OTP vs. smartkort

Olika typer av engångskoder (OTP) och kryptografiska smartkort är de i dag dominerande metoderna för tvåfaktorsautentisering.

Smartkort och OTP är olika bra på att bemöta olika typer av attacker. De båda metoderna bjuder också på olika svårigheter när det gäller provisionering och användning av slutanvändare. Valet av lösning beror också på skillnader i komplexitet när det gäller driftsättning på den plattform man önskar skydda.

Under hösten 2011 har Linköpings universitet deltagit i ett Swami-projekt kring tvåfaktorsautentisering. I detta projekt har kryptografiska smartkort av typen Gemalto .NET samt en OTP-lösning från Mideye AB utnyttjats. Smartkorten har försetts med egenutfärdade certifikat från vår Windows AD PKI. OTP-lösningen är baserad på dels på engångskoder levererade via SMS, dels på engångskoder genererade genom användning av dosa. Både formerna av engångskoder är tidsberoende.

Kryptografiska smartkort

Kryptografiska smartkort tillhandahåller en skyddad lagringsplats för lagring av privata nycklar samt publika nycklar och certifikat relaterade till den privata nyckeln. Smartkort kan också användas för lagring av symmetriska nycklar samt generella binära dataobjekt. Skyddet som smartkort tillhandahåller går i princip ut på att göra det omöjligt (eller i vart fall mycket svårt) att extrahera hemliga nycklar. Efter inloggning med PIN-kod tillhandahåller dock kortet möjlighet att utföra operationer såsom signering och dekryptering. På detta sätt skall det alltså krävas att en användare har det fysiska kortet i sin besittning för att exempelvis kunna genomföra en inloggning.

Användning av smartkort kräver förstås att kortet ansluts till den dator från vilken kortet skall användas. Därför måste datorn i fråga förses med drivrutin för kort och kortläsare, att identifiera fungerande drivrutiner på olika plattformar kan vara en utmaning. En faktor att beakta när det gäller hotbild är att ett kort kan missbrukas från en kompromitterad dator om en attackerare har kommit över PIN-koden till kortet (vilket exempelvis kan ske genom tangentbordssniffning).

Smartkort från Gemalto .NET

Vi har tidigare tittat på kort från olika leverantörer men inom detta projekt fokuserat på kort av typen Gemalto .NET. Dessa kort har valts utifrån att det varit möjligt att få tag på fungerande drivrutiner för både Windows och Linux. Se kommentar om drivrutiner för Mac OS X under förutsättningar.

I vissa tillämpningar kan det också vara en fördel att relativt många nyckelpar (upp till 15 stycken) kan samlas på ett och samma kort.

Förutsättningar

- För användning av smartkort för inloggning i Windows och mot Remote Desktop krävs en PKI-infrastruktur för AD. Notera att CA-datorer i PKI:n är mycket skyddsvärda då övertagande av en CA i princip innebär ett övertagande av hela det AD som CA:n ansvarar för.
- De klientdatorer som skall användas måste ha fysika förutsättningar för användning av smartkort. Det vill säga antingen inbyggd kortläsare eller USB-port för anslutning av extern läsare. Detta innebär att kryptografiska smartkort inte kan användas med ett flertal mobila tillämpningar.
- På de klientdatorer som skall användas krävs drivrutiner för kortläsare. För såväl Windows, Mac OS X och Linux är det tämligen oproblematiskt att få igång drivrutiner för kortläsare. Man bör välja en CCID-kompatibel läsare för att undvika problem (CCID står för Chip/Smart Card Interface Devices och är en standard för smartkortsläsare).
- På de klientdatorer som skall användas krävs även drivrutiner för själva smartkorten. För Gemalto:s .net-kort finns drivrutiner för Windows XP, Vista och Windows 7 samt för Mac OS X 10.6 tillgängliga på Gemaltos webbsidor. På begäran kan man också få källkod till drivrutin för Unix /Linux som också bör kunna användas på Mac OS X. Vårt intryck är att drivrutiner för Windows och Linux fungerar mycket bra. Den färdigkompilerade drivrutinen för Mac OS X har vi dock inte fått att fungera (på Mac OS 10.6.8). Vi har dock framgångsrikt använt en tidigare version av denna drivrutin.

Provisionering

Vid användning av korten i PKI-tillämpningar måste ett certifikat utfärdas av en betrodd utgivare (exempelvis en intern CA). Det finns ett antal kommersiella verktyg för att underlätta hantering av detta. En utvärdering av sådana verktyg har dock inte ingått i detta projekt. Vi har utnyttjat vår befintliga AD PKI och har använt Microsofts CertMgr som medföljer Windows för att utfärda inloggningscertifikat och provisionera korten med dessa.

Smartkorten kan också användas för exempelvis ssh, i dessa fall kan användare själva provisionera sitt smartkort genom att generera ett nyckelpar med kommandoradsverktyg som exempelvis pcks11-tool.

Erfarenheter av användning med Remote Desktop och fildelning

Användning har med framgång testats för användare som använder Windows XP och Windows 7 för anslutning till en Remote Desktop Farm som kräver inloggning med smartkort. Det pågår också tester för utdelning av fileshares som kräver smartkort.

För att stänga ute användare som inte använder smartkort krävs, vid inloggning med Remote Desktop till aktuella servrar, medlemskap i en särskild säkerhetsgrupp. Användare blir enbart medlemmar i denna grupp om de loggar in med smartkort.

Vi har inte testat Remote desktop eller fildelning med smartkortsinloggning på Mac OS X eller Linux.

Erfarenheter av användning med SSH

Ett mindre antal användare använder smartkort för användning av ssh. Användning av smartkort för ssh kräver ingen särskild konfiguration på serversidan (annat än normalt förfarande med upplägg av auktoriserade nycklar).

På klientsidan är erfarenheterna blandade. Tekniskt så fungerar användning med openssh på Linux överlag tillfredställande. Bland nackdelarna kan noteras att varje inloggning tar ett par sekunder då nyckeln verifieras på kortet. Vidare så fungerar hantering av smartkort med ssh-agent otillfredsställande då agenten får problem när kortet avlägsnas. I dessa lägen krävs en omstart av agenten för att den skall kunna kommunicera med smartkortet på nytt. Detta problem kan vara relaterat till kortets drivrutin.

Användare upplever också att det är lätt att glömma kvar kortet i kortläsaren när man lämnar arbetsplatsen.

På Windows har PuttySC, en fork av Putty med stöd för smartkort, använts. På Windows XP fungerar PuttySC hyfsat medan den i vissa lägen kraschar på Windows 7 (i synnerhet upplevs detta om smartkortsinloggning begärs utan att kortet är tillgängligt).

Erfarenheter av användning med Cisco:s Any Connect

Försök har gjorts att konfigurera en VPN baserad på Cisco ASA för användning av smartkort. Resultatet var dock inte tillfredställande. Så länge smartkort krävdes i en avskild testmiljö så fungerade lösningen relativt bra. När smartkortsinloggningen sedan driftsattes på en VPN som servar ett flertal olika grupper (varav vissa inte har behov av tvåfaktorsautentisering) så slutade detta med att samtliga användare avkrävdes smartkort innan de fick upp inloggningsrutan där användarna kunde välja den inloggning som inte kräver smartkort. Vi noterade också svårheter för VPN-klienten Any Connect att korrekt identifiera isatt smartkort vid upprepad in- och utmatning av själva kortet.

Efter ett antal arbetsdagar ägnade åt problematiken ovan avbröts försöken med smartkortsautentisering för Any Connect.

OTP

Engångskoder bygger i allmänhet på en delad nyckel mellan autentiseringsserver och en dedikerad hårdvara som användaren har i sin besittning (exempelvis bankdosa). Genom att kombinera nyckeln med en räknare som hålls synkroniserad mellan dosa och autentiseringsserver genereras så kallade händelsebaserade engångskoder. Om nyckeln i stället (eller dessutom) kombineras med en klocka så åstadkoms tidsbaserade engångskoder. För generering av OTP kan den delade nyckeln kombineras med ett värde (challenge) som autentiseringsservern ber användaren mata in i sin dosa för att signera exempelvis en banktransaktion. Engångskoder kan också baseras på leverans *out of band* (via ett annat medium än det som autentiseringen avser) till exempel genom SMS.

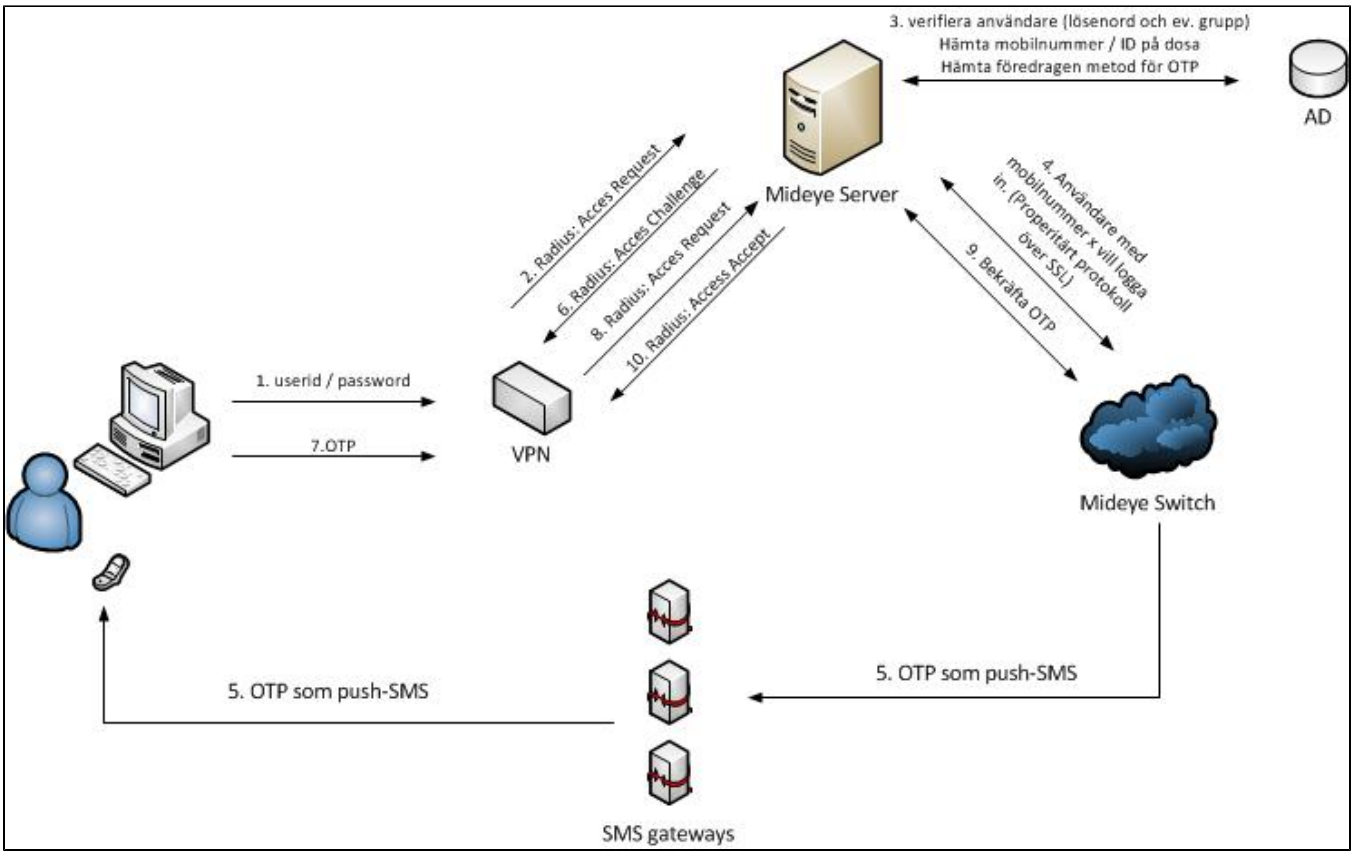
Tids- och challengebaserade engångskoder ger ett något starkare skydd än rent händelsebaserade engångskoder då de inte på förhand kan noteras av en attackerare som tillfälligt fått tillgång till en OTP-dosa.

Mideye

Det system som vi har utvärderat kommer från Mideye AB. Systemet bygger på att *Mideye server* installeras lokalt. System som vill utnyttja tvåfaktorsautentisering kommunicerar med Mideye server genom Radius. Det finns enligt uppgift också ett properitärt API som kan användas i stället för Radius; detta har dock inte undersökts vidare inom projektet.

Mideye server autentiserar användarens statiska lösenord mot lärosätets lokala användarkatalog (AD, LDAP, lokal databas etc.) och läser ur katalogen också ut om användare föredrar inloggning med SMS eller OTP-dosa samt hämtar mobilnummer eller ID för OTP-dosa. För leverans av SMS och verifiering av engångskoder kommunicerar Mideye server med *Mideye switch* som drivs externt av leverantören.

Arbetsflödet för en lyckad inloggning på en VPN med kodleverans via SMS inloggning illustreras något sammanfattat med nedanstående bild



Systemet kan ur användarkatalogen läsa ut om SMS eller OTP-dosa skall användas som default för en viss användare. En användare som vanligtvis utnyttjar SMS-leverans kan tillfälligt begära att autentisera med OTP-dosa genom att lägga till "@TOKEN" efter sitt användarid. På motsvarande sätt kan en användare som normalt använder dosa lägga till "@MOBILE" för att få SMS-kod.

Dosorna som används köps genom Mideye AB. Varje dosa har ett unikt ID som matas in i användarkatalogen. Den faktiska nyckeln i dosan hanteras inte av oss som kund utan det är Mideye switch som bekräftar huruvida en kod är giltig eller ej. Dosorna bygger på både räknare och klocka. Aktuellt värde för räknare och klocka kan enkelt läsas ur dosan för omsynkronisering om till exempel för många koder tryckts fram utan att användas.

Förutsättningar

För att komma igång med OTP från Mideye krävs (förutom avtal med Mideye AB) följande:

- Server för Mideye server (Linux eller Windows)
- Databas (MS SQL eller Mysql)
- Användarkatalog samt systemkonto som låter Mideye server läsa relevanta fält i användarkatalogen (alternativt kan Mideye servers lokala databas användas)
- Eventuellt dosor (köps genom Mideye AB)

Eftersom OTP-dosan och mobilen är helt externa så finns inga särskilda krav på användarnas datorer. Det går lika bra att använda lösningen på en Mac som på en iPad. System som skall interagera med lösningen måste kunna hantera Radius challenge response.

Provisionering

Om användarkatalogen redan har uppgift om användarnas mobilnummer krävs ingen särskild provisionering mer än att sätta upp behörighetskontroll för respektive system, Mideye stödjer exempelvis användandet av AD-grupper. För att använda dosor så måste varje dosas serienummer kopplas till respektive användare, detta är ett manuellt arbete som kan underlättas något genom avläsning av serienummer med streckkodsläsare.

Erfarenheter av användning med Cisco:s Any Connect

Vi har testat Mideye på en grupp om 16 tekniker i deras användning av klientbaserad VPN med Cisco:s Any Connect. Erfarenheterna har varit mycket positiva. I synnerhet uppskattas möjligheten att själv välja om SMS eller dosa skall användas vid ett visst tillfälle. Driftsättningen på Cisco ASA skiljer sig inte mot driftsättning av autentisering mot någon annan Radius-server.