Bruk av Honeynett for å analysere trusler mot datanettverk

Forord

Vi vil gjerne takke Nils Ulltveit-Moe for veiledning og gode råd. Resten av Proseq for hjelp når det var nødvendig. Vi vil også takke IKT- avdelingen ved HiA for å ha gjort det mulig å utføre forsøk ved skolen. Forhåpentligvis vil denne rapporten være interessant lesing, gi innblikk i hvordan sette opp Honeynett og bruk av dette for å skaffe informasjon. Vi håper også at det arbeidet vi har gjort vil kunne hjelpe andre som vil forske videre på bruk av Honeynett.

Frank Aguilar Mortensen Asbjørn Setekleiv

Innholdsfortegnelse

Figurliste

1 Innledning

1.1 Beskrivelse av oppgaven

Denne oppgaven vil ta utgangspunkt i begrepet honeypott[14], nærmere bestemt Honeynett[21], en nettverksvariant av honeypott og gå et skritt videre derfra. Et nett av denne typen vil bli brukt som et verktøy for våre undersøkelser.

Det vi ønsker å se nærmere på er:

- Teknikker på hvordan en kan tiltrekke ønsket trafikk til et slikt nett. Med ønsket menes aktivitet som gir observatørene ny informasjon/kunnskap om eventuelle trusler, fremgangsmåter og angripere.
- Psykologien i hvorfor noen servere/tjenester blir valgt som mål for misbruk, mens andre ikke blir det. Erfaringer herfra vil klart være av interesse for nettverksansvarlige.
- Se på hvilke følger ens egne handlinger og aktivitet på Internett kan være med på å utsette en selv og det nett man er tilknyttet til for potensielle trusler og farer som en kanskje ikke ville blitt utsatt for ellers. Det vi tenker å se på er f.eks. innbrudd hos en selv, internettbrukere med fastnett/bredbånd og misbruk/manipulering av ens egen maskin eller nettverk for videre angrep mot andre, såkalte distribuerte angrep.

Til dette formål skal vi altså sette opp et "narrenettverk", et Honeynett som vi så skal bruke til å monitorere potensielle angripere. Hva et Honeynett er og hvordan vi har satt dette opp vil bli beskrevet i mer detalj senere i rapporten, men helt kort vil det for angriperne se ut som et helt vanlig nett og disse skal ikke være klar over (eller ha mulighet til å finne ut av) at de blir monitorert.

Det vil selvfølgelig også være interessant å se og lære hvordan angripere jobber, selv om dette strengt tatt ikke er hovedformålet med dette studiet.

Oppgaven ble gitt til oss av Proseq[31], et norsk firma med hovedsete i Arendal. Datasikkerhet er hovedområdet deres, og virksomheten spenner vidt innen design av sikkerhetsløsninger og monitorering av nettverk til kunder.

Siden Asbjørn jobbet her sommeren 2001 hadde vi kontakter som resulterte i at denne oppgaven ble tildelt oss. Området datasikkerhet interesserer oss begge og det falt dermed naturlig å ta denne oppgaven.

1.2 Avgrensninger

Det som er viktig å få frem er at denne oppgaven ikke skal fokusere på oppsett og design av Honeynett alene, da dette er gjort før. En stor del av oppgaven må likevel settes av til dette, da et slikt nett er en nødvendighet for å få gjennomført forsøkene.

Med psykologien bak hvorfor noen servere blir valgt fremfor andre mener vi ikke å psykoanalysere angriperne våre da vi ikke har noen form for kommunikasjon mellom oss selv og de som kommer på "besøk". Ingen av oss har noen kompetanse på dette området. Vi skal bare observere endringer i aktivitet inn på vårt nett ettersom vi endrer på tjenester og oppsett av maskinene i nettet. Resultater herfra kan diskuteres da dette vil være basert på passiv observasjon. Vi skal likevel ta noen

diskusjoner om hva slags formål angripere kan ha og hvem de kan være. Dette vil bli basert på hva de gjør med maskinene og liknende dokumenterte tilfeller på nettet.

Med hensyn på skolens sikkerhet og ikke minst for å holde Honeynettet hemmelig vil vi utføre eksperimentene i en viss rekkefølge og stigende kompleksitet. Kort oversikt:

- Til å begynne med vil vi gå forsiktig ut og bare observere hva som skjer på nettet.
- IP-er som angriper oss vil ikke bli kontaktet. Selv ikke om de kommer fra kjente bedrifter/organisasjoner.
- Ved aktiv bruk ut på Internett vil vi ikke prøve å hisse opp eventuelle angripere, da dette kan føre til angrep på skolens nett.
- Kontakt med potensielle angripere vil begrense seg til eventuell normal oppførsel på Internett. Ikke noe vil bli nevnt om nettet vårt. Normal oppførsel er bruk av applikasjoner og tjenester vi installerer uten å forsøke å la noen få vite at vi kjører et Honeynett.

2 Oppsummering

Denne oppgaven er et studie i hvordan bruke et Honeynett som et verktøy for å observere og analysere de trusler som dag finnes på Internett. Det ble først foretatt et litteraturstudium, både teknisk om Honeynett-teknologi og mer teoretisk om de aktuelle trusler som er aktive på dagens Internett. Design av Honeynettet ble utført i samråd med både Proseq og HiA[12], der skolen hadde det endelige ord da deres nettverksressurser ble benyttet.

Det ble utført fem forsøk som omhandlet både passiv og aktiv lokking. Resultatene av disse forsøkene viste indikasjoner på at aktiv lokking ved normal bruk av nettet, ikke nødvendigvis er mer effektiv enn ved passiv lokking. Det viste seg at oppsett av standardtjenester som FTP og HTTP var effektivt som tiltrekningmetoder. Ved tre av forsøkene ble maskinene tatt grunnet feil ved FTP-tjenesten.

Det ser også ut til at det finnes et stort antall angripere på Internett. Det kan virke som om at de fleste angrep på oss ikke var selektive, eller målrettet, men angrep blindt etter tjenester som lett kan utnyttes. Dette vil selvsagt være spekulasjoner, men ut ifra den tidsperioden vi hadde til rådighet var dette det vi kom fram til. Angriperne våre hadde forskjellige formål med å ta over maskinene, men det viste seg også at i store trekk at formålet var å bruke disse i videre angrep. En mer omfattendes analyse ble ikke foretatt grunnet begrensinger i tid og de begrensinger vi fikk fra HiA.

3 Datasikkerhet

Datasikkerhet er et vidt begrep som lenge har vært aktuelt, men med innføring av Internett som en viktig kommunikasjonskanal for en stor del av verdens befolkning har dette teamet blitt mer og mer aktuelt for alle som tar sikkerhet for seg selv og andre seriøst.

3.1 Generelt

Datasikkerhet generelt går mye på at uvedkommende ikke skal komme inn på maskinen og stjele eller ødelegge informasjon. Dette er ikke ment å gjøres fysisk, men gjennom inntrenging fra andre maskiner som er koblet på Internett. Man kan tenke seg følgende scenario som kan oppstå:

- Adgang til maskin ved hjelp av stjålet brukernavn/passord.
- Adgang ved hjelp av gjetting av passord.
- Bruk av feil i programvare for å skaffe seg tilgang, for eksempel ved bruk av ”buffer overflow”.
- Trojanere
- Feilkonfigurering av programvare.
- Spyware[38]

Det kan også tenkes andre mulige måter å komme seg inn i maskiner på, men disse som nå er nevnt er måter som er mest brukt og dermed mest kjent. Målet er å forhindre at uvedkommende får tilgang.

3.1.1 Passord

Adgang til en privat arbeidsmaskin eller server er aller enklest å oppnå om man kan passordet. Det optimale er en administrator-/root-konto. Da har man full tilgang til maskinen og kan utføre alt av forandringer man ønsker. Man kan legge til brukere, endre på konfigurasjonsfiler, legge inn bakdører og legge til nye brukere. Total kontroll er oppnådd.

Selvsagt kan man ikke forhindre adgang til maskiner 100% ved å benytte sikre passord, men det er mye som kan gjøres. Det viktigste er å ikke benytte seg av enkle passord. Eksempel på et dårlig passord er navnet til noen man kjenner, og ord som står i ordlister. Eksempel på et godt passord er et ord som inneholder mange tall og bokstaver sammensatt. Bytting av tall isteden for bokstaver er et gammel og greit triks. Helst bør man velge en passordsetning som ikke kan gjettes (gjerne over 20 tegn). Problemet for folk flest er at passordet er noe man må skrive inn ofte og da velger man naturlig et lett passord.

Dersom det er bedrifter eller skoler som har mange brukere bør man kreve at brukerne benytter seg av lange passord. Dette kan ofte styres fra administrativ hold, så brukerne selv ikke har noen mulighet for å overstyre dette.

Stjeling av passord kan forekomme, men det er enkle metoder man kan benytte for å gjøre det vanskeligere for dette å oppstå. Det viktigste er at passordet aldri skrives ned, men huskes. Det bør også være et krav at passordet byttes jevnlig og da i et annet. Et triks som brukes er å skrive inn samme passord om igjen. Dette må ikke være mulig.

Holder man seg til disse reglene som er beskrevet så har man oppnådd en hel del i forhold til å høyne datasikkerheten.

3.1.2 Feil i programvare

Microsoft[26] Internet Information Server (IIS) er en web server som er mye brukt i både kommersielt og privat henseende. Den er eksempel på programvare som er veldig nyttig å bruke exploits på for å oppnå kontroll på serveren den kjører. Siden den blir benyttet på mange og store servere vil den typisk være veldig utsatt for angrep. De angrep som er mest kjent for IIS er såkalte Unicode-angrep[25].

Det vil si at man sender spesielle tegn til webserveren slik at man får tilgang til steder på maskinen der man egentlig ikke skal ha det.

Eks: http://X.X.X.X/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\ Man oppnår her tilgang til filer man ikke skal ha tilgang til, f.eks. cmd.exe. Da kan man kjøre systemkommandoer.

En annen måte er å bruke "buffer overflow". Dette er feil fra programmerers side. Har man datatypen string i et program og setter av 25 tegn, hva vil da skje når man sender 400 tegn? Man kan da få overflow av variabelen. De overflødige tegn kan inneholde ondsinnet kode som kan brukes for å oppnå tilgang til maskinen. Å beskytte seg mot dette er vanskelig dersom man ikke lager programvaren selv og benytter rammeverk som er designet med tanke på sikkerhet. Det er derfor viktig å hele tiden være oppdatert på hva som skjer på programvarefronten, dvs. ha den nyeste versjonen av programmet man bruker. Dette vil ikke beskytte deg 100 %, men vil hjelpe betydelig.

3.1.3 Trojanere

For noen år siden var det et fjernadministreringsprogram som ble laget av en hacker[11]gruppe som kaller seg "Cult of the dead cow"[8]. Programmet heter Netbus[28]. Dette er et program som legger seg inn på en Windowsmaskin, temmelig skjult som en serverapplikasjon. Har man klient versjonen kan man logge seg på og ta kontroll over maskinen som kjører server versjonen. Det finnes lignende programmer for fjernadministrering, f.eks. PCanywhere[39], men det som var spesielt med Netbus var at det som oftest ble det spredt gjennom exe-filer som gav seg ut for å være noe annet enn selve programmet.

Brukerne merket ikke noe om man trykket på fila. Ingenting skjedde, men Netbus la seg inn og startet hver gang maskinen ble startet på nytt. Dette var en såkalt trojaner. Programmet gav seg ut for å være noe annet enn det egentlig var.

Et betydelig problem i dag er at brukere som laster ned programmer fra Internett, ofte er svært ukritiske til hva de installerer. Er man ikke forsiktig så oppnår man å få trojanere på maskinen sin. Nimda[29] viruset som herjet rundt vinteren 2001 var en trojaner som man kunne få dersom man kjørte en versjon av IIS som ikke hadde fått installert en utbedring (det fantes også andre måter å bli infisert på). Man risikerte blant annet at maskinen begynte å sende e-post til kontakter i adresseboka til Outlook-programmene med vedlegg som man ikke var interessert i å videreformidle. Dette kunne være jobbrelaterte dokumenter eller annet personlig. Utbedring for å stoppe Nimda kom raskt ut, men enda kan man finne mange maskiner som ikke har gjort noe med dette problemet. De er fremdeles infisert og fortsetter å infisere andre maskiner på lokalnett og Internett.

Det er generelt vanskelig å forhindre feil i programvare, men hadde man benyttet nyeste utbedring på IIS så kunne ikke Nimda gjort noen skade. Benytter man antivirusprogrammer så vil disse som oftest oppdage og fjerne problemet. Dette forutsetter selvsagt at man benytter seg av de nyeste virus definisjonene og at virusene er kjente.

3.1.4 Feilkonfigurering av programvare

Ved innføring av ADSL og andre bredbåndtjenester har bruken av Internett økt betraktelig. Man laster ned programmer som WinMX[47] og Bearshare[6] som deler harddisken for andre brukere på nettet. Dette gjøres for å få tilgang til andre sine fildelinger, akkurat som et stort lokalnettverk.

Sikkerhetsrisikoen her er enorm. Det finnes feil som er oppdaget i flere av disse programmene hvor

brukere har fått lov til å kjøre ondsinnet kode på maskinen til den uheldige, eller hente ut filer fra kataloger som ikke var ment å være delt. Ofte deler uerfarne brukere hele harddisken sin med personlige dokumenter og konfigurasjonsfiler. Det behøver ikke å være hull i selve programvaren som fører til angrep på brukeren, men rett og slett at man avgir for mye informasjon om seg selv og datamaskinen man bruker.

Fildelingsprogrammer er enkelt å bruke dersom man vil ha tilgang til filmer, mp3 osv, men sikkerhetsrisikoen kan som nevnt være høy. Programmene bør derfor optimalt kjøres på en egen maskin som er dedikert til jobben, og som ikke inneholder noen kritiske filer. Eller ikke kjøres i hele tatt.

Andre programmer som har er kjent for å ha åpne hull i programvaren er ”Instant Messaging” tjenester som ICQ[15], Microsoft Messenger[24] og AOL sin Messenger[5]. De avgir veldig mye informasjon om deg selv, og er derfor en sikkerhetsrisiko i seg selv. All kommunikasjon skjer fra klient til klient, selv om en del lagres sentralt på server. Igjen er det lureste å kjøre slike programmer på en ”sikker” maskin, men dette er igjen veldig lite brukervennlig.

Det finnes veldig mange eksempler på åpne hull i programvare, dette er bare noen eksempler, men måten programmene har feil eller sikkerhetshull på kan ofte likne mye på hverandre.

3.1.5 Spyware

Shareware[35] programmer er mye brukt i databransjen for at brukeren kan teste ut programmet før man kjøper det. I realiteten laster mange ned en crack eller en serial til programmet på Internett og unngår dermed å kjøpe det. Dette førte til at flere og flere produsenter gikk bort fra shareware prinsippet og heller introduserte Adware[3]. Dette betyr at man viser reklame i programmet som brukeren benytter. Problemet er at det ikke lenger er så lønnsomt å ha reklame banner på Internett som det var før, derfor innførte mange produsenter gjemt funksjonalitet som følger med programmene sine, men skjult for brukeren. Dette er såkalte Spyware-programmer. Ikke alle bruker det, men det er blitt vanlig.

Det disse programmene gjør er å sende informasjon tilbake til enten produsent eller en tredjepart. Denne informasjonen kan bli solgt videre til selskaper som er interessert i hvilke web sider du besøker, hvilke mp3 sanger du hører på osv. Alle er enige om at dette er et grovt brudd på datasikkerheten og personvernet. Man har ofte 3 valg. Enten kjøpe programmet, det skal da ikke fortsette å logge, selv om det ofte skjer likevel. Annet valg er å ikke bruke programmet. Det tredje er å fjerne spywaren. Det kan skje at programmet ikke vil virke lenger etter at spywaren er fjernet, men det må man teste ut på hvert enkelt program. Det finnes program som er gratis på Internett, eksempel Ad-aware[2] fra Lavasoft, som fjerner spyware, eller man kan fjerne dem selv dersom man har kunnskap nok. Eksempel på program som inneholder spyware er Kazaa[19].

3.2 Honeypott

Kort sagt er en honeypott et verktøy som benyttes til å analysere de trusler et datasystem kan være utsatt for samt til å analysere innbrudd i disse. Formålet med å implementere en honeypott i et datasystem er å lettere lære av feil og mangler ved ens egen datasikkerhet ved å kunne se hvordan utenforstående jobber under et innbrudd, hvilke teknikker og verktøy disse benytter og hva formålet med innbruddet kan være. Denne typen informasjon kan så senere brukes til enten å spore opp angriperen (selv om dette ofte kan vise seg å være bortkastet tid i mange tilfeller) eller å rette opp mangler og feil i sikkerhetssystemet slik at lignende angrep blir vanskeliggjort i fremtiden. I tillegg

vil en honeypott i et datasystem være med på å raskere gi advarsel om at et angrep er i ferd med å skje og på denne måten gi ansvarlige mer tid til å reagere. En honeypott kan også brukes til å avlede angrep fra viktige deler av et nett som ellers ville gått mot sentrale datasystemer. En ting som er viktig å merke seg er at en honeypott ikke erstatter andre sikkerhetssystemer, men er bare et tillegg.

Måten en honeypott fungerer på er å tilby en rekke tjenester til omverdenen (gjerne falske sådan) for så å overvåke disse for hendelser og aktivitet. Den kan hvis ønskelig registrere og loggføre alt fra nettverksaktivitet til kommandolinjeaktivitet. En honeypott vil kunne være alt fra en tjeneste som kjører på en server til en hel maskin satt av til dette formålet. For å forsikre seg om at det en honeypott fanger inn av data ikke blir kompromittert, er det svært viktig at loggfiler ikke blir liggende på selve honeypotten.

Loggfilene skal senere undersøkes og brukes til å analysere angrepet i ettertid. Kan vi ikke stole på loggene så faller en stor verdi av en honeypott bort. Det som kjennetegner en honeypott er at eventuelle besøkende aldri bør være i stand til å merke at de blir holdt under oppsikt. Hvis disse mot formodning finner ut at de er inne på en slik en vil denne da miste mye av sin verdi og kan bli forbigått ved senere anledninger.

Når en først velger å sette opp en honeypott i sitt system er det en rekke viktige punkter en må ha i bakhodet. Man er selv ansvarlig for de skader og angrep som kan genereres fra honeypotten når fremmede får kontroll over denne. Derfor er det svært viktig at man har et opplegg som sørger for at man ikke risikerer at egne ressurser kan bli benyttet til angrep mot andre (relay, DOS). Noe som er med på å vanskeliggjøre dette, er selvfølgelig det faktum at man må implementere dette samtidig som man ikke må røpe for de ”besøkende” at maskinen er en honeypott. En vanlig måte å gjøre dette på er å benytte seg av brannmurløsninger rettet inn mot trafikk som kommer fra honeypotten. Disse brannmurene vil ofte være konfigurert slik at de vil slippe noe trafikk ut, men avhengig av regelsett vil de stoppe forbindelser etter en stund. Eksempel her vil være å godta en FTP-forbindelse fra honeypotten, slik at inntrengere kan hente ned de verktøy som de trenger for videre misbruk, men stenge ved portskanning.

Som tidligere nevnt trenger man ikke kjøre en helt separat maskin som en honeypott. Man kan også simulere dette. Det finnes verktøy for dette på markedet. De to løsningene som er valgt som eksempel er Deception Toolkit[41] og Specter[37]. Felles for dem begge er at de baserer seg på å lure angripere og føre disse ned blindgater.

Deception Toolkit er et enkelt program en kjører på maskinen en vil skal fungere som en honeypott. Programmet vil så simulere en rekke virkelige tjenester og kjente svakheter på systemet. Disse tjenestene vil virke reelle utenfra, men er i virkeligheten bare et spill og virker egentlig ikke. Programmet kjenner igjen forespørsler og vil svare på disse, selv om det den gir tilbake ikke alltid er like forståelig (tull med andre ord). Slik vil en angriper kaste bort mye tid og ressurser i den tro at angrepet fungerer bare for å finne ut senere at det hele bare var tøv. Et slikt verktøy vil gi den som blir angrepet tid til å sikre sitt system ytterligere, samt få en oversikt over hvordan forsøkene på å utnytte tjenestene fungerer. På denne måten lærer man angriperens teknikker.

Et delmål til de som lager Deception Toolkit er at programmet vil virke avskrekkende ved å merke maskiner slik at det er mulig å se at denne honeypott-løsningen kjører på disse. Dette gjøres ved at de har reservert, eller gjort kjent port 365. Dette vil kunne fungere som en avskrekker da angripere etter hvert vil sjekke om denne porten eksisterer før de setter i gang et omstendig angrep, spesielt hvis mange nok kjører opp Deception Toolkit, slik at dette blir et kjent fenomen.

Specter er en lignende, men litt større løsning. Det er fremdeles ett program som kjøres, men her simuleres en hel maskin. Slik kan man skape et mye mer attraktivt mål blant mer viktige maskiner i systemet og på denne avlede angrep som ellers ville kunne få store konsekvenser. Det vil bare være den virtuelle maskinen som er utsatt for angrep, vertsmaskinen vil ikke være tilgjengelig utenfra. Specter gir mulighet til å simulere 13 forskjellige operativsystem. På den virtuelle maskinen har man valget mellom 14 forskjellige tjenester/feller. Man har også mulighet til å velge mellom forskjellige adferdsinnstillinger som angir hvordan Specter vil oppføre seg mot en angriper.

Her har man fem forskjellige innstillinger:

- Åpen - I denne modusen vil Honeypotten se ut som en dårlig konfigurert server. Denne modusen kan være nyttig for å kartlegge trusler systemet kan være utsatt for, i og med at mange kan bli tiltrukket av et ”lett bytte”.
- Sikker - Dette vil få systemet til å fremstå som en fornuftig satt opp server. En person som setter i gang et angrep her, vil enten ikke vite hva han/hun gjør eller være en som er ute etter en verdig utfordring.
- Sviktende - Her vil man få inntrykk av at maskinen lider av en rekke hardware og softwareproblemer. Dette kan føre til at angriper blir nysgjerrig og kan bli værende lenge nok til at sporing og varsling kan bli gjort.
- Rar - Denne modusen er ment for å forvirre angriperen og få denne til å kaste bort tid på honeypotten på liknende måte som ved Deception Toolkit. Her vil Specter gjøre at maskinen svarer ulogisk og oppfører seg såpass uforutsigbart at en angriper vil bli sittende og lure på hva som egentlig skjer.
- Agressiv - Angriper vil holdes opptatt og uinteressert helt til Specter har samlet opp nok informasjon om angrepet og angriper (ikke kan skaffe mer informasjon). Når dette er gjort vil Specter så presentere seg for angriperen og kutte forbindelsen. Dette er ment å virke avskrekkende og forhindre angrep fra angriperen i fremtiden.

Specter er et automatisert overvåkningssystem. Når systemet først er satt opp, vil det selv sørge å alarmere hvis innbrudd oppdages samt sette i gang loggføring av hendelsene. I tillegg vil systemet også forsøke å finne ut mest mulig informasjon om hvor og fra hvem angrepet kommer fra. Til dette benyttes en rekke funksjoner som for eksempel finger, portscan, traceroute og dns (bare for å nevne noen).

3.3 Honeynett

Vi skal her beskrive den type honeypott vi har benyttet i vår oppgave. Det som er spesielt med Honeynett er at her består honeypotten av et fungerende nett og ikke bare en maskin.

I 3.2 har vi gått gjennom hva en honeypott er og hvordan man kan sette opp en. Enten som selvstendig maskin eller et verktøy. Honeynett er en honeypott variant som består av mange maskiner. Vanligvis har man også en dedikert loggserver og et IDS[17] system i tillegg.

Målene med å sette opp et Honeynett er de samme som ved bruk av en honeypott, men ved å benytte flere maskiner kan man lettere simulere et større miljø. Dette kan enten være en typisk server park, eller et vanlig hjemmenett. I vårt forsøk simulerer vi en forskningslab med arbeidsstasjoner og noen servere. I ett Honeynett bruker vi ikke verktøy som Specter, men vi

benytter oss bare av normale produksjonssystemer (dette er ekte maskiner). Systemene endres ikke for å gjøre dem mindre sårbare. Specter og liknende system er begrenset til ferdigdefinerte tjenester og tillater ikke simulering av et ekte produksjonsmiljø med ekte tjenester og svakheter.

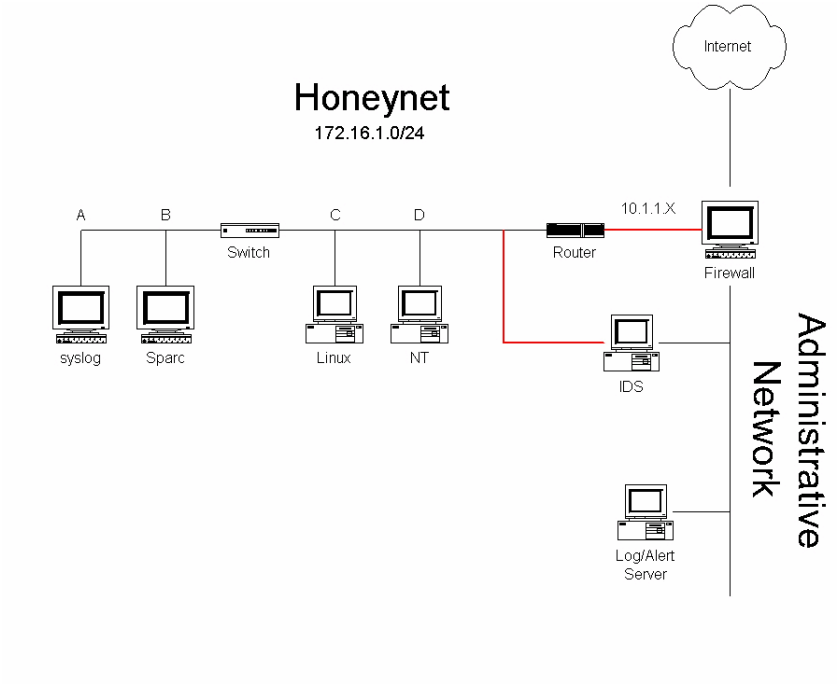
Hvordan man designer et Honeynett er opp til hver enkelt, men vi har benyttet oss av retningslinjer som "Project Honeynet"[30] har gitt ut. Dette er personer som har gjort forsøk med dette i lang tid og har kommet fram til løsninger som fungerer meget bra. Selvsagt må alle ta hensyn til hvordan utstyr man har tilgjengelig og gjøre det beste ut i fra dette. Men dette er retningslinjer som kan være lurt å følge hovedgangen i. Under følger en kort versjon av hvordan "Project Honeynet" anbefaler oppsett av Honeynett, og hvordan man skal bruke det.

I et Honeynett er meningen å lokke angripere inn, men disse må ikke få muligheten til å bruke nettet til videre angrep. Målet er å lære hva en angriper gjør, og hvordan han/hun utfører angrepet. Derfor må man ha en brannmur. Denne skal typisk tillate alle forbindelser inn, men antall forbindelser ut skal reguleres. Kommer det for mange forbindelser ut på et kort tidsrom skal alle stenges ned. Mellom brannmuren og Honeynettet er det dessuten lurt å plassere en router for å skape et mer naturlig miljø, og dessuten hindre at man kan se brannmuren.

Sitat fra Project Honeynet: "Additionally, a router is placed between the firewall and the Honeynet. This is done for two reasons. One, the router hides the firewall. When a honeypot is compromised, blackhats will find a production router between them and outside networks. This creates a more realistic environment and obscures the firewall from being discovered. The second purpose of the router is to act as second access control device. The router can supplement the firewall, ensuring compromised honeypots are not used to attack systems outside the Honeynet."

Her ser man ett typisk oppsett som er hentet fra "Project Honeynet".

Enhet	Formål
Firewall:	Slippe trafikk inn, men begrense forbindelser ut.
IDS:	Logge alt som skjer på nettet.
Log/Alert Server:	Loggmaskin som brukes til lagring av logger, feks. NT logger over SMB.
Syslog:	Loggserver som lagrer lokale logger sentralt ved bruk av syslog-daemon.
Sparc:	Honeynett maskin
Linux:	Honeynett maskin
NT:	Honeynett maskin
Svitsj:	Vanlig svitsj
Router:	Brukes som supplement til brannmur og hindrer brannmur i å bli oppdaget.



Figur 1 – Oversikt: Et mulig Honeynett

Det er tre ting som er viktig når man kjører et Honeynett. Det er kontroll med data, datafangst og datasamling. Kontroll med data er kanskje det viktigste. De som angriper maskinene er folk som ikke skyr noen midler for å få utført det de skal gjøre. Enten det er å bruke maskinene til å ta over andre maskiner, eller å bruke dem i et DDOS angrep. Uansett er det veldig viktig å sørge for at ingen av maskinene våre blir brukt til ulovlige aktiviteter. Dette kan føre til at man kan bli ansvarlig for aktivitetene til angriperen. Ved å bruke en IDS-system, typisk SNORT[36] som er gratis og er forholdsvis enkel å sette opp, og en brannmur, oppnår man mulighet for å stoppe angripere fra å gjøre skade fra våre egne maskiner og ut på Internett. Dette er selvsagt ingen garanti og krever kontinuerlig oppfølging.

Datafangst innebærer at man må vite hvordan angriperen kom inn, når han kom inn og hva han/hun benyttet eller benytter de kompromitterte maskinene til. Brannmuren[32] skal typisk logge alle forbindelser ut og inn fra Honeynettet. Her kan man se senders IP, mottakers IP, porter, og diverse flagg som er satt. Brannmuren kan settes opp til å gi beskjed til oss dersom det kommer forbindelser inn. Det vi må huske på er at alle innkommende forbindelser er mistenkelige siden dette nettet helst ikke skal brukes til andre ting enn å bare stå i ro. IDS-systemet brukes for å logge all aktivitet som skjer på nettet. Det skal også kunne gi alarmer dersom et system blir kompromittert basert på kjente signaturer og hendelser som det kjenner til. Eksempel på dette er en IRC forbindelse fra en av maskinene på Honeynettet. Siden IDS-systemet tar og logger alt som skjer er det viktig at den er veldig godt sikret. Man må kunne stole på dataene som lagres her.

Datafangst involverer også logging av ting som utføres på selve maskinene dersom de blir tatt. Her kan ikke IDS-systemet lytte. Vi trenger noe som kan logge tastetrykk og prosesser som kjøres på maskinen. Det som er vanlig er å benytte en modifisert versjon av bash (shell) som logger til en fjernlogg-server. Dersom maskinen blir tatt og loggingen blir skrevet over har vi i alle fall logger fram til det skjedde på en sikker maskin. Disse hadde gått tapt uten fjernlogging.

Datasamling betyr måter å lagre dataene på. Under er en oversikt over hvordan ”Project Honeynet”

anbefaler å lagre data.

```
Snort Alerts,      Full     -> MySQL database - real time
Snort Alerts,      Full     -> ASCII text     - daily
Snort Alerts,      Fast     -> ASCII text     - daily
Snort binary log,  Full     -> snort.log      - daily
Brannmur logs,     Full     -> ASCII text     - daily
Brannmur logs,     Unique   -> ASCII text     - daily
```

Dette var en kort forklaring av hvordan et Honeynett kan settes opp.

3.4 Virtuelle Honeynett

Virtuelle Honeynett er nesten det samme som vanlige Honeynett. Forskjellen ligger i at man kjører ofte hele nettet i software på en kraftig maskin i motsetning til å ha et ekte, fungerende nettverk. Det er dette vi benytter oss av i diplomoppgaven. VMware[45,13] er programmet vi har valgt å ta i bruk, selv om det er andre programmer på markedet også. Eksempel er User mode Linux[44]. Vmware finnes i Linux og Windows versjon. Vi valgte å bruke Windows-versjonen av VMware, da vi hadde en Windows 2000 server tilgjenglig fra Proseq og vi fant det enklest å bruke det ferdige oppsettet, og sette nettet opp på denne. Dessuten hadde vi testet ut VMware i tidligere versjoner, og vi likte hva programmet kan gjøre, hele konseptet med virtuelle maskiner (som oppfattes som ekte av systemet) som en så på disse kan installere fritt de operativsystemer en vil.

3.4.1 Fordeler

For å sette opp et ekte Honeynett uten bruk av programmer som VMware trenger man like mye hardware som antall maskiner. I tillegg kommer da selvfølgelig annet nettverksutstyr som kabler til de enkelte maskinene, samt de nødvendige svitsjer/hubber. Her har virtuelle Honeynett den fordelen at man i teorien bare trenger en maskin (dog en kraftig en), i tillegg til at man har muligheten til å sette opp virtuelle nett innad i VMware-maskinen mellom de virtuelle maskinene, noe som da kutter ned på en del tilleggsutstyr. Dette gjør et virtuelt Honeynett både plassbesparende og oversiktlig, i tillegg vil en også ende opp med et mer mobilt system, da en vil ha det meste i en boks (som en så kan flytte rundt etter behov).

3.4.2 Ulemper

Bruk av virtuelle maskiner vil ha en del ulemper, noe som da også vil gjelde for bruk av virtuelle Honeynett. Det vi tenker på da er hardwarebegrensninger; tilgjengelig prosessorkapasitet, minne og lagringsplass. Dette vil da selvfølgelig kunne påvirke de valg en til enhver tid vil måtte ta med tanke på de tjenester og operativsystem man ønsker å kjøre. I tillegg vil man etter hvert kunne støte på problemer med hensyn på vertsmaskinens minne og lagringsplass, noe som helt klart vil være med på å begrense antallet virtuelle maskiner en til en hver tid vil kunne klare å kjøre opp gjennom VMware. Koblet opp mot dette har man også problemstillingen hastighet versus antall noder i det virtuelle nettet. Vertsmaskinen vil til enhver tid være flaskehalsen i systemet og man må hele tiden veie opp kostnadene rundt vertsmaskinen opp mot det man ville ha brukt på et tilsvarende fysisk nett. Til slutt kan det også nevnes at med en vertsmaskin som en sentral del av nettet, vil man ha en viss sårbarhet i det at man er prisgitt at denne funger tilfredsstillende og ikke faller fra, da dette vil ødelegge nettet (alt går ned med vertsmaskinen).

3.4.3 Konklusjon

Vi har her diskutert en del fordeler og svakheter knyttet til bruk av virtuelle Honeynett. Vi har også gjort rede for vårt valg av VMware som plattform for vår bruk av denne type teknologi i denne oppgaven. Slik vi ser det, spesielt med tanke på de ressurser vi har til rådighet, vil bruken av VMware til det formål vi har valgt være det mest hensiktsmessige.

3.5 Angripere - Hvem – Hva – Hvordan

I denne delen skal vi presentere en del metoder for tiltrekning av trafikk mot et Honeynett. Både der Honeynettet brukes til analyse og der det brukes som del av datasikkerheten som avledning fra kritiske systemer. Disse metodene er på ingen måte oppskrevne sannheter, men antagelser tatt ut i fra tidligere erfaringer til andre som har drevet med Honeynett (”Project Honeynet”) og egne etter lang tids bruk av Internett.

Før vi kan se nærmere på måter å tiltrekke folk til honeynettet vårt, bør vi kanskje se litt på hva slags personer en kan tenke seg vil ha interesse av å komme innom nettet vårt. Angripere, la oss kalle dem det, kan ikke på noen måte sies å være en homogen gruppe. Her finner en alle typer mennesker, alle med forskjellige motiver. Likevel er det visse grupperinger som er store nok til at vi kan differensiere litt og gjøre det mulig å snakke om typer av angripere. Alle vil kanskje ikke være enige i følgende beskrivelser, men dette er altså vår personlige oppfatning av grupperingene:

Først har vi de såkalte scriptkiddies. Dette er ofte unge mennesker som bruker fritiden sin til å leke med skript lagd av andre enn dem selv, for å ta over servere rundt om på Internett. Disse skriptene er skrevet spesielt mot kjente svakheter og vil ofte automatisk gi root-aksess på maskiner de rettes mot for brukeren. Scriptkiddies har ikke nødvendigvis den kunnskap som trengs for å forstå hvordan disse programmene virker selv, og har ofte heller ikke noen spesiell grunn for å sette i gang et angrep, annet enn at de synes det er moro. Ofte bryr de seg heller ikke om hvilket operativsystem eller tjeneste serveren de angriper kjører. Mye tyder på at dette er den største gruppen ute på nettet (og den mest aktive).

Deretter finner vi den klassiske hackeren. Dette er personer som sitter inne med mye kunnskap og best kan beskrives som entusiaster innen emnet. En ekte hacker har egentlig ingen mørke motiver for å angripe noen. En hacker er ute etter kunnskap og vil ikke ødelegge for andre. En hacker leter etter svakheter ved (helst nye) programmer, tjenester og servere. Hvis svakheter blir funnet, vil en hacker vanligvis ta kontakt med de ansvarlige og opplyse om den oppdagede svakheten. Ironisk nok er det ofte hackere som står bak de verktøyene som scriptkiddies benytter. Disse blir laget som dokumentasjon og i test øyemed, og er egentlig ikke ment for å hjelpe folk til å angripe andre, men som motivasjon for å utbedre mangler og for å lære.

Til slutt kommer vi til den farligste grupperingen, crackerne. Disse individene er minst like kunnskapsrike som hackerne (og begynte kanskje slik), men har helt andre motiver for å gjøre det de gjør. Dette kan f.eks. være at de er ute etter å straffe noen hvis de føler seg urettferdig behandlet av offeret, eller de er ute etter personlig vinning. Disse er ofte ute etter å ødelegge eller stjele, og i noen tilfeller endre informasjon i henhold til egne motiver. Angrep fra en cracker er ofte svært målrettet og godt gjennomtenkt. Hvis en cracker angriper deg, er det nesten garantert en grunn for dette. Noen ganger vil slike crackere også operere på oppdrag fra en tredjepart, da oftest i vinningssaker f.eks. stjeling av informasjon ved industrispionasje.

I tillegg må det nevnes at vi også har automatiserte angrep som stammer fra virus / ormetrafikk. Disse angrepene minner mye om angrep fra scriptkiddies og bruker faktisk også mye av de samme

teknikkene som disse tar i bruk. Disse angrepene er heller ikke diskriminerende og angriper hvem som helst, når som helst. Heldigvis kan slike standardiserte angrep ofte være lette å skille ut, slik at de kan oppdages og blokkeres av egnet programvare.

3.5.1 Hvordan tiltrekkes disse individene inn mot nettet vårt?

Under skisserer vi noen metoder vi mener kan være nyttige for å gjøre nettet vårt mer interessant for uvedkommende. Før vi kommer med forslag på slike metoder, er det kanskje nyttig å se litt på hva disse menneskene ser etter før de setter i gang et angrep. Her er noen mulige årsaker:

- Linjekapasitet – Dette er kanskje først og fremst noe en scriptkiddie har interesse av. Det å ta over et punkt i nettet med bra linjekapasitet har sine fordeler hvis en for eksempel er ute etter å bruke dette videre i et DDOS angrep, eller benytte den aktuelle serveren som et relay-punkt. Stikkord her vil muligens være å benytte en slik ressurs som en tredjepart i andre operasjoner. Det er også svært nyttig å ha en rask forbindelse når man kjører IRC boter. Dette for å forhindre at man selv blir slengt av nettet pga DOS angrep.

- Feilkonfigurering eller mangler – En server som ligger åpen på denne måten, for eksempel ved å benytte tjenester som det allerede er funnet mangler ved (er kjente i miljøet) vil være et lett bytte. Angripere er vel som folk flest og tar veien med minst motstand. Igjen vil scriptkiddies være å finne på grunn av de verktøy de bruker, men her kan også en snill hacker kunne komme innom for å si ifra om manglene.

- Viktighet på serveren (f. eks servernavn/domene) – Hvis en viktig server er tilgjengelig (og dette oppfattes av angriperne), kan den før eller senere bli angrepet. Her vil ikke hvor sikker serveren er satt opp være det avgjørende når det gjelder interessen for å komme inn, men mer ønsket om enten å ødelegge operasjonen til serveren eller endre data på denne til egen vinning (f. eks. økonomisk).

- Personlige motiv – Her menes at angrepet ikke skyldes at målet direkte virker interessant, men at det ligger en grunn bak hos angriper ved valg av mål. Dette kan for eksempel skyldes at vedkommende gjør dette for å straffe noen eller utøve makt. Herunder kommer også situasjoner der en innleid angriper er med i bildet. Stikkord her er at angrepet ikke direkte skyldes en handling gjort av offeret, men heller en oppfatning angriper gjør (av offeret) som setter i gang angrepet. Et eksempel her kan være høyreradikale websider som blir angrepet av venstreradikale.

Videre i oppgaven skal vi se på hvordan vi kan bruke disse momentene for å tiltrekke denne type trafikk som er beskrevet inn mot vårt Honeynett. Gjennom senere forsøk vil vi, forhåpentligvis, kunne få indikasjoner på om disse metodene vi nå nevner faktisk virker, eller om det er andre ting vi heller burde ha sett nærmere på.

3.5.2 Tiltrekningsmetoder

Når det gjelder tiltrekning av trafikk mot nettet vårt har vi to fremgangsmåter å velge mellom: Passiv og Aktiv. Med passiv menes det at vi kort sagt sitter og venter på å bli oppdaget, mens aktiv betyr det motsatte, at vi selv går ut og reklamerer, eller lokker til oss trafikk.

Noen forslag til en passiv fremgangsmåte:

- Tjenester – Her installerer vi tjenester, eller operativsystem vi ofte vet det finnes svakheter ved, samt tjenester som er interessante da disse er nye. Hvis noen skulle finne oss mens vi har et slikt oppsett burde de også kunne se manglene ved vårt system og bli lokket til å sette i gang et angrep. Denne måten er kanskje også interessant å benytte for å simulere hvordan maskiner settes opp i områder uten spesiell kompetanse, som for eksempel hjemme hos folk. Her må det kanskje passes på at en tenker igjennom hva som er naturlig for serveren å kjøre, slik at en ikke vekker mistanke hos en potensiell angriper ved å kjøre svært gamle tjenester på en ellers sikker og oppdatert maskin. Man må muligens også passe på å ha tjenester som det virker fornuftig å ha på samme maskin.

- Konfigurasjon – Dette kan kanskje sies å være en variant, eller utvidelse av forrige punkt men her ser vi for oss at vi med vilje konfigurerer dumt eller galt. Eksempel kan være feil oppsatt brukerrettigheter eller ved deling av ting en vanligvis ikke deler. Eksempelvis gjennom nettverksdeling eller FTP. Dette vil gjelde både for de tjenester vi velger å kjøre og for operativsystemet. Igjen vil kanskje dette lure folk til å tro at det her er snakk om mangelfull kunnskap og at det ikke er en felle. Dette må selvfølgelig gjøres på en slik måte at det ikke vekker mistanke eller hemmer nettet i særlig stor grad, men heller å gjøre serveren litt mer attraktiv ved å virke mer mottakelig for angrep.

- Navngiving – En ting å vurdere er å se på selve navnet en gir en maskin. Hvis en gir navn som røper hvilken rolle i nettet maskinen er tiltenkt å ha, vil dette kanskje kunne være med på å rette inn angrep mot visse servere i motsetning til andre. For eksempel kan det tenkes at bruk av navn som inneholder ord som SAP, SQL, DATABASE eller TRANSACTION kan få noen maskiner mer attraktive enn andre.

Hvis det viser seg at passiv lokking ikke har ønsket effekt, kan det være ønskelig å aktivt lure til seg trafikk. Her kommer aktiv lokking inn i bildet. I motsetning til passiv lokking, vil vi her altså selv gå ut og gjøre første steget i angrepsprosessen. Det er ikke meningen å oppsøke potensielle angripere, men heller å synliggjøre oss mer enn vi ville ha gjort bare ved å vente. Noe en imidlertid må tenke over her, vil være de juridiske konsekvenser det har å aktivt lokke folk til å gjøre lovbrudd. I en bevisst situasjon med bruk av aktiv lokking, vil det nok være vanskelig å få tatt angriperen for noe, da denne da kan hevde at denne ble lurt til å gjøre det som ble gjort. Denne lovgivningen er forskjellig fra land til land og vi vil derfor ikke nevne noen spesielle lover for et enkelt land som for eksempel Norge.

Noen forslag til fremgangsmetoder:

- Publisere servertjenester – Her ser vi for oss at en servertjeneste som for eksempel en webserver, eller nærmere bestemt adressen til denne serveren blir registrert og offentliggjort hos kjente søkemotorer. Slik vil det bli enklere for individer på Internett å finne oss, samtidig som vi kan sette serveren i den sammenheng vi vil den skal stå i ved å benytte de nøkkelord/søkeord vi må oppgi ved en slik registrering.

- Eksponere IP-adresser – Forsøke å vekke interesse ved å nærmest publisere IP-adresser i Honeynettet gjennom bruk, altså trafikk ut fra nettet. Benytte seg av webleser, surfe litt rundt på nettet, teste ut IRC og også peer-to-peer programvare som Kazaa eller Direct Connect. I tillegg kunne det også være en tanke å teste ut ICQ, eller andre Instant Messaging programmer for å se om dette også har en effekt på aktivitet inn mot testnettet. Denne måten å gjøre ting på vil kanskje også være nyttig for å undersøke hvordan vanlig bruk av nettresurser kan være med å

endre trusselbildet. Vi kan anta at slik aktivitet fra vår side kan ha en viss effekt på antall skanninger / angrep, men dette får senere forsøk vise.

- Uforsiktig oppførsel – Dette vil likne mye på forrige punkt. Også her er det snakk om trafikk ut på Internett fra oss. Her vil vi med vilje oppføre oss uforsiktige, både med tanke på valg av steder vi velger å kontakte, men også på hva vi foretar oss ved disse stedene. For eksempel ser vi for oss at vi surfer oss inn på steder som det ikke alltid er like lurt å gå inn på. I tillegg til å bare klikke seg inn på disse stedene, vil en da også tillate alt av cookies, plugin og andre tillegg. Det kan også være en ide å faktisk teste ut all reklamen som popper opp rett som det er. Hva skjer egentlig hvis en er så dum og sier ja?

- Registrere e-post – For å eksponere oss litt mer kunne det vært en ide å registrere e-post adressene våre rundt om på nettet. Dette kan gjøres i forbindelse med forrige punkt, ved å legge igjen adressen vår på forum og ved å registrere oss på diverse mailinglister det ofte reklameres for på slike steder. I tillegg kunne det være en ide å gjøre det samme mens vi benytter for eksempel IRC.

Dette var noen forslag på mulige måter å tiltrekke seg trafikk inn mot et honey-nett. Vi kan ikke si med sikkerhet om disse vil ha noen innvirkning her og nå, dette er bare antagelser, men vi vil gjøre forsøk med honey-nettet vårt for å se om det er mulig å trekke noen slutninger om hvorvidt det merkes noen endring i mengden angrep, eventuelt angrepstype.

3.6 CASE

Vi skal her se på et typisk angrep som blir foretatt av en scriptkiddie med det fiktive navnet, 100z3R. Dette er selvfølgelig en rent hypotetisk case, men flere av metodene som blir vist her er tatt fra kilder som har erfaring med lignende angrep[43,20]. Vi har bestemt oss på å fokusere denne casen på en typisk scriptkiddie og ikke en proff hacker/cracker. Årsaken er at slik vi har mulighet til å sette opp nettet vårt, er sjansene størst for at vi får besøk av denne typen inntrengere, og ikke av de litt mer ressursfulle. I tillegg er også dokumentasjonen rundt slike angrep mye større og lettere tilgjengelig, da denne typen angrep er mest synlige og er svært standardisert (rootkit).

La oss si at angrepet starter klokken syv om kvelden (dato er ikke relevant). 100z3R har nettopp satt seg ned ved maskinen sin og har logget inn på IRC kanalen, der han/hun møter likesinnede og planlegger angrep. En kamerat har nettopp fått tak i en ny exploit som han deler villig med de andre deltagerne på kanalen. Denne exploiten benytter en kjent svakhet ved en tjeneste som følger med nyeste versjon av Red Hat[33]. Deltagerne i kanalen har i lang tid planlagt et DDOS angrep mot en ISP i USA. Årsaken er at en av disse har fått stengt kontoen sin på grunn av misbruk og målet er da selvsagt å ta hevn. For å gjennomføre dette angrepet er det behov for 50 kraftige maskiner med tilgang til høyhastighetsnettverk.

100z3R er en av hovedpersonene bak dette angrepet og trenger å ta over flere maskiner, siden de til sammen bare har 30 maskiner tilgjengelig. Først trenger de imidlertid å finne servere der ute som kjører den aktuelle tjenesten som exploiten utnytter. 100z3R setter da i gang en portskanning over en rekke IP-serier, som rommer til sammen ca. 2 millioner potensielle ofre. Denne skanningen vil ikke bli gjort fra privatmaskinen, men fra en maskin som allerede er tatt over tidligere. Da skanningen vil komme til å ta litt tid, velger vår venn å ta kvelden og komme tilbake senere for å få en oversikt over de mål det automatiske skanningsprogrammet har funnet.

Etter 10 timer er skanningen ferdig og det foreligger en liste med rundt 100 potensielle mål. En av disse blir valgt ut og den nye exploiten blir kjørt mot denne maskinen. Etter 5 sekund får vedkommende tilbakemelding på at root-aksess ble oppnådd og maskinen er klar til bruk. Tjenesten som hadde en svakhet sikres slik at ingen andre oppnår root på denne maskinen. En sjekk på hvor maskinen befinner seg viser at målet er i Storbritannia og har 100Mbit Internett-tilgang. Dette er informasjon som gjør at 100z3R velger å beholde maskinen og installerer en IRC-bot som gjør at maskinen kan fjernstyres fra kanalen vennene okkuperer.

Som oppsummering ble følgende gjort for å få en maskin med i et DDOS angrep mot en tredjepart:

- Skanningen ble foretatt, noe som resulterte i 100 potensielle ofre.
- Exploiten ble kjørt mot et av disse ofrene og root-tilgang ble oppnådd på kort tid.
- Lokasjon og hastighet ble undersøkt.
- Til slutt ble en IRC-bot installert og angriperne kan fortsette til neste offer.

Denne prosessen beskriver en mulig vei for å oppnå root status. En annen måte som også blir benyttet, er at skanning, root og bot-installering kan gjøres i en og samme operasjon(automatisk).

Formålet med dette angrepet var å få tak i nok båndbredde til å føre et DDOS angrep mot en ISP. Vår venn 100z3R kunne likegodt brukt den maskinen han/hun tok til å utføre annet kriminalitet.

Som eksempel kan nevnes:

- Plattform for videre angrep.
- Passordsniffing, dersom maskin står på et interessant nettverk.
- Installering av ”bouncer”, en proxy for IRC[18] for å skjule IP.
- E-post spam.
- Mp3 og pornodistribuering.

4 Metodedel

Dette er en oversikt over gangen i prosjektet for å vise hvordan vi har gått fram fra begynnelse til slutt. Mer utfyllendes informasjon finnes i andre deler av rapporten.

1. Litteraturstudium - Dette involverte generelle studier om datasikkerhet. Hvem som angriper, hva de angriper og metoder de benytter er beskrevet her. I kapittel 3 beskrives Honeynett nærmere.

2. Designfase av nettet - Her ble det utført samtaler mellom skolen, Proseq og oss for sammen å utbedre en løsning som endte i at vi kunne få satt opp nettet på IKT labben ved HIA Grimstad. Her var det viktig å sette opp klare regler for hva vi kunne få lov til å gjøre på nettet, og hvilke forholdsregler som skulle gjelde under forsøket. Skolen var veldig strikse på dette området siden de ikke ville utsette sitt nett for potensielt misbruk.

3. Anskaffelse av hardware - Skolen stilte med Internetttilgang, og en del nettverksutstyr. Fra Proseq fikk vi 2 servere i rack. Den ene var en IDS boks med Linux som brukes til å monitorere nettet. Den andre var en litt kraftigere maskin med Windows 2000 server som skulle brukes til å

kjøre VMware. I tillegg stilte vi med en egen maskin da skolen ikke kunne gi oss noen hardware som vi kunne benytte som brannmur. Vi vurderte selvsagt muligheten for å kjøre brannmuren på VMware maskinen, men det var veldig praktisk å kunne ha brannmur på egen maskin, da vi kunne observere trafikken i sanntid på egen skjerm. Det er også flere grunner til dette valget, men det beskrives bedre seinere.

4. Oppsett og konfigurering av nettverket - Denne fasen involverte oppsett av nettverksutstyr og datamaskiner. Ved fysisk oppsett av nettet ble det foretatt en del kompromiss, da vi ikke fikk all hardware som vi ønsket. Enkelte ting ble forkastet fra ønsket design. Eksempel: ruter, mer minne og ekstra harddisker.

5. Gjøre oss kjent med VMware - VMware ble benyttet til kjøring av virtuelle maskiner og var nytt for oss begge i bruk. Naturlig nok ble det benyttet en del tid til å sette seg inn i konfigurasjon og oppsett. Siden nesten hele Honeynettet kjøres som virtuelle maskiner er riktig oppsett av VMware viktig for å unngå feil.

6. Gjøre oss kjent med Linux - Redhat Linux 7.2 ble valgt som plattform for å kjøre de virtuelle IX-maskinene. Dette er den nyeste distribusjonen fra Redhat per dags dato. Ingen av oss kunne så mye om Linux fra begynnelsen og det ble benyttet en del tid for å sette oss inn i det aller mest grunnleggende. Debian Linux[9] ble valgt til å kjøre på brannmuren, da Debian generelt har ett bedre sikkerhets rykte enn Redhat. Denne maskinen kjørte ikke noen andre tjenester enn ruter/brannmur. Målet med denne er å slippe all trafikk inn, men begrense trafikk ut.

7. Gjøre oss kjent med Windows operativsystem - Her snakker vi her om grunnleggende sikkerhet og loggmuligheter i Windows XP.

8. Oppsett av tjenester som kjøres på de virtuelle maskinene - Dette er tjenester som f.eks. Webserver, FTPserver og lignende. Disse er lokkemiddel for eventuelle innbrudd.

9. Oppsett av logging - Siden vi ikke alltid er tilstede når ting skjer, er logger vitale for å kunne oppdage om maskinen er kompromittert og hva angriperen foretok seg.

10. Monitorering av nettet og analyser - Dette var en prosess som ble kjørt kontinuerlig under hele perioden. Selvsagt etter nettet ble satt opp.

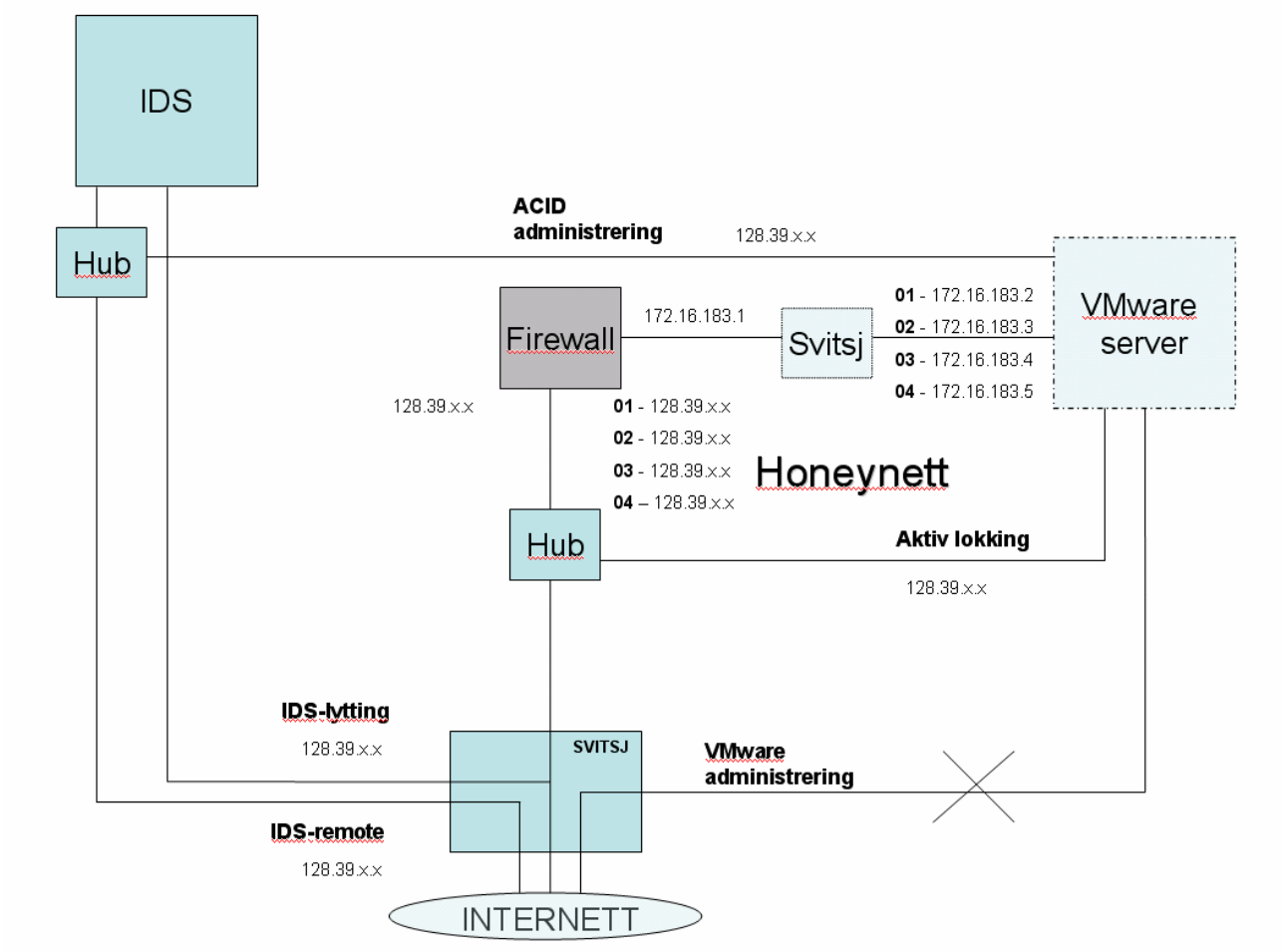
11. Aktiv lokking - Her går vi aktivt ut på nettet og prøver å tiltrekke trafikk til Honeynettet. Eksponering av oss selv utføres med vanlige applikasjoner som ICQ og Mirc[27].

Hele tiden under forsøket ble det installert nye tjenester og operativsystem, så fase 4-11 ble utført nesten kontinuerlig gjennom hele perioden. Rapporten ble selvsagt også skrevet parallelt med kjøring av Honeynettet.

5 Design av nett

For å sette opp et fungerendes nett trengte vi hjelp av IT-ledelsen ved HIA. IKT labben ved skolen passet bra og det ble satt opp forbindelser der. Det har blitt lagt til noen få komponenter etter hvert, blant annet et par ekstra hubber for å kunne koble på maskiner. Men i hovedsak er nettet som det ble designet fra begynnelsen av. Under er en oversikt over nettet, og en dypere forklaring til komponentene.

5.1 Forklaring



Figur 2 – Oversikt over vårt Honeynett

5.1.1 Forklaring til komponenter og forbindelser

- VMware server - Dette er server maskinen som inneholder alle de virtuelle maskinene. Denne maskinen er levert av Proseq og kjører Windows2000 server. Den har 2 prosessorer og 512MB RAM.
- IDS - Debian Linux boks som kjører IDS fra Proseq. All logging av nettet utføres her. På denne maskinen kjøres TCP dump og Snort. Man har også mulighet for å bruke ACID[4].
- Brannmur - Dette er brannmuren, en Debian Linux boks som står og monitorerer forbindelsene

ut og inn og logger dem. Antall forbindelser ut reguleres også.

- SVITSJ - Dette er hovedsvitsjen inn til IKT labben. På denne svitsjen finnes det en monitoreringsport som er koblet direkte i IDS boksen som muliggjør logging av alt som skjer på Honeynettet. 3 andre forbindelser er også satt opp på svitsjen.
- Svitsj - Dette er en liten svitsj som er satt opp mellom brannmuren og VMware serveren. Den er ikke strengt tatt nødvendig, men muliggjør tilkobling av eksterne maskiner til Honeynettet på en enkel og grei måte.
- Hub - Det brukes 2 hubber i nettet. Begge 2 brukes for å kunne koble til maskiner på segmenter når det er hensiktsmessig å gjøre dette.
- IDS-remote - SSH forbindelse inn til IDS som gjør at Proseq kan fjernstyre IDS boksen fra Arendal.
- IDS-lytting - Logging av alt som skjer på Honeynettet gjøres over denne forbindelsen.
- ACID administrering - Dette er en midlertidig forbindelse som settes opp etter behov. Forbindelsen går fra Windows 2000 serveren og inn på IDS boksen sånn at vi kan få hentet logger, TCPdump og bruke ACID til monitorering av nettet. Det er viktig å påpeke at denne forbindelsen ikke er fra de virtuelle maskinene som er på honey nettet, men fra Verts OS som ikke er på Internett.
- VMware administrering - Dette er en forbindelse som var tenkt å brukes til fjernstyring av VMware maskinen og uthenting av logger, men er ikke brukt siden vi fysisk ved å sitte på IKT-labben klarer å utføre alt som trengs.
- Aktiv lokking - En forbindelse som ble brukt til aktivt å trekke trafikk til nettverket. Forbindelsen går utenfor brannmuren og har dermed ingen begrensinger på data som blir sendt.

NAT brukes i brannmuren, så hver Internett IP har en lokal IP. For eksempel 172.16.183.1 har Internett adresse 128.39.x.x

Skolen var veldig bekymret for at eventuelle angripere skulle benytte skolens nett til å gjøre angrep på andre maskiner, så oppsettet måtte godkjennes av Proseq før vi fikk lov til å sette det opp. Brannmuren er ekstremt viktig her siden den begrenser trafikken og muligheter som angripere har til å benytte maskinene. Derimot er denne også et svakt punkt da dersom denne tas, så har angriperen total kontroll på Honeynettet og kan bestemme trafikkmønsteret selv. Vi kan ikke gardere oss 100 % mot angrep, så IDS sensoren er ekstremt viktig for å monitorere nettet og oppdage angrep. Siden den står passivt og overvåker kan ikke en eventuell angriper vite om denne.

5.2.2 Ting ved designet som kunne vært gjort annerledes

”Project Honeynet” anbefaler en ruter mellom VMware maskinen og brannmuren for å skjule den og skape et mer realistisk miljø. Dette var ikke mulig å få til da vi ikke hadde noen ruter tilgjenglig. Ved oppsett av et annet Honeynett ville vi hatt med denne løsningen.

Brannmuren kunne kjørt på VMware maskinen siden denne maskinen har tilstrekkelig med nettverkskort. Grunnen til at vi valgte å kjøre en egen fysisk maskin var at VMware maskinen hadde ikke tilstrekkelige mengder med RAM, som vi etter hvert begynte å gå tom for. Dessuten var det praktisk å ha denne på en egen skjerm også. Ved å sette i flere skjermkort i VMware maskinen, samt mer RAM, kunne vi med enkelhet kjørt brannmuren også på VMware maskinen og spart en

del ressurser.

IDS-boksen logger all trafikk som kommer ut og inn fra Honeynettet, men det hadde vært hensiktsmessig å ha ett nettverkskort i IDS boksen til slik at vi kunne logget det som skjedde internt på Honeynettet. Dette ble ikke gjort pga tidsproblemer, men det hadde helt klart vært nyttig. For eksempel har vi en loggserver internt på nettet som vi ikke kan monitorere siden den bare kjører interne IP-er. En potensiell angriper vil likevel ikke ha mulighet for å sende data ut fra denne dersom den blir tatt, da dette krever oppsett i brannmuren. Vi fant derfor ut at vi ikke ville bruke tid på ekstra monitorering internt.

5.2 Hardware

For å bygge et fungerendes Honeynett, så trenger man en del hardware. Dette er en mer detaljert liste over hva slags hardware som ble brukt for å designe nettet.

- VMware maskin - Dual P3-733MHz, 512MB RAM, 2x3Com 10/100 Mb, Dual Intel 10/100 Mb, 40GB IDE disk, 2x36GB SCSI UW160
- IDS boks - AMD K7 1GHz, 512MB RAM, 40GB IDE Harddisk
- Brannmur - AMD K7, 384MB RAM, 2xUnex 10/100 Mb.
- 2 hubber - 3com 10Mbit.
- 1 liten svitsj - Unex 10/100Mb.
- 1 stor svitsj - Cisco 10/100/1000Mb.
- 11 TP-kabler - RJ45, cat5.

5.3 Oppsett og konfigurering av nettverket

Vi fikk som sagt ikke all hardware som vi ønsket oss og det tok litt tid å bli enig med skolen om oppsett og design av nettet. VMware maskinen har 4 nettverksporter og vi bruker bare en av dem, pluss en til ACID eller aktiv lokking. I begynnelsen tenkte vi å sette opp hver virtuell maskin med sitt eget nettverkskort siden vi fikk såpass mange porter ledig når vi kjørte separat brannmur. Poenget med dette falt veldig fort bort når VMware bare trenger ett nettverkskort. Grunnen til dette er at VMware tillater at flere virtuelle maskiner får tilgang til samme nettverkskort under hver sin IP (WMvare ruter informasjonen videre). Meningen med flere nettverkskort var å minimere risikoen for at nettet skulle gå ned, da alt står og faller på den ene TP-kabelen som serveren er koblet til med. Vi fant likevel fort ut at dersom vi skulle kjøre det gjennomført med en kabel til hver virtuell maskin, hadde vi ikke nok porter uansett. Denne løsningen gikk vi fort bort fra.

På oversikten over nettet ser vi alle de fysiske koblingene, men det er også mulig å opprette virtuelle nettverkskoblinger mellom de virtuelle maskinene og Verts-OS. Man bruker et såkalt virtuelt nettverkskort. Denne koblingen brukes når vi skal hente ut logger fra loggserveren og til Verts-OS, som er Windows2000. De virtuelle maskinene er avskilt fra Verts-OS og skal i teorien ikke kunne få kontakt med vertsmaskinen. Skulle derimot det umulige skje, så vil denne koblingen være en sikkerhetsrisko. Vi sørget derfor for at denne bare er oppe i veldig korte perioder av gangen, og er dermed ikke noen betydelig risiko. Den største fordelen ved å benytte en sånn kobling

til å hente logger er at en eventuell angriper ikke kan se at det går trafikk på nettet. Den er totalt skjult, om ikke loggserveren er tatt over.

5.4 Konfigurasjon av VMware - Oppsett og problemer

VMware er programmet som muliggjør oppsett av virtuelle maskiner. Under er oversikt over forskjellige måter vi testet ut VMware på, hva som ikke gikk og hvilken løsning vi endte med.

Hardwaren vi hadde tilgjenglig var:
2-prosessor maskin (P3), med Windows 2000 Server
3 nettverkskort med totalt 4 RJ45 plugger
3 harddisker, 1 IDE og 2 SCSI (UltraWide 160)

Software vi hadde tilgjengelig var:
VMware 2.04 Workstation.

Det viktigste for oss var å få logget alt som skjer på de virtuelle maskinene, og vi valgte en løsning der Windows2000 Server skal logge alt som skjer og lagre loggene lokalt. Siden denne maskinen ikke er synlig for de virtuelle maskinene og ikke er koblet på nettverk skal dette være en veldig sikker løsning. Man har mulighet for å gi tilgang til de virtuelle maskinene så de kan få aksess til Verts OS, men dette er ikke ønskelig.

Løsning 1.1:
Vi gir de virtuelle maskinene en disk hver.

Fordeler:
Vi kan nå aksessere maskinene veldig enkelt fra verts-OS. Diskene er fritt tilgjengelig og vi kan lese loggene veldig enkelt. (Dette fant vi ut var ikke helt sant, da vi bare kunne ha lese tilgang til diskene samtidig, ikke skrive tilgang, dvs. at ingen informasjon kan lagres på de virtuelle maskinene.)

Ulemper:
Vi kan bare ha et virtuelt OS på hver harddisk, og vi kaster bort veldig mye plass. Ulempen er også at vi bare kan ha 2 virtuelle OS kjørende til en hver tid. Løsningen er å sette i flere disker, men dette har vi ikke tilgjengelig så det er uaktuelt.

Løsning 1.2:
Vi bruker virtuelle disker

Fordeler:
Vi kan nå enkelt lage så mange virtuelle maskiner som vi vil. Disse kan enten plasseres på samme disk eller spres ut over flere disker. Man kan enkelt starte opp flere OS enn det er harddisker. Dette muliggjør en fleksibilitet som man ikke har i løsning 1.1.

Ulemper:
Disk filene som de virtuelle maskinene ligger på kan ikke leses av Windows direkte. VMware har et Perlskript[1] som man kan kjøre under Linux, men altså ikke på noe annet operativsystem uten eventuelle emuleringer.

Siden vi ikke har nok harddisker er vi avhengig av å gå for løsning 1.2. Dette innebærer at vi må ha en måte å aksessere disse diskfilene på.

Vi har igjen 2 løsninger som vi ser for oss:

Løsning 2.1:
Dele disse virtuelle filene over nettverk til en Linux maskin som igjen monterer disse og får lese tilgang.

Løsning 2.2:
Kjøre opp en virtuell Linux maskin på vertsmaskinen som vi bare skal bruke til logging.

Den siste er den meste elegante løsningen da man ikke trenger noe ekstra hardware og man ikke bruker noen nettverkskobling. Selv om denne logg maskinen vil ta litt ekstra ressurser gir dette oss muligheten til å logge, da perl skripet som følger med VMware kan brukes i Linux.

Et problem er at versjon 2.04 har vanskeligheter med å lese SCSI disker som er større enn 8.4 GB. Det skal i teorien gå, men kan være litt problemfylt. Dette kan løses med at den virtuelle maskinen kan kjøre på versjon 3.0 av VMware.

En løsning ville selvsagt vært å ha brukt versjon 3.0 til alle de virtuelle maskinene, men perl skriptet for lesing som følger med støtter ikke formatet på de virtuelle filene som versjon 3.0 bruker. Versjon 3.0 deler de virtuelle diskene i mange små filer.

Det viser seg at Perl skripet som føler med VMware må monteres hver gang man skal lese oppdateringer på den virtuelle maskinen. Dette er uholdbart og gjør bruk av dette skriptet verdiløst. Vi kan ikke kjøre skriptet hver gang vi skal se oppdateringer i loggene. Vi har følgende løsninger som vi ser for oss.

Løsning 3.1:
Forkaste lesing av loggfiler direkte og heller sette opp en dedikert loggserver på Honeynettet.

Løsning 3.2:
Lese loggfiler direkte, men montere de virtuelle maskinene hver gang vi skal lese logger.

Siden løsning 3.2 som nevnt er uaktuell går vi for løsning 3.1. VMware 2.04 har dessuten såpass vesentlige mangler at vi velger å gå for versjon 3.0, da vi ikke lenger trenger å lese loggfiler med Perl-skriptet. Versjon 3.0 støtter filer større enn 2GB og har bedre støtte for SCSI disker. Dessuten følger det med en del oppdateringer som er nødvendige når man skal kjøre OS som Windows XP. Disse kan lastes ned til versjon 2.04 også, men siden de følger med 3.0 er det praktisk å bruke denne versjonen.

5.5 Gjøre oss kjent med Linux

Redhat Linux 7.2 var den nyeste Linux distribusjonen når vi satt opp Honeynettet. Den kommer med Linux kjerne 2.4.10 og går for å ha den største markedsandelen (Linux) på nettet i dag. Hvor sikker er en standard distribusjon? Dette var et spørsmål som var interessant å finne ut. For å tiltrekke seg angripere til Honeynettet er det en fordel å bruke programmer og tjenester som er kjent. Derfor valgte vi ikke en mer ukjent Linux distribusjon.

Ingen av oss var eksperter på Linux, og Redhat har masse informasjon om hvordan man skal sette opp Linux som en server. Dette gjelder generelt for alle Linux versjoner, men det er likevel mindre

forskjeller mellom distribusjonene. Redhat har også veldig gode grafiske konfigurasjonsverktøy om dette er ønskelig å benytte. Etter hvert valgte vi likevel for det meste å editere config filer manuelt. Dette er mer praktisk etter som man lærer seg mer om oppsett av Linux.

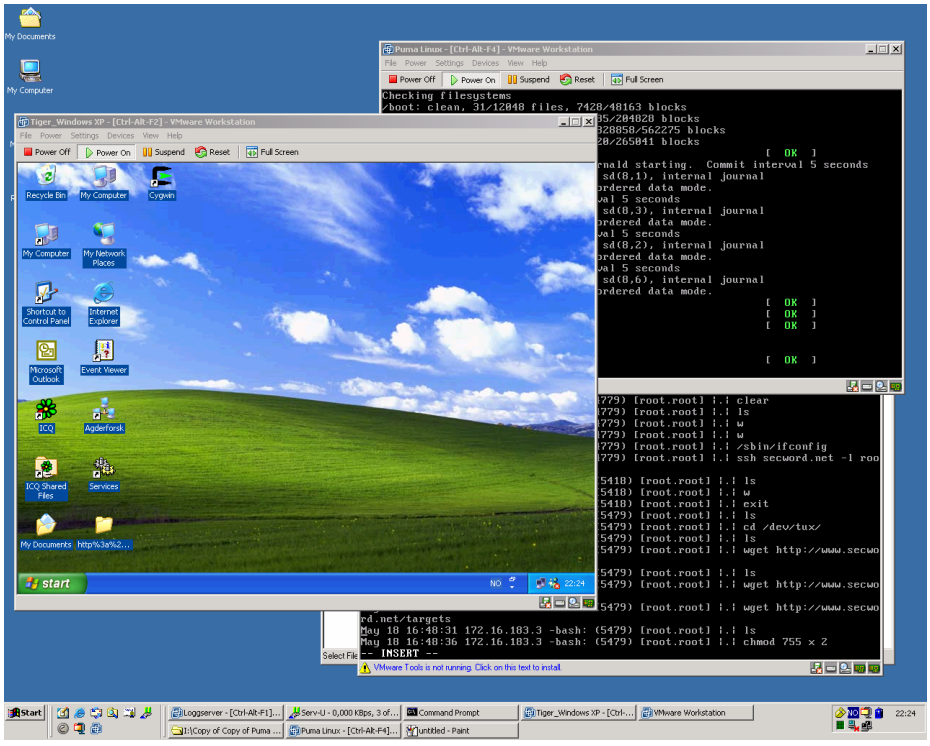
Debian har en overlegen pakkeadministrering synes mange, og vi har noe erfaring ved å bruke Debian som brannmur før. Den er også generelt regnet som sikrere distribusjon. IDS-boksen kjører som nevnt også Debian.

5.6 Gjøre oss kjent med Windows

Windows er et operativsystem som vi bruker både som virtuell maskin og verts-OS. Det som vi måtte sette oss dypere inn i her var loggmuligheter. Vi hadde begge to gode kunnskaper om Windows generelt.

5.7 Oppsett av virtuelle maskiner

Dette er virtuelle maskiner som lagres som diskfiler på harddisken. Under vises 3-VMware maskiner som kjører under Windows 2000. 2 Linux maskiner og 1 Windows maskin.



Figur 3 – VMware i aksjon

5.7.1 Oppsett av Windows XP

Vi valgte Windows XP som virtuell maskin, da dette er det nyeste operativsystemet til Microsoft. OS-et ble installert uten oppdateringer, da det ofte er slik de fleste installerer det. WindowsXP er et OS som krever forholdsvis mye med RAM. 128MB ble satt av, men dette måtte etter hvert skaleres ned til 64, pga av minnemangel. Vi fikk ikke lov til å installere den virtuelle maskinen med mindre enn 128MB minne dersom man valgte standardoppsett. Dette viser selvsagt at VMware ikke anbefaler noe mindre. Vi fikset mangel på minne ved å manuelt konfigurere dette etter at Windows

var installert. Annen virtuell hardware som ble installert var 10mbit nettverkskort, 4GB Harddisk, CDROM, Diskettstasjon og USB støtte.

All VMware hardware blir lik på alle de virtuelle maskinene.

5.7.2 Oppsett av Redhat 7.2

Installering av Redhat 7.2 som virtuell maskin består av samme fremgangsmåte som WindowsXP. Vi bruker samme hardware, med et unntak. Vi satt bare av 32MB minne. Dette er nok for denne maskinen siden vi som oftest kjører denne i kommandolinje modus.

5.7.3 Oppsett av Loggserver (Redhat 7.2)

Til loggserver benyttet vi lik hardware som i den andre Redhat 7.2 maskinen, også her med 32MB minne, men med et unntak. Vi har i denne maskinen 2 nettverkskort. Det ene er standard kort som går ut på Honeynettet, det andre brukes til å lage et nett mellom Verts-OS og Loggserver. Dette vil være et privat nett som bare gjelder mellom disse maskinene.

5.7.4 Oppsett av WindowsXP utenfor brannmur

Denne Windows XP-maskinen er en tro kopi av den maskinen som vi bruker i Honeynettet. Vi brukte denne maskinen til å aktivt eksponere nettet mot omverdenen. Nettverkskoblingen til denne maskinen gikk gjennom et annet fysisk nettverkskort og ble koblet til utenfor brannmuren.

5.7.5 Oppsett av Redhat 6.2

Den siste Redhat maskinen som ble satt opp var en Redhat 6.2 versjon, altså en litt eldre server. Likevel benyttet vi samme hardware som en Redhat 7.2, og med samme mengde minne.

5.8 Oppsett av fysiske maskiner

Dette er de fysiske maskinene og inkluderer verts-OS og brannmur.

5.8.1 Oppsett av Windows2000 server (Verts-OS)

Dette er en Windows 2000 server standard installasjon og ble satt opp av Proseq. Ingen unødvendige tjenester ble installert, da denne maskinen ikke skulle stå og ha noen tjenester mot Internett. Klientprogrammer eksisterer likevel til dette formålet. Opprinnelig var det planlagt å fjernstyre denne maskinen for henting av logger osv, men denne løsningen ble forkastet. Den eneste kontakten denne maskinen har med andre(noen ganger) er IDS-boks, brannmur og virtuell forbindelse til loggserver.

5.8.2 Oppsett av brannmur

Brannmuren har 2 nettverkskort, der det ene går ut mot Internett og det andre mot Honeynettet. På maskinen kjører vi IPtables under Debian Linux(unstable version, dvs mest moderne). IPtables skriptet lastet vi ned fra ”Project Honeynet”, og dette virker slik at all trafikk slipper inn, men antall forbindelser ut reguleres. Hver gang skripet startes, restartes disse reglene. Dette brannmurskripet er veldig komplisert, men heldigvis er det enkelt å konfigurere. Oppsettet vi benyttet følger med som

vedlegg.

5.9 Logging

Logging er vitalt for å finne ut hva en angriper gjør eller hva han/hun har gjort. Logger er noe av det første en angriper vil prøve å ødelegge, så fjernlogging er en nødvendighet på de fleste maskiner. Alle maskiner logger, og under følger en oversikt hvordan de ulike maskinene logger, hva de logger og hvordan vi bruker loggene.

5.9.1 WindowsXP

Windows XP inneholder en eventlogger som håndterer Application, Security og System logger. Disse lagres som filer på Windows maskinen. For å få forbedret loggmuligheter gjorde vi noen forandringer på standard oppsettet. Dette beskrives mer detaljert i vedlegg om logging.

Loggfiler ble lagt i en mappe som heter C:\\logs. Denne forandringen ble utført i registeret. Katalogen ble delt slik at loggserver kunne få lesetilgang til disse filene over nettverket.
Det optimale hadde vært at XP-maskinen kunne logge direkte til loggserveren, istedenfor at loggserver henter filene. Dette ble prøvd i begynnelsen, men siden vi måtte ha SMB fildeling med skriverettigheter på loggserveren, ble dette en for stor risiko å ta. Det eksisterer også løsninger som logger direkte til syslogd på Linux maskinen, eksempel er et program som heter EventReporter[10], men dette valgte vi å ikke ta i bruk. De beste løsningene må man dessuten betale for, som ikke vi hadde mulighet til.

5.9.2 Redhat 7.2 Server

Linux maskiner har veldig gode loggmuligheter, men når man benytter disse i Honeynett så er ofte de originale loggmulighetene for dårlige. Det første en angriper vil gjøre når han/hun kommer inn på maskinen er å slette alle spor, derav loggene. I begynnelsen prøvde vi ut muligheter for logging ved å bruke tail-kommandoen og dermed overføre logger over SMB nettverket. Denne løsningen var en altfor tungvint løsning og vi fikk satt opp loggserveren til å stå og motta logging remote fra Linux maskinene.

Et problem når en angriper kommer inn på maskinen er at det er veldig enkelt å slå av sysloggen. Dermed vil vi bare ha logger fram til dette skjer. En løsning vi implementerte på slutten av forsøkene var å kjøre en falsk syslogg, som vi kalte for noe helt annet. Da vil ikke en eventuell angriper få gjort noe med den originale sysloggen som er slått av, da bare én syslogg kan kjøre om gangen. Dette ble ikke benyttet før i aller siste oppsett av server. Grunnen til at vi ikke benyttet en falsk syslogg helt fra begynnelsen var at vi ikke så betydningen av å finne ut hva en angriper gjorde på maskinen etter han hadde fått root. All informasjon opp mot hvordan maskinen ble tatt tok IDS sensoren seg av, og vi tok alltid ned maskinen med en gang uansett. Likevel valgte vi å sette opp en falsk syslogg i et siste forsøk da vi fant oss tid til å sette dette opp, selv om ikke vi synes det var streng tatt nødvendig. Vi visste ikke helt hvor mye jobb det ville bli med å lage en falsk syslogg, så dette ble utsatt pga. tidsnød og fordi det ikke var kritisk for oppgavegjennomføring.

Kommandolinje logging ble også lagt til da det kunne være nyttig å finne ut nøyaktig hvilke kommandoer angriperen vår utførte.

5.9.3 Redhat 7.2 Loggserver

Denne maskinen ble oppsatt som remote loggserver. Den mottok all logging fra andre Linux

maskiner, samtidig som den logget lokalt. Hele tiden tok den også siste endringer i WindowsXP logg filene og tok sikkerhetskopi på serveren. Denne operasjonen ble utført over SMB nettverk. Tail - kommandoen ble benyttet til denne jobben.

5.9.4 Redhat 6.2 Server

Redhat 6.2 er en eldre Redhat distribusjon, men logging er akkurat likt som på versjon 7.2. Vi logget remote til loggserver, men vi hadde ikke kommandolinjelogging. Vi fant ut at denne ikke hadde noen mening uten en falsk syslogg. IDS tar alt uansett før de oppretter en kryptert forbindelse, og da er forsøket vårt som oftest over.

5.9.5 Brannmur

Branmuren logger bare lokalt, men alle forbindelser ut og inn lagres i loggene. Dette er meget nyttige logger og brukes ved analysering av angrep. Logger ble overført til Verts-OS på VMware maskinen over FTP når Honeynettet var nede.

5.9.6 IDS

IDS-boksen er den klart viktigste loggingsmuligheten vi har. Her lagres all trafikk som TCP-dump og Snort analyserer trafikken og legger ut alarmer til oss som vi kan vi kan analysere med ACID. Uten denne ville det blitt vanskelig å finne ut at maskinen vår hadde blitt tatt. For å analysere TCP-dumpen benyttet vi et program som kalles Ethereal[42]. Her kan vi enkelt observere enkelte pakker inn på nettet, eller hele forbindelsen til en angriper. Hvilke kommandoer som kjøres osv. Så lenge dette pågår i klartekst må nevnes.

5.9.7 Tips til forbedret logging

Ved slutten av forsøk 4 fant vi ut at vi skulle satt opp full logging på alle Linux maskinene. (Remote loggserver, falsk syslogg og kommandolinjelogging) Det var noen angripere som satt over på kryptert forbindelse litt for fort uten at vi fikk med oss den informasjon vi ville. Det må likevel nevnes at vi gikk gjennom en læringsprosess med å sette opp Honeynettet da ingen av oss hadde gjort dette før, og det er veldig enkelt å se tilbake og si at vi kunne gjort ting annerledes nå.

6 Forsøk

Forsøksdelen inkluderer 5 forsøk. Disse vil bli nøyere forklart under hver del og er utført etter hverandre over tid.

6.1 Forsøk 1 – Passivt Honeynett

Her vil vi beskrive og forklare første forsøk som ble gjort i Honeynettet. I tillegg til oppsett, vil vi også vise til de forventninger vi hadde, samt den konklusjon vi kom frem til etter at forsøket ble ansett som ferdig.

6.1.1 Beskrivelse

I dette forsøket ville vi holde en lav profil og se hva som ville skje med et utvalg maskiner satt ut på

Internett. Det var ikke meningen å på noen måte gå aktivt ut for å lokke til seg trafikk utenifra. Her mente vi det var to punkter som var spesielt viktig. En: Sette opp Honeynettet tilfredsstillende. Det ville både ødelagt mye for prosjektet og vært utrolig flaut om nettet var satt opp feil. Og to: Finne ut hvor lang tid det ville ta før noen fant oss (om i det hele tatt) og hva som ville hende da.

Siden dette var første forsøk, var det i tillegg nødvendig å sette opp og installere følgende maskiner:

- 2 x Linux Red Hat 7.2
- Windows XP

For å sikre loggfiler samt ha et sentralt sted hvor disse kunne observeres valgte vi å sette opp en loggserver. Til dette formålet valgte vi å benytte Red Hat 7.2 og under installeringen av maskinen tok vi valget custom og valgte selv de tjenester som skulle installeres. Siden denne maskinen ikke skulle brukes til annet enn det å være loggserver, kjører den ingen unødvendige tjenester ut mot Honeynettet eller har X installert. Slik kunne vi også spare VMware-maskinen for unødvendig ressursbruk.

Til neste maskin valgte vi også Red Hat, men her tok vi en generell server installasjon, da det ville være interessant å finne ut hvor sikker denne ville være. Slik endte vi opp med en maskin med en rekke ferdig installerte tjenester. Det ble i tillegg startet WU-ftpd, da vi var interessert i å ha en FTP server kjørendes i tillegg.

Siste maskin ble en Windows XP maskin, standard installasjon, rett fra CD. Her ble det ikke gjort annet enn å sette opp SMB nettverk, samt legge til gjeste-kontoen. Cygwin ble installert for å hjelpe til med logging, men dette var bare klientverktøy.

6.1.2 Forventninger

Siden dette tross alt var det første forsøket, så vi det ikke som særlig sannsynlig at vi ville oppleve den riktig store pågangen øyeblikkelig etter vi fikk Honeynettet online. Vi antok at det ville gå en viss tid før vi ble oppdaget. Hvis det ble aktivitet ville dette mest sannsynlig ikke komme fra virkelige mennesker, men heller være automatisert ormetrafikk og skanninger. Før eventuelle angrep, ville det antakeligvis komme skanningstrafikk for å kartlegge nettet vårt. Denne type trafikk sjekker ofte hele IP-serier og er ikke nødvendigvis rettet mot noen spesielt. Alt i alt ventet vi oss ikke at nettet ville bli angrepet under dette forsøket, men at det hele heller ville være en testperiode for konfigurasjonen og design av nettet. Dessuten ville vi få tid til å sette oss nærmere inn i innstillingene hos de forskjellige maskinene vi hadde kjørende.

Når det gjelder oppfatningen om hvor sikre de forskjellige maskinene egentlig var, var vi begge av den oppfatning at hvis noen maskiner var spesielt sårbare, ville dette mest sannsynlig være Windows-maskinen og ikke de Linux-baserte serverne. Vi har vel alle hørt historiene om hvor sikre Linux maskiner er i forhold til Windows og hvor mange sikkerhetshull det er å finne i Microsofts sine produkter.

En siste ting vi var litt bekymret over var hvor sikker loggserveren vår var mot angrep. Loggserven befinner seg på et intern-nett mellom honey-maskinene og er ikke direkte knyttet opp mot Internett. For å ta loggserveren måtte en hacker i så fall allerede ha tatt over en av de andre maskinene, noe vi forhåpentligvis ville ha merket og kunne stoppe.

Da dette var første gang Honeynettet skulle kobles til, hadde vi noen innkjøringsproblemer, vi ville være så sikre som mulig på at tingene var satt opp skikkelig, så nettet var ikke skikkelig i gang før

5-tiden på ettermiddagen den 11. April.

6.1.3 Resultat

Det var meningen å la nettet stå eksponert en tid for å samle opp måldata i loggene. Vi ble enige om at hvis noen av maskinene ble tatt, ville vi ta disse ned, mens det resterende nettet ville fortsette å stå eksponert. Dette skjedde, vi fikk e-post fra Nils Ulltveit-Moe og vi valgte da å ta ned den infiserte maskinen så raskt som mulig. Det resterende nettet fikk fortsette som før. De hendelsene som vises til i dette kapitlet gjelder både før og etter dette. Vi har valgt å ikke dele opp resultatdelen, selv om en maskin ble tatt ned.

Loggfilene viste seg fulle av aktivitet som skyldtes ormetrafikk (skripts og/eller virus), samt en god del skanningstrafikk jevnt fordelt over testmaskinene. Disse angrepene var ikke særlig gjennomtenkt og var ikke rettet mot noen spesiell maskin. Dessuten var denne typen trafikk forventet og ble derfor ikke ansett som spesielt interessant for dette forsøket, annet enn en bekreftelse på at vi "var der ute".

Men det var også mer å finne i loggene. Rett over klokken ni på kvelden den 11. April, startet det første av en rekke, mer direkte og interessante angrep mot honey-nettet. Angrepet var rettet mot Linux maskinen og angrepet kom fra adresse 128.X.14.X. Dette var et angrep som brukte en svakhet i ftp serveren vi hadde kjørende (WU-ftpd) og angriperen fikk raskt root-aksess gjennom denne exploiten. Linux-maskinen hadde holdt i nærmere 4 timer (6 hvis en ser bort i fra innkjøringsproblemene).

Neste angrep kom klokken syv på morgenen den 12. April. Denne gangen kom angrepet fra 65.31.X. X og det hele varte i omtrent fire minutter. Igjen var det Linux maskinen som ble utsatt for angrep og også her var det WU-ftpd tjenesten som ble utnyttet. Igjen ble angrepet vellykket og root-aksess ble gitt.

Disse angrepene mot Linux-maskinen skjedde før vi rakk å koble denne maskinen fra nettet. Følgende aktivitet i loggene ble observert etter at vi gjorde dette:

Klokken 19:58 den 12. April ble det igjen logget et forsøk på kontakt fra 65.31.X.X mot Linux maskinen, som nå ikke var tilgjengelig. Her var tydeligvis angriperen tilbake og prøvde å få gjenopprettet kontakt med en maskin denne tidligere på dagen hadde fått root-aksess til. Hva denne personen ville bruke denne maskinen til var ikke mulig å finne ut, da denne ikke var på nett. Det ble ikke funnet andre forsøk på gjenopprettelse av kontakt fra denne adressen i dette forsøket.

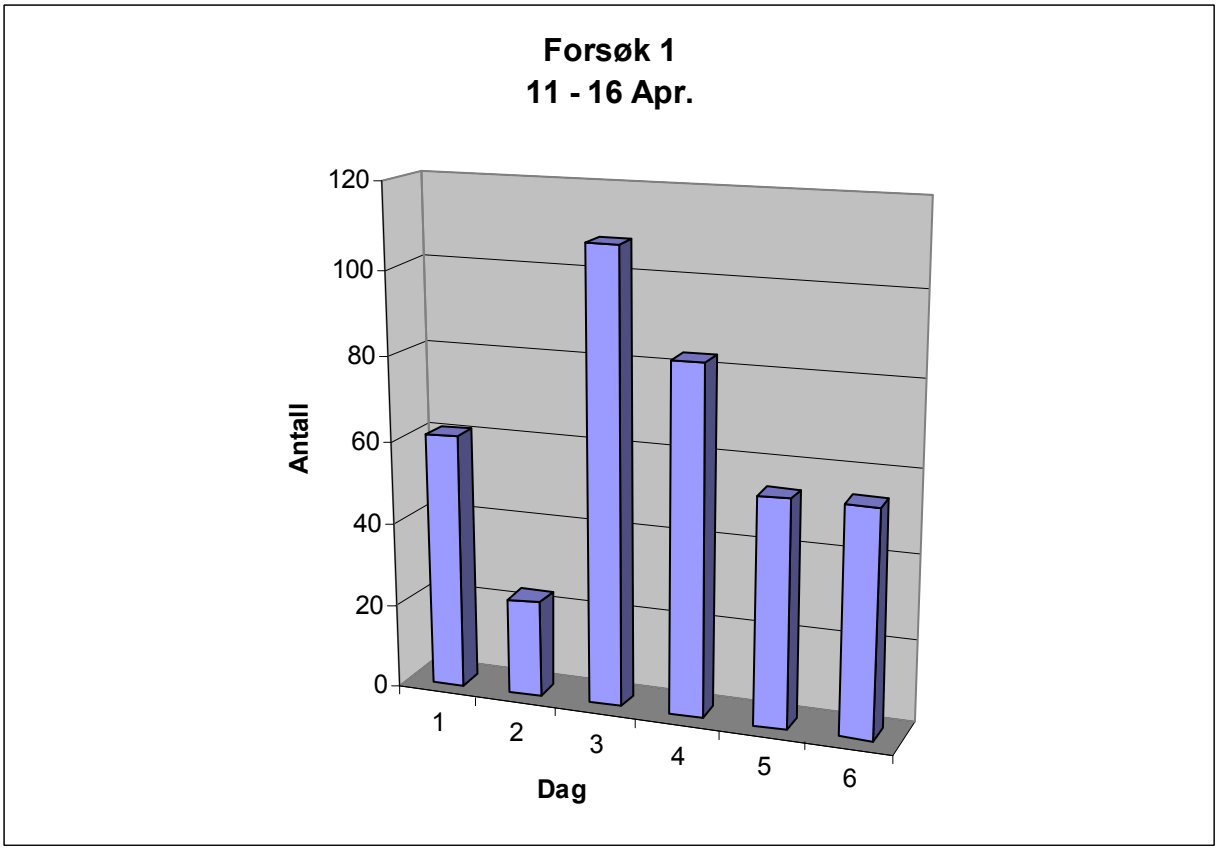
Videre ble det påvist forsøk på SSH trafikk mot Linux maskinen fra adresse 208.59.X.X. Dette kunne være et forsøk på å ta kontroll på maskinen, siden denne hadde blitt kompromittert tidligere, men her ble det forsøkt med flere IP adresser, så vi antar at dette var en form for skanning.

Som en kuriositet ble det klokken 00:40 den 13. April observert mistenkelig trafikk fra 128.39.X.X (skolens hybelnett). Dette var aktivitet på protokoller som: SMB, ICMP, TCP, NBSS, DCERCP, NBNS. Dette vil mest sannsynlig være en maskin som er infisert av et virus/orm. Vi antar dette var NIMDA, da denne personen har hatt dette før.

Det siste av interesse å finne i disse loggene var fra 05:05 den 13. April. Her ble det sendt SYN fra 194.183.X.X, igjen mot Linux maskinen (som fremdeles var nede). Det interessante her var at både sender og mottakers port var 21, noe som er litt rart. Dette kan indikere et forsøk på en oppkobling fra en ftp til en annen (server til server), men dette vil bare være en observasjon. Vi kan ikke si dette

med noen som helst sikkerhet.

Ingen varsling av de involverte har blitt gjort og vil heller ikke bli gjort av oss. I tillegg: For mer utfyllende informasjon om angrepene vises det til loggfiler som er vedlagt.



Figur 4 – Antall forbindelser mot Honeynett under første forsøk

6.1.4 Konklusjon

Det må nevnes at Honeynettet kanskje i ettertid burde ha fått mer tid online før vi tok det ned. Særlig ville det vært interessant å finne ut hva angriperne egentlig ville ha gjort med Linux maskinen hvis de fikk sjansen til å gå videre med angrepene. Likevel er det en del begrensninger vi må gjøre for i det hele tatt få tid til de forsøk vi ønsker å gjøre. Dette var tross alt første forsøk av en rekke og vi kan da trekke noen konklusjoner, selv om materialet kunne vært rikere.

For det første tyder aktivitetsloggene på at det ikke er riktig å anta at det tar en viss tid før folk der ute på Internett finner nye maskiner etter hvert som de kommer online. Faktum er at mengden skanninger der ute, samt at det som oftest skannes hele serier av adresser, gjør at en ny maskin vil bli oppdaget nesten øyeblikkelig. Hvis en skal sikre seg burde dette settes opp før en går på nett og ikke i ettertid.

For det tredje tyder ting ved første øyekast (vi kan ikke si at vi har nok data enda til å si ting med større sikkerhet) på at mesteparten av den trafikk en er utsatt for ikke nødvendigvis kommer fra virkelige mennesker, men fra virus, ormer og lignende. Det meste er ormer som blindt skanner og prober nettet for så å rapportere sine funn et sted, eller selv å gjennomføre standardiserte angrep på kjente svakheter. I tillegg virker det som om et angrep følger etter en viss tid med skanning, noe som virker logisk da dette sparer angriperen for tid ved ikke å bruke ressurser på maskiner som ikke

så lett kan taes over. Det finnes alltid enklere maskiner å ta og folk velger vanligvis det enkleste.

Til slutt viste forsøket oss noe litt uventet: En standard Linux-installasjon er ikke nødvendigvis mer sikker enn en standard Windows installasjon. Selv om vi hadde både en Windows og en Linux-maskin på nett med standard installasjoner, var det Linux-maskinen som hadde blitt tatt. Windows-maskinen hadde ingen vist noen særlig interesse for. Dette var noe vi ikke hadde ventet oss. Nok en myte avlivet, men det må nevnes at Linux maskinen kjørte flere og mer varierte tjenester enn Windows-XP maskinen. Ved installering av en typisk Linux klient maskin kunne resultatet blitt annerledes, eventuelt installere flere tjenester på Windows maskinen.

6.2 Forsøk 2 - Videreføring av passivt Honeynett

Dette er forsøk 2 og er en naturlig videreføring av forsøk 1. Vi fører nå Honeynettet ett steg videre ved å teste ut tilleggstjenester, ikke bare rene default installasjoner.

6.2.1 Beskrivelse

Det første forsøket var over raskt da den ene Linux maskinen vår ble tatt på få timer. Den ene XP maskinen vår fortsatte å stå oppe og gå alene, men fortsatt med en standard installasjon. Etter noen dager satt vi opp Linux maskinen igjen(dette var selvsagt en kopi før kompromittering), denne gangen uten ftp tjenesten som ble kompromittert. Alle andre tjenester kjørte fremdeles som når den var nyinstallert.

Siden XP maskinen vår ikke hadde mottatt noen interessante angrep, dersom man ser bort i fra automatiske ormer, så bestemte vi oss for å gjøre den mer attraktiv. IIS 5.1 ble installert, med Web og FTP-server. Ingenting ble forandret på, så vi hadde nå 1 Web server og en FTP server kjørendes på XP maskinen. Det viste seg at FTP serveren ble installert med anonym bruker som default. Dette ble heller ikke gjort noen forsøk på å endre.

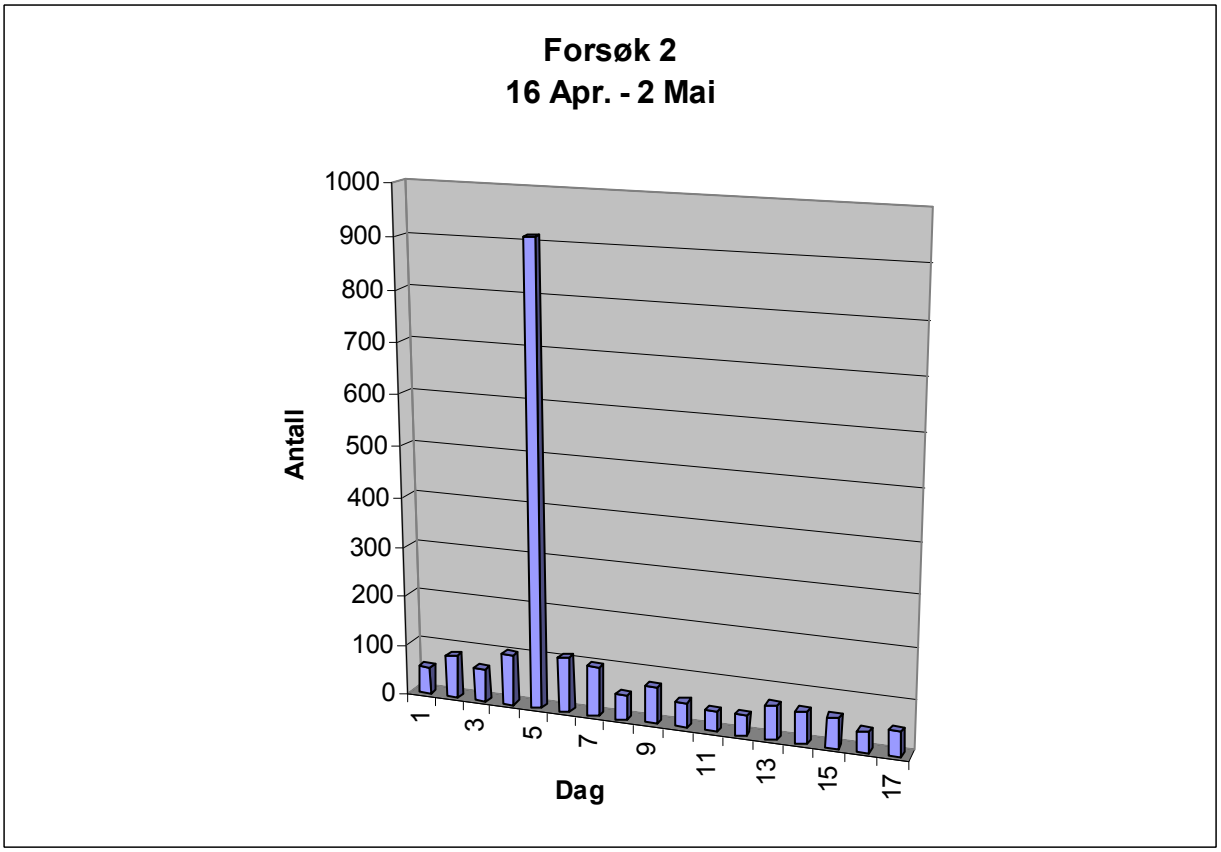
6.2.2 Forventninger

Siden vi hadde suksess med Linux maskinen som ble tatt forventet vi at denne maskinen inneholdt flere sårbare tjenester som kunne bli utnyttet til en angripers fordel. XP maskinen var helt standard innstall som nevnt og hadde nå en Web server kjørende. Alt dette tatt i betraktning forventet vi at begge maskinene ville bli infisert med automatiske ormer, eller at en angriper manuelt ville gå til angrep på dem.

IIS har en lang historie med angrep fra ormer og særlig Nimda viruset herjet på Internett for en stund siden. Det er fremdeles i omløp, men antall webservere og maskiner som er infisert er gått betraktelig ned. Webserver 5.1 har feilfiks for Nimda, men vi forventet at det var andre ukjente ormer som også ville prøve seg. Likevel var vi litt bekymret for at det ikke ville skje noe med XP maskinen. Det kan virke som Microsoft har gjort en god jobb med sikkerheten i Windows XP, og de har lært av feilene de har gjort tidligere.

6.2.3 Resultat

Dagene gikk og vi sjekket loggene jevnlig. Det viste seg at ingen alvorlige angrep kom. Det var nok av forsøk på portskanning og automatiske ormer. En spesiell adresse klarte å oppnå 913 hendelser i brannmurloggene. Dette var bare rettet mot XP-maskinen og over et tidsrom på ca. 90 sekunder. Hva dette skyldes er uklart. Alle var rettet mot port 80, dvs Web-server.



Figur 5 – Antall forbindelser mot Honeynett under andre forsøk

At alle angrep var automatiske var helt tydelig når typiske Windows exploits kom gjentatte ganger på Apache[40] web server som kjører på Linux plattformen. Etter hvert dukket det også opp noen interessante angrep som kan være fra angripere i loggene. Det viste seg at noen hadde vært inne på XP maskinen. Angriperen logget på som anonym bruker og prøvde å eksekvere kommandoer, dette mislyktes på grunn av at angriperen ikke hadde rettigheter til å utføre disse.

6.2.4 Konklusjon

Andre forsøk med oppsett av passivt Honeynett var ikke så vellykket som det første. Det viste seg at vi hadde nok av attraktive tjenester kjørende, men ingen av disse var satt opp på en slik måte at noen av angriperne kunne benytte seg av svakhetene. Dersom vi hadde installert en eldre versjon av IIS så hadde vi mest sannsynlig fått masse infisering av automatiske ormer, men dette var ikke formålet med dette forsøket. Selv om ikke resultatet ble helt som vi hadde tenkt oss, fikk vi bekreftet at installasjon av nyere operativsystem og tjenester forhindrer de mest vanligste angrep.

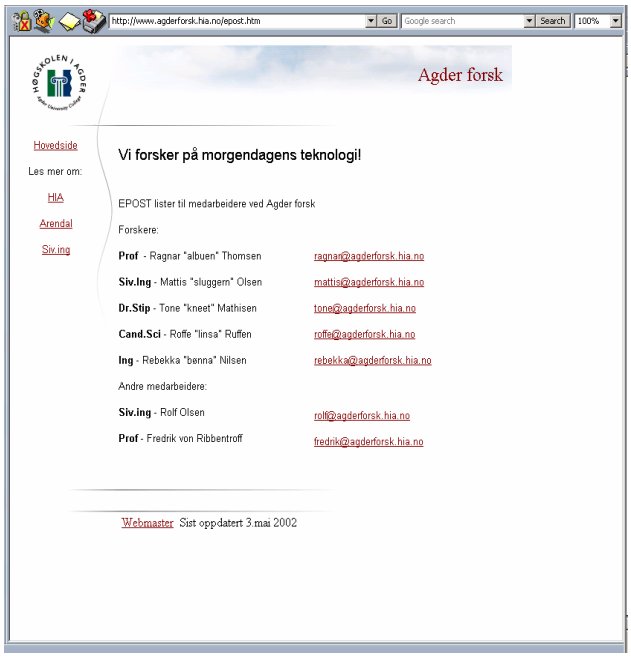
6.3 Forsøk 3 - Aktivt forsøk for å tiltrekke trafikk

Nok et skritt i retning for å tiltrekke ønsket oppmerksomhet mot Honeynettet vårt. Denne gangen i en mer aggressiv rolle.

6.3.1 Beskrivelse

Etter å ha kjørt et passivt Honeynett over en lang periode uten å ha fått noen interessante resultater var tiden inne til å gå mer aktivt ut for å forsøke å tiltrekke seg trafikk. Vi hadde fremdeles 2 maskiner kjørende i Honeynettet vårt, her regner vi ikke med loggserveren. Siden tidligere forsøk hadde både gitt positive og negative resultater bestemte vi oss for å prøve å tiltrekke trafikk på en mer aktiv måte. Vi kjørte opp en tredje maskin i Honeynett, en Windows XP maskin. På denne installerte vi ICQ, Mirc og Kazaa. Denne skulle brukes som ”surfemaskin”. I tillegg installerte vi en mailserver på den andre XP maskinen og lagde relevante web sider som vi la ut på Apache web server. På denne siden la vi også ut e-post til fiktive ansatte ved Agderforsk HIA. Litt humor ble brukt ved oppsett av siden, dette for seriøse folk som virkelig er ute etter informasjon ikke skal bli feilledet. De lurer de som bare er ute etter å ødelegge (forhåpentligvis).

Figur 6 – Fiktiv hjemmeside



Figur 7 – E-postliste

Ved å legge ut websidene så eksponerer vi oss på en måte, da web roboter kan finne sidene og registrere dem i søkemotorene sine, men vi valgte også å registrere sidene på forskjellige søkemotorer manuelt. Formålet med web sidene og e-post adressene var å finne ut om eksponering av informasjon ville føre til mer trafikk på Honeynettet, men også om det ville ta lang tid før vi fikk spam e-post. Dette var den mer passive delen av dette forsøket.

For den mer aktive delen brukte vi maskinen med ICQ, Mirc og Kazaa. Disse brukte vi aktivt til å besøke diverse hacker og warez kanaler på nettet. Vi surfet også mye på sikkerhetsrelaterte og undergrunnssider. Forskjellige tjenester ble også testet ut ved at vi registrerte oss med brukeren ”Ragnar Thomsen”. Hele tiden eksponerte vi e-post adressen for de som ville ha tak i den. Vi gjorde ingen forsøk på å skjule hvilken IP eller domene vi var fra og disse forsøkene ble gjort over en 1-2 ukers periode annenhver dag.

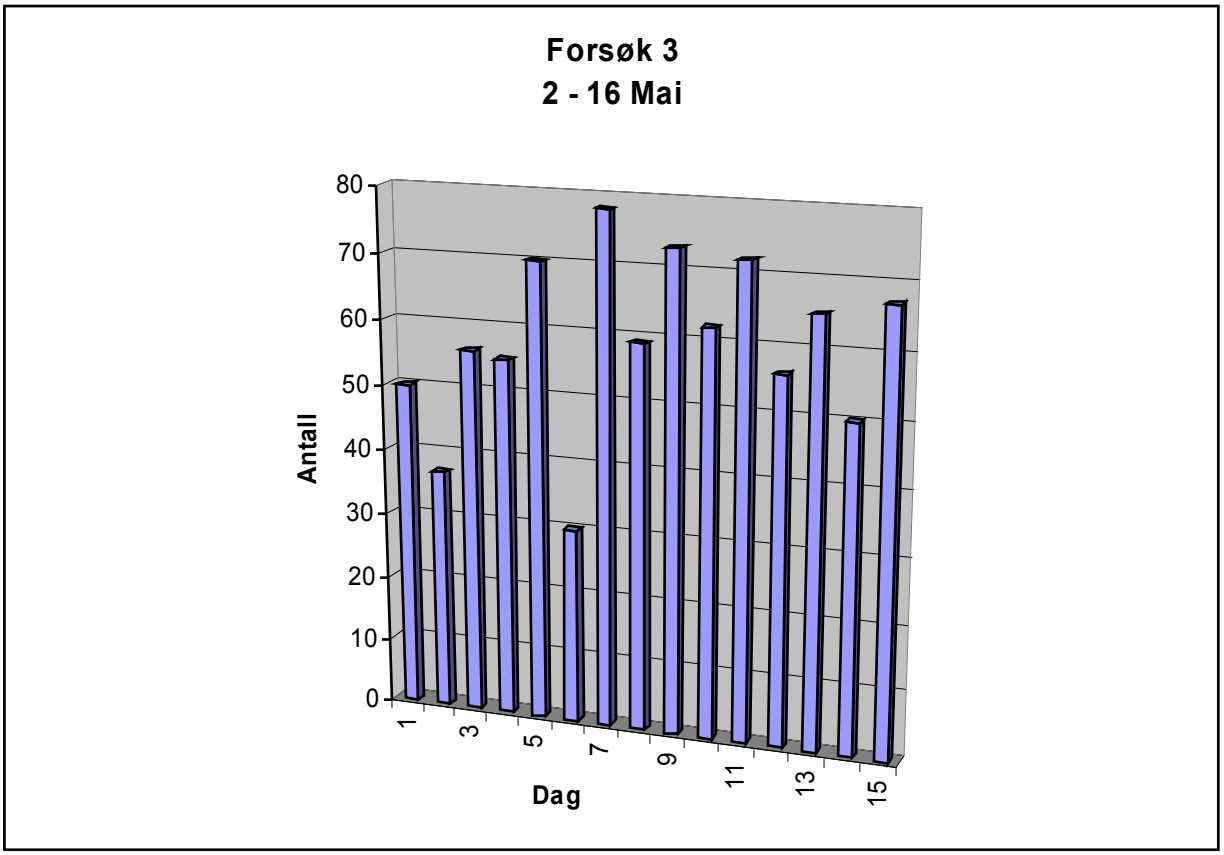
6.3.2 Forventninger

Forventningene var store framfor dette forsøket. Vi regnet med at vi ville motta masse spam e-post,

og at folk ville skanne IP adressen vår på IRC og bruke denne til å utforske Honeynettet videre. Siden angripere som oftest bruker andre maskiner enn deres egen til å ta maskiner med, så er det selvsagt vanskelig å si direkte at akkurat "den" handlingen førte til mer trafikk. Vi måtte ta utgangspunkt i trafikkmengden og se om vi fikk andre angrep enn de typiske ormeangrepene som vi pleide å få. ICQ brukeren var heller passiv, her registrerte vi jo bare en bruker, så vi forventet bare å motta spam meldinger. Det var selve web surfinga vi var veldig optimistiske rundt. Det at vi la igjen elektroniske spor på alle sidene vi besøkte håpte vi kunne føre til interesse hos de typiske cracker og hacker relaterte sidene. Vi var for eksempel inne på sider som crackere og hackere bruker som informasjonskilder om sikkerhetshull og feil i programmer som kan utnyttes. Dette inkluderer grupper som lager exploits som brukes av scriptkiddies til angrep.

6.3.3 Resultat

Vi så ikke noen betydelig trafikkøkning på Honeynettet, og ingen alvorlige angrep ble observert. Alt vi hadde i denne perioden var typisk portskanning og ormer. Det må likevel nevnes at det ble observert flere portskanninger enn vanlig til tider, dette kan være tilfeldig for vi har hatt masse portskanning i perioder før, men det kan også være et tegn på at tiltrekkingen har hatt en virkning. Forøvrig fikk vi ikke noen epost-spam og ICQ meldinger begrenset seg også til en enkelt melding rett etter installasjon.



Figur 8 – Antall forbindelser mot Honeynett under tredje forsøk

6.3.4 Konklusjon

En kan vel si at forsøket ikke var helt som vi hadde forventet, da vi faktisk hadde trodd vi ville få mer tiltrekking ved å gå aktivt ut. Grunnen kan være at vi ikke gjorde det lenge nok, grunnet tidsmangel, men kan også være at farene ved å bruke nettet aktivt ikke er så store. Forsøket skulle

helt klart være kjørt lenger, da det er for kort tid å ta en endelig konklusjon etter 1-2 uker, men det kan gi en pekepinn på hvordan bruk av Internett utsetter nettet ditt for farer. Vi kunne dessuten ikke ha forbindelsen oppe hele tiden, for denne trafikken genererer altfor mye støy i Honeynettet, så all logging som skjer i denne perioden må sjekkes nøye.

6.4 Forsøk 4 - Oppsett av eldre distribusjon

Her vil vi gå gjennom oppsett av en litt eldre distribusjon til Redhat og se om denne trekker til seg flere angripere enn nyere oppsett.

6.4.1 Beskrivelse

Redhat 6.2 er en eldre distribusjon, men er fremdeles i bruk over store deler av serverparken på Internett. Denne distribusjonen har mange velkjente sikkerhetshull, og forsøk med oppsett av disse maskiner som honeypott har gitt gode resultater for de som har testet dette ut[16]. Dette blir nest siste forsøket i diplomoppgaven. Installasjonen vil være en default installasjon som server.

6.4.2 Forventninger

Siden Redhat 6.2 har en del kjente sikkerhetshull kan vi til en viss grad forutse hva som kommer til å skje. En mulighet er at den vil bli tatt av en automatisk orm som kjører exploit og kanskje installere en IRC bot på maskinen. Redhat 6.2 har flere sikkerhetshull, men de mest kjente er feil i WU-ftpd og RPC (Remote Procedure Call) tjenester. RPC inkluderer tjenester som rstatd, ypbind og mountd. Denne maskinen blir helt klart det letteste byttet i Honeynettet vårt til nå, men det er interessant å se på hva en litt eldre distribusjon trekker til seg av trafikk og angripere.

6.4.3 Resultat

Vi hadde mistanke om at denne maskinen ville bli tatt veldig raskt, da dette var en eldre versjon av Redhat og vi hadde lest om andre honeypott forsøk som hadde vært vellykket. Når vi sjekket loggen på ACID dagen etter den ble satt opp så vi at den var tatt 2 ganger på samme kvelden. Begge gangene ble det benyttet en expolit i WU-ftpd. De to angriperne som hadde vært inne oppførte seg helt forskjellig og vi fikk ikke en enkel oversikt over hva de hadde tenkt å bruke datamaskinen til. Grunnen til dette kunne være at de byttet til kryptert forbindelse for fort, men vi fikk en oversikt over hva de lastet ned før loggingen stoppet opp. Den første angriperen lastet ned et rootkit. Dette kan tyde på at formålet med å ta over maskinen var å bruke denne videre i angrep. Andre angriperen lastet ned programmer for å drive skanning og flooding. Dette er klare indikasjoner på at formålet til han/hun også kan ha vært å bruke maskinen til angrep på andre maskiner på Internett.

6.4.4 Konklusjon

Dette forsøket viste seg å være svært vellykket, men vi savnet litt mer logging, da vi ikke fikk entydig informasjon om hva angriperene hadde som formål å bruke maskinen til. Det at maskinen ble tatt med WU-ftpd akkurat som i forsøk 1. viser at dette tydeligvis er en svært kjent svakhet i de Redhat versjoner vi har testet ut. Vi fikk likevel ikke bekreftet at denne maskinen trakk til seg flere angripere enn en nyere versjon da vi fikk akkurat like mange angrep som i forsøk 1. Tidsperioden er selvsagt altfor liten til å trekke sånne konklusjoner. I tidligere forsøk der vi har hatt enten angrep eller forsøk på angrep har vi sett at det først har kommet portskanning, deretter angrep. Her

observerte vi ikke noen portskanning før angrepene kom. Hva dette kommer av er uklart. En grunn kan være at IDS-sensoren ikke plukket dem opp, en annen kan være at vi har eksponert oss såpass at noen gikk direkte mot nettet vårt og testet exploits, selv om vi tviler sterkt på dette. Det kan også tenkes at det her er snakk om folk som har vært innom tidligere, uten at vi kan si noe sikkert (snakk om andre adresser her).

Vi hadde også såpass mange mangler ved loggingen at vi bestemte oss for å gjøre et siste forsøk og utvide loggingen til også å gjelde falsk syslogg og shell-logging.

6.5 Forsøk 5 - Ny distribusjon med utvidet loggmulighet

Her utfører vi det siste forsøket vårt før Honeynettet blir tatt ned.

6.5.1 Beskrivelse

I forsøk 1. fikk vi en del informasjon om hva angriperen hadde tenkt å bruke maskinen til da vedkommende lastet ned en IRC-bot, men forsøk 4 viste oss at kommandolinje logging er en nødvendighet sammen med en falsk syslogg for å avsløre hva angriper gjør dersom han/hun bytter over til kryptert forbindelse. Siden dette mest sannsynlig skjedde i forsøk 4, utvidet vi Redhat 7.2 maskinen med falsk syslogg og shell-logging for å finne ut om dette førte til at vi ville få mer kunnskaper om hva angriper hadde i tankene når maskinen ble angrepet. WU-ftpd ble startet opp da forsøk 1. og 4. viste oss at denne tjenesten var ettertraktet som exploit.

6.5.2 Forventninger

Siden dette forsøket var nesten identisk med forsøk 1, regnet vi med at det ville forløpe akkurat på samme måte, men vi håpte at vi ville oppdage litt mer av hva angriperen gjorde med maskinen når den først var tatt.

6.5.3 Resultat

Resultatet ble akkurat som forventet. Maskinen ble tatt med WU-ftpd som exploit og root tilgang ble oppnådd. Akkurat samme program ble lastet ned som under forsøk 1. Det kan se ut som den første gangen byttet angriperen ut passord filer for å legge til en bruker som han/hun kan logge inn med og deretter skrive su for å få root tilgang. Exploiten logges fullt og helt av IDS sensoren, men når angriper bytter til kryptert forbindelse har vi bare fjernlogging av shell.

6.5.4 Konklusjon

Uten fjernlogging av shell kunne vi ikke funnet ut hva slags formål brukeren skulle benytte maskinen til, dette pga nedlasting av rootkit som inneholder IRC bot (som så senere kan beordres sentralt til å videre angripe andre) gjøres over kryptert forbindelse. Vi kan altså konkludere med at vi ikke oppnådde noe mer nyttig informasjon enn vi gjorde fra forsøk 1 og 4, men uten ekstra logging ville vi ikke fått denne informasjonen på forsøk 5. Det er tydeligvis stor forskjell på hvordan eventuelle angripere operer, så full logging bør nok alltid være med for å være helt sikker på å få tak i all relativ informasjon.

6.6 Ting vi ville utført uten begrensinger tilstede

Hvis det var bare en ting vi kunne ha gjort annerledes må det være å ha mer tid til forsøkene. Forsøkene ble satt opp altfor raskt etter hverandre og dette kan vekke mistanke. Under forsøk 3 var formålet å eksponere nettet aktivt, og dette ble gjort over en 2 ukers periode, igjen altfor kort tid. Resultatene kunne blitt annerledes med bedre tid, men det er ikke sikkert. Det hadde også vært ønskelig å kunne gå mer aktivt ut ved å framprovosere angrep fra angripere som har litt mer kunnskaper enn de automatiserte exploits som angriperne våre benyttet.

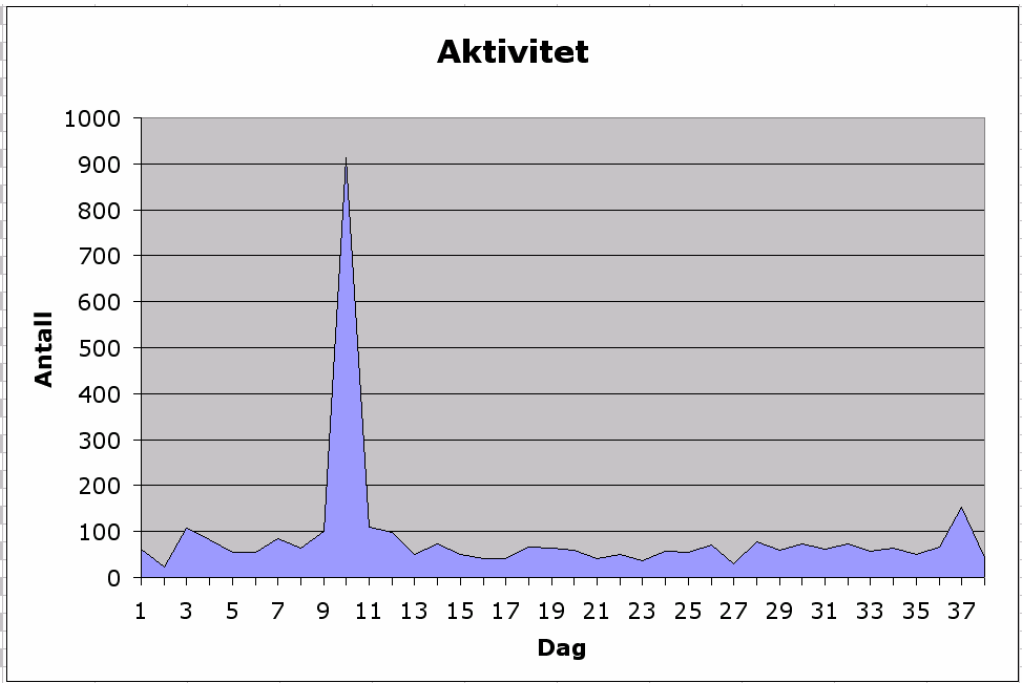
Under alle forsøkene benyttet vi oss av en 5-6 IP'er som vi hadde fått tildelt. Dersom noen fulgte med på oss under gjennomføringen av alle disse forsøkene ville vi nok avslørt at vi hadde et Honeynett og ikke et vanlig nett. Vi hadde tilgjengelig noen flere IP'er, men alle disse var i samme serie, og ville avslørt oss uansett. Bytte av domene kunne også vært ønskelig.

VMware maskinen vår har fungert tilfredsstillende, men det hadde helt klart vært en fordel med mer RAM og mer diskplass. Optimalt burde hver virtuelle maskin ha en harddisk hver og 128MB med RAM eller mer. En PIII prosessor med 32 MB ram vekker mistanke. Vi så at noen kjørte kommandoer for å finne ut hvilken prosessor vi hadde. I tillegg er det ikke lurt å installere VMware tools(grafikkdrivere til VMware), siden disse kommer opp som system prosesser. Man kan da lett bli avslørt for å kjøre et Honeynett

DNS og navngiving: Hvis vi hadde hatt full kontroll på domenet (noe vi ikke hadde under prosjektet), kunne vi ha byttet navn mellom hvert forsøk, noe som kanskje kunne vært med på å forhindre overlapp i forsøksdata mellom forsøk.

7 Konklusjon

Dag01	11.apr	61
Dag02	12.apr	23
Dag03	13.apr	108
Dag04	14.apr	83
Dag05	15.apr	54
Dag06	16.apr	54
Dag07	17.apr	84
Dag08	18.apr	65
Dag09	19.apr	101
Dag10	20.apr	913
Dag11	21.apr	110
Dag12	22.apr	98
Dag13	23.apr	50
Dag14	24.apr	73
Dag15	25.apr	51
Dag16	26.apr	42
Dag17	27.apr	41
Dag18	28.apr	67
Dag19	29.apr	63
Dag20	30.apr	59
Dag21	01.mai	40
Dag22	02.mai	50
Dag23	03.mai	37
Dag24	04.mai	56
Dag25	05.mai	55
Dag26	06.mai	70
Dag27	07.mai	30
Dag28	08.mai	78
Dag29	09.mai	59
Dag30	10.mai	73
Dag31	11.mai	62
Dag32	12.mai	72
Dag33	13.mai	56
Dag34	14.mai	65
Dag35	15.mai	50
Dag36	16.mai	67
Dag37	17.mai	152
Dag38	18.mai	44



Figur 9 – Oversikt over forbindelser mot Honeynett under hele forsøksperioden

Etter å ha gjennomført 5 forsøk og brukt Honeynettet til å prøve å trekke til seg trafikk, sitter vi igjen med en del svar. Det viste seg etter gjentatte forsøk at å trekke til seg trafikk ikke var så lett. Vi kan ikke entydig si at lokkingen har hatt noen effekt som grafen over viser. Trafikken var jevn gjennom hele forsøket, uten om en spesiell hendelse under forsøk 2 (som kan observeres på graf, dag 10). Det mest effektive var som vi fant ut i forsøkene, å installere tjenester på maskinene som vi vet er attraktive. Vi kan ikke si at aktiviteten endres, men sjansen øker betraktelig ved å installere maskiner vi vet er sårbare. Etter kort tid vil eventuelle angripere finne deg uansett hvor lite eller mye du annonserer lokasjon.

Under har vi en liste over unike angrep som ble logget på IDS under forsøksperioden. Som vi kan se er ormangrep dominerendes. Selv om man ikke ønsker denne type trafikk, må man ofte sortere denne ut i fra den mer interessante trafikken. Dette er viktig å tenke på ved oppsett av Honeynett.

Attempted-recon	138
Bad-unknown	14
Misc-activity	33
Portscans	84
Shellcode-detect	13
Unclassified	23
Web-application-attack	978

Motivet til angriperne våre viste seg, så langt vi overvåket de, å være basert på å ta over maskiner for å bruke dem i videre angrep senere. Eksempelvis DOS/DDOS eller som et springbrett for å ta over andre maskiner. Problemet vårt var at tidsaspektet gjorde det vanskelig for oss å finne noen generelle motiv. Med bedre tid kunne vi nok fått en bedre oversikt over profiler på angripere.

Den eneste sikre konklusjonen vi kan gi er at det tydeligvis eksisterer veldig mange angripere på Internett, og ingen, selv ikke hjemmebrukere, kan føle seg sikre. Selv ved å ha den nyeste programvaren er det viktig å ha kunnskap om oppsett av maskiner. Alle forsøk på angrep virket som kom helt uavhengig av hvilke type maskiner vi hadde i nettverket. Det kan virke som Linux maskiner er mer utsatte for typiske exploits. De automatiske ormene virket som de var myntet på Windows maskiner. Dette er selvsagt spekulasjoner. Til slutt må vi vel si at forsøkene var delvis vellykkete tiden tatt i betraktning, men det er likevel en del spørsmål som står ubesvart. En ting vi derimot fikk vist (gjentatte ganger), var at WU-ftpd er en svært populær exploit.

8 Referanser

1.	Accessing Virtual Hard Disks Outside of GSX Server, VMware, [2002], Online: http://www.vmware.com/support/gsx/doc/loopback_gsx_Linux.html
2.	Ad-aware, [2002], Online: http://www.lavasoft.nu/
3.	Adware, Techtarget, [2002], Online: http://whatis.techtarget.com/definition/0,289893,sid9_gci521293,00.html
4.	Analysis Console for Intrusion Databases, Roman Danyliw, [2002], Online: http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html
5.	AOL messenger, [2002], Online: http://quicken.aol.com/
6.	Bearshare, [2002], Online: http://www.bearshare.com/
7.	Complete Reference Guide to Creating a Remote Log Server Remote syslog, Eric Hines, [14. Aug. 2000], Online: http://www.Linuxsecurity.com/feature_stories/remote_logserver-1.html
8.	Cult of the dead cow, cDc communications, [2002], Online: http://www.cultdeadcow.com/
9.	Debian, [2002], Online: http://www.debian.org
10.	EventReporter - The NT Event Monitor, [2002] http://www.eventreporter.com/en/
11.	Hacker, Wikipedia, the free encyclopedia, [last upd. 25 Mai, 2002], Online: http://www.wikipedia.com/wiki/hacker
12.	HIA, [2002], Online: http://www.hia.no
13.	Honeypotting with VMware – basics, Kurt Seifried, [last upd. 15 Febr. 2002], Online: http://www.seifried.org/security/ids/20020107-Honeypot-vmware-basics.html
14.	Honeypots,Definitions and Value of Honeypots, Lance Spitzner, [last upd. Mai, 2002], Online: http://www.enteract.com/~lspitz/Honeypot.html
15.	ICQ, [2002], Online: http://www.icq.com/
16.	Incident Analysis of a Compromised RedHat Linux 6.2 Honeypot, Stephen Holcroft, [April 2002], Online: http://www.lucidic.net/whitepapers/sholcroft-4.1-2002.html
17.	Intrusion Detection FAQ, What is a Honeypot? Honey Pot Systems Explained, Loras R. Even, [12. Juli, 2000], Online: http://www.sans.org/newlook/resources/IDFAQ/Honeypot3.htm
18.	IRC proxy An Introduction to psyBNC 2.2.1 tutorial, jestrix, [2002], Online: http://www.jestrix.net/tuts/psy.html
19.	Kazaa, [2002], Online: http://www.kazaa.com
20.	Know Your Enemy – Motives, Project Honeynet, [Juni, 2002], Online: http://project.honeynet.org/papers/motives/
21.	Know Your Enemy, Project Honeynet, [11. Mai 2002], Online: http://project.Honeynet.org/papers/honeynet/
22.	Link til patch for bash2.03, Project Honeynet, [2002],

	Online: http://project.honeynet.org/papers/honeynet/bash.patch
23.	Link til patch for bash2.05, Luciano's personal Homepage, Luciano Rocha, [2002], Online: http://strange.nsk.yi.org/
24.	Messenger, [2002], Online: http://messenger.msn.com/
25.	Microsoft IIS Unicode Exploit: Nate Miller, Lucent Technologies Worldwide Services, [August 2001], Online: http://www.lucent.com/livelink/197020_Whitepaper.pdf
26.	Microsoft, [2002], Online: http://www.microsoft.com
27.	Mirc, [2002], Online: http://www.mirc.com
28.	Netbus Analysis, Alexey Podrezov & Mikko Hypponen, F-Secure Corp, [1998-2001], Online: http://www.europe.f-secure.com/v-descs/netbus.shtml
29.	Nimda Analysis, K. Tocheva, G. Erdelyi, A. Podrezov, S. Rautiainen & M. Hypponen; F-Secure Corp, [September 18-19, 2001], Online: http://www.europe.f-secure.com/v-descs/nimda.shtml
30.	Project Honeynet, [2002], Online: http://project.honeynet.org/
31.	Proseq, [2002], Online: http://www.proseq.no
32.	rc.firewall, Rob McMillen & Michael Clark, [2001], Online: http://project.honeynet.org/papers/honeynet/rc.firewall
33.	Redhat, [2002], Online: http://www.redhat.com
34.	Setting up a Linux Log Server to enhance System Security, Chl0ie, [31. Aug 2000], Online: http://www.Linuxsecurity.com/feature_stories/logserver-2.html
35.	Shareware, Techtarget, [2002], Online: http://whatis.techtarget.com/definition/0,289893,sid9_gci212977,00.html
36.	Snort, [2002], Online: http://www.snort.org
37.	Specter, [2002], Online: http://www.specter.com
38.	Spyware, Techtarget, [2002], Online: http://whatis.techtarget.com/definition/0,289893,sid9_gci214518,00.html
39.	Symantec pcAnywhere, Symantec [2002], Online: http://enterprisesecurity.symantec.com/products/products.cfm?productID=2
40.	The Apache Software Foundation, [2002] http://www.apache.org/
41.	The Deception Toolkit, [2002], Online: http://www.all.net/dtk
42.	The Ethereal Network Analyzer, [2002] http://www.ethereal.com/
43.	The Strange Tale of The Denial of Service Attack Against GRC.COM, Steve Gibson, Gibson Research Corporation, [last upd. 5 Mars 2002], http://grc.com/dos/grcdos.htm
44.	User mode Linux: The User-mode Linux Kernel Home Page, [2002], Online: http://user-mode-Linux.sourceforge.net/
45.	VMware, [2002], Online: www.vmware.com

46.	Windows NT Event Log explained, NtWak0, [12, Sept. 2000], Online: http://www.securiteam.com/windowsntfocus/5EP0E0K2KW.html
47.	WinMX, [2002], Online: http://www.winmx.com/

9 Stikkordsliste

ACID	Analysis Console for Intrusion Databases. Et PHP-basert verktøy for søk og behandling av databaseinformasjon fra IDS, brannmur og lignende.
Administrator	Øverste bruker på maskinen. Har alle rettigheter (se root)
ADSL	Asymmetric Digital Subscriber Line.
Adware	Programvare som benytter reklame som betalingsmetode.
Aktiv lokking	Her menes å aktivt prøve å trekke til seg trafikk til et Honeynett.
AOL	America Online. Stor nettleverandør I USA.
Apache	Kjent webserver.
Bakdør	Skult funksjonalitet i programmer.
Bash	Kommandolinjetolkeren i GNU.
Bearshare	Et peer-to-peer fildelingsprogram.
Bot	Et dataprogram som fungerer som en agent for en bruker eller et annet program.
Bouncer	Program for å sende trafikk videre.
Brannmur	Programvare for å begrense nettilgangen inn/ut fra en maskin. Brukes til å sikre en nettressurs.
Buffer overflow	Hendelse der variabler overskriver sitt avsatte område i minnet.
CD-ROM	Enhet for å lese CD-ROM – plater.
Codec	En algoritme eller et spesialisert program som oppgave har å minske størrelsen på datafiler (COmpression/DECompression).
Crack	Program som har blitt strippet for sikkerhetsmekanismer.
Cracker	Dataterrorist med onde hensikter.
Cygwin	Programpakke som tilbyr et Linux-miljø på Windows-plattformen.
Datatype	Fellesbetegnelse på variabler som angir typen.
DCERCP	Dataprotokoll?
DDOS	Distribuert Denial of Service. Distribuert form for DOS-angrep der en angriper fra mange steder samtidig og dermed øker datamengden.
Debian	Linuxvariant
DNS	Domain Name Server
DOS	Denial of Service. Angrep der man ved overbelastning forsøker å hindre en tjener å formidle en tjeneste.
E-post	Elektronisk post.
Eventlogger	Loggingsmekanisme for systemhendelser.
Exe	Med dette menes en eksekverbar fil, altså et program.
Exploit	Utnyttelse av en (kjent) svakhet i et system.
Finger	Tjeneste som muliggjør å finne ut info om hvem som er logget på en maskin.
Fjernadministrering	Kort sagt å styre, eller manipulere en maskin på avstand.
Flagg	En variabel som brukes til å angi en tilstand.
FTP	File Transfer Protocol. Dataoverføringsprotokoll som er mye brukt i datanett.
Hacker	Dataterrorist uten nødvendigvis onde hensikter.
Honeynett	Flere Honeypoter som til sammen utgjør et nett. Hele nettet vil da fungere som en Honeypot.
Honeypot	En ressurs som brukes til tiltrekning. Formålet er å studere det

	som tiltrekkes.
Hub	Et koblingspunkt i nettet der trafikk fra et tilkoblingspunkt i hub'en blir sendt ut til de andre (broadcast).
ICMP	Internet Control Message Protocol. Meldingsprotokoll som benyttes til meldingskontroll og feilrapportering ved bruk av IP.
ICQ	Program som benyttes blant annet til instant messaging, chat og filoverføring.
IDE	Integrated Drive Electronics. Standardisert interface mellom harddisker (ide) og hovedkortressurser.
IDS	Intrusion Detection System. System (software/hardware) som brukes til innbruddsoppdagelse og reaksjon i datasystemer.
IIS	Internet Information Services. Microsoft serverprogramvare for Web og FTP.
Instant Messaging	Kommunikasjonskonsept der meldinger mellom partene sendes og besvares i sanntid hvis mulig.
IP	Internet Protocol. Dataoverføringsprotokollen som benyttes i internett.
IP-serie	En gruppe IP-adresser.
Iptables	Program som brukes for å håndtere forbindelser inn/ut gjennom nettverket.
IRC	Internet Relay Chat. Kommunikasjonssystem som involverer regelsett og klient/server-programvare i nettverk som ofte spenner over flere servere.
ISP	Internet Service Provider.
IX	Fellesforkortelse for Unix/Linux systemer.
Kazaa	Peer-to-peer fildelingsprogram.
Kommandolinje	Brukergrensesnitt der en gir kommandoer ved å skrive disse inn klartekst.
Kryptering	Mekanisme for å forsøke å gjøre data uleselig for uvedkommende.
Linux	Gratis operativsystem.
Linuxkjerne	De innerste grunnfunksjonene (kjerne) i Linux operativsystemet.
Loggfiler	Filer som brukes til å lagrer systemhendelser.
Mailingsliste	Liste over e-post adresser som alle mottar samme e-post.
Mbit	Megabit. Benyttes for å angi mengde, her: million (1024*1024) bits.
Messenger	Microsoft program for å utveksle meldinger (instant messaging).
Mirc	IRC-klient for Windows.
Mp3	MPEG-1 Audio Layer-3. Et standardisert format for komprimering av audiosekvenser til svært små filer.
NAT	Network Address Translation. Oversetting av IP-adresser fra et nettverk til et annet.
NBNS	NetBIOS Name Server, WINS sitt egentlige navn.
NBSS	NetBIOS Session Service
Netbus	En trojan som gir andre mulighet til å fjernadministrere maskinen som er infisert.
Nimda	Automatisk orm som ble kjent vinteren 2001 for å infisere stort antall PC-er.
Node	En punkt i nettet.
Orm/Worm	Automatiserte programmer med forhåndspesifisert motiv.
OS	Operativsystem

Passiv lokking	Metode for å tiltrekke trafikk til et Honeynett uten å aktivt reklamere, eller kontakte noen selv.
Peer-to-peer	Klient til klient kommunikasjon, altså ingen server med i bildet.
Pentium3	Prosessorfamilie fra Intel.
Perl	Skriptspråk spesielt bra på strengoperasjoner.
Plugin	Programsnutt som kan kobles til et større program for å gi økt funksjonalitet.
Port	Port angir hvilken applikasjon som skal motta nettverksdata.
Portscan	Aktivitet som brukes til å undersøke hva en maskin reagerer på i nett. Dette gjøres ved forespørsler for hver port.
Proxy	En punkt i nettet som virker som et mellomledd mellom bruker på en arbeidsstasjon og Internett. Denne kan tilby sikkerhet.
Rack	Mange servere plassert kompakt i et skap.
RAM	Random Access Memory.
Rammeverk	Sett med funksjoner som gjør det enklere for programmerere å implementere ny funksjonalitet.
Redhat	Linux-variant.
Registeret	Registeret i Windows. Inneholder blant annet informasjon om systeminnstillinger, samt inststallerte program.
Relay	Videresende data uten mellombehandling.
Remote Loggserver	Loggserver som tar imot logging fra andre maskiner sentralt
Root	Brukerstatus på IX-systemer. Root har alle rettigheter på systemet.
Root aksess	Hvis noen oppnår root-aksess vil disse få alle rettigheter.
Rootkit	Programvareverktøy hvis oppgave er å gi brukeren root-aksess.
Router	Nettverksdel som kommuniserer på nettverkslaget laget. Står mellom forskjellige nett.
RPC	Remote Procedure Call
SAP	Firma som kom med orginal ide om å benytte felles databaser i et selskap og koble applikasjonene sammen med disse. Teknologien kalles ofte for SAP også.
Scriptkiddies	Betegnelse på unge hackere som benytter ferdige programverktøy for datakriminalitet.
SCSI	Small Computer System Interface. Et dataoverførings-grensesnitt for perifere enheter.
Serial	Ofte en passord-setning for aktivere et program slik at man kan utnytte alle muligheter som programmet tilbyr.
Server	Tilbyder av en tjeneste.
Shareware	Programvare som kan fritt kan benyttes privat, men betaling ønskes.
Shell	Se kommandolinje.
Skanning	Metode for å finne ut hvilke porter som maskiner har åpne og lytter på.
SMB	Server Message Block Protocol. En filoverføringsprotokoll
Snort	En open source IDS sensor.
Spam	Uadresserte meldinger som sendes ut i stort antall.
Spyware	Programmer som kjører i det skjulte for å avdekke ulike ting på maskinen din, ofte med hensikt å benytte disse til salgsøyemed.
SQL	Structured Query Language, et standard programmeringsspråk for å hente og legge til data til databaser,
SSH	Secure Socket Shell, et interface og protokoll for å få sikker

	overføring ved fjernadministreing av en datamaskin.
String	Datatype som brukes til å inneholde ord, eller setninger (en rekke tegn).
Svitsj	Nettverksenhet som opererer på linklaget.
SYN	Synkroniseringspakke i TCP
Syslog	En tjeneste som har som formål å samle all logging og sende disse til en loggserver eller til et medium for oppbevaring.
Tail	Program som kan brukes til å skrive ut siste endringer i en fil.
TCP	Transmission Control Protocol, brukes mye for å garantere overføring av datapakker sammen med IP
Tcp-dump	Program som skriver TCP aktivitet ned på disk, i filer som så senere kan undersøkes nærmere.
TP-RJ45-Cat5	Nettverkskabel som brukes til overføring av data.
Traceroute	Program som brukes til å undersøke veien/ruten en pakke tar i nettet.
Trojanere	Programmer som utgir seg for noe annet enn det de er.
UDP	User Datagram Protocol, en kommunikasjonsprotokoll.
Unicode	Koder for å representere bokstaver og tegn på.
USB	Universal Serial Bus, et plug and play interface mellom enheter og datamaskinen.
Vertsmaskin	Maskin som kjører den aktuelle tjenesten, eller deler fila som forespørres.
VertsOS	Operatisvsystemet som man kjører programmer under.
Virtuell Maskin	Ikke-fysisk maskin som kjøres i software (VMware).
Virtuelle disker	Diskfiler som emulerer en ekte harddisk
Virus definisjoner	Informasjon (database) om virustyper. Brukes av antivirusprogramvare for å kunne gjenkjenne flest mulig virus.
VMware	Programvare som tilbyr opprettelse av virtuelle maskiner kjørende på samme hardvare.
Warez	Betegnelse på filer en kan laste ned fra Internett. Brukes mest om piratkopierte programmer/spill.
Windows	Operativsystem fra Microsoft.
WinMX	Peer-to-peer fildelingsprogramvare.
Wu-ftpd	FTP serverprogram.

10 Vedlegg

10.1 HOWTO - Logging i Honeynett

Dette er en gjennomgang over hvordan man kan sette opp logging til et Honeynett. Her går vi kort gjennom hvordan vi satt opp loggserver, WindowsXP og Redhat Linux. Denne gangen mer teknisk forklart.

10.1.1 WindowsXP:

I Windows XP har en 3 logger: System log, Application log og Security log. Windows XP logger ikke alt som default så det er lurt å legge til litt mer logging[46]. Nettlinken under ble benyttet for å forandre loggmulighetene i WindowsXP. Det må nevnes at denne siden gjelder for NT så noen av lokasjonene er litt forskjellige. Loggingen ble begrenset til å forandre på "policies" under Administrative tools. Ikke noe mer ble gjort.

Link til forklaring til logging i NT:
http://www.securiteam.com/windowsntfocus/5EP0E0K2KW.html

10.1.2 Linux Redhat 7.2

Følgende må gjøres på klientsiden:

1. Sett opp loggserver til å motta fjernlogging[7, 34]:
Finn syslog PID og drep den:
```
[root@linux]#ps -aux | grep "syslogd"
```
Dersom PID er 1200 skriv:
```
[root@linux]#kill -9 1200
```
Start syslog med "remote" lagt til:
```
[root@linux]# syslogd -rm 0
```
Sjekk om syslogd ble startet opp med "remote".
```
[root@linux]#cat /var/log/messages
```
Dersom det står at loggserver ble startet med "remote reception" helt nederst er alt klart for logging.

2. Logging av bash:
Last ned bash 2.03[22] eller bruk 2.05[23] som redhat 7.2 blir innstallert med.
Dersom man velger å laste ned bash 2.03 og bruker patch på denne må man sette symlinks slik at når man starter et shell så sendes all data til bash2.03. Det samme må gjøres med default shell også, dersom man vil ha mulighet til å logge kommandoer fra andre shell på maskinen også.
Si at bash2.03 heter bash203, da setter man symlink fra andre shell til denne.
```
[root@linux]# ln -sf bash203 $LFS/bin/<navn på shell>
```

Link til patch for bash2.03: http://project.honeynet.org/papers/honeynet/bash.patch
Link til patch for bash2.05: http://strange.nsk.yi.org/

3. Oppsett av falsk syslogg:
Last ned syslogd som source RPM. I sourcen går man inn og forandrer loggfil lokasjon fra `/etc/syslog.conf` til noe man velger selv. Feks. `/etc/fakesyslog`. Poenget er å velge noe slik at en inntrenger ikke oppdager den så lett. Ett å ha kjørt `make` legger man også binær filen man fikk et hemmelig sted. Før man starter opp den falske sysloggen lager man en config fil som den skal lese fra. Dette er en vanlig tekst fil.

Dette legges til i filen:

```
*.*         @X.X.X.X
```

Husk også å stoppe sysloggen som kjører fra før:

```
[root@linux]# /etc/init.d/syslog stop
```

Start så den falske sysloggen og vi har en skjult syslogg som en eventuell inntrenger skal ha problemer med å oppdage ved første øyekast.

Denne guiden virker på alle Linux distribusjoner, men lokasjon til filer kan være forskjellig. Denne virker på Redhat 7.2.

Programmer brukt til logging ligger vedlagt på CD-rom.

10.2 Brannmur script

Dette skriptet ligger vedlagt på CD-rom.

10.3 Exploit script brukt

De skript vi fikk tak i ligger vedlagt på CD-rom

10.4 TCPdump

Disse ligger vedlagt på CD-rom

10.5 Loggfiler

Disse ligger vedlagt på CD-rom

10.6 Orginal oppgavebeskrivelse

Datasikkerhetsfirmaet Proseq ser for seg å benytte såkalte "Honey-nett" i arbeidet med å få økt datasikkerhet for sine kunder og dem selv.

Målsettingen med denne oppgaven er å trekke dette konseptet ett steg videre. Oppgaven skal fokusere på følgende: Forske på teknikker for å tiltrekke trafikk, se på psykologien bak hvorfor noen servere velges framfor andre og se på hvordan det jeg gjør på nettet kan påvirke trusselnivået. For å få gjennomført prosjektet, må man bygge en prototyp på et "Honeynett" og utføre tester med forskjellige tjenester. Dette inkluderer både velkjente tjenester med eksisterende sårbarheter og nye. Deretter skal man prøve å tiltrekke trafikk til disse tjenestene.

Målsetningen går mer i retning av trusselanalyse av nye og eksisterende tjenester enn de tekniske aspektene med å få opp Honeynettet. Dette er gjort før av andre og kan dermed trekkes erfaringer av.

10.7 TCPstreams og loggfiler brukt i forsøk

Alle klokkeslett regnes ut i fra UTC og beregnes ut ifra når TCP strømmen begynte og når den stoppet. Under forsøk 5 baserte vi oss på siste kommando som ble skrevet i kommandolinje. 2 timer må legges til ved beregning til norsk tid. Vedlegg ligger på CD-rom.

10.7.1 – Første forsøk

128.39.x.x (WUftpd exploit)

Angrep begynte: Apr 11 – 2002 – 20:55:39
Angrep sluttet: Apr 11 – 2002 – 21:17:39

65.31.x.x (WUftpd exploit)
Angrep begynte: Apr 12 – 2002 – 07:34:50
Angrep sluttet: Apr 12 – 2002 – 07:55:32

10.7.2 – Andre forsøk

24.205.x.x (Orm)
Angrep begynte: Apr 20 – 2002 – 11:04:29
Angrep sluttet: Apr 20 – 2002 – 11:05:52

10.7.3 – Fjerde forsøk

202.181.x.x (WUftpd exploit)
Angrep begynte: Mai 16 – 2002 – 20:38:39
Angrep sluttet: Mai 16 – 2002 – 20:46:21

203.198.x.x (WUftpd exploit)
Angrep begynte: Mai 17 – 2002 – 00:35:36
Angrep sluttet: Mai 17 – 2002 – 00:51:22

10.7.4 – Femte forsøk

200.203.x.x (WUftpd exploit)
Angrep begynte: Mai 18 – 2002 – 13:43:21
Angrep sluttet: Mai 18 – 2002 – 14:42:51

Redigerte TCP-streams ligger vedlagt på slutten av rapporten og på CD-rom. De på papir er med redigerte IP adresser.