MNFIT 291 - Prosjektarbeid i informatikk

Mac- og Windows-integrasjon i Skolelinux

Sluttrapport
Gruppe 7

Prosjektdeltagere:
Svein Magne Bang, Sigurd Thune og Odd Rune Dahle

Oppdragsgiver: Terje Rydland

FAKULTET FOR INFORMASJONSTEKNOLOGI, MATEMATIKK OG ELEKTROTEKNIKK
NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET

Sammendrag

Som en del av faget MNFIT291 Informatikk prosjektarbeid II ved Institutt for datateknikk og informasjonsvitenskap ved NTNU, har denne prosjektgruppen i løpet av vårsemesteret 2003 jobbet med et oppdrag fra Skolelinux. Skolelinux er en norsk linuxdistribusjon tilrettelagt for det norske skoleverket. Distribusjonen er basert på frivillig arbeid fra et landsdekkende nett av entusiaster.

Skolelinux hadde behov for å la skolenes Windows- og Macintosh-maskiner kobles til deres arkitektur. Prosjektgruppen har gjennomført denne utvidelsen av arkitekturen på en vellykket måte. Oppgaven har bestått mer av å tilpasse og dokumentere eksisterende programvare enn å implementere ny kode.

Innhold

Kapittel 1

Introduksjon til prosjektet

1.1 Innledning

1.1.1 Bakgrunn

Ved introduksjonsmøtet i MNFIT291 20. januar 2003 ved NTNU, ble det lagt frem en rekke oppdrag man kunne velge å arbeide med. Denne gruppen valgte der et oppdrag gitt av Skolelinux, via Terje Rydland (terje.rydland@idi.ntnu.no), som gikk ut på å integrere Macintosh- og Windows-maskiner i Skolelinux' arkitektur. Representant fra Skolelinux og ekstern veileder, var Per Harald Westby (per.h.westby@broadpark.no).

Gruppen har alle tidligere erfaring med drift av forskjellige typer nettverk, hvor man er avhengig av at datamaskiner med forskjellige typer operativsystemer snakker med hverandre. Gruppen har også en del erfaring med Linux, og mener at Skolelinux er et interessant prosjekt, med en god baktanke.

Prosjektets varighet er satt til å være fra januar 2003 til innleveringsfrist 28. april 2003 i henhold til tidsskjemaet som er satt opp i dette faget.

1.1.2 Hvordan rapporten skal leses

Rapportens innhold er delvis ganske teknisk, og skrevet for et varierende publikum. Den er primært en presentasjon av vårt arbeide, men også dokumentasjon for Skolelinux.

IT-ansvarlige som skal bruke Skolelinux bør som et minimum lese vedleggene som omhandler installasjon av deres plattform (mac og/eller windows). Det anbefales også å lese utviklingsrapporten spesielt, og helst utredningen.

Utviklere fra Skolelinux bør i tillegg lese utviklingsrapporten spesielt nøye, samt vedleggene med konfigurasjon og kildekode.

Representanter fra Instituttet har nok mest utbytte av introduksjon, kravspesifikasjon og avslutning, selv om vi håper at de klarer å lese hele dokumentet :-).

Gruppen minner om ordlisten i vedlegg A på side 47.

1.1.3 Oppdragsgiver

Dette prosjektet er bare en liten del av et langt større utviklingsprosjekt kalt Skolelinux [1]. Skolelinux er en Linux-distribusjon som utvikles i regi av Linux i skolen [2]. Dette er en idealistisk medlemsorganisasjon som tilrettelegger og informerer om bruk av fri programvare (opensource) i norske skoler. Linux i skolen ble stiftet 16. juli 2001 og fungerer som en "paraply" for flere prosjekter som arbeider med å få Linux inn i norske skoler [1].

Målet er å tilby skolen en gratis programvareplattform som er bedre tilpasset skolens behov enn tilsvarende komersielle løsninger som finnes i dag. Skolelinux kommer her inn som en komplett løsning bestående av et operativsystem og tilhørende programvare utviklet spesielt med tanke på skolen som bruker.

Alle programmene som leveres med skolelinux skal være tilgjengelig på bokmål, nynorsk og i stor grad samisk. Dette er helt på linje med de anbefalinger Læringssenteret på oppdrag fra Utdanningsdepartementet har kommet med. I Språkrådets tolkning av forskriftene til opplæringsloven [2] kommer det klart frem at regelen om paralellutgaver av alle læremidler gjelder, uansett om det er et læremiddel som er papirbasert, eller på elektronisk form.

Skolelinux er et distribuert prosjekt og har i dag over 50 medlemmer som arbeider med oversetting og utvikling. Arbeidet er basert på frivillig innsats. Prosjektet har en flat organisasjonsstruktur og baserer seg rundt "gjørokratiet". Dette innebærer at alle kan jobbe med akkurat den biten de vil til en hver tid, noe som ofte er vanlig fremgangsmåte i opensource-miljøet. Skolelinux baserer seg på Debian GNU/Linux [3] og dermed også GNU GPL [4] lisensiering. Dette innebærer at alt som utvikles også må gjøres tilgjengelig for andre. GPL tillater fri kopiering/-distribuering og modifisering av programvaren.

[1] http://www.skolelinux.no/
[2] http://www.linuxiskolen.no/
[3] http://www.debian.org/
[4] http://www.gnu.org/licenses/

1.2 Oppgaven

1.2.1 Oppgavebeskrivelse

Oppgaven går ut på å vurdere og beskrive hvordan en Mac med Mac OS X og en PC med Microsoft Windows best kan kobles inn i Skolelinux' nettverksarkitektur og utnytte de tjenester Skolelinux (tjeneren) tilbyr. Det skal med andre ord etableres en profil for mac- og/eller windows-klienter i Skolelinux. Videre skal det implementeres utvalgte deler av den/de etablerte profilen(e) for å automatisere og legge til rette for dette.

Aktuelle teknologier er bl.a. OpenLDAP, Samba, Netatalk, Cups, NFS og X-tjenere, for å eventuelt kunne kjøre applikasjoner på tjenere fra mac og/eller windowsklient. Implementasjonsdelen av oppgaven vil for det meste bestå av å tilpasse allerede eksisterende og velkjent programvare for å passe in i Skolelinux' arkitektur.

1.2.2 Hvorfor?

Skolelinux er et frivillig prosjekt igangsatt for å hjelpe skoler med å få mer ut av sine IT-budsjett. I dag går veldig mye penger med til å betale lisenser til Microsoft for Windows og Office-programmer[3].

Mange skoler har fremdeles mange maskiner som er kraftige nok til å kjøre Windows, og ønsker å fortsette med dette. Vår oppgave var å la disse få bruke tjenester som fil- og printerdeling fra et Skolelinux-nettverk, slik at en konvertering til kun Linux skal gå lettest mulig.

1.2.3 Avgrensning

Gruppen har valgt å avgrense oppgaven til å omfatte fil- og skriver-deling til Mac OS X og Windows 98, Windows NT, Windows 2000 og Windows XP Professional. Det er verdt å merke seg at Windows 95 er valgt bort på grunn av sikkerhetsmessige hensyn, mens Windows XP Home i stor grad mangler mulighet til å jobbe som en nettverksklient (mangel på funksjonalitet).

Det vil vektlegges å få til automatikk i tilknytningen i form av scripting og tilrettelegging for å få konfigureringarbeidet ned på et minimum. Resterende tjenester vil bli tatt med i utrednings- og dokumentasjonsfasen, men blir ikke implementert på grunn av den begrensede tidsrammen prosjektet har. Enkelte tjenester vil i utgangspunktet også være tilgjengelig as-is for mac og windows, uten nødvendig

modifisering av hverken klient eller tjener. Valg av programvare på klientsiden for disse tjenestene (f.eks e-post) vil være opp til hver enkelt administrator å velge.

1.3 Planlegging og rapportering

1.3.1 Inndeling av prosjektet

Oppgaven er i utgangspunktet delt opp i to deler. En utredningsdel hvor man skal finne løsninger som gir mac- og windows-klienter tilgang til tjenester på Skolelinux' tjener, og en del som består av å implementere utvalgte deler av denne løsninger for Skolelinux. Vi velger også å legge til en tredje "usynlig" del, som består i å sette seg inn i det omfattende materialet som omhandler alle løsningene Skolelinux baserer seg på. Det er et omfattende system som det tar tid å bli kjent med før man i det hele tatt kan begynne på selve oppgaven.

1. Oppstartsfase
    - Ta kontakt med oppdragsgiver
    - Få satt opp maskiner i et testnett
    - Få tilgang til Skolelinux CVS
    - Påmelding i Skolelinux mailinglister
2. Studier/research
    - Studere Skolelinux arkitektur
    - Utrede kravspesifikasjon og avgrense oppgave
    - Lære om autentiseringssystemet i Skolelinux (LDAP)
    - Finne informasjon om alternative aktuelle programvareløsninger
3. Utredningsfase
    - Prøve ut forskjellige tjenester og sammenligne dem
    - Danne en profil for mac/Windows-klienter i Skolelinux' arkitektur.
4. Implementasjon
    - Få opp valgte programvare for fil- og skriverdeling på Skolelinux tjener med full LDAP-støtte
    - Testing av løsning lokalt
    - Feilretting, korrigering

5. Avslutningsfase

    - Skrive ferdig prosjektrapport og dokumentasjon

1.3.2 Møtevirksomhet og rapportering

Som en del av opplegget i faget mnfit291 skal gruppen levere en ukentlig kort statusrapport til veileder. Gruppen har etter avtale satt opp et møte hver tirsdag kl 12 med veileder Sven Ziemer hvor siste ukes statusrapport gjennomgås og spørsmål vedrørende de praktiske forhold rundt prosjektet tas opp.

Kontakt med arbeidsgiver skjer også på ukentlig basis gjennom et møte hver tirsdag kl 11. Resten av kontakt med andre utviklere av Skolelinux foregår via mail, samt en utviklersamling i februar hvor to av gruppens medlemmer deltok. Viktige spørsmål vedrørende oppgavens innhold taes opp på møter med Rydland, på disse møtene blir det skrevet et kort referat fra i ettertid som sendes på mail.

Det er blitt tildelt et rom til prosjektet ved i IT-Syd på Gløshaugen. Dette rommet står fritt tilgjengelig for prosjektgruppen og de enkelte gruppemedlemmene disponerer egen nøkkel hit. Det er ingen faste tider satt opp for arbeid på dette rommet, medlemmene benytter seg av det etter som det passer inn i timeplanen deres.

Viktige beskjeder og planlegging innad i gruppen foretas på egen mailingliste opprettet ved NTNU eller ved møter som avtales etter behov.

1.3.3 Organisering

Studenter

Gruppeleder	Svein Magne Bang
Utviklere	Sigurd Thune
	Odd Rune Dahle

Institutt

Oppdragsgiver	Terje Rydland
	Per Harald Westby
Veileder	Sven Ziemer

1.4 Fremdrift og tidsplanlegging

1.4.1 Milestones

Vi har satt opp en rekke milestones utover i prosjektet for å få bedre kontroll med tiden vi bruker. Vi har valgt å fastsette det på ukesbasis. En del av oppgavene strekker seg over flere uker. For å tilpasse fremdriften med de ukentlige møtene hver tirsdag velger vi å si at hvert enkelt punkt skal være fullført innen tirsdagen den gjeldende uka.

- Testnett på skolen - 11. februar
  Eget rom må være tildelt og maskiner satt på plass. Utstyret må være koblet opp i henhold til skolelinux arkitektur og nettet må fungere som et vanlig skolelinux-nett for å ha dette som utgangspunkt.

- Kravspesifikasjon fastsatt - 18. februar
  En komplett kravspesifikasjon skal være skrevet. Denne skal avklares og godkjennes sammen med oppdragsgiver.

- Utredelse av en profil/teknisk løsning - 11. mars
  En oversikt over hvordan en integrering av gitte plattformer i Skolelinux kan gjennomføres skal skrives ned. Valg av teknisk løsning skal skal dokumenteres og legges fram for oppdragsgiver.

- Rapport for midterm innlevering - 14. mars
  Midterm-rapport skal være ferdig skrevet og klar for levering. Alle punkter som kreves å være med i denne må være på plass.

- Ferdig mer implementering - 8. april
  Fil- og skriver-deling mot windows og må være ferdig implementert. Synkronisering av brukerdata mot samba må være ferdig og fungere. Alt skal være klart for demonstrasjon overfor kunde. Endringer skal være oppdatert i CVS.

- Endelig demonstrasjon overfor kunde - 24. april
  Alt må være ferdig implementert og tilstrekkelig testet slik at det er klart for komplett demonstrasjon.

- Ferdig skrevet dokumentasjon og rapport - 28. april
  Dokumentasjon i form av en howto om hvordan man skal koble til henholdsvis mac- og windows-klienter skal være klar og lagt inn i CVS. Dokumentasjonen skal også inn i rapport som vedlegg. Rapport skal være ferdig skrevet og klar til levering. Alle punkter som kreves til innleveringen skal

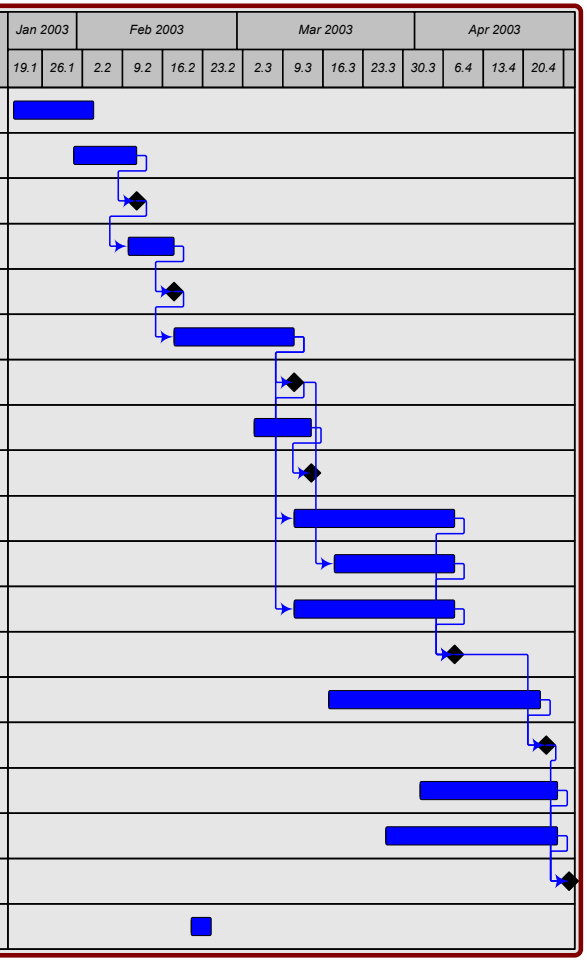
være dekket. Produktet i form av dokumentasjon og programvare skal være lagt inn i CVS hos kunde.

1.4.2 Gantt-diagram

De fleste oppgavene i diagrammet utføres av utføres av samtlige av gruppens medlemmer med unntak av implementasjonen som er delt opp. Inndelingen har vært som følger:

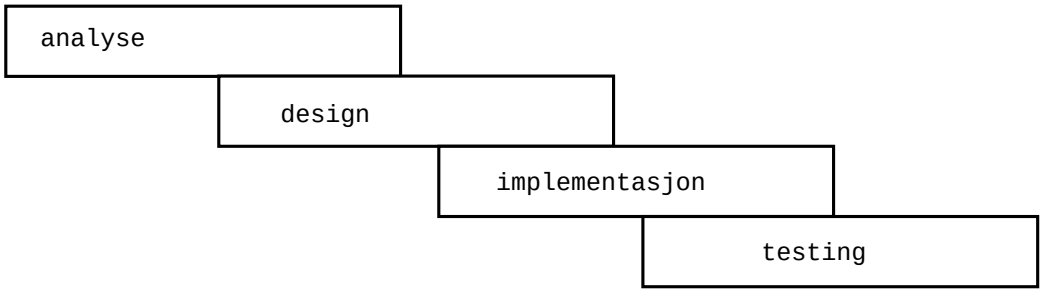
Implementering av passordsynkronisering	Odd Rune Dahle
Implementering av windowstilknytning	Svein Magne Bang
Implementering av mactilknytning	Sigurd Thune
Andre oppgaver	Alle

ID	Task Name	Start	Finish	Duration
1	Milestones, arbeidsfordeling og fremdriftsplan	21.01.2003	03.02.2003	10d
2	Reservere rom og koble opp nettverk	31.01.2003	10.02.2003	7d
3	Ferdig testnett klart til bruk	11.02.2003	11.02.2003	0d
4	Utarbeide kravspesifikasjon	10.02.2003	17.02.2003	6d
5	Avklart kravspesifikasjon	18.02.2003	18.02.2003	0d
6	Utrede tekniske løsninger, valg av tjenester	18.02.2003	10.03.2003	15d
7	Ferdig utredning og avklart valg av tekniske løsninger	11.03.2003	11.03.2003	0d
8	Renskrive rapport til mid-term	04.03.2003	13.03.2003	8d
9	Innlevering av mid-term rapport	14.03.2003	14.03.2003	0d
10	Implementere windows-tilknytning	11.03.2003	07.04.2003	20d
11	Implementere mac-tilknytning	18.03.2003	07.04.2003	15d
12	Implementere synkronisering av brukere	11.03.2003	07.04.2003	20d
13	Ferdig med implementeringsdel, alt skal være på plass	08.04.2003	08.04.2003	0d
14	Testing på eget nett	17.03.2003	22.04.2003	27d
15	Endelig demonstrasjon for oppdragsgiver på Brundalen VGS	24.04.2003	24.04.2003	0d
16	Skrive dokumentasjon og veiledning	02.04.2003	25.04.2003	18d
17	Ferdigstille endelig rapport	27.03.2003	25.04.2003	22d
18	Innlevere ferdig rapport	28.04.2003	28.04.2003	0d
19	Utviklersamling på Bryne (Time kommune) - fredag, lørdag og søndag	21.02.2003	24.02.2003	1,5d



1.5 Utviklingsmodell

Etter litt dikusjon og grubling ble det valgt i å ikke bruke noe avansert og moderne utviklingsmodell. Prosjektet har i stedet basert seg rundt fossefallsmetoden. Det finnes flere årsaker til gruppen endte opp med dette. Først og fremst skyldes det at en iterativ prosess designet for programvareutvikling fort ville bli vanskelig å gjennomføre i et slikt prosjekt. Med den knappe tidsrammen og det litt uoversiktelige arbeidet som ventet tok gruppen den “trygge” veien.



Figur 1.1: Fossefall – vår utviklingsmodell.

1.6 Risikovurdering

Sannsynlighet og konsekvens er her delt inn i tre nivåer: lav, middels og kritisk. Ordene er i utgangspunktet ganske selvforklarende, og jo høyere risiko og jo større konsekvens man har, desto mer alvorlig er hendelsen.

Hendelse	Sannsynl.	Konsekvens	Følger	Tiltak
Gruppemedlem slutter eller blir syk	Lav	Kritisk	Gruppen får ikke ferdig det planlagte produktet til leveringsdato	Redusere omfang av ferdig produkt underveis dersom en slik situasjon oppstår.
Mid-term rapport blir avvist som ikke tilfredstillende	Lav	Kritisk	Gruppen får ikke fullføre prosjektet	Sette av nok tid til å gi en tilfredstillende rapport til uke 11, samt sørge for jevnt arbeidstempo.
Oppgavedefinisjon endres underveis. Nye krav kommer legges frem	Middels	Middels	Tidsskjema sprenges, planer må endres. Lett å miste oversikt.	Sørge for en klar definisjon og målsetting i oppstart. Ved nye krav må dette forhandles med kunde og eventuelt bli plassert inn i prosjektet på bekostning av andre tidligere vedtatte krav.
Krav blir viser seg å være uoppnåelig uten omfattende (les: tidkrevende) programvareutvikling.	Middels	Middels	Det endelige resultat vil ikke fullt ut oppfylle kravspesifikasjon.	Få tidlig oversikt over de forskjellige løsningsalternativene og eventuelt gå tilbake til kunde for å diskutere funksjonelle krav mot tidsfrist.
Valg av teknisk løsning som viser seg å by på problemer.	Middels	Middels	Mindre tid til implementasjon av rett løsning ettersom man bruker av denne tiden til feil løsning.	Bruke tid på å undersøke de forskjellige løsningene nøye og utrede disse før man begynner implementasjon.
Ferdig løsning viser seg å være ustabil/manglende funksjonalitet i praktisk bruk.	Lav	Middels	Resulterende løsning som blir levert til den faste datoen er ikke fullt ut tilfredstillende.	Sette av tid til testing mot slutten med mulighet for å luke ut feil i god tid før leveringsdato.

Kapittel 2

Kravspesifikasjon

2.1 Innledning

Dette prosjektet er valgt som del av faget ≪Informatikk prosjektarbeid II≫ ved Institutt for datateknikk og informasjonsvitenskap ved NTNU våren 2003.

Skolelinux-prosjektet har etablert en nettverksarkitektur, et antall profiler for maskiner i et Skolelinux-nettverk (tjener, tynnklient-tjener, arbeidsstasjon og tynnklient) alle basert på Debian Linux, og et sett tjenester (autentisering, autorisasjon osv).

Første del av oppgaven er å vurdere og beskrive hvordan Mac med OS X og/eller PC med Windows best kan koples inn i denne nettverksarkitekturen og utnytte tjenestene den tilbyr, og derved etablere en profil for Mac- og/eller Windows-klienter i Skolelinux.

Andre del av oppgaven er å implementere utvalgte deler av den/de etablerte profilene.

2.1.1 Overordnet

Ordene "MÅ", "MÅ IKKE", "BØR", "BØR IKKE", "PÅKREVD", "ANBEFALT", "KAN" og "VALGFRI" i dette dokumentet skal tolkes likt sine engelske motstykker som beskrevet i RFC 2119[4].

Overordnede krav og målsettinger

- Det MÅ foreligge en rapport som beskriver hvordan OS X og/eller PC med Windows best kan kobles inn i Skolelinux' nettverksarkitektur.
- Prioriterte versjoner av Microsoft Windows er 98 og XP.
- Prioriterte versjoner for Macintosh er MacOS X 10.2 og nyere.

- Tilknytning av klienter MÅ være «plug and play» med et minimum av teknisk arbeid.
- Autentisering av brukere MÅ skje mot den sentrale brukerdatabasen (LDAP).
- Passord MÅ gå kryptert over nettverket.
- Autentiserte brukere MÅ få automatisk tilgang til sine hjemmeområder på Skolelinux-tjeneren.
- Tilgang til web/webproxy BØR være automatisk. Løsning for web/mail er VALGFRITT.

2.1.2 Tidsfrister

Prosjektet har leveransedato 28. april 2003. Andre tidsfrister og milepæler vurderes løpende i samtale med kunden.

2.2 Situasjonsanalyse

Skolelinux-arkitekturen er per i dag primært rettet mot skoler som vil ha en full overgang til Skolelinux-distribusjonen. Noen integrasjon av Windows- og Macintosh-klienter er ikke foretatt i dag.

2.2.1 Mål

Skolelinux-prosjektet satser på at overgangen til Skolelinux skal være så enkel som mulig. For å gjøre overgangen så smertefri som mulig for skolenes IT-ansvarlige er det ønskelig å la dem bruke eksisterende arbeidsstasjoner mot en Skolelinux-tjener. Dette prosjektet fokuserer på Windows og Macintosh.

2.2.2 Bruker-/Målgruppe

Skolelinux er tiltenkt elevnettverk på skolene, og alle oppsettvalg som er gjort har dette som ramme. Målgruppen er altså skoler og driftansvarlige på skoler.

Krav til brukeren

- Systemansvarlig må ha noe erfaring med bruk av Linux for å kunne installere (Gjerne igjennom kurs i regi av skolelinux for brukeradministrasjon.)

- Elever må ha en grunnleggende itk-kunnskap gitt av skolen for å kunne skifte passord og logge seg på maskinene selv.

2.2.3 Behovsanalyse

Kunden må ha en løsning som gjør det enkelt for skoler å beholde sine gamle klienter på allerede eksisterende operativsystemer. Det er ønskelig at klientene skal kunne dualboote - enten gammelt operativsystem eller Linux tynnklient.

Løsningen MÅ være godt testet før leveringsdato. Å oppfylle få krav godt er fra kundens ståsted bedre enn å oppfylle alle krav halvveis.

2.3 Teknisk løsning

2.3.1 Funksjoner

Primærkrav til funksjoner er autentisering, fil- og skriverdeling. Det legges vekt på at driftsansvarlige MÅ ha minst mulig jobb med oppsett og vedlikehold utover førstegangs installasjon. I tillegg er det ønskelig å ha løsninger for webproxy og epost.

Det er IKKE PÅKREVD at Mac- og Windows-klienter skal kunne konfigureres/administreres vha cfengine.

2.3.2 Bruk av standard programvare

All programvare som leveres som en del av løsningen skal være lisensiert under GNU General Public License. Til dokumentasjon benyttes GNU Free Documentation License.

2.3.3 Levering

Endelig levering er i form av at gruppens løsninger (og dokumentasjon, rapport) er med i Skolelinux-distribusjonen. Gruppens arbeide skal sjekkes inn i Skolelinux' CVS tjener.

Endelig levering til NTNU/IDI:

- demonstrasjon
- rapport

2.4 Dokumentasjon

Prosjektgruppen MÅ produsere en rapport som beskriver hvordan MacOS X og/eller PC med Windows best kan kobles inn i Skolelinux' nettverksarkitektur. Denne rapporten skal brukes som utgangspunkt i prosjektets implementasjonsfase.

2.4.1 Brukerdokumentasjon

Det skal produseres dokumentasjon for hvordan driftsansvarlige på skoler kan bruke prosjektets løsning til å integrere sine eldre klientmaskiner i Skolelinux-arkitekturen.

2.4.2 Systembeskrivelse

Som en del av sluttarbeidet skal det foreligge en systembeskrivelse av implementasjonsarbeidet som tar for seg minimum følgende punkter:

- Systemkrav.
- Installasjonsprosedyre.
- Drift / vedlikehold.
- Brukte komponenter.

Kapittel 3

Utredning

3.1 Innledning

Oppgaven består av to hoveddeler. Først en som består av å utrede hvordan en integrering av mac og windows mot Skolelinux kan gjøres, og deretter en implementering av noen spesielt prioriterte elementer.

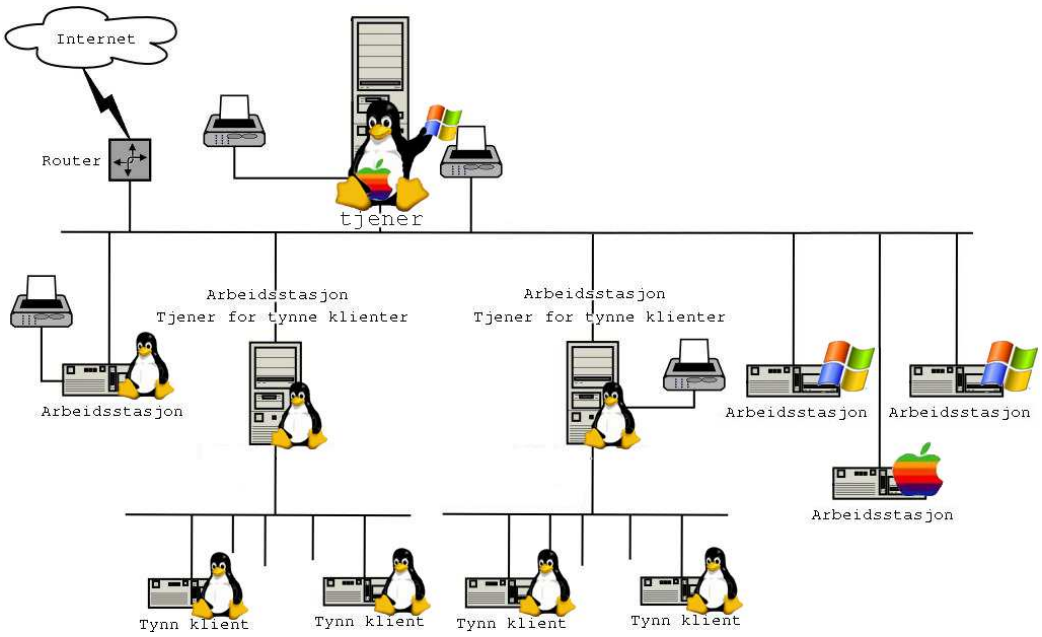
Det første som må bringes på det rene er hvordan maskiner som kjører Mac OS eller windows kan plasseres inn i skolelinuxarkitekturen. Først må den fysiske plasseringen i nettet avklares, deretter må ressursene som tilbys fra Skolelinux sees i sammenheng med disse operativsystemene.

Skolelinux-tjeneren har en rekke tjenester den tilbyr på nettet til sine klienter. En viktig del av denne oppgaven er å utrede hvordan man best kan få utnyttet disse fra de plattformene som skal integreres. Målet med denne delen av rapporten er å finne frem til hvilke tjenester støttes i utgangspunktet, hva som trenger ytterligere dokumentasjon og hva skal implementeres i dette prosjektet.

3.2 Ønsket arkitektur

Som utgangspunkt har Skolelinux etablert en arkitektur bestående av en tjener, tykke klienter, tynnklient-tjenere og tynnklienter. Hver tynnklient-tjener har sitt eget separate nettverk for sine klienter. Hovednettet består av tjeneren, tynnklient-tjenere og tykke klienter (arbeidssstasjoner). Første del av dette prosjekter er å etablere en profil for klienter som kjører Mac OS og windows. Første ledd i utredningen av en slik profil blir å plassere maskinene fysisk inn i nettverksarkitekturen.

Figur 3.1 er et forsøk på å vise hvordan gruppen ser for seg at slike klienter er integrert med den tidligere arkitekturen. Alle klienttyper er representert. Som det fremgår av figuren blir både mac- og windowsklienter plassert på hovednettet. Dette valget er gjort ettersom begge plattformer hører hjemme under definisjonen



Figur 3.1: Ny Skolelinux-arkitektur

av en tykk klient. I noen tilfeller vil det kanskje være ønskelig å kunne plassere disse på et tynnklient-nett av rent praktiske (fysiske) årsaker, men dette anbefales ikke. For Mac kan dette muligens fungere ved hjelp av en bra konfigurert NATing på tynnklient-tjeneren, men vil lett kunne få problemer. Dersom windows skal baseres rundt samba/windows-fildeling vil en slik ruting av trafikk neppe fungere tilstrekkelig. Slik trafikk er av erfaring svært vanskelig å få transportert på tvers av subnett. Den sentrale tjeneren får utvidelser i form av konfigurasjon, ny programvare og nye script for å håndtere denne nye situasjonen.

3.3 Tjenester

De tjenestene som tilbys av Skolelinux som er aktuelle i forbindelse med tilknytning av mac- og Windows-klienter er:

- Autentisering via LDAP
- Tildeling av ip-adresser (DHCP)
- Navnetjener (DNS)
- E-post-tjener

- Web-tjener (Apache)
- Web-tilgang via proxy (Squid)
- Fildeling (NFS og Samba)
- Skriverdeling (CUPS og Samba)
- Klokkesynkronisering (NTP)

Ved å gå igjennom dem punkt for punkt forsøkes det å komme frem til en konklusjon om hva som behøves for å få tilknyttet klientene som skal integreres. Deretter dykkes det litt mer inn i dybden på de aktuelle teknologiene som finnes til en implementering.

3.3.1 Autentisering

Noe som skiller Skolelinux fra mange andre linux-distribusjoner er måten brukerdata blir lagret på. I stedet for å lagre brukere og deres passord i spesielle tekstfiler, bruker Skolelinux LDAP til dette. LDAP kan for enkelthets skyld sees på som en slags database. Dette har en rekke fordeler som for eksempel at man kan distrubuere brukeredataene over nett.

I forbindelse med windows-tilknytning byr dette derimot på en utfordring siden windows kan ikke benytte seg av LDAP for autentisering. For ordens skyld kan det nevnes at windows ikke kan få tilgang til tekstfilene som er alternativet heller, men mellomleddet, samba, er bedre testet og dokumentert for bruk sammen med den tradisjonelle måten å lagre brukerdata på. For mac er denne historien ganske annerledes. OS X har full støtte for LDAP og kan autentisere brukere direkte mot tjeneren.

3.3.2 Tildeling av IP-adresser

Tildeling av ip-adresser gjøres automatisk av tjeneren ved help av DHCP dersom klienten er satt opp til å be om dette. Man må med andre ord sette opp klienten til å be om ip-adresse, noe som er standard på en ny installasjon av både windows og MacOS. Dersom man allerede har en ip-adresse satt manuelt, må man gå inn i nettverks-oppsettet på klienten og sette den til å tilordne adresse automatisk via DHCP. DHCP vil også tildele annet nettverksoppsett som netmask og default gateway.

3.3.3 Navnetjener

Tjeneren tar seg av navnetildeling (DNS) til alle maskiner lokalt, og gir videre tilgang til eksterne navnetjenere automatisk. Oppsett av dette vil være automatisk forutsatt at man har satt klienten til å motta dette via DHCP som nevnt over for ip-adresser.

3.3.4 E-post

Skolelinux tilbyr epost-tilgang gjennom Squirrelmail (en webbasert epost-leser), eller direkte mot POP[1]/ IMAP[2]-protokollene. Hvis man velger å bruke protokollene direkte så må brukeren sette opp en egen post-klient på maskinen sin (f.eks. Outlook, Eudora e.l.). Med Squirrelmail, som er en web-basert klient, vil ikke brukeren ha behov for noen ekstra programvare utover en nettleser.

Mac- og Windows-klienter vil i dag få automatisk tilgang til nettverksressurser som f.eks. Skolelinux' Squirrelmail-løsning. Prosjektgruppen anbefaler at disse klientene benytter seg av den webbaserte løsningen. Dette har følgende fordeler:

- Ingen lokal konfigurasjon.
- Meldinger forblir lagret sentralt.
- Unngår å bruke Outlook, som er velkjent for sine sikkerhetsproblemer[3].

3.3.5 Tilgang til web

Skolelinux har en egen web-tjener som kan benyttes både for interne og eksterne websider. For å få tilgang til disse behøver man kun å legge inn url til tjeneren i webleseren (http://tjener).

For å gi klienter tilgang til web ut bruker skolelinux en proxy. Denne må legges inn i klientens web-leser for å kunne komme ut på internett. Til windows finnes det en rekke web-lesere. De mest kjente er Internet Explorer, Netscape og Opera. Alle disse har hvert sitt sted å lagre konfigurasjon og det finnes derfor ingen standard for å legge inn denne informasjonen. Det finnes dog noe som til nøds kan kalles standard for å automatisere dette ved hjelp av DHCP, web-tjener og scripting. Samme problematikk finnes på Mac hvor det også finnes en rekke forskjellige web-lesere.

[1] http://www.faqs.org/rfcs/rfc1939.html
[2] http://www.ietf.org/rfc/rfc2060.txt
[3] http://www.securityfocus.com/search?category=23&query=outlook

3.3.6 Fildeling

I Linux har man flere måter å dele filer med andre maskiner på. Med fildeling mener vi her det som også kalles nettverks-filsystemer. Du kan under visse omstendigheter også se på ftp-tjenere som fildeling, men de har et litt annet bruksområde enn det som det er snakk om her. Et nettverks-filsystem lar deg få full tilgang til filer på samme måte som om de lå på en lokal disk. I utgangspunktet kjører unix/linux, Windows og mac hver sine filsystemer, også over nettverk. Mellom Linux-maskiner brukes som oftest NFS (network file system), mens Windows bruker CIFS (common internet file system), kanskje mer kjent under navnet Samba. Apple har tradisjonelt sett på sin side hatt en egen løsning kalt appletalk.

I dag er det mulig å kjøre flere av disse systemene på tvers av plattformer, men ikke alt fungerer like bra og er like lett å administrere. For Windows-klienter synes det å være helt klart at samba er det rette valget. Først og fremst fordi dette er innebygget i windows fra før, men også ettersom det er en velkjent løsning på tjener-siden som er i bruk i store systemer og har mye dokumentasjon tilgjengelig. Å koble en Windows-klient opp mot NFS eller Appletalk vil kunne la seg gjøre ved hjelp av tredjeparts programvare, men er hverken godt dokumentert eller regnet for å være stabile løsninger.

På mac-klienter stiller dette seg litt annerledes. Dagens operativsystem, Mac OS X har støtte for alle tre tjenestene i varierende grad. Fra gammelt av har den støtte for Appletalk, som ble brukt i eldre versjoner av operativsystemet. På grunn av sitt utgangspunkt i unix har også OS X støtte for NFS, men da først og fremst ved hjelp av kommandoer i et skall. Det som kan se ut til å være den beste løsningen for OS X er samba, på lik linje med Windows. Fra og med versjon 10.2 har OS X på papiret full støtte for samba, også grafisk. Utviklere av netatalk, som er appletalk-tjenesten for Linux, støtter også opp om dette som den beste løsningen for OS X. Appletalk-implementasjonen som finnes pr i dag på Linux har ikke støtte for en del praktisk funksjonalitet som filnavn over 32 tegn. Den har også en veldig tungvindt måte å lagre brukerens passord på (en fil i hver brukers hjemmekatalog) som gjør at vi ikke ønsker å legge denne inn i Skolelinux.

Etter en del testing, og samtaler med macdriftere ved instituttet har vi allikevel besluttet å gå for nfs for fildeling på mac. Med de tjenestene som til dags dato kjører på tjeneren er dette den enkleste og ryddigste måten å gjør det på. Samba ble lagt til side til fordel for nfs ettersom mac allerede har støtte for autentisering med den innebygde støtten for LDAP. Fildeling med nfs er også den fildelingsmetoden som Skolelinux bruker mot tykke klienter og tynnklient-tjeneren. Det ble derfor naturlig for oss å velge denne også for mac. Mac'en vil da bruke de samme tjenestene som linux arbeidsstasjoner, tynnklienter osv for fildeling og autentisering. Flere personer tilknyttet Skolelinux ønsket en løsning basert på samba fremfor NFS. Samba er som nevnt fullt støttet på papiret, men i praksis førte det til proble-

mer i vårt nett. OS X ville rett og slett ikke autentisere opp mot samba-tjeneren. Tilknytning av enkeltshares fungerte, men autentisering i form av pålogging på mac fungerte ikke. Vår knappe tidsramme og begrensede kunnskaper om Mac førte til at NFS ble veien å gå for å få en enkel løsning som fungerte etter de krav som ble satt.

3.3.7 Skriverdeling

På samme måte som for fildeling har man flere alternative protokoller og tjenere for å dele skrivere over nettverket. I utgangspunktet benytter Skolelinux seg av CUPS (common unix printing system) til dette, men kan også dele dem videre via andre systemer som samba og appletalk. Det er ikke snakk om å velge en tjeneste på bekostning av en annen. Samba vil f.eks fungere som et mellomledd mellom windows-skriverdeling og CUPS lokalt på tjeneren.

For Windows-klienter er valget ganske enkelt. Samba tilbyr skriverdeling og dette er det systemet som er innebygget i Windows. Skal man knytte en Windows-maskin direkte opp mot f.eks CUPS vil dette innebære tredjeparts programvare.

Mac OS X bruker samme skriverdeling som skolelinux (CUPS) som standard. Det har også støtte for skrivere som er delt via samba og appletalk. Valg av skrivertilkobling vil derfor være fritt alt etter hva man ønsker og hva som viser seg å fungere best i den gitte situasjonen.

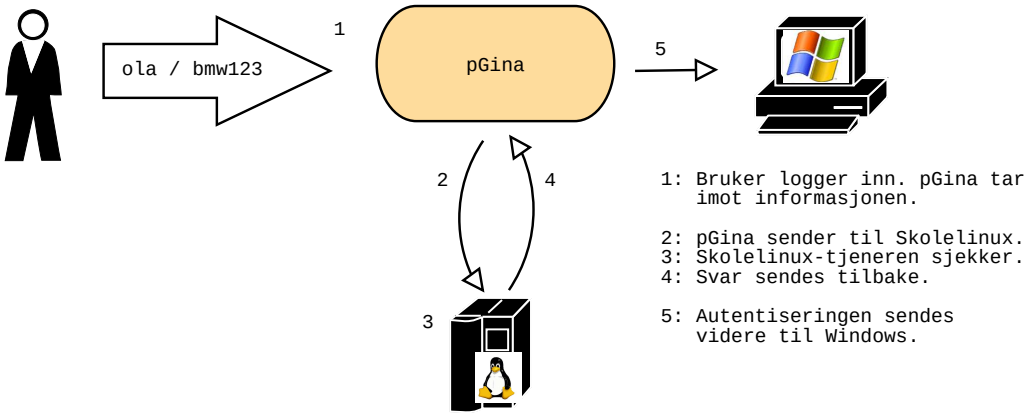
3.3.8 Klokkesynkronisering

For å sørge for at alle klienter har samme tid på sin klokke tilbyr Skolelinux tjener synkronisering via NTP. På mac har man en innebygd klokkefunksjon som kobler seg mot nettverksklokketjenere, mens Windows krever tredjeparts programvare for å koble seg mot NTP. Det er derimot mulig å stille inn klokka på windowsmaskiner ved hjelp av påloggingsscript i samba uten noe ekstra programvare, men dette fungerer kun på windows 9x siden man ikke får lov å justere klokka uten å ha administrator-rettigheter i NT/2000/XP.

3.4 Aktuelle teknologier

3.4.1 pGina

En av de viktigste punktene i forbindelse med mac- og windows-integrering er å få til en felles autentisering. Med dette menes at man kun har en brukerdatabase



Figur 3.2: Autentisering med pGina.

og at brukerene benytter seg av samme brukernavn og passord for å autentisere seg, uansett hvilket operativsystem klienten kjører.

pGina[4] er et nytt autentiseringsgrensesnitt som er under utvikling. Dette fungerer rundt samme prinsipp som PAM gjør på linux, men gjør det mulig å bytte ut windows' standard passordkryptering med andre moduler. Den eneste modulen som finnes pr i dag er laget for LDAP og kan i teorien settes opp til å kjøre sammen med Skolelinux. Dessverre er pGina ikke egnet for skolelinux på grunn av sin dårlige støtte til eldre windowsversjoner. pGina er kun laget for Windows 2000 og XP. Den kan muligens portes bakover til NT4, men det er ikke overhodet støtte for Windows 9x eller Windows ME som det finnes mye av ute i skolene pr dags dato. Dette vil med andre ord ikke gå overens med kravet om hvilke operativsystem som skal støttes i kravspesifikasjonen og vil derfor ikke bli brukt i dette prosjektet.

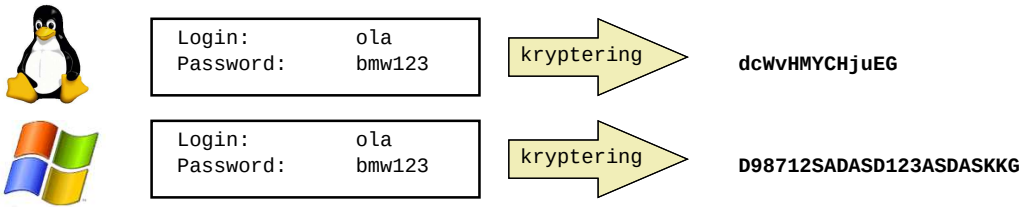
3.4.2 Samba

Samba er en stor programbareløsning som kan tilby store deler av det man trenger for å få windowsklienter opp mot en linux-tjener. Den kan ta seg av alle tre punktene som er vektlagt i dette prosjeket: autentisering, fil- og skriverdeling. I og med ta det er en stor løsning, betyr det også at den har mange punkter som må undersøkes og redegjøres for før en implementasjon.

En av de store utfordringene med samba/windows sammen med Linux er passordautentisering. Disse to systemene kjører hver sin krypteringsalgoritme slik at Windows kun kan autentisere passord som er kryptert med windows' algoritme, mens linux kun kan autentisere passord som er kryptert med linux' algoritme. Det

[4]http://pgina.xpasystems.com/

finnes flere forskjellige måter å få disse systemene til å samarbeide, alle har sine fordeler og ulemper.



Figur 3.3: Ulike krypteringsalgoritmer.

ukrypterte windowspassord

Den metoden som har blitt brukt en del opp igjennom tiden er å sette opp Windows til å sende og motta passord ukryptert over nett. Dette forenkler autentiseringen ettersom Linux-tjeneren da får passordet i klartekst og kan sjekke dette opp mot sin egen nøkkel. Den store ulempen med dette er at man da har veldig dårlig sikkerhet. Man kan på enkelt vis sniffe nettverket og få passordene frem i klartekst. Dette gjør metoden uegnet for bruk med Skolelinux ettersom det bryter med de kravene til sikkerhet et slikt system har.

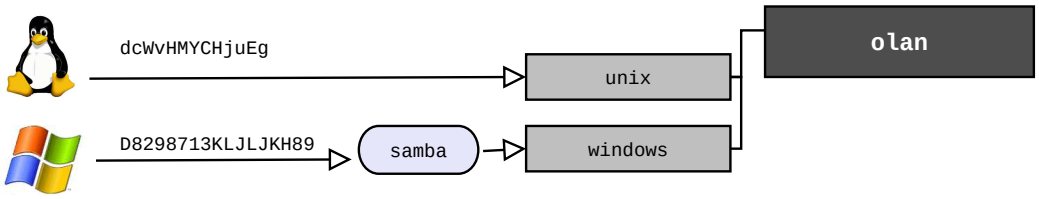
PAM vs LDAP

Skolelinux støtter et grensesnitt for autentisering kalt PAM. Dette fungerer som en generell "plugg" hvor man kan putte inn moduler for forskjellige systemer. Det finnes en flere pam-moduler for samba, ment å benyttes til forskjellige oppgaver. En av dem gjør det blant annet mulig å autentisere brukere mot en ekstern NT-server, men er ikke til nytte i denne sammenhengen.

Selve samba-tjeneren har derimot mulighet for å autentisere innkommende brukernavn og passord mot linux via PAM. Dermed kan dette benyttes til å la samba kommunisere mot LDAP siden det finnes en PAM-modul som tilbyr dette. Selv om dette er et steg i riktig retning er man allikevel tilbake til problemet omkring ukrypterte passord. Det er ikke mulig for samba å dekryptere et windows-passord for så å sjekke det opp mot Linux' autentiseringssystem, ei heller andre veien. I det du logger på en windows-maskin vil den motta passordet og kryptere det med sin algoritme. Deretter vil denne krypterte passord-hashen bli sendt over nettet til samba-tjeneren. Samba mottar den krypterte hashen, men kan ikke sammenligne denne med den hashen som Skolelinux har i sin brukerdatabase siden den er laget med en annen algoritme.

Det som derimot er nyttig i denne sammenheng er å godkjenne brukeren. Vi kan benytte denne funksjonaliteten til å undersøke om en bruker finnes på systemet før vedkommende får lov å logge inn, eller i det hele tatt å bli opprettet for innlogging via windows. Ingen brukere skal kunne ha tilgang fra windows uten å først være lagt til i linux.

Et kompromiss



Figur 3.4: Separat lagring av ulike passord-hasher

Skal man beholde sikkerheten med tanke på krypterte passord over nett, og samtidig ha støtte for alle versjoner av windows (og helst uten å modifisere klienten), vil det eneste alternativet være å lagre separate passord-hasher for hver enkelt bruker. Som vist i figuren over har man da en hash for linux, og en for windows for hver enkelt bruker (brukernavn i dette tilfellet er “olan”. Dette kan føre til problemer med at disse passordene ikke er synkronisert. I praksis vil det bety at brukeren da risikerer å ha forskjellige passord på windows og linux. Derfor er det viktig å få til en samkjøring av oppdateringen av passord mellom skolelinux og samba i et felles grensesnitt. Endrer man passordet til en bruker, så skal man endre det både i Linux og samba samtidig. Slik kan man klare å opprettholde samme passord for brukeren til en hver tid uavhengig av hvilken maskin han eller hun logger inn på.

Klientkontoer

Det finnes enda et viktig punkt man må ta med i forbindelse med itegrering av samba i Skolelinux. NT-baserte utgaver av windows (NT4, 2000 og XP Pro) har en helt annen nettverksarkitektur enn win95,98 og ME. For at en maskin som bruker et av disse operativsystemene skal kunne logge seg på et domene, må det i tillegg til en konto tilhørende brukeren, også finnes en konto tilhørende maskinen. Har man for eksempel en Windows 2000-klient som har fått navnet “windows2” så må man ha en konto på tjeneren som heter “windows2”.

Dette er bare en tom konto som det ikke er mulig å logge inn med, men er der for å gi en økt sikkerhet og kontroll med hvilke maskiner som skal få lov å

logge seg på ditt (windows)domene. Utfordringen med dette i forbindelse med Skolelinux blir å få disse spesielle kontoene inn i LDAP og da helst automatisk for å slippe unødig arbeid ved oppkobling av nye maskiner til nettverket.

3.4.3 Webmin

Skolelinux bruker webmin som administreringsverktøy. Her er det under utvikling en egen modul for å administrere brukere. Dessverre er ikke denne klar før dette prosjektet skal være avsluttet, men det finnes en "forsmak" på denne allerede som blir brukt som utgangspunkt for de som skal utvikle den endelige modulen. Webmin har også en modul som er laget for å konfigurere samba. Denne har støtte for å samkjøre oppdatering av brukerdata mellom Linux og samba, men da kun med den tradisjonelle "users & groups"-modulen som er laget for brukerdata lagret i tekstfiler og ikke med LDAP-administrasjonen som Skolelinux bruker. Måten denne er implementert på er at den hekter seg på en hver oppdatering av brukeren på linux og utfører samme opptadering i samba. En fornuftig måte å gjennomføre samkjøringen av bruker/passord-oppdateringen på i skolelinux er å skrive en ny modul eller funksjon som tilsvarende hekter seg på oppdateringer i LDAP-administrasjonen.

3.4.4 CUPS

Skolelinux tilbyr skriversystemet CUPS som en del av sin arkitektur. CUPS står for "Common Unix Printing System", og er en komplett pakke for deling og administrasjon av skrivere under UNIX-baserte operativsystemer.

Dette kan enkelt integreres med samba, og vi får tilbudt skrivere til windows-maskiner.

3.4.5 cfengine

Siden skolelinux er en spesialisert distribusjon, betyr det at en rekke konfigurasjonsfiler er ferdig satt opp for å passe inn i et standardisert nett. For å sette opp dette benyttes et program kalt cfengine. Konfigurasjonsendinger som blir gjort på tjeneren i forbindelse med dette prosjektet må derfor legges inn i den pakken som inneholder filene til cfengine for å bli med i distribusjonen.

3.5 Oppsummering

Som det står fastsatt i kravspesifikasjonen vil hovedfokus i dette prosjektet bli lagt på fil- og skriver-deling. Med dette medfølger også autentisering ettersom man ikke kan gi tilgang til noen av disse tjenestene før brukeren først er autentisert. De andre tjenestene vil ellers høre til to grupper: En som i utgangspunktet er støttet av klienten og dermed ikke trenger noen modifikasjon, og en annen som ikke lar seg implementere til samme grad og dermed heller blir forsøkt beskrevet hvordan man kan ta i bruk.

Ved å sette opp tjenestene i en tabell og dele dem inn i disse gruppene får man følgende:

Tjeneste:	Std. mac/win	implementeres	beskrives
Tildeling av ip (DHCP)	X		
Navnetjener (DNS)	X		
E-post			X
Tilgang til webtjener	X		
Web-proxy	X		X
Fildeling	X	X	X
Skriverdeling	X	X	X
Klokkesync		X	X

Tildeling av ipadresser og navnetjenester er som tidligere beskrevet fullt støtte av både windows og mac som standard. Det samme må kunne sies om tilgang til webtjeneren siden begge systemer leveres med en eller annen form som web-leser og har en rekke gratis alternativer til disse. E-post og web-proxy krever litt mer arbeid for å sette opp og vil derfor bli beskrevet i grove trekk i dokumentasjonen til de respektive plattformene. Fildeling, skriverdeling og klokkesynkronisering vil bli forsøkt implementert så langt det lar seg gjøre og tilhørende dokumentasjon skrives. Siden valgene gjøres for å unngå så mye klientmodifisering og tredjeparts programvare som mulig, kan man også sette fil- og skriver-deling inn som standard støttet.

Ut fra denne tabellen kan man trekke frem at fil- og skriverdeling samt klokke-synkronisering skal implementeres i dette prosjektet, mens resten bare skal dokumenteres i denne omgang.

Kapittel 4

Utviklingsrapport

4.1 Forberedelser til windows-støtte

4.1.1 Installasjon av samba

Skolelinux er i utgangspunktet en linux-distribusjon kalt debian som har blitt videreutviklet og modifisert. Flesteparten av programmene som følger med er de samme som finnes til debian og hentes automatisk ned når det lages en ny cd av skolelinux. For å kunne ta i bruk samba i skolelinux har vi et par krav som ikke blir møtt av de standardpakkene med samba som følger med debian. For det første må samba være kompilert med støtte for ldap. Dette medfører også at det hele må baseres på en sambaversjon nyere enn 2.2.4 for å ha en stabil støtte for ldap. På grunn av dette er det kompilert og laget egne pakker av samba for skolelinux. I etterkant er det også blitt lagt til et ekstra verktøy i den ene pakken for å kunne hjelpe til med automatiseringen.

4.1.2 Konfigurasjon av samba

Når man har en ferdig klargjort programvare må den så konfigureres for å passe inn i systemet. Det følger med en standard konfigurasjonsfil i sambapakkene, men denne egner seg ikke særlig til bruk med skolelinux. Derfor har gruppen valgt å opprette en ny konfigurasjon spesielt for skolelinux som blir lagt inn som erstatning via cfengine. Detaljene i denne kan sees i tillegg B på side 50. Navnet smb-skolelinux.conf kommer av at den er spesifikk for skolelinux. Cf-engine vil ved installasjonen lenke denne til smb.conf og dermed overskrive den eksempelfila som følger med samba (og dermed aktivere den nye konfigurasjonen).

Noen av de viktigste valgene som er satt opp her er arbeidsgruppen “skolelinux”, konfigurasjon for å koble seg opp mot ldap og oppsett av skriversystem. Videre er hjemmekataloger med tilhørende rettigheter satt opp, samt et område for

pålogging med tilhørende script. Se gjerne dokumentasjon på “http://www.samba.org” og manualsiden til smb.conf for en mer detaljert beskrivelse av de forskjellige kommandoene som er brukt.

4.1.3 Konfigurasjon av LDAP

For at samba skal kunne lagre sine data i LDAP må selve LDAP-tjenesten konfigureres. Til dette trengs et såkalt “shema” som beskriver hvilke data som skal legges inn og hvor de skal plasseres. I dette prosjektet blir det benyttet et schema som er hentet fra “Samba (v 2.2) PDC LDAP v.3 howto”[6]. Dette gjøres for å sikre et så standard oppsett i LDAP som mulig. Det er spesielt viktig med tanke på eventuelle oppgraderinger til nyere versjoner av samba i framtiden. Versjon 3 av samba kommer til å ha en utvidet støtte for LDAP og vil trolig basere sitt oppsett rundt et schema som ligner på dette. Tillegg B.1 på side 52 gjengir dette schemaet i sin helhet.

En siste ting som kreves for at LDAP-tjeneren skal benytte seg av schemaet, er at det ligger en referanse til dette i konfigurasjonsfilen slapd.conf. På samme måte som for konfigurasjonsfilen til samba er dette en erstatningsfil og filen endringene er gjort i heter da slapd-skolelinux.conf. Endringene som har blitt gjort i konfigurasjonen er da å legge inn referanse til samba.schema og inetorgperson.schema. Sistnevnte må med ettersom samba.schema er avhengig av referanser i dette. De to linjene som er lagt inn ser ut som dette:

Listing 4.1: endringer i slapd-skolelinux.conf

```
include         /etc/ldap/schema/inetorgperson.schema
include         /etc/ldap/schema/samba.schema
```

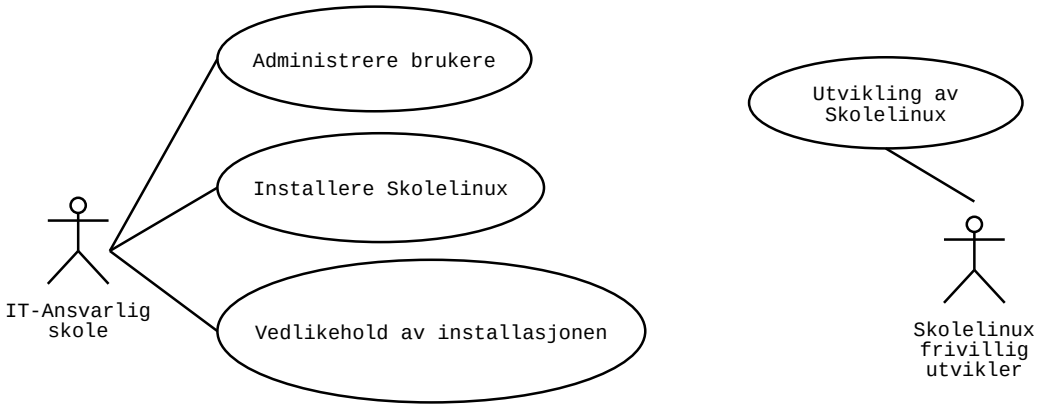
4.2 Implementering av script for windows

4.2.1 Use Case

Vår del av implementeringen vil bestå av et lite program for å samkjøre data fra et administreringsverktøy og en brukerdatabase som benyttes av windows. Dette vil ikke være synlig for brukren da det blir kallet opp av systemet der det behøves. En litt vanskelig situasjon oppstår ettersom administreringsverktøyet utvikles av en annen studentgruppe og ikke er ferdig tidsnok til at vi får tatt den i bruk.

Vårt prosjekt må derfor benytte seg av et minimalistisk verktøy for å legge til og fjerne brukere som ligger i distribusjonen i dag. Denne har ingen teknisk dokumentasjon i form av UML eller usecase-diagrammer og vi må derfor foreta en del grove antakelser på dette stadiet.

Business use case



"IT-Ansvarlig skole" er en som er ansatt ved/av en skole og har ansvar for administrasjon av skolens IT-tilbud. "Skolelinux frivillig utvikler" er en som engasjerer seg for Skolelinux og jobber for videreutvikling av prosjektet. Vårt prosjekt skal tilrettelegges slik at IT-ansvarlig ikke får mer arbeid med "Administrere bruker"-biten.

Selve administreringsverktøyet er det altså en annen gruppe som skal ta seg av, dette er kun ment for å skissere hvordan administrator skal benyttet det fremtidige administrasjonsverktøyet som igjen vil kalle opp programbiten vi skal lage. Som de fleste andre distribuerte prosjekter basert på åpen kildekode er mangel på god dokumentasjon for utviklere et problem og vi bruker derfor et grovmasket nett når vi beskriver hva som skal skje.

Opprette bruker

Aktør	Systemet (webmin-modul for brukeradministrasjon)
Prekrav	Administrator må ha fylt inn den nødvendige informasjonen i webmin for å legge til en bruker og trykket på knapp for å legge til bruker. Forutsetter at webmin-modulen luker ut potensielle feil som at brukeren allerede finnes på systemet.
Hovedflyt	1. Webminmodulen kaller opp script for synkronisering av brukerdata. 2. Script mottar nødvendig informasjon fra webminmodul på kommandolinje. 3. Script oppretter bruker ved hjelp av smbpasswd. 4. Script returnerer beskjed om operasjon er vellykket eller om en feil oppstod.
Avvik	Ingen

Endre passord

Aktør	Systemet (webmin modul for brukeradministrasjon)
Prekrav	Nytt passord må være fylt ut i webminmodulen og administrator må ha valgt knappen for å endre passord. Krever at webminmodulen ikke tillater operasjonen dersom brukeren ikke finnes
Hovedflyt	1. Webminmodulen kaller opp script for synkronisering av brukerdata. 2. Script mottar brukernavn og passord fra webminmodul på kommandolinje. 3. Script oppdaterer passord ved hjelp av smbpasswd. 4. Script returnerer beskjed om operasjon er vellykket eller om en feil oppstod.
Avvik	Ingen

4.2.2 Script for bruker- og passord-synkronisering

Skolelinux benytter seg av et egenutviklet brukergrensesnitt for å administrere brukere på web. Dette kaller de *webmin-ldap-skolelinux*.

Det er meningen at dette brukergrensesnittet skal byttes ut med et som lages av seks studenter ved NITH[1]. Dessverre er ikke det prosjektet over før i mai. Gruppen har derfor gitt beskjed til utviklerene av det nye systemet om hvilke føringer dette prosjektet legger på den jobben de skal gjøre.

I mellomtiden har vi valgt å modifisere det eksisterende brukergrensesnittet litt, slik at man fikkk demonstrert vårt prosjekt på en best mulig måte. Disse modifikasjonene går ut på at brukere som opprettes må få noe tilleggs-informasjon for at de skal kunne logge inn fra windows også i form av data fra samba. Det er lagt til en opsjon i konfigurasjonen av webmin-ldap-skolelinux som spesifiserer om brukere skal synkroniseres eller ikke. Når denne opsjonen er slått på, vil et ekstra script bli kjørt når brukere opprettes (se tillegg C.1 på side 59 for kildekoden). Selve webgrensesnittet vil også vise informasjon om brukere er synkroniserte eller ikke i form av et grønt eller rødt ikon som indikerer det.

Scriptet som utfører oppdateringen fungerer som et eget lite selvstendig program man kjører fra kommandolinjen. Det er ikke avhengig av andre komponenter bortsett fra smbpasswd, som er en kommando tilhørende samba. Denne blir kalt opp og kjørt ved behov. Scriptet arver ingen egenskaper og har bare en funksjon

[1] Norges Informasjonsteknologiske Høgskole; http://www.nith.no/

som utfører hele jobben. Integreringen av dette scriptet i webmin gjøres ved å kalle det opp som en ekstern kommando:

Listing 4.2: Oppkall av script

```
# check if sambasync is set, add user if it is
if ($config{'sambasync'}) {
  # hack to make sure root has a samba-account (first time run)
  my $rootuser = &ldap_get_user($config{'basedn'},"root");
  if (not $rootuser) {
    run_script("smb_create.pl", latin1("root")->utf8, $rootpw, $rootpw);
  }
  # add the new user to samba
  run_script("smb_create.pl", latin1($uid)->utf8, $userpw, $rootpw);
}
```

Dersom "sambasync" er satt, vil våre endringer blir utført. Sambasync blir satt av eller på i konfigurasjonen til webmin-modulen. Det første som gjøres er å sjekke om root finnes i samba, dette må den gjøre og blir derfor opprettet ved å kalle opp smb_create.pl med "root" som brukernavn og rootpassord som brukerens passord. Root må eksistere i samba for at de nødvendige rettighetene ved innmelding av windowsklienter skal være på plass. Deretter kjøres et nytt kall til smb_create.pl med brukernavnet til den nylig opprettede brukeren, det nye passordet og passordet til root (for å få lov til å legge inn brukerdata). Dette vil bli kjørt uavhengig av hva den første sjekken om root-konto har medført.

Siden det bare finnes to operasjoner i det lille administreringsverktøyet som ligger der nå, oppretting og sletting av brukere, er det også kun de to operasjonene man får utført med disse kallene. Til tross for dette er smb_create.pl laget slik at det også vil fungere ved oppdatering av passord. Dersom brukeren (brukernavnet) allerede finnes, vil bare passordet bli endret. Sletting av brukere krever ingen modifikasjoner da rutinene for å fjerne brukere fra LDAP tar seg av det på egenhånd.

Manualside for smb_create.pl

```
smb_create.pl

Script for å oppdatere en brukers windows-passord. Oppretter brukeren i
samba dersom den ikke finnes. Returnerer beskjed til stdout om brukeren er
lagt inn eller om en feil oppstod.

 Bruk:

   smb_create.pl $<$brukernavn$>$ $<$passord$>$  $<$ldap-admin-passord$>$



 Interne funksjoner:
```

```
  sub change_samba(student_id, newpass, rootpass)
  - Hovedrutine som øutfrer selve oppdateringen.
Interne variabler:
  student_id - brukernavn
  newpass    - brukerens (nye) passord
  rootpass   - rootpassord (ldap admin passord)
  tempfile   - filnavn til temp-fil
  ret        - returmelding fra smbpasswd
  retmsg     - returmelding fra smb_create.pl
```

4.2.3 Automatisering av klientkontoer

Som nevnt i utredningen krever windows NT-baserte operativsystem at hver enkelt klient får sin egen konto. En slik konto benytter i samba dollartegnet $ bak maskinnavnet, og skiller seg sådan ut fra de vanlige brukerkontoene også visuelt. Den har selvfølgelig ikke de samme mulighetene som andre kontoer. Den har ikke noe passord, ikke hjemmekatalog og ikke noe tilhørende skall. Alt dette gjør det umulig å logge inn via en slik konto. Den er kun til for å autentisere selve maskinen mot tjeneren.

Å opprette slike kontoer manuelt for hver enkelt NT-klient man skal ha i sitt nett er både tidkrevende og unødvendig. Samba har mulighet for å opprette slike kontoer automatisk for hver gang du melder inn en ny maskin i ditt domene (som root). Maskiner legges inn i domenet med en spesiell veileder innebygget i windows. På tjenersiden gjøres dette ved at samba da kaller opp et eksternt script. Som oftest benyttes kommandoen “adduser” som finnes på de fleste systemer. Dette oppretter unix-kontoen, og deretter ordner samba sine samba-spesifikke data selv.

I Skolelinux er derimot adduser lite egnet, ettersom den bruker passordfiler for å lagre sine data i stedet for LDAP. Vi har derfor laget et eget script i perl som tar seg av dette. Se tillegg C på side 55 for å studere kildekoden. Dette scriptet kan kun brukes for å opprette klient-kontoer, og tar derfor kun ett argument fra kommandolinjen – klientnavnet (med dollartegn). Alle andre data blir satt av scriptet. Vi har valgt å reservere et sett med bruker-id og en gruppe-id til disse kontoene. En gruppe “machines” er oppretet med gid=70, og bruker-id velges fra 7000 og oppover, dette gjør at man ikke får konflikter med de nummerseriene som er reservert for ordinære brukere som får tildelt tall fra 10000 og oppover. Det vil riktignok bli en konflikt om man da har over 3000 windows-klienter i sitt nett, men vi ser det

som et lite sannsynlig antall å ha på en skole og velger derfor å akseptere denne begrensningen.

Rent praktisk har et slikt script en del utfordringer. Til forskjell fra et system som lagrer brukere i passordfiler, krever skolelinux en del mer informasjon for å la et script koble seg opp mot LDAP og opprette brukere. Følgende informasjon er nødvendig:

- ldap server
- ldap basedn
- ldap admin dn
- ldap admin passord

Ettersom scriptet skal kjøres automatisk av samba, er det ikke mulig å hente inn noe av dette fra brukeren, ikke en gang passordet. LDAP admin passord er i realiteten det samme som root-passord. Samba har denne informasjonen i sin konfigurasjonsfil, bortsett fra passordet som lagres i en spesiell .tdb-fil som kun er tilgjengelig for root. For å få disse dataene går derfor vårt script først inn i samba konfigurasjonen (smb.conf) og henter ut ldap-konfigurasjonslinjene her. Deretter benytter den tdbdump-verktøyet som følger med samba for å gå igjennom secrets.tdb og henter ut adminpassordet der. Det er en omfattende jobb, men fungerer på samme måte som samba selv gjør det og man er dermed garantert at konfigurasjonen er oppdatert så sant samba er det.

Når disse dataene er tilgjengelig har scriptet alt som trengs og kan benytte ldap-kommandoer for å gå inn og opprette kontoer. Det sørger først for at machines-gruppa eksisterer og går deretter igjennom de eksisterende klient-kontoene for å finne laveste ledige uid innenfor det reserverte området. Dette er samme fremgangsmåte som modulen for å legge til brukere i webmin benytter seg av. Når en ny ledig id er funnet opprettes kontoen med standardverdier for hjemmekatalog, skall osv. Samba overtar etter at kontoen er opprettet og ordner sine windows-spesifikke data som tillater maskinen å logge på domenet.

4.2.4 Loginscript

For å kunne utføre kommandoer ved pålogging bruker windowsklienter en batchfil den får tildelt fra tjeneren. I vår konfigurasjon har vi valgt å kalle denne “login.bat”. Se tillegg C.2 på side 60 for kildekoden.

Det finnes en rekke mer eller mindre avanserte operasjoner man kan foreta seg i et slikt script, men dette faller litt utenfor vårt område å gå i detalj på hvordan slikt gjøres. Bøker og websider om administrering av windows-nettverk har mye

tips og triks om dette temaet. Vår lille standardfil er ment som et eksempel som man selv kan modifisere og legge til ekstra funksjoner ettersom man måtte ha behov.

Slik det er lagt opp nå gjør det følgende tre ting:

- Kobler opp hjemmeområde
- Sykroniserer klokke
- Oppretter snarvei til hjemmeområde på skrivebord

Oppkobling av hjemmeområde gjøres av scriptet ved hjelp av "net use" kommandoen til windows (se kildekode). Dette gjelder først og fremst til windows 9x/ME siden windows NT-baserte systemer kobler opp hjemmeområdet av seg selv uten at denne kommandoen blir kjørt.

Listing 4.3: Oppkopling av hjemmeområde

```
net use h: /HOME /yes
```

Klokkesynkronisering gjøres også ved hjelp av "net" kommandoen til windows (se kildekode). Dessverre fungerer dette kun på systemer basert på windows 9x/ME som standard siden windows NT-baserte systemer nekter andre enn administrator å få endre klokka.

Listing 4.4: Klokkesynkronisering

```
net time \\tjener /set /yes
```

Opprettelsen av snarvei til hjemmeområde er lagt inn for å vise hvordan man kan kalle opp script skrevet i andre språk eller rene programfiler for å utføre mer avanserte oppgaver enn batchscriptet i utgangspunktet tillater. Dette gjøres ved at den kaller opp et vbs-script kalt "shortcut.vbs" (se tillegg C.3 på side 60) som ligger på tjeneren.

Listing 4.5: Oppstart av annet script

```
start \\tjener\netlogon\shortcut.vbs
```

4.2.5 Avansert loginscripting

Som nevnt kan man utvide funksjonaliteten til et login-script ved å kalle opp andre scriptspråk eller programvare. Vi har valgt å lage et lite vb-script som oppretter en snarvei på skrivebordet. Andre scriptspåk kan sikkert med fordel benyttes, vi valgte vbscript i dette tilfellet for å slippe å måtte inkludere ekstra (runtime) programvare. Vbscript forutsetter at du har et system som er oppgradert ved hjelp av Microsoft Update (følger typisk med oppgradering av Internet Explorer).

Kildekoden finnes i tillegg C.3 på side 60. Koden er forhåpentligvis ganske selvforklarende for den som kjenner til dette språket. Hovedpoenget er ikke hva scriptet gjør, men at det viser hva som lar seg gjøre ved hjelp av eksterne scriptspråk som kalles fra batchfilen. Mer avanserte script kan også inneholde subrutiner som blir kallet opp avhengige av forskjellige kriterer (for eksempel hvilken windowsversjon det kjøres på).

4.3 MacOS X

4.3.1 Oppsett av OS X mot Skolelinux

I motsettning til Windows har OS X innebygd støtte for tjenestene som i Skolelinux brukes for å autentisere brukere og dele filer. En maskin med OS X vil derfor kunne virke veldig likt en arbeidsstasjon med linux i Skolelinux nettverket.

OS X maskinen vil montere brukerenes filer under oppstart ved hjelp av nfs. Når en bruker logger inn vil mac'en sende forespørsel til tjeneren for å sjekke brukernavn og passord. Stemmer dette blir brukeren logget inn, og får tilgang på sine egne filer fra tjeneren.

Vi har skrevet en howto på hvordan å konfigurere tjener og OS X klient for å virke sammen i et Skolelinux nettverk, samt et skript for å sette opp nfs monteringen.

Kapittel 5

Testing

5.1 Testrapporter

En rekke tester har blitt foretatt underveis i prosjektet. De fleste for å se til at alle filene som ble lagt inn i CVS kom med ved oppgradering av systemet. Dette for å sikre seg at de som eventuelt måtte ønske å prøve ut dette eksternt er sikret et oppdatert system. Testene ble også brukt for å luke ut bugs fra konfigurasjonsfiler og script. En del bugs vil ikke bli oppdaget på et system man har jobbet med en stund (konfigurasjon og diverse innstillinger henger igjen), det har derfor vært viktig å installere på ny for hver gang man har gjennomført slike tester.

Det har også blitt holdt flere demonstrasjoner overfor oppdragsgiver med presentasjon av hva som har blitt implementert underveis. To av disse ble holdt på kontoret på NTNU, mens en avsluttende demonstrasjon ble holdt på Brundalen VGS 24. april.

5.1.1 Test 1

Dato

17.03.2003

Målsetning

Testen ble utført for å se at windows 98 fungerer mot vårt oppsett før demonstrasjon neste dag og at brukere blir opprettet mot samba.

Hva ble gjort

Installert to klienter med windows 98, konfigurert dem til å koble opp mot sambatjeneren. Tjener er vår opprinnelige test-installasjon som vi har startet utviklingen

på. Brukere er opprettet tidligere, delvis manuelt på grunn av manglende funksjonalitet i koden ennå.

Resultat

Windows 98 lar seg koble opp mot systemet, loginscript ble ikke kjørt. Feilsøkt og endret samba konfigurasjonsfil, testet ok. Hjemmekatalog ble ikke koblet opp rett. Endrer på loginscript, testet ok.

5.1.2 Test 2

Dato

7.04.2003

Målsetning

Testen ble utført for å se at windows 2000 og XP fungerer mot vårt oppsett. Macdel prøves også ut før neste demonstrasjon.

Hva ble gjort

Installert ny tjener. Oppdatert via apt. Lagt til windowsmaskiner manuelt i samba (dette er ikke automatisk ennå). Oppdatert innstillinger på windows XP for å la den kjøre mot samba. Forsøkt å legge en windows XP og en windows 2000-maskin inn i samba. Brukere ble lagt inn (test1-4) og disse ble så forsøkt brukt på de to windowsklientene og mac.

Resultat

Filene ser ut til å være på plass. Det mangler ennå automatikk i å legge til nt/2k/xp-maskiner (kjent problem). Maskinene lot seg ellers melde inn i domenet. Brukere ble korrekt opprettet og fungerte ved bruk både fra W2K, XP og OS X.

5.1.3 Test 3

Dato

20.04.2003

Målsetning

Denne testen er nok en gang foretatt for å se at alle våre endringer kommer med ved oppgradering av systemet og at det fungerer å legge til brukere via webmin-ldap i samba. Testing av mactilknytning ble også foretatt.

Hva ble gjort

Installerte PR37, la til woody-testing i sources.lst og kjørte en full oppgradering. Deretter ble de nøvendige filer kopiert over (følger med cfengine) og schema aktivert i slapd.conf. Alt i henhold til test-veiledningen på web. Etter en restart av de involverte tjenestene ble det aktivert windows-syknronisering i webmin-ldap og brukere ble forsøkt lagt inn. Mac oppkobling fungerte greit, bortsett fra at det krever mye manuelle instillinger.

Resultat

Nok en gang ble ikke brukere lagt inn. Feilsøking viste at ldap-admin passord ikke ble satt av samba. Feilsøkte smb_create.pl og fant skrivefeil i pathname: sbin ble endret til bin og testen foretatt på ny. Denne gangen fungerte alt som det skulle. Endring i script ble lagt inn i cvs.

5.1.4 Test 4

Dato

20.04.2003

Målsetning

Følgende test ble gjort for å prøve brukeroppdateringen i større skala ved å legge inn en rekke brukere og deretter visuelt sjekke og ta stikkprøver av samba-login. Ferdig installert system med alle oppgraderinger ble brukt.

Hva ble gjort

Lagt inn 100 brukere (kalt test00 til 99) ved hjelp av et lite script. Dette ble gjort både mot linux og samba-biten med samme kall som webinterfacet ellers gjør. Visuelt undersøkt tilfeldig utvalgte brukere ved hjelp av vlad. Deretter ble 10 tilfeldig utvalgte brukere testet med login via smbclient.

Resultat

Alle brukere så ut til å bli lagt inn uten problemer. Alle brukere som ble forsøkt logget inn på samba med sitt tilhørende passord fungerte.

5.1.5 Test 5

Dato

23.04.2003

Målsetning

Testen ble foretatt for å se til at alle endringer var på plass og at ting fungerte som planlagt i forkant av demonstrasjonen neste dag.

Hva ble gjort

Installerte PR37, la til woody-testing i sources.lst og kjørte en full oppgradering. Deretter ble de nøvendige filer kopiert over (følger med cfengine) og schema aktivert i slapd.conf. Alt i henhold til test-veiledningen på web. Etter en restart av de involverte tjenestene ble det aktivert windows-syknronisering i webmin-ldap og brukere ble forsøkt lagt inn.

Resultat

Oppdaget feil ved innlegging av brukere. Feilsøkt webmin-modul og funnet en variabel som var blitt byttet om. Rettet feil, lagt inn endring i cvs og testet på ny. Alt fungerte som det skulle. Fant warning i sama-logg, lagt til en linje i konfigurasjon (printcap name), oppdatert i cvs. Warning er borte.

Kapittel 6

Konklusjoner

6.1 Egne erfaringer

I løpet av dette 5 vekttalsfaget i prosjektarbeid har vi fått erfaring i å jobbe mot en ekstern kunde i et utviklings/imlementasjons-prosjekt.

Dette prosjektet har vært forskjellig fra de andre prosjektene faget hadde å velge i mellom, da dette ikke har vært et rent utviklingsprosjekt. Her har det vært mer implementasjon og oppsett enn ren programutvikling. Vi har likevel vært igjennom de fleste fasene som et vanlig utviklingsprosjekt har; utreding av brukerkrav, formell spesifikasjon, utvikling, testing, demonstrasjon og nå sluttrapport.

Prosjektet har vært en nyttig erfaring i å jobbe i en prosjektgruppe hvor en ekstern kunde setter krav.

6.2 Kravspesifi kasjon

Gruppen brukte ganske lang tid på utvikling og finpuss på kravspesifikasjonen. Det har sannsynligvis lønnet seg, da det ikke har vært nevneverdige avvik fra denne kravspesifikasjonen under prosjektets gang.

6.3 Risikovurdering

Gruppen har ikke fulgt opp risikovurderingen i særlig grad. Den burde bli kontinuerlig oppdatert, og det burde bli referert til konkrete risikoforebyggende tiltak underveis. Heldigvis har gruppen vært skånet for store problemer underveis.

6.4 Løsningen

Gruppen er godt fornøyd med løsningen som er levert. Kravene fra spesifikasjonen er oppfylt, og løsningen er demonstrert for kunde og sluttbruker. Jobben vi har gjort er av en slik natur at det innbyr til å jobbe videre med Skolelinux etter prosjektets avslutning.

Bibliografi

[1] Ole J. Utnes *Manifest for Linux i skolen* http://www.linuxiskolen.no/omlis/manifest.html

[2] Norsk Språkråd *Oppheving av unntaket fra kravet om språklige parallellutgaver for kontorstøtteprogrammer ("administrativ programvare") som benyttes i grunnskolen og videregående opplæring* http://www.sprakrad.no/ufd2002.htm

[3] Britt Wang Løvvik *"Norges største dugnadsgjeng"* SkoleMagasinet 3/2002. http://developer.skolelinux.no/info/samisk/sm_3_2002_s9.pdf

[4] RFC2119 *Key words for use in RFCs to Indicate Requirement Levels* http://www.ietf.org/rfc/rfc2119.txt

[5] David Lechnyr *The Unofficial Samba HOWTO* http://hr.uoregon.edu/davidrl/samba/

[6] Ignacio Coupeau *SAMBA (v 2.2) PDC LDAP v.3 howto* http://www.unav.es/cti/ldap-smb/ldap-smb-2_2-howto.html

Tillegg A

Ordliste

A.1 Innledning

Denne ordlista er til for å forklare en del av faguttrykkene vi har brukt i denne rapporten. Ikke alle som leser denne har inngående kjennskap til Linux, og vil vi prøve å dekke de mest brukte fagtermene.

cfengine

Cfengine er en avansert løsning for å utføre diverse automatiserte konfigurasjonsoppgaver. Den kan f.eks oppdatere konfigurasjonsfiler, passe på at filer ikke blir endret osv. Skolelinux benytter cfengine til å installere og sette opp konfigurasjonsfiler ved installasjon.

LDAP

LDAP står for Lightweight Directory Access Protocol, og kort fortalt er dette en slags database over brukere med tilleggsinformasjon. LDAP kan brukes til en rekke forskjellige oppgaver, men i denne sammenheng brukes den til å lagre informasjon om brukerene på systemet. Den inneholder fullt navn, brukernavn, hvilke grupper brukeren tilhører, passordet, hjemmekatalogen og hvilke rettigheter brukeren har osv.

linux

Linux er et operativsystem, på samme måte som Microsoft Windows og Mac OS X, opprinnelig utviklet av finnen Linus Torvalds. Forskjellen mellom Linux og de fleste andre kjente operativsystem er at Linux er utviklet under GPL lisensen. Dette betyr at når et program gjøres tilgjengelig, skal kildekoden følge med. Dette

betyr ikke at program under GPL lisensen er gratis, det betyr bare at koden følger med, og at du, om du vil, kan forandre på den eller kopiere og dele ut videre så mye du vil. Endrer du noe er du forpliktet til å gjøre forandringene dine tilgjengelig for andre igjen. Men mange program under GPL lisensen er gratis, og blir utviklet som dugnadsprosjekter. Linux er et kjempestort dugnadsprosjekt hvor alle som vil er med på utviklingen.

Selve Linux er det vi kaller en kjerne, kort fortalt et program som "snakker" med de forskjellige delene av datamaskinen. De andre programmene kommer i kontakt med maskinen, via kjernen.

linuxdistribusjon

En linuxdistribusjon er en linux kjerne satt sammen med en hel del programmer som til sammen danner et komplett operativsystem med tilhørende programvare. Linuxdistribusjoner har ofte et ytre utseende eller grensesnitt som gjør de litt spesielle på hver sin kant. Men i bunn er det stor sett den samme programvaren som ligger til grunn. En dårlig sammenligning i windowsverdenen kan være windows xp home og professional som hver sin distribusjon av windows xp-operativsystemet. Eksempler på linuxdistribusjoner er Skolelinux, Debian, RedHat og Suse.

passord-hash

En passord-hash er en kryptert nøkkel som lages av passordet. Man krypterer et passord med seg selv slik at man må vite passordet for "låse det opp". Dette er en vanlig måte å lagre passord på et system og er regnet for å være ganske sikkert i motsetning til å lagre passordet i klartekst hvor man kan lete seg fram til der det er lagret og lese det rett ut derfra.

proxy

Når en benytter en proxy-tjener betyr at web-leseren din først tar kontakt med proxy-tjeneren før den henter ned et dokument eller en fil. Dersom proxy-tjeneren ikke har fila går den ut og henter den ned samtidig som den overføres til din web-leser. Fila blir også lagret på proxy-tjeneren. Neste gang noen skal se på det samme dokumentet blir proxy-tjeneren kontaktet, den ser at dette dokumentet ligger lagret der, kontakter den web/ ftp-tjener der dokumentet stammer fra for å se om det er blitt endret, og dersom det ikke er endret overføres dokumentet fra proxy-tjeneren til web-leseren. Jo flere som benytter seg av denne proxy-tjenesten, jo oftere vil man finne dokumentene hos proxyen på lokalnettet, og man slipper

å gå ut på internett via tregere linjer. Det betyr at trafikken på linja ut blir kraftig redusert, slik at surfingen vil virke raskere for brukeren.

samba

Samba er et sett med programmer som sørger for fil og skrivererdeling under Linux og andre unix-lignende systemer. Samba benytter en protokoll/filsystem som heter CIFS som er det samme som windows benytter når den deler filer og skrivere over nettverk.

skolelinux

SkoleLinux er en linux distribusjon (se linuxdistribusjon) utviklet av Linux i Skolen. Som tidligere beskrevet har de fleste linux distribusjoner noe som gjør dem forskjellig fra andre. I Skolelinux er alt laget med tanke på å bruke systemet i skolen. De aller fleste tjenester og programmer som brukerne på en skole vanligvis trenger følger med. Det jobbes også med å gjøre oppgavene lettere for de som skal administrere slike nettverk. Skolelinux er som mange andre linux distribusjoner helt gratis og kan kopieres fritt.

webmin

Webmin er et web-basert konfigureringsverktøy for linux. Her kan en endre på en del vanlige instillinger, alt etter hvilke komponenter som er installert. Mange ser på det som enklere å bruke webmin framfor å lete seg frem etter rett konfigurasjonfil for så å redigere denne manuelt. Du finner webmin ved å koble web-leseren din mot maskinen som kjører webmin på port 10000. Skolelinux bruker webmin som administrasjonsverktøy, her endres stort sett alt, inkludert å administrere brukere.

Tillegg B

Konfigurasjon

B.1 smb-skolelinux.conf

Dette er konfigurasjonsfilen til samba. I stedet for å modifisere på standardkonfigurasjonen som følger med debian-pakkene av samba har vi valgt å opprette en egen mer eller mindre fra bunnen av.

Listing B.1: smbaddclient.pl

```
#
# Skolelinux configuration file for the samba suite
#
# Please read the smb.conf(5) manual page
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not many any basic syntactic
# errors.
#

# Modified for use with skolelinux by Svein Magne Bang 2003/04/02

#======================= Global Settings =======================

[global]

# server name
   netbios name = tjener

# server string/NT Description field
   server string = %h server (Samba %v)

# Workgroup/NT-domain name

   workgroup = skolelinux

# OpenLDAP configuration

   ldap server = tjener
   ldap port = 389
   ldap ssl = no
   ldap suffix = "ou=People,dc=skole,dc=skolelinux,dc=no"
```

```
   ldap admin dn = "cn=admin,ou=People,dc=skole,dc=skolelinux,dc=no"

# PAM setup

   obey pam restrictions = yes

# Printer settings

   load printers = yes
   printing = cups
   printcap name = cups

# Network logon

   logon home = \\tjener\%U
   logon drive = h:
   logon script = login.bat

;   invalid users = root

# Logfiles

   log file = /var/log/samba/log.%m
   max log size = 1000
   syslog = 0

# Security options

   security = user
   encrypt passwords = true

# Networking options

   socket options = TCP_NODELAY

# Browser Control Options

  local master = yes
  domain logons = yes
  domain master = yes
  preferred master = yes

# WINS Support

;   wins support = no
;   wins server = w.x.y.z

# DNS proxy for NetBIOS

   dns proxy = no

# Add NT clients

   add user script = /etc/samba/smbaddclient.pl %u


#======================= Share Definitions =======================

[homes]
   comment = Home Directories
   browseable = no
   writable = yes
```

```
   create mask = 0700
   directory mask = 0700

[netlogon]
   comment = Network Logon Service
   path = /etc/samba/netlogon
   guest ok = yes
   writable = no
   share modes = no

[printers]
   comment = All Printers
   browseable = no
   path = /tmp
   printable = yes
   public = no
   writable = no
   create mode = 0700
```

B.2 samba.schema

For at samba skal kunne legge sine (bruker)data inn i ldap trengs et schema som inneholder strukturen på dette. Følgende fil er ikke produsert av gruppen bak dette prosjektet, men kommer rett fra “Samba (v 2.2) PDC LDAP v.3 howto”[6] skrevet av Ignacio Coupeau.

Listing B.2: samba.schema

```
##
## schema file for OpenLDAP 2.0.x
## Schema for storing Samba's smbpasswd file in LDAP
## OIDs are owned by the Samba Team
##
## Prerequisite schemas - uid (cosine.schema)
##                      - displayName (inetorgperson.schema)
##
## 1.3.1.5.1.4.1.7165.2.1.x - attributetypes
## 1.3.1.5.1.4.1.7165.2.2.x - objectclasses
##

##
## Password hashes
##
attributetype ( 1.3.6.1.4.1.7165.2.1.1 NAME 'lmPassword'
        DESC 'LanManager Passwd'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{32} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.2 NAME 'ntPassword'
        DESC 'NT Passwd'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{32} SINGLE-VALUE )

##
## Account flags in string format ([UWDX      ])
##
attributetype ( 1.3.6.1.4.1.7165.2.1.4 NAME 'acctFlags'
```

```
        DESC 'Account Flags'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{16} SINGLE-VALUE )

##
## Password timestamps & policies
##
attributetype ( 1.3.6.1.4.1.7165.2.1.3 NAME 'pwdLastSet'
        DESC 'NT pwdLastSet'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.5 NAME 'logonTime'
        DESC 'NT logonTime'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.6 NAME 'logoffTime'
        DESC 'NT logoffTime'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.7 NAME 'kickoffTime'
        DESC 'NT kickoffTime'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.8 NAME 'pwdCanChange'
        DESC 'NT pwdCanChange'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.9 NAME 'pwdMustChange'
        DESC 'NT pwdMustChange'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

##
## string settings
##
attributetype ( 1.3.6.1.4.1.7165.2.1.10 NAME 'homeDrive'
        DESC 'NT homeDrive'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{4} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.11 NAME 'scriptPath'
        DESC 'NT scriptPath'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{255} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.12 NAME 'profilePath'
        DESC 'NT profilePath'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{255} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.13 NAME 'userWorkstations'
        DESC 'userWorkstations'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{255} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.17 NAME 'smbHome'
        DESC 'smbHome'
```

```
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{128} )

attributetype ( 1.3.6.1.4.1.7165.2.1.18 NAME 'domain'
        DESC 'Windows NT domain to which the user belongs'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{128} )

##
## user and group RID
##
attributetype ( 1.3.6.1.4.1.7165.2.1.14 NAME 'rid'
        DESC 'NT rid'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.15 NAME 'primaryGroupID'
        DESC 'NT Group RID'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

##
## The smbPasswordEntry objectclass has been depreciated in favor of the
## sambaAccount objectclass
##
#objectclass ( 1.3.1.5.1.4.1.7165.2.2.1 NAME 'smbPasswordEntry' SUP top AUXILIARY
#        DESC 'Samba smbpasswd entry'
#        MUST ( uid $ uidNumber )
#        MAY  ( lmPassword $ ntPassword $ pwdLastSet $ acctFlags ))

objectclass ( 1.3.1.5.1.4.1.7165.2.2.2 NAME 'sambaAccount' SUP top STRUCTURAL
        DESC 'Samba Account'
        MUST ( uid $ rid )
        MAY  ( cn $ lmPassword $ ntPassword $ pwdLastSet $ logonTime $
               logoffTime $ kickoffTime $ pwdCanChange $ pwdMustChange $ acctFlags $
               displayName $ smbHome $ homeDrive $ scriptPath $ profilePath $
               description $ userWorkstations $ primaryGroupID $ domain ))

##
## Used for Winbind experimentation
##
objectclass ( 1.3.1.5.1.4.1.7165.1.2.2.3 NAME 'uidPool' SUP top AUXILIARY
        DESC 'Pool for allocating UNIX uids'
        MUST ( uidNumber $ cn ) )

objectclass ( 1.3.1.5.1.4.1.7165.1.2.2.4 NAME 'gidPool' SUP top AUXILIARY
        DESC 'Pool for allocating UNIX gids'
        MUST ( gidNumber $ cn ) )
```

Tillegg C

Kildekode

C.1 smbaddclient.pl

Dette er et perlscript som benyttes av samba for å automatisk opprette klient-kontoer når en ny maskin legges inn i domenet. Vi har valgt å legge det inn under /etc/samba, kanskje litt snodig sted å legge en executable, men vi har gjort det for å ha litt mer kontroll på hvor de forskjellige delene befinner seg. Dette er uansett et script som ikke skal brukes av annet enn samba og regnes som en intern kommando.

Listing C.1: smbaddclient.pl

```
#!/usr/bin/env perl
#
# A script adding machines to LDAP for use by samba
#
# Authors: Odd Rune Dahle <oddrune@pvv.org>
#          Svein Magne Bang <sveinmb@stud.ntnu.no>
#
# Date:    2003-04-15


$DEBUG = 1;

$smbconf="/etc/samba/smb.conf";
$secretstdb="/var/lib/samba/secrets.tdb";
$tdbdumpbin="/usr/bin/tdbdump";
$machine_gid=70;

################## end of user configuration ##################
###############################################################


##
# First we need to read the ldap configuration from smb.conf
##

open(FILE,$smbconf) || die "Can't open $smbconf: $!";    # open smb.conf
```

```
$ldapdata[0] = "ldap server";
$ldapdata[1] = "ldap suffix";
$ldapdata[2] = "ldap admin dn";
$ldapdata[3] = "ldap rootpw";

while ($line = <FILE>) {     # parse file and extract ldap config
  if ($line =~ /ldap/i) {
   for($n=0;$n<3;$n++){
    if ($line =~ /$ldapdata[$n]/i) {
      $line =~ s/$ldapdata[$n]//i;
      $line =~ s/=//;
      $line =~ s/^\s+//;
      $line =~ s/"//g;
      $ldapdata[$n]=$line;
    }
   }
  }
}

for(n=0;$n<3;$n++){      # get rid of newlines
  chomp($ldapdata[$n]);
}

$ldapdata[1] =~ s/ou=people,//i;  # remove ou=people from basedn

close(FILE);   # close smb.conf



##
# Now it's time to dump det secrets.tdb and retrive the ldap admin pw
##


$tdbret=`$tdbdumpbin $secretstdb`;  # execute tdbdump

$foundkey = "no";  # boolean used for loop

foreach $line (split ("\n", $tdbret)){  # parse tdbdump output
  if ($foundkey eq "yes"){   # if key is found, then extract rootwd
    $line =~ s/data = //i;
    $line =~ s/"//g;
    $ldapdata[3] = $line;
    last;
  } else {                   # still not found, keep searching
    if ($line =~ s/key = //i) {
      $line =~ s/"//g;
      $line =~ s/\//,/g;
      if($line eq $ldapdata[2]){ # see if this is the right key
        $foundkey="yes";
      }
    }
  }
}

# done reading configuration


$config{'server'} = $ldapdata[0];
$config{'basedn'} = $ldapdata[1];
$config{'rootdn'} = $ldapdata[2];
$config{'rootpw'} = $ldapdata[3];
```

```
$config{'machine_group'} = $machine_gid;


##
# Now it's time to actually add the machine
##


use Net::LDAP;
use Net::LDAP::Entry;


my $ldap;


sub ldap_connect($$$){
    my ($server) = @_;

    $ldap = Net::LDAP->new($server) || &error("Server=$server");
    $ldap->bind();
}

sub ldap_add_machine($$$$$$$$$){
    ($cn, $uid, $userpw, $uidNumber, $gidNumber,
     $rootpw, $rootdn, $basedn, $homedir, $maildir, $shell) = @_;

    $ldap->bind($rootdn, password => $rootpw);
    my $entry = Net::LDAP::Entry->new();
    $entry->dn("uid=$uid,ou=People,$basedn");
    $entry->add(
                objectclass => ['posixAccount'],
                cn => $cn,
                uid => $uid,
                uidNumber => $uidNumber,
                gidNumber => $gidNumber,
                homeDirectory => $homedir,
                userPassword => "{crypt}". $userpw,
                loginShell => $shell,
                );

    return $entry->update($ldap);
}

sub ldap_add_group($$$$$){
    ($uid, $gidNumber, $rootpw, $rootdn, $basedn) = @_;

    $ldap->bind($rootdn, password => $rootpw);
    my $entry = Net::LDAP::Entry->new();
    $entry->dn("cn=$uid,ou=Group,$basedn");
    $entry->add(
                objectclass => 'posixGroup',
                cn => $uid,
                gidNumber => $gidNumber,
                );

    return $entry->update($ldap);
}

sub ldap_close(){
    $ldap->unbind();
}
```

```
sub ldap_get_max_uid_for_workstations($){
    my ($basedn) = @_;
    my $mesg = $ldap->search (base   => "ou=People,$basedn",
                              filter => "objectClass=posixAccount"
                              );
    my $minval = 7000;
    my $maxval = 9000;  # maxval must be below 10000 to avoid conflicts
                        # with the skolelinux user-range.
    foreach $entry ($mesg->all_entries()){
        my $val = $entry->get_value('uidNumber');
        $minval = $val if ($val > $minval) and ($val < $maxval);
    }
    return ++$minval;
}

sub ldap_machine_exists($$$){
    my ($uidNumber, $basedn, $filtername) = @_;

    my $mesg = $ldap->search (base   => "ou=People,$basedn",
                              filter => "$filtername=$uidNumber"
                              );
    return $mesg->count();
}


if (!$ARGV[0]) {
        print "Usage: smbaddclient <clientname\$>\n";
        exit 2;
}

my $uid = lc($ARGV[0]);  # lowercase

&ldap_connect($config{'server'}, $config{'rootdn'});

my $uidnumber=&ldap_get_max_uid_for_workstations($config{'basedn'});


# This should be a unneccessary test, but I want it here
# anyway -- prevents a lot of mess if there are bugs elsewhere.
my $count = &ldap_machine_exists($uidnumber, $config{'basedn'}, 'uidNumber');
if ($count) {
        print "[debug] uidNumber $uidnumber already exists.\n" if $DEBUG;
        exit 1;
}

$count = &ldap_machine_exists($uid, $config{'basedn'}, 'uid');
if ($count) {
        print "[debug] uid $uid already exists.\n" if $DEBUG;
        exit 1;
}

# make sure the machines-group exists - ugly way to do it, but it works :)
&ldap_add_group('machines',$config{'machine_group'}, $config{'rootpw'},
        $config{'rootdn'}, $config{'basedn'});

$result = &ldap_add_machine("Machine $uid",$uid,'*',$uidnumber,
                $config{'machine_group'},$config{'rootpw'},$config{'rootdn'},
                $config{'basedn'},'/var','/tmp','/bin/false');
if ($result->code()) {
        print "[error] $result \n";
} else {
        print "Successfully added client $uid.\n";
```

```
}

&ldap_close();
```

C.2 smb_create.pl

Dette perlscriptet brukes av webminmodulen for brukeradministrasjon i ldap for å synkronisere brukere mellom linux og windows på opprettelses-stadiet. Det kalles som en shellkommando av modulen hver gang en bruker legges til. Det kan også brukes for å endre windows-passordet for en bruker.

Listing C.2: smb_create.pl

```
#!/usr/bin/perl
# $Id: smbcreate.pl,v 1.2 2003/04/27 12:10:32 sveinmb Exp $
#
# Author: Odd Rune Dahle <oddrune@pvv.org>
# Date:   2003-03-26
#
# Modified by Svein Magne Bang <sveinmb@stud.ntnu.no> to read ldap-admin
# passwd from commanline (2003-04-11)
#
# Add a user to samba, or just change the password
# if the user already exists.
#


sub change_samba {
        $student_id = $_[0];
        $newpass = $_[1];
        $rootpass = $_[2];
        $tempfile = "/tmp/.smbhack.$$";
        umask(0177);
        open TEMP, ">$tempfile";
        print TEMP "$newpass\n$newpass";
        close TEMP;

        # try to add the user to samba
        $ret = `/usr/bin/smbpasswd -a -s $student_id < $tempfile`;

        # if it failed we change the rootpw samba uses, and try again
        # we assume the ldap admin pw has changed (or it is the first time)
        $retmsg = "";
        if($?) {
                $ret = `/usr/bin/smbpasswd -w $rootpass`;
                if($?) {
                        $retmsg .= "Failed to update ldap admin passwd in samba!";
                }
                $ret = `/usr/bin/smbpasswd -a -s $student_id < $tempfile`;
                if($?) {
                        $retmsg .= "Failed to add user $student_id to samba!\n";
                }
        }

        unlink "$tempfile";

        if($retmsg) {
```

```
                return $retmsg;
        } else {
                return "Added user $student_id to samba\n";
        }
}

if ($#ARGV < 1) {
        print "syntax: smb_create.pl <username><password><ldap-adminpw>\n";
        print "\ta user that already exist will only get the password changed\n";
        exit(1);
        }

print change_samba($ARGV[0], $ARGV[1], $ARGV[2]);
```

C.3 login.bat

Dette er et eksempel på et loginscript som blir kjørt hver gang man logger på samba fra en windowsmaskin. Det er et helt vanlig batch-script som kjent fra dos. Her kan man legge inn kommandoer og få disse utført ved pålogging, f.eks å koble koble opp nettverksstasjoner eller kopiere over konfigurasjonsfiler man ikke ønsker at brukerene skal få endre på.

Listing C.3: login.bat

```
@echo off
echo "Kobler til tjeneren"
echo "Kobler opp hjemmekatalog"
net use h: /HOME /yes
echo "Synkroniserer klokke"
net time \\tjener /set /yes
echo "Oppretter link til ~ahjemmeomrde"
start \\tjener\netlogon\shortcut.vbs
```

C.4 shortcut.vbs

Shortcut.vbs er et lite vbscript som vi har valgt å ta med for å vise hvordan man kan kalle andre scriptspråk eller programmer fra login.bat og dermed utføre mer avanserte operasjoner enn hva som er mulig med batch-fil. Dette scriptet oppretter en snarvei på skrivebordet kalt “Min hjemmekatalog” som henviser til h:

Listing C.4: shortcut.vbs

```
' VBScript.
Dim Shell, DesktopPath, URL
Set Shell = CreateObject("WScript.Shell")
DesktopPath = Shell.SpecialFolders("Desktop")
Set URL = Shell.CreateShortcut(DesktopPath & "\Min hjemmekatalog.LNK")
URL.TargetPath = "H:\"
URL.Save
```

C.5 mac-nfs-oppsett

mac-nfs-oppsett er et lite script som setter opp monteringen av hjemmeområdene ved hjelp av Netinfo Mananger på mac. Dette er ment som en hjelp ved oppsett av mac-klienter mot Skolelinux.

Listing C.5: mac-nfs-oppsett

```
#/bin/sh

# Scriptet åm økjres som root
if [ ! "`id 2>&1 | egrep "uid=0" | cut -d "(" -f1 `" = "uid=0" ]; then
        echo "Du åm ævre root forå  kunne økjre dette scriptet."
        exit
fi

# Oppsett av åhjemmeomrder
echo Dette scriptet setter opp åhjemmeomrdet i NetInfo Manager
mkdir /skole/
mkdir /skole/tjener
cd /skole/tjener
ln -s /automount/skole/tjener/home0 home0
nicl / -create /mounts/fu dir /skole/tjener/home0
nicl / -create /mounts/fu type nfs
nicl / -create /mounts/fu name tjener:/skole/tjener/home0
echo Ferdig...
```

Tillegg D

Tilknytning av windowsklienter mot Skolelinux

D.1 Introduksjon

Dette dokumentet er ment å være til hjelp når man skal koble opp maskiner som kjører Microsoft Windows i et Skolelinux-nettverk. Ingen punkter vil bli gjennomgått i stor detalj, det skal bare være en introduksjon til hvordan det *kan* gjøres. Drifting og administrering av windowsmaskiner er et omfattende tema og er langt bedre dokumentert og beskrevet andre steder.

Forfatteren av dokumentet har kun hatt tilgang på engelske versjoner av windows og referanser og navn på komponenter vil derfor bære preg av det. Dette burde i de fleste tilfeller allikevel gå greit å følge siden norske oversettelser av windows har en tendens til å være veldig direkte.

D.1.1 Revisjonshistorie

Versjon 1.0 - skrevet av Svein Magne Bang (April 2003)

D.2 Administrativt

D.2.1 Copyright

D.2.2 Disclaimer

Forfatteren tar intet ansvar for eventuelle problemer som måtte oppstå som følge av handlinger utført basert på informasjon i dette dokumentet. Det er ikke mulig å prøve ut alle ting under enhver konfigurasjon. En del av dette dokumentet vil derfor trolig inneholde feil og ikke stemme med ditt oppsett. Dersom du finner noen feil, vær vennlig å oppdatere dette dokumentet. Orginalen ligger som et latex-dokument i Skolelinux CVS.

Tillegg E

Installasjon

E.1 Konfigurasjon av tjeneren

Skolelinux er ferdig satt opp med et standardoppsett for å koble til windowsklienter. Løsningen baserer seg rundt Samba. Dersom du ønsker å endre noe av dette oppsettet anbefales en av de mange howtoene som er skrevet om Samba. Disse finnes under “documentation” på www.samba.org.

E.2 Konfigurasjon av klienten

E.2.1 Nettverksoppsett

E.2.2 Autentisering, fildeling og skriverdeling

Skolelinux har som standard et windows-domene kalt “skolelinux”. Hvordan man setter opp en windowsklient til å delta i dette domenet varierer alt etter hvilken versjon av windows man kjører. Her følger et kort sammendrag av hva som må gjøres (noe er sakset og oversatt fra “The Unofficial Samba HOWTO”)

Windows 95

Windows 95 og 95A har ikke støtte for kryptering av passord, og er derfor ikke støttet av skolelinux. Dette valget er gjort av sikkerhetsmessige årsaker.

Windows 98/ME

Disse versjonene er de enkleste å få lagt inn i domenet. Du kan gjøre dette ved å endre instillingene til “Client for Microsoft Networks“ under “Network” i kontrollpanelet. Der finnes et valg kalt “Connect to a windows NT-domain” som må

aktiveres og domenetnavnet “skolelinux” skrives inn. Etter reboot vil maskinen være klar for å logge på skolelinux.

Windows NT og 2000

For å koble til en maskin som kjører et av disse operativsystemene må du gå igjennom en veiviser. Disse systemene har en litt mer omfattende nettverksarkitektur og trenger derfor litt mer endringer i systemet. Veilederens startes ved å gå på kontrollpanelet og system. Under “Network Identification” velger du “Network ID”. I veiviseren velger du så at maskinen skal være med i et “business network” og “a network with a domain”.

Veiviseren spør etter et brukernavn, her fyller du inn “root”, passordet er administrator (root) passordet ditt på skolelinux-tjeneren og domenet settes til “skolelinux”. Det neste veiviseren spør etter er maskinnavn og domene. La maskinnavnet være det som står og fyll inn “skolelinux” nok en gang om ikke dette står fra før. Nok en gang vil maskinen be om brukernavn, passord og domene - fyll inn samme data som første gang. Etter en tenkepause, og dersom alt gikk bra, vil det dukke opp et vindu som spør om du vil legge til brukere. Her kan du velge å ikke legge til flere og trykke neste. Veiviseren vil fortelle deg at den er ferdig og at du er nødt til å restarte maskinen. Etter at maskinen er kommet opp igjen vil du i påloggingsvinduet kunne velge om du vil logge på den lokale maskinen eller skolelinux-domenet i en “drop-down”- boks.

Windows XP Home

Windows XP Home er et artig produkt som har mistet en rekke funksjoner som kan være nyttige å ha. En av de funksjonene som mangler er muligheten til å koble seg på et domene. Microsoft har bestemt seg for at en hjemmemaskin ikke har noe i et nettverk å gjøre, og har dermed utelatt denne muligheten. Windows XP Home edition støttes dermed ikke som klient i skolelinux-arkitekturen.

Windows XP Professional

Windows XP Professional er i utgangspunktet basert på Windows 2000 og dermed er prosedyren for å legge den til i skolelinux-domenet veldig lik. Den eneste forskjellen er at XP Pro har noen sikkerhetsinnstillinger som ikke går overens med oppsettet skolelinux benytter seg av. Disse innstillingene må endres før man får lagt til maskinen. Dette kan gjøres ved hjelp av “local policy”, men er enklest gjort i form av å endre to verdier i registeret. Dette kan gjøres ved hjelp av en liten .reg-fil som kjøres på maskinen. Etter at disse innstilligenen er endret kan man legge til maskinen etter samme oppskrift som for Windows 2000.

E.3 Andre tjenester

E.3.1 Konfigurering Web-proxy

For å benytte seg av proxyen må man konfigurere hver enkelt browser. I Internet Explorer gjøres dette i “Internet Options”. I Opera settes det i “Preferences”. Navnet på proxyserveren er “webcache” og port er 3128.

Siden hver browser har sitt sted å lagre denne informasjonen på finnes det ikke en enkelt måte å oppdatere dette på alle maskinene i nettverket. En metode som ofte brukes er å kopiere inn endringene via login-scriptet til samba i form av en .reg-fil, eller .ini (avhengig av browser).

E.3.2 Konfigurering av E-post

Skolelinux tilbyr flere måter å benytte mailtjenesten på. Noen er mer egnet i et nettverk på en skole enn andre. Hvilken løsning man ønsker på sitt nettverk får være opp til hver enkelt administrator å velge. Mange har nok en løsning som benyttes på sin skole allerede og ønsker å fortsette med denne. Noen er også underlagt spesielle retningslinjer for hvordan elevene skal få tilgang til epost på.

POP er kanskje den som passer dårligst inn. Dette er den protokollen som benyttes mest når man sjekker epost fra en ISP. POP innebærer at meldingen blir lastet ned og lagret lokalt hos brukeren før den kan leses i epostleseren. I et skolenettverk vil det bety at posten vil lagres flere steder i systemet, både på tjeneren og på klienten. Bruk av POP vil også kreve at epostleseren blir satt opp for hver enkelt bruker på klientene.

IMAP er langt bedre i denne sammenhengen. Med IMAP vil epostleseren være oppkoblet mot tjeneren hele tiden og bare bli et grensesnitt mot eposten. Posten vil hele tiden befinne seg på et sted, tjeneren. I noen tilfeller vil også dette innebære at epostleser må settes opp for hver enkelt bruker. Noen epostlesere kan derimot konfigureres slik at de krever en egen innlogging med brukernavn, og dermed behøves kun et felles oppsett for alle brukere.

Web-mail er den løsningen som gir minst arbeid med tanke på windowsklienter. Alt som behøves fra klientens side er en webleser. Webmail kan derimot kreve en større innsats for å bli satt opp på tjeneren, men vil være veldig lett å administrere i etterkant. Samtidig blir man spart for alle sikkerhetsproblemene epostlesere på windows har hatt de siste årene (virus og ormer som utnytter sikkerhetshull). Selv om et slikt web-grensesnitt kan være vidt forskjellig fra operativsystemet tilbyr, viser det seg at unge brukere lett forstår hvordan det skal brukes. Veldig mange av de som har privat epostadresse benytter en web-basert løsning. (F.eks hotmail eller telenor onlines epostleser).

Tillegg F

Hvordan sette opp en mac med OSX som arbeidsstasjon i et skolelinux nettverk

F.1 Forberedelser

F.1.1 Software

Vi tar i denne guiden utgangspunkt i at du har en helt ny installasjon av OS X på mac'en du vil tilknytte SkoleLinux nettverket. Det forutsettes også at du har oppdatert operativsystemet til siste versjon. Til dette bruker du den automatiske oppdateringsfunksjonen i OS X.

F.1.2 Kunnskap

Denne guiden krever at du har litt erfaring med OSX. Det forutsettes at du kjenner til en del enkle begrep og prinsipp som er vanlige i OS X. Du må også under hele oppsettet være logget inn på mac'en som root. Disse forandringene er ikke mulige som en vanlig bruker. Root konto er i utgangspunktet deaktivert. Du aktivere denne i Netinfo Manager på Sikkerhet menyen. Logg så inn på nytt med root brukeren.

F.2 Nettverksoppsett

F.2.1 Plassering av mac'er på Skolelinux nettverket

Mac'er i et Skolelinux nettverk vil oppføre seg som arbeidsstasjoner. Det er derfor naturlig å plassere dem sammen med Linux arbeidsstasjoner. For nærmere info om dette se på http://developer.skolelinux.no/arkitektur/arkitektur.html

F.2.2 DHCP

For å virke opp mot skolelinux nettverket, er det greiest om mac'en er konfigurert til å bruke DHCP for å få ip adresse og andre innstillinger fra serveren. DHCP er vanligvis aktivert som standard i OS X. Med DHCP i orden vil du ha nett-tilgangsom. Dette vil være en fordel, for da kan du enklere ta ned scriptet for oppsett av nfs på mac'en. Dette gjøres på følgende måte:

- Gå inn på Systemvalg
- Velg Netverk
- Velg nettkort. Dette er vanligvis Innebygd Ethernet
- Klikk på TCP/IP delen. Konfigurer skal her bruke Med DHCP alternativet.
- Klikk Ta i bruk

Merk:

- For å finne ut om nettverket er oppe, kan du kjøre ifconfig fra terminal. en0 skal ha flaggene UP,RUNNING.
- Om maskinen ikke har stått på nett tidligere, er dette et bra tidspunkt å kjøre oppdatering av softwaren på mac'en. Du finner Programvareoppdatering i System delen av Systemvalg. Denne guiden forutsetter at du har oppdatert OS X til siste versjon.

F.2.3 DNS

DNS blir automatisk satt av DHCP. Om man ønsker å forandre navnetjener kan en gjøre dette i oppsettet på Nettverk som finnes under Systemvalg.

F.2.4 Proxy

For å kunne benytte web proxyen som er installert på skolelinux tjeneren går man som følger:

- Gå inn på Systemvalg
- Velg Netverk
- Velg Proxy
- Merk av Webproxy
- Skriv inn ipadressen til tjener og porten til webproxyen. Ipen er vanligvis 10.0.2.2 og porten er 3128.
- Klikk Ta i bruk

F.3 Autentisering

F.3.1 LDAP oppsett i OS X

Gå inn på Katalogtilgang som ligger under Verktøy på Programmer.

- Kryss av LDAPv3.
- Klikk så Konfigurer. Her skal Sted stå på Automatisk.
- Klikk så på Ny. Her skal Konfigurasjonsnavn være tjener, Servernavn eller ipadresse skal og være tjener.
- På LDAP-Standardtyper velger du RFC 2307(Unix).
- Du får nå opp en dialogboks hvor du skal skrive Søkebaneendelse. Her skriver du følgende: dc=skole,dc=skolelinux,dc=no.
- Klikk ok. SSL skal ikke være aktivert.

Trykk på Verifisering tab'en.

- Her skal det søkes i Egendefinert bane.
- Katalognode lista skal innholde /LDAPv3/tjener. Har du ikke denne må du legge den til. Klikk i så fall Legg til og velg LDAPv3/tjener.
- Det samme gjelder for Kontakter tab'en. Har du ikke /LDAPv3/tjener på katalognode lista, må den legges til ved å trykke Legg til og velg LDAPv3.
- Lukk Katalogtilgang

F.4 Innloggingsmeny i OS X

- Gå på Kontoer innstillingene under Systemvalg
- Velg Påloggingsvalg
- Merk av Felt for navn og passord

F.5 NFS

F.5.1 Sette opp nfs

Den enkleste måten å sette opp nfs er å kjøre det ferdiglagde scriptet mac-nfs-oppsett i et skall på mac'en. Det er viktig at du er logget inn som administrator/root på mac'en før du kjører dette scriptet. Du vil ikke få kjørt det som noen annen bruker.

Code listing 5.1: nfs oppsett

Listing F.1: Oppstart av annet script

```
# scp root@tjener:/mac/mac-nfs-oppsett mac-nfs-oppsett
# chmod u+x mac-nfs-oppsett
# ./mac-nfs-oppsett
```

Merk:

- Scriptet sier i fra når det er ferdig. Du kan da avslutte terminalen.
- Scriptet ligger i sin helhet i slutten av dette dokumentet.

F.6 Tidsinstillinger i OS X

Gå på Dato og tid innstillingene under Systemvalg.

- Velg Nettverkstid
- Skriv inn adressa til tjeneren (10.0.2.2)
- Kryss av på Bruk en nettverkstidstjener

F.7 Sette opp nettverksskriver i OS X

For å sette opp en nettverksskriver i OS X går man inn på Programmer, så Verktøy og velger Utskriftssenter.

- Velg ”IP-utskrift”i rullegardinmenyen
- Skriv inn adressa til skriveren
- Velg modell og driver
- Klikk Legg til

F.8 Forandringer på tjener

F.8.1 Forandringer i oppsettet på NFS

Logg inn på tjeneren for arbeidsstasjoner med ssh som root. Hos oss heter denne maskinen tjener. Gjør som følger:

Code listing 8.1: Tjener login

Listing F.2: Oppstart av annet script

```
# ssh root@tjener
```

Oppgi passord, og du logger inn. Editer /etc/exports Dette kan for eksempel gjøres ved å bruke nano (/bin/nano)

Code listing 8.2: Editer /etc/exports

Listing F.3: Oppstart av annet script

```
# nano /etc/exports
```

Code listing 8.3: /etc/exports

Listing F.4: Oppstart av annet script

```
# /etc/exports: the access control list for filesystems which may
# be exported to NFS clients.  See exports(5).
#
# Orginal linje
# /skole/tjener/home0 10.0.2.0/255.255.254.0(rw)
#
# Forandret linje
/skole/tjener/home0 10.0.2.0/255.255.254.0(rw,insecure)
```

Her vises en ferdig modifisert /etc/exports. Det er bare å legge til et insecure parameter. Dette betyr i praksis at du lar nfs bruke porter over 1024. Såvidt vi vet utgjør dette ingen risiko. Det neste vi skal gjøre er å restarte nfs-eksporteringen.

Code listing 8.4: Restart av nfs eksporteringen

Listing F.5: Oppstart av annet script

```
# /etc/init.d/nfs-kernel-server restart
```

Logg ut fra serveren ved å skrive:
Code listing 8.5: Logge ute fra server

Listing F.6: Oppstart av annet script

```
# exit
```

Lukk nå alle programmene på mac'en og start den på nytt. Det skal nå kunne gå an å logge inn på den med brukere som er lagt inn i LDAP, som en vanlig linux arbeidsstasjon.

F.9 Tillegg

Code listing 9.1: mac-nfs-oppsett

Listing F.7: Oppstart av annet script

```
#/bin/sh
echo Dette scriptet setter opp automatisk mounting av åhjemmeomrdene
mkdir /skole ; mkdir /skole/tjener ; cd /skole/tjener
ln -s /automount/skole/tjener/home0 home0
nicl / -create /mounts/fu dir /skole/tjener/home0
nicl / -create /mounts/fu type nfs
nicl / -create /mounts/fu name tjener:/skole/tjener/home0
echo Ferdig...
```