Separation säkrar uppkopplade bilar

EXPERTARTIKEL

Smidigast är en hypervisor – en mjuk mur med minimal målyta

Av Lee Cresswell, Lynx Software Technologies

Lee Cresswell började på Lynx som försäljningsansvarig för EMEA år 2013. Innan dess arbetade han på Real-Time Innovations (RTI), Texas Instruments och Wind River. Han började sin karriär som mjukvaruutvecklare och har en kandidatexamen från universitet i York och en MBA från London Business School.

Ingen som läser den här artikeln behöver övertygas om allvaret i säkerhetsfrågorna kring uppkopplade och självkörande fordon. Följderna av bristande säkerhet kan man bland annat läsa om i rapporten ”Remote Exploitation of an Unaltered Passenger Vehicle” av Miller och Valasek.

Deras exempel är inte en isolerad händelse. I princip alla fordon på marknaden har någon form av trådlös kommunikation som utgör en sårbarhet som kan utnyttjas av hackare. Majoriteten av biltillverkarna erbjuder möjligheten att samla historik över körningen och trådlöst överföra den till datacenter inklusive tredjepartscenter. De flesta av dessa överföringar sker utan effektivt dataskydd.

Nästan alla fordon använder en blandning av trådlösa tekniker, inklusive mobilnät, Bluetooth och NFC (närfältskommunikation). Det finns typiskt en direkt väg från den trådlösa kommunikationslänken till den centrala fordonsbussen vilket inte bara ger tillgång till navigation, säkerhet och andra smarta tillämpningar, utan också till bromsar, styrning och farthållare.

När uppkoppling har blivit normen finns en brant inlärningskurva för företag och individer innan de förstått problemen och kunnat implementera lämpliga lösningar.

Standarden för processen heter ISO 26262 och ger ”ett fordonsspecifikt och riskbaserat angrepssätt för att bestämma integritetsnivåerna (Automotive Safety Integrity Levels – Asil))”.

För att kunna tillämpa Asil i olika fordonssystem är en förutsättning att systemen är uppdelade så att mindre kritiska system inte kan komprometterа de mest kritiska systemen.

TRADITIONELLT HAR DET fungerat bra. Styrketsar har varit dedicerade för specifika funktioner som motorstyrning, ABS eller något annat. Men dagens mer holistiska angreppssätt har lett till allt mer interaktion dem emellan.

Ett enkelt exempel på detta är en automatisk växellåda där motorn rapporterar varvtalet och växellådan berättar för de andra modulerna när den byter växel. Tidigare skedde kommunikationen via dedicerat kablage men allteftersom allt fler system, som intelligenta farthållare och bromsar, behöver informationen ledde dedicerade kablar till allt för komplexa system.

Lösningen blev fordonsnätverk, vanligen i form av Can (Controller Area Network) som gör det möjligt att utbyta data samtidigt som kablaget inte växer i takt med att antalet funktioner ökar. Nya processormoduler pluggas enkelt in.

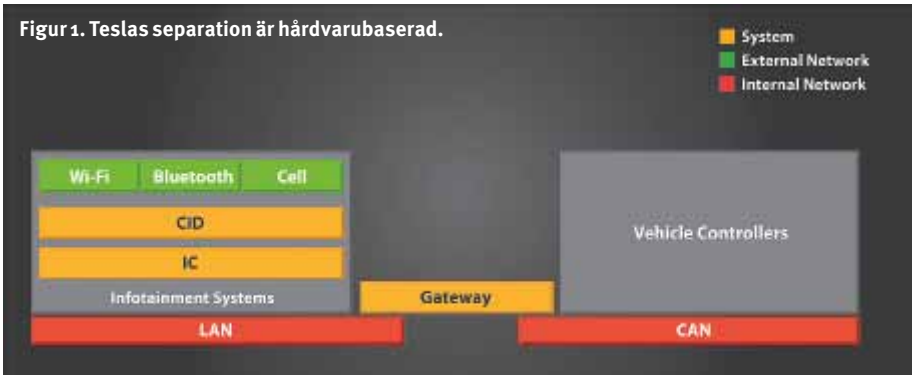
Så länge som fordonet var isolerat från omvärlden var förbindelserna mellan processornoderna ett minimalt säkerhetshot för fordonet. Så snart man bevisat att kommunikationen på bussen inte komprometterade något av de system som redan var anslutna, så kunde de anses vara separerade och principen i ISO 26262 var upprätthållen.

Den uppkopplade bilen förändrade allt. En extern ingång för med sig möjligheten för hackare att angripa en svag punkt eller en attackyta. En sådan måltavla utgör en signifikant risk även om i ett icke-kritiskt system eftersom det i ett nästa steg kan ge tillgång ett kritiskt system. För att säga det rätt ut: att någon har hackat din bilradio innebär inte att de inte har tillgång till bromssystemet.

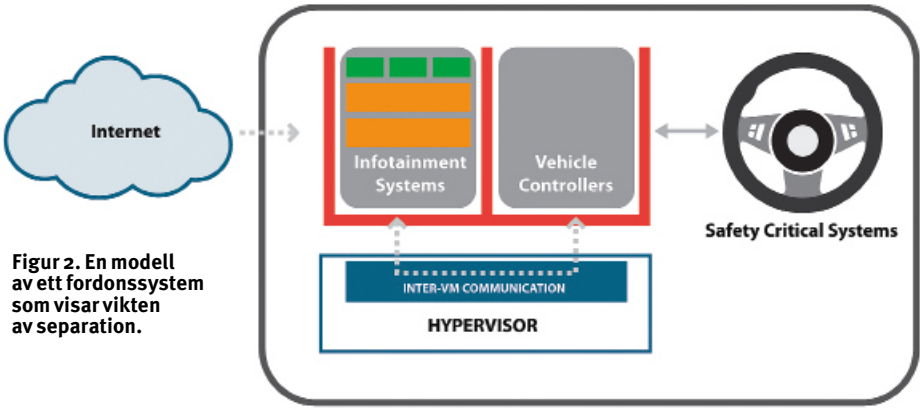
DET STÅR KLART att en uppkopplad bil aldrig kommer att kunna erbjuda samma säkerhet som en som inte är uppkopplad. Men för att en uppkopplad bil ska kunna anses säker och följa principerna i ISO26262 är det nödvändigt att attackytorna minimeras och att separationen mellan systemen optimeras.

Model S från Tesla använder en fysisk gateway för att isolera infotainmentsystemet från säkerhetskritiska delar. I gatewayen finns en strukturerad API som stödjer ett begränsat antal kommandon mellan de två nätverken vilket innebär att om de säkerhetskritiska ska accessas krävs ingående kunskap om API:erna.

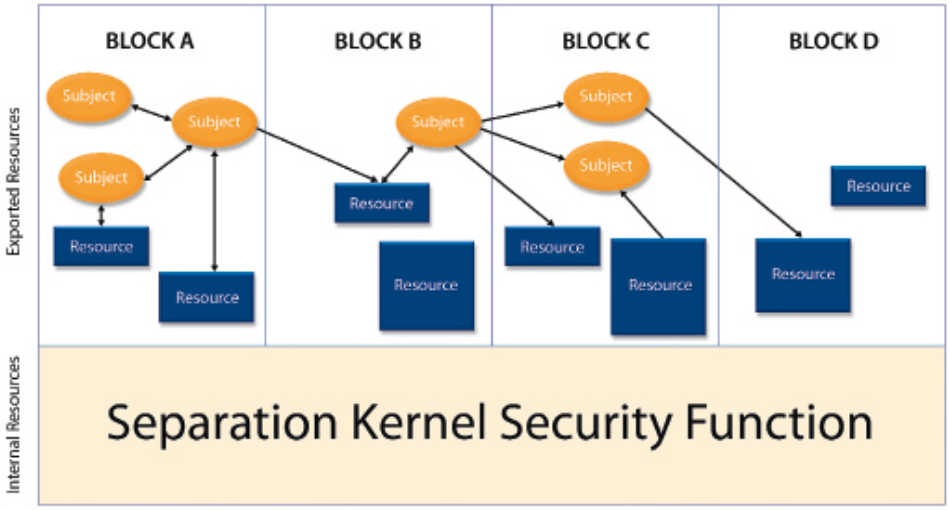
Dock har det visat sig att detta upplägg inte är helt säkert vilket visar hur svår utmaningen är i en uppkopplad bil. Nästan lika viktigt är att metoden för med sig en ansenlig kostnad i form av hårdvara.



Figur 1. Teslas separation är hårdvarubaserad.



Figur 2. En modell av ett fordonssystem som visar vikten av separation.



Figur 3. Att tillämpa lägsta möjliga privilegium i en separerande kärna ger en hög granularitet vad gäller flödeskontroll per subjekt och per resurs.

Även om Tesla har en effektiv metod för att separera systemen så är det uppenbart att motsvarande lösning i mjukvara blir billigare.

Sök efter ”embedded hypervisor” på någon populär sajt för fordonselektronik och det skulle vara lätt att tro att de många lösningar som presenteras är likvärdiga. Men för att utforska det hela lite djupare, utgå från ett system som liknar Teslas som har en hypervisor istället för hårdvara.

FUNKTIONALITETEN I HYPERVISORN är viktig eftersom den ger systemet möjlighet att återspegla funktionen med begränsningar i kommunikationen mellan de två helt olika applikationerna (virtuella maskiner) där den ena hanterar styrsystem och den andra infotainment.

Vad gäller säkerheten är ett fundamentalt krav att om det utåtriktade infotainmentsystemets virtuella maskin hackas, ska den virtuella maskinen som styr fordonet ändå inte vara sårbar. Om de virtuella maskinerna verkligen är separerade av hypervisorn – och inte är sammankopplade via denna – är det viktigt att attackytan görs så liten som möjligt genom att minimera de delade resurserna.

För att illustrera påståendet, ta en KVM, en virtualiseringsinfrastruktur för Linuxkärnan som gör den till en hypervisor. KVM tjänar som exempel på ett problem som delas med alla hypervisorer av typen 1.

Om KVM Linux var den valda hypervisorn i det teoretiska fordonssystemet skulle säkerheten hos den virtuella maskinen för fordonsstyrning vara beroende av en monolitisk kärna som skulle ha minst 390 portar med hundratusentals olika parametrar, implementerade i 19,5 miljoner rader kod som ständigt ändras. Men inte bara det, I/O-stacken skulle ligga i hypervisorns monolitiska kärna vilket gör den till en enkel attackvektor in till den virtuella maskinen som styr fordonet.

KVM Linux ger exakt den funktionalitet för hypervisorn som behövs för att stödja systemets funktionalitet. Men metoden för att separera de virtuella maskinerna med sina olika behov av skydd är långt från optimalt.

ETT BÄTTRE ANGREPPSSÄTT är att behålla funktionaliteten i hypervisorn men att göra separation och minimering av attackyta till det huvudsakliga fokuset för arkitekturen.

För att förstå hur man kan uppnå detta underlättar det att först fundera på principerna för lägsta möjliga privilegier och separerande kärnor (Least Privilege and Separation Kernels).

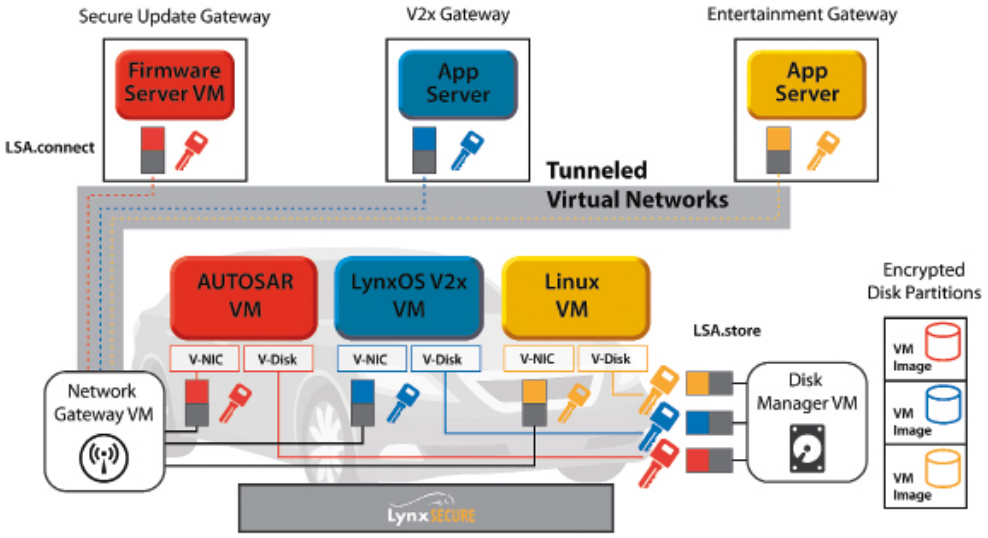
Konceptet med separation stammar från John Rushby som 1981 föreslog att det skulle bestå av en kombination av hårdvara och mjukvara som tillåter multipla funktioner att realiseras på samma fysiska resurser utan att de interagerar på ett oönskat sätt.

På liknande sätt förslog Saltzer and Schroeder redan för 30 år sedan att varje program och varje användning av systemet skulle utföra sina uppgifter med så begränsade privilegier som möjligt.

Ett angreppssätt som baseras på en blandning av de två koncepten garanterar att varje aktiv och exekverbar enhet (varje ”subjekt”) har tillräckliga privilegier och resurser för att utföra sin uppgift – men inte mer.

DESSA PRINCIPER är inte nya. De finns i hårdvaruassisterad virtualisering (Intel VT, ARMv7 & v8, NXP Virtualization Extensions, MIPS Virtualization, med mera) som har allt som krävs men har tagit bort den overhead som kommer med den praktiska implementationen, genom att flytta den från mjukvara till hårdvara.

Kärnan som separerar dem kombinerar principerna med lägsta möjliga privilegium och separerade kärnor med hårdvarubaserad virtualisering så att varje teoretiskt ”subjekt” implementeras som en virtuell maskin. I motsats till en hypervisor som Type 1 KVM med 19,5 miljoner rader källkod har en lösning med en separerande kärna bara 25 000 rader kod, vilket är en mycket bättre resurs att dela. ▶



Figur 4. Virtuella maskiner med så låga privilegier som möjligt hanterar datalagring och nätverksfunktionalitet i en separerande kärna.

Principen med lägsta möjliga prioritet uppfylls genom att försäkra sig om att de separerade kärnorna inte innehåller något som skulle kunna användas från användarsidan inklusive drivrutiner, monitorer för virtuella maskiner och kritiska I/O-stackar.

Det finns ingen kompromiss i hypervisorns funktionalitet när varje virtuell maskin får sitt eget virtuella moderkort som innehåller systemets resurser som tilldelats av systemarkitekten vid konfigureringen. Varje operativsystem som stöds av den underliggande hårdvaran kan köras på en separerande hypervisor.

DET ÄR VÄRT ATT NOTERA att principen om lägsta möjliga privilegier inte är unik för separerade kärnor. Tekniken har rötterna i en era från tiden före hårdvaruvirtualisering när det bara fanns två nivåer för prioritet som arkitekter till mikrokärnor kunde använda (administratör och användare) och många använde dem på ett förnuftigt sätt för att köra sårbar kod som att låta drivrutinerna köras på användarens nivå och inte på övervakarens.

När hårdvaruvirtualisering blev verklighet kom samtidigt möjligheten till olika nivåer för monitorer till virtuella maskiner (VMM) och med det ett dilemma för arkitekter som arbetade med mikrokärnor. Om privilegier för en högre nivå ignorerades skapades en sårbarhet och en stor attackyta som inte fanns med i beräkningarna. Men om privilegier för äldre kod för övervakning följde med till den nya nivån skulle den ta med sig mycket kod som inte var nödvändig för funktionen.

DESIGN AV MIKROKÄRNOR har utvecklats till att hantera virtualisering med tillhörande nivåer. Vissa mikrokärnor kan till och med själva dra nytta av virtualiseringen. Det är dock oundvikligt att en sådan utveckling leder till kompromisser där delar av koden körs med högre privilegier än vad som föreskrivs av den lägsta nivån. För att en säkerhetskritisk separation ska minimera attackytan måste den garantera att varje programkomponent har så låga privilegier som möjligt i den här nya världen med hårdvaruvirtualisering.

I fordonsvärlden kan separationskärnor mycket väl baseras på väl underbyggda principer men de är bara användbara om de kan appliceras så att de ger den säkerhetsmiljö som behövs för att garantera att fordonet inte är sårbart för cyberattacker.

Det här måste sättas i relation till existerande infrastruktur som till största delen består av det öppna Internet. Ansvaret för säkerheten landar därmed på gatewayen och i det här fallet systemet i fordonet.

Det är också viktigt att den höga kostnaden för att bygga och serva olika fordonsnät måste undvikas.

Figur 4 visar hur ett sådant system kan designas. Nätverket och datahanteringen implementeras som små virtuella maskiner med lägsta privilegier samtidigt som de är snålt kodade och körs som ”bare metal”-applikationer för att minimera fotavtrycket och därmed sårbarheten.

APPARNA KÖRS i tre virtuella maskiner. Eventuellt är de mappade till individuella kärnor i en flerkärnig processor där det finns hård realtidskoppling mellan dessa.

Kryptonycklar garanterar integriteten i data hos de tre medan separationskärnan ger den underliggande garantin att systemets kodbas är minimal.

Resultatet är en robust lösning som ger ett motståndskraftigt applikationsgränssnitt som hindrar skadlig programvara att påverka den virtuella mjukvaruarkitekturen. Den säkrar integriteten hos kritiska applikationer och skyddar dem från att manipuleras av andra applikationer. Kostnaden minimeras och trots det garanteras applikationens säkerhet av en enda nätverksstruktur. Det går att visa att fordonsapplikationerna är korrekta och att krypteringen i nätverket inte kan kringgås.

Att koppla lägsta möjliga privilegier med principen om separerade kärnor och hårdvaruvirtualisering ger en optimal lösning för utmaningen att säkra den uppkopplade bilen. ■