UNI•C

Notat

03.06.05 SLI/sli

Vedrørende:	Teknisk beskrivelse af UNI-C Single Sign-On
Fordeling:	
Skrevet af:	Steen Lindén
Version nr.:	3. juni 2005

Dette notat beskriver, hvorledes eksterne systemer kan integreres i Single Sign-On (SSO) infrastrukturen, der anvendes hos UNI-C på web-applikationer i den danske undervisningssektor.

Som baggrundsforståelse indledes der med et afsnit om den Single Sign-On model, der anvendes *internt* hos UNI-C. Derefter beskrives hvorledes *eksterne* systemer kan indlejres i modellen.

Pubcookie single sign-on

UNI-C anvender Pubcookie fra University of Washington til implementering af Single Sign-On (http://www.pubcookie.org). Pubcookie er en del af Internet2 WebISO projektet (http://middleware.internet2.edu/webiso/)

Pubcookie er baseret på ”secure cookies” og løsningen er derfor begrænset til web-servere, der benytter SSL (https://) inden for et enkelt DNS domæne (p.t. *emu.dk*).

Pubcookie består af en standalone login-server, der er integreret med UNI-Cs brugerdatabase (HUGO), der indeholder omkring 500.000 ansatte og studerende i den danske undervisningssektor, og plug-in autentificeringsmoduler til web-serverne Apache og Microsoft IIS.

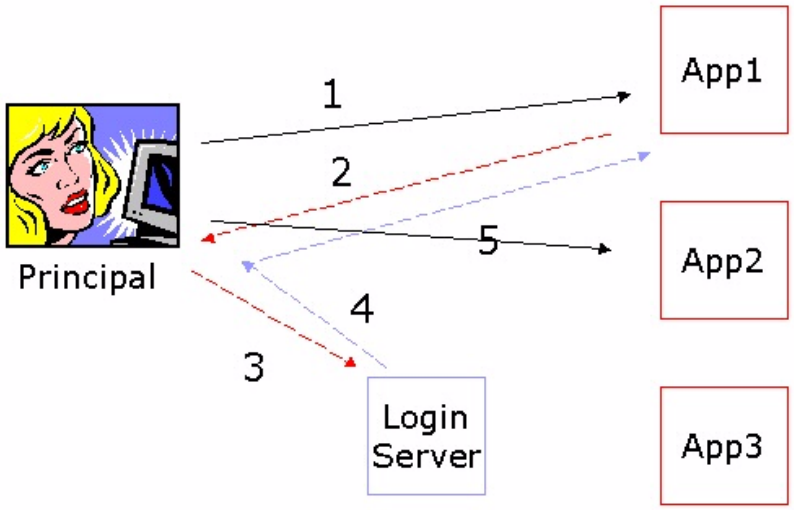
Som udgangspunkt tilbyder Pubcookie kun autentificering, men UNI-C har lokalt tilføjet check af en enkelt ”servicekode” til simpel adgangskontrol, så simple anvendelser ikke nødvendigvis behøver at kode adgangskontrollen selv.

Pubcookie-modellen

Pubcookie benytter en sikkerhedsmodel, der i høj grad minder om Kerberos’ model. Den illustreres bedst ved et eksempel: (Numrene i parentes refererer til request/reply-pilene på tegningen på næste side.)

1. Brugerens browser kontakter en applikation (App1), der er beskyttet af SSO (pil 1).
2. SSO autentificeringsmodulet checker om brugeren har en billet (session-cookie) til App1.
3. Dette er ikke tilfældet initielt, så browseren omdirigeres til login-serveren. (pil 2 og pil 3)
4. En Login-formular præsenteres for brugeren, som indtaster brugernavn og password, der checkes mod HUGO brugerdatabasen.

5. Login-serveren udsteder en login-cookie, som giver fremtidig adgang til login-serveren selv uden fornyet autentificering. Login-billetten er gyldig i 8 timer eller indtil browseren lukkes.
6. Login-serveren udsteder desuden en kortlivet grant-cookie, som autentificerer brugeren i forhold til App1.
7. Endelig dirigeres browseren tilbage til App1 af Login-serveren. (pil 4)
8. SSO autentificeringsmodulet, som beskytter App1, inspicerer grant-cookie og udsteder en session-cookie til sig selv, såfremt grant-cookie er OK.
9. Resten af sessionen kan brugeren anvende applikationen App1 uden fornyet login.
10. Når andre SSO-beskyttede applikationer kontaktes i løbet af sessionen (pil 5), sker omdirigeringen via login-serveren som beskrevet ovenfor for App1. I disse tilfælde vil der dog ikke være behov for at genindtaste brugernavn og password, da login-serveren jo ved første kontakt har udstedt en login-cookie til sig selv. Brugeren bemærker således ikke de efterfølgende round-trips til login-serveren, hvilket opleves som Single Sign-On.



Integration af applikationer uden for Pubcookie-infrastrukturen

I nogle tilfælde er det ikke muligt eller ønskeligt at benytte Pubcookie-modulerne direkte i forbindelse med en applikation, når denne ønskes integreret i SSO-infrastrukturen.

Eksempler på sådanne tilfælde:

1) Applikationen ønsker ikke at benytte SSL af performance-hensyn.
2) Applikationen ligger i en andet DNS-domæne end SSO-infrastrukturen.
3) Applikationen er agent for brugeren og skal kunne autentificere på brugerens vegne.

4) Applikationen benytter hverken en Apache eller Microsoft IIS web-server.
5) Applikationen ønsker at benytte SSO, som alternativ autentificeringsmulighed (tredjeparts autentificering)

UNI-C har implementeret en metode til integrationen af sådanne applikationer i SSO infrastrukturen. Konceptet er baseret på en dedikeret Pubcookie SSO-beskyttet applikation hos UNI-C, *ssoproxy*, som kommunikerer autentificeringsinformation til eksterne applikationer. I praksis omstiller den eksterne applikation ved login til *ssoproxy*, som redirigerer tilbage med en kort-livet billet indkodet i URL'en. Billetten indeholder brugernavn og tidspunkt for udstedelsen og et fingeraftryk, baseret på en fælles off-line hemmelighed.

Dette tillader de eksterne applikationer at ligge i et hvilket som helst DNS domæne og SSL er ikke længere et krav. Prisen er et lidt lavere sikkerhedsniveau, men det er stadig tilstrækkeligt for de fleste applikationer, der baserer sig på UNI-Cs brugerdatabase.

Igen anskueliggøres konceptet bedst med et eksempel:

1) En brugers browser forbinder til en ekstern applikation, som checker internt om brugeren allerede er logget ind, f.eks. ved at inspicere en privat session cookie.
2) Er brugeren ikke logget ind omdirigeres til *ssoproxy*, der er beskyttet af Pubcookie og brugeren autentificeres i forhold til SSO-infrastrukturen, som beskrevet i sektionen om pubcookie-modellen ovenfor.
3) Efter successfuldt login omdirigerer *ssoproxy* browseren tilbage til applikationen med en URL query string, der rummer et fingeraftryk, som bekræfter brugerens identitet i forhold til den eksterne applikation.
4) Applikationen verificerer fingeraftrykket, checker at det er udstedt inden for en fastsat tidsramme, og logger brugeren ind internt.

Eksterne applikationen er nødt til at administrere deres eget sessionsbegreb for at virke i denne model og være i stand til at håndtere dialogen omkring det URL indkodede fingeraftryk med *ssoproxy*. Der findes på nuværende tidspunkt ingen plug-and-play moduler, der transparent integrerer ssoproxy-modellen direkte i web-serverne.

Som fingeraftryk anvendes den hexadecimale MD5 checksum af en streng sammensat af minimum brugernavn, tidspunkt og en fælles hemmelighed mellem *ssoproxy* og applikationen. Brugernavn og tidspunkt er givne størrelser i kommunikationen. Tidspunktet skal forhindre genanvendelse af fingeraftrykket og den fælles hemmelighed autentificerer *ssoproxy* i forhold til applikationen. Dermed er checksummen i realiteten en signering af brugernavnet. Applikationen har dermed fået bekræftet brugerens identitet og kan efterfølgende aktivere sit eget sessionsbegreb.

Eksempel på URL-indkodet fingeraftryk:

Brugernavn = testuser
Tidsstempel = 20030505125952
Fælles Hemmelighed= abc123

The fingerprint is calculated as follows:

MD5(<Tidsstempel><Fælles Hemmelighed><Brugernavn>) =
MD5(20030505125952abc123testuser) =

5e55280df202c8820a7092746b991088

Det URL-indkodede fingeraftryk bliver dermed:

http://<addr>?timestamp=20030505125952&user=testuser&auth=5e55280df202c8820a7092746b991088

Informationer om institutioner, brugere og servicekoder

Oplysninger om institutioner som er omfattet af UNI-Login kommer væsentligst fra Undervisningsministeriets register over dets institutioner. Oplysninger om øvrige institutioner har UNI-C indhentet. Oplysningerne omfatter: unikt nummer (uvm-nr), institutionens navn, adresse, beliggenhedskommune, telefonnummer, mail-adresse, hjemmeside og institutionstype. Se nærmere på vejviser.emu.dk

Enhver bruger har et unikt brugerID, og ud fra det vil man kunne skaffe oplysninger om følgende: fulde navn, rolle, skole, email-adresse og de rettigheder som den givne producent har tildelt. Rolle kan være lærer eller elev. Elever kan være knyttet til en klasse.

Der opereres endvidere med servicekoder, som er grundelementet i UNI-Cs adgangskontrolsystem. Servicekoderne anvendes primært til angivelse af, om brugeren har abonnement på en given tjeneste, men de kan også være mere finkornede og angive en speciel rolle eller rettighed.

Servicekoderne er normalt knyttet til UNI-Cs CRM-system, men det er også muligt partnere at administrere servicekoder fra egne systemer via en grænseflade til UNI-C.

Til at skaffe disse oplysninger stiller UNI-C en web-service grænseflade til rådighed med WSDL-definitioner. Til administration af rettigheder findes der såvel en we-service som en række andre web-baserede grænseflader.

Individuelle aftaler og tilpasning

Hvilke konkrete grænseflader, der tages i anvendelse i forhold til integrationen af et eksternt system i UNI-Cs SSO-infrastruktur, afhænger i høj grad af den enkelte anvendelse og niveauet af integration. F.eks. kan nogle applikationer kræve et abonnement, mens andre blot ønsker information om en given brugers rolle. Kundeforhold vil i nogle tilfælde være varetaget af UNI-C og i andre af den eksterne organisation. Osv.

Der er rige muligheder for tilpasning i forhold til det konkrete projekt, og basis for den enkelte integration er derfor et møde, hvor de individuelle behov afklares og integrationen aftales nærmere.