Universitetet i Sørøst-Norge

Av Fredrik Villo og Per Nyberg

TSD2090V-1

Lab 3 (TCP/UDP/ICMP)

Innholdsfortegnelse

Sammendrag

I denne labben (Lab 3) fikk gruppe 4 i oppgave å utforske og forstå funksjonaliteten til TCP, UDP og ICMP protokollene i transport og nettverkslaget. Arbeidet inkluderte bruk av verktøyet TCPdump for å fange og analysere pakker sendt mellom to maskiner på et lokalt nettverk i et kontrollert miljø på USN Vestfolds datalab. Gruppen opplevde oppgaven som lærerik og mer forståelig enn tidligere labber, og følte at de fikk et godt læringsutbytte av gjennomføringen.

1 Innledning

Transmission Control protocol (TCP) og User Datagram Protocol (UDP) er to av de viktigste protokollene som brukes til å overføre data mellom enheter på et nettverk. TCP sørger for pålitelig og strukturert dataoverføring med feil behandling, mens UDP tilbyr en raskere, men mindre pålitelig tilkobling der det ikke er noen garanti for at pakker når frem i riktig rekkefølge eller i det hele tatt. ICMP er en protokoll som brukes av nettverkslaget til å rapportere feil, i denne labben brukes ICMP for å utføre nettverk diagnostikk og for å gi eventuelle feilmeldinger hvis noen av pakkene ikke når frem mellom de ulike virtuelle maskinene**[1], [2]**

Hensikten med Lab 3 er at gruppe 4 skal lære hvordan TCP og UDP fungerer, skille mellom de to, og bruke TCPdump til å analysere hvordan pakkene fra de ulike protokollene beveger seg mellom to maskiner.

2 Problemstilling

Hvordan fungerer TCP og UDP i nettverkskommunikasjon, og hvordan kan TCPdump brukes til å analysere pakkene fra disse protokollene når data overføres mellom to maskiner? Hva er de viktigste forskjellene i måten TCP og UDP overfører data, og hvordan kan vi visuelt observere forskjellen ved hjelp av TCPdump?

3 Teoridel

3.1

TCP og UDP

TCP og UDP er 2 kommunikasjonsprotokoller som brukes for å sende forespørsler. De skiller seg signifikant på måten de opererer på.

UDP sender pakker, såkalte datagrammer, fra server til klient - men sjekker ikke om de kommer frem eller ikke. Det betyr at pakker kan komme i feil rekkefølge eller at man får pakketap (packet loss) og at pakken faktisk ikke kommer frem og er borte for alltid.

TCP starter med en så kalt «*handshake*» som består av 3 deler.

1. Klient sender ett Syn (synchronize) flagg til server «*Kan jeg få koble til*?».
2. Server svarer med Syn+ACK (acknowlegde) som betyr at «*Ja, du får koble til*».
3. Klienten svarer så med ACK som betyr «*Ja takk. Jeg kobler gjerne til deg.*»

Neste del sendes det en f.eks. en GET forespørsel om en fil. Serveren svarer på ny med ACK og sender pakker. Når klienten har fått det den ønsket, avsluttes det hele med et ACK om at alle filer har blitt tatt imot og serveren sender en closing som klient svarer ACK på og koblingen avsluttes.[1]
På denne måten er du mer sikker på at du får alle pakker du ønsket.[1]

3.2

ICMP

Datamaskin.biz skriver:

> «*ICMP står for Internet Control Message Protocol , et lett sett av applikasjoner opprinnelig designet for å oppdage og rapportere feil forholdene på datanettverk . ICMP er en forlengelse av Internet Protocol (IP) og er definert av en intensjonsavtale , kjent som Request for Comments (RFC) 792 , utgitt av Internet Engineering Task Force . Formål*
>
> *ICMP tillater verter, rutere og andre nettverksenheter å utveksle grunnleggende kontroll, for eksempel feil og statusinformasjon, når data sendes fra én enhet til en annen. Så ICMP er nyttig for sondering et nettverk for å bestemme dens generelle egenskaper eller for feilsøking problemer med nettverkstilkobling. Hvis en nettverksadministrator forstår ICMP og mulige årsaker til bestemte typer ICMP-meldinger , er han eller hun bedre rustet til å diagnostisere nettverksproblemer.*»[3], [4]

3.3

Hva gjør TCPDump?

TCPDump lytter til trafikken som flyr over nettverket. Den ser pakker som sendes og tas imot og fra hvilke IP-kommunikasjonen går mellom.
Hvis man ønsker å søke etter spesiell trafikk, f.eks. type trafikk, port eller IP kan man filtrere med hjelp av opsjonene som er innebygget i programmet. F.eks.: “tcpdump tcp port 80” som da lytter etter TCP på port 80 (HTTP).
Når det kommer til porter burde man merke seg at port 0 - 1024 er standardporter som ikke bør brukes når du setter opp ny kommunikasjon. Det er også lurt å unngå å bruke port 1024 til 49151 ettersom disse portene er registrerte porter hos IANA (Internet Assigned Numbers Authority) som brukes av spesifikke programvarer eller protokoll. 49152 til 65535 er lurest å bruke da disse er “private” porter som ikke har noen forbehold fra IANA.[5]

3.4

Metode og fremgangsmåte

I denne labben brukes det flere metoder og fremgangsmåter for å forstå og analysere hvordan TCP, UDP og ICMP fungerer. Labben legger vekt på å gi gruppen verktøyene de trenger for praktisk oppsett og testing ved hjelp av følgende tilnærminger:

1. **Oppsett av Nettverk**
   - **Virtuelle maskiner:** Gruppen ble bedt om å sette opp to ulike maskiner til denne øvelsen, Maskin A og Maskin B. Begge skulle bli konfigurert med 2 nettverksadaptere, der eth0 skulle settes til «bridged adapter», men eth1 skulle settes til «host only adapter».

   - **IP-konfigurasjon:** Maskin A og maskin B blir konfigurert med de statiske IP-adressene 1.4.0.1/24 og 1.4.0.2/24 henholdsvis, for at sikre at maskinene kan kommuniserer med hverandre.

2. **Bruk av verktøy for nettverksanalyse**
   - **TCPdump:** Dette verktøyet brukes til å fange og analysere nettverkspakker i sanntid, gruppen brukte kommandoen «tcpdump -n -i eth1 net 10.4.0.0/24 and not arp» til å skille mellom de ulike pakkene og unngå støy fra ARP-protokollen.[6]

3. **Testing av protokoller**
   - **ICMP:** Gruppen testet ICMP protokollen ved å bruke *ping* kommandoen. Dette sender echo forespørsler mellom de to maskinene for å verifisere at de kan kommunisere. TCPdump fanger opp ICMP-pakkene som sendes og mottas.
     Dataen fra dette ser slik ut:

     06:52:22.741985 IP 10.4.0.2 > 10.4.0.1: ICMP echo request, id 3, seq 1, length 64
     06:52:22.742006 IP 10.4.0.1 > 10.4.0.2: ICMP echo reply, id 3, seq 1, length 64
     07:02:23.048503 IP 10.4.0.2.45855 > 10.4.0.1.search-agent: UDP, length 9
     07:02:32.272665 IP 10.4.0.2.45855 > 10.4.0.1.search-agent: UDP, length 12
     07:05:51.056074 IP 10.4.0.2.48634 > 10.4.0.1.tram: * wb-32!
     07:05:51.056098 IP 10.4.0.1 > 10.4.0.2: ICMP 10.4.0.1 udp port tram unreachable, length 55

   - **UDP:** Gruppen bruker netcat (nc) på maskin A for å lytte på port *1234* ved hjelp av kommandoen *nc -n -v -u -l 1234*. *-u* flagget som blir brukt her står for UDP, og gjør at pakkene sendes over UDP istedenfor TCP som er standard. Gruppen sender deretter pakker fra maskin B til maskin A ved å skrive tekst inn i kommando linjen. Dataene vi får ut av dette ser slik ut:

     04:37:55.662487 IP 10.4.0.2.45900 > 10.4.0.1.dbm: UDP, length 7
     04:37:56.830417 IP 10.4.0.2.45900 > 10.4.0.1.dbm: UDP, length 6
     04:38:01.838302 IP 10.4.0.2.45900 > 10.4.0.1.dbm: UDP, length 16

     Ved å analysere disse dataene så kan gruppen se at det ikke er noe bekreftelse som går imellom maskinene og at maskin A heller ikke viser noe tilbakemelding, og TCPdump logger kun når pakkene blir sendt (når tekst blir skrevet).

- **TCP:** For å teste og analysere TCP tilkoblingen så gjentar gruppen de samme stegene, men uten *-u* flagget slik at det overføres via TCP og ikke UDP. Dataene gruppen fikk ut av dette var dette:

  Første segment:(Handshake)
  04:40:10.277027 IP 10.4.0.2.39848 > 10.4.0.1.dbm: **Flags [S],** seq 3803048943, win 29200, options [mss 1460,sackOK,TS val 3719038258 ecr 0,nop,wscale 7], length 0
  04:40:10.277050 IP 10.4.0.1.dbm > 10.4.0.2.39848: **Flags [S.],** seq 903611455, **ack 3803048944**, win 28960, options [mss 1460,sackOK,TS val 2708990543 ecr 3719038258,nop,wscale 7], length 0
  04:40:10.277384 IP 10.4.0.2.39848 > 10.4.0.1.dbm: Flags [.], **ack 1**, win 229, options [nop,nop,TS val 3719038259 ecr 2708990543], length 0

  Andre segment:
  04:41:44.861237 IP 10.4.0.2.39848 > 10.4.0.1.dbm: Flags [P.], seq 4:10, ack 1, win 229, options [nop,nop,TS val 3719132842 ecr 2709078975], length 6
  04:41:44.861269 IP 10.4.0.1.dbm > 10.4.0.2.39848: Flags [.], ack 10, win 227, options [nop,nop,TS val 2709085127 ecr 3719132842], length 0
  04:41:46.957587 IP 10.4.0.2.39848 > 10.4.0.1.dbm: Flags [P.], seq 10:15, ack 1, win 229, options [nop,nop,TS val 3719134939 ecr 2709085127], length 5
  04:41:46.957604 IP 10.4.0.1.dbm > 10.4.0.2.39848: Flags [.], ack 15, win 227, options [nop,nop,TS val 2709087223 ecr 3719134939], length 0
  04:41:52.029863 IP 10.4.0.2.39848 > 10.4.0.1.dbm: Flags [P.], seq 15:31, ack 1, win 229, options [nop,nop,TS val 3719140011 ecr 2709087223], length 16
  04:41:52.029880 IP 10.4.0.1.dbm > 10.4.0.2.39848: Flags [.], ack 31, win 227, options [nop,nop,TS val 2709092296 ecr 3719140011], length 0

  Tredje segment:
  04:42:23.950279 IP 10.4.0.2.57902 > 10.4.0.1.dbm: Flags [F.], seq 1, ack 1, win 229, options [nop,nop,TS val 3719171932 ecr 2709120173], length 0
  04:42:23.950404 IP 10.4.0.1.dbm > 10.4.0.2.57902: Flags [F.], seq 1, ack 2, win 227, options [nop,nop,TS val 2709124216 ecr 3719171932], length 0
  04:42:23.950586 IP 10.4.0.2.57902 > 10.4.0.1.dbm: Flags [.], ack 2, win 229, options [nop,nop,TS val 3719171932 ecr 2709124216], length 0

  Gruppen observerte at det ble loggført mer data i TCPdump ved overføring over TCP. De la merke til at TCPdump ikke bare registrerte data når noe ble skrevet (andre segment), men også når tilkoblingen ble opprettet (første segment) og når tilkoblingen ble lukket (tredje segment).

4. Feilsøking og validering

Etter hvert avsnitt av labben ble gruppen oppmerksom på at dersom netcat eller ping-kommandoene ikke fungerte, skulle de gjenta avsnittet for å kontrollere at IP-adressene og nettverksinnstillingene var korrekte. Basert på dette konkluderte gruppen med at netcat og ping kan brukes til å feilsøke og validere om oppsettet er korrekt.

4 Konklusjon

I denne labben fikk gruppe 4 en praktisk forståelse av TCP, UDP og ICMP protokollene. De fikk erfaring med å bruke TCPdump til å analysere nettverkstrafikk, og gjennom testing av disse protokollene skjønte gruppen mer om hvordan de fungerer på de ulike nivåene i nettverkstakken og hvordan de håndterer data.

UDP: Ut ifra oppgavene og det gruppen har lært så er resultatene som forventet. UDP viser at den ikke sender noen form for bekreftelse verken til eller fra når TCPdump leser loggen.

TCP: Her ser gruppen en klar forskjell på dataene som blir loggført, ved å analysere loggen så lærte gruppen hvordan «handshaket» ser ut.

TCPdump: Effektivt nettverksanalyseverktøy som fungerte veldig bra i labben med kun to maskiner i nettverket og dermed lite trafikk. Skal man analysere trafikk over større nettverk må man lære seg hvordan man filtrerer trafikken som den plukker opp – ellers vil det bli for mange nye rader for hvert sekund som går og da mister man oversikten.

5 Kilder og referanser

[1] «Differences between TCP and UDP», GeeksforGeeks. Åpnet: 17. september 2024. [Online]. Tilgjengelig på: https://www.geeksforgeeks.org/differences-between-tcp-and-udp/

[2] «What is ICMP? | Internet Control Message Protocol». Åpnet: 17. september 2024. [Online]. Tilgjengelig på: https://www.cloudflare.com/learning/ddos/glossary/internet-control-message-protocol-icmp/

[3] «Internet Control Message Protocol», Internet Engineering Task Force, Request for Comments RFC 792, sep. 1981. doi: 10.17487/RFC0792.

[4] «Hva er ICMP». Åpnet: 17. september 2024. [Online]. Tilgjengelig på: https://www.datamaskin.biz/Nettverk/internet-networking/67754.html

[5] «tcpdump(1) man page | TCPDUMP & LIBPCAP». Åpnet: 17. september 2024. [Online]. Tilgjengelig på: https://www.tcpdump.org/manpages/tcpdump.1.html

[6] «An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware», Internet Engineering Task Force, Request for Comments RFC 826, nov. 1982. doi: 10.17487/RFC0826.