Datavetenskap

Robert Gustavsson och Per-Ove Ringsby

Footprint Toolbox Framework

Examensarbete

2000:07

Footprint Toolbox Framework

Robert Gustavsson och Per-Ove Ringsby

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är vårt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Robert Gustavsson

Per-Ove Ringsby

Godkänd, 2000-05-31

Handledare: Hans Hedbom

Examinator: Stefan Lindskog

Sammanfattning

De flesta organisationer sprider idag ut information genom många kanaler, mängden information gör att inte ens de själva vet hur den samlade bilden ser ut. Så här uttrycker vi oss i vår inledning och menar då att om fokus sätts på den samlade bilden, kan den förmedla information som organisationen inte alls har menat. Illasinnade personer kan lägga pussel och på detta sätt skapa underlag för en attack. Den informationskanal som mer och mer dominerar är Internet och ses på många håll som helt nödvändig för organisationen.

I vårt arbete har vi utvecklat en metod som med utgångspunkt i det informationsavtryck som närvaron på Internet innebär skapar ett avtryck (footprint). Ett footprint ger en överblick över den utlagda informationen och blir ett hjälpmedel för att kunna sortera bort onödig information. Metoden är repeterbar det vill säga upprepade sökningar ger samma resultat, vi har också utvecklat en programvara som utför vissa sökningar automatiskt. För att skapa ett footprint måste många olika sökningar i flera oberoende källor genomföras. Just automatiken är en viktig del eftersom den sammanfattar dessa rätt så komplexa sökningar till ett enda knapptryck.

Footprint Toolbox Framework

Most organisations today spreads information about them self through many channels. The volume of the information makes it almost impossible for an organisation to have a clear picture of what information they give out. This is how we express ourself in the first chapter, and what we mean by that is that if you can get the hole picture it is possible to give away more and different information than what was actually intended by the organisation. An evil minded person can put the pieces of the pussle together and use this against the organisation. The information channel that more and more dominates today is the Internet and it is viewed by many as essential to the organisation.

In our work we have developed a metod that tries to make an information footprint of the information an organisation leaves on the Internet.This footprint gives an overview of what information an organisation has put on the Internet and will work as an aid to find out what needs to be there and not. The metod is repeatable, that is repeated searches will give the same result. We have also developed a program that will perform some of the searches automatically. To create a footprint many different searches in different sources has to be made. This is where the automation is essential because it makes all these rather complex searches availible through a simple push on a button.

Innehåll

Figurer

Tabeller

1 Inledning

Är den information vi förmedlar enbart informativ i det syfte som vi själva menar ?
Om du meddelar på internet att du idag har köpt och installerat en alldeles ny webserver, du vill ge besökare och användare tillgång till det allra senaste inom tekniken, och talar då gärna om vilken typ av maskin det är och vilket operativsystem du kör. En illasinnad person tackar dig för detta och börjar direkt testa om du fixat alla kända svagheter som ditt system lider av. Exemplet ovan visar att information kan användas i olika syfte. De flesta organisationer sprider idag ut information genom många kanaler, mängden information gör att inte ens de själva vet hur den samlade bilden ser ut. Trots att all information finns tillgänglig så är det inte trivialt att sammanställa den. I ett aktivt säkerhetsarbete är det ändå viktigt att kunna veta vilket pussel som är möjligt att lägga med den information som finns.

Finns det någon metod man kan använda för att söka, hitta och sammanställa ett sådant avtryck ?

1.1 Syfte

Varje gång man vill avbilda sitt informationsavtryck måste man börja söka information, resultat från en fråga kan bilda uppslag och indata till nya frågor. Man tvingas välja vilka informationskanaler man vill söka i och det är inte svårt att inse att man från gång till annan missar information om man inte går metodiskt tillväga. Vi vill med vårt arbete undersöka om det är möjligt att utarbeta en metod som med god repeterbarhet ger ett bra avtryck, samt undersöka möjligheterna för en automatiserad informationssökning över information tillgänglig på Internet.

Det finns otaliga program som kan användas för att söka information på Internet. Om man vill samla information från flera körningar med olika program eller med olika sökargument, sitter man snart med en diger lunta med mycket redundans. Vi vill utreda

möjligheten att i ett ramverk, läs verktygslåda, samla de olika program som man tänker använda sig av och genomföra en körning som automatiskt kör respektive delprogram och filtrerar resultatet.

Vi vill poängtera att syftet med en sådan kartläggning som vi beskriver är att organisationen själv skall kunna skaffa sig en överblick över den information som finns tillgänglig och använda som ett underlag i sitt säkerhetsarbete. Vi är medvetna om att metoden kan användas av illasinnade personer som vill skaffa sig underlag för en eventuell attack. Men genom att öka medvetenheten om hur information kan användas och missbrukas så anser vi att det i längden kommer att gynna organisationen själv som då motiveras till att skaffa sig kontroll över den information som sprids.

1.2 Förutsättningar

Om man tänker sig en organisations hela informationsavtryck så kan det finnas bitar som kan användas av konkurrenter som vill kartlägga en organisation, det kan gälla maskinpark, patent, sökta patent, vilken kompetens som finns inom företaget mm. Vi kommer inom ramen för vårt arbete endast att ägna oss åt information som vi hittar på Internet och då bara det som rör datasäkerhet. Frågan i sin helhet är ändå intressant och vi tror att den i vårt informationssamhälle kommer att vara mycket aktuell.

Vi kommer i vår footprint metod att vara mycket restriktiva med att inkludera information som man kan få från verktyg som nmap (portscanner) och liknande. Anledningen till denna avgränsning är att vi menar att ett footprint inte skall penetrera datorer utan endast samla tillgänglig information. Ett verktyg som hamnar i gränslandet är traceroute men vi har valt att inkludera detta eftersom det kan ge information om nätets topologi utan att vara för aggressivt.

1.3 Lösningsförslag

Vi delar in uppgiften i två delar, där den ena delen består av att utarbeta en metod och den andra delen att skapa ett verktyg, FTF (Footprint Toolbox Framework), som ingår som en del i metoden. Det som inte ryms i FTF är sådant som kräver en mer kvalificerad bedömning och som vi inte anser att vi kan implementera med någon sorts logik.

För metodarbetet har vi infört en checklista som skall vara ett levande dokument, där det med en bestämd konvention kan infogas nya checkpunkter. Denna lista möjliggör då en metodisk och repeterbar metod som bygger på tidigare erfarenheter.

FTF är en verktygslåda som innehåller ett antal subprogram som kan konfigureras att köras i sekvens, ett programs utdata bildar indata till ett annat. Om vi använder ett objektorienterat program är det naturligt att varje subprogram är en instans av ett objekt, varje objekt innehåller då en "att göra" lista och en "spara" lista. De program som bildar objekt är färdiga standard program, helst konsolprogram till Linux. Utdata kan vi sen enkelt filtrera med ett perl script och på så sätt dela ut uppgifter till andra objekt.

Exempel på program som kan användas:

nslookup — För att fråga en viss dns server efter information, går att köra med olika "mode". Ett standard program för Linux[1].

host — Ett program som liknar nslookup och listar dns information, en bra sak är att den själv testar olika dns servers. Ett standard program för Linux[2].

whois — För att fråga i registerdatabaser, här finns nätblock och kontaktinformation. Ett standard program för Linux[3].

traceroute — Ett standard program som visar ett pakets väg genom nätet[4].

dejasearch — Söker i nyhetsgrupper efter inlägg, använder mail adresser och fria ord som indata[5].

2 Vad är "footprinting"

Att systematiskt kartlägga en organisation för att skaffa sig kunskap om densamma har man alltid kunnat göra. Syftet med en sådan kartläggning varierar, men vi kan tänka oss flera anledningar: en illasinnad person som planerar ett angrepp, organisationen själv som vill hitta förslag till förbättringar och så vidare.

Eftersom organisationens lagrade informationen idag på många sätt utnyttjar Internet så har vi en situation där den också kan vara tillgänglig för långt fler än de som den är ämnad för. Man kan tala om att organisationer och företag sätter ett avtryck (footprint) ute på Internet.

Att söka genom informations källor och lägga ett pussel av den information som finns om organisationen och därigenom skapa detta footprint kallar vi footprinting. I begreppet footprinting vill vi också lägga ett systematiskt arbetssätt så att vi uppnår en repeterbarhet och en möjlighet att styra valet av information så att det passar det egna syftet. Vår definition av footprinting är:

- Att metodiskt och efter en i förväg utarbetad plan samla information om en organisation, baserad på organisationens närvaro på Internet.

2.1 Vilken information kan man hitta

Man kan omvänt fråga sig vilken information vill man hitta? Om vi vidgar synen till att inte bara gälla datorsäkerhet utan även industrispionage, fortfarande med Internet som informationskälla, kan vi med utgångspunkt i de metoder som vi nedan listar få uppslag till olika typer av information. Listan är en fri anpassning på Dr Worth Wade's metoder att skaffa sig information [6], som vi har hittat i ett papper[7]:

1. Periodisk hämtning och spegling av konkurrenters WWW-sidor och filarkiv.

2. Hämtning av öppen information på Internet i form av tidningsarkiv.

3. Kartlägga ett företag med hjälp av sökverktyg samt söka de sidor som har "länkar" till presentationerna för att finna intressenter, typ kunder och leverantörer.

4. Avlyssna och filtrera informationsflödet i nyhetsgrupper och mailinglistor.

5. Öppen analys av konkurrenters datanät och användare, undersöka nätet utifrån med hjälp av bland annat kataloginformation.

6. Kartlägga användarnas intressen på Internet, dvs vilken information söker de mest.

7. Delta i konkurrenters mailinglistor, interna WWW-sidor mm under falskt namn eller identitet som disponeras via konkurrenters kunder.

8. Kartlägga konkurrenters interna datanät i samband med samarbetsprojekt eller via konsultarbeten.

9. Kamp om domännamn och uttag av domännamn i förebyggande syfte.

Punkterna 4-6 är mycket tveksam moraliskt etisk medan punk 7-9 är direkt illegala. Vi har tagit med ovanstående för att visa vad information kan användas till samt belysa att motivet styr vilken information man söker. Det som är intressant är att information, som är utspridd, tillgänglig och i sig själv harmlös, om den sammanställs kan bilda underlag för en ganska aggressiv kartläggning.

Det som inte nämns ovan är den information som behövs om man vill göra ett dataintrång eller kanske till och med orsaka skadegörelse. Detta är nog det första man tänker på när ordet datasäkerhet kommer på tal. Det kan vara mail-adresser, nätadresser, ip-block, maskintyper, operativsystem mm.

2.1.1 Hemsidan

Den information som finns här har organisationen själv lagt upp, oftast i syfte att marknadsföra sig. Man vill skapa en bild som man önskar att andra skall ta till sig. Den infor-

mation som kan komma ifråga för ett footprint är domännamn, adresser och telefonnummer till både organisationen och dess personal, dessutom intressanta nyheter om inköp av datautrustning. Vi kan också tänka oss att en tredje part som på uppdrag av organisationen, presenterar information om organisationen. Ett exempel på tredje parts information är företaget kompass som presenterar mycket fyllig information om en klient, givetvis med klientens samtycke. Anställda kan ha egna sidor som informerar om deras arbetsuppgifter och befattning. Kunder och leverantörer kan ha länkar eller ge annan direkt information om deras affärsrelationer till organisationen .

2.1.2 Nyhetsgrupper & Chat sidor

Ibland kan anställda berätta om vilka system de kör, administratören kanske vill ha hjälp med inställningar av ny hård / mjukvara, vi har sett exempel där systemadministratörer berättar om sina datorer för att få hjälp med konfigureringar. Genom att systematiskt söka här kan man hitta intressant information. ICQ[8] användandet är också utbrett, i denna publika databas finns ofta information om anställda och deras befattning.

2.1.3 Register uppgifter

För att Internet skall fungera krävs att någon ansvarar för registreringen av domännamn och ip-nummer. Idag är domännamn för organisationer, företag med mera underordnade olika toppdomäner. För varje toppdomän finns ackrediterade registerhållare. Likadant så finns det för ip-nummer (nätinformation) en uppdelning, registrering och skötsel av ip-nummer görs idag av tre regionala organisationer, Regional Internet Registries (RIR). All denna information finns lagrad i sökbara databaser som allmänheten kan nå via hemsidor och webläsare eller speciella sökprogram.

Domäner

För domänrelaterad information, det vill säga vem som äger en viss domän, vilka DNS servrar som ansvarar för respektive domän liksom poster innehållande namn, adresser, telefonnummer och emailadresser till kontaktpersoner för domänen.

NIC-SE [9]Network Information Center Sweden AB, en organisation helägd av stiftelsen Internetinfrastruktur. NIC_SE tillhandahåller, koordinerar och står för drift av det nationella registret för internetdomännamn under .SE på Internet. I denna databas kan man hitta information om domäner under .SE.

INTERNIC [10]Inter-Network Information Center är ett samarbete mellan ett flertal organisationer och U.S Government som ansvarar för registreringar under .NET, .COM och .ORG.

Liknande databaser finns för varje toppdomän, men vi kommer att röra oss mest under ovanstående toppdomäner.

Internet Protocol (IP) nummer

De tre regionala organisationerna som sköter alla ip-nummer är:

- RIPE [11]Europa, Mellanöstern och delar av Afrika.
- APNIC [12]Asien.
- ARIN [13]Amerika och delar av Afrika.

RIPE (Resaux IP Eropean) En organisation lokaliserad i Amsterdam som ska serva Internet Service Providers (ISP) i Europa, Detta är den databas som är mest väldokumenterad av de vi undersökt. Databasen innehåller mycket information om Autonoma System (AS) och ip-nummer. En grundlig sökning över en

organisation i denna databas ger mycket information om deras nät, vilka nätblock som routas, tekniska och administrativa kontaktpersoner, adresser samt nätnamn.

ARIN (American Registry for Assigned Numbers). Liksom RIPE en icke vinstgivande organisation som administrerar och sköter registreringen av IP nummer inom sitt område. En sökning kan ge följande information: nätverksnamn, nätverksnummer, AS och ip-nummer samt namn, adresser, telefonnummer och email adresser till kontaktpersoner.

2.1.4 DNS-Servrar

DNS-servrar (namnservrar) innehåller routing information som måste vara tillgänglig på något sätt för att man skall kunna adressera datorer. Vi redogör inte för hur namnservrar används i nätverket utan endast för den information de innehåller. Om man vet ett namn, till exempel www.hemligheter.se, så används informationen i en namnserver för att mappa detta namn mot en nät-adress (ip-nummer). Namnservern som är ansvarig för domänen hemligheter.se måste känna till alla dess adresser och vid en förfrågan meddela desamma.

Det som är intressant är att dessa servrar kan replikera sin databas. Med standard verktyg som *nslookup* och *host* kan man be om just detta. Efter en sådan körning sitter man med en detaljerad lista över routing informationen däribland ip-nummer till alla datorer i domänen. Ibland har servrarna blivit instruerade att inte replikera sin databas, men givet ett ip-nummer talar de alltid om adressen. I dessa fall får man utnyttja att whois-databaserna ofta talar om vilken sekvens av ip-nummer som domänen använder och helt enkelt göra ett uppslag manuellt på alla adresser.

2.2 Bedömning av information

Vad bestämmer om information är harmlös eller inte ur ett säkerhets perspektiv ? Det footprint som vi arbetar fram är uteslutande skapat med hjälp av information tillgänglig på Internet. Av denna anledning är det inte en komplett bild, men trots det ganska detaljerad vad gäller en organisations närvaro på Internet.Då avtrycket är tänkt att ligga till grund för ett vidare säkerhetsarbete kan avtryckets detaljrikedom berätta lite om vilket säkerhets tänkande som råder på den spårade organisationen.

Med utgångspunkt i ett footprint som är hämtad med en standardiserad metod, som till stor del är automatiserad, kommer ett vidare arbete med dessa frågor att skapa en erfarenhet som kan användas för att tolka ett footprint. Den princip vi använder och anser skall råda är att metoden och den automatiserade processen inte skall tolka informationen utan endast utifrån programmets logik rekursivt arbeta sig fram tills ingen ytterligare information hittas. Det vill säga ta med alla epostadresser och inte bara de som innehåller domänen. Antag att en kontaktperson hos en IP (Internet Provider) är mycket frispråkig och diskuterar tekniska frågor för en klients räkning ute på någon nyhetsgrupp, detta gör att relevansen i en bit information måste tolkas av footprintets beställare. En anmärkning mot denna princip är att ett för stort footprint med både redundans och irrelevant information skymmer det som är verkligt intressant, och att man för att avbryta sökningen vid någon punkt måste tolka informationen. Vi har försökt hitta naturliga avgränsningar och konfigurerat programmet utifrån dessa.

3 Metod

Metoden kan indelas i tre delar, dessa delar kan betraktas som steg som skall genomföras i en bestämd ordning. Steg 1 och 3 är sådant som inte går att automatisera utan kräver ett visst mått av erfarenhet och mänsklig kreativitet medan steg 2 är den automatiserade delen som vi kallar FTF.

Nedan kommer vi att göra en genomgång som visar hur ett footprint skapas helt manuellt, det vill säga utan hjälp av FTF verktyget, vi kommer på detta sätt att tydligare åskådliggöra vilka beslut som måste fattas och varför sökningar naturligt görs i en viss ordning.

3.1 Websökning med standard sökmotorer och checklista

Det första man måste ta reda på om man vill kartlägga en organisations footprint är dess domännamn. Om organisationen är ett börsnoterat företag kan det enklaste sättet vara att börja med att gå in på Stockholms börsens hemsida och leta upp företaget. Där finns nästan alltid en länk till bolagets hemsida. Annars får man ta till nån av de många sökmotorer som finns på nätet, till exempel altavista. För att inte få med alla sidor där företagets namn nämns, det kan bli ganska många, kan man använda sig av sökordet "title:organisationens namn" i altavista. Med det sökordet får man bara resultat med sidor som innehåller söksträngen i sin titel. När man fått ett sökresultat går man igenom det och skriver upp alla intressanta länkar och får på så sätt en lista över hemsidor och domäner som tillhör organisationen.

För att få tag på ytterligare hemsidor som har med organisationen, dess anställda eller information om organisationen att göra kan man prova en sökning på altavista med "link:www.organisationen.org" som söksträng. Då får man en lista på siter som har länkar till sidan i söksträngen. På det sättet kan man hitta anställdas personliga hemsidor och andra delar av organisationen som man inte hittade i den första sökningen. De anställdas

personliga hemsidor kanske inte verkar så intressant vid en första anblick men om man tänker efter kan de ge en hel del intressant information. De flesta organisationer har till exempel användar namn och email adresser som på ett regelbundet sätt kan fås av namnet på den anställde. Dessutom kan den anställde tala om vad han har för arbetsuppgifter hos organisationen kanske till och med en bild på sig själv. Allt detta kan vara till stor nytta för någon som tänker göra lite social engineering.

Nu kan man gå igenom listan av web sidor med anknytning till organisationen och letar efter intressanta personer och deras e-mail adresser, nyheter om dator utrustningen och kanske information om organisationens struktur, var olika delar är lokaliserade till exempel. För att få fram denna information så snabbt som möjligt kan man använda sig av altavista nu också. Om man är intresserad av email adresser kan man göra en sökning med "host: www.organisationen.se" och kanske bara ett "@" tecken. Detta borde ge alla sidor hos den angivna hosten med ett "@" tecken i sig, vilket oftast innebär att det finns en epostadress. Dessutom har många organisationers hemsidor ofta egna sök funktioner ibland både för nyheter och epostadresser och då kan det vara bättre att använda sig av dem.

Men kom ihåg att sökmotorer inte alltid hittar allt så även om man tycker sig hittat det mesta ska man alltid surfa runt lite för hand. Då kan man hitta en del sidor som sökmotorerna inte fann och dessutom kan man komma på nya saker att söka på så att metoden kan utvecklas. En annan sak som man alltid ska undersöka är om det går att få en katalog listning från något bibliotek på nån server. Det brukar sällan gå men om man lyckas hitta någon katalog utan index.html (eller liknande) som inte är låst för listning vet man aldrig vad man hittar för filer där. Speciellt farligt blir det om det går att lista /etc/ biblioteket eller cgi-bin eftersom där kan finnas filer med lösenord till funktioner på sidan och servern själv.

3.2 whois

När man vet domän namnet kan man söka igenom nic-databasen för respektive topp domän. Detta skall resultera i en lista över dns servrar för domänen. Dessutom kan man beroende på vilken nic-databas man söker i få reda på kontakt personer, telefonnummer, adresser osv.

DNS servrarnas ip-nummer används som söknyckel i arin.net eller ripe.net databaserna beroende på topp-domän. För denna sökning används också whois. Sökningen resulterar i information om vilket nät den sökta IP adressen tillhör, vilka andra IP adresser som hör till nätet och vilka personer som ansvarar för det. Eftersom samma person ofta ansvarar för flera nät inom samma organisation kan det vara bra att göra ytterligare en sökning med namnen, eller deras register id, för att se om så är fallet.

3.3 host

Nu kan man köra host på alla domäner man hittills hittat. Host letar själv upp en domäns alla dns servrar och med rätt växlar försöker den hämta all dns-data om den sökta domänen. På detta sätt kan man få reda på vad alla domänens datorer heter och deras IP nummer, vilka datorer som hanterar mail och ibland även vilka operativsystem som de olika datorerna har. Om dns-servern är konfigurerad för att inte överföra sitt innehåll kan man utnyttja den serie av ip-nummer, som man fick från arin/ripe- sökningen för att få tag på datornamnen. Datorns namn säger mer än ip-numret, ex datorer som heter www.företaget.se har troligen en www-server på någon port. Vi har skrivit ett litet program, ip2name, som tar ett nätblock som indata och redovisar de symboliska datornamnen för respektive ip-nummer.

3.4 traceroute

För att få reda på hur organisationens nättopologi ser ut kör man programmet traceroute (UNIX) / tracert (NT) på de värddatornamn som host körningen resulterade i. Med detta

resultat kan man se hur paketen routas i nätet inom domänen och på så sätt få en bild av topologin. I bästa fall är det bara eventuella brandväggar som syns eftersom dessa ofta inte låter traceroute paketen slinka igenom.

3.5 dejasearch

Nu vi vet vad alla organisationens domäner och under domäner heter och känner till ett antal personers email adresser. Vad kan man då ha det till? Jo, man kan till exempel kontrollera om någon av de personer vi hittat, eller någon annan hos organisationen, är aktiv på någon nyhetsgrupp. Det kan vara någon systemadministratör som ställer frågor angående sitt system för att få hjälp att rätta till något problem. Sådana frågor kan vara mycket avslöjande för organisationen eftersom folk ofta talar om vilken hårdvara och mjukvara som används för att få bra svar. Om man nu vill söka efter meddelanden på alla nyhetsgrupper finns det flera sökmotorer på Internet. Bland annat en som heter deja.com. Vad dejasearch programmet gör är att det använder web tjänsten deja.com för att söka bland nyhetsgrupper och ladda hem de meddelanden som hittats.

3.6 Kompletterande websökning

En kompletterande websökning, steg 3 i metoden. Denna sökning bygger på ny vetskap om organisationen och möjligheten finns nu att komplettera och fylla på med de eventuella pusselbitar som saknas. Dessutom kan denna informationssökning användas för att avgöra om någon del i avtrycket inte är relevant och därför kan strykas.

3.7 Ett exempel.

Antag att företaget Hemligheter AB vill göra ett footprint på sig själva för att ta reda på vad det finns för information om dem på nätet. Det första de isåfall skulle göra vore att gå till någon av de stora sökmotorerna som Altavista eller liknande och göra en sökning

på företagsnamnet. Man söker med "title: hemligheter AB" på altavista och får då en mängd sidor med företagets namn i titeln. Först undersöks vilka av de olika länkarna som tillhör Hemligheter AB och när det är gjort finns en lista över företagets hemsidor och några domäner. Hemligheter AB verkar ha två domäner med websidor hemligheter.se och hemligheter.com.

Nu kan man fortsätta undersöka vad det finns för information om företaget på deras egna och andras websidor enligt metod som beskrivits ovan. När detta är klart och dokumenterat börjar man leta mer information om de domäner som hittats på de olika NIC databaserna. Man hittade till exempel en domän hemligheter.se som undersöks med kommandot *whois hemligheter.se@whois.nic-se.se*. Detta resulterar i en lista på domänens dns servrar och en del kontaktinformationen som talar om vem som ansvarar för domänen och så vidare, se figur 3.1. Kontakt information sparas undan medan man går vidare med dns-servrarna.

En sökning i RIR databaserna med dns-servrarnas ip-nummer blir nästa steg. Men för att kunna göra detta behöver man ip-numret till dns-servrarna. Ett sätt att skaffa sig ip-nummer är med kommandot *host ns1.hemligheter.se*, man får då mappningen
ns1.hemligheter.se 127.34.54.65. Eftersom dessa dns-servrar tillhör en .se domän så räcker det antagligen att söka i den europeiska databasen RIPE om dem, det görs med kommandot *whois 127.34.54.65@whois.ripe.net*. Från denna sökning sparas all information om nätet som dns-servern tillhör och de personer som ansvarar för nätet, se figur 3.2.

Sedan gräver man djupare genom att gå vidare med de ansvariga personernas namn och nic-handle och gör en inverterad sökning för att se om de har ansvar för andra nät också. Figur 3.3 visar resultatet från en möjlig sökning med kommandot
whois "-i admin-c,tech-c HA000-RIPE"@whois.ripe.net.
Här syns det att det att HA000-RIPE är ansvarig för flera nät. Denna information sparas och samma sökning görs på de andra personerna som hittades i den första ripe sökning.

När detta är klart fortsätter sökningen med den andra domänen, hemligheter.com. Då

```
*domainname.name:               hemligheter.se
*domainname.status:             REGISTRERAT
*domainname.regdate:            1986-05-29

*domainname.dns.data:
hemligheter.se.                     IN  NS  ns1.hemligheter.se.
hemligheter.se.                     IN  NS  ns2.hemligheter.net.
ns1.hemligheter.se.           IN  A  127.34.54.65

*contract_number:               999-888-777
*status:                        UPPRATTAT
*holder:                        Hemligheter Information Technology AB
*orgno:                         006103-1111
*contact_info.coaddr:           -
*contact_info.address:          Hemligheter IT
*contact_info.zipcode:          123 45
*contact_info.postal_town:      Hemligby
*contact_info.countrycode:      SE
*contact_info.phone_no:         555-123 45 67
*contact_info.fax:              555-12 34 56
*contact_info.contact:          Hemlige Hemligsson
*contact_info.email:            it1.dns@hemligheter.se
```

Figur 3.1: Resultat whois hemligheter.se@whois.nic-se.se

```
inetnum:     127.34.54.0 - 127.34.54.255
netname:     NET-HEMLIG2
descr:       Hemligheter Data AB
country:     SE
admin-c:     Hemlige Arne
tech-c:      Hemlige Arne
changed:     uffe@nogonstans.net 19940606
changed:     ripe-dbm@nogonannanstans.net 20000225
source:      RIPE

route:       127.34.54.0/24
descr:       NET-HEMLIG2
origin:      AS0002
advisory:    AS0001 1:1800 2:1133 3:1240
notify:      staff@hemligheter.net
mnt-by:      AS0000-MNT
changed:     Hemlige.Hemligheter@hemligheter.net 19950712
source:      RIPE

person:      Hemlige Arne
address:     Hemligheter Data AB
address:     123 45 Hemligby, Sweden
phone:       +46 555 123456
fax-no:      +46 555 234567
e-mail:      Hemlige@hemligheter.se
nic-hdl:     HA000-RIPE
changed:     lunkan@nogonstans.se 19971111
changed:     hostmaster@nogonannanstans.it 19971124
source:      RIPE
```

Figur 3.2: Resultat whois 127.34.54.65@whois.ripe.net

```
[joshua.ripe.net]

% Rights restricted by copyright. See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:     127.54.34.0 - 127.54.34.255
netname:     SE-HEMLIG1
descr:       Hemligheter Data AB
country:     SE
admin-c:     HA000-RIPE
tech-c:      HA000-RIPE
status:      ASSIGNED PI
mnt-by:      NOGON-MNT
changed:     ber@nogonstans.se 19940919
changed:     fredrik@nogonstans.se 19990216
changed:     fredrik@nogonstans.se 19991105
source:      RIPE

person:      Hemlige Arne
address:     Hemligheter Data AB
address:     123 45 Hemligby, Sweden
phone:       +46 555 123456
fax-no:      +46 555 234567
e-mail:      Hemlige@hemligheter.se
nic-hdl:     HA000-RIPE
changed:     lunkan@nogonstans.se 19971111
changed:     hostmaster@nogonannanstans.it 19971124
source:      RIPE
```

Figur 3.3: Resultat whois "-i admin-c,tech-c HA000-RIPE@whois.ripe.net

detta är en .com domän finns informationen om den hos internic. Sökningen görs med kommandot *whois hemligheter.com* och resultatet finns i figur 3.4. Här får man inte lika mycket information som hos nic-se. Men man får reda på domänens dns servrar.

```
Registrant:
Hemligheter Inc (HEMLIG-DOM)
   7921 Hemliga gatan Box 12121
   Hemligswille, NC 09090-0808

   Domain Name: HEMLIGHETER.COM

   Administrative Contact, Technical Contact, Zone Contact:
      Hemlighet, Hemlig  (HH0000)  HH@HEMLIGHETER.COM
      Hemligheter IT
      7821 Hemliga gatan  Box 21212
      Hemligswille, NC 09090-0808
      336.393.3319
   Billing Contact:
      Domain Billing  (DB1713-ORG)  billing@DOMAINNETWORK.SE
      Domain Network
      Nybrogatan 55
      Stockholm, S-114 85
      SWEDEN
      +46 8 527 90 600
      Fax- +46 8 527 90 630

   Record last updated on 20-Mar-2000.
   Record expires on 13-Sep-2000.
   Record created on 12-Sep-1995.
   Database last updated on 18-Apr-2000 04:36:44 EDT.

   Domain servers in listed order:

   NS1.HEMLIGHETER.SE  127.235.196.33
   NS2.HEMLIGHETER.SE  127.138.110.250
   NS3.HEMLIGHETER.COM 127.43.54.65
```

Figur 3.4: Resultat whois hemligheter.com

Liksom tidigare använder man dns-servrarnas IP-nummer för en ny sökning denna gång i ARIN-databasen för att få reda på vilka nät de tillhör, *whois 127.43.54.65@whois.arin.net,* se figur 3.5. Nu upptäcker man att IP adressen tillhör ett nät som heter HEMLIG1 och

```
[whois.arin.net]
[No name] (NS108-HST) NS3.HEMLIGHETER.COM 127.43.54.65
Hemligheter Data America (NETBLK-HEMLIGDATA-US) HEMLIGDATA-US
 127.235.192.0 - 127.235.207.0

To single out one record, look it up with "!xxx", where xxx is the
handle, shown in parenthesis following the name, which comes first.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.
```

Figur 3.5: Resultat whois 127.43.54.65@whois.arin.net

man gör då ytterligare en sökning i ARIN-databasen med HEMLIG1 som nyckel, *whois NET-HEMLIG1@whois.arin.net*, se figur 3.6. Denna information sparas. Man skulle kunna gå vidare och söka på handlern på personen som står som coordinator men eftersom all intressant information om denne redan finns med här är detta inte nödvändigt. Nu görs givetvis samma sak med de andra dns servrarnas IP nummer också. Ifall de skulle vara på något annat nät. Eftersom en organisation mycket väl kan ha fler nät än de som dns servrarna ingår i och eftersom det hos arin är möjligt att söka på en organisations namn så gör man även detta, *whois hemligheter@whois.arin.net*. Denna sökning visar att företaget har ytterligare nät som inte tidigare upptäckts, se figur 3.7. Dessa nät utforskas på samma sätt som ovan.

Kom ihåg att en så här bred sökning kan ge en hel del svar som egentligen inte har med just den organisation vi är intresserade av att göra, så man får vara lite selektiv innan man går vidare. Nu har förhoppningsvis all den information som finns fritt tillgänglig om de domäner och nät som Hemligheter AB har hittats.

Nu övergår man till att ta reda på om företaget självt ger ut fri information om sina datorer. Det enklaste sättet att göra detta är att med hjälp av programmet host försöka lista all information de olika dns servrarna har om företagets domäner. Sökningen görs med kommandot *host -lva hemligheter.se*, se figur 3.8 och *host -lva hemligheter.com*, se figur 3.9.

```
[whois.arin.net]
Hemligheter America (NETBLK-HEMLIGDATA-US)
   7821 Hemliga gatan
   Hemligby, NC 09099
   US

   Netname: HEMLIGDATA-US
   Netblock: 127.235.192.0 - 127.235.207.0

   Coordinator:
      Hemlig, Robert  (RE101-ARIN)  RH@HEMLIGHETER.COM
      910-393-2698

   Domain System inverse mapping provided by:

   NS1.HEMLIGHETER.SE 127.235.196.33
   NS2.HEMLIGHETER.SE 127.138.110.250
   NS3.HEMLIGHETER.COM 127.43.54.65

   Record last updated on 12-Mar-1999.
   Database last updated on 19-Apr-2000 05:38:13 EDT.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.
```

Figur 3.6: Resultat whois NET-HEMLIG1@whois.arin.net

```
[whois.arin.net]
HEMLIGHETER (NETBLK-HEMLIGHETER10) HEMLIGHETER10     127.28.92.0 - 127.28.92.63
HEMLIGHETER PERU S.A (NETBLK-HEMLIGHETER-PE)HEMLIGHETER-PE   127.49.210.0 - 127.49.210.31
Hemligheter  (NET-HEMLIGHETERVUV) HEMLIGHETERVUV   127.160.23.0
Hemligheter  (NET-HEMLIGHETERPV) HEMLIGHETERPV    127.171.0.0
Hemligheter (NET-VTV1) VTV1  127.138.117.0
Hemligheter  of  America (NETBLK-HEMLIGHETER) HEMLIGHETER
 127.154.248.0 - 127.154.255.0
Hemligheter  (NET-HEMLIGHETERV1) HEMLIGHETERV1    127.157.8.0
Hemligheter  (NET-HEMLIGHETERV2) HEMLIGHETERV2    127.157.9.0
Hemligheter  (NET-HEMLIGHETERV3) HEMLIGHETERV3   127.157.10.0
Hemligheter  (NET-HEMLIGHETERV4) HEMLIGHETERV4   127.157.11.0
Hemligheter  (NET-HEMLIGHETERV5) HEMLIGHETERV5   127.157.12.0
Hemligheter  (NET-HEMLIGHETERV6) HEMLIGHETERV6   127.157.13.0
Hemligheter  (NET-HEMLIGHETERV7) HEMLIGHETERV7   127.157.14.0
Hemligheter Data AB (NET-SE-SDE) SE-SDE    127.138.0.0
Hemligheter Data  America (NETBLK-HEMLIGHETERDATA-US) HEMLIGHETERDATA-US
 127.235.127.0 - 127.235.207.0
Hemligheter Do Brasil  Ltda. (NET-HEMLIGHETERBRASIL) HEMLIGHETERBRASIL     127.0.89.0
Hemligheter Do Brasil  Ltda.
 (NETBLK-HEMLIGHETERBRASIL3) HEMLIGHETERBRASIL3
     127.0.100.0 - 127.0.102.0
Hemligheter Sales & Service (NETBLK-ATWORK-HEMLIGHETER) ATWORK-HEMLIGHETER
     127.125.196.128 - 127.125.196.255

To single out one record, look it up with "!xxx", where xxx is the
handle, shown in parenthesis following the name, which comes first.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.
```

Figur 3.7: Resultat whois hemligheter@whois.arin.net

```
rcode = 0 (Success), ancount=2
Found 1 addresses for ns.hemligheter.se
Found 1 addresses for lisa.hemligheter.se
Trying 195.178.160.1
hemligheter.se 86400 IN SOA lisa.hemligheter.se hostmaster.hemligheter.se(
2000010701 ;serial (version)
28800 ;refresh period
7200 ;retry refresh this often
604800 ;expiration period
86400 ;minimum TTL
)
hemligheter.se 86400 IN NS lisa.hemligheter.se
hemligheter.se 86400 IN NS ns.kulan.se
hemligheter.se 86400 IN MX 10 brandy.hemligheter.se
news.hemligheter.se 86400 IN CNAME brandy.hemligheter.se
karkis.hemligheter.se 86400 IN A 193.15.51.252
karkis.hemligheter.se 86400 IN MX 10 karkis.hemligheter.se
tomas.hemligheter.se 86400 IN CNAME tomasks.hemligheter.se
brandy.hemligheter.se 86400 IN A 195.178.163.141
localhost.hemligheter.se 86400 IN A 127.0.0.1
tomasks.hemligheter.se 86400 IN A 195.178.163.135
www.hemligheter.se 86400 IN A 193.219.246.231
titanic.hemligheter.se 86400 IN CNAME brandy.hemligheter.se
lisa.hemligheter.se 86400 IN CNAME brandy,hemligheter.se
lisa.hemligheter.se 86400 IN A 195.178.163.141
kulan-gw.hemligheter.se 86400 IN A 195.178.163.142
victory.hemligheter.se 86400 IN CNAME brandy.hemligheter.se
ftp.hemligheter.se 86400 IN A 195.178.163.130
ns.hemligheter.se 86400 IN CNAME brandy.hemligheter.se
hemligheter.se 86400 IN SOA lisa.hemligheter.se hostmaster.hemligheter.se(
2000010701 ;serial (version)
28800 ;refresh period
7200 ;retry refresh this often
604800 ;expiration period
86400 ;minimum TTL
)
```

Figur 3.8: Resultat host -lva hemligheter.se

Den information som man får här sparas. Eftersom *host -lva hemligheter.com* inte gav något svar antas att de dns servrarna var konfigurerade att inte replikera sitt data. Men genom att använda programmet ip2name på de olika nätblock man fått i tidigare sökningar och se om där finns några datorer med namnet hemligheter i sitt symboliska namn. En sådan sökning kan endast resultera i en mappning mellan datorernas IP och symboliska namn men ingen information om vilka datorer som hanterar mail för domänen och sådana saker.

```
rcode = 0 (Success), ancount=3
Found 1 addresses for GATEKEEPER.HEMLIGHETER.SE
Found 1 addresses for GATE.HEMLIGHETER.SE
Found 1 addresses for GATEKEEPER.HEMLIGHETER.com
Trying 127.138.110.250
Server failed, trying next server: Query refused
Trying 127.138.110.253
Server failed, trying next server: Query refused
Trying 127.235.196.33
Server failed: Query refused
```

Figur 3.9: Resultat host -lva hemligheter.com

För att få en bättre bild av hur hemligheter AB's nät ser ut för en utomstående kan man nu köra traceroute på de datornamn som ip2name och host körningarna resulterade i. För att denna sökning ska vara meningsfull bör den ske från en dator som inte är placerad i företagets eget nätverk.

traceroute www.hemligheter.se

När man nu fått en hyfsad bild av hur hemligheter AB's nät ser ut för utomstående kan det vara dags att gå tillbaka till webben och gå in på någon av de sidor där man kan söka på nyhetsgrupper som till exempel deja.com. Tidigare sökningar har gett några epostadresser till personer som är ansvariga för Hemligheters olika domäner och nät. Så en sökning görs på dessa för att se om någon av dem skrivit något meddelande som kan vara allt för avslöjande. Dessutom talade host om alla Hemligheter AB's domäner och subdomäner då kan man göra ytterligare sökningar inom nyhetsgrupper för att se om någon som har en

epostadress på någon av dessa domäner skrivit något som innehåller känslig information.

För att ytterligare förtydliga var respektive information kan hittas hänvisar vi till tabell 3.1.

Tabell 3.1: Översiktstabell källa - information

källa	Information
NIC-SE, INTERNIC	DNS-server, Kontaktinformation
RIPE,ARIN	Nätblock, Vem som tilldelat nätblocket , nätblock som routas,kontaktinformation
DNS-Server	Hostname, mailserverar, IP-nummer och alias (hela databasen)
deja.com	Inlägg i nyhetsgrupper (utnyttjas av dejasearch)
Altavista (söktjänster på Internet)	Domännamn, sidor som länkar till hemsidan och länk till hemsidan
Organisationens hemsidan	Epostkonton, information om anställda, adresser och telefonnummer
www.ebl.se	Organisationsnummer (utnyttjas av orgnumsearch)

4 Lösning FTF

Vår lösning på problemet att automatisera stora delar av footprintingen bygger på att olika delprogram används för att leta upp den informationen vi är intresserade av, till exempel whois och host. Informationen tolkas sedan av perlscript och skickas till ett C++ program som är själva ramverket som sköter körningen av delprogrammen och deras perlscript. Ramverket sparar sedan informationen som kommer från perlscript:en eller skickar den vidare till andra delprogram som skall köras senare. För att ramverket ska kunna veta hur de olika programmen skall hanteras använder det sig av två konfigurations filer config.dat och sequence.dat. Ramverket i sig är uppbyggt av två centrala objekt, manager och program. Manager sköter körningen av program instanserna i den ordning som specificerats i sequence.dat. Den tar även hand om att producera ramverkets utdata, också detta efter hur det är specificerat i sequence.dat.

4.1 Config.dat

I config.dat specificeras ett antal strängar som används av ramverket för att veta hur programmet och perlscriptet skall köras och hur data skall tolkas. Ett typiskt program kan specificeras enligt exemplet i figur 4.1 "name:", "cmd:", "stdin:", "pearl:" och "end." är nyckelord som talar om vad strängarna skall användas till.

```
name: program
cmd: kommando $1
stdin: ..
perl: ./program.pl
spara_info: £0
program1 $2 £2
end.
```

Figur 4.1: Exempel på config.dat

name: Talar om att programmet kallas "program" så att andra program kan skicka information till det genom ramverket.

cmd: Säger att programmet ska startas med kommandot “kommando“ för att ge rätt utdata. Variablerna "$0"-"$9" är ett sätt att adressera olika delar av den sökträng som ramverket skickar med till alla körningar av programmen. Söksträngen delas upp i ord, så att varje ord av strängen kan adresseras separat. $1 blir alltså ord nummer 1 i söksträngen $2 ord nummer 2 osv. $0 Däremot är lite speciell eftersom den innehåller hela söksträngen.

stdin:.. Här skriver man in vad man vill att programmet skall få in på sin stdin när det startats. Denna strängen måste avslutas med ".." till skillnad mot de andra som avslutas med enter. Detta beror på att interaktiva program ofta behöver enter slag efter kommandon och därför inget bra tecken att kontrollera strängslut med.

perl: Denna rad fungerar ungefär som cmd: raden, fast här är det perlscriptet det gäller. Perlscripten som ska tolka informationen som programmen ger ifrån sig, kan vara konstruerade hur som helst, så länge deras utdata följer ett visst format som ramverket kan läsa.

end. Detta nyckelord avslutar respektive programs specifikation.

Formatet på utdata från perlscript skall följa nedanstående konvention:

det data man vill spara eller skicka vidare::progg1,spara_info,progg2::Nästa data...

Allt innan de första "::" är det data man vill spara eller skicka vidare. Efter "::" kommer namnet på det program man vill skicka datat till, skilt med komma. Ibland vill man inte skicka vidare datat utan istället spara det för senare utskrift. Då använder man sig av ordet "spara_", som program namn, följt av den etikett man vill spara datat under. Etiketten används senare av ramverket för att formatera utskriften. När man listat alla program som just detta data skall skickas till avslutar man med "::" igen och sedan följer nästa data bit som man vill göra något med.

Alla rader från nyckelorden fram till “end." är en lista där alla de program och spara_ etiketter som perlscripet kan generera finns följt av en sträng som talar om hur datat skall formateras till just det programmet. Här används variablerna $0-$9 igen för att kunna skicka med olika delar av detta programmets indata till nästa program. En ny variabel £0-£9 används även här. Den fungerar precis som $0-$9 fast här är det data fältet i perlscriptets utdata som adresseras. Så rad 5 i exemplet (spara_info:) skulle ha sparat hela data strängen under etiketten info. Den sjätte raden däremot skulle ha tagit det andra ordet i programmets indata följt av ett mellanslag och det andra ordet i data strängen och skickat till ett program med namnet "program1".

4.2 Sequence.dat

I sequence.dat, se figur 4.2, listas alla program som specificerats i config.dat i den ordning man vill köra dem avskilda med enter. Listan avslutas med "end." på en egen rad och sedan kommer en ny lista som talar om i vilken ordning de olika etiketterna skall skrivas ut när utdatat genereras av ramverket. Den sista delen av filen, före sista end., listar indata information som en javaklient använder för att ge ramverket ett rätt formaterat indata. Indatas formatering redovisas nedan. OR emellan host och ip2name betyder att om host misslyckas med att få någon information kommer ip2name att köras annars körs bara host.

4.3 Manager

Managern har en lista av program instanser och en save-lista. Save listan används för att spara de resultat man vill skriva ut som utdata från ramverket senare. Varje data som sparas taggas först med en etikett så att man senare kan skriva ut en etikett i taget.

```
orgnumsearch
nic-se
internic
ipresolv
hostlight
ripe
deep_ripe
arin
host OR ip2name
dejasearch
end.
info
net
host
news
trace
scan
end.
email dejasearch
domain internic,nic-se
org orgnumsearch,arin
orgnr nic-se
end.
```

Figur 4.2: Exempel på sequence.dat

4.4 Program

En instans av ett program objekt skapas med information från en viss del av config.dat. Detta resulterar i:

- ett antal strängar för program körning
- en lista där man kan matcha program och spara-etiketter mot deras format sträng.
- en ToDo lista med alla de söksträngar som ett visst program objekt ska köra.
- en HaveDone lista som lagrar alla sökningar som ett program har genomfört.

När en program instans har kört en söksträng flyttas denna från ToDo till HaveDone listan. När något program vill lägga till en söksträng till ett annat program objekt tittar det i både sin ToDo och HaveDone lista för att se att det inte redan har eller kommer att köra den söksträngen.

4.5 Exekvering av ett delprogram (program instans)

En körning av de program som specificerats i exemplet för sequence.dat, figur 4.2, fungerar enligt följande.

När ramverket startas skapas en instans av managern. Managern skapar sedan i sin tur så många instanser av program objektet som det finns program specificerade i config.dat och sparar dom i en lista. Managern läser in indatat från en tillfällig fil, skapad av en javaserver formaterad enligt samma konvention som gäller för delprogrammens perlscript, och lägger till data delarna som nya söksträngar i respektive delprograms ToDo lista.

Managern anropar nu run funktionen hos varje program instans i den ordning som de är listade i sequence.dat filen. Detta avslutas när alla programs ToDo-listor är tomma.

Nedan följer en översikt över ett delprograms exekvering (run anrop till programinstans):

- När en program instans får ett run anrop pop:ar den den första söksträngen i sin ToDo lista och använder denna för att byta ut alla variabler i programmets kommando och perl sträng.

- Programmet körs med det genererade kommandot och resultatet från denna körning pipas till pearlscriptet som formatterar resultatet och sparar det till en resultatfil.

- Resultatfilen som består av data och program öppnas av delprogrammet. Resultatet från körningen skall förmedlas till andra programinstanser eller sparas som utdata. Programmets match-lista utnyttjas nu så att respektive delresultat kan skickas till rätt målprogram.

- I målprogrammets indata sträng, en rad i delprogrammets config.dat, byts nu alla variabler ut mot de rätta delarna i datadelen och söksträngen. Den nya indata strängen läggs till i målprogrammets ToDo lista. Om program delen innehåller spara

etiketter fungerar det på ungefär samma sätt fast istället för att lägga den nya strängen i ToDo listan hos någon program instans läggs den i managerns save-lista.

Efter det att alla delprograms ToDo-listor är tomma börjar managern skriva ut alla strängar i sin save-lista. Detta sker i den ordning som specificerats efter första "end." i sequence.dat. Denna utskrift formateras av ett pearlscript till en html sida vars namn skickas tillbaka till javaservern som svar på sökningen.

4.6 Exempel

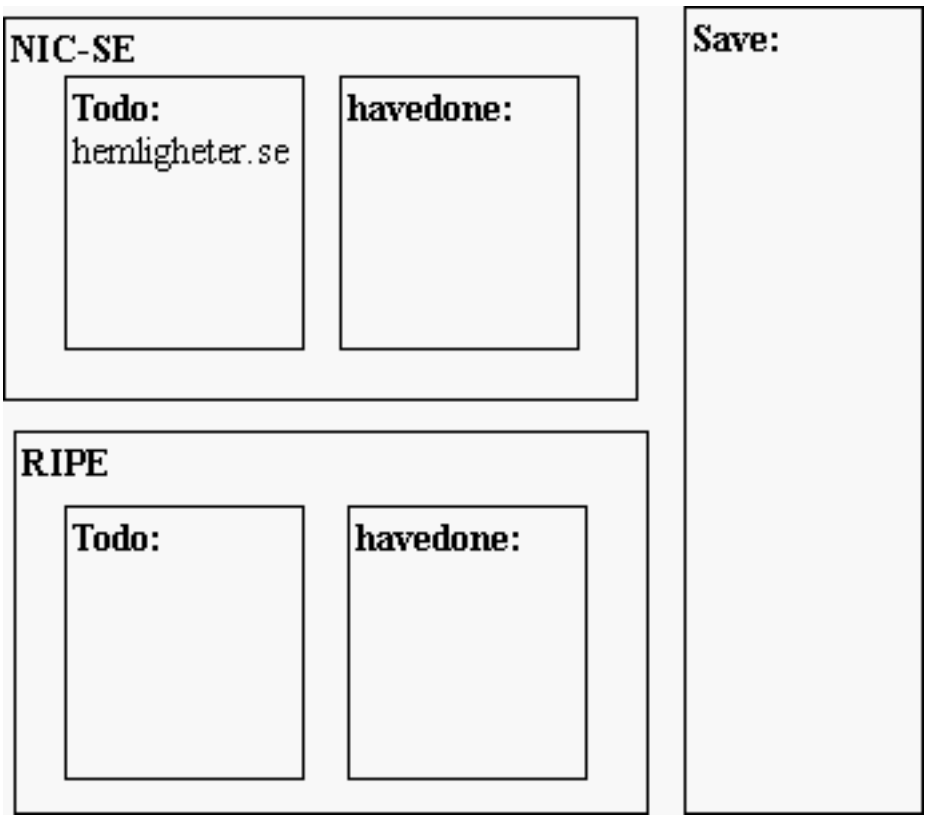
För att åskådligöra hur en körning kan gå till kan vi anta att vi har en config fil som ser ut som den i figur 4.3. Denna fil specifierar två program NIC-SE och RIPE, man kan se på cmd raden att det egentligen rör sig om två olika whois sökningar.

```
name: NIC-SE
cmd: whois $0@whois.nic-se.se
stdin..
perl: ./nic-se.pl
spara_info £0
ripe £0
end.

name: ripe
cmd: whois $0@whois.ripe.net
stdin..
perl: ./ripe.pl
spara_net £0
spara_info £0
end.
```

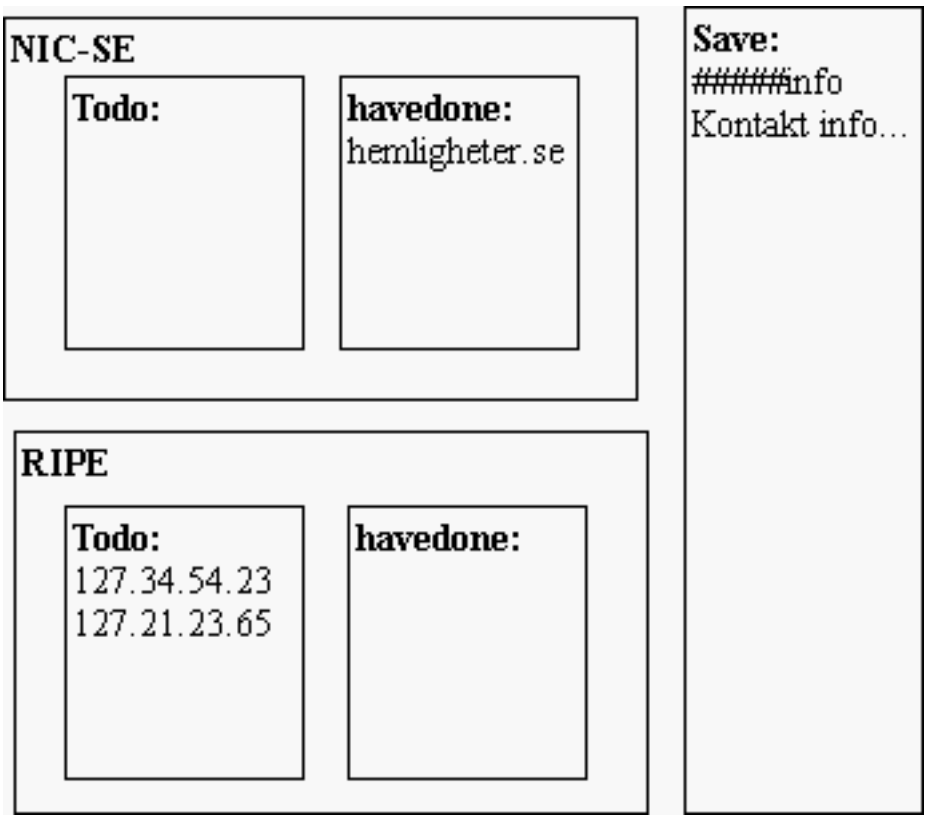
Figur 4.3: Exempel på config.dat

När vi startar programmet med hemligheter.se som indata kommer programmet NIC-SE att innehålla information enligt figur 4.4. Vi ser att NIC-SE i sin ToDo-lista har ett entry som manager objektet (MO) har lagt dit. Nu kommer MO att utifrån den ordning som finns i sequence.dat att titta igenom ToDo-listorna för respektive program. I vårt exempel kommer nu NIC-SE att utföra de operationer som finns i sin ToDo-lista.



Figur 4.4: Steg 1

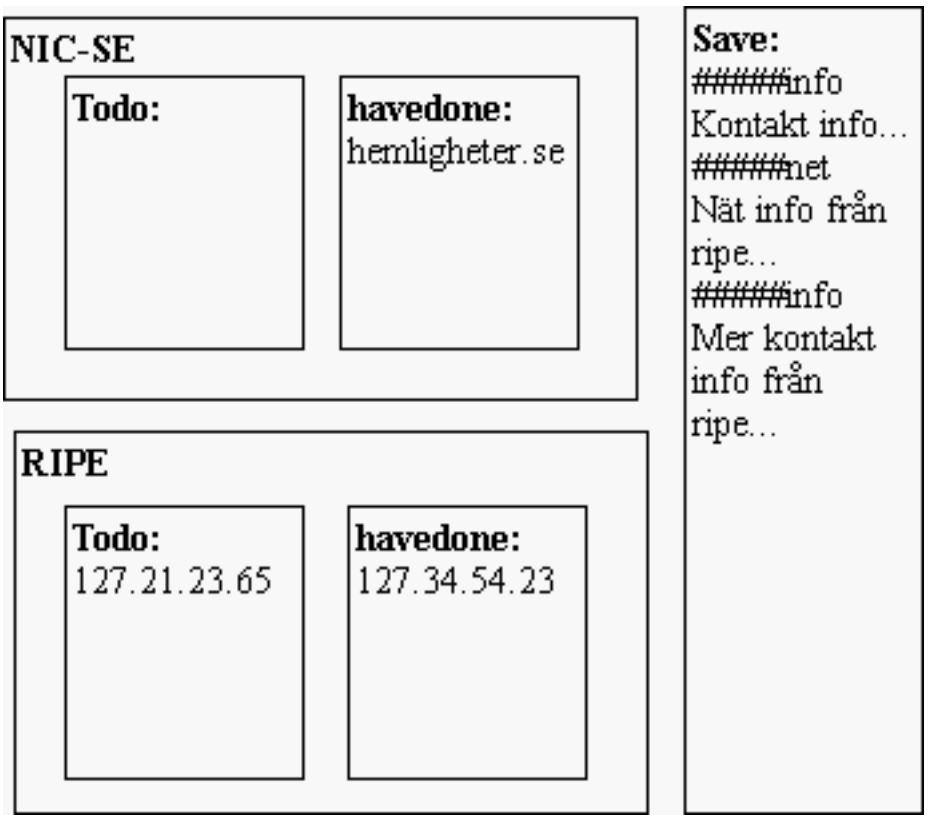
Nu har NIC-SE kört sitt kommando, *whois hemligheter.se@whois.nic-se.se*, och resultatet som är filtrerat med perlscriptet, ./nic-se.pl, från denna körning har MO tagit hand om och fördelat. Vi ser i figur 4.5 att det entry som tidigare fanns i ToDo-listan nu finns i HaveDone-listan. Samt att utdatat var två ip-nummer och kontaktinformation med etiketten "info".



Figur 4.5: Steg 2

När RIPE objektet har utfört det första av de två kommando som finns i ToDo-listan är situationen enlig figur 4.6. Ett entry mindre i ToDo som är flyttat till HaveDone och två entry i spara-listan med etiketterna "net" och "info".

Programmet eller egentligen MO kommer att fortsätta tills alla ToDo-listor är tomma. Då ett program kan generera utdata som hamnar i ToDo listan för ett objekt som ligger tidigare i sequence.dat filen kommer MO att börja från början igen och köra så många



Figur 4.6: Steg 3

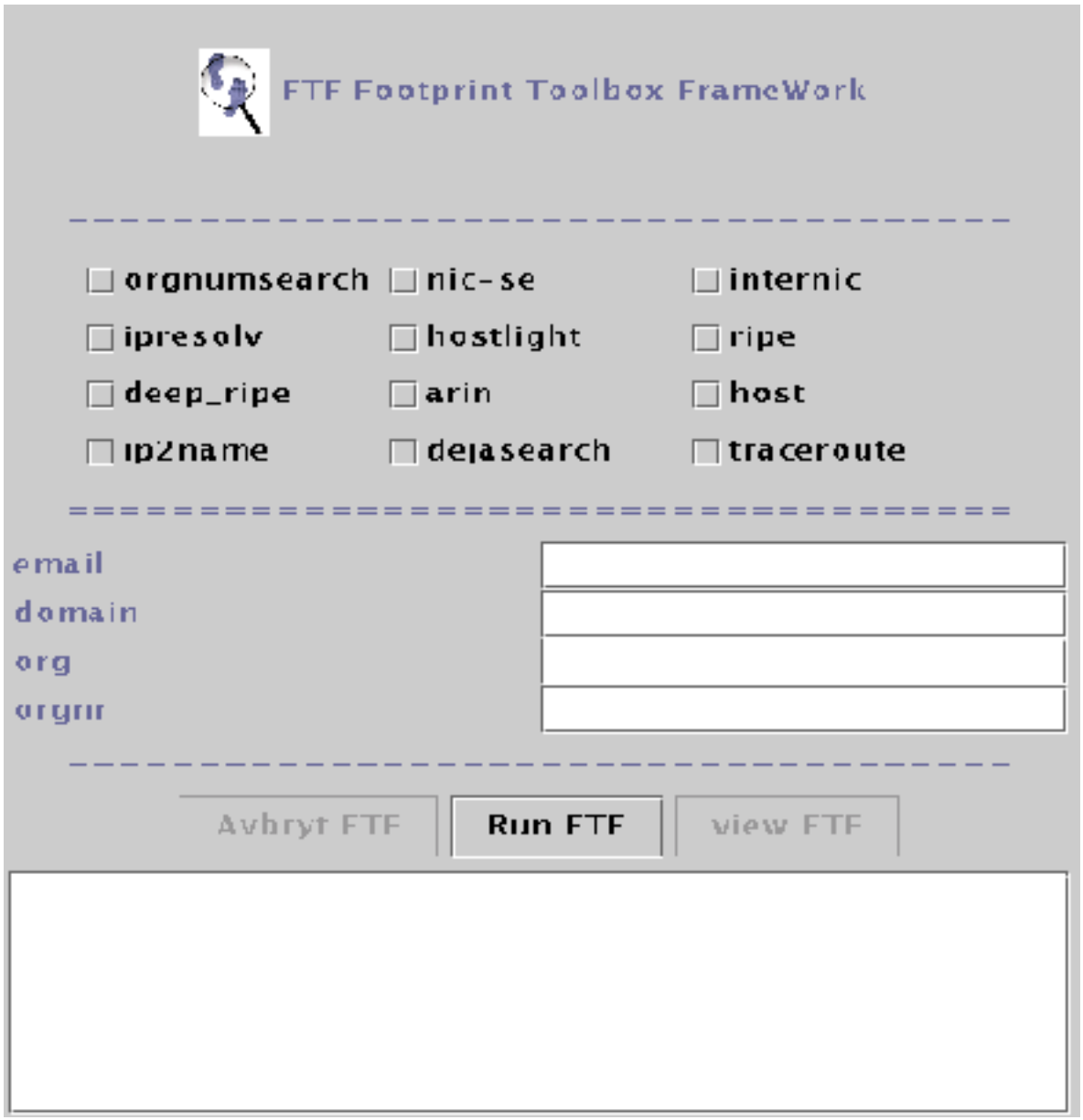
varv som krävs, det vil säga tills alla ToDo-listor är tomma. Detta fyller två syften, det ena är att man kan få ett rekursivt förfarande och på så sätt kunna söka i en databas med flera olika söknycklar, det andra att man inte missar något om man i sin sequence.dat fil lägger in programmen i fel ordning. Dock kommer en "rätt" ordning att göra körningen mer effektiv. HaveDone-listans uppgift är att spara gamla sökningar och på det sättet förhindra att samma sökning körs två gånger.

4.7 Gränssnitt

För att FTF:en ska bli lätt att använda oavsett var man befinner sig har vi gjort en klient utformad som en Java applet, figure 4.7. Appletten använder sig av RMI för att fråga en server vilka små program som finns i den nuvarande konfigurationen och vad de kan ta för indata. När användaren kryssat i vilka program han vill köra och skrivit in den indata han har. Skickas denna informationen över till RMI-servern som konstruerar en indata fil till FTF:en och sedan startar den. När FTF:en har slutfört sökningen resulterar det i en html sida som Appletten efterfrågar och visar för användaren.

4.8 Hur ser vår konfiguration ut?

Vi kommer här förklara hur den konfiguration vi gjort åt ramverket fungerar. Det sägs ofta nedan i texten att perlscriptet skickar något till ett annat program eller sparar något till någon etikett. Detta är inte riktigt sant. Perlscriptet producerar bara en resultat fil där den talar om vart saker borde sparas eller skickas som sedan tolkas av ramverket beroende på vad som står i config.dat för det aktuella perl scriptets program. Så själva sparandet eller skickandet sker i ramverket men eftersom man i perl scriptet trots allt talar om var man vill göra av datat så har vi valt att skriva på det sättet.



Figur 4.7: Skärmdump på klienten

4.8.1 orgnumsearch

För att ta reda på en organisations olika organisationsnummer använder vi ett egettillverkat program, orgnumsearch. Orgnumsearch använder en sökmotor hos www.ebl.se. Indatat hit kommer bara ifrån indatafilen och innehåller ett organisations namn. Perlscriptet formaterar om organisations numret lite så det stämmer med det formatet som används hos whois.nic-se.se och skickar sedan detta till programmet nic-se.

4.8.2 nic-se

Som man kan se i figur 4.8 så använder vi här programmet whois för att kunna göra förfrågningar om domäner på nic-se:s whois server. Indata kommer från indatafilen, orgnumsearch eller nic-se självt och innehåller bara en domän eller ett organisationsnummer. Om domänen existerar eller om någon domän hittats med det sökta organisationsnumret tar perlscriptet ut all kontakt information och sparar den under info etiketten. Alla IP nummer till dns-servrar skickas till arin och ripe. Domän namnet skickas till host, hostlight och dejasearch. Om det finns ett organisations nummer i kontakt informationen skickas det till nic-se igen. Dessutom antas det att domänen har några vanliga hosts till exempel mail, www och ftp. Dessa konstrueras och skickas till ipresolv för att sedan provas i ripe och arin.

4.8.3 internic

Internic använder också whois precis som nic-se fast här är det internics whois server som frågas. En bild av internics del av config.dat kan ses i figur 4.9. Internic får sitt indata från indatafilen eller internic självt. Indatat innehåller antingen ett organisations namn eller ett domän namn. Om indatat är ett organisations nummer skickar perlscriptet bara alla domän namn den hittar tillbaka till internic. Skulle indatat vara en domän som existerar hos internic sparas all kontakt information till etiketten info. Alla IP nummer till dns servrarna skickas till arin och ripe. Domän namnet skickas som hos nic-se till host, hostlight

```
# whois sökning i NIC-SE databasen
# indata från indata, nic-se
# format: domän|orgnum
name: nic-se
cmd: whois $0@whois.nic-se.se
stdin: ..
perl: ./nic-se.pl
nic-se £0 #Org-num
ripe £0 #Orgdomän IP
dejasearch £0 #email | domän-mail
spara_info £0 #domaninfo
host £0 #domän
hostlight £0 #domän
ipresolv £0 #Orgdomän hostname
end.
```

Figur 4.8: Konfigurering nic-se.

och dejasearch. Sedan antages också här att en domän troligen har host med namn som mail, www och ftp. Så dessa konstrueras och skickas till ipresolv.

```
# whois sökning i INTERNIC databasen
# indata från indata, internic
# format: Org|domän
name: internic
cmd: whois $0
stdin: ..
perl: ./inter.pl $0
internic £0 #domän
arin £0 #Orgdomän IP
ripe £0 #Orgdomän IP
host £0 #domän
spara_info £0 #domän-info
hostlight £0 #domän
dejasearch £0 #domän-email
ipresolv £0 #Orgdomän hostname
end.
```

Figur 4.9: Konfigurering internic.

4.8.4 ipresolv

Ipresolv använder host för att ta reda på vad ett visst hostname har för IP. Indatat till ipresolv kommer från internic och nic-se och innehåller en förkortad version av domännamnet, domän namnet utan toppdomän namnet hemligheter.se exempelvis blir hemligheter, som vi kallar orgdomän följt av det hostname som skall kontrolleras. Perlscriptet tar ut IP numret från svaret från host och skickar det vidare till arin och ripe ihop med orgdomänen. Figur 4.10 visar hur ipresolv är konfigurerat i config.dat.

```
# Kör host för att skaffa IP till en host
# indata från internic,nic-se
# format: Orgdomän hostname
name: ipresolv
cmd: host $2
stdin: ..
perl: ./ipresolv.pl $1
ripe £0 #Orgdomän IP
arin £0 #Orgdomän IP
spara_host $2 £0 #host A IP
spara_scan £0 #IP
traceroute £0 $2 #Orgdomän host
end.
```

Figur 4.10: Konfigurering ipresolv

4.8.5 hostlight

Som syns i figur 4.11 använder hostlight host för att ta reda på grundläggande information om en domän. Indatat är bara ett domän namn och kommer ifrån nic-se eller internic. När perlscriptet tolkar resultatet av host körningen plockar det ut alla IP-nummer den hittar och skickar dem till ripe och arin. Alla hostnames skickas till traceroute och hostnames med deras IP sparas till etiketten host.

```
# Försöker göra en mindre listning... kan funka för endel där host inte funkar.
# indata från nic-se, internic
# format: domän
name: hostlight
cmd: host -v -a $0
stdin: ..
perl: ./hostlight.pl $0
traceroute £1 £3 #Orgdomän hostname
spara_host £0 #hostname type IP(eller nått annat beroende på type)
dejasearch £0 #domän-email
spara_scan £2 #IP
ripe £1 £2 #Orgdomän IP-ns
arin £1 £2 #Orgdomän IP-ns
end.
```

Figur 4.11: Konfigurering hostlight

4.8.6 ripe

I figur 4.12 framgår det att vi även här använder whois men nu mot whois.ripe.net. Indatat till ripe består av en orgdomän följt av ett IP-nummer och kommer antingen från nic-se, internic, ipresolv eller hostlight. Det här perlscriptet sparar undan all information om det nätblock som IP numret tillhör i etiketten net och all information om de personer som står som ansvariga till dessa nät under etiketten "info". Dessutom skickas det högsta och lägsta IP-numret i nätblocket vidare till ip2name. Alla epostadresser skickas till dejasearch och de ansvarigas namn och ripe-nick skickas till deep_ripe.

4.8.7 deep_ripe

Deep_ripe använder sig också av whois mot whois.ripe.net men nu med lite andra växlar som möjliggör sökningar där man får reda på alla nät som en viss person är ansvarig för. De nya växlarna beskrivs i figur 4.13. Till deep_ripe kommer en orgdomän följt av antingen en ripe-nick eller en persons namn. Indatat till deep_ripe kommer alltid från ripe. Perlscriptet till deep_ripe fungerar precis som det till ripe med den skillnaden att det inte skickar vidare något till deep_ripe.

```
# Söker i ripe databasen
# indata från nic-se, internic, ipresolv, hostlight
# format: Orgdomän IP
name: ripe
cmd: whois "$2$3"@whois.ripe.net
stdin: ..
perl: ./ripe.pl $1
ip2name £0 #lågIP högIP Orgdomän
dejasearch £0 #email
deep_ripe £0 #Orgdomän (RIPE-nick | namn)
spara_info £0 #person-info
spara_net £0 #nät-info
end.
```

Figur 4.12: Konfigurering ripe

```
# Söker djupare i ripe med rekursion
# indata från ripe
# format: Orgdomän pers-nick|namn
name: deep_ripe
cmd: whois "-i admin-c,tech-c,zone-c $2 $3 $4"@whois.ripe.net
stdin: ..
perl: ./deep_ripe.pl $1
spara_net £0 #nät-info
ip2name £0 #lågIP högIP Orgdomän
ripe £0 #används ej nu.
end.
```

Figur 4.13: Konfigurering deep_ripe

4.8.8 arin

Arin är den amerikanska motsvarigheten till ripe så här används också whois fast mot whois.arin.net databasen. Indatat till arin består av en orgdomän följt av ett IP-nummer, ett nät namn eller ett arin-nick för en person och kommer från samma program som indatat till ripe. Eftersom utdatat från arin ser annorlunda ut så fungerar perlscriptet också annorlunda. Om indatat är ett ip nummer letar scriptet upp namnet på det nät som IP-numret tillhör och skickar det vidare till arin igen. Är indatat ett nätnamn eller nätblocksnamn så sparas den nät information som hittas till etiketten net, det högsta och lägsta IP-numret i nätblocket skickas till ip2name och om någon ansvarig persons arin-nick hittas skickas det till arin igen. Om indatat är ett arin-nick sparas person informationen till etiketten "info" och eventuella e-post adresser skickas till dejasearch. Figur 4.14 visar arins del av config.dat.

```
# Söker i arin databasen
# indata från nic-se, internic, ipresolv, hostlight
# format: Orgdomän IP|net(block)-nick|pers-nick
name: arin
cmd: whois $2@whois.arin.net
stdin: ..
perl: ./arin.pl $2 $1
ip2name £0 #lågIP högIP Orgdomän
arin $1 £0 #Orgdomän (ARIN-nick | nät | nätblock)
spara_net £0 #nätinfo
spara_info £0 #person-info
dejasearch £0 #email
end.
```

Figur 4.14: Konfigurering arin

4.8.9 host

Använder host med växlarna -l -v -a för att försöka få de dns-servrar som hör till domänen att lista vad de vet om den. Indatat hit är bara ett domän namn och kommer ifrån nic-se och internic. Perl scriptet sparar alla rader i listan till etiketten host. Dessutom skickas alla hostnames på A rader till traceroute. IP-nummer på A rader sparas även i ytterligare en etikett scan för att kunna användas som indatafil om man vill göra en port scan senare.

4.8.10 ip2name

Ip2name använder ett special skrivet program som också heter ip2name. Detta program tar ett nät block och går igenom det IP-nummer för IP-nummer och skriver ut alla IP som har ett hostname. Indatat hit kommer ifrån ripe eller arin och består av nätblockets lägsta och högsta IP-nummer följt av en orgdomän. Perl scriptet kontrollerar om en host innehåller orgdomänen. Om så är fallet gör det en rad som ser ut som en A rad från host och sparar den under host etiketten. IP-numret sparas också här under scan etiketten och hostnamet skickas till traceroute.

4.8.11 dejasearch

Använder programmet dejasearch för att söka efter nyheter skrivna av de epostadresser som skickas hit med indatat. Så indatat hit är antingen en epostadress eller en domän för en bredare sökning på alla adresser inom den domänen. Indatat kommer från arin, ripe, nic-se och internic. Detta perlscript sparar alla hittade nyheter till etiketten news.

4.8.12 traceroute

Traceroute körs för att ta reda på vilken väg paket tar till en viss host. Indatat hit består av en orgdomän och ett hostname och kommer från host och ip2name. Här plockar perlscriptet ut alla rader som innehåller orgdomänen plus de tre som kommer innan och sparar alla dessa rader i etiketten trace.

5 Genomförande och dokumentation

Vi har ovan redovisat hur footprinting kan göras manuellt, samt beskrivit hur FTF verktyget fungerar och vilka sökningar som genomförs med den nuvarande konfigureringen. Det här kapitlet visar ett sätt att arbeta fram ett footprint baserat på vår metod med FTF-verktyget som en integrerad del.

5.1 Checklista

För att få en repeterbar metod och försäkra sig om att alla de sökningar som skall göras verkligen görs kan man använda en checklista. Tanken är då att uppgifter som listas i detta dokument skall prickas av efter hand som arbetet fortskrider. För att en checklista skall fungera måste den vara enkel att arbeta efter och kännas som en nödvändig hjälp. När användaren är van och hittar ny viktig information med sökningar som inte anvisats av checklistan, skall det finnas incitament som gör att checklistan förändras. Träffsäkerheten i metoden är avhängig av ett dokumenterat tillvägagångsätt som ger en möjlighet att kunna skjuta in sig och därmed på ett effektivt sätt hitta den information man söker.

Vi har tagit fram ett förslag till hur en sådan checklista kan se ut men har inte haft tid att testa denna ordentligt, se bilaga 1. Man kan mycket väl tänka sig att implementera checklistan i något program som kopplat till en databas även samlar in dokumentationen och därigenom blir en hjälp att utveckla metoden, erfarenhetsåterkoppling.

5.2 Genomförande

Det footprint som skapas med en manuell genomgång består av flera olika resultatfiler och behöver på något sätt sammanställas så att redundansen försvinner samt att resultatet blir läsbart för en bredare grupp än de som utför arbetet. Lyckligtvis så plockar FTF verktyget bort mycket av redundansen på samma gång som det samlar allt resultat till en resultatfil.

När ett footprint skall arbetas fram, är det viktigt att de sökningar som görs i den

inledande web-sökningen dokumenteras så att alla sökningen kan härledas utifrån de resultat man finner intressanta. Ett footprint är ett led i organisationens säkerhetsarbete och finner man känslig eller ur en angripares synvinkel nyttig information måste man korrigera detta. Informationskällan måste därför kunna lokaliseras. De automatiserade sökningarna i FTF:et är dokumenterade i resultatfilen.

5.3 Principer för dokumentation

Gjorda sökningar kan dokumenteras enligt mallen i tabell 5.1, sökningar samt resultat redovisas liksom en bedömning hur resultatet påverkar säkerheten, vilka hot som öppnas och så vidare. Om man dessutom direkt har förslag till åtgärder skall detta också dokumenteras. FTF sökningen dokumenteras på så många rader man finner lämpligt.

Tabell 5.1: Principer för dokumentation

Sökning	Resultat	Bedömning	Förslag till åtgärd

6 Diskussion

6.1 Problem vi stött på

Det största problemen vi stött på har varit när det gäller att automatisera tolkningen av resultatet från de olika databaser och program som vi använder, vi har identifierat följande problem:

- Deras utdata är inte i första hand tänkt att tolkas utav ett program utan av människor.

- Formatet på utdatan från whois databaser och websidor ändras ibland och då måste man ändra i perlscriptet. Detta kan leda till en del underhålls arbete om man vill att sökningarna ska fungera ordentligt.

- Det är svårt att avgöra vilka uppgifter som man ska använda för vidare sökningar. Går man vidare med för mycket i till exempel RIPE kan man få en sökning som kanske aldrig tar slut eller resulterar i alldeles för mycket information. Men man får heller inte vara för restriktiv för då kanske man missar nyttig information. Detta bekymret visar sig bäst när man undersöker organisationer som inte verkar ha något eget nät utan kanske bara har en web server hos något nät företag och lite information i deras dns-server som talar om vilken dator som hanterar mail för organisationens domän. Då kan det bli så att man får en mängd information om olika personer och nät som tillhör nät företaget, vilket i och för sig kan vara interessant. Men ofta blir det alldeles för mycket information eftersom nät företagen har många nät och många olika administratörer.

- All sökning kan inte automatiseras. När man söker information på organisationens web-sidor och andra web- sidor som kan ha med organisationen att göra krävs ofta en hög grad av intelligens för att få ut det interessanta. Detta är något som är mycket

svårt att få en dator att göra på ett bra sätt. Så en del av det måste någon helt enkelt göra för hand.

6.2 Alternativa lösningar

De alternativa lösningar som man kan tänka sig handlar mest om hur implementationen av ramverket och klient/server delarna skulle kunnat gå till. Själva idén om hur de olika små programmen skall köras och de olika listorna med saker att göra och saker att spara är nog ganska givna. Däremot kunde man kanske ha tänkt sig att implementera hela ramverket i java eftersom klient/server delarna är i java. Då hade man kunnat bygga in hela ramverket i den del som nu fungerar som RMI server. Ytterligare en idé är att behålla ramverket i C++ och göra kommunikationen med hjälp av någon CORBA lösning mot java appleten istället för RMI. Men frågan är om det verkligen hade varit interessant.

En annan sak man kunde gjort annorlunda skulle vara att sköta matchningen av utdatat från små programmen med hjälp av lex[14] och kanske eventuellt yacc[15]. Man kunde ha specificerat olika strängar av reguljära uttryck att matcha i utdatat i programmets del av config.dat. Dessa matchsträngar kunde sedan lex ha plockat fram ur utdatat och gett till ramverket. Men vi tror att lösningen med perl script ger en mycket större flexibilitet även om den har den nackdelen att man måste kunna lite perl.

7 Slutsats och summering

I inledningen formulerar vi två frågor som vi vill undersöka och eventuellt finna en lösning på. Den ena frågan rör möjligheten att utarbeta en metod som med god träffsäkerhet hittar information som i ett säkerhetsperspektiv kan vara skadlig för organisationen. Den andra frågan var om det är möjligt att till stora delar automatisera en sådan sökning.

7.1 Slutsats metod

Om man skall sammanfatta metoden så är själva kärnan att man enligt en uppgjord plan skall söka information, och sedan gå vidare eller spara resultatet som en del i footprintet. Resultatet är mycket beroende av den plan som ligger till grund för sökningen. Vi har sett att den tid man lägger ner på att noga tänka igenom vilken information man vill att varje sökning skall resultera i avtecknar sig i footprintets kvalitet. Vi vill poängtera att metoden kan och kommer att utvecklas vidare i takt med en ökad användningen av verktyget men vi anser att vi med vårt arbete skapat en metod för footprinting som uppfyller de önskemål vi hade från början.

7.2 Slutsats FTF

Ett önskemål som vi hade med FTF-verktyget och som det uppfyller väl är att den arbetsinsats som ligger till grund för en bra metod kan tas tillvara och därmed användas i upprepade sökningar. De metodsteg som vi utarbetat har med hjälp av konfigurationsfilen och perlscripten kunnat överföras till programmet.

När vi arbetat med automatiseringen av informationssökningen så kan vi konstatera att själva motorn i verktyget, som implementerades tidigt i projektet, inte har behövts ändrats i sin design på något sätt. Funktionaliteten bygger på att verktyget själv skapar objekt utifrån konfigurationsfilen och kan på så sätt ändras med omgivningen. Genom att angripa automatiseringen på detta sätt har det visat sig att graden av automatisering är

ganska hög. Inte minst av den anledningen att man ganska enkelt kan bygga in nya moduler i ramverket.

Trots att ytterligare funktionalitet kan byggas kring vår lösning och då mest för att underlätta användningen så anser vi att den drivrutin eller motor som vi har tagit fram har visat att det är möjligt att automatisera stora delar av footprintingen.

8 Hur går vi vidare

Under tiden vi har arbetat fram metoden och FTF-verktyget så har vi varit tvingade att avgränsa vårt arbete. Denna avgränsning påverkar inte vår slutsats men lösningen rymmer en fortsättning som vi försöker sammanfatta i detta kapitel.

8.1 Bygg vidare

Vi har i vårt arbete kommit fram till att det är möjligt att i ett ramverk bygga in moduler och göra sökningar utifrån en viss konfigurering av dessa. Just vårt val av moduler är baserat på den kunskap som vi hade och skaffade oss i inledningen av vårt arbete. Med tanke på den mängd information som finns på Internet gör vi inga anspråk på att vi med dagens konfiguration och modulval har fångat allt av intresse. Det finns därför anledning att gå vidare och försöka hitta och implementera fler moduler som kan komplettera eller ersätta någon av de just nu aktuella.

Med perlscriptet som används till dejasearch som vi refererat till ovan kan man också själv göra egna sökverktyg på sidor som innehåller intressant information, vi har endast implementerat en egen sökning för att få tag i organisationsnummer.

8.2 Mer aggressiv

Undantaget traceroute så är vår sökning ganska snäll och lämnar inga spår efter sig som kan röja att någon har gjort ett footprint över en organisation. Detta kan man ändra på, om man implementerar moduler som till exempel använder nmap eller andra ännu mer aggressiva program. Man kan då försöka lista tillgängliga serverprocesser och operativsystem, mappat mot en lista med kända svagheter kan dessa skapa ytterligare information. Denna typ av sökning kommer i ett övervakat nät att synas och den som attackeras kan aktivera något typ av försvar.

Vi anser att en portscanning mer är en avbildning än ett avtryck och har därför inte

denna typ av analyser i vårt footprint. Men modellen och ramverket kan mycket väl utvidgas till en sådan djupare och mer aggressiv kartläggning och därmed bilda bas för en mer uttömmande säkerhetsanalys.

8.3 Social engineering

Vi nämner lite om detta inledningsvis, det är svårt att lista vilka typer av attacker som den i footprintet hittade informationen kan användas till. Man kan tänka sig att lägga upp strategier och träna sig på denna typ av attack, kanske med hjälp av någon teaterbegåvning. Genom att implementera ett program som skickar falsk epost och utge sig att vara en sysadministrator kan man få svar från intet ont anande mottagare där de röjer sitt lösenord eller kanske till och med passerkoder. Listan kan göras lång, det är nog bara fantasin som sätter gränser.

8.4 Andra informations källor

Telefonkataloger, företagskataloger, patent och registreringsverket, upplysningscentraler mm. Vi har bara ägnat oss åt den information som finns fritt tillgänglig på Internet. men som nämns ovan finns det även andra informationskällor. Man kan använda telefonnummer till en organisation och utifrån nummerserier försöka hitta modem som kan användas får att komma åt ett nät innanför en brandvägg. En del datorer har service modem anslutna. Att provringa ett antal telefonnummer kan bli en uppgift för en modul i vårt FTF.

8.5 Dokumentation

För att få en kvalitativ bild av ett footprints säkerhetsvärde kan man tänka sig att införa någon typ av poängsättning. Detta skulle ge organisationen större möjlighet att prioritera rätt åtgärd samt kanske lättare kunna koppla säkerhetskostnad mot skyddskostnad.

Referenser

[1] Andrew Cherenson, Linux man pages: man nslookup. 4th Berkeley Distribution, June 24, 1990.

[2] Linux man pages: man host. 4th Berkeley Distribution, December 15, 1994.

[3] Whois.net- Domain Based Research Service. URL http://www.whois.org/index.html.June 8, 2000.

[4] Van Jacobson, Linux man pages: man traceroute. April 22, 1997.

[5] Dejasearch-Your usenet search assistant. URL http://homemade.hypermart.net/dejasearch/, June 2000.

[6] Chemical Engineering:article by Dr Worth Wade ,May 23 1996, McGraw-Hill Inc.

[7] Mikael Simovits och Thomas Forsberg, Business intelligence på internet, Nordic Workshop on Secure Computer Systems, Nordsec 96, Göteborg, Sverige, 7-8 novemer, 1996.

[8] what is ICQ?. URL http://www.icq.com/products/whatisicq.html, June 2000.

[9] Välkommen till toppdomänen .SE!. URL http://www.nic-se.se/index.html, september 12, 1998.

[10] Welcome to the InterNIC Website!. URL http://www.internic.net/index.html, May 1, 2000.

[11] RIPE Network Coordination Centre-About RIPE NCC. URL http://www.ripe.net/ripencc/about/index.html, June 2000.

[12] Asia Pacific Network Information Centre-FOR YOUR INFORMATION URL http://www.apnic.net/general.html, December 31, 1999.

[13] Arin-About Arin. URL http://www.arin.net/arinintro.htm, June 2000.

[14] Vern Paxson, flex - fast lexical analyzer generator, Linux man pages: man lex, April 1995.

[15] Yacc - an LALR(1) parser generator, Linux man pages: man yacc. July 15, 1990.

9 Bilagor