Medlemsblad for

Dansk UNIX-system Bruger Gruppe

DKUUG-Nyt

Nummer 49, 1. april 1992

Indhold

Redaktionelt

DKUUG-Nyts redaktion består af Søren O. Jensen (ansvarshavende) og Christian Damsgaard Jensen.

Vi er naturligvis altid interesserede i indlæg fra folk. Det behøver ikke være lange artikler, men kan også være annonceringer, opfølgninger af tidligere artikler, eller andet. Hvis I blot har ønsker eller gode ideer til artikler, er I også meget velkomne til at kontakte os. Bidrag til bladet bør indleveres på maskinlæsbar form.

Indlæg, foreslag, ønsker, etc. til nr. 50 kan sendes med elektronisk post til redaktionen på adressen:

`dkuugnyt@dkuug.dk`

eller, hvis man foretrækker almindelig sneglepost, til:

Søren O. Jensen
Datalogisk Institut
Universitetsparken 1-3
2100 København Ø

Deadline for nr. 50 er d. 15. april
DKUUG kan kontaktes på følgende måder:

DKUUG, sekretariatet (Inge og Mogens Buhelt)
Kabbelejevej 27B
2700 Brønshøj
Telefon: 31 60 66 80 (mandag, tirsdag og torsdag, kl. 13-14)
Telefax: 31 60 66 49 (NB: NYT NUMMER!)
Giro: 1 37 86 00
Email: `sek@dkuug.dk`

DKUUGs netpassere (Jørgen Jensen og Kim Chr. Madsen)
Telefon: 31 39 73 22
Email: `netpasser@dkuug.dk`

DKUUGs formand (Keld Simonsen)
Telefon: 33 13 00 23
Email: `keld@dkuug.dk`

Tema: Sikkerhed og UNIX

Af *Christian D. Jensen*
DKUUG–Nyt

I dette nummer af DKUUG–Nyt sætter vi fokus på sikkerheden i UNIX. Mange mener at sikkerhed i UNIX er en selvmodsigelse. Denne fordom stammer helt tilbage fra UNIX's barndomsdage, hvor det i første række var et operativsystem til forskning og undervisning på universiteterne. Det er klart at disse universitetssystemer ikke havde en særlig god sikkerhed, da det ofte ikke er ønskeligt, afvejet mod den fleksibilitet man mister, når sikkerheden forbedres. Idag bruges UNIX i stor udstrækning udenfor universitetsmiljøerne, hvorfor behovet for sikkerhed naturligt er steget.

Vi vil i dette nummer fosøge at aflive den myte at UNIX ikke kan gøres sikkert, set i forhold til andre operativsystemer. Der bruges idag mange ressourcer på at gøre UNIX mere sikkert, f.eks. er Kerberos — et generelt system til at autorisere brugere overfor services — indført i OSFs DCE (Distributed Computing Environment). Et andet eksempel, på at man idag tager sikkerheden alvorligt, er at der er indført autorisation i X11 (R4/R5), der er markedets førende vinduessystem på UNIX arbejdsstationer.

Selvom der fra leverandørernes side gøres meget for at UNIX bliver sikkert, ligger ansvaret i sidste ende hos systemadministratoren. Vi retter derfor specielt vores fokus mod, hvad man kan gøre med et eksisterende UNIX system for at øge sikkerheden.

AALBUG — Aalborg Unix bruger gruppe

Versions kontrol
Tirsdag den 21. April 1992
Kl. 19:00 - 22:30

Aalborg Universitetscenter
Institut for eletroniske systemer
Fredrik Bajersvej 7, bygning D2

Kresten Krab Thorup der er en del af Juntaen på IESD vil fortælle om hvordan man med fordel anvender versions kontrol i software udviklingen. Vægten vil blive lagt på RCS der tilgængelig fra GNU.

AALBUG har fast mødetid og sted nemlig den næstsidste tirsdag i hver måned kl 19:00 på ovenstående lokalitet. Døren til Instituttet er normalt låst udenfor normal arbejdstid, så kom rimeligt præcis, eller bank paa vinduet til auditoriet, hvis du møder en lukket dør.

Der vil være kaffe og te ad libitum, mens øl og vand kan købes

Tilmelding er ikke strengt nødvendig, men meld dig alligevel til således at den fornødne proviant kan tilvejebringes.
Tilmelding:

E-mail	aalbug@dkuug.dk
telefax	98 15 17 39 (att. Peter L. Petersen)
alm. post	Peter L. Petersen
	Inst 8, AUC
	Fredrik Bajersvej 7C
	9220 Aalborg Ø

Vel mødt.

Er UNIX sikkert

Af *Christian D. Jensen*
DKUUG–Nyt

Torsdag d. 12. marts var der medlemsmøde hos Digital i Hørsholm. Mødets titel var "er UNIX sikkert", hvilket også var overskriften på dagens første indlæg af Jørgen Bo Madsen fra UNI•C.

Efter at UNI•C sidste år deltog i opklaringen af danmarks første "hacker–sag", var det naturligt at Jørgen Bo Madsen fortalte om "hacker" problematikken. Gennem jagten på hackerne, har han fået et enestående indblik i "hackernes" verden, og de muligheder der står åbne for dem. Der findes rundt om i verdenen en mængde "bulletinboards", som specialiserer sig i "hackermateriale". Her skal man kende de mest almindelige sikkerhedshuller for at kunne forbinde sig. Når man først har adgang til disse "bulletinboards", bliver mulighederne meget større. Her er det muligt at finde "gør det selv" bøger om hackning, hvor de forskellige leverandørers sikkerhedshuller gennemgås skridt for skridt.

En af grundene, til at "hackerne" har så meget vind i sejlene, er at leverandørerne leverer systemerne meget åbne. Dette skyldes naturligvis konkurrence hensyn, hvor man skal kunne mindst lige så meget som de andre leverandører. Problemet er at de fleste systemer forbliver åbne efter, at de er leveret, fordi kunderne ikke er vidende om eventuelle sikkerhedsproblemer ved den store åbenhed.

Selvom leverandørerne har en del af skylden for de usikre systemer, gør de også meget for, at hjælpe dem der kan hjælpe sig selv. Mange UNIX varianter har idag mulighed for C2 sikkerhed, samt forældelse af "passwords". Når man har disse muligheder, er det vigtigt, at man benytter dem for at højne sikkerheden. Mens talen er ved "passwords" er det meget vigtigt at vælge et godt password. Det er "passwordet" som hindrer at "hackeren" kommer ind på systemet i første omgang (Der er andetsteds i bladet en artikel af Jørgen Bo Madsen om "passwords" og valg af disse).

Som sidste punkt viste Jørgen Bo Madsen, hvordan han – på ejerens opfordring – havde brudt ind på et UNIX system. Metoden var ganske enkelt at lade en PC, som han havde fået på nettet, foregive at være

en betroet maskine. Resultatet var at han i løbet af ganske kort tid var root på den pågældende maskine.

Sikkerhed i distribuerede systemer?

Peter Krogh fra Datacentralen fortalte om hvordan man sikrer sig, når maskinerne er nødt til at have forbindelse til omverdenen.

Hovedvægten blev lagt på konfiguration af ens netværk, med broer og routere, således at man mindsker chancen for ubudne gæster på netværket.

Med brug af filter–broer kan man sikre at kun maskiner med bestemte adresser har adgang til ens netværk. Dette giver samme sikkerhed på netværket, som der er på de forbundne maskiner.

Sikkerhed ved indgangen – phLOGIN

phLOGIN havde sendt Alain Williams, for at fortælle om firmaets `login`–program, der er udvidet med en mængde sikkerheds "features".

Før man beslutter sig for at indføre tredieparts sikkerhedsprogrammel, er det vigtigt at organisationen beslutter sig for en sikkerhedspolitik. Ved en sådan sikkerheds politik skal man overveje følgende:

- Hvad skal beskyttes.
- Hvor meget besvær vil man skabe for den almindelige bruger, for at højne sikkerheden.
- Er alle punkter dækket? (daglig backup er også en del af sikkerheden).
- Hvad sker der hvis politikkerne overtrædes

Gennem det daglige arbejde er der fare for at de sikkerhedsansvarlige stirrer sig blinde på deres eget system. For at undgå dette foreslog Alain Williams, at man allierer sig med en venligsindet sikkerhedsansvarlig/systemadministrator, således at man skiftes til at gennemgå sikkerheden på hinandens systemer. Hvis der kigges på systemet med fremmede øjne, er der en chance for at oplagte huller, eller spor efter "hackeraktivitet", bliver opdaget.

Alain Williams havde lavet en kort liste, over de bøger han selv havde stående på sin reol, som vi bringer her.

- Security Advice for UNIX Machines on a TCP/IP Network
  Jean-Luc Archimaud <jla@imag.fr>
  EurOpen Newsletter, Volume 11 No 2 — Summer 1991.
- UNIX Operating System Security
  F. T. Grampp and R. H. Morris
  AT&T Bell Labs Technical Journal, Volume 63 No 8 Part 2 — October 1984.
- Practical UNIX Security
  Simon Garfinkel and Gene Spafford
  O'Reilly & Associates, ISBN 0-937175-72-2.
- UNIX System Security
  Rick Farrow
  Addison Wesley, ISBN 0-201-57030-0.
- UNIX System Security
  Patrick H. Wood and Stephen G. Kochan
  Simon & Schuster, ISBN 0-672-48494-3.

sikkerhed i netværk

Dagens andet internationale indslag var Martin Bergling fra Communicator Nexus AB, som fortalte om sikkerhedsaspekter vedrørende datanet. Som udgangspunkt definerede han et sikkert system som "et system hvor den *rette* information når den *rette* person i *rette* tid".

For at man kan tale om sikkerhed er det nødvendigt at gøre sig klart, hvad der er fjendebilledet. Truslerne mod et sikkert system udgøres af:

- "Insiders"
- Administrative fejl, herunder
  - ansvarsfordeling
  - ny software
- Driftsforstyrelser
  - fejl i softwaren
  - strømsvigt

- Indbrud, tyveri
- Virus
- Crackere

Det maner til eftertanke, at virus og crackere kommer så langt nede på listen. Selvom det er vigtigt at sikre sine modemmer, er det altså mindst lige så vigtigt at sikre sin strømforsyning.

Den største del af foredraget blev brugt til at fortælle om den standardisering, der er på vej i forbindelse med datanet. Der blev desuden brugt en del tid på at forklare om kryptering og sikker transport af data over potentielt usikre net. Det var en meget grundig og udemærket gennemgang af begge emner, som jeg dog føler det vil føre for vidt at komme ind på her.

UNI•D projektet

Som rosinen i pølse–enden fortalte Ole Carsten Pedersen fra UNI•C om deres eksperimenter med et distribueret miljø. Som organisation bag DE–Net, som er den danske gren af det verdensomspændende internet, yder UNI•C en række ekstra tjenester som tilbydes brugere af DE–Net.

UNI•D projektet går ud på at finde ud af, hvordan disse tjenester kan tilbydes på en nem og sikker måde. Som en del af svaret fandt UNI•C ud af at de måtte basere sig på standarder som AFS (Andrew File System) og Kerberos autorisationssystem. Disse systemer er grundstenene i det kommende DCE (Distributed Computing Environment) fra OSF, som lader til at blive standard.

Konklusion

Hvis der skal drages nogen konklusion på mødet, må det blive at man kan få sit UNIX system så sikkert som man ønsker, men at den højnede sikerhed koster noget af fleksibiliteten. Det er altså op til de enkelte sikkerhedsansvarlige/systemadministratorer, hvor god sikkerheden skal være. En andn vigtig konklusion er at UNIX er ved at blive markedets sikreste system, fordi der foregår et stort og seriøst arbejde på standardiseringsområdet.

Modem'er & Sikkerhed under UNIX

Af *Kim Chr. Madsen*
KIMCM Consult

Introduktion

I de senere år er det blevet mere almindeligt at udnytte modem'er, til overførsler af data mellem EDB-systemer. Det være sig til elektronisk post, filoverførsler eller til terminalsessioner. Ligeledes er EDB-udstyr inden for de seneste år blevet så prisbilligt, at det er blevet allemandseje, med hvad deraf følger af crackere .

Systemadgang

Den nemmeste måde at beskytte sit EDB-udstyr mod eksterne crackerangreb, er ved at sørge for at systemet ikke har nogen eksterne forbindelser. Dette er dog ikke altid ønskeligt, idet man derved afskærer sig fra alle de muligheder der ligger i at kunne kommunikere med andre systemer.

Det næstbedste man kan gøre er at sørge for at eventuelle modem'er på ens system, kun er udgående modem'er, hvilket vil sige at man kan bruge dem til at ringe ud på, men ikke accepterer indgående opkald. Dette sker normalt ved ikke at starte en getty eller uugetty på den port hvor modem'et er tilsluttet.

Det siger dog sig selv at hvis alle følger dette råd, er der ikke nogen man kan kalde op til! Og derfor kan vi prøve at finde ud af hvad man kan gøre, hvis man er villig til at lade andre kalde op til ens system, for at beskytte sig mod ubudne gæster.

Password-beskyttelse

Først og fremmest skal man gøre sig klart, at hvis man anvender standard UNIX-programmel, vil man starte en getty- eller uugettyproces på modemporten, hvilket giver en person der ringer til modem'et

adgang til systemet, på samme måde som sad personen ved en almindelig terminal. Derfor er det, ligesom ved al anden UNIX-sikkerhed, vigtigt at sørge for at alle konti på systemet har "gode" passwords, der er svære at knække.

Hvis ens system er udstyret med "shadow password"-beskyttelse, er det yderligere en god ide at anvende dette, idet det beskytter mod at en cracker der er kommet ind blot kan hente passwordfilen og knække denne på sit eget system (eller en supercomputer hvortil der også er skaffet adgang). Hvis der køres med "shadow password"-beskyttelse, skal en cracker skaffe sig priviligeret adgang til systemet for blot at læse de krypterede passwords.

Hemmeligt Telefonnummer

At have et hemmeligt, ikke registreret telefonnummer til ens modem kan selvfølgelig filtrere de uprofessionelle crackere fra. Men det er ikke nogen videre sikkerhed mod de såkaldte wardialers .

En wardialer er opkaldt efter den metode der blev vist i filmen War Games, hvor en ung knægt leder efter et modem-telefonnummer på et spillefirma, og derfor instruerer sin hjemmedatamat til at ringe op til alle telefonnumre indenfor et område, notere på hvilke telefonnumre der var et modem der svarede. Wardialer programmer findes i mange forskellige udgaver, og er tilgængelige på forskellige arkivmaskiner rundt omkring.

Tilbagekald

Der findes to forskellige metoder man kan anvende for at sikre sig at den der ringer op også er den hun udgiver sig for at være. Dette sker ved på forhånd at have noteret hvilket telefonnummer denne person har. Når personen kalder op til systemet, vil dette ringe tilbage til vedkommende og først på det tidspunkt tillade brugeren normal adgang til systemet.

Dette kan enten gøres ved hjælp af specielle tilbagekaldsmodem'er, eller ved at have to modemgrupper, en indgående og en udgående gruppe. Disse to metoder beskrives nedenfor.

Tilbagekald har dog den ulempe at det er ejeren af systemet der kommer til at betale for telefonregningen, hvorfor det til tider kan være praktisk at have en eller anden form for udligning af omkostningerne, f.eks. ved at lade brugere af systemet betale for opkoblet tid.

Tilbagekaldsmodem'er

Der findes forskellige modemtyper på markedet, der har indbygget en tilbagekalds-mulighed. Ved sådanne modem'er kommer man ikke direkte ind i systemet, men logger ind på modemet ved at angive et modem-password.

Når man har fået forbindelse til modemet, vil dette lægge røret på og ringe tilbage til det telefonnummer, der er blevet gemt for den bruger der loggede ind herpå.

Problemet med tilbagekaldsmodem'er er at det danske telefonsystem, er indrettet således at når A ringer op til B, og B derefter lægger røret på uden at A gør det samme, afbrydes forbindelsen ikke med det samme, men først efter et par minutter. Dette bliver der normalt ikke taget højde for i disse modem'er, hvorfor en cracker blot kan afspille en klartone hvorefter modemet vil ringe tilbage, i den tro at alt er ok, men uden at forbindelsen nogensinde har været afbrudt.

Modemgrupper

Hvis man ønsker den bedste sikkerhed med både ind- og udgående modem'er, kan dette opnås ved at dele sine modem'er ind i en indgående gruppe og en udgående gruppe. Der skal dog normalt laves specialprogrammel til håndtering af dette.

De indgående modem'er tager imod opkald udefra, hvorefter der logges ind gennem et specialprogram, der noterer hvem det er der ønsker at komme i kontakt med systemet, undersøger om denne person har lov til dette, og finder telefonnummeret til denne person frem. Herefter afbrydes forbindelsen og der ringes tilbage på et af de udgående modem'er.

Denne metode sikrer at man kun ringer op til foruddefinerede telefonnumre, men koster selvfølgelig mindst et ekstra indgående modem,

specialprogrammel samt tab af fleksibilitet, idet en "godkendt" person ikke kan kontaktes på andet telefonnummer end det i systemet noterede.

Konklusion

Hvis man ønsker at have modem'er tilsluttet sit system, åbner man mulighed for at crackere kan bryde ind på ens system. Men man kan gøre mange ting for at minimere denne risiko.

- Lade modem'et være et udgående modem, hvis man ikke har brug for at der er nogen der kalder op til modem'et.
- Konventionel beskyttelse, bl.a. password beskyttelse. Sørger for at crackere får svært ved at komme ind på systemet.
- Hemmeligt telefonnummer, er ikke nogen særlig beskyttelse mod "professionelle" crackere.
- Tilbagekalds-metoder, er en af de mest sikre måder at give autoriserede personer lov til at anvende systemet over modem. Der er dog problemer med tilbagekaldsmodem'er, og ingen standardiserede softwareløsninger.

Automatiseret sikkerhed?

Af *Morten Welinder*
DIKU

Indledning

Kan sikkerhed automatiseres og er det i givet fald en fordel eller en sovepude? Det er spørgsmål, jeg vil besvare i denne artikel, hvori jeg vil beskrive to af de frit tilgængelige programpakker, som forsøger at hjælpe med automatiseringen af sikkerheden i et UNIX-system, CRACK og COPS.

Lad mig dog starte med at slå fast, at hvis man vil sikre sit UNIX-system ordentligt, er der kun én ting at gøre: Spurt hen til maskinen og flå alle stik ud af den. Maskinen vil så være sikret mod ethvert misbrug, der ikke involverer fysisk kontakt til maskinen, men heller ikke mere!

En praktisk detalje, der også skal nævnes er, at man skal sørge for at veksle et par ord med de højere magter i sin organisation, *inden* man begynder at beskæftige sig nærmere med sagen. Samarbejdsånden kan få nogle alvorlige knæk, hvis man blot går i gang. Sikkerhedsproblemer betragtes af nogen som følsomme oplysninger.

Hvad kan automatiseres?

Der er tre hovedårsager til huller i sikkerheden på et UNIX-system, nemlig brugernes valg af passwords, brugernes opsætning af deres konti og endelig leverandørens/systemadministratorens opsætning. CRACK tager sig af passwords, medens COPS ser på opsætningen.

I denne tilfælde skal man være klar over, at programmerne ikke gør deres arbejde perfekt; ikke fordi de ikke gerne ville, men fordi det ikke kan lade sig gøre. Det er en umiddelbar konsekvens af Alan M. Turings berømte artikel om beregnelighed fra 1936.

Det eneste, man programmæssigt kan beskytte sig mod eller opdage, er allerede kendte forseelser og fejl. Nye fejl og bommerter skal først

opdages (af de rigtige folk), rapporteres og indbygges i programmerne; i mellemtiden er man ubeskyttet. Det svarer nøje til situationen med computervirus og der er ikke noget at gøre ved det.

Selv om det perfekte ikke er opnåeligt, er der imidlertid ingen grund til apati. Det er tilladt at være naiv og gøre det så godt, man kan.

Crack

CRACK er det massive angreb på en flok brugeres passwords. Efter, at man har kørt CRACK vil mængden af helt tåbelige passwords på ens system være drastisk nedsat. For at forstå virkemåden af CRACK er det først nødvendigt at vide lidt om passwords i UNIX-sammenhæng.

UNIX-passwords

Et password i UNIX-sammenhæng er en tegnstreng på op til otte bogstaver med hver syv betydende bits; ubenyttede tegn svarer til nultegn. Det giver $2^{56} = 72057594037927936$ muligheder, lidt mindre når man ser det upraktiske i at have \n i sit password.

Når man skifter sit password, får man automatisk tildelt en tilfældig kode af 4096 mulige, ens såkaldte "salt". Denne kode bruges til at modificere kryperingsalgoritmen, der anvendes på passwordet. Den krypterede udgave af passwordet er sammen med "salt" til en vis grad offentlig og er at finde i `passwd`-filen. Der findes ingen kendt, simpel metode til at regne baglæns.

Et password checkes ved at kryptere det på samme måde som det oprindelige password og sammenligne resultaterne. Det er med vilje en meget langsom proces, så det er helt urealistisk med almindelige maskiner at prøve alle kombinationer igennem blot for en enkelt bruger. 1000 forsøg i sekundet er i denne sammenhæng kun muligt for meget kraftige maskiner og selv da kun med finpudset maskinkode.

Det er en designmæssig brøler, at "salt" kun vælges blandt 4096 mulige. Det betyder, at det ikke volder væsentligt mere besvær at angribe 100000 brugere end at angribe 10000. Brugere med samme "salt" kan nemlig checkes for samme ord samtidigt.

Hvis brugere spredte deres valg af passwords jævnt over alle mulighederne var der ikke noget password-problem. Det gør brugere desværre ikke.

Crack's første fase

Alt for mange brugere lider af den vrangforestilling, at man kun kan afprøve passwordgæt ved at sætte sig hen til en terminal og forsøge at logge ind. Derfor mener de, at deres navn bagfra — "netroM" — eller måske loginnavnet fordoblet — "useruser" — er glimrende password. Det gør CRACK ikke!

Første fase af et CRACK-angreb er derfor alverdens forvanskninger af brugerens navne, der hentes i `passwd`-filen. Alt prøves forlæns og baglæns, med store og små bogstaver, forskellige sammensætninger og for eksempel tillæg af cifre. "Morten42" er altså ikke et CRACK-sikkert password.

Da det kun er en enkelt bruger, det går ud over, er det realistisk at prøve mange tusinde kombinationer.

Cracks anden fase

Anden fase er langt mere tung. Den går ud over brugere, der er en anelse mere snedige og derfor vælger et password som for eksempel kærestens kats navn, deres foretrukne ølmærke eller en ting, de så, da de valgte password.

Til det formål skal CRACK bruge en stak ordbøger; jo flere og større jo bedre. På UNIX-systemer findes som regel en engelsk ordbog og CRACK indeholder en tåbeordbog med ord som "password", "qwerty", "zxcvbn" og navne. Man bør selv supplere med en dansk ordbog og en liste over danske navne. Man kunne indlæse "Hvad skal barnet hedde?"

Ordbøger gør det imidlertid ikke alene. Man skal også bruge et sæt regler til forvanskning af ordene. Det skyldes dels, at ordbøgerne oftest ikke indeholder bøjningsformer, og dels at mange benytter simple ændringer til at frembringe passwords. Eksempler på sådanne regler er:

- Udskift "to" med "2", så "tonedoev" bliver til "2nedoev".

- Hvis ordet ikke ender på "e" og ikke er mere end fire bogstaver, så tilføj "erne". Altså "flod" bliver til "floderne", men "blaa" bliver til "blaaerne".

- Spejlvend ordet og skriv det med stort begyndelsesbogstav. "spejlet" bliver altså til "Teljeps".

Reglerne kan skræddersys efter behov og evner. Det eneste problem er, at tidsforbruget er proportionalt med gættenes antal.

Cracks evner

Foruden at være fleksibel med hensyn til, hvilke ord der prøves, rummer CRACK en lang række smarte detaljer:

- CRACK kan køre effektivt i netværk, således at en række maskiner kan dele arbejdsbyrden mellem sig efter deres relative kapaciteter.

- CRACK er robust over for nedbrud — man starter blot, hvor man slap.

- CRACK kan undgå dobbeltarbejde ved kun at arbejde på ændrede passwords i forhold til tidligere kørsler.

- CRACK-pakken indeholder en C-funktion, der kan vurdere et password, til at bruge i et nyt passwordprogram.

Cops

COPS er programmet, der kan bruges til at fange egne og andres fejl i opsætningen samt til dels afsløre, om nogen har efterladt bagdøre på ens system. Formålet med COPS er at informere — ikke at rette. Man forventes selv at vide, hvad man skal gøre ved huller, som programmet måtte finde.

COPS er et detaljernes program. Det må det være, eftersom sikkerhedshuller oftest opstår, fordi nogen overser en detalje. Det går ud over en artikel som denne, men jeg håber alligevel den er læseværdig.

Systemopsætning

Et UNIX-system er en kompleks sag — der er altid et eller andet i vejen. Ingen enkeltperson kan forventes at have helt styr på alt og nye huller bliver til stadighed fundet. Her er COPS et glimrende redskab til at henlede opmærksomheden på problemer, inden de bliver alvorlige.

Når man installerer et nyt program, kan det ske, at man efterlader det, så alle kan skrive i det, måske er ens kerne opsat, så `/dev/nit` eller `/dev/kmem` kan læses af alle eller måske indeholder root's søgesti det aktuelle katalog. En lang række af sådanne selvstændigt set harmløse detaljer kan med den rigtige (forkerte?) viden bruges skadeligt.

Et helt andet problem er kendte fejl i programmer. Her sørger COPS for at undersøge, om nogle programmer er dateret før fejlene opdagedes op derfor er sårbare. Metoden er ikke skudsikker, men hvis man ellers bruger nyeste version af COPS har man nogen sikkerhed. Metoden med datachecket er i øvrigt elegant; man fortæller ingen, hvorfor et givet program er sårbart, kun at man bør skifte det ud.

Opsætning af konti

Mange brugere udviser uvidenhed eller skødesløshed ved opsætningen af deres konti, selv skrivbare hjemmekataloger er ikke noget særsyn, så COPS undersøger, om en række forhold, primært med hensyn til skriverettigheder, for hver enkelt bruger er i orden.

Skriverettigheder for alle eller ens gruppe bør begrænses mest muligt. I praksis undersøger COPS hjemmekataloget og en række (forudbestemte) punktumfiler, der kan benyttes til at starte programmer. Hvem tænker for eksempel på, at `.forward` er et meget direkte adgangskort til en konto? Her kommer automatikken i øvrigt til kort. Mange brugere laver struktureret kodning i deres setup og kalder underprogrammer. Typisk ses "`source .alias`", hvilket ikke får COPS til at undersøge `.alias`.

Bagdøre

Har en cracker været på besøg, kan han meget vel tænkes at have efterladt en såkaldt bagdør, der senere giver ham mulighed for at vende tilbage uden for stort besvær. Bagdøre findes i mange forskellige

udgaver; ikke alle er lige lette at opdage. I et forsøg på at finde bagdøre gør COPS mange ting:

- Finder suid-filer ejet af root. Kommer en ekstra fil på listen, er det grund til overvejelse.
- Undersøger, at `passwd`-filen har det rigtige format og ikke indeholder for mange brugere med superuserstatus.
- Ser efter, at `cron` ikke udfører programmer, nogen kan rette i.
- Undersøger, om `/etc/hosts.equiv` indeholder et "+", der giver alle netadgang til ens maskine.
- Vedligeholder et checksumsregister for ens programmer, så man kan se, hvis nogen har installeret nye versioner af kritiske programmer, `/bin/login` til eksempel.

En konklusion

Kan de så bruges til noget, disse to programmer? Efter min mening må svaret være et "tja..."

På den ene side, kan det være nyttigt at tage pulsen på ens system og få stoppet de værste huller med jævne mellemrum. Da de to beskrevne programmer ikke kræver den store menneskelige indsats, er det nok i det lange løb en fordel at bruge dem.

På den anden side, vil systemadministratorer altid være mindst et skridt bag efter crackere. CRACK og COPS kan eliminere mange crackere, men ikke alle og de der bliver tilbage er de bedste! De ved, hvordan de bliver superuser i frokostpausen uden at efterlade andet spor end en urørt madpakke.

Jeg vil selv anbefale at køre COPS med et par måneders mellemrum for at holde brugerne lidt i ørerne. CRACK kan jeg ikke anbefale at køre mere end én gang, for efter min mening bør dårlige passwords stoppes allerede, når de foreslås. Endelig erindrer jeg om, at man ikke skal føle sig sikker, bare fordi automatikken ikke kan finde noget galt.

Vide mere

COPS kan fås ved anonym `ftp` til `ftp.diku.dk` i `/pub/cert/cops/`. Pakken henviser til CRACK. Begge pakker indeholder fyldige beskrivelser og masser af forbehold.

Klubaften i København

Tirsdag den 28. april 1992
Kl. 19:00 – 22:30

Datalogisk Institut (DIKU)
Universitetsparken 1
(indgang fra Nørre Alle)
Tema:
Bridges, Routers og Gateways

Klubben vender hermed tilbage til temaet netværk, samtidigt med at der legges vægt på sikkerheden. Broer, rutere og "gateways" benyttes alle til at sektionere netværker, og er med til at give en forbedret sikkerhed.

Sikkerhed og Netværk

Af *Peter Holm*
Danosi

Sikkerheden omkring kommunikation og netværker kan opdeles i to områder:

- Sikkerhed mod at uvedkommende tapper eksisterende kommunikationsforbindelser, og dermed kan komme i besiddelse af informationer.
- Sikkerhed mod at uvedkommende gennem kommunikationsnetværket får adgang til systemer, således at de derved kan udtrække eller ændre informationer.

Sikkerhed mod tapning

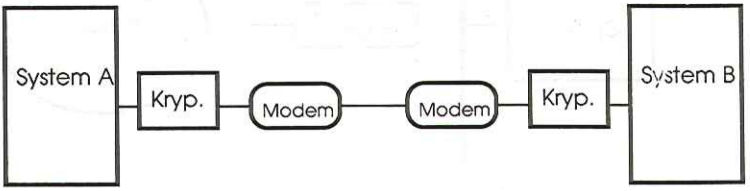
Sikerhed mod tapning af oplysninger er normalt kun aktuelt, hvis man har meget fortrolige oplysninger, men i disse tilfælde, stilles der selvfølgelig meget store krav, både til beskyttelse mod tapning og adgang til systemet.

Hvis man skal have mulighed for at tappe oplysninger på kommunikationsforbindelser, vil det i de fleste tilfælde kræve at man er rimelig nær forbindelserne. Den vigtigste beskyttelse er derfor at sørge for at uvedkommende ikke har adgang til kommunikationsforbindelser som lokalnet, telefonledninger, terminalforbindelser mv. Åbner man først adgang til de nævtne forbindelser, er det rimeligt let at tappe disse for informationer. Den eneste beskyttelse i disse tilfælde, er at kryptere de data der sendes. Denne kryptering kan foretages på flere måder:

I lokalnet miljøer, er der mulighed for at vælge specielle net, der har indbygget kryptering. Netværket sørger således for at al trafik på nettet sendes krypteret, men præcenteres på normal vis over for de tilsluttede systemer.

En anden, og måske lettere tilgængelig løsning, er at kryptere informationerne i systemerne inden de overføres på lokalnettet. Mange operativsystemer anvender for eksempel denne metode ved overførsel af passwords.

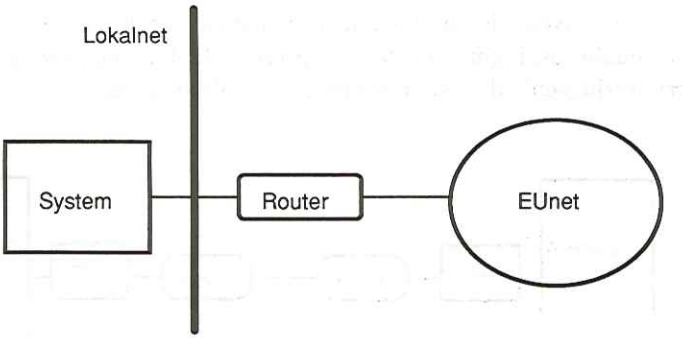
Ved anvendelse af telefonledninger til datakommunikation, findes der enkle løsninger, der kan kryptere data blot ved indskydelse af en krypteringsenhed mellem system og modem (se fig.).



Krypteringsenhederne vil her kryptere data, og linien optræder helt transparent over for de tilsluttede systemer.

Sikkerhed mod uvedkommendes adgang gennem kommunikationsnet

Uvedkommende kan have mange muligheder for at komme i kontakt med systemet. En af de i øjeblikket mest omtalte, er adgang ved hjælp af opkaldsmodem. Hvordan man kan sikrer sig mod denne form for misbrug, vil jeg ikke omtale her, da der allerede har været skrevet en del om dette emne. Mange systemer er idag også forbundet indbyrdes via (verdens) omspændene net (f.eks Internet, EUnet, EARN etc.) og her ønsker man selvfølgelig at kunne bruge disse net uden at uvedkommende kan få adgang. Dette er på mange måder en svær opgave, men der findes dog muligheder for at øge sikkerheden i sådanne net. En forudsætning, hvis ens system er tilsluttet et sådan net, er selvfølgelig at sikkerheden på selve systemet er i orden. I disse store net giver nettet normalt mulighed for at alle kan kommunikere med alle, og den første opgave vil derfor normalt være at få filtreret en del af kommunikationsmulighederne fra. Til dette formål vil man normalt anvende en router (se fig):



I mange routere kan man indsætte filtre, der begrænser adgangen til ens eget net og dermed også ens systemer. Routere findes i mange forskellige udgaver og prislag, men interesserer man sig for sikkerhed skal blikket nok vendes mod de større modeller som f.eks. Wellfleet og Cisco, da filtrering er en rimelig ressourcekrævende opgave.

Anvendes for eksempel TCP/IP i ens kommunikationsnet vil man blandt andet kunne filtrere på følgende:

Afsender adresse:

Netværkstrafik vil normalt indeholde adressen på modtageren og afsenderen. Dette giver en umiddelbar mulighed for at filtrere trafik fra "ukendte" adresser fra, således at man kun lader trafik fra kendte adresser passere. Dette giver en rimelig sikkerhed mod at uvedkommende får adgang til systemet, men det skal understreges at denne sikkerhed ikke er fuldstændig. En fuldstændig sikkerhed vil kræve at hele nettet er sikkert, og det har man normalt ingen garanti for.

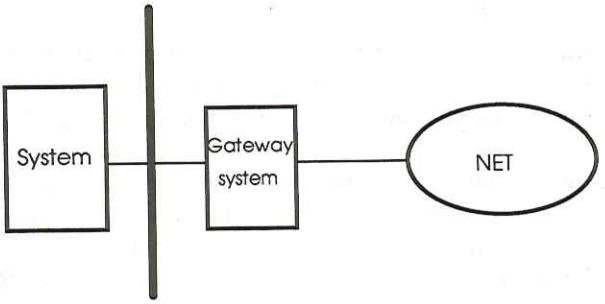
Services:

I TCP/IP net kan man også på netværkstrafikken adskille forskellige services, såsom terminaltrafik, filoverførsler, elektronisk post, navneservice etc. Ønsker man for eksempel kun at udveksle post med omverdenen, vil det være fornuftigt at indføre filtrering således at kun elektronisk post kan komme igennem. I denne forbindelse vil det selvfølgelig være væsentligt at man har sikret sin modtagelse af elektronisk post,

således at dennes vej ind i systemet ikke kan misbruges. En total filtrering af f.eks. terminaltrafik er måske ikke altid ønskelig, idet man ofte gerne selv vil kunne anvende netværket. I disse tilfælde vil det ofte være muligt at filtrere på retningen af forbindelsen, således at terminalopkald ud i nettet tillades, men at opkald ind til ens egne systemer forbydes.

Gateway systemer

En anden måde at sikre sine systemer på, er anvendelsen af et "gateway" system (se fig.).



Ved denne metode tilsluttes de eksterne net til et enkelt system, der så kan give en videreforbindelse til ens øvrige systemer. Denne løsning baserer sig udelukkende på sikkerheden i gateway systemet, dvs. login og password, og det er derfor vigtigt at alle uvedkommende bremses her. Denne løsning er ikke helt så fleksibel, som den tidligere, da filoverførsel f.eks. må foregå med flere hop.

Husk!

Ved vurdering af sikkerheden omkring tilslutning til netværk, er det væsentligt at prøve at samle al kontrol af uønsket access på så få komponenter som muligt. Det er enklest at foretage en grundig filtrering og kontrol et enkelt sted, frem for at have mange forskellige adgangsveje til ens systemer. Og husk: Faste linier giver ikke bedre sikkerhed end den sikkerhed, der findes i den anden ende af linien.

Er UNIX Sikkert?

Af *Bjørn Johannesen*
Control Data

Proprietære systemer bliver erstattet af åbne systemer, hvor decentrale UNIX-maskiner er bundet sammen i netværk. Den tidligere centrale maskine bliver erstattet af et åbent system.

Fordelene ved denne "downsizing" eller "rightsizing" er mange. Jeg skal ikke komme ind på alle fordelene ved UNIX, idet dette kunne fylde hele bladet. Jeg vil udelukkende koncentrere mig om en relativ ny, stærk side ved UNIX: – Sikkerhed.

Åbne systemer og netværk har tidligere betydet, at brugerne måtte give afkald på den adgangskontrol og datasikkerhed, der er en selvfølge i mainframe verdenen. Dette har givet afholdt mange fra at vælge UNIX.

UNIX har efterhånden en del år på bagen og fremstår i dag som et stabilt styresystem, der kan afvikles på maskiner, der ydelsesmæssigt kan måle sig med mainframes. Den kritik, der tidligere var vedrørende den manglende sikkerhed, eksisterer nu ikke mere. Selv Registertilsynets krav tilgodeses fuldt ud.

UNIX er blevet et reelt alternativ til de centrale, proprietære systemer.

Datasikkerhed

Der findes i dag flere teknikker til at undgå tab af data. Sikkerhedskopiering, der foretages jævnligt, er med til at reducere omfanget af katastrofen, den dag disken står af, eller data på anden måde går tabt.

Som en ekstra sikkerhedsfaktor har de moderne database systemer logningsfaciliteter, der yderligere reducerer risikoen for tab af data.

Vitale data kan sikres med spejlede diske, som er en kendt teknik til at håndtere diskfejl. En endnu bedre sikring, oven i købet med mindre lagerforbrug end ved spejlede diske, er Disk Array Systemer, der kan konfigureres med paritetsdiske. Systemet fungerer på den måde at der parallelt foregår læsning/skrivning på 4 diskenheder. Een af disse er en

paritetsdisk. Hvis ulykken skulle ske, kan den defekte disk udskiftes og data genetableres. Dette foregår, medens systemet kører og varer kun nogle få minutter. På denne måde er man principielt sikret mod tab af data ind til dommedag. Oven i købet får man en ydelsesmæssig fordel, idet brugerdata læses og skrives samtididigt på flere fysiske enheder.

Åbne systemer er – åbne

Åbne systemer har tidligere været kendetegnet ved, at de har været åbne, når det gjaldt adgangskontrol. UNIX er ofte blevet kritiseret for manglende adgangskontrol og brugerstyring i det hele taget. Disse mangler er blevet mere og mere åbenlyse i takt med UNIX's voksende udbredelse. Brugere af lukkede systemer har haft den nødvendige sikkerhed og savner den selvfølgelig ved overgang til åbne systemer. At man downsizer, rightsizer, decentraliserer — eller hvilken betegnelse man måtte foretrække — er ikke ensbetydende med, at krav om sikkerhed og ydeevne ændres. Det centrale system erstattes af netværket. Den sikkerhedsstyring, der tidligere naturligt lå hos datacenteret, skal nu implementeres i et distribueret, åbent miljø.

Adgangskontrol

Det amerikanske forsvar har beskrevet forskellige klassifikationsgrader, der går fra ingen sikkerhed til et sikkerhedsniveau, der er så restriktivt, at det ikke er praktisk anvendeligt.

Et sted midt imellem disse to yderpunkter findes et sikkerhedsniveau, C2, der giver mulighed for sikker adgangskontrol uden at en stor, fordyrende organisation skal etableres.

Adgangskontrol er en mulighed — ikke en restriktion — og tilpasses den enkelte organisation.

Monitorering

Hvis der skulle opstå mistanke om uautoriseret adgang til data, er logning af kritiske operationen et uundværligt værktøj. Som en del af de sikkerhedsfunktioner, der kan leveres til UNIX, findes Audit-logning, der er en historikfil med registrering af operationer som:

Systemopstart- og nedlukning
Login og Logoff
Oprettelse, sletning eller rettelse af filer eller attributter
Oprettelse/sletning af kataloger
Kald af specifikke kommnadoer eller programmer.

Envidere er der mulighed for at registrere afviste adgangsforsøg og forsøg på uretmæssig anvendelse af systemressourcer. Da både tidspunkt, brugerid og beskrivelse af udførte operation registreres, kan analyseprogrammer danne rapporter til sporing af eventuelle forsøg på sikkerhedsbrud.

Kerberos

Adgangskontrol på den enkelte UNIX maskine styres med C2-faciliteter. Men der kan opstå problemer med udveksling af passwords og adgang til foreskellige ressourcer i netværket. For at styre dette, er der beskrevet en protokol, Kerberos.

Kerberos autoriserer serviceudbydere, klienter og brugere på nettet og er udformet til at kunne modstå selv de mest sofistikerede indtrængningsforsøg.

Brugere, maskiner og services identificeres over for Kerberos i en særlig database, som er under kontrol af den såkaldte Kerberos-server. Denne er placeret på en central maskine på nettet — selvfølgelig kryptograferet.

Kerberos-serveren udsteder samtlige tilladelser til brugeren på basis af brugerens password. Brugeren behøver således ikke at foretage fornyet login på andre systemer eller andre maskiner, da brugerens rettigheder er beskrevet i Kerberos' database.

En speciel finesse er, at brugerpassword ikke sendes på nettet, men udelukkende anvendes lokalt på arbejdsstationen til dekryptering af login-tilladelsen fra Kerberos. Det er således ikke muligt at opsnappe passwords ved lytning på nettet.

Genbrug af fælles ressourcer

I et flerbrugersystem anvendes fælles ressourcer som internt lager og eksterne lagringsmedier som disk og magnetbånd.

I forbindelse med læsning og skrivning til ydre enheder, vil data være tilgængeligt for andre programmer. Data kan ligge i hukommelsen, efter at læse/skrive operationen er udført. Desuden vil data på en disk ikke blive slettet fysisk. Det er kun indekset, der fjernes.

For at hindre uautoriseret adgang til data, skal data derfor slettes fysisk, både på disk og hukommelse, når de ikke længere er ibrug. Denne, ikke uvæsentlige detalje, kan også varetages af UNIX i dag.

Magnetbånd er et "klassisk" medie fra mainframe-miljøet, og UNIX kan i dag give den samme form for sikkerhed, der kendes fra de store, centrale systemer. Med magnetbåndlabels efter ANSI-standarden kan UNIX sikre, at det korrekte bånd anvendes til en given funktion, og at båndet ikke overskrives før sin udløbsperiode.

Er UNIX sikkert?

Sikkerhed er mange ting. Det er datasikkerhed, adgangskontrol og netværkssikkerhed.

Svaret på spørgsmålet i overskriften må være:

Ja, UNIX er sikkert. Både adgangskontrol, netværkssikkerhed og sikkerhed mod tab af data kan i dag leveres til UNIX, så det tilfredstiller de krav, der kan stilles til sikkerhed. Selv mainframe-brugere vil kunne føle sig trygge ved UNIX.

UNIX er i dag et stabil operativ system, hvortil der kan leveres adgangskontrol, monitorering og netværkssikkerhed.

Spejlede diske og konfigurationer med paritets diske giver sammen med moderne database systemer en meget høj grad af data tilgængelighed.

Valget af kodeord er vigtigt på et EDB–system

Af *Jørgen Bo Madsen*
UNI•C

Denne artikel beskriver algoritmen, der krypterer brugernes kodeord (password) på et UNIX–system og hvorfor det er så vigtigt at vælge et godt kodeord.

NBS (National Bureau of Standards) opfordrede i 1972 til at indsende forslag til en offentlig tilgængelig krypteringsalgoritme (dvs. uden kopibeskyttelse osv.).

Kriterierne for udformningen var blandt andet:

- Højt sikkerhedsniveau
- Tilgængelig for alle brugere
- Kildeteksten må ikke være hemmelig
- Skal kunne implementeres elektronisk (i en chip)
- Portabel

IBM udviklede i 1974 DEA (Data Encryption Algorithm) baseret på "Lucifer"–algoritmen. Efter analyser og kontrol blev DEA i 1976 til DES, der herefter blev indført i UNIX. Algoritmen har kun en privat nøgle (kodeord) hvilket betyder, at både sender og modtager skal kende den samme nøgle. Nøglelængden i IBM's Lucifer–nøgle var 128 bit, hvorimod DES–nøglen kun er 56 bit (dvs. otte 7–bit tegn).

DES er en kompleks kombination af to fundamentale byggeklodser: Substitution og permutation. Styrken i algoritmen består i at anvende disse teknikker 16 gange for hver 8 byte-blok.

Tre krypteringsalgoritmer

Lad mig med det samme slå fast, at der i UNIX-sammenhæng er tale om tre forskellige krypteringsalgoritmer:

1. des (1) er en kommando. DES (16 iterationer) krypterer og dekrypterer DATA og kan implementeres som en chip. Denne kommando eksisterer stort set kun i USA.

2. crypt (3) er et funktionskald. Crypt.c er baseret på DES, (25 iterationer) men algoritmen er envejs, og kun beregnet til kodeord.

3. crypt (1) er en kommando. Crypt baserer sig på en algoritme tilbage fra anden verdenskrig. Den er let at bryde så lad være med at stole på den.

På grund af eksportbegrænsninger, må kildeteksten til DES ikke eksporteres fra USA og Canada uden særlig tilladelse til trods for, at algoritmen er beskrevet i adskillige bøger og tidsskrifter verden over.

I det følgende vil jeg udelukkende beskrive password – DES. Iøvrigt har det amerikanske telefonselskab AT&T kopirettighederne til algoritmen.

Selvom kildeteksten til krypteringsalgoritmen er tilgængelig ved mange installationer, er der endnu ikke set noget tegn på, at det er muligt at dekryptere det krypterede kodeord.

Envejs-kryptering er en funktion hvor kryptering er relativt nemt, men hvor dekryptering er relativt svært. Et eksempel herpå er funktionen x^3 som er let at udregne, mens den omvendte funktion, $\sqrt[3]{x}$, er svær at udregne.

Brugerregistreringsfilen

En af de vigtigste datafiler på et UNIX–system er brugerregistreringsfilen /etc/passwd. Hver linie i denne fil er inddelt i 7 felter: brugernavn, krypteret kodeord, brugernummer, gruppenummer, brugerens fulde navn, hjemmekatalog og kommandofortolker. Et eksempel på en registrering af brugernavnet uniok:

```
uniok:E6vPXbZPMec7c,o.KF:217:101:Ole Kj{rgaard:/unic/uniok:/bin/ksh
```

Det er nødvendigt, at alle brugere har læseadgang til brugerregistreringsfilen af hensyn til alle de kommandoer og funktioner, der læser brugernes data. For eksempel har kommandoerne `ls`, `find` og

`chown` brug for at konvertere mellem brugernavn og nummer, idet alle brugernes filer er registreret med numre.

Feltet med det krypterede kodeord er 13 tegn. De 2 første tegn er SALT–værdien. SALT er et 12–bit nummer (4096 muligheder) der er sammensat af procesnummeret og systemtiden. De næste 11 tegn er selve det krypterede kodeord. Ydermere kan der være tilføjet et komma og 1 til 4 tegn (18 tegn i alt). De 3 tegn efter kommaet angiver: Maximal levetid, Minimal levetid og hvornår kodeordet sidst er skiftet.

Ved login på et UNIX–system indtaster brugeren først sit brugernavn og derefter sit kodeord. Loginprogrammet søger i brugerregistreringsfilen efter det netop indtastede brugernavn og tilhørende SALT–værdi. En blok af nuller krypteres med brugerens kodeord som nøgle og med anvendelse af SALT–værdien. Resultatet af krypteringen sammenlignes med brugerens krypterede kodeord fra brugerregistreringsfilen. Er de to krypterede kodeord identiske, får brugeren adgang til maskinen. Krypteringshastigheden er den samme, uanset om der er valgt et kodeord på 1 eller 8 tegn. Afgives der forkert kodeord, venter loginprogrammet et antal sekunder, før der promptes for et nyt. Denne ventetid sikrer, at afprøvning af mange forskellige kodeord på det samme brugernavn forsinkes betydeligt.

SALT–værdien sikrer, at to brugere med samme kodeord ikke har det samme krypterede kodeord. Faktisk kan man kopiere det krypterede kodeord til et nyt brugernavn på et andet UNIX–system, forudsat at man dér er priviligeret bruger, for herefter at logge ind med det nye brugernavn og det "gamle" kodeord. Det er samtidig et bevis på, at der er noget kode, der er standard på alle UNIX–systemer :-).

Gætter kodeord

Umiddelbart ser det ikke ud til at være noget problem, at alle kan læse brugerregistreringsfilen, *men* da alle kan læse brugernes krypterede kodeord, kan alle prøve at gætte det. Og det er lige netop det hackerne gør. De er helt klare over, at det er umuligt at dekryptere brugernes kodeord. Istedet gætter de brugernes kodeord udfra forskellige informationer om brugerne og en stor ordbog. Ordbogen kan være op til 100.000 ord. En af de hurtigste crypt–funtioner krypterer ca. 750 kodeord pr. sekund, på en DECstation 3100.

Brugere anvender næsten altid simple kodeord der kan huskes. Det er ofte navne på familiemedlemmer eller bekendte. Også omskrivninger af brugernavn og personlig information anvendes som kodeord. Og det er netop det Cracker–programmerne udnytter. Eksempler på knækkede Kodeord er: SECRET, gyldespreder, qwerty, mastermind, wiseguy, astrid90, 8jenser, idjjdi og kotobuki (betyder lykke pe japansk). Generelt skal man undgå at bruge enkeltord, og ord der findes i en ordbog eller på elektronisk form. Det er nemt at give eksempler på dårlige kodeord hvorimod eksempler på gode kodeord er svære.

At få adgang til brugerregistreringsfilen er normalt ikke noget problem. Hvis hackerne ikke er registreret på den UNIX–maskine de vil cracke, anvender de programmerne: 'tftp', 'ypcat', 'finger', 'ftp' med flere. Tftp og ypcat bruges til at kopiere brugerregistreringsfilen til hackermaskinen uden angivelse af brugernavn eller kodeord. Finger anvendes til at få overblik over, hvem der er registreret på en fremmed UNIX–maskine og deres personlige informationer. Ftp anvendes til at angribe den fremmede UNIX–maskine med en dedikeret liste af brugernavne og kodeord. Langt de fleste UNIX-systemer registrerer *ikke* forkerte kodeord, der stammer fra ftp!

Superbruger privillegier

En anden vigtig datafil på et UNIX-system er grupperegistreringsfilen /etc/group. Hver linie i denne fil er inddelt i mindst 3 felter: gruppenavn, krypteret kodeord og gruppenummer. Det fjerde felt indeholder en liste over tilknyttede brugernavne. Et eksempel på en registrering af gruppenavnet staff:

staff:VIghIpei55H0w:3:hansen,jensen,frandsen

Det er nødvendigt, at alle brugere har læseadgang til grupperegistreringsfilen af hensyn til alle de kommandoer og funktioner, der læser gruppernes data.

Kodeordet til en priviligeret gruppe kan udnyttes til at opnå Super-User–privilegier.

Sikkerhedsniveauet på de forskellige maskiner afhænger af, hvilke typer data der behandles. Uanset hvad maskinen anvendes til, vil jeg

anbefale sikkerhedsniveau C2 (Discretionary Access Control + Audit) eller højere. C2–sikkerhedsniveauet flytter brugernes og gruppernes krypterede kodeord over i særlig datafiler, hvor der *ikke* er læseadgang for alle. Samtidig er det muligt at registrere hver gang der afgives forkert kodeord. C2 øger sikkerheden *men* det hjælper ikke at forbedre sikkerheden på et UNIX–system, eller at investere i software til at øge sikkerhedsniveauet, hvis brugernes kodeord er for nemme at gætte.

Gode kodeord

Gode kodeord skal have følgende egenskaber:

- læres udenad
- lette at huske
- hurtige at indtaste
- ikke nedskrevet
- hemmelige og personlige
- skiftes mindst hvert halve år
- mindst 8 tegn lange, hvis de er længere trunkeres automatisk til 8 tegn

Desuden skal gode kodeord være svære at gætte. Kodeord med tastatursekvenser (som qwerty eller zxcvbn), navne, tlf. nr. nummerplader og ord der findes i ordbøger er alle *nemme* at gætte. Et godt kodeord skal være kryptisk, opbygget af store og små bogstaver samt tal og/eller specialtegn. Specialtegnene bør ikke være de første eller de sidste, da nogle "password–crackere" såkan gætte dem. Eksempler på gode kodeord er:

- avMINryg
- steN+Tag
- ur+-12ti mer
- spiS14fi sk
- TINES2ru nde
- moS8Toma ter
- henT6Kru s

- bil.BAKK ER
- TEGN-lig e

En god måde at lave et kryptisk kodeord er at vælge en linie fra et digt eller en sang, man tager så det første eller de to/tre første bogstaver i hvert ord, og sætter dem sammen til et kodeord. Disse kodeord er nemme og huske, men samtidigt meget svære at gætte. Eksempler på sådanne kodeord er:

- `LPE,sovo`, som stammer fra "Lille Peter Edderkop, stod og vaskede op".
- `Iekssk,adn`, som stammer fra "I en kælder sort som kul, aller dybest nede".
- `Ospekos`, som stammer fra "Ole sad på en knold og sang".

Kerberos: Sikkerhed i åbne netærk

Af *Peter Holst Andersen*
DIKU

I et åbent miljø hvor arbejdsstationer og netværk er frit tilgængelige, kan man ikke umiddelbart stole på en arbejdsstations identifikation af en bruger. Kerberos er et system udviklet under MIT's Athena projekt, hvor en trejde-part er betroet til at stå for identifikationen. Denne artikel er baseret på *Kerberos : An Authentication Service for Open Systems* af J. G. Steiner, C. Newman og J. I. Schiller og beskriver hvordan Kerberos systemet fungerer. Centralt for Kerberos systemet er at nogen, der lytter med på netværket, ikke vil være i stand til at opsamle information, der giver ham mulighed for at maskere sig som en anden bruger.

Hvis man vil tilbyde forskellige tjenester (services) over et netværk, så som e-mail, printer tilgang og NFS, og man ønsker at sikre sig at disse ikke kan misbruges, er der forskellige muligheder : I et lukket miljø, hvor alle maskiner er under kontrol fysisk, kan man overlade sikkerheds-check til maskinen, hvor brugeren logger ind. I et mere åbent miljø, kan man nøjes med at stole på de maskiner, som man ved er under kontrol. I et helt åbent miljø, må man kræve at brugeren beviser sin identitet for hver netværks tjeneste. Kerberos benytter den sidstnævnte fremgangsmåde, selvom det ikke betyder at en bruger skal indtaste sit password, hver gang han skal udskrive på printeren.

Nøgler og *passwords*. Kerberos bruger "private key" (i modsætning til "public key") kryptografering, som baserer sig på at to, der ønsker at kommunikere uden andre skal vide hvad de kommunikerer om, kender en nøgle som ingen andre kender. I et "public key" system har alle en privat nøgle, som kun de kender, og en offentlig nøgle, som alle kender.

Kryptograferingen i Kerberos er baseret på DES (Data Encryption Standard), hvilken desværre er omfattet af USA's eksport-restriktioner. Der er dog en mulighed for at udskifte kryptograferings-modulet i Kerberos.

Lige som hver bruger har et password, er der til hver netværk tjeneste knyttet et password, som kun er kendt af den pågældende tjeneste og Kerberos.

Billetter og *legalisatorer* (eng. *authenticator*). I Kerberos bruges billetter til hemmeligt at sende identiteten af den bruger, som billetten oprindeligt var udstedt til, mellem Kerberos og den server som tilbyder den ønskede tjeneste.

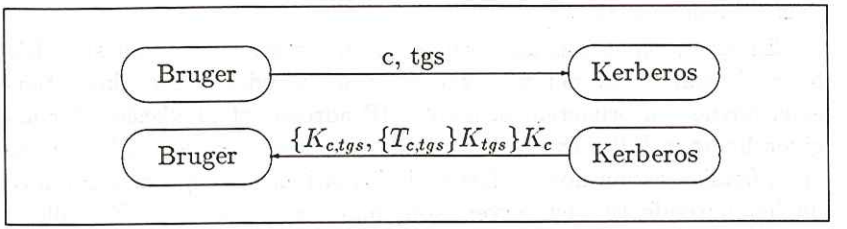
En given billet kan kun bruges af én bruger til én tjeneste. Billetten består af navnet på serveren, som tilbyder den ønskede tjeneste, navnet på brugeren, brugerens IP adresse, et klokkeslet, der angiver hvornår billetten er blevet udstedt, levetiden for billetten, og en tilfældig session-nøgle. Denne information er kryptograferet med nøglen hørende til den server, som billetten passer til. En billet : $\{\text{server, client, addr, timestamp, life, } K_{s,c}\}K_s$. Når billetten først er blevet udstedt, kan brugeren anvende den flere gange for at opnå en given tjeneste hos den pågældende server, indtil billettens levetid udløber.

En legalisator kan derimod kun bruges én gang. En ny skal oprettes hver gang brugeren ønsker at benytte en tjeneste. Dette er ikke noget problem, idet bruger selv er i stand til at oprette en legalisator. Den består af navnet på brugeren, brugerens IP adresse og arbejdsstationens nuværende klokkeslet. Legalisatoren er kryptograferet med session-nøglen, som er en del af billetten : $\{\text{client, addr, timestamp}\}K_{s,c}$.

Den initielle identifikation. En bruger går hen til sin arbejdsstation og indtaster sit brugernavn. Over netværket sendes en forespørgsel til Kerberos, bestående af brugernavnet og navnet på en speciel tjeneste, nemlig den *billet-udstedende tjeneste* (eng. *ticket-granting service*).

Hvis Kerberos kender brugeren, oprettes en tilfældig session-nøgle ($K_{c,tgs}$), som senere vil blive brugt mellem brugeren og den billet-udstedende tjeneste. Der oprettes også en billet til den billet-udstedende tjeneste, bestående af brugerens navn, et klokkeslet, levetiden for billetten, brugerens IP adresse og den session-nøgle, som lige er blevet oprettet. Denne billet ($T_{c,tgs}$) bliver kryptograferet med en nøgle K_{tgs}, som kun er kendt af Kerberos og den billet-udstedende tjeneste.

Kerberos sender nu den kryptograferede billet $\{T_{c,tgs}\}K_{tgs}$, en kopi af session-nøglen $K_{c,tgs}$ og noget yderligere information tilbage til brugeren. Svaret kryptograferes med en nøgle K_c, som kun er kendt af Kerberos og brugeren, idet den er genereret ud fra brugerens password. Se figur 1.



Figur 1: Den initielle identifikation.

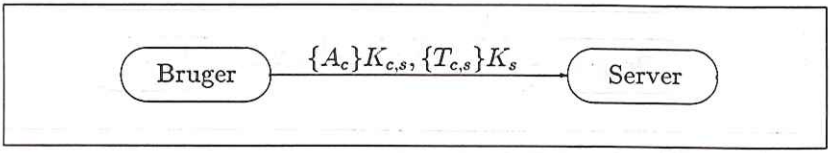
Når svaret bliver modtaget af brugerens maskine, bliver brugeren bedt om at indtaste sit password. Maskinen vil nu være i stand til at dekryptografere svaret fra Kerberos. Billetten ($\{T_{c,tgs}\}K_{tgs}$) og session-nøglen ($K_{c,tgs}$) gemmes til senere brug, mens brugerens password og den dertil hørende nøgle K_c slettes fra maskinens lager.

En tjeneste forespørgsel. Lad os antage at brugeren ønsker at benytte en bestemt netværk tjeneste (f.eks. e-mail), og at brugeren allerede har en billet til at benytte denne tjeneste. I næste afsnit bliver det forklaret hvordan man får en billet. Billetten vil være krypteret med en nøgle K_s, som kun er kendt af Kerberos og den ønskede tjeneste. Billetten vil blandt andet indeholde en kopi af vores session-nøgle $K_{c,tgs}$.

Applikationen opretter en legalisator, bestående af brugerens navn og IP adresse, samt det nuværende klokkeslet. Legalisatoren kryptograferes med vores session-nøgle $K_{c,tgs}$ og sendes sammen med billetten ($T_{c,s}$) til den server, som tilbyder den ønskede tjeneste.

Når serveren modtager legalisatoren og billetten, bliver billetten dekryptograferet og da billetten indeholder en kopi af session-nøglen vil serveren også være i stand til at dekryptografere legalisatoren. Serveren sammenligner informationen i billetten med informationen

i legalisatoren og det nuværende klokkeslet. Hvis alt stemmer vil serveren lade forespørgslen gå igennem. Se figur 2.



Figur 2: En tjeneste forspørgsel.

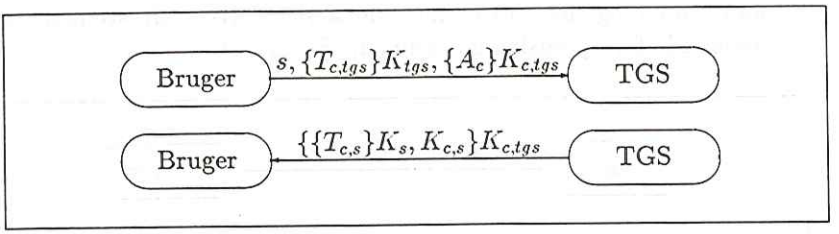
Det forudsættes at maskinernes ure er synkroniseret, så de ikke afviger med mere end et par minutter. Hvis en billet er dateret ude i fremtiden eller er for gammel, vil serveren behandle forespørgslen som et forsøg på at genbruge en tidligere forespørgsel. Serveren kan også holde styr på tidligere forespørgsler, der endnu ikke er blevet forældede, så den kan identificere forsøg på at genbruge forespørgsler.

Hvis brugeren ønsker det, kan han kræve, at serveren også skal bevise sin identitet. Dette gøres ved at serveren lægger 1 til klokkeslettet, som brugeren sendte som en del af sin legalisator og sender resultatet tilbage kryptograferet med session-nøglen.

Efter disse udvekslinger har serveren Kerberos' garanti for at brugeren er den, som han udgiver sig for at være. Og hvis brugeren ønsker det, har han Kerberos' garanti for at det ikke er en falsk server. Ydermere deler brugeren og serveren en nøgle, som kun de to kender, og kan derfor regne deres senere kommunikation for sikker.

Hvordan får man billetterne ? Som nævnt tidligere kan en billet kun bruges til én tjeneste. Brugeren må således få fat på en billet for hver tjeneste han ønsker at benytte. Billetterne får man af den billet-udstedende tjeneste. Da denne selv er en tjeneste benyttes den samme fremgangsmåde som beskrevet i sidste afsnit.

Når et program har brug for en billet, som det ikke har i forvejen, sender det en forespørgsel til den billet-udstedende tjeneste. Forespørgslen består af navnet på den server, der ønskes en billet til, den billet-udstedende billet $T_{c,tgs}$ og en legalisator A_c som beskrevet i sidste afsnit. Se figur 3.



Figur 3: Hvordan får man billetterne ?

Den billet-udstedende server undersøger legalisatoren (A_c) og den billet-udstedende billet ($T_{c,tgs}$) som før. Hvis alt stemmer oprettes der en ny tilfældig session-nøgle som skal benyttes mellem brugeren og den nye server. Dernæst oprettes der en billet til den nye server, bestående af brugerens navn, serverens navn, det nuværende klokkeslet, brugerens IP adresse og den nye session-nøgle. Levetiden for den ny billet sættes til at være det mindste af den resterende levetid for den billet-udstedende billet og standard levetiden for en billet til den pågældende tjeneste.

Den billet-udstedende server sender nu billetten, sammen med session-nøglen og noget yderligere information tilbage til brugeren. Alt dette er kryptograferet med den oprindelige session-nøgle, som er en del af den billet-udstedende billet. På denne måde undgår bruger at skulle indtaste sit password på ny.

Problemer ved Kerberos. Hvor lang levetid skal en billet have ? Det er en afvejning mellem sikkerhed og bekvemmelighed. Jo længere en billets levetid er, jo længere tid er der en risiko for, at den kan blive misbrugt. Dette kunne f.eks. ske hvis brugeren glemmer at logge ud. Jo kortere en billets levetid er, jo oftere er brugeren nødt til at forny den.

Hvordan kan en "legaliseret" bruger lade en server forespørge om andre netværk tjenester på hans vejne ? Et eksempel herpå er hvis en tjeneste kræver tilgang til beskyttede filer hos en fil-server. Et andet eksempel er hvis brugeren fra arbejdsstationen har logget ind på en anden maskine, og derfra ønsker at benytte en netværk tjeneste. Her kunne det være bekvemt at brugerens rettigheder blev overført til den

nye maskine. Det kan man dog ikke gøre, med mindre brugeren har garanti for at den nye maskine er sikker.

Et andet problem handler om at sikre arbejdsstationerne. Hvis det ikke drejer sig ens private arbejdsstation, som man har kontrol over, men en frit tilgængelig arbejdsstation, kan man ikke være sikker på om nogen f.eks. har ændret i `login` programmet.

Yderligere information. Der findes en nyhedsgruppe, som handler om Kerberos, den hedder comp.protocols.kerberos. Kildetekst og dokumentation er tilgængelig via `ftp` fra `athena-dist.mit.edu` i kataloget `pub/kerberos`. Kildeteksten for modulet, der implementerer DES-kryptograferingen er dog ikke tilgængelig.

Konkrete spørgsmål om f.eks. dokumentation eller kildeteksten kan rettes til info-kerberos@athena.mit.edu.

Reklame-karrusel i DKUUG-Nyt

Af *John Plate, plate@infotek.dk*
InfoTek aps

I DKUUG-Nyt nr. 48 skriver Frank Bagge, datakontoret Aps, en artikel, hvori han skamroser UNIX-varianten Esix, som han selv markedsfører. Det gør han som svar på en artikel i nr. 47, hvor Peter Frenning, der sælger SCO UNIX, har kastet smuds på sine konkurrenter, herunder Esix.

Gang i karrusellen

Nu har vi karusellen kørende og der er intet, der kan afhjælpe en redaktørs artikel-hunger, som provokationer i spalterne. Det ligefrem driver læserne til tastaturet. Også mig.

Frank Bagge skriver, at det er for galt, at Peter Frenning omtaler Esix som "en fejlbehæftet UNIX, hvilket absolut ikke er tilfældet". Herefter lover Frank Bagge at "få sat fakta i det korrekte perspektiv", og han får derefter præsenteret sig selv, sit produkt og fremtidsversioner for læseren. Blot mangler priserne og de detaljerede handelsbetingelser!

Sagen er, at begge artikler er ganske mangelfulde. Selv med mit begrænsede kendskab til tingene, er det let at se. Men lad mig holde mig til Esix, som jeg selv har haft kørende siden efteråret.

Fælles frustration

Det fungerer såmænd udmærket, men problemfrit er det ikke. Af den Esix "mailing-liste", jeg har abonneret på længe, fremgår det, at Everex (der står bag Esix) ikke mere svarer på henvendelser om deres SVR3.2 version. For Esix SVR4 gælder, at henvendelser pr. elektronisk post får ganske ringe eller ingen service. Det er ret kendt, at det er en sej proces at få rettet fejl og rettelserne sendt ud. Brugerne er meget afhængige af deres egen "mailing-liste" og det aktive sammenhold mellem brugerne.

I frustation over den dårlige service på Esix, har mange søgt over mod Dell's version af SVR4, som godt nok også har visse problemer,

men langt færre end Esix. Priserne er stort et identiske, nemlig ca. 8.000 kroner for en 2-bruger udviklingsversion ved køb direkte i USA. (Det må/kan man ikke, men det er nu meget let.)

For meget reklame

Jeg mener at indlæg, som de to her omtalte, focuserer entydigt på reklame for egne produkter og for lidt på reel information.

I adskillige tilfælde har DKUUG-Nyt bragt indlæg fra folk med umiskendelige kommercielle interesser i det de skriver om. Husker du ikke artiklen fra en medarbejder hos Dansk Data Elektronik, der fortalte, at DDE altid havde fulgt åbne standarder. Ak ja, firmaets gode samvittighed står i dyb taknemmelighedsgæld til den dårlige hukommelse. Sjældent har noget firma forsøgt at låse kunderne fast i lukkede løsninger som DDE. Men det er da glædeligt, hvis en forandring er indtrådt.

Brug et skrappere filter

Som i mange andre af den type indlæg fristes forfatteren til at nedpriotere det egentlige budskab til fordel for en gang rendyrket reklame, der retteligt burde indrykkes som "informativ annoncering" — mod betaling.

Redaktøren burde gøre reklame-maskerne mindre i det redaktionelle filter, og prioritere de mere saglige indlæg. Som f.eks. artiklen om "Demand Printing", der godt nok fik nævnt firmanavnet, men som var særdeles sober i sit indhold.

Reklame–karrusel i DKUUG–Nyt?

Af *Christian D. Jensen*
DKUUG–Nyt

John Plate rejser i sit indlæg en kritik af den redaktionelle linie i bladet. Denne kritik drejer sig mest om, at alt for reklameprægede artikler får lov at slippe igennem det redaktionelle filter. Som redaktør af bladet mener jeg at indlægget kræver et svar samtidig med, at jeg vil opfordre andre til at give deres meninger til kende.

Kritikken drejer sig blandt andet om det seneste tema–nummer, der drejede sig om "UNIX til hobbybrug". Nummeret forsøgte at belyse mulighederne for at køre UNIX hjemme. Målet var at få beskrevet dels de gratis muligheder, dels de komercielt tilgængelige UNIX varianter, der kører på en billig maskine.

De nævnte artikler er en oversigtsartikel af Peter Frenning fra SCO, som gennemgår markedet for UNIX til PC'er. Personligt mener jeg at det var en sober og god artikel, der gav en introduktion til markedet uden at forherlige SCO.

Bagefter føler Frank Bagge, at det produkt han leverer er blevet uretfærdigt behandlet og han ønsker at gøre skaden god igen. Det kan godt være at ESIX artikelen ikke virker så godt som svar til Peter Frennings artikel. Den ligger til gengæld fuldstændig på linie med et af de oprindelige formål med tema–nummeret, nemlig at belyse det komercielle marked.

Da vi ikke har en fast stab af journalister, kan det være meget svært at få dækket markedet, hvorfor vi går til leverandørerne og beder dem om selv at beskrive deres produkter. Den bedste reklame leverandøren kan få er en god og saglig artikel, der ikke forsøger at love mere end varen kan holde. Det er derfor både i læsernes og leverandørens interesse at der leverers en saglig artikel.

Selvom jeg ikke er enig i den konkrete kritik, vil jeg gerne vedgå at der har været reklameprægede artikler i bladet. At den utilslørede reklamegejl bør nedtones er vi helt enige om, men sålænge der også er et egentligt indhold i artikelen, tror jeg godt at bladets læsere kan læse udenom salgsgassen.

DKUUG
Dansk UNIX-system Bruger Gruppe
Bestillingsliste medlemstilbud

Listen sendes til:

DKUUG
Sekretariatet
Kabbelejevej 27 B
2700 Brønshøj

(tlf 3160 6680; fax 3160 6649)

Afsender:
Medlemsnr.:

att:

Medlems-
navn og
adresse:

Medlemsnavn og adresse tages normalt fra vor database, men bedes angivet her (gerne stempel) af hensyn til kontrol.

Prissatte medlemstilbud (priser EXCL moms).	Antal	Medl.pris	Beløb
UNIX-bogen (dansk udg. af "UNIX - the book")		150,00	
Dansk UNIX markedsoversigt, 4. udgave 1991		70,00	
UniForum products catalog 1990		400,00	
Administrative systemer. Børsen rapport ...		250,00	
Ekspeditionsgebyr, pr. samlet bestilling ..		50,00	
		Ialt, excl moms:	

Overskydende sæt af foredragsnoter fra medlemsmøderne tilsendes mod et ekspeditionsgebyr på 50 kr + moms. Ring og hør, om vi har det ønskede.

Øvrige medlemstilbud, der fremsendes gratis	Antal
Tilmeldingsblanket/rekvisition til:	
- Ekstra abonnement på DKUUG udsendelser (abonnementet er gratis, højst 2 stk pr. organisations-medlem, højst 9 stk pr. stormedlem)	
- Ekstra abonnement på EUUG Newsletter og DKUUG udsendelser (350,- kr/år, kun org.- og stormedlemmer samt studerende)	
- Affilieret medlemsskab af UniForum (200 kr/år) (incl CommUNIXations 4 gange pr år).	
- Fuldt medlemsskab af UniForum (ca. ??,-/??,- USD/år for associeret/generelt medlemsskab)	
- Abonnement på PC World og/eller Computerworld (50 % af normal abonnementspris)	
DKUUG's nye brochure (til PR-formål)	
DKUUG Nyt specialnummer MicroData 91 (også til PR-formål) ...	
DKUUG's medlemsliste	
Medlemsinformation (vedtægt, formandsberetn., regnskab, budget)	
Netinformation (m. tilmeldingsblanket for login/post/nyheder)	
Magnetbåndsinformation (m. bestill.blanket for gratissoftware)	

Dato:

Underskrift:

Forbeholdt DKUUG:
Modt. d.

Eksp. d.

Medltilb 91-10-22

Oversigt over medlemsmøder i 1992

Dato	Sted	Emne
23/04	Helsingør	UNIX i den offentlige sektor
3/06	Odense	Netværk og kommunikation
4/06	Odense	Systemudvikling – 4GL – CASE
18/06 †	København	Multimedia
26/08	Helsingør	UNIX–markedet
24/09 †	København	Administrative systemer
29/10	Odense	Arbejdsstationer, hardware og software
26/11	København	Generalforsamling m.m.

De med † markerede møder er eftermiddagsmøder.

Detaljeret program for hvert enkelt møde vil blive udsendt separat og evt. annonceret i DKUUG-Nyt.