Pålitliga plattformar

JACOB LÖFVENBERG OCH KRISTOFFER LUNDHOLM

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--3136--SE
ISSN 1650-1942

Underlagsrapport
December 2010

Informationssystem

Jacob Löfvenberg och Kristoffer Lundholm

Pålitliga plattformar

Titel	Pålitliga plattformar
Title	Trustworthy platforms
Rapportnr / Report No.	FOI-R--3136--SE
Rapporttyp	Base data report
Report Type	Underlagsrapport
Månad / Month	December / December
Utgivningsår / Year	2010
Antal sidor / Pages	21
ISSN	1650-1942
Kund / Customer	FM
Projektnr / Project No.	E7138
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

Mjukvarubaserade plattformar med hög tillförlitlighet har många användningar inom IT-säkerhetsområdet. De kan benämnas med olika namn, men gemensamt för dem är att det på något sätt går att verifiera att mjukvaran inte förändrats sedan den installerats i maskinen.

I rapporten definieras och diskuteras begreppet ”pålitlig plattform” för denna typ av system. Olika tekniker för att bygga pålitliga plattformar beskrivs och nyttan med sådana plattformar för Försvarsmakten förklaras.

Nyckelord: IT-säkerhet, säker plattform, trusted platform, TPM

Abstract

Software-based platforms with a high level of trustworthiness has many uses in IT security. There are different names for them, but they all have in common that it is possible to verify that the software has not been modified since installed in the machine.

In this report the concept ”pålitlig plattform” (approximately ”trustworthy platform” in English) is defined and discussed for this type of system. Different technologies for building trustworthy platforms are described, and the benefit of such platforms for the Swedish Armed Forces is explained.

Keywords: IT security, trusted platform, TPM

Innehållsförteckning

1 Inledning

De senaste decennierna har uppvisat en enorm ökning i antalet IT-system i samhället och vi ser fortfarande en tilltagande digitalisering av allt fler tekniska och administrativa system. En av de tydligaste trenderna just nu är den ökade sammankopplingen av system. Inte bara blir systemen sammankopplade utan de kommunicerar i allt högre grad själva, utan inblandning av människor. Denna sammankoppling ställer krav på att systemen själva, utan direkt stöd av människor, måste klara av att skilja på information som ska förmedlas och information som måste hållas inom systemet. För att kunna lita på att information hanteras korrekt krävs en viss nivå av säkerhet och därmed krav på tilltron till att systemens filtreringsfunktion fungerar korrekt. Detaljerna i situationen avgör hur hög denna nivå behöver vara.

Frågor kring tilltro till system är extra relevanta när funktioner implementerats i mjukvara. Anledningen till detta är att mjukvarubaserade system oftast är mer komplexa än helt hårdvarubaserade system. Detta beror i sig på att mjukvara är billigt att replikera och att det därför är lockande enkelt att infoga existerande, generella mjukvaror på ett sätt som gör att även relativt enkla system kan komma att innehålla stora mängder kod. Mjukvara är dessutom mycket enklare än hårdvara att modifiera i efterhand.

1.1 Ett inledande exempel

För att exemplifiera hur ett till synes enkelt system egentligen är ganska komplext, väljer vi att titta på bredbandsroutrar[1] för konsumentbruk. I dessa finns i allmänhet, förutom den uppenbara näthårdvaran, en generell processor som kör ett operativsystem, på vilket ett flertal servertjänster kör. Det finns till exempel nästan alltid en webbserver med vars hjälp administration och konfiguration kan göras och det är inte ovanligt att det också finns säkerhetsrelaterade funktioner. Exempel på detta är blockering av vissa externa webadresser och möjlighet att begränsa internetåtkomsten för interna datorer baserat på tidpunkt (klockslag och veckodag) och trafiktyp.

För en hemmamiljö är det troligen tillräckligt att tillverkaren försäkrar att systemet är säkert för att de flesta nyttjare ska känna sig tillräckligt trygga med att det fungerar som det ska. Men i ett sammanhang med höga krav på säkerhet skulle ett antal frågor infinna sig:

- Hur pålitliga är de som designat hård- och mjukvaran i routern?
- Har designerna gjort rätt i sin bedömning av hotbild, risknivå och säkerhetsbehov?
- Hur skickliga är utvecklarna och hur väl fungerar utvecklingsmetoden de använt för att göra säkra implementationer?

[1] En bredbandsrouter är oftast en ethernetswitch för ett lokalt nät, ihopkopplat med en gateway för anslutning till externt nät. Ofta finns ytterligare funktioner för att hantera det interna och det externa nätet.

- Har tillverkaren av routern använt hård- eller mjukvara som de inte själva har utvecklat? I så fall, hur pålitlig är denna?
- Hur har överföringen av källkod till maskinkod gjorts? Kan det finnas svagheter i kompilatorn, antingen på grund av buggar eller på grund av att någon medvetet fått den att bete sig på något speciellt sätt?
- Innehåller just det här exemplaret av routern den mjukvara den ska innehålla och ingenting annat?
- Är konfigurationen av routern korrekt?
- Går det att påverka routern efter uppstart så att den beter sig felaktigt?

Dessa frågor handlar i stor utsträckning om riktighet och pålitlighet. Frågorna visar att även om det finns kända säkerhetsfunktioner, som är anpassade för den existerande hotbilden, är frågorna kring riktighet och pålitlighet ett kvarstående problem som måste lösas separat.

1.2 Syfte

Med denna rapport vill vi beskriva och tolka innehållet i begreppet ”pålitlig plattform” (engelska: trusted platform) och vilket värde och funktion en pålitlig plattform har i samband med de säkerhetsproblem som finns inom Försvarsmakten. Vidare vill vi ge en översikt över de kommersiella tekniker som finns för att skapa en pålitlig plattform.

1.3 Motivering

I sammanhang som kräver hög säkerhet måste det vara möjligt att vara säker på att det aktuella systemet säkerhetsmässigt beter sig som det ska. Detta görs i allmänhet genom olika former av granskning för att verifiera att: kraven uppfylls av designen, designen uppfylls av källkoden, källkoden implementeras av den körbara koden, den körbara koden laddas in i systemet vid tillverkningen samt att systemet med tillhörande hanteringsregler omöjliggör otillbörlig förändring av systemet. Omöjliggörande av otillbörliga förändringar brukar, i sammanhang med hög säkerhet, kräva att systemet väsentligen är frikopplat från andra system och/eller nät. I de sammanhang där frikoppling inte är möjligt krävs speciella former av lösningar för att erhålla tillräcklig tilltro till säkerheten, vilket är vad den här rapporten handlar om.

Skälet för att göra denna genomgång är att pålitliga plattformar finns som en ingående, nödvändig komponent i många säkerhetskritiska system. I diskussioner och dokument nämns pålitliga plattformar ibland explicit, om än med andra namn eller bara till funktionen, men lika ofta saknas direkt omnämnande av sådana. Behovet framgår dock när systemfunktionen analyseras närmare. Bristen på omnämnande, och den otydliga beskrivningen i de fall frågan ändå berörs, gör att vi tror att en närmare genomgång och diskussion är av värde.

2 Bakgrund

Vi inleder detta avsnitt med att presentera några begrepp som är centrala för diskussionen vi för senare. Efter terminologidelen beskriver vi en del material som behövs som bakgrund för resten av rapporten.

2.1 Begrepp

Ett centralt begrepp i den här rapporten är *assurans*, som i [1] definieras som

> tillit till att ett systems eller en produkts *säkerhetsfunktioner* uppfyller specificerade säkerhetskrav.

Oftast finns i betydelsen också en underton av att det handlar om hur mycket tillit systemet är förtjänt av, och att assuransen är kopplad till systemet i sig. Detta till skillnad från engelskans *trust*, som är ett mått på tillit men som kan tänkas vara oförtjänt eller välförtjänt. Assurans associeras vidare ofta till graden av validering av systemets säkerhetsfunktioner. Denna association kan antagligen förklaras med termen *evaluation assurance level*, som är ett begrepp från Common Criteria och som anger ett numeriskt värde som är kopplat till hur noggrant säkerhetsutvärderat systemet är.

Vi kommer att använda begreppet assurans i betydelsen

> graden till vilken det finns skäl att lita på att ett systems säkerhetsfunktioner fungerar enligt specifikation.

I praktiken kommer detta ”skäl” att lita på säkerhetsfunktionerna oftast från någon typ av formell granskning eller godkännande. Inom Försvarsmakten är det MUST som granskar och godkänner säkerheten i IT-system.

Ett annat väsentligt begrepp är *riktighet*, vilket i [1] definieras som

> skyddmål [sic!] att informationen inte förändras, vare sig obehörigen, av misstag eller på grund av funktionsstörning.

Ofta används också begreppet *integritet* med ungefär denna betydelse. Integritet kan dock ha delvis andra betydelser och associationer, varför vi kommer att använda ”riktighet”. Notera att det inte krävs någon sekretess för att uppfylla definitionen av riktighet.

Det engelska begreppet *trusted computing base* används som ett samlande namn på de delar i ett system som måste fungera korrekt för att säkerheten ska kunna upprätthållas. En mer formell definition ges i [2]

> the totality of protection mechanisms within it, including hardware, firmware, and software, the combination of which is responsible for enforcing a computer security policy.

Det för rapporten mest centrala begreppet är pålitlig plattform (engelska: *trusted platform*). I annan litteratur används ibland begreppet säker plattform med likartad betydelse; en tänkbar tolkning är att en sådan plattform är säker i det avseende som är relevant. Vår bild är dock att detta är ett för långtgående krav. En teknisk lösning kan inte vara säker i sig själv eftersom säkerhet endast går att erhålla i kombination med ett lämpligt användarbeteende. Vi kommer istället att använda *pålitlig plattform* i betydelsen

> en plattform på vilken det är möjligt att verifiera riktigheten i mjukvaran som körs.

En tekniskt nogrannare definition ges i avsnitt 3.1.

2.2 Motiverande tekniska trender

Det finns tekniska trender som gör att bland annat nyttan av mjukvarulösningar med hög assurans blir allt mer tydlig. Här presenterar vi två sådana trender som vi ser som underliggande drivkrafter för detta.

2.2.1 Allt mer blir mjukvara

Allt fler tekniska apparater och system är helt eller delvis byggda med mjukvara, något som i många fall inte alls är uppenbart. Orsaken är att det är jämförelsevis billigt och enkelt att utveckla mjukvara jämfört med att utveckla hårdvara. Mjukvara har dessutom den fördelen att den kan utvecklas, vidareutvecklas och korrigeras efter det att tillhörande hårdvara är designad, eller till och med efter det att hårdvaran är levererad. Detta medför både en ökad flexibilitet och en kortare utvecklingstid, men tyvärr också att produkten kan levereras eller säljas trots att mjukvaran ännu är i ett dåligt skick och egentligen behöver förbättras ytterligare. Konsekvensen för konsumenten är allt oftare att produkter inte har en välfungerande mjukvara vid leveranstillfället och att produkten fram tills dess att mjukvaran korrigerats fungerar sämre än den borde.

Ofta används standardprocessorer för att exekvera mjukvaran som finns i mjukvaruprodukter. Orsaken är att de idag är billiga att köpa in och snabba och enkla att utveckla till. Det finns också en stor mängd existerande programkod att återanvända, vilket ytterligare underlättar och reducerar utvecklingskostnader. I många sammanhang där det inte finns behov av speciellt utrymmes- eller energisnåla lösningar, används av samma skäl till och med datorer av PC-typ med allmänt spridda operativsystem (vanligast är Windows, Linux och BSD). Dessutom finns idag PC-lösningar som är jämförelsevis snåla, både utrymmesmässigt och energimässigt. Både Linux och BSD går också att anpassa till just de behov som finns, vilket innebär att de kräver mindre i form av hårdvaruprestanda.

Att mjukvara fortsätter att utvecklas även efter att systemet levererats syns även i den militära teknikutvecklingen, även om problemen med halvfärdig mjukvara i tidiga versioner av produkter inte är tydlig på samma sätt som i konsumentkretsar. Att

väsentliga funktioner kommer i senare versioner av mjukvaran förekommmer visserligen, men de funktioner som finns brukar vara färdigutvecklade och välfungerande.

Ett exceptionellt exempel, där flexibiliteten hos mjukvara utnyttjas till sin fulla kapacitet, är mjukvarudefinierad radio (SDR, från engelskans software defined radio). I en SDR implementeras själva radiofunktionaliteten i mjukvara, vilket innebär att samma hårdvara kan användas till att implementera vilken sorts radio som helst (inom de gränser som sätts av hårdvarans förmåga). Eftersom radiofunktionen körs som mjukvara blir det i princip möjligt att byta radiofunktion bara genom att ladda in en ny fil. En radio som kommunicerar med en svensk militär radiostation kan strax efteråt ta emot information från en civil sändare som arbetar på ett helt annat sätt, för att därefter vidarebefordra något till en koalitionspartner från ett annat land med ytterligare en annan teknik.

2.2.2 Allt mer kopplas samman

Som nämnts tidigare finns en fortgående trend att i allt högre utsträckning koppla samman system. Denna trend finns även inom den militära sfären. System som tidigare varit separerade av säkerhetsskäl kopplas samman, eftersom det möjliggör en så mycket bättre funktion och effekt. Samtidigt innebär det en stor utmaning att koppla samman system där inte all information får delas.

2.3 Ett klassiskt exempel på en högassuranslösning för sammankoppling

Ett klassiskt sätt att hantera kommunikation över osäkra medier är kryptering. Krypterad kommunikation kan ses som ett sätt att koppla samman system på olika platser genom att skapa en säker tunnel. Information från insidan skickas över det osäkra mediet, men bara i skydd av krypteringen. Även här krävs hög assurans hos krypteringssystemet, vilket i de flesta militära sammanhang har inneburit kryptolösningar i hårdvara med en arkitektur som är särskilt lätt att granska. Denna arkitektur brukar kallas röd-svart-separering och idén är att den röda sidan (den med känslig information) och den svarta sidan (den oskyddade) är helt separerade bortsett från krypteringsfunktionen. Denna funktion implementeras i sin tur i en form som är så enkel och lättförståelig som möjligt så att det går att få tillräcklig assurans för den säkerhetsnivå som är aktuell. I och med att krypteringsutrustningen är i hårdvara och måste hållas under betryggande uppsikt kan nyttjaren vara säker på att utrustningen uppför sig på samma sätt som när den levererades. Genom noggrann granskning och övervakning under utveckling och produktion kan leverantören vara säker på att utrustningen vid leverans gör det den ska enligt specifikationen, och kedjan är därmed hopbunden hela vägen. Värt att notera är vidare att i och med att all kommunikation genom krypteringsutrustningen krypteras (eller dekrypteras, beroende på kommunikationsriktningen) så kan inte en utomstående angripare heller skicka falsk information till den skyddade insidan, den röda sidan, utan att ha tillgång till krypteringsnyckeln.

2.4 Behov

Det finns, särskilt i militära sammanhang, många tillfällen då det behövs system med hög assurans. Kryptering är ett tydligt exempel men varje apparat som är i kontakt med entiteter med olika sekretessnivå och/eller sekretessägare har behov av assurans på någon nivå. Detta gäller såväl för system som kopplar samman eller separerar olika nät (brandväggar, dataslussar) som för apparater som kan komma att handhas av personer med olika behörighet (kryptotelefon, hemligdator, hemlighårddisk). I det senare fallet är behovet separation i tiden snarare än i rummet. Hemligheter som kommunicerats eller lagrats av en person vid ett givet tillfälle får ibland inte vara tillgängligt för nästa person. Ett aktuellt exempel är radering av hårddiskar, där det relativt nyligen godkänts ett program som anses ha tillräcklig assurans för att det ska gå att återanvända hårddiskar som innehållit material klassat HEMLIG/SECRET; dock bara i verksamhet med minst samma sekretessnivå. Raderingsprogrammet anses alltså säkert nog för att (i tiden) separera entiteter med *samma* höga sekretessnivå. Vill man däremot göra en hårddisk med material klassat HEMLIG/SECRET till klassen ÖPPEN finns ingen lösning med tillräcklig assurans som bibehåller funktionen hos hårddisken; den måste förstöras.

Som framgår ovan används standarddatorer även i sammanhang där man inte förväntar sig generella datorer och där det inte syns att det är vanliga datorer som används. Detta i kombination med de beskrivna behoven av assurans och riktighet implicerar att det finns ett behov av vanliga datorer som det går att lita på mer än vad som normalt är möjligt. Det går att komma en bra bit på väg genom att minimera mängden installerad mjukvara, konfigurera systemet på ett restriktivt och säkerhetsmedvetet sätt och sedan granska lösningen noggrant. Det går dock att komma ännu längre med olika typer av hårdvarustöd. Signerad kod och krypterad lagring med krav på dekryptering i separat hårdvara, t.ex. med hjälp av ett smart kort, kan skydda mot vissa typer av attacker. Dock finns det inget som hindrar att angriparen ger sig på de i mjukvara installerade kontrollmekanismerna som verifierar de kryptografiska kontrollmekanismerna. Problemet är att vi måste kunna lita på kontrollfunktionerna, och de är förstås lika möjliga att påverka som den övriga mjukvaran. För att nå en högre assuransnivå behövs således hårdvarustöd som kan verifiera dessa ”lägre” mjukvarunivåer.

3 Säkerhet i mjukvarubaserade plattformar

I detta avsnitt diskuterar vi vad som krävs för att ett mjukvarubaserat system ska kunna vara säkert, och mer specifikt beskriver vi vad vi menar med begreppet ”pålitlig plattform”.

3.1 Säkerhetskedjan

Att bygga ett system som kan ha hög säkerhet innebär att smida en kedja med flera sammanhängande länkar som alla måste vara starka. Följande uppräkning är ett försök att beskriva vilka dessa länkar är. Beskrivningen är gjord på en relativt hög nivå, vilket innebär att det är möjligt att bryta ner de olika delarna ytterligare och därmed få en noggrannare beskrivning. För oss räcker dock denna ganska grova beskrivning eftersom vi bara vill använda den till att peka ut var i kedjan den pålitliga plattformen kommer in.

1. Utvecklarna

   De som designar systemet måste vara betrodda. Detsamma gäller för programmerarna som implementerar designen.

2. Systemdesign

   Designen av systemet behöver vara gjord på ett säkerhetsmedvetet sätt som hanterar de säkerhetsfrågor som är relevanta för det färdiga systemet. Designen behöver verifieras för att den ska gå att lita på.

3. Utvecklingsprocessen

   Om systemet är säkerhetskritiskt måste utvecklingsprocessen vara utformad så att den stödjer utvecklarna i att både göra rätt och att producera lättgranskad källkod.

4. Kompilering

   Källkoden omvandlas till körbar maskinkod av en kompilator. Detta är en automatisk process som endast i undantagsfall är möjlig att verifiera med avseende på riktighet; istället måste kompilatorn i sig ha hög assurans. Vidare måste det säkerställas att det är exakt den källkod som utvecklingsprocessen resulterade i som faktiskt kompileras.

   Här kan användning av kryptografiska hashfunktioner vara ett verktyg. Genom att verifiera att hashvärdet är oförändrat sedan utvecklingsprocessen avslutades kan det verifieras att källkoden inte modifierats. Detta gäller förstås bara om det går att lita på implementationen och exekveringen av hashalgoritmen.

5. Installation av mjukvara i hårdvaran

   Vid installation av mjukvaran, dvs. resultatet av kompileringen, i hårdvaran måste det säkerställas att det är just det som kompilatorn producerade som faktiskt

installeras. Allt från kompileringen, utan ändringar, och inget annat ska installeras. Återigen kan kryptografiska hashfunktioner vara till hjälp, på samma sätt som för källkoden, och med samma begränsningar.

6. Uppstart av hårdvaran

   När hårdvaran startas ska den korrekta mjukvaran laddas. Ingenting får ha modifierats, lagts till eller tagits bort. Detta kan verifieras med hjälp av kryptografiska hashfunktioner, men återigen måste det gå att lita på implementationen och exekveringen av denna kontroll. För att detta ska vara möjligt bör denna kontroll vara implementerad i hårdvara så att den inte går att påverka utan fysiskt tillträde. Med en väl uttänkt lösning räcker det med en minimal kontroll i hårdvara, för att verifiera riktigheten hos mjukvara som sedan kan kontrollera resten av det som ska laddas. Detta kan ske genom att systemet startas i steg där allt större och mer komplexa delar kontrolleras och laddas i ”lager” utanpå varandra.

7. Säkerhet under drift

   När hårdvaran är uppstartad och systemet är igång måste säkerheten fortsatt upprätthållas. Inget har vunnits om systemet korrekt granskar sig självt vid uppstarten, bara för att genast bli komprometterat när det kommer i drift. Ett sådant skydd utgörs av vanligt IT-skydd, men kan vidare stödjas av särskilda komponenter som verifierar att mjukvaran i minnet inte har modifierats. Detta kan ske antingen kontinuerligt eller inför varje säkerhetskritisk händelse.

Definitionen av pålitliga plattformar i avsnitt 2.1 kan med hjälp av denna kedja noggrannare beskrivas som att en plattform är pålitlig om stegen 5 till 7 är uppfyllda.

4 Existerande och tidigare lösningar

I detta avsnitt betraktar vi existerande tekniker och metoder för att erhålla pålitliga plattformar. Det rör sig både om principiella förhållningssätt, dvs. lösningar med egenskaper som liknar pålitliga plattformar och existerande kommersiella tekniker som behandlar just de frågor som är relevanta för pålitliga plattformar. Genomgången är inte fullständig men fungerar för att visa på spännvidden i problemdomänen och lösningsförslagen.

4.1 Ren hårdvara

En ren hårdvarulösning är ofta robust eftersom den är svår att få att ändra beteende, särskilt utan fysiskt tillträde. Ett exempel på denna typ av lösning, kryptering, beskrevs i avsnitt 2.3, och det kan ses som typiskt.

4.2 Digital rights management

Digital rights management (DRM) är ett begrepp som används för tekniker som används för att styra och begränsa hur köpare kan använda digitala mediafiler de köpt.

Upphovsrättsinnehavare, distributörer och mediaindustrin i övrigt fick ett stort intresse för denna typ av teknik när mediainspelningar på allvar började distribueras och användas i digitalt format. På den analoga tiden gjorde kvalitetsförsämringar vid kopiering att storskalig kopiering, i flera led, inte var möjligt för privatpersoner. Vid digital lagring är kopian dock identisk med originalet, vilket gör att varje ny kopia kan fungera som ett original vid en ny kopiering. Detta gör att även kopiering mellan enskilda personer, sett över hela populationen, kan bli mycket omfattande.

Tanken med DRM är att tekniskt begränsa de möjligheter som den digitala tekniken ger, nämligen kopiering utan kvalitetsförlust. Antingen tillåts inte kopiering alls, eller så begränsas kopieringen på något sätt, t.ex. till ett fåtal kopior. För att uppnå detta måste hårdvaran som används vara begränsande för användarna, och dessa begränsningar får inte gå att förändra. Ur rättighetsägarens perspektiv ska alltså hårdvaran utgöra en pålitlig plattform. Ur samma perspektiv är användarna att betrakta som motståndare. Värt att notera är att DRM-systemen även kan begränsa legal kopiering, beroende på hur harmoniserade upphovsrättslagstiftningen och DRM-systemens policy är.

System för DRM finns, eller har funnits, i ganska stort antal. Exempel är: digital-TV, film på DVD och blu-ray, TV-spel, musik och musikspelare, e-böcker och datorspel.

4.3 Trusted platform module och Trusted computing

Trusted platform module (TPM) är en plattform vars utveckling drivs av Trusted computing group (TCG) som är ett initiativ startat av AMD, HP IBM, Intel, Microsoft m.fl.

Målet för TCG är att utveckla *Trusted computing*, något som ska implementeras i en TPM. Med Trusted computing menas huvudsakligen att det ska gå att verifiera att endast auktoriserad mjukvara kör i ett system, dvs. det vi har kallat en pålitlig plattform.

På den kommersiella arenan finns inte något riktigt alternativ till TPM. Det finns en kinesisk lösning som kallas Hengzhi-chippet, men denna följer inte TCG:s specifikation och det verkar inte finnas någon detaljerad tillgänglig information vilken funktion chippet erbjuder.

TPM är däremot relativt väl spridd. TPM-stöd finns sedan flera år hos komponenter från de flesta stora tillverkare av datordelar, från moderkort och CPU:er till bärbara och stationära datorer. Visst stöd i mjukvara finns hos både Windows och Linux och fortsatt utveckling pågår. Som exempel kan nämnas att Department of Defence i USA kräver att alla nya datorer ska innehålla ett TPM-chip.

4.3.1 Ingående funktioner

TPM innehåller flera tekniker och funktioner. De som vi ser som de viktigaste är:

- Signeringsnyckel (engelska: endorsement key)

  Varje TPM-chip innehåller en hemlig signeringsnyckel som kan användas för att skapa digitala signaturer. Nyckeln är en 2048-bits RSA-nyckel och skapas vid tillverkningen av chippet och kan inte ändras eller läsas i efterhand.

- Minnesskydd (engelska: memory curtaining)

  Minnesskyddet i TPM-plattformar är mycket starkare än i en vanlig dator. Delar av primärminnet kan göras helt avskilda från resten av systemet, så att varken användare eller operativsystem kan nå det. Den exakta utformningen av skyddet är implementationsberoende och kan variera mellan tillverkare. En implementation är Trusted execution environment från Intel, som kräver ett TPM-chip och särskilt hårdvarustöd i CPU och moderkort för att fungera.

- Förseglad lagring (engelska: sealed storage)

  Förseglad lagring gör det möjligt att göra information tillgänglig bara till en viss kombination av hårdvara och mjukvara. På detta sätt kan man hindra känslig information från att bli tillgänglig för hård- eller mjukvara som inte är pålitlig ur informationsägarens perspektiv.

- Fjärrattest (engelska: remote attestation)

  Fjärrattesten gör det möjligt för TPM-chippet i en dator att på ett pålitligt sätt visa för en annan maskin, via ett nät, vilken hård- och mjukvara som används. På detta sätt får den andra maskinen ett underlag på vilket den kan basera beslut angående om den ska betrakta den första maskinen som pålitlig eller inte.

Värt att påpeka är att TPM alltid måste aktiveras för att kunna användas. Den som köper en ny dator riskerar alltså i dagsläget inte att bli påtvingad TPM-funktionerna. En risk som påtalats är annars att TPM-chippet ska användas för otillbörlig begränsning av vad användaren kan göra med sin egen information, hård- och mjukvara.

4.3.2 Attacker

Ett antal sätt att bryta den säkerhet som TPM-chippet skapar har presenterats; här beskrivs två av dessa:

- Cold boot

  Minnesskyddet innebär att information som lagras i de minnesareor som är skyddade inte är åtkomliga för icke auktoriserad hård- och mjukvara. Genom att kraftigt kyla ner minnenas RAM-kretsar är det möjligt att stänga av datorn och flytta minnena utan att informationen i RAM-kretsarna försvinner. Minnena kan sedan placeras i en dator utan begränsningar och på så sätt kan innehållet i minnena kopieras och analyseras.

- Öppning av TPM-chippet

  Med hjälp av vanligt förekommande kemikalier har det visat sig möjligt att öppna TPM-chip på ett sätt som gör det möjligt att fysiskt avlyssna den interna kommunikationen på chippet. Denna kommunikation är inte krypterad eller skyddad på annat sätt eftersom den sker inuti chipet, och därför ansetts tillräckligt säker.

  Även om attacken inte kräver någon avancerad kemisk utrustning behövs en Focused Ion Beam (FIB), ett sorts elektronmikroskop. Det är med hjälp av denna FIB som det är möjligt att koppla in sig på de mycket små ledningsbanorna i chippet. Proceduren, med rätt utrustning, sägs ta 6–7 timmar.

Gemensamt för de två attackarna är att de kräver fysisk tillgång till maskinen som ska attackeras. Att motståndskraften mot sådana attacker är begränsad är rimligt då detta inte har varit något designmål för TPM. Om det är ett rimligt hot eller inte beror på den enskilda situationen.

4.4 High assurance platform

Amerikanska NSA har ett program med namnet High Assurance Platform (HAP) [3] vars vision är att definiera ett ramverk för utvecklingen av nästa generations plattform för säker databehandling (engelska: secure computing platform). Tanken är att HAP ska koppla kommande COTS[2]-säkerhetstekniker med ett antal assuranstekniker. Syftet är att ge kommersiella aktörer möjlighet att utveckla verifierbart säkra, managerbara och användbara datorkomponenter som i sin tur kan användas till att bygga COTS-baserade, assurerbara, kommersiella produkter för slutanvändare.

HAP är tänkt att utvecklas och demonstreras i tre steg:

- HAP Release 1

  Första steget ska innehålla hårdvarustöd för virtualisering och hårdvarustöd för fjärrattest. Stödet för virtualisering ska erhållas från moderkortet och är tänkt att ge både bättre prestanda och starkare separation mellan de virtuella maskinerna. Hårdvarustödet för fjärrattestering ska erbjudas via TPM.

[2]COTS–Commercial Off The Shelf: en term för teknik som är allmänt tillgänglig och går att köpa färdig

- HAP Release 2

  I steg två ska teknisk administration och användbarhet hos plattformen förbättras. Huvudsakligen rör det sig om olika typer av fjärradministration som ska möjliggöras, men det ingår även stöd för isolering av drivrutiner, stöd för fjärrattestering av mjukvara inuti virtuella maskiner, periodiskt återkommande fjärrattestering och minnesskydd mot DMA.

- HAP Release 3

  Steg 3 ska lyfta assuransnivån till en nivå som är tillräcklig för att kunna hantera information från multipla, godtyckliga informationssäkerhetsklasser i samma fysiska plattform.

General Dynamics C4 Systems erbjuder HAP-datorer byggda på arbetsstationer och bärbara datorer från HP och Dell som alla synes vara av standardtyp. Enligt specifikationen [4] tillåter datorerna material som spänner från Unclassified till Secret eller från Secret till Top Secret.

5 Nytta för Försvarsmakten

En pålitlig plattform kan vara en väsentlig komponent i många säkerhetskritiska sammanhang. Detta gäller oavsett om det är en civil eller militär frågeställning. Dock är det få civila tillämpningar som kan förvänta sig en så resursstark angripare som det som anses normalt i militära sammanhang. Att Försvarsmakten har behov av pålitliga plattformar är därför inte överraskande. I detta avsnitt ger vi några för Försvarsmakten relevanta exempel, explicita eller principiella, där pålitliga plattformar är väsentliga för att upprätthålla en tillräcklig säkerhetsnivå.

5.1 Gemensamt taktiskt radiosystem

Gemensamt taktiskt radiosystem, GTRS, är ett kommunikationssystem som är under utveckling med sikte på att tillhandahålla såväl förbättrade 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.

En central del i GTRS är ett delsystem kallat Crypto sub system (CSS). CSS är väsentligen en egen dator vars syfte är att implementera kryptofunktioner, säker transportlagring av mjukvara och verifiering av systemets tillstånd. Dess funktion påminner alltså om TPM-funktionerna (se avsnitt 4.3), men med högre krav på manipuleringsskydd och assurans.

Vid uppstart börjar CSS med att verifiera sin egen riktighet, både med avseende på hård- och mjukvara. Därefter verifieras och startas delar som är mer perifera och för huvudfunktionerna mer centrala. Som synes utgör GTRS, med CSS i spetsen, i alla avseenden det vi kallar en pålitlig plattform.

5.2 Brandvägg

En mer principiell funktion som är lämplig att betrakta är en brandvägg. Syftet med en brandvägg är att separera två eller flera nät med viss kontrollerad, bibehållen kommunikationsförmåga. Det är väsentligt för brandväggens funktion att den lever upp till de specifikationer som finns och att den inte är modifierad på ett sätt som förändrar dess beteende. Detta är precis de egenskaper som en pålitlig plattform kan bidra med, nämligen att verifiera att ett system innehåller rätt mjukvara och konfigurationsparametrar.

5.3 Datasluss

En datasluss är ett tekniskt system som kan kopplas in mellan två andra tekniska system, med olika säkerhetsklass eller systemägare, för att styra flödet av data mellan systemen. Till skillnad från en brandvägg ska en datasluss inte bara stoppa otillåtna anslutningar eller protokoll utan även kontrollera att endast tillåten information överförs.

6 Diskussion och slutsatser

Vi har sett att det finns trender inom IT-området som driver på behovet av pålitliga plattformar, särskilt för Försvarsmakten. Sammankoppling av nät med känslig information måste göras med hjälp av system med hög assurans, och av pris- och flexibilitetsskäl är det mycket fördelaktigt om dessa system kan vara mjukvarubaserade. Detta leder till ett behov av pålitliga plattformar där det går att lita på att mjukvaran inte förändrats sedan den installerades i hårdvaruplattformen.

Det finns gott om sammanhang där pålitliga plattformar är nödvändiga för att uppnå tillräcklig IT-säkerhet inom Försvarsmakten. Exemplen i avsnitt 5 är inte en uttömmande genomgång; vi ser även fall då en pålitlig plattform inte är absolut nödvändig, men där den skulle bidra med ökad säkerhet eller robusthet. Med anledning av detta menar vi att den potentiella nyttan av pålitliga plattformar inom Försvarsmakten är stor.

De kommersiella försöken med pålitliga plattformar har oftast handlat om att förhindra användare från att använda sina inköpta mediefiler på ett sätt som bryter mot leverantörernas policy. Att lyckas med att införa detta är extra besvärligt eftersom de tekniska systemen av nödvändighet måste vara mycket billiga och lättanvända, och eftersom angriparen (som är densamma som användaren) har fysisk tillgång till den pålitliga plattformen.

För Försvarsmakten handlar det snarare om att kunna verifiera att mjukvaran i en hårdvaruplattform är riktig, dvs. att det inte modifierats. Den tänkta attacken är dock antagligen en nätbaserad attack, dvs. angriparen har inte fysisk tillgång till den pålitliga plattformen. Denna skillnad har stor betydelse för möjligheten att uppnå tilltro till lösningen, eftersom fysisk tillgång är en oerhört stor fördel för angriparen. Skillnaden leder vidare till en naturlig uppdelning i fall där den pålitliga plattformen ska användas i en fysiskt kontrollerad miljö där endast nätbaserade attacker är möjliga, och fall där den måste tåla att komma under angriparens kontroll.

Att funktionen hos pålitliga plattformar är värdefull menar vi är uppenbart. Ett ytterligare steg vore att utforma en standardiserad, pålitlig plattform som skulle kunna återanvändas i många system med sådana behov. En sådan standardiserad, pålitlig plattform skulle förmodligen inte kunna nå upp till den allra högsta assuransnivån. Frågan är snarare om det skulle gå att konstruera den så att den når en nivå som är allmänt användbar. Av ekonomiska skäl bör den vara byggd till allra största delen av kommersiellt tillgänglig teknik.

TPM är en metod för att erhålla just det som behövs för att skapa en pålitlig plattform. Vår bedömning är dock att det är svårt att tänka sig att en kommersiell teknisk lösning, även om den behandlar precis rätt problem, är möjlig att lyfta till en assuransnivå som är tillräcklig för att kunna använda inom Försvarsmakten. Civil IT-säkerhetsteknik är mycket sällan anpassad för den hotbild och risknivå som finns i militära sammanhang. Något som dock skulle kunna vara realistiskt vore att använda TPM-bestyckade maskiner för att öka robustheten *inom* ett nät, där säkerheten i systemet inte står och faller med de egenskaper som TPM-chippet tillför.

Den högassuransplattform som beskrivs i avsnitt 4.4 är dock intressant, då den i förlängningen är tänkt att uppnå tillräcklig säkerhet för att användas för separation av militärt sekretessklassad information.

Referenser

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

[2] Department of Defense trusted computer system evaluation criteria, DoD 5200.28-STD, 1985. I terminologilistan (glossary) under ”Trusted Computing Base (TCB)”.

[3] High Assurance Platform Program, webplats med adress http://www.nsa.gov/ia/programs/h_a_p/index.shtml, kontrollerat tillgänglig 2010-12-08.

[4] Webplats med adress http://www.gdc4s.com/content/detail.cfm?item=35a995b0-b3b7-4097-9324-2c50008b3a75&page=3, kontrollerat tillgänglig 2010-12-08.