The Kerberos 5 Experience - A three-headed beast of Kerberos, Linux and Windows

Lars*Anundskås
Jon†Langseth
Thomas‡Austad

18. november 2004

*BOFH
†gjør SuSEn
‡5lacker

Innhold

1 Introduksjon

Vi har tatt for oss Kerberos som prosjekt i System administrasjon høsten 2004. Kerberos er en nettverksautentiseringsprotokoll som tar i bruk avansert kryptografisk teknologi, og som antar at vi hele tiden befinner oss i et usikkert nett med potensielle angripere på innsiden. Vi har hørt mye positivt om denne protokollen og var derfor nysgjerrige på å få belyst denne protokollen nærmere. Siden kurset ikke tar for seg denne spesifikt i forbindelse med sikkerhet, bestemte vi oss for å ta det som prosjekt. Målet med oppgaven var å :

1. Gjøre oss kjent med Kerberos protokollen.
2. Sette opp en Master Server.
3. Sette opp en Linux klient, og autentisere denne mot master serveren.
4. Sette opp en Windows XP Klient og autentisere denne mot master serveren.
5. Forstå hva som skjer i prosessen med autentisering/autorisering.

Vårt prosjekt har vært todelt; En praktisk og en teoretisk del. Den praktiske biten besto i å sette opp maskinene for Kerberos. Den teoretiske delen har bestått av en del research på blant annet internet, og andre nett.

2 Kerberos

Det finnes i dag både kommersiell og ikke-kommersiell programvare som benytter protokollen, dette kan være alt fra enkle tjenester som ssh og ftp, til mer sentrale funksjoner som sentral autentisering av brukere på domenebasis og tilgangskontroll til clustersystemer og distribuerte filsystemer. Den primære kilden for Kerberos software for Unix plattfomen er MIT Kerberos[1], som er OSI lisensiert. Både GPL baserte alternativer, som Heimdal, og kommersielle implementasjoner, slik som Microsoft og CyberSafe[1] sine løsninger er tilgjengelige. En oversikt over noen andre aktører kan finnes på adressen `http://www.ornl.gov/~jar/commerce.htm`.

[1] http://www.cybersafe.ltd.uk/

2.1 Noen viktige begreper

I forbindelse med Kerberos er det viktig å få klarhet i noen sentrale begreper.

- Credentials - Legitimasjon, dvs Shared Secret key, ticket etc, kan sees ved å skrive kommandoen `klist`
- Fysisk sikrede hoster - Hoster bak brannmurer, routere etc ?
- Kerberized - Nett-programmer som støtter kerberos protokollen.
- Kerberisert - Samme begrep, på norsk.
- Klient - En som autoriserer seg med Kerberized programvare mot en Kerberos autentiserings-server.
- Master Server - Et system kan ha flere autentiserings-servere, men bare en er Master (default)
- Principle - En entitet i et system, dvs en host, server, klient etc.
- Private Key - Nøkkel som ingen andre enn dens eier vet om.
- Public Key - Nøkkel som identifiserer en host, distribueres til alle i nettverket og kan derfor aldri brukes til å autentisere en principle alene.
- Realm - Kan sammenlignes med et NT domene. Dette er et identifiserende navn som benyttes som gruppering for brukere, hosts og tjenester som hører logisk sammen. Knyttes ofte naturlig opp imot DNS domenenavn. Autentisering gjort i ett Realm, er kun gyldig i dette, medmindre det finnes eksplisitt 'trust' med et annet Realm.
- Shared Secret Key - En hemmelighet som er delt og kjent kun for kerberos server og klienten som ønsker autentisering.
- Ticket - En kombinasjon av nøkler, timestamp og brukerinformasjon, som tilsammen benyttes som bekreftelse på at en bruker, maskin eller tjeneste er autentisert. Tickets er begrenset i gyldighetsområde, og er kun gyldige en kort periode.
- TGT - Ticket Granting Ticket, en Ticket som bekrefer at brukeren er autentisert, som kan benyttes til å autentisere tilgang til en tjeneste eller host.

2.2 Protokollen

Kerberos opererer som en tredjeparts autentiseringsserver som tar i bruk konvensjonell kryptografi, feks Shared Secret keys. Protokollen implementerer en metode for å verifisere en host sin [2] identitet, det være seg bruker av en workstation eller en nettverksserver, i et åpent, ubeskyttet nettverk. Kerberos oppnår dette ved å ikke stole på tradisjonelle indikatorer for identifisering av entiteter, dvs IP adresse, operativsystem etc og uten å kreve fysisk sikring av hoster i nettet [3]. Kerberos antar også at pakker som reiser rundt i nettet kan bli lest og modifisert, forfalsket av angripere, uønskede gjester eller andre uvedkommede.

2.2.1 Hvordan virker det?

Kerberos er altså vår *tjener* i forbindelse med autentisering, den er ikke en *erstatning* for kryptert komminikasjon, som SSL, men et tillegg.

Man har en eller flere autentiseringsservere som kjøres på fysisk sikrede hoster. Disse serverne ivaretar en database med Principals og deres credentials, og en principal er en unik entitet(host, server, klient etc) som Kerberos kan utstede tickets til.

2.2.2 Autentiserings Prosessen

Autentiseringsprosessen kan beskrives som følger:

1. Klient sender forespørsel til sin Autentiseringsserver om credentials for en gitt server. Denne serveren er Master i den Realm klienten hører hjemme under.

2. Autentiseringsserver slår opp i sin database med credentials for de forskjellige principals, og svarer med credentials kryptert med klientens nøkkel. Denne Credential består av en ticket og en midlertidig kryperingsnøkkel, bedre kjent som en session key.

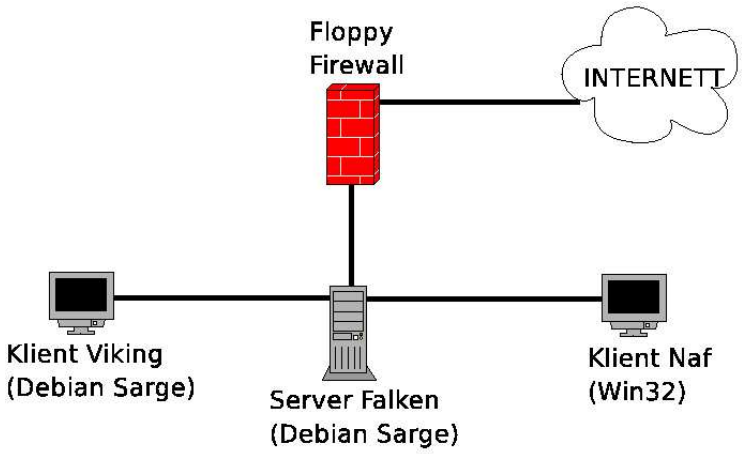
3. Klient sender nå sin ticket til den oppgitte server (som inneholder klientens identitet, kopi av session key, kryptert i servers key).

4. Session key, er nå delt mellom klient og server og brukes til å autentisere klienten, og muligens også server. Den kan også brukes til å kryptere fremtidige meldinger mellom de to.

[2] eller Principal som det heter i Kerberos

[3] firewalls, betong vegger, stål grinder, spisse pinner etc

For å kunne verifisere identiteten til en principal i en transaksjon, overfører klienten som nevnt sin ticket til server. Denne sendes i klartekst, og det er derfor nødvendig å ettersende informasjon som verifiserer at meldingen virkelig kom fra eieren av den ticket som nettopp ble overført. Det siste er kryptert i session key og inneholder et *tidsstempel*. *Tidsstemplet* beviser at meldingen var nylig generert og at den ikke er en gjengivelse utført av en eventuell angriper. Kombinasjonen session-key og *tidsstempel* gir en sikker kommunikasjonsprosess da en session key aldri overføres i klar tekst, og kun er kjent for de to kommuniserende principles.



Figur 1: Arkitektur

3 Installasjon

3.1 falken: *NIX Kerberos server

Før vi satte igang med installasjonen av Kerberos pakkene på maskinen som skulle bli vår master server, tok vi oss endel tid, og leste dokumentasjon, slik at vi skulle ha et visst overblikk. Hoved-dokumentasjonen vi leste var MIT sine egne artikler om installasjon[2], brukermanual[3], Jim Rome sin installasjonsveiledning[4] og endel ytterligere dokumenter fra MIT[5]. Deretter kom en omgang med `apt-cache search` og `Google`, for å finne frem til spesifike instruksjoner for Debian. Dette ble litt her, og litt der, vi fant igrunn ingen ren Debian Sarge installasjonsgjennomgang.

Etter å ha avdekket hvilken software vi trengte på serveren, installlerte vi pakkene:

```
krb5-admin-server
krb5-config
krb5-kdc
krb5-user
libkadm55
libkrb53
libpam-krb5
ssh-krb5
krb5-doc
```

Under installasjonen av disse fikk vi spørsmål om hostname for *Password changing server*. Vi oppga da `falken` som hostname. Neste steg ble å sette opp et nytt Kerberos Realm.

```
shell$> krb5_newrealm
```

Denne kommandoen resulterte i et spørsmål om kdc database master key. Etter å ha konsultert man-sidene, kom vi frem til at dette er en debian kommando, som gjør endel av jobbene for oppsett automagisk for oss. Vi

valgte master key '01579'. Deretter la vi til en *principality* for en bruker som vi kom til å bruke som admin.

```
shell$> kadmin.local
kadmin.local: addprinc luckystrike
passord:
passord
```

La til 'limited kerberos services' i */etc/inetd.conf* (se A.6).

Deretter erigerte vi tilgangskontroll-listen i `/etc/krb5kdc/kadm5.acl` og la til '*luckystrike*' som eneste administrator med alle rettigheter. Erigerte `/etc/krb5.conf`, og konfigurerte der vår realm i henhold til et av eksemplene vi hadde merket oss under dukumentasjonslesningen. Opprettet også `/etc/krb5kdc/kadm.acl` og `/etc/krb5kdc/kdc.conf` som brukes av kdc[4]. Benyttet så kadmin.local til å legge inn nye *keytables* før vi restartet krb5-serverene:

```
shell$> kadmin.local
kadmin.local: ktadd -k /etc/krb5kdc/kadm5.keytab kadmin/admin kadmin/changepw
kadmin.local: quit

shell$> /etc/init.d/krb5-kdc restart
shell$> /etc/init.d/krb5-admin-server restart
shell$> /etc/init.d/inetd restart
```

Vi la så inn principalities for alle brukere og maskiner som skulle autentiseres via kerberos i første omgang, med andre ord falken og viking, våre to Linux maskiner.

```
shell$> kadmin.local
kadmin.local: addprinc viking
kadmin.local: addprinc falken
kadmin.local: addprinc host/viking.redningstjenesten.no
kadmin.local: addprinc host/falken.redningstjenesten.no
kadmin.local: ktadd host/viking.redningstjenesten.no
kadmin.local: ktadd host/falken.redningstjenesten.no
```

Etter å ha konfigurert Kerberos måtte vi fortelle PAM at den skulle bruke Kerberos som autentiseringsmekaniske. For å få dette til måtte følgende linje:

```
auth     sufficient pam_krb5.so
```

legges til i `/etc/pam.d/common-auth`(se A.1). Dette medførte at vi plustelig hadde lokal autorisasjon av brukere på serveren via kerberos. PAM[5] sjekker først med krb5kdc om maskinen kan autentiseres, deretter om brukeren kan autentiseres, utsteder eventuell Ticket, og tillater til slutt login, dersom brukeren i tillegg har en gyldig brukerkonto på maskinen.

Dette gjennomføres senere også på klienten, slik at lokal pålogging på Linuxmaskinene våre autentiseres mot den sentrale Kerberos Master Serveren. Mer om dette i neste kapittel.

[4] Key Distribution Center

[5] Pluggbare Autentiserings Moduler

3.2 viking: *NIX Kerberos klient

For å sette opp viking som Kerberos Linux Klient, måtte vi apt-get'e noen pakker først:

```
shell$> apt-get install krb5-clients krb5-doc libpam-krb5 ssh-krb5
```

- krb5-clients inneholder *Kerberized versions* av telnet, ftp og rsh.
- krb5doc Dokumentasjon for Kerberos 5
- libpam-krb5 er kerberized versjon av dette API'et for autentiseringsrelaterte tjenester.
- ssh-krb5 Kerberized SecureSHell.

Vi fikk melding om at denne openSSH versjonen inneholdt nye 'privelege separation options' noe som reduserer faren som sikkerhetshullene i sshd representerer. Denne kunne vi deaktivere i `/etc/ssh/sshd_conf` med 'UsePrivilegeSeparation no', noe vi også måtte gjøre.

Redigerte så `/etc/krb5.conf` slik at klienten kunne finne riktig Kerberos server. Disse tre er det vesentligste fra denne filen:

```
realm = redningstjenesten.no
servers = falken
adminservers = falken
```

For å få utstedt ticket fra Falken måtte vi utføre følgende kommando;

```
shell$> kinit
     passord: this_is_cloacked_on_purpose
```

Etter å ha fått tildelt ticket brukte vi den kerberiserte SSH klienten som nylig ble installerte til å logge oss på falken[6]:

```
shell$> ssh viking@falken
--(viking@falken)-(483/pts/1)-(08:22/18-Nov-04)--
--($:~/)-- ls -lars ; ls -thomas ; ls -ion
```

Dette gikk fint. Vi kunne nå avslutte SSH sesjonen og skrive kommandoen `klist` for å få en liste over tildelte tickets samt hvor lenge de var gyldige:

```
shell$> klist
     liste over tickets, bl.a.
     tidspunkt   tidspunkt   host/falken.redningstjenseten.no@REDNINGSTJENESTEN.NO
```

Hvis vi nå skrev `ssh falken` ble vi ikke spurt etter passord. Dette ettersom vi hadde fått tildelt en ticket med gyldighet i 24000 sekunder.

[6]Legg her merke til de personlige versjonene av LS kommandoen.

3.3 Uttesting av *NIX Kerberos installasjon

Siden Kerberos fungerte på *NIX maskinene hadde vi noen spørsmål i forbindelse med installasjonen vi hadde gjort.

Validering som root: Gikk det ann å validere seg som `root` og dermed få samme tilgang som før? For å teste dette prøvde vi både `su` og `ksu` kommandoene, og fant at *root* heldigvis ikke hadde magiske tilgang via Kerberos. Det skulle root heller ikke ha i følge vår installasjon siden denne brukeren ikke var oppført i Kerberos sin database.

Legge til nye brukere: Hva skjer når man legger til en bruker på en av *NIX maskinene, vil Kerberos automagisk bli oppdatert? For å teste dette scenario så brukte vi både `useradd` og `adduser` kommandoene for å opprette brukere. Deretter prøvde vi en login. Vi fant at ingen av de to kommandoene var '*kerberisert*', og at for å forenkle innlegging av brukere ble vi nødt til å skrive scripts som gjorde oppdatering både i *NIX og Kerberos brukerdatabasene.

Scripting av brukeradministrering: Er det mulig å skripte operasjoner som gjør det enklere for administrator å legge til og avslutte brukere? Vi prøvde å automatisere disse operasjonene, og fant at `kadmin.local` kunne brukes i denne sammenheng siden den hadde et kommandolinjeinterface som var kraftig nok. `kadmin.local` ble så satt i en sammenheng sammen med `adduser` og `deluser` for å opprette og slette brukere. Vi opprettet så to script i `/usr/local/sbin` og kalte dem beskrivende nok `spawn_user` og `slay_user`. For å få et overblikk i hva som ble gjort i skriptene henviser vi til D.1 og D.2.

3.4 naf: Win32 Kerberos klient

Lastet ned siste versjon av MIT Kerberos for MS Wintendo[1]. Vi fulgte alle default valg under installasjonen bortsett fra at vi valgte å legge kerberos i `C:\Kerberos`, og ikke i `C:\Program Files\Kerberos` slik installasjonsprogrammet ville. Da installasjonen var ferdig måte vi konfigurere `C:\Windows\krb5.ini` (se appendiks C.1). Dette for å få LEASH (ticket manager) til å validere brukere mot `falken.redningstjenesten.no`.

Det viste seg at LEASH var enkel å konfigurere siden vi allerede hadde satt oss inn i konfigurasjonsfilene som brukes under *NIX. Etter at MIT installasjonen var ferdig konfigurert startet vi LEASH, og prøvde en login med principal satt til `falken`. Vi fikk dermed ticket fra falken.redningstjenesten.no.

Vi reiste oss så opp for å få XP til å bruke Kerberos validering ved login. Dette viste seg å by på en del hodebry. Vi prøvde først å følge kerberosdokumentasjonen til Microsoft[6] For å følge denne måtte vi installere Windows

ResourceKit (fra Windows XP CDen). Fulgte deretter instruksjonene som ble gitt i MS dokumentet, og kjørte følgene kommandoer i kommandoskall på windows burken:

```
ksetup /setdomain REDNINGSTJENESTEN.NO
ksetup /addkdc REDNINGSTJENESTEN.NO falken.redningstenesten.no
ksetup /setmachpassword 01763
ksetup /MapUser AllUsers *
shutdown -r
```

La også til relevante data på Kerberos Serveren(falken).

```
shell> kadmin.local
kadmin.local: addprinc bruker
kadmin.local: addprinc host/naf.redningstjenesten.no
kadmin.local: ktadd host/naf.redningstjenesten.no
```

Dette viste seg å være lite fruktbart, og vi kom ikke vesentlig nærmere en løsning. Problemet var at vi ikke fikk windows logon service til å bruke `kdc`[7] for å validere påloggingsforsøk.

Vi ga opp å følge Microsoft sitt dokument, og brukte en del tid på å finne dokumentasjon [7] [8] [6] [9] om avvik fra Kerberos-standarden i Microsoft sin implementasjon. Dette viste seg å være en bedre rute å ta, og vi fikk flere pekepinner på små men kritiske detaljer som Microsoft dokumentet ikke sa noe om. Det viste seg at Microsoft sin implementasjon av Kerberos ikke var helt standard, og den implementerte bl.a. kun to typer hash-funksjoner[8], og krevde at `pre-authorization` var satt.

Dette medførte at vi raderte all informasjon i Windows sitt Kerberos oppsett. Dette ble gjort med `ksetup.exe` fra ResourceKit. Desverre så '*glemte*' den å ta vekk mapping mellom lokale brukere og kerberos principals. Vi måtte derfor bruke `regedit` til denne jobben. Etter at alle spor etter det tidligere Kerberos oppsettet var fjernet tok vi en reboot av maskinen[9]. For så å oppdatere `c:\Windows\krb5.ini`. I denne filen la vi til `admin_keytab`, `acl_file` i `libdefaults` seksjonen. La også til `master_key_type`, `supported_enctypes` og `default_principal_flags` i `realms` seksjonen.

Deretter ble Kerberos satt opp på nytt ved hjelp av `ksetup`:

```
ksetup /SetRealm REDNINGSTJENESTEN.NO
ksetup /AddKdc REDNINGSTJENESTEN.NO falken.redningstjenesten.no
ksetup /AddKpasswd REDNINGSTJENESTEN.NO falken.redningstjenesten.no
ksetup /MapUser * *
ksetup /SetComputerPassword secret
```

[7] Key Distribution Center

[8] MIT Kerberos og Microsoft Kerberos har kun des-cbc-crc, des-cbc-md5 som felles hash funksjoner.

[9] N'te reboot siden vi startet Windows eXPerimentet..

For å ikke få problemer med tidsangivelsen ble også SNTP service startet:

```
net time /setsntp:ramstind.gtf.ol.no
net start w32time
```

og denne ble satt til å starte når maskinen bootet. Vi tok deretter en reboot[10] for å sjekke at alt var i orden(så langt i alle fall). Vi fikk dermed muligheten til å bruke Kerberosvalidering ved login, men dette fungerte ikke, og windows xp valgte å bruke lokal autorisasjon i stedet.

La så til `default_principal_flags = +preauth, +tgt-based` i `C:\krb5.ini`, men etter et login foorsøk fant vi at dette ikke hjalp oss videre.

Fikk så den 'glimrende' ideen at '`man`' sikkert hadde svaret på vårt problem, og i `krb5.conf` sin man-page fant vi at de samme flaggene kunne brukes på serversiden. Vi la dermed til den ovenfornevnte informasjon på `falken.redningstjenesten.no`. Restartet så kerberos på serversiden, og logget ut av windows xp maskinen. Da vi prøvde login ble det registrert at brukeren hørte til et Kerberos domene. Kerberos ticket ble sendt slik at brukernavn ble akkreditert, og vi kunne endelig fullføre en kerberized login.

Det siste vi gjorde var å forsøke å en login, med en kerberos principal kalt *luckystrike*, som ikke hadde en lokal konto på windows boksen. Etter å ha angitt brukernavn og passord, så opprettet windows en default profil og fullførte logon prosessen for *luckystrike*. Grunnen til at dette ble sjekket var at under *NIX, så må man ha satt opp hjemmekataloger etc. på forhånd, og ikke tatt det underveis slik som Windows XP. Vi ser at Windows XP sin måte å takle denne situasjonen på er hendig. Dette siden man slipper å tenke på synkronisering mellom principals på Kerberos server og brukerprofiler lokalt.

4 Praktisk konklusjon

Med dette er den oppgaven vi satte oss fore å fullført. Vi har lært oss hvordan Kerberos 5 fungerer, vi har klart å sette opp Kerberos 5 autentisering av brukere på Unix/Linux plattform, vi har fått krb5 autentisering av login via PAM på Linux til å fungere, og vi har *single-sign-on* for kerberiserte *NIX tjenester som SSH. Vi har klart å få Windows XP til å utføre brukerautentisering mot en *NIX basert Kerberos server.

[10] N+1'te ... urk!

5 Hvorfor Kerberos?

Etter å ha gjennomført dette prosjektet blir det logisk å reise noen spørsmål om hvorvidt Kerberos er *vegen å gå* .

Hva har egentlig Kerberos å tilby av sikkerhet og skalerbarhet, og hvilke konsekvenser vil disse valg medføre?

- Kerberos forhandler med autentiserte overføringer som kan være krypterte mellom to endepunkter som kan være situert i to forskjellige nett. Dette gjør så Kerberos har ett ekstra lag med sikkerhet som ikke er avhengig av hvilken side av firewall/pakkeswitch/router klientene er. Dette er viktig siden de fleste angrep på nettverk kommer fra insiden.

- Brukere er notorisk dårlige til å huske vanskelige passord, og i denne sammenhengen kan Kerberos hjelpe siden den bruker ett ticket system[10]. På denne måten kan brukere slippe å forholde seg til selve passordet så ofte, og i stedet la Kerberos ticket gjøre valideringen.

- Kerberos gjør det også mulig å automatisere autentisering, slik at passord kan utelukkes helt selv om man beholder autentisering og mulighet for kryptering. Dette gir for eksempel muligheten til å kryptere kommunikasjon mot distribuerte filsystemer, og det gir mulighet for økt produktivitet for folk som benytter interaktive passord som krever autentisering mye i sin arbeishvedag.

- Kerberos 4 er brukt på diverse nettverk rundt omkring i verden og versjon 5 av Kerberos er bakoverkompatibel[11]. Noe som gjør overgangen til versjon 5 enklere siden dette kan gjøres gradvis.

- Hvis man er administrator for ett Windows nettverk hvor store deler av services kjøres på *NIX, vil Kerberos hjelpe administrering av brukere siden brukerne automagisk får opprettet en lokal profil ved login. Kobler man dette sammen med SAMBA som filserver(se 6), og automatisk mounting av brukerens hjemmeområde, burde dette være en løsning som skalerer bra siden man slipper å administrere Windows maskinene etter at Kerberos er satt opp.

- Kerberos server er ikke vanskelig å sette opp, noe denne rapporten burde være ett vitnesprov på. Det handler mest om å lese dokumentasjonen[11] *før* man begynner å installere programvaren. Det bør også gjøres nøye vurderinger i forhold til hvilke applikasjoner/services som må kerberiseres.

[11] Dette er også kjent under begrepet RTFM, og ikke helt uten grunn.

- Kerberos er en svært sterk løsning, ikke bare for initiell autorisasjon, men kontinuerlig autentisering, gjennom ticket-systemet. I motsetning til de fleste andre systemer, hvor en bruker, maskin eller tjeneste bekreftes kun en gang i starten av kommunikasjon, passer Kerberos på at riktig identitet bekreftes jenlig, og legger opp til at krypteringsnøkler også skal erstattes jevnlig.
- MIT Kerberos server på BSD eller Linux, gjerne i kombinasjon med LDAP, er en svært kostnadseffektiv løsning for autorisasjon, sammenlignet med Microsoft Windows 2003 Server med Active Directory, ellet Sun NIS med Sun Solaris Kerberos.
- Kombinasjonen av *BSD/Linux, Kerberos 5, LDAP og Samba, er fullt mulig (se 6) å sette opp som en komplett erstatning for Micosoft Windows NT Domain Controller og/eller Microsoft Active Directory, ikke bare for blandede nett, men også som en kraftig løsning i rene Microsoft Windows 2000/XP baserte nettverk.

Siden Kerberos vanskeligjør mange typer angrep og sikkerhetsrisikoer så må det være en grunn til at så få bruker denne type løsning? Det kan være mange grunner til dette, og vi har her prøvd å sammenfatte en del av disse:

- Det finnes ingen rask *script-o-matic* løsning(er) for migrasjon av brukerdatabaser fra standard *NIX til Kerberos. Det lar seg gjøre å migrere, men standard *out-of-the-box* konverteringsskript er ikke tilgjengelig.
- Kerberos er kun delvis kompatibel med PAM systemet som brukes av de fleste *NIX baserte OSer. Dette er noe en del folk på nett hevder, men vi må si at det var relativt enkelt å få Kerberos og PAM til å fungere sammen.
- For at en applikasjon skal kunne gjøre bruk av Kerberos så må kildekoden forandres. Dette for å kunne bruke de rette funksjonskallene i Kerberos biblioteket. For en del applikasjoner vil dette medføre relativt stor arbeidsmenge før den er kerberisert. På den annen side vil det være helt umulig å få *closed-source* programmer til å fungere sammen med Kerberos hvis de ikke er kerberisert på forhånd.
- Ved å bruke Kerberos som autoriseringsmekanisme på ett nettverk, så må man tenkte godt gjennom dette siden Kerberos er en *alt-eller-ingenting* løsning. F.eks. hvis noen *services* fortsatt sender *plain-text* passord, så vil det fortsatt være store muligheter for at nettverket kan bli utsatt for eksterne farer, og som sådan vil det ikke bli vesentlig sikrere. For å sikre seg maksimalt må man kerberisere alle applikasjoner/services som sender *plain-text* passord eller slutte å bruke disse applikasjonene på nettverket.

6 SAMBA

I samtale med Erik Hjelmås på onsdag(17 Nov 2004) fikk vi servert ideen om å kombinere Kerberos og SAMBA. Selv om det var *dårlig* med tid igjen, bestemte vi oss for å gjøre ett tappert forsøk.

Denne tilleggsoppgaven blir dermed; sette opp SAMBA slik at man kan bruke Kerberos tickets for å validere brukere, og på denne måten kan de mounte shares og/eller hjemmekataloger på f.eks. en *NIX maskin.

6.1 Installasjon

Før vi installerte SAMBA tok vi et søk `apt-cache` for å finne de pakker vi trengte. Disse ble så installert vha `apt-get`.

Pakkene som ble installert var:

- samba
- smbfs
- samba-doc
- smbclient

I løpet av installasjonsprosessen fikk vi spørsmål om hvilket *domain* vi skulle bruke. Her svarte vi `redningstjenesten.no`.

Vi redigerte deretter `/etc/samba/smb.conf`, og etter å ha konsolidert dokumentasjonen så redigerte vi litt i oppsettet til *Authentication* delen:

```
security = ADS
realm = REDNINGSTJENESTEN.NO
password server = falken.redningstjenesten.no
```

restartet deretter SAMBA service, og testet om vi kunne browse samba serveren vha `smbclient` fra localhost. Dette fungerte ikke, og vi satte deretter opp SAMBA med default bruker autorisering, samt `obey pam restrictions = yes` for å sjekke om vi i det hele tatt kunne koble oss opp mot serveren. Noe som heller ikke fungerte smertefritt.

Etter å ha skumlest nylig oppdaterte filer i `/var/log` fant vi at `cifs/falken` tjenesten ikke lå i Kerberos databasen. Vi la derfor til denne.

```
falken$ kadmin.local
kadmin.local> addprinc cifs/falken
              satte passord 01579
kadmin.local> ktadd cifs/falken
```

Vi redigerte også `/etc/samba/smb.conf` for å prøve å tvinge SAMBA til å lytte på nettverksinterface i stedet for å se på IP adresser.

```
hosts deny = none
hosts allow = ALL
interfaces = eth2 eth1 eth0 lo
bind interfaces only = yes
```

Ved testing av dette oppsettet fant vi at dette heller ikke fungerte!

I ren desperasjon tok vi så å redigerte `/etc/samba/smb.conf` enda en gang. Denne gangen tok vi vekk endringene som er nevnt ovenfor, og satte denne opsjonen i stedet:

```
encrypt passwords = no
```

Da vi hadde restartet SAMBA service fant vi at Kerberos faktisk fikk en ticket forespørsel fra `cifs/falken`. Dette gjorde oss optimistiske, og vi fant at det var tid for å lese mer dokumentasjon i håp om å bringe Samba og Kerberos sammen :)

6.2 Samba vs Kerberos

Den eneste måten vi fant at SAMBA kunne bruke Kerberos for autentisering på er ved å sette opp SAMBA som en ADS[12]. Dette gjøres ved å legge til:

```
realm = REDNINGSTJENESTEN.NO
security = ADS
password server = falken.redningstjenesten.no
encrypt passwords = yes
```

i `/etc/samba/smb.conf`. Deretter må `/etc/pam.d/samba` redigeres litt siden default oppsett i denne var under pari. Vi erstattet alt i denne filen med auth,account,password moduler som fungerte bedre en default Debian oppsett. Noe som ikke er så rart når man tenker på at Debian ikke bruker Kerberos by-default.

Den nye `/etc/pam.d/samba` filens innhold ble:

```
# THOMAS: test config for kdc-pdc
#
auth       requisite   pam_nologin.so
auth       requisite   pam_krb5.so
auth       optional    pam_smbpass.so migrate
account    required    pam_krb5.so
password   requisite   pam_cracklib.so retry=3
password   optional    pam_smbpass.so nullok use_authtok try_first_pass
password   required    pam_krb5.so use_authtok try_first_pass
session    required    pam_krb5.so
```

Da dette var gjort restarte vi SAMBA service, og rebootet Windows boksen(just in case:). Da samba kom opp igjen lagde vi en maskinkonto[13] ved å kjøre følgende kommando:

```
falken$ net ads join -U Administrator%password
```

Kjørte så en `net` kommando på windows xp boksen for å sjekke at oppsettet fungerte.

[12] Active Domain Server

[13] Kalles vel strengt tatt en Computer Account i SAMBA lingo.

```
C:> net use * \\falken\bruker
```

Mot formodning så fungerte dette, og vi foretok så en mounting av netverks-share(d.v.s. vi mountet bruker sin hjemmekatalog på serveren) i filbehandleren, logget så ut, og logget oss på igjen. Da vi åpnet filbehandleren på ny hadde vi fortsatt den tilgang vi skulle ha. Vi foretok deretter noen flere logon/logoff for å teste om alt fungerte smertefritt. Noe det gjorde i ca 20 minutter, men da ville ikke Windows XP boksen mer, og bestemte at vi skulle få se en BSOD[14]. Da vi fikk startet opp maskinen igjen hadde `registry` fått en del feil, og den velkjente runddansen med '*fixing registry*' og rebooting var et faktum. Vi hadde ikke tid til å finne ut hvor feilen som forårsaket dette var grunnet tidspress(må levere denne rapporten i dag :)!

[14]Blue Screen Of Death

A falken

A.1 /etc/pam.d/common-auth

```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#

#auth required pam_unix.so nullok_secure
auth       sufficient   pam_krb5.so
auth       sufficient   pam_unix.so nullok try_first_pass
auth       required     pam_deny.so
```

A.2 /etc/krb5.conf

```
[libdefaults]
 default_realm = REDNINGSTJENESTEN.NO
 ticket_lifetime = 24000
 dns_lookup_realm = false
 dns_lookup_kdc = false
 krb4_config = /etc/krb.conf
 krb4_realms = /etc/krb.realms
 kdc_timesync = 1
 ccache_type = 4
 forwardable = true
 proxiable = true

[realms]
 REDNINGSTJENESTEN.NO = {
  kdc = falken:88
  admin_server = falken:464
  default_principal_flags = +preauth, +tgt-based
 }

[domain_realm]
 .redningstjenesten.no = REDNINGSTJENESTEN.NO
 redningstjenesten.no = REDNINGSTJENESTEN.NO

[kdc]
 profile = /etc/krb5kdc/kdc.conf

[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 24000
   renew_lifetime = 24000
   forwardable = true
   krb4_convert = false
 }

[login]
 krb4_convert = true
 krb4_get_tickets = true


[logging]
 default = FILE:/var/log/krb5lib.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind
```

A.3 /etc/krb5kdc/kdc.conf

```
[kdcdefaults]
 kdc_ports = 750,88

[realms]
 REDNINGSTJENESTEN.NO = {
  database_name = /var/lib/krb5kdc/principal
  admin_keytab = FILE:/etc/krb5kdc/kadm5.keytab
  acl_file = /etc/krb5kdc/kadm5.acl
  key_stash_file = /etc/krb5kdc/stash
  kdc_ports = 750,88
  max_life = 10h 0m 0s
  max_renewable_life = 7d 0h 0m 0s
  master_key_type = des3-hmac-sha1
  supported_enctypes = des3-hmac-sha1:normal des-cbc-crc:normal
                     => des:normal des:v4 des:norealm des:onlyrealm
                     => des:afs3
  default_principal_flags = +preauth
 }
```

A.4 /etc/krb5kdc/kadm5.acl

```
# This file Is the access control list for krb5 administration.
# When this file is edited run /etc/init.d/krb5-admin-server restart to activate
# One common way to set up Kerberos administration is to allow any principal
# ending in /admin  is given full administrative rights.
# To enable this, uncomment the following line:
# */admin *

# Kerberos_principal permissions [target_principal] [restrictions]
root@REDNINGSTJENESTEN.NO *

# eof
```

A.5 /etc/ssh/sshd_conf

```
# Package generated configuration file
# See the sshd(8) manpage for details
Port 22
Protocol 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
UsePrivilegeSeparation yes
KeyRegenerationInterval 3600
ServerKeyBits 768
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 600
PermitRootLogin yes
StrictModes yes
RSAAuthentication yes
PubkeyAuthentication yes
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PermitEmptyPasswords no
PasswordAuthentication no
```

```
# To change Kerberos options
KerberosAuthentication yes
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver
KerberosTgtPassing no

X11Forwarding no
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
KeepAlive yes

Subsystem sftp /usr/lib/sftp-server

UsePAM yes

# added by thomas
UsePrivilegeSeparation yes
```

A.6 /etc/inetd.conf

```
# /etc/inetd.conf:  see inetd(8) for further informations.
#

#:INTERNAL: Internal services
discard stream tcp nowait root internal
discard dgram udp wait root internal
daytime stream tcp nowait root internal
time stream tcp nowait root internal
time dgram udp wait root internal

# Limited Kerberos services
#
krb5_prop stream tcp nowait root /usr/local/sbin/kpropd  kpropd
eklogin   stream tcp nowait root /usr/local/sbin/klogind klogind -5 -c -e
```

B viking

B.1 /etc/pam.d/common-auth

```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
#auth required pam_unix.so nullok_secure
auth sufficient pam_krb5.so

auth sufficient pam_unix.so nullok try_first_pass
auth required pam_deny.so
```

B.2 /etc/krb5.conf

```
[libdefaults]
 default_realm = REDNINGSTJENESTEN.NO
 krb4_config = /etc/krb.conf
 krb4_realms = /etc/krb.realms
 kdc_timesync = 1
 ccache_type = 4
 forwardable = true
 proxiable = true

[realms]
 REDNINGSTJENESTEN.NO = {
  kdc = falken
  admin_server = falken
 }


[domain_realm]
 .mit.edu = ATHENA.MIT.EDU
 mit.edu = ATHENA.MIT.EDU
 .media.mit.edu = MEDIA-LAB.MIT.EDU
 media.mit.edu = MEDIA-LAB.MIT.EDU
 .whoi.edu = ATHENA.MIT.EDU
 whoi.edu = ATHENA.MIT.EDU
 .stanford.edu = stanford.edu

[login]
 krb4_convert = true
 krb4_get_tickets = true
```

B.3 /etc/ssh/sshd_conf

```
# Package generated configuration file
# See the sshd(8) manpage for details

Port 22
Protocol 2,1
HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
```

```
UsePrivilegeSeparation no
KeyRegenerationInterval 3600
ServerKeyBits 768
SyslogFacility AUTH
LogLevel INFO
LoginGraceTime 600
PermitRootLogin yes
StrictModes yes
RSAAuthentication yes
PubkeyAuthentication yes
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PermitEmptyPasswords no
PasswordAuthentication no

# To change Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver
#KerberosTgtPassing yes

X11Forwarding no
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
KeepAlive yes
Subsystem sftp /usr/lib/sftp-server

UsePAM yes
```

C naf

C.1 c:\windows\krb5.ini

```
[domain_realm]
 .redningstjenesten.no = REDNINGSTJENESTEN.NO
 redningstjenesten.no = REDNINGSTJENESTEN.NO

[libdefaults]
 default_realm = REDNINGSTJENESTEN.NO
 dns_lookup_kdc = true
 admin_keytab = /etc/krb5kdc/kadm5.keytab
 acl_file = /etc/krb5kdc/kadm5.acl


[realms]
 REDNINGSTJENESTEN.NO = {
  kdc = falken:88
  admin_server = falken:464
  default_domain = redningstjenesten.no
  master_key_type = des3-hmac-sha1
  supported_enctypes = arcfour-hmac:normal arcfour-hmac:norealm
                     => arcfour-hmac:onlyrealm des3-hmac-sha1:normal
                     => des3-hmac-sha1:norealm des3-hmac-sha1:onlyrealm
  default_principal_flags = +preauth, +tgt-based
 }

[login]
 krb4_convert = true
 krb4_get_tickets = true
```

D Brukeradministrering

D.1 spawn_user

Her kommer scriptet for å legge til brukere..

D.2 slay_user

Her kommer scriptet for å slette brukere..

Disse to ble desverre ikke kopiert med i rapporten innenfor tidsfristen..

Referanser

[1] Massachusetts Institute of Technology. Mit kerberos distribution page. http://web.mit.edu/kerberos/www/dist/index.html#krb5-1.3.5-bin.

[2] Massachusetts Institute of Technology. Kerberos v5 installation guide. http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3.5/doc/krb5-install.html.

[3] Massachusetts Institute of Technology. Kerberos v5 user guide. http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3.5/doc/krb5-user.html.

[4] Jim Rome. Kerberos installation help. http://www.ornl.gov/~jar/HowToKerb.html.

[5] Massachusetts Institute of Technology. Kerberos 5 release 1.3.5 (documentation). http://web.mit.edu/kerberos/www/krb5-1.3/#documentation.

[6] Microsoft. Step-by-step guide to kerberos 5 (krb5.10) interoperabillity. http://www.microsoft.com/windows2000/techinfo/planning/security/kerbsteps.asp.

[7] "darkness". Kerberos v, afs, and windows 2000. http://www.codefu.org/weblogs/darkness/archives/000092.html.

[8] Turbo Fredriksson (kerberos mail list). Win logon to a mit kerberos v kdc? http://mailman.mit.edu/pipermail/kerberos/2002-October/001857.html.

[9] (Unknown). Getting a windows xp workstation to authenticate to an mit kerberos server. http://cfm.gs.washington.edu/~adioso/HOWTO/Microsoft/WinXP-MIT-Kerberos.txt.

[10] B. Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networksw. *IEEE Communications*, 32(9):33-38, September 1994.

[11] John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so. The evolution of the kerberos authentication system. *Distributed Open Systems*, 1994. pages 78-94. IEEE Computer Society Press.

[12] Jim Rome. Commercial kerberos vendors. http://www.ornl.gov/~jar/commerce.htm.

[13] news.answers comp.protocols.kerberos, comp.answers. Faq's about kerberos. http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html.

[14] B. Clifford Neuman. Proxy-based authorization and accounting for distributed systems. *Proceedings of the 13th International Conference on Distributed Computing Systems*, May 1993. pages 283-291.

[15] Marlena E. Erdos and Joseph N. Pato. Extending the osf dce authorization system to support practical delegation. February 1993. Proceedings of the 1993 PSRG Workshop on Network and Distributed System Security.