2026-09-15

HÖGSKOLAN VÄST

Jämförelse av transportprotokoll för syslog-baserad logghantering under nätverksbelastning

Kadija Abdi **Mohamed Mahamed**

EXAMENSARBETE

Nätverksteknik med IT-säkerhetsprogrammet, 120 hp
Institutionen för ingenjörsvetenskap, Högskolan Väst

EXAMENSARBETE

Jämförelse av transportprotokoll för syslog-baserad logghantering under nätverksbelastning

Sammanfattning

Logghantering är en stor del av dagens IT-säkerhet då loggar kan användas till att upptäcka incidenter och analysera systemhändelser. Syftet med detta arbete är att undersöka hur valet av transportprotokoll TCP respektive UDP påverkar tillförlitligheten i logginsamlingen genom syslog under nätverksbelastning.

För att undersöka detta genomfördes olika tester i en labbmiljö. En experimentell testmiljö byggdes upp bestående av en router, två klientdatorer och SIEM plattformen Graylog. I denna miljö utfördes kontrollerade logghändelser samtidigt som nätverksbelastningen ökade för att simulera realistiska förhållanden.

Varje testscenario upprepades sju gånger (n=7) för att säkerställa statistisk tillförlitlighet. Resultaten visar att syslog via TCP uppvisade en genomsnittlig loggförlust på 1,18 % (medelvärde 4 941 av 5 000 loggar mottagna; värsta uppmätta fall 1,32 %) medan UDP uppvisade en genomsnittlig loggförlust på 44,14 % (medelvärde 2 793 av 5 000 loggar mottagna; värsta fall 44,84 %). CPU användning och minnesanvändning förblev stabila i samtliga scenarier, vilket tyder på att loggförlusten vid UDP i huvudsak beror på protokollets inneboende begränsningar. Andra faktorer, till exempel Graylogs mottagningsbuffer eller bakgrundstrafik i labbmiljön, har inte mätts separat och kan därför inte uteslutas som bidragande orsaker.

Studien bidrar med ett empiriskt underlag för protokollval i säkerhetskritiska IT miljöer, avgränsat till den testade miljön.

Datum:	2026-03-20
Författare:	Kadija Abdi, Mohamed Mahamed
Examinator:	Hani Alid
Handledare:	Johan Larsson
Program:	Nätverksteknik med IT-säkerhet, 120 hp
Huvudområde:	Datateknik
Utbildningsnivå:	Grundnivå
Kurskod:	EXn300, 7,5 hp
Nyckelord:	*Logghantering, syslog, SIEM, nätverksbelastning, nätverkssäkerhet*
Utgivare:	Högskolan Väst, Institutionen för ingenjörsvetenskap 461 86 Trollhättan Tel: 0520-22 30 00, www.hv.se

DEGREE PROJECT

Comparison of Transport Protocols for Syslog-Based Log Management Under Network Load

Summary

Log management is an important part of modern IT security since log data can be used to detect incidents and analyse system events. The purpose of this study is to empirically investigate how the choice of transport protocol (TCP versus UDP) affects the reliability of log collection via syslog under network load.

To investigate this, a series of experiments were conducted in a laboratory environment. An experimental test environment was built consisting of a router, two client computers and the SIEM platform Graylog. In this environment, controlled log events were generated while network load was gradually increased in order to simulate realistic conditions.

Each test scenario was repeated seven times (n=7) to ensure statistical reliability. The results show that syslog via TCP exhibited an average log loss of 1.18% (mean 4,941 of 5,000 logs received; worst case 1.32%), while syslog via UDP exhibited an average log loss of 44.14% (mean 2,793 of 5,000 logs received; worst case 44.84%) under network load. CPU and memory utilisation remained stable across all scenarios, indicating that log loss under UDP is primarily attributable to the protocol's inherent limitations, although other unmeasured factors (e.g. the Graylog receive buffer or background traffic) cannot be ruled out.

The study contributes an empirical basis for transport protocol selection in security critical IT environments, within the scope of the tested setup.

Date:	March 20, 2026
Author:	Kadija Abdi, Mohamed Mahamed
Examiner:	Hani Alid
Advisor:	Johan Larsson
Programme:	Network Technology and IT-security, 120 HE credits
Main field of study:	Computer Engineering
Education level:	First cycle
Course code:	EXN300, 7.5 HE credits
Keywords:	*Log management, syslog, SIEM, Network Load, Network security*
Publisher:	University West, Department of Engineering Science SE-461 86 Trollhättan, Sweden Phone: +46 520 22 30 00, www.hv.se

Förord

Denna rapport har genomförts som en del av examensarbetet inom nätverksteknik och IT-säkerhet vid Högskolan Väst. Arbetet har utförts av Mohamed Mahamed och Kadija Abdi-kadir Abdi. Arbetet har delats upp så att Kadija har ansvarat för installation och konfiguration av den centrala loggservern samt analys av loggdata. Mohamed har ansvarat för nätverksdesign, routing och konfiguration av nätverksenheter. Planering, genomförande av tester, analys och rapportskrivning har vi gjort tillsammans. Vi vill rikta ett tack till vår handledare Johan Larsson för stöd och vägledning under arbetets gång.

Innehållsförteckning

Sammanfattning ..... ***i***

Summary ..... ***ii***

Förord ..... ***iii***

Nomenklatur ..... ***vi***

1 Inledning ***1***

1.1 Problemformulering 2

1.2 Syfte 2

1.3 Frågeställning 2

2 Bakgrund ***3***

2.1 Logghantering i IT-system 3

2.2 Central logginsamling 3

2.3 Syslog-protokollet 4

2.4 SIEM-system 4

2.5 Graylog 5

2.6 Transportprotokoll för loggöverföring 5

2.7 Relaterad forskning 6

3 Metod ***6***

3.1 Forskningsdesign 6

3.2 Testmiljö 7

3.3 Loggkonfiguration 7

3.4 Testprocedur och logggenerering 8

3.5 Datainsamling och mätvärden 9

4 Resultat ***10***

4.1 Genererade vs. mottagna loggar 10

4.2 Loggförlust per metod (%) 11

4.3 CPU användning (obelastat vs. belastat) 12

4.4 Minnesanvändning (obelastat vs. belastat) 13

5 Analys/diskussion ***13***

5.1 Lokal loggning 13

5.2 Central loggning via TCP 14

5.3 Central loggning via UDP 14

5.4 CPU och minnesanvändning 15
5.5 Skillnader mellan metoderna 15
5.6 Begränsningar och framtida arbete 16
6 Slutsats 17
Referenser 18

Nomenklatur

- Syslog – Ett protokoll för att skicka loggmeddelanden mellan enheter i ett nätverk.
- SIEM – Security Information and Event Management, ett system för att samla in och analysera loggdata.
- Graylog – En plattform för central logghantering och analys av loggar.
- TCP – Transmission Control Protocol, ett tillförlitligt transportprotokoll.
- UDP – User Datagram Protocol, ett snabbare men mindre tillförlitligt transportprotokoll.
- Logghantering – Processen att samla in, lagra och analysera loggdata.
- Loggförlust – När loggmeddelanden inte når fram till mottagaren.
- Nätverksbelastning – Mängden trafik som skickas i nätverket.

1 Inledning

Loggar har en avgörande roll inom modern IT-säkerhet och nätverksdrift, och genom dem kan säkerhetsanalytiker och säkerhetsadministratörer övervaka och identifiera systemhändelser och avvikelser samt analysera säkerhetsincidenter. Olika system såsom routrar, servrar, brandväggar och applikationer genererar loggdata ständigt. Informationen i dessa loggar inkluderar systemfel, trafikflöden och autentiseringsförsök, vilket gör det till en betydelsefull källa vid felsökning och hantering av incidenter [1].

Loggar har traditionellt lagrats på den lokala enheten, men denna metod har sina begränsningar eftersom lokal lagring vanligtvis är begränsad och det finns risk att loggar skrivs över när bufferten når sin kapacitet. Dessutom kan lokal lagring vara komplicerad när analys ska göras på flera enheter samtidigt. För att lösa dessa problem är det vanligt att använda en central logginsamling där loggar förs över till en central server med hjälp av syslog-protokollet. Syslog är ett protokoll som används för att skicka loggmeddelanden från olika nätverksenheter till en central loggserver, där informationen lagras och analyseras på ett mer effektivt sätt [2].

I avancerade säkerhetsmiljöer används även SIEM, en förkortning av Security Information and Event Management. SIEM är utvecklat för att samla in loggar från flera olika källor och standardiserar samt korrelerar loggdata, vilket gör att loggdata kan analyseras i realtid. Genom att analysera loggar från flera olika system kan en säkerhetsanalytiker samtidigt upptäcka mönster och identifiera attacker som annars vore svåra att upptäcka. Därför är SIEM plattformar vanliga inom säkerhetsövervakning och incidenthantering i organisationer [3].

Trots de många fördelarna med central logginsamling finns det även utmaningar gällande prestanda och tillförlitlighet. När stora mängder loggdata skickas över nätverket kan det leda till nätverksbelastning som påverkar överföringen av loggmeddelanden, vilket i sin tur kan leda till att meddelanden försenas eller går förlorade innan de når den centrala loggservern. I det scenariot kan förmågan att upptäcka och analysera säkerhetsincidenter påverkas. Valet av transportprotokoll har också stor betydelse för om loggöverföringen når fram eller inte. Syslog använder till exempel oftast UDP (User Datagram Protocol), som saknar mekanismer för att bekräfta att loggar kommer fram säkert. TCP är däremot mer pålitligt och säkerställer att överföringen når sin destination [2], [4].

NIS2 direktivet ställer krav på tillförlitlig loggning, incidentrapportering och spårbarhet inom kritisk infrastruktur [5], vilket gör frågan om protokollval direkt relevant för organisationer som måste uppfylla dessa krav.

Det som denna forskning bidrar med, utöver vad som redan framgår av teorin, är en konkret empirisk indikation på omfattningen av skillnaden i en specifik labbmiljö. Att TCP erbjuder högre tillförlitlighet än UDP har sedan länge varit etablerat, men att förlusten för

UDP i denna studie når i genomsnitt 44,14 % under de testade nätverksförhållandena, medan TCP ligger på 1,18 %, ger en mätbar grund som kan vara till hjälp vid val av protokoll i miljöer som omfattas av NIS2 direktivets krav [5]. Resultaten gäller för den testade miljön och bör verifieras innan de tillämpas i andra sammanhang.

Den aktuella studien analyserar hur valet av transportprotokoll inverkar på loggförlust under hög nätverksbelastning. Lokal loggning används som en referenspunkt eftersom den inte påverkas av nätverksbelastningen och därför inte kan jämföras direkt med TCP och UDP [1], [6].

1.1 Problemformulering

Även om central loggning och SIEM-system används i stor utsträckning inom moderna IT-miljöer finns det ändå utmaningar relaterade till prestanda och pålitlighet vid överföring av loggdata. När stora mängder loggmeddelanden förs över nätverket kan det leda till trafikbelastning som påverkar överföringen, vilket resulterar i att vissa loggmeddelanden försenas eller går förlorade. Om loggdata inte når fram till loggservern kan viktig eller känslig information om säkerhetshändelser saknas, vilket påverkar organisationens möjlighet att upptäcka, analysera och hantera säkerhetsincidenter. Det är därför väsentligt att undersöka hur olika metoder för logghantering påverkas när nätverket är belastat.

1.2 Syfte

Syftet med detta examensarbete är att empiriskt undersöka hur valet av transportprotokoll TCP respektive UDP påverkar tillförlitligheten i logginsamling via syslog under nätverksbelastning, samt att kvantifiera vad skillnaden innebär ur ett säkerhetsperspektiv [7], [8]. Lokal loggning inkluderas enbart som referenspunkt för att illustrera skillnaden mot nätverksbaserad loggning; den är inte direkt jämförbar med TCP och UDP eftersom den inte påverkas av nätverksbelastning.

Studien mäter tre specifika mätvärden:

- Loggförlust: antal genererade loggar i förhållande till antal mottagna loggar.
- CPU användning på routern i obelastat och belastat tillstånd.
- Minnesanvändning på routern i obelastat och belastat tillstånd.

1.3 Frågeställning

- I vilken utsträckning påverkar valet av transportprotokoll (TCP respektive UDP) tillförlitligheten i syslog-baserad logginsamling under simulerad nätverksbelastning i en kontrollerad labbmiljö, och vad innebär den uppmätta skillnaden konkret ur ett säkerhetsperspektiv?

- Under vilka förhållanden kan loggförlust vid UDP-baserad syslog äventyra ett SIEM-systems förmåga att detektera säkerhetsincidenter, och vilka faktorer utöver protokollet i sig kan bidra till förlusten?
- Hur påverkar nätverksbelastning routerns CPU och minnesanvändning i den testade miljön, och vad säger detta om var flaskhalsen i loggöverföringen faktiskt finns?

2 Bakgrund

2.1 Logghantering i IT-system

Logghantering är en essentiell del av moderna IT-system och spelar en betydande roll inom systemadministration och IT-säkerhet. Med hjälp av loggar registreras händelser om allt som sker i en enhet eller nätverksmiljö. Dessa händelser kan omfatta systemfel, inloggningsförsök och aktiviteter kopplade till säkerhet. Analysen av loggar ger säkerhetsadministratören en större inblick i nätverket, hur systemet fungerar samt möjlighet att identifiera potentiella problem och säkerhetshot [1].

I moderna ITmiljöer skapas loggar kontinuerligt av olika system och nätverksenheter som switchar och routrar. Nätverksenheterna dokumenterar data som rör nätverkstrafik och anslutningar. Servrar och operativsystem dokumenterar systemprocesser och användarinteraktion. Brandväggar och andra säkerhetssystem genererar loggar kopplade till säkerhetspolicyer och trafikfiltrering. Den omfattande loggdata som genereras i moderna IT-system möjliggör analys av hur systemet fungerar samt identifiering av avvikelser från normal aktivitet [1].

Loggdata används inte enbart för säkerhetsändamål utan även för att övervaka systemets prestanda och stabilitet. Genom att undersöka och analysera loggarna kan en administratör identifiera konfigurationsfel, flaskhalsar eller andra tekniska utmaningar inom infrastrukturen. Loggdata är därmed en essentiell resurs för både drift och säkerhetsarbete inom organisationer [1].

2.2 Central logginsamling

Loggar har traditionellt lagrats på den lokala enheten där loggen genereras. Denna metod har sina begränsningar även om den är den enklaste formen av lagring. Lokal loggning innebär att loggarna enbart finns tillgängliga på den enhet där de genererades, vilket försvårar analys av incidenter som involverar flera system. Lagringsutrymmet på nätverksenheter är dessutom begränsat, vilket resulterar i att äldre loggar skrivs över när kapaciteten uppnås.

För att lösa dessa problem används centraliserad logginsamling. Vid central loggning skickas loggmeddelanden från flera enheter till en central loggserver där meddelanden analyseras och lagras. Den centrala loggservern fungerar som ett nav för loggdata och möjliggör analys av information från många system på ett och samma ställe, vilket förenklar både felsökning och incidentanalys.

En ytterligare fördel med central logginsamling är att loggar kan bevaras även om en enhet kompromitteras eller slutar fungera. Om loggarna enbart lagras lokalt finns risk att de går förlorade om en angripare raderar loggfiler eller om systemet kraschar. Genom att lagra loggar på en separat server kan en organisation säkerställa att väsentlig data bevaras och kan analyseras i efterhand [1]. Centraliserad loggarkitektur är en erkänd grundprincip inom nätverkssäkerhet, och central logghantering framhålls som nödvändig för effektiv incident-respons [9].

2.3 Syslog-protokollet

Syslog är ett av de mest använda protokollen för logginsamling i nätverksmiljöer. Det är ett standardiserat protokoll som används för att skicka loggmeddelanden mellan olika system och en central loggserver. Syslog stödjer både UDP och TCP som transportprotokoll [2].

Protokollet fungerar genom att en enhet, ofta kallad syslog klient, genererar loggmeddelanden som sedan skickas över nätverket till en syslog server. Den centrala loggservern tar emot loggarna och lagrar dem i en loggfil eller databas där de kan analyseras vid ett senare tillfälle. Syslog meddelanden innehåller information om händelsens allvarlighetsgrad, vilken enhet som genererade loggen samt en beskrivning av händelsen.

En av de viktigaste egenskaperna hos syslog är flexibiliteten — protokollet gör det möjligt att samla loggar från ett stort antal enheter och system av olika typer och tillverkare. Syslog kan även använda TCP som transportprotokoll [10]. Det bör noteras att detta transportsätt numera har status Historic inom IETF och att TLS-krypterad syslog [11] är den rekommenderade transporten för TCP-baserad syslog i moderna miljöer; den testade miljön i denna studie använde dock oformaterad TCP, eftersom TLS-kryptering inte var en del av studiens avgränsning. Genom att använda TCP erbjuder syslog högre tillförlitlighet än UDP, eftersom TCP använder inbyggda mekanismer för kvittering och åter överföring vid paketförlust [10]. Säkerhetsfrågor relaterade till syslog transmission har fått ökad uppmärksamhet i den senaste forskningen, där det otillräckliga skyddet av integritet i UDP-baserad loggöverföring identifieras som en betydande svaghet [8].

2.4 SIEM-system

I stora IT miljöer kan mängden loggdata bli omfattande, och för att hantera informationen används SIEM (Security Information and Event Management). SIEM utvecklades för att samla in, lagra och analysera loggmeddelanden från olika källor i en nätverksmiljö. Genom att samla data på ett och samma ställe kan SIEM ge en samlad bild av det som sker i nätverket [4].

En central funktion i SIEM-system är korrelation av loggdata, vilket innebär att systemet analyserar många händelser simultant för att identifiera mönster och samband mellan olika

nätverksaktiviteter. Exempelvis kan ett flertal misslyckade inloggningsförsök från olika system indikera ett pågående intrångsförsök — något som kan upptäckas snabbare med SIEM än om loggarna lagrats lokalt [4].

Kvaliteten och fullständigheten av loggdata är avgörande för SIEM-systemets kapacitet att upptäcka anomalier. Avvikelsedetekteringssystem är starkt beroende av att loggdata är både kompletta och konsekventa — bristande loggposter skapar blinda fläckar som kan undergräva effektiviteten i hotdetektionen [12]. Analys av nätverksloggar i realtid spelar en central roll inom dagens säkerhetsövervakning, och i moderna SIEM plattformar kombineras nätverksflödesdata ofta med loggdata för att erbjuda en mer heltäckande insikt [13].

SIEM används främst inom säkerhetsövervakning och incidenthantering, och kompletta loggdata är en förutsättning för att kunna följa etablerade metoder för incidenthantering, såsom identifiering, analys och åtgärd av säkerhetsincidenter [14]. Genom att centralisera analysen och automatisera larm kan säkerhetsanalytiker snabbt identifiera potentiella hot och vidta nödvändiga åtgärder [4]. Det operativa värdet av ett SIEM-system är direkt beroende av hur fullständig den insamlade loggdatan är, vilket innebär att loggförlust utgör ett allvarligt säkerhetsproblem [15].

2.5 Graylog

Graylog är en SIEM plattform som används för central logghantering och analys av loggdata. Plattformen kan ta emot loggdata via flera protokoll såsom syslog, HTTP/HTTPS, GELF och NetFlow, och presenterar informationen i ett grafiskt gränssnitt där användare kan filtrera, söka och analysera loggar [16].

En central funktion hos Graylog är möjligheten att analysera stora mängder data i realtid. Genom olika verktyg och filter kan användaren snabbt identifiera specifika händelser och mönster i loggarna. Användaren kan även skapa dashboards och visualiseringar som ger en samlad överblick av säkerhetstillståndet i systemet [4], [16].

2.6 Transportprotokoll för loggöverföring

Vid överföring av loggmeddelanden över nätverket används vanligtvis transportprotokollen UDP eller TCP. Dessa protokoll har olika egenskaper gällande tillförlitlighet och prestanda.

UDP (User Datagram Protocol) är ett anslutningslöst protokoll som skickar data utan att bekräfta om mottagaren har tagit emot informationen. Detta gör UDP snabbt och effektivt men innebär att paket kan gå förlorade under överföringen om nätverket är belastat. UDP används typiskt i tillämpningar som video och röst där låg fördröjning prioriteras framför garanterad leverans [3], [6].

TCP (Transmission Control Protocol) är ett förbindelseorienterat protokoll som kräver en anslutning mellan avsändare och mottagare innan data kan skickas. TCP innehåller mekanismer för felkontroll och återöverföring av paket, vilket gör protokollet mer tillförlitligt.

Nackdelen med TCP är att kontrollmekanismerna skapar mer overhead och kan påverka prestandan i nätverket [3].

UDP utvecklades från början utan mekanismer för felkontroll, i syfte att hålla nere overhead och latens [3], [13]. När nätverket är belastat fortsätter UDP att skicka paket i samma takt, vilket förvärrar situationen med paketförluster [17]. I kontrast till detta arbetar TCP:s mekanismer för kongestionskontroll genom att sänka sändningshastigheten vid överbelastning, men de kan fortfarande inte garantera att redan buffrad data inte går förlorad [18].

Vid loggöverföring kan valet mellan TCP och UDP ha stor påverkan på tillförlitligheten i logginsamlingen, särskilt i miljöer där nätverket utsätts för hög belastning. Transportprotokollet är därmed en avgörande faktor vid analys av logghantering i nätverksmiljöer [3], [6].

2.7 Relaterad forskning

Tidigare forskning kring syslog prestanda och transportprotokoll ger ett viktigt sammanhang för denna studie. Huang et al. beskriver hur ett centraliserat syslog system samlar in och analyserar loggmeddelanden från routrar, switchar och brandväggar, och visar att centraliserad loggning effektivt kan identifiera avvikande nätverksbeteende [19].

Forskning om syslog transportprotokoll har visat på en tydlig avvägning mellan hastigheten hos UDP och pålitligheten hos TCP; TCP introducerades för att öka tillförlitligheten, vilket dock kan resultera i högre latens [7]. Genomgående analyser av SIEM-system har belyst hantering av stora datamängder i realtid som en av de mest avgörande funktionerna [4]. Det operativa värdet av SIEM-system är direkt kopplat till hur fullständig loggdatan är, vilket innebär att förlust av loggar utgör ett allvarligt säkerhetsproblem [15].

Nyligen genomförd forskning tyder på att UDP-baserad syslog kan leda till avsevärd loggförlust under kraftig nätverksbelastning, vilket kräver aktiva kompensationsåtgärder för att säkerställa loggintegritet [8]. Dessutom är SIEM baserade avvikelsedetektionssystem beroende av att loggdata är både komplett och konsekvent; saknade loggar skapar blinda fläckar i hotdetekteringen [12].

3 Metod

3.1 Forskningsdesign

Denna studie tillämpar en kvantitativ experimentell metod för att undersöka hur valet av transportprotokoll påverkar tillförlitligheten i logginsamlingen via syslog under nätverksbelastning. Metoden valdes eftersom studien syftar till att mäta konkreta och numeriska värden — loggförlust, CPU användning och minnesanvändning — som möjliggör objektiv jämförelse mellan de testade scenarierna [7].

En kontrollerad labbmiljö användes för att minimera externa störvariabler och säkerställa reproducerbarhet. För att öka den statistiska pålitligheten genomfördes varje testscenario

sju gånger (n=7). Resultaten från de sju körningarna visade konsekventa mönster, vilket tyder på att mätningarna är stabila och kan reproduceras. I resultat och analysavsnitten redovisas därför medelvärde, min–maxintervall och samtliga sju enskilda mätvärden per körning. Sett ur ett säkerhetsperspektiv är det värsta tänkbara utfallet minst lika viktigt som medelvärdet, då en enda missad loggsekvens kan ha stor betydelse vid analys av incidenter [15].

Den enda variabeln som ändrades mellan testerna var antingen transportprotokollet (TCP eller UDP) eller nätverksbelastningsnivån, medan hårdvara, mjukvara och testprocedur hölls konstanta genom hela studien. Detta säkerställer att observerade skillnader i loggförlust i huvudsak kan kopplas till den justerade variabeln snarare än till externa faktorer [19].

Det primära syftet med jämförelsen är att utvärdera TCP mot UDP som transportprotokoll för syslog. Lokal loggning ingår i studien som en referenspunkt för att illustrera skillnaden mot nätverksbaserad loggning, men eftersom lokal loggning sker utan nätverksöverföring påverkas den inte av nätverksbelastning och är därmed inte direkt jämförbar med TCP och UDP under samma förhållanden [1], [6].

3.2 Testmiljö

Testmiljön byggdes upp i högskolans laborationssal vid Högskolan Väst och bestod av följande komponenter:

- Router (Cisco, hostname: LOGGAR): konfigurerad som loggkälla och syslog klient.
- PC A (192.168.206.103): klientdator som genererade nätverksbelastning.
- PC B (192.168.205.203): klientdator som fungerade som trafikmottagare.
- Graylog VM (192.168.205.230): virtuell maskin med SIEM plattformen Graylog installerad som central loggserver.

För att separera trafiken segmenterades nätverket med VLAN via en switch. Routern hanterade kommunikationen mellan segmenten via subinterfaces: G0/0/0 (192.168.206.1), G0/0/1.30 (192.168.205.1) och G0/0/1.40 (192.168.204.1). Graylog valdes som SIEM plattform på grund av dess flexibla kapacitet att ta emot loggar via både TCP och UDP, samt dess inbyggda möjligheter att visualisera data i realtid [4], [16].

3.3 Loggkonfiguration

Routern konfigurerades för tre separata testscenarier:

- Lokal loggning (referenspunkt): loggar lagrades enbart i routerns interna buffer (logging buffered 64000 debugging) utan vidarebefordran till extern server. Detta scenario är inte direkt jämförbart med de nätverksbaserade scenarierna.
- Central loggning via TCP: routern konfigurerades att skicka syslog meddelanden till Graylog (192.168.205.230) på port 514 med oformaterad TCP som transportprotokoll [10] (numera med status Historic; TLS-krypterad syslog [10] är den rekommenderade

transporten i moderna miljöer men ingick inte i denna studies avgränsning). TCP erbjuder ökad tillförlitlighet via kvitteringsmekanismer och återöverföring vid paketförlust [10].

- Central loggning via UDP: samma konfiguration som ovan men med UDP som transportprotokoll på port 1514. UDP-baserad syslog saknar mekanismer för att bekräfta leverans [6].

På serversidan konfigurerades Graylog med separata inputs för TCP (port 514) och UDP (port 1514). Valet att testa båda protokollen motiveras av deras fundamentalt olika tekniska egenskaper, TCP erbjuder ökad tillförlitlighet via kvitteringsmekanismer och återöverföring vid paketförlust medan UDP skickar paket utan bekräftelse och saknar flödeskontroll [3], [6], [11].

3.4 Testprocedur och logggenerering

Logghändelser genererades via routerns extended-ping kommando mot Graylog servern: LOGGAR#ping 192.168.205.230 repeat 5000.

För att varje ICMP förfrågan skulle generera ett motsvarande loggmeddelande konfigurerades routern med kommandot debug ip icmp, vilket innebär att varje mottagen ICMP echo reply skrivs ut som ett loggmeddelande (exempel: ”ICMP: echo reply rcvd, src 192.168.205.230, dst 192.168.205.1”). Debug meddelanden skickas som standard endast till konsolen, men genom att konfigurera logging trap debugging vidarebefordrades meddelanden på debug nivå och allvarligare till de konfigurerade syslog mottagarna (Graylog via TCP respektive UDP, se avsnitt 3.3). Denna koppling mellan varje enskild ping förfrågan och ett tydligt, räkningsbart loggmeddelande är förutsättningen för att kunna beräkna loggförlust genom att jämföra antalet genererade ICMP förfrågningar med antalet mottagna loggmeddelanden i Graylog.

Valet av 5 000 ICMP förfrågningar baserades på två skäl: dels ger det en tillräckligt stor mängd data för att loggförlustens procentandel ska kunna bedömas som statistiskt tillförlitlig, dels är det praktiskt genomförbart utan att överbelasta routerns loggbuffer under ett enskilt test. ICMP valdes som trafiktyp eftersom varje pingförfrågan skapar ett tydligt och spårbart loggmeddelande i routern, vilket gör det möjligt att exakt räkna skillnaden mellan genererade och mottagna loggar [19]. Tester av TCP och UDP genomfördes under samma förhållanden med samma nätverkstopologi, liknande belastningsnivå och samma konfiguration av Graylog där transportprotokollet var den enda faktorn som varierade.

Varje test skickade exakt 5 000 ICMP förfrågningar med 100 byte paketstorlek, vilket genererade ett förutbestämt och kontrollerbart antal loggposter. Routerns loggbuffer rensades (clear logging buffer) innan varje testkörning för att säkerställa att enbart det aktuella testets loggar mättes. Varje scenario upprepades sju gånger med identisk procedur; routerns loggbuffer rensades inför varje ny körning.

Nätverksbelastning simulerades parallellt genom att köra flera instanser av ping kommandon med maximal paketstorlek från PC A: ping 192.168.205.1 -n 1 -l 65500 1>nul.

Maximal paketstorlek (65 500 byte) valdes för att skapa maximal nätverkslast. Belastningens effekt bekräftades empiriskt genom att ping till routern gav ”Request timed out” under pågående belastningstester, vilket indikerar att nätverket befann sig i ett tillstånd med aktiv paketdropp. Metoden att jämföra genererade mot mottagna loggar är en etablerad metod för att mäta loggförlust i experimentella studier [19], [7].

3.5 Datainsamling och mätvärden

Följande mätvärden registrerades för varje test:

- Genererade loggar: konstant 5 000 ICMP förfrågningar per test (kontrollerad variabel).
- Mottagna loggar: fastställdes genom att avläsa värdet ”Total results” i Graylogs sökgränssnitt (Search). Sökfältet lämnades tomt (ingen filtrering på meddelandeinnehåll), eftersom endast en logginput (TCP respektive UDP) var aktiv åt gången, vilket gjorde att samtliga inkommande meddelanden under den valda perioden härrörde från det aktuella testet. Källfältet (source) visade routerns interfaceadress (192.168.205.1) för samtliga meddelanden, vilket bekräftade att loggarna kom från rätt enhet. Tidsintervallet (From/Until) ställdes in manuellt för varje testkörning så att det motsvarade den tidsperiod då respektive test genomfördes. Graylogs index rensades mellan varje testkörning för att säkerställa att endast det aktuella testets meddelanden ingick i resultatet.
- Loggförlust (%): beräknades enligt formeln (genererade − mottagna) / genererade × 100.
- CPU användning: avläst med kommandot ”show processes cpu” på routern i obelastat och belastat tillstånd.
- Minnesanvändning: avläst med kommandot ”show processes memory” totalt Used-värde samt per process.

På routern kontrollerades att loggning skedde genom att inspektera den interna loggbufferten (show logging), vilket bekräftade att meddelanden genererades och skickades vidare under testerna. Det exakta antalet mottagna loggmeddelanden fastställdes dock i Graylog, eftersom detta utgjorde den centrala mottagningspunkten för samtliga tre testscenarier.

En metodologisk fråga är huruvida varje ICMP förfrågan garanterat genererar exakt ett loggmeddelande, eller om till exempel dubbla loggningar (t.ex. både vid sänd förfrågan och mottaget svar) skulle kunna resultera i fler mottagna meddelanden än genererade förfrågningar. Studien inkluderar ingen oberoende protokollnivåverifiering av detta, men det kan valideras empiriskt utifrån de insamlade mätvärdena: i samtliga fjorton testkörningar (sju för TCP, sju för UDP) var antalet mottagna loggmeddelanden i Graylog aldrig högre än antalet genererade ICMP förfrågningar (5 000) högsta uppmätta värdet var 4 948 TCP-test 2. Om varje ping hade genererat mer än ett loggmeddelande hade mottagna värden konse-

kvent överstigit 5 000, vilket inte observerades i något test. Detta stödjer att varje ping resulterade i högst ett motsvarande loggmeddelande, och att genererade mottagna jämförelsen därmed ger en giltig grund för att beräkna loggförlust.

CPU och minnesanvändning mättes en gång per tillstånd obelastat respektive belastat) och inte per enskild testkörning, till skillnad från loggförlusten som mättes i samtliga sju körningar. Dessa mätvärden möjliggör en kvantitativ och objektiv jämförelse mellan de testade scenarierna [3], [7]. För loggförlust anges både medelvärde, min–maxintervall och samtliga sju enskilda mätvärden i resultatkapitlet, vilket möjliggör en jämförelse som tar hänsyn till både normala och exceptionella resultat [3], [7].

4 Resultat

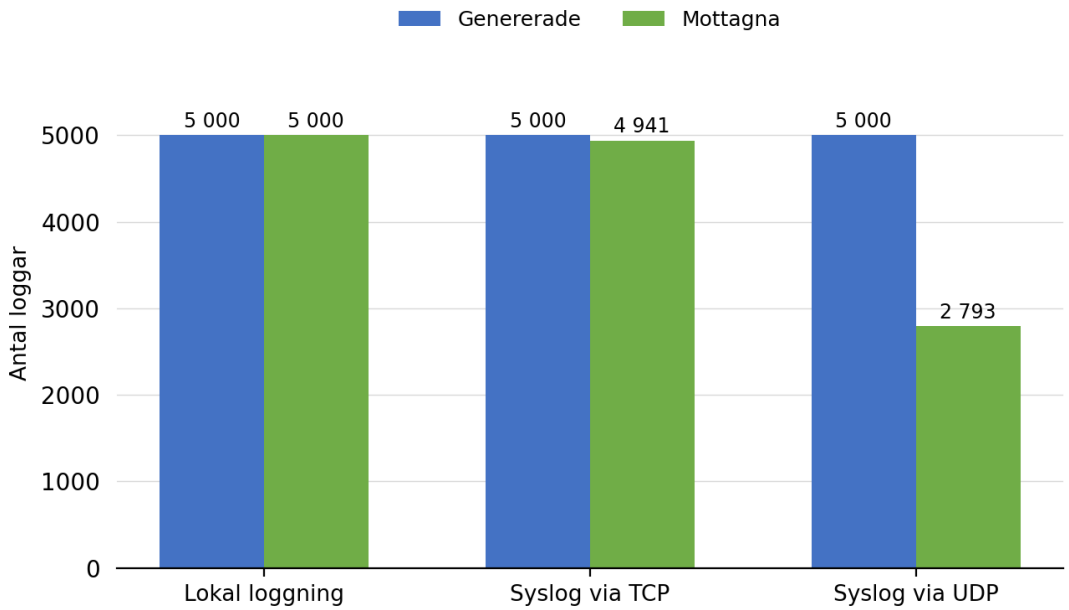
Detta kapitel redovisar resultaten från samtliga sju testkörningar (n=7) för respektive scenario: lokal loggning, syslog via TCP och syslog via UDP. Varje enskild körning redovisas i tabellform, tillsammans med medelvärde, så att spridningen mellan körningarna kan bedömas.

4.1 Genererade vs. mottagna loggar

Tabell 1 visar antal genererade och mottagna loggar för samtliga sju testkörningar för syslog via TCP respektive UDP. Vid lokal loggning registrerades samtliga 5 000 loggar utan förlust i samtliga sju körningar och redovisas därför inte i egen tabell. Figur 2 nedan illustrerar medelvärdet av genererade och mottagna loggar för de tre scenarierna.

Tabell 1. Egen tabell baserad på egna testresultat. Genererade och mottagna loggar samt beräknad loggförlust för samtliga sju testkörningar, syslog via TCP respektive UDP (n=7).

Test	Gen. TCP	Mott. TCP	Förlust TCP	Gen. UDP	Mott. UDP	Förlust UDP
Test 1	5 000	4 934	1,32 %	5 000	2 758	44,84 %
Test 2	5 000	4 948	1,04 %	5 000	2 764	44,72 %
Test 3	5 000	4 937	1,26 %	5 000	2 769	44,62 %
Test 4	5 000	4 941	1,18 %	5 000	2 780	44,40 %
Test 5	5 000	4 946	1,08 %	5 000	2 806	43,88 %
Test 6	5 000	4 943	1,14 %	5 000	2 816	43,68 %
Test 7	5 000	4 939	1,22 %	5 000	2 858	42,84 %
Medel	**5 000**	**4 941**	**1,18 %**	**5 000**	**2 793**	**44,14 %**



Figur 2. Egen figur baserad på egna testresultat. Genererade jämfört med mottagna loggar (medelvärde av sju körningar) för lokal loggning, syslog via TCP och syslog via UDP.

4.2 Loggförlust per metod (%)

Vid lokal loggning noterades en förlust på 0 % i samtliga sju körningar. Tabell 2 och Tabell 3 nedan visar den beräknade loggförlusten per körning för TCP respektive UDP. För TCP varierade loggförlusten mellan 1,04 % och 1,32 %, med ett medelvärde på 1,18 %. För UDP varierade loggförlusten mellan 42,84 % och 44,84 %, med ett medelvärde på 44,14 %. Figur 3 nedan visar den genomsnittliga loggförlusten per scenario.

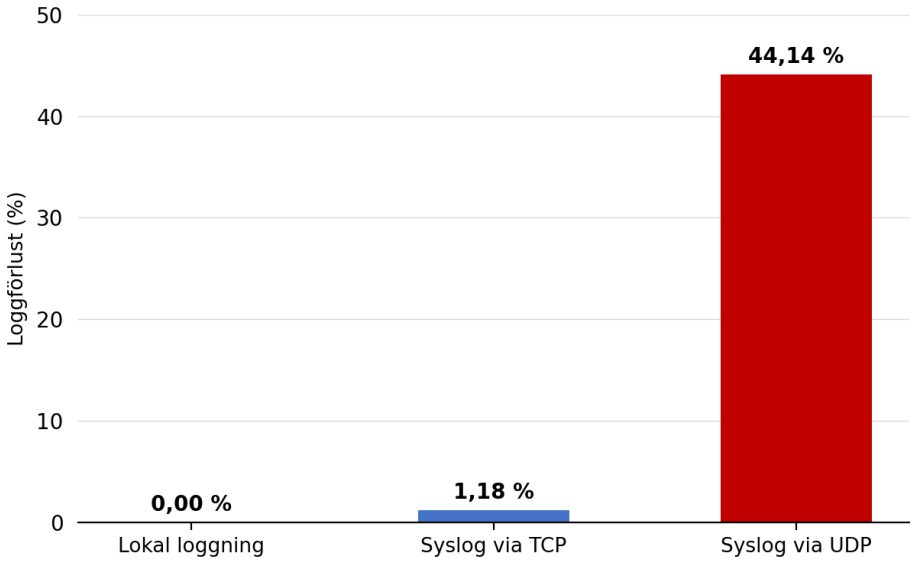
Tabell 2. Egen tabell baserad på egna testresultat. Loggförlust per testkörning, syslog via TCP (n=7).

Testkörning	Mottagna loggar	Loggförlust (%)
Test 1	4 934	1,32 %
Test 2	4 948	1,04 %
Test 3	4 937	1,26 %
Test 4	4 941	1,18 %
Test 5	4 946	1,08 %
Test 6	4 943	1,14 %
Test 7	4 939	1,22 %
Medel	**4 941**	**1,18 %**

Tabell 3. Egen tabell baserad på egna testresultat. Loggförlust per testkörning, syslog via UDP (n=7).

Testkörning	Mottagna loggar	Loggförlust (%)
Test 1	2 758	44,84 %
Test 2	2 764	44,72 %
Test 3	2 769	44,62 %
Test 4	2 780	44,40 %
Test 5	2 806	43,88 %
Test 6	2 816	43,68 %
Test 7	2 858	42,84 %

Medel	**2 793**	**44,14 %**

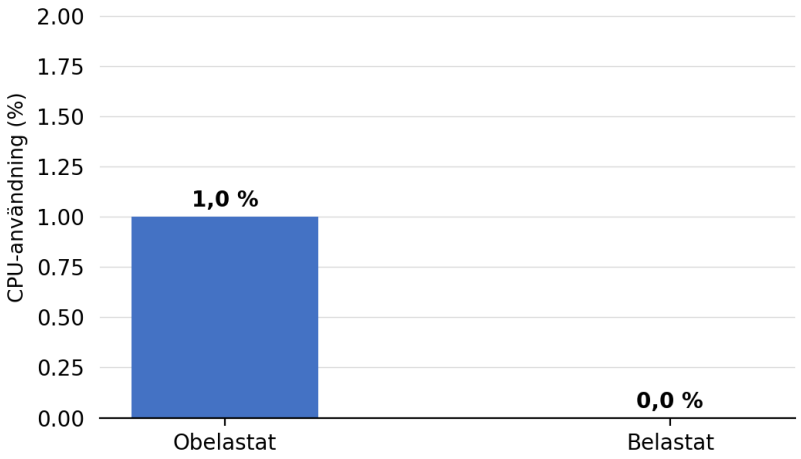


Figur 3. Egen figur baserad på egna testresultat. Genomsnittlig loggförlust (%) per scenario (medelvärde av sju körningar).

Resultaten visar att loggförlusten var konsekvent låg för TCP och konsekvent hög för UDP i samtliga sju körningar (se Tabell 2 och Tabell 3), utan enstaka avvikande värden i någon av mätserierna. Det tyder på att skillnaden mellan protokollen är stabil under de testade förhållandena, snarare än ett resultat av en enskild extremkörning.

4.3 CPU användning (obelastat vs. belastat)

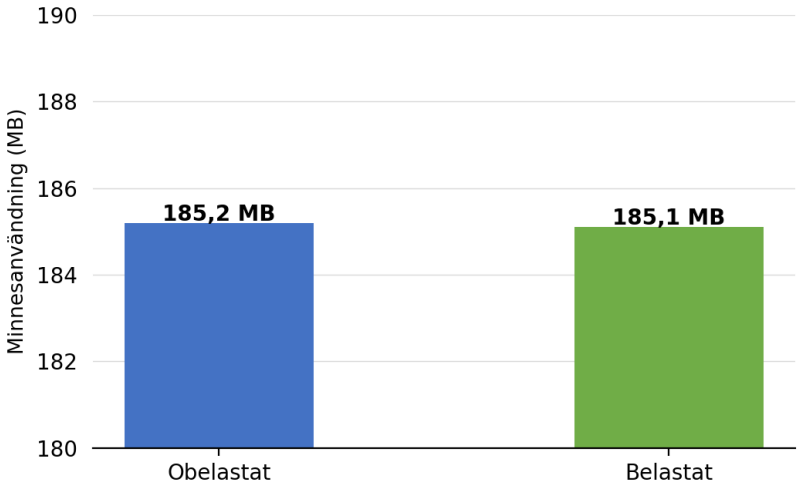
CPU användningen på routern mättes till 1 % i obelastat tillstånd och 0 % i belastat tillstånd, se Figur 4. Mätningen genomfördes en gång per tillstånd och inte per enskild testkörning. Resultatet indikerar att nätverksbelastningen inte påverkade routerns processoranvändning på ett märkbart sätt, men eftersom mätningen inte upprepades sju gånger på samma sätt som loggförlusten kan slumpmässig variation mellan enskilda körningar inte uteslutas i samma utsträckning.



Figur 4. Egen figur baserad på egna testresultat. CPU användning på routern, obelastat vs. belastat tillstånd.

4.4 Minnesanvändning (obelastat vs. belastat)

Minnesanvändningen låg stabilt på 185,2 MB i obelastat tillstånd och 185,1 MB i belastat tillstånd, en skillnad på cirka 100 kB, se Figur 5. Liksom för CPU användningen mättes minnesanvändningen en gång per tillstånd, inte per enskild körning.



Figur 5. Egen figur baserad på egna testresultat. Minnesanvändning på routern, obelastat vs. belastat tillstånd.

5 Analys/diskussion

Syftet med den här studien var att på empirisk basis undersöka hur valet av transportprotokoll påverkar logginsamlingens tillförlitlighet via syslog under belastning på nätverket, samt att kvantifiera betydelsen av dessa skillnader ur ett säkerhetsperspektiv. De sju upprepade testkörningarna har resulterat i ett konsekvent empiriskt underlag som belyser tydliga skillnader mellan de tre testscenarierna.

5.1 Lokal loggning

Lokal loggning var bäst ur loggförlustperspektiv eftersom den inte visade några tappade loggar, vilket var förväntat eftersom loggarna sparas direkt på enheten utan att behöva skickas över nätverket. Det är väsentligt att framhålla att lokal loggning inte kan jämföras direkt med nätverksbaserad loggning under liknande omständigheter. Eftersom ingen information överförs via nätverket påverkas lokal loggning inte av nätverksbelastningen, vilket gör att 0 % loggförlust framstår som ett förväntat och självklart utfall snarare än en rättvis jämförelse mot TCP och UDP [1], [6]. Därför inkluderas lokal loggning i studien enbart som en referenspunkt.

Detta gör lokal loggning pålitlig, men den har också sina begränsningar: bufferten blir full relativt snabbt, och loggarna är inte tillgängliga centralt för analys och korrelation med

andra system. Om en av enheterna kompromitteras och loggarna raderas lokalt finns inget annat sätt att spåra händelsen, vilket är en allvarlig säkerhetsbrist i praktiska miljöer [1].

5.2 Central loggning via TCP

Central loggning via TCP visade en genomsnittlig loggförlust på 1,18 %, vilket motsvarar ungefär 59 loggar per körning. I det värsta uppmätta fallet (test 1) förlorades 66 loggar (1,32 %).

TCP har kvitteringsmekanismer som validerar att varje paket kommer fram korrekt och skickar om paket som inte bekräftas [3]. Den kvarstående förlusten kan förklaras av hur TCP-stacken hanterar sändningsbufferten. Avsändaren betraktar ett meddelande som levererat så snart det skickas till operativsystemets sändningsbuffert, i stället för när mottagaren faktiskt bekräftar mottagandet [11]. Vid nätverksöverbelastning kan data som redan buffrats gå förlorad innan TCP hinner upptäcka förlusten och påbörja återsändning, även om TCP:s kongestionskontroll minskar sändningshastigheten [18]. Vid nätverkskongestion kan alltså buffrad data gå förlorad trots protokollets kvitteringsmekanismer [11].

Sammanfattningsvis visade TCP en betydligt lägre loggförlust än UDP i den testade miljön. Huruvida en förlust på 1,18 % är godtagbar beror på kraven i den specifika verksamheten och kan inte generaliseras utifrån denna studies resultat.

5.3 Central loggning via UDP

Central loggning via UDP visade en genomsnittlig loggförlust på 44,14 % över de sju testerna, med ett värsta uppmätt fall på 44,84 %. UDP är ett anslutningslöst protokoll som skickar paket utan att bekräfta om de har mottagits eller inte [3]. UDP saknar mekanismer för att korrigera och detektera loggförlust [6]. Under nätverksbelastning kastas UDP paket bort eftersom protokollet saknar flödeskontroll, och förlorade loggar kan inte rekonstrueras i efterhand.

Att ping till routern gav ”Request timed out” bekräftar att nätverket var under belastning under testerna. När nätverket är belastat begränsar TCP:s flödeskontroll sändningsfrekvensen, medan UDP fortsätter att skicka med hög hastighet, vilket bidrar till paketförlust.

En genomsnittlig loggförlust på 44,14 % innebär att nästan hälften av alla nätverkshändelser inte loggades i Graylog under de testade förhållandena. Eftersom studien mätte loggförlust för ICMP trafik och inte faktiska säkerhetshändelser, kan inga direkta slutsatser dras om hur detta skulle påverka ett SIEM-systems förmåga att detektera verkliga intrångsförsök. Resultatet visar dock att en hög andel förlorade loggmeddelanden generellt minskar mängden information tillgänglig för säkerhetsanalys, vilket är relevant att beakta vid val av transportprotokoll [15], [12].

På mottagarsidan kan Graylog VM:ens mottagningsbuffer överskridas om loggmeddelanden anländer snabbare än de kan bearbetas, vilket kan bidra till ytterligare loggförlust på applikationsnivå [16].

Det är viktigt att inte enbart tillskriva förlusten till UDP protokollet som en isolerad faktor. Andra faktorer som kan spela in inkluderar mottagningsbufferten hos Graylog, UDP socket bufferten i operativsystemet på den mottagande enheten samt eventuell bakgrundstrafik i labbmiljön [16]. Dessa faktorer har inte mätts direkt i denna studie. Det som kan observeras är att förlusten uppträder konsekvent vid hög nätverksbelastning och uteblir i TCP scenariot under liknande förhållanden, vilket tyder på att protokollmekanismen är den huvudsakliga förklaringen, men detta kan inte betraktas som slutgiltigt bevisat inom ramen för denna studies mätomfång.

5.4 CPU och minnesanvändning

CPU användningen på routern förblev extremt låg i samtliga testscenarier: 1 % i obelastat tillstånd och 0 % i belastat tillstånd. Detta kan verka paradoxalt att CPU användningen sjunker under belastning men förklaras av att Cisco IOS XE routerns paketförmedling hanteras av ett hårdvaruaccelererat ASIC chip snarare än av routerns huvudprocessor. Processorn används primärt för kontrollplanet, medan dataplanet, inklusive paketförmedling, hanteras av dedikerad hårdvara [3].

Loggningsprocesserna i syslog (PID 524) och EEM ED syslog (PID 476) visade ingen ökad CPU aktivitet under testerna, vilket stödjer slutsatsen att logghantering i sig inte belastar routerns processor nämnvärt.

Minnesanvändningen var stabil på cirka 185 MB i samtliga scenarier; skillnaden mellan belastat och obelastat tillstånd var cirka 100 kB, vilket ligger inom normal variation och inte kan anses signifikant.

Det mest betydelsefulla resultatet är att routerns hårdvarukapacitet inte tycks vara den begränsande faktorn för loggförlusten vid UDP. Detta stärker slutsatsen att protokollets avsaknad av flödeskontroll och kvitteringsmekanismer är den huvudsakliga förklaringen [3], [6], men eftersom inte samtliga tänkbara faktorer (t.ex Graylogs mottagningsbuffer eller switchens hantering av trafiken) har mätts separat, kan studien inte bevisa att UDP är den enda orsaken.

5.5 Skillnader mellan metoderna

Den mest meningsfulla jämförelsen i studien är mellan TCP och UDP som transportprotokoll för syslog under identiska nätverksförhållanden. Resultaten visar en stor skillnad: TCP uppvisade en genomsnittlig loggförlust på 1,18 % (värsta fall 1,32 %), medan UDP uppvisade en genomsnittlig loggförlust på 44,14 % (värsta fall 44,84 %).

Det är viktigt att notera att nätverksbaserad loggning och lokal loggning inte sker under samma förhållanden, vilket gör en direkt jämförelse akademiskt begränsad. Nätverksbelastningen har ingen effekt på lokal loggning eftersom data inte skickas över nätverket, därför är 0 % loggförlust ett förväntat och självklart utfall snarare än direkt jämförbart med transportprotokollen TCP och UDP [1]. Den verkliga och rättvisa jämförelsen i studien görs därför mellan TCP och UDP under likadana belastningsförhållanden.

Ur säkerhetsperspektiv är skillnaden relevant. SIEM-systems förmåga att detektera hot är beroende av fullständigheten hos loggdata [4], [15]. Studiens resultat visar att en betydande andel loggmeddelanden går förlorade vid UDP under hög nätverksbelastning, vilket i förlängningen skulle kunna minska mängden information tillgänglig för säkerhetsanalys. Eftersom studien inte undersökt hur denna typ av loggförlust påverkar detektion av faktiska säkerhetshändelser kan ingen slutsats dras om konkreta konsekvenser för hotdetektion i den testade miljön.

Det bidrag som denna forskning ger, utöver vad teorin redan antyder, är en mätning av den faktiska skillnaden i en kontrollerad miljö som simulerar ett litet kontorsnätverk, med ett tillräckligt antal upprepningar (n=7) för att minska effekten av slumpmässig variation. Resultaten visar empiriskt att TCP i denna miljö inte bara var mer pålitligt, utan att skillnaden också hade en nivå som är relevant för praktisk säkerhet.

Tidigare studier visar att det operativa värdet av SIEM-system i hög grad beror på hur kompletta loggdata är [4], [15], [12]. Mot bakgrund av NIS2 direktivets krav på både spårbarhet och pålitlig loggning ger resultaten från denna studie ett empiriskt stöd för att UDP, i den testade miljön, inte uppfyller de krav som kan förväntas för säkerhetskritisk loggning under nätverksbelastning [5].

5.6 Begränsningar och framtida arbete

Studien har flera begränsningar som bör beaktas vid tolkningen av resultaten. Testmiljön bestod bara av en enda Cisco router som loggkälla, vilket inte är representativt för stora produktionsmiljöer med många simultana loggkällor [19].

En mer realistisk testmiljö hade inkluderat ett antal olika nätverkskomponenter, såsom routrar, switchar och brandväggar, som samverkar för att producera loggdata samtidigt.

Nätverksbelastningen simulerades med ICMP paket av maximal storlek (65 500 byte), vilket är enklare jämfört med trafiken i verkliga nätverk [7]. Resultaten reflekterar därför detta specifika belastningsscenario och bör inte generaliseras utan vidare tester.

Testerna genomfördes i högskolans laborationssal där bakgrundstrafik från andra studenter inte kan uteslutas. En fullt isolerad nätverksmiljö hade gett mer precisa resultat.

För framtida arbete skulle det vara värdefullt att: (1) bredda testmiljön genom att inkludera fler parallella loggkällor, till exempel switchar och brandväggar, (2) utforska om TLS-kryptering över TCP har någon effekt på loggförlusten under hög belastning, och (3) undersöka

om syslog-ng eller r-syslog med kömekanismer kan minska den kvarstående förlusten vid TCP ytterligare.

6 Slutsats

Syftet med denna studie var att empiriskt analysera hur valet av transportprotokoll, TCP eller UDP, påverkar logginsamlingssystemets tillförlitlighet via syslog under förhållanden av nätverksbelastning, samt att kvantifiera de praktiska säkerhetskonsekvenserna kopplade till denna skillnad. Resultaten från sju genomförda och konsekventa testkörningar (n=7) ger svar på studiens tre centrala frågeställningar.

Gällande protokollvalets påverkan på tillförlitligheten visade resultaten att TCP hade en genomsnittlig loggförlust på 1,18 % (värsta uppmätta utfall 1,32 %), medan UDP uppvisade en loggförlust på 44,14 % i genomsnitt (värsta fall 44,84 %). Skillnaden förklaras i huvudsak av TCP:s inbyggda mekanismer för bekräftelse och återöverföring, till skillnad från UDP, som saknar dessa funktioner och fortsätter att sända data i samma takt även under belastning. Det innebär att nästan hälften av loggmeddelandena i den testade UDP konfigurationen aldrig nådde Graylog.

Angående säkerhetskonsekvenserna visar resultaten att en betydande andel loggmeddelanden går förlorade vid UDP under hög nätverksbelastning i den testade miljön. Eftersom det operativa värdet hos SIEM-system är beroende av loggdatans fullständighet [15], kan en sådan förlust i förlängningen minska mängden information tillgänglig för säkerhetsanalys. Studien har dock inte undersökt hur denna loggförlust påverkar förmågan att faktiskt detektera säkerhetshändelser, och några direkta slutsatser om detta kan därför inte dras.

NIS2 direktivet ställer krav på att berörda organisationer ska ha förmåga att förebygga, upptäcka och hantera säkerhetsincidenter, vilket förutsätter tillförlitlig logghantering [5]. Studiens resultat visar att UDP under hög nätverksbelastning i den testade miljön uppvisade en betydligt högre loggförlust än TCP, vilket är en faktor att beakta vid protokollval i verksamheter som omfattas av sådana krav. Studien gör dock ingen fullständig regelefterlevnadsbedömning, vilket skulle kräva en bredare analys än vad denna undersökning omfattar.

När det gäller CPU och minnesanvändning förblev routerkapaciteten stabil och låg: CPU användningen var under 1 %, och minnesanvändningen varierade med endast cirka 100 kB. Detta indikerar att loggförlusten vid UDP i denna studie inte tycks bero på hårdvarans kapacitet, utan snarare på begränsningar inom protokollet.

Dessa resultat bör inte tolkas som generellt gällande för alla typer av nätverksmiljöer. Undersökningen genomfördes med en specifik Cisco router, en enda trafiktyp (ICMP) och en definierad labbkonfiguration. I produktionsmiljöer med andra loggkällor, andra trafiktyper eller annan hårdvara kan mönstren för loggförlust se annorlunda ut. Inom ramen för den testade miljön rekommenderas TCP framför UDP som transportprotokoll för syslog-baserad logghantering i situationer där kompletta loggdata är ett krav. Resultatet bör dock

inte tolkas som en generell rekommendation för alla nätverksmiljöer, utan som en indikation som bör verifieras i den egna produktionsmiljön innan den tillämpas.

Framtida arbete: för kommande arbete skulle det vara värdefullt att utvidga testmiljön med ytterligare loggkällor som switchar och brandväggar, att utforska om TLS-kryptering över TCP har någon inverkan på loggförlusten, samt att undersöka om användning av syslog-ng eller r-syslog med kömekanismer kan minska den kvarstående loggförlusten vid TCP ytterligare.

Referenser

[1] W. Stallings, Network Security Essentials, 6th ed. Boston, MA, USA: Pearson, 2016.[Länk]

[2] R. Gerhards, ”The Syslog Protocol,” RFC 5424, Internet Engineering Task Force (IETF), Mar. 2009.[Länk]

[3] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 8th ed. Boston, MA, USA: Pearson, 2021.[Länk]

[4] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, ”Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures,” Sensors, vol. 21, no. 14, art. 4759, Jul. 2021, doi: 10.3390/s21144759.[Länk]

[5] European Parliament and Council of the European Union, ”Directive (EU) 2022/2555 of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive),” consolidated text (CELEX:02022L2555), incorporating Corrigendum OJ L, 22.12.2023, Off. J. Eur. Union, EUR-Lex, accessed Mar. 2026.[Länk]

[6] C. Lonvick and A. Okmianski, ”Transmission of Syslog Messages over UDP,” RFC 5426, IETF, Mar. 2009.[Länk]

[7] M. Rajiullah, R. Lundin, A. Brunstrom, and S. Lindskog, ”Syslog performance: Data modeling and transport,” in Proc. 2011 3rd Int. Workshop on Security and Communication Networks (IWSCN), Gjøvik, Norway, 2011, pp. 31–37, doi: 10.1109/IWSCN.2011.6827714.[Länk]

[8] A.-A. Anton, P. Csereoka, E. A. Capota, and R.-D. Cioargă, ”Enhancing Syslog Message Security and Reliability over Unidirectional Fiber Optics,” Sensors, vol. 24, no. 20, art. 6537, Oct. 2024, doi: 10.3390/s24206537.[Länk]

[9] K. Kent and M. Souppaya, Guide to Computer Security Log Management, NIST Special Publication 800-92, National Institute of Standards and Technology, Gaithersburg, MD, USA, Sep. 2006.[Länk]

[10] R. Gerhards and C. Lonvick, ”Transmission of Syslog Messages over TCP,” RFC 6587, IETF, Apr. 2012.[Länk]

[11]F. Miao, Ed., Y. Ma, Ed., and J. Salowey, Ed., ”TLS Transport Mapping for Syslog,” RFC 5425, IETF, Mar. 2009.[Länk]

[12]M. Wurzenberger, F. Skopik, G. Settanni, and R. Fiedler, ”AECID: A Self-learning Anomaly Detection Approach Based on Light-weight Log Parser Models,” in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP 2018), Funchal, Madeira, Portugal, Jan. 2018, pp. 386–397.[Länk]

[13]A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, ”An Overview of IP Flow-Based Intrusion Detection,” IEEE Commun. Surveys Tuts., vol. 12, no. 3, pp. 343–356, Third Quarter 2010, doi: 10.1109/SURV.2010.032210.00054.[Länk]

[14]K. Scarfone, T. Grance, and K. Masone, Computer Security Incident Handling Guide, NIST Special Publication 800-61 Revision 1, National Institute of Standards and Technology, Gaithersburg, MD, USA, Mar. 2008.[Länk]

[15]S. Bhatt, P. K. Manadhata, and L. Zomlot, ”The Operational Role of Security Information and Event Management Systems,” IEEE Security & Privacy, vol. 12, no. 5, pp. 35–41, Sep./Oct. 2014, doi: 10.1109/MSP.2014.103.[Länk]

[16]J. Manzoor, A. Waleed, A. F. Jamali, and A. Masood, ”Cybersecurity on a budget: Evaluating security and performance of open-source SIEM solutions for SMEs,” PLoS ONE, vol. 19, no. 3, e0301183, Mar. 2024, doi: 10.1371/journal.pone.0301183.[Länk]

[17]S. Floyd and V. Jacobson, ”Random early detection gateways for congestion avoidance,” IEEE/ACM Trans. Netw., vol. 1, no. 4, pp. 397–413, Aug. 1993, doi: 10.1109/90.251892.[Länk]

[18]V. Jacobson, ”Congestion avoidance and control,” ACM SIGCOMM Comput. Commun. Rev., vol. 18, no. 4, pp. 314–329, Aug. 1988, doi: 10.1145/52325.52356.[Länk]

[19]J.-h. Huang, M.-q. Zhang, and Y.-l. Jiang, ”The design and implement of the centralized log gathering and analysis system,” in Proc. 2012 IEEE Int. Conf. Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, 2012, pp. 268–273, doi: 10.1109/CSAE.2012.6272772.[Länk]