FOI-R--1019--SE
November 2003
ISSN 1650-1942

Användarrapport

Mats Persson

Tilltro och policies för mobil kod i heterogena nät och system

Ledningssystem
581 11 Linköping

TOTALFÖRSVARETS FORSKNINGSINSTITUT

Ledningssystem
Box 1165
581 11 Linköping

FOI-R--1019--SE

November 2003

ISSN 1650-1942

Användarrapport

Mats Persson

Tilltro och policies för mobil kod i heterogena nät och system

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 Linköping	**Rapportnummer, ISRN** FOI-R--1019--SE	**Klassificering** Användarrapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år November 2003	**Projektnummer** E7083
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41 Ledning med samband och telekom och IT-system	
Författare/redaktör Mats Persson	**Projektledare** Alf Bengtsson	
	Godkänd av Martin Rantzer	
	Uppdragsgivare/kundbeteckning FM	
	Tekniskt och/eller vetenskapligt ansvarig Alf Bengtsson	

Rapportens titel

Tilltro och policies för mobil kod i heterogena nät och system

Sammanfattning (högst 200 ord)

För det framtida militära ledningssystemet ställs en del särskilda krav på den tekniska arkitekturen som skiljer sig från den kommersiellt tillgängliga tekniken. Ledningssystemet skall vara flexibelt i meningen dynamiskt, decentraliserat, interoperabelt och mobilt. Detta innebär bland annat att man måste använda speciella metoder för hantering av policies och tilltro (värdering av förmågan hos något att tillfredställande utföra en handling). Detta är speciellt viktigt när det gäller de olika formerna av mobil kod. I denna rapport diskuteras en idéskiss, som innebär att man i varje nod i nätet inför en särskild komponent vilken agerar som en slags pålitlig vakt. Denna komponent bestämmer sig för vilka andra komponenter den skall lita på, avgör vilka policies som skall gälla och kontrollerar trafiken mellan nod och nät.

Nyckelord

IT-säkerhet, tilltro, policies, mobil kod

Övriga bibliografiska uppgifter	**Språk** Svenska
ISSN 1650-1942	**Antal sidor:** 18 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	FOI-R--1019--SE	User report
	Programme Areas 4. C4ISR	
	Month year November 2003	**Project no.** E7083
	General Research Areas 5. Commissioned Research	
	Subcategories 41 C4I	

Author/s (editor/s)	Project manager
Mats Persson	Alf Bengtsson
	Approved by Martin Rantzer
	Sponsoring agency FM
	Scientifically and technically responsible Alf Bengtsson

Report title (In translation)

Trust and Policies for Mobile Code in Heterogenous Networks

Abstract (not more than 200 words)

The requirements on the future military command and control system and its technical architecture differ somewhat from the available commercial technology. The command and control system should be flexible in the meaning of dynamic, decentralised, interoperable and mobile. This means that different methods for trust and security policy management must be utilized. This is especially important when concerning the different forms of mobile code. The outline discussed in this report shows how a special component, that acts lika a trusted guard, is inserted in each node in the net. This component decides which other components it should trust, decides which policies are in force and controls the traffic between the node and network.

Keywords

Computer security, trust, policies, mobile code

Further bibliographic information	Language Swedish
ISSN 1650-1942	**Pages** 18 p.
	Price acc. to pricelist

Tilltro och policies för mobil kod i heterogena nät och system

Mats Persson

26 november 2003

Sammanfattning

För det framtida militära ledningssystemet ställs en del särskilda krav på den tekniska arkitekturen som skiljer sig från den kommersiellt tillgängliga tekniken. Ledningssystemet skall vara flexibelt i meningen dynamiskt, decentraliserat, interoperabelt och mobilt. Detta innebär bland annat att man måste använda speciella metoder för hantering av policies och tilltro (värdering av förmågan hos något att tillfredställande utföra en handling). Detta är speciellt viktigt när det gäller de olika formerna av mobil kod. I denna rapport diskuteras en idéskiss, som innebär att man i varje nod i nätet inför en särskild komponent vilken agerar som en slags pålitlig vakt. Denna komponent bestämmer sig för vilka andra komponenter den skall lita på, avgör vilka policies som skall gälla och kontrollerar trafiken mellan nod och nät.

Innehåll

1 Inledning

Det finns två viktiga säkerhetsfrågeställningar för mobil kod i heterogena system. Dessa två är tilltro och säkerhetspolicies. När det gäller tilltro är frågan hur mycket förtroende man ska ha för okända noder oberoende om man kan autenticera dem eller ej. För säkerhetspolicies är frågan hur man undviker brister och konflikter i mellan policies i de ingående systemen.

Denna rapport kommer att diskutera dessa frågeställningar om tilltro och policies för mobil kod. Till att börja med ges en del bakgrund om projektet och dess inriktning. En del relaterad forskning tas också upp i bakgrunden. De följande kapitlen visar och diskuterar kraven på det framtida ledningssystemet. Rapporten avslutar med några idéskisser till lösningar på problemen.

De centrala utgångspunkterna och frågeställningarna i denna rapport är: mobil kod är det farligaste säkerhetshotet, hur hanterar man säkerhetspolicies i ett distribuerat system, hur hanterar man tilltro, och hur går detta ihop med sammankopplade system av system?

2 Bakgrund

Traditionella distribuerade system kan bestå av ett antal sammankopplade datorer med en gemensam mellanskiktsnivå. Med detta menas gemensamma gränssnitt och protokoll för kommunikation, vilka ofta är skrivna i samma programspråk. I någon mening är det ett enda system utspritt på flera datorer. I ett sådant system kan alla lita på alla, eller åtminstone på en central server. Systemet har i stort sett gemensamma säkerhetsregler. Ofta är systemet isolerat från resten av Internet. Möjligen kommunicerar det med några andra system, eller är åtminstone skyddat bakom en brandvägg om det är kopplat till Internet.

I datorsystem där man väljer att vara mer öppen och kommunicera med flera andra system måste man vara beredd på mer svårlösta problem. Till att börja med måste man bestämma sig för vilka öppna och gemensamma standardiserade protokoll man ska utnyttja. Det finns många öppna protokoll, till exempel HTTP-protokollet. En annan fråga man måste lösa är vilka säkerhetspolicies man ska ha, och vidare vilken autenticeringsmekanism och åtkomstkontroll man behöver. Frågan blir svårare av att det finns många olika system inblandade. Den tredje frågan är vem man ska lita på. Man brukar oftast lita på sig själv, men vilka andra okända system kan man ha förtroende för?

Mobil kod introducerar alltid ytterligare säkerhetsfrågeställningar. Slipper man problemen om man spärrar all form av mobil kod?

De följande underkapitlen utreder en del begrepp och visar även på relaterade forskningsartiklar.

2.1 System av system

Projektet “System av system” startade 2003 och har för närvarande två inriktningar. En inriktning där vi tittar på skydd av system, främst identitetsverifiering och namngivning. Den andra inriktningen studerar mobil kod i systemen - främst skillnader i policies mellan systemen. Detta projekt har beskrivit målbilden på följande sätt:

> Det framtida tekniska ledningssystemet måste byggas med en evolutionär utvecklingsmodell. Systemet kommer att innehålla många delsystem som har en lång kravlista. Dessa delsystem kommer att vara oberoende av varandra, självständiga, mobila och interoperabla med andra typer av system. Delsystemen ska vara lätta att byta ut eller modifiera.

Den beskrivningen skiljer sig en del från, och är till en viss grad oförenligt med, det idealiska och säkra militära datornätet. Ett sådant borde ha dessa egenskaper:

- Litet nät med få noder.
- Säkra kommunikationsprotokoll som IPSEC eller SSL.
- Hög tilltro på varje ingående nod.
- Varje nod är hårt säkrad och verifierad.
- Det finns en gemensam auktoritet i nätet som alla litar på (kap 2.5).
- Minimal mängd mobil kod, möjligen enbart i form av säkerhetspatchar.

Men det är inte ett sådant system som denna rapport har i åtanke utan ett som beskrivs lite mer i detalj i kapitel 3.

2.2 Webtjänster

I ett distribuerat heterogent system kommer det att finnas en mängd tjänster. Den tekniska utvecklingen pekar ut webtjänster (Web Services) som den mest troliga tekniken.

En webtjänst är en tjänst som är tillgänglig över ett nätverk, som använder sig av standardiserade XML-meddelanden, och som inte är bundet till

ett särskilt operativsystem eller programspråk. Webtjänster tillhandahåller operationer eller metoder via nätet som andra applikationer kan anropa. Detta innebär alltså att en applikation på en dator i nätet kan skicka förfrågningar till en annan dator i nätet och få ett svar. Webtjänster kan använda flera olika underliggande protokoll, men det vanligaste är HTTP. Därav namnet “webtjänst”.

Webtjänster använder sig av öppna protokoll och formatet på XML-meddelandena är standardiserat. Det protokoll som specificerar formatet på dessa meddelanden kallas SOAP (Simple Object Access Protocol). Webtjänster är även självbeskrivande och dessa skrivs i WSDL (Web Services Description Language). Dessa tjänster kan sedan registreras på en UDDI-server (Universal Discovery Description Integration) i nätet så att andra kan hitta dem [1].

I och med denna öppenhet och standardisering kan man skriva webtjänster i vilket programspråk som helst, och på vilket plattform man vill. Detta gör att webtjänster har hög interoperabilitet. På samma sätt som nästan alla datorer kan hantera IP-protokollet på nätverksnivå, kanske alla datorer kommer att förstå webtjänster på “tjänste-nivån”.

Det kan vara värt att notera att webtjänster inte specificerar vad som skickas i meddelandena, utan dessa kan innehålla vilka data som helst. Det går utmärkt att skicka mobil kod via en webtjänst, till exempel kan man konstruera en tjänst som beskriver sig som “execute” och exekverar den kod skickas till den.

För att åskådliggöra vilka risker som kan uppkomma i webtjänster så ges här ett exempel på farlig kod som utnyttjar en dåligt skyddad webtjänst. Exempelkoden söker igenom alla möjliga kreditkortsnummer, kontrollerar deras validitet, och skriver sedan ut de giltiga numren i en lista.

```
foreach number in 0000000000000000..9999999999999999 {
  if (requestWebService("www.ccvalidator.com",
                        "ValidateCreditCardNo",
                        number) == "OK")  print number
}
```

2.3 Mobil kod

Mobil kod är programkod som förflyttar sig runt i ett nätverk och exekverar på datorerna. För att den mobila koden ska kunna exekvera måste det finnas en exekveringsmekanism på den dator där koden ska köras. Koden måste även ha tillräckliga rättigheter att utföra nödvändiga operationer.

Hur mycket mobil kod kan man förvänta sig skickas genom webtjänster? Eftersom det inte finns någon begränsning i vad ett XML-meddelande kan innehålla, så kan man anta att det kommer att göras. Förhoppningsvis

är de som bygger sådana system medvetna om riskerna med mobil kod och lägger in lämplig skydd i form av sandlådor. Ett intressant exempel i detta sammanhang är exekverbar XML-kod kallad “o:XML” [2] som är ett programspråk där man skriver koden i XML.

Är systemet med webtjänster i sig självt en distribuerad dator där enskilda system exekverar, och därmed har samma problem som vanlig mobil kod? Likheterna kan uttryckas så här: mobil kod anropar funktioner i den dator den exekverar; en mobil enhet som dyker upp i ett nytt system anropar webtjänster i det nya system där den blivit ansluten. Säkerhetsproblemen borde bli samma i båda fallen. Däremot är skillnaden att den mobila enheten utnyttjar sin egen processorkraft, men utnyttjar nätverket desto mer.

Alltså kan man säga att området programexekvering i distribuerade nät inkluderar både mobil kod och webtjänster. Nätet av webtjänster utgör en exekveringsomgivning med motsvarande risk för brister som inuti en enskild dator.

Det finns mycket som kan klassificeras som mobil kod, till exempel maskar och virus, distribution av säkerhetspatchar och programuppgraderingar, eller självgående mobila agenter.

En mobil agent kan tillhöra ett system, vandra in i ett annat och kommunicera med ett tredje system. Agenten kan bli begränsad av olika behörigheter i flera nivåer, kan bära med sig rättigheter till speciella resurser, samtidigt som den stöter på brandväggar som kan sätta den i karantän. Situationen blir snart väldigt komplex och risken för säkerhetsluckor ökar om man lättar på reglerna för att få agenten att fungera.

2.4 Policies

De tre grundpelarna för datasäkerhet är sekretess, integritet och tillgänglighet. Säkerhetspolicies är regler och förordningar som beskriver hur sekretessen ska hanteras, hur integriteten ska hållas, och vilka som ska få tillgång till vad. Man kan säga att säkerhetspolicies är det som beskriver hela säkerhetssystemet. Policies kan vara statiska eller dynamiska som förändras över tiden. Man kan härleda alla säkerhetsangrepp till brister i policies, att de saknas eller har inte följts (ofta på grund av implementationsfel).

Exempel på enklare policies är brandväggsregler eller åtkomstregler för ett filsystem. Dessa regler är ofta enkelt beskrivna med instruktioner: <Tillåt / Förbjud> tillgång till <object> från <subject>. Ett antal sådana regler utgör då policyn. Det kan också finnas mer generella policies som “Tillåt allt som inte uttryckligen förbjuds”, eller dess motsats “Förbjud allt som inte uttryckligen tillåts”. Det senare är vanligt i säkerhetskritiska system.

En något mer sofistikerad idé är att göra ett "programmerbart" policysystem. Ett exempel är LGI (Law-Governed Interaction) [3] [4] som hanterar dynamiska policies i distribuerade system. En policy i LGI kan uppdateras dynamiskt från andra datorer i systemet. Vilka som får uppdatera andras policies styrs också av policysystemet. Det är alltså policysystem med tillstånd som kan förändras. LGI har ett Prolog-liknande språk för att programmera policies.

I detta sammanhang är det lämpligt att nämna det policysystem som är tänkt att användas i webtjänster. Detta kallas XACML (Extensible Access Control Markup Language). XACML definierar vilka som får tillgång till informationen, vilken information som är tillgänglig, när man får tillgång till informationen och hur man får tillgång till informationen [5]. Ett XACML-dokument är skrivet med XML-syntax och innehåller fyra delar: vem, vad, hur, och villkor. Dessa delar bildar en policy som sedan någon auktoriseringstjänst kan tolka och från denna dela ut eller neka rättigheter.

Policies är ett viktigt redskap för att hantera säkerheten för mobil kod. Det bör finnas regler som avgör om mobil kod får exekveras, med villkoret att den är autenticerad, och i så fall vilka resurser den får utnyttja [6]. Problem uppkommer när de olika systemen den mobila koden vandrar genom har olika policies. Till exempel kan en policy vara att en mobil agent i ett system har rätt att få reda på alla användarnamn men kan inte göra något inloggningsförsök med något användarnamn. Ett annat system kan däremot ha en policy att inte lämna ut några namn, men tillåta inloggningsförsök.

Problemet är alltså att policysystem bygger på att alla ingående system använder samma regelbas. Det går inte att ett av systemen kommunicerar utan kryptering när alla andra krypterar. Ur säkerhetssynvinkel krävs att det finns en central databas med policies som helst ska vara statiska. Dynamisk ihopkoppling av ad-hoc system kan resultera i ett kaos av motsägelsefulla och felaktiga policies.

Det finns forskning om säkerhetspolicies för mobil kod [7] där man konstruerat ett policyspråk, kallat SPL (Security Policy Language), och har en central policymonitor som tar hand om alla förfrågningar från det distribuerade systemet.

2.5 Tilltro

Tilltro är egentligen en subjektiv värdering av någon annans förmåga att utföra en handling på ett tillfredsställande sätt. I datorsammanhang betyder tilltro en värdering av säkerhetsnivån i datorn, kommunikationskanaler eller den omgivning som datorn befinner sig. Man kan ungefär säga att tilltron är summan av tilltron till de ingående delarna i systemet.

I ett distribuerat system har man vanligtvis en gemensam aktör i nätet

som alla litar på, en så kallad Trusted Authority (TA). Denna aktör utför säkerhetskritiska uppgifter såsom betrodd part i säkerhetsprotokoll, tillverkar och delar ut krypteringsnycklar eller är kontrollant av säkerheten i programkomponenter i form av mobil kod. Problemet med TA är att man i väldigt stora distribuerade system, eller sammankopplade system, inte kan komma överens om vem man litar mest på, eller om den man litar mest på är tillräckligt pålitlig. Hur ska man då hantera tilltro i ett distribuerat system utan en central TA?

Tilltro är transitiv under vissa villkor. Om A litar på B och B litar på C, så litar A på C, under villkoret att A litar på B:s förmåga att bedöma pålitligheten hos C. Det vill säga, A behöver inte lita på B:s alla förmågor utan bara vissa specifika. I en artikel om en distribuerad tilltro-modell [8] tas detta upp och där ges även ett förslag på hur man rent numeriskt kan värdera tilltro.

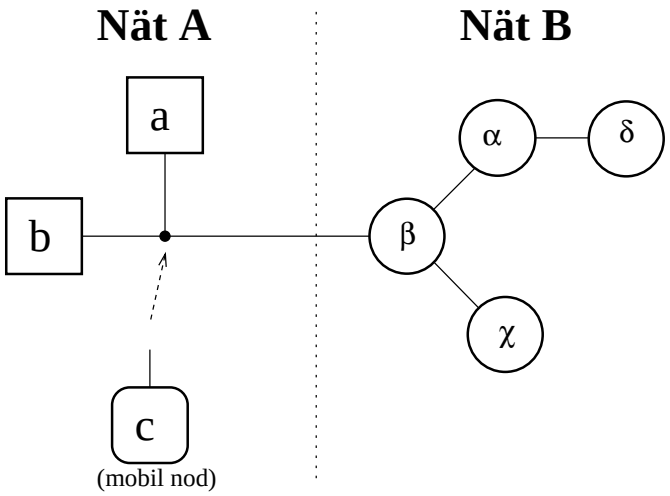
Om man underlåter att sätta upp villkor för den transitiva tilltron eller utnyttjar transitiviteten rakt av så kan detta utnyttjas för säkerhetsattacker. Det innebär att man skaffar sig högre tilltro genom att gå via ett annat objekt och utnyttja systemets tilltro till detta objekt. Det kallas ofta i litteraturen för “elevated trust” eller “luring attack”. I .NET, som är ett distribuerat system, försöker man lösa detta genom att kontrollera tilltron till varje objekt i anropskedjan [9].

Det finns annan forskning om upprättande av tilltro-relationer inom mobila adhoc-nät [10]. Denna bygger på att varje mobil nod i nätet samlar in stöd för hur mycket man bör lita på andra noder, summerar dessa och bestämmer sig sedan för “hög”, “medium”,”låg”,”ingen” tilltro till en annan nod. Denna metod liknar den aritmetik som beskrivs i [8]. Om både B och C litar på D, så kan A också tänka sig att lita på D. Detta liknar det naturliga sättet för till exempel när människor upprättar förtroende för varandra. Det som kanske saknas hos dessa artificiella aritmetiker för tilltro är en tidsaspekt. Normalt ökar förtroendet för någon ju längre tiden går, och inget negativt händer.

3 Krav på framtida arkitektur

Det finns ett antal krav på det framtida datoriserade ledningssystemet. Ett enkelt exempel på ett sådant system visas i figur 1. Kraven på detta nya system kan man sammanfatta med en enkel lista som innehåller några önskvärda egenskaper:

- Systemet skall kunna hantera både dynamisk avknoppning och ihopkoppling av noder eller system.
- Noderna i nätet ska kunna vara mobila.



Figur 1: *Ett system bestående av två olika sammankopplade nät, varav ett med en mobil nod*

- Systemet ska vara interoperabelt med andra nationers system.
- Noderna ska kunna fungera självständigt utan beroende av centrala delar.
- Det finns tjänster i nätet, då i form av webtjänster. Dessa kan i någon omfattning hantera mobil kod.
- Nätets utseende ska kunna konfigureras på många olika sätt.
- Ett mer övergripande krav är att det ska kunna hantera system av system. Delsystem bildar ett större system när de ihopkopplas.

Kraven på arkitekturen kan sammanfattas med dessa ord: dynamiskt, decentraliserat, interoperabelt, mobilt.

4 Problemformulering

Ur de tidigare bakgrundsbeskrivningarna och de krav som ställs på en framtida arkitektur, kan man ta fram ett antal centrala problemställningar:

- I ett system av system kan det bli ett problem att få fram en betrodd auktoritet (TA) som alla litar på. Då får man vanligen nöja sig med någon som man nästan litar på. Detta gör att autenticeringen blir svagare och tyngdpunkten måste istället ligga i andra skyddsmekanismer.
- Säkerhetskrav gör det nödvändigt att autenticera mobil kod i distribuerade nät. Detta ställer högre krav på en korrekt policy i systemen.

- Ett distribuerat nät med ett antal olika system kan ha varierande säkerhetspolicies eftersom det administreras av olika personer på olika sätt. Olika system kan ha helt olika policies för hur inkommande data, anrop eller mobil kod hanteras. Detta innebär att en angripare kan söka igenom nätet efter den information eller funktionalitet som har lägst restriktioner, och sedan utnyttja detta för att skaffa sig mer information, skaffa sig högre rättigheter eller orsaka annan skada.
- Mobila noder kan koppla in sig på olämpliga platser i nätet och där anropa funktioner på sådant sätt som ej var tänkt, på samma sätt som mobil kod.

Det finns förmodligen ännu fler problem än de ovanstående, men de är mest framträdande när det gäller mobil kod. De två viktigaste problemen som man bör lösa är att det kan saknas en gemensam TA och att man måste ha ett system för att synkronisera policies så att de blir likadana mellan systemen.

5 Lösningar

Det finns både färdiga produkter och forskning inom området tilltro och policyhantering. De färdiga produkterna är dock i hög grad centraliserade. Forskning har visat några andra lösningar på problemen. Några av dessa lösningar har till viss del redan beskrivits i kapitel 2.

5.1 Distribuerad policyhantering

Det finns två metoder för att hantera och distribuera olika policies:

- Gemensam policyserver som övervakar och bestämmer policies i nätets noderna. Som ett exempel finns denna lösning i 3Com's policyserver som styr brandväggskort [11].
- En distribuerad policyhantering där noderna styr varandra och uppdaterar policies enligt något regelsystem [12]. Detta har beskrivits tidigare i kapitel 2.4

5.2 Distribuerad tilltro

En ännu svårare fråga är att hantera tilltro. Hur hantera tilltro mellan helt okända parter? Det finns ett traditionellt sätt att lösa detta, samt två andra där man bygger upp en tilltro med olika metoder. De senare är egentligen det mer naturliga eftersom människor beter sig på samma sätt när de bygger upp tilltro till varandra.

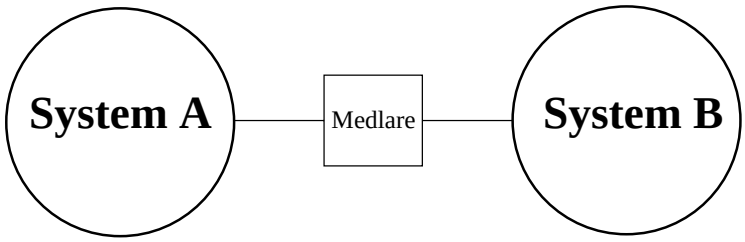
- Det traditionella sättet är att komma överens om en gemensam central auktoritet som alla litar på. Denna auktoritet har till uppgift att säkerställa aktiviteter, i synnerhet autenticering men även kontroll av mobil kod i vissa system.

- Man bygger upp relationer av tilltro med hjälp av rekommendationer från andra noder som man litar på [8]. Denna modell har tagits upp tidigare i kapitel 2.5. Kortfattat innebär den att en nod sätter ett numeriskt värde på hur mycket den litar på en annan nod genom att rent aritmetiskt beräkna värde utifrån rekommendationer från övriga noder.

- Man samlar in stöd för uppbyggnad av tilltro från flera olika typer av källor såsom identitet, nycklar, position, version på säkerhetspatchar, tidigare beteende, et cetera [10]. Denna metod använder sedan ett logiskt resonemang för att räkna ut hur mycket man kan lita på någon annan.

5.3 Proxies och medlare

Ett annat sätt att lösa problemen med tilltro och policies är att låta en självständig nod i nätet agera proxy eller medlare. Denna kan fungera som en mellanhand som alla mer eller mindre litar på. Det är ungefär så som dagens brandväggar fungerar. En principskiss visas i figur 2.



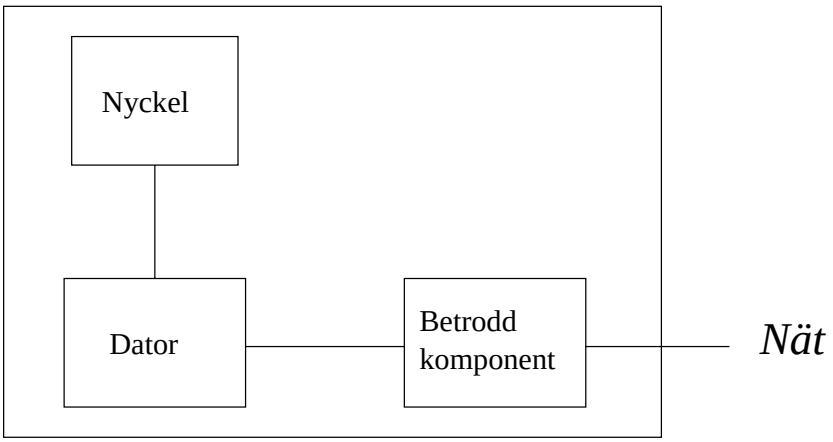
Figur 2: *En medlare mellan två system.*

Detta är inte en särskilt bra lösning eftersom någon måste tillverka medlaren, och då litar kanske inte andra på den. Dessutom ger den enbart separering av näten i skyddade zoner, vilket i och för sig kan vara bra, men det ger inte tillräckligt skydd för de enskilda noderna eller skydd av nätet från de enskilda noderna - den så kallade “insider-problematiken”.

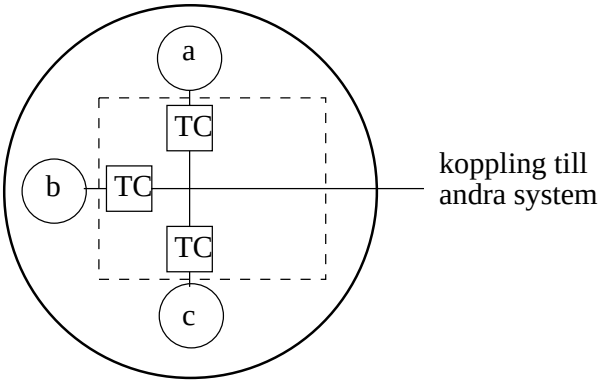
5.4 Idéskiss till “Trusted Component”

En idé är att konstruera en hårdvarubaserad enhet som har sin egen programvara för att övervaka och styra nätverkstrafiken mellan en dator och

resten av nätverket. Denna enhet kan ha formen av ett smartcard eller kretskort med ett inbyggt enkelt operativsystem. Tanken är att denna enhet ska ersätta nätverkskortet. I figur 3 visas en enkel bild på var den sitter i datorn. Låt oss kalla denna enhet för “betrodd komponent” eller “trusted component” (TC). En principskiss på hur denna enhet ska sitta i nätverket visas i figur 4.



Figur 3: *En enhet bestående av en dator, en nyckel som autenticerar mot datorn, och en komponent “betrodd komponent”*



Figur 4: *En system bestående av tre noder a,b,c med vardera en TC. Dessa tre TC bildar en gemensam medlare.*

Denna betrodda komponent kan ha flera funktioner. Ur de tidigare föreslagna lösningarna i kapitel 5.1 och 5.2 kan man tänka sig att komponenten ska hantera distribuerade policies och bygga upp förtroenden med andra komponenter i nätet. Det finns även annan funktionalitet som man skulle vilja lägga i denna komponent. En TC kan innehålla program- eller hårdvara för:

- Handskaknings- och kommunikationsprotokoll för att upprätta nya förbindelser med okända TC eller återuppta gamla.

- Autenticering eller identifiering. Detta kan göras med certifikat för att ge ett högt förtroende, eller genom identifiering med hjälp av ip-nummer vilket förstås ger ett lägre förtroende.
- Upprättande av tilltro genom någon slags bevisning eller värdering av andra TC i nätet. Här har man även en tidsaspekt då någon som har uppträtt korrekt under en längre tid får ett högre förtroende.
- Utbyte och uppdatering av policyregler. Flera TC kan bilda en grupp som uppdaterar varandras regler för att hålla dem konsistenta.
- Brandvägg, proxy och filterfunktionalitet för alla nätverkslager. Från paketnivå till tjänstenivå. Den kan alltså fungera som ett email-filter eller som en brandvägg för webtjänster.
- Agera som nätsensor för intrångsdetektering.
- Kryptering och dekryptering för att avlasta CPUn i värddatorn.
- Routing. Om en TC har flera nätverksanslutningar kan den agera som en router.
- Sammanställning av ett antal andra närliggande TC och den gemensamma tilltro och policydatabasen.

En TC blir förhoppningsvis en mer kompakt och kanske en bättre lösning än att integrera samma funktionalitet i värddatorns operativsystem. De tidigare uppräknade decentraliserade lösningarna var tänkta som en del av operativsystemet eftersom det är lättast att bygga dessa lösningar med hjälp av programvara. En nackdel är att de blir känsligare för angrepp eftersom de lättare kan sättas ur spel genom några kommandon från till exempel ett virus eller en mask.

Fördelarna med en TC är flera. Man får säkerhetsfunktionaliteten samlat på ett ställe vilket gör det enklare att analysera; man kan bygga en manipuleringssäker komponent; man får även en avlastning av värddatorns CPU. En nackdel är att man kan få en ganska hög komplexitet i en TC då den kan innehålla ett helt operativsystem (som då visserligen är reducerat).

Det finns redan en variant på en TC i form av ett nätverkskort med inbyggda brandväggar [11] där det som motsvarar TC kallas Policy Enforcement Point (PEP). Något som också liknar TC är det som tidigare kallades “Palladium” av Microsoft, men som numera heter Next-Generation Secure Computing Base [13].

Tidigare i kapitel 5.3 nämndes att det kunde vara lämpligt med någon slags medlare i nätet. I detta fall blir medlaren det virtuella konglomeratet[1] av betrodda komponenter som kommit överens om hur mycket de litar på varandra och vilka regler som gäller.

[1] Samling av löst förenade element utan inre samband med varandra.

6 Diskussion och slutsatser

Den betrodda komponenten som beskrivs i kapitel 5.4 är bara på idéstadiet. Det finns fortfarande en del att utveckla på denna. Vilka funktioner ska en sådan komponent ha? Hur ska protokollet mellan komponenter se ut? Hur bygger man upp ett förtroende? Hur styr man de distribuerade policyreglerna? Hur administreras den betrodda komponenten?

Det är sannolikt att vi i framtiden kommer att se ännu mer mobil kod, men då dolt under andra benämningar eller bakom annan funktionalitet. Många lösningar på säkerhetsproblem har en tendens att inkludera någon form av mobil kod, till exempel automatisk distribution av säkerhetspatchar. Det blir då speciellt viktigt att kontrollera varifrån dessa kommer, hur mycket man kan lita på de som signerat dem, och om man vågar installera samt köra dem.

En nackdel med många säkerhetslösningar är att de baseras på en central administration. Till exempel har många företag en central brandvägg som även kan fungera som webproxy. Övervakningssystem och intrångsdetekteringssystem sitter också på en central plats i nätet. Det kan vara svårt att komma ifrån detta eftersom det är så praktiskt vid systemadministration. Men om man vill ha ett mobilt och dynamiskt nät av heterogena system måste man hitta andra lösningar.

Framtida forskningsmöjligheter

En tidigare rapport om aktiva nät [14] tog upp möjligheterna att lägga in en aktiv nod i varje dator. Det skulle vara intressant att kombinera detta med den betrodda komponenten som diskuteras i denna rapport.

Ett annat område värt att kombineras med idéerna i denna rapport är distribuerad intrångsdetektering. En speciell del i den betrodda komponenten kan agera som nätsensor. Den skulle kunna analysera nätverkstrafik och notera vilka policy- eller brandväggsregler som appliceras. Baserat på denna analys kan den skicka varningar till andra komponenter i nätet som då kan anpassa sig själva till den nya situationen. Alternativt kan varningarna skickas till en central tjänst som sammanställer och presenterar dessa för en mänsklig övervakare.

Ett annat problem som kanske borde studeras närmare är vilka säkerhetsrisker som finns vid anrop av distribuerade webtjänster. En distribuerad samling av webtjänster kan liknas vid en “global dator” där webtjänsterna agerar som operativsystemsanrop. Den kanske har samma säkerhetsproblem som ett vanligt operativsystem.

Referenser

[1] H. Deitel, *Web Services: A Technical Introduction*. Prentice Hall, 2003. ISBN 0-13-046135-0.

[2] "o:XML – Objectoriented XML." `http://www.o-xml.org/`.

[3] N. Minsky, "Coordination and trust in open distributed systems," tech. rep., Rutgers University, LCSR, 1995.

[4] N. Minsky, V. Ungureanu, W. Wang, and J. Zhang, "Building reconfiguration primitives into the law of a system," in *Proceedings of the Third International Conference on Configurable Distributed Systems*, pp. 89–97, May 1996.

[5] "OASIS eXtensible Access Control Markup Language (XACML)." `http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf`.

[6] M. Persson, "Mobil kod och säker exekvering," Användarrapport FOA-R–98-00807-503–SE, FOA, april 1998.

[7] C. Ribeiro and P. Guedes, "SPL: an access control language for security policies with complex constraints," in *Network and Distributed System Security Symposium, NDSS'01*, february 2001.

[8] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *New Security Paradigms 97*, pp. 48–60, 1997.

[9] B. A. LaMacchia and S. Lange, *.NET Framework Security*. Addison Wesley, 2002. ISBN 0-672-32184-X.

[10] L. Eschenauer, V. D. Gligor, and J. Baras, "On trust establishment in mobile ad-hoc networks," tech. rep., University of Maryland, ISR, 2002.

[11] T. Markham and C. Payne, "Security at the network edge: A distributed firewall architecture," in *Proceedings of DISCEX II*, 2001.

[12] N. Minsky and V. Ungureanu, "Unified support for heterogeneous security policies in distributed systems," *Proceedings of the 7th security conference. (USENIX Association: Berkeley, CA)*, January 1998.

[13] P. England, B. Lampson, J. Manferdelli, M. Peinado, and B. Willman, "A Trusted Open Platform," *IEEE Computer*, pp. 55–62, July 2003.

[14] M. Persson, "Säkerhet i aktiva nät," Användarrapport FOI-R–0309–SE, FOI, december 2001.