularan za relativno kratko vreme. U okvirima IM servisa, moguće je kreiranje imenika lica s kojim se komunicira, kao i pregled trenutne (ne)dostupnosti za komunikaciju. Dopisivanje je prilično olakšano jer korisnik skoro momentalno vidi poruku koju je poslala druga strana i obratno.

2. INSTANT MESSAGING

2.1. Osnovni pojam

Instant messaging (IM) je jedan oblik servisa koji nudi prenos poruka u realnom vremenu (real-time), preko računarskih mreža i Interneta. U tu svrhu se koriste klijentske programske aplikacije (messenger). Kratke poruke se tipično prenose u oba smera između dve strane, u trenutku kada korisnik osmisli poruku i pritisne opciju za slanje. Pojedine IM aplikacije koriste tehnologiju kojom se obezbeđuje slanje i prikaz teksta u realnom vremenu, karakter po karakter, u trenutku kucanja poruke.

Većina IM programa ima razne napredne mogućnosti kao:

- **Instant messages** - trenutno slanje i prijem poruka sa prisutnim korisnicima;
- **Chat rooms** – kreiranje zajedničkog chat prostora gde više korisnika može istovremeno da se dopisuje;
- **Web links** - postavljanje linkova za razne veb sajtove;
- **Video** - slanje i pregled video sadržaja, razgovor sa direktnim video prenosom lica s kojim se komunicira;
- **Images** - pregled, slanje i prijem slika;
- **Sounds** – razmena i pregled audio sadržaja;
- **Files** – razmena fajlova između korisnika;
- **Talk** – moguće korišćenje Interneta kao posrednog medijuma za razgovor umesto klasične telefonije;
- **Streaming content** - prikaz raznog multimedijalnog sadržaja, vesti i sl. u realnom vremenu;
- **Mobile capabilities** - slanje brzih poruka preko mobilnih telefona.

Kao što je prethodno navedeno, IM obuhvata multicast prenos, u vidu takozvanih "chat soba", gde učesnici mogu biti anonimni ili već poznati od ranije (npr. saradnici koji rade na nekom projektu koriste chat u komunikaciji). Današnji IM sistemi teže ka uspostavljanju konekcija između već poznatih korisnika (često koristeći kontakt liste tipa "buddy list" ili "friend list").

U zavisnosti od IM protokola, arhitektura servisa u tehničnom smislu može da bude peer-to-peer (direktno tačka-tačka prenos) ili klijent-server (gde server rutira poruke među korisnicima).

IM predstavlja u suštini skup komunikacionih tehnologija korišćenih u tekst-baziranoj komunikaciji između dva ili više učesnika preko Interneta ili drugih tipova računarskih mreža. IM–chat se dešava u realnom vremenu, zbog toga je neophodno napomenuti da se online chat i instant messaging razlikuju od drugih tehnologija kao što je e-mail zbog prividne sinhronizacije korisnika u komunikaciji. Pojedini sistemi čak dozvoljavaju da poruke budu poslate korisnicima koji nisu prisutni (offline messages), umanjujući razlike između IM i e-mail servisa. IM omogućava trenutnu i efikasniju komunikaciju, dozvoljavajući trenutni prijem potvrde ili odgovora.

U većini slučajeva, IM obuhvata dodatne opcije kao npr. to da korisnici mogu da se vide međusobno preko web-kamera, ili da razgovaraju besplatno preko Interneta koristeći mikrofoni i slušalice ili zvučnike. Mnoge aplikacije dopuštaju transfer fajlova, mada su ograničene na određenu maksimalnu veličinu fajla. Moguće je sačuvati tekst sa konverzacijom korisnika radi eventualne upotrebe. Brze poruke se često pamte u lokalnoj bazi, otprilike kao u programima elektronske pošte.

2.2. Kratak istorijat razvoja

Iako sam pojam potiče još iz devedestih godina prošlog veka, IM prethodi Internetu, pojavljujući se prvi put na višekorisničkim operativnim sistemima Compatible Time-Sharing System (CTSS) i Multiplexed Information and Computing Service (Multics), sredinom šezdesetih godina prošlog veka [1]. Inicijalno, neki od ovih sistema su bili u upotrebi signalizacije i obaveštavanja servisa za štampanje, ali ubrzo su iskorišćeni za usklađivanje komunikacije sa drugim korisnicima koji su ulogovani na istoj mašini. Paralelno sa razvojem mreža, implementirali su se i protokoli na tim istim mrežama. Neki od njih su se koristili kao P2P (peer-to-peer) protokoli (talk, ntalk i ytalk), dok je za druge bila neophodna konekcija sa serverom (talker i IRC).

The Zephyr Notification Service je napravljen u okviru projekta Atena, Instituta tehnologije Masačusets osamdesetih godina da bi omogućio lociranje i slanje poruka korisnicima od strane provajdera servisa. Pored navedenih sistema, jedan od tada postojećih online chat kapaciteta je bio Talkomatic (1973) u okviru sistema PLATON. Fenomen elektronske oglasne table (BBS) koji je doživeo vrhunac osamdesetih godina, predstavlja još jedan primer gde su pojedini sistemi ugradili chat funkcionalnosti koje su bile približne IM; Freelancin' Roundtable je glavni primer.

Prvi dostupni online chat servis komercijalnog tipa (nasuprot PLATON-a koji je služio u edukativne svrhe) je bio CompuServe CB Simulator iz 1980 [2]. Raniji IM programi su uglavnom bili tekst u realnom vremenu, gde su se karakteri pojavljivali u trenutku kucanja istih. Ovo uključuje Unix "talk" command line program, koji je bio popularan osamdesetih i početkom devedesetih. Neki BBS chat programi (Celerity BBS) su koristili sličan interfejs. Moderne implementacije teksta u realnom vremenu takođe postoje u IM aplikacijama, kao što je AOL Real-Time IM kao opcionalna funkcionalnost.

Kompanija Quantum Link je u ponudi imala servis za Commodore64 računare, koji je omogućavao direktne poruke između konektovanih kupaca, sa nazivom "On-Line Messages" (OLM), a kasnije "FlashMail". Quantum Link je postao America Online, da bi potom napravio AOL Instant Messenger (AIM) [3]. Dok je klijentski softver Q-Link radio na Commodore64, koristeći samo Commodore PETSCII¹ tekstualnu grafiku, ekran je bio vizuelno podeljen u sekcije i OLM bi se pojavili kao žuta traka gde je pisalo "Message from:" i ime pošiljaoca zajedno sa porukom na vrhu, i predstavljenom listom opcija.

Moderniji softverski klijenti za razmenu poruka, grafičkog interfejsa, koji su danas poznati, pojavili su se prvi put sredinom devedesetih godina: PowWow, ICQ i AOL Instant Messenger. U novembru 1996. godine, firma Mirabilis je predstavila čuveni ICQ ("I seek

¹ PET Standard Code of Information Interchange

you"). Nekoliko godina kasnije, AOL je preuzeo firmu Mirabilis, autore ICQ-a. U međuvremenu, druge kompanije su razvijale sopstveni softver (Excite, MSN, Ubiq i Yahoo!), svaka sa svojim vlasničkim protokolima i klijentskim programima.

1998. godine, kompanija IBM je predstavila IBM Lotus Sametime, proizvod zasnovan na tehnologiji stečenoj kada je IBM kupio Haifa-baziran Ubiq i Lexington-baziran Databeam.

2000. godine, pokrenuta je aplikacija otvorenog koda i protokol otvorenog standarda koji se zove Jabber. Protokol je standardizovan pod nazivom Extensible Messaging and Presence Protocol (XMPP). XMPP serveri su mogli funkcionisati kao kapija za ostale IM protokole, smanjujući potrebu za korišćenje više različitih klijenata. IBM Lotus Sametime je novembra 2007. godine dodao podršku IBM Lotus Sametime Gateway za XMPP.

Od 2010. godine, provajderi društvenih mreža često nude IM mogućnosti. Facebook Chat je jedan oblik brzih poruka, i Twitter može da se posmatra kao sistem za brze poruke. Mnogi IM servisi nude video pozive, VoIP i web konferencije. Web konferencijski servis može integrisati video pozive i brze poruke. Pojedine IM kompanije takođe nude deljenje desktopa, IP radio i IPTV u sklopu funkcionalnosti za govor i video.

2.3. Osnovna IM komunikacija

U narednim koracima je opisan postupak osnovnog funkcionisanja i komunikacije korisnika preko IM:

1. Korisnik pristupa određenoj web stranici IM provajdera i preuzima kopiju besplatnog klijentskog softvera za svoj računar.
2. Instalira preuzetu aplikaciju na računar.
3. Prilikom pokretanja, klijent pokušava da se konektuje na server i pritom koristi vlasnički protokol datog provajdera za komunikaciju.
4. Kada se klijent poveže sa serverom, korisnik može uneti svoje ime i lozinku da se prijavi na serveru. Ukoliko nema korisnik nalog, radi se prijava za nalog, i odmah nakon toga može da se koristi. Tada server proverava kreirano ime i lozinku, i samim tim se korisnik prijavljuje.
5. U trenutku uspostavljanja veze, klijent šalje serveru informaciju za povezivanje (IP adresu i broj TCP porta) od računara koji koristi. Klijentski program takođe prikazuje korisniku sačuvanu kontakt listu.
6. Server pravi privremenu datoteku sa informacijama za povezivanje i listu kontakata. Tada i proverava da li je ko od korisnika na listi kontakata trenutno prijavljen.
7. Ako server pronade da je neko kontakata prijavljen, onda šalje poruku nazad klijentu, sa informacijama o prisutnosti tih korisnika. Istovremeno šalje informaciju o prisutnosti datog korisnika drugim prijavljenim korisnicima sa kontakt liste.
8. Kada klijent dobije informacije o prijavljenim korisnicima u listi kontakata, onda menja status tih korisnika u "online". Klikom na korisnika sa liste kontakata koji je na mreži, otvoriće se prozor u koji može da se unese tekst u polje.
9. S obzirom da klijent ima podatak o IP adresi i broju porta za računar osobe s kojom komunicira, poruka se šalje direktno klijentu na računar te osobe. Drugim rečima,

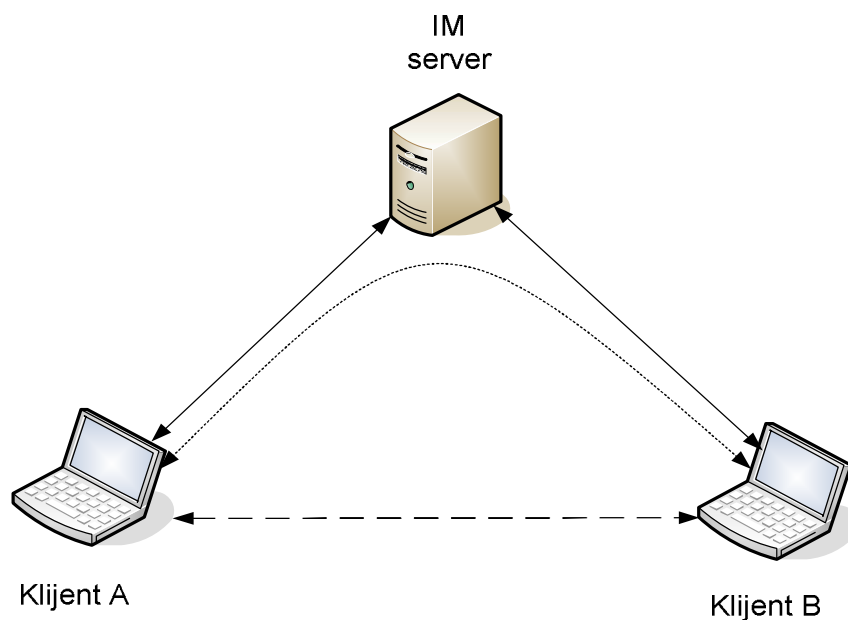
- server nije direktno uključen u ovom trenutku. Sva komunikacija je neposredna između dva klijenta (naravno, u zavisnosti od primenjene tehnologije).
10. Kada drugi korisnik dobije poslatu poruku, onda može da odgovori. U trenutku razmene poruka oba korisnika, te iste poruke se prikazuju istovremeno (ili približno) na oba računara.
 11. Nakon završetka razmene poruka, u trenutku odjavljivanja korisnika, šalje se poruka serveru za prekidanje sesije. Server odjavljuje korisnika i šalje poruku o statusu svim klijentima sa liste kontakata koji su trenutno na mreži. Na kraju, server briše privremeni fajl koji je sadržavao podatke za povezivanje datog klijenta.

2.4. IM komunikacioni modeli

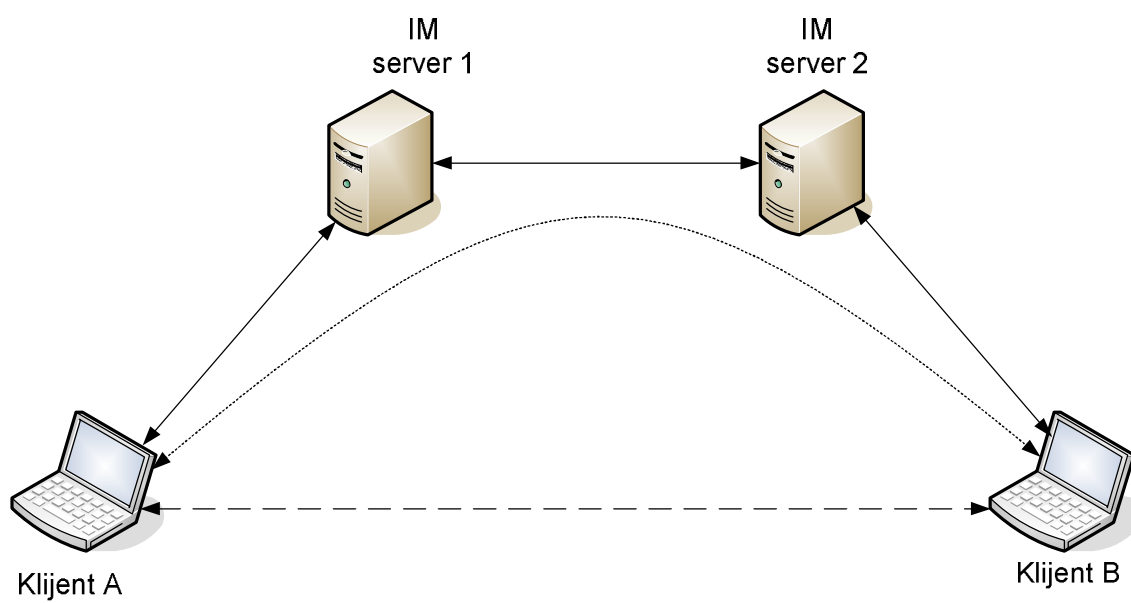
Iako se za klijenta, IM server pojavljuje kao jedini posredni entitet preko kojeg komunicira sa drugim korisnikom, to može da bude i grupa servera, s kojim upravlja isti provajder IM servisa (npr. AOL) ili više nezavisnih provajdera. Postoje dva osnovna IM modela komunikacije [4]:

- Centralizovani IM serverski model (prikazan na Slici 1.),
- Distriburani IM serverski model (prikazan na Slici 2.).

Kada korisnik A želi da se dopisuje sa korisnikom B, oba moraju da se prijave na isti IM servis. Poruke od A do B će server da prosledi u zavisnosti od podešavanja privatnosti korisnika B. Za direktnu komunikaciju između korisnika A i B, server može da pruži neophodne informacije (destinaciona IP adresa) za obe strane.



Slika 1.



Slika 2.

3. XMPP (Extensible Messaging and Presence Protocol)

3.1. Osnovni pojmovi

XMPP je komunikacioni protokol baziran na XML-u (Extensible Markup Language). Protokol se ranije nazivao Jabber, kreiran je 1999. godine od strane Jabber open-source zajednice za potrebe brzih poruka (IM), prikazivanja informacija, održavanja listi kontakata u približno realnom vremenu[5]. Dizajniran da bude fleksibilan sa mogućnostima proširivanja, protokol je takođe korišćen za sisteme za objavu i dopis, signalizaciju VoIP, video sadržaj, fajl transfer, video igre, Internet of Things (IoT)² aplikaciju kao pametna mreža, i servise društvenih mreža. Za razliku od većine IM protokola, XMPP je definisan kao otvoreni standard i koristi pristup otvorenih sistema razvoja i aplikacija, po kojem bilo ko može implementirati XMPP servise i povezivati sa rešenjima drugih organizacija. S obzirom da je XMPP otvoren protokol, implementacije se mogu razvijati korišćenjem bilo koje softverske licence. Iako se mnoge serverske, klijentske implementacije i biblioteke distribuiraju besplatno i kao open-source softver, brojne implementacije komercijalnog softvera i freeware-a takođe postoje.

IETF (The Internet Engineering Task Force) organizacija je formirala XMPP radnu grupu 2002. godine radi formalizacije esencijalnih protokola kao IETF tehnologija brzih poruka i dostupnosti[6]. XMPP radna grupa je sastavila četiri specifikacije (RFC 3920³, RFC 3921, RFC 3922, RFC 3923), koje su priznate kao Predloženi Standardi u 2004. godini. 2011. godine, RFC 3920 i RFC 3921 zamenjene sa RFC 6120 i RFC 6121 respektivno, sa RFC 6122 specificirajući XMPP adresni format. Kao dodatak ovim osnovnim protokolima standardizovanim u IETF, XMPP Standards Foundation (prethodno poznata kao Jabber Software Foundation) je aktivna u razvoju otvorenih XMPP ekstenzija. XMPP-bazirani softver je distribuiran po celom Internetu, i od 2003. godine, je u široj upotrebi od strane desetina miliona ljudi u svetu, po podacima XMPP Standards Foundation.

3.2. Kratak istorijat

Džeremi Miler, programer iz Ajove, je počeo sa radom na Jabber tehnologiji 1998. godine da bi pokrenuo prvu verziju jabberd servera u januaru 1999. godine. Nedugo posle toga se pojavljuje, Jabber programerska zajednica koja se fokusirala na open-source softver, pre svega na jabberd server (verzija 1.0 u maju 2000, verzija 1.2 u oktobru 2000, i verzija 1.4 u februaru 2001. godine), ali kao glavni proizvod se pokazao razvoj XMPP protokola. Raniji Jabber protokol, razvijan 1999. i 2000. godine, je formirao osnovu XMPP protokola koji je objavljen u RFC 3920 i RFC 3921 (glavne promene tokom formalizovanja od strane IETF

² IoT – sveobuhvatna mreža fizičkih objekata povezana elektronikom, softverom, senzorima...

³ RFC – Request for Comments

XMPP radne grupe su dodali TLS za enkripciju kanala i SASL za autentifikaciju) [7]. RFC 3920 i RFC 3921 zamenjene sa RFC 6120 i RFC 6121, i to je objavljeno 2011. godine.

Prvi IM servis zasnovan na XMPP protokolu je Jabber.org, koji je funkcionisao od 1999. godine i nudio je besplatne naloge za korisnike XMPP protokola. U periodu od 1999. do 2006. godine, servis je koristio jabberd kao serverski softver, da bi potom migrirao u ejabberd (oba su bila besplatni softverski aplikacioni serveri) tokom istog perioda. Nešto kasnije, 2010. godine, servis je postao komercijalni M-Link serverski softver kojeg je proizveo Isode Ltd.

Avgusta 2005. godine, Google je predstavio Google Talk, kombinaciju VoIP i IM sistema koji koristi XMPP za brze poruke i kao osnovu za protokol signalizacije prenosa glasa i fajlova, koji se zove Jingle. Prvobitna verzija nije obuhvatala serverske međukomunikacije, ali je Google ubacio i tu funkcionalnost u januaru 2006. godine. Od tada je Google implementirao i podršku za video prenos u Google Talk, koristeći Jingle protokol za signalizaciju. Maja 2013, Google je objavio odbacivanje kompatibilnosti sa Jabber-om iz Google Talk-a, konkretno za server-server komunikaciju, mada je zadržana podrška za klijent-server. Google je takođe odbacio XMPP podršku za Google Voice u maju 2014. godine.

AOL je predstavio XMPP podršku ekperimentalnog tipa za svoj AOL Instant Messenger (AIM) servis, omogućavajući AIM korisnicima da komuniciraju koristeći standardizovan, open-source XMPP. U martu 2008. godine, rad sa ovim servisom je prekinut, a u maju 2011. godine AOL nudi ograničenu XMPP podršku. Septembra 2008. godine, Cisco Systems je kupio Jabber Inc., kreator komercijalnog proizvoda Jabber XCP.

Društvena mreža Facebook, u 2010. godini je otvorila svoju chat opciju za nezavisne aplikacije preko XMPP protokola. Facebook razvojni tim je objavio da Facebook Chat nema XMPP servere u svojoj infrastrukturi, nego da samo prezentuje XMPP interfejs klijentima; kao posledica toga, neke promene serverskog podešavanja ne mogu da se rade preko XMPP.

Microsoft je izbacio XMPP interfejs 2011. godine za svoj (sada van funkcije) Microsoft Messenger servis. Skype, kao naslednik navedenom servisu, takođe pruža ograničenu XMPP podršku. Međutim, ovo nisu izvorne XMPP implementacije.

3.3. Karakteristike protokola

3.3.1. Prednosti

- **Decentralizacija.**

Arhitektura XMPP umrežavanja je slična arhitekturi elektronske pošte; svako je u mogućnosti da uspostavi sopstvenu XMPP serversku infrastrukturu bez potrebe glavnog centralnog servera.

- **Otvoreni standardi.**

IETF je usvojio i formalizovao XMPP kao priznatu tehnologiju brzih poruka i dostupnosti pod nazivom XMPP (poslednje specifikacije su RFC 6120, 6121 i 6122).

Za razvoj i implementaciju podrške ovih specifikacija nije neophodna posebna privilegija, tj. nema direktne povezanosti sa proizvođačima.

- **Istorija.**

XMPP tehnologije su u upotrebi još od 1999. godine. Višestruke implementacije XMPP standarda postoje za klijentske, serverske komponente i biblioteke kodova.

- **Bezbednost.**

XMPP servere možemo izdvojiti iz javne XMPP mreže (npr. interna chat mreža kompanije na zatvorenom LAN-u), implementirani su solidni bezbednosni protokoli (za autentifikaciju i enkripciju, SASL i TLS) u jezgro XMPP specifikacije.

- **Fleksibilnost.**

Proizvoljne funkcionalnosti se mogu nadograditi na XMPP; u cilju održanja interoperabilnosti, XMPP Standards Foundation je odgovoran za zajedničke ekstenzije i dodatke. XMPP aplikacije van klasičnog IM (Instant messaging) obuhvata grupni chat, upravljanje mrežom, spajanje sadržaja, zajedničke alate, deljenje fajlova (file sharing), gejming, daljinsku sistemsku kontrolu i nadzor, geolokaciju, posredni softver (middleware) i računarstvo u oblacima (cloud computing), VoIP i servis za identitete.

3.3.2. Nedostaci

- Nema optimizovan QoS (Quality of Service).
Protok podataka XMPP protokola uglavnom čini oko 70% na serveru a od toga oko 60% se ponavlja.
- Komunikacija bazirana na tekstu XML formata.
S obzirom da XML zapisi imaju poseban format, XMPP protokol ima malo veći mrežni paket prilikom slanja poruka na adrese više primalaca. Ovaj problem je pomenut u probnom dokumentu XEP-0322⁴.
- S obzirom na dužinu jednog XML fajla kojim je kodiran XMPP protokol, ne može da se obezbedi modifikacija binarnih podataka. Sa tim, upotreba eksternog protokola za transfer fajlova kao HTTP je neminovna.
- Nema mogućnosti end-to-end enkripcije.

3.3.3. Podržani servisi

Servisi u kontekstu samog protokola predstavljaju razne opcije i funkcionalnosti koje se koriste u aplikacijama. XMPP implementacije tipično podržavaju ključne servise[8]:

- **Channel encryption**

Ovaj servis, omogućava enkripciju konekcije između klijenta i servera, ili između dva servera. Enkripcija u ovom slučaju predstavlja važan bezbednosni segment aplikacija.

- **Authentication**

Dati servis predstavlja još jedan nivo bezbednosti aplikacije. Autentifikacija omogućava stranama koje pokušavaju da komuniciraju preko mreže, da se prijave na server, tj. autentifikuju. Time je server stavljen u ulogu kontrolne tačke.

⁴ XEP – XMPP Extension Protocol

- **Presence**

Omogućava saznanje o dostupnosti drugih entiteta na mreži. Jednostavnije rečeno, servis dostupnosti odgovara na pitanje, "Da li je korisnik prisutan, dostupan za komunikaciju ili nedostupan (offline)?"⁵. Podaci o dostupnosti mogu da sadrže i dodatne informacije (npr. da li je korisnik na sastanku). Naravno, razmena informacija prisutnosti se zasniva na eksplicitnom pristanku između dva entiteta kako bi se zaštitila privatnost korisničkih informacija.

- **Contact lists**

Servis omogućava kreiranje i skladištenje listi kontakata, ili tzv. roster, na XMPP serveru. Najuoobičajenija upotreba ovog servisa je kao IM "lista prijatelja", i bilo koji entitet koji ima nalog na serveru može da koristi servis radi održavanja liste poznatih ili poverenih entiteta.

- **One-to-one messaging**

Ovaj servis omogućava slanje poruke jednom entitetu. Klasična upotreba razmene poruka jedan na jedan predstavlja personalni ili lični IM, ali poruke mogu biti proizvoljan XML, bilo koja dva entiteta na mreži mogu da razmene poruke – botovi⁵, serveri, komponente, uređaji, XMPP-uključeni web servisi, ili neki drugi XMPP entitet.

- **Multi-party messaging**

Ovaj servis je definisan u XEP-0045[9], njime je dozvoljen pristup virtuelnoj chat sobi radi razmene poruka između više učesnika, nešto slično Internet Relay Chat (IRC). Poruke mogu da budu običan tekst, ili da sadrže XML ekstenzije za naprednije funkcionalnosti, kao što je podešavanje sobe, sobno glasanje, i druge kontrolne poruke u sesiji.

- **Notifications**

Servis je definisan u XEP-0060[10], služi za generisanje i dostavljanje obaveštenja za više korisnika (pretplatnika). Postoji sličnost sa višeučesničkim prenosom poruka, ali je optimizovan za jedan-prema-svima slanje sa eksplicitnim dopisom na posebne kanale ili teme.

- **Service discovery**

Ovaj servis je definisan u XEP-0030[11] i omogućava pretragu funkcionalnosti koje podržava drugi entitet, kao i neki dodatnih entiteta koji su povezani sa tim (npr. sobe hostovane od strane chat room servisa).

- **Capabilities advertisement**

Ovaj servis je definisan sa XEP-0115[12] i predstavlja ekstenziju na servis prisutnosti (presence) koji pruža obaveštenje "iz prve ruke" za podatke pronadjenih servisa koji su podržani od drugih entiteta, radi evidentiranja u keš memoriju aplikacije.

- **Structured data forms**

Servis definisan u XEP-0004[13], omogućava razmenu struktuirane ali i fleksibilne forme sa drugim entitetima, slično HTML formi. Često se koristi za konfiguraciju i druge zadatke gde je neophodno prikupiti ad-hoc informaciju od drugih entiteta.

- **Workflow management**

Servis je definisan u XEP-0050[14], obezbeđuje učešće u interakciji strukturnih radnih procesa sa drugim entitetom, sa podrškom za uobičajene akcije radnog procesa, kao

⁵ Bot – program koji služi za automatizovane procese ili zadatke.

npr. kretanje u sledeću fazu poslovnog procesa ili izvršavanja naredbe. Često se koristi u kombinaciji sa formama podataka (data forms).

- **Peer-to-peer media sessions**

Ovaj servis je definisan u XEP-0166[15] i omogućava iniciranje i upravljanje medijskih sesija sa drugim entitetom. Takva sesija može da se koristi u svrhu audio i video chat-a, prenosa fajlova i drugih interakcija u realnom vremenu.

Ovo su samo neki od osnovnih servisa koji su dostupni korisniku (ili aplikaciji) kao učesniku XMPP mreže.

3.3.4. Aplikacije za implementiranje

S obzirom da imamo mnogo osnovnih servisa na raspolaganju, postoje razne mogućnosti za implementaciju IM aplikacija:

- **Instant messaging**

Predstavlja klasične IM sisteme koje većina ljudi poznaje, sa kombinacijom tri osnovna servisa: prisutnost (dostupnost), liste kontakata, direktno slanje poruka (P2P). Takvi sistemi mogu, pa i uključuju više drugih servisa i funkcija, ali samo sa ova tri servisa se može napraviti kostur IM aplikacije.

- **Groupchat**

Predstavlja višeučesnički servis za slanje poruka koji omogućava kreiranje groupchat sistema sličnim IRC. Često, groupchat sistemi su korišćeni za specifične aplikacije, kao sisteme za trgovinu u realnom vremenu u finansijskoj industriji, situacione prostorije za brzo reagovanje i vojsku, virtuelne učionice.

- **Gaming**

U kombinaciji sa posebnim ekstenzijama, i P2P i višeučesničko slanje poruka pružaju mogućnost pravljenja jednostavnih sistema za igre. Programeri za razvoj igara koriste XMPP radi dodavanja funkcionalnosti kao što su prisutnost i kontakt liste u već postojeće višekorisničke igre.

- **Systems control**

Kombinovanjem P2P slanjem poruka i posebnih obrazaca podataka može se dobiti manji sistem za kontrolu i interakciju sa udaljenim sistemima. U ovakve aplikacije spadaju upravljanje mrežom, naučna telemetrija, upravljanje mašinama.

- **Geolocation**

XMPP servis za obaveštavanje generalno može biti sa ili bez dodatnih podataka u informacijama koje se prosleđuju. Primer jednog takvog oblika za obaveštavanje je geolociranje, koje omogućava pravljenje zanimljivih aplikacija za lociranje, kao što je praćenje vozila.

- **Middleware and cloud computing**

Određen broj kompanija i istraživačkih grupa aktivno rade na XMPP-baziranim sistemima servisa za računanje, lakog posrednog softvera (middleware), i upravljanje cloud računarskih infrastruktura. Iako je upotreba XMPP protokola u ovom slučaju možda iznenađujuća, jer se takve tehnologije za slanje poruka oslanjaju uglavnom na zahtevnije platforme, XMPP se polako probija na manjem tržištu. Neke kompanije koje već poseduju XMPP infrastrukturu su shvatile da njena je upotreba moguća i u ne-IM

slučajevima. Takvi sistemi obično koriste radne procese za strukturnu razmenu poruka. Specifične aplikacije tog tipa uključuju bioinformatiku.

- **Data syndication**

Popularne aplikacije društvenih mreža u sve većem broju koriste XMPP servis obaveštavanja da bi rešili određen problem koji imaju: stalna potražnja za ažurnim informacijama. Postojeće HTTP-bazirane distribucije nisu skalabilne, jer često, određen odgovor ili povratna informacija se ne promeni od poslednjeg potraživanja. Suprotno tome, XMPP servis obaveštavanja šalje podatak radi ažuriranja samo kada dodje do promene, optimizujući u velikoj meri propusni opseg i serverske resurse koji bi inače bili uzalud potrošeni.

- **Voice over IP (VoIP)**

Google Talk aplikacija koja je lansirana 2005. godine, prva je koja je popularizovala upotrebu XMPP protokola za voice chat. Od tada, XMPP dodaci za medijske sesijske servise (zване Jingle) su formalizovani u XSF, implementirani i distribuirani od strane tadašnje firme Nokia i projekta "Jedan laptop po detetu". Sami dodaci mogu da se koriste za sinhronizaciju širokog opsega tipova medijskih sesija, uključujući video, transfer fajlova, whiteboarding-a, i kolaborativnog editovanja.

- **Identity services**

S obzirom na postojanje stalnih identifikatora (JabberID) kompleksnog servisa za autentifikaciju, moguća je upotreba XMPP protokola za formiranje servisa za identitete i autorizaciju kao što su OpenID i OAuth.

Ostali primeri aplikacija koje nismo naveli uključuju prenos podataka, chat uživo - integrisan u veb sajtove, komunikacije mobilnih uređaja, i direktorijuma za aktivnu dostupnost.

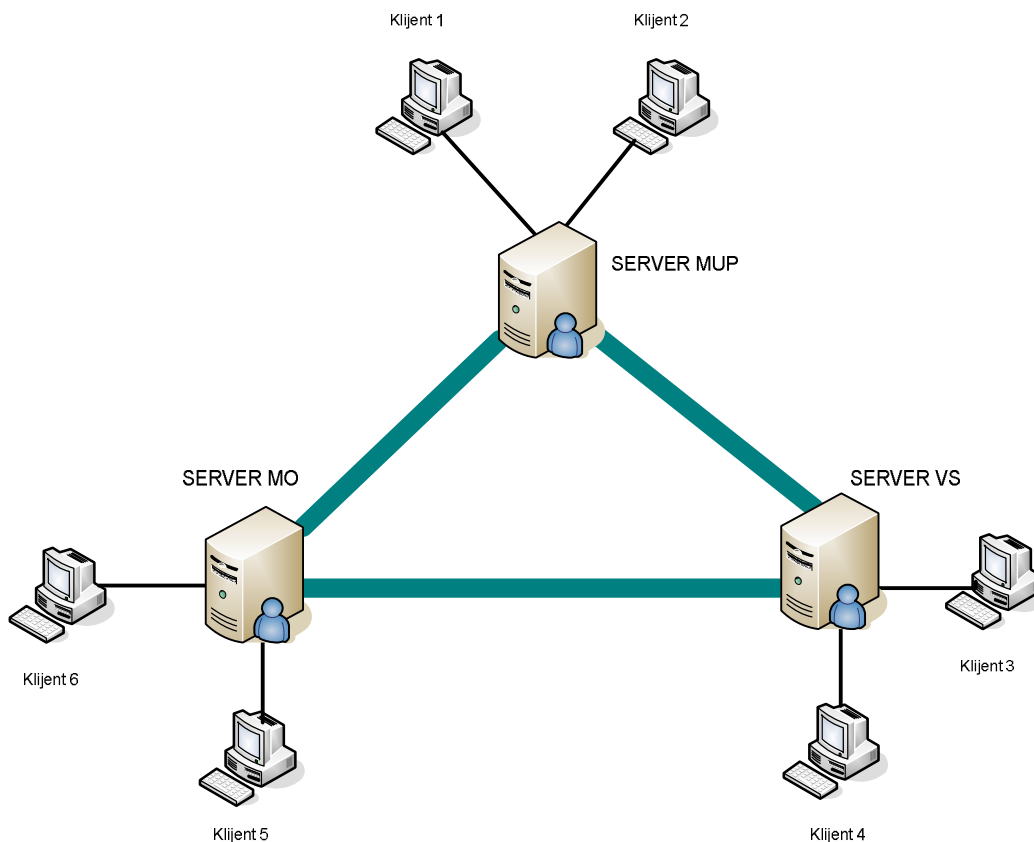
3.4. Arhitektura

Infrastruktura za brze poruke, prisutnost, i druge oblike komunikacije u realnom vremenu, uglavnom se sastoji od stotina hiljada Jabber servera na kojima rade programi kao ejabberd i Openfire, i miliona Jabber klijenata na kojima su programi kao Adium, Spark, Gajim, Pidgin, ili Psi, gde svi oni koriste standardni protokol XMPP.

XMPP tehnologije koriste decentralizovanu klijent-server arhitekturu, sličnu arhitekturama korišćenim za World Wide Web i mrežu elektronske pošte. Šema prikazana na Slici 3. je uprošćena prezentacija koja pokazuje tri servera, i po dva klijenta za svaki server.

Prednost upotrebe decentralizovane klijent-server arhitekture je u podeli segmenata sistema (razvojni deo klijentskih aplikacija može da se fokusira na korisničko iskustvo, a deo za servere na pouzdanost i proširivost), mnogo je olakšano vođenje P2P tehnologije od strane organizacija, sistem je robustan i nema jednu tačku prekida⁶ (svako može da koristi sopstvene XMPP polise kao što je autentifikacija korisnika, enkripcija kanala, i prevencija lažiranja adrese).

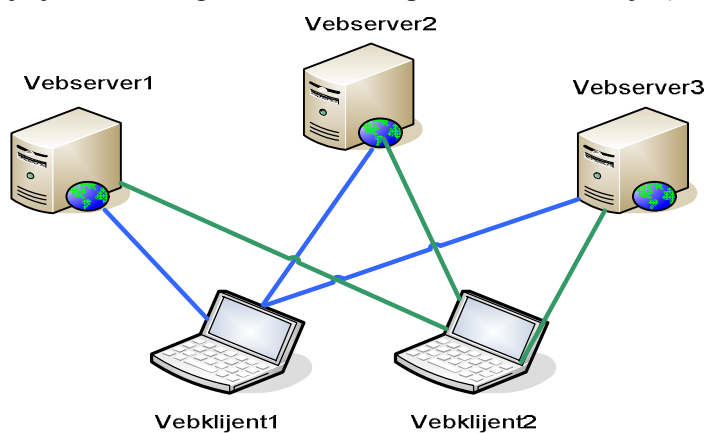
⁶ Single point of failure



Slika 3.

Između ostalog, XMPP programerska zajednica je dosada radila na tome da klijentske aplikacije budu jednostavne, a da se sva kompleksnost svede na servere, pritom dozvoljavajući šire usvajanje same tehnologije.

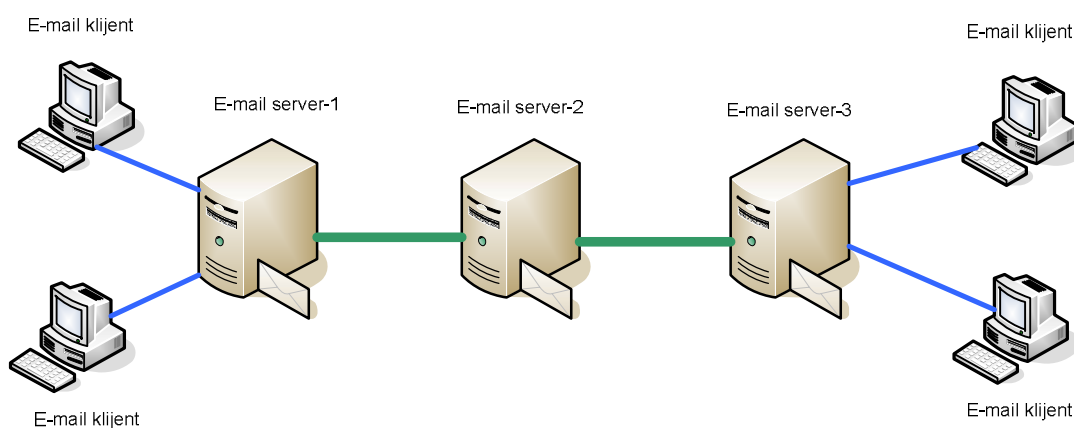
Ipak, postoje neke bitne razlike u arhitekturi između Web-a, email-a, i Jabber-a. Prilikom posete nekom veb sajtu, pretraživač (pregledač) se konektuje na vebserver, ali se vebserveri obično ne povezuju jedan sa drugim da bi se kompletirala transakcija (Slika 4.).



Slika 4.

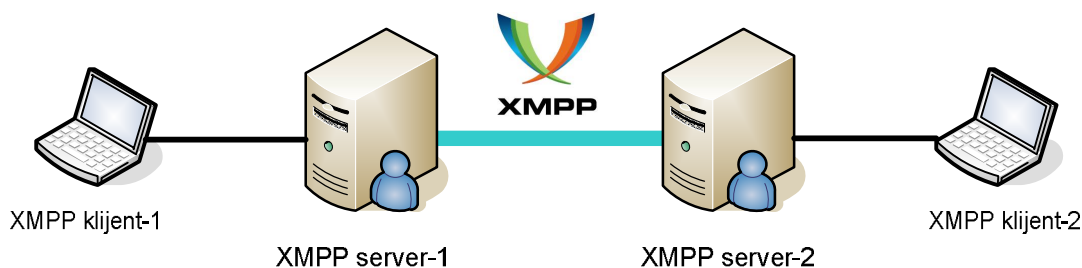
Umesto toga, HTML web strana može da daje reference na druge vebservere (npr. učitavanje slika ili skripti), tako da pretraživač otvara sesije sa pomenutim vebserverima da bi učitao punu stranicu. Tako da Web tipično ne uključuje među-domenske konekcije (često nazivanim federacijama).

Kada šaljemo elektronsku poštu nekom korisniku sa drugog domena, e-mail klijentska aplikacija se povezuje sa podešenim e-mail serverom, koji dalje trasira put za poruku namenjenom datom kontaktu. Za razliku od Web-a, e-mail sistem se sastoji od združene (federalizovane) mreže servera. Naravno, poslata poruka može biti rutirana kroz višestruke posredne e-mail servere pre nego što stigne na predviđenu destinaciju (Slika 5.). Tako da e-mail mreža koristi više tačaka između servera za dostavljanje poruka.



Slika 5.

Slično e-mail sistemu, a za razliku od Web-a, XMPP sistemi uključuju veliki broj među-domenskih konekcija. Međutim, prilikom slanja XMPP poruke nekom korisniku drugog domena, klijent se prvo konektuje na lokalni server, koji dalje uspostavlja direktnu vezu sa serverom tog korisnika bez posrednih tačaka (Slika 6.).



Slika 6.

Ovaj model direktne federacije ima bitne prednosti u odnosu na model indirektne federacije, korišćenog u elektronskoj pošti (konkretno, može da doprinese sprečavanju lažiranja adrese i neke oblike spamova).

S obzirom da XMPP opisuje klijent-server arhitekturu na kojoj je XMPP komunikacija bazirana, jedan od aspekata takve komunikacije je prethodno pomenuta federacija, gde postoji mogućnost da XMPP serveri različitih domena razmene XML zapise u vidu poruka. Postoje najmanje četiri nivoa federacije[16]:

1. **Permissive Federation** - server prihvata konekciju od bilo kog hosta na mreži, bez utvrđivanja identiteta hosta zasnovanog na DNS pretragama. Nedostatak verifikacije ili autentifikacije hostova može prouzrokovati da se domeni lažiraju.
2. **Verified Federation** - server prihvata konekciju samo nakon što se identitet hosta verifikuje (slab tip) putem XEP-0220 (Server Dialback) protokola. Server Dialback koristi DNS kao osnovu za utvrđivanje identiteta; osnovni pristup je da kada server-primaoac prihvata konekciju tipa server-server od drugog servera, ne obrađuje saobraćaj preko te veze sve dok ne utvrdi ključ sa autoritativnog servera za domen datog od servera koji inicira konekciju. Iako Server Dialback ne pruža snažnu potvrdu identiteta ili pouzdanu federaciju i iako to može biti predmet DNS poisoning napada, u velikoj meri je efektivno sprečio da se većina adresa sa raznih instanci lažira putem XMPP mreže, od početka njenog razvoja u 2000. godini[17].
3. **Encrypted Federation** - server prihvata konekciju sa hosta samo ako taj host podržava TLS (Transport Layer Security) protokol kao što je definisano za XMPP u RFC 3920 i host predstavi digitalni sertifikat. Međutim, sertifikat može biti samo-potpisan, u tom slučaju uzajamna potvrda identiteta obično nije moguća. Stoga, nakon pregovora STARTTLS⁷, strane će nastaviti sa slabijom proverom identiteta koristeći Server Dialback. Ova kombinacija rezultira enkriptovanom konekcijom sa slabom verifikacijom identiteta.
4. **Trusted Federation** - server prihvata konekciju sa hosta samo ako taj host podržava TLS (Transport Layer Security) protokol i host predstavi digitalni sertifikat izdat od strane pouzdanog CA tela (Trusted root CA). Lista poverenih CA tela je određena od strane polise lokalnih servisa, kao što je nivo poverenja prema različitim tipovima sertifikata (tj. Klasa 1, Klasa 2, ili Klasa 3). Upotreba pouzdanih sertifikata domena efikasno sprečava DNS poisoning napade ali otežava kreiranje federacija s obzirom da obično takve sertifikate nije lako dobiti.

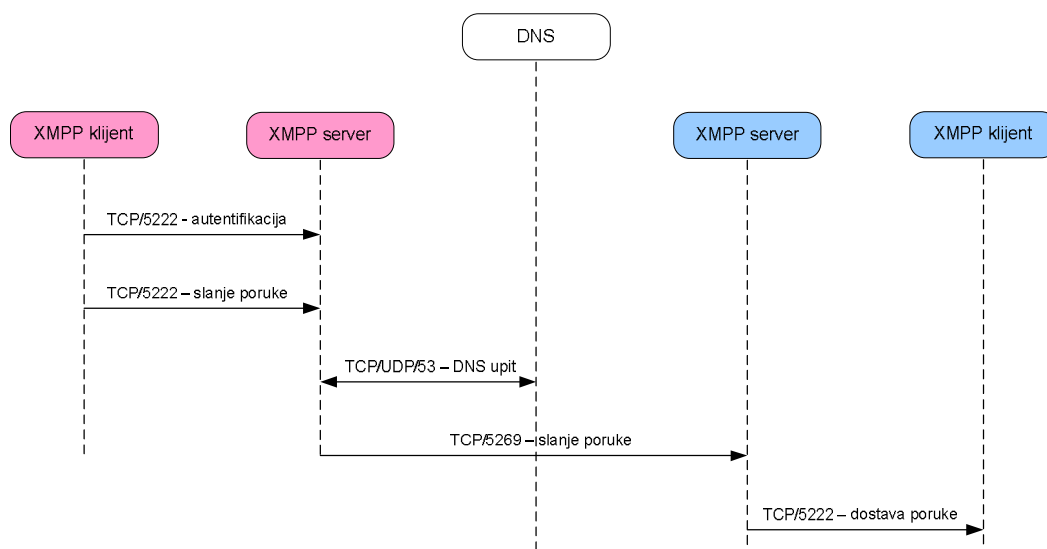
Iako su klijenti i serveri fundamentalni entiteti na XMPP mreži, postoje i drugi entiteti koji imaju ulogu u datoj mreži. Automatizovani klijenti koji se nazivaju botovima, omogućavaju širok spektar komunikacionih servisa, uključujući pomoć u chat sobama i olakšanim interfejsima za servise koji nisu tipa XMPP, kao aplikacije društvenih mreža. Samim tim, većina XMPP servera su napisani u modularnom obliku tako da omogući administratorima da dodaju posebne servise ili serverske komponente, kao što su višekorisničke chat sobe, parametre za igre, i slično.

⁷ STARTTLS - ekstenzija za enkripciju klasičnih tekstualnih protokola.

3.5. Funkcionisanje XMPP sa DNS protokolom

XMPP protokol je prevashodno arhitekture tipa klijent-server, ali je naravno podržana i komunikacija server-server.

- Poruke se ne šalju direktno između klijenata, nego se umesto toga šalju od klijenta pošiljaoca ka svom serveru i njegov server dalje šalje poruku ka serveru primaoca koji onda dostavlja poruku klijentu primaoca.
- Ako primalac nije prisutan, onda ako je to omogućeno na serveru, poruke će čuvati za naknadnu dostavu kada se korisnik vrati na mrežu.
- Serveri potražuju jedni druge koristeći servis DNS. Ime servera primaoca je naziv hosta sa desne strane JID korisnika. Na primer, u JID djordje@chat.vs.rs, ime XMPP servera je chat.vs.rs.
- DNS je širok sistem na Internetu koji se koristi za pretvaranje naziva hostova (jchat.vs.rs) u IP adrese (kao 10.90.76.246).
- Računari koriste IP adrese i DNS pretrage da razgovaraju jedni s drugima, što je i prikazano na Slici 7.



Slika 7.

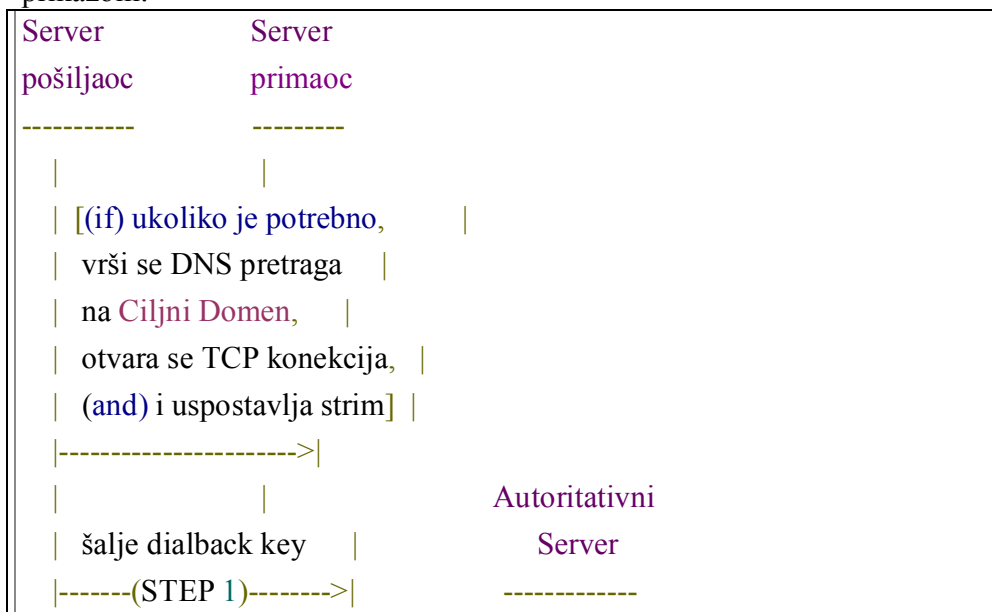
- XMPP klijent pošiljaoca otvara TCP konekciju na portu 5222 ka XMPP serveru i vrši se autentifikacija korisnika.
- XMPP klijent pošiljaoca koristi TCP konekciju da pošalje XMPP paket brze poruke svom serveru.
- XMPP server pošiljaoca pretražuje IP adresu XMPP servera primaoca na osnovu JID primaoca.

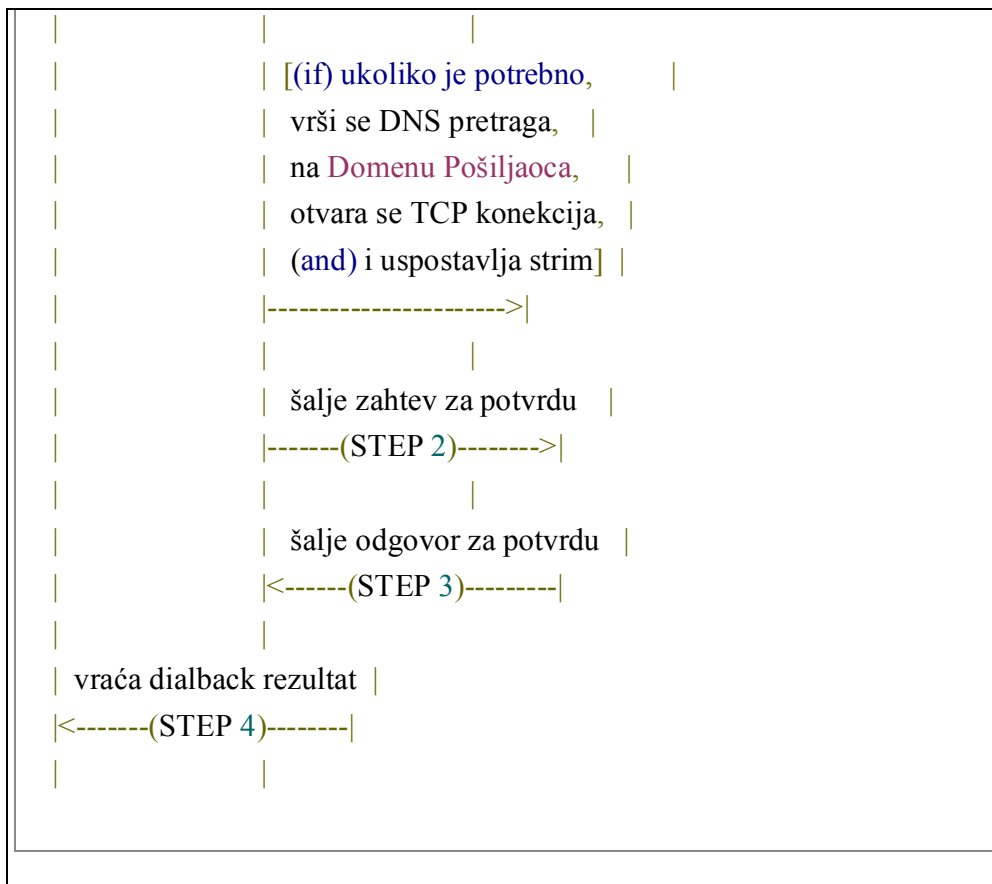
- XMPP server pošiljaoca otvara server-server TCP/IP konekciju na portu 5269 ka XMPP serveru primaoca i šalje spremljenu poruku.
- XMPP server primaoca koristi client-server konekciju da prosledi poruku XMPP klijentu primaoca.

XEP 220 definiše osnovni tok procesa za Server Dialback po sledećem:

1. Server-pošiljaoc generiše dialback ključ i šalje tu vrednost putem svog XML toka (stream) sa serverom-primaocem. Ako server-pošiljaoc još uvek nema uspostavljen XML tok sa serverom-primaocem, onda mora da obavi DNS pretragu ciljanog domena i na taj način da pronade server-primaoc, otvori TCP konekciju pronađene IP adrese i port, i uspostavi XML tok sa serverom-primaocem.
2. Umesto da odmah prihvati XML zapis konekcije servera-pošiljaoca, server-primaoc šalje isti dialback ključ putem svog XML toka sa autoritativnim serverom za verifikaciju. Ako server-primaoca još uvek nema XML tok sa autoritativnim serverom, onda će morati da obavi DNS pretragu domena pošiljaoca i na taj način da pronade autoritativni server, otvori TCP konekciju za datu IP adresu i port, i uspostavi XML tok sa autoritativnim serverom).
3. Autoritativni server obaveštava server-primaoc da li je ključ validan ili nije.
4. Server-primaoc obaveštava servera-pošiljaoca da li je njegov identitet potvrđen ili nije.
5. Nakon koraka 4, server-pošiljaoc je autorizovan da šalje zapise iz svog domena do namenjenog domena, što se izražava u komunikaciji na "da" i "od" atributima dialback pregovora. Pored slabe verifikacije identiteta domena pošiljaoca, ovim se osigurava da server-primaoc prihvata zapise za ciljni domen.

Za standardna okruženja autoritativni server je istovremeno server-pošiljaoc, ali to može biti zaseban računar u istoj mreži (gde je "mreža" definisana sa znanjem deljene tajnosti za verifikaciju dialback ključeva). Prethodno navedeno se može predstaviti grafičkim prikazom:





3.6. Adresiranje

Zbog toga što se XMPP komunikacije dešavaju u mrežnom okruženju, svaki XMPP entitet mora da ima adresu, koja se zove JabberID (JID) [18]. XMPP se tipično oslanja na Domain Name System (DNS) da bi obezbedio potpunu strukturu za adresiranje, umesto upotrebe osnovnih Internet Protocol (IP) adresa. Svakako je lakše zapamtiti da je XMPP servis dostupan na jabber.org, nego na adresi 208.68.163.220. Shodno tome, JabberID-ovi za korisnike imaju izgled kao e-mail adrese (npr. aleksandarv@jabber.org) jer je format user@domain.tld već poznat ljudima; pored toga, ovaj format koristi kompletnu DNS infrastrukturu kao svoj adresni prostor, za razliku od starijih IM sistema koji su koristili brojeve ili imena bez domenskog identifikatora.

3.6.1. Domeni

Svaki JabberID sadrži deo naziva domena, koji se tipično mapira kao fully qualified domain name (FQDN). Prilikom instalacije XMPP serverskog softvera, odabira se ime domena za distribuciju, kao vs.rs ili gmail.com. Koristeći zapise DNS servisa, naziv domena se mapira na jednu ili više određenih mašina, kao npr. doshke.vs.rs ili

talk.google.com. Ti nazivi mašina se zauzvrat mapiraju u određene IP adrese, kao npr. 208.68.163.220 ili 72.14.253.125.

Naravno, za svrhu adresiranja na mreži najbitniji je naziv domena (vs.rs gmail.com), nego imena mašina nižeg nivoa i IP adrese. Za ASCII karaktere, domenski deo JID **case-insensitive** (tako da je JABBER.ORG isto što i jabber.org).

3.6.2. Korisnici

Kada kreiramo nalog na XMPP servisu kao što je jabber.org, onda biramo JabberID koji funkcioniše kao virtuelni identitet na mreži. Alternativno, taj JabberID može biti dodeljen automatski. JabberID uglavnom izgleda kao e-mail adresa. U zavisnosti od primenjenih polisa, može biti identična kao e-mail adresa nekog servisa ili kompanije (npr. Google Talk adresa na XMPP mreži izgleda kao Gmail adresa na e-mail mreži). Kao i kod domenskog dela naziva JabberID, korisnički deo JID je **case-insensitive** ASCII karaktere (tako da DjordjeD@vs.rs je isto što i djordjed@vs.rs). XMPP programeri obično nazivaju adresu u formi **user@domain.tld** "**čisti ili goli**" JID.

3.6.3. Resursi

Kada se konektujemo sa klijentom na XMPP server, biramo (ili nam server dodeljuje) identifikator resursa (resource identifier) za određenu konekciju. Ovaj resurs služi za rutiranje saobraćaja ka toj konekciji umesto bilo koje druge konekcije koju smo možda otvorili u datom trenutku. Resurs se dodaje na kraju adrese naloga, kao npr. djordjed@vs.rs/roundabout ili rashan@uzice.net/home. Ovim se omogućava upit ili razmena poruka preko određenog uređaja koji je povezan sa nalogom; takođe predstavlja i to da je uređaj posebna "tačka prisutnosti" sa različitim statusima dostupnosti, mogućnostima, itd. Resurs često predstavlja naziv računara, lokacije, klijentskog softvera koji se koristi, ali može biti bilo kakav string (uključujući prazan karakter i druge specijalne karaktere). Suprotno drugim celinama JID, deo koji čini resurs je **case-sensitive** (djordjed@vs.rs/home se razlikuje od djordjed@vs.rs/Home). XMPP programeri obično nazivaju adresu u formi **user@domain.tld/resource** "**punim**" JID.

3.6.4. Internacionalizacija

Velika razlika između JabberID i e-mail adrese je ta da je XMPP u potpunosti internacionalizovan. To znači da XMPP nazivi domena i korisnička imena nisu ograničeni na stari ASCII karakterni opseg, nego mogu da obuhvate bukvalno svaki Unicode karakter. Primera radi, ukoliko imamo Jabber server na domenu čistač.rs, onda može postojati adresa kao žika@čistač.rs. Ili ako se bavimo matematikom, naš JabberID može biti nešto kao ∞@matematika.rs. Za karaktere koji nisu tipa ASCII, nije case-sensitivity u prvom planu, nego case-folding. Iako neka komplikovana pravila poređenja i dekompozicije karaktera mogu da se primene koristeći Unicode karaktere, ova case-folding pravila (definisana tehnologijom stringprep) su forsirana od strane programskih biblioteka nižeg nivoa, tako da većina programera ne mora da se bavi time direktno[19].

3.6.5. XMPP URI

Na samoj XMPP mreži, JabberID su obezbeđene kao kao obične adrese bez Uniform Resource Identifier (URI) šeme. To bi izgledalo otprilike kao kada posećujemo www.google.com (umesto <http://www.oreilly.com/>), ili šaljemo e-mail na standards@xmpp.org (umesto <mailto:standards@xmpp.org>). Ipak, postoji XMPP URI šema koja se može upotrebiti za identifikaciju JabberID kao URI, kao npr. <xmpp:jabber.org> ili <xmpp:djordje@jabber.org> (bez `"/`). Ova URI šema je definisana u RFC 5122. XMPP zajednica je takođe definisala načine za obuhvatanje raznih komandi u XMPP URI, kao <xmpp:user@domain.tld?message> za slanje poruke [20XEP-0147].

3.7. Strimovanje XML-a

XMPP je u suštini, tehnologija za prenos (strimovanje) XML-a[21]. Kada se pokreće sesija sa XMPP serverom, otvara se dugotrajna TCP konekcija i onda se uspostavlja XML prenos do servera (server takođe otvara prenos u suprotnom smeru). Sam strim može da se predstavi kao XML dokument koji se povećava vremenom, tokom sesije između klijenta i servera. Nakon uspostave XML strima sa serverom, klijent i server mogu da razmene tri specijalna XML isečka putem strima: `<message/>`, `<presence/>`, i `<iq/>`. Ovi isecci, koje se nazivaju XML "strofama" (eng. stanza), predstavljaju osnovne tekstualne strukture XMPP-a, i nakon uspostave XML strima, može se slati neograničen broj XML strofa preko tog istog strima.

Kako izgleda jedna uprošćena XMPP sesija, ilustrovano je u sledećem primeru:

```
C: <stream:stream>
C:   <presence/>
C:   <iq type="get">
      <query xmlns="jabber:iq:roster"/>
    </iq>
S:   <iq type="result">
      <query xmlns="jabber:iq:roster">
        <item jid="djordje@vs.rs"/>
        <item jid="marko@vs.rs"/>
        <item jid="aleksandar@vs.rs"/>
      </query>
    </iq>
C:   <message from=nacelnik@vs.rs
           to="marko@vs.rs">
      <body>Napisi izvestaj.</body>
    </message>
S:   <message from=direktor@vs.rs
           to="informatika@conference.vs.rs">
      <body>Slobodni ste.</body>
    </message>
C:   <presence type="unavailable"/>
C: </stream:stream>
```

Unutar XMPP sesije, sami elementi strima imaju ulogu enkapsulacije za neograničen broj odlaznih i dolaznih XML strofa; odlazne strofe poslate od klijenata su obeležene sa C, a dolazne koje su dostavljene od servera sa S.

Pristup XMPP-a za otvaranje dugotrajnih TCP konekcija i asinhronu razmene neograničenog broja XML isečaka se drastično razlikuje od tradicionalnog pristupa korišćenog u web i e-mail tehnologijama, gde se otvara TCP konekcija, izvrši transakcija i zatvori konekcija. Ovakav tip konekcija nije uvek na raspolaganju komunikacijama u realnom vremenu, jer server nema "stalno-uključen" kanal koji je dostupan za protok informacija ka klijentu. Kao rezultat toga, jeste konstantna potražnja za novim i svežijim informacijama.

Suprotno tome, u XMPP protokolu, klijent može da pošalje više zahteva bez "kočenja" dok čeka za odgovore, i server će vratiti te odgovore dinamički čim dobije odzive.

3.8. Primitive u komunikaciji

XMPP strofa kao takva, može da se shvati kao jedan vid osnovne jedinice u komunikaciji, slično paketu ili poruci u drugim mrežnim protokolima⁸. Nekoliko faktora određuju značenje strofe:

- Nazivi elemenata strofe, a to su poruka, prisutnost, ili iq. Svaki tip strofe se preusmerava drugačije na serverima i prihvata drugačije na klijentima.
- Vrednost tipa atributa, koja varira u zavisnosti od vrste date strofe. Ova vrednost dalje diferencira način na koji će se tip strofe obraditi od strane primaoca.
- Elementi "dece" koji definišu veličinu transporta strofe. Veličina transporta može da se predstavi korisniku ili da se procesira po nekom automatizmu što je i određeno specifikacijom koja definiše imenski prostor (namespace) za veličinu transporta.

3.8.1. Poruka

XMPP <message/> strofa je osnovna "gurajuća" metoda za dobijanje informacije sa jednog mesta do drugog. S obzirom da su poruke tipično nepotvrđene, tj. one su na neki način "ispali i zaboravi" mehanizam za brzo dobijanje informacija sa jednog mesta na drugo. Poruke se koriste za IM, grupni chat, alarme i obaveštenja, i druge slične aplikacije. Postoji pet vrsta strofa za poruke, a razlikuju se po tipovima atributa:

Normal

Normalni tipovi poruka su najbližnji e-mail porukama, jer su to pojedinačne poruke na čiji se odziv može ali i ne mora očekivati.

Chat

Chat poruke se razmenjuju u sesiji u približno realnom vremenu, između dva entiteta, najčešće kao IM između dva prijatelja.

⁸ Izraz je predložen od strane Lize Disol, jednom od predsedavajućih u IETF XMPP Radnoj grupi.

Groupchat

Poruke grupnog chat-a se razmenjuju u višekorisničkoj chat sobi, slično IRC (Internet Relay Chat).

Headline

Naslovne poruke se koriste za slanje upozorenja i obaveštenja, a odgovor se uopšte ne očekuje (klijent koji prima ovaj tip poruke treba da onemogući korisnika da odgovori).

Error

Ukoliko se pojavi greška vezana za prethodno poslatu poruku, entitet koji detektuje problem vratiće poruku o grešci u kucanju.

Kao dodatak tipu atributa, strofa poruke sadrži adrese pošiljaoca i primaoca, može da sadrži id atribut za praćenje. Adresa pošiljaoca nije navedena od strane klijenta koji šalje, nego server pošiljaoca vrši upis radi izbegavanja lažiranja adrese.

Poruke takođe sadrže elemente korisnog transporta. Osnova XMPP specifikacija definiše neke veoma bazične korisne prenose, kao što su <body/> i <subject/>, koji se koriste za osoba-osoba chat poruke. Primera radi, jednostavna poruka može da izgleda ovako:

```
<message from=aleksandar@vs.rs/admin
        to=djordje@vs.rs
        type="chat">
  <body>Who are you?</body>
  <subject>Query</subject>
</message>
```

Redosled atributa nije bitan. Poruke takođe mogu da sadrže koirnsni transport koji nije definisan osnovama XMPP specifikacija.

3.8.2. Prisutnost

Jedna od prepoznatljivih funkcionalnosti komunikacionih sistema u realnom vremenu, jeste prisutnost (dostupnost) [22]. Prisutnost oglašava dostupnost drugih entiteta na mreži, čime pruža saznanje o tome da li su drugi entiteti online i dostupni za komunikaciju. Prisutnost predstavlja katalizator u smislu komunikacije i kolaboracije preko Interneta, jer ljudi češće ulaze u interakciju sa drugim ljudima kada znaju da su online.

Naravno ne može doći do komunikacije sve dok se ne odobri ista. Ova autorizacija je na neki način pretplata na korisničku dostupnost i dopisivanje (presence subscription). Da bi neko video da li ste prisutni, ta osoba mora da vam pošalje zahtev za dopisivanje, koji vi odobravate. Nakon odobrenja, korisnik koji je slao zahtev može automatski da prima regularna obaveštenja o vašoj prisutnosti.

Ovakav model implicira da je XMPP <presence/> strofa u suštini jednostavna, specijalizovana objavi-pretplati (publish-subscribe) metoda, gde korisnici koji se pretplate na vašu dostupnost primaju ažurne informacije o prisutnosti kada ste online, promenite status "na sastanku sam" ili "na ručku sam" i predete u offline status. Osnova XMPP prisutnosti (online ili offline) se često nadograđuje sa nekim uobičajenim statusima kao što su "odsutan" i "ne uznemiravaj". Ovakva statusi se mogu personalizovati koristeći statusne poruke tipa "na vozu" ili "Trenutno pišem, ne uznemiravaj me sada". Na primer:

```
<presence from="aleksandar@vs.rs/mobile">  
  <show>away</show>  
  <status>na otpustu sam!</status>  
</presence>
```

U XMPP IM aplikacijama, prisutnost je tipično prikazana u kontakt listi. Lista kontakata sadrži pregled JabberID-eva i status dostupnosti za dopisivanje sa korisnicima na listi.

3.8.3. IQ

Info/Query (IQ) strofa omogućava strukturu za zahtev-odziv interakcije i jednostavne radne procese, nešto slično GET, POST i PUT iz HTTP-a. Za razliku od <message/> strofe, IQ strofa uključuje samo jedan korisni transport (payload), kojim se definiše zahtev koji treba da se obradi ili akciju koju treba da preuzme primalac. Pritom, entitet koji šalje IQ strofu mora uvek da primi odgovor (obično generisan od strane namenjenog primaoca ili njegovog servera). Zahtevi i odzivi se prate pomoću id atributa, kojeg generiše zahtevajući entitet i prihvata primajući entitet. Tip atributa ima posebnu vrednost za IQ strofe:

Get

Zahtevajući entitet traži informaciju, kao što su zahtevi registrovanje naloga (slično HTTP GET).

Set

Zahtevajući entitet daje neke informacije ili pravi zahtev (slično HTTP POST ili PUT).

Result

Odzivni entitet vraća rezultat get operacije (npr. informaciju koju entitet mora da obezbedi da bi registrovao nalog), ili priznaje set zahtev (slično HTTP 200 statusnom kodu).

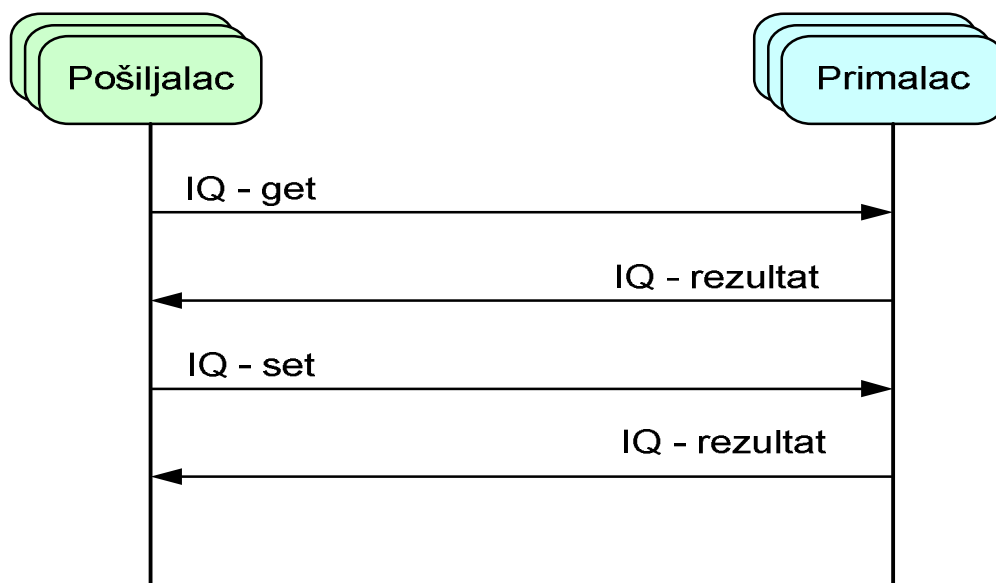
Error

Odzivni entitet ili posredni entitet, kao XMPP server, obaveštava zahtevajući entitet koji nije u mogućnosti da obradi get ili set zahtev (jer je zahtev loše kreiran, zahtevajući entitet nema privilegiju da izvodi operaciju itd.). Ranija upotreba HTTP-stilizovanih numeričkih kodova za greške je zamenjena sa XML elementima za uslove proširenih grešaka.

Koristeći vrednosti tipa IQ strofe, može se generisati solidno struktuirana IQ interakcija između dva entiteta, što je i prikazano na Slici 8. Radi ilustracije ovakvog načina interakcije u detaljima, razmotrićemo kroz sledeći primer proces dobijanja liste kontakata i njegovog ažuriranja:

```
<iq from="aleksandar@vs.rs/mobile  
  id="rr82alz7"  
  to="aleksandar@vs.rs"  
  type="get">  
  <query xmlns="jabber:iq:roster"/>
```

</iq



Slika 8.

Ovde Aleksandar traži od svog servera (vs.rs) kontakt listu koja se nalazi na serveru, šaljući IQ-get zahtev koji sadrži prazan korisni transport označen jabber:iq:roster imenskim prostorom. Server odgovara sa ispunjenim korisnim transportom označenim sa datim imenskim prostorom, u ovom slučaju sadržavajući jedan `<item/>` element za svaki kontakt na listi. (klijentu je poznato da se rezultat ove liste odnosi njegov inicijalni zahtev jer je server uvrstio id atribut sa vrednošću rr82alz7.)

```

<iq from="aleksandar@vs.rs"
    id="rr82alz7"
    to="aleksandar@vs.rs/mobile"
    type="result">
  <query xmlns="jabber:iq:roster">
    <item jid="djordje@vs.rs"/>
    <item jid="marko@vs.rs"/>
    <item jid="milica@vs.rs"/>
    <item jid="dragana@home.rs"/>
  </query>
</iq>

```

Aleksandar takođe može da upotrebi IQ-set za dodavanje novog kontakta na listu:

```

<iq from="aleksandar@vs.rs/mobile"
    id="ru76lvd7"
    to="aleksandar@vs.rs"
    type="set">
  <query xmlns="jabber:iq:roster">
    <item jid="slobodan@vs.rs"/>
  </query>
</iq>

```

Njegov server onda jednostavno potvrđuje ažuriranje liste kontakata vraćajući prazan IQ-rezultat:

```
<iq from="aleksandar@vs.rs"
    id="ru761vd7"
    to="aleksandar@vs.rs/mobile"
    type="result"/>
```

Kao što se može primetiti iz ovih prethodno navedenih primera, korisni transport IQ-get ili IQ-set je uvek definisan sopstvenim formatom, označenim određenim XML imenskim prostorom. Svaki oblik korisnog transporta možemo zamisliti kao komandu koju treba da obradi primalac.

IQ-get zahteva određenu vrstu informacije, kao što je registracioni obrazac, podaci za konfiguraciju, ili kontakt listu. An IQ-set kreira, ažurira, ili briše određenu vrstu informacije, kao što je popunjen obrazac, ažurirane podatke za konfiguraciju, ili dopunu kontakt liste.

3.8.4. Proširivost

XML strofa može da sadrži bilo koji broj elemenata "dece", uključujući XHTML formatirana tela poruka, pokazivače ka URL-ovima, RSS ili Atom obaveštenja, obrasce koji treba da se popune (ili već popunjene obrasce), XML-RPC ili SOAP podatke za web servise, geografske lokacije, i druge razne vidove korisnog transporta.

Zbog toga što je XMPP isključivo XML tehnologija, samim tim prilično iskorišćava XML imenske prostore kao način proširivanja korisnog transporta strofe. Ove imenske prostore možemo predstaviti kao XML ekvivalentne pakete i imenske prostore u programiranju.

Dosada, XMPP programerska zajednica je definisala popriličan broj ekstenzija za osnovne XMPP slojeve strofa. Većina ovih ekstenzija se objavljuju od strane XMPP Standards Foundation na <http://xmpp.org/>, ali svako može definisati sopstvene privatne ekstenzije za posebne funkcionalnosti. Ekstenzije se izjednačavaju i po nazivu elementa i po imenskom prostoru. U počecima Jabber open source projekata, programeri su koristili <x/> element za ekstenzije koji bi postavili u strofama poruke i prisutnosti, i <query/> element za ekstenzije koji bi postavili u IQ strofe.

4. PRETNJE I RIZICI

Brze poruke (Instant messaging) sve više predstavljaju pretnju kao posrednik za maliciozni softver (malware). Mnogo ljudi koristi IM servis, za lične ili poslovne potrebe, IM mreže im omogućavaju ne samo prenos tekstualnih poruka, nego i prenos fajlova. Kao posledicu takvih transfera, IM programi mogu da prenesu crve i drugi maliciozni softver. Mnoge ranjivosti su otkrivene i verovatno će da se otkriju u klijentskim aplikacijama.

Takve ranjivosti ne samo da daju hakerima daljinski pristup, nego i obezbeđuju brzo širenje mogućih pretnji. Trenutne pretnje su ograničene svojim mogućnostima za nalaženje ranjivih hostova, ali sa IM kontakt listama, pronalaženje slabo zaštićenih hostova postaje značajno lakše što za rezultat ima objedinjenu pretnju koja može da se propagira brže od CodeRED i Slammera.⁹ U ovom poglavlju ćemo izvršiti analizu pretnji i bezbednosti kroz razne mehanizme i metode. Najuoobičajnije pretnje po neki komunikacioni sistem jesu:

- prisluškivanje komunikacionih kanala,
- neautentifikovani ili korisnici sa slabom autentifikacijom,
- lažni serveri,
- lažiranje adresa,
- DoS napadi,
- virusi, crvi, i drugi maliciozni softver,
- neželjene komunikacije (spam),
- imitiranje i lažno predstavljanje (phishing),
- curenje ličnih ili tajnih (privilegovanih) informacija,
- neadekvatno arhiviranje i vođenje elektronskih dnevnika (logging),
- prekoračenja bafera i drugi bezbednosni propusti u kodu.

4.1. Opšte pretnje

4.1.1. Nesigurne (otvorene) konekcije

Jedna od najvećih pretnji za IM mreže predstavljaju njihove otvorene, nesigurne konekcije. IM konekcije su podložne na preuzimanje tokom klijent-server, klijent-klijent i server-server komunikacije. Kada se izvrši autentifikacija nakon logovanja, sve ove konekcije ispostavljaju slabije mere bezbednosti (broj sekvence ili identifikator transakcije, koji može biti lako lažiran) ili su bez bezbednosnih mera. Sa tim, dosta popularnim IM konekcijama nedostaje autentifikacija (osim u poruci logovanja), poverljivost i integritet.

Time se otvaraju vrata mnogim drugim bezbednosnim ranjivostima uključujući lažno predstavljanje (impersonation), denial of service (DoS), man-in-the-middle, itd. Primera radi, čak iako korisnik odluči da prima poruke samo od prijatelja sa kontakt liste, moguće je

⁹ CodeRED i Slammer crvi

da prima i neželjene poruke od nepoznatih entiteta; sve što je potrebno, jeste da napadač uhvati jednu od tih otvorenih TCP konekcija sa nekim korisnikom iz liste kontakata[23].

4.1.2. Denial of Service (DoS) napad

DoS napadi mogu da se izvrše na različite načine. Pojedini napadi mogu da onemoguće sam klijentski program za slanje poruka. Napadači takođe mogu da iskoriste klijenta da izvrše propterećenje memorije ili mikroprocesora računara da bi srušili sistem. DoS napadi mogu da obuhvate i IM server i propusni opseg mreže. Pretrpavanjem neželjenim porukama je prilično olakšano kada korisnici prihvataju poruke od svakoga.

Današnji IM klijenti generalno imaju podršku za blokiranje korisnika. Međutim, iako žrtva može da blokira korisnički ID, napadači mogu da probiju i ovu barijeru kompromitujući naloge više korisnika istovremeno.

4.1.3. Imitiranje

Napadači mogu da imitiraju validne korisnike na najmanje dva načina. Kada sazna za korisničku lozinku, napadač koristi automatizovane skripte u cilju imitacije datog korisnika prema njegovim kontaktima. Naravno, napadači mogu da preuzmu klijent-server konekcije, odmah nakon prijavljivanja korisnika, uoči iniciranja konekcije sa drugim korisnikom ili kada se konekcija naglo prekine. Server će držati konekcije otvorenim neko određeno vreme (vreme isteka - keepalive), pa napadači mogu da iskoriste dati vremesni period da preuzmu konekciju ka serveru.

4.1.4. Lažiranje DNS i IM servera

Štetni programi (npr. QHosts-1) mogu da menjaju TCP/IP podešavanja u sistemu pogođenog korisnika i da ga usmeri na drugi DNS server. Zlonamerni hakeri mogu da postave IM server i upotrebe lažiranje DNS servera tako da se korisnici prijave na pogrešan server umesto regularnog servera.

IM klijenti nemaju efikasniji način da provere da li komuniciraju sa legitimnim serverima. Serveri uglavnom proveravaju identitet klijenta preko hash-a korisničko imena i lozinke. Ovakav tip klijentske autentifikacije je ranjiv na (IM) man-in-the-middle napade gde se maliciozni server predstavlja kao legitimni. Napadi kao što su prikupljanje informacija vezanih za naloge, prisluškivanje i lažno predstavljanje su mogući ukoliko je IM server kompromitovan.

4.2. Pretnje iz IM funkcionalnosti

IM klijenti danas pružaju korisnicima mnogo korisnih mogućnosti (npr. prenos fajlova, deljenje aplikacija i dr.) pored osnovnih opcija za slanje poruka. Naravno, postoje i prateće ranjivosti i pretnje datih uslužnih IM servisa.

4.2.1. Razmena fajlova

Crvi mogu lako da se propagiraju kroz IM mreže, koristeći funkcionalnost prenosa fajlova. Generalno, korisnici nisu mnogo obazrivi prilikom prijema fajla od poznatog kontakta. Naime, crvi često eksploatišu ove situacije imitirajući pošiljaoca. Takođe, IM transfer fajlova koji sadrže maliciozni softver, relativno lako prolazi firewall nego fajl poslat putem elektronske pošte. To se pretežno dešava zbog poteškoća u razlikovanju IM saobraćaja na izlaznoj tački mreže, i upotrebe tuđih (vlasničkih protokola) od strane IM provajdera [24].

Kao što koriste e-mail adresar, IM crvi takođe koriste korisničku listu kontakata (roster) kao sredstvo širenja; i za razliku od offline i sporih e-mail propagacija, IM liste kontakata daju veoma brzo mete za brzo širenje.

Neki IM klijenti imaju opcije pokretanja ili otvaranja fajlova odmah nakon preuzimanja, što za posledicu ima širenje malicioznog koda bez neke kontrole samog korisnika. Na primer u ICQ, korisnici mogu da izaberu da li će automatski prihvatati sve nadolazeće razmene fajlova. Sa novijim IM klijentima, korisnici mogu da podese automatsko skeniranje fajlova koji se preuzimaju, ali antivirusni programi i alati ne skeniraju sve moguće fajlove i njihov kod (kao npr. MPEG fajlove). Takvi fajlovi mogu da sadrže posebne programske sekvence koje bi sa pokretanjem, srušile video aplikaciju ili instalirale backdoor program [25].

4.2.2. Registri u formi običnog teksta i arhiviranje poruka

Postoje bezbednosna podešavanja i kod IM klijentskih aplikacija koje napredniji korisnici mogu da konfigurišu u zavisnosti od njihovih potreba. IM aplikacije imaju nekoliko podešavanja u registru operativnog sistema Windows. Bilo koji tehnički obrazovan korisnik Windows-a može da pročita vrednosti registara, a još ako ima administratorske privilegije, može da ih menja. Neke od bezbednosnih podešavanja vezana za IM program koja se čuvaju u registrima su: kriptovana lozinka; korisnički ID; da li se skeniraju antivirusom dolazeći fajlovi; softverska putanja antivirusa; da li je potrebna dozvola za dodavanje u nečijoj listi kontakata; ko može da kontaktira korisnika (korisnici sa liste kontakata ili svi); da li da se dele fajlovi sa ostalima; putanja deljenog direktorijuma; da li da se traži lozinka prilikom promene ovih podešavanja itd. MSN Messenger je skladištio korisničku kontakt listu, listu blokiranja i listu dozvole u registrima u formatu razumljivog ljudima. Modifikacija registarskih zapisa omogućava napadačima da zaobiđu pojedine bezbednosne opcije (kao dozvole za razmenu fajlova, ovlašćenje dodavanja kontakata i dr.). Prikupljanjem ID korisnika i kriptovanih lozinki, napadači preuzimaju kontrolu nad korisničkim nalogima. Takođe, lozinke formata običnog teksta se relativno lako dešifruju koristeći razne alate [26].

IM klijenti uglavnom imaju opciju arhiviranja poruka. Konverzacije korisnika se čuvaju u formatu običnog teksta u predvidljivoj sistemskoj lokaciji. Poverljivost ovih poruka može da bude kompromitovana kako za korporativne tako i za obične korisnike.

4.2.3. Nesigurna osnovna (default) podešavanja

Ono što je uobičajen sigurnosni nedostatak za većinu korisničkih IM aplikacija jeste osnovno podešavanje koje ima prilikom prve upotrebe. Dosta IM klijenata, podrazumevano dozvoljava svima koji koriste isti IM servis da kontaktiraju korisnika. Dozvoljeni prijem

poruka od bilo koga otvara vrata ka smetnji kao što je "spim" - tzv. spam poslat putem IM sistema. Ova opcija može da bude ograničena na to da dozvoljava unose od korisničke liste kontakata, zato što IM korisnici ne komuniciraju sa strancima često [27]. Osnovna lokacija za IM preuzimanje fajla na korisničkom računaru može da bude eksploatisana radi pokretanja malicioznog koda (npr. ranjivost fajla ICQ scm) [28]. Kod ICQ, podrazumevano podešavanje za autorizaciju liste kontakata je izgledalo otprilike ovako: svi korisnici mogu da me dodaju na kontakt liste kada vide moj online/offline status. Očigledno da ovo nije baš najbolje sigurnosno podešavanje.

4.2.4. Deljenje IM funkcionalnosti sa drugim aplikacijama

MSN Messenger kontakt lista i druge funkcionalnosti su dostupne od drugih aplikacija kao što su Microsoft Outlook Express e-mail klijent i web-bazirani Hotmail e-mail servis. Microsoft je takođe objavio IM API-je programere aplikacija radi zasebne integracije IM funkcionalnosti sa bilo kojim softverskim proizvodom. Microsoft-ov Live Communication Server (LCS) je integrisao MSN, AIM i Yahoo! javne IM servise sa Microsoft Office programima da bi pojačao saradnju u realnom vremenu. Paralelno sa integracijom IM mogućnosti sa različitim aplikacijama, povećavaju se bezbednosni rizici i za IM servise i za date aplikacije. Sigurnosni upad u IM servis može da pogodi i aplikacije koje integrišu IM funkcionalnosti, i obrnuto.

4.2.5. Zlonamerni hiperlinkovi

Linkovi ka web stranama koje imaju zlonameran sadržaj mogu da se pošalju putem regularnih brzih poruka. Neki IM klijenti poseduju opciju da li da prihvate ili odbiju poruke sa hiperlinkovima (mada je podrazumevano podešavanje da se primaju URL poruke od svih). Na primer u AIM, korisnik je mogao da kreira hiperlinkove gde vidljiv tekst nema uopšte veze sa ponuđenim web linkom. Ovakva situacija može lako da dovede korisnika u zabludu da pristupa bezbednom linku. Maliciozni hiperlinkovi mogu da usmere korisnike na fišing web stranice.

4.2.6. URI (Universal Resource Identifier) moduli

YIM i AIM klijenti instaliraju prilagođene URI module ymsgr i aim. Ovi URI pružaju pomoć u pisanju korisnih skripti za obrađivanje aplikacija kao Microsoft IE, Netscape Navigator, Mozilla Firefox, Microsoft Outlook, ili Windows command shell. URI može biti poslat od strane drugog YIM ili AIM korisnika u IM tekstualnoj poruci ili HTML e-mail poruci. Korisnici mogu da ugrađuju ove URI linkove na sopstvene web strane. Web pregledači i komandno okruženje može da se upotrebi za pokretanje AIM ili YIM obrađuju ove URI. Primera radi, ako YIM korisnik izvrši URI ymsgr:addfriend?drugar iz IE (klikom na hiperlink), YIM klijent će se pokrenuti (ili je već pokrenut), i korisnik će biti obavešten da doda drugara u kontakt listu.

Nedostatak obaveznog proveravanja parametara ovih protokola je dozvolio zlonamernim hakerima da pokreću razne buffer overflow napade[29]. Programske putanje YIM/AIM klijenata odgovornih za obradu ymsgr/aim URI se nalaze u Windows registru. Promenom takvih registarskih zapisa u bilo koji maliciozni program, napadači mogu da aktiviraju te

programe prilikom pozivanja URI. Skripte koje su napisane korišćenjem ovih URI otvara prostor za automatizovane napade.

4.3. Crvi

IM crvi predstavljaju sve crve koji se šire po IM mrežama, eksploatacijom funkcionalnosti i ranjivosti IM klijentskih programa i protokola. Pojam crva možemo definisati kao maliciozni kod koji se propagira preko mreže, sa ili bez uticaja čoveka [30]. Prvi IM crv se pojavio 2001. godine [31], da bi od tada IM mreže postajale sve više ranjive na crve. Najčešći načini širenja i replikacije IM crva su:

- razmenom, odnosno transferom malicioznog fajla (zahteva učešće korisnika),
- malicioznim URL-om u tekstualnoj poruci (zahteva učešće korisnika),
- eksploatacijom grešaka implementacije IM klijentskih aplikacija (sa ili bez učešća korisnika),
- eksploatacijom operativnih sistema ili najčešće korišćenih softverskih aplikacija (sa ili bez učešća korisnika).

4.3.1. Ranjivosti klijentskih aplikacija

Postoji mnogo ranjivosti u IM klijentima koje zlonamerni hakeri mogu da iskoriste za kreiranje IM crve. Na primer, prekoračenja bafera (buffer overflow) su otkrivena i zakrpljena u ranijim verzijama AIM i YIM klijentima. Eksploatacijom navedene ranjivosti u YIM, napadači su bili u mogućnosti da brišu fajlove zaraženog hosta [32].

Ranjivost fajla zvučnog programa ICQ klijenta se mogla koristiti za smeštanje zlonamernog sadržaja korisnički računar. Time se omogućavalo pokretanje maliciozne skripte na ugroženom računaru korišćenjem poznatih Internet Explorer ranjivosti. Bizex crv [33] je koristio ovu ranjivost da se propagira ICQ mrežom.

Dok većina poznatih ranjivosti klijenata traži neko učešće korisnika za eksploataciju hosta, MSN Messenger PNG (Portable Network Graphics) ranjivost [34] nije morala da ima interakciju korisnika da bi se eksploatisala za širenje crva. MSN Messenger koristi PNG format da bi uključila opciju "Display Picture" (koja prikazuje malu sliku kontakta u chat prozoru). Podaci slike su se slali korisniku kad god je želeo da pošalje poruku kontaktu iako slika primaoca možda ne bi bila prikazana na prozoru pošiljaoca. Tako da zloupotrebu ovog propusta, nije moglo da spreči ni isključivanje date opcije. Microsoft je ispravio ranjivost i naterao korisnike da ažuriraju IM klijenta (u suprotnom, nisu mogli da koriste servis).

Slično PNG ranjivosti, MSN Messenger ranjivost obrada slika tipa GIF je mogla da dovede do daljinskog izvršenja koda [35]. Ovakav propust je bio eksploatisan slanjem malicioznog emotikona¹⁰ ili prikazom slike na online MSN Messenger klijentu.

¹⁰ Emotikon – likovni i grafički prikaz ljudskog lica kojim se izražavaju osećaji.

4.3.2. Primeri IM crva

- **W32.Choke** crv [36] se kači na MSN Messenger, i kada korisnik inicira IM konverzaciju sa zaraženim hostom, crv šalje tekstualnu poruku sa pozivnicom za pružanje fajla sa zaraženog hosta.
- **W95.SoFunny** crv [37] se širi kao fajl-dodatak koristeći AIM. On krade AIM informacije za logovanje, i šalje korisnički ID i lozinku elektronskom poštom autoru datog crva. Prikazuje se kao servisni proces Windows operativnih sistema da bi se sakrio.
- **JS Menger** crv [38] se pojavljuje kao poruka u MSN Messenger-u sa URL-om web strane gde se hostuje kod crva. Iskorišćava slabosti Microsoft Internet Explorer pregledača.
- Različite varijacije **Bropia/Kelvir** crva su pokušale da se rašire preko online kontakata kao fajl koristeći MSN Messenger. Bropia-M [39] onemogućava nekoliko sistemskih procesa kompromitovanog hosta, uključujući Windows System Configuration Utility, Registry Editor i Task Manager. Bropia-F [40] onemogućava debugging alate inficiranog sistema, i instalira varijantu Agobot/Spybot crva [41], otvarajući backdoor inficiranih sistema.
- **Agobot** crv se propagira eksploatacijom poznatih slabosti kao što su prekoračenje bafera SQL servera, IIS (Internet Information Services) ili WebDAV (Webbased Distributed Authoring and Versioning) ranjivosti, i služi za registrovanje kucanja na tastaturi i preusmerenja spamova.
- **Serflog** varijacije [42] se šire preko MSN Messenger-a (kao URL) ili P2P mreža (npr. eMule2). Isključuje antivirus i druge poznate sistemske i bezbednosne procese, i modifikuje sistemski fajl HOSTS radi blokiranja pristupa poznatim bezbednosnim web portalima.

Većina IM crva instalira pozadinske programe zbog dalje eksploatacije zaraženih hostova. Ipak, takvi crvi dosada nisu nosili neki preterano štetan teret osim koda za njihovo množenje. Tehnike socijalnog inženjeringa se tipično koriste radi ubeđivanja korisnika da prihvati razmenu fajla sa crvom ili da poseti maliciozni URL link.

4.4. Analiza potencijalnih XMPP ranjivosti

Pre nego što se instalira XMPP aplikacija, neophodno je pratiti razvoj XMPP tehnologije u pogledu slabosti i propusta. Iako se otvorena XMPP mreža nalazi na javnom Internetu od 1999. godine, nisu uočeni neki veći bezbednosni problemi: nije bilo značajnijih DoS napada, vrlo malo spama, nije bilo fishing-a, virusa, crva lažnih servera, osim nekih manjih propusta vezano za pojedine klijente i servere. Naravno da to ne znači da neće biti incidenata u budućnosti, s obzirom da zlonamerni hakeri nikad ne spavaju.

1. TLS enkripcija je jedan od mehanizama zaštite komunikacije XMPP servera i klijenata. Međutim, čak iako korisnik kriptuje konekciju ka svom serveru, njegov kontakt kriptuje svoju konekciju ka serveru, i veza između ta dva servera se kriptuje, poruke koje se

razmenjuju među korisnicima su nezaštićene unutar samih servera. To znači da lice sa administratorskim privilegijama (legitimno ili uljez) može da ima uvid u razmenjene poruke.

Rešenje za dati problem jeste end-to-end enkripcija, gde krajnji korisnici dogovaraju metodu za enkripciju njihovih poruka. Ipak ova enkripcija je malo problematična. Jedan od open-source pristupa je Pretty Good Privacy (PGP), ili OpenPGP. Nažalost, obični krajnji korisnici nemaju PGP ključeve. Drugi pristup problemu je preporučen od strane IETF eksperata bezbednosti, a to je da XMPP koristi Secure/Multipurpose Internet Mail Extensions (S/MIME), ali ovaj pristup nikad nije prihvaćen u XMPP zajednici (svaki obični krajnji korisnik je morao imati digitalni sertifikat). Encrypted Sessions, još jedno rešenje, nije u potpunosti validirano zbog teške implementacije.

2. Autentifikacija je drugi metod zaštite, pri čemu server ne dozvoljava klijentima pristupanje XMPP mreži dok ne obezbedi legitimne parametre (korisničko ime i lozinka). Slabost ovog metoda može da predstavlja upotreba kompletnog skupa Unicode znakova za JabberID, što dovodi do mogućnosti napada lažnog predstavljanja (fishing), zato što znakovi u različitim skriptama mogu da budu slični. Primera radi, jednostavan način implementacije u ASCII adresama je zamena slova "l" sa brojem "1", kao paypal.com (što može da prevari korisnike). Kada adresa obuhvata različite znakove koji ne mogu da se jasno razlikuju (npr. latinično "a" i ćirilično "a"), to može bude potencijalna ranjivost.
3. Svaki današnji komunikacioni sistemi su pogodni za priličan broj neželjenih poruka (spamova) i drugih sličnih oblika. Pretpostavlja se da je preko 90% e-mail poruka na Internetu - spam. XMPP mreža takođe nije imuna na spamove ali u odnosu na e-mail sistem postoje razlike:
 - Smatra se da u većini slučajeva može da se šalje e-mail bilo kom korisniku na Internetu, a da pritom nema adekvatnog sistema autorizacije. Takav sistem postoji u IM (roster). U XMPP protokolu, moguće je na osnovu prisutnosti blokirati poruke korisnika koji nisu na listi kontakata.
 - Spamovanje je u nekim segmentima posao. Ukoliko je previše skupo za spamere da rade u XMPP mreži, nego u e-mail mreži, onda će da se drže e-mail mreže. Operatori nekog XMPP servisa mogu da traže digitalni sertifikat od korisnika da bi pristupio mreži što znatno otežava spamovanje.
 - XMPP serverski procesi i servisi ograničavaju korisnike da šalju veliki broj poruka za kratko vreme. Spameri mogu da zaobiđu navedeno tako što će da registruju što više naloga, ali XMPP serveri mogu da ograniče broj prijava sa određene IP adrese i opsega (tako što će registrovati crne liste domena i IP adresa) [43] itd.

5. Mehanizmi zaštite XMPP protokola

5.1. Uspostavljanje XMPP strima

Bazni oslonac XMPP protokola je XML strim, koji u klijent-server komunikaciji predstavlja sesiju. Klijentska sesija počinje konekcijom na server i uspostavljanjem sesijskih parametara, nakon čega klijent može da šalje strofe – poruka, prisutnosti i IQ drugim entitetima na mreži. Sesija se završava kada se klijent (ili server) odluči da zatvori konekciju.

Uopšteno govoreći, uspostavljanje klijent-server sesije se sastoji od sledećih faza:

1. Iniciranje TCP konekcije na server
2. Otvaranje XML strima
3. Uspostava raznih opcija strima
4. Autentifikacija sa serverom
5. Odabir resursa
6. Za IM aplikacije, zahtev za listom kontakata i slanje inicijalne prisutnosti.

Kada sesija započne, konkretna XMPP komunikacija (koja se sastoji od strofa poruke, prisutnosti, IQ) takođe startuje.

Pretpostavimo da korisnik djordje@vs.rs treba da se konektuje na server. Prvo što je neophodno da pronade je fizička mašina koja pruža XMPP servise za dati domen. XMPP omogućava po svojoj specifikaciji, funkcionisanje sopstvenog XMPP servisa za dati domen na bilo kojem hostu. Da bi korisnik našao naziv mašine koja hostuje XMPP servis, onda mora da izvrši DNS pretragu. To znači da korisnik upituje DNS SRV zapis za _xmpp-client._tcp.vs.rs. servis. Tada bi DNS odgovor bio sledeći rezultat:

```
_xmpp-client._tcp.vs.rs. 86400 IN SRV 10 20 5222 xmpp1.vs.rs  
_xmpp-client._tcp.vs.rs. 86400 IN SRV 10 5 5222 xmpp2.vs.rs
```

Iz ovog prikaza se može zaključiti da postoje dva XMPP servera dostupna vs.rs servis, xmpp1 i xmpp2, i oba su otvorena na portu 5222 (standardni port korišćen za prijem XMPP klijentskih konekcija). Da bi nastavio dalje, korisnik će uraditi sledeće korake:

1. U zavisnosti od prioriteta i važnosti zapisa, bira jednu od ove dve mašine (npr. xmpp1.vs.rs).
2. Izvršava "A" (ili "AAAA") pretragu radi utvrđivanja IP adrese izabrane mašine.
3. Otvara TCP konekciju ka datoj IP adresi i portu.

Onda XMPP uspostava počinje. Korisnički klijent šalje inicijalno zaglavlje strima (*stream header*) serveru:

```
<?xml version="1.0"?>
```

```
<stream:stream to="vs.rs"
  version="1.0"
  xmlns="jabber:client"
  xmlns:stream="http://etherx.jabber.org/streams">
```

- Ovo je XML tekstualna deklaracija i njeno ubacivanje je opcionalno pre nego što se pošalje samo zaglavlje strima.
- Elementi kvalifikovani od `http://etherx.jabber.org/streams` imenskog prostora moraju da imaju uvek prefiks *stream:* (npr. `<stream:stream>`).
- *To* atribut sadrži domenski deo korisničkog JID. U pitanju je logički naziv domena, što znači da ne more da bude isti kao naziv mašine.
- *Version* atribut pokazuje verziju XMPP protokola, i uvek je 1.0.
- *Xmlns* atribut određuje šablonski (default) imenski prostor za sve XML-ove koji idu preko strima. S obzirom da je ovo klijent-server strim, šablonski imenski prostor je `jabber:client`.
- `<stream/>` element nije zatvoren. U ovom trenutku klijent šalje samo nalepnicu za otvaranje da bi otpočeo strim, tj. `<stream:stream>`. Od sada pa nadalje, svi podređeni XML elementi se šalju kao deca ovog root elementa, dok se sesija ne završi zatvaranjem root elementa sa završnom `</stream:stream>` nalepnicom.

Kao odgovor, server otvara odzivni strim (*response stream*) u suprotnom smeru, šaljući uvodnu `<stream:stream>` nalepnicu korisniku (postoji jedan strim, razmena podataka je u dva smera):

```
<?xml version="1.0"?>
<stream:stream from="vs.rs"
  id="l8d1m79pk53ht"
  version="1.0"
  xmlns="jabber:client"
  xmlns:stream="http://etherx.jabber.org/streams">
```

- Pre nego što pošalje odzivno zaglavlje strima, server generiše jedinstveni ID strima za sesiju.

Odmah nakon otvaranja odzivnog strima, server pokušava da se usaglasi sa klijentom o daljem statusu i postupanju konekcije. Prvo, server obaveštava klijenta o funkcionalnostima strima kojem daje podršku:

```
<stream:features>
  <starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls">
    <optional/>
  </starttls>
  <mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl">
    <mechanism>PLAIN</mechanism>
    <mechanism>DIGEST-MD5</mechanism>
    <required/>
  </mechanisms>
  <compression xmlns="http://jabber.org/features/compress">
    <method>zlib</method>
  </compression>
</stream:features>
```

U ovom konkretnom primeru, server ima podršku sledećih funkcionalnosti:

- Enkriptovane konekcije sa XMPP profilom *Transport Layer Security (TLS)* [44]. Server pokazuje da je TLS uspostava opcionalna, ali je generalno preporučljiva za sve XML strimove.
- Autentifikacija putem *Simple Authentication and Security Layer (SASL)* [45]. U ovom slučaju, jedine podržane metode autentifikacije su PLAIN i DIGEST-MD5 mehanizmi. Server pokazuje da je SASL uspostava neophodna.
- Kompresija strima za više optimalnih upotreba propusnog opsega [46]; I ovo je opcionalna funkcionalnost (server nije naveo `<optional/>` element-dete, ali se podrazumeva).

Klijent može dalje da uspostavi enkriptovanu i kompresovanu konekciju, nakon toga se autentifikuje na XMPP server sa jednim od ova dva mehanizma i potražuje resurse od servera.

5.2. Opcije autentifikacije (SASL)

Jedna od prednosti SASL radnog okvira (okruženja) je u tome što omogućava programerima i administratorima podršku različitim mehanizmima autentifikacije u zavisnosti od potreba. Neki od SASL mehanizama koji se trenutno koriste ili su planirani za buduću implementaciju u XMPP su sledeći:

- *PLAIN*
PLAIN mehanizam [47] pruža veoma jednostavan metod autentifikacije baziran na lozinkama. S obzirom da se lozinka šalje bez bezbednosne zaštite (u čistom tekstu), PLAIN je jedino može koristiti ukoliko je sam XML strim zaštićen upotrebom Transport Layer Security (TLS).
- *DIGEST-MD5*
DIGEST-MD5 mehanizam [48] pruža bolju zaštitu u odnosu na PLAIN mehanizam, jer se lozinka kriptuje prilikom prenosa (tako da nije neophodan TLS za samu lozinku). Međutim, mnogi problemi u interoperabilnosti su se pojavili kod DIGEST-MD5, ne samo u XMPP-u, nego i drugim sistemima, kao što je IMAP. IETF je zbog toga prevazišao DIGEST-MD5 sa SCRAM mehanizmom.
- *SCRAM*
SCRAM (Salted Challenge Response Authentication Mechanism) predložena zamena za DIGEST-MD5 od strane IETF. Mehanizam omogućava prilično jaku zaštitu za brojne bezbednosne pretnje kada se koristi u konjukciji sa TLS.
- *EXTERNAL*
EXTERNAL mehanizam omogućava inicirajućem entitetu (bilo klijentu ili serveru) da prezentuje digitalni sertifikat koristeći TLS uspostavu (pregovaranje) i da se referencira na dati sertifikat prilikom autentifikacije, smanjujući potrebu za lozinkama sveukupno. Ipak, nema mnogo krajnjih korisnika van određene specijalizovanih okruženja koji imaju digitalne sertifikate, pa se u praksi EXTERNAL mehanizam koristi više za server-server nego za klijent-server konekcije.

- *GSSAPI*
GSSAPI mehanizam [49] je prihvaćen kao opšti SASL mehanizam (priključenje unutar priklučenja!), ali do danas je upotrebljen samo u kombinaciji sa Kerberos V. GSSAPI mehanizam je još manje rasprostranjen od EXTERNAL mehanizma, ali je mnogo moćan kod organizacija koje koriste Kerberos.
- *ANONYMOUS*
ANONYMOUS mehanizam [50] omogućava korisnicima da se autentifikuju bez registracije naloga na serveru. Ovaj mehanizam je najkorisniji za aplikacije sa direktnim pristupom za korisnike, kao što centri za pozivanje (call centres) i online pomoćni sistemi, gde registracija nema smisla za jednokratnu upotrebu XMPP aplikacije.

Neke od ovih metoda mogu da rezultuju u zajedničkoj sesiji autentifikacije aktivirajući bezbednosni sloj, čime omogućuju zaštitu integriteta i enkripciju. Ipak, ovaj aspekt SASL nije zastupljen u široj upotrebi sa svim svojim mogućnostima.

5.3. Enkripcija konekcije

Povezivanje na XMPP server koristeći standardnu TCP konekciju predstavlja nedostatak zbog slabe zaštite. Samim tim, onaj koji prisluškuje i prati saobraćaj na mreži može da presretne sve što se šalje na server, uključujući lične poruke, i lozinku za prijavljivanje. Autentifikacija može da se obezbedi mehanizmom koji šalje lozinku na takav način da ne može da se rekonstruiše iz saobraćaja zbog enkripcije (npr. DIGEST-MD5). Ipak, u ovom slučaju ostaje nezaštićena preostala komunikacija sa serverom.

Još jedan problem može da predstavlja i to da li uopšte klijent šalje poruku pravom serveru, tj. maliciozni entitet može da preusmeri saobraćaj ka svom hostu, prikazivajući sebe kao pravog servera što može da rezultira u odavanju bitnih podataka samog korisnika.

Za navedene probleme je ponuđeno rešenje u vidu protokola Secure Sockets Layer (SSL) koji se naziva danas *Transport Layer Security*. Kao i većina Internet protokola koji koriste SSL ili TLS, XMPP takođe koristi TLS za bezbednost konekcija. Upotrebu TLS protokola u XMPP-u i samo funkcionisanje je prikazano kroz sledeći primer.

Pretpostavimo da je na vs.rs server instalirana nova sigurnosna polisa. Kada korisnik započinje sesiju, server mu odgovara sa strimom:

```
<stream:stream from="vs.rs" id="98hihht0n4"
version="1.0"
xmlns="jabber:client"
xmlns:stream="http://etherx.jabber.org/streams">
<stream:features>
<starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls">
<required/>
</starttls>
</stream:features>
```

Iz date XML strofe se vidi da server vs.rs pokazuje klijentu da ima podršku za sigurne konekcije tipa TLS, i da zahteva enkripciju. Pored toga, u okviru <starttls/> elementa,

server nije objavio podršku za neki mehanizam autentifikacije, tako da korisnik mora da prihvati konekciju sa enkripcijom:

```
<starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls"/>
```

Ovaj zahtev dobija odgovor od servera da će biti uspostavljena TLS konekcija:

```
<proceed xmlns="urn:ietf:params:xml:ns:xmpp-tls"/>
```

Nakon elementa `<proceed/>`, regularni XMPP strim se završava, i počinje TLS uspostava (handshake) preko same TCP konekcije. Ovaj korak obuhvata slanje sertifikata od strane servera klijentu, i sa tim klijent validira sertifikat čime se uverava da je dati server pravi. Uspostava se realizuje direktno putem TCP konekcije koristeći standardne TLS poruke (a ne XML enkapsulaciju TLS-a), što je i definisano TLS specifikacijom. Nakon uspešnom povezivanja, klijent restartuje svoju XMPP sesiju, i onda počinje da kriptuje sve što šalje koristeći ključ dobijen od servera:

```
<stream:stream to="vs.rs"  
version="1.0"  
xmlns="jabber:client"  
xmlns:stream="http://etherx.jabber.org/streams">
```

Sad kada je obezbeđena sigurna konekcija, server odgovara sa više opcija strima, kao što su mehanizmi autentifikacije, dozvoljavajući klijentu da počne autentifikovanje svoje XMPP sesije:

```
<stream:stream from="vs.rs"  
id="c8b7rn6p6"  
version="1.0"  
xmlns="jabber:client"  
xmlns:stream="http://etherx.jabber.org/streams">  
<stream:features>  
<mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl">  
<mechanism>PLAIN</mechanism>  
<mechanism>DIGEST-MD5</mechanism>  
</mechanisms>  
</stream:features>
```

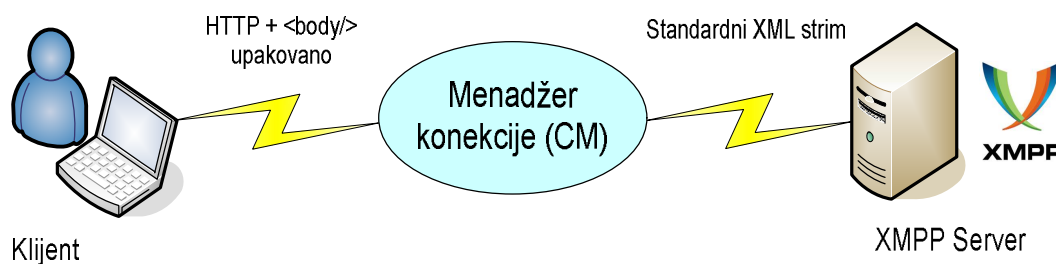
5.4. BOSH

Klijent-server i server-server konekcije uglavnom podrazumevaju dugotrajne TCP konekcije, koje su otvorene dokle god je potrebno. U pojedinim situacijama, koje nisu retke, teško je održavati dugotrajne TCP konekcije zbog više razloga:

- Nestabilne mreže često mogu da prekidaju TCP konekcije, što za posledicu ima to da klijent pravi mnogo pokušaja kreiranja XMPP sesije, uspostavom zaglavlja strima, enkripcije, autentifikacije, povezivanje resursa, traženjem liste korisnika, i inicijalne prisutnosti.
- Na prenosnom uređaju, konstantna upotreba dugotrajnih TCP konekcija prilično brzo troši bateriju.
- pojedine web-bazirane aplikacije ne mogu da održavaju dugotrajne konekcije.
- Ograničavajuća podešavanja firewall-a može da spreči i standardnog XMPP klijenta da otvori regularnu TCP konekciju ka XMPP serveru itd.

Rešenje za ovakav problem je u načinu konekcije koji ne zahteva od klijenta da održava dugotrajne TCP konekcije. Standardan metod je emulacija dvosmernih strimova koji su poznati od TCP povezivanja, ali sa efikasnim korišćenjem višestrukih, sinhronizovanih HTTP zahtev/odziv parova. Ovakav metod se naziva Bidirectionalstreams Over Synchronous HTTP, ili skraćeno BOSH. [51].

BOSH-bazirani sistemi se uglavnom instaliraju sa posebnim menadžerom konekcije (CM) koji predstavlja posrednu komponentu (proxy) između klijenta i XMPP servera (Slika 9). U ovom slučaju server obuhvata standardni XMPP proces (daemon), a menadžer konekcije se povezuje XMPP protokolom ka serveru, HTTP protokolom ka klijentu. Klijent se ponaša kao hibridni XMPP+HTTP agent koji funkcioniše kao XMPP klijent iz korisničke perspektive, ali tokom komunikacije sa menadžerom konekcije sav XMPP saobraćaj uokviruje <body/> elementom HTTP-a.



Slika 9.

5.4.1. Funkcionisanje BOSH

Ukoliko dati XMPP servis obuhvata i BOSH menadžer konekcije, klijent može na dva načina da mu pristupi: manuelno da konfiguriše klijentsku aplikaciju ili da pokrene servis pretraživanja (što je i uobičajeno iz perspektive običnog korisnika). Kada se pokrene pretraga, klijent šalje DNS upit za TXT zapise vezane traženim domenom. U DNS odgovoru dobija sledeću formu zapisa[52]:

```
_xmppconnect IN TXT "_xmpp-client-xbosh="https://bosh.domen.com/webclient"
```

Pokretanje XMPP sesije sa XMPP serverom se realizuje slanjem inicijalnog HTTP zahteva ka linku `https://bosh.domen.com/webclient` (u narednim primerima `vs.rs`). Zahtev se sastoji od HTTP POST paketa i sadrži `<body/>` element; koji je podrazumevano prazan, ali obuhvata određene attribute koji se koriste za uspostavu BOSH sesije.

```
POST /webclient HTTP/1.1
Host: httpcm.jabber.org
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=utf-8
Content-Length: 164

<body content='text/xml; charset=utf-8'
  from='djordje@vs.rs'
  hold='1'
  rid='3346378901'
  to='vs.rs'
  route='xmpp:vs.rs:9999'
  wait='60'
  xml:lang='en'
  xmpp:version='1.0'
  xmlns='http://jabber.org/protocol/httpbind'
  xmlns:xmpp='urn:xmpp:xbosh' />
```

Hold atribut definiše maksimalan broj HTTP zahteva za potraživanje koji je dozvoljen BOSH menadžeru konekcije radi slanja poruka za klijenta u jednom trenutku. Podrazumevana vrednost ovog atributa je 1.

Prilikom kreiranja BOSH sesije, klijent mora da generiše veliki, proizvoljan broj koji ima ulogu inicijalnog ID zahteva. Svako slanje zahteva menadžeru konekcije povećava *rid* atribut za 1.

Wait atribut maksimalno vreme čekanja (u sekundama) menadžera konekcije na odziv i predstavlja vreme čekanja HTTP/TCP konekcije.

BOSH menadžer konekcije dalje otvara regularnu XMPP konekciju ka `vs.rs` i prima XMPP odgovor koji se dalje prosleđuje korisniku kao HTTP odgovor:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 582

<body wait='60'
  inactivity='30'
  polling='5'
```

```

    requests='2'
    hold='1'
    from='vs.rs'
    accept='deflate,gzip'
    sid='SomeSID'
    charsets='ISO_8859-1 ISO-2022-JP'
    xmpp:restartlogic='true'
    xmpp:version='1.0'
    authid='ServerStreamID'
    xmlns='http://jabber.org/protocol/httpbind'
    xmlns:xmpp='urn:xmpp:xbosh'
    xmlns:stream='http://etherx.jabber.org/streams'
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    <mechanism>SCRAM-SHA-1</mechanism>
    <mechanism>PLAIN</mechanism>
  </mechanisms>
</stream:features>
</body>

```

Requests atribut definiše maksimalan broj HTTP zahteva za potraživanje koji je dozvoljen klijentu da otvara ka menadžeru konekcije u jednom trenutku. Podrazumevana vrednost datog atributa je 2.

Kada BOSH menadžer konekcije odgovori na zahtev klijenta za kreiranje sesije, onda generiše jedinstveni ID sesije (Session ID), koji je različit u odnosu na ID strima (Stream ID) generisan od strane XMPP servera. Svaki dalji zahtevi i odgovori moraju da imaju *sid* atribut. Autentifikacija sa serverom se realizuje slanjem drugog HTTP zahteva čiji <body/> element sadrži prvi korak SASL autentifikacije:

```

POST /webclient HTTP/1.1
Host: httpcm.vs.rs
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=utf-8
Content-Length: 237

<body rid='1573741821'
  sid='SomeSID'
  xmlns='http://jabber.org/protocol/httpbind'>
  <auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
    mechanism='SCRAM-SHA-1'>
    biwsbj1qdWxpZXQscjlvTXNUQUF3QUFBQU1BQUFBTlAwVEFBQUFBQUJQVTBBQQ==
  </auth>
</body>

```

Uspešna autentifikacija klijenta sa serverom se izvršava u još tri koraka:

Drugi korak

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 284

<body xmlns='http://jabber.org/protocol/httpbind'>
  <challenge xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>

```

```

    cjlvtXNUQUF3QUFBQU1BQUFBTLAwVEFBQUFBQUJQVTBBQUxMjQ2OTViLTY5Y
    TktNGRlNi05YzMwLWI1MWIzODA4YzU5ZSxzPU5qaGtZVE0wTURndE5HWTBaaT
    AwTmPkUxUa3hNbVV0TkRsbU5UTm1ORE5rTURNeixpPTQwOTY=
  </challenge>
</body>

```

Treći korak

```

POST /webclient HTTP/1.1
Host: httpcm.vs.rs
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=utf-8
Content-Length: 295

<body rid='1573741822'
  sid='SomeSID'
  xmlns='http://jabber.org/protocol/httpbind'>
  <response xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    YzliaXdzLHI9b0lzVEFBd0FBQUFNQUFBQU5QMFRBQUFBQUFCUFUwQUFlMTI0N
    jklYi02OWE5LTRkZTYtOWMzMCIiNTFiMzgwOGM1OWUscD1VQTU3dE0vU3ZwQV
    RCa0gyRlhzMfDEWHZKWXc9
  </response>
</body>

```

Četvrti korak

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 152

<body xmlns='http://jabber.org/protocol/httpbind'>
  <success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    djlwTk5ERlZFUXhlWHhDb1NFaVc4R0VaKzFSU289
  </success>
</body>

```

Nakon uspešne SASL autentifikacije, klijent mora da restartuje strim, slanjem posebnog zahteva:

```

POST /webclient HTTP/1.1
Content-Type: text/xml; charset=utf-8
Content-Length: 240

<body rid='1573741824'
  sid='SomeSID'
  to='vs.rs'
  xml:lang='en'
  xmpp:restart='true'
  xmlns='http://jabber.org/protocol/httpbind'
  xmlns:xmpp='urn:xmpp:xbosh' />

```

Xmpp:restart atribut ukazuje BOSH menadžeru konekcije da restartuje XML strim koji je otvoren sa XMPP serverom u ime klijenta. Prema tome, BOSH CM restartuje strim sa XMPP serverom i vraća sledeće:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 221

<body xmlns='http://jabber.org/protocol/httpbind'
      xmlns:stream='http://etherx.jabber.org/streams'>
  <stream:features>
    <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'/>
  </stream:features>
</body>
```

Klijent nakon ponovne uspostave, traži resurse XMPP servera:

```
POST /webclient HTTP/1.1
Content-Type: text/xml; charset=utf-8
Content-Length: 242

<body rid='1573741825'
      sid='SomeSID'
      xmlns='http://jabber.org/protocol/httpbind'>
  <iq id='bind_1'
    type='set'
    xmlns='jabber:client'>
    <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
      <resource>httpclient</resource>
    </bind>
  </iq>
</body>
```

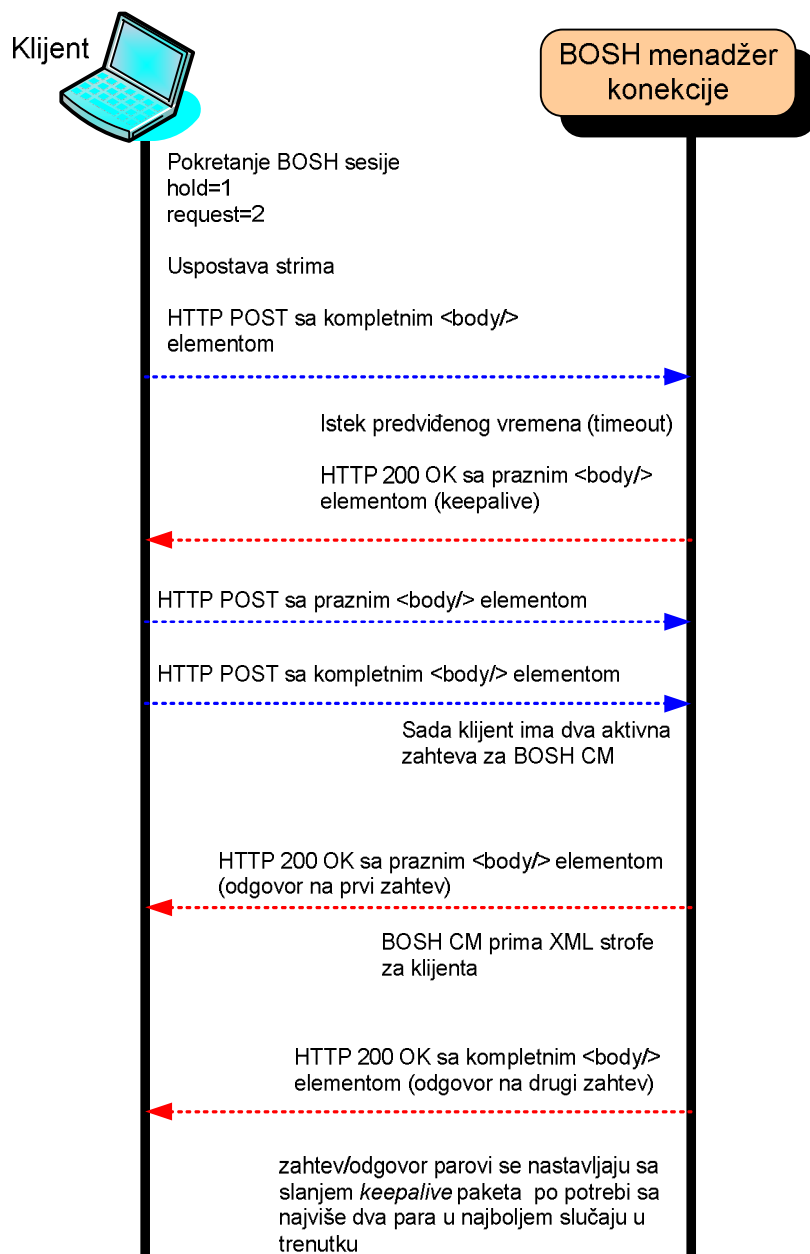
IQ strofa uključuje deklaraciju imenskog prostora zbog toga što je podrazumevani imenski prostor za BOSH `http://jabber.org/protocol/httpbind` umesto podrazumevanog imenskog prostora za klijent-server XMPP strim (npr. `jabber:client`). Kao odgovor, menadžer konekcije vraća vezani rezultat kojeg je primio od XMPP servera:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 221

<body xmlns='http://jabber.org/protocol/httpbind'>
  <iq id='bind_1'
    type='result'
    xmlns='jabber:client'>
    <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
      <jid>user@example.com/httpclient</jid>
    </bind>
  </iq>
</body>
```

Upotreba BOSH dolazi do izražaja u slučaju kada korisnik ne šalje ili ne primi XML strofu u vremenskom periodu dužem od vremena isteka (timeout). Tada menadžer konekcije šalje

pakete korisniku ukoliko je ističe *timeout* period, ili vraća prazan `<body/>` element korisniku (ukoliko je poslao više HTTP zahteva u trenutku).



Slika 10.

Kao što se može videti iz Slike 10, kada imamo dva zahtev/odgovor para u izuzetnom slučaju u bilo kojem jednom trenutku, i ugrađene nadolazeće XMPP strofe u odgovorima na HTTP zahteve koje je klijent poslao, BOSH može lako da mapira XMPP dvosmerne strimove u HTTP semantiku. Time se postiže efikasnost, a bez forsiranja klijenta potražuje menadžera konekcije. Stepenn efikasnosti BOSH zavisi od vrednosti *hold*, *requests*, i *wait* atributa.

5.5. Slanje poruka bez servera

Kao što je ranije pomenuto, XMPP je zasnovan na klijent-server arhitekturi, što znači da je uvek neophodan server kao posrednik komunikacije između korisnika. Međutim, u pojedinim situacijama, konekcija na XMPP server može biti problematična:

- Korisnik može da se nalazi na udaljenoj lokaciji bez ili sa jakom lošom Internet konekcijom (seosko područje i sl.), a neophodan mu je XMPP za komuniciranje sa ljudima u okolini.
- Korisnik može da bude na međunarodnoj konferenciji gde ima potrebe za tematskom raspravom sa ljudima u prostoriji, ali ne zna njihov JabberID.
- Korisnik nema pristup, zbog toga što je blokiran sav izlazni saobraćaj ka Internetu itd.

Kompanija Apple Computer je definisala skup tehnologija poznatih kao "zero-configuration networking" ili "umrežavanje nulte-konfiguracije". Ove tehnologije su primenjene u komunikacijama u realnom vremenu koristeći XMPP, i to je standardizovano od strane XSF pod nazivom *Serverless Messaging* [53].

Ukoliko postoji lokalna računarska mreža, korisnik može da izvrši pretragu za ostalim korisnicima. Osnovni pristup je da će svaki klijent slati multicast posebnih DNS zapisa ka svima u lokalnoj mreži, što rezultuje time da će svi kolektivno upravljati rezervisanim top-level domenom .local.

Da bi korisnik našao svoje prijatelje na mreži, mora da pusti broadcast multicast-a DNS upita za sve PTR zapise koji se poklapaju sa _presence._tcp.local. servisom. Odgovor bi izgledao kao sledeći zapisi

```
_presence._tcp.local. PTR ognjen@refzinf._presence._tcp.local.  
_presence._tcp.local. PTR aleksandar@refis._presence._tcp.local.
```

U ovom primeru, korisnik nalazi dva druga korisnika na lokalnoj mreži: ognjen (naziv računara je refzinf) i aleksandar (naziv računara refis). U slučaju da korisnik (u cilju daljeg objasnjenja kroz primer, korisnik će biti djordje sa nazivom računara refzs) planira da ostvari chat komunikaciju sa prethodno navedenim korisnicima, treba mu ime hosta i broj porta njihovih računara. Onda šalje broadcast sa mDNS upitom, tražeći SRV zapise servisa prisutnosti, da bi mu se vratilo kao odgovor:

```
ognjen@refzinf._presence._tcp.local. SRV 5562 refzinf.local.  
aleksandar@refis._presence._tcp.local. SRV 5562 refis.local.
```

Naravno, korisnik djordje je sada upoznat da XMPP klijenti ognjen@refzinf i aleksandar@refis slušaju na portu 5562 hostova refzinf.local i refis.local. Poslednje što je preostalo su IP adrese ovih hostova. Slanjem broadcast poruke mDNS upita za A (ili za IPv6, AAAA) DNS zapisa hostova, dobija sledeće:

```
refzinf.local. A 10.6.1.76  
refis.local A 10.6.1.30
```

Sa svim neophodnim poznatim parametrima korisnik djordje može da otvori TCP konekciju prema ovim korisnicima:

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  from='djordje@refzs'
  to='ognjen@refzinf'
  version='1.0'>
```

Odgovor:

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  from='ognjen@refzinf'
  to='djordje@refzs'
  version='1.0'>
```

Dalje primalac šalje karakteristike strima:

```
<stream:features>
  <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
    <optional/>
  </starttls>
  <query xmlns='http://jabber.org/protocol/disco#info'
node='http://code.google.com/p/exodus#QgayPKawpkPSDYmwT/WM94uAlu0='>
  <identity category='client' name='Exodus 0.9.1' type='pc'/>
  <feature var='http://jabber.org/protocol/caps'/>
  <feature var='http://jabber.org/protocol/disco#info'/>
  <feature var='http://jabber.org/protocol/disco#items'/>
  <feature var='http://jabber.org/protocol/muc'/>
  </query>
</stream:features>
```

Nakon ove razmene, korisnik može da šalje <message/> i <iq/> strofe, kao normalnu XMPP komunikaciju.

Ako želi da sazna detaljnije informacije o korisnicima s kojima komunicira, dati podaci se nalaze u mDNS zapisima, konkretno u DNS TXT zapisima koji sadrže varijetet ključ-vrednost parova.

```
ognjen@refzinf._presence._tcp.local. IN TXT "
txtvers=1 (1)
status=avail (2)
msg=Citam neku knjigu (3)
lst=Ognjen
last=Mojovic
nick=mojke
node=http://psi-im.org/caps (4)
ver=QgayPKawpkPSDYmwT/WM94uAlu0= (5)
```

hash=sha-1

(6)

1. Ovaj broj je uvek 1, i *txtvers* polje mora da bude prvo u vrednosti TXT zapisa.
2. Ovo je ekvivalentno <show/> elementu standardne XMPP prisutnosti, sa izuzetkom što je samo status dostupnog mapiran u TXT prikazu (*avail*).
3. Ovo je ekvivalentno <status/> elementu standardne XMPP prisutnosti.
4. Ovo je ekvivalentno *node* atributu iz XMPP dodatka mogućnosti entiteta (Entity Capabilities).
5. Ovo je ekvivalentno *ver* atributu iz XMPP dodatka mogućnosti entiteta.
6. Ovo je ekvivalentno *hash* atributu iz XMPP dodatka mogućnosti entiteta.

Korisnik djordje ukoliko želi da oglasi svog XMPP klijenta ostalim drugim klijentima, može da objavi iste tipove mDNS i DNS-SD zapisa koje je koristio za otkrivanje drugih klijenata:

```
refzs.local. A 10.6.61
_presence.tcp.local. PTR djordje@refzs._presence._tcp.local.
djordje@refzs._presence._tcp.local. SRV 5562 refzs.local.
djordje@refzs._presence._tcp.local. IN TXT "txtvers=1
      status=avail
      lst=Djordje"
```

S obzirom na to da XMPP koristi TLS za enkripciju komunikacije, SASL za autentifikaciju, DNS za slabu validaciju imena servera, ove tehnologije omogućavaju sigurnost identiteta korisničkih entiteta enkripcijom XML strimova. Nasuprot tome, umrežavanje nulte-konfiguracije koristi dinamičku pretragu potvrđene nazive računara kao osnovu identiteta pošiljaoca. Samim tim, slanje poruka bez servera nema autentifikovane identitete kao XMPP, niti omogućava enkripciju.

Da bi osigurali entitete, preporučuje se upotreba TLS i SASL za XML strim kao što je opisano u dokumentu RFC 6120. Ali pošto u ovakvoj situaciji nema servera (gde je TLS i SASL obavezan kao bezbednosni mehanizam), primena bezbednosnih parametara zavisi od lokalne politike klijenta

6. OPENFIRE

Openfire (prethodno poznat kao **Wildfire**, i **Jive Messenger**) je server za IM i grupni chat koji je baziran na XMPP serveru napisanom u Java programskom jeziku i licenciran sa Apache License 2.0 licencom [54]. Nastao je kao proizvod open-source zajednice programera i običnih korisnika koja se naziva Ignite Realtime. Poslednja verzija servera je 3.10.2 [55].

Openfire sadrži sledeće funkcionalnosti:

- web-bazirani interfejs za administraciju i konfiguraciju,
- instalaciju dodataka (plugins),
- poseduje prilagodljivo okruženje,
- podržava implementaciju SSL/TLS enkripcije,
- prilično jednostavnu instalaciju sa prostim interfejsom,
- konektivnost sa bazama (ugrađena HSQLDB ili neka druga DBMS sa JDBC 3 drajverom) radi lakšeg skladištenja poruka i informacija o korisnicima,
- uvezivanje sa LDAP
- nezavisnu platformu, u potpunosti Java,
- kompletna integracija i kompatibilnost sa Spark XMPP klijentom,
- podržava više od 50,000 istovremeno prisutnih korisnika.

6.1. Instalacija

Osnovna postavka IM i chat servera Openfire, obuhvata nekoliko faza:

- pokretanje i instalacija serverske aplikacije na operativni sistem,
- instalacija i povezivanje serverske aplikacije sa bazom podataka (eksternog tipa),
- inicijalna konfiguracija servera.

Instalacioni program je prilagođen za MS Windows i Linux operativne sisteme (rpm i tar.gz instalacioni paketi za Linux). Struktura instaliranog Openfire programa je sledeća:

```
openfire/  
|- readme.html  
|- license.html  
|- conf/  
|- bin/  
|- jre/  
|- lib/  
|- plugins/  
    |- admin/  
|- resources/  
    |-database/  
    |-security/  
|- documentation/
```

- **conf** direktorijum sadrži konfiguracione fajlove.
- **bin** direktorijum sadrži izvršne serverske fajlove, razlikuju se u zavisnosti od verzije (distribucije) operativnog sistema.
- **jre** direktorijum sadrži Java pokretačko okruženje koje je ugrađeno u Windows i RPM verzijama Openfire aplikacije.
- **lib** direktorijum sadrži programske biblioteke, neophodne za funkcionisanje Openfire servera.
- **plugins** direktorijum sadrži serverske dodatke, osnovni dodatak je web-bazirani administratorski konzolni dodatak.
- **resources/database** direktorijum sadrži SQL organizacione fajlove zbog kreiranja Openfire baza podataka, i nadogradnje postojećih instalacija sa skriptovima.
- **resources/security** je direktorijum gde Openfire skladišti i koristi ključeve za podršku TLS konekcije.
- **documentation** direktorijum sadrži serversku dokumentaciju.

Openfire skladišti podatke na dva načina: u već ugrađenu bazu serverske aplikacije (ograničenih mogućnosti) ili koristeći eksterne baze podataka tipa MySQL, MSSQL и Oracle.

Web-bazirani čarobnjak i konfiguracioni alat je implementiran u Openfire instalaciji. Jednostavno pokretanje Openfire i upotreba web pregledača inicira administratorsku konzolu. Podrazumevani port za web administraciju i konfiguraciju je 9090 za nezaštićenu konekciju, odnosno 9091 zaštićenu konekciju (SSL/TLS).

Pored osnovne instalacije moguće je instalirati i razne dodatke, koju pružaju dodatne opcije i funkcionalnosti drugih protokola.

6.2. Povezivanje eksterne baze podataka sa serverom

Openfire skladišti sve podatke u back-end bazu podataka. Ukoliko ne izaberemo ugrađenu bazu, onda je neophodno instaliranje i podešavanje eksterne baze podataka za aplikaciju. JDBC 2.0 drajveri su potrebni za komunikaciju Openfire serverske komponente sa bazom podataka. Openfire instalacioni paket sadrži JDBC drajvere za MySQL, Postgres, Microsoft SQL Server i HSQLDB.

Sve šeme baza podataka se nalaze u direktorijumu `resources/database` Openfire instalacije. Podržani su i drugi tipovi baza pored već navedenih.

6.2.1. MySQL

U alatima za podešavanje Openfire-a, treba koristiti i prepodesiti sledeće parametre:

- `driver: com.mysql.jdbc.Driver`
- `server: jdbc:mysql://[YOUR_HOST]/[DATABASE_NAME]`

pri čemu su `[YOUR_HOST]` i `[DATABASE_NAME]` parametri servera. U većini slučajeva, server i baza podataka su na istom mestu, pa se koristi `localhost` kao parametar za `[YOUR_HOST]`.

Instrukcije za podešavanje se sastoje od nekoliko koraka:

1. verzija MySQL mora bude najmanje 4.1.18 ili novija (5.x preporučena),
2. kreiranje baze za Openfire tabele:
`mysqladmin create [databaseName]`
3. ubacivanje šematskog fajla iz direktorijuma `resources/database`:
Unix/Linux: `cat openfire_mysql.sql | mysql [databaseName];`
Windows: `type openfire_mysql.sql | mysql [databaseName];`
4. pokretanje Openfire alata za podešavanje, i upotreba odgovarajuće JDBC konekcije.

6.2.2. Oracle

U alatima za podešavanje Openfire-a, treba koristiti i prepodesiti sledeće parametre:

- `driver: oracle.jdbc.driver.OracleDriver`
- `server: jdbc:oracle:thin:@[HOST]:[PORT]/[SERVICE]`

pri čemu su `[HOST]`, `[PORT]` i `[SERVICE]` parametri servera. U većini slučajeva se koriste `localhost` i `1521` kao odgovarajuće vrednosti za `[HOST]` i `[PORT]` kada su baza podataka i server na istom hostu.

Nakon toga poželjno je napraviti "Jive" korisnički nalog ili bilo koji drugi nalog koji će posedovati i raspolagati sa Jive tabelama.

Sledeće je ubacivanje šematskog fajla iz direktorijuma `resources/database` instalacije, koristeći `sqlplus`:

1. kopiranjem "openfire_oracle.sql" fajla u instalirani folder `sqlplus-a` (`/Oracle/Ora81/bin/`),
2. prijavom na `sqlplus` i izvršavanjem komande `@ openfire_oracle`.

6.2.3. SQL Server

Openfire uključuje Free TDS JDBC drajver, koji je preporučen za konekcije sa SQL Server bazama podataka. Da bi se konektovali na SQL Server, u alatima za podešavanje Openfire-a, treba koristiti i prepodesiti sledeće parametre

- `driver: net.sourceforge.jtds.jdbc.Driver`
- `server: jdbc:jtds:sqlserver://[host-name]:[port-number]/[database-name];appName=jive`

gde je `[host-name]` naziv hosta ili IP adresa servera, `[port-number]` je broj porta koji SQL Server koristi (obično 1433) i `[database-name]` je ime baze (nije obavezan parametar). Instrukcije za podešavanje su sledeće:

1. prvo treba napraviti praznu bazu preko Enterprise Manager-a (ukoliko ne postoji nikakva baza),
2. kreiranje korisničkog naloga za zaseban pristup bazi podataka (ukoliko ne postoji),

3. pokrenuti Query Analyser izvršiti konekciju na server,
4. potom se selektuje baza namenjena za Openfire,
5. pokretanje `openfire_sqlserver.sql` datoteke,
6. pokretanje skripte kojom će se kreirati neophodne tabele i
7. u alatima za podešavanje Openfire-a upotrebiti adekvatna JDBC podešavanja.

6.2.4. PostgreSQL

U alatima za podešavanje Openfire-a, treba koristiti i prepodesiti sledeće parametre:

- `driver: org.postgresql.Driver`
- `server: jdbc:postgresql://[HOST_NAME]:[PORT_NUMBER]/dbname`

Ukoliko se izostavi, podrazumevani host je localhost (ne 127.0.0.1) i broj porta je 5432. Instrukcije za podešavanje su sledeće:

1. kreirati bazu podataka, upotrebom PostgreSQL "createdb" opcije:
`createdb -E UNICODE openfire`
2. ubaciti šematski fajl iz direktorijuma `resources/database` korišćenjem `psql` funkcije:
`psql -d openfire -f openfire_postgresql.sql`
3. u alatima za podešavanje Openfire-a upotrebiti adekvatna JDBC podešavanja.

6.3. Integracija baze korisnika

Integracija zahteva unošenje zasebnih upita radi pristupa datoj bazi podataka, što znači da administrator mora da zna tabelarnu strukturu baze i prosti SQL jezik. Posebno napravljena baza može da bude drugačija i na drugačijem serveru od Openfire baze podataka – zato se unosi informacija o konekciji na bazu podataka u sklopu konfiguracije. Da bi konfigurisali server za integrisanje sa drugim posebnim baznim tabelama, neophodno je:

1. stopirati Openfire,
2. izmeniti fajl `conf/openfire.xml` u Openfire instaliranom direktorijumu,
3. restartovati Openfire.

6.3.1. Podešavanja konekcije sa bazom podataka

Pre svega mora da se odredi tačan zapis za bazu podataka i JDBC drajver.

- `jdbcProvider.driver` – naziv klase JDBC drajvera koji se koristi za konekciju na datu bazu. Drajver mora da bude takođe u klasnoj putanji Openfire (`lib/` direktorijum).
- `jdbcProvider.connectionString` – pun konekcionis tring za bazu podataka.

Primer malog segmenta konfiguracionog fajla:

```
<jive>
...
<jdbcProvider>
  <driver>com.mysql.jdbc.Driver</driver>
```

```

<connectionString>jdbc:mysql://localhost/dbname?user=username&password=secret</connectionString>
  </jdbcProvider>
  ...
</jive>

```

6.3.2. Integracija autentifikacije

Najjednostavnija integracija sa zasebnom spoljnom bazom podataka je integracija autentifikacije. Za datu integraciju se koriste sledeći parametri:

- `provider.auth.className` – vrednost treba promeniti u:
`org.jivesoftware.openfire.auth.JDBCAuthProvider`.
- `jdbcAuthProvider.passwordSQL` – SQL string za selektovanje korisničke lozinke. SQL izjava treba da sadrži jedan "?" znak, koji će se dinamički zameniti korisničkim imenom prilikom izvršavanja.
- `jdbcAuthProvider.passwordType` – tip lozinke. Validne vrednosti su:
 - "plain",
 - "md5",
 - "sha1",
 - "sha256",
 - "sha512".

Ukoliko vrednost nije upisana, podrazumevani tip lozinke je plain.

Primer malog segmenta konfiguracionog fajla:

```

<jive>
  ...
  <provider>
    <auth>

      <className>org.jivesoftware.openfire.auth.JDBCAuthProvider</className>
    </auth>
  </provider>
  <jdbcAuthProvider>
    <passwordSQL>SELECT password FROM user_account WHERE
username=?</passwordSQL>
    <passwordType>plain</passwordType>
  </jdbcAuthProvider>
  ...
</jive>

```

Prijavljivanje na administratorskoj konzoli ima samo korisnik sa korisničkim imenom "admin". Promena autorizacije korisničkih imena za prijavu na datu konzolu se radi u `admin.authorizedUsernames` polju. Primer u kodu:

```

<jive>
  ...
  <admin>
    ...
    <authorizedUsernames>djordje, marko</authorizedUsernames>
  </admin>

```

```
...
</jive>
```

6.3.3. Integracija korisnika

Openfire takođe može da učitava korisnike i njihove podatke iz zasebne baze. Integracija korisnika ide u kombinaciji sa integracijom autentifikacije. Podešavanje integracije se sastoji iz sledećih koraka:

- provider.user.className – vrednost treba promeniti u:
org.jivesoftware.openfire.user.JDBCUserProvider.
- jdbcUserProvider.loadUserSQL – SQL izjava za učitavanje imena i e-mail adrese korisnika za dato korisničko ime. SQL izjava treba da sadrži jedan "?" znak, koji će se dinamički zameniti korisničkim imenom prilikom izvršavanja.
- jdbcUserProvider.userCountSQL – SQL izjava za učitavanje ukupnog broja korisnika u bazi podataka.
- jdbcUserProvider.allUsersSQL – SQL izjava za učitavanje svih korisničkih imena u bazi podataka.
- jdbcUserProvider.searchSQL – deo SQL izjave koji služi za pretragu korisnika u bazi. Izjava treba da se završi sa "WHERE" – korisničko ime, ime, i e-mail polja će biti dinamički dodata u izjavi, u zavisnosti od pretrage. Ako se ne podesi ova vrednost, pretraga neće biti moguća.
- usernameField – naziv polja korisničkih imena u bazi podataka, koji se koristi za pretrage.
- nameField – naziv polja imena u bazi podataka, koji se koristi za pretrage.
- emailField – naziv polja e-mail u bazi podataka, koji se koristi za pretrage.

Primer jednog dela konfiguracionog fajla:

```
<jive>
...
<provider>
  <auth>

<className>org.jivesoftware.openfire.auth.JDBCAuthProvider</className>
  </auth>
  <user>

<className>org.jivesoftware.openfire.user.JDBCUserProvider</className>
  </user>
</provider>
<jdbcAuthProvider>
  <passwordSQL>SELECT password FROM user_account WHERE
username=?</passwordSQL>
  <passwordType>plain</passwordType>
</jdbcAuthProvider>
<jdbcUserProvider>
  <loadUserSQL>SELECT name,email FROM myUser WHERE
username=?</loadUserSQL>
  <userCountSQL>SELECT COUNT(*) FROM myUser</userCountSQL>
  <allUsersSQL>SELECT username FROM myUser</allUsersSQL>
```

```

    <searchSQL>SELECT username FROM myUser WHERE</searchSQL>
    <usernameField>username</usernameField>
    <nameField>name</nameField>
    <emailField>email</emailField>
  </jdbcUserProvider>
  ...
</jive>

```

6.3.4. Grupna integracija

Openfire može da učitava grupu podataka iz zasebne baze podataka. Grupna integracija se kombinuje sa integracijama autentifikacije i korisnika. Podešavanje integracije se sastoji iz sledećih koraka:

- `provider.group.className` – vrednost treba promeniti u:
`org.jivesoftware.openfire.group.JDBCGroupProvider`.
- `jdbcGroupProvider.groupCountSQL` – SQL izjava za učitavanje ukupnog broja grupa u bazi podataka.
- `jdbcGroupProvider.allGroupsSQL` – SQL izjava za učitavanje svih grupa u bazi podataka.
- `jdbcGroupProvider.userGroupsSQL` – SQL izjava za učitavanje svih grupa za određenog korisnika. SQL izjava treba da sadrži jedan "?" znak, koji će se dinamički zameniti korisničkim imenom prilikom izvršavanja.
- `jdbcGroupProvider.descriptionSQL` – SQL izjava za učitavanje opisa grupe. SQL izjava treba da sadrži jedan "?" znak, koji će se dinamički zameniti imenom grupe prilikom izvršavanja.
- `jdbcGroupProvider.loadMembersSQL` – SQL izjava za učitavanje svih članova grupe. SQL izjava treba da sadrži jedan "?" znak, koji će se dinamički zameniti imenom grupe prilikom izvršavanja.
- `jdbcGroupProvider.loadAdminsSQL` – SQL izjava za učitavanje svih administratora u grupi. SQL izjava treba da sadrži jedan "?" znak, koji će se dinamički zameniti imenom grupe prilikom izvršavanja.

Primer jednog dela konfiguracionog fajla:

```

<jive>
  ...
  <provider>
    <auth>

    <className>org.jivesoftware.openfire.auth.JDBCAuthProvider</className>
    </auth>
    <user>

    <className>org.jivesoftware.openfire.user.JDBCUserProvider</className>
    </user>
    <group>

    <className>org.jivesoftware.openfire.group.JDBCGroupProvider</className>
    </group>
  </provider>
  <jdbcAuthProvider>

```

```

    <passwordSQL>SELECT password FROM user_account WHERE
username=?</passwordSQL>
    <passwordType>plain</passwordType>
</jdbcAuthProvider>
<jdbcUserProvider>
    <loadUserSQL>SELECT name,email FROM myUser WHERE
username=?</loadUserSQL>
    <userCountSQL>SELECT COUNT(*) FROM myUser</userCountSQL>
    <allUsersSQL>SELECT username FROM myUser</allUsersSQL>
    <searchSQL>SELECT username FROM myUser WHERE</searchSQL>
    <usernameField>username</usernameField>
    <nameField>name</nameField>
    <emailField>email</emailField>
</jdbcUserProvider>
<jdbcGroupProvider>
    <groupCountSQL>SELECT count(*) FROM myGroups</groupCountSQL>
    <allGroupsSQL>SELECT groupName FROM myGroups</allGroupsSQL>
    <userGroupsSQL>SELECT groupName FROM myGroupUsers WHERE
username=?</userGroupsSQL>
    <descriptionSQL>SELECT groupDescription FROM myGroups WHERE
groupName=?</descriptionSQL>
    <loadMembersSQL>SELECT username FROM myGroupUsers WHERE
groupName=? AND isAdmin='N'</loadMembersSQL>
    <loadAdminsSQL>SELECT username FROM myGroupUsers WHERE groupName=?
AND isAdmin='Y'</loadAdminsSQL>
</jdbcGroupProvider>
...
</jive>

```

6.4. LDAP

LDAP (Lightweight Directory Access Protocol) predstavlja jedan od vodećih standarda za autentifikaciju korisnika i skladištenje profilnih podataka korisnika. Kao moćan alat, koristi se u velikim organizacionim celinama da bi olakšao upravljanje korisnicima. Poznati LDAP serveri su Open LDAP, Active Directory i Novell's eDirectory.

Openfire u osnovi, skladišti sve korisničke podatke u sopstvenoj bazi podataka i izvršava autentifikaciju potraživanjem u bazi. LDAP modul zamenjuje tu funkcionalnost i dozvoljava Openfire da:

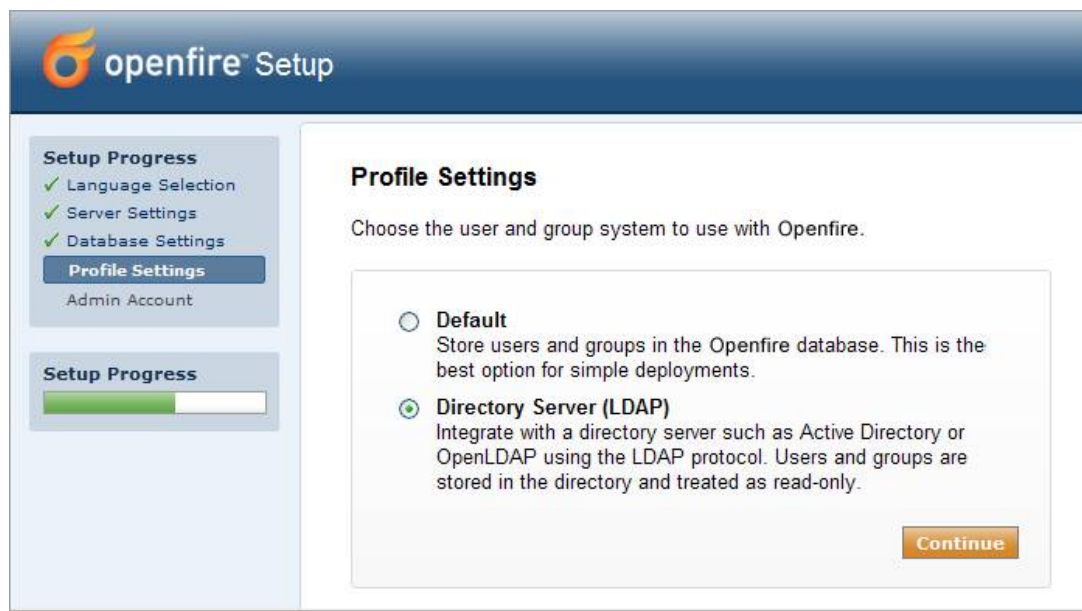
- koristi LDAP server autentifikaciju korisničkog identiteta,
- učitava informaciju korisničkog profila iz LDAP direktorijuma,
- učitava skup informacija iz LDAP direktorijuma.

Openfire može samo da očitava LDAP direktorijum (read-only).

6.4.1. Konfiguracija

Openfire konfiguracioni alat obuhvata LDAP čarobnjak za podešavanje. Na Slici 10 je prikazana stranica za podešavanje profila, gde je dat izbor opcija. Odabirom LDAP opcije

se pristupa daljem podešavanju uz pomoć čarobnjaka. U ovom poglavlju će biti predstavljen rad sa aktivnim direktorijumom kao primer.



Slika 11.

Ukoliko je već odrađen proces podešavanja Openfire servera, a želite da omogućite upotrebu LDAP, treba da se uradi:

1. stopiranje Openfire,
2. izmena datoteku `conf/openfire.xml` u Openfire instalacionom direktorijumu i prepodešavanje `<setup>true</setup>` u `<setup>>false</setup>`.
3. restart Openfire i pokretanje alata za podešavanje.

6.4.2. Rad sa aktivnim direktorijumom

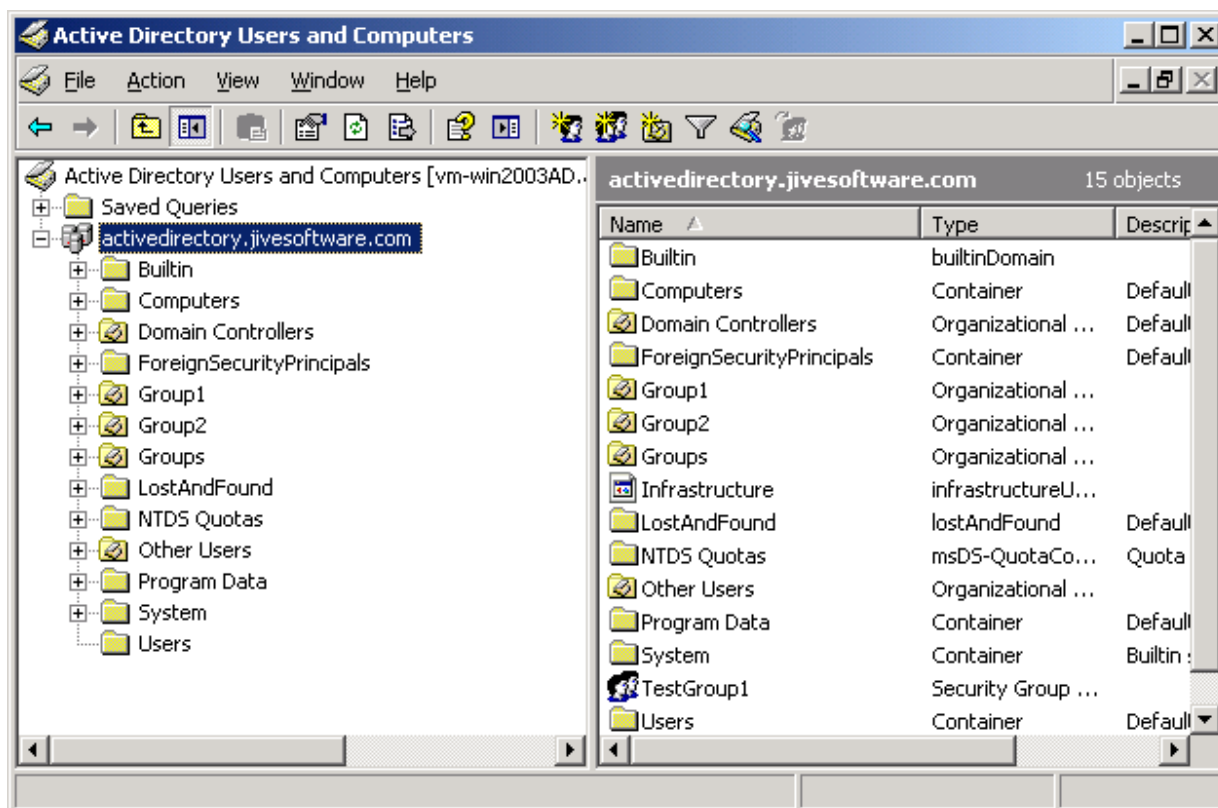
MS Active Directory (u daljem tekstu AD) je široko rasprostranjen sistem direktorijuma koji podržava LDAP protokol (Slika 11). Dva specifična polja na koja nailazimo u toku konfiguracije a tiču se konekcije sa AD serverima, su:

- **Base DN**

Base DN opisuje gde da se učitavaju korisnici i grupe. Ako se koristi podrazumevano AD podešavanje, svi korisnički nalozi i grupe se nalaze u "Users" folderu datog domena. LDAP forma je `cn=Users;dc=<Nas domen>`. Za domen `dc1.vs.rs`, base DN zapis bi izgledao `cn=Users;dc=dc1,dc=vs,dc=rs`.

- **Administrator DN**

AD ne dozvoljava anonimne LDAP konekcije. Zato je neophodno uneti DN korisnika sa privilegijama za konektovanje na server i čitanje svih podataka korisnika i grupa. Ukoliko ne postoji namenski nalog, najpodesniji je sistemski administratorski nalog. Administratorski DN je `cn=Administrator,dc=<Nas domen>`. Za domen `dc1.vs.rs`, base DN zapis bi izgledao `cn=Administrator,cn=users;dc=dc1,dc=vs,dc=rs`.



Slika 12.

6.4.3. Manualna izmena konfiguracionog fajla

Podešavanje LDAP može i direktno da se izvrši izmenom konfiguracionog fajla `conf/openfire.xml` u Openfire instalaciji koristeći tekstualni program (editor). Svojstva označena sa (*) su obavezna za podešavanje, svojstva označena sa (**) moraju da se podese za aktiviranje LDAP grupe podrške i ostala svojstva su opcionalna:

Glavna podešavanja

- `provider.user.className` * – vrednost treba promeniti u: `"org.jivesoftware.openfire.ldap.LdapUserProvider"`.
- `provider.auth.className` * – vrednost treba promeniti u: `"org.jivesoftware.openfire.ldap.LdapAuthProvider"`.
- `ldap.host` * – LDAP serverski host; npr. `dc1.vs.rs...` Moguća je upotreba više LDAP servera ali svi oni bi trebalo da dele istu konfiguraciju (npr. SSL, baseDN, administratorski nalog i dr.).

- `ldap.port` – LDAP serverski broj porta. Podrazumevana vrednost je 389.
- `ldap.connectionTimeout` – vrednost ovog svojstva je broj (integer u obliku string-a) koji predstavlja vreme isteka konekcije u milisekundama. Ako se LDAP ne poveže u predviđenom vremenu, prekida se pokušaj konekcije. Vrednost treba da bude veća od nule, a ako nije podešena, podrazumevana vrednost je 10 sekundi. Ne važi za SSL/TLS konekcije.
- `ldap.readTimeout` – vrednost ovog svojstva je broj (integer u obliku string-a) koji predstavlja vreme isteka vremena očitavanja u milisekundama za LDAP operacije. Ukoliko LDAP provajder ne dobije LDAP odziv u predviđenom vremenu, otkazuje dalje očitavanje. The integer should be greater than zero. Vrednost treba da bude veća od nule, a ako nije podešena, podrazumevana vrednost je beskonačno vreme. Zahteva Java 1.6 ili noviju verziju.
- `ldap.baseDN` * – početni DN koji pretražuje korisnike s kojima treba da radi.
- `ldap.alternateBaseDN` – drugi DN u direktorijumu je opcionalan. Ako se aktivira, onda se koristi za autentifikaciju, učitavajući pojedinačno korisnike i prikazujući listu korisnika.
- `ldap.adminDN` – DN od administratora direktorijuma. Sve operacije sa direktorijumom se izvršavaju ovim nalogom. Administrator mora da radi pretragu i da učitava zapise korisnika.
- `ldap.adminPassword` – lozinka administratora direktorijuma.
- `ldap.usernameField` – polje preko koga se vrše pretrage korisničkih imena. Kada nijepodešeno ovo svojstvo, podrazumevana vrednost je `uid`. AD korisnici treba da postave `sAMAccountName`.
- `ldap.nameField` – polje koje sadrži ime datog korisnika. Podrazumevana vrednost je `cn`. AD korisnici treba da postave `displayName`.
- `ldap.emailField` – polje koje sadrži e-mail adresu korisnika. Podrazumevana vrednost je `mail`. AD korisnici treba da postave `mail`.
- `ldap.searchFields` – LDAP polja koja se koriste za pretrage korisnika. Ako ovo svojstvo nije podešeno, onda se pretražuju polja korisničkog imena, imena i e-mail adrese. Primer: "Username/uid,Name/cname".
- `ldap.searchFilter` – opcionalni filter za pretraživanje prilikom učitavanja korisnika. Podrazumevani filter za pretraživanje se kreira koristeći attribute iz svojstva `ldap.usernameField`.
- `ldap.subTreeSearch` – podrazumevano, Openfire će pretraživati celo LDAP podstablo (počev od osnovnog DN) kada učitava korisnike. Ovo svojstvo je podešeno na `false`, i podstablo je onemogućeno, tako da se korisnici učitavaju direktno iz osnovnog DN. Ovo može da ubrza performanse, ali i onemogućiti učitavanje korisnika iz podstabla DN.

Grupna podešavanja

- `provider.group.className` ** – vrednost treba promeniti u: "org.jivesoftware.openfire.ldap.LdapGroupProvider".
- `ldap.groupNameField` ** – naziv polja na kojem se realizuju pretrage imena grupa. Podrazumevana vrednost je `cn` ako nije podešeno ovo svojstvo.
- `ldap.groupMemberField` – naziv polja koje sadrži članove grupe. Podrazumevana vrednost je `member` ako nije podešeno ovo svojstvo.

- `ldap.groupDescriptionField` – naziv polja koje sadrži opis grupe. Podrazumevana vrednost je `description` ako nije podešeno ovo svojstvo.
- `ldap.posixMode` ** – vrednost "true" predstavlja to da se korisnici čuvaju unutar grupe samo poimenično. Vrednost "false" predstavlja to da se korisnici čuvaju po celokupnom DN unutar grupe. Ako nije podešeno svojstvo, podrazumevana vrednost je `false`. Posix mod za AD je `false`.
- `ldap.groupSearchFilter` – opcionalni filter za pretraživanje prilikom učitavanja grupa. Osnovni filter za pretragu grupa se kreira koristeći atribut određen od strane `ldap.groupNameField`.

Podešavanje konekcije

- `ldap.debugEnabled` – vrednost je "true" ako treba debugging da radi. Kada je aktivno svojstvo, tragovi informacija o baferima poslatim i primljenim od strane LDAP provajdera se zapisuju u `System.out`.
- `ldap.sslEnabled` – vrednost je "true" kada treba da se omoguće SSL konekcije na LDAP server. LDAP serverski broj porta bi trebalo da bude 636.
- `ldap.initialContextFactory` – naziv klase koji bi se koristio kao polazna tačka programske biblioteke. Podrazumevana vrednost je `"com.sun.jndi.ldap.LdapCtxFactory"`. Uglavnom nije neophodna.
- `ldap.autoFollowReferrals` – vrednost "true" indicira da LDAP pomoćne reference moraju da se automatski prate. Ako je vrednost "false", referentna polisa koja se koristi yavisi od provajdera. Referenca je entitet koji služi za preusmeravanje klijentskih zahteva na drugi server. Ona sadrži imena i lokacije drugih objekata.
- `ldap.connectionPoolEnabled` – vrednost "false" onemogućava povezivanje LDAP konekcija. Podrazumevana vrednost je "true", ako se ne podesi ovo svojstvo.

Primer jednog dela konfiguracionog fajla:

```
<jive>
...
<ldap>

    <host></host>
    <port>389</port>
    <usernameField>uid</usernameField>
    <nameField>cn</nameField>

    <emailField>mail</emailField>
    <baseDN>ou=People;dc=example;dc=com</baseDN>
    <adminDN>cn=Directory Administrator</adminDN>
    <adminPassword></adminPassword>

</ldap>
<provider>
    <user>

<className>org.jivesoftware.openfire.ldap.LdapAuthProvider</className>
    </user>

    <auth>
```

```
<className>org.jivesoftware.openfire.ldap.LdapAuthProvider</className>
  </auth>
</group>

<className>org.jivesoftware.openfire.ldap.LdapGroupProvider</className>

  </group>
</provider>
...
</jive>
```

Prijavljivanje na administratorskoj konzoli ima samo korisnik sa korisničkim imenom "admin". Međutim, postoje korisnici LDAP direktorijuma kojimogu da budu administratori. Promena autorizacije korisničkih imena za prijavu na datu konzolu se radi u `admin.authorizedUsernames` svojstvu. Primer u kodu:

```
<jive>
  ...
  <admin>
    ...
    <authorizedUsernames>djordje, marina</authorizedUsernames>
  </admin>
  ...
</jive>
```

6.5. SSL instrukcije

SSL podrška za Openfire je napravljena korišćenjem standardne Java bezbednosne SSL implementacije (`javax.net.ssl.SSLServerSocket`). Od verzije 3.2 Openfire servera, upravljanje i uvođenje sertifikata i enkripcije je moguć putem administratorske konzole, ali i manualno pomoću standardnog Sun JDK 1.5 alata.

Sun JDK (verzija 1.5.x) se distribuira sa svim bezbednosnim alatima potrebnim za konfiguraciju SSL u Openfire-u. najbitniji je `keytool` koji se nalazi u `JAVA_HOME/bin` direktorijumu JDK. Sun JVM (Java virtual machine) održavaju repozitorijume sigurnosnih sertifikata (sopstvenih i drugih učesnika), privatnih i javnih ključeva koji se nazivaju "keystore" i "truststore", na fajl-sistemu kao kriptovane fajlove. `keytool` se koristi za kreiranje, čitanje, ažuriranje, i brisanje zapisa ovih fajlova. Openfire se isporučuje sa samo-potpisanim (self-signed) tipskim sertifikatom dizajniranim za inicijalno probno testiranje.

Za konfiguraciju SSL na serveru potrebno je da se izvrše sledeći koraci:

1. određivanje Openfire serverskog domena.
2. kreiranje samo-potpisanog SSL serverskog sertifikata za serverski domen. Nakon inicijalnog podešavanja se kreira samo-potpisani sertifikat, tako da ukoliko već postoje (Openfire serverski domen se poklapa sa postojećim web domenom sa SSL-om), treba preći na 4.korak.

3. CA telo može da sertifikuje SSL serverski sertifikat (opcionally).
 - a. generiše se zahtev za potpisivanje sertifikata (CSR).
 - b. pošalje se CSR određenom CA telu za potpis.
4. importovanje serverskog sertifikata u keystore (ako koristimo samo-potpisani sertifikat generisan u 2.koraku, sertifikat je već uvežen i možemo preskočiti ovaj korak).
5. uklanjanje podrazumevanih (default) sertifikata iz keystore-a.
6. importovanje klijentskih sertifikata u truststore.
7. prilagođavanje Openfire konfiguracije sa adekvatnim keystore i truststore podešavanjima.

Određivanje domena servera

Openfire serverski domen treba da se poklapa sa imenom serverskog hosta, npr. "vs.rs". Korisnički nalozi bi bili oblika "korisnik@vs.rs" kao e-mail adrese. U daljim instrukcijama ćemo koristiti domen "vs.rs" kroz primere.

Kreiranje samo-potpisanog serverskog sertifikata

Da bi kreirali samo-potpisani serverski sertifikat, pristupamo komandnoj liniji i menjamo direktorijume u `resources/security` direktorijumu Openfire instalacije. Trebalo bi da se vide podrazumevani keystore i truststore fajlovi. Prvo treba promeniti osnovnu keystore lozinku:

```
keytool -storepasswd -keystore keystore
```

keytool naravno traži staru lozinku (a to je `changeit`) i posle toga novu lozinku. Potom se pravi sertifikat uz pomoć keytool-a:

```
keytool -genkey -keystore keystore -alias vs.rs
```

gde se i menja ime servera u `vs.rs`. Keytool će tražiti skladišnu lozinku, i nekoliko informacija za sertifikat. Treba uneti sve potrebne informacije i kompletirati upisivanjem naziva servera kada se traži ime i prezime. Nakon unetih informacija, keytool traži da se verifikuje informacija i podesi lozinka ključa (skladišna i ključna lozinka mora biti ista). Ako se kasnije izmeni keystore lozinka, treba zameniti i unosnu lozinku koristeći keytool:

```
keytool -keypasswd -alias vs.rs -keystore keystore
```

Keytool će kreirati sertifikate koristeći DSA algoritam (podrazumevano). Neki klijenti očekuju od servera da poseduje RSA sertifikate ili neće koristiti TLS/SSL. Tako da je poželjno napraviti i RSA sertifikate u keystore-u.

```
keytool -genkey -keystore keystore -alias vs.rs -keyalg RSA
```

Dobijanje CA potpisanog sertifikata

Za potraživanje CA potpisanog sertifikata, prvo treba eksportovati sertifikat u standardnom CSR formatu. To se može uraditi uz pomoć keytool-a:

```
keytool -certreq -keystore keystore -alias vs.rs -file certificate_file
```

gde se i menja ime servera u `vs.rs` kao i naziv sertifikacionog fajla koji pravimo za `certificate_file`. Dalje se šalje CSR CA telu i prate njihove instrukcije.

Importovanje serverskih sertifikata

U slučaju da je CA potpisao sertifikat za server, ili već poseduje SSL sertifikat, neophodno je da se uveze preko keytool-a.

```
keytool -import -keystore keystore -alias vs.rs -file  
signed_certificate_file
```

Uklanjanje postojećih (default) sertifikata

```
keytool -delete -keystore keystore -alias rsa
```

```
keytool -delete -keystore keystore -alias dsa
```

Importovanje klijentskih sertifikata

Ukoliko se zahteva od klijenata da potvrde svoj identitet sertifikatima, neophodno je da se izuzmu njihovi sertifikati i importuju u truststore fajl a ne u keystore. Prvo se menja podrazumevana lozinka za truststore:

```
keytool -storepasswd -keystore truststore
```

keytool naravno traži staru lozinku (a to je `changeit`) i posle toga novu lozinku. Potom se uvozi svaki sertifikat pojedinačno uz pomoć keytool-a:

```
keytool -import -keystore truststore -alias user_name -file  
certificate_file
```

Konfiguracija Openfire-a

Konfiguracija se radi pokretanjem Openfire administratorske konzole preko web pregledača i menjanju se sistemska svojstva:

- `xmpp.socket.ssl.active` – podešavamo "true" za aktiviranje SSL-a.
- `xmpp.socket.ssl.port` – koji port koristi za SSL (podrazumevani 5223 za XMPP)
- `xmpp.socket.ssl.storeType` – koji se skladišni tip koristi ("JKS" je Sun Java Keystore format koji upotrebljava JDK keytool). Ako se ne definiše, Openfire koristi vrednost "jks".
- `xmpp.socket.ssl.keystore` – lokacija keystore fajla povezanog sa Openfire instalacionim root direktorijumom.

- xmpp.socket.ssl.keypass – lozinka truststore i ključa koja je promijenjena u 2.koraku.
- xmpp.socket.ssl.truststore – lokacija keystore fajla povezanog sa Openfire instalacionim root direktorijumom, ako ostavimo prazno onda nije u upotrebi.
- xmpp.socket.ssl.trustpass – lozinka truststore i ključa koja se menja u 6.koraku.

Nakon svih podešavanja mora da se restartuje server.

7. ZAKLJUČAK

IM je definitivno promenio način komunikacije, dozvoljavajući neposrednu interakciju, kao i stalnu dostupnost za kontakt među ljudima. Te promene se mogu posmatrati sa više aspekata, kako pozitivnih tako i negativnih. Sa društvenim i tehnološkim napretkom, kao posledica užurbanog načina života u privatnom i poslovnom smislu, nastao je novi sistem komuniciranja brzih poruka. IM je eksplodirao u okviru raznih društvenih mreža kao što su Facebook, Twiter i sl. Naravno, prateći faktor u razvoju svakog softvera i tehnologije su maliciozni hakeri i pregršt zlonamernog softvera i zloupotreba. Gde se tu nalazi XMPP?

Ogromna prednost XMPP protokola je u njegovoj suštini gde se bavi bezbednosnim pitanjima koja nisu direktno povezana sa problemima svetske mreže. To znači da je isključiv fokus na razvoj samog protokola i servisa u svakom pogledu, a pritom da se oslanja na već postojeće standardne bezbednosne protokole i pravila.

Sa centralizovanom arhitekturom, sav saobraćaj se preusmerava kroz server i identitet datog servera može da se potvrdi potpisom Sertifikovnih Autoriteta (CA), krajnji korisnici ne moraju da razmišljaju o složenim stvarima kao sertifikat za uspostavljanje sigurne komunikacije. Kao u e-mail svetu, poverenje se prepušta serveru koji mora da zadovoljava određene propise i kriterijume.

Pored toga, XMPP je dizajniran da šalje sve poruke u realnom vremenu koristeći vrlo efikasne mehanizme. Mnogi postojeći web bazirani mehanizmi često prave nepotrebne zahteve zagušujući mrežu a ne funkcionišu u realnom vremenu.

S obzirom da je XMPP od svog nastanka, protokol otvorenog tipa i ima punu podršku programerske Jabber zajednice, koja prati sve moderne tokove razvoja drugih standarda i protokola, mogućnosti za dalje napredovanje su prilično velike. Sve više je prisutna implementacija na različite platforme i društvene mreže, tako da će verovatno XMPP u budućnosti imati još veću perspektivu i primenu, pre svega zbog svoje fleksibilnosti.

LITERATURA

- [1] Tom Van Vleck. "Instant Messaging on CTSS and Multics". Multicians.org. 2012-05-11.
- [2] Wired and Inspired, The Columbus Dispatch (Business page), by Mike Pramik, November 12, 2000.
- [3] "AOL Instant Messenger's Real-Time IM feature". Help.aol.com. 2012-05-11.
- [4] A Review of XMPP based chat Servers with a vision of Security Enhancement using Cryptography, by Poonam Taneja, Asha Mishra, Pawan Bhadana, March 2014.
- [5] <http://xmpp.org/about-xmpp/>
- [6] <https://www.ietf.org/>
- [7] <http://xmpp.org/about-xmpp/history/>
- [8] Saint-Andre, P. (March 2011). Extensible Messaging and Presence Protocol (XMPP): Core. IETF. RFC 6120. Retrieved May 4, 2014.
- [9] "XEP-0045: Multi-User Chat", Saint-Andre, Peter. xmpp.org.
- [10] "XEP-0060: Publish-Suscribe", Millard, Peter. Saint-Andre, Peter. Meijer, Ralph xmpp.org.
- [11] "XEP-0030: Service Discovery", Hildebrand, Joe. Millard, Peter. Eatmon, Ryan. Saint-Andre, Peter. xmpp.org.
- [12] "XEP-0115: Entity Capabilities", Hildebrand, Joe. Saint-Andre, Peter. Tronçon, Remko. Konieczny, Jacek. xmpp.org.
- [13] "XEP-0004: Data Forms", Eatmon, Ryan. Hildebrand, Joe. Miller, Jeremie. Muldowney, Thomas. Saint-Andre, Peter. xmpp.org.
- [14] "XEP-0050: Ad-Hoc Commands", Miller, Matthew. xmpp.org.
- [15] "XEP-0166: Jingle", Ludwig, Scott. Beda, Joe. Saint-Andre, Peter. McQueen, Robert. Egan, Sean. Hildebrand, Joe. xmpp.org.
- [16] "XEP-0238: XMPP Protocol Flows for Inter-Domain Federation", Saint-Andre, Peter. xmpp.org.
- [17] "XEP-0220: Server Dialback", Jeremie Miller, Peter Saint-Andre, Philipp Hancke. xmpp.org.
- [18] Saint-Andre, P. (September 2015). Extensible Messaging and Presence Protocol (XMPP): Core. IETF. RFC 7622.
- [19] Preparation of Internationalized Strings ("stringprep"). P. Hoffman, M. Blanchet Viagenie. IETF. RFC 3454. December 2002.
- [20] "XEP-0147: XMPP URI Scheme Query Components", Saint-Andre, Peter. xmpp.org.

- [21] "XEP-0322: Efficient XML Interchange (EXI) Format", Peter Waher, Yusuke DOI. xmpp.org.
- [22] "XEP-0310: Presence State Annotations", Kevin Smith. xmpp.org.
- [23] L.Joncheray. A simple active attack against TCP. In Proceedings of the 5th USENIX Security Symposium, pages 7–19, June 1995.
- [24] N.Hindocha. Threats to Instant Messaging, 2003. Symantec Security Response. <http://securityresponse.symantec.com/avcenter/reference/threats.to.instant.messaging.pdf>
- [25] Techworld.com Staff. Real Player struck by massive security hole, Feb. 2004. News article, <http://www.techworld.com/news/>; Macworld.co.uk Staff. Microsoft warns of JPEG threat, Sept. 2004. News article, <http://www.macworld.co.uk/news/>.
- [26] ElcomSoft. Advanced instant messengers password recovery. <http://www.elcomsoft.com/aimpr.html>.
- [27] R. E. Grinter and L. Palen. Instant Messaging in teen life. In Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work, pages 21–30. ACM Press, 2002.
- [28] Secunia. ICQ predictable file location weakness, Feb. 2004. Secunia Advisory SA10970, <http://secunia.com/advisories/10970/>.
- [29] US-CERT. AOL Instant Messenger contains buffer overflows in parsing of AIM URI handler requests, Jan. 2002. <http://www.kb.cert.org/vuls/id/474592>.
- [30] D.M.Kienzle and M.C.Elder. Recent worms: A survey and trends. In Proceedings of the 2003 ACM Workshop on Rapid Malcode, pages 1–10, Washington, DC, USA, Oct. 2003. ACM Press.
- [31] USA Today Staff. IM viruses opening a new can of worms, Aug. 2001. News article, <http://www.usatoday.com>.
- [32] News.com Staff. Yahoo fills in Messenger hole, May 2005. News article, <http://news.com.com/>
- [33] Symantec.W32.Bizex.Worm <http://securityresponse.symantec.com/avcenter/venc/data/w32.bizex.worm.html>.
- [34] Microsoft TechNet. Microsoft security bulletin MS05-009: Vulnerability in PNG processing could allow remote code execution (890261) <https://technet.microsoft.com/library/security/ms05-009>.
- [35] Microsoft TechNet. Microsoft security bulletin MS05-022: Vulnerability in MSN Messenger could lead to remote code execution (896597). <https://www.microsoft.com/technet/security/Bulletin/MS05-022.msp>
- [36] Symantec. W32.Choke.Worm. <http://securityresponse.symantec.com/avcenter/venc/data/w32.choke.worm.html>.
- [37] Symantec.W95.SoFunny.Worm@m.https://www.symantec.com/security_response/writeup.jsp?docid=2001-071013-3230-99.

- [38] TrendMicro.JSMENGER.GEN.http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=JS_MENGER.GEN.
- [39] Symantec.W32.Bropia.M.<http://securityresponse.symantec.com/avcenter/venc/data/w32.bropia.m.html>.
- [40] TrendMicro.WORMBROPIA.F.http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BROPIA.F.
- [41] TrendMicro.WORMAGOBOT.AJC.http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_AGOBOT.AJC.
- [42] Symantec.W32.Serflog.A.<http://securityresponse.symantec.com/avcenter/venc/data/w32.serflog.a.html>.
- [43] "XEP-0205: Best Practices to Discourage Denial of Service Attacks", Saint-Andre, Peter. xmpp.org.
- [44] The Transport Layer Security (TLS) Protocol Version 1.2. Rescorla, Eric. Dierks, Tim. IETF. RFC 5246. August 2008.
- [45] Simple Authentication and Security Layer. Melnikov, Alexey. Zeilenga, Kurt. IETF. RFC 4422. June 2006.
- [46] "XEP-0138: Stream Compression", Hildebrand, Joe. Saint-Andre, Peter. xmpp.org.
- [47] The PLAIN Simple Authentication and Security Layer (SASL) Mechanism. Zeilenga, Kurt. IETF. RFC 4616. August 2006.
- [48] Using Digest Authentication as a SASL Mechanism. Leach, Paul. Newman, Chris. IETF. RFC 2831. May 2000.
- [49] The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2. Zhu, Larry. Jaganathan, Karthik. Hartman, Sam. IETF. RFC 4121. July 2005.
- [50] Anonymous Simple Authentication and Security Layer (SASL) Mechanism. Zeilenga, Kurt. IETF. RFC 4505. June 2006.
- [51] "XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH)", Paterson, Ian. Smith, Dave. Saint-Andre, Peter; "XEP-0206: XMPP Over BOSH", Paterson, Ian. Saint-Andre, Peter. xmpp.org.
- [52] "XEP-0156: Discovering Alternative XMPP Connection Methods", Hildebrand, Joe. Saint-Andre, Peter. xmpp.org.
- [53] "XEP-0174: Serverless Messaging", Saint-Andre, Peter. xmpp.org.
- [54] <https://en.wikipedia.org/wiki/Openfire>
- [55] <http://www.igniterealtime.org/projects/openfire/documentation.jsp>