

NNTP – Network News Transfer Protocol

NNTP specificira protokol za distribucijo, sprejemanje in pošiljanje novic preko varnega komunikacijskega kanala (TCP). NNTP je izdelan tako, da so članki (novice) shranjeni v centralni podatkovni bazi, katera naročenemu uporabniku omogoča izbiro in pregled novic katere si sam želi ogledati.

NNTP protokol deluje na TCP transportnem sistemu in uporablja vrata 119. Za komunikacijo med odjemalcem in strežnikom je uporabljen zelo podoben način kot pri prenosu elektronskih sporočil. Ukazi in odgovori so zgrajeni iz znakov iz ASCII tabele. Vsak ukaz je zaključen z kombinacijo znakov CR-LF (Carriage Return – Line Feed – skok na začetek vrstice, skok na naslednjo vrstico). Nekaj NNTP ukazov:

1xx – Informativno sporočilo

2xx – Ukaz uspešen

3xx – Ukaz do sedaj v redu, pošlji še preostanek

4xx – Ukaz je v redu, ampak iz neznanega razloga ga ni mogoče izvesti

5xx – Ukaz ni implementiran, ali napačen ali pa je prišlo do napake v programu

Iz zgornjih ukazov vidimo, da je prva številka odgovora rezervirana za sporočilo. Poglejmo si drugi del:

x0x – Povezavno, namestitveno ali drugo sporočilo

x1x – Izbira novičarske skupine

x2x – Izbira članka

x3x – Distribucijske funkcije

x4x – Pošiljanje

x5x – Nestandardna extenzija(dodatek)

x6x – Izpis za testiranje (debugging code)

Nekaj odjemalcev, ki podpira NNTP protokol:

- Microsoft Outlook Express
- Microsoft Outlook
- Netscape Navigator
- Mozilla
- Mozilla Thunderbird
- Google News
- Pine

Nekaj NNTP strežnikov:

- nntp
- leafnode
- Exchange

Telnet – terminal emulation

Program telnet je zagnan na lokalnem računalniku, kateri se poveže na oddaljeni računalnik in tam lahko izvajamo ukaze. Ukaze se vtipka na lokalni računalnik izvajajo pa se na oddaljenem računalniku. Izvajanje se dogaja v realnem času, tako da vsak ukaz, katerega vtipkamo dobimo takoj odgovor. Protokol telnet so izdelali leta 1969 kot nadomestilo za terminale. Na univerzah je tak način dostopa omogočal povezavo iz šibkih računalnikov na močnejše računalnike in tako izvajanje procesorsko zahtevnejših aplikacij. V takratnih lokalnih omrežjih varnost ni bila tako pomembna.

Da zgradimo telnet povezavo potrebujemo računalnik, do katerega imamo dostop. Poznati moramo uporabniško ime in geslo.

Povezava se zgradi na TCP protokolu in uporablja vrata 23. Več informacij o samem telnet protokolu se nahaja v RFC854 in RFC855.

Glavna slabost telnet protokola je varnost. Podatki se namreč po prenosnem kanalu ne prenašajo prekrito, ampak to kar vtipkamo se dejansko prenese po omrežju. Zlonamerni uporabniki omrežja na tak način zelo hitro izkoristijo to slabost protokola in se okoristijo z uporabniškimi imeni in gesli drugih uporabnikov.

Ravno zaradi nevarnosti se protokol zelo hitro opušča in zamenjuje ga varna različica protokola imenovana SSH (secure shell). SSH ponuja vse možnosti telneta z dodatki enkripcije.

Telnet odjemalci:

- dtelnet
- console telnet
- putty
- ...

Telnet stražniki:

- telnetd

IRC – Internet Relay Chat

IRC je primarno namenjen komuniciranju različnim skupinam (many-to-many) in tako imenovanih kanalih. Omogočena je tudi privatna komunikacija (one-to-one). IRC je bil izdelan leta 1988, da bi zamenjal program MUT (MultiUser Talk).

IRC je odprt protokol, ki uporablja TCP za prenos sporočil po omrežju in navadno vrata 6667. IRC strežniki so med seboj povezani in vsak odjemalec se poveže na en strežnik, preko katerega so mu vidni vsi uporabniki (ne glede na strežnik). Uporabniki uporabijo enega od IRC odjemalcev, da se povežejo na strežnik. Večina IRC strežnikov uporablja prijavo preden se povežemo, a ta prijava je navadno le nastavitev uporabniškega imena (včasih imenovano tudi nickname).

IRC uporablja čiste ASCII znake, kar pomeni, da lahko uporabimo IRC tudi z povezavo direktno na strežnik z aplikacijo TELNET – prenos znakov po omrežju. Protokol je opisan v dokumentu RFC1459.

Nekaj IRC omrežij:

- EFnet
- IRCnet
- Undernet
- ...

Nekaj irc odjemalcev:

- mIRC
- ChatZilla
- ircII
- BitchX
- XChat

Bots - roboti:

- Eggdrop
- EnergyMech

Nekaj irc strežnikov:

- ircd
- ircXpro
- ...

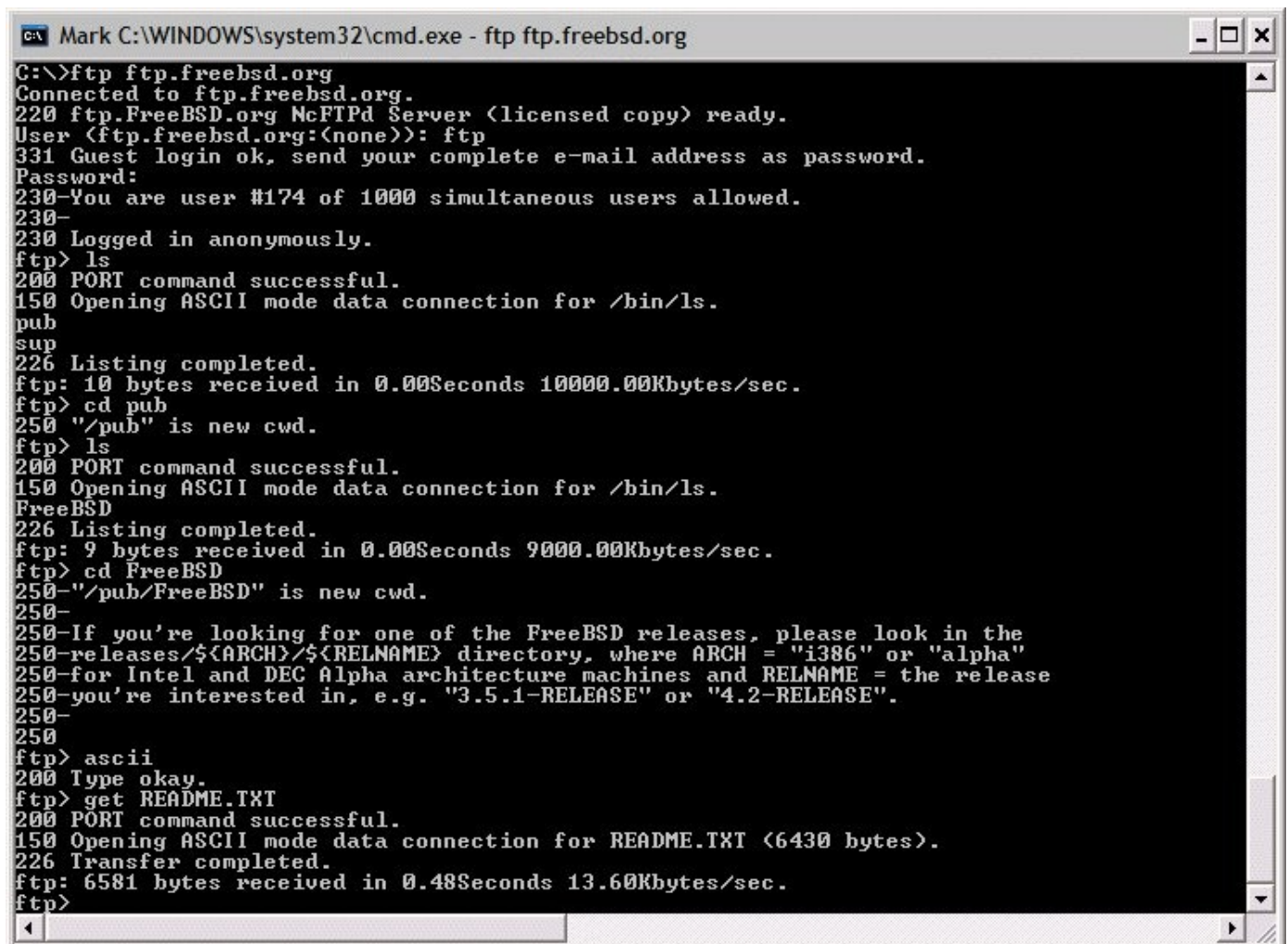
FTP – File Transfer Protocol

FTP ali File Transfer Protocol je protokol, katerega uporabljamo za izmenjavo datotek preko TCP/IP omrežja. V komunikacijo sta vpletena dva računalnika. Prvi je računalnik, ki nudi storitev FTP – imenovan tudi FTP strežnik. Ta je vedno pripravljen, da nudi svoje storitve drugim računalnikom – imenovani tudi FTP odjemalci. FTP odjemalci (aplikacije) ko so enkrat povezani na strežnik lahko opravljajo različne operacije za delo z datotekami (kopiranje, brisanje, premikanje, preimenovanje,...).

Vsakdo lahko izdelava aplikacijo FTP strežnik ali odjemalec, kajti protokol je odprtega tipa in specifikacije si lahko ogleda vsak. Vsak računalnik, ki podpira TCP/IP protokol lahko uporablja FTP storitev za manipulacijo z datotekami. FTP protokol ni vezan na operacijski sistem in tudi izmenjava datotek iz enega operacijskega sistema v drugega je mogoča.

FTP uporablja vrata 21 in deluje samo preko TCP transportnega sistema. FTP stražnik posluša na vratih 21 na povezave iz FTP odjemalcev. Hkrati lahko imamo povezanih več FTP odjemalcev hkrati na en FTP strežnik. Ko smo povezavo vzpostavili se za prenos datotek uporablja druga vrata (ne vrat 21). Katera vrata so uporabljena se odjemalec in strežnik sama sporazumeta.

Da ustvarimo dejansko povezavo na FTP strežnik potrebujemo aplikacijo imenovano FTP odjemalec. Aplikacije so lahko tekstovne, obstajajo pa tudi grafične. Ko zaženemo program, je preden se povežemo na strežnik potrebno najprej vnesti ime oddaljenega FTP strežnika. Prvi ukaz na strežniku je identifikacije z uporabniškim imenom in geslom. Žal se podatki v FTP protokolu pošiljajo neprekriti, tako da vsak ki prisluškuje vidi naše uporabniško ime in geslo. Če se pravilno prijavimo v sistem imamo na voljo ukaze za delo z datotekami.



```
C:\>ftp ftp.freebsd.org
Connected to ftp.freebsd.org.
220 ftp.FreeBSD.org McFTPd Server <licensed copy> ready.
User <ftp.freebsd.org:(none)>: ftp
331 Guest login ok, send your complete e-mail address as password.
Password:
230-You are user #174 of 1000 simultaneous users allowed.
230-
230 Logged in anonymously.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
pub
sup
226 Listing completed.
ftp: 10 bytes received in 0.00Seconds 10000.00Kbytes/sec.
ftp> cd pub
250 "/pub" is new cwd.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
FreeBSD
226 Listing completed.
ftp: 9 bytes received in 0.00Seconds 9000.00Kbytes/sec.
ftp> cd FreeBSD
250 "/pub/FreeBSD" is new cwd.
250-
250-If you're looking for one of the FreeBSD releases, please look in the
250-releases/${ARCH}/${RELNAME} directory, where ARCH = "i386" or "alpha"
250-for Intel and DEC Alpha architecture machines and RELNAME = the release
250-you're interested in, e.g. "3.5.1-RELEASE" or "4.2-RELEASE".
250-
250
ftp> ascii
200 Type okay.
ftp> get README.TXT
200 PORT command successful.
150 Opening ASCII mode data connection for README.TXT (6430 bytes).
226 Transfer completed.
ftp: 6581 bytes received in 0.48Seconds 13.60Kbytes/sec.
ftp>
```

Slabosti:

- gesla niso prekrita
- uporabljenih je več povezav. Ena za ukaze in druge povezave za prenos datotek med odjemalcem in strežnikom.
- ni vgrajenega algoritma, ki preverja pravilnost podatkov. Če se podatki med prenosom popačijo ne moremo ugotoviti, da so ti podatki popačeni.

Varnost:

Kot že omenjeno se podatki med prenosi ne šifrirajo, tako da na enostaven način lahko ugotovimo podatke o uporabniku, ki prenaša in izvaja raznorazne ukaze na strežniku. Trenutna rešitev za pošiljanje datotek z prekrivanjem je uporaba ssh aplikacij.

Anonimni dostop

FTP strežniki, ki ponujajo vsebine navadno imajo tudi tako imenovan anonimni dostop do strežnika. Z uporabo anonymous uporabniškega imena in elektronskega naslova kot geslo se uporabnik prijavi na tak strežnik in lahko brska po javnih datotekah. Navadno teh datotek ni mogoče brisati ali preimenovati, kot tudi ni mogoče datotek nalagati na sistem. Tak sistem je prvensteno namenjen prenosu datotek iz strežnika.

Načina prenosa datotek

Ko prenašamo datoteke imamo dva možna načina prenosa datoteke:

- ASCII način
- BINARY način

ASCII način uporabljamo, ko hočemo prenesti tekstovno datoteko. Ta način je uporabljen, ko prenašamo tekstovne datoteke (znak po znak) in sistem poskrbi za pravilni zapis datoteke na disku (na windowsih svoj zapis, na unix sistemih svoj zapis, na mac svoj zapis). Pri binarnem prenosu se prenašajo bit po bit in na drugi strani se biti shranjujejo nazaj v datoteko. Naprednejši odjemalci sami poskrbijo za pravilen prenos datotek.

FTP odjemalci:

- CuteFTP
- BulletProof
- FileZilla
- gFTP
- ncftp
- WS ftp
- ...

FTP strežniki:

- FileZilla Server
- War FTP Daemon
- Serv-U FTP Server
- BSD ftpd
- wu-ftp
- ProFTPD
- ...

Za razvoj aplikacij najdemo FTP specifikacije v RFC 959.

SSH – Secure Shell

Secure Shell ali na kratko SSH je računalniški program in hkrati protokol izdelan za prijavo in izvajanje programov na oddaljenem računalniku. Razvijalcem SSH protokola je bil cilj zamenjava protokolov, ki niso bili varni, kot so rlogin, TELNET in rsh. Torej prenos iz enega računalnika na drugi računalnik naj bi bil prekrit in dvem povezanim računalnikom omogočal varen prenos preko prenosnega kanala, ki ni varen. Uporabniki SSH lahko uporabljajo še za tuneliranje (tunneling), posredovanje (forwarding) in prenos datotek. SSH strežnik prevzeto posluša na vratih 22, TCP transportnega sistema.

SSH se večinoma uporablja :

- v kombinaciji z sftp, kot varno nadomestilo za prenos datotek (FTP)
- v kombinaciji z scp, kot varno nadomestilo za prenos datotek (RCP)
- za posredovanje ali tuneliranje. Na tak način ustvarimo varen kanal, preko katerega prenašamo neprekrute podatke (HTTP over SSH, X11 over SSH,...)
- z uporabo ssh odjemalca lahko na oddaljenem računalniku popravljamo nastavitve strežnika.
- za izvajanje raznoraznih skript preko varne povezave.

Arhitektura SSH-2 protokola je definirana v RFC 4251. Vsak od nivojev znotraj samega protokola pa ima svoj standard.

- transportni nivo (RFC 4253). Ta nivo definira začetno izmenjavo ključev in avtentikacijo na strežnik. Vzpostavi se enkripcija, kompresija in kontrola integritete (samega prenosa). Na tem nivoju se definira tudi ponovni prenos ključev po prenešanem 1GB podatkov ali ko je pošla 1 ura povezave – kar dosežemo prej (prevzeto je 1GB in 1 ura).
- Nivo uporabnikove prijave (RFC 4252). Ta nivo definira odjemalčevo avtentikacijo in ponuja več različnih načinov avtentikacije. Avtentikacija se izvaja na odjemalcu (torej ko se zahteva geslo, to geslo zahteva odjemalec in ne strežnik!). Nekaj načinov avtentikacije:
 - password – uporabnik se prijavi s pomočjo gesla
 - publickey – prijava s pomočjo ključev (navadno DSA ali RSA par ključev)
 - GSSAPI – mogoč je zunanji način prijave kot primer kerberos, ki nam omogoča način enkratne prijave za vse storitve (Single SIGN On).
- povezavni nivo (RFC 4254). Ta nivo definira princip delovanja kanalov, katere SSH ponuja.

SSH odjemalci:

- PuTTY
- MindTerm
- WinSCP
- OpenSSH

SSH strežniki:

- OpenSSH server
- GlobalSCAPE SSH Server

SNMP

Simple Network Management Protocol, na kratko SNMP. Protokol nam omogoča nadziranje naprav, ki so povezane na omrežje.

MIB – SNMP protokol je že po sami zasnovi zelo razširljiv. Ta razširljivost je dosežena preko množice nadzorne informacijske baze ali na kratko MIB (Management Information Base), ki nam specifikira nadzor podatkov specifičnega dela naprave, ki podpira SNMP. MIB uporablja hierarhični način, opisani z objektnimi identifikatorji (OID – Object IDentifier). Hierarhija samega MIB-a se začne z brezimenskim korenem in nadaljuje se z posameznimi imeni različnih organizacij. Tak model omogoča nadzor nad vsemi nivoji OSI referenčnega modela, razširljivo do aplikacij kot do podatkovne baze, email,...

Arhitektura – SNMP sistem je sestavljen iz treh komponent:

1. Master Agent
2. Subagents
3. Management Stations

Master Agent je aplikacija, ki se izvaja na SNMP napravi (primer router), ki se odziva na SNMP zahteve, katere povzroča management station. Torej lahko povzamemo, da Master Agent se obnaša kot server, management station pa kot odjemalec. Strežnik navadno zahteve, ki jih je prejel izvaja na svojih sub agentih (Subagents) in rezultate samo posreduje nazaj odjemalcu.

Subagent je programski del, ki izvaja povpraševanja prejeta od Master Agenta. Povpraševanja izvaja na podlagi, kateri MIB ta naprava podpira. En Master Agent lahko vsebuje več Subagentov, torej lahko podpira več MIBov.

Manager ali management station je aplikacija, ki ustreza imenu odjemalec. Na odjemalcu željene rezultate preglejujemo.

Prevzeta vrata, katera uporablja SNMP Master Agent za svoje delovanje so 161, na odjemalcu pa 162 - vse na UDP transportnem sistemu. Torej odjemalec se poveže na vrata 161 agenta, le ta pa posreduje rezultate odjemalcu na vratih 162.

Standard je definiran v več RFCjih:

- 1065 – struktura in identifikacija nadzora
- 1066 – MIB
- 1067 – SNMP

Verzija 1

Šibka lastnost te verzije je varnost. Avtentikacija odjemalcev je izvedena v obliki "community string", kot geslo. Geslo ni prekrito!

Verzija 2

Zaradi raznoraznih nestrinjanj z varnostjo se v2 ne uporablja množično. Definiran v RFC 1441, 1452 – znano kot SNMPv2 ali v2p, vključuje popravke nad v1 in prinaša določene izboljšave na področju performans, varnosti in komunikaciji manager to manager. Community based SNMPv2 ali SNMPv2c definiran v RFC 1901 in RFC 1908, z razliko prvotne verzije SNMPv2 ne vsebuje izboljšav v varnosti, ampak se sklicuje na avtentikacijo, kot jo ima SNMPv1 (community string), a vsebuje izboljšave v performansah v primerjavi z SNMPv1. User based SNMP ali SNMPv2u je definiran v RFC 1909 in RFC 1910. Ta verzija je vključuje varnost SNMPv2, ampak ne vsebuje kompleksnosti značilne za SNMPv2. Zmeda v SNMPv2 različicah vodijo v razvoj nove verzije 3.

Verzija 3

SNMP se je reorganiziral z verzijo 3 (definira RFC 3411 in RFC 3418) in je postavil standard za SNMP. IETF (Internet Engineering Task Force) je postavil vse ostale verzije (v1, v2*) za pretekle. V praksi večina odjemalcev in master agentov podpira več SNMP različic.

SNMP Management Station

- MG-SOFT MIB Browser
- AdventNet MIB Browser
- SNMP Mib Browser
- NET-SNMP

SNMP Master Agent

- MS Master Agent
- MG-SOFT Master Agent
- NET-SNMP