Bidrag till en IT-säkerhetsarkitektur för GTRS

Identifierade säkerhetsfunktioner och säkerhetsmekanismer

AMUND HUNSTAD, HENRIK KARLZÉN, JACOB LÖFVENBERG

FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

FOI
Totalförsvarets forskningsinstitut
Informationssystem
Box 1165
581 11 Linköping

Tel: 013-37 80 00
Fax: 013-37 81 00

www.foi.se

FOI-R--3148--SE
ISSN 1650-1942

Underlagsrapport
December 2010

Informationssystem

Amund Hunstad, Henrik Karlzén, Jacob Löfvenberg

Bidrag till en IT-säkerhetsarkitektur för GTRS

Identifierade säkerhetsfunktioner och säkerhetsmekanismer

Titel	Bidrag till en IT-säkerhetsarkitektur för GTRS: Identifierade säkerhetsfunktioner och säkerhetsmekanismer
Title	Contributions to an IT Security Architecture for GTRS: Identified security functions and mechanisms
Rapportnr / Report No.	FOI-R--3148--SE
Rapporttyp	Underlagsrapport
Report Type	Base Data Report
Månad / Month	December / December
Utgivningsår / Year	2010
Antal sidor / Pages	37
ISSN	1650-1942
Kund / Customer	FM
Projektnr / Project No.	E53195
Godkänd av / Approved by	Anders Törne

FOI, Totalförsvarets Forskningsinstitut
Avdelningen för informationssystem
Box 1165
581 11 LINKÖPING

FOI, Swedish Defence Research Agency
Information Systems
P.O. Box 1165
SE-581 11 LINKÖPING

Sammanfattning

I denna rapport redovisas resultatet av FOI:s arbete med en IT-säkerhetsarkitektur för Gemensamt Taktiskt Radiosystem (GTRS). Med utgångspunkt i FOI:s tidigare redovisade riskinventering för GTRS har ett antal problemområden identifierats genom att gruppera relaterade riskfaktorer. Grupperingen gjordes för att få ett hanterbart antal problemområden att betrakta.

Försvarsmakten har i sitt dokument ”Krav på säkerhetsfunktioner” beskrivit ett antal generella säkerhetsfunktioner för IT-system. Vi har analyserat dessa och har redovisat vilka säkerhetsfunktioner som är relevanta för vilka problemområden. I vissa fall har också ett behov av andra, nya säkerhetsfunktioner identifierats och dessa har beskrivits.

Säkerhetsfunktionerna är både generella och översiktliga, varför en nedbrytning och specificering har gjorts för att nå en detaljnivå som kan fungera som ett bidrag till en IT-säkerhetsarkitektur. Resultatet av specificeringen blev en ganska omfattande lista med säkerhetsmekanismer som på ett tydligare sätt än säkerhetsfunktionerna korresponderar till behov som problemområdena genererar.

Längs vägen har också ett antal mer specifika observationer gjorts som inte passade in i redovisningsstrukturen som beskrivs ovan. Dessa observationer, med tillhörande diskussioner, beskrivs i ett separat kapitel i rapporten.

Nyckelord: GTRS, IT-säkerhet, riskinventering, sårbarheter, arkitektur

Abstract

This report accounts for the result of a study performed at the Swedish Defence Research Agency (FOI) regarding IT security architecture for the software defined radio system GTRS. Using FOI's previously presented risk inventory for GTRS, a number of problem areas have been identified by clustering related risk factors. The clustering was done to reduce the number of problem areas to be studied to a manageable level.

In their document "Krav på säkerhetsfunktioner" the Swedish Armed Forces have described a number of general security functions for IT systems. We have analyzed these and have described which of the security functions are relevant for which problem areas. In some cases a need for other, new security functions have been identified and these have been presented.

The security functions are very general, why a decomposition and specification has been done to reach a level of detail that work as a contribution to a IT security architecture. The result of the specification was a rather comprehensive list of security mechanisms, which correspond to the needs generated by the problem areas more palpably than the security functions do.

During the project a number of more specific observations have been done, which do not fit into the report structure described above. These observations, with corresponding discussions, are presented in a separate chapter.

Keywords: GTRS, IT security, risk inventory, vulnerabilities, architecture

Innehåll

1 Inledning

Gemensamt Taktiskt Radiosystem, GTRS, är ett system som är under utveckling med sikte på att tillhandahålla förbättrade såväl som nya tjänster för Försvarsmakten. GTRS kommer att vara markant mer flexibelt än existerande kommunikationslösningar. Detta sker dock till priset av en påtagligt ökad komplexitet och komplexitet är ett problem så fort säkerhetsaspekter ska beaktas.

Det är värt att notera att GTRS brukar betecknas som en mjukvaruradio, men att det i lika hög grad är ett mjukvarukrypto. Detta är två funktioner som var för sig är mycket avancerade. Till detta kommer ett IT-system som kontrollerar krypto- och radiofunktionerna och binder samman dem till en fungerande helhet. GTRS är alltså också ett avancerat IT-system där IT-komponenterna är nödvändiga för att systemet i sin helhet ska fungera korrekt.

1.1 Syfte och problemformulering

GTRS-utvecklingen präglas av betydande grad av parallella utvecklingsinsatser, där idéer, problem och lösningar diskuteras och analyseras av en uppsättning aktörer. Detta medför kreativt, men inte alltid lättstyrd verksamhet. Anpassning av GTRS till existerande så kallade arvssystem har också signalerats som önskvärd. Utvecklingsarbetet rörande säkerhet förenklas inte av dessa omnämnda faktorer. För att kunna uppnå och påvisa lämplig nivå av IT-säkerhet är det nödvändigt med en välgrundad struktur för IT-säkerhetsfunktionerna i systemet, det vill säga en IT-säkerhetsarkitektur. Med tanke på denna situation är det önskvärt att se framåt och bedöma hur säkerheten bör hanteras och byggas upp även på sikt.

Balansgången mellan flexibilitet och komplexitet medför betydande IT-säkerhetsutmaningar. Inom FOI:s projekt om IT-säkerhet i GTRS ingår deluppgifter med fokus på att identifiera sårbarheter, analysera hur hot utnyttjar sårbarheter och vilka risker detta medför samt att med detta som utgångspunkt formulera en IT-säkerhetsarkitektur för GTRS.

Syftet med att låta FOI ta fram en möjlig IT-säkerhetsarkitektur för GTRS är att tillåta ett mer fritt och framåtblickande tänkande än vad som är möjligt i den del av GTRS-projektet som arbetar med de sista faserna av första generationens GTRS. Denna rapport är alltså inte tänkt som ett underlag i arbetet med den nu (år 2010) aktuella beställningen från leverantören Rockwell Collins. Istället ska den kunna användas i arbetet inför kommande generationer av systemet.

1.2 Genomförande

Framtagande av en IT-säkerhetsarkitektur sker som en fortsättning på en redan genomförd GTRS-riskinventering och -scenariobyggande [3], i vilken en riskinventeringsmodell var en utgångspunkt. Riskinventeringen och riskinventeringsmodellen, som redovisades i [3], relaterar även till den typ av analys som FM:s gemensamma riskhante-

ringsmodell, [1], innebär.

Arbetet som redovisas i denna rapport har tagit sin utgångspunkt i FOI:s rapport *IT-säkerhet i GTRS: Riskinventering och scenarier* [3] där det belystes hur kedjan sårbarhet-hot-skada-risk påverkar dynamisk IT-säkerhetshantering i programvaruintensiva system i nätverk. Den riskinventering som gjordes där har använts för att välja ut ett antal relevanta problemområden som i någon mening kan anses täcka IT-säkerhetsområdet för GTRS. Problemområdena är översiktliga områden, som var för sig kan rymma flera av de problem som riskinventeringen utmynnade i. Problemområdena har sedan analyserats utifrån de säkerhetsfunktioner som beskrivs i Försvarsmaktens (FM) *Krav på säkerhetsfunktioner* (KSF) [4]. På detta sätt har problemområdena belysts och analyserats ur ett flertal vinklar i syfte att se vilka säkerhetsfunktioner som behövs för varje problemområde. Därefter har analysresultaten samlats ihop för varje säkerhetsfunktion och brutits ner i mer detaljerade säkerhetsmekanismer, vilket ska betraktas som det underlag till IT-säkerhetsarkitektur som rapporten levererar.

Riskbedömningen av GTRS som redovisas i [3] utgör ett underlag för att ta fram en IT-säkerhetsarkitektur. Denna riskbedömning är övergripande och indikerar övergripande och verksamhetsrelaterade IT-säkerhetsbehov respektive till behoven relaterade övergripande IT-säkerhetskrav. En noggrannare behovs- och kravanalys hade krävt en mer omfattande dialog med FM-aktörer, till exempel i form av enkätstudier eller intervjuserier.

Med utgångspunkt i riskbedömningen, respektive indikerade behov och krav identifierar och beskriver vi i denna rapport säkerhetsfunktioner och underliggande säkerhetsmekanismer. I ytterligare steg kan *interaktionen* mellan säkerhetsfunktionerna och -mekanismerna samt övriga systemkomponenter närmare analyseras och specificeras. På detta sätt kan en IT-säkerhetsarkitektur systematiskt och stegvis byggas ihop, men denna studie begränsar sig till att identifiera och beskriva säkerhetsfunktioner och säkerhetsmekanismer.

Våra frågeställningar rör i princip GTRS som system, oberoende av vilken apparat som används. Dock är diskussionen med naturlighet präglad av Ra7201 eftersom det i dagsläget är den enda apparat som finns för GTRS. Vi kommer genomgående att använda begreppet GTRS även om det rör egenskaper som är knutna till apparaten. Detta gör vi för att slippa gränsdragningsfrågor angående vad som är systemegenskaper och vad som är apparategenskaper. Just den gränsdragningsfrågan är heller inte så betydelsefull så länge det bara finns en sorts apparat för GTRS.

1.3 Rapportstruktur

Rapportens struktur är följande: Kapitel 1 och 2 presenterar ramarna för rapporten och ger viss bakgrundskunskap som behövs för resten av rapporten. Kapitel 3 beskriver ett antal övergripande problemområden för GTRS och identifierar vilka säkerhetsfunktioner som behövs i de olika problemområdena. Kapitel 4 går igenom säkerhetsfunktionerna som identifierades i kapitel 3 och bryter ner dem i mer detaljerade säkerhetsmekanismer och beskriver var i systemet de passar in. I kapitel 5 ges exempel på riskfaktorer och hantering av dessa med hjälp av en eller flera säkerhetsfunktioner.

Under arbetet har några relaterade, men inte renodlade arkitekturfrågor identifierats. Då dessa indikerar viktiga utvecklingsfrågor eller -områden diskuteras dessa i korthet i kapitel 6. En avslutande och sammanfattande diskussion av arbetets bidrag till utvecklingen av en IT-säkerhetsarkitektur för GTRS redovisas i kapitel 7.

2 Bakgrund

I detta kapitel beskrivs inledningsvis kortfattat KSF [4] och terminologin rörande IT-säkerhetsarkitekturer. Därefter relateras arkitekturresonemangen i kapitel 1 till riskinventeringsmodellen för GTRS [3].

2.1 FM:s krav på säkerhetsfunktioner

FM har tagit fram sju generella och grundläggande säkerhetsfunktioner. Till dessa har det också tagits fram en uppsättning krav och tillsammans utgör dessa Krav på säkerhetsfunktioner (KSF) [4]. Dessa krav beskriver hur säkerhetsfunktionerna avses fungera, men endast på en relativt övergripande nivå. Kraven tar sin utgångspunkt i militär operativ verksamhet och de varierande grader av sekretessbehov som detta medför i olika situationer. I och med att KSF är framtagen av FM finns det fördelar med att använda dessa krav i GTRS-sammanhang.

Följande säkerhetsfunktioner beskrivs i KSF och återges här med aningen sammanfattande definitioner:

- Behörighetskontroll

  Denna säkerhetsfunktion ska, på alla normala vägar in i och ut ur systemet, ge rätt nivå av tillgång till varje användare.

- Säkerhetsloggning

  Denna säkerhetsfunktion ska registrera samtliga säkerhetsrelevanta förändringar och åtgärder i en säkerhetslogg.

- Intrångsskydd

  Denna säkerhetsfunktion ska skydda mot intrång genom, bland annat, nätburna angrepp, anslutande system och fjärradministration.

- Intrångsdetektering

  Denna säkerhetsfunktion ska identifiera pågående samt genomförda försök att kringgå, bland annat, säkerhetsfunktionen intrångsskydd.

- Skydd mot skadlig kod

  Denna säkerhetsfunktion ska skydda systemet mot skadlig kod och hindra denna från att köras och spridas till andra delar av systemet.

- Skydd mot RÖS

  Denna säkerhetsfunktion ska skydda systemet mot passiv avlyssning genom att eliminera röjande signaler.

- Skydd mot obehörig avlyssning

  Denna säkerhetsfunktion ska skydda mot obehörig avlyssning genom exempelvis kryptering.

Ett representativt exempel på hur kraven kan vara utformade framgår i Appendix 7.3. Där redovisas de krav som ställs på säkerhetsfunktionen *Behörighetskontroll* för IT-system inom FM som inte är avsedda för behandling av hemliga uppgifter. Vad som kan noteras är att dessa krav är fokuserade på *vad* som ska uppfyllas, men inte på *hur* detta på mer detaljerad teknisk nivå ska realiseras, det vill säga säkerhetsmekanismerna. Det kan vidare noteras att man har som mål att i framtiden även beskriva en standardiserad uppsättning säkerhetsmekanismer.

Säkerhetsfunktionerna *Skydd mot RÖS* samt *Skydd mot obehörig avlyssning* kommer inte att diskuteras närmare då de bedöms ligga utanför denna rapports ramar. Dessa säkerhetsfunktioner berör passiv och aktiv avlyssning och har begränsad IT-säkerhetskoppling, utan ligger närmare frågor kring fysisk säkerhet. Koppling finns även till kryptografi, vilket dock är ett ämne som endast kommer att kort beröras i rapporten.

2.1.1 Terminologi

Begreppet *arkitektur* används flitigt inom IT-området i samband med utveckling och beskrivningar av system. Den exakta betydelsen varierar, men gemensamt är att det brukar handla om en inte alltför detaljerad beskrivning av omfattande tekniska system.

Det mer specifika begreppet *säkerhetsarkitektur* definieras enligt SIS [5] på följande sätt:

> övergripande teknisk beskrivning av i ett system ingående säkerhetstjänster inklusive samverkan och gränssnitt mellan olika komponenter.

Även om inte begreppet säkerhetstjänst definieras bedömer vi att ovanstående stämmer väl överens med användningen av säkerhetsarkitektur i allmänna IT-sammanhang. I den här rapporten har vi ytterligare specificerat begreppet till *IT-säkerhetsarkitektur*, vilket är samma sak som SIS definition av säkerhetsarkitektur, men med förtydligandet att endast den IT-relaterade arkitekturen är i fokus.

Det kan här observeras att terminologin rörande relaterade begrepp varierar något inom litteraturen. Speciellt gäller det begreppen säkerhetsfunktion, säkerhetsmekanism och säkerhetstjänst. SIS-terminologin använder, enligt vår bedömning, följande begreppshierarki:

säkerhetsarkitektur – säkerhetstjänst – säkerhetsfunktion

KSF-terminologin [4] använder däremot, enligt vår bedömning, följande begreppshierarki:

säkerhetsarkitektur – säkerhetsfunktion – säkerhetsmekanismer

KSF använder begreppet säkerhetsfunktion på ett sätt som placerar det på motsvarande abstraktionsnivå som det av SIS använda begreppet säkerhetstjänst. SIS använder begreppet säkerhetsfunktion på ett sätt som placerar det på liknande, någorlunda tekniskt konkreta nivå som KSF:s begrepp *säkerhetsmekanismer*, ett begrepp som SIS inte använder. I SIS definieras *säkerhetsfunktion* som

> specifik teknisk egenskap hos en systemkomponent som svarar för viss del av säkerheten.

Denna rapport använder huvudsakligen KSF:s definitioner av säkerhetsfunktion och säkerhetsmekanism. Med säkerhetsmekanismer avses då mer detaljerade lösningsförfaranden, implementationer, protokoll och liknande som behövs för att realisera respektive säkerhetsfunktion.

Även andra komponenter i systemet kan ha en inverkan på IT-säkerheten utan att explicit ingå i en säkerhetsfunktion. De kan därmed vara rimliga att relatera till en IT-säkerhetsarkitektur. Med beaktande av detta och av valet att använda KSF-terminologin, anpassar vi SIS definition av säkerhetsarkitektur som följer:

> *Säkerhetsarkitektur*: Övergripande teknisk beskrivning av i ett system ingående säkerhetsfunktioner inklusive samverkan och gränssnitt mellan olika säkerhetsmekanismer. Samverkan och gränssnitt med övriga relevanta komponenter kan också beskrivas där inverkan på IT-säkerheten kan observeras.

2.2 Riskinventeringsmodell för GTRS

I [3] introducerades en riskinventeringsmodell som användes för att analysera händelsekedjor av aktörer, aktiviteter, sårbarheter och skador samt för analys av scenarier, se Bild 1 nedan.

I riskinventeringsmodellen ingår:

- *aktörer*: outsiders, insiders, legitima aktörer
- *aktiviteter*: initiala attacker, manipulation, skadliga aktiviteter
- *sårbarheter*: IT-sårbarheter i GTRS
- *skador*: skador i GTRS, skador i verksamhet
- *händelsekedjor*: en kombination av aktör, aktivitet, sårbarhet och skada som utgör en möjlig oönskad händelse.

Aktörer i modellen är de som initierar eller genomför hot, dvs. hotaktörer. I och med att legitima aktörer kan utsättas för manipulation är dessa i detta avgränsade sammanhang också hotaktörer.

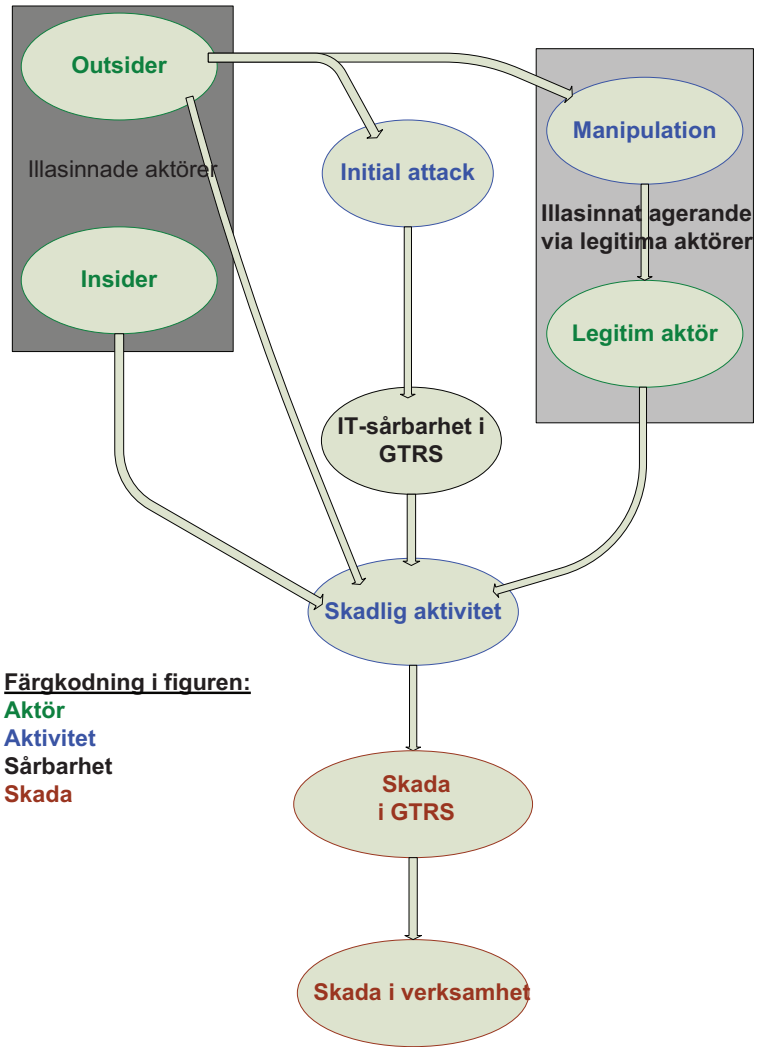

Bild 1: Riskinventeringsmodell med GTRS-relevanta aktörs- och aktivitetskategorier, sårbarheter och skador.

Händelsekedjor beskrivs i riskinventeringsmodellfiguren med hjälp av pilar som binder ihop aktörer, aktiviteter, sårbarheter och skador. Dessa åskådliggör hur hot emanerar från hotaktörer – eventuellt via en initial attack eller manipulation av legitima användare – fram till att skada uppstår i GTRS och i verksamheten vilken GTRS stödjer. Med andra ord är det i ett samspel mellan aktörer och aktiviteter som hot uppstår. Detta samspel är i sig komplext och intrikat, vilket medför att riskanalysen lätt blir detsamma. Vidare indikerar detta att vad som utgör adekvata och relevanta IT-säkerhetsfunktioner

och -mekanismer kan vara svårt att avgöra. En god IT-säkerhetsarkitektur kräver effektiva tekniska skydd. Samtidigt räcker detta inte med tanke på att de mänskliga aktörerna kan anpassa sitt beteende för att överlista vad som i utgångspunkten var effektiva tekniska skydd. En mer detaljerad beskrivning av riskinventeringsmodellen återfinns i [3].

Skador i den av en IT-incident direkt berörda verksamheten kan ofta anses vara allvarligare än den direkta konsekvensen i GTRS. Exempelvis kan sekretessbelagt innehåll i en fil tänkas bli läst av en illegitim användare. Detta behöver inte medföra någon egentlig konsekvens för GTRS, annat än att någon har forcerat skyddsmekanismerna. Däremot kan skadorna inom en berörd verksamhet vara omfattande. Det kan till exempel vara tal om att taktisk information röjs för en motståndare, vilket i en stridssituation kan innebära livsfara.

Den skadliga aktivitetens omfattning och komplexitet kan variera och detsamma gäller för de skador som kan uppstå i såväl GTRS som verksamheten. Detta implicerar behov av riskanalys och riskhantering. Riskinventeringsmodellen utgör en god utgångspunkt för riskanalys och riskhantering, genom att den möjliggör identifiering av händelsekedjor fram till skador i GTRS och berörd verksamhet. Däremot ingår inte uttrycklig bedömning av risker i modellen. Försvarsmaktens gemensamma riskhanteringsmodell [1] är för en sådan fortsättning ett adekvat och relevant ramverk att använda som redskap.

3 Problemområden belysta ur KSF-perspektiv

För att utreda vad som kan utgöra GTRS-relevanta säkerhetsfunktioner och i ett nästa steg, i Kapitel 4, vilka de underliggande säkerhetsmekanismerna till respektive säkerhetsfunktion är, har vi utgått från ett antal GTRS-relevanta problemområden och granskat hur dessa kan hanteras med hjälp av KSF:s säkerhetsfunktioner och eventuella kompletteringar till dessa. Dessa GTRS-relevanta problemområden baseras på den riskinventering och scenariokonstruktion som gjordes i [3].

Säkerhetsmekanismerna är mer detaljerade tekniska lösningar än vad säkerhetsfunktioner normalt specificerar. De i denna rapport diskuterade säkerhetsmekanismerna avses utgöra större byggstenar i en arkitektur, dock utan att täcka alla nivåer och implementationsdetaljer. Vår bedömning är att detta kan utgöra ett enkelt, men konstruktivt bidrag till en IT-säkerhetsarkitektur. Målet är att en noggrant vald uppsättning säkerhetsmekanismer ska underlätta hantering av olika riskfaktorer och därigenom uppfylla FM:s säkerhetsbehov och -krav associerade till dessa faktorer.

Riskfaktorer är – beskrivet med terminologin från riskinventeringsmodellen – aktörer, aktiviteter, sårbarheter, hot och skador i systemet GTRS och även i verksamheten i vilken GTRS används. [3] påpekar att det är i samspelet mellan aktörer och aktiviteter som hot uppstår, och resonerar i termer av händelsekedjor – i vilka sårbarheter, hot och skador också ingår – för att indikera sådana samspel.

De riskfaktorer som identifierades i [3] låter sig relativt enkelt grupperas till ett mindre, och därmed lättare hanterbart, antal problemområden. Enskilda problemområden kategoriserar händelsekedjor som ger eller beskriver olika typer av problem med

GTRS. Det har eftersträvats att de valda problemområdena på ett adekvat sätt ska beskriva den variation av informationssäkerhetsrisker som GTRS kan medföra. En begränsning som dock bör omnämnas är att vilka olika säkerhetsbehov som mer explicit finns inom FM och till exempel bland tänkbara koalitionspartners till FM, inte är kartlagt. Detta kommer att påverka såväl prioritering av vilka problemområden som bör beaktas och vilka säkerhetsfunktioner och -mekanismer som behövs.

Säkerhetsfunktioner torde vara närmare kopplade till samspelet i händelsekedjor än till enskilda aktörer, aktiviteter etc. Vår bedömning är därmed att valet av relevanta säkerhetsfunktioner underlättas av en analys relaterad till problemområden beskrivna med utgångspunkt i händelsekedjor.

Utifrån dessa avvägningar och ställningstaganden har följande problemområden valts ut:

- obehörig åtkomst,

  vilket innebär att åtkomst till system, resurs eller objekt sker i strid med gällande säkerhetspolicy

- uppgraderingsproblematik,

  all mjuk- och hårdvara är behäftad med fel och detta medför olika varianter av problem rörande uppgradering

- mjukvarubrister,

  all mjukvara är behäftad med fel som medför IT-sårbarheter och möjlig skadlig aktivitet

- hårdvarubrister,

  all hårdvara är behäftad med fel som medför sårbarheter och möjlig skadlig aktivitet

- handhavandebrister,

  vilket innebär att den mänskliga faktorn respektive brister i användargränssnitt medför att handhavande av GTRS fallerar

- tillgänglighetsangrepp,

  vilket innebär att tillgängligheten till GTRS minskar på grund av angrepp som reducerar systemets prestanda

- nätburna angrepp,

  vilket innebär att GTRS nätanslutningar öppnar upp för olika varianter av angrepp

- protokollangrepp,

  vilket innebär att defekter i en specifikation av ett protokoll i GTRS kan utnyttjas för angrepp på GTRS.

Varje problemområde granskas och lämpliga säkerhetsfunktioner beskrivs per problemområde. KSF har utgjort den uppsättning säkerhetsfunktioner ur vilken ett relevant första urval säkerhetsfunktioner för respektive problemområde har gjorts. Därefter har eventuella nödvändiga kompletterande säkerhetsfunktioner också identifierats och beskrivits.

Som omnämnt i avsnittet 2.1 har säkerhetsfunktionerna *skydd mot obehörig avlyssning* och *skydd mot RÖS* begränsad IT-säkerhetskoppling. Därför har de inte varit aktuella som säkerhetsfunktioner för den IT-säkerhetsarkitektur som diskuteras här.

3.1 Obehörig åtkomst

Obehörig åtkomst innebär, enligt [5], att åtkomst till system, resurs eller objekt sker i strid med gällande säkerhetspolicy. Detta kan ske genom att komma åt användaridentitet, inloggningsuppgifter eller på annat sätt kringgå systemets säkerhetsfunktioner. Effekter av obehörig åtkomst kan, i de generella termer SIS resonerar, vara att information avslöjas, förändras eller förstörs eller att otillåtna bearbetningar utförs.

För att skydda systemets konfidentialitet och integritet finns ett behov av att stänga ute obehöriga och inte ge någon mer tillgång än de har rätt till. För att uppfylla detta behövs säkerhetsfunktionen *behörighetskontroll* på alla vägar in i och ut ur GTRS och vid alla tillfällen, inklusive vid reparation och omstart. Vad gäller vägar ut måste exempelvis dolda kanaler regleras. Behörighetskontroll måste användas för användare, anslutande system och, i form av bland annat signering, till viss del för kod som förs in. Eventuella sidokanaler och förbikopplingsfunktioner, som ibland behövs i IT-system, måste hanteras på ett säkerhetsmedvetet sätt då de kringgår den normala säkerhetsfunktionaliteten.

Riskinventeringsmodellen indikerar att obehörig åtkomst kan initieras av insiders eller outsiders till verksamheten i vilken GTRS används. I grova drag kan outsiders initiala attacker hanteras med hjälp av *behörighetskontroll*, *intrångsdetektering* respektive *intrångsskydd*. Likaså kan i grova drag deras manipulation av legitima aktörer till att agera som insiders hanteras med hjälp av en *säkerhetsmedveten anställningsprocess* samt olika former av *utbildning*. Legitima aktörer som utan manipulation agerar som insiders kan också hanteras med hjälp av anställningsprocesser och utbildning.

Det är önskvärt att obehörig åtkomst kan spåras och dokumenteras. Här kan *säkerhetsloggning* bidra med data för utredning och analys och för att utkräva ansvar i samband med obehörig åtkomst. GTRS kommer även kräva olika behörighetsnivåer, vilket också medför en betydande utmaning att utforma och hantera.

3.1.1 Säkerhetsfunktioner för problemområdet obehörig åtkomst

Den övergripande probleminventeringen rörande problemområdet obehörig åtkomst pekar ut följande säkerhetsfunktioner ur KSF som nödvändiga för att hantera problemområdet:

- behörighetskontroll

- säkerhetsloggning
- intrångsdetektering
- intrångsskydd.

Utöver dessa har följande ytterligare säkerhetsfunktioner identifierats som nödvändiga för att hantera problemområdet:

- säkerhetsmedveten anställningsprocess
- utbildning.

3.2 Uppgraderingsproblematik

I princip all mjuk- och hårdvara av större storlek är behäftad med fel. En del upptäcks efterhand vid drift och måste normalt åtgärdas. Detta ger upphov till olika varianter av problem rörande uppgradering. En aspekt av problemområdet av vikt att beakta är vilka specifika uppgraderingsbehov som dyker upp. Utöver detta tillkommer att beakta hur uppgraderingarna ska distribueras och genomföras. Med ett så starkt distribuerat system som GTRS utgör är detta av betydande vikt. I de fall där GTRS används inom koalitioner med sina olika aktörer med olika användningsmönster, accentueras dessa behov än mera. En väl uttänkt *uppgraderingspolicy* är därmed viktig för att distribuera och genomföra uppgraderingarna.

Då GTRS består av nät med sammankopplade datorer är det, för att de inte ska förlora kommunikation och interoperabilitet, viktigt att alla noder uppdateras samtidigt för att ha samma version av mjukvaran. Om det inte rör sig om mycket kritiska mjukvarufel är det lämpligt att paketera ett antal uppdateringar till ett större uppgraderingspaket. Normalt är hårdvarufel svårare att åtgärda i fält än mjukvarufel, genom att det krävs att man öppnar utrustningen och byter alternativt reparerar komponenter. Att ta apparaturen till verkstad kan vara lämpligt i dessa fall, även om reservdelar kan hållas tillgänglig i fält. Vanligen är dock sådana fel mer sällsynta, men i militära sammanhang finns situationer där fysisk utrustning kan utsättas för direkt skada.

Även uppgraderad mjukvara kan vara behäftad med fel, vilket innebär att vidare uppgraderingsmöjligheter måste finnas, liksom kanske möjligheten att gå tillbaka till tidigare tillstånd och således att avinstallera uppdateringarna. Ett möjligt alternativ till så kallade kumulativa uppdateringar är att vid varje uppgraderingstillfälle återställa systemet till ett ursprungligt tillstånd och sedan göra en fullständig installation med den nya programvaran. *Säkerhetsloggning* är i sammanhanget en viktig säkerhetsfunktion.

Uppgradering kräver att policy och praxis utarbetas för att hantera olika säkerhetsrelaterade situationer. Till exempel behöver man bestämma hur man agerar i samband med att en nod med avvikande versionsnummer ansluter till nätet. När ska den uppdateras? Kan det i sämsta fall vara tal om ett angrepp? Ska varningar ges till nätet i övrigt? Policy krävs också för att klargöra vilken behörighet som krävs för uppgradering.

Behörighetskontroll tillsammans med funktioner som *testning, signering och verifiering av uppdateringar* krävs för att hindra obehöriga respektive olämpliga uppdateringar från att installeras. Detta är särskilt viktigt eftersom uppgraderingsmöjligheten

är en inbyggd funktion särskilt designad för att föra in ny kod i systemet och den kan därmed utgöra en svag punkt i skyddet mot skadlig kod. Detta innebär att *intrångsskydd* och *intrångsdetektering* och, som försvar på djupet, *skydd mot skadlig kod* är lämpliga säkerhetsfunktioner för att hantera uppgraderingsproblematiken.

Även om många delar av systemet, utöver renodlad mjukvara även exempelvis kryptoalgoritmer, vågformer och konfigurationsfiler, går att byta ut och/eller uppgradera, finns det en gräns för hur långt man kan gå innan man istället måste tillverka helt nya system eller noder. Just att föra in nya typer av noder, såsom handhållna enheter, eller totalt nya versioner är en utmaning då man måste utforma det grundläggande systemet för att klara av stora förändringar. Dessutom är system som är under ständig utveckling svåra att ackreditera.

3.2.1 Säkerhetsfunktioner för problemområdet uppgraderingsproblematik

Den övergripande probleminventeringen rörande problemområdet pekar ut följande säkerhetsfunktioner ur KSF som nödvändiga för att hantera problemområdet:

- behörighetskontroll
- säkerhetsloggning
- intrångsskydd
- intrångsdetektering
- skydd mot skadlig kod.

Utöver dessa har följande ytterligare säkerhetsfunktioner identifierats som nödvändiga för att hantera problemområdet:

- uppgraderingspolicy
- testning, signering och verifiering av uppgraderingar.

3.3 Mjukvarubrister

Övergången mellan problemområdena uppgraderingsproblematik respektive mjukvarubrister är flytande, i och med att mjukvarubrister till stor grad kan åtgärdas med hjälp av uppgraderingar. Några korta observationer rörande mjukvarubrister är relevanta att göra separat från uppgraderingsproblematiken. Problematiken med mjukvarubrister kan lindras med uppgradering, men inte totalt elimineras.

GTRS är mjukvaruintensivt, säkerhetskritiskt och av hög komplexitet. Detta ger som konsekvens en miljö där mjukvarubrister är kritiska att detektera och hantera.

Kodgranskning på olika nivå och av olika grupper eller instanser är en möjlighet, men systemets omfattning gör det svårt att överblicka och simulera den verkliga miljön

och därmed att kontrollera alla exekveringsvägar och tillstånd. Skillnader i tidsuppfattning mellan noder kan medföra olika konkurrerande tillstånd i systemet, vilket vore svårt att kontrollera och hantera.

En i utvecklingsorganisationen väl rotad *säkerhetsmedveten utvecklingsmetodik* är central för att på bäst möjliga sätt lyckas hantera mjukvarubrister, i och med att bristerna inte alltid låter sig upptäckas via enkelt definierade kontrollrutiner. Likaså behövs för kvalitetsgranskning, uppföljning och för att säkra spårbarhet säkerhetsfunktionen *testning, signering och verifiering av mjukvara.*

Ett exempel på delproblematik som dessa säkerhetsfunktioner kan behöva hantera är frågan om bakdörrar i system. Vid testning är det vanligt att man använder så kallade bakdörrar för att styra systemet på ett sätt som inte ska vara möjligt i verklig drift. Om dessa lämnas kvar utgör de allvarliga brister i systemet.

Bland KSF:s säkerhetsfunktioner bedöms *säkerhetsloggning* vara viktigt för att säkra spårbarhet rörande mjukvarubrister. *Skydd mot skadlig kod* är av vikt för att hindra mjukvarubrister från att utnyttjas för att introducera eller öppna upp för skadlig kod i GTRS. Eftersom mjukvarubrister kan utnyttjas för att forcera behörighetskontroller spelar även säkerhetsfunktionerna *behörighetskontroll* och *intrångsskydd* en roll. *Intrångsdetektering* kan på liknande sätt vara en sekundär säkerhetsfunktion.

3.3.1 Säkerhetsfunktioner för problemområdet mjukvarubrister

Den övergripande probleminventeringen rörande problemområdet pekar ut följande säkerhetsfunktioner ur KSF som nödvändiga för att hantera problemområdet:

- behörighetskontroll
- säkerhetsloggning
- intrångsskydd
- intrångsdetektering
- skydd mot skadlig kod.

Utöver dessa har följande ytterligare säkerhetsfunktioner identifierats som nödvändiga för att hantera problemområdet:

- säkerhetsmedveten utvecklingsmetodik
- testning, signering och verifiering av mjukvara.

3.4 Hårdvarubrister

Medan mjukvara är relativt väl skyddad av hårdvaruskal ligger däremot hårdvaran i sig utan eller med begränsat yttre skydd. Det innebär att det finns mindre möjligheter att ta

hand om och hindra fel i hårdvara. Därmed är det, för någon som är nära systemet, relativt enkelt att påverka det via hårdvaran, exempelvis genom att utsätta det för någon form av strålning eller rent mekanisk påverkan. Vidare kan övertagna hårdvarukomponenter tänkas användas på andra sätt än vad som från början var tänkt. Dessutom är hårdvara svårare att uppdatera eller byta ut, åtminstone med samma snabbhet som mjukvara.

Kompletterande fysiska skydd och personalens konstruktiva agerande utgör *intrångsskydd* i KSF:s mening, även om skyddsfunktionen i detta fall inte främst är IT-baserad. De fysiska skydden kan bland annat hindra att komponenter otillbörligen avlägsnas från apparaturen eller läggs till densamma. Dessutom bör skyddsfunktionen hindra attacker som baserar sig på att genom exempelvis bestrålning påverka noden.

Behörighetskontroll är viktig även för att förhindra otillbörlig påverkan av hårdvarukonfiguration. Fysisk *behörighetskontroll* är viktig för att kontrollera fysisk tillgång medan *intrångsdetektering* kan bestå av olika former av larm och plombering.

Ett annat hårdvarurelaterat problem är att loggar kan påverkas felaktigt, särskilt då krypterade data, av design, är mycket känsliga för även små ändringar vilket leder till problem med exempelvis spårbarhet. Vidare måste loggar av hårdvaruförändringar normalt skrivas och underhållas utanför systemet. Dessa observationer innebär särskilda krav på säkerhetsfunktionen *säkerhetsloggning*.

3.4.1 Säkerhetsfunktioner för problemområdet hårdvarubrister

Den övergripande probleminventeringen rörande problemområdet pekar ut följande säkerhetsfunktioner ur KSF som nödvändiga för att hantera problemområdet:

- behörighetskontroll
- säkerhetsloggning
- intrångsskydd
- intrångsdetektering.

3.5 Handhavandebrister

Brister i användargränssnitt och den mänskliga faktorn gör att handhavandebrister kan uppkomma. Som noterats i avsnittet behörighetskontroll i detta kapitel finns ett antal potentiella användare av systemet såsom militära styrkor, militära koalitionspartners, icke-statliga organisationer samt olika typer av administratörer och produktions- och verkstadsanställda. De olika användarna har olika bakgrund och kunskap och olika mål med sitt användande av systemet. Handhavandebrister som uppkommer genom fel av användare med mer omfattande befogenheter kan vara mer allvarliga. Vidare finns risken att användare väljer att använda ett reservsystem istället för GTRS om det senare upplevs som svåranvänt eller långsamt, vilket leder till andra säkerhetsproblem.

Det finns en lång rad olika typer av handhavandebrister. Exempelvis kan en användare glömma utrustning igång eller råka behandla den på ett sätt så att plombering av misstag lossnar. En felinställd nod kan leda till att hela nät slås ut och man kan råka komma åt nollställningsknappen om den är dåligt utformad. Annan närliggande utrustning kan störa radionätet eller fungera som avlyssning av talbaserad kommunikation genom att inget krypto hunnits läggas på. Användare kan vara oroliga att de ska glömma bort lösenord varför de skrivs ner och därmed lättare kan hamna i orätta händer. Den röd-svarta separationen kan också brytas genom att switchar används som exempelvis förlängningssladdar utan att hålla isär sladdar av olika ”färg”. Det faktum att systemet använder vissa standardprotokoll och standardgränssnitt gör att nät som inte borde ha kontakt med varandra lättare kan få det. Kontaktdon som inte är fysiskt kompatibla med fel utrustning kan delvis eller helt eliminera sådana problem.

För att i möjligaste mån undvika handhavandefel är *utbildning* och andra stödfunktioner av vikt. Vidare kan *säkerhetsloggning* till viss del minimera sannolikheten för att vissa typer av sådana här fel inträffar men framförallt kan loggar användas för att i efterhand se vad som gick fel. Utformningen av användargränssnitt är kritisk för att minimera risken för handhavandefel.

3.5.1 Säkerhetsfunktioner för problemområdet handhavandebrister

Primärt bör problemområdet handhavandebrister hanteras med hjälp av

- utbildning.

.

Som komplettering kan följande säkerhetsfunktion ur KSF bidra:

- säkerhetsloggning.

3.6 Tillgänglighetsproblematik

I ett så komplicerat system som GTRS finns många komponenter och delsystem som måste fungera för att systemet som helhet ska vara brukbart och därmed tillgängligt. Om någon del av GTRS exempelvis belastas av större eller andra datamängder än den är designad för kan prestanda reduceras.

Överbelastning kan ske på många olika sätt, till exempel i form av användares misstag eller som ett led i ett angrepp eller en skadlig aktivitet av en hotaktör. Exempelvis kan stora mängder trafik skickas genom att utnyttja systemets trådbundna anslutning eller brister i protokoll, exempelvis vad gäller routing. Rent fysiska attacker kan också påverka tillgängligheten.

Övertagna apparater kan användas för att störa via radio utan att behöva knäcka hoppsekvensen. En insider eller manipulerade anställda kan omkonfigurera en nod på ett sätt som leder till att hela nätet påverkas. Fienden kan provocera fram nollställning av en apparat genom exempelvis ett fysiskt angrepp och det senare kan även leda

till skada på plomberingen och därigenom ett potentiellt förbud mot användning av apparaten. Omkringliggande system eller fjärradministration kan attackeras och radiokommunikation kan störas med telekrigsinsatser. Lyckas hotaktören påverka systeminformationens riktighet kan tilliten till systemet skadas.

Av de säkerhetsfunktioner som nämns i KSF är det främst *intrångsskydd* som här kan vara av värde. En brandvägg som filtrerar olämplig trafik på både ytlig och djupare, mer applikationsnära, nivå kan bidra till att minska risken för överbelastningsangrepp. *Intrångsdetektering* och *säkerhetsloggning* kan fungera som larmsystem och som hjälp vid analys av angrepp.

3.6.1 Säkerhetsfunktioner för problemområdet tillgänglighetsproblematik

Den övergripande probleminventeringen rörande problemområdet pekar ut följande säkerhetsfunktioner ur KSF som nödvändiga för att hantera problemområdet:

- säkerhetsloggning
- intrångsskydd
- intrångsdetektering.

3.7 Nätburna angrepp

Då GTRS till stor del är IP-baserat och dessutom kan komma att ha anslutning till internet, avsiktligt eller av misstag genom handhavandefel, kan olika typer av attacker den vägen vara möjliga om exempelvis vissa allvarligare mjukvarufel föreligger. Genom systemets svarta ethernetport färdas normalt ingen okrypterad hemlig information, men lyckade angrepp kan fortfarande leda till att systemet som helhet, eller delar därav, kraschar eller ger angriparen viss eller fullständig tillgång till detsamma. Det senare kan vidare leda till exempelvis falsksignalering eller ett totalt kompromitterat system.

Vidare kan den svarta sidans router – vilken är central för anslutande nät och bland annat bär hoppsekvensdata från CSS:en (Crypto Sub System) till UT:arna (Universal Tranceiver) – angripas liksom eventuell fjärradministration. Även radiogränssnitt kan bli utsatta för angrepp exempelvis genom att en del av det utökade skyddet, frekvenshoppandet, kringgås genom att skicka data på alla frekvenser. Detta är troligtvis kostsamt för en angripare men kan leda till överbelastning eller till och med, för angriparen, möjlighet till egen kommunikation via UT:arna.

Angriparen kan använda olika tillvägagångssätt för att angripa GTRS. Manuella hackningsförsök kan genomföras med en eller flera relativt kompetenta individer som manuellt spanar, kartlägger, undersöker och slutligen angriper. Alternativt kan mer automatiserade angrepp användas. Dessa är normalt inte riktade mot ett visst system utan innebär snarare ett bakgrundsbrus av angrepp på internet. GTRS som är specialdesignat för hög säkerhet torde stå emot sådana angrepp, men den stora mängden kan trots

allt innebära vissa problem, exempelvis i form av överbelastning. Vidare kan anslutande nät vara mindre väl skyddade. GTRS bör även förhindra exekvering av olika typer av obehörig körbar kod som kan spridas med de nämnda automatiserade angreppen.

Intrångsskydd bedöms vara den viktigaste säkerhetsfunktionen bland de i KSF vad gäller nätburna angrepp. Funktionen måste även skydda mot angrepp som kommer från anslutande system eller andra behöriga vägar in, såsom fjärradministration. *Skydd mot skadlig kod* är också en viktig säkerhetsfunktion i sammanhanget liksom *behörighetskontroll* för att undvika falsksignalering och otillbörlig privilegiehöjning. *Säkerhetsloggning* och *intrångsdetektering* utgör larmfunktion och säkrar spårbarhet. Då en angripare ofta utnyttjar ett systems svagaste länk är även många av de övriga problemområdens föreslagna säkerhetsfunktioner av intresse för att hantera nätangrepp. Exempelvis är ofta handhavandebrister en anledning till att angrepp baserade på automatiserad skadlig kod kan lyckas.

3.7.1 Säkerhetsfunktioner för problemområdet nätburna angrepp

Den övergripande probleminventeringen rörande problemområdet pekar ut följande säkerhetsfunktioner ur KSF som nödvändiga för att hantera problemområdet:

- behörighetskontroll
- säkerhetsloggning
- intrångsskydd
- intrångsdetektering
- skydd mot skadlig kod.

Utöver dessa kan ett antal övriga säkerhetsfunktioner bidra i viss utsträckning.

3.8 Protokollangrepp

Protokollangrepp är ett specialfall av de nätangrepp som berördes i det förra avsnittet. Detta specialfall är angrepp som utnyttjar defekter i en specifikation av ett protokoll snarare än i en implementering av detsamma.

Det underliggande problemet ligger med andra ord på ett tidigt designstadium och en uppdatering skulle behöva vara relativt grundlig och därmed omfattande, inte minst då många GTRS-protokoll är standardiserade. Denna senare anledning gör vidare att svagheter troligen upptäcks lättare av angripare. Protokoll som har använts av många under längre tid är dock rimligen bättre studerade än andra. Att använda standardprotokoll är kostnadsbesparande, men å andra sidan har man mindre kontroll över utvecklingsprocessen. Standardprotokoll är i allmänhet inte framtagna för användande i system med så hög säkerhetsnivå som GTRS. Problemet med protokollangrepp förvärras av systemets komplexitet med gränssnitt till andra system och ingående arvssystem.

Protokollproblemen måste delvis lösas av säkerhetsmekanismer knutna till *behörighetskontroll* och *intrångsskydd*, men framförallt bör man i designfasen se till att valda protokoll studeras tillräckligt och att de från början är lämpade för säkerhetskritiskt arbete. *Säkerhetsloggning* och *intrångsdetektering* kan identifiera och larma om problem som uppstått och därmed bidra till att hindra att dessa sprider sig. *Säkerhetsmedveten utvecklingsmetodik* är också av vikt.

3.8.1 Säkerhetsfunktioner för problemområdet protokollangrepp

Den övergripande probleminventeringen rörande problemområdet pekar ut följande säkerhetsfunktioner ur KSF som nödvändiga för att hantera problemområdet:

- behörighetskontroll
- intrångsskydd
- säkerhetsloggning
- intrångsdetektering.

Utöver dessa har följande ytterligare säkerhetsfunktioner identifierats som nödvändiga för att hantera problemområdet:

- säkerhetsmedveten utvecklingsmetodik.

4 Säkerhetsfunktioner och säkerhetsmekanismer

Detta kapitel består av beskrivningar av varje säkerhetsfunktion samt till varje säkerhetsfunktion associerade relevanta säkerhetsmekanismer. Varje säkerhetsfunktion kan implementeras, eller realiseras, med hjälp av ett antal så kallade säkerhetsmekanismer. En säkerhetsmekanism kan vara specifik för en viss säkerhetsfunktion eller delas mellan flera.

Fokus i kapitel 3 var på att identifiera GTRS-relevanta problemområden ur vilka säkerhetsfunktioner kunde identifieras. I detta kapitel är fokus på att som ett nästa steg, för respektive säkerhetsfunktion, identifiera säkerhetsmekanismer som realiserar säkerhetsfunktionerna utgående från de generella krav GTRS ställer. De olika säkerhetsmekanismerna är resultat av egen reflektion och analys, men en första uppsättning identifierades i form av en genomgång av säkerhetsåtgärder beskrivna i [5] respektive [2].

En reflektion är att det inte är klart vilka detaljkrav GTRS ställer på säkerhetsfunktionerna och på deras underliggande säkerhetsmekanismer. Vi vet att GTRS förväntas användas i militära sammanhang, eventuellt där samverkan sker med civila koalitionspartners. Detta ställer krav på exempelvis behörighetskontroll, men på vilket sätt kräver närmare specifikationer och studier. Uppsättningen av säkerhetsmekanismer är därför

att se som förslag och utgångspunkt för vidare utvecklingsarbete. Det är inte orimligt att anse varje säkerhetsfunktion med tillhörande säkerhetsmekanismer vara ett område värt separat granskning och studium.

4.1 Behörighetskontroll

Enligt [5] ingår autentisering (verifiering av användares uppgivna identitet) och auktorisering (reglering av användares åtkomsträttigheter till information och resurser) i behörighetskontroll. Dessutom ingår enligt [5] även förande av logg, det vill säga registrering av användares aktiviteter i behörighetskontroll. Vi har här valt att hantera loggförning med avseende på säkerhetskritiska händelser separat under säkerhetsfunktionen säkerhetsloggning.

Säkerhetsfunktionen behörighetskontroll ska, på alla vägar in i och ut ur systemet – inklusive uppdateringsmöjligheter, fjärradministration samt fysisk tillgång – ge rätt nivå av tillgång till varje användare. Nya användare behöver registreras för första gången i systemet och existerande användare behöver förvaltas. Exempelvis är det av vikt att hindra otillbörlig privilegiehöjning då det är ett viktigt steg i många typer av IT-angrepp. Dessa observationer visar behovet av identitetsadministration, -registrering och -tilldelning.

Det bör ställas vissa krav på att säkerhetsattribut, framförallt lösenord, inte ska vara enkla att gissa eller prova sig till. För att ytterligare skydda mot att säkerhetsattribut av olika typ röjs är det lämpligt att de, i de fall det är möjligt, byts ut efter viss tid. Eftersom en användares rättigheter kan variera över tid bör systemet även vara designat för detta, vilket kan noteras inte är ett tydligt krav i KSF. Där noteras enbart att revokering av rättigheter ska vara möjlig om säkerhetsattributen hamnat i orätta händer, vilket inte täcker in möjligheten att en användares roll helt eller delvis förändrats.

Vid upprepade autentiseringsfel bör någon form av åtgärd vidtas. En användare som gör fel kan exempelvis låsas ute under viss tid. Att upptäcka en större mängd upprepade autentiseringsfel kan göras i samverkan med säkerhetsfunktionen *intrångsdetektering*.

Inloggade användare kan ibland av olika skäl vara tvungna att göra kortare pauser i sitt arbete med GTRS. För att sedan snabbt kunna återuppta arbetet är det lämpligt att det finns möjlighet att logga ut utan att behöva spara och avsluta sitt arbete. Av misstag kan användare glömma att logga ut. För att hindra obehörig åtkomst kan det därför vara bra att systemet automatiskt frågar efter inloggningsuppgifter efter exempelvis en viss tids inaktivitet.

Ingen användare ska ha fullständiga rättigheter i systemet, exempelvis för att ingen ska ha tillgång till både säkerhetslogg (se nedan) och ha övrig åtkomst i systemet. Det kan vidare exempelvis vara aktuellt att implementera åtkomstkontroll baserad på roller användare har eller baserad på åtkomstregler för olika resurser och användare.

Utifrån dessa resonemang bedöms följande säkerhetsmekanismer vara relevanta för GTRS:

- autentiseringsrelaterade säkerhetsmekanismer

  identitetsadministration, -registrering och -tilldelning

användare autentiserar sig mot apparatur och tvärtom

ny inloggning vid privilegiehöjning

tidsbegränsat användande (regelbunden återautentisering)

starka säkerhetsattribut

byte av säkerhetsattribut regelbundet

kontolåsning vid upprepad felautentisering

låsning av konto efter inaktiv period

möjlighet för tillfällig utloggning

- auktoriseringsrelaterade säkerhetsmekanismer

  ingen tillåts ha fullständig tillgång

  lista på åtkomsträttigheter per användare, roller och/eller resurser

  ändring av åtkomsträttighetslista inklusive tillägg av användare och revokering.

4.2 Säkerhetsloggning

En säkerhetslogg definieras enligt [5] som en logg över säkerhetskritiska händelser och den kan exempelvis innehålla information om förändringar i behörighetsinformation och förändringar i datorsystemets konfiguration. Likaså påpekar [5] vikten av att säkerhetsloggen skyddas mot obehörig påverkan.

Denna säkerhetsfunktion bör registrera samtliga säkerhetsrelevanta förändringar och åtgärder, såsom autentiseringshändelser, eventuell användning av bypassfunktionalitet och åtkomst till sekretessbelagd information. I allmänhet kräver säkerhetsloggning att viktiga händelser i övriga säkerhetsfunktioner registreras i säkerhetsloggen.

En viktig egenskap säkerhetsloggning medför är oavvislighet (eng. non-repudiation), vilket [5] beskriver som att en handling inte i efterhand ska kunna förnekas av utföraren. Vad som utfördes, vem som utförde det, när det utfördes och hur det utfördes, bör registreras. För spårbarhet är det därför viktigt att en systemgemensam klocka finns.

Rutiner för hantering av säkerhetsloggen bör finnas för att förhindra att loggen blir full och förhindrar ytterligare registreringar. Detta kan exempelvis hanteras genom att kopior av äldre loggar lagras under en viss tid som bedöms som adekvat. Eventuellt kan viss information gallras ur dessa äldre loggar för att minska utnyttjat utrymme. Möjligheter att genom mjukvara ändra eller förstöra loggen bör förhindras. Kryptering av loggar bör övervägas som skydd mot obehörig åtkomst.

Loggarna bör före kryptering kunna användas av automatiska verktyg för larm, övervakning, spårbarhet, felsökning och återställning och då i samband med säkerhetsfunktionen för intrångsdetektering. Exempelvis kan loggarna användas för att identifiera lokala inloggningar av användare som man vet inte har fysisk tillgång till systemet för tillfället.

Utifrån dessa resonemang bedöms följande säkerhetsmekanismer vara relevanta för GTRS:

- loggningsförfarande – registrera data från övriga säkerhetsfunktioner i logg
- logganalys med hjälp andra verktyg
- lagring av loggar
- kryptering av loggar
- larm i samverkan med intrångsdetektering
- tillförlitlig systemgemensam klocka.

4.3 Intrångsskydd

Denna säkerhetsfunktion ska skydda mot intrång från bland annat nätburna angrepp, anslutande system och fjärradministration samt sådant som kan utgöra dolda kanaler. Behörighetskontrollen skyddar mot mer rättframma angrepp via kanaler för normalt användande vilket kan beskrivas som systemets dörrar. Intrångsskyddet innebär däremot skydd av vad som på motsvarande sätt kan beskrivas som systemets väggar, det vill säga övriga potentiella vägar in och ut. Intrångsskyddet kan delvis bestå av en brandvägg som filtrerar olämplig trafik på både ytlig och djupare, mer applikationsnära, nivå för att skydda mot exempelvis överbelastningsattacker. Dessutom bör brandväggen vara tillståndskänslig, det vill säga komma ihåg vilken trafik som tidigare skickats i en session, utan att för den sakens skull vara allt för känslig för överbelastningsangrepp.

Intrångsskyddet bör inte enbart, som [4] ger som alternativ, förlita sig på att krypterad information utsänd från en autentiserad avsändare inte kan göra skada då apparatur och inloggningsuppgifter kan hamna i orätta händer och apparaturen då missbrukas på ett sätt som intrångsskyddet i så fall inte upptäcker. Istället bör intrångsskyddet kontrollera även andra egenskaper än sådana som kommer av korrekt autentisering, exempelvis trafikegenskaper. Även om inte intrångsskyddet kan läsa den krypterade informationen kan det till exempel reagera på att alltför stora mängder data sänds från en och samma avsändare.

Försök att ta sig genom intrångsskyddet kan istället definieras på liknande sätt som för säkerhetsfunktionen för skadlig kod, det vill säga genom svart- eller vitlistning samt eventuellt viss heuristik/avvikelseanalys. Sådana försök bör identifieras av och, utöver intrångsskyddets direkta hindrande försvar, hanteras av säkerhetsfunktionen *intrångsdetektering*, exempelvis med hjälp av larm.

Utifrån dessa resonemang bedöms följande säkerhetsmekanismer vara relevanta för GTRS:

- filtrera genom kontroll av trafikegenskaper
- filtrera genom kontroll av applikationsegenskaper
- vit-/svartlistning

- avvikelseanalys/heuristik
- skicka data till intrångsdetektering för utvärdering.

4.4 Intrångsdetektering

Denna säkerhetsfunktion ska *upptäcka och identifiera* pågående och genomförda försök att kringgå säkerhetsfunktionen intrångsskydd (så kallade *intrångsförsök*), vare sig dessa lyckats eller ej och ska utgöra ett komplement till säkerhetsfunktionen *intrångsskydd*. Dessutom kan *intrångsdetektering* reagera på andra typer av avvikelser såsom upprepade felaktiga inloggningsförsök och annat som främst skyddas av andra säkerhetsfunktioner än intrångsskydd.

Vid upptäckt av intrångsförsök och liknande, vilka ska registreras i säkerhetsloggen i lättanalyserbar form, ska larm aktiveras. Analys bör genomföras av säkerhetsloggar för att uppnå full spårbarhet och förbättrade möjligheter för återställning. Av samma skäl är det viktigt att all intrångsdetekteringsinformation loggas.

Utifrån dessa resonemang bedöms följande säkerhetsmekanismer vara relevanta för GTRS:

- inhämta data och larm från övriga säkerhetsfunktioner
- aggregera och korrelera data och loggar för analys
- larma.

4.5 Skydd mot skadlig kod

Denna säkerhetsfunktion ska hindra skadlig kod från att köras och manipulera systemet, samt spridas till andra delar av detsamma. Detta kan göras genom att ständigt analysera systemet för att hitta eventuell ny kod och därefter avgöra om denna är skadlig. Att arbeta i anslutning till de designade kanalerna in i och ut ur systemet är särskilt lämpligt. Detta ger ett försvar på djupet där skadlig kod kan upptäckas även om den lyckats ta sig förbi intrångsskydd och liknande. Även kod som införs direkt via hårdvara eller som indata till program kan till viss del hanteras av *skydd mot skadlig kod* genom exempelvis indatavalidering och rimlighetskontroller.

Säkerhetsfunktionen bör automatiskt försöka att eliminera upptäckta problem och spara undan data för att i efterhand kunna inspektera koden. Även om man inte lyckas rädda den enskilda noden från att infekteras av skadlig kod ska man genom inbyggda larm, genom intrångsdetekteringsfunktionen, bli varse problemet för att bland annat kunna se till att övriga noder inte drabbas. I GTRS ska normalt all kod som inte finns med i en uppdaterbar vitlistning klassificeras som skadlig. Digitala signaturer kan ingå som ett sätt att kontrollera att kod/programvara inte manipulerats.

För något ökad dynamik kan det i vissa fall vara lämpligt med den något mindre strikta mekanismen svartlistning eller avvikelsekontroll, vilken förbjuder sådant som avviker från det normala. En striktare variant, där man kräver att systemets kod eller

delsystemkod ser ut på ett och endast ett sätt och inte kan ändras, är också möjlig. Exempelvis kan man jämföra en kontrollsumma, som tagits fram vid tillverkning eller uppdatering, med hur systemet ser ut vid varje tidpunkt. Detta kan liknas vid en vitlistning med krav på att allt på listan ska finnas med och ger ett djup i försvaret generellt vad gäller integriteten, det vill säga systemets riktighet.

Utifrån dessa resonemang bedöms följande säkerhetsmekanismer vara relevanta för GTRS:

- stoppa skadlig kod från att få åtkomst och ändra
  - vid gränssnitt
  - genom kompletterande systemgenomsökning
- placera skadlig kod i karantän
- spara undan bevis genom exempelvis loggning
- larma genom intrångsdetekteringsfunktionen
- definiera skadlig kod definieras genom:
  - vit-/svartlistning
  - avvikelseanalys/heuristik
  - systemkontrollsumma
- indatavalidering
- digitala signaturer.

4.6 Säkerhetsfunktioner utöver KSF-säkerhetsfunktionerna

Vid analysen av problemområden i kapitel 3 identifierades en uppsättning säkerhetsfunktioner som inte ingår i KSF-strukturen. Dessa ytterligare säkerhetsfunktioner är följande:

- säkerhetsmedveten anställningsprocess, vilken identifierades i problemområdet obehörig åtkomst
- utbildning, vilken identifierades i problemområderna obehörig åtkomst respektive handhavandebrister
- uppgraderingspolicy, vilken identifierades i problemområdet uppgraderingsproblematik
- testning, signering och verifiering av uppgraderingar, vilken identifierades i problemområdet uppgraderingsproblematik
- säkerhetsmedveten utvecklingsmetodik, vilken identifierades i problemområderna mjukvarubrister respektive protokollangrepp

- testning, signering och verifiering av mjukvara, vilken identifierades i problemområdet mjukvarubrister
- fysiskt skydd, vilken identifierades i problemområdet hårdvarubrister.

Dessa säkerhetsfunktioner har inte brutits ner i uppsättningar av säkerhetsmekanismer. Vår bedömning är att processen med att bryta ner dessa säkerhetsfunktioner i säkerhetsmekanismer hade resulterat i ett nära på arbiträrt utfall, åtminstone inom existerande resursramar.

5 Relation mellan riskfaktorer och säkerhetsfunktioner

För att tydliggöra nyttan med säkerhetsfunktionerna ges här ett antal exempel på allvarligare risker, hämtade från [3], som kan minskas med hjälp av en eller flera säkerhetsfunktioner. Riskerna bedömdes i Riskinventering efter sin allvarlighetsgrad, med 1 som relativt harmlöst och 5 som mycket allvarligt.

5.1 Utvecklingsprocess

I [3], avsnitt 4.6.9 Utvecklingsprocess, med bedömd risknivå 5, och till viss del även i avsnitt 4.6.6 Mjukvaruuppdateringsmekanism, finns beskrivet vilka risker som kan kopplas till utvecklingsprocessen. På grund av sin storlek och många delsystem, inklusive arvssystem och anslutande system är GTRS ett komplext system som är svårt att utveckla. Vidare är det mödosamt att verifiera implementeringen, inte minst avseende säkerhet och att erhålla hög assurans. Man måste bland annat undvika att testfunktionalitet såsom bakdörrar lämnas kvar från utvecklingsprocessen. Sammantaget krävs säkerhetsfunktionen *säkerhetsmedveten utvecklingsmetodik*. Denna säkerhetsfunktion utgör på sätt och vis ett aggregat av ett flertal olika säkerhetsfunktioner och hela utvecklingsprocessen kan ses som ett eget system som måste skyddas med samma noggrannhet som den färdiga produkten. Exempelvis krävs säkerhetsfunktionerna *behörighetskontroll* och *intrångsskydd* för att säkerställa att endast behöriga får tillgång till systemet och *skydd mot skadlig kod* kan vara lämpligt för att skydda mot att skadlig kod förs in i systemet vid dess tillverkning. Samma säkerhetsfunktion kan till viss del avhjälpa problem rörande kvarlämnade bakdörrar. Slutligen krävs även exempelvis områdesbevakning och insiderskydd, för att veta vilka nya hot som finns och därmed vilka ytterligare krav som finns vad gäller säkerhetsfunktioner.

5.2 Utnyttjande av övertagen apparat

I [3] avsnitt 4.7.1 Utnyttjande av övertagen fungerande apparat och i avsnitt 4.7.8 Nyttjande av GTRS för falsksignalering beskrivs risken att en GTRS-apparat övertas,

främst genom fysiska medel, och utnyttjas för exempelvis falsksignalering, störning, avlyssning samt undersökning av själva apparaten. Risknivån bedömdes till fem på grund av förlustens allvarlighetsgrad trots att problemet som sådant troligen är övergående då det förmodligen snart upptäcks, varpå bland annat kryptonycklar byts. Reservsystem och säkerhetskopiering av information är viktig för att i någon mån kunna fortsätta kommunicera säkert utan GTRS och inte förlora viktig information.

Då den övertagna apparaten känner hoppsekvensen är någon form av störning via radio inte svår att genomföra, se även avsnitt 4.6.2 Öppen svart ethernetport i [3]. Vidare kan felaktig routinginformation skickas ut i nätet, vilket kan göra särskild skada om övertagandet inte upptäckts.

För att skydda mot att hårdvara avlägsnas för analys bör denna i möjligaste mån sättas fast på ett sådant sätt att den inte går att få loss utan att förstöras. Då tillbörliga reparationer kan komma att behövas bör dock en sådan lösning användas relativt sparsamt. Viktigt är att det åtminstone går att detektera om en apparat öppnats, och att det inte går att återförsluta den och återställa den plombering som bör finnas. Detta för att man ska kunna veta hur en apparat hanterats om den återtas. Här är även säkerhetsfunktionerna *intrångsdetektering* och *säkerhetsloggning* av vikt. Mjukvara kan, genom exempelvis kodfördunkling, vilket hör till säkerhetsfunktionen *säkerhetsmedveten utvecklingsmetodik*, delvis skyddas från att studeras.

Passiv avlyssning av nätet, vilket kan vara ett legitimt behov i situationer då lokal radiotystnad är av värde, är svårt att skydda sig mot om motståndaren utan upptäckt övertagit en fungerande apparat. Även vid trafik från noden ifråga är det inte säkert att det går att upptäcka att den övertagits då det är svårt att skilja på en legitim och illegitim användare, speciellt då talbaserad kommunikation normalt inte används. Denna skulle annars kunna utgöra en form av biometrisk autentiseringsmetod under förutsättning att operatörerna känner varandras röster. Att låta användare autentisera sig på nytt med jämna mellanrum är ett sätt att förkorta tiden en motståndare kan dra nytta av att ha övertagit en inloggad nod. Säkerhetsfunktionen *behörighetskontroll* är av detta skäl mycket viktig i sammanhanget. Noggrannare riskbedömningar är nödvändiga för att avgöra hur säkerhetsfunktionen behörighetskontroll ska utformas. Den behöver även kunna konfigureras på ett säkert sätt för att medge anpassning till varierande hotnivåer och säkerhetsbehov.

5.3 Öppen svart ethernetport

I avsnitt 4.6.2 Öppen svart ethernetport i [3] behandlas den enskilda GTRS-nodens svarta ethernetport. Denna port kan användas till mycket, bland annat anslutning till andra nät, oavsett om de är GTRS-nät eller ej, samt för att använda noden som radiogränssnitt för ett annars trådbundet nätverk. På grund av detta, samt det faktum att nodintern information möjligen kan läcka ut den vägen, bedömdes i [3] risknivån till 4. Eftersom det trots allt rör sig om den svarta sidan bör i alla fall den allra mest känsliga informationen vara skyddad genom kryptering, varför aspekter kring konfidentialitet inte vidare behöver skyddas av säkerhetsfunktioner som beskrivits i denna rapport. Däremot måste porten ifråga fredas från illegitim trafik, dels för att inte felaktig in-

formation ska nå in i systemet och därmed skada integriteten, dels för att inte överbelastning och andra typer av tillgänglighetsangrepp ska vara möjlig denna väg. För att hantera dessa risker måste säkerhetsfunktionerna *behörighetskontroll* och *intrångsskydd* användas, i kombination för att förhindra obehörig åtkomst, men framförallt för att skydda mot just tillgänglighetsangrepp. Intrångssskyddet bör, på både ytligare och djupare nivå, kunna reagera på avvikelser från normala beteenden. Då det inte alltid är lätt att förutsäga hur överbelastning och liknande kommer att utföras, räcker det troligen inte med svart- eller vitlistning vad gäller vilken trafik och data som är tillåten. Istället måste säkerhetsfunktionen ifråga, i samverkan med *intrångsdetektering*, kunna reagera på avvikelser från normala beteenden och skeenden för att upptäcka och hantera detta.

5.4 Användbarhet

I [3] diskuterades användbarhet bland annat i avsnitt 6.1 Koppling mellan säkerhet och användbarhet samt i avsnitt 4.7.6 Byte till dåligt krypto, med bedömd risknivå 3 och vidare i avsnitt 4.7.5 Nollställning styrd av användare samt i avsnitt 4.7.12 Felkopplade sladdar. Oavsett hur bra säkerhet man har i övrigt så måste systemet användas på rätt sätt för att vara säkert och överhuvudtaget användbart. Svåra och långsamma samt allmänt dåliga användargränssnitt i GTRS, och i perifera system, kan leda till att andra system används istället, trots att de troligen är mindre säkra i övrigt. För att erhålla lättanvända system bör man efterlikna sådana system som idag är vanliga och användarna är väl förtrogna med. Detta är vidare viktigt för att undvika handhavandefel såsom oavsiktlig nollställning eller felkoppling av sladdar, även om det senare även till viss del kan hindras med hjälp av andra åtgärder så som fysiskt inkompatibla kontaktdon system emellan.

Säkerhetsfunktionen *utbildning* är central för att uppnå hög grad av användbarhet. Användare behöver utbildning rörande hantering av GTRS, systemutvecklare rörande hur hantering av GTRS kan underlättas för användare. Även för de flesta andra säkerhetsfunktioner existerar användbarhetsaspekter, i och med att användarna är beroende av dessa i sin interaktion med GTRS. Upplever man interaktionen som komplex och svårbegriplig, innebär detta i sig presumtiva direkta eller indirekta säkerhetsproblem.

5.5 Skadlig kod

I [3], avsnitt 4.6.7 Exekvering av körbar kod, beskrevs exekvering av potentiellt skadlig körbar kod. Problemet bedömdes till risknivå 3 på grund av att det finns så många vägar in i systemet och framförallt för att mängden kod som redan finns i systemet är så stor att den är svårgranskad vad gäller att undvika möjligheter till körning av skadlig kod. Vidare kan konsekvenserna av eventuellt intrång av denna art vara mycket allvarliga. I avsnitt 4.5.2 Virus, maskar och trojaner i [3] beskrivs GTRS-relaterade risker med olika varianter av skadlig kod. Där nämns att skadlig kod kan spridas till GTRS från alla typer av anslutande system, även om de har mindre grad av behörighet i GTRS. Sä-

kerhetsfunktionen *skydd mot skadlig kod* bör kunna hantera de flesta oriktade hot från skadlig kod, men mer riktade sådana är mer problematiska, särskilt om de utvecklats och används av en resursstark motståndare. Dessutom utgör uppdateringsmekanismen en synnerligen känslig del som kan utnyttjas för att föra in skadlig kod.

6 Särskilda frågor

I detta kapitel beskrivs några enskilda frågor som är relevanta för IT-säkerhetsarkitekturen, men som inte funnits passa in i rapportstrukturen ovan. Det rör sig om frågor som är relevanta, men av mer problematiserande karaktär än resten av rapporten.

6.1 Röd/svart-separering

I ett traditionellt kryptosystem definierar kryptokanalen gränsen mellan röd och svart sida. Vi kallar denna gräns för kryptogränsen. Genom att ingenting från röd sida tillåts slippa ut genom kryptokanalen utan att samtidigt krypteras, och inga andra kanaler mellan röd och svart sida får finnas, erhålls en lösning som har tillräckligt låg komplexitet för att kunna verifieras som säker. Även GTRS är uppbyggd på detta sätt och kryptogränsen finns någonstans i CSS:en.

Förutom den röda och svarta sidan finns i de flesta kryptosystem en administrativ kanal för att föra in nycklar och konfigurationsinställningar i systemet. Denna kanal behöver inte vara en kommunikationskanal i traditionell mening. Även ett vred för att välja en inställning eller ett hålkortsfack för en nyckel ingår i vad vi menar med administrativ kanal. Gemensamt för denna typ av kanal är att den är skyddad från motståndarsidan, i allmänhet genom att den hålls under fysisk kontroll.

Vad som normalt inte påpekas är att kryptogränsen också skyddar den röda sidan från vissa typer av angrepp från den svarta sidan. Detta eftersom endast kommunikation som går att dekryptera tillåts passera. Det är alltså svårt för en extern angripare att lyckas föra in data i de delar som anses känsliga. Kryptogränsen fungerar alltså som ett sekretesskydd utåt och ett integritetsskydd inåt.

GTRS har en delvis annan arkitektur än den vi beskrivit ovan. Vissa delsystem på den svarta sidan (router och UT:ar) har behov av integritetsskydd så till vida att den mjukvara som de laddas med måste vara korrekt för att systemet som helhet ska fungera. Denna mjukvara lagras i krypterad form på CSS:ens röda sida och överförs efter dekryptering till respektive mottagare under GTRS-apparatens uppstartssekvens. Det finns alltså ett integritetsbehov även utanför kryptogränsen, något som alltså måste uppnås med andra verktyg än kryptografiska. En stor del av de IT-säkerhetsbehov som vi identifierat i projektets två rapporter har sin grund i separationen mellan sekretessgräns och integritetsgräns, där sekretessgränsen har fått bestämma placeringen av kryptogränsen.

Det är värt att notera att UT:arna erbjuder ett grundläggande integritetsskydd tack vare frekvenshopp och länkkrypto. Däremot finns inte nödvändigtvis något motsvarande skydd på den svarta ethernetanslutningen som eventuellt kommer att finnas. Ett

kryptografiskt skydd även på den ingången skulle påtagligt förbättra situationen för integritetsskyddet i den svarta delen av GTRS. Det är väsentligt att en sådan lösning implementeras så att det är det första skydd inkommande kommunikation behöver passera. Ingen underliggande behandling i mjukvara på insidan av GTRS kan förekomma, eftersom det skulle öppna upp för attacker som sker innan det kryptografiska skyddet kommer till användning, se avsnitt 4.5.4 Manuella nätbaserade attacker i [3]. Överhuvudtaget behövs ett sådant skydd på varje anslutande nät som man inte har kontroll över på något annat sätt.

6.2 Nätaspekter

Diskussion i rapporten har mestadels fokuserat på säkerhet rörande enskilda noder. I detta avsnitt analyseras frågor som är mer relaterade till hela nät, exempelvis routingprotokollsfrågor och frågor avseende tillgänglighetsangrepp på nätnivå, samt perifera systemaspekter av teknisk karaktär såsom anslutande system.

Det finns flera skillnader mellan en dator som används isolerad och en som är en del av ett nät av likadana datorer. Först och främst måste man hålla reda på vilka som är en del av nätet och hantera att noder ansluter, lämnar och återansluter till detsamma. Dessutom vill man normalt skydda sig mot att vem som helst kan gå med i nätet eller att utge sig för att vara en del av det. Säkerhetsproblem kan därmed uppstå om någon utomstående kan utge sig för att vara en nod som redan är, eller varit med i nätet, för att antingen ta del av och medverka i nätet, så kallad ”spoofing”, eller för att helt enkelt blockera den nod man utger sig för att vara från att återfå kontakt med nätet, så kallad ”squatting”. I GTRS kan radioskugga ge upphov till liknande problem även om spoofing bör kunna minimeras med hjälp av ömsesidig autentisering som bygger på i förhand framtagna säkerhetsattribut. Att någon försöker hindra en nod från att få kontakt med nätet, genom exempelvis jamming, kan dock vara ett möjligt tillgänglighetsproblem. Vidare finns möjlighet att övertagna noder, vilka är autentiserade, påverkar routingen inom nätet, se avsnitt 5.2.

Användbar bandbredd måste delas upp på lämpligt sätt. För att kommunicera med övriga noder på ett säkert sätt måste man se till att inte alla pratar ”i mun på varandra” och att rätt destinationsnod nås. Vidare är det viktigt att all kommunikation är skyddad från utomstående, vilket löses med kryptering. Man måste också hantera situationer då någon försöker ändra på eller förstöra information som skickas och därmed hindra lyckad kommunikation. Tillgänglighetsangrepp noterades även i förra stycket medan angrepp mot riktigheten i informationen som överförs borde upptäckas då ändrad kryptotext torde leda till oläslig eller obegriplig klartext.

Ytterligare en fråga är nättopologin, det vill säga hur nätet är strukturerat logiskt och fysiskt. Kanske är det lämpligt med någon form av hierarki eller så föredrar man exempelvis en helt decentraliserad variant. Ibland vill man dela olika typer av resurser inom nätet, såsom någon typ av hårdvara. Var denna då förläggs är av intresse och om resursen kräver extern konnektivitet kan det ibland vara intressant att placera den i en avskiljd del av nätet, en så kallad demilitariserad zon. Ett exempel på en sådan resurs kan vara en säkerhetsmekanism av typen ”honeypot”. På så sätt kan man utnyttja de

eventuellt starkare resurser som ett nät kan erbjuda och man slipper utföra samma säkerhetsprocedurer på flera ställen. Å andra sidan kan de enskilda noderna skilja sig något från varandra och därmed kräva olika säkerhetslösningar.

GTRS-noderna är ibland anslutna till mer än bara varandra. Anslutande nät kan vara koalitionspartners motsvarighet till GTRS och små skillnader i protokoll näten emellan kan leda till oförutsägbara resultat. Även arvssystem av olika typ och modernare system kan vara svåra att föra ihop med GTRS och därmed kräva mycket arbete och utveckling av passande gränssnitt. Vad gäller äldre system kan gamla sårbarheter plötsligt finnas i GTRS absoluta närhet och i allmänhet bör man vara försiktig med att låta andra system få tillgång till det egna systemet.

6.3 Handhållna enheter

I nästa utrustningsgeneration kan man komma att vilja introducera nya typer av GTRS-noder, exempelvis handhållna enheter, vilket ställer nya krav på säkerheten. Det är viktigt att redan från början designa GTRS för att kunna genomföra sådana förändringar i näten utan att behöva bygga om hela systemet. Det är därför av intresse att notera hur kravbilden förändras av mindre och portabla noder.

Det finns ett antal skillnader mellan handhållna enheter och större sådana. Mindre storlek innebär mer begränsade möjligheter avseende gränssnitt.

Processorkraft kan bli lidande och handhållna enheter är därmed mer sårbara för tillgänglighetsangrepp genom överbelastning. Fjärradministration och kryptosystem kan också bli lidande av begränsad processorkraft. Annan hårdvara riskerar att medföra skillnader i mjukvara, vilket leder till fler typer av system att uppdatera och underhålla. Begränsad lagringskapacitet är en annan faktor, liksom andra strömförsörjningsmöjligheter samt mindre uteffekt och antennstorlek.

Eftersom handhållna enheter är så enkla att ta med sig kan de lättare hamna i okontrollerade miljöer där exempelvis förhållanden rörande avlyssningsskydd kan vara annorlunda. Portabla enheterna kan även lättare hamna i orätta händer.

7 Diskussion

SIS definition av begreppet säkerhetsarkitektur är okomplicerat. Med den definitionen som utgångspunkt innebär en IT-säkerhetsarkitektur att ge en övergripande teknisk beskrivning av ett system, dess ingående säkerhetsfunktioner och säkerhetsmekanismer och deras samverkan. Att realisera en IT-säkerhetsarkitektur är däremot inte okomplicerat. Det är nödvändigt att klargöra vad som avses med begrepp som exempelvis säkerhetsfunktion och säkerhetsmekanism och därefter komma fram till vad begreppen innebär i ett aktuellt fall med ett visst system och en viss kontext. I fallet GTRS är en första generation av systemet under hård- och mjukvarumässig utveckling, samtidigt som resonemang pågår kring hur arkitekturen för nästa generation bör byggas upp. Utöver detta är kontexten för GTRS inte helt självklar. Att systemet ska användas i militära kommunikationssammanhang, möjligen i samverkan med civila aktörer, är klart.

Konkreta säkerhetsbehov rörande GTRS för kommande GTRS-generation är däremot tillgängliga endast på en översiktlig nivå.

7.1 Försvarsmaktens Krav på säkerhetsfunktioner

Vi har i vårt arbete valt att utgå från KSF för att få en bra struktur på den resulterande IT-säkerhetsarkitekturen. Att vi valde KSF framför andra möjliga ramverk var för att i så stor utsträckning som möjligt erhålla en struktur som överensstämmer med andra liknande FM-relaterade studier. KSF innehåller generella krav avseende generella säkerhetsfunktioner för FM:s system.

Det finns alltså behov av att bryta ner KSF:s säkerhetsfunktioner i mindre delar för att nå fram till något som kan fungera som en IT-säkerhetsarkitektur. I rapporten har vi för GTRS gjort denna nedbrytning genom att peka på säkerhetsmekanismer som passar för de GTRS-specifika behov som identifierats. Vi har alltså gått vidare, förbi KSF:s säkerhetsfunktioner, till något annat som är mer detaljerat. Detta innebär inte att KSF:s uppdelning i säkerhetsfunktioner är utan värde. Säkerhetsfunktionerna är relevanta och pekar på säkerhetsfrågor som är närmast allmängiltiga och måste hanteras, men just allmängiltigheten gör att de, när de tillämpas på ett visst system, behöver tolkas i form av mer specifika tekniker, dvs. säkerhetsmekanismerna.

Olika system ser olika ut och även om samma säkerhetskrav föreligger så finns ofta stora skillnader i implementeringsmöjligheterna. Säkerhetsmekanismer implementerade i olika system skiljer sig därmed åt till åtminstone viss del. KSF har däremot en strävan mot standardisering. Även om enskild säkerhetsfunktionalitet kan lösas på ett förhållandevis standardiserat sätt med hjälp av en viss säkerhetsmekanism är det inte säkert att denna kan fungera tillfredsställande tillsammans med övriga säkerhetsmekanismer och systemet ifråga som helhet. Vidare kan det finnas säkerhetsmässiga fördelar med olika lösningar i olika system eller delsystem.

Arbete pågår inom FM med en ny version av KSF. Detta pågående arbete är en faktor som framöver kan tänkas påverka uppbyggnaden av IT-säkerhetsarkitekturen för GTRS. Vilka säkerhetsfunktioner som FM önskar ska användas kan alltså ändras i framtiden. Detta skulle, men behöver inte, påverka vilka säkerhetsmekanismer som krävs i GTRS och hur deras samverkan förväntas realiseras.

7.2 Styrkor och svagheter i analysen

Som man kan notera är utvecklingssituationen i utgångspunkten komplex och kräver flexibilitet från de involverade aktörerna. Detta implicerar vidare att det i rapporten redovisade utkastet till IT-säkerhetsarkitektur ska ses som ett underlag snarare än en färdig IT-säkerhetsarkitektur. Eftersom vi inte haft tillgång till några tydligt formulerade säkerhetsbehov eller detaljerade fallbeskrivningar har IT-säkerhetsarkitekturen blivit generell och översiktlig till sin natur. Speciellt tydligt blir detta med avseende på de framtagna säkerhetsmekanismer som rapporten redovisar. De redovisade säkerhetsmekanismerna kan återfinnas i de flesta IT-system. Detta kan ses som en svaghet

så till vida att vi inte kunnat ge någon omfattande beskrivning av säkerhetsfunktioner eller -mekanismer som är specifika för GTRS. Sett ur en annan synvinkel så har de två rapporterna lyft fram, och visat, att ur ett IT-säkerhetsperspektiv gäller att *GTRS har alla de säkerhetsegenskaper man kan förvänta sig av en nätansluten dator*, varken mer eller mindre. Detta är inte självklart utan visar bara ytterligare en gång att det finns flera perspektiv att anlägga vad gäller GTRS och att dessa kan vara sinsemellan mycket olikartade.

7.3 Specifika behov

De skillnader som finns mellan GTRS och en vanlig dator handlar huvudsakligen om att GTRS också är en kryptoprodukt och en radioprodukt. Dessa egenskaper för med sig ytterligare säkerhetsbehov, men det rör sig inte specifikt om IT-säkerhetsbehov.

Vi tror att tydliga säkerhetsbehov och fallbeskrivningar är en förutsättning för att explicit kunna lyfta fram GTRS-specifika säkerhetsegenskaper. Specifika militära säkerhetsbehov (inte bara IT-säkerhetsbehov) rörande GTRS torde rimligen utgöra underlag för att formulera specifika IT-säkerhetskrav för GTRS och dessa torde medföra behov av anpassning och vidareutveckling av den typ av generella säkerhetsmekanismer som redovisats.

Kombinationen kryptosystem/radiosystem/IT-system kan också medföra speciella säkerhetsbehov. Den i GTRS-sammanhang ofta diskuterade röd/svart-separeringen i anknytning till kryptokärnan är en aspekt på GTRS som torde medföra särskilda säkerhetsbehov och -krav. Röd/svart-separationen är direkt knuten till kryptokärnan, men separationens påverkan på säkerhetsfunktionerna och -mekanismerna i övrigt är i behov av att studeras närmare, se avsnitt 6.1 Röd/svart-separering. Valet av placering av röd/svart-gränsen påverkar t.ex. hur routing kan göras i radionätet.

Referenser

[1] *Försvarsmaktens gemensamma riskhanteringsmodell – Metodanvisning*, Försvarsmakten M7739-350012, 2009.

[2] Gollmann, D., Computer Security, 2nd ed, John Wiley & Sons, ISBN 978-0470862933, 2006.

[3] A. Hunstad, H. Karlzén och J. Löfvenberg, *IT-säkerhet i GTRS: Riskinventering och scenarier*, FOI-R--2980--SE, april 2010.

[4] Försvarsmakten (2004). Krav på säkerhetsfunktioner – Grunder. 10 750: 78976. 2004-12-20.

[5] SIS, *SIS HB 550, Utgåva 3, Terminologi för informationssäkerhet*, 2007.

Appendix A Exempel på Försvarsmaktens krav på säkerhetsfunktioner

Som ett representativt exempel på hur FM:s krav på säkerhetsfunktion (KSF) kan vara utformade listas här de krav som ställs på säkerhetsfunktionen behörighetskontroll för IT-system som inte är avsedda för behandling av hemliga uppgifter [4]:

- Säkerhetsfunktionen för behörighetskontroll ska förhindra åtkomst till IT-systemets subjekt och objekt av användare och subjekt som inte har behörighet och åtkomsträttigheter till IT-systemet.
- Säkerhetsfunktionen för behörighetskontroll ska unikt identifiera och autentisera en användare innan åtkomst av någon funktionalitet eller tilldelning av åtkomsträttigheter får ske i det IT-system som skyddas av säkerhetsfunktionen.
- Säkerhetsfunktionen för behörighetskontroll ska autentisera en användare vid:
  - inloggning,
  - upphävande att [sic] tillfälligt åtkomstskydd,
  - byte av säkerhetsattribut för autentisering
  - när tiden för tidsbegränsad användning av IT-systemets resurser har gått ut.
- Säkerhetsfunktionen för behörighetskontroll ska säkerställa en viss kvalité på det säkerhetsattribut som används för autentisering genom att kontrollera att säkerhetsattributet ges:
  - en minsta giltighetstid,
  - ett minsta antal tillåtna tecken används för skapandet av säkerhetsattributet
  - en längsta giltighetstid används för säkerhetsattributet.
- Säkerhetsfunktionen för behörighetskontroll ska säkerställa att alla användare kan göras individuellt ansvariga (dvs oavvislighet) för sina vidtagna åtgärder i IT-systemet.
- Säkerhetsfunktionen för behörighetskontroll ska använda lösenord eller motsvarande som kontrollmekanism och säkerhetsattribut för autentisering.
- Säkerhetsfunktionen för behörighetskontroll ska kunna vidta automatiska åtgärder vid autentiseringsfel. Sådana åtgärder ska omfatta nekande av åtkomst till IT-systemet samt låsning av berörd användares konto under en viss tid.
- Säkerhetsfunktionen för behörighetskontroll ska använda användares, subjekts och objekts säkerhetsattribut som kontrollmekanism vid styrning av åtkomst.
- Säkerhetsfunktionen för behörighetskontroll ska möjliggöra olika definierade roller i en hierarkisk struktur.

- Säkerhetsfunktionen för behörighetskontroll ska säkerställa låsning av sådana säkerhetsattribut som kan anses vara röjda till sådana användare eller subjekt som inte har behörighet och åtkomsträttigheter till IT-systemet. Låsningen kan ske direkt eller vid nästa inloggning.
- Det ska gå att lägga till, ta bort eller på annat sätt förändra vilka kontroller som ska utföras för att säkerställa säkerhetsattributens kvalité.
- Det ska gå att lägga till, ta bort eller på annat sätt förändra vilka åtgärder som ska vidtas vid autentiseringsfel.
- Det ska gå att lägga till, ta bort eller på annat sätt förändra vid vilka tillfällen låsning av sådana säkerhetsattribut som kan anses vara röjda ska ske.
- Det ska finnas dokumentation som ur ett användarperspektiv beskriver hur säkerhetsfunktionen fungerar och som innehåller instruktioner och riktlinjer om hur man på ett säkert sätt använder säkerhetsfunktionen.