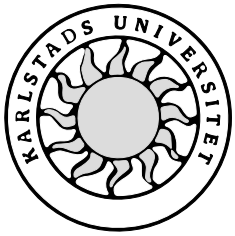


Datavetenskap

Anders Bergström

Mattias Hansson

Matrix – ett system för centralisering och administration av logginformation i ett nätverk

---

Examensarbete, C-nivå

2003:08

Matrix – ett system för centralisering och administration av logginformation i ett nätverk

Anders Bergström

Mattias Hansson

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.

Anders Bergström & Mattias Hansson

Godkänd, 2003-06-03

Handledare: Johan Eklund

Examinator: Stefan Lindskog

Sammanfattning

Denna rapport beskriver arbetet att vidareutveckla Veriscan Security AB:s prototyp för logghanteringssystemet Matrix. Bakgrunden till detta projekt är att arbetet med att kontrollera/granska loggfiler ofta är mycket tidskrävande, då det genereras enorma mängder information i loggfiler i ett nätverk. Veriscan Security AB:s mål med projektet är att förenkla och effektivisera systemadministratörers arbete med kontroll av logginformation. Detta skulle öka chansen att upptäcka t.ex. intrångsförsök.

Tyngdpunkten i detta arbete ligger på design och implementation av ett protokoll för kommunikation mellan olika datorsystem och en server i ett nätverk. Ytterligare delar som täcks in av examensarbetet är design och implementation av både servern, som hela systemet ska kretsa kring, och programvara för insamling av logginformation från de olika datorsystemen. En managerkonsol för konfiguration och fjärrstyrning av systemet, samt ett grafiskt användargränssnitt till denna, ska även konstrueras.

Resultatet av arbetet har blivit ett fungerande, körbart system, vilket styrs från ovan nämnda managerkonsol med tillhörande användargränssnitt. Systemet Matrix behöver dock vidareutvecklas ytterligare innan det blir komplett.

Matrix – a network-based system for centralizing and administration of log-information

Abstract

This report describes the further development of a system-prototype for log-handling called Matrix, which is owned by Veriscan Security AB. The reason for this project is, that inspection of log-files often is very time-consuming since the amount of log generated in a network is huge. Veriscan Security AB's purpose with this project is to simplify and increase the efficiency of the work that system-administrators have to perform in order to inspect log-information. As a result, the probability of detecting attempts of intrusion for example would be significantly higher. The emphasis of this project lies on the design and implementation of a protocol for communication between computer-systems and a server in a network. Also included in the project is the design and implementation of both the server, which is the central part of the system, and the application used to gather log-information from the computer-systems. A manager-console with a graphical user interface for configuration and remote control of the system is also to be constructed.

The result of this bachelor's project is a working system, which is controlled from the manager-console mentioned above. The manager-console has a graphical user interface. Further development of the system Matrix needs to be done before it's complete.

Förord

Ett stort tack går till handledare Johan Eklund på Karlstads Universitet som hjälpt och inspirerat oerhört mycket under hela examensarbetet. Tack går även till personalen på Veriscan Security AB, där detta projekt utförts.

Innehållsförteckning

1 Inledning 1

1.1 Introduktion till uppgiften 1

1.2 Arbetet beskrivet i denna rapport 2

2 Bakgrund 3

2.1 Veriscan Security AB 3

2.2 Matrix 3

2.2.1 Översikt
2.2.2 Bakgrund till och syfte med projekt Matrix
2.2.3 Agent
2.2.4 Server
2.2.5 Managerkonsol
2.2.6 Databas
2.2.7 Tidigare arbete inom projekt matrix

2.3 Uppgift 6

2.4 Förutsättningar 7

3 Genomförande av projektet 8

3.1 Analys av tidigare arbete 8

3.2 Tidssynkronisering 8

3.3 Design av nytt protokoll mellan agent och server 8

3.3.1 Översikt
3.3.2 Specifik design
3.3.3 Exempelscenario

3.4 Design, implementation och ändring av agenten 11

3.5 Implementation och utveckling av servern 12

3.6 Design av protokoll mellan server och managerkonsolen 13

3.7 Design av en enkel grafisk managerkonsol 14

3.8 Implementation av server-manager protokollet samt en enkel managerkonsol 14

3.9 Resultat 17

4 Diskussion 18

4.1 Analys av resultat 18

4.2 Kritik 18

4.3 Avgränsningar 18
4.4 Vidare arbete 19
4.5 Allmänna reflektioner 19
5 Slutsats 20
Referenser 21

Figurförteckning

Figur 1: Översikt av systemet Matrix 4
Figur 2: Kommunikation från server till agent 9
Figur 3: Kommunikation från agent till server 9
Figur 4: Exempelscenario på kommunikation mellan agent och server 11
Figur 5: Kommunikation från managerkonsol till server..... 13
Figur 6: Kommunikation från server till managerkonsol 14
Figur 7: Huvudfönster i managerkonsol 15
Figur 8: Dialogruta ”Manage logging for agent” i managerkonsol 15
Figur 9: Dialogruta ”Set allowed IP-number(s) for agent” i managerkonsol 16

1 Inledning

1.1 Introduktion till uppgiften

Varje datorutrustning i ett nätverk, exempelvis en brandvägg genererar, så fort en händelse sker, logginformation som skrivs till en loggfil. En loggfil innehåller med andra ord information om händelser och operationer som har utförts på datorutrustningen. Administratörer och säkerhetsansvariga har ibland intresse av att läsa i loggfilen för att se vad som exempelvis orsakat ett problem, eller för att analysera ett intrångsförsök.
Ofta är arbetet med att kontrollera/granska loggfilerna tidskrävande då dessa innehåller oerhört mycket information.
Detta examensarbete är utfört på uppdrag av Veriscan Security AB, i rapporten kallat Veriscan. De arbetar med ett projekt kallat Matrix, som avser att förenkla logghantering i ett nätverk genom att samla ihop data (logg) till en databas via en central server. En användare, t.ex. en systemadministratör, skall där ges möjlighet att sortera informationen och därigenom på ett enkelt sätt få ut den data som, från hans/hennes perspektiv, är relevant ur den totala informationsmängden .

1.2 Arbetet beskrivet i denna rapport

Arbetet som denna rapport behandlar är en del av projektet Matrix och avser framför allt kommunikationen till och från servern. Till servern ska agenter, som är programvara på en dator i nätverket, samt en managerkonsol, som är det grafiska verktyg systemadministratören använder för att konfigurera agenter, kunna ansluta och kommunicera. Arbetet innefattar även design och implementation av servern, med inriktning på kommunikationen både mot agent och managerkonsol. En sådan managerkonsol, från vilken en administratör ska kunna konfigurera systemet, måste även ha ett grafiskt gränssnitt. Modifiering/förbättring av den befintliga agenten ska även utföras, eftersom den nuvarande agenten enbart är en prototyp och måste anpassas efter den nya, mer utförliga, designen av agenten som är framtagen för att passa in mer i systemet Matrixs helhet. För att kunna testa de implementerade komponenterna i Matrix på ett smidigt sätt innebär arbetet även att implementera ovan nämnda managerkonsol.

2 Bakgrund

2.1 Veriscan Security AB

Veriscan Security AB bildades 1999 i syfte att utgöra en oberoende part åt organisationer vad det gäller IT-säkerhet. Företaget arbetar som konsult åt andra företag och utvärderar huvudsakligen andra företags IT-säkerhet [1].

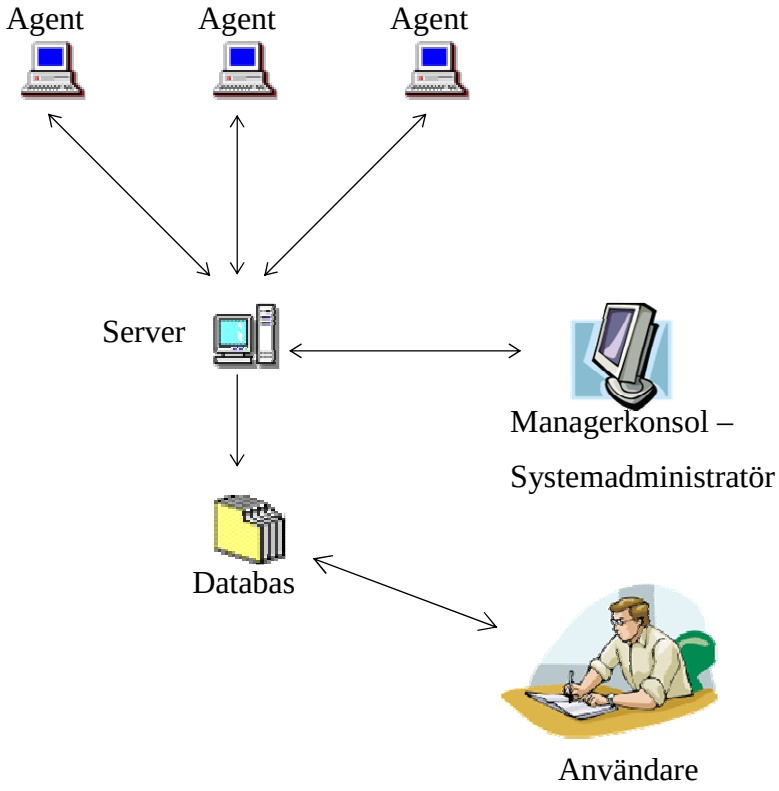
2.2 Matrix

2.2.1 Översikt

Anledningen till projektet Matrix uppkomst var att Veriscan önskade ett sätt att samla in och sortera loggar ifrån ett flertal datorsystem i ett nätverk, såsom arbetsstationer, routrar och brandväggar. Detta skulle kraftigt förenkla arbetet med kontroll av logg för systemadministratörer. Veriscan avser med Matrix ett komplett system för att samla in, filtrera, skicka, lagra och undersöka logg.

Systemets ”hjärna” är en central server. All kommunikation, t.ex. konfigurering av systemet från en managerkonsol eller insamlande av logginformation från agenten till databasen, går via denna server. Här lagras även information om vilken/vilka agenter som är tillåtna att ansluta till systemet.

Agenternas uppgift är att ansluta till servern, ta emot information från managerkonsolen (via servern) om vilken/vilka loggfiler som ska övervakas och sedan skicka all nyuppkommen logginformation till servern. Servern skickar sedan vidare insamlad logginformation till en databas, där den lagras och ska kunna tas fram av en användare, exempelvis en systemadministratör, vid begäran, för kontroll. Exakt hur användaren ska kunna få fram information från databasen är inte designat än, men en idé är att använda sig av ett program, en klient, som ansluter direkt till databasen och tar fram begärd information. Genom managerkonsolen kan systemets administratör ställa in vilka agenter som ska vara aktiva på vilka datorer, samt vilka loggfiler dessa ska övervaka. Figur 1 visar en bild av hur det kompletta systemet Matrix är tänkt att se ut [2].



Figur 1: Översikt av systemet Matrix

2.2.2 Bakgrund till och syfte med projekt Matrix

För att förklara problemet som föranlett Veriscans projekt Matrix måste först förklaras vad en logg i projektets betydelse är. En datorutrustning såsom en arbetsstation, brandvägg eller router skriver till en loggfil när en händelse som berör datorutrustningen i fråga inträffar. Som exempel skriver ett antivirusprogram på en arbetsstation till sin loggfil när ett virus har upptäckts, brandväggen skriver till sin logg när oönskad trafik upptäcks och routern skriver till sin logg så fort ett IP-paket har inkommit. Alternativen för loggning går att ställa in efter önskemål. En loggfil innehåller med andra ord information om händelser och operationer som har utförts på datorutrustningen. Dessa loggar växer ständigt och som ett exempel kan nämnas att en router kan skriva 1000-tals rader logginformation per minut. För en administratör är det en mycket svår uppgift att hålla uppsikt över alla dessa loggar. Speciellt svårt är att se samband mellan samtidiga händelser på olika datorsystem inom samma nätverk.

Många organisationer använder idag ett flertal system för att implementera och kontrollera säkerhet i sina egna nät och i anslutningspunkter med andra nät. Alla dessa system producerar loggar, i olika format och i stor mängd. Det mesta av denna information är dock oftast ointressant, så det finns ett intresse från Veriscan att ta fram ett system för att kunna urskilja det viktiga från det oviktiga i loggarna. Detta är det första motivet till att projektet Matrix startades.

Det andra motivet är att i ett nätverk finns det alltid en eller flera personer som ansvarar för nätverket, så kallade systemadministratörer. Om man tänker sig ett medelstort nätverk på exempelvis 20 arbetsstationer, en router och en brandvägg medför det nästintill obegränsat arbete för systemadministratörerna endast att kontrollera loggar på alla datorutrustningar. Anledningen till att man kontrollerar logginformation är bl.a. att man vill upptäcka eventuella intrångsförsök och kunna undersöka vad som orsakat ett eventuellt fel på en datorutrustning. Avsikten är att detta skall förenklas med projektet Matrix genom att all logg skickas till en central server, som eventuellt har inställningsbara filter som sorterar bort ointressant logg. Dessa filter kan senare under implementationen av Matrix komma att ligga hos agenten istället, eller eventuellt både hos agent och hos server med möjlighet att ställa in var man vill att logginformationen ska filtreras. Detta är ännu inte bestämt eftersom designen av filtreringen inte är påbörjad. Den filtrerade logginformationen läggs sedan in i en databas, från vilken systemadministratörerna på ett snabbare och mer effektivt sätt kan få fram relevant information om händelser i nätverket. I slutänden kommer denna ökade förmåga till selektiv och samordnad övervakning att kunna bidra till ökad säkerhet och mindre risk för felaktigheter i ett datornätverk [2].

2.2.3 Agent

En agent är programvara som ligger på varje enskilt datorsystem i nätverket som ska använda Matrix. Det som händer när en agent startas upp (manuellt), är att den försöker ansluta till servern. Lyckas inte denna uppkoppling så försöker agenten automatiskt att koppla upp igen efter 60 sekunder. Efter att agenten anslutit till servern skickas information från en managerkonsol via servern om vilken/vilka loggfiler som agenten ska övervaka. Om de angivna loggfilerna existerar på datorn som agenten körs på skickas all nyuppkommen logginformation i dessa filer till servern. Om någon loggfil ej skulle existera på agentens dator skickas ett felmeddelande tillbaka till servern [2].

2.2.4 Server

Servern är navet i hela systemet Matrix. Till denna kan ett godtyckligt antal agenter i nätverket, samt en managerkonsol, ansluta samtidigt. All kommunikation i hela Matrix, t.ex. logginformation som ska från agent till databas, går genom denna server [2].

2.2.5 Managerkonsol

Från managerkonsolen kan en systemadministratör konfigurera systemet Matrix, såsom att ställa in vilka agenter som ska vara aktiva eller vilka loggfiler agenterna ska övervaka. Till denna managerkonsol finns ett grafiskt användargränssnitt, ofta förkortat GUI [2].

2.2.6 Databas

I databasen lagras all logginformation som skickas till den från servern. Mot denna databas ska en användare, t.ex. en systemadministratör, kunna ansluta och få fram begärd data på ett smidigt sätt [2].

2.2.7 Tidigare arbete inom projekt matrix

Veriscan har haft ett tidigare examensarbete [3], vars mål varit att konstruera en agentprototyp samt en enkel server-stubbe (endast för att tillfredsställande kunna testa sin agent). Detta examensarbete är den enda tidigare utveckling som skett på projektet Matrix.

2.3 Uppgift

Veriscans uppgift för examensarbetet som denna rapport innefattar har varit att fortsätta utveckla systemet Matrix för att det i framtiden ska bli komplett. Veriscan har dock inga krav/förhoppningar på att detta examensarbete ska kunna leverera ett användbart system. Veriscan har givit uppgiften att:

- Ta fram en specifik design av hela systemet Matrix med tyngdpunkt på kommunikationen mellan agent och server. Det är ett krav att systemet ska vara plattformsoberoende.
- Vidareutveckla agenten. Agentprototyp finns att tillgå som resultat av tidigare examensarbete [3]. Denna prototyp är från början tänkt att i huvudsak användas i

befintligt skick. Möjligtvis måste viss modifikation göras för att få agenten att fungera med det protokoll för kommunikation med servern som ska skapas (vid implementation av managerkonsolen kan eventuellt större delar av agenten behöva ändras/skrivas om). Agenten ska kunna ansluta till servern och ta emot konfigurations-information från managerkonsolen (via servern) om vilken/vilka loggfiler som ska övervakas. När detta är avklarat ska all nyuppkommen logginformation i övervakade loggfiler skickas till servern.

- Utveckla en server. Det finns en mycket begränsad server-stubbe att tillgå från tidigare arbete med projektet. Denna kan vidareutvecklas om så önskas men troligtvis måste dock servern byggas upp från grunden, då aktuell implementation har avsevärda begränsningar. Servern är det centrala navet i systemet, all trafik över nätverket ska gå genom denna. Den måste vara en multi-user-server, d.v.s. den ska stödja att flera agenter kan vara anslutna samtidigt.
- Utveckla ett protokoll för kommunikation mellan den logg-insamlande agenten och servern. Detta protokoll måste noga designas innan implementation. Det måste förutom att kunna hantera exportering av logg från agent till server även vara utformat för att stödja fjärrstyrning av agenten. Med fjärrstyrning menas en möjlighet att kunna modifiera inställningar hos agenten från en managerkonsol.
- Tillverka en managerkonsol, ett GUI för administratören av Matrix-systemet. Genom denna managerkonsol ska agenterna kunna fjärrstyras, d.v.s. det ska gå att ställa in för varje enskild agent om den ska vara aktiv eller inte samt vilka loggfiler som agenten ska övervaka.
- Implementera stöd för tidssynkronisering. Med detta menas en lösning som synkroniserar systemklockorna på samtliga datorsystem det körs en agent på. Anledningen till detta är att man vill upptäcka eventuella samband mellan logg-händelser på olika datorutrustningar, då främst vid intrångsförsök.

2.4 Förutsättningar

Goda kunskaper i Java-programmering är en förutsättning för att detta examensarbete skall kunna genomföras. All tidigare kod är nämligen skriven i Java och Veriscan rekommenderar användning av Java. Med detta i åtanke så står inläsning av detta programspråk högst på agendan i initialskedet. Med tanke på att ingen av deltagarna i detta projekt har några som

helst förkunskaper i Java så utgör denna inläsning en relativt stor del av projektet. När tillräckliga kunskaper i Java införskaffats så blir nästa steg att läsa in och förstå koden till den befintliga agentprototypen.

3 Genomförande av projektet

3.1 Analys av tidigare arbete

Det visade sig ganska omgående att designen av den redan utvecklade agenten [3] inte var anpassad att passa in i det fullständiga systemet. De viktigaste delarna som inte var kompatibla med det övriga systemet var:

- Inget enhetligt sätt att få fram tid från logg-händelser.
- All konfiguration görs på plats på agenten (ohållbart i ett stort nätverk).

Detta innebär för detta examensarbete att mycket större delar av den befintliga agentprototypen än beräknat måste designas och skrivas om för att passa in i helheten – systemet Matrix.

3.2 Tidssynkronisering

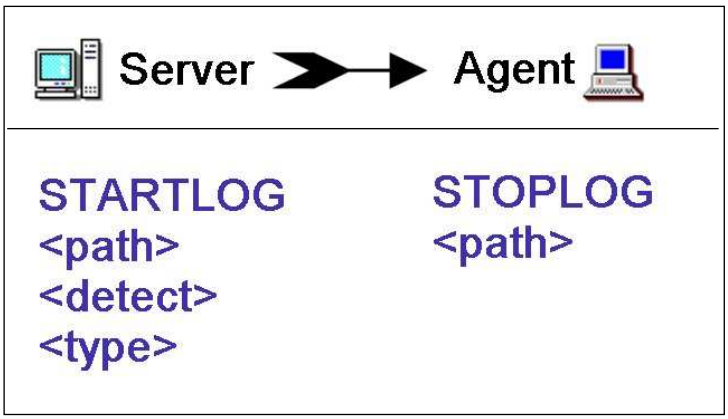
Planen var först att integrera tidssynkronisering i systemet Matrix. Vid undersökning av olika sätt att kunna förverkliga detta upptäcktes att det var mycket mer komplicerat än vad som först antagits. Tidssynkroniseringen avgränsades därför från detta examensarbete (Se kapitel 4.3).

3.3 Design av nytt protokoll mellan agent och server

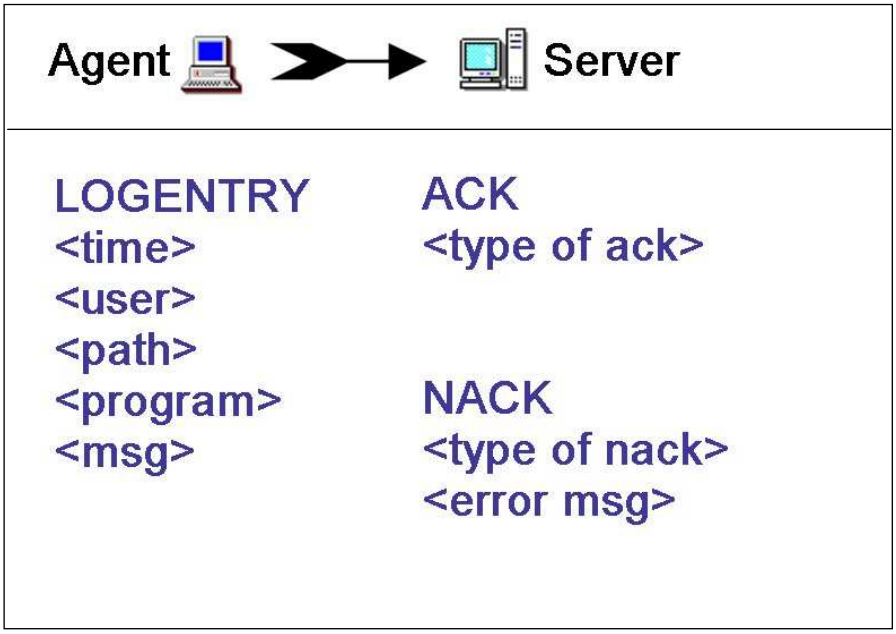
3.3.1 Översikt

Först bestämdes att kommunikationen i protokollet måste vara dubbelriktad. I riktningen server till agent skickas information om vilka loggfiler som ska övervakas (loggas) och när

loggning av en fil ska stoppas (figur 2). I motsatt riktning (agent till server) skickas logg-händelser från de övervakade filerna, samt även ACK:ar och NACK:ar (figur 3).



Figur 2: Kommunikation från server till agent



Figur 3: Kommunikation från agent till server

3.3.2 Specifik design

Som separator i protokollet, d.v.s. för att skilja informationen som skickas åt, används nyradstecken. Första raden talar alltid om vad det är för typ av data som kommer. Till exempel om en NACK ska skickas från agent till server så skickas texten ”NACK” först. På detta sätt vet mottagande sidan (servern) på ett enkelt sätt vad det är för data som ska komma. I NACK-fallet åtföljs ”NACK” av ett nyradstecken varpå vilken typ av NACK det är, t.ex.

STARTLOG. Efter detta kommer ännu ett nyradstecken varpå ett felmeddelande kommer, t.ex. ”No such file/path” om loggfilen inte fanns.

Betydelse av övriga protokollfält:

<path> - Sökväg till loggfilen.

<detect> - Typ av metod som ska användas för att upptäcka när loggfilen blivit uppdaterad.

<type> - Typ av loggfil. För närvarande finns det enbart stöd för unix syslog, kallat ”syslog”.

<time> - Tid för logghändelsen (konverterat till ett enhetligt format).

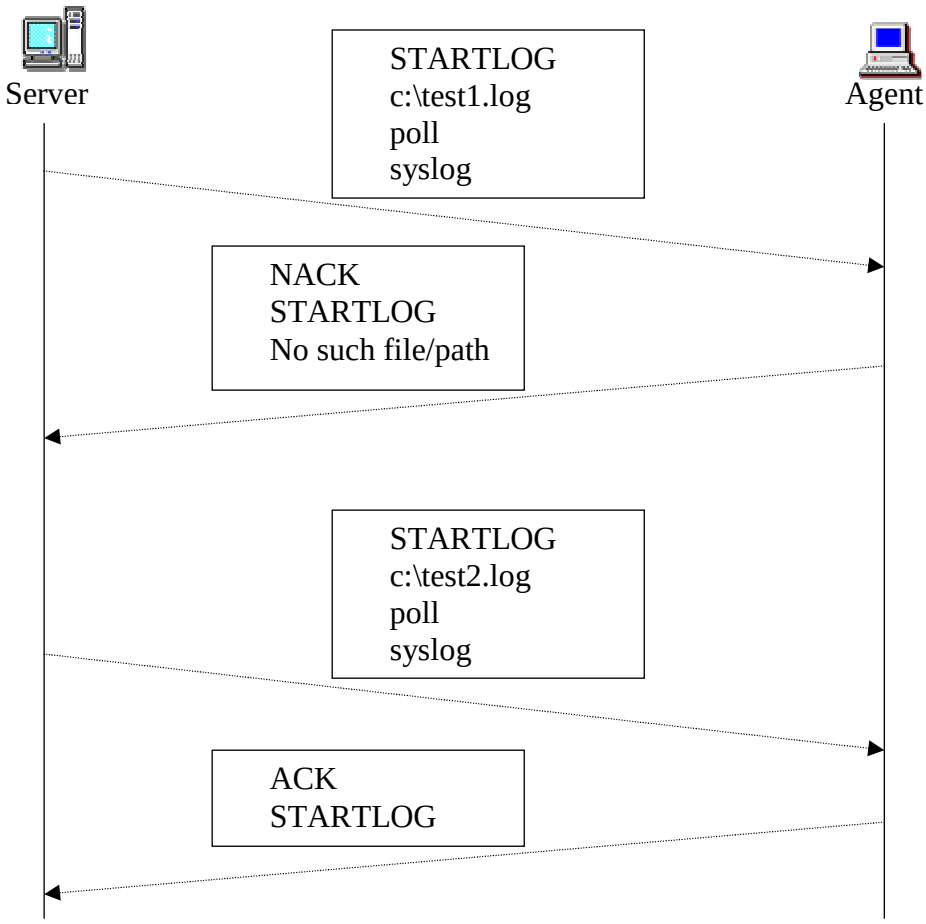
<user> - Vilken användare det gäller (om tillgängligt).

<program> - Vilket program det gäller (om tillgängligt).

<msg> - Övrig information i logghändelsen (om tillgängligt).

3.3.3 Exempelscenario

Figur 4 visar ett exempelscenario när en systemadministratör från managerkonsolen via servern vill att en agent ska övervaka en specifik loggfil. Systemadministratören skickar en begäran till agenten att övervaka loggfilen med sökvägen ”c:\test1.log” av typen syslog med poll-metoden. (Poll-metoden innebär att agenten med jämna mellanrum undersöker om loggfilen ändrats. Har den ändrats så skickas nytillkommen logginformation till servern. Detta är den hittills enda implementerade metoden i projektet, men ”detect”-fältet finns med för framtida implementation av alternativ metod.) Agenten hittar ingen fil med den sökvägen, så en NACK av typen STARTLOG skickas tillbaka till servern med felmeddelandet ”No such file/path”. Systemadministratören mottar vid managerkonsolen detta felmeddelande från servern och inser då att fel sökväg angetts och försöker igen, den här gången med sökvägen ”c:\test2.log”. Agenten hittar begärd fil vid angiven sökväg och skickar tillbaka en ACK av typen STARTLOG. Filen ”c:\test2.log” övervakas nu av agenten.



Figur 4: Exempelscenario på kommunikation mellan agent och server

3.4 Design, implementation och ändring av agenten

Ändringarna i agenten visade sig större än vad som först beräknats. Detta berodde på att agentprototypen som var framtagen i tidigare examensarbete inte passade in den nya designen där större tanke ligger på helheten av Matrix-systemet. Största ändringarna i agenten var sättet att skicka och hantera logg-händelser, samt att göra den i största möjliga mån styrd från managerkonsolen. Ett exempel ur en Unix-syslogfil:

May 12 23:25:10 veriscan42 sshd(pam_unix)[2712]: session closed for user root

I detta exempel består klockslaget för händelsen av ”May 12 23:25:10”. Med den befintliga agentprototypen kan man inte exportera tiden som en enskild post utan man är tvungen att skicka ”May” ”12” och ”23:25:10” var för sig med separata identifierare. Detta är ingen

effektiv lösning då man istället måste ta itu med problemet på server-sidan för att kunna spara händelsen, inklusive tiden för händelsen, på ett enhetligt sätt i databasen.
Konverteringen av datum till ett enhetligt format fastställdes därför till att implementeras i agenten. Detta medför att det tidigare implementerade fältet ”type”, som anger vilket format loggfilen som ska övervakas har, i agenten ändras från att kunna vara antingen ascii eller binary (binär) till vad det är för typ av loggfil som loggas, t.ex. syslog. Ändringar som krävdes för att göra agenten styrd från managerkonsolen i största möjliga mån var att ta bort allt förutom server-IP, serverport och agentnamn från konfigurationsfilen. Vid uppstart av agenten, som sker manuellt på varje enskilt datorsystem, försöker den ansluta till servern för att få information om vilka filer som eventuellt ska loggas. Om uppkoppling misslyckas försöker den igen efter 60 sekunder. Om anslutning lyckas skickar servern information om vilka filer som ska loggas, varpå agenten kontrollerar om filerna finns och skickar ACK för varje fil som existerar, respektive NACK för varje fil som ej existerar tillbaka till servern.

3.5 Implementation och utveckling av servern

Föregående examensarbete efterlämnade en enkel serverstubbe med följande funktionalitet:

- Öppna en port och låta en agent ansluta.
- Ta emot logginformation från ansluten agent.
- Skriva ut mottagen logginformation på skärmen.

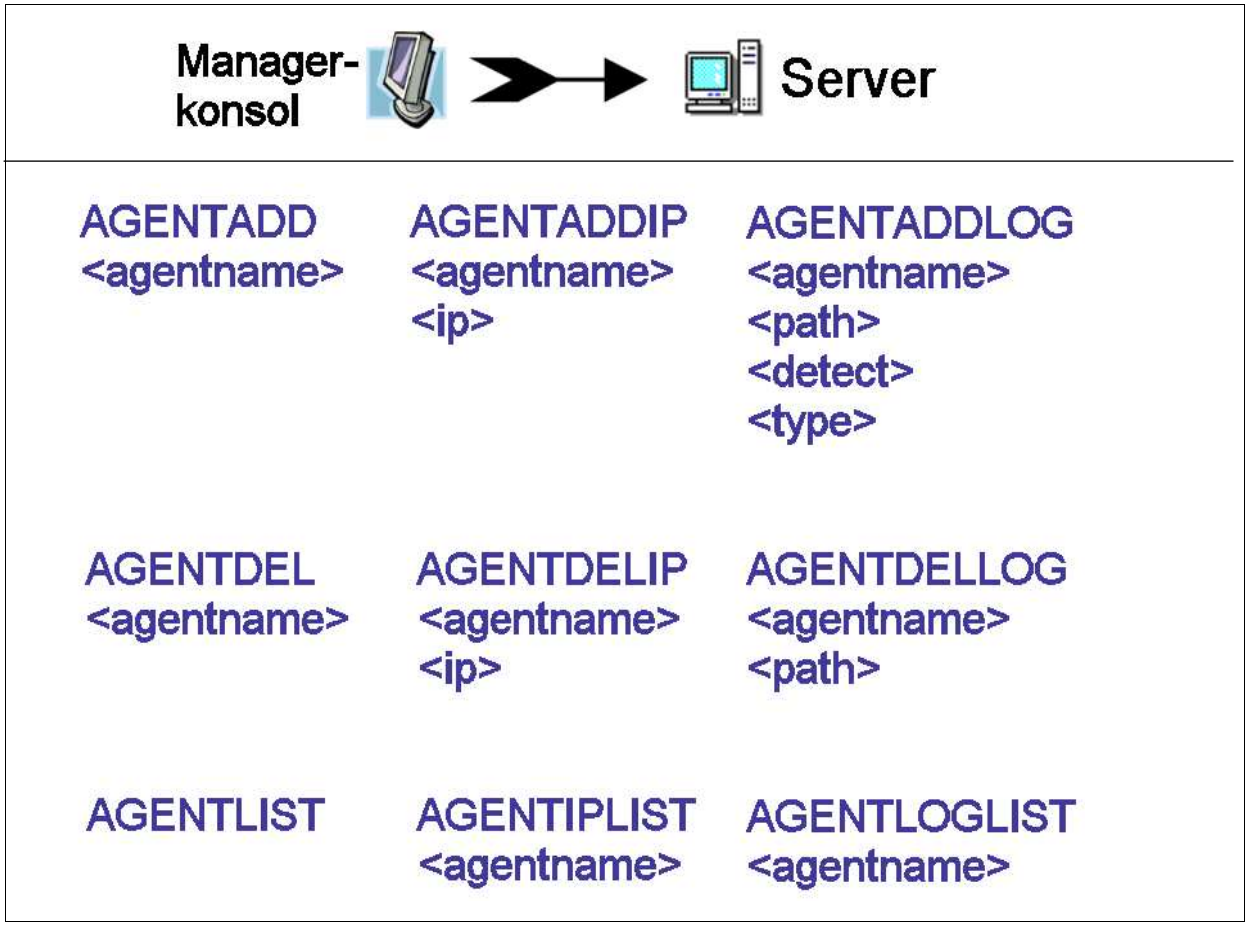
Denna server är nu helt omskriven och har stöd för de nydesignade protokollen, d.v.s. följande funktionaliteter:

- Obegränsat antal agenter kan ansluta.
- Stöd för att en managerkonsol kan ansluta.
- Dubbelriktad kommunikation mot både agent och managerkonsol.
- Kontroll att IP-adressen är legitim för en agent när den ansluter till servern.

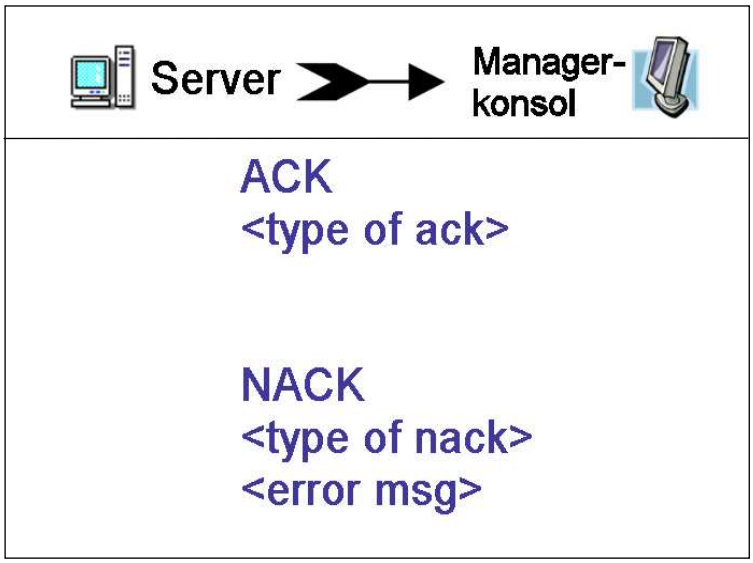
För att kunna hantera obegränsat antal agenter och en managerkonsol samtidigt används flertalet trådar. För varje agent/managerkonsol startas två nya trådar, en hanterar indata och den andra hanterar utdata.

3.6 Design av protokoll mellan server och managerkonsolen

Grunden i detta protokoll valdes till samma som mellan agent och server, d.v.s. att först ha en identifierare för att mottagaren ska veta vad det är för data som kommer, varpå datan följer (allt separerat av nyradstecken). Se Figur 5 och 6.



Figur 5: Kommunikation från managerkonsol till server



Figur 6: Kommunikation från server till managerkonsol

3.7 Design av en enkel grafisk managerkonsol

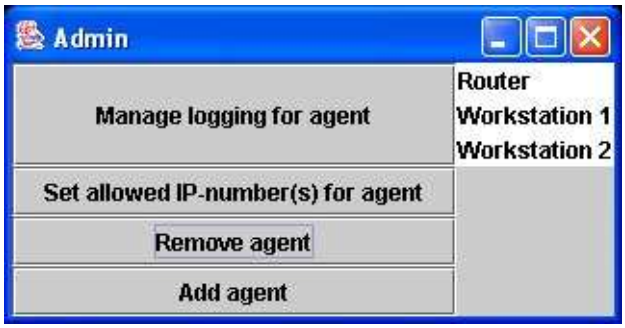
Tidsåtgången vid framför allt förändringen av agenten var massiv och när det var dags för att påbörja designen av managerkonsolen fanns p.g.a. tidsbrist två vägar att gå:

1) Avgränsa managerkonsolen.
2) Inte lägga ner så mycket tid på ett avancerat GUI, utan göra en enkel version.

Valet blev alternativ två, alltså att göra ett enkelt användargränssnitt. Designen av managerkonsolen bestod av skisser föreställande det tilltänkta GUI:et på papper som sedan visades för uppdragshandledaren [4] som förutom att godkänna den även erbjöd hjälp i början för att implementationen snabbt skulle komma igång.

3.8 Implementation av server-manager protokollet samt en enkel managerkonsol

Implementationen av server-manager protokollet samt managerkonsolen gick mycket hand i hand. Dels för att tiden var knapp, dels för att man lätt upptäckte under implementation av managerkonsolen vilken data som behövdes från servern och i vilken form/ordning datan borde skickas. Implementationen av protokollet samt managerkonsolen blev klar efter ungefär en veckas arbetstid.



Figur 7: Huvudfönster i managerkonsol

I Figur 7 ses managerkonsolens huvudfönster, som består av fyra knappar till vänster och en lista av agenter till höger. Knapparna ”Manage logging for agent”, ”Set allowed IP-number(s) for agent” samt ”Remove agent” kräver att managerkonsolanvändaren, i detta stycke kallad användare, först markerat en agent i listan. Om inte en agent är markerad när man trycker på dessa knappar visas ett felmeddelande. Om däremot en agent är markerad när man trycker på ”Manage logging for agent” kommer fönster enligt Figur 8 att visas.

Figur 8: Dialogruta ”Manage logging for agent” i managerkonsol

I denna dialogruta kan användaren lägga till logging av en specifik fil på tidigare markerad agent. Med knappen ”Remove logfile” kan användaren dessutom stoppa loggning av angiven fil. I Figur 8 finns det för tillfället ingen fil som loggas på angiven agent. Om användaren lägger till loggning av en fil kommer även en lista att synas med tillagd loggfil som enda element i listan.

Figur 9: Dialogruta ”Set allowed IP-number(s) for agent” i managerkonsol

Dialogrutan i Figur 9 visas när användaren först markerat en agent och sedan tryckt på knappen ”Set allowed IP-number(s) for agent” i managerkonsolens huvudfönster. Här kan man ställa in ifrån vilka IP-adresser specificerad agent kan ansluta till servern. Detta för att öka säkerheten så att inga obehöriga ska kunna ansluta till servern. Även här kommer en lista att synas när det finns IP-adresser tillagda för specificerad agent.

3.9 Resultat

Inom ramen för detta projekt har:

- Designats och konstruerats ett protokoll mellan agent och server, som klarar att både skicka loggposter och konfigurationsdata.
- Utförts ändringar i agenten så att den i största möjliga mån är styrd från servern samt implementerats konvertering av datum från logghändelser till ett enhetligt format.
- Designats och implementerats en server som kan handha en managerkonsol och obegränsat antal agenter samtidigt.
- Designats ett protokoll mellan managerkonsol och server som kan skicka konfigurationsdata till server med tillhörande bekräftelse.
- Tillverkats ett grafiskt användargränssnitt för managerkonsolen som är plattformsoberoende (endast Java krävs) och har funktionalitet för att lägga till, ta bort och administrera agenter.

D.v.s. samtliga delar enligt uppgiften (se kap. 2.3), förutom tidssynkroniseringen som avgränsades, har genomförts.

4 Diskussion

4.1 Analys av resultat

Författarna anser sig ha konstruerat en produkt med hög potential. Framtida arbete krävs för att få en produkt som kan fungera på marknaden. Veriscan hade inte heller för avsikt med detta examensarbete att få en färdigutvecklad produkt, utan har troligtvis tänkt utföra ytterligare examensarbete gällande produkten Matrix. Under projektets gång har mindre misstag utförts, t.ex. implementerades ibland funktioner som var onödiga eftersom de redan fanns inbyggda i programspråket. Detta måste dock anses naturligt p.g.a. projektdeltagarnas begränsade förkunskaper av Java. Tidsplanen kunde även varit bättre utformad, men även detta problem känns naturligt eftersom författarna aldrig utfört liknande arbete förut.

4.2 Kritik

- Managerkonsolen utvecklades snabbt och med bristfällig design vilket resulterade i mycket kompakt kod.
- Dokumentationen av kod, framförallt funktioner, har delvis varit bristfällig.
- Få kommentarer i koden.

4.3 Avgränsningar

I samråd med uppdragsgivaren avgränsades tidssynkroniseringen ifrån detta examensarbete. Problemet med tidssynkroniseringen är att Java inte har något inbyggt sätt att ställa systemklockan på datorutrustningen som programmet exekverar på. För att lösa detta måste istället plattformsspecifika lösningar konstrueras, vilket inte är acceptabelt eftersom ett krav på systemet Matrix är att det ska vara plattformsoberoende. Istället bestämdes att tidssynkroniseringen ska skötas med hjälp av annan programvara.

4.4 Vidare arbete

Nedan följer ett antal exempel på tänkbara vidareutvecklingar av projektet Matrix:

- Kryptering av all TCP trafik.
- Designa databas och lägga till funktionalitet i servern så att loggarna sparas i databasen.
- Lägga till filtrering av logginformation.
- Utveckla stöd för loggning av fler filtyper, t.ex. Microsoft Windows loggar.
- Utveckling av klient-GUI som verkar gentemot servern alternativt databasen.
- Implementera en dialogruta i managerkonsolen som frågar dess användare efter ett lösenord. I detta projekts managerkonsol har lösenordet hårdkodats och verifieras sedan med servern.
- Lägga till stöd i managerkonsolen samt servern för att kunna ändra lösenordet för användaren av managerkonsolen.
- Kunna ange struktur på en viss loggfiltyp genom managerkonsolen (hur loggen ser ut i t.ex. syslog).
- Designa samt besluta om vilken typ av databas som ska användas.

4.5 Allmänna reflektioner

Arbetet med systemet Matrix har varierat i svårighetsgrad under projekttiden, men på det hela taget har projektet legat på en lagom svårighetsnivå. Några större tidskrävande problem påträffades inte utan arbetet flöt på bra. Projektet har varit både lärorikt och en nyttig erfarenhet. Arbetet har varit en god försmak på att komma ut i arbetslivet.

5 Slutsats

Detta examensarbete har givit författarna både utökade programmeringskunskaper och nyttiga arbetslivserfarenheter. Vad gäller programmeringen så har detta projekt öppnat dörren till Java för projektdeltagarna, vilka anser detta som mycket värdefullt eftersom programspråket säkerligen kommer att nyttjas av dem ytterligare i framtiden. Lärdom har även tagits från den smått tröga starten av projektet, som antagligen kunde ha undvikits med en striktare och mer detaljerad planering. Produkten av detta examensarbete är ett mellanled i utvecklingen av det fullständiga systemet Matrix, så resultatet av projektet kommer antagligen att utvecklas ytterligare av Veriscan. Eventuellt har Veriscan för avsikt att handha ytterligare examensarbete inom projektet Matrix. Projektdeltagarna anser det dock inte troligt att projektet kommer att bli komplett i och med detta, utan det krävs nog ännu mer arbete. Systemet Matrixs potential har av författarna hela tiden setts som mycket god. Projektdeltagarna anser att om Matrix vidareutvecklas till ett komplett system kommer det att vara ett mycket nyttigt och värdefullt verktyg för att underlätta systemadministratörers arbete. Författarna känner att fullgott arbete utförts och lägger med stolthet detta examensarbete till handlingarna.

Referenser

[1] www.veriscan.se (2003-04-10 13:20)

[2] Per-Ove Ringsby: per-ove.ringsby@veriscan.se

[3] Leander L-O, Törnqvist D, (2002). Matrix, ett loggadministreringsssystem

[4] Robert Gustavsson: robert.gustavsson@veriscan.se