Centar Informacijske Sigurnosti

Mrežni protokoli za razmjenu datoteka

CIS-DOC-2011-02-004

Upozorenje

Podaci, informacije, tvrdnje i stavovi navedeni u ovom dokumentu nastali su dobrom namjerom i dobrom voljom te profesionalnim radom CIS-ovih stručnjaka, a temelje se na njihovom znanju i petnaestak godina iskustva u radu u informacijskoj sigurnosti. Namjera je da budu točni, precizni, aktualni, potpuni i nepristrani.

Ipak, oni su dani samo kao izvor informacija i CIS ne snosi nikakvu izravnu ili posrednu odgovornost za bilo kakve posljedice nastale korištenjem podataka iz ovog dokumenta.

Ukoliko primijetite bilo kakve netočnosti, krive podatke ili pogreške u ovom dokumentu, ili imate potrebu komentirati sadržaj molimo Vas da to javite elektroničkom poštom na adresu info@CIS.hr.

O CIS-u

CIS izrađuje pregledne dokumente (eng. white paper) na teme iz područja informacijske sigurnosti koji će biti korisni zainteresiranoj javnosti, a u svrhu **podizanje njezine svijesti o informacijskoj sigurnosti** i **sposobnosti za čuvanje i zaštitu informacija i informacijskih sustava**. Pored toga, CIS razvija i održava mrežni portal www.CIS.hr kao referalnu točku za informacijsku sigurnost za cjelokupnu javnost; izrađuje obrazovne materijale namijenjene javnosti; organizira događaje za podizanje svijesti o informacijskoj sigurnosti u javnosti i pojedinim skupinama te djeluje u suradnji sa svim medijima.

CIS **okuplja mlade** zainteresirane za informacijsku sigurnost i radi na njihovom pravilnom odgoju i obrazovanju u području informacijske sigurnosti te pripremu za **profesionalno bavljenje informacijskom sigurnošću**.

Centar informacijske sigurnosti [CIS] nastao je 2010. godine na poticaj Laboratorija za sustave i signale[LSS] Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu, a kao posljedica 15togodišnjeg rada na istraživanju, razvoju i primjeni informacijske sigurnosti. LSS je među ostalim potaknuo osnivanje CARNetovog CERTa i sudjelovao u izradi Nacionalnog programa informacijske sigurnosti RH.

Smisao CISa je da bude **referentno mjesto za informacijsku sigurnost** za javnost, informatičare i posebno za mlade te da sustavno podiže njihovu svijest i sposobnosti u području informacijske sigurnosti.

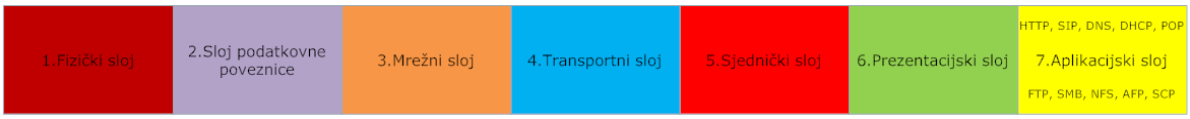
Rad CISa podržava Ministarstvo znanosti, obrazovanja i sporta Republike Hrvatske, a omogućuju sponzori.

Prava korištenja

Sadržaj

1. Uvod

U posljednjem desetljeću došlo je do značajnog širenja Interneta kao jednog od vodećih informacijskih medija u smislu porasta broja njegovih korisnika te usluga javno dostupnih putem te globalne mreže. Sam Internet počiva na čitavom nizu protokola bez kojih sva postojeća infrastruktura i funkcionalnost, koju korisnici uočavaju i koriste, ne bi bila moguća. Općenito, protokol je dogovoreni postupak kojeg treba slijediti u određenoj situaciji [1]. U slučaju Interneta, protokoli omogućuju uspostavu komunikacije između pojedinih elemenata mrežne infrastrukture. Spomenuti protokoli teoretski su kategorizirani u sedam OSI mrežnih slojeva koji čine takozvani OSI mrežni model [2]. To su fizički sloj, sloj podatkovne poveznice, mrežni, transportni, sjednički, prezentacijski te aplikacijski sloj. Mrežni protokoli za razmjenu datoteka pripadaju aplikacijskom sloju OSI mrežnog modela. Navedeni model te položaj tih protokola ilustrira slika 1.



Slika 1. OSI mrežni model
Izvor: Wikipedia

U grupu takvih protokola svrstavaju se mrežni protokoli poput protokola FTP, SFTP, TFTP, SMB, AFP, NFS i RFS. Postavlja se pitanje zašto su uopće nastali takvi protokoli. Odgovor je zapravo vrlo jasan i jednostavan. Ljudi su od davnine prenosili podatke odnosno datoteke s jedne lokacije na drugu [3], a to je općenita definicija transfera datoteka. Prije pojave računala, podaci su bilježeni na papiru ili bušenima karticama pa su ljudi razmjenjivali spise papira ili kolekcije bušenih kartica fizičkim prijenosom s jedne lokacije na drugu. Pojavom računala, datoteke su se i dalje fizički razmjenjivale samo putem drugih medija poput disketa (eng. *diskette*), kompaktnih diskova (eng. *Compact Disk*, CD) ili tvrdih diskova (eng. *Hard Disk*). Pojavom Interneta stvorena je mogućnost brže direktne razmjene datoteka putem računalne mreže. Kako bi takva razmjena bila moguća, bilo je potrebno standardizirati način razmjene datoteka. Standardizacijom postupka omogućena je efikasna suradnja i komunikacija računala koja sudjeluju u razmjeni datoteka, bez obzira na razlike u korištenoj programskoj podršci i sklopovlju. Mrežni protokoli za razmjenu datoteka predstavljaju taj globalni standard. Preciznije, mrežni protokol za razmjenu datoteka je konvencija koja opisuje kako prenijeti datoteke između dva računala odnosno dvije krajnje točke komunikacije [4]. Osnovna svrha im je isključivo prijenos toka bitova pohranjenog u obliku jedinice datotečnog sustava te uz datoteku vezanih metapodataka (podataka o podatcima) poput naziva datoteke, veličine datoteke i vremenske oznake. Navedeni protokoli obično se u svom radu oslanjaju na protokole niže razine OSI modela. Primjerice, FTP se oslanja na protokol TCP (eng. *Transmission Control Protocol*) na transportnom sloju, a na protokol IP (eng. *Internet Protocol*) na mrežnom sloju OSI modela. U kontekstu Interneta, pojam razmjene datoteka definiran je kao čin prijenosa datoteka kroz mrežu računala. Računala koja pružaju uslugu prijenosa datoteka nazivaju se datotečni poslužitelji. Ovisno o smjeru komunikacije, gledajući iz perspektive klijenta, prijenos datoteka može se nazivati preuzimanje (eng. *download*) ili slanje (eng. *upload*)(klijent predaje datoteku drugom klijentu ili poslužitelju).

Postoje dva tipa razmjene datoteka:

- *pull* metoda - razmjena datoteka koja je inicirana od strane primatelja datoteka,
- *push* metoda - razmjena datoteka koja je inicirana od strane pošiljatelja datoteka.

Postoji više različitih načina putem kojih se može ostvariti razmjena datoteka:

- transparentna razmjena datoteka putem mrežnih datotečnih sustava,
- eksplicitna razmjena datoteka putem namjenskih usluga razmjene datoteka poput FTP-a i SMB-a,
- distribuirana razmjena datoteka putem *peer-to-peer* mreža (poput Bittorenta ili Gnutelle),
- razmjena datoteka putem programskih alata za razmjenu poruka u stvarnom vremenu (eng. *instant messenger*) ili putem sličnih alata koji se primjenjuju u lokalnim mrežama,
- razmjena datoteka između računala i perifernih, vanjskih uređaja,
- razmjena datoteka putem direktnih modemskih ili serijskih poveznica poput XMODEM-a, YMODEM-a i ZMODEM-a.

2. Opis poznatih mrežnih protokola za razmjenu datoteka

U ovom poglavlju dokumenta bit će opisani sljedeći protokoli:

- FTP (eng. *File Transfer Protocol*),
- FTPS (eng. *File Transfer Protocol over SSL*)
- TFTP (eng. *Trivial File Transfer Protocol*),
- SFTP (eng. *Simple File Transfer Protocol*),
- SFTP (eng. *SSH File Transfer Protocol*),
- FXP (eng. *File eXchange Protocol*),
- FSP (eng. *File Service Protocol*),
- UFTP (eng. *UDP-based File Transfer Protocol*),
- SMB/CIFS (eng. *Server Message Block/Common Internet File System*),
- AFP (eng. *Apple Filing Protocol*),
- NFS (eng. *Network File System*),
- RFS (eng. *Remote File System*) i
- SCP (eng. *Secure Copy*).

2.1. File Transfer Protocol (FTP)

FTP je standardni mrežni protokol za kopiranje datoteka s jednog računala na drugi putem mreže zasnovane na TCP/IP OSI modelu [10]. FTP je izgrađen na temelju klijent-poslužitelj arhitekture te koristi dva zasebna kanala za ostvarivanje komunikacije između njih: **podatkovni** i **kontrolni kanal**. FTP klijenti mogu se autentificirati na poslužitelju razmjenjujući autentifikacijske podatke u obliku otvorenog teksta ili se spojiti na njega kao anonimni korisnici ukoliko to poslužitelj dozvoljava. Obično u anonimnom pristupu korisnik unosi riječ „anonymous“ kao korisničko ime, a adresu svoje elektroničke pošte kao lozinku. Unesene podatke FTP poslužitelj uopće ne verificira te se stoga izbjegava takav način prijave korisnika ukoliko se želi uspostaviti relativno pouzdaniji i sigurniji pristup podacima putem ovog protokola.

Za FTP protokol rezervirani su TCP priključci 20 i 21. Komunikacija FTP protokolom uspostavlja se kada klijent s poslužiteljem uspostavi vezu odnosno sjednicu na priključku 21. Klijent ju ostvaruje slanjem zahtjeva PORT i dobivanjem potvrdnog odgovora od poslužitelja. Takva veza naziva se kontrolna veza odnosno kontrolni kanal, a ostaje otvorena tijekom čitavog trajanja sjednice. Kontrolni kanal se koristi za administriranje sjednice odnosno za razmjenu naredbi i identifikacijskih podataka između klijenta i poslužitelja koristeći protokol po funkcionalnosti sličan protokolu TELNET. Primjerice, naredba „RETR <naziv datoteke>“ predstavlja zahtjev klijenta za prijenos određene datoteke s poslužiteljskog na klijentsko računalo. Poslužitelj na tu i druge naredbe odgovara s troznamenkastim ASCII statusnim kodom i opcionalnim tekstom. Primjerice, ako poslužitelj odobri spomenuti RETR zahtjev njegov odgovor klijentu bit će poruka „200 OK“. Ta poruka označava uspješnu obradu prethodno zaprimljenog zahtjeva. Brojevi predstavljaju kodni broj, a tekst objašnjenje ili tražene parametre. Komunikacija kontrolnim kanalom odvija se na temelju čitavog niza različitih naredbi čiji se detaljni popis i opis nalazi na sljedećoj poveznici:

http://www.techmynd.com/all-essential-ftp-commands-list/

Po uspostavi kontrolnog kanala, pokreće se uspostava druge veze odnosno podatkovnog kanala. Njenu uspostavu može inicirati ili poslužitelj ili klijent ovisno da li se koristi aktivni ili pasivni način rada protokola FTP. U aktivnom načinu, klijent šalje zahtjev za uspostavu podatkovnog kanala koji sadrži njegovu IP adresu i broj priključka na kojem osluškuje konekciju, a poslužitelj kao odgovor na zahtjev uspostavlja TCP konekciju na svom priključku 20 ili odbija konekciju. Vrlo kvalitetna ilustracija ovog načina komunikacije između FTP klijenta i poslužitelja dostupna je na sljedećoj poveznici:

http://www.visualland.net/view.php?cid=1163&protocol=FTP&title=1.%20FTP%20Basics&ctype=1

U situacijama kad se klijent nalazi iza vatrozida i ne može primati dolazne TCP konekcije, koristi se pasivni način rada protokola FTP. U njemu klijent šalje PASV naredbu poslužitelju, a kao odgovor dobiva poslužiteljevu IP adresu i priključak (20). Klijent koristi te podatke za uspostavu podatkovnog kanala. Očito je da klijent pri uspostavi oba kanala koristi proizvoljne TCP priključke, a poslužitelj uvijek rezervirane priključke 20 i 21. Osim vatrozida, FTP protokolu probleme mogu zadavati i NAT (eng. *Network Address Translation*) uređaji koji također mogu sprječavati dolazne konekcije. Drugi problem za NAT uređaje jest činjenica da će klijenti, koji se u mreži nalaze iza njih, u RETR zahtjev zapisati svoju privatnu IP adresu i privatni priključak, a ne javnu IP adresu i javni priključak koji određuje NAT uređaj. Postoje dva rješenja tog problema. Jedan je da FTP klijent i poslužitelj koriste PASV naredbu odnosno pasivni način uspostave podatkovnog kanala. Drugi je da NAT uređaj mijenja vrijednosti zapisane u odaslanom PORT zahtjevu. Najčešće se koristi prvo rješenje.

Podaci razmijenjeni FTP protokolom mogu se slati u četiri različita formata:

- ASCII format - koristi se za tekst. Ukoliko je potrebno, pošiljateljevi podaci se transformiraju u osam bitni ASCII format prije prijenosa te se na odredištu transformiraju u format čitljiv primatelju. Zbog te transformacije ovaj format pogodan je samo za razmjenu tekstualnih datoteka.
- Binarni format (image format) - pošiljatelj šalje oktet po oktet datoteke, a primatelj pohranjuje tok okteta kako ga prima. Podrška za ovaj format preporučena je za sve implementacije protokola.
- EBCDIC format - koristi se za razmjenu otvorenog teksta između računala koja podržavaju EBCDIC[1] znakovni format, po svemu drugome identičan ASCII formatu.
- Lokalni format - dopušta dvama računalima s istim postavkama da razmjenjuju podatke u vlastitom formatu bez potrebe za transformacijom u ASCII format i natrag.

Razmjena podataka može se ostvariti na tri različita načina:

- **tok podataka** - podaci se prenose u obliku kontinuiranog toka pri čemu je FTP protokol oslobođen od bilo kakvog procesiranja podataka. Procesiranje odnosno obradu podataka radi protokol TCP.
- **blok podataka** - FTP protokol dijeli podatke u niz blokova te ih predaje dalje protokolu TCP. Svaki blok čini zaglavlje bloka, broj okteta i polje podataka.
- **komprimirani podaci** - podaci su komprimirani korištenjem određenog algoritma za kompresiju.

2.1.1. Sigurnosne ranjivosti FTP protokola

FTP nije dizajniran kao siguran protokol te tako posjeduje brojne sigurnosne slabosti. Te ranjivosti uključuju:

- napade koji omogućuju napadaču da pristupa priključcima FTP poslužitelja putem PORT naredbe indirektno koristeći računalo žrtvu kao posrednika njegovog PORT zahtjeva (eng. *Bounce attack*). Na taj način poslužitelj misli da komunicira s legitimnim klijentom na siguran način, a nije svjestan da zlonamjerni napadač ima uvid u komunikaciju [19].
- napade IP zavaravanja (eng. *IP spoofing*) koji predstavljaju slanje IP datagrama s lažnom adresom pošiljatelja odnosno lažnom adresom izvorišta paketa.
- napade „grubom silom“ (eng. *Brute Force Attack*) koji su zapravo pokušaj probijanja korisnikovog imena i lozinke isprobavanjem svih mogućih kombinacija znakova.

[1] EBCDIC je engleska skraćenica za Extended Binary Coded Decimal Interchange Code koja označava 8-bitni znakovni sustav. Navedeni kodni sustav razvio se od 6-bitnog BCD sustava za kodiranje bušenih kartica. Razvila ga je tvrtka IBM tokom kasnih 50-ih godina prošlog stoljeća za svoja prva računala.

- prisluškivanje prometa (eng. *traffic sniffing*) koje je temelj za brojne napade, uključuje postupak u kojem napadač postavlja svoju mrežnu karticu u promiskuitetni način rada te tako vidi sav mrežni promet u svom segmentu mreže.
- nedostatak zaštite korisničkog imena i lozinke.
- krađu priključka odnosno ARP napad putem kojeg LAN komutator šalje pakete napadačevom računalu, a ne legitimnom odredištu. Na taj način napadač dobiva neovlašten uvid u FTP komunikaciju [20].

FTP protokol ne predviđa šifriranje prometa. Čitav prijenos obavlja se u obliku otvorenog teksta te tako svatko tko posjeduje mogućnost snimanja i presretanja mrežnog prometa može jednostavno pratiti čitavu komunikaciju odnosno saznati korisnička imena, lozinke te dobiti pristup razmijenjenim podacima i naredbama. Ovaj problem je zajednički svim internetskim protokolima dizajniranim prije nego što su razvijeni enkripcijski mehanizmi poput protokola TLS ili SSL. Navedeni problem rješava se korištenjem sigurnosnog proširenja protokola FTP, a to je tuneliranje protokola pomoću protokola SSH ili protokol FTPS. Navedena rješenja opisana su u sljedećim potpoglavljima dokumenta. Alternativno rješenje jest odustajanje od FTP protokola te korištenje drugih protokola koji posjeduju istu funkcionalnost uz mnogo višu dostupnu razinu sigurnosti poput protokola SFTP ili SCP.

2.1.2. Budućnost protokola FTP

Može se reći da za izvornu inačicu protokola FTP budućnost nije blistava, s obzirom na nepostojanje sigurnosnih mehanizama koji su nužni u današnjoj komunikaciji putem Interneta. No, u kombinaciji sa sigurnosnim proširenjima (FTPS) ovaj protokol je i dalje koristan. Njegov razvoj još uvijek postoji u okviru IETF što se vidi po uvedenoj podršci za protokol IPv6 te po dva RFC dokumenta izdana u zadnje četiri godine koji sažimaju kozmetičke izmjene protokola iz zadnjih par godina. Spomenuti RFC dokumenti dostupni su na sljedećim poveznicama:

http://tools.ietf.org/html/rfc5797 i http://tools.ietf.org/html/rfc3659

2.1.3. FTP over SSH

Koncept *FTP over SSH* predstavlja praksu tuneliranja FTP sjednice preko SSH veze. Tuneliranje preko SSH protokola otežano je činjenicom da FTP istovremeno uspostavlja više TCP veza. Mnogi SSH klijenti samo tuneliraju početni kontrolni kanal, a kasnije uspostavljeni podatkovni kanali mimoiđu SSH tunel te tako nemaju osiguranu povjerljivost i integritet komunikacije. Kako bi se čitava komunikacija šifrirala SSH protokolom, SSH klijenti moraju poznavati način funkcioniranja FTP protokola, nadgledati i mijenjati poruke u FTP kontrolnom kanalu te autonomno otvarati nova prosljeđivanja odnosno tunele za FTP podatkovne kanale.

2.1.4. File Transfer Protocol over SSL (FTPS)

FTPS je proširenje protokola FTP u smislu dodane podrške za kriptografske protokole TLS (eng. *Transport Layer Security*) i SSL (eng. *Secure Sockets Layer*), uključujući i upotrebu PKI (eng. *Public Key Infrastructure*) infrastrukture [16]. Protokol FTP dizajniran je 1971. godine za korištenje u znanstvenoj i istraživačkoj mreži ARPANET. U to vrijeme pristup ARPANET-u bio je limitiran na mali broj vojnih sjedišta i sveučilišta zbog čega nisu postojali zahtjevi za implementaciju mehanizma koji će pružati sigurnost i privatnost prenesenih podataka. Kako se iz ARPANET-a razvio Internet, podaci su se počeli razmjenjivati po sve dužim rutama između klijenata i poslužitelja. S tim porastom proporcionalno je rasla mogućnost uplitanja (neželjenog) u razmjenu podataka. Kompanija Netscape 1994. godine razvija sigurnosni „omotač“ aplikacijskog sloja odnosno protokol SSL. Taj protokol omogućava aplikacijama da komuniciraju na siguran i povjerljiv način putem nesigurne mreže, onemogućujući prisluškivanje, krivotvorenje poruka i neovlašteni pristup podacima. SSL može pružiti sigurnosne mehanizme svakom protokolu koji se temelji na pouzdanoj konekciji (npr. TCP konekcija), ali najčešće se koristi u kombinaciji s protokolima HTTP (HTTPS) i FTP (FTPS). FTPS za podatkovni kanal koristi priključak 989, a za kontrolni 990 što omogućuje istovremeni rad FTP i FTPS protokola na istom poslužitelju.

Razvijene su dvije metode uspostave protokola FTPS, eksplicitna i implicitna. Prva podržava i sigurni i nesigurni oblik komunikacije (sa i bez SSL-a), a druga samo sigurnu komunikaciju. U eksplicitnom načinu rada klijent mora eksplicitno zatražiti uspostavu sigurne komunikacije od FTPS poslužitelja te ostvariti međusobno dogovorenu metodu enkripcije. Ukoliko klijent ne zatraži sigurnu komunikaciju, poslužitelj će klijentu dopustiti uspostavu nesigurne konekcije ili odbaciti konekciju. Kako bi se uspostavila sigurna konekcija, klijent uvijek mora zahtijevati od poslužitelja uspostavu međusobno poznatog sigurnosnog mehanizma korištenjem naredbe AUTH. Ukoliko klijent pošalje poslužitelju zahtjev za korištenje mehanizma koji mu nije poznat, poslužitelj će klijentu javiti poruku s kodom greške 504 *(not supported)*. Klijent može saznati koje mehanizme poslužitelj poznaje slanjem naredbe FEAT, no poslužitelj ne mora nužno biti potpuno iskren pri otkrivanju podataka o razini sigurnosti koju podržava. Uobičajeno FTPS poslužitelji podržavaju TLS i SSL mehanizme koji se pozivaju AUTH TLS odnosno AUTH SSL naredbama s time da se predlaže da klijenti uvijek pri pregovaranju o sigurnosnom mehanizmu koriste AUTH TLS naredbu. U ovom načinu rada klijent ima potpunu kontrolu nad enkripcijom sjednice odnosno određuje koji dio konekcije će biti šifriran. Klijent može u bilo kojem trenutku zatražiti uspostavu ili prekid šifriranja komunikacije putem podatkovnog i kontrolnog kanala. Jedino ograničenje predstavlja FTPS poslužitelj koji može odbiti određene zahtjeve na temelju vlastite sigurnosne politike.

U implicitnom načinu rada nije dopušteno pregovaranje o korištenju ili ne korištenju sigurnosnih mehanizama. Očekuje se da klijent pri inicijalizaciji konekcije poslužitelju pošalje TLS/SSL *ClientHello* poruku. FTPS poslužitelj odbija uspostaviti konekciju ukoliko ne primi tu poruku. U posljednjoj specifikaciji protokola FTPS ova metoda nije definirana pa se kao takva smatra odbačenom i izvan uporabe. U ovom načinu rada čitava FTPS sjednica je kriptirana.

Uspostava sigurne komunikacije putem protokola FTPS podrazumijeva šifriranje podatkovnog i kontrolnog kanala. Šifriranje komunikacije duž kontrolnog kanala uspostavlja se kada klijent poslužitelju pošalje AUTH TLS/AUTH SSL zahtjev. Preporučljivo je da se to dogodi prije autentifikacije i autorizacije korisnika kako bi se onemogućio neovlašten pristup korisničkom imenu i lozinki od treće strane. Klijent prekida šifriranje komunikacije u kontrolnom kanalu slanjem naredbe CCC (eng. *Clear Command Channel*). Šifriranje komunikacije duž podatkovnog kanala ne uspostavlja se nakon slanja naredbe AUTH TLS već klijent u tu svrhu mora poslati zasebnu naredbu PORT. Klijent prekida šifriranje komunikacije u podatkovnom kanalu slanjem naredbe CDC (eng. *Clear Data Channel*).

Koliko god je korištenje sigurne komunikacije nužnost, postoje određene situacije kada ju je bolje izbjeći. Za podatkovni kanal to su sljedeće situacije:

- ne razmjenjuju se osjetljivi podaci pa je šifriranje nepotrebno,
- razmjenjuju se podaci koji su već šifrirani na razini podatka pa je dodatno šifriranje redundantno,
- dostupni TLS ili SSL mehanizmi ne pružaju zadovoljavajuću razinu enkripcije. Takve situacije moguće su u starijim implementacijama FTPS klijenata ili poslužitelja.

Za kontrolni kanal to su sljedeće situacije:

- korištenje FTPS-a kada se klijent ili poslužitelj nalaze iza vatrozida ili NAT uređaja koji osluškuje kontrolni kanal kako bi dinamički otvorio priključak za podatkovni kanal (opisano detaljnije u nastavku teksta).
- anonimni klijenti unutar iste sjednice učestalo koriste AUTH i CCC/CDC naredbe. Takvo ponašanje se zna koristiti za resursno temeljen napad uskraćivanjem usluge (eng. *Denial of Service Attack, DoS Attack*) pošto SSL/TLS sjednica svaki put mora biti pokrenuta ispočetka, a to dodatno opterećuje procesor (odnosno zahtijeva dodatno procesiranje).

Protokol FTPS zna biti nekompatibilan s određenim implementacijama vatrozida u pojedinim mrežama. Naime, određeni vatrozidi znaju biti konfigurirani tako da snimaju FTP kontrolni kanal kako bi otkrili broj priključka kojeg će koristiti podatkovni kanal kako bi mogli dinamički propustiti promet na tom priključku. No, kod protokola FTPS to nije moguće zbog šifriranja kontrolnog kanala te je tako moguća situacija da u određenim mrežama neće prolaziti FTPS promet, a hoće obični FTP. Ovaj problem rješava se ručnom konfiguracijom dopuštenih priključaka koji će biti otvoreni na vatrozidu.

2.1.5. File eXchange Protocol (FXP)

FXP odnosno FXSP (eng. *FXP over SSL*) je metoda razmjene podataka koja koristi FTP za prijenos podataka od jednog udaljenog poslužitelja do drugoga, bez usmjeravanja podataka preko klijentske konekcije [15]. Uobičajena FTP komunikacija podrazumijeva postojanje jednog poslužitelja i jednog klijenta te razmjenu datoteka između njih. S druge strane, u FXP sjednici klijent održava standardnu FTP konekciju s dva FTP poslužitelja te može potaknuti bilo kojeg od tih poslužitelja da otvori konekciju prema drugom poslužitelju s ciljem inicijacije razmjene podataka. Prednost korištenja FXP protokola naspram FTP-a evidentna je kada poslužitelj sa širokim prijenosnim pojasom zahtjeva resurse od drugog poslužitelja s jednako širokim prijenosnim pojasom, a samo klijent s uskim prijenosnim pojasom, (primjerice administrator mreže prisutan na udaljenoj lokaciji) ima omogućen pristup resursima na oba poslužitelja. FXP će za tu razmjenu datoteka koristi poveznicu između poslužitelja koja je većeg kapaciteta od poveznica klijenta s poslužiteljima, a one bi se koristile u slučaju korištenja običnog FTP-a. Očito je da će komunikacija putem FXP-a biti brža i efikasnija od komunikacije putem FTP-a. Usprkos tome što se FXP može klasificirati kao samostalni protokol, on je zapravo samo proširenje protokola FTP.

Određene implementacije FTP poslužitelja poput glFTPd-a, cuftpd-a, RaidenFTPD-a i wzdftpd-a podržavaju pregovaranje o sigurnom podatkovnom kanalu između dva poslužitelja koristeći FTPS. Navedena komunikacija uspostavlja se korištenjem specifičnih naredbi CPSV ili SSCN. Spomenute naredbe klijent poslužitelju mora poslati prije nego što pošalje PASV naredbu te tako daje uputu poslužitelju da uspostavljenu podatkovnu konekciju zaštiti korištenjem TLS ili SSL mehanizma.

Ukoliko FTP poslužitelj omogući podršku za FXP postaje ranjiv na tzv. FTP *bounce* napad. Zbog toga većina implementacija FTP poslužitelja u inicijalnoj konfiguraciji ima isključeno FXP proširenje.

Uspostava FXSP sjednice, odnosno korištenje naredbi CPSV i SSCN, podložna je tzv. *Man In The Middle* napadu pošto oba FTP poslužitelja jedan drugome ne provjeravaju SSL certifikate. Tu činjenicu napadač može iskoristiti kako bi lažirao njihove certifikate te se lažno predstavljao tim poslužiteljima. Napadač se postavi usred komunikacije oba poslužitelja, lažira certifikate te se obojici poslužitelja predstavlja kao druga strana. Na taj način dobiva punu kontrolu nad komunikacijom, a poslužitelji toga uopće nisu svjesni jer smatraju da sigurno komuniciraju s drugom stranom.

2.1.6. File Service Protocol (FSP)

FSP je zamjena za protokol FTP temeljena na UDP (eng. *User Datagram Protocol*) protokolu [13]. Omogućuje anoniman pristup poslužitelju s manjim zahtjevima po sklopovlje računala i mrežu u usporedbi s FTP protokolom. S obzirom da koristi UDP umjesto TCP-a, izbjegava probleme koje mnogi FTP poslužitelji imaju s opterećenjem procesora i memorije (zbog potrebe za zasebnim procesom za svakog klijenta) te jednostavnije upravlja nastavkom prijenosa podataka zbog prekida izazvanog ispadom u mreži. IANA (eng. *Internet Assigned Numbers Asossiation*) nikad nije službeno priznala FSP te stoga nema rezervirani broj priključka. Pošto je on UDP ekvivalent FTP-a, službeni FSP poslužitelji koriste UDP priključak 21 koji je očiti ekvivalent FTP-ovom TCP priključku 21. Iz toga se može zaključiti da oba protokola mogu istovremeno raditi na istom poslužitelju bez ikakvog konflikta. Neslužbeni FSP poslužitelji znaju koristiti priključak 2121. FSP nikada nije uspio dostići popularnost protokola FTP u pogledu legitimne uporabe, ali je postao iznimno korišten početkom i sredinom 90-ih godina prošlog stoljeća od strane podzemnih sjedišta (eng. *underground sites*) koja su sadržavala pornografiju i piratske inačice komercijalne programske podrške. U te svrhe je korišten ponajviše zbog već navedene činjenice da FSP poslužitelj zahtjeva samo jedan proces što otežava njegovo uočavanje od strane administratora sustava, a pošto koristi UDP na transportnom sloju, jednako teško njegov promet uočava administrator mreže. Naposljetku, povećana upotreba vatrozida, smanjena upotreba korisničkih računa za pokretanje i izvršavanje poslužitelja ili većine klijenata te nedostatak potpore za FSP u *web* preglednicima uzrokovali su smanjenje njegovog korištenja.

2.1.7. UDP-based File Transfer Protocol (UFTP)

UFTP istovremeno označava i protokol i naziv programskog alata koji ga implementira [14]. UFTP je šifrirani višeodredišni mrežni protokol za razmjenu datoteka. Razvio i dizajnirao ga je Dennis Bush u svrhu ostvarenja iznimno efikasne, sigurne i pouzdane razmjene podataka u slučajevima kada se podaci razašilju većem broj računala, višeodredišno adresiraju i šalju ili prenose preko bežične poveznice (primjerice, satelitski prijenos). Međutim, u slučajevima prijenosa putem poveznica s malim brojem grešaka, sa širokim prijenosnim pojasom ili s velikim kašnjenjem, UFTP uspijeva postići barem dvostruko bolje performanse od protokola za razmjenu datoteka temeljenih na protokolu TCP (poput „običnog“ FTP-a). Temelji se na *Starburstovom* MFTP (eng. *Multicast File Transfer Protocol*) protokolu. Sigurna komunikacija ostvaruje se pomoću TLS mehanizma.

2.1.8. Trivial File Transfer Protocol (TFTP)

TFTP je protokol za razmjenu datoteka poznat po svojoj jednostavnosti [6]. TFTP je prvotno definirao IEN (eng. *Internet Experiment Note*) 1980. godine, a dijelom se temelji na ranijem protokolu EFTP (eng. *Easy File Transfer Protocol*) koji je bio dio protokolnog paketa PUP (eng. *PARC Universal Packet*). Većinom se koristi za automatiziranu razmjenu konfiguracijskih ili inicijalizacijskih datoteka između mrežnih čvorova u lokalnoj mreži. U usporedbi s FTP-om, TFTP je jako limitiran pošto ne pruža mogućnost autentikacije korisnika te se stoga rijetko koristi u interaktivnom obliku. Također, protokol nema nikakvih sigurnosnih mehanizama pa je njegova upotreba na Internetu opasna i ranjiva na razne sigurnosne prijetnje poput presretanja, izmjene ili ubacivanja novog malicioznog prometa. Stoga se isključivo koristi u privatnim, lokalnim mrežama. Iz tih razloga određene Unix implementacije protokola ograničavaju mogućnost razmjene podataka na samo jedan za tu svrhu konfiguriran direktorij te čitanje i pisanje datoteka koje imaju postavljenu globalnu mogućnost čitanja odnosno pisanja. Na transportnom sloju koristi UDP protokol te njegov priključak pod brojem 69. Dizajniran je kako bi bio malen i jednostavan za implementaciju pa stoga samo omogućuje preuzimanje i postavljanje datoteka ili elektroničkih poruka na udaljeni poslužitelj. Ne omogućuje prikaz sadržaja direktorija.

S obzirom na jednostavan dizajn, TFTP se implementira uz korištenje male količine memorije te je zbog toga koristan za inicijalizaciju računala poput usmjeritelja koji nemaju sklopove za pohranu podataka. Element je mrežnog inicijalizacijskog protokola PXE (eng. *Preboot Execution Environment*) gdje je implementiran u *firmware*-u (program ugrađen u sklopovlju) BIOS-a (eng. *basic input/output system*) mrežne kartice računala. Također, koristi se za prijenos male količine podataka između mrežnih čvorova poput *firmware*-a IP telefona ili slika operacijskog sustava kada se određeni udaljeni tanki klijent inicijalizira s mrežnog čvora ili poslužitelja. Početne faze mrežne instalacije određenih operacijskih sustava (Solaris Jumpstart, Red Hat Kickstart, Symantec Ghost, Windows NT Remote Installation Services) koriste TFTP za učitavanje jezgre operacijskog sustava.

TFTP definira tri načina razmjene podataka, a to su:

- netascii,
- oktetni te
- elektronička pošta.

Netascii je osam bitno proširenje sedam bitnog ASCII oblika koje uključuje osam kontrolnih znakova (npr. null znak te znakovi novog reda CR i LF). Oktetni način podržava osam bitni prijenos proizvoljnih znakova pri čemu primljena datoteka jest identična poslanoj. Točnije, ukoliko poslužitelj dobije oktetnu datoteku i vrati ju natrag pošiljatelju ona mora biti identična primljenom originalu. Oblik elektroničke pošte koristi netascii format pri čemu se datoteka prenosi elektroničkom poštom na adresu koja je zapisana kao naziv prenošene datoteke. Ovaj način razmjene datoteka proglašen je zastarjelim te se više ne koristi.

Svaka razmjena datoteka započinje sa zahtjevom za čitanje ili pisanje koji istovremeno predstavlja zahtjev za uspostavu konekcije. Ukoliko poslužitelj odobri zahtjev, konekcija prema klijentu se otvara te se datoteka šalje putem fiksnih blokova veličine 512 okteta. Svaki poslani podatkovni paket sadrži točno jedan blok podataka te mora biti potvrđen putem paketa potvrde kako bi se mogao poslati sljedeći paket u nizu. Paket veličine manje od 512 okteta predstavlja oznaku završetka razmjene podataka. Ukoliko se određeni paket izgubi u mreži, primatelju će isteći vremenska kontrola primitka tog paketa te će ponovno

poslati svoj posljednji poslani paket (potvrdu ili podatkovni paket). Time će pošiljatelju paketa dati do znanja da nije primio posljednje poslani paket (te će pošiljatelj ponovno poslati navedeni paket). Pošiljatelj mora pamtiti samo posljednji poslani paket za potrebe retransmisije (ponovnog slanja istog paketa) pošto posljednja primljena potvrda garantira da je primatelj primio sve prethodne pakete. Ovakav mehanizam moguć je zahvaljujući tome što TFTP numerira sve poslane pakete koji pripadaju istom slijedu paketa odnosno čine jednu datoteku. Vrijedi primijetiti da su oba čvora uključena u razmjenu ujedno primatelji i pošiljatelji. Jedan šalje podatkovne pakete i prima potvrde, dok drugi šalje potvrde i prima podatkovne pakete.

2.1.9. Simple File Transfer Protocol (SFTP)

SFTP je predložen kao nesigurni protokol za razmjenu datoteka čija je razina kompleksnosti manja od FTP-a, ali veća od TFTP-a [5]. Nije se uspio izboriti za šire i ozbiljnije korištenje na Internetu te mu je stoga IETF (eng. *Internet Engineering Task Force*) dodijelio status povijesnog protokola. Posjeduje isti akronim kao i protokol *SSH File Transfer Protocol*, no ne smije ga se miješati s tim protokolom jer se radi o dva posve različita koncepta. Za njegovu uspostavu rezerviran je priključak 115, sadrži skup od 11 mogućih naredbi i podržava tri tipa prijenosa podataka, a to su ASCII, binarni i kontinuirani. Implementacija binarnog i kontinuiranog tipa je ista za sve sustave čija je veličina riječi (eng. *word size*) višekratnik od osam. SFTP podržava sljedeće mogućnosti:

- Prijavu korisnika na temelju korisničkog identifikatora odnosno putem kombinacije korisničkog identifikatora i lozinke.
- Hijerarhiju direktorija.
- Upravljanje datotekama odnosno preimenovanje datoteka, njihovo brisanje, postavljanje, preuzimanje, preuzimanje s prepisivanjem te preuzimanje s pridodavanjem.

2.2. SSH File Transfer Protocol (SFTP)

SFTP je mrežni protokol koji omogućuje pristup datotekama, razmjenu datoteka i upravljanje datotekama putem pouzdanog toka podataka [11]. Razvio ga je IETF kao proširenje protokola SSH-2 (eng. *Secure Shell version 2*) koje pruža mogućnost sigurne razmjene datoteka. Iako je razvijen u kontekstu SSH-2 protokola, može se koristiti i u kombinaciji s različitim drugim protokolima poput sigurne razmjene podataka preko TLS protokola i razmjene upravljačke informacije u VPN (eng. *Virtual Private Network*) aplikacijama. Protokol pri svom radu podrazumijeva funkcioniranje iznad sigurnog kanala, primjerice SSH tunela, zatim da je poslužitelj već autentificirao klijenta te da je identitet klijentskog korisnika dostupan protokolu. SFTP radi na TCP priključku protokola SSH, a to je priključak 22. Protokol nije Internet standard, ali je uvelike korišten i implementiran. Tokom vremena protokol je unaprjeđivan te su objavljene njegove nove inačice. Trenutno je aktualna inačica 6 SFTP protokola nastala 2006. godine.

U usporedbi sa starijim SCP protokolom, koji omogućuje samo razmjenu datoteka, SFTP pruža čitav niz operacija nad udaljenim datotekama te po tom pitanju djeluje više kao protokol udaljenog datotečnog sustava. Spomenuti niz operacija uključuje mogućnost SFTP klijenta da nastavlja nedovršene razmjene datoteka, pregledava sadržaj direktorija te briše udaljene datoteke. SFTP nastoji biti više neovisan o platformama na kojima se koristi nego SCP. SCP se ponajviše implementira na Unix operacijskim sustavima, a SFTP je više manje dostupan na većini postojećih operacijskih sustava. SFTP nije FTP ostvaren iznad SSH protokola već posve novi protokol neovisan o FTP protokolu. SFTP sam po sebi nema mehanizme sigurnosti i autentifikacije korisnika već očekuje da će protokol na kojeg se oslanja osigurati navedene mehanizme. SFTP se najčešće implementira kao podsustav SSH-2 protokola, no moguće ga je implementirati i u kombinaciji s SSH-1 protokolom. Takva implementacija nije neovisna o platformi na kojoj se implementira pošto SSH-1 nema podršku za koncept podsustava. SFTP klijent koji se pokušava spojiti na takvu implementaciju mora poznavati putanju do binarne datoteke SFTP poslužitelja na poslužiteljskoj strani. Datoteke postavljene na SFTP poslužitelj mogu u svojim metapodacima sadržavati neke osnovne atribute poput vremenskih oznaka odnosno datuma nastanka originalne datoteke. Ova karakteristika prednost je SFTP protokola nad FTP protokolom koji bez dodatnih proširenja nema takve mogućnosti. Iako i SCP i SFTP koriste isti oblik SSH enkripcije, s istom razinom viška prenesene informacije (zbog šifriranja

tokom razmjene datoteka) obično je zbog dvosmjerne, cikličke prirode protokola SFTP, SCP prijenos brži. S druge strane zahvaljujući takvom dizajnu, SFTP omogućava prekidanje razmjene podataka bez prekidanja uspostavljenje sjednice, što kod protokola SCP nije moguće.

S obzirom da se ovaj protokol koristi u kombinaciji s SCP protokolom ili kao sigurna zamjena za protokol FTP, u njegovu budućnost ne treba sumnjati. Sve dok ne nastane protokol sigurniji i kvalitetniji od protokola SSH, protokol SFTP će se koristiti na jednak način kao što se i danas koristi.

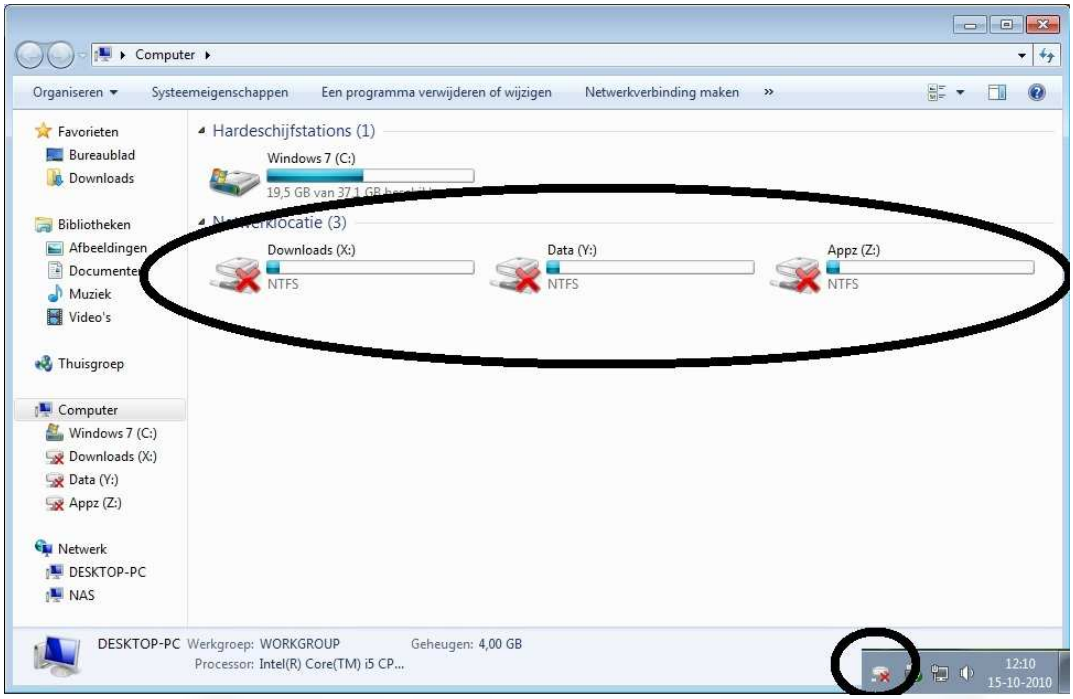
2.3. Server Message Block/Common Internet File System (SMB/CIFS)

SMB je aplikacijski protokol ponajviše korišten za pružanje dijeljenog pristupa datotekama, pisačima, serijskim priključcima i različitim oblicima komunikacije između mrežnih čvorova [7]. Također, pruža međuprocesni komunikacijski mehanizam s autentikacijom korisnika. SMB se ponajviše koristi u operacijskom sustavu Microsoft Windows, no razvijene su njegove implementacije za operacijske sustave Unix, Linux i Mac OS X. Razvio ga je Barry Feigenbaum u kompaniji IBM s ciljem pretvaranja DOS „Interrupt 33“ sustava za lokalni pristup datotekama u mrežni datotečni sustav. Daljnji razvoj protokola preuzeo je Microsoft, spojivši ga sa svojim LAN *Manager* proizvodom. Originalno je SMB bio dizajniran za rad temeljen isključivo na aplikacijskom programskom sučelju NetBIOS/NetBEUI koji nije prihvaćen u sklopu Interneta. Stoga je Microsoft bio primoran uvesti podršku za TCP protokol tako da se od 2000. godine SMB standardno na transportnom sloju oslanja na navedeni protokol te njegovi poslužitelji promet osluškuju na TCP priključku 445, dok klijenti koriste proizvoljni dinamički priključak.

Kao i FTP, SMB se zasniva na klijent-poslužitelj arhitekturi u kojoj klijent postavlja specifične zahtjeve, a poslužitelj na njih adekvatno odgovara. Određeni dijelovi SMB protokola upravljaju pristupom datotečnom sustavu tako da omogućuju klijentu slanje zahtjeva podatkovnom poslužitelju, dok drugi upravljaju međuprocesnom komunikacijom iz čega je nastao međuprocesni dijeljeni direktorij (eng. *Inter-Process Communication share, IPC share*). To je mrežni dijeljeni direktorij kojeg dijele računala umrežena protokolom SMB koja koriste operacijski sustav Microsoft Windows. Prikaz takvih direktorija u operacijskom sustavu Microsoft Windows 7 donosi slika 2. Takav virtualni dijeljeni direktorij koristi se za komunikaciju između procesa i računala putem protokola SMB te često za razmjenu podataka između autenticiranih računala. SMB se koristi za dijeljenje datoteka u lokalnim mrežama, ali i između različitih podmreža[2] u Internetu što često iskorištavaju napadi fokusirani na zlouporabu dijeljenja datoteka ili ispisa. SMB poslužitelji svoj datotečni sustav i druge resurse daju na raspolaganje klijentima putem mreže, a klijenti ostvaruju pristup tim dijeljenim resursima. To je osnovna funkcionalnost protokola SMB. No, ona ne bi bila moguća bez paketa protokola NT domene (eng. *NT domain suite of protocols*) koji u najmanju ruku osiguravaju NT domensku autentikaciju za validaciju korisničkog pristupa resursima.

Tijekom implementacije i nakon povećanja korištenja protokola, otkriveni su određeni propusti u njegovim performansama koji su naknadno ispravljeni. Kod prvih implementacija, korištenje SMB protokola je značilo povećanje razaslanog prometa unutar mreže. No, sam protokol nije razašiljao te podatke već je to radio NetBIOS protokol za otkrivanje i oglašavanje lokacije usluga. Problem je riješen korištenjem protokola DNS (eng. *Domain Name Service*) umjesto NetBIOS-a. Također, uočeno je kako se performanse protokola degradiraju, više nego kod konkurentnih protokola, u slučaju pojave kašnjenja odnosno latencije u komunikaciji. Latencija se pojavljuje pri komunikaciji između geografski udaljenijih lokacija, tipično prisutnih u WAN mrežama. Razlog leži u činjenici što inačica 1 SMB protokola prenosi podatke u obliku blokova veličine 64 kilobajta, a ne obliku toka. Također, problem je u tome što SMB sigurnosno potpisivanje blokova donosi višak informacija u komunikaciji te što veličina TCP prozora nije optimizirana za WAN poveznice. Predložena rješenja problema uključuju korištenje nove inačice protokola (inačica 2), *offline* datoteka, skaliranja TCP prozora te uređaja za ubrzavanje WAN komunikacije koji koriste privremenu memoriju i optimiziraju inačicu 1 SMB protokola.

[2] Internet je globalna mreža koja se sastoji od čitavog niza međusobno povezanih mreža. Takve mreže često se nazivaju podmreže u kontekstu Interneta, dok se manji dijelovi tih mreža također nazivaju podmreže u kontekstu tih mreža.



Slika 2. Dijeljeni diskovi protokola SMB u operacijskom sustavu Windows 7
Izvor: Synology Inc.

SMB posjeduje karakteristiku koja ga razlikuje od konkurentnih protokola, a zove se oportunističko zaključavanje. To je mehanizam zaključavanja datoteka s ciljem poboljšavanja performansi protokola kontrolirajući spremanje datoteka u privremenoj memoriji klijenata. Za razliku od tradicionalnog zaključavanja, oportunističko se ne koristi za međusobno isključivanje već za sinkronizaciju spremanja datoteka u privremenu memoriju. Postoje tri tipa oportunističkih semafora:

- ***batch* semafori** - ovi semafori se koriste kako bi se razriješio problem učestalog otvaranja i zatvaranja iste datoteke. Takvo ponašanje očito utječe na performanse protokola pošto se u kratkom periodu vremena mogu razmijeniti veće količine skoro pa identičnih informacija što predstavlja nepotrebnu redundanciju. Korištenjem ovog semafora klijent odgađa slanje zahtjeva za zatvaranje datoteke (pohranjuje se u privremenu memoriju). Ukoliko kasnije pošalje zahtjev za otvaranjem iste datoteke, zahtjevi se međusobno poništavaju, a mreža nije opterećivana nepotrebnim prometom.
- **ekskluzivni semafori** - kada klijent otvori neku dijeljenu datoteku na SMB poslužitelju, koju nitko drugi ne koristi, onda s njom dobiva i ovaj tip semafora. Zahvaljujući njemu, klijent sve promjene koje učini ne šalje poslužitelju već pohranjuje u svojoj privremenoj memoriji. Na taj način, ubrzava se rad protokola pošto nema mrežne komunikacije između klijenta i poslužitelja pri svakoj učinjenoj promjeni nad datotekom. Umjesto toga obavi se samo jedna razmjena podataka po završetku rada klijenta ili kada neki drugi klijent zatraži pristup toj datoteci. U potonjem slučaju, poslužitelj klijentu s ekskluzivnim semaforom šalje zahtjev za prekid kojim mu poništava posjedovanje semafora.
- **semafori druge razine** - ovaj tip semafora klijent može koristiti nakon što izgubi pravo na ekskluzivni semafor. Semafor druge razine omogućava mu da sve zahtjeve za čitanjem obavlja pomoću pohrane u privremenoj memoriji, ali sve zahtjeve za pisanjem u datoteku mora slati SMB poslužitelju.

Također, SMB poslužitelji koriste četvrtu vrstu poruke pri radu s oportunističkim zaključavanjem, a to su poruke prekida. One odskaču od uobičajene SMB komunikacije pošto ih šalje poslužitelj klijentu, a ne klijent poslužitelju. Tom porukom poslužitelj obavještava klijenta da oportunistički semafor kojeg je posjedovao više ne vrijedi, jer je drugi klijent zatražio pristup datoteci nad kojom je korišten semafor, te da sve lokalne promjene na korištenoj datoteci mora poslati poslužitelju uključujući potvrdu primanja prekidne poruke. Po primitku potvrde, poslužitelj dopušta rad nad tom datotekom drugom klijentu.

Microsoft je 2006. godine, zajedno s izdavanjem operacijskog sustava Windows Vista, izdao i novu inačicu (drugu) protokola SMB. Inačica 2 korigira sve mane prethodne inačice SMB protokola. Među korekcijama vrijedi istaknuti sljedeće:

- poboljšanje performansi protokola na poveznicama s velikom latencijom - omogućeno smanjenjem količine razmijenjenih poruka između klijenta i poslužitelja tijekom komunikacije. Navedeni napredak ostvaren je smanjenjem broja naredbi sa stotinjak na 19, uvođenjem cjevovoda odnosno slanjem dodatnih zahtjeva prije nego što dođe odgovor na prethodno poslani zahtjev te uvođenjem mogućnosti komponiranja većeg broja akcija u jedan odaslani zahtjev poslužitelju.
- poboljšanje performansi protokola pri prijenosu većih datoteka - ostvareno dopuštanjem da veličina korištenog međuspremnika (odnosno blokova kojima se razmjenjuju podaci) bude veća od 64 kilobajta.
- uvođenje trajnih *handleova* datoteka - omogućavaju robusnost SMB sjednice, odnosno da preživi gubitke mrežne konekcije karakteristične za bežične mreže bez potrebe za ponovnom pregovaranjem i uspostavom.
- ostale promjene koje uključuju podršku za simboličke poveznice, spremanje postavki datoteka u privremenu memoriju, poboljšano potpisivanje poruka putem algoritma za generiranje sažetka SHA-256, korištenje 32-bitnog i 64-bitnog datotečnog sustava te bolju skalabilnost sustava koja omogućuje veći broj aktivnih korisnika, dijeljenih datoteka i otvorenih datoteka po pojedinom SMB poslužitelju.

Ovaj protokol ima svijetlu budućnost pošto je popularan na Windows platformi te sa svakom novom verzijom operacijskog sustava Windows dolazi i nova inačica ovog protokola. S druge pak strane, zahvaljujući Sambi, implementaciji protokola za sve druge operacijske sustave, protokol se koristi na svim dostupnim platformama. To znači da ima iznimnu korisničku podršku, dostupne implementacije na raznim platformama i adekvatne sigurnosne mehanizme. Te tri karakteristike garancije su uspjeha protokola u budućnosti.

2.4. Secure Copy (SCP)

SCP je način sigurne razmjene računalnih datoteka između lokalnog i udaljenog računala, dva lokalna računala ili dva udaljena računala [17]. Pojam SCP odnosi se na dvije povezane stvari, SCP protokol i SCP program.

SCP je mrežni protokol koji podržava razmjenu datoteka na TCP priključku 22 koji pripada protokolu SSH. SCP protokol je zapravo BSD RCP (eng. *Remote File Copy*) protokol tuneliran pomoću SSH protokola (kako bi se osiguralo šifriranje i autentifikacija). Zbog toga se često SCP ne smatra zasebnim protokolom već kombinacijom protokola RCP i SSH. Protokol RCP obavlja razmjenu datoteka, a protokol SSH omogućuje autentikaciju i šifriranje komunikacije. Na taj način SCP štiti autentičnost i povjerljivost podataka koji se razmjenjuju te sprečava neovlašteno snimanje, kopiranje i otuđivanje razmijenjenih informacija. SCP, kao i svaki drugi protokol ovog tipa, omogućava postavljanje i preuzimanje datoteka s drugog računala koje sudjeluje u komunikaciji. Pri postavljanju datoteka, klijent šalje poslužitelju datoteke koje želi postaviti na poslužitelj. Opcionalno, može u prijenos uključiti osnovne atribute datoteka poput dozvola i vremenskih oznaka. Pri preuzimanju datoteka, klijent šalje zahtjev poslužitelju za određenim datotekama ili direktorijima koje želi preuzeti s poslužitelja. U slučaju preuzimanja direktorija, poslužitelj klijentu isporučuje sve poddirektorije i datoteke koje taj direktoriji i svi poddirektoriji sadrže. Ta karakteristika predstavlja sigurnosnu prijetnju ukoliko se preuzimaju podaci sa zlonamjernog poslužitelja. Naime, poslužitelj može u sklopu poddirektorija čuvati maliciozne datoteke poput trojanaca, virusa i *rootkita* koje klijent nije tražio, a mogu itekako ugroziti sigurnost njegovog operacijskog sustava.

SCP kao program objašnjen je u poglavlju „Implementacije mrežnih protokola za razmjenu datoteka“.

Protokol SCP dosta se koristi na operacijskim sustavima Linux/Unix za sigurnu razmjenu datoteka među klijentskim računalima. Također, njegova implementacija na operacijskom sustavu Windows, WinSCP, je iznimno popularna na toj platformi za istu svrhu. S obzirom na to, predviđa se svijetla budućnost protokola u postojećem obliku korištenja.

2.5. Network File System (NFS)

NFS je protokol mrežnog datotečnog sustava kojeg je 1984. godine razvila kompanija Sun Microsystems [8]. NFS dopušta korisniku na klijentskom računalu da pristupa datotekama putem mreže na način sličan pristupu datotekama pohranjenim na lokalnom računalu. NFS se temelji na sustavu ONC RPC (eng. Open *Network Computing Remote Procedure Call*) odnosno na metodologiji udaljenog poziva procedure te je kao i svi drugi protokoli otvoren standard. NFS se koristi na svim poznatim platformama s time da je najpopularniji na Linux/Unix sustavima dok su SMB odnosno AFP češće korišteni na operacijskim sustavima Windows i Mac OS. Najnovija inačica protokola NFSv4.1 objavljena je u siječnju 2010. godine. Svaka nova inačica dodatno je unaprijedila mogućnosti protokola.

Verzija 1 protokola korištena je samo unutar kompanije Sun. Čim je razvojni tim napravio značajnija unaprjeđenja te inačice i napravio ju dostupnom izvan kompanije nastala je inačica 2. Ona se originalno potpuno temeljila na protokolu UDP. Dizajneri protokola su time nastojali sačuvati karakteristiku protokola da i dalje ne pamti stanje sustava, a da se zaključavanje datoteka implementira izvan jezgre protokola. Protokol je mogao baratati datotekama maksimalne veličine 2GB.

Verzija 3 uvodi sljedeće karakteristike:

- potporu za 64-bitni datotečni sustav,
- upravljanje datotekama većim od 2GB,
- asinkrono pisanje datoteka na poslužitelju kako bi se poboljšale performanse pisanja,
- dodatne atribute datoteka u raznim odgovorima poslužitelja kako bi se izbjegla potreba za njihovim ponovnim dohvaćanjem te
- READDIRPLUS operaciju koja služi za dohvaćanje broja (eng. *file handle*) i atributa datoteka zajedno s njihovim nazivima tijekom pregleda direktorija.

Najvažnija razlika u odnosu na prošlu inačicu jest prestanak oslanjanja na UDP protokol te njegovo zamjenjivanje s protokolom TCP. Navedena promjena omogućila je kvalitetniju izvedbu NFS protokola u WAN (eng. *Wide Area Network*) mreži. Inačice 2 i 3 uključuju dodatno proširenje protokola nazvano „WebNFS“, koje omogućava lakšu integraciju s *web* preglednicima i slanje podataka kroz vatrozide.

Verzija 4, nastala pod utjecajima protokola AFS (eng. *Andrew File System*) i CIFS, zahtjeva jaku sigurnost, uvodi pamćenje stanja sustava odnosno sjednice te više poslužiteljski imenik. Ta inačica prva je koju nije razvio Sun već IETF (kojem je Sun prepustio daljnji razvoj protokola NFS). Najnovija inačica (4.1) nastoji iskoristiti prednosti implementacije grozdova poslužiteljskih računala uključujući sposobnost pružanja skalabilnog paralelnog pristupa podacima distribuiranim između većeg broja poslužitelja. Tu mogućnost ostvaruje pNFS (eng. *Parallel NFS*) proširenje protokola. Paralelni pristup podacima moguć je zahvaljujući odjeljivanju metapodataka datotečnog sustava od lokacije gdje su pohranjeni sami podaci koji su podijeljeni među podatkovnim poslužiteljima. To je poprilična promjena u odnosu na tradicionalni NFS koji drži podatke i njihove metapodatke u sklopu jednog poslužitelja. U starijim inačicama postoje poslužiteljski sustavi sačinjeni od više čvorova, no sudjelovanje klijenata u odjeljivanju metapodataka od podataka bilo je limitirano. U inačici 4.1 klijent može biti jedna od lokacija na kojoj se čuvaju podaci ili poznavati lokaciju te tako izbjeći interakciju s NFS poslužiteljom kada prenosi podatke. U toj inačici protokola, NFS poslužitelj je kolekcija poslužiteljskih računala koju kontrolira poslužitelj metapodataka. Klijent traži podatke preko poslužitelja metapodataka, a kad ih pronađe i kreće razmjenjivati direktno komunicira s podatkovnim poslužiteljima koji pripadaju pNFS kolekciji poslužitelja.

2.6. Remote File System (RFS)

RFS je bio distribuirani datotečni sustav kojeg je razvio AT&T u suradnji s kompanijom Bell Laboratories 80-ih godina 20. stoljeća [9]. Prvi put je implementiran u sklopu operacijskog sustava UNIX *System V Release 3*. Njegove karakteristike se podosta razlikuju od karakteristika protokola NFS. Umjesto fokusiranja na pouzdane operacije u prisutnosti kvarova, RFS se fokusirao na očuvanje semantike UNIX datotečnog sustava u sklopu mreže. Za razliku od NFS poslužitelja (prije NFS inačice 4), RFS poslužitelj je čuvao stanje sustava kako bi pratio koliko je puta određena datoteka otvarana, je li neki proces zaključao datoteku te slične podatke vezane uz postojeće datoteke.

Osnovne karakteristike RFS sustava su:

- pružanje potpune UNIX/POSIX datotečne semantike (zaključavane datoteka, pristup, brisanje, čitanje, mijenjanje datoteka, listanje direktorija, mijenjanje metapodataka, dozvola itd.),
- dopuštanje mrežnog podizanja (eng. *mount*) uređaja (npr. moguć udaljen pristup RFS direktoriju /dev/cdrom) te
- transparentan pristup datotekama (korisnik ne mora poznavati lokaciju datoteke).

2.7. Apple Filing Protocol (AFP)

AFP je mrežni protokol koji pruža datotečne usluge korisnicima koji koriste operacijske sustave Mac OS X [12]. Osim protokola AFP, operacijski sustav Mac OS X podržava njemu slične protokole SMB, FTP, NFS i WebDAV. AFP trenutno podržava Unicode (oblik zapisa teksta koji podržava znakove svih pisama koji postoje na svijetu) nazive datoteka, POSIX[3] dozvole te dozvole listi kontrole pristupa, imenovane proširene atribute, napredno zaključavanje datoteka te Mac OS konstrukciju za pohranu strukturiranih podataka u datoteci (eng. *Resource fork*). U Mac OS 9 i starijim inačicama AFP je bio primarni protokol za razmjenu datoteka.

Počevši od inačice 3.0, AFP protokol oslanja se isključivo na TCP protokol za uspostavu komunikacije, a AppleTalk protokol koristi samo kao protokol otkrivanja usluge. Verzije 2.x i starije koristile su oba protokola za uspostavljanje komunikacije. Upravo je ta promjena uzrok nedostatka interoperabilnosti starih i novih implementacija AFP poslužitelja. Zapravo, inačica 3.0 ovog protokola te sve nakon nje donose znatan napredak mogućnosti protokola dizajniranih isključivo za Mac OS X klijente. AFP inačica 3.0 je uvela model POSIX dozvola, Unicode UTF-8 kodiranje naziva datoteka te omogućila maksimalnu dopuštenu veličinu datoteke od dva tebibajta. AFP inačica 3.1 uvodi podršku za Kerberos sustav autentifikacije korisnika, automatsko ponovno spajanje klijenta, NFS i sigurne AFP konekcije temeljene na SSH protokolu. Maksimalna veličina datoteke je prvotno proširena na osam tebibajta, a kasnije i na 16 tebibajta. Verzija 3.2 uvodi podršku za liste kontrole pristupa i proširene atribute. Verzija 3.2+ donosi osjetljivost na velika i mala slova u imenima datoteka te poboljšava podršku za Mac OS *Time Machine* funkcionalnost (sustav arhive). Konačno, inačica 3.3 dodaje podršku za *Replay Cache* funkcionalnost (služi AFP poslužitelju za pohranu odaslanih odgovora na zahtjeve klijenta) koja je potrebna za *Time Machine* sustav.

2.7.1. Budućnost protokola AFP

AFP se izričito koristi na operacijskom sustavu Mac OS, s time da se sve više kao njegova alternativa na tom sustavu koriste protokoli SMB i NFS. Zbog toga je njegova budućnost neizvjesna. Nema sumnje da će Apple i dalje pružati podršku za taj protokol, no pitanje je koliko će sami korisnici koristiti navedeni protokol. Naime, problem je u tome što protokol mogu koristiti samo Mac OS korisnici u međusobnoj komunikaciji. Ukoliko požele razmjenjivati datoteke s korisnicima drugih platformi, moraju koristiti neki drugi protokol poput protokola SMB, FTP ili NFS pa se postavlja legitimno pitanje zašto uopće koristiti protokol AFP kad sve ono što on nudi i više pružaju konkurentski protokoli.

2.8. Usporedba opisanih mrežnih protokola za razmjenu datoteka

S obzirom da su u dosadašnjem tekstu već međusobno uspoređivani pojedini opisani protokoli, ovo poglavlje neće ponoviti već napisano nego donosi tabličnu usporedbu svih obrađenih protokola po procijenjenim najvažnijim karakteristikama mrežnim protokola za razmjenu datoteka. Na taj način na jednom mjestu postoji sustavan pregled zajedničkih karakteristika, ali i razlika pojedini protokola. Navedeni pregled donosi tablica 1.

[3] POSIX je zajednički naziv za skupinu IEEE (eng. *Institute of Electrical and Electronics Engineers*) standarda kojima se definira sučelje za programiranje aplikacija (eng. *Application Programming Interface, API*) usklađenih s različitim izvedenicama operacijskog sustava UNIX.

Tabela 1: Usporedba protokola za razmjenu datoteka

	FTP	FTPS	FXP	FSP	UFTP	SFTP	TFTP	SSHFTP	SMB	SCP	NFS	RFS	AFP
autentifikacija korisnika	da	da	da	da	da	da	ne	da	da	da	da	da	da
povjerljivost komunikacije	ne	da	ne	ne	ne	ne	ne	da	da	da	da	da	da
autentičnost, integritet, neporecivost komunikacije	ne	da	ne	ne	ne	ne	ne	da	da	da	da	da	da
klijent-poslužitelj arhitektura	da	da	da	da	da	da	da	da	da	ne	da	da	da
dijeljenje datoteka među klijentima	ne	ne	ne	ne	ne	ne	ne	ne	da	ne	da	da	da
transportni protokol	TCP	TCP	TCP	UDP	UDP	TCP	UDP	TCP	TCP	TCP	TCP	TCP	TCP
podržani operacijski sustavi	svi	svi	svi	svi	svi	svi	svi	svi	svi	svi	svi	Unix	MAC
protokol je aktivan	da	da	ne	ne	ne	ne	da	da	da	da	da	ne	da
sigurnosni protokol	nema	SSL	SSL	nema	TLS	nema	nema	SSH	NT domene	SSH	SSH	nema	SSH

3. Implementacije mrežnih protokola za razmjenu datoteka

U nastavku poglavlja dan je pregled najkorištenijih besplatnih implementacija klijentskih i poslužiteljskih aplikacija pet najvažnijih mrežnih protokola za razmjenu datoteka obrađenih o ovom dokumentu (FTP, SFTP, SMB, SCP i NFS). RFS je ipak rijetko korišten protokol te je obrađen iz povijesnih razloga i poveznice s protokolom NFS, a AFP se samo koristi u operacijskom sustavu MAC OS X pa stoga nisu prikazane implementacije za te protokole.

3.1. Implementacija protokola FTP

U ovom poglavlju opisane su tri implementacije protokola FTP, jedna implementacija FTP klijenta te dvije implementacije FTP poslužitelja.

3.1.1. FileZilla klijent

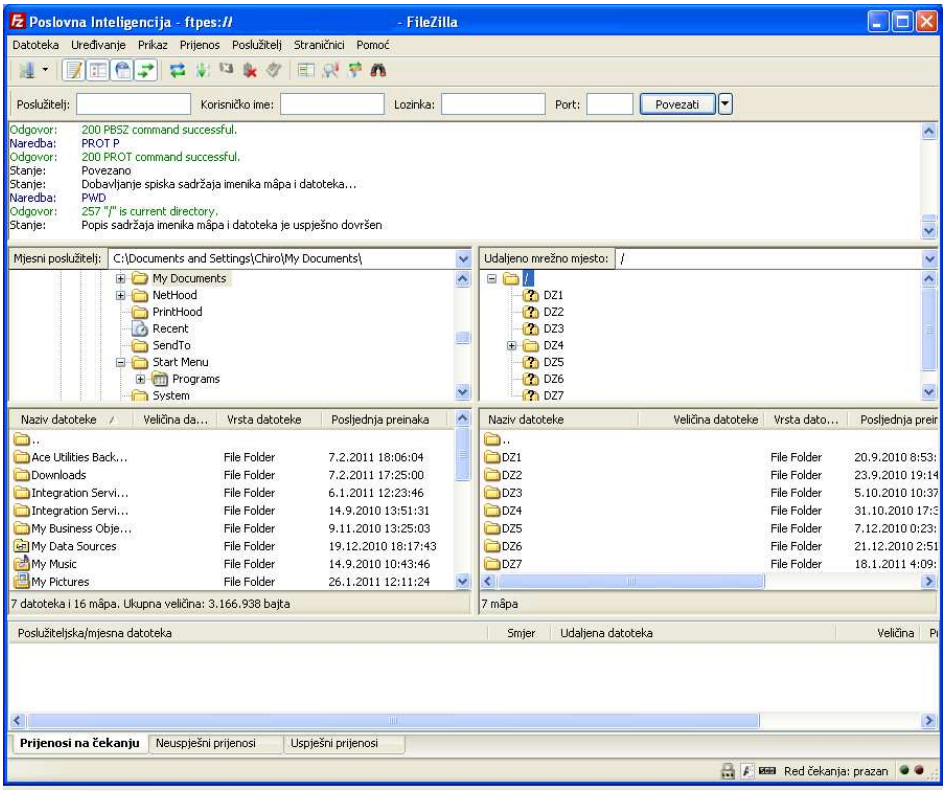
FileZilla je besplatni, višeplatformski programski alat otvorenog koda koji implementira protokol FTP te se sastoji od FileZilla klijenta i poslužitelja (opisan u sljedećem potpoglavlju). Instalacijske datoteke alata dostupne su za operacijske sustave Windows, Linux i Mac OS X [23]. Alat omogućuje komunikaciju putem protokola FTP, SFTP i FTPS. U listopadu 2010. godine utvrđeno je da je FilleZilla klijent sedmi najviše preuzeti alat sa stranice SourceForge.net. Zanimljivost vezana uz alat jest da je nastao kao studentski projekt Tima Kossea i dvojice njegovih kolega na kolegiju računalne znanosti te da je projekt otvorenog koda pošto njegovi autori, zbog mnogobrojne konkurencije, nisu vjerovali da bi uspjeli prodati niti jedan primjerak programa ukoliko bi bio komercijalan. Najnovija stabilna inačica programa jest inačica 3.3.5. koja je dostupna na sljedećoj poveznici:

http://filezilla-project.org/download.php?type=client

Najvažnije odlike FileZilla klijenta su sljedeće:

- **upravitelj mrežnih mjesta** - omogućuje korisniku da stvori listu FTP sjedišta pamteći njihove konekcijske podatke poput broja priključka, tipa protokola te tipa prijave na poslužitelj (anonimno ili uobičajeno). Za uobičajeni način pamti se korisničko ime te opcionalno lozinka.
- **usporedba imenika mapa i datoteka** - omogućuje usporedbu lokalnih i udaljenih direktorija.
- **dnevnik poruka** - prikazan je pri vrhu prozora grafičkog sučelja alata. Prikazuje konzolski ispis naredbi koje šalje FileZilla te odgovore udaljenog poslužitelja.
- **pregled direktorija i datoteka** - nalazi se ispod dnevnika poruka, predstavlja grafičko sučelje za FTP. Korisnici mogu pregledavati direktorije te mijenjati njihov sadržaj bilo na lokalnom ili na udaljenom računalu koristeći sučelje slično programu Windows Explorer. Korisnici vrlo jednostavno mogu grafički prebacivati datoteke s lokalnog na udaljeno računalo te u obratnom smjeru.
- **red za čekanje prijenosa** - nalazi se pri dnu prozora grafičkog sučelja alata, a prikazuje status svakog aktivnog prijenosa datoteka ili prijenosa koji je na čekanju u realnom vremenu.

Slika 3. prikazuje izgled grafičkog sučelja alata u kojem se može primijetiti dio opisanih odlika.



Slika 3. Grafičko sučelje alata Filezilla

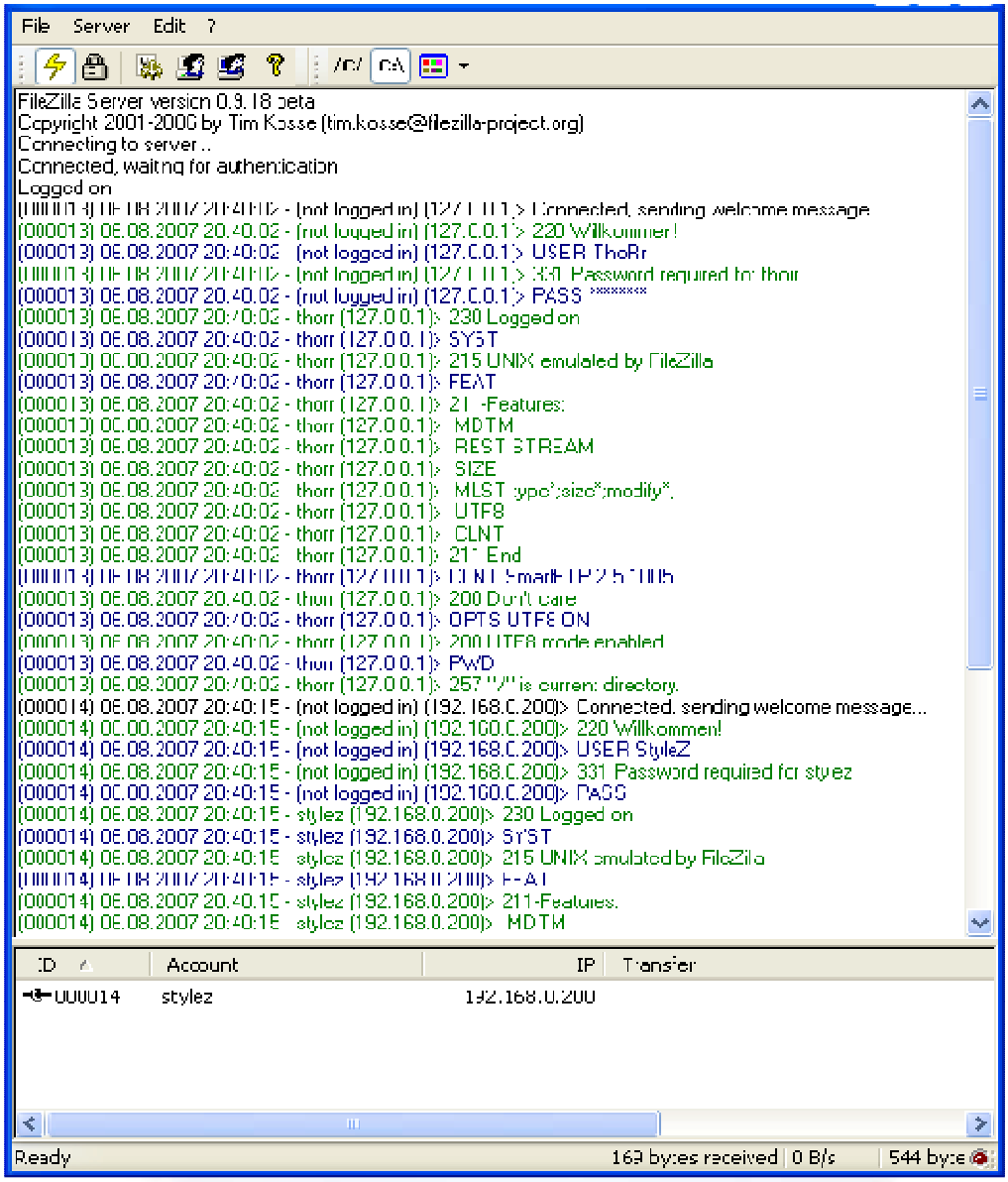
Alat posjeduje određena ograničenja. Radi se o sljedećim stavkama:

- **skriptiranje** - ne podržava skriptno pozivanje i rukovanje putem komandne linije već samo ručno, interaktivno putem grafičkog sučelja.
- **Windows 9x -** od inačice 2.2.23 FileZilla interno koristi Unicode format te kao rezultat toga ne može raditi na operacijskim sustavima Windows 9x/ME.
- **Mac OS X** - trenutna inačica alata ne podržava inačicu MAC OS X v10.4 niti sve starije od nje. Kako bi alat mogao raditi na tim inačicama potrebno ga je ručno prevesti (eng. *compile*) iz izvornog koda.
- **vremenske oznake pri preuzimanju i postavljanju datoteka na poslužitelj** - kod postavljenih datoteka na FTP ili SFTP poslužitelj, vremenskim oznakama se može pristupiti samo ako poslužitelj podržava MFMT (eng. *Modify Fact: Modification Time*) naredbu. Kod preuzetih datoteka, do vremenskih oznaka nastanka datoteka može se doći ukoliko datotečni sustav na kojem su pohranjene podržava vremenske oznake datuma i vremena nastanka datoteke. Primjerice, datotečni sustavi FAT32 i NTFS ih podržavaju.

3.1.2. FileZilla poslužitelj

FileZilla poslužitelj je besplatni FTP poslužitelj otvorenog koda koji radi na operacijskom sustavu Microsoft Windows [22]. FileZilla poslužitelj podržava protokole FTP i FTPS. Uključuje čitav niz funkcionalnosti među kojima vrijedi istaknuti sljedeće:

- ograničenja prijenosnog pojasa u oba smjera komunikacije (preuzimanje i slanje),
- komprimiranje podataka,
- šifriranje pomoću protokola SSL i TLS za potrebe FTPS protokola,
- dnevnik poruka korišten za praćenje rada poslužitelja i informacije o prometu u realnom vremenu,
- mogućnost ograničavanja dozvole pristupa samo na interni LAN (eng. *Local Area Network*) promet ili samo vanjski promet te
- mogućnost korištenja virtualnog datotečnog sustava.

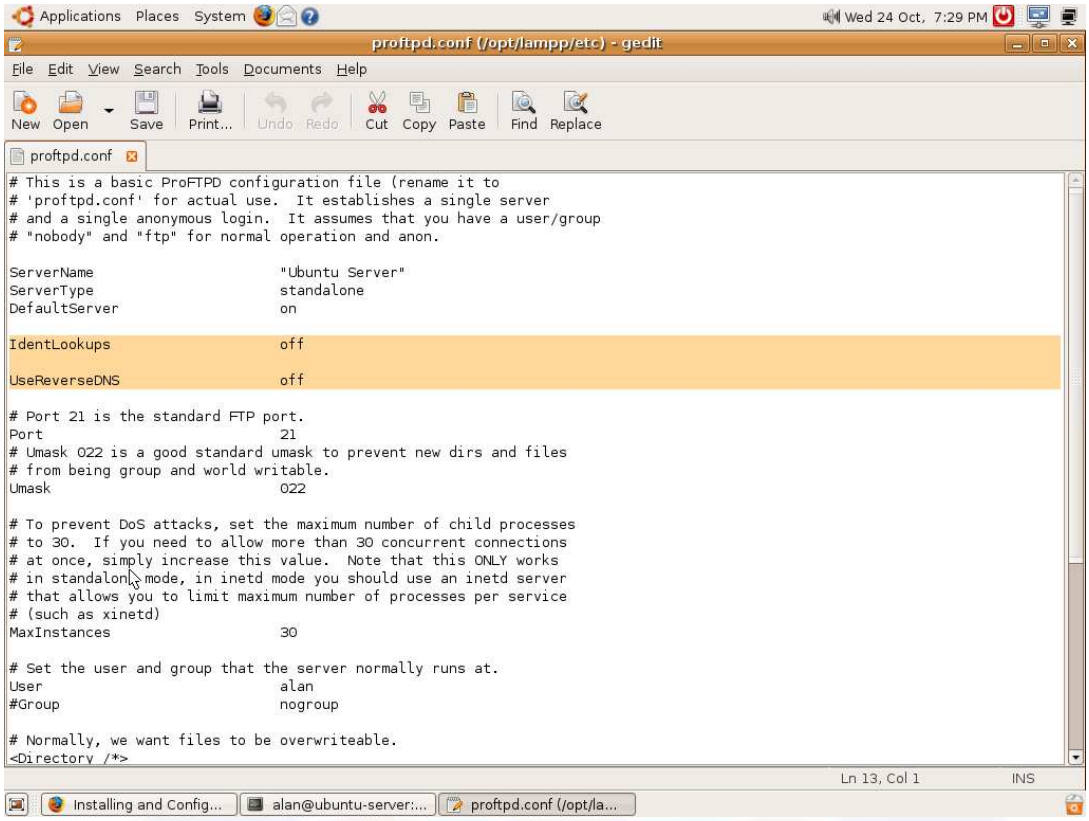


Slika 4. Grafičko sučelje FileZilla poslužitelja
Izvor: Wikipedia

Upravitelj korisničkom sjednicom, prikazan pri dnu prozora grafičkog sučelja FileZilla poslužitelja (slika 4.), omogućuje administratoru sustava uvid u popis trenutno aktivnih korisnika te u njihove transakcije (odnosno u status aktivnih prijenosa podataka). Trenutno postoje dvije operacije koje administrator može obaviti nad aktivnim korisnicima ukoliko zloupotrebljavaju svoje ovlasti. Može poništiti korisničku sjednicu ili zabraniti pristup poslužitelju s određene korisničke IP adrese.

3.1.3. ProFTPd poslužitelj

ProFTPd je konfigurabilni FTP poslužitelj namijenjen operacijskim sustavima Linux i Unix [21]. Koristi samo jednu konfiguracijsku datoteku „/etc/proftpd.conf“. Sadržaj te datoteke prikazan je na slici 5.



Slika 5. Konfiguracijska datoteka ProFTPd poslužitelja
Izvor: Alan Edwardes blog

Može se koristiti za jednostavnu konfiguraciju većeg broja virtualnih FTP poslužitelja te, ovisno o korištenom datotečnom sustavu, može imati tzv. *chroot* mogućnost (stvaranje većeg broja virtualnih root direktorija). ProFTPd može raditi kao samostalni poslužitelj ili kao inetd[4] usluga te ima podršku za rad u IPv6 okruženju. Njegova je arhitektura modularna što je omogućilo razvoj različitih proširenja poslužitelja u obliku modula. Popularni su moduli za SSL/TLS šifriranje te podrška za RADIUS (eng. *Remote Authentication Dial In User Service*), LDAP (eng. *Light Directory Access Protocol*) protokole.

ProFTPd poslužiteljem se upravlja putem komandne linije (odnosno ljuske operacijskog sustava), no razvijeno je i grafičko sučelje pod nazivom gProFTPd za korisnike koji preferiraju grafička sučelja.

3.2. Implementacija protokola SFTP

Pojam SFTP klijent može se odnositi na „*Secure file transfer program*“. To je naredba komandne linije koja implementira funkcionalnost SFTP klijenta sličnu onoj koja je dostupna u kombinaciji s alatom OpenSSH. Zapravo je riječ o programu sftp koji je također naredba komandne linije te pruža interaktivno sučelje slično sučelju tradicionalnih FTP klijenata. Čak i određene implementacije SCP protokola koriste protokol SFTP za razmjenu datoteka. Međutim, takve implementacije posjeduju sposobnost povratka na SCP protokol u slučaju da poslužitelj, s kojim se komunicira, ne podržava SFTP uslugu.

Što se tiče SFTP poslužitelja, postoji niz implementacija za UNIX i Windows operacijske sustave. Najpoznatija implementacija jest besplatni programski paket OpenSSH.

Značajke alata OpenSSH su sljedeće [24]:

- projekt otvorenog koda,
- osigurava pouzdane mehanizme (algoritme) šifriranja, primjerice, trostruki DES (eng. *Data Encryption Standard*), Blowfish, AES (eng. *Advanced Encryption Standard*), Arcfour,

[4] Inetd je poslužiteljski pozadinski proces, na mnogim Unix operacijskim sustavima, koji upravlja internetskim uslugama.

- X11 prosljeđivanje (šifriranje prometa X Window sustava),
- prosljeđivanje priključaka (šifriranje kanala za naslijeđene protokole),
- pouzdana autentifikacija korištenjem mehanizama poput javnog i privatnog ključa, lozinke za jednokratnu uporabu te Kerberos autentifikacije,
- interoperabilnost odnosno podrška za inačice 1.3, 1.5 i 2.0 protokola SSH,
- podrška za SFTP klijente i poslužitelje koji su bazirani na SSH1 i SSH2 protokolima te
- komprimiranje podataka.

3.3. Implementacija protokola SMB

Samba je besplatna implementacija protokola SMB/CIFS koja uspješno radi na različitim operacijskim sustavima [18]. Od svoje inačice 3, Samba pruža usluge dijeljenja datoteka i pisača za različite Microsoft Windows klijente, može se integrirati s Windows Server domenom te biti dio Active Directory domene. Samba radi na većini Unix sustava poput operacijskih sustava GNU/Linux, Solaris, AIX i BSD varijanti, uključujući i Mac OS X Server. Također, standardna je na skoro svim distribucijama operacijskog sustava Linux. Samba poslužitelj ne može se koristiti na operacijskom sustavu Microsoft Windows. No, kao što je već rečeno, interoperabilan je s računalima na kojima je implementiran taj operacijski sustav. Sam naziv implementacije izveden je iz kratice protokola kojeg implementira.

Samba omogućuje dijeljenje datoteka i pisača između računala koja rade na operacijskim sustavima Microsoft Windows i Unix. Implementacija je većeg broja usluga i protokola:

- NetIOS preko TCP/IP infrastrukture (NBT),
- SMB/CIFS,
- MSRPC (eng. *Microsoft Remote Procedure Call*),
- paket protokola Network Neighborhood,
- WINS (eng. *Windows Internet Naming Service*) poslužitelj,
- paket protokola NT domene,
- modificirana inačica Kerberos sustava te
- modificirana inačica protokola LDAP.

Samba omogućava mrežno dijeljenje odabranih Unix direktorija, uključujući njihove poddirektorije. Korisnicima operacijskog sustava Windows, ti direktoriji se prikazuju poput normalnih Windows direktorija dostupnih preko mreže. Unix korisnici mogu dijeljene direktorije dodati u njihovu datotečnu strukturu pomoću naredbe smbmount ili koristeći alat smbclient (dolazi u paketu sa Sambom), zatim ih mogu čitati putem sučelja sličnog standardnim FTP programima komandne linije. Svaki direktorij može imati drugačije privilegije pristupa postavljene preko standardne Unix datotečne zaštite. Primjerice, *home* direktoriji imaju omogućeno čitanje i pisanje za sve poznate korisnike koji tako mogu pristupiti vlastitim datotekama. No, nemaju pravo pristupa datotekama drugih korisnika u tim direktorijima ukoliko to dopuštenje nije eksplicitno postavljeno.

Samba za svoj rad koristi dva pozadinska procesa:

- smbd - koji pruža usluga dijeljenja datoteka i pisača te
- nmbd - koji pruža uslugu mapiranja NetBIOS računalnih imena u IP adrese (ovo mapiranje ekvivalentno je mapiranju kod protokola DNS, samo što se koristi protokol NetBIOS, znači IP adresa zamjenjuje se simboličkim imenom i obratno).

Samba se konfigurira izmjenom njene konfiguracijske datoteke koja se obično nalazi u direktoriju /etc/smb.conf (ili /etc/samba/smb.conf). Samba može pružati sučelja za prijavu korisnika i skripte za implementaciju grupne politike pomoću naredbe *poledit*. Vrijedi spomenuti da Samba paket uključuje alat Samba Web Administration Tool (SWAT), koji se koristi za administraciju putem web preglednika.

3.4. Implementacija protokola SCP

SCP kao program je alat koji implementira SCP protokol u obliku pozadinskog procesa (eng. *service demon*) ili u obliku klijenta [17]. Naime, SCP program je istovremeno i SCP poslužitelj i

SCP klijent. Najčešće korišteni SCP alat jest naredba scp, prisutna u većini SSH implementacija. Ta naredba je zapravo analogna naredbi rcp uz sigurnosne dodatke. S obzirom da ta naredba pruža i funkciju SCP poslužitelja, mora biti instalirana na svim SSH poslužiteljima koji žele pružati SCP uslugu svojim klijentima. Određene implementacije SSH poslužitelja koriste naredbu scp2 koja koristi protokol SFTP umjesto protokola SCP. Navedena naredba ponaša se i koristi na jednak način kao i naredba scp.

Što se same sintakse naredbe tiče, ona je jednaka sintaksi naredbe cp.

Ukoliko se želi prebaciti neku datoteku s poslužitelja na klijentsko računalo, potrebno je izvršiti sljedeću naredbu:

```
scp izvorišna_datoteka
korisnik@računalo:odredišni_direktorij/odredišna_datoteka
```

Ukoliko se želi prebaciti neku datoteku s klijentskog računala na poslužiteljsko, izvršava se sljedeća naredba:

```
scp korisnik@računalo:/izvorišni_direktorij/izvorišna_datoteka
odredišna_datoteka
```

S obzirom da SCP protokol implementira samo razmjenu datoteka, klijenti s grafičkim sučeljem su vrlo rijetki (pošto bi njihova implementacija zahtijevala uvođenje dodatnih funkcionalnosti poput izlistavanja sadržaja direktorija). Primjerice, WinSCP klijent se zapravo oslanja na protokol SFTP. Čak i kad rade u SCP načinu rada, klijenti poput WinSCP alata nisu potpuni SCP klijenti jer koriste druge protokole za implementaciju dodatnih funkcionalnosti koje posjeduju. Ta karakteristika pak izaziva probleme s platformskom neovisnošću. Tako su moguće situacije da se s određenim SCP poslužiteljem istovremeno ne može komunicirati putem SCP klijenta s grafičkim sučeljem, a može putem tradicionalnog klijenta u obliku naredbe ljuske operacijskog sustava.

3.5. Implementacija protokola NFS

Ukoliko se pretpostavi tipični Unix scenarij da jedno računalo (klijent) zahtijeva pristup podacima pohranjenim na drugom računalu (NFS poslužitelj), onda se implementacija protokola NFS sastoji od sljedećih koraka [8]:

1. Poslužitelj implementira NFS pozadinski proces (obično nfsd) kako bi učinio svoje podatke dostupnim klijentima.
2. Administrator poslužitelja odlučuje što će točno biti dostupno klijentima, objavljujući u mreži imena i parametre dostupnih direktorija. Obično za taj postupak koristi konfiguracijsku datoteku „/etc/exports“ i naredbu komandne linije exportfs.
3. Sigurnosna administracija poslužitelja osigurava da poslužitelj može prepoznati i odobriti autorizirane klijente.
4. Mrežna konfiguracija poslužitelja osigurava da odgovarajući klijenti mogu pregovarati s njime bez obzira na korištene vatrozide.
5. Klijentsko računalo traži pristup objavljenim podacima, obično koristeći *mount* naredbu. U tom procesu klijent mora propitati poslužiteljsko računalo o priključku na kojem promet osluškuje NFS poslužitelj te se spojiti na navedeni poslužitelj (nfsd). Sam *nfsd* potom mora primljeni zahtjev za podizanjem datotečnog sustava predati *mountd* procesu.
6. Ako čitav postupak protekne uspješno, korisnici na klijentskim računalima mogu koristiti podignuti datotečni sustav na poslužitelju unutar dopuštenih parametara.

Sam proces podizanja i uspostave datotečnog sustava NFS poslužitelja može se automatizirati, primjerice, korištenjem datoteke „/etc/fstab“.

4. Zaključak

Na kraju se može zaključiti da je dalek put iza mrežnih protokola za razmjenu datoteka. Njihovi počeci su ujedno bili i počeci samog razvoja internetskih protokola. Primjerice, FTP je svakako jedan od najstarijih aplikacijskih protokola. Zbog tako ranog razvoja, njihova specifikacija je s vremenom zastarjela te sadržavala čitav niz sigurnosnih ranjivosti. To je potaklo njihovo poboljšanje, razvoj sigurnosnih proširenja ili nastanak posve novih protokola čiji će temelji počivati na sigurnoj i pouzdanoj komunikaciji. Tako se na temelju protokola FTP razvio čitav niz protokola (SFTP, SCP, UFTP, TFTP, FSP, FXP, FTPS, AFP, SMB, NFS, RFS). Neki od tih protokola su se pokazali uspješnima te se i dalje razvijaju. Dapače, čak se koriste i više od njihovog pretka. Drugi pak nisu uspjeli pronaći svoj put do željenih korisnika (ili se situacija na Internetu odigrala u korist njihovih konkurenata) pa su se prestali koristiti i razvijati.

Svaki od obrađenih protokola posjeduje čitav niz implementacija, a najpoznatije i najkorištenije od njih opisane su u ovom dokumentu. Navedene implementacije prate razvoj i napredak svojih protokola. Uglavnom, mrežni protokoli za razmjenu datoteka zbog svoje funkcionalnosti i mogućnosti koje pružaju bili su važan dio TCP/IP OSI modela , još uvijek jesu i vrlo vjerojatno će biti i u daljnjoj budućnosti razvoja Interneta.

5. Reference

[1] Wikipedia: Protokol, http://hr.wikipedia.org/wiki/Protokol, svibanj 2009.

[2] Wikipedia: OSI model, http://en.wikipedia.org/wiki/OSI_model, veljača 2011.

[3] SearchNetworking.com Definitions: File transfer, http://searchnetworking.techtarget.com/definition/file-transfer, travanj 2001.

[4] Wikipedia: File Transfer, http://en.wikipedia.org/wiki/File_transfer, veljača 2011.

[5] Wikipedia: Simple File Transfer Protocol, http://en.wikipedia.org/wiki/Simple_File_Transfer_Protocol, rujan 2010.

[6] Wikipedia: Trivial File Transfer Protocol, http://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol, veljača 2011.

[7] Wikipedia: Server Message Block, http://en.wikipedia.org/wiki/Server_Message_Block, veljača 2011.

[8] Wikipedia: Network File System, http://en.wikipedia.org/wiki/Network_File_System_(protocol), veljača 2011.

[9] Wikipedia: Remote File System, http://en.wikipedia.org/wiki/Remote_File_System, siječanj 2011.

[10] Wikipedia: File Transfer Protocol, http://en.wikipedia.org/wiki/File_Transfer_Protocol, veljača 2011.

[11] Wikipedia: SSH File Transfer Protocol, http://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol, veljača 2011.

[12] Wikipedia: Apple Filing Protocol, http://en.wikipedia.org/wiki/Apple_Filing_Protocol, veljača 2011.

[13] Wikipedia: File Service Protocol,

http://en.wikipedia.org/wiki/File_Service_Protocol, veljača 2010.
[14] UFTP - Encrypted UDP based FTP with multicast, http://www.tcnj.edu/~bush/uftp.html, prosinac 2010.
[15] Wikipedia: File eXchange Protocol, http://en.wikipedia.org/wiki/File_eXchange_Protocol, siječanj 2011.
[16] Wikipedia: FTPS, http://en.wikipedia.org/wiki/FTPS, prosinac 2010.
[17] Wikipedia: Secure copy, http://en.wikipedia.org/wiki/Secure_copy, siječanj 2011.
[18] Wikipedia: Samba, http://en.wikipedia.org/wiki/Samba_(software), veljača 2011.
[19] Wikipedia: FTP bounce attack, http://en.wikipedia.org/wiki/FTP_bounce_attack, rujan 2010.
[20] Toolbox.com: Man-In-The-Middle-Attack, http://it.toolbox.com/wiki/index.php/Man-in-the-Middle_Attack, prosinac 2008.
[21] Wikipedia: ProFTPd, http://en.wikipedia.org/wiki/ProFTPd, siječanj 2011.
[22] Wikipedia: FileZilla Server, http://en.wikipedia.org/wiki/FileZilla_Server, prosinac 2010.
[23] Wikipedia: FileZilla, http://en.wikipedia.org/wiki/FileZilla, veljača 2011.
[24] OpenSSH: Features, http://www.openssh.com/features.html, srpanj 2005.