NTNU
Det skapende universitet

Effektiv logganalyse i et heterogent driftsmiljø

Vegard Fremstad

Master i lærerutdanning med realfag
Oppgaven levert: April 2009
Hovedveileder: Anders Christensen, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

Forord

Denne oppgaven er det avsluttende arbeidet i en mastergrad ved Norges Teknisk-Naturvitenskapelige Universitet (NTNU). Oppgaven er utført i perioden september 2008 til april 2009, og er utført ved Institutt for Datateknikk (IDI).

Først og fremst vil jeg takke min kjære Rannveig for å ha tatt hovedansvar hjemme og tildels nektet meg deltakelse i husarbeidet, uten at det har resultert i store protester. At du i tillegg har godtatt alle mine forsøk på å systematisere papir på alle mulige ledige plasser i huset har også vært til stor hjelp, overgått kun av bidraget på korrekturlesing. Håper jeg kan gjengjelde noe av dette i tiden fremover! Skal naturligvis stille på alle kino-, teater- og tur-forslag uten å mukke.

En stor takk sendes til min gode medarbeider Steinar, da han har tatt for seg av alle mulige oppgaver som har kommet underveis, og latt meg fokusere på noen få ting på jobb, som igjen har gjort det mulig å konsentrere seg om denne oppgaven utenom arbeidstiden. Du vet jeg setter stor pris på det!

Her skal heller ikke glemmes mine veiledere. Vi har etterhvert hatt mange sesjoner, og de har bidratt sterkt til at fokus har blitt holdt på hovedoppgavene, og ikke forsvunnet ut i de små mindre vesentlige detaljene. Har også fått mange gode tips til fremgangsmåte, formaliteter og rapportskrivingen. Tusen takk for god veiledning!

Jeg setter virkelig pris på dette, og håper denne oppgaven holder like høy kvalitet som hjelpen jeg har fått underveis!

_____

Trondheim, 07.04.2009
Vegard Fremstad

Sammendrag

Oppgaven tar for seg utredning og testimplementasjon av en delvis automatisk logganalyse for servere og nettverkskomponenter på Kristen Videregående skole Trøndelag (KVT). Dette har som formål å effektivisere analysen av loggdata, på en slik måte at problemer kan finnes og håndteres på et tidligst mulig tidspunkt, uten at manuell gjennomgang av logger kreves for å finne problemene.

Innledningsvis blir terminologien og teorien rundt logging tatt opp, da en forståelse av dette er nødvendig for å kunne vurdere hva en delvis automatisert logganalyse vil kunne gi av fordeler og eventuelle ulemper. Validering av resultatene krever også god innsikt i enkeltkomponentene som ligger til grunn for resultatene.

Loggsystemene som blir omtalt og anvendt i oppgaven er syslog, syslog-ng og Event Viewer, hvor syslog-ng blir valgt som loggsystem for loggserverne.

Tre loggservere som mottar samme loggdata fra andre servere og nettverkskomponenter på KVT blir satt opp med henholdsvis Lire, Sawmill og Splunk, og resultatene fra analysene som disse 3 logganalyseverktøyene gir blir drøftet.

Det kan ikke på bakgrunn av denne oppgaven konkluderes med sikkerhet hvorvidt de omtalte verktøyene medfører en mer effektiv drift av nettverket på KVT. Hver for seg har de vesentlige mangler i forhold til å kunne utføre en komplett logganalyse, men samtidig tilfører de enkeltelementer som kan være effektiviserende.

Innhold

1 Oppgavetekst

Tittel:

"Effektiv logganalyse i et heterogent driftsmiljø."

Beskrivelse:

Utredning og testimplementasjon av delvis automatisert logganalyse av loggdata fra ulike nettverkskilder, som inkluderer både servere og enkeltstående nettverkskomponenter. Formålet er å gjøre de IT-ansvarlige mer effektive gjennom å få bedre oversikt over hendelser på enheter i nettverket.

2 Introduksjon

Kristen Videregående skole Trøndelag (heretter KVT) er en videregående skole som holder til i Trondheim kommune. De har omlag 500 elever, og en stab på cirka 60 personer. KVT har siden midten av 1990-tallet profilert seg som en foregangsskole hva anvendelse av teknologi angår, hvilket har resultert i en stor og omfattende maskinpark, både på server og klientsiden.

I de siste årene har antallet bærbare klientmaskiner økt betraktelig, samtidig som flere og flere digitale tjenester blir tatt i bruk, både av elever og ansatte. Dette har ført til knapphet i personellressursene på driften av IT-anlegget, og det er dermed ønskelig å kunne effektivisere driften, fortrinnsvis uten å tape kvalitet på det som blir utført. Et av elementene som kan forbedres er oppfølging og kontroll av servere, som i dag gjøres per manuell inspisering av logger, og typisk først når det oppdages feil eller problemer.

Dette resulterte i et ønske fra KVT om å undersøke hvorvidt det med deres maskinpark var mulig å implementere en løsning for logganalyse som kunne gjøre det mulig å gjøre oppfølgingen av servere og andre nettverkskomponenter mer effektiv. Og da ideelt sett på en slik måte at det blir mulig å finne eventuelle problemer og rette opp i de før brukerne merket noe til de.

2.1 Kravspesifikasjon

Systemet for logganalyse som skal utredes og utvikles skal oppfylle disse kravene:

- Løsningen skal kunne takle alle typer servere som er i produksjonsmiljøet ved KVT, hvilket inkluderer Windows 2000 Server, Windows 2003 Server, Ubuntu Server, FreeBSD, Novell Netware/Open Enterprise Server[1].
- Løsningen skal sørge for at rapporter genereres/tilgjengeliggjøres på gitte tidspunkt, slik at de IT-ansvarlige kan få en god oversikt over ve-

[1]Et skifte til Novell/SUSE OES er på trappene i løpet av høsten, så om systemet kun er tilpasset det nye systemet, kan det godtas

sentlige hendelser på de enkelte systemene. Disse rapportene skal føre til mindre behov for manuell logganalyse, og dermed være tidsbesparende. Ideelt skal de inneholde god nok informasjon slik at man finner problemer før de får vesentlig konsekvenser, hvis mulig.

- Det skal være mulig å utvide løsningen på et senere tidspunkt, slik at man får inkludert nye servere, samt også ta inn vanlige nettverkskomponenter som switcher, routere og skrivere. Dette forutsetter dog at disse komponentene har støtte for å sende loggdata over nettverket.
- Det er ønskelig å kunne utrede og implementere en løsning som medfører minst mulig ekstrakostnader for KVT. Om dette viser seg vanskelig skal ledelsen ved KVT orienteres og en beslutning tas i fellesskap. KVT stiller med maskinvare til løsningen.
- Bruk av Open Source er ønskelig, for lettere å kunne modifisere løsningen om det skulle dukke opp nye behov på et senere tidspunkt.

3 Bakgrunn

3.1 Logghistorie

Helt siden de første datamaskinene kom i bruk har det vært et behov for å kunne se tilbake på hva som skjedde under en "kjøring" av et sett med instruksjoner. På slutten av 50-tallet, da forskerne begynte å få tilgang til å utføre beregninger på datamaskiner ble instruksjonene gjerne matet som serier med hullkort og resultatet av ferdig kjøring var enten et resultat (som ønsket), eller beskjed om at en feil hadde oppstått underveis, og at de måtte gjøre om programmeringen (endre på hullkortene). Sammen med beskjeden om at en feil hadde oppstått, var det knyttet en logg med oppføringer over hva som gikk galt[8]. Logging var altså en vesentlig del av datamaskinsystemene fra de ble tatt i bruk.

Etterhvert som datamaskinene har fått utvidet funksjonalitet, kapasitet, ytelse og utbredelse har logging av hendelser fortsatt å være en selvsagt del av ethvert operativsystem, både på ordinære personlige datamaskiner, servere, nettverkskomponenter og også etterhvert andre mer spesialiserte komponenter som alarmenheter, UPS'er og lignende.

Disse loggene har, avhengig av plattform, maskinvare og bruk blitt lagret i ulike formater og med ulik type informasjon, men har som fellesnevner at de i hovedsak inneholder en oppføring per hendelse på den aktuelle enheten. De mest brukte loggtypene i dag er syslog for Unix, Linux og Macintosh, og Windows Event Log[2] for nettopp Windows. Hvilke hendelser som blir lagret i logger varierer fra enhet til enhet, og kan også ofte styres av administratoren på den aktuelle enheten.

I dag, hvor datamaskinene blir brukt som støtte til det meste av arbeidsoppgaver, er behovene for logging blitt stadig større[3]. Det er ønskelig, for ikke å si nødvendig, å ha gode og stabile datasystemer, og da er logging av hendelser og den tilhørende analysen avgjørende faktorer for å sette inn tiltak som

[2]Windows Event Log er en utvidet utgave av det som tidligere het Event Viewer
[3]Perspektivet er her arbeidslivet, og ikke privatsfæren.

forbedrer systemene.

3.2 Hva og hvorfor logger vi?

Et datasystem, det være seg en separat konfigurert enhet som for eksempel en switch, eller en server som yter en eller flere tjenester, har et eller flere spesifikke formål det skal oppfylle som en del av nettverket. Om begrepet normalmodus blir brukt om den ønskede/ideelle situasjonen, er det altså ønskelig at enheten skal operere i normalmodus så ofte som mulig.

Av og til hender det ting som bringer enheten ut av normalmodus. Dette kan være en switch som feiler, et strømbrudd, en tjeneste som går i stå eller andre forstyrrelser i forbindelse med nettverket. Det må da eksistere et system som gjør at feilen for det første blir oppdaget, og deretter gjør administratorene i stand til å sette i verk tiltak for å bringe enheten tilbake til det ønskede normalmoduset igjen. Om dette i tillegg skjer før brukerne merker at noe er i uorden vil administratorene være spart for mange forespørsler.

Logging er en av de mekanismene som både gjør det mulig å finne ut at en enhet har gått ut av normalmodus, og i tillegg kan hjelpe til med informasjon om hvordan enheten igjen kan komme tilbake i orden. Logging ivaretar ideelt alle vesentlige[4] hendelser som inntreffer på det aktuelle systemet, og tar vare på disse i henhold til hvordan selve loggsystemet er konfigurert. Normalt betyr dette logging til filer på harddisken, eller rett inn i en databaseløsning.

Unntak fra dette vil kunne være en maskinvarefeil som fører til maskinstans. En datamaskin/komponent som "fryser" vil ikke være i stand til å sende loggmeldinger, selv om den kan registrere at en feil har forekommet neste gang den startes opp igjen. Selv om det ikke vil kunne dekke absolutt alle hendelser, er logging nødvendig for ethvert datasystem, ettersom uforutsette hendelser innimellom vil inntreffe, og det da må eksistere en metode for å diagnostisere og isolere årsaken[2] til problemet.

[4]Mengden og hvilke hendelser som skal logges kan og bør konfigureres i henhold til normalsituasjonen til den enkelte enhet. Om man har diskplass kan man naturligvis velge å logge alt.

Logging har sin sterke side på å melde om feil som ikke er maskinvarekritiske. Når dette er sagt, så kan logging riktignok inkludere temperaturvarsling, feilmeldinger fra harddisk som ikke har kritiske feil og andre "intelligente" komponenters meldinger, men om et strømbrudd eller andre hendelser sørger for at en maskin ikke blir tilgjengelig, mister den også loggevne. Dette kompenseres i enkelte tilfeller gjennom andre komponenters logg som angir feil ved tilkobling til den aktuelle enheten, men ikke ved direkte logging.

3.3 Hva kan en logg inneholde?

Som oppgaven kommer tilbake til i 3.4 finnes det flere ulike typer loggsystemer, men felles for disse er at de registrerer hendelser på et gitt system i et forhåndsbestemt format. Hvilke hendelser som skal registreres vil som nevnt variere basert på hva man ønsker å få ut av loggingen gjennom konfigurasjonen, men dette er også begrenset av hva applikasjonene/komponentene er i stand til å logge. For å konkretisere inneholder listen under noen eksempler på hendelser det er ønskelig å bevare gjennom logger på KVT:

- Suksessfull innlogging av spesifikk bruker
- Mislykket innlogging
- Feil på planlagt operasjon/oppgave (for eksempel daglig sikkerhetskopiering)
- Anvendelse av administrator/rotbruker for å utføre kommandoer
- Godkjenning/avvisning av trådløs tilkobling mot RADIUS-server[5]
- Manglende kontakt med deler eller hele nettverket
- Restart av servere/enheter
- Maskinvarefeil (som det er mulig å varsle om)

Det finnes, i tillegg til de ovennevnte, et vell av mulige hendelser som det kan være interessant å ha med i et loggsystem, avhengig av hva man ønsker å se

[5]RADIUS - Remote Authentication Dial In User Service, anvendes for å autentisere tilgang til et eller flere systemer.

på, hvor i nettverket enheten er plassert og hvilken oppgave den har. Hos en RADIUS-server som står for autentisering av maskiner og personer som skal ha tilgang til nettverket vil det sannsynligvis være nyttig å logge store deler av aktiviteten knyttet til forsøkene på tilkobling, mens hos en printserver på et lokalt subnett[6] vil det ofte kunne være tilstrekkelig å logge hvem som skriver ut når, og om det ble suksessfullt eller ikke.

Elementene i listen ovenfor kan ytterligere kategoriseres, i generelt nyttig informasjon som kan være interessant å se på som del av den ordinære oppfølgingen av aktiviteten, og kritisk informasjon som må undersøkes nærmere umiddelbart. Bruk av ordet "anomalier" senere i oppgaven går på tilfeller av denne kritiske informasjonen, selv om ordet i utgangspunktet er definert noe bredere.

Følgende linjer er eksempel på hvordan loggoppføringer i syslogformat ser ut i loggfilene:

```
daemon.info 2009-03-27T15:13:03+01:00 ub-dhcp dhcpd: \
DHCPACK to 158.38.101.103 (00:12:3f:1e:1e:6d) via eth0

daemon.info 2009-03-27T15:13:08+01:00 dhcppers dhcpd: \
DHCPREQUEST for 158.38.14.54 from 00:15:c5:06:87:7d  \
(larerutlan01) via eth0
```

Linjen under er et eksempel på hvordan en Windows-event kan se ut når den blir sendt til syslog:

```
local7.notice 2009-03-27T15:12:45+01:00 158.38.14.20 Security:
538: VISMA\ellen-margrethe: User Logoff: User Name: ellen-
margrethe Domain: VISMA Logon ID: (0x0,0xC0DE4D6) Logon Type: 3
```

Rent teknisk så er det et element som er felles og viktig for alle loggformater; nemlig tid[1, Kapittel 9.1]. For at en hendelse skal kunne gi informasjon og kontekst for å finne feil, samt forhåpentligvis mulighet for å anvende informasjonen til å rette opp feilen, må tidspunktet hvor hendelsen inntraff være

[6]Med Lokalt subnett menes her et begrenset sett med komponenter som er skilt fra resten av nettverket med en ruter eller liknende, og ikke inneholder vesentlige serverfunksjoner delt med resten av nettverket.

med som en del av hendelsesoppføringen. Det kan dog i spesielle applikasjoner være spesielle måter å anvende tidsbegrepet på, for eksempel kan en printserver anvende utskriftsjobbnummer som tidsreferanse. I de aller fleste tilfeller er dog også tiden, slik den normalt fremstår, med i oppføringen.

Dette er viktig av flere årsaker:

- En systeminnlogging på webserveren kan være naturlig om hendelsen inntreffer i løpet av arbeidstiden, eller i et vedlikeholdsvindu[7], mens kan antyde et sikkerhetsbrudd om den eksempelvis inntreffer mellom klokken 24.00 og 06.00 av en bruker som ikke burde ha grunn til å gjøre arbeid på det tidspunkt.

- Om man skal korrelere flere logger er det også her avgjørende at det eksisterer en korrekt tid som er felles for de ulike loggsystemene. Om dette ikke er tilfelle, kan man og vil man gå glipp av at hendelser fra ulike kilder er tidsmessig sammenfallende, hvilket vil øke kompleksiteten i feilsøkingen, dersom de skulle høre sammen årsaksmessig.

Mengden loggdata fra enhetene kan konfigureres gjennom å velge hvilke typer loggmeldinger det er ønskelig å ta med[8]. Det må her gjøres vurderinger på hvilken mengde data man ønsker eller har behov for, med mål om å finne en balanse mellom mengden av loggdata og hvor kritisk enheten er i systemet. Om loggmengden blir stor kan det det vanskelig å finne de loggmeldingene man søker etter når man utfører et manuelt feilsøk. Om man derimot kun inkluderer meldinger som man på forhånd definerer som kritiske, kan man risikere å fjerne litt for mye, og stå igjen uten feilsøkingsverktøy. Det må altså gjøres bevisste og gjennomtenkte valg på hvilke loggmeldinger man ønsker å ha med, med mindre man har et effektivt analyseverktøy som tåler belastningen.

[7]Et vedlikeholdsvindu er et på forhånd planlagt tidsintervall hvor vedlikehold av enheter kan foretas, selv om det også vil kunne medføre nedetid for systemet eller enheten.

[8]Se tabell med facilities som syslog anvender under punkt 3.4.2

3.4 Ulike loggsystemer:

Det finnes et hav av ulike loggsystemer[9], både systemer som er lagd spesifikt for windows eller *nix og systemer som tar sikte på å takle begge deler. Årsaken til at de 3 under er valgt ut er først og fremst på grunn av utbredelsen og brukermassen, men også at det kan være nyttig å skille mellom loggsystemer og logganalyseverktøy. Dette skillet gir fleksibilitet til å forholdsvis enkelt kunne skifte ut analyseverktøy, uten å gjøre inngrep i infrastrukturen ellers. Flere logganalyseverktøy kan selv motta logger over nettverket[10], men fleksibiliteten ved å ha dette skillet er kraftig vektlagt i denne oppgaven. Disse systemene er forøvrig også eksisterende på KVT fra før.

Event Viewer er standard på alle Windows-installasjoner frem til og med Windows 2003 Server (nå erstattet av Windows Event Log), mens syslog eller syslog-ng er standard på UNIX, FreeBSD, samt de fleste større linux-distribusjoner og er bredt utprøvd og dokumentert.

Systemer som har en slik utbredelse vil også bli oppdatert og fulgt opp nøye[11], hvilket er en trygghet for administratorer og brukere som anvender systemene.

Eksempler på andre loggsystemer for linux/UNIX er rsyslog som brukes av Fedora fra og med versjon 8, metalog, sysklogd og socklog. Når det gjelder Windows følger Event Viewer med som standard, og brukes i bunn, mens man kan velge å kjøre andre systemer på toppen. Disse erstatter dog ikke Windows Event Viewer, men ligger som klienter i tillegg og eksporterer data til et annet ønsket loggsystem.

[9] Med loggsystem så menes her programmene som kjører på den enkelte datamaskin eller server og som styrer og tar i mot logger fra applikasjoner lokalt på denne maskinen, i tillegg til å lagre de/sende de videre i henhold til hvordan det er konfigurert.

[10] Splunk som oppgaven tar for seg i delkapittel 4.8.2 har denne funksjonen.

[11] Når mange brukere klager på feil, er det som regel mer effektivt i forhold til utbedring enn om få brukere klager.

3.4.1 Event Viewer

Event Viewer er navnet på loggsystemet som er brukt som standard av NT-plattformen til Windows, som inkluderer Windows NT, 2000, XP og Server 2003. Med Windows Vista og Windows Server 2008 har det kommet en ny versjon med benevning Windows Event Log, men det er fremdeles det samme systemet som ligger i bunn, selv om det har blitt gjenstand for en videreutvikling og med tilhørende ny navngiving. Grunnen til at "Event Viewer" er valgt som tittel her, er at oppdragsgiver ikke anvender noen servere med operativsystemet Windows Server 2008, og at det dermed ikke har relevans for valgene som blir gjort senere i oppgaven. Datamaskiner med Windows Vista er heller ikke aktuelle å innhente loggdata fra på dette tidspunkt.

Event Viewer har vært gjenstand for en stadig oppgradering parallelt med utviklingen av nye operativsystemer, men hovedelementene har siden Windows 2000 vært stabile. Loggstrukturen har følgende oppbygging:

- System
- Application
- Security
- Egendefinerte seksjoner

Security er den viktigste enkeltkomponenten av disse, ettersom den inneholder hendelser som berører pålogging, tilgang til objekter, konto-operasjoner, prosesser med mer. Denne kan kun skrives til av Local Security Authentication Subsystem Service (lsass.exe), mens System er brukt av selve operativsystemet og Application av nettopp applikasjonene som er i bruk på den aktuelle maskinen.

På Windows-plattformen medfører tilgang til å lese logger også tilgang til å slette logger, hvilket ikke er en ideell situasjon, da man generelt ønsker å på en sikker måte ta vare på loggene[5, kapittel 7], i alle fall frem til de er analysert. Denne funksjonaliteten er dog begrenset til brukere med Administratorrettigheter, men bruk av en ekstern loggserver til å ta i mot og ta vare på loggdataene blir her en veldig aktuell problemstilling. En annen ting man må

merke seg er at Event Viewer har en standard mengdebegrensning på 512KB for hver av System, Application og Security-loggene[5, kapittel 7], hvorpå overskytende loggmengde enten overskriver gamle hendelser, eller ikke blir lagt inn. Dette er ikke heldig, og nok et argument for å sende hendelser til en ekstern loggserver som har rutiner for trygg lagring. Det er dog muligheter for å øke denne maksimale loggmengden også med Event Viewer, men dette fjerner ikke fundamentet for de andre argumentene. Bruk av en loggserver er dermed uansett en god idé.

Windows har ingen innebygd funksjon for å sende logghendelser til en ekstern server, så en tredjepartprogramvare må installeres for å muliggjøre dette. Dette er nærmere beskrevet i kapittel 5.3.

Event Viewer har mulighet for å kobles til Task Scheduler[12] hvor bestemte hendelser setter i gang bestemte oppgaver, men da dette kommer under monitorering, er drøfting av dette utenfor denne oppgavens omfang.

3.4.2 syslog

syslog er loggsystemet som tradisjonelt har vært standard på UNIX, FreeBSD og de fleste Linux-avarter etter at det ble utviklet av Eric Allman på 80-tallet[13]. Det er bygd som en klient/server-protokoll, hvor klienten sender hendelsen som skal registreres til serveren, kalt syslog daemon eller syslogd[13].

Som standard ligger klienten og serveren på samme maskin eller enhet, men konfigurasjon av syslogd (serverkomponenten) tillater at man kan sette opp logging til en annen maskin eller enhet ved hjelp av noen få forholdsvis enkle steg.

Det må her anføres at protokollen som da eventuelt brukes til ekstern overføring av loggdata er User Datagram Protocol (UDP), hvilket ikke er ideelt med tanke på sikkerhet. I motsetning til Transmission Control Protocol (TCP),

[12]Task scheduler sørger for å kjøre spesielle programmer eller script på bestemte tider, eller ved bestemte triggere

[13]Eric Allman laget syslog som en del av sitt arbeid med å utvikle sendmail for UC Berkeley. For detaljert informasjon om syslog, se rfc3164[12]

som sørger for sikker[14] levering eller feilmelding, har UDP ingen slike kontrollelementer. Dette gjør at selve protokollen krever mindre "overhead" for å sende meldinger, men man kan altså ikke vite om alle meldinger er kommet frem til mottaker, hvilket er uheldig i loggsammenheng. Det er naturlig nok ikke mulig å vite hvorvidt en melding som ikke kommer frem er viktig eller ikke.

Hendelser som lagres av syslog lagres med et tidsstempel, facility (typenavn), level (alvorlighetsgrad), avsender samt selve beskjeden/meldingen. Facility må være et av 14 forhåndsbestemte alternativ, hvor noen er reserverte, og andre kan tilpasses spesielle behov. Level er delt opp i deler fra den mest kritiske; "emerg", til den minst kritiske: "none".

Tabell 1 og 2 viser en oversikt over henholdsvis facilities og levels. Behandlingen av hendelser kan settes opp i henhold til hvilken facility og level den enkelte hendelse har. Dette gjelder både hvordan de skal lagres, hvem som da skal ha tilgang til å se loggene, og om andre oppgaver skal settes i gang som en respons til hendelsen.

syslog har fått stor utbredelse også blant lukkede enheter, som rutere, brannmurer og andre nettverkskomponenter, hvilket gjør at man forholdsvis enkelt kan samle loggene på et eller flere sentrale punkter. Årsaken til at syslog der er valgt som loggsystem, kan være at de ofte anvender et operativsystem med Linux-kjerne, hvor syslog er lett tilgjengelig for implementasjon.

syslog er i utgangspunktet ikke kompatibel med Event Viewer, men det finnes tredjepartsprogramvare som kan ligge som en tjeneste på en Windows-maskin, og derfra hente inn loggene, oversette de til syslogformat og sende de til en maskin som er konfigurert til å ta i mot syslog-oppføringer fra eksterne kilder.

[14]Med sikker menes her trygghet for at dataene ikke forsvinner ubemerket på veien, og ikke sikkerhet som i kryptering.

Tabell 1: syslog Facilities Messages

Message	Description
LOG_AUTH	Security and authorization messages (DEPRECATED).
LOG_AUTHPRIV	Security and authorization messages (private).
LOG_CRON	Clock daemon (cron and at).
LOG_DAEMON	System daemons without separate facility values.
LOG_FTP	Ftp daemon.
LOG_KERN	Kernel messages.
LOG_LOCAL0-7	Reserved for local use.
LOG_LPR	Line printer subsystem.
LOG_MAIL	Mail subsystem.
LOG_NEWS	USENET news subsystem.
LOG_SYSLOG	Messages generated internally by syslogd.
LOG_USER (default)	Generic user-level messages.
LOG_UUCP	UUCP subsystem.

3.4.3 syslog-ng

syslog-ng (syslog next generation) bygger videre på den opprinnelige syslog, men inneholder en del utvidelser som gjør det i stand til å være et bedre loggsystem for dagens sammensatte systemer[7].

syslog-ng anvender også i utgangspunktet syslog-formatet, men er konfigurerbart i større grad enn den opprinnelige syslog. Om man samler loggdata fra flere kilder, tar syslog-ng vare på navnet på serveren som loggoppføringen opprinnelig kom fra, i motsetning til syslog som kun registrerer servernavnet den mottok meldingen fra. Denne forskjellen kan virke subtil, men om man har et større/seksjonert nettverk, kan nøsting av loggservere være aktuelt og gi behov for denne funksjonen.

syslog-ng støtter lagring av loggmeldinger direkte inn i en katalogstruktur, hvor katalogene kan være dato, servernavn eller tilsvarende, og endres dynamisk uten at administratoren trenger intervenere. Dette kan gjøre log-

Tabell 2: syslog Priority Levels

MESSAGE	Description
LOG_EMERG	System is unusable.
LOG_ALERT	Take action immediately
LOG_CRIT	Critical conditions have occurred
LOG_ERR	Error conditions.
LOG_WARNING	Warning conditions.
LOG_NOTICE	Normal but significant conditions that warrant attention.
LOG_INFO	Informational message.
LOG_DEBUG	Debug-level message.
LOG_NONE	N/A

grotasjon overflødig[15], og samtidig også gjøre unna arkivering. Et eksempel på en mulig filplassering med bruk av syslog-ng på brannmurlogger er: */var/log/Firewall/2009/03/22.log*. Hver måned genereres nye mapper, og loggdataene fra hvert døgn legges rett inn i egne filer.

Den kanskje viktigste enkeltforskjellen mellom syslog-ng og syslog, er at syslog-ng støtter bruk av TCP til mottak og sending av loggmeldinger over nettverk. Dette innebærer større trygghet for at loggmeldinger kommer frem til mottaker, og har som konsekvens en mer pålitelig loggmengde.

3.5 Samhandling og samling av loggdata

For å få en best mulig logganalyse, som oppgaven skal komme tilbake til i kapittel 4, må alt av loggdata være tilgjengelig på en plass og på samme format. Dette innebærer at man må ta i bruk en dedikert loggserver, som tar i mot loggdata fra alle komponenter som skal være inkludert i logganalysen.

Dette har flere fordeler:

- Administratorene forholder seg til kun en rapport per tidsperiode som dekker loggdata fra alle enhetene.

[15]Dette er avhengig av hvilken konfigurasjon man setter opp. Det er ikke noe i veien for å anvende samme filplassering og loggrotasjon som med tradisjonell syslog.

- Det innebærer en sikrere oppbevaring av loggdata. Et sikkerhetsbrudd på en server innebærer ikke at loggene kan endres eller fjernes[3, sd. 710].
- Konfigureringen av analysen kan skje på en plass, og gjelde for alle enhetene. Dette minsker muligheten for feil.
- Lagring og arkivering av logger blir enklere, når det kun er en server å forholde seg til.
- Korrelering av loggdata fra ulike kilder blir direkte tilgjengelig, uten behov for filkopiering i mellom servere.

For å få fullt utbytte av disse momentene kreves det dog at den dedikerte serveren blir beskyttet i større grad enn hva tilfellet ellers kanskje ville vært[7]. Et eventuelt sikkerhetsbrudd på denne serveren vil kunne føre til at alt av loggdata man har blir kompromittert, hvilket vil være et uheldig og svært uønsket scenario. Bruk av chroot[16] for å kjøre loggsystemet, samt krav om fysisk tilgang til serveren for innlogging kan her være nødvendig for å ha et sikkert system.

Et siste moment som er nødvendig for å få full effekt av å anvende en sentral loggserver, er å sikre seg at alle enhetene som anvender systemet har samme oppfattelse av riktig tid. Hvorvidt man anvender en ntp-server til dette eller finner andre løsninger er underordnet det kriteriet at en løsning eksisterer. Om en felles tid er korrekt implementert vil tiden i loggoppføringen kunne antas å være riktig, og dermed kunne brukes som en parameter i logganalysen.

3.6 Personvern

Datatilsynet gir omfattende retningslinjer om hvordan logger skal anvendes, hvilket også er styrende for hvordan vi teknisk går frem for å samle våre logger. Det tilsynet først og fremst påpeker som viktig, er at det gjennom bruk av logger ikke skal være mulig å spore hva den enkelte person gjør til

[16] chroot kan blant annet begrense hva loggsystemet kan utføre, samt hvilke kommandoer som kan kalles derfra.

en hver tid, da dette anses som overvåkning av den enkelte. Dette er med unntak av informasjon om hvem som logger på, eller forsøker å logge på, systemet og til hvilken tid dette skjer[9].

Dette kommer tildels i konflikt med formålet med logging og logganalyse, hvor hensikten først er å oppdage anomalier for deretter å sette inn nødvendige tiltak. Dette innebærer at en er avhengig av å ha så komplett informasjon som mulig, slik at analysen kan gi "riktige" resultater.

Det som dog kan gjøres, slik at begge behov blir dekket, er en viss anonymisering av logger. Man kan her se for seg at man maskerer IP-adressene på steder hvor IP-adressene ikke er kritiske. Dette er lite formålstjenlig på loggdata fra en *nix-daemon som sshd[17], hvor IP-adressen kan være avgjørende for å kunne vite hvorvidt hendelsen er ok eller ikke, men kan være uproblematisk å utføre på loggdata som kommer fra en DNS-server. Det vil sjelden være kritisk for oppetiden til systemet å vite hvem som har vært inne på hvilke nettsider til hvilken tid.

Datatilsynets retningslinjer med tanke på personvern setter standarder som gjør at en til en hver tid må være bevisst på hvorvidt identifiserende informasjon forekommer i loggene, og ta grep under lagring for å unngå muligheten for overvåkning. Om det i enkelte spesialtilfeller vurderes til at lagring av slik informasjon må forekomme, må dette gjøres kjent blant de som kan være omfattet av det. Dersom anonymisering ikke blir utført, vil krav om sletting av informasjon, som igjen betyr sletting av logger, komme i konflikt med ønsket om et godt grunnlag for logganalyse.

En gylden middelvei kan være om man etter et visst tidspunkt anonymiserer loggene som nevnt over. Mer konkret kan man da tenke seg at man daglig kjører en rensing av logger på en slik måte at de i etterkant ikke kan anvendes til overvåkning. En slik variant bør sannsynligvis oversendes Datatilsynet for godkjenning, for å være sikker på at ingen lover brytes.

[17] sshd er prosessen som styrer kryptert fjernpålogging, i dette tilfellet påloginger til servere.

4 Logganalyse hos KVT

Etter at logging har blitt satt opp på alle datamaskiner/komponenter som det er ønskelig å inkludere for analyse, kommer et punkt hvor man skal hente ut nyttig og god informasjon fra dataene. I utgangspunktet ligger det da kun en mengde med data i loggfilene, og disse må behandles for å kunne gi merverdi i forhold til før-status hvor administratorene manuelt sjekket hver logg. Dette skjedde typisk etter at en situasjon hadde oppstått, som en reaksjon på en feilmelding.

Dette kapitlet tar sikte på å vise hva logganalyse er, hvorfor og hvordan det kan brukes, samt spesielle momenter det må tas hensyn til underveis.

4.1 Formål med logganalyse

For å kunne ta gode beslutninger om hvordan logger skal analyseres, er det nødvendig å vurdere hva formålet med analysen er. Abe Singer i 'Building a Logging Infrastructure' peker på 3 ulike alternativer som, avhengig av hvilke(n) som blir prioriterert, har påvirkning på hvordan veien går videre[5]:

Hvorfor er det ønskelig med bedre logganalyse?

1. Forbedret sikkerhet
   Om dette er det viktigste punktet, medfører det et hovedfokus på angrep, kontroll av loggintegritet og restriksjoner på servertilgang- og synlighet.

2. Effektivisere drift og administrasjon
   Dette punktet setter fokus på automatisering av analyse samt gode rapporteringsmuligheter

3. Pålegg og krav

Disse punktene er ikke gjensidig utelukkende, men setter betingelser for hva som er hovedfokus. I tilfellet KVT og denne oppgaven er punkt 2 hovedargumentet[18] for at behandling av logging på KVT skal gjennomgå en revisjon.

[18] se Kravspesifikasjon

Dette får konsekvenser for noen av valgene som blir tatt, men blir kommentert underveis.

Ved å få en mest mulig automatisert logganalyse vil målet om en mer effektiv hverdag for administratorene være mer oppnåelig. Prinsipielt er også lesing av logger manuell logganalyse, men det vil være ineffektivt ettersom mengden data som man *ikke* ser etter kan være uforholdsmessig stor.

Det vil dog i den nærmeste fremtid ikke kunne bli en realitet med en perfekt helautomatisk logganalyse[4] i et heterogent nettverk, som det her er snakk om. Dette kommer blant annet av at nye feil kan oppstå, nye komponenter kobles til, og signaturer til kjente komponenter kan endres ved oppgradering til nye versjoner. En manuell kontroll av resultatene fra logganalysen vil bli en del av arbeidsoppgavene til administratorene fremover, selv om dette vil gi bedre oversikt enn hva tilfellet er i dag.

4.2 Teknologivalg

Som delkapittel 3.5 omtaler, så vil en sentral loggserver stå sentralt for på en effektiv måte å kunne analysere loggdata. Med loggdataene på et felles format, lagret i en felles på forhånd planlagt struktur, blir det mulighet til å sammenholde og analysere data på en effektiv måte. En implementasjon av dette krever dog at man må ta noen strategiske valg, hva teknologi angår.

Disse punktene må besluttes før man går i gang med implementasjonen:

- Loggformat/Loggsystem
- Tidshåndtering

Valg av logganalyseverktøy blir ikke kommentert her, men når den ovennevnte infrastrukturen kommer på plass vil det i utgangspunktet være uproblematisk å implementere ulike logganalyseverktøy på toppen. Dataene mottas på samme format, på angitt plass, klare til behandling. Dette forutsetter at logganalyseverktøyene har støtte for valgene som her blir tatt, og blir nærmere kommentert under 4.8.

Loggsystem

Ettersom Event Viewer ikke har innebygd støtte for verken å sende eller motta loggdata, som både syslog og syslog-ng har, er det ikke mulig å anvende det på loggserveren uten at man introduserer nye verktøy. Med kravspesifikasjonens ønske om å minimere kostnad ved innføring av dette nye systemet, kan bruk av Windows på loggserveren utelukkes, da det finnes alternativer.

Likhetene og ulikhetene mellom syslog og syslog-ng er beskrevet nærmere i 3.4, hvor syslog-ng fremstår som noe tryggere og mer fleksibelt, mens syslog er og har vært en meget utbredt og gjennomprøvd standard over lengre tid på UNIX, FreeBSD og de fleste Linux-distribusjoner.

Ettersom syslog-ng "arver" fordelene som syslog har med tanke på kompabilitet, i tillegg til forbedringer med tanke på sikkerhet og administrasjon, blir syslog-ng et naturlig valg i løsningen av denne oppgaven. Det resulterer i mulighet for å angi tidsformat som inkluderer år, sortering av filer og kataloger ut i fra hvilken server som er original avsender, i tillegg til støtte for trygg levering.

Tidshåndtering

Som beskrevet i 3.5 er innføringen av en felles tid essensiell for å kunne utnytte loggdataene på en best mulig måte. Alle komponenter som skal logge til loggserveren anvender *ntp.kvt.vgs.no* som tidstjener, med *ntp.ubuntu.com* som sekundær tjener hvis det er mulig å definere flere.

ntp.kvt.vgs.no henter selv tiden fra *ntp.ubuntu.com*, så et utfall av den lokale tidsserveren vil ikke føre til et umiddelbart hopp i loggtidspunktene.

4.3 Logghåndtering og sikkerhet

For å kunne stole på loggdata må man ha rutiner og oppsett som fjerner/minsker risikoen for at uautoriserte personer får tilgang til loggene (Se punkt 3.5.). Dette kan gjøres på servere ved hjelp av tilgangspolitikk, både med tanke på tidsrom man kan logge på, krav til passordkvalitet samt begrense tilgang til spesielle ip-adresser, eller på enkleste måte ved kun å tillate lokal

pålogging. Utilsiktet tilgang til servere som sender logger til loggserveren vil også kunne sørge for loggmeldinger som ikke har bakgrunn i reelle hendelser på den enkelte server. Dette medfører ikke risiko for fjerning eller endring av loggdata som allerede ligger på loggserveren, men kan redusere kvaliteten på loggene som kommer fra den aktuelle serveren i perioden etterpå.

Implementasjon av en streng tilgangspolitikk vil ikke alltid kunne gjennomføres i like stor grad på printere og andre nettverkskomponenter[19], men da må man ta hensyn til dette når man trekker konklusjoner fra logganalysen. Der man har mulighet til å kryptere tilkobling og stille krav til passord må man gjøre det for å kunne ha mest mulig tillit til loggdataene.

Et annet moment, som også er relevant for validering av resultater fra logganalyse, er at loggoppføringer, spesielt med tanke på facility og level, er satt av applikasjonsutviklerne. Behovene de begrunnet sine valg med i utviklingen trenger ikke å tilsvare behovene som eksisterer på dette nettverket. Tilsynelatende er dette ikke noe stort problem, da enighet om hva som karakteriseres som kritisk ser ut til å være nådd. Når nye tjenester og programmer tas i bruk må allikevel dette undersøkes, slik at det ikke resulterer i unødig "støy" i loggene.

4.4 Logganalyse vs. monitorering

Monitorering går utenfor målet til denne oppgaven. For presisjonen i oppgaven er det allikevel viktig å trekke noen skillelinjer i forhold til hvor monitorering slutter og logganalyse tar over.

Monitorering er et sanntidssystem som på en eller annen måte implementerer triggere som utfører forhåndsbestemte funksjoner om en enkelt hendelse, eller en bestemt sekvens av hendelser inntreffer. Generelt kan dette beskrives på denne måten: Om en forhåndsdefinert hendelse skjer, så skal en reaksjon forekomme automatisk[1, Kapittel 10]. Hvorvidt denne reaksjonen medfører en restart av en tjeneste, nekt av fremtidig brukertilgang eller kun en beskjed

[19] I et større nettverk kan det være naturlig at andre enn administratorene har brukernavn og passord til for eksempel skriverne for å utføre vedlikehold.

til ansvarlig personell vil være avhengig av den enkelte hendelsen og den tilhørende konfigurasjonen. Et eksempel kan være at en epost automatisk blir sendt til administrator om det kommer en melding om at en disk er gått full.

Logganalyse er ikke et sanntidssystem, selv om hendelsene loggføres i sanntid[20]. Her er det i utgangspunktet bestemte tidspunkt hvor rapporter genereres, på bakgrunn av alle hendelser i et gitt tidsrom. Hendelsene som medfører reaksjoner i forbindelse med monitorering vil også kunne dukke opp som prioriterte hendelser i en logganalyse, men dette er avhengig av hvilke parametre som er satt i konfigureringen av logganalyseverktøyet. Flere logghendelser er vesentlige kun på bakgrunn av sin kontekst, hvilket monitorering ikke har mulighet til å fange opp. Kapittel 4.5 tar videre for seg behovet for logganalyse.

Det må her ikke misforstås slik at monitorering på noen måte kan erstatte logganalyse og oppbevaring av logger. Monitorering kan ikke ta for seg alle mulige hendelser som kan inntreffe, da må man i tilfelle ha et svært begrenset system med kun et sett med angitte mulige hendelser, som ikke er aktuelt i et nettverk som KVT har. Overdreven bruk av monitorering som resulterer i meldinger til administratorer kan også føre til at man lettere kan overse de viktige meldingene, blant alle som egentlig ikke krever umiddelbart tilsyn (Om det sendes en melding om at gul toner begynner å bli lav hver gang en utskrift sendes til en skriver, vil det ikke være til hjelp for administratorene.).

4.5 Teknisk gjennomgang av elementer i logganalyse

I et sett med logger finnes det som nevnt tidligere en stor variasjon i de ulike hendelsene som er loggført. En god analyse av et sett med loggdata klarer å hente ut og prioritere de viktigste hendelsene, mens rutinemessige eller mindre vesentlige hendelser får mindre prioritet. I tillegg er det også normalt at et sammendrag av aktiviteten i det gitte tidsrommet er inkludert.

[20]Fjernlogging fra Windows kan ha en forsinkelse før en serie med logghendelser overføres, avhengig av hvilken agent som anvendes for eksportering av loggdata.

4.5.1 Baseline

Her melder det seg dog umiddelbart utfordringer. Når blir en hendelse, eller en serie med hendelser vesentlig? For å kunne gi et svar på dette kan man blant annet inkludere baseline som en parameter. En baseline kan forklares som en normaltilstand, og i loggsammenheng kan det være de hendelser som normalt inntreffer ved et komplett fungerende system i et gitt tidsrom[5, kapittel 6].

Hva er så motivasjonen for å logge rutinemessige, tilsynelatende uvesentlige data om man ikke ønsker å se på de?

Svaret på dette spørsmålet er at det ikke er gitt på på forhånd om de er uvesentlige eller ikke. Om en baseline sier at det på en normal ukedag forekommer 1000 suksessfulle innlogginger, vil det kunne være relevant om man en dag har 50000 innlogginger. Disse hendelsene vil ikke i seg selv alltid skille seg fra hverandre, men det store antallet kan for eksempel tyde på at en server ikke får svar ved innlogging til en tjeneste, og dermed prøver gjentatte ganger. Dette vil kunne være vanskelig å finne manuelt, selv om man ser på loggstørrelsen at et eller annet er i emning.

På en videregående skole vil det i tillegg være høyst relevant hvilken dag man analyserer data fra. Lærerne og elevene har ulike planer dag for dag, men som gjentas uke for uke, noe som påvirker loggmengden fra dag til dag. Om halvparten av elevene skulle starte klokken 12.00 på mandag, men 08.00 på fredag, vil dette ha betydning for loggdataene som genereres. Når loggdata fra en mandag blir analysert, er det mer relevant å se på hva som har foregått på tidligere mandager enn statistikk for ukedager sett under ett.

4.5.2 Alvorlighetsgrad oppgitt i loggoppføringen

Dette er det enkleste punktet å implementere uansett type verktøy man anvender for logganalyse. Hver enkelt oppføring har som nevnt i 3.4.2 en definert alvorlighetsgrad, og man kan/bør sette logganalyseverktøyet til å inkludere alle oppføringer som har en level større enn en satt grense. Oppføringene kan

i tillegg plasseres i filer etter hvilken facility som er angitt, og fører til at det blir mulig å sortere det videre. En oversikt over disse ligger i tabell 1 og 2.

Dette fører for eksempel til at alle oppføringer som har alvorlighetsgraden EMERG alltid vil komme med i loggrapporten. For hver ny grad man inkluderer, risikerer man å inkludere falske positiver, men i et enkelt nettverk kan det likevel fungere med flere enn de mest alvorligste. Dette må rett og slett testes ut, for å se hvilke oppføringer som eksisterer, hvilken alvorlighetsgrad de har og i hvilken grad disse er å oppfatte som kritiske.

4.5.3 Maskering og statistikk

Logghendelser kan ha ulikt tekstlig innhold, selv om det i analysesammenheng er samsvarende hendelser. Innlogginger over vpn vil for eksempel ha ulike ipadresser for påloggingssted (altså hvor brukeren er tilknyttet internett når den logger på), men dette trenger ikke å være relevant med mindre enkeltadresser er knyttet til spesielle triggere. En metode som kan anvendes for å kunne samle slike hendelser mest mulig hensiktsmessig er å maskere 'felter' som inneholder for eksempel ipadresser og brukernavn.

Ved en slik maskering vil man kunne anvende statistikk sammenholdt med baseline, da man på enkelt vis kan finne antall av de ulike hendelsene. Dette kan igjen sette i gang flere analyse-teknikker, dersom det viser seg at antallet av en bestemt hendelse avviker vesentlig fra antallet på en ordinær dag (baseline). Om dette er tilfellet vil den neste gjennomgangen av den samme loggsekvensen kunne maskere kun brukernavn, og hente ut informasjon på ipadresser. Den nøyaktige responsen vil kunne variere med oppsettet av hvert enkelt system.

Det må her bemerkes at grunndataene ikke endres, selv om en permanent implementasjon vil måtte justeres til å ta hensyn til personvern i henhold til betraktningene i 3.6.

4.5.4 Korrelering

Korrelering går i loggsammenheng ut på å se sammenhenger mellom hendelser fra samme loggkilde som opptrer som en sekvens [21], og mellom hendelser som opptrer på ulike loggkilder, men er omtrent sammenfallende i tid. Mer konkret kan en enkelthendelse, som en feilmelding ved sending av epost, være av mindre betydning, dersom mailserveren kjører en jobb som gjør at den er utilgjengelig akkurat i det øyeblikket den blir forsøkt kontaktet.

Om det derimot viser seg i loggfilene at mange ulike servere som forsøker å sende epost mellom klokken 09.00 og 10.00 blir avvist, blir den korrelerte informasjonen mer vesentlig for administratorene å forholde seg til. Dette kan være vanskelig å plukke opp, om loggene ikke er samlet sentralt. (Forutsetter her at man ikke har direkte tilgang til epostserveren, som Uninett kontrollerer for KVT).

Korrelering er ikke begrenset til analysering av slike aktive tjenester, det er også mulig å se for seg at feil ved nettverkskomponenter kan oppdages. Dersom flere servere på samme subnett plutselig blir utilgjengelige kan det tyde på at en router er ute av drift.

4.5.5 Spesifikke tilpasninger og triggere

Det finnes ikke *en* uniform, standardisert måte som nettverk er bygd opp på, hvilket gjør ethvert forsøk på "one-size-fits-all" komplisert. Hver enkelt avdeling/enhet som skal sette opp et nettverk har ulike behov, og dermed ulike komponenter, og ulik sammensetning av komponentene innad. Dette fører til at logganalyse blir mer komplisert å sette opp, da man enten må tweake alle lokale tilpasninger, slik at de blir vektet riktig, eller at man bruker en form for erfaringslæring. Dette kan skje ved hjelp av en form for tilbakemelding fra administratorene, for på sikt å bedre logganalysen.

Et eksempel på hvor komplisert en slik tilpasning kan være, er et som Jon Stearley beskriver i "Towards Informatic Analysis of Syslogs". Artikkelen om-

[21] Den opprinnelige hendelsen resulterer i flere etterfølgende loggmeldinger.

taler blant annet et cluster med Linux-servere hvor administratorene har laget 1221 håndskrevne regler[6] til et regelbasert loggmonitoreringsverktøy (logsurfer). En slik tilpasning vil ikke være aktuelt i forhold til denne oppgaven, og valg av logganalyseverktøy tar hensyn til dette.

4.6 Nåstatus hos oppdragsgiver

Før arbeidet med denne oppgaven tiltok eksisterte det ikke noen form for automatisert logganalyse på KVT. Etter en hendelse inntraff ble utvalgte logger gått gjennom manuelt, for å finne feil eller anomalier. Dette fungerte, ut i fra hva de IT-ansvarlige kunne fortelle, men var en ren reaktiv oppgave når en feil som berørte brukere eller administratorer allerede hadde oppstått.

De ulike loggsystemene som da måtte sjekkes var syslog på *nix-baserte servere og nettverkskomponenter, og Event Viewer/Windows Event Log) på Windows-baserte servere. Loggene ble arkivert etter standardinnstillingene, det vil si rotert på uke og månedsbasis på *nix, mens nettverkskomponenter og Windows overskrev de eldste oppføringene når den definerte loggstørrelsen var oppfylt. Det var altså ingen mulighet til å gå tilbake i tid (utover en måned) for å spore hendelser eller sammenligne med dagens loggdata.

4.7 Utvalgskriterier for logganalyseverktøy

På bakgrunn av kravspesifikasjonen for oppgaven, samt nettverksarkitekturen på KVT, blir følgende punkter vesentlige når man skal vurdere hvilke løsninger som kan egne seg som del av en løsning for KVT:

* Støtte for syslog/syslog-ng.
  Verktøyet må kunne ha støtte for å lese logger som er på syslog-format, da basestasjonene ikke har mulighet til å eksportere på annet format, og de fleste serverne er *nix og anvender syslog-ng for levering av logger.
* Støtte for å lage og sende rapporter til angitte tidspunkt.
* Støtte for maskering av logger.

* Evne til å lage rapporter basert på loggstatistikk, og sammenligning av statistikk over tid, gjerne med mulighet for å angi konfidensintervall for maksimal endring som skal kunne godtas som normale variasjoner. Dette må i tilfelle være konfigurerbart slik at man kan angi dager som ikke skal være med i utvalget på grunn av testing eller andre hendelser som fremkaller store endringer i loggdataene.
* Open Source-verktøy vil være en fordel da inkludering av nye komponenter som krever tilpasninger kan utføres kostnadsfritt lokalt, om en ser bort fra tidsbruk. At det ikke tilkommer kostnad ved innkjøp vil også være en fordel.
* Kvalitet på brukergrensesnitt.
* Konfigurerbarhet i forhold til å behandle spesielle servere/komponenter på en bestemt måte.

4.8 Eksisterende teknologi

Et søk på *google.com* på "log analyzer" gir omlag 16300000 treff, hvilket antyder at det her eksisterer mange ulike verktøy og varianter. Dette innebærer ikke at alle disse treffene beskriver ulike verktøy, men materialet med informasjon er uansett voldsomt. Svært mange av analyseverktøyene som der kommer opp retter seg mot logganalyse på webservere, med tanke på hvilke sider som ble besøkt av hvem, og til hvilken tid. Dette er ikke innenfor hva denne oppgaven skal behandle, og det er behov for en annen tilnærming for å kunne sile ut reelle alternativer for analyse av loggdata i syslogformat, også fra andre kilder enn kun webservere.

Denne oppgaven tar ikke mål av seg til å gjøre en kvalitativ undersøkelse av hvilke logganalyseverktøy som "er best", da det ville medført et voldsomt volum av verktøy som skal undersøkes.

Et nytt søk på google ble utført, men denne gangen med følgende søketerm: "syslog log analyzer automatic reporting ". Dette ga omtrent 33400 treff, og leverte Sawmill som første resultat. Etter en gjennomgang av funksjonene opp

mot utvalgskriteriene ble Sawmill besluttet tatt med som et av verktøyene for implementasjon i denne oppgaven. Dette er på ingen måte en vitenskapelig tilnærming, men ettersom Sawmill tilsynelatende fyller kriteriene og kommer høyt opp i treffrangeringen til Google ble verktøyet valgt ut til å være med.

Splunk ble oppdaget etter litteratursøk, som resulterte i ulike artikler og websider med sammenligninger av logganalyseverktøy. Konteksten for disse sammenligningene var amerikanske forhold, med voldsomme loggmengder og størrelser for øvrig, så den direkte relevansen var i utgangspunktet ikke stor, satt opp mot denne oppgaven. Det viste seg dog at Splunk kom i flere versjoner, også tilpasset mindre systemer, og ettersom utvalgskriteriene ikke ekskluderte Splunk, ble det valgt som det andre verktøyet.

Det siste logganalyseverktøyet som ble valgt ut er Lire, hvilket står på listen over logganalyseverktøy som blir vedlikeholdt under *www.syslog.org*[22]. Lire ble valgt ut blant disse som det som utfylte kriteriene på en best mulig måte, og er i likhet med de andre på denne listen ikke et kommersielt verktøy. SLAPS-2[23] var også aktuelt med tanke på funksjonalitet, men krever Solaris som operativsystem på serveren og ble dermed valgt vekk.

Det eksisterer store mengder andre logganalyseverktøy, men valget i denne oppgaven falt på de tre ovenstående, da de var tilgjengelige, oppga at de oppfylte utvalgskriteriene og i tillegg var gratis å anskaffe[24].

4.8.1 Sawmill

Sawmill er et kommersielt verktøy levert av "FlowerFire", et California-basert programvarefirma, og som blir anvendt av mange større firma og institusjoner, blant andre University of York, British Telecom og Accenture[10]. Det kan installeres på de aller fleste plattformer, da kildekoden er tilgjengelig for kompilering, men ikke for editering.

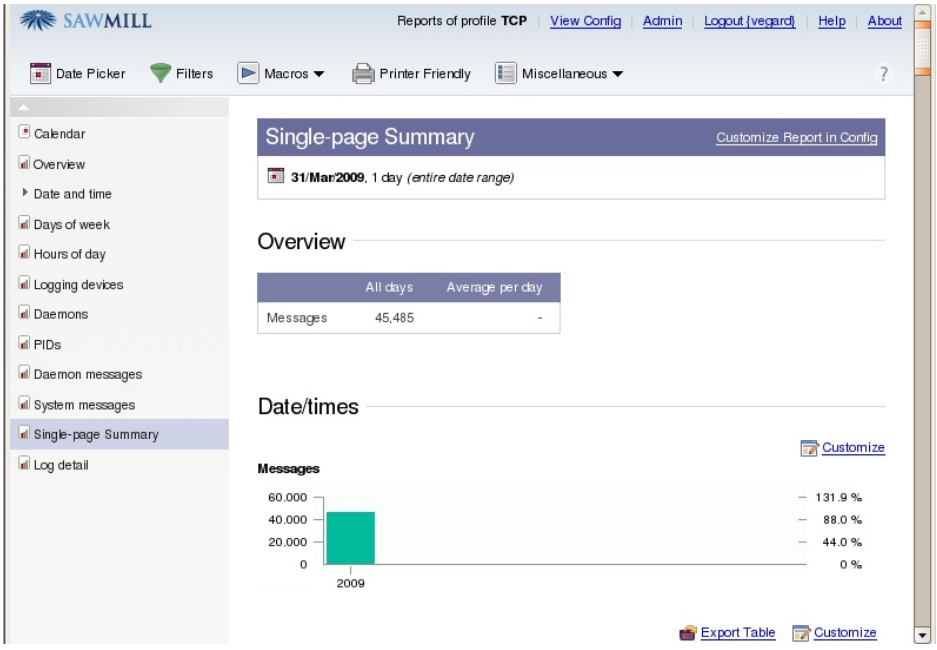
Fra et logganalyseperspektiv er Sawmill statistisk rettet, men inneholder mu-

[22]Se http://www.syslog.org/wiki/Main/LogAnalyzers for detaljer
[23]Se http://www.openchannelfoundation.org/projects/SLAPS-2
[24]Sawmill er riktignok gratis bare i de første 30 dagene.

lighet for å sette alt av filtre som man måtte ønske. Alle konfigurasjonsendringer gjøres fra et web-grensesnitt og med verktøyet følger det bred dokumentasjon. Et skjermbilde av grensesnittet er presentert i figur 1.

Sawmill støtter automatisk generering og sending av rapporter, men dette gjelder ikke den billigste utgaven (Lite), kun Professional, Advanced og Enterprise. Pris er styrt fra hvor mange rapporter man skal generere fra hvilke servere, og fra hvilke loggformater de skal genereres. For KVT ser det ut til at man innledningsvis trenger 8-10 ulike rapporter, hvilket gir en kostnad på omtrent 5000 kroner per år, ved bruk av Professional-versjonen.

Av loggformater støtter Sawmill det meste som finnes på markedet. Disse installeres som plugins i selve verktøyet, og blir umiddelbart klart til bruk. I tillegg lager de nye plugins om man har et loggformat som ikke støttes automatisk, dette uten ekstra kostnad og krever kun at man sender over loggdata på det ønskede loggformatet.



Figur 1: Illustrativt skjermbilde av grensesnittet til Sawmill

4.8.2 Splunk

Splunk[11] er et kommersielt verktøy som leveres av "Splunk", også det et California-basert programvareselskap. Av firmaer og instanser som anvender Splunk kan Cisco, Dow Jones, VISA og George Washington University nevnes, selv om listen også inneholder mange andre store selskaper.

I motsetning til Sawmill, som er statistisk rettet, tilbyr Splunk et mer søkeorientert grensesnitt. Grensesnittet er flerdelt, med en grafisk oversikt over loggmengde per intervall på en valgt loggkilde øverst, og en søkefunksjon under, som gir tilgang til søk i alle loggkilder som man har satt opp. Faste filtre og rapporter settes også opp her. Et skjermbilde er presentert i figur 2.

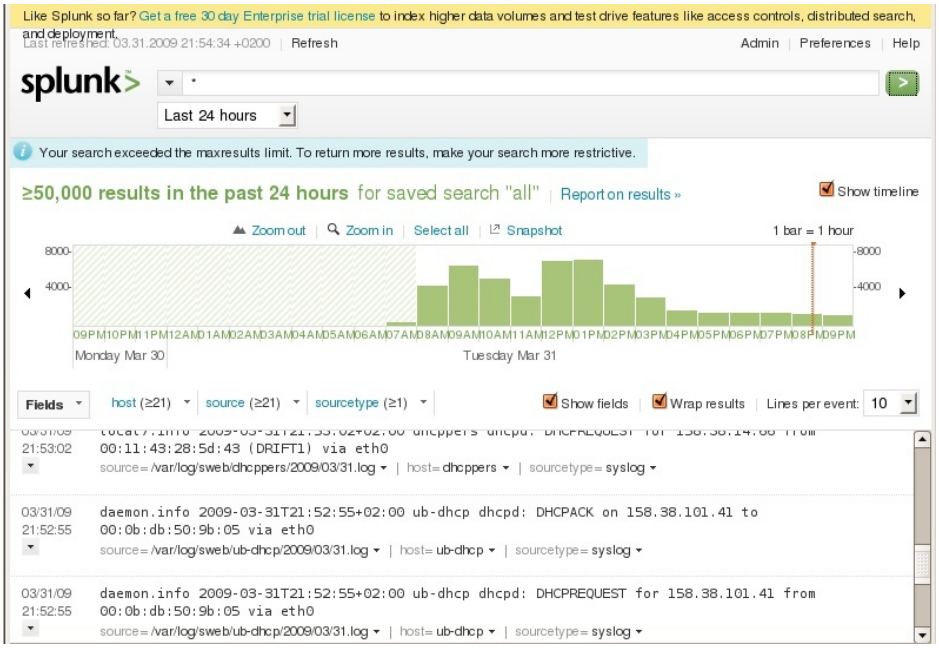
I tillegg til den rene søkefunksjonen, har Splunk en ekstrafunksjonalitet som tilsynelatende har stor nytteverdi. De har implementert direkte korrelering ved at så godt som alle elementer i de enkelte loggoppføringene er direkte søkbare ved klikk. Dette har nok størst nytteverdi ved manuelt feilsøk, men gir også fleksibilitet når man skal sette opp filtre for rapportering ved at de fleste elementer i loggoppføringene er indeksert.

Prispolitikken hos Splunk varierer med hva man trenger av funksjonalitet, for eksempel må man betale for å automatisk sammenholde data fra flere separate instanser av Splunk-servere. Om man derimot ikke har behov for den funksjonaliteten og holder seg under 500MB med rå loggdata per dag, vil versjonen som er gratis være tilstrekkelig.

I tillegg til å støtte lesing av filer som ligger på serveren (for eksempel generert av syslog-ng eller syslog), kan Splunk settes opp til å lese "pipes" på den aktuelle maskinen, samt til å med stå som ren mottaker av data fra andre maskiner eller servere, på samme måte som syslog-ng tar i mot data fra andre som sender syslog-data over nettverket. Dette er dog ikke aktuelt for denne oppgaven, og blir derfor ikke utprøvd eller vurdert.

Splunk oppgir å støtte de aller fleste loggformater[25], uten at man trenger å konfigurere dette selv. Kategorisering av loggformat er en automatisk prosess ved lesing av loggfilene/strømmene.

[25]Se nærmere beskrivelse på http://www.splunk.com/view/SP-CAAACH8



Figur 2: Illustrativt skjermbilde av grensesnittet til Splunk

4.8.3 Lire

Lire er et Open Source-verktøy som opprinnelig ble utviklet gjennom "Log-Report Foundation", men som etter 2004 har hatt en brukerdrevet utvikling. Siste oppdatering (Versjon 2.1) kom 14. mars 2009, så utviklingen er fremdeles aktiv.

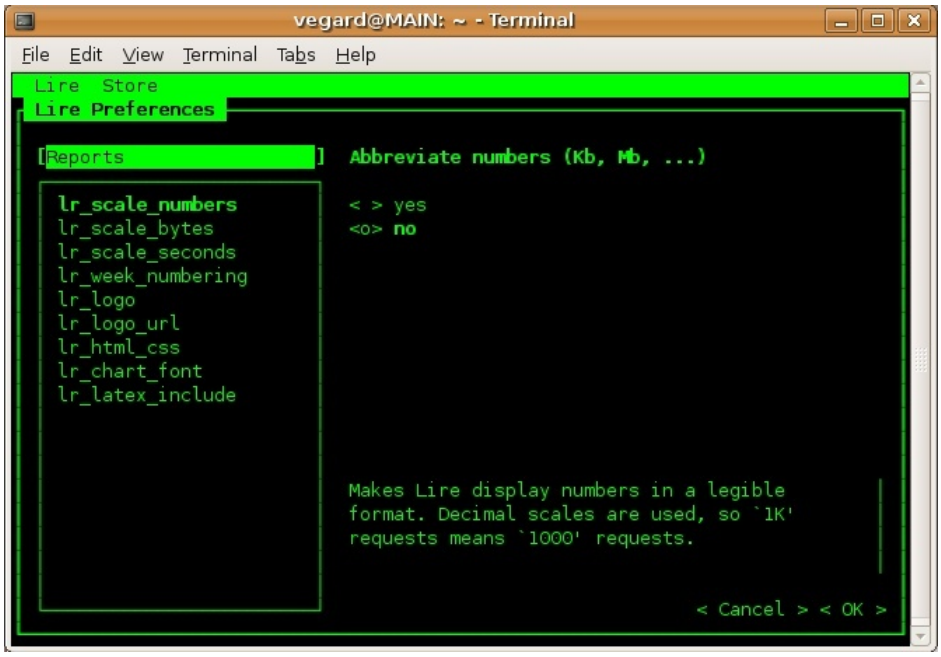
I motsetning til Sawmill og Splunk er Lire implementert for et mindre utvalg spesifikke tjenester, som mySQL, BIND, IPfilter, syslog, CISCO IOS, CUPS og Apache[26]. Tjenester som ikke er støttet kan man, ettersom det er Open Source, selv implementere, men dette krever både innsikt og innsats.

Brukergrensesnittet til Lire er ikke oppdatert med AJAX og tilsvarende slik som Sawmill og Splunk. Det er allikevel oversiktlig og forholdsvis greit å konfigurere, selv om det kan oppfattes som noe primitivt. Det viktigste er tross alt funksjonaliteten så lenge brukervennligheten er akseptabel. Rapportene

[26]For fullstendig liste, se: http://download.logreport.org/pub/current/doc/user-manual/ch01s03.html

kan genereres som hvilken som helst cron-jobb[27], og man får dermed frihet til å justere tidspunkt for kjøring, og hvilket datagrunnlag kjøringen skal anvende, på en forholdsvis enkel måte. Figur 3 viser et skjermbilde av Lire.

Lire leser kun fra filer lokalt på serveren, men har mulighet for å levere rapportene enten til epost eller lagret som fil. Det er også støtte for flere formater for rapportene, blant annet html, ps, pdf og text.



Figur 3: Illustrativt skjermbilde av grensesnittet til Lire

[27]Detaljer om oppsettet av crontab kommer i kapittel 5.6

5 Testoppsett

5.1 Testmiljø

Maskinvaren som ble anvendt var en Dell Poweredge 2950 med 8GB RAM. Dette er en virtualiseringsserver som har hypervisoren VMWare ESXi 3.5 i bunn, med 3 virtuelle servere installert. Ingen andre tjenester som ligger på disse serverne er aktive i produksjonsmiljøet ved KVT.

Det er i utgangspunktet mulig å sette spørsmålstegn ved om det er gunstig å teste tre oppsett som kjører på samme hardware. Belastningen på serveren er imidlertid så liten at dette ikke betyr noe i praksis. Alle serverne har mer enn nok ressurser til å håndtere mottak og behandling av loggdata, og loggmengden hver dag overstiger ikke 20MB. CPU-bruk er monitorert underveis, og det er kun ved intense søk med Splunk at den overstiger 30%. Det gjelder da kun den virtuelle maskinen som Splunk ligger på.

Serveren er plassert i produksjonsmiljøet, men med en brannmurkonfigurasjon som gjør at kun andre servere, men ikke vanlige datamaskiner kan kommunisere med den.

Enheter som ble konfigurert for å logge til disse maskinene er følgende:

- RADIUS/VPN-server - Debian
- DNS/DHCP-server - Ubuntu
- FOG-server (utrulling av image, oppdatering av nye maskiner)
- WEB-server (ikke besøkslogg)
- Windows 2003-server med Visma Lønn
- Windows 2003-server med Oracle
- Windows 2000-server med MicroMarc (biblioteksystem)
- 13 trådløse basestasjoner (DLINK DWL3200AP)

5.2 Installasjon av loggservere

Alle serverne ble initielt likt installert, med unntak av hostname og ipadresse. Et ISO-bilde av Ubuntu Server 8.04 LTS[28] ble lastet ned fra *www.ubuntu.com* og de virtuelle maskinene ble startet med ISO-bildet som oppstartsmedium.

Følgende valg ble tatt under installasjonen:

1. Språk ble satt til Norsk
2. Nettverk ble satt opp med DHCP
3. Hostname ble satt: Logg1, Logg2 og Logg3
4. Harddisk ble satt opp med valget: *Veiledet, bruk hele disken*
5. Navn, brukernavn og passord ble skrevet inn
6. LAMP og OpenSSH ble valgt som tjenester som skulle installeres fra menyen.
7. Passord ble satt i Mysql

Kun en bruker ble konfigurert. Om resultatet av denne oppgaven skulle føre til at man velger å anvende et eller flere av verktøyene permanent, må en ny installasjon til. Denne må ta hensyn til flere brukere, rettigheter til lesing/skriving av logger, eventuell bruk av chroot med mer, men dette er forholdsvis trivielle detaljer som kan løses om man bestemmer seg for å anvende dette på et senere tidspunkt. Hvordan dette løses vil også være avhengig av hvilke andre tjenester som eventuelt kjører på samme server, og hvilken tilgangspolitikk som er implementert. Ettersom man her kun har en bruker, vil tilgang til logger ikke være en problemstilling under arbeidet med denne oppgaven.

Av valgene for hvilke tjenester som skulle installeres, ble LAMP[29] og openSSH valgt (Kunne også vært satt opp med kommandoen tasksel i etterkant, men ble satt under installasjonen for enkelthets skyld).

[28] LTS - Long Term Support

[29] En samlebetegnelse for en webserver som inkluderer Linux, Apache, Mysql og PHP

openSSH ble valgt for å kunne aksessere maskinene eksternt på en enkel og sikker måte, da lokal tilgang (tty) til de virtuelle maskinene uansett må gjøres via et eksternt grensesnitt for hypervisoren. Dette grensesnittet krever Windows, og er i tillegg ikke like responsivt som openSSH.

LAMP ble satt opp på alle serverne, for å ha muligheten til å velge logg-analyseverktøy med web-grensesnitt senere. Installasjonen av hovedserverne ble gjort tidlig i prosessen, før utvalg av verktøy ble gjort, og de kan derfor ha mer installert programvare enn det som strengt tatt er nødvendig for det enkelte loggverktøyet.

Nettverksoppsett (IP-adresse, subnet mask, gateway og DNS-servere) ble gjort på DHCP-serveren på KVT, hvor de ble satt opp med faste ipadresser knyttet til maskinvareadressen til det virtuelle nettverkskortet.

Etter grunninstallasjonen var ferdig ble følgende kommandoer kjørt:

```
sudo apt-get update && sudu apt-get dist-upgrade
(Yes for å godta alle oppdateringer)
sudo apt-get install ntp
sudo apt-get install syslog-ng
```

ntp.kvt.vgs.no og *ntp.ubuntu.com* ble satt som tidsservere i */etc/ntp.conf*.

De ovenstående kommandoene sørger først for å oppdatere alle programvarepakker som er installert på den virtuelle maskina, for deretter å installere ntpd, som sørger for å holde tiden synkron med det interne nettverket (ntp.kvt.vgs.no), eller med tidsserveren til ubuntu om den interne skulle være ute av drift. Dette tilsvarer oppsettet på resten av servere på KVT, og sikrer at man har en felles riktig tid. *ntp.kvt.vgs.no* henter forøvrig også tiden fra *ntp.ubuntu.com*, slik at et eventuelt utfall av den interne ikke fører til tidsforskyvning. Syslog-ng ble også installert, som argumentert for i 4.2.

Til slutt ble et snapshot[30] tatt av den virtuelle maskinen, for å hindre at hele installeringen måtte gjentas om vesentlige feil skulle oppstå på et senere

[30]Dette er en funksjon i VMWare, som tillater at man tar et øyeblikksbilde av den virtuelle maskinen, og har mulighet til å gjenopprette dette automatisk om nødvendig.

tidspunkt.

Tabell 3 viser en oversikt over navn, ipadresser, og hvilke verkøy som er installert.

Tabell 3: Serveroversikt

Servernavn	IP-adresse	Verktøy
Logg1	158.38.100.32	Sawmill
Logg2	158.38.100.31	Splunk
Logg3	158.38.100.33	Lire

5.3 Konfigurering av syslog og syslog-ng

På loggserverne ble altså syslog-ng installert, og konfigurasjonsfilen syslog-ng.conf ble endret i henhold til filene som ligger i appendix under A.

Det kan her bemerkes at loggserveren Logg1 inneholder kode for å sende logger til Logg2 og Logg3 som både inkluderer intern logg fra Logg1 samt loggdata som kommer over nettverket. Loggdataene som de to "redundante" loggserverne genererer er besluttet ikke tatt med i den samlede loggmengden, da det i en reell implementasjon vil være kun en loggserver, og dataene fra disse serverne ikke vil bidra til en bedre logganalyse. Dette sikrer i tillegg at all analyse foregår på nøyaktig samme datamengde, hvilket er en forutsetning for å kunne sammenligne resultater generert på de ulike serverne.

Filplassering og detaljert loggformat er tilpasset det enkelte verktøy, som filene viser.

Det er brukt et felles oppsett for "options", men deler av dette må endres i en permanent implementasjon om denne foregår på en server med flere brukere, i henhold til argumentasjonen om brukere i forrige delkapittel.

På klientsiden eksisterer det 4 ulike oppsett:

- Det er ikke mulighet til å endre loggsystemet på basestasjonene, som fra før er syslog. Disse settes opp til å fjernlogge mot ipadressen til

Logg1[31].

- På serverne med en *nix-avart installert installeres syslog-ng på lik linje som over, og syslog-ng.conf endres i henhold til den generelle syslog-ng.conf for klienter som ligger vedlagt under punkt A.2 i Appendix.
- Et spesialtilfelle eksisterer for RADIUS-serveren, da denne ikke logger til syslog som standard, men til en egen fil. Denne er vedlagt under punkt A.2.1 i Appendix.
- På Windows-serverne kunne det vært brukt en syslog-ng-klient levert av Balabit Software, utviklerne av syslog-ng, men denne kommer kun med lisensierte versjoner av syslog-ng. Valget falt på en klient ved navn "*Eventlog to Syslog Utility (evtsys)*" som PURDUE har utviklet, og som er distribuert gratis, med åpen kildekode[32]. Denne sender data i syslogformat over nettverket umiddelbart når de kommer inn i Event Viewer i Windows, altså ikke som en batch-jobb med jevne mellomrom. Dette sørger for at rekkefølgen inn i loggfilene blir bevart i større grad enn en batch-jobb ville vært i stand til, hvilket gjør det lettere å få oversikt når manuell logglesing må til. "Evtsys" ble lagt inn som en tjeneste, og ble ved første gangs kjøring satt opp med ipadressen til Logg1.

Med dette oppsettet samles alt av loggdata og distribueres likt til loggserverne. Analyser som gjøres på samme tidsgrunnlag på de ulike serverne skulle dermed gi tilsvarende resultater, og dermed en god mulighet for en kvalitetssikret evaluering.

5.4 Installasjon og konfigurering av Sawmill

Sawmill krever registrering av bruker for nedlasting, men kun med epostadresse, og man får da tilgang til en 30-dagers fullversjon. Man velger i installasjonen hvilken versjon man ønsker å teste, og valget i denne oppgaven falt på "Professional", som beskrevet tidligere.

[31] Logg1 er navnet på en av loggserverne.
[32] Mer informasjon finnes ved søk etter evtsys på http://engineering.purdue.edu/

Nedlastingsdetaljer:

- *Versjonsdetaljer:* Linux (glib2.5(...); e.g., RedHat ES/AS 5)
- *Maskinvarearkitektur:* x86 (32-bit; e.g., Intel Pentium, 32-bit Xeon, AMD Athlon, Core Duo/Core2Duo)

Etter registreringen og nedlastingen ble følgende kommandoer kjørt for å installere og starte Sawmill:

```
tar -xvzf sawmill.tar.gz
sudo sawmill/sawmill8.0.5.1 #starter sawmill
```

Det kan her bemerkes at Sawmill ikke installeres i tradisjonell forstand, men kjører direkte fra utpakket arkiv. Endringer av konfigurasjon blir lagret i tekstfiler i filhierarkiet til Sawmill. På samme måte som Splunk, må konfigureringen av Sawmill gjøres via en nettleser, og da på port 8988.

Startkonfigurasjon

Første gang man aksesserer websnittet til Sawmill, kommer det opp en serie med valg som må gjennomføres.

Standardvalgene ble brukt, og disse valgene ble i tillegg satt:

- 30 dagers prøveversjon
- brukernavn/passord
- Professional ble valgt som versjon

Man får deretter opp websnittet til Sawmill, hvor man videre må opprette en profil for å angi hva som skal analyseres. Der ble disse valgene satt:

- 'Filer ligger på lokal disk'
- 'Filplassering:' /var/log/sawmill/
- Godtok forslag på 'syslog' som format

Da er startkonfigurasjonen ferdig, og man kan se siste loggrapport ved å trykke på view reports, og deretter 'Single-page Summary' som ligger nest nederst i menyen til venstre. Her er det mulighet for å sende rapporten som

epost, men ikke som en automatisk hendelse, dette må manuelt gjøres hver gang. Konsekvensen er her at man må bruke websnittet for å få tilgang til rapporten, men da er den også fullt oppdatert til enhver tid. Det krever en ekstrainnsats for administratorene, men vurderinger omkring det kommer oppgaven tilbake til i kapittel 8.

5.5 Installasjon og konfigurering av Splunk

Splunk kan brukes gratis på en loggmengde under 500MB, men det kreves også her en registrering for å få tilgang til nedlastingen. Etter registreringen ble Splunk lastet ned, og installert og startet via følgende kommandoer:

```
sudo dpkg -i splunk.deb
/opt/splunk/bin/splunk #Starter Splunk
(Velg 'Yes' for å godta lisensavtale)
```

Splunk er da installert og startet, og ligger under /opt/splunk. For å sette startparametrene må man bruke en nettleser og aksessere maskina det ble installert på, da på port 8000. Valg av port kan om ønskelig endres etter at programmet er startet.

Startkonfigurasjon

På førstesiden som kommer opp når man får opp websnittet til Splunk, må man første gang velge "Index Files", for å sette opp at Splunk skal hente de riktige loggfilene. Bruker da disse innstillingene:

- Monitor directory: /var/log/splunk/
- Fqdn: Logg2
- Set source type: Manual
- Source type: syslog

Etter å ha valgt submit, og deretter linken til 'main page', kommer man til hovedbildet til Splunk, som viser nåstatus, med de siste loggmeldinger, grafisk oversikt over loggmengden de siste 30 dager, sammen med flere andre måter å

filtrere loggmeldinger på. For oversikt anvendes de siste 7 dager som standard utvalg, men andre tidsutvalg er lett tilgjengelig via en nedtrekksboks.

5.6 Installasjon og konfigurering av Lire

Lire er tilgjengelig i ubuntu-arkivene (repositories) for tilgjengelig og tilpasset programvare, og installeres ved hjelp av følgende kommandoer:

```
### Installasjon av Lire
sudo apt-get install lire
### Kommandoene under sørger for installasjon av
### ekstra programvarepakker, og er er nødvendig
### for å gi støtte for alle output-formater.
sudo apt-get install texlive-omega
sudo apt-get install ghostscript
sudo apt-get install ploticus
### Starter programmet:
lire
### Linker til de sist installerte programmene settes
### opp under Preferences i selve programmet.
```

Versjonen som er installert fra repository er 2.02 (fra Juli, 2006), men ikke er den siste versjonen. Endringene som står oppført på versjonsoversikten[33] fra denne versjonen til siste versjon har ikke konsekvenser for denne oppgaven. Endringene går i hovedsak på flere støttede tjenester, som ikke blir benyttet på KVT.

I motsetning til Splunk og Sawmill som har konfigurasjonen via et websnitt, settes innstillingene til Lire opp i fra et grensesnitt som holder seg i terminalen. Når man kjører den siste kommandoen som står ovenfor (lire), blir man presentert for et grensesnitt for innstillinger, hvor alt av konfigurasjon blir satt.

Initielt måtte en Store settes opp, som er termen Lire bruker på å benevne et sett med logger og tilhørende rapporter. Plassering av Store ble satt til

[33] http://download.logreport.org/pub/current/ChangeLog

/var/log/lirestore. Etter at denne er satt opp, må det defineres hvilke loggfiler som skal inngå, under *input*, samt hvilket format som skal anvendes under *report*.

I KVT's tilfelle ble */var/log/lire/syslog* valgt som kildefil, innhenting av logger ble satt til en gang i timen, og *syslog* valgt som loggformat, med resultatet presentert med pdf. Alt av logging anvender enten syslog eller syslog-ng som system, og i begge tilfeller kan det opprinnelige syslog-formatet anvendes på loggene. Lire har i utgangspunktet ikke innebygd støtte for det utvidete formatet syslog-ng er i stand til å anvende, som inkluderer facility, level og et utvidet tidsstempel i selve loggoppføringen, så loggformatet på serveren som Lire kjører på må reflektere dette ved å anvende standard-formatet til syslog-ng. Det er mulighet for å tilpasse loggformatet slik at Lire også kan utnytte den ekstra informasjonen som nevnt over, men ytterligere detaljer omkring dette blir presentert under "Resultater".

Det neste steget er å definere når og hvordan rapporter skal genereres. Det eksisterer her valg på frekvens og format som rapporten skal anvende. Daglig rapportering (daily) ble valgt, med pdf som output-format. Alternativer for hvordan resultatet skal presenteres er blant annet ps, html og tekst, men ettersom det her, for effektivitetens del, er essensielt å levere lett tilgjengelige rapporter med god lesbarhet, fremstår pdf som et godt valg. Det er også trivielt å sende tekst eller pdf-dokument som vedlegg i epost, som vil være den foretrukne formen for å gjøre rapportene tilgjengelig for administratorene. Bruk av epost vil føre til at mindre intervensjon er nødvendig kreves for å lese rapportene, da epost, uavhengig av logganalyse, anvendes som et verktøy gjennom arbeidsdagen.

En egenart ved Lire, sett i sammenheng med de andre verktøyene, er at verktøyet i utgangspunktet ikke kjører permanent i bakgrunnen på den aktuelle maskinen. Dette medfører et behov for å manuelt sette opp en kjøreplan i *crontab*[34]. Under følger crontab som ble konfigurert på Logg2.

[34]crontab er en tekstfil som Ubuntu bruker for å sette i gang oppgaver som skal utføres til planlagte tidspunkt.

```
#mm hh dom mo dow cmd
30  23  *   *  *   lr_cron daily /var/log/lire
25  *   *   *  *   lr_cron hourly /var/log/lire
```

Dette betyr at en kjøring av rapportgenereringen foregår hver dag, klokken 23.30, i tillegg til at den siste logginformasjonen hentes hver time. Dette sørger for mindre belastning enn om den skulle kjørt en gang i døgnet da det blir mindre data å håndtere for hver gang, samt gir muligheter for ytterligere rapporter til andre tider av døgnet om ønskelig, da dataene allerede er innhentet.

Et mer generisk oppsett av crontab tilpasset andre tidsintervaller for rapporter kan hentes via kommandoen *info lr_cron* på en maskin som har lire installert.

6 Gjennomføring

Loggdata sendes automatisk fra de konfigurerte komponentene til loggserverne, man har med andre ord kontinuerlig oppdaterte loggdata for analyse.

Med gratisversjonen av Splunk følger det ikke med mulighet til å begrense tilgang til websnittet, hvilket ikke er ideelt. Passord, ipadresser og andre typer 'avslørende' informasjon kan og vil ligge i loggene, og en begrensning av tilgang til serveren måtte til. Dette ble løst ved å sette opp en regel i brannmuren på KVT, som sørger for at kun ipadresser anvendt av de IT-ansvarlige fikk tilgang. Det bør her bemerkes at nettet elevene anvender er separat atskilt fra personalnettet, slik at de ikke har mulighet for å angi ipadresser inkludert i unntaket satt opp i brannmuren. Andre ansatte kan teoretisk få tilgang ved å endre ipadresse, men dette vil ikke være et sannsynlig scenario. Om det skulle skje, medfører det ikke mulighet for endring av logger, så konsekvensen for oppgaven er liten. Dette må dog tas hensyn til ved en permanent implementasjon.

På versjonen av syslog-ng som ble installert på loggserverne er standardinnstillingen at informasjon om facility og level ikke er med i lagringen av loggoppføringen. Splunk støtter denne type ekstraoppføring i loggdataene, og for at Splunk skulle ha best mulig loggdata tilgjengelig, ble en template (t_filter) konstruert på Logg2 for å sørge for bevaring av denne informasjonen (Se A.1.2 for selve konfigureringen). Ettersom det opprinnelige filhierarkiet til syslog ikke anvendes, forsvinner i utgangspunktet noe av kontekstinformasjonen som gis gjennom hvilken fil som blir lest[35], og dette kan ses på som kompensering. Her ble det også satt opp at dato skulle være på ISO-format, som tar vare på dato inkludert år og tidsforskjell, og navn på avsender.

Loggrotasjon måtte endres på alle serverne, da filstrukturen for loggene ble endret. Oppføringer i /etc/logrotate.d/syslog-ng ble tilpasset det nye filhierarkiet, med daglig rotasjon over 365 repetisjoner. Det eksisterer fra før ikke rutiner for å arkivere logger, så dette er en temporær løsning, som bør justeres når dette er på plass. Dette er dog en oppgave for administratorene

[35] Filene som i utgangspunktet genereres er fordelt på facilities, som tabellen i 3.4.2 viser en oversikt over.

ved KVT, og berører ikke oppgaven direkte. På Logg2 ble kun de lokale kontrollfilene (Se A.1.2) satt opp i logrotate, da arkivering av loggdataene blir foretatt automatisk av syslog-ng.

Når rapporter skulle genereres fra Lire, viste det seg vanskelig å få satt disse til å komme i pdf-format automatisk. Rapportene ble korrekt generert, men kun i et råformat. En ekstra kommando måtte kjøres manuelt for å generere rapporten i riktig format. Årsaken til dette er ennå ikke kjent, men ved hjelp av et script og crontab-oppføring ble det funnet en løsning som fungerer, selv om dette bør rettes opp i ved en permanent installasjon. Dette skriptet kjøres 10 minutter etter at rapporten genereres, og resulterer i en pdf-fil, som blir liggende på hjemmekatalogen til brukeren på serveren.

Sawmill og Splunk støtter ikke automatisk sending av rapporter som gir nok informasjon til å evaluere resultatene på. Dette fører til at webgrensesnittene med utvalgene som er beskrevet under startkonfigurasjon og videre under Resultater ble lastet på samme tid som Lire-rapporten ble generert. Dette gir resultater som er direkte sammenlignbare, da de genereres på samme mengde data, men eksisterer ikke i samme form eller format. Når det gjelder Splunk er det heller ingen utskriftsfunksjon som inkluderer grafer, og en eventuell utskrift av resultatvinduet måtte inkludert bruk av "printscreen", hvilket heller ikke er ideelt når resultatene fyller flere skjermhøyder. Formen på presentasjonen av resultatene er i tillegg så vidt forskjellige på disse verktøyene at en direkte sammenligning av data ikke er mulig. Resultater og diskusjon av disse må derfor være beskrivende i henhold til tidligere omtalt teori, og ikke statistisk rettet, som hvis mulig ville ha vært den foretrukne metoden.

6.1 Avgrensning

syslog-ng.conf ble på de ulike serverne konfigurert innledningsvis i henhold til vedlegget i Appendix. Denne konfigureringen ble gjort etter tester av hvilke loggformat som de ulike verktøyene støtter, hvor det mest innholdsrike ble valgt der dette var mulig (Dette gjelder initielt Splunk, men som det skal vise seg også Lire, selv om dette ikke ble oppdaget før senere). Dette blir tatt nærmere opp under "Resultater".

Loggdataene som ble anvendt for analyse er de reelle som systemet på KVT genererer, og rapportene ble generert på bakgrunn av data fra ett døgn, men inkluderer også de siste 30 døgns data der det er mulig og relevant. Datamengden ble generert torsdag 26 mars, og loggdata for dette døgnet inneholder 80411 linjer med loggdata fra 20 servere/nettverkskomponenter.

For å kunne vurdere resultatene som rapportene/resultatvisningene inneholdt ble en manuell gjennomgang av loggdataene utført, i tillegg til at resultatene fra de ulike verktøyene ble sammenholdt.

På bakgrunn av resultatene fra den manuelle analysen sett opp mot de genererte rapportene, ble innstillingene for rapportering i de ulike verktøyene justert, med mål om å ende opp med best mulig resultater. Hvilke endringer som ble utført er beskrevet og kommentert under "Resultater". Den første kjøringen ble utført med innstillingene fra startkonfigurasjonen, som beskrevet under "Testoppsett".

Nettverket på KVT er ikke stort i logganalysesammenheng, og antallet kritiske feil vil som regel være lavt eller ikke-eksisterende. Det ble innledningsvis gjort en vurdering på om kritiske hendelser manuelt skulle plasseres i loggene, for å undersøke om de ulike analysene ville plukke opp disse, men dette ble vurdert til å kunne gjøre valideringen av resultatene mer komplisert. I verste tilfelle kunne en underbevisst justering av verktøyene, til å fange opp akkurat disse oppføringene, føre til en falsk trygghet for kvaliteten på logganalysen. Det var dog ikke til å unngå at det foregikk noe arbeid som krevde restart på servere i tiden denne oppgaven ble utført.

Rapportene som ble sammenholdt ble utført på samme tidspunkt, med de samme rådata, og er dermed i utgangspunktet direkte sammenlignbare. Dette betyr ikke at de inneholder nøyaktig de samme resultatene, men dette kommer som nevnt av de ulike tilnærmingene verktøyene anvender, kvaliteten på verktøyene samt i hvor stor grad de er tilpasset logginformasjonen som blir generert på KVT.

Dette inkluderer ikke loggdata fra Novell Netware-serverne, som viste seg vanskelig å implementere i perioden oppgaven ble utført. Som nevnt tidligere skal en oppdatert versjon tas i bruk til høsten, og der anvendes SUSE OES

som operativsystem. Dette operativsystemet støtter syslog-ng, og logger derfra kan da inkluderes i systemet på samme måte som det har blitt gjort med andre loggkilder.

7 Resultater

Resultatene fra analyserapportene de tre verktøyene genererer blir her presentert, om enn ikke i sin fullstendige form. Som kommentert tidligere er formen på resultatene fra de ulike verktøyene svært ulik. Her vil derfor de mest vesentlige og/eller interessante resultatene bli trukket ut for evaluering og sammenligning. Denne delen er oppdelt i 2 kjøringer, hvor kjøring 1 anvender startkonfigurasjonen til verktøyene, mens kjøring 2 anvender en tilpasset konfigurasjon hvor endringene er beskrevet under hvert enkelt verktøy. Disse endringene i konfigurasjonen blir utført for å om mulig forbedre rapportene, slik at det ender opp med en konfigurasjon som gir en best mulig logganalyse for KVT.

En manuell gjennomgang ved hjelp av *grep* med ulike parametre ga følgende spesielle hendelser som skiller seg ut, og som vil være interessant for nærmere kontroll:

- restart av server ubuntu-fog
- oppstart av basestasjon med ipadresse 158.38.100.215
- restart av syslog-ng på server MAIN

Disse resultatene ble altså funnet ved hjelp av stikkordssøk, hvilket godt illustrerer behovet for en automatisert logganalyse, da man kun finner spesifikke elementer som man søker etter. Korrelering av data, baseline-anvendelse mot statistikk er ikke mulig ved en slik manuell gjennomgang. Til tross for dette, vil disse resultatene være ønskelige at stikker seg ut, men med en så omfattende loggmengde er det umulig å vite om dette er et komplett bilde av anomalier.

7.1 Kjøring 1

7.1.1 Sawmill

Som nevnt tidligere anvendes Sawmill ved hjelp av et web-grensesnitt, og alle innstillinger og rapporter endres og vises på skjermen.

Første rapportgenerering av Sawmill (ved valg av Single-Page Summary) gir i utgangspunktet ut den totale analysen over de 30 dagene som det eksisterer loggdata for. Ved et nytt valg av dagens dato i kalenderen i skjermbildet genereres samme rapportoppsett nok en gang, men da med kun siste døgns data. Oppsettet av denne rapporten viser seg da å være mindre tilpasset en daglig rapport som innholdsoversikten under viser:

- Overview
- Date/Times
- Years
- Months
- Days
- Days of week
- Hours of day
- Logging devices
- Daemons
- PIDs
- Daemon messages
- System messages

Alle punktene inneholder en tabell med oppføringer relativ til overskriften, mens de første punktene til og med "Hours of day" i tillegg består av søylegrafer som beskriver mengden loggdata i henhold til det enkelte punkt.

Når en "dagsrapport" genereres blir det generert kun en søyle i alle de første punktene, med unntak av Hours of day, som da blir bestående av loggmengden de siste 24 timer, fordelt med en søyle for hver time. De første punktene blir da overflødige, og gjenstand for utbedring til Kjøring 2.

Statistikk er hva Sawmill har som hovedfokus, som også resten av innholdet av rapporten gjenspeiler. Hvert av punktene fra "Logging devices" og ut gir en statistisk fremstilling av loggkilder, hvilke daemons som har blitt anvendt, system-beskjeder med mer. Her er det sortert fra mest bruk til minst brukt, med en fremvisning av de 10 mest brukt, angitt i reelle tall og prosent av total, samt en tallmessig oppsummering for hvert punkt.

I denne rapporten vises ingen meldinger som antyder restart av servere, restart av syslog-ng eller andre hendelser som har inntruffet på nettverket som i en ideell logganalyse burde ha blitt trukket frem.

Totalt sett er dette altså ikke en god rapport med tanke på å finne anomalier i loggene. Den gir derimot en god oversikt over den generelle anvendelsen av nettverket, men dette er ikke alene nok til å hjelpe administratorene med en mer effektiv logganalyse. Grafisk oversikt over loggmengde per time kan gi antydninger på at noe ikke er som det skal være dersom det inntreffer loggmengder som virker unaturlige, enten på grunn av mengde, tidspunkt eller en kombinasjon.

7.1.2 Splunk

Splunk omtaler seg som et logganalyseverktøy med søk som sin sterkeste side[11], som klart kommer frem i brukergrensesnittet. Det er på lik linje med Sawmill også kun web-basert, men har en annen tilnærming enn utgangspunktet er med Sawmill.

Førstesiden til Splunk er definert som et "dashboard", hvilket betyr at man selv kan velge hva som skal ligge der gjennom å endre profilen. Første kjøring anvender dog det opprinnelige oppsettet, men blir endret noe til kjøring 2.

Når det gjelder Splunk blir det feil å omtale loggoversikten som en rapport. Her kan man velge hvilket tidspunkt man ønsker å se på, og hva man ønsker

å se på, men alt endrer seg dynamisk i hovedvinduet etter hvert som man gjør valg eller endringer. Som figur 2 viser så er førstesiden tre-delt, med et søkefelt, en søylegraf med loggmengde i det utvalgte tidsrommet, samt de siste loggmeldingene i det samme tidsrommet.

Heller ikke her er det i utgangspunktet trivielt å hente ut data om anomalier som krever administratorenes oppmerksomhet. Den grafisk oversikten over loggmengde i den aktuelle tidsperioden (forrige døgn ble utvalgt), gir samme oversikt som Sawmill, og kan være nyttig, men alene gis det ingen direkte mulighet for å finne enkeltfeil blant loggene. Heller ikke her blir restarter av server eller andre hendelser som har vesentlig relevans uthevet. Det kan her allikevel hende at de vil kunne dukke opp, men det er avhengig av de da er inne blant de siste hendelsene når denne loggoversikten blir åpnet, da det kun er de siste hendelsene som kommer frem direkte i skjermbildet.

Et standardoppsett av Splunk gir dermed heller ikke veldig god støtte til administratorene når det gjelder å plukke ut anomalier i loggene, selv om den grafiske oversikten kan antyde at noe ikke er som det skal, dersom loggmengden har plutselige endringer på spesielle tidspunkt.

Mulighet for korrelering av data er også tilgjengelig med Splunk, selv om det ikke går automatisk. Ved å trykke på et tidspunkt i loggene kommer man rett inn i loggsekvensen som ligger rundt det tidspunktet, og man kan umiddelbart se hva som foregår rett før og rett etter. Dette fordrer dog at det finnes en grunn til å sjekke akkurat det tidspunktet, og generell korrelering som ikke er knyttet til spesielle feilmeldinger er ikke mulig. Statistisk sammenligning over tid som Sawmill har mulighet for er her ikke direkte tilgjengelig, men det er mulig å bygge opp søk som erstatter noe av dette behovet. Dette må gjøres manuelt og kan være en naturlig del av en manuell analyse etter det er funnet feil.

7.1.3 Lire

Lire skiller seg fra de andre gjennom å levere resultatene i en ren rapportform, uten mulighet for endringer der og da. Rapporten i pdf-format består av 19

sider, hvilket i utgangspunktet virker noe omfattende i forhold til å skulle være effektiviserende for administratorene. Rapporten er oppdelt i følgende punkter:

```
1: Overview Reports
--Facility
--Level
--Top 10 Hosts
--Number of Messages Logged per 1h Period
2: Most Common Event Reports
--Top 10 Processes
--Top 50 Messages
--Top 5 Messages by Process, Top 30 Processes
3: By 1h Period Reports
--Top 10 Processes
--Top 10 Messages
4: Warning or Higher Level Events
--Top 0 Messages
--Top 0 Process
A - DLF Schema for Syslog Superservice
```

I utgangspunktet ser denne rapporten lovende ut, med tanke på at punkt 1 og 4 antydes å inneholde henholdsvis informasjon om meldinger sortert på level, og en oversikt over meldinger som er registrert med level "warning" eller mer alvorlig. Dette er informasjon som verken Sawmill eller Splunk ga mulighet for med startkonfigurasjonen. Det synes noe merkelig at punkt 4 har som standard at ingen av oppføringene skal fremvises (Top 0 Messages/Process), men dette skal være mulig å konfigurere.

Det viser seg derimot at med standardformatet for syslog som er anvendt på serveren (som Lire kjører på) får ikke Lire nok informasjon om loggoppføringene til å kunne plukke ut noen hendelser under punkt 1 og 4, og rapporten ender dermed som ren statistikk over inntrufne hendelser i perioden den er satt til å analysere. Level og Facility inkluderes ikke i loggoppføringen og er dermed ikke tilgjengelig, årsaken til dette er kommentert under delkapittel 5.6.

Siste del av punkt 2 inkluderer derimot noe som kan være av interesse utover statistikk, nemlig de 5 meldingene med høyest frekvens fordelt på hver prosess. Dette kan gi nyttig informasjon, da enkelte prosessers tilstedeværelse i loggen i seg selv kan være tegn på at noe er galt. En forekomst av *su*[36] på en av serverene kan antyde at en nærmere kontroll er på sin plass.

Den foreløpige slutningen når det gjelder rapporten som Lire genererer er at Lire, med det gjeldende loggformatet, heller ikke blir en løsning som sørger for å finne anomalier i loggene på en måte som bidrar til en mer effektiv logganalyse for administratorene.

7.2 Kjøring 2

7.2.1 Sawmill

Det opprinnelige oppsettet i Sawmill kan, selv om det ikke er ideelt for den rene dagsrapporten, med litt justering allikevel anvendes for å bidra til en bedre analyse. Dette kommer av at en god oversikt over tidligere loggmengde kan antyde hvorvidt en dags loggmengde er "naturlig", som diskutert i punkt 4.5. Her fremheves viktigheten av å sammenligne loggmengden med baseline, som vi her kan få gjennom denne først genererte rapporten. For å gi en mer oversiktlig og gjennomførbar mulighet for dette ble det før kjøring 2 opprettet filtre i Sawmill, som sørger for at statistikk fra den enkelte ukedag blïr filtrert.

Det finnes dog ingen mulighet til å justere Sawmill for å tilrettelegge for umiddelbar identifikasjon av anomalier/alvorlige feil i loggene. Til tross for dette blir det her forsøkt å forbedre rapporten slik at den gir så gode resultater som mulig.

Rapportvalget "Single-page summary" ble justert slik at den nå inneholder følgende deler:

- Days (kun graf)
- Hours of day (kun graf)

[36] su er et program/kommando på *nix som kalles på når man ønsker å anvende en annen bruker med tilhørende rettigheter.

- Logging devices (kun tabell)

Denne rapporten inneholder altså ikke kun informasjon om siste døgn, men som, med filteret for ukedag aktivert, gir en baseline som administratorene kan bruke for sammenligning når siste døgns rapport skal evalueres. Filteret sørger altså for å generere en baseline kun for den aktuelle ukedagen. Den siste søylen i grafen som viser aktivitet per aktuell ukedag vil da være gårsdagens/utvalgt dags loggmengde.

I tillegg til denne baseline-rapporten, ble det satt opp en ny rapport (Single-page Summary-PreviousDay), som tar utgangspunkt i den første, og inkluderer følgende komponenter:

- Hours of day (kun graf)
- Logging devices
- Daemons
- Daemon Messages
- System Messages

Resultater

Denne rapporten viser først en graf over loggmengde per time, som kan sammenholdes med baseline fra rapporten over, og deretter tabeller med logghendelsene med høyest frekvens, sortert på de ulike komponentene i listen over. I det aktuelle tilfellet gir ikke dette mye mening, da de foregående ukene har vært preget av ulike prosjekter og liknende på KVT, slik at det ikke ender opp som nyttig å sammenligne ukedag for ukedag i et så lite intervall.

Det er ikke utført endringer fra kjøring 1 som endrer innholdet i disse komponentene, og en inspeksjon av disse gir bekreftelse på at det er de samme resultatene som i første kjøring. Det kan være nyttig statistikk med tanke på å beregne nettverksbelastning og liknende, men inneholder altså ingen indikasjoner på at servere har restartet eller tilsvarende.

7.2.2 Splunk

Første kjøring av Splunk ga et tredelt oppsett, med søk, loggmengde på utvalgt tidsperiode og siste hendelser, men inneholdt som kjent ingen komponenter som kunne bidra til å finne detaljer om anomalier i loggdataene. For å rette på dette ble følgende elementer lagt til før kjøring 2:

- Søylegraf som viser loggmengde fordelt på de ulike serverne.
- Søylegraf som viser "Error last 24 hours"

Loggmengden fordelt på de ulike serverne fremkommer som en søylegraf, hvor hver søyle er fargedelt etter de ulike serverne, og har de siste 24 timer som standard utvalg. Denne erstatter den tidligere varianten som kun hadde med loggmengde.

"Error last 24 hours" er en innebygd funksjon i Splunk som også anvender en søylegraf, men som kun inkluderer loggoppføringer som inneholder spesielle termer. Disse kan justeres, og i dette tilfellet endte justeringen opp med disse termene:

- warning
- .err
- .emerg
- .alert
- restart
- shutdown
- failure
- "wrong password"

Resultater

Kjøring 2 av Splunk ga mer hensiktsmessige resultater enn kjøring 1. Spesielt er grafen som viser "Error last 24 hours" effektiv, da man for det første får

et grafisk bilde av hvordan feilene er fordelt, og samtidig kan klikke direkte på grafen for å se de konkrete meldingene, ferdig sortert. Her ble både loggoppføringene som inneholdt informasjon om restart av server, restart av syslog-ng hos en av klientmaskinene, samt også oppstart av en basestasjon inkludert.

Oversikt over loggmengde fordelt på timer som også inkluderer hvilke servere/komponenter som er mest aktive kan også være nyttig når spesielle hendelser inntreffer som fører til store antall loggmeldinger fra enkeltkomponenter. I loggdataene som her ble analysert kom ikke dette til nytte, men det er lett å se for seg at dette kan være meningsfullt å ha med.

7.2.3 Lire

Første kjøring av Lire bar preg av at loggformatet som ble anvendt umuliggjorde gode resultater. Etter studier av dokumentasjonen til Lire ble et annet loggformat funnet til å kunne gjøre jobben bedre, men dette krevde noe bearbeiding av loggdata. Ettersom loggdataene som lå på serveren Lire kjørte på ikke inneholdt informasjon om level og facility i filene, ble loggfilene på Logg2 kopiert over og modifisert til å passe dette nye formatet.

Detaljer om denne endringen ligger i appendix, under B. Det må her påpekes at selve loggbeskjeden ikke ble endret, loggformatet ble endret til å inkludere level og facility, samt en modifisert tidsangivelse.

Forsøk på endringer slik at "Warning or Higher Level Events" kommer ut med 10 hendelser i stedet for 0 lyktes ikke. Det er uvisst om dette kommer av feilkonfigurasjon eller en feil i programmet.

Resultater

Resultatene fra kjøring 2 med Lire hadde samme oppbygging som i kjøring 1, ettersom ingen endringer ble gjort i selve rapportstrukturen.

Rapporten var nå komplett, med tanke på innhold i de ulike punktene, og det faktiske resultatet var også vesentlig mer hensiktsmessig. På første side var det nå en oversikt over antall meldinger fordelt på hver facility, antall

meldinger fordelt på hver level, samt de 10 komponentene som genererte flest meldinger.

At forsiden gir oversikt over antall meldinger fordelt på level gjør at et øyekast fra administratorene er nok for å registrere hvorvidt en loggoppføring med hendelse kritisk har inntruffet. Som nevnt tidligere i oppgaven vil hvilken level man ønsker å sortere etter være avhengig av komponenter og tjenester i nettverket, men uansett hvor man setter grensen for hva som er mest interessant er indikasjonen på det her direkte tilgjengelig.

Strukturen på rapporten er som nevnt ikke endret, men det nye loggformatet medfører noen endringer i utseendet og innholdet til noen av oppføringene. Der det tidligere var slik at hver prosess var fordelt på kilde og prosess, er dette nå endret til å kun være oppdelt i prosesser, som intensjonen også er med dette punktet. Delen med "Warning or Higher Level Events" er fremdeles tom for oppføringer, men inneholder nå i det minste summen av hendelser som er registrert, nemlig 28. Dette tallet stemmer overens med antallet hendelser som ble funnet ved manuell gjennomgang. Ettersom Lire kun anvender level som sikt, kan det konkluderes at Lire fant de samme loggoppføringene som den manuelle, selv om de ikke ble eksplisitt presentert.

Resultatene inkluderer altså informasjon om antall hendelser med level "warning" eller høyere, men inneholder ikke informasjon om korrelering, sammenlignende statistikk i forhold til baseline eller liknende.

8 Diskusjon

Resultatene bærer preg av at verktøyene har svært ulik tilnærming til logganalyse. Der hvor Sawmill gir mulighet for etablering av en baseline, har Splunk mulighet for å filtrere på angitte søkefelt(maskering og statistikk), og Lire gir en rapport basert på level i loggoppføringen sammen med oppsummeringer fordelt på de ulike enhetene som leverer, tjenestene som er i bruk, og loggoppføringer i sin helhet.

Dette gjør det ikke trivielt å sammenligne resultatene seg i mellom, men gir en god mulighet til å se på hva programmene faktisk leverer av analyse. Som beskrevet i delkapittel 4.5, benytter en ideell logganalyse både baseline, alvorlighetsgrad (level), maskering og statistikk, samt korrelering som grunnlag for sin analyse. På bakgrunn av dette kan det fastslås at de utprøvde verktøyene ikke er ideelle, da ingen av verktøyene berører mer enn et av disse punktene som grunnlag for sine resultater.

På tross av dette kan det allikevel være mulig å utnytte disse verktøyene som del av en logganalyse. Sawmill sin presentasjon av baseline kan være nyttig, spesielt med tanke på at det med filter for ukedag, kan etableres baseline for hver ukedag, som det har blitt nevnt tidligere er nyttig i en skole-sammenheng. Hvorvidt dette vil være verdt innsatsen med å tilpasse verktøyet, med tanke på både arbeidstimer og bruk av ekstra lagringsplass, må vurderes i hvert enkelt tilfelle. At man må aksessere grensesnittet for å få tilgang til rapporten vil da måtte inngå i vurderingen.

Splunk er heller ikke en perfekt løsning for en tilnærming mot en automatisert logganalyse, selv om verktøyet tilbyr veldig gode muligheter for sporing av feil etter at man fra andre kilder har funnet grunn til etterkontroll. Måten Splunk gjør så godt som alle deler i loggene direkte søkbare ved klikk, gjør at slik sporing vil være svært effektiv, og korrelering kan finnes manuelt. Manglene på den utprøvde logganalysen er derimot så store at dette kun bør komme i tillegg til et annet system. Den grafiske fremstillingen av feil under den siste perioden er oversiktlig, men finner kun forhåndskonfigurerte termer som er definert som anomalier. Dette fører dog til at Splunk kan anvendes

som en automatisering av den manuelle kontrollen av stikkord.

Lire fant de samme feilene som ble oppdaget ved den manuelle analysen, som beror på at de samme metodene blir brukt. Her er det dog ikke mulighet for statistisk sammenligning eller korrelering, men ettersom rapporten tar med seg loggmeldingene som er merket med alvorlighetsgrad "Warning" eller verre kan det tilsynelatende gi inntrykk av at jobben er utført, når man har funnet noen feil med høy alvorlighetsgrad. Fordelene med Lire er at det kan settes opp helt automatisk, og kan erstatte det manuelle søket etter disse alvorlighetsgradene.

Ingen av de 3 anvendte verktøyene gir en fullverdig analyse. En anvendelse av de sterke sidene til alle 3 vil ikke være effektiviserende for administratorene, selv om bruk av et eller flere kan gjøres i enkelttilfeller, for å undersøke spesifikke logger. At Sawmill og Splunk krever bruk av et webgrensenitt for å få tilgang til loggene har ikke blitt tillagt vekt når det gjelder drøfting av verktøy. Det kreves ikke mer av administratorene å legge inn disse webadressene som startsider i en nettleser, enn det kreves å lese en epost eller en rapportfil.

Hvorvidt disse systemene vil kunne føre til at maskinvarefeil blir oppdaget i en tidligere fase enn hva tilfellet tidligere har vært, er umulig å si noe sikkert om. Det har, under tiden arbeidet med oppgaven har foregått, ikke vært noen maskinvarefeil på systemene på KVT, og det har dermed ikke vært mulig å evaluert hvordan disse eventuelt ville fremkommet i logganalysene.

Tidsutvalget for den endelige logganalysen kan synes noe smalt, men det er blitt vurdert til å være bredt nok for å analysere disse verktøyene, da de har helt ulike egenskaper. En utprøving av andre loggsystemer som i større grad har uniforme logganalyser kan ha behov for en større loggmengde, men i dette tilfellet ble et vindu på et døgn vurdert til å være bredt nok.

Bruk av syslog-ng som loggsystem har vist seg å fungere uten problemer, selv om tilpasning av format til Lire ikke ble perfekt første gang. Det ble tidligere i oppgaven antatt at, med et utvidet loggformat generert av syslog-ng, et sømløst skifte av logganalyseverktøy kunne foretas på toppen av dette. Dette har vist seg å være en feilslutning, da formatet må tilpasses hvert enkelt verktøy.

9 Videre arbeid

Videre arbeid omkring dette temaet vil måtte inneholde flere logganalyseverktøy, da de tre som er utvalgt sannsynligvis ikke er best av mengden som totalt er tilgjengelig, og heller ikke nødvendigvis best tilpasset behovene på KVT. Det er dog en utfordring å finne frem til logganalyseverktøy, da det som nevnt tidligere ikke finnes en fullstendig rangering eller sammenligning av ulike verktøy.

Både Splunk og Sawmill har mulighet til selv å motta loggdata over nettverket, en funksjon som i denne oppgaven ikke har blitt utprøvd. Om dette resulterer i samme loggdatagrunnlag kan det gjøre bruk av syslog-ng overflødig på loggserveren. En reduksjon i kompleksitet medfører ikke bedre logganalyse, men kan gjøre prosessen enklere å forholde seg til.

Ved ytterligere anvendelse av Lire må det tas rede på årsaken til at automatisk generering av pdf ikke fungererer, samt modifisering av rapporten til å inkludere selve loggoppføringene som har level høyere enn "Warning".

En permanent implementasjon av en eller flere av løsningene beskrevet i denne oppgaven krever at utbedringer blir gjort på systemet. Dette gjelder spesielt maskering/anonymisering av logger, inkludering av den nye versjonen av Novell Netware (som bruker SUSE OES), samt behandling av lese og skriverettigheter på loggfilene. Generell sikring av loggserveren må også utføres, med tanke på hvilke tjenester som skal kjøre og hvem som skal ha tilgang til å logge på.

10 Konklusjon

Løsningene som her er implementert og utprøvd støtter utvidelse i form av flere loggkilder uten videre konfigurasjon, og er gjennomført uten kostnader for KVT. syslog-ng og Lire er open source-verktøy, mens Splunk og Sawmill er kommersielle produkter. Splunk er gratis å bruke med loggmengden som eksisterer på KVT, mens en permanent anvendelse av Sawmill vil medføre en kostnad i størrelsesorden 5000 kroner i året.

Implementasjonen av de tre logganalyseverktøyene i denne oppgaven har *ikke* gitt resultater som entydig sier at driften på KVT blir mer effektiv med disse i bruk. De kan brukes som del av en løsning, men vil slik de nå fremstår heller føre til at administratorene bruker mer tid på logganalyse, selv om de da også vil kunne komme over flere anomalier og elementer for oppfølging. Hvorvidt dette vil føre til en netto tidsfortjeneste, med tanke på at problemer nå kan oppdages på et tidligere tidspunkt, er på dette grunnlaget umulig å konkludere om.

Appendix

A Konfigurasjon av syslog-ng.conf

A.1 Loggservere

"Options" er felles for alle Loggserverne og er plassert først i syslog-ng.conf på de respektive serverene:

Options

```
#########  global options  ########################
options {
        time_reap(60);
        create_dirs(yes);
  perm(0644);
        dir_owner(www-data);
  dir_perm(0775);
  use_fqdn(no);
        keep_hostname(yes);
        long_hostnames(on);
        use_dns(yes);
        dns_cache(yes);
};
```

A.1.1 Logg1

```
#    Options removed
#########  local settings  ########################
source loghost {
        unix-stream("/dev/log");
        internal();
        udp(ip(0.0.0.0) port(514));
};
source loghost_t {
        unix-stream("/dev/log");
        internal();
        tcp(ip(0.0.0.0) port(5000) max_connections(1000));
};
# backup/control logs
destination localhost {
```

```
        file("/var/log/syslog-ng.all");
};
destination tcplocalhost {
        file("/var/log/tcpsyslog-ng.all");
};
# for sawmill
destination sawmill {
        file("/var/log/sawmill/syslog");
};
# to second loghost
destination loghost2 {
        tcp("158.38.100.31" port (5000));
};
# to third loghost
destination loghost3 {
        tcp("158.38.100.33" port (5000));
};
# backup/control
log {
        source(loghost); destination(localhost);
};
log {
        source(loghost_t); destination(tcplocalhost);
};
# to loghost 2
log {
        source(loghost);source(loghost_t); destination(loghost2);
};
# to loghost 3
log {
        source(loghost);source(loghost_t); destination(loghost3);
};
# to file for sawmill
log {
        source(loghost);source(loghost_t); destination(sawmill);
};
```

A.1.2 Logg2

```
#    Options removed
#########  local settings  ########################
source loghost_t {
```

```
        unix-stream("/dev/log");
        internal();
        tcp(ip(0.0.0.0) port(5000) max_connections(1000));
};
source s_me {
        internal();
        unix-stream("/dev/log");
        file("/proc/kmsg" log_prefix("kern: "));
};
# Destinations
destination localhost {
        file("/var/log/syslog-ng.all");
};
template t_filter {
        template("$FACILITY.$PRIORITY $ISODATE $HOST $MSG\n");
                template_escape(no);
};
destination d_tcplocalhost {
        file("/var/log/tcpsyslog-ng.log" template(t_filter));
};
destination d_splunk {
        file( "/var/log/splunk/$HOST/$YEAR/$MONTH/$DAY.log"
                template(t_filter));
};
# local log (only from this server)
log { source(s_me); destination(localhost); };
# local file from loghost (used as backup/control)
log { source(loghost_t); destination(d_tcplocalhost); };
# local file directory for Splunk
log { source(loghost_t); destination(d_splunk);  };
```

A.1.3 Logg3

```
#    Options removed
#########  local settings  ########################
source loghost {
        unix-stream("/dev/log");
        internal();
        udp(ip(0.0.0.0) port(514));
};
source loghost_t {
        unix-stream("/dev/log");
```

```
        internal();
        tcp(ip(0.0.0.0) port(5000) max_connections(1000));
};
# backup/control logs
destination localhost {
        file("/var/log/syslog-ng.all");
};
destination tcplocalhost {
        file("/var/log/tcpsyslog-ng.all");
};
# for Lire
destination  Lire {
        file("/var/log/lire/syslog");
};
log {
# backup/control
        source(loghost); destination(localhost);
};
log {
        source(loghost_t); destination(tcplocalhost);
};
# to file for Lire
log {
        source(loghost);source(loghost_t); destination(lire);
};
```

A.2 Klienter

```
\sububsection{Generell konfigurasjon}
### Erstatter dhcp med relevant servernavn
source dhcp {
        unix-stream("/dev/log" max_connections(1000));
        internal();
};
destination remote {
        tcp("158.38.100.32" port(5000));
};
log {
        source(dhcp); destination(remote);
};
```

A.2.1 Klient - RADIUS-server

```
options {
        log_fifo_size(1000);
};
source radius-vpn {
        unix-stream("/dev/log" max_connections(1000));
        internal();
};
source s_tail {
        file("/var/log/freeradius/radius.log"
                follow_freq(1)
                flags(no-parse));
        internal();
};
destination remote {
        tcp("158.38.100.32" port(5000));
};
destination d_file {
        file("/var/log/vradius.log" perm(0644));
};
#remote
log {
        source(radius-vpn);
        source(s_tail);
        destination(remote);
};
#local
log {
        source(radius-vpn);
        source(s_tail);
        destination(d_file);
};
```

B Omformatering av loggdata for Lire

Det opprinnelige loggformatet som serveren Lire kjører på ble satt opp med, var syslogformatet som syslog-ng anvender som standard, men lagret i én fil. Det viste seg at Lire måtte ha eksplisitt tilgang på level og facility i loggfilene for å kunne produsere best mulige rapporter, og en endring måtte dermed til.

Studier av dokumentasjonen til Lire viste av et syslogformat anvendt av et loggsystem som heter Kiwi, var det nærmeste i forhold til formatet anvendt av loggserveren Splunk kjører på (Som var det eneste formatet loggene var tilgjengelig på som inkluderte facility og level). Dette formatet har med seg dato inklusiv år og tidsforskjell fra CET, i tillegg til facility, level og loggbeskjed. Dette er formatet på loggene som Splunk anvender:

```
facility.level YYYY-MM-DDTHH-MM:SS+02:00 host Message (FIX)
```

Under er formatet som Kiwi anvender:

```
YYYY-MM-DD-HH-MM-SS facility.level host Message
```

I tillegg til ulik rekkefølge, har altså ikke Kiwi-formatet informasjon om tidsforskjell, og har heller ikke med bokstaven T i tidsformatet.

Det ble laget et lite script som sørger for å rette opp disse forskjellene, slik at rapporten kunne bli generert med full funksjonalitet:

```
#!/bin/bash
awk '{temp = $1; $1 = $2; $2 = temp}' file.log
sed -e 's/T/ /' -e 's/\+01\:00//' -e 's/\+02\:00//' \
file.log >lire.log
```

awk sørger først for å bytte om på de to første ordene, som her er facility.level og tidsstemplet, mens sed fjerner T'en i tidsstemplet samt informasjonen om tidsforskjellen.

Dette er ingen permanent løsning, men det er nok til å kunne generere rapporten på en korrekt måte, så får en permanent løsning utvikles dersom resultatene skulle tilsi et behov for det.

Referanser

[1] Kirk Bauer. *Automating UNIX and Linux Administration*. Apress, 2003.

[2] Ian Eaton. The Ins and Outs of System Logging Using Syslog. *Sans Information Security Reading Room*, 2003.

[3] Thomas A. Limoncelli, Christina J. Hogan, and Strata R. Chalup. *The practice of system and network administration, second edition*. Addison Wesley, 2007.

[4] John P. Rouillard. Real-time logfile analysis using the Simple Event Correlator (SEC). In *18th USENIX System Administration Conference (LISA -04) Proceedings*, November 2004.

[5] Abe Singer and Tina Bird. *Building a Logging Infrastructure*. The USENIX Association, 2004.

[6] Jon Stearley. Towards Informatic Analysis of Syslogs. In *IEEE Conference on Cluster Computing*, September 2004.

[7] Indian Computer Emergency Response Team. Implementation of Central Logging Server using syslog-ng. *CISG*, July 2004.

[8] webside. *The Art of Unix Usability*. catb.org, 2009. `http://www.catb.org/ esr/writings/taouu/html/ch02s01.html`.

[9] webside. *Personvern og informasjonssikkerhet*. Datatilsynet, 2009. `http://www.datatilsynet.no/templates/article____407.aspx#19`.

[10] webside. *Sawmill Homepage*. Sawmill, 2009. `http://www.sawmill.net`.

[11] webside. *Splunk homepage*. Splunk, 2009. `http://www.splunk.com`.

[12] webside tekst. *RFC 3164*. The Internet Engineering Task Force, 2009. `http://www.ietf.org/rfc/rfc3164.txt`.

[13] website. *Article on Syslog*. Wikipedia - The Free Encyclopedia, 2009. `http://www.wikipedia.com/syslog`.