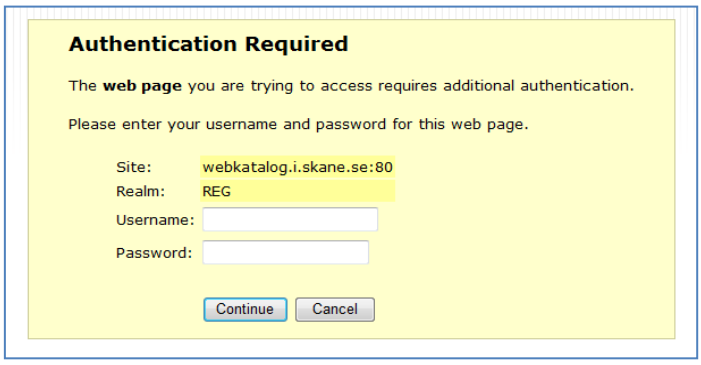
Hur certifikat ”enablar” man en webbapplikation?

Här ser vi ett inte helt ovanligt webbformulär för inloggning klippt från en webbsida. Vi önskar alltså ersätt detta formulär med en certifikatinloggning utan att behöva riva upp himmel och jord.

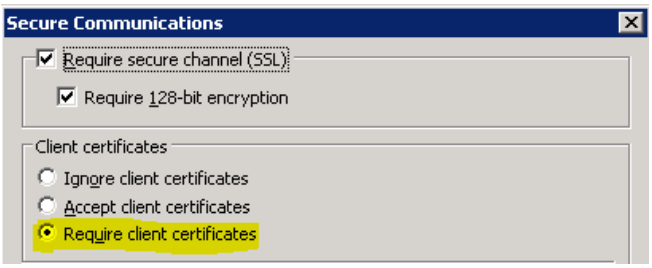


Figur 1. Inloggning till webb applikation.

Steg 1. I detta tänkta exempel står en IIS eller Apache med en .Net eller Tomcat-baserad applikationslogik och en databas:

Index	UserID	Password
000001	jonoho1	[hash of password]
000002	jenalm1	[hash of password]
000003	andmos1	[hash of password]

Steg 2. Konfigurera webbservern så att den kräver clientcertifikat för att upprätta en SSL-session, se figur x. Glöm inte att addera de rotcertifikat som ska vara betrodda.



Figur 2. Konfiguration av webb server.

Steg 3. Testa uppkopplingen med en webbläsare och det aktuella smarta kortet (SITHS). Lyckas inloggningen? Om du använder ett Net iD-paket med SSO aktiverat kan du först logga på en annan site (e-Dos) och där ange PIN. Stäng webbläsaren, öppna den igen och surfa till din nykonfigurerade site, ingen PIN-kod behöver då anges.

Steg 4. Har man kommit såhär långt är det dags att modifiera webbapplikationens inloggningslogik. Inget formulär behöver visas utan man styr användarna direkt till applikationens startsida. Men här uppstår ett problem, certifikaten lär knappast innehålla *jonoho1*, *jenalm1* och *andmos1*. Inte heller kommer något lösenord att lämnas till applikationen. Vill man i detta läge slippa integrera mot AD eller annan datakälla utan tills vidare hålla fast vid den bakomliggande kontodatabasen nödgas man skapa en egen mappningslogik baserat på inloggningen via certifikat och SSL. Då bestämmer man sig först vilket fält i klientcertifikaten som ska användas som ID-begrepp. Använder man HCC på SITHS-kort så är troligen HSA-ID det man vill använda. HSA-ID är alltid unikt inom SITHS precis som personnummer är unikt för privatpersoner i Sverige. Tre exempel på Subject/Ämne:

SERIALNUMBER = SE162321000255-154453
G = Jonas
SN = Öholm
CN = Jonas Öholm
O = Region Skåne
L = Skåne Län
C = se

E = jonas.oholm@secmaker.com
SERIALNUMBER = SE165565968202-3PF0
G = Jonas
SN = Öholm
CN = Jonas Öholm
OU = SecMaker
OU = Samarbetspartners
O = Carelink AB
C = se

T = Konsult
E = jonas.oholm@secmaker.com

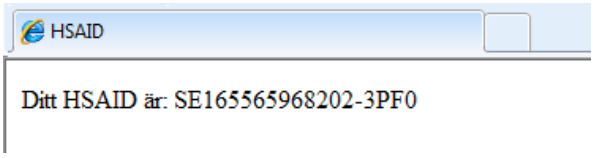
SERIALNUMBER = SE2321000016-5G74
G = Jonas
SN = Öholm
CN = Jonas Öholm
O = Stockholms Läns Landsting
L = Stockholms län
C = se

I alla lägen hittas HSA-ID på samma ställe i certifikaten. Det gör det enkelt att parsa fram HSA-ID vid inloggningen och skicka detta mot bakomliggande logik. En enkel konverteringstabell är förstås inte helt optimalt, men har man bråttom så kan det kanske tillåtas att passera i väntan på en större ”renovering”, t.ex. enligt följande:

Index	HSA-ID	UserID	Password
000001	SE165565968202-3PF0	jonoho1	
000002	SE165565968202-3PDX	jenalm1	
000003	SE165565968202-3PDW	andmos1	

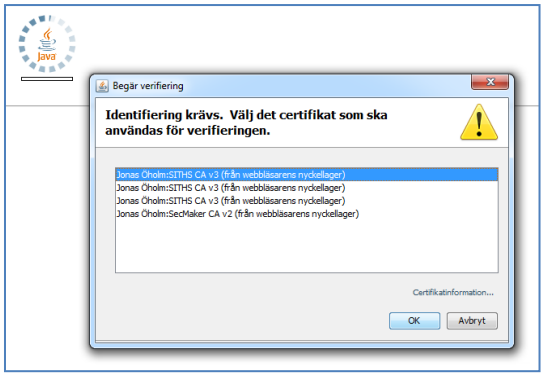
Att parsa fram HSA-ID kan göras på flera sätt men ett enkelt ASP-script som nyttjar inbyggd IIS-funktionalitet kan se ut såhär:

```
1 <html>
2 <head>
3 <title>HSAID</title>
4 </head>
5 <p>Ditt HSAID är: <%=Request.ClientCertificate("Subject2.5.4.5")%></p>
6 </body>
7 </html>
```



Dvs. lyckas SSL-inloggningen och HSA-ID matchar databasen så har man loggat på i god ordning och SSO-känslan kommer på köpet.

OBS! En liten varning rörande laddningen av Java Applets från sidor som kräver klientcertifikat. I JRE 1.3 fungerade det som så att appleten laddades över befintlig, av webbläsaren framförhandlad SSL-session. Gör man samma sak med en senare JRE så vill appleten förhandla fram en egen session med en egen och synnerligen illa designad certifikatvalsdialog. Det går dock att lösa med en ”intrikad” cookie-hantering.



Figur 3. Cookie hantering.