Intrusion

Part 1 of chapter 9

Frederik Alkærsig & Henrik Holmgren-Jensen

Intrusion

Er defineret som en uautoriseret adgang eller brug af ressourcer på et system. Dette er oftest servere, enten forbundet til internettet eller på et lokalt netværk.

Intruders

Intruders inddeles oftest i følgende 3 kategorier: Hackere, kriminelle og interne, dvs. medarbejdere eller personer der i forvejen har adgang til systemet.

	Hacker	**Kriminel**	**Interne**
Motiv	Opnå status eller nysgerrighed	Økonomisk	Berettigelse, hævn
Mål	Ofte tilfældige, opportunistisk	Specifikke	Specifikke, eks. kundedatabaser
Metode	Port scanning af ip ranges og brute forcing	Målrettet scanning og exploitation. "In-n-out"	Udnytter allerede eksisterende rettigheder
Intrusion Prevention/ Detection	Bloker scanning/ bruteforcing	Samme, sværere at opdage	Least privilege, logging

Hackere er individuelle personer eller små grupper som forsøger at få adgang til systemer på grund af nysgerrighed, status eller f.eks. udnytte ressourcerne til egen brug såsom fil-deling. Deres metoder er brede port scanning af ip ranges hos f.eks. data centre og større firmaer eller systemer som findes mere eller mindre tilfældigt. Der søges her efter services som hver kører på sin egen port og muligvis også giver et “banner” med information om servicen, f.eks. ssh. Adgang fåes som regel med velkendte exploits, som der kan scannes specifikt efter, eller via brute forcing. Når først der er opnået adgang til systemet, bliver dette ofte misbrugt i sådan grad at det relativt nemt kan opdages ved f.eks. drastisk forøget netværkstrafik eller ændringer i systemets opsætning, f.eks. en ftp server på et system der ellers aldrig har haft dette. I nogle tilfælde bruges der også automatiserede scripts som ofte efterlader sig spor der er nemme at opdage.

Kriminelle hackere er oftest grupper af hackere der samarbejder om at bryde ind i specifikke og udvalgte mål for økonomisk gevinst. Dette kunne f.eks. være en webshop som gemmer kunders kreditkort informationer. Metoderne er delvist de sammen som almindelige hackere, dog mere specialiserede og bedre udført med få spor efterladt.

Den sidste gruppe er interne brugere som allerede har adgang og muligvis adgang til mere data end nødvendigt. Det kunne eksempelvis være opsagte ansatte som tog en kundedatabase med sig. Denne type intruder kan være meget svær at opdage, men der kan tages forholdsregler ved at have audit logs og bruge least privelige princippet hvor en bruger udelukkende har rettigheder til de ressourcer brugeren har brug for.

Blandt disse typer er der yderligere inddeling: Masqueraders, Misfeasors og Clandestine users.

Masqueraders er udnyttelse af almindelige brugeres rettigheder f.eks. en hacker som finder login informationer til en bruger, logger sig ind og udnytter dette. En ændret adfærd i brugerens opførsel på systemet kan gøre denne type intruder nemmere at opdage.

Misfeasors er svarende til den interne type, som udnytter allerede eksisterende rettigheder.

Clandestine brugere er en kompromitteret root bruger hvilket må anses for at være den værste form for angreb. Med fuld adgang til systemet er der mulighed for effektivt at stoppe simple former for logging og auditing.

Teknikker

De mest almindelige teknikker til at få adgang til systemer er:

Brute forcing af logins og passwords til services kørende på systemet. Dette er eventuelt suppleret med word lists og default passwords til servicen. Denne metode er meget nemt at opdage.

Trojanske heste på en maskine tilhørende en bruger med autoriseret adgang. Disse kan komme ind på maskinen via email eller anden indgang og kræver som regel at en bruger aktivt henter og eksekverer programmet med et trojansk payload, som heller ikke detekteres af eventuel anti-virus software installeret på maskinen eller eksempelvis mailserver.

Man-in-the-middle attack hvor en intruder får fat i login information ved at være mellem en autoriseret bruger og et system. Enten rent fysisk eller virtuelt.

Exploiting af kendte sårbarheder i systemet. Undgås bedst ved at holde al software opdateret hvilket kan være besværligt. Visse sårbarheder er muligvis heller ikke kendt af software udvikleren og kan derfor ikke patches.

Intrusion Detection

Intrusion detection er forskellige teknikker til at opdage intruders. Problemet i at opdage indtrængende i et netværk skyldes at der er et vist overlap imellem en godartet og ondartet brugers opførsel (se figur nedenfor).

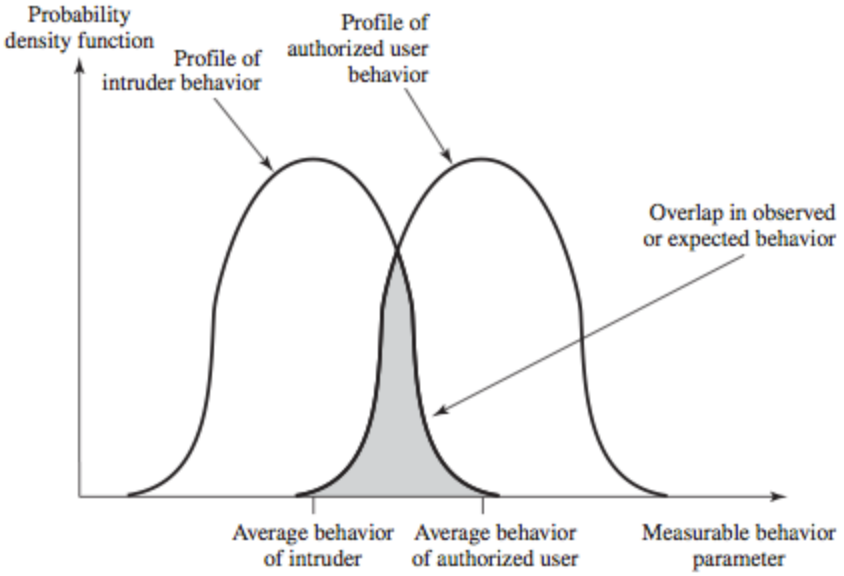

Figure 9.1 Profiles of Behavior of Intruders and Authorized Users

Dette overlap mellem brugermønstre leder til problemer med falske positiver (en godartet bruger bliver registreret som ondartet) eller værre, falske negativer (en ondartet bruger bliver opfattet som godartet)

Til trods for denne problematik, er der udviklet en række teknikker til at forsøge at imødekomme det store behov for detektion af indtrængende på netværket. Groft set kan disse inddeles i statistiske, og regelbaserede teknikker.

Statistisk baseret detektion baserer sig på at generere en profil af en given bruger, eller evt. en repræsentativ metabruger, imod hvilken observeret opførsel kan sammenlignes. Hvis en brugers nuværende opførsel afviger væsentligt fra hans hidtidigt observerede opførsel, rejses et alarmflag.
Regelbaseret detektion baserer sig på at opbygge en database af regler som indikerer (hvis ikke påviser) et angreb hvis disse brydes.

Statistisk baseret detektion bygger på en antagelse at en brugers adfærd i fremtiden vil ligne vedkommendes adfærd i fortiden, og er særligt anvendelig imod masqueraders og til en vis grad misfeasors. Masqueraders vil hurtigt afsløres da deres brugsmønster næsten med sikkerhed vil adskille sig markant fra den legitime bruger hvis credentials de benytter. Misfeasors vil være sværere at afsløre, da de kan opføre sig tæt på normalt. Her vil ting som pludselig tilgang til fortrolige filer brugeren ikke tidligere har brugt kunne være røde flag, men en bruger som tager

en ekstra kopi af en fortrolig fil vedkommende har haft adgang til, og brugt, i månedsvis vil være svær at fange.

Regelbaseret detektion bygger på en antagelse af at normal-opførsel kan defineres, og alle handlinger som ikke er normal-opførsel derfor kan listes og sammenlignes med en aktiv brugers opførsel.
Regelbaseret detektion falder ind i to underkategorier:
a) Anomaly Detection, hvor regler opbygges for at vise afvigelser fra tidligere brugsmønstre og
b) Penetration Identification, hvor kendte angrebsvektorer og svagheder bruges til at skabe et regelsæt som kan identificere en bruger som forsøger at udføre et af disse angreb.

Anomaly Detection lider under det problem at blot relativt simple brugsmønstre meget hurtigt eksploderer i antallet af regler. Et eksempel nævnt i bogen fra '89, taler om størrelsesordner som 10.000 til 1.000.000 regler.
Penetrations Identifikation kan ofte klare sig med færre regler, og kan med fordel opbygges ved bl.a. at holde sig opdateret på internettet omkring CERT bulletiner og nye angrebsscripts og -applikationer hvorefter regler som identificerer brugere som anvender disse angreb kan skrives af specialiseret personale. Systemet kan så enten vælge at føre log over hændelsen og lade netværksadministrationen skride til handling, eller udføre et automatiseret respons, som f.eks. at lukke den indkommende netværksforbindelse og tilføje den angribende IP til en blokeringsliste.
Denne type systemer har særlig værdi i større organisationer hvor det at holde software fuldt opdateret til enhver tid ikke er forretningsmæssigt realistisk, da en opdatering af en enkelt komponent af et komplekst system kan lede til kompatibilitetsproblemer med andre komponenter og anden uønsket adfærd.
Som et kompromis kan netværksadministrationen vælge at tilføje regler for de eventuelle svagheder som programopdateringen retter, således at et angreb som forsøgte at udnytte disse huller vil blive stoppet hurtigst muligt, og i mellemtiden give driftsafdelingen tid til at teste hvorvidt den nye version introducerer problemer i systemet.

Audit Records

En vigtig ting for at kunne implementere nogle former for intrusion detection er audit records, log over hvad brugere foretager sig og på hvilke tidspunkter. Dette findes allerede til en hvis grad i de fleste styresystemer, men disse er som regel ikke nok til intrusion detection. I stedet bruges mere specificerede logs. Et eksempel på denne slags logging fra [DENN87] indeholder følgende records:

- Subject: Hvem udfører handlingen
- Action: Hvad er handlingen
- Object: Hvad bliver handlingen udført på
- Exception-Condition: Benævnelse af ulovlig handling, 0 hvis ikke ulovlig
- Resource-Usage: Brugte ressourcer, f.eks. CPU forbrug
- Timestamp: Tidspunkt hvor handlingen udføres

Udføres eksempelvis kommandoen:
COPY GAME.EXE TO <Library>GAME.EXE (Kopierer executable til på systemet "fælles" library mappe)

Vil følgende audit records blive genereret:

Smith	execute	<Library>COPY.EXE	0	CPU = 00002	11058721678
Smith	read	<Smith>GAME.EXE	0	RECORDS = 0	11058721679
Smith	execute	<Library>COPY.EXE	write-viol	RECORDS = 0	11058721680

Den enkelte kommando bliver opdelt i 3 separate handlinger, som hver logges for sig, da eksempelvis det at eksekvere eller læse en executable ikke nødvendigvis er uautoriseret.

Base Rate Fallacy

En meget vigtig pointe i forbindelse med enhver form for intrusion detection er at man til enhver tid skal holde for mente ikke blot en tests merit i det enkelte tilfælde, men også tage i betragtning den samlede incidensrate i befolkningen. Sagt på en anden måde, hvis det er meget sjældent at en person er en intruder, så kræver det en usandsynligt nøjagtig test at undgå uacceptabelt mange falske positiver. Når testen er så nøjagtigt tunet at næsten ingen falske positiver sker i det enkelte tilfælde, vil man så til gengæld ofte se mange falske negativer slippe igennem.
At man generelt har en tendens til at glemme at se på den samlede incidensrate i en befolkning kaldes for 'base rate fallacy', og kan illustreres effektivt ved følgende eksempel:
Given en test som for et enkelt tilfælde har en sandsynlighed for at give det rigtige svar på 87% - det vil sige at hvis man tager en tilfældig person ud af en normalfordelt befolkning og udfører testen på vedkommende så vil den 87% af tiden give det rigtige svar - og en viden om at ud af befolkningen så er der kun 1% individer som er netværksindtrængere.
Hvad er nu sandsynligheden for at en person som tager testen bliver benævnt en netværksindtrænger på trods af at vedkommende ikke er det. Ved brug af bayes teorem, når vi frem til at den samlede sandsynlighed for dette tilfælde, et falsk positiv, faktisk er helt oppe på 93.7%, altså vil over 9 ud af 10 udslag være uskyldige personer på trods af den 'gode' test med 87% nøjagtighed.
Det bliver ikke meget bedre når vi finder at selv hvis testens nøjagtighed skrues helt op til 99.9%, altså en chance på een promille for at et isoleret udslag er forkert, så får vi stadig 9% falske positiver.
Generelt har vi i netværkssikkerhed så lav en incidensrate i befolkningen at det i praksis er umuligt at konstruere en tilstrækkeligt god test til at undgå en uforholdsmæssig høj rate af falske positiver.