

DA-NET2002 Nettverk, sikkerhet og
datakommunikasjon

Prosjekt2

Gruppe 9

Gruppe prosjekt



Innhold

1. Innledning.....	3
2. Ordliste.....	4
3. Problemstilling	5
3.1 Nettverk	5
3.2 DNS	5
3.3 DHCP	5
3.4 Webserver.....	5
3.5 Epost.....	6
3.6 Brannmur	6
3.7 HIDS.....	6
4. Programvalg	6
4.1 Webserver Apache	6
4.2 Hvorfor Apache.....	7
4.3 Mail service (MTA) – Postfix	7
4.4 Hvorfor Postfix?	8
4.5 Brannmur – Firewalld	8
4.6 Hvorfor Firewalld?	9
4.7 DNS - Bind9	9
4.7.1 Adresseløsningsmekanisme	10
4.7.2 Rekursiv og hurtigbufret navneserver	10
4.7.3 DNS-resolver	10
4.7.4 Registrer caching.....	10
4.7.5 Omvendt oppslag	11
4.7.6 Klientoppslag.....	11
4.7.7 BIND.....	11
4.8 Hvorfor Bind9?	12
4.9 HIDS – OSSEC.....	12
4.10 Hvorfor OSSEC?	12
4.11 IMAP – Roundcube	13
4.12 Hvorfor Roundcube?	13
5. Valg av oppsett og struktur.....	14
5.1 Utstyr.....	14
5.2 Systemer.....	14
5.3 Oversikt:	15
5.4 Webserver.....	15

5.4.3 Valg av CMS.....	16
5.5 Mailserver.....	17
5.6 Gateway	17
5.7 DMZ	18
5.8 VLAN Tagging.....	18
5.9 DNS	20
5.10 HIDS og brukere.....	20
5.11 HIDS.....	20
5.12 Brukere	20
6. Konklusjon.....	21
7. Referanser.....	22
8. Vedlegg.....	25

1. Innledning

Denne rapporten baserer seg på oppsett av et nettverk for en bedrift ifm. av et skoleprosjekt.

Det vil være en stegvis gjennomgang av det teoretiske samt fysiske oppsettet av utstyret for gjennomføringen av oppgaven. Av fordelaktige årsaker vil rapporten ikke inneholde kommandoer/syntax'er, da dette i utgangspunktet ikke vil gi en økt forståelse for utføringen av oppsettet eller tilføye ytterligere nødvendige opplysninger. Det har blitt lagt vekt på beskrivelse av programmer og servicer nødvendig for utføring av oppgaven. Disse blir i all hovedsak beskrevet i henhold til funksjonalitet i et nettverk generelt, og deres hensikt ved implementasjon i dette prosjektet. Illustrasjoner ifm. av bilder vil gi en bedre visuell forståelse av oppsett, og fremheve de aktuelle enheter og tjenester som videre blir nevnt. Formålet med denne rapporten er å gi en grunnleggende forståelse for et oppsett av et lokalt nett med tilknytning til internett. Hvordan dette kan settes opp, og hvilke utfordringer som følger med. Hvor bedriftens sikkerhet ivaretas, men samtidig også tillater både brukere intern, og ekstern å kommunisere og å ha tilgang i selektive soner av nettverket, slik som dette prosjektet har tatt utgangspunkt i. Det er på forhånd utarbeidet en gruppe-kontrakt til utførelsen av prosjektet for konkretisering, og enighet om hver enkelt gruppemedlem sitt ansvar for arbeidsoppgaver, oppmøteplikt og bidrag.

2. Ordliste

DMZ	Demilitarized Zone
HIDS	Host Intrusion Detection System
NIDS	Network Intrusion Detection System
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
NIC	Network Interface Controller
CMS	Content Management System
MTA	Mail Transfer Agent
SMTP	Simple Mail Transfer Protocol
FQDN	Fully Qualified Domain Name
MX	Mail Exchange Record
GUI	Graphic User Interface

3. Problemstilling

Problemstilling er tatt ut ifra oppgavebeskrivelse:

Dette er et prosjekt som skal simulere en mellomstor bedrift med diverse tjenester som bedriften trenger. Dette er et prosjekt som skal gjennomføres med maskiner på studentserverrom D3-101b eller hjemmefra. Hver gruppe(4 personer)får tildelt 3 maskiner + 2 stk. RaspberryPi eller tilsvarende som skal brukes i prosjektet. I prosjektet skal disse settes opp med Linux med valgfri distribusjon. Alle servere skal ha høy sikkerhet og være oppdatert til siste versjon av programvare. Tjenester er beskrevet under.

3.1 Nettverk

Bedriften skal bruke IP versjon 4 og har ikke mulighet for å ha mere enn en publik IP adresse. Det vil si at innenfor bedriftens gateway skal det brukes private IP adresser. Nettverket skal segmenteres i to soner, en sone for DMZ og en for sikker sone. Bedriften skal ha en fleksibel løsning og det skal tas i bruk VLAN tagging (802.1q) av pakker. Bedriftens gateway skal ha brannmur med høy sikkerhet som vil si at det er kun pakker som skal til bedriftens public tjenester i DMZ zone som skal kunne komme inn her. Disse vil da bli transportert inn i bedriftens nettverk via NAT/NAPT funksjon i gateway. Mellom DMZ og sikker sone skal det være brannmur som hindrer pakker å komme inn til sikker sone. Oppkobling som er startet fra sikker sone skal få svar tilbake. Sikker sone skal ha dynamisk tildelt IP adresser fra DHCP mens DMZ sone skal ha statisk tildelte adresser fra DHCP.

3.2 DNS

DNS skal inneholde sone for eksterne(internett)og interne (bedriftens private nettverk) forespørsler. Det vil si at alle som er på bedriftens nettverk skal få intern IP, og alle fra internett skal få bedriftens offentlige IP adresse. Det skal være oppslag for alle bedriftens tjenester i DMZ i tillegg til reversoppslag. For at klienter og servere i bedriften skal kunne bruke internett må DNS serveren kunne slå opp eksterne navn/ekstern DNS.

3.3 DHCP

DHCP skal tildele nettverksinformasjon til sikker sone, slik at klienter kan koble seg på nettverket med dynamiske innstillinger på klienten. DHCP skal tildele statiske IP adresser til servere i DMZ sone.

3.4 Webserver

Webserveren skal ha funksjon som bedriftens hjemmeside, som skal være en dynamisk side av typen CMS (content management system). Webserveren skal også ha webløsning for bedriftens e-post.

3.5 Epost

Bedriften skal ha mulighet for å ta imot og sende e-post fra egen server. Det skal kun brukes webløsning for e-post.

3.6 Brannmur

Brannmurer i bedriften skal ha høy sikkerhet.

3.7 HIDS

Alle servere i bedriften skal ha HIDS installert og varsle ansvarlig person via e-post når det blir utført ureglementert aktivitet på serveren. Denne tjenesten skal ha en felles server hvor alle loggede aktiviteter blir samlet (agent/server funksjon).

4. Programvalg

4.1 Webserver Apache

Apache er en gratis webserver service, med open source som lever innhold til klienter over Internett igjennom klientenes web utforskere som Firefox, Chrome, Edge osv. Apache er ikke en fysisk webserver men en service som kjører på en server. Det er regnet ut at Apache står for omtrentlig 34,4 % av alle webservere som er i bruk i dag i følge w3techs.com per 19.05.2021. [1]

Apache var lenge alene på tronen, men mange utfordrere kommer med nye mer brukervennlige webserver applikasjoner. For kun noen få år siden lå denne markedsandelen på nesten 70 %. Apache fikk navnet i en hyllest til de innfødte i USA, selv om det er vanlig og tro at det kommer fra at det er ”a patchy” webserver. Apache er skrevet i C språket.

I februar 1995 var den mest populære serverprogramvaren på nettet HTTPdaemonen for offentlig domene utviklet av Rob McCool ved National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

I mellomtiden hadde utviklingen av HTTPD stoppet etter at Rob forlot NCSA i midten av 1994, og mange nettredaktører hadde utviklet sine egne ”extensions” og ”bug fixes” som hadde behov for en felles distribusjon.

En liten gruppe av disse nettredaktørene, samarbeidet via privat e-post, de samlet seg for å koordinere endringene i form av ”patches”. Brian Behlendorf og Cliff Skolnick satte sammen en adresseliste, delt informasjonsplass og pålogginger for kjerneutviklerne på en maskin i Californias Bay Area, med båndbredde donert av HotWired. I slutten av februar utgjorde åtte hovedbidragsyttere grunnlaget for den opprinnelige Apache Gruppen.

4.2 Hvorfor Apache

Som vår webserver så valgte prosjektgruppa å bruke Apache. prosjektgruppa så også på NGINX og LightHTTPD, men prosjektgruppa så bort ifra LIGHTHTTPD da prosjektgruppa leste at det var optimert med tanke på tunge applikasjoner som CGI baserte utvidelser (noe prosjektgruppa ikke kommer til å bruke) og at det var svært brukervennlig. [2]

Det høres kanskje rart ut at prosjektgruppa valgte den webserver som er regnet som den webservern som er minst brukervennlig, men siden hovedmålet med dette prosjektet faktisk er læring for vår egen del så resonerte prosjektgruppa oss frem til at hvis prosjektgruppa kan lære oss å installere og bruke Apache så vil være mye bedre rustet til fremtidige prosjekter og kan da velge hvilken som helst webserver med god samvittighet.

Vi så også på NGINX og selv om Apache og NGINX som kanskje er de 2 likeste og største webserverne på markedet nå. Selv om de er svært like, så er det noen viktige forskjeller, Arkitekturen er grunnleggende forskjellig, Apache lager en ny "tråd" for hver forespørsel, mens NGINX kan behandle flere forespørsler på samme "tråd". Det er mange forskjeller, men de som gjorde at utfallet falt på Apache var at Apache støtter alle UNIX systemer, noe NGINX ikke gjør, at Apache er kjent for å være den sikreste av alle webservere og igjen det at Apache er den som vil gi oss mest læringsutbytte da NGINX er mer brukervennlig. [3]

Apache er ikke like lett å konfigurere, men er til gjengjeld den er jevnlig oppdatert den mest fleksible av de 3, når det kommer til tilpassninger. Med tanke på læringsmål og tanken bak prosjektet samt at alle 3 gruppene som jobbet sammen på dette prosjektet hadde erfaring med Apache og disse var utelukkende gode følte prosjektgruppa seg trygg på at Apache var det rette valget. Det var det disse tingene som veide tyngst og som gjorde at valget falt på Apache. [4]

4.3 Mail service (MTA) – Postfix

MTA er en programvare som overfører e-post meldinger fra en pc til en annen ved hjelp av SMTP.

Postfix er en open-source MTA som dirigerer hvor e-posten skal gå og leverer den. Postfix ble originalt skrevet i 1997 men har fått regelmessige oppdateringer. [v2]

Som en SMTP-server så iverksetter Postfix første laget av forsvar mot spambots og malware. Man kan kombinere Postfix med andre programmer som for eksempel Amavisd-new som gir spam/virus filtrering. Man kan også slå den sammen med Dovecot for å lagre e-postene på serveren. Som en SMTP Client iverksetter Postfix en høytytelse parallellisert leveringsmotor. [5]

En enkelt Postfix-forekomst har blitt målt på 300 meldinger sendt per sekund over internett, og kjørt på en gjennomsnittlig maskinvare.

E-post systemer som Postfix oppnår høy ytelse ved å levere post i parallelle økter, mens Sendmail og Exim som oppretter en forbindelse om gangen kan oppnå høy ytelse ved å sende begrensede partier med epost parallelt, slik at hvert parti leveres ved en annen prosess. Postfix krever parallell innsending i forskjellige MTA-forekomster når de når sin egen ytelse

grense, eller ytelsesgrensen til maskinvare eller operativsystemet. Leveringshastighetene som er sitert ovenfor er i stor grad teoretiske. Ved massepostlevering bestemmer den sanne leveringsfrekvensen primært av mottakerens policy for mottak av e-post og av avsenderens omdømme.

Postfix består av en kombinasjon av serverprogrammer som kjører i bakgrunnen, og klientprogrammer som påkalles av brukerprogrammer eller av systemadmin.[6]

Postfix kjernen består av flere dusin av serverprogrammer som kjører i bakgrunnen, og hver håndterer ett spesifikt aspekt ved leveringen av e-post. For sikkerhet og skadekontrollformål kjører de fleste serverprogrammene med begrenset privilegier, og avsluttes frivillig etter behandling av et begrenset antall forespørsler. For å spare på ressursene i systemet blir de fleste serverprogrammene avsluttet når de er inaktive.

4.4 Hvorfor Postfix?

Av Postfix Exim og Sendmail er Postfix den mest sikre. Sendmail kan ikke bli sett på som sikker. Postfix ble opprinnelig designet for å redusere sårbarhetene til Sendmail. Exim er ganske sikkert i de fleste tilfellene, men taper til Postfix. En god konfigurert Postfix gir forberedt forsvar mot spam, misbruk og lekkasje av sensitive data.

Sendmail er kjent for å være ineffektiv sammenliknet med konkurrenter. Både Postfix og Exim er erstatninger for Sendmail og er nesten like når det gjelder pålitelighet. Imidlertid er Postfix et skritt foran fordi den har en modulær arkitektur. Den består av uavhengige systemdeler som kan byttes ut ved feil, noe som gir et høyere nivå av pålitelighet.[7]

Vi valgte Postfix fordi det er den mest sikre av de prosjektgruppen så på. Exim er mer konfigurert, men prosjektgruppa skulle ikke konfigurere så mye i dette prosjektet og derfor ble dette ikke en prioritet. Postfix er også mer pålitelig enn Exim og Sendmail, det kunne like gjerne blitt Exim ettersom det er mest konfigurert og siden ingen av oss hadde brukt det før så ville det hatt bedre læringsutbytte, dessverre rant tiden ut for oss og vi endte med å bruke Postfix siden vi hadde brukt det før, var svært fornøyd med sist og siden det er den MTA'n med høyest sikkerhet så falt valget på Postfix også denne gang. [8] [9]

[v3]

4.5 Brannmur – Firewallld

Brannmur er en programvare som beskytter nettverk mot uønsket kommunikasjon. Oppgaven til brannmuren er å kontrollere og tilpasse trafikken mellom nettverk som har ulike tillitsforhold. [10] [11]

Firewalld er en dynamisk styrt brannmur med støtte for nettverk og brannmur soner som kan definere tillitsnivået for nettverkstilkoblinger eller interface. Firewalld støtter IPv4, IPv6 brannmuringstillinger, Ethernet-broer og IPsett.

Det er et skille mellom mulighetene for kjøretid og permanente konfigurasjoner. Dette gir også et interface for tjenester eller applikasjoner for å legge til brannmurregler direkte.

Det er flere fordeler med å bruke firewalld. Mens programmet kjører kan endringer bli utført umiddelbart. Det trenger ikke å restarte daemon eller service som kjører. med firewalld D-bus

interface er det lett for services, applikasjoner og brukere å tilpasse brannmurinnstillinger. Interfacen er komplett og brukes til brannmur konfigurasjonsverktøyene firewall-cmd, firewall-config og firewall-applet.

Separasjonen av runtime og permanente konfigurasjoner gjør at det er mulig å gjøre endringer og teste om disse endringene er bedre før man gjør de permanente. De blir ikke permanente og vil fjernes neste gang service reload og restart eller når systemet rebooter. Da går innstillingene tilbake til de permanente innstillingene. Med runtime miljøet er det mulig å bruke innstillinger for en begrenset periode. Hvis disse innstillingene ble fullført og man er fornøyd med dem, kan man gjøre de permanente.

Sette opp tillatelse for http og gjøre den permanent [v4][12]

4.6 Hvorfor Firewallld?

Vi så på flere andre brannmurer enn Firewallld, deriblant Iptables. Iptables er et program/applikasjon som tillater en bruker til å konfigurere en brannmur sikkerhetstabellen av Linux kjerne brannmur og kjedene, sånn at en bruker kan legge til eller fjerne brannmur regler som passer til brukerens behov. Iptables blir brukt for IPv4 og ip6tables blir brukt for IPv6 for begge tcp og udp. Man må ha root for å konfigurere Iptables reglene i brannmuren. Alle reglene i Iptables håndteres direkte av Linux-kjernen, dette er kjent som Kernel duty. Det har ikke noe å si hvilket GUI verktøy eller andre sikkerhetsverktøy kan brukes til konfigurere serverens brannmur, siden alle innstillingene konvergerer til Iptables regler og sendt til kjerner for å utføre operasjonen. [13]

Vi valgte Firewallld fordi den er dynamisk og dette gjør at prosjektgruppa kan gjøre endringer og sjekke om disse fungerer for å så lagre disse. Prosjektgruppa er ganske nye i nettverk og selv om vi har lært mye gjennom prosjekt 1 og prosjekt 2 og så er det fortsatt mye vi ikke kan, det er lett å sette seg fast hvis vi ikke finner svaret i litteraturen eller på nettet så prosjektgruppa ønsket ikke å ha mulighetene til å låse oss selv ute og måtte begynne på nytt. Vi så også at gjennom prosjekt 1 (og merket det også under prosjekt 2) at det er mye forsøk på datainnbrud , på det meste hadde vi over 250 forsøk på noen få minutter. Det å kunne tilpasse brannmuren kjapt og til gruppens behov sto høyt. Vi hadde alle gode erfaringer med Firewallld fra prosjekt 1.

4.7 DNS - Bind9

DNS står for “Domain Name System” og er navnetjenerstandarden spesifisert i TCP/IP protokollsuiten. DNS er internettjenesten som kobler domenenavn sammen med IPadressen til en tjener på internett, dette gjør det mulig å sende informasjon til det riktige stedet på internett. [14]

Operasjonene til DNS er Adresseløsningsmekanisme, Rekursiv og hurtigbufret navneserver, DNS-resolver, Registrer caching, Omvendt oppslag, Klientoppslag.

4.7.1 Adresseløsningsmekanisme

Domenenavn resolver bestemmer domenenavnservere som er ansvarlig for det aktuelle domenenavnet. Dette blir gjort ved en sekvens med spørsmål fra den høyeste domene etiketten. For at operasjonen skal bli gjort riktig av domenenavn resolveren er en nettverkskonfigurert vert med en innledende hurtigbuffer av de kjente adressene til rot navneserverne.

[28] [30] [14]

4.7.2 Rekursiv og hurtigbufret navneserver

Teorien er autoritative navneservere tilstrekkelig for driften av internett. Imidlertid med bare autoritative navneservere som opererer må alle spørsmål fra DNS starte ed rekursive spørsmål i rotsonen til DNS. Alle brukersystemer må iverksetter resolver-programvare som er i stand til rekursiv drift. For at effektiviteten skal bli bedre og redusere DNS trafikk over internett og øke ytelsen støtter DNS cache servere som lagrer DNS søkeresultater i en periode som er bestemt i konfigurasjonen av den aktuelle domenenavneposten.

[28] [30] [14]

4.7.3 DNS-resolver

Klientsiden av DNS er det som kalles DNS resolver. En resolver er ansvarlig for å initiere og sekvensere spørsmål som til slutt fører til en full oppløsning av den søkte ressursen. Dette kan være å oversette fra domenenavn til en IPadresse. DNS resolver er klassifisert etter en rekke spørring metoder, som for eksempel rekursiv, ikke-rekursiv. En resolver prosess kan bruke kombinasjoner av disse metodene.

I en ikke-rekursivt spørsmål spør DNS resolver om en DNS-server som gir en post som serveren er autoritativ for, eller den kan gi et delvis svar uten å spørre andre servere. I et tilfelle en cache DNS resolver gir den ikke rekursive spørringen til den lokale DNS cacheten et resultat og reduserer belastningen på oppstrøms DNS servere ved å cache DNS ressursposter i en periode etter et første svar fra oppstrøms DNS servere.

I et rekursivt spørsmål spør en DNS resolver om en enkelt DNS server. Som igjen kan spørre andre DNS-servere på vegne av rekvirenten. Dette kan være en enkelt stub resolver som kjører på en hjemmerouter.

[28] [30] [14]

4.7.4 Registrer caching

Vanligvis for implementering av navneløsning i applikasjoner er å redusere belastningen på DNS serveren ved å cache søkeresultatene lokalt. Søkeresultatene som er oppnådd fra DNS forespørsel har alltid en TTL (Time to live) som sier hvor lenge den kan leve i cache og være lette å finne og sleppe i søke opp i DNS for å finne. Etter TTL må resultatene oppdateres eller

kastes. TTL er satt av admin av den autoritative DNS serveren. TTL kan vare fra noen sekunder til dager eller uker.

Denne distribuerte cache arkitekturen vil gi et resultat som gjør at endringer ikke skjer i hele nettverket når noen endringer skjer hos en enkelt. Dette krever at alle hurtigbufferne utløper og oppdateres etter TTL.

[28][29][30][31][14]

4.7.5 Omvendt oppslag

En omvendt DNS oppslag er et spørsmål om DNS for domenenavn når IPadressen er kjent. Flere domenenavn kan være tilknyttet en IPadresse. DNS lagrer IPadresser i form av domenenavn som spesielt formaterede navn i peker poster innenfor infrastrukturen toppnivå domenenavn ARPA. For IPv4 er det omvendte oppslagsrommene ip6.arpa. IPadressen er representert som et navn i omvendt ordnet oktettrepresentasjon for IPv4.

Når et omvendt oppslag skjer konverterer DNS-klienten adressen til disse formatene før den spør etter navnet på PTR-posten etter at det har blitt delegert som DNS spørsmål.

[28] [30] [14]

4.7.6 Klientoppslag

Brukere kommuniserer vanligvis ikke direkte med DNS resolver. Det som skjer, er DNS oppløsning usynlig i applikasjoner som nettlesere og andre internettapplikasjoner som epostklienter. Når noe gir en forespørsel som f.eks et program ønsker tilgang til en nettside eller noe som krever domenenavnsoppslag sender slike forespørsler til DNS resolveren i det lokale operativsystemet som håndterer den kommunikasjonen. DNS oppleseren vil nesten alltid ha cache som har innhold. Hvis cachen kan gi svaret på forespørselen vil resolveren returnere verdien i cachen uten å gå videre i DNS serveren. Hvis den ikke inneholder svaret på forespørselen, vil den sende forespørselen videre til en eller flere DNS servere.

[28] [30] [14]

4.7.7 BIND

Bind står for «Berkeley Internet Name Domain» var designet i tidligere 1980 tallet på universitetet i California Berkeley. Det ble skrevet av fire studenter, BIND blir fortsatt oppdatert og er ofte betraktet som standard konvensjonell DNS-server. [v4] BIND9 er en stor omskriving av nesten alle aspekter av den underliggende BINDarkitekturen. Noen av de viktige funksjonene i BIND9 er DNSsikkerhet(DNSSEC,TSIG), IPv6, protokollforbedringer (IXFR, DDNS, DNS Notify, EDNS0), Views, Multiprocessor Support og en forbedret bærbarhetsarkitektur. Bind9 er et fleksibelt, fullverdig DNS system. BIND9 har det fleste nødvendige funksjonene til de fleste applikasjonene. BIND9 er gjennomskiktig åpen kildekode, lisensiert under MPL2.0 lisensen. Unbound Unbound er en gratis open-source validering, rekursiv, hurtigbufret DNS resolver programvare under BSD-lisensen. Det er et

nylig utviklet DNS-system som kom inn i DNS-rommet for å gi et raskt og magert system som inneholder moderne funksjoner basert på åpne standarder.[15]

4.8 Hvorfor Bind9?

Response Rate Limiting RRL er en funksjon som blir brukt i autoritative navneservere som fungerer som et avbøtende verktøy for problemet med DNS-forsterkningsangrep. Dette gjør at DDOS (Distributed Denial of Service) blir mindre betydelig på klientmaskiner. DNS Security Extension DNSSEC er en funksjon som gir primært opprinnelse godkjenning av DNS-data, godkjent nektelse av eksistens og dataintegritet. Unbound støtter DNS-over-TLS som lar klienter kryptere kommunikasjonen. Videre støtter det forskjellige moderne standarder som begrenser mengden data som utveksles med autoritative servere, Disse moderne standardene inkluderer Query Name Minimization, Aggressive Use of DNSSEC-Validated Cache og støtte for autoritetssoner, som kan brukes til å laste en kopi av root zone. Vi valgte BIND9 fordi den har mange funksjoner som en DNS-server. Den har forsvar mot DDOS og er ganske sikker. Unbound er også sikker ved at man kan kryptere meldingene. BIND9 har også de fleste funksjonene man trenger til de fleste programmene prosjektgruppa brukte. [16] [17] [18]

4.9 HIDS – OSSEC

HIDS står for “host-based intrusion detection system” . Det er en applikasjon som overser en data eller et nettverk for mistenkelig virksomhet/utførelser både gjennom inntrengere utenfra samt misbruk av tillatelser, ressurser og data av interne agenter, enten i form av uregelmessigheter fra brukere og/eller prosesser eller en mismatch i filer og sertifikater ved å konstant overvåke nettverket. [19]

OSSEC er kort for “Open Source Security Event Correlator” og er den mest brukte HIDS per i dag, den er som den sier, open source og dermed gratis. OSSEC har også en server til agent modell noe som betyr at en dedikert server gir analyse og aggregering for hver host. Den har et svært godt loggsystem som gir brukerne god og tilgjengelig oversikt over uregelmessigheter i systemet i sanntid. [20][21]

4.10 Hvorfor OSSEC?

Vi vurderte et par andre HIDS en OSSEC, men som gratis open source var OSSEC flere hestehoder foran de andre som var Tripwire og Wazuh. Ulempen med OSSEC er at alle regler blir satt tilbake når en oppdaterer, men bortsett fra det har den alle de samme gode egenskapene som de 2 andre.

Dessverre så er ulempene til de to andre for store til å overse, Tripwire gir deg ikke beskjed i sanntid, noe som betyr at du må gå inn i ettertid av et angrep eller lignende og så rette opp i feil eller mangler istedenfor å kunne iverksette tiltak med en gang en merker at en inntrenger prøver å ta seg inn i systemet. Da kan skaden allerede være for stor.

Med Wuzah som har samme opprinnelse som OSSEC, det er faktisk en fork av OSSEC, er det ikke store forskjellene, men Wuzah kan være vanskelig å få server installasjon delen og

API til og fungere. Så valget her var egentlig om det var mellom 2 serviser med store likheter og relativt små svakheter, siden prosjektgruppa lett kan komme forbi ulempene i OSSEC ved å ferdigstille reglene i OSSEC i en annen config fil og kopiere reglene tilbake inn etter oppdatering er denne ulempen nesten lik null. Utover dette er disse 2 servisene tilnærmet like i utførelse og arkitektur. [22]

4.11 IMAP – Roundcube

Roundcube er en webbasert e-post klient som er kjent for sin gjennomgående bruk av AJAX teknologi. Roundcube er et gratis open-source programvare.

Roundcube er en webbasert e-post klient som er kjent for sin gjennomgående bruk av AJAX teknologi. Roundcube er et gratis open-source programvare. Ajax er en forkortelse for "Asynchronous JavaScript and XML") er et sett med webutviklingsteknikker som bruker mange webteknologier på klientsiden for å lage asynkrone webapplikasjoner. Med Ajax kan webapplikasjoner sende og hente data fra en server asynkront (i bakgrunnen) uten å forstyrre visningen og til den eksisterende siden. Ved å koble datautvekslingslaget fra presentasjonslaget, tillater Ajax websider og, webapplikasjoner å endre innhold dynamisk uten å måtte laste hele siden på nytt. Ulempen med dette er at hvis brukerens nettleser ikke støtter JAVA eller HTML så vil brukeren få en redusert opplevelse eller ikke få lastet siden slik den er ment til å vises. [23]

4.12 Hvorfor Roundcube?

Når prosjektgruppa gjorde litt research for å finne ut hvilken webbaserte e-post klient prosjektgruppa ville bruke så endte prosjektgruppa med 3 hovedkandidater, Horde, Squirrelmail og Roundcube akkurat som i prosjekt1. Squirrelmail, Horde og Roundcube blir beskrevet slik:

Squirrelmail er for brukere som kun vil lese og sende e-post. Den har ikke støtte for HTML. [24]

Horde er for "high end" brukere som trenger full tilgang til utvidet "features", avanserte verktøy og mobil tilgang til e-post.

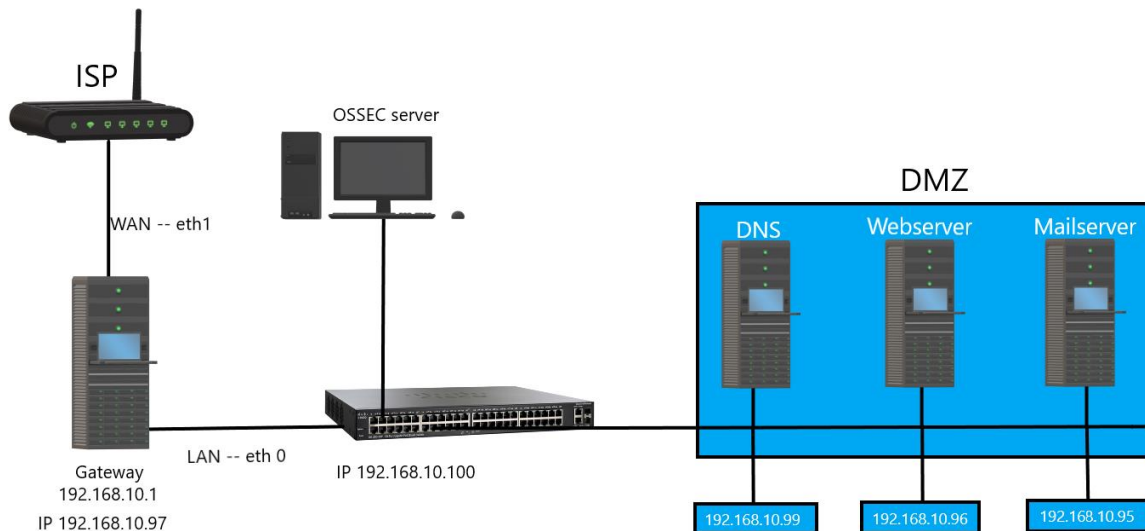
Roundcube er en brukervennlig e-post interface som med en bred tilgang på tilleggsmuligheter og utvidelser, dog ikke like mye som Horde.

Etter kort tid så ble Squirrelmail ute av ligningen da den var rett og slett for nedstrippet, den er så "basic" som en kan få det, når prosjektgruppa skulle få læringsutbytte av dette og mulighet til å lage en side med HTML og andre "features". Da gjensto kun Roundcube og Horde igjen. Disse har begge to mange muligheter og "features".

Utover dette hadde begge kalendere, adressebok søkefunksjoner, "drag and drop" funksjoner for å endre strukturen på mappeoppsettet for brukere og det meste annet prosjektgruppa er vant med i en e-postservice. [25][26]

Vi så ikke behovet for å ha tilgang på mobil selv om det kunne vært fin læring så var det overflødig for denne oppgaven dermed falt valget på Roundcube denne gang.

5. Valg av oppsett og struktur



5.1 Utstyr

Til utføring av oppdraget, ble hver gruppe tildelt nødvendig utstyr som bestod av: 2 stk. Raspberry Pi, 3 stk. pc'er, 1 stk. cisco switch og tilhørende nødvendige kabler for strøm og oppkobling. Gruppene fikk selv velge enhetenes formål og funksjonalitet i nettverket, samt operativsystemene som skulle installeres.

5.2 Systemer

Valget av operativsystem for pc'er ble gjort i form av en avstemning i gruppa, og ble bestemt til å være CentOS 8 da medlemmene allerede var kjent med syntakser og kommandoer for systemet. Dette var dog ikke tilfellet for Raspberry Pi, da installasjon av CentOS på disse viste seg å være en større utfordring enn antatt grunnet programvare for installasjon. Det ble derfor bestemt at gruppen ville gå for det ordinære operativsystemet for Raspberry Pi, Raspbian, da dette ikke ville ha noen innvirkning på systemets funksjonalitet, grunnet tilstrekkelig likhet i samtlige linux distribusjoner.

Ettersom utførelsen av prosjektet ble gjort hjemmefra (grunnet covid 19), viket strukturen av nettverket og oppsettet noe fra foreleserens fremgangsliste for innstillinger. Det bes derfor ta hensyn til at enkelte detaljer oppgitt ifm. av IPadresser, gateway mm. ikke fremstilles ifølge dokumentasjon gitt til studenter for oppsett.

5.3 Oversikt:

Oppsettet av nettverket ble i felles enighet i gruppen satt til å se følgende ut:

Raspberry Pi 1: Webserver med Apache underliggende [IP 192.168.10.96]
Raspberry Pi 2: Mailserver med Roundcube underliggende [IP 192.168.10.95]
Pc 1: DNS med BIND underliggende for drift av DNS tjenester [IP 192.168.10.99]
Pc 2: HIDS med OSSEC underliggende for sikkerhet og varsling [IP 192.168.10.98]
Pc 3: Gateway med brannmur for oppsettet [IP 192.168.10.97]

Cisco switchen fikk tildelt IPadresse 192.168.10.100 for oppsett. Trafikken ble konfigurert til å rutes via Vlan 2 og alle endringer ble utført ved bruk av terminal emulatoren Putty, hvor gruppene fikk tildelt fremgangsmåte for konfigureringer.

DHCP tildelingen fra hoved-router (ISP) ble manuelt endret til å tildele IPadresser i ønsket område 192.168.10.101 – 192.168.10.199 for å unngå uønskede konflikter i utstyr med statisk ip, som senere skulle kobles til via egen gateway. Tanken bak dette var å separere utstyr ment for prosjektet og private enheter ved å tildele statiske ip'er til utstyr som skulle settes i DMZ-sonen, mens utstyret på LAN (internt i bedriften og hjemmet) ville få tildelt dynamiske ip'er ved bruk av DHCP.

5.4 Webserver

For oppsett av webserver ble Apache valgt til å drifte innholdet på serveren. Systemet ble installert på Raspberry Pi med Raspbian som OS. Det ble satt opp Omeka Classic som CMS for webserver med port 80 konfigurert til å håndtere forespørsler på severen eksternt.

Databasen for Omeka konfigures til slutt for serveren, og siste del av installasjonen foregår i nettleser for oppkobling av DNS.

En webserver er Hardware og software som bruker HTTP (Hypertext Transfer Protocol) og andre protokoller for å svare på forespørsler fra Internett. En webserver har som hovedoppgave å vise innhold på nettsider gjennom lagring, behandling og levering av nettsider til brukere. Det finnes to typer nettsider: statiske og dynamiske. Statiske nettsider er de som er løst og viser det samme innholdet for hver bruker, vanligvis utelukkende skrevet i Hypertext Markup Language (HTML). Et dynamisk nettsider, derimot, er et som kan vise forskjellig innhold og gi brukerinteraksjon, ved å bruke avansert programmering og databaser i tillegg til HTML.

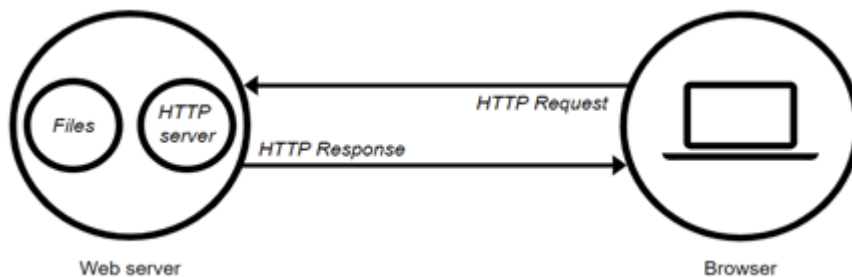
For å etablere en forbindelse med en webserver, trenger man en nettleser. En nettleser er et verktøy som kommuniserer med en webserver gjennom en effektiv HTTP designet for å arbeide med hypertekst- og hypermediedokumenter som kan inneholde ren tekst, bilder, lyd

og video. Websider er skrevet i HTML og lagres vanligvis i filer med suffikset (filename extension) .html eller .htm. Alt innholdet i et hypermedia-dokument er lagret på datamaskiner som kan være en vanlig datamaskin eller server. [33]

Webserver svarer forespørselene på en av de to følgende måter:

- Sende filen til klienten som er tilknyttet den forespurte URL
- Genererer respons ved å påkalle et skript og kommunisere med databasen

Webserveren til bedriften skal ha IP 192.168.10.95 og kjøres på en av datamaskinene i DMZ sone. For at alle forespørsler fra eksterne maskiner på port 80 skal kunne sendes til webserveren, legges port forwarding inn i brannmuren. forespørsler på port 80 vil bli sendt videre til bedriften hjemmeside.



Demonstration of how web server works.[33]

Vi skal ha en bedriftsside slik at den blir dirigert til selskapets side ved forespørsler i stedet for apache-testsiden. Hjemmeside til bedriftens skal bare en dynamisk side av typen CMS (Content management system) med webbasert mail server.

5.4.1 Hva er CMS?

Et Content Management System, er et Graphic User Interface som gjør interaksjon med nettsider database brukervennlig. [34]

5.4.2 Hvorfor CMS?

- Brukervennlig
 - Raskere oppdateringer
 - Støtte flere brukere
 - Sikkerhet
 - Lett å vedlikeholde
- [34]

5.4.3 Valg av CMS

Apache CMS-systemet er ikke lenger åpent for nye nettsideprosjekter pga det, valgte vi Apache Lisens åpen kildekode Amethyst CMS. det finnes flere alternativer som kunne støtte apache, men amethyst har den beste GUI og er brukervennlig og letter å komme i gang med.

5.5 Mailserver

E-postserver er en digital posttjeneste. Det er en maskin eller applikasjon som er ansvarlig for håndtering av meldinger. Med andre ord mottar og leverer en e-postserver e-post

For oppsett av mailserver, ble det i likhet med webserver, valgt en Raspberry Pi med Raspbian OS til dette. Ut over ordinære innstillinger og konfigurasjoner for Postfix, Dovecot, SmtP og POP3, ble løsningen for webmail konfigurert til å driftes ved bruk av Roundcube. Programvaren ble satt opp med 4 brukere i databasen, som representerer gruppemedlemmene i gruppe 9.

Mailserveren til bedriften har IP 192.168.10.96 og kjører på en av datamaksinene i DMZ sone. SMTP bruker TCP og sender trafikken gjennom port (443).

Først sender avsenderen e-postadressen til mottaker med meldinger ved hjelp av et e-postprogram. Når avsenderen har trykket på send knappen, vil e-posten gå til MTA. Denne kommunikasjonen gjøres via SMTP-protokoll.

Når avsenderen sendes en e-post, sender systemet en forespørsel om å finne ut den tilsvarende MTA (Mail Transfer Agent) for mottakeren. Dette vil bli gjort ved hjelp av MX-record. I DNS-sonen vil det være en MX-record for mottakeradressens domene. Dette er en DNS-record som spesifiserer e-postserveren til et domene. Så, etter DNS-oppslaget, blir det gitt et svar til den forespurt e-postserveren med IP-adressen til mottakerens e-postserver. På denne måten identifiseres 'til' e-postserveren. SMTP-protokollen brukes til å overføre meldingene mellom e-postservere. Nå er meldingen vår i mottaker mailserveren (MTA) og overføres deretter til mottakerens datamaskin.[32]

Bedrift skal ha webbasert mail server(Roundcube) Det er en webmail pakke som er enkel å konfigurere og er tilgjengelig på alle plattformer som støtter PHP. Roundcube støtter Drag-&-drop message management, Sophisticated privacy protection, Tilgjengelig på over 80 språk, etc [27]

5.6 Gateway

Pc 3 ble valgt ut til å fungere som nettverkets gateway med CentOS 8 som underliggende OS. Ettersom maskinen ble levert med kun et nettverkskort (NIC) måtte det opprettes virtuelle adaptere for videre tilkoblinger i nettet. Disse Interfaces ble konfigurert til å samsvare med tilkoblinger for enheter (nevnt i oversikt) som senere skulle plasseres i DMZ-sonen. Fremgangsmåten her er i prinsippet å bare kopiere første interface og sette et "alias" foran navnet på interface filen. Deretter kan man konfigurere filen med innstillingene man ønsker.

Samtlige innstillinger for interfaces som skulle være i ekstern sone, ble satt til statisk ip ved boot og tildelt gateway mot ISP. Default DNS for ekstern sone ble satt til 8.8.8.8 for trafikk ut mot nett. For tilkoblingen inn mot LAN nettverket, ble det fjernet DNS konfigurasjon og gateway konfigurasjon på Interface slik at trafikken på forblir lokal på LAN. Konfigurasjon for IP ble dog satt til å være dynamisk ved boot.

IPforwarding for de respektive enhetene lokalt, ble lagt ved å opprette en egen konfigurasjonsfil etc/sysctl.d/ip_forward.conf, med innholdet net.ipv4.ip forward = 1.

Filen ble satt til å være permanent og blir aktiv ved hver oppstart.

Brannmurreglene ble i henhold til oppdragsbeskrivelsen tilpasset med høy sikkerhet for nettverket, hvor kun enheter i DMZ-sonen kunne nås fra internett med port 80 for webserver og 443 og 25 for mail. Maskering av IPadresser benyttes for enheter i DMZ-sonen samt enheter på intern sone.

5.7 DMZ

DMZ-sonen ble konfigurert til å inneholde serverne DNS, webserver og mailserver. Sonen ligger allerede konfigurert i oppsettet til CentOS 8 og velges ut ifra firewalld sitt innhold av soner. En oversikt over disse kan enkelt synliggjøres med kommandoen: cat /usr/lib/firewalld/zones/public.xml.

Enheter og/eller tilkoblinger som skal plasseres i sonen, konfigureres i terminal med tilhørende innstillinger og porter via firewalld-kommandoer.

5.8 VLAN Tagging.

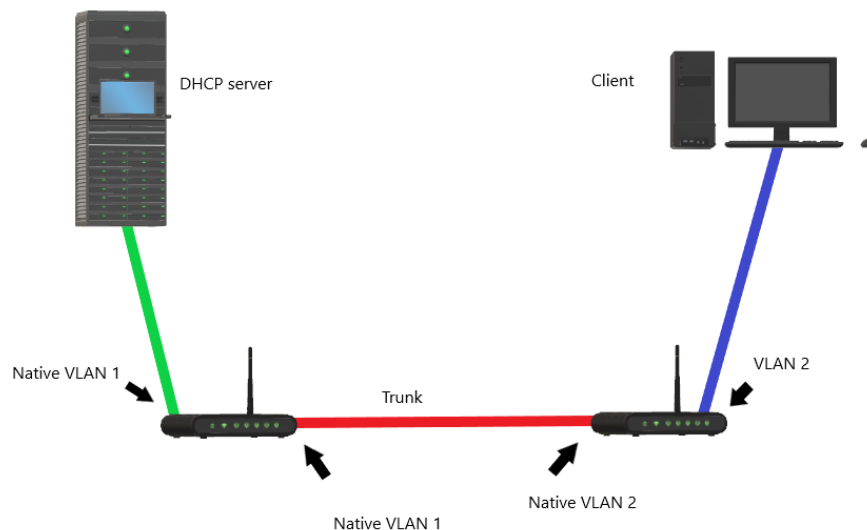
Virtuelle lokale nettverk, eller VLAN, adskiller trafikk i et nettverk. VLAN holder trafikk fra forskjellige nettverk atskilt når de krysser delte lenker og enheter innenfor en topologi. Denne prosessen, også kjent som VLAN-tagging, er uvurderlig for å begrense kringkastingstrafikk og sikre nettverkssegmenter. VLAN-merking er en integrert del av nettverk i alle størrelser og støttes på MX Security Appliances, MR Access Points og MS Series Switches. Dette kan gjøres for både data og management traffic.

Terminologi:

VLAN- Virtual Local Area Network, logisk identifikator for å isolere et nettverk
Trunk - En port aktivert for VLAN-tagging
Access - En port som ikke tagger og bare accepts et enkelt VLAN.
Encapsulation - Prosessen med å endre datarammer for å inkludere tilleggsinformasjon.
802.1Q - Den vanligste innkapslingsmetoden for VLAN-merking. Dette er metoden som brukes av Meraki-enheter.
Native VLAN - VLAN assosiert med all umerket trafikk på en trunk.

VLAN-aktiverte porter er generelt kategorisert på en av to måter, tagged eller untagged. Disse kan også refereres til som henholdsvis "trunks" eller "access". Formålet med en tagged eller "trunked" port er å passere trafikk for flere VLAN-er, mens en umerket eller accessport aksepterer trafikk for bare et enkelt VLAN. Generelt sett vil trunkportene koble switcher, og accessporter vil koble til slutenheter.

En klient er koblet til en VLAN 1-accessport og ønsker en adresse fra DHCP-serveren på VLAN 1-subnettet (192.168.1.0/24). Det er en native VLAN- mismatch på trunk linken mellom de to switchene, noe som forhindrer klienten i å motta riktig adresse. Når det kommer fra en access VLAN 1-port, når DHCP-forespørselen kommer til trunken på switchen, vil det være untagged trafikk, da det opprinnelige VLAN er 1. Når trafikken kommer til den andre switchen på den andre siden av trunken, native VLAN er 10. Den untaggede trafikken fra switchen til høyre vil bli behandlet som VLAN 10 på switchen til venstre. DHCP-serveren vil svare på DHCP-forespørselen for VLAN 10 (192.168.10.0/24) og sende adressen tilbake til klienten. Nok en gang, da VLAN 10 ikke er merket på venstre switch, vil den bli behandlet som VLAN 1 på høyre switch på grunn av den native VLAN-mismatch, og klienten vil til slutt få en adresse i feil subnet. [43]



Et enkelt eksempel på hvordan VLAN tagging/trunking kan se ut. Selvlaget figur

5.9 DNS

Vi har vår DNS som er BIND9 som en egen PC og har egen IPadresse som er 192.168.10.99. Denne datamaskinen koblet vi sammen med de andre datamaskinene og raspberry-piene ved hjelp av IPadresser, dette gjorde det enklere ved å ha statisk IPadresse som da alltid er den samme istedenfor å ha en dynamisk IPadresse som gjør at den kan endre seg hver gang man skruer på serveren. Dette er et problem fordi konfigureringen er satt for 1 IPadresse og vi må lage et script eller sette spesielle innstillinger for å finne riktig IPadresse hver gang.

Vi lagde et lokal domene som inneholder forward sone og revers sone og vi lagde forward sonen og revers sonen. Forward sone er en sone som vil lagre host sin IPadresse-forhold. Når hosten blir spurt om en IPadresse gir den IPadressen til host systemet ved hjelp av vertsnamnet. Reverssonen returnerer den omvendte DNS sonen FQDN for serveren i forhold til IPadressen.

Lagde forward DNS sone fil for domenet og konfigurerte filen for at den skal inneholde e-posten og nettsiden på IPadressene 192.168.10.96 for nettsiden og 192.168.10.95 for mail. Den inneholder også MX som gjør at mail kan komme og gå fra mailen igjennom DNS serveren uten å bli stoppet siden den finner utsiden av lokale nettverket. I revers DNS sone filen lagde vi en fil for revers DNS oppslag. Som gjør at det går fra IPadresse til nettside. Vi la også tilgang til DNS igjennom brannmuren sånn at alt fungerer.

5.10 HIDS og brukere

Vi satt opp en PC til å ha HIDS og lagre brukerdata. Vi brukte en PC til å ha og kjøre HIDS og lagre brukerdata. Denne datamaskinen har en statisk IP, dette er fordi vi har satt opp konfigurasjonen med 1 IP og ønsker ikke at den skal kunne endre seg og ikke fungere. Denne IPadressen er 192.168.10.98. En annen ting vi gjorde var å sette IP utenfor de vanlige IP portene som er (192.168.10.101-192.168.10.199), dette er fordi vi ikke ønsker at andre enheter skal kunne få denne IPadressen og gjøre at ting ikke fungerer.

5.11 HIDS

Programmet vi brukte til HIDS var OSSEC. Vi satt opp en hybrid HIDS som er server og agent. Vi koblet sammen alle datamaskinene og raspberry-piene ved hjelp av IPadressene og dette gjorde at vi kun trengte å ha DNS på den ene datamaskinen og ikke på alle. Vi satt opp HIDS sånn at vi kunne overvåke alle tilganger og varsle når noen prøver å gjøre noe med systemet. Vi la inn integritet sjekk på systemet som gjør at det sjekker om systemet fungerer sånn det skal i sanntid. Vi aktiverte også rootkit-deteksjonsmotor som gjør at det sjekker mot rootkits som kan ha kommet inn i systemet og varsler hvis det oppdager noe.

5.12 Brukere

Denne datamaskinen vil inneholde brukerdata fra interne og eksterne brukere. Interne brukere som er root brukere eller andre brukere som har tilgang til systemet, men også eksterne brukere som er brukerne som bruker e-posten fra raspberry-pien. Dette vil kunne være epost

adresser, navn og andre brukerinnstillinger og personinfo fra brukerne. Vil også lagre passord som vil være kryptert og ikke være tilegnelig kun fra root eller brukere som har tilgang til root. Vi lagde 4 brukere, 1 til hver person i gruppen som har root tilgang og kunne gjøre endringer ved sudo command.

5.13 Backup

Backup (Sikkerhetskopiering) er duplikatkopi av filer og database for å bevare og gjenopprette data umiddelbart etter en hendelse med opprinnelig tap av data.

For nettverket prosjekt 2 valgt vi Rsync. Rsync er et verktøy som gir utrolig allsidighet for sikkerhetskopiering og synkronisering av data. Den kan brukes lokalt til å sikkerhetskopierte filer til forskjellige kataloger eller kan konfigureres for å synkronisere over Internett til andre server eller host. [x4]

6. Konklusjon

I dette prosjektet fikk tatt i bruk alt vi lærte i prosjekt 1, samt at vi måtte tilegne oss mye ny kunnskap om hvordan vi skulle sette opp dette nettverket for å få dette til å fungere. Vi hadde et spørsmål til faglærer dagen før innlevering noe som gjorde at vi fant ut at vi dessverre hadde vi misforstått oppgaven litt og 24 timer før levering fikk vi vite at vi skulle sette opp en gateway manuelt. Dette hadde vi ikke gjort på dette tidspunktet og vi måtte hive oss rundt for å komme i mål da dette var en av de, hvis ikke den største operasjonen som måtte gjennomføres. Vi kom heldigvis i mål, men det ble ikke noe søvn natt til 28. mai.

Nå er det prosjekt 2 gjennomført og nettverket til bedrift 9 er oppe og fungerer som det skal. Nettverket prosjekt har fått diverse tjenester: DNS, Web, Mail og DHCP.

7. Referanser

Apache:

[1]" Usage Statistics and Market Share of Apache, March 2021", W3techs.com, 2021. [Online]. Available: <https://w3techs.com/technologies/details/ws-apache>. [Funnet Funnet 19.Mai 2021].

[2]"Apache Vs NGINX – Which Is The Best Web Server for You?", ServerGuy.com, 2021. [Online]. Available: <https://serverguy.com/comparison/apache-vs-nginx/>. [Funnet 19.Mai 2021].

[3]"Apache HTTP Server vs lighttpd vs NGINX | What are the differences?", StackShare, 2021. [Online]. Available: <https://stackshare.io/stackups/apache-httpd-vs-lighttpd-vs-nginx>. [Funnet 19.Mai 2021].

[4]"Prosjekt 1 Gruppe 12 side 3 og 4."

Postfix:

[5]"7 best Linux mail servers - Dade2", Dade2, 2021. [Online]. Available: <https://dade2.net/7-best-linux-mail-servers/>. [Funnet 19.Mai 2021].

[6]"Postfix (software)", En.wikipedia.org, 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Postfix_\(software\)](https://en.wikipedia.org/wiki/Postfix_(software)). [Funnet 19.Mai 2021].

[7]"Postfix vs Sendmail vs Exim - Choosing MTA %page% %sep% %sitename%", Mailtrap Blog, 2021. [Online]. Available: <https://blog.mailtrap.io/postfix-sendmail-exim/>. [Funnet Funnet 19.Mai 2021].

[8]"Postfix vs. Sendmail – Linux Hint", Linuxhint.com, 2021. [Online]. Available: https://linuxhint.com/postfix_vs_sendmail/. [Funnet 19.Mai 2021].

[9] "Prosjekt 1 Gruppe 12 , side 5."

Firewalld

[10]T. Woerner and T. Woerner, "Home", firewalld, 2021. [Online]. Available: <https://firewalld.org/>. [Funnet 19.Mai 2021].

[11]"Firewall (computing)", En.wikipedia.org, 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Firewall_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing)). [Funnet 19.Mai 2021].

[12]"IPTABLES VS FIREWALLD | Unixmen", Unixmen.com, 2021. [Online]. Available: <https://www.unixmen.com/iptables-vs-firewalld/>. [Funnet 19.Mai 2021].

[13] "Prosjekt 1 Gruppe 12 , side 7."

Bind9:

[14]"Domain Name System", No.wikipedia.org, 2021. [Online]. Available: https://no.wikipedia.org/wiki/Domain_Name_System. [Funnet 19.Mai 2021].

[15]"The Top DNS Servers And What They Offer - DNSimple Blog", Blog.dnsimple.com, 2021. [Online]. Available: <https://blog.dnsimple.com/2015/02/top-dns-servers/>. [Funnet 19.Mai 2021].

[16]"DNS, BIND Nameserver, DHCP, LDAP and Directory Services", Bind9.net, 2021. [Online]. Available: <https://www.bind9.net/>. [Funnet 20.Mai 2021].

[17]Computingforgeeks.com, 2021. [Online]. Available: <https://computingforgeeks.com/bind-vs-dnsmasq-vs-powerdns-vs-unbound/>. [Funnet 20.Mai 2021].

[18] "Prosjekt 1 Gruppe 12 , side 8."

[28] En.wikipedia.org. 2021. Domain Name System - Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/Domain_Name_System [Accessed 19 Mai 2021].

[29] |. 2021. DNS Zones and Zone Files Explained. [online] Available at: <http://www.steves-internet-guide.com/dns-zones-explained/> [Accessed 19 Mai 2021].

[30] Cope, S., 2021. Understanding DNS - Beginners Guide to DNS. [online] |. Available at: <http://www.steves-internet-guide.com/dns-guide-beginners/> [Accessed 19 Mai 2021].

[31] Cope, S., 2021. DNS Lookups Explained. [online] |. Available at: <http://www.steves-internet-guide.com/dns-lookups/> [Accessed 19 Mai 2021].

OSSEC:

[19]Logz.io, 2021. [Online]. Available: <https://logz.io/blog/open-source-hids/>. Funnet 20.Mai 2021].

[20]"OSSEC - World's Most Widely Used Host Intrusion Detection System - HIDS", OSSEC, 2021. [Online]. Available: <https://www.ossec.net/>. [Funnet 20.Mai 2021]

[21]"8 Best HIDS Tools - Host-Based Intrusion Detection System - DNSstuff", Software Reviews, Opinions, and Tips - DNSstuff, 2021. [Online]. Available: <https://www.dnsstuff.com/host-based-intrusion-detection-systems>. [Funnet 20.Mai 2021].

[22] "Prosjekt 1 Gruppe 12 , side 9."

Roundcube:

[23]"Roundcube", En.wikipedia.org, 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Roundcube>. [Funnet 20.Mai 2021].

[24]"Ajax (programming)", En.wikipedia.org, 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)). [Funnet 20.Mai 2021].

[25]S. Milojkovic, "Horde vs Roundcube vs Squirrelmail - Which Works Best", LinOxide, 2021. [Online]. Available: <https://linoxide.com/linux-how-to/horde-vs-roundcube-vs-squirrelmail/>. [Funnet 20.Mai 2021].

[26] "Prosjekt 1 Gruppe 12 , side 9.

[27] "About Roundcube", roundcube.net, 2021 [online] Available: <https://roundcube.net/about/>

[32]"How Internet Email Works" mailenable.com, 2021[online]. Available at: https://www.mailenable.com/documentation/10.0/Standard/How_Internet_Email_Works.html /

[33] MDN contributors, May 4, 2021," What is a web server?"[online] Available: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server/

[34] Velocity consultancy, March 20, 2020, "What is CMS?" [online] Available: <https://www.velocityconsultancy.com/what-is-a-cms-website/>

[35] Amethyst, 27.05.2021 (online) Available: <https://www.ametys.org/community/en/index.html/>

[36] "Rsync", Rsync.samba.org, 2021[Online]. Available: <https://rsync.samba.org/>

Installing and configuring a Linux gateway

In-text: (Installing and configuring a Linux gateway, 2021)

[37]TechRepublic. 2021. Installing and configuring a Linux gateway. [online] Available at: <https://www.techrepublic.com/article/installing-and-configuring-a-linux-gateway/> [Accessed 27 Mai 2021].

[38] *How to Turn a Linux Server into a Router to Handle Traffic Staticly and Dynamically - Part 10.* [online] Tecmint.com. Available at: <https://www.tecmint.com/setup-linux-as-router/> [Accessed 27 Mai 2021].

[39]How To Install Omeka Classic Cms On Centos 8. [online] Available at: <https://www.youtube.com/watch?v=Dp0KKXQSdf6w> [Accessed 27 Mai 2021].

[40]Linux Ubuntu As Router Gateway [online] Available at: <https://www.youtube.com/watch?v=NsUNKUtGw8o> [Accessed 27 Mai 2021].

[41]How to configure sub Interface on Centos [online] Available at: https://www.youtube.com/watch?v=fTeZj_8lzlK [Accessed 27 Mai 2021].

[42]2021. [online] Available at: <<https://www.cyberciti.biz/faq/how-to-set-up-a-firewall-using-firewalld-on-centos-8/>> [Accessed 27 Mai 2021].

[43] Fundamentals of 802.1Q VLAN Tagging

Your Bibliography: Cisco Meraki. 2021. Fundamentals of 802.1Q VLAN Tagging. [online] Available at:

<https://documentation.meraki.com/General_Administration/Tools_and_Troubleshooting/Fundamentals_of_802.1Q_VLAN_Tagging> [Accessed 28 May 2021].

8. Vedlegg

[v2]

Postfix Announcements

2021

April 29, 2021: [Postfix stable release 3.6.0](#).

April 11, 2021: [Postfix stable release 3.5.10 and legacy releases 3.4.20, postfix-3.3.17, 3.2.22](#).

January 17, 2021: [Postfix stable release 3.5.9 and legacy releases 3.4.19, 3.3.16, 3.2.21](#).

2020

November 7, 2020: [Postfix stable release 3.5.8 and legacy releases 3.4.18, 3.3.15, 3.2.20](#).

August 30, 2020: [Postfix stable release 3.5.7 and legacy release 3.4.17](#).

July 26, 2020: [Postfix stable release 3.5.6 and legacy releases 3.4.16, 3.3.14, 3.2.19](#).

July 24, 2020: [Postfix stable release 3.5.5 and legacy releases 3.4.15, 3.3.13, 3.2.18](#).

June 27, 2020: [Postfix stable release 3.5.4 and legacy releases 3.4.14, 3.3.12, 3.2.17](#).

June 14, 2020: [Postfix stable release 3.5.3 and legacy releases 3.4.13, 3.3.11, 3.2.16](#).

May 16, 2020: [Postfix stable release 3.5.2 and legacy releases 3.4.12, 3.3.10, 3.2.15](#).

April 18, 2020: [Postfix stable release 3.5.1 and legacy releases 3.4.11, 3.3.9, 3.2.14](#).

March 16, 2020: [Postfix stable release 3.5.0](#).

March 12, 2020: [Postfix legacy releases 3.4.10, 3.3.8, 3.2.13](#).

February 3, 2020: [Postfix stable release 3.4.9 and legacy releases 3.3.7, 3.2.12, 3.1.15](#).

2019

November 24, 2019: [Postfix stable release 3.4.8](#).

September 22, 2019: [Postfix stable release 3.4.7 and legacy releases 3.3.6, 3.2.11, 3.1.14](#).

June 29, 2019: [Postfix stable release 3.4.6 and legacy releases 3.3.5, 3.2.10, 3.1.13](#).

March 30, 2019: [Postfix stable release 3.4.5 and legacy releases 3.3.4, 3.2.9, 3.1.12.](#)

March 14, 2019: [Postfix stable release 3.4.4.](#)

March 10, 2019: [Postfix stable release 3.4.3.](#)

March 10, 2019: [Postfix stable release 3.4.2.](#)
















March 7, 2019: [Postfix stable release 3.4.1.](#)

February 27, 2019: [Postfix stable release 3.4.0.](#)

February 26, 2019: [Postfix legacy releases 3.3.3, 3.2.8, 3.1.11, 3.0.15.](#)

[v3]

Postfix vs Exim vs Sendmail overblikk

	Postfix	Exim	Sendmail
Number of servers worldwide (as of October 2019)	306,907 (34.91%) 	499,992 (56.88%) 	34,551 (3.93%) 
Security	 1	 2	 3
Performance	 1	 2	 3
Reliability	 1	 2	 3
Configurability	 2	 1	 3
Portability	 2	 3	 1
Summary	15	13	8

[v4] Releases Bind9

BIND 9					
VERSION	STATUS	DOCUMENTATION	RELEASE DATE	EOL DATE	DOWNLOAD
9.17.10	Development	BIND 9.17 ARM (HTML PDF) Release Notes (HTML)	February 2021	TBD	Download
9.16.12	Current-Stable	BIND 9.16 ARM (HTML PDF) Release Notes (HTML)	February 2021	TBD	Download
9.11.28	Current-Stable, ESV	BIND 9.11 ARM (HTML PDF) Release Notes (HTML PDF)	February 2021	December 2021	Download