Autentifikacija i autorizacija korisnika na jednom mjestu

T. Pavić i L. Jelenković
Fakultet elektrotehnike i računarstva
Adresa: Unska 3, Zagreb, R. Hrvatska
E-mail: tomislav_p@net.hr

Sažetak – Proces autentifikacije iznimno je važan element informacijske sigurnosti, budući da predstavlja prvi korak prijave korisnika u sustav.

***Single Sign-On* (skraćeno SSO) predstavlja proces autentifikacije koji omogućava korisniku predočenje svojih akreditacijskih podataka samo jednom kako bi mogao pristupiti svim dozvoljenim resursima. Nakon što se korisnik autentificira može izvršavati aplikacije za koje je autoriziran. Autentifikacija se temelji na oznakama.**

SSO je tradicionalno smatran sigurnosnom tehnologijom koja obuhvaća dvije grane sigurnosti: autentifikaciju i autorizaciju. Danas se najviše razvija SSO za Web, gdje se osim autentifikacije i autorizacije velika važnost pridaje prezentaciji podataka koja ovisi o autentificiranom principalu.

I. UVOD

Od prvih računala do danas zahtjeva se kontrola pristupa podacima. Moguće je sama računala fizički zaštiti i ograničiti im pristup, ili zahtijevati identifikaciju korisnika pa na temelju toga dozvoliti pristup određenim podacima.

Proces autentifikacije, odnosno provjere korisničkog identiteta, iznimno je važan element informacijske sigurnosti. Budući da predstavlja prvi korak prijave korisnika u sustav, sigurnosni zahtjevi koji se pred njega postavljaju prilično su visoki. Također, osim visoke razine sigurnosti, da bi bio upotrebljiv u praksi, proces autentifikacije mora zadovoljavati i brojne druge zahtjeve (npr. praktičnost, financijska isplativost, jednostavnost održavanja i upravljanja i sl.). Kao primjer slabe isplativosti mogu se navesti biometrijski uređaji koji, usprkos visokoj razini sigurnosti koju nude, još uvijek nisu šire prihvaćeni kao mehanizam autentifikacije. U svakodnevnom poslu postoji potreba za čestim prijavljivanjem na više sustava, pri tome se upotrebljavaju različita korisnička imena i autentifikacijske informacije. Upravo u tom području pomaže uvođenje protokola "autentifikacije i autorizacije korisnika na jednom mjestu" ili "jednostruke autentifikacije". Taj protokol smanjuje ljudske pogreške jer je potrebno pamtiti mnogo manje lozinki i ujedno štedi vrijeme

II. JEDNOSTRUKA PRIJAVA

Jednostruka prijava (poznatije pod engleskim nazivom *Single Sign-On* – u daljnjem tekstu će se koristiti engleski naziv ili skraćeno SSO) predstavlja proces autentifikacije koji omogućava korisniku predočenje svojih akreditacijskih podataka samo jednom kako bi mogao pristupiti različitim resursima (npr. aplikacijama). Nakon što se korisnik autentificira može izvršavati aplikacije za koje je autoriziran.

Drugim riječima, SSO predstavlja dijeljenje autentifikacijskih podataka. Za primjer se može uzeti neko poduzeće, gdje zaposlenik, kako bi izvršavao svoje obveze, mora imati pristup određenim podacima, ovisno o vrsti posla koje obavlja. Očito je poduzeću potreban autentifikacijski mehanizam koji će dodjeljivati uloge radnicima na temelju kojih se određuju dozvole pristupa podacima. Dakle, poduzeće može imati nekoliko aplikacija kojima se pristupa korištenjem korisničkog imena i lozinke. Korisnik jedne aplikacije u nekom trenutku želi koristiti drugu aplikaciju. Postavlja se pitanje kako to omogućiti? Postoje dvije mogućnosti:

- Imena i lozinke se mogu duplicirati i postaviti u bazu na svakoj aplikaciji. Tako pojedina aplikacija zasebno provjerava valjanost imena i lozinke. Očito je da postoji zalihost jer se kod pristupa svakoj aplikaciji potrebno ponovo prijaviti.
- Drugi način je bez zalihosti. Ako se korisnik prijavi na jednoj aplikaciji, tada se njegovi podaci prosljeđuju ostalim aplikacijama. Jedini zahtjev je da si dvije aplikacije međusobno vjeruju.

Uobičajeno je da se *Single Sign-On* modul izdvaja u zasebni dio. Od tuda i naziv "autentifikacija na jednom mjestu". Sve aplikacije vjeruju da će taj modul provjeriti ime i lozinku korisnika. Korištenje SSO-a donosi sljedeće prednosti:

- Povećana produktivnost korisnika. Omogućuje da korisnici unose autentifikacijske podatke na samo jednom mjestu za sve usluge koje koriste u jednom trenutku,
- Autentifikacija je bazirana na jedinstvenoj bazi sigurnosnih podataka,
- Propagiranje sigurnosnih atributa kroz sustav,
- Korisnik pamti samo jedno korisničko ime/lozinku za sve aplikacije,
- Administratori održavaju centralizirani repozitorij korisnika za cijeli sustav,
- Reducirani troškovi uvođenja i održavanja,
- Jednostavnije razvijanje novih aplikacija. SSO pruža jedinstveno autentifikacijsko sučelje.

Programeri uopće ne moraju brinuti o autentifikaciji. Mogu pretpostaviti da je korisnik uspješno autentificran kada dođe zahtjev za aplikacijom i zajedno s njim korisničko ime,
- Ostvarenje SSO rješenja može se provesti u koracima, prvo na jednom dijelu aplikacija da bi onda krenuli na preostali dio. Nije potrebno napraviti ogroman skok.

Problemi koji se javljaju sa SSO rješenjima uključuju sljedeće:

- Promjena postojećih aplikacija. Ostvarenje SSO rješenja može biti kompliciranа, dugotrajna, skupa za promjenu postojećih aplikacija.
- Računala bez nadzora. Na primjer, ako se korisnik uspješno prijavi na SSO poslužitelj, te se udalji od računala i ostavi ga bez nadzora. Iako je to generalno pitanje sigurnosti, u SSO slučaju je posebno opasno jer su svi autorizirani resursi kompromitirani.
- Jedna točka napada. Sve aplikacije koriste usluge jednog poslužitelja koji je glavna meta zlonamjernih korisnika.

Dakle, SSO nije bez mana, ali ipak pruža više od problema koji se javljaju, pa se SSO sve više koristi.

III. ZAHTJEVI KOJI SE POSTAVLJAJU PRED SSO

Različita tumačenja funkcionalnosti SSO sustava dovela su do različitih zahtjeva i načina primjene. Svim tumačenjima je zajedničko to da su funkcionalnost i procjena koristi SSO-a najčešće svedeni na realizaciju samo jednokratne identifikacije korisnika ili korisničkog procesa (npr. na temelju certifikata ili biometrijskih obilježja). Tek potom na realizaciju autentifikacije i autorizacije na sve naknadno spojene sustave. U radu kompleksnog informatičkog okruženja, pred SSO se postavljaju i drugi, ne manje važni zahtjevi:

- središnje upravljanje korisničkim podacima i analiza istih, temeljeno na prethodno definiranim ulogama,
- automatizirano prebacivanje aplikacija (u slučaju prestanka rada) bez prekidanja usluge,
- uspostavljanje središnjeg sustava podataka o reviziji i vremensko praćenje ponašanja korisnika,
- središnji pregled nad sustavima i uslugama te pripravljanje odgovarajućih specifičnih dojava o kvaru i alarmiranje bez dodatnih troškova,
- autentifikacija specifična za aplikacije i uloge (npr. stalno zaposleni/vanjski suradnici, klijenti, dobavljači, pristupi održavanju).

IV. NAČIN RADA SSO SUSTAVA

Danas se u svakodnevnom poslu istovremeno koristi nekoliko sustava (aplikacija). Administratori su zaduženi za upravljanje korisničkim podacima na svakom od sustava kako bi se njima moglo pristupiti uz očuvanje integriteta i sigurnosti.

Gledano kroz povijest, distribuirani sustavi su nastali povezivanjem komponenti koje se ponašaju kao neovisne sigurnosne. Te komponente obuhvaćaju individualne platforme s pridruženim operacijskim sustavima i aplikacijama.Komponente djeluju kao neovisne domene tako da se krajnji korisnik mora predstaviti i autentificirati na svakoj domeni s kojom želi raditi. Ovaj scenarij je prikazan na slici 1.

Korisnik komunicira s primarnom domenom s kojom uspostavlja sjednicu. To je na slici prikazano kvadratom u kojemu piše “Prijava na primarnu domenu” pri čemu se od korisnika zahtjeva predočenje autentifikacijskih podataka, na primjer korisničkog imena i lozinke. Sjednica s primarnom domenom je tipično predstavljena ljuskom operacijskog sustava na korisničkoj strani. S primarne domene korisnik poziva usluge na drugim domenama.

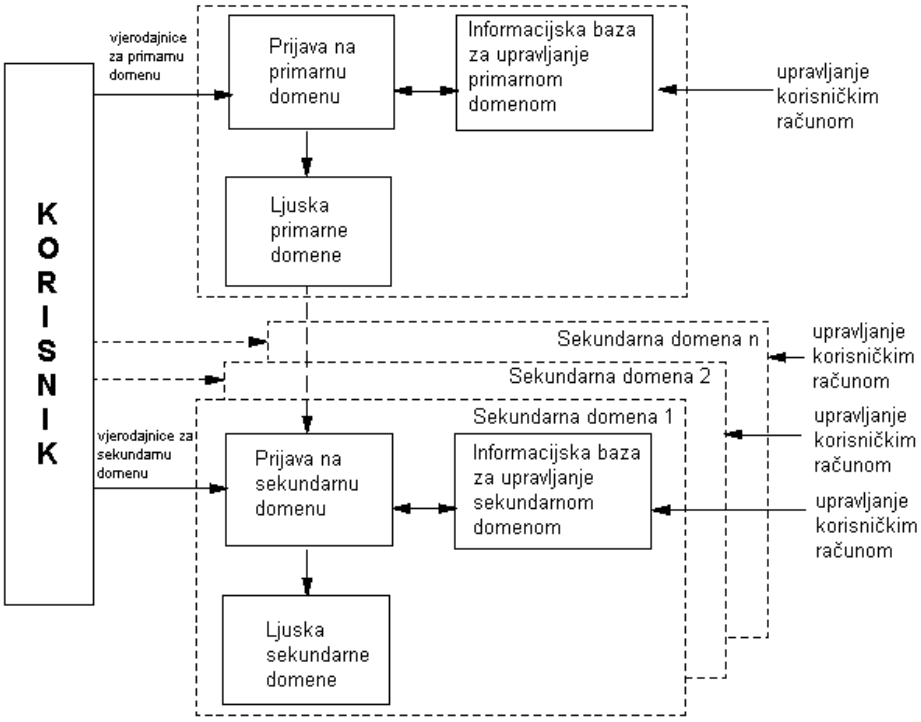
Da bi se pozvala usluga na sekundarnoj domeni korisnik je dužan prijaviti se na toj domeni. To zahtijeva od korisnika predočenje autentifikacijskih podataka primjenjivih na sekundarnu domenu. Korisnik mora voditi odvojene dijaloge sa svakom sekundarnom domenom koju želi koristiti. Sekundarnu domenu tipično predstavlja ljuska operacijskog sustava ili aplikacije. S gledišta upravljanja zahtjeva se neovisno pristupanje svakoj domeni i mogućnost višekorisničkog pristupa. Zahtjevi sigurnosti i iskoristivosti zahtijevaju koordinaciju funkcija za prijavu svih domena. Usluga koja pruža takvu koordinaciju i integraciju ujedno smanjuje i određene troškove, poput:

- vremena potrebnog za prijavljivanje na pojedinim domenama,
- količinu podataka koju korisnik pamti,
- vrijeme potrebno administratoru za dodjeljivanje dozvola i održavanje.

Takva usluga, koja pruža koordinaciju funkcija za prijavu svih domena, naziva se engleskim nazivom *Single Sign-On*. Ovaj pristup prijave, autentifikacije na sustave je prikazan na slici 2.

U *Single Sign-On* metodi sustav koji je dio primarnog SSO-a zadužen je za prikupljanje svih korisničkih podataka potrebnih za autentificiranje korisnika na svaku od sekundarnih domena kojoj bi korisnik mogao tražiti pristup. Ti se podaci mogu iskoristiti na nekoliko načina:

- direktno, korisnički podaci su izravno proslijeđeni sekundarnoj domeni,
- indirektno, korisnički podaci služe za dohvaćanje drugih podataka iz baze, koji se onda koriste prilikom prijave na sekundarnu domenu,
- trenutno uspostavljanje sjednice sa sekundarnom domenom prilikom prijavljivanja na primarnoj domeni,
- podaci su privremeno spremljeni i korišteni na sekundarnoj domeni prilikom zahtijevanja usluge.



Sl. 1. Višestruka prijava korisnika na višestruke sustave

Sa stajališta upravljanja SSO model pruža jedno korisničko sučelje kroz koje se koordinirano i sinkronizirano upravlja sa svim komponentama domena.

Značajni sigurnosni aspekti SSO modela su:

- sekundarna domena mora vjerovati primarnoj domeni da će:
  - provjeriti identitet i autentifikacijske podatke
  - zaštiti autentifikacijske podatke,
- zaštita kod prijenosa podataka između primarne i sekundarne domene

V. SIGURNOST I SSO

SSO je tradicionalno smatran sigurnosnom tehnologijom. Posebice obuhvaća dvije grane sigurnosti: autentifikaciju i autorizaciju. Autentifikacija je proces kojim se utvrđuje korisnikov identitet, dok se autorizacijom utvrđuju dozvoljene radnje korisnika (obično se dodjeljuju uloge koje imaju dozvole). Većina SSO rješenja centralizira autentifikaciju, dok se autorizacijom upravlja na ciljanim resursima (npr. ciljanoj aplikaciji).

Kada se govori o SSO-u i njegovom sigurnosnom aspektu, treba uvijek sagledati malo širu sigurnosnu sliku. Vatrozidi (engl. *firewalls*), virtualne privatne mreže (VPN), enkripcija i druge tehnologije djeluju nezavisno na različitim sigurnosnim rješenjima. Sve tehnologije se mogu povezivati u skladnu cjelinu i međusobno nadopunjavati već kako to zahtjeva korisnik.

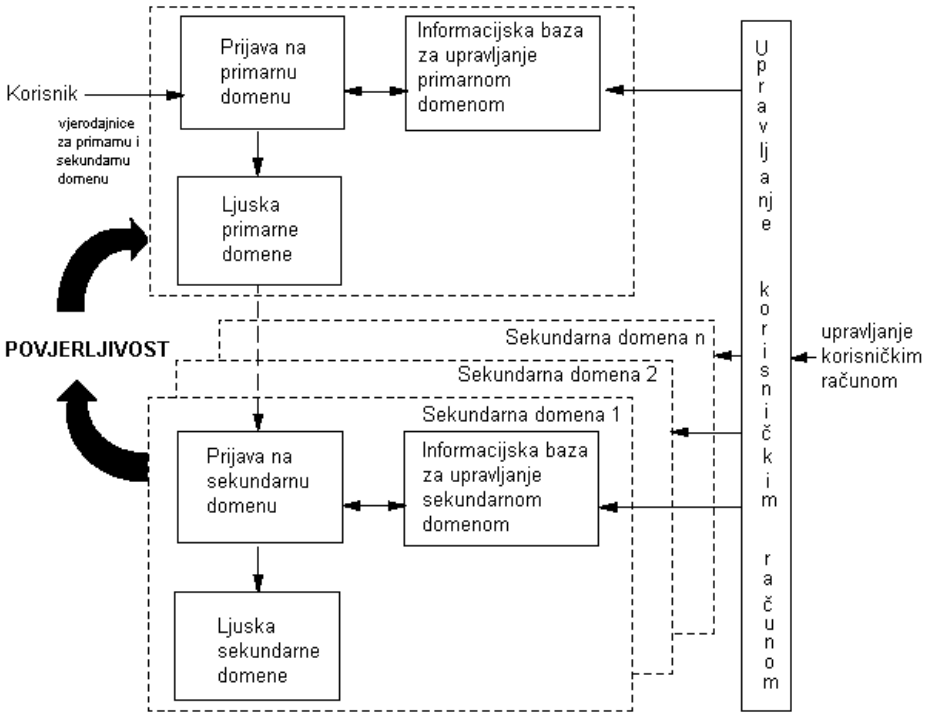
VI. SINKRONIZACIJA LOZINKI

Ukoliko se želi samo smanjiti trošak održavanja zbog zaboravljenih/izgubljenih lozinki, nije nužno koristiti SSO. Sinkronizacija lozinki je manje kompleksna i osigurava istu lozinku za pojedinog korisnika na svim sustavima. No, korisnik se mora prijaviti na svaki sustav koji želi koristiti, ali sada pamti samo jednu lozinku. Sinkronizacija lozinki je bazirana na poslužiteljskim agentima. Kada jedan poslužitelj ustanovi da se lozinka promijenila, tada prosljeđuje promjene drugim poslužiteljima kojima korisnik može pristupiti. Sinkronizacija lozinki korisnicima može biti transparentna. Jedino što korisnik mora znati jest kada se promijeni lozinka na jednom sustavu, onda se mijenja na svim sustavima. Tu se javlja problem ako je lozinka na jednom mjestu komprimirana, svi sustavi su u opasnosti.

Još jednom valja napomenuti da sinkronizacija lozinki nije SSO rješenje. Korisnik se svaki put mora predstavljati kada želi koristiti novu aplikaciju, što zahtjeva dodatni trošak održavanja i vremena.

VII. AUTOMATIZIRANA PRIJAVA

Najjednostavnije SSO rješenje je automatizirana prijava. Svi podaci potrebni za prijavu su spremljeni na posebnom autentifikacijskom poslužitelju. Korisnik se prijavljuje na taj poslužitelj koristeći klijentski program, kojemu onda poslužitelj daje popis svih resursa kojima prijavljeni korisnik može pristupiti. Korisnik izabire željeni resurs i pristupa mu preko autentifikacijskog poslužitelja.



Sl. 2. Jednostruka prijava na višestruke sustave

Automatizirana prijava predstavlja pomak u sigurnosti u odnosu na sinkronizaciju lozinki. Kod sinkronizacije ista lozinka je pohranjena na svakom sustavu, što nije slučaj kod automatizirane prijave. Tako autentifikacijski poslužitelj može održavati bazu različitih lozinki povezanih sa svakim ciljanim resursom, tj. sustavom.

VIII. AUTENTIFIKACIJA TEMELJENA NA OZNAKAMA

Ako se uvođenje SSO-a razmatra isključivo iz razloga smanjenja troškova i povećanja efikasnosti, tada je dovoljno uvesti automatiziranu prijavu. Ali s time se ne dobiva na povećanju sigurnosti resursa. Ostaje moguća direktna prijava na pojedini resurs kao i bez SSO-a. Sofisticiranija rješenja pružaju veću sigurnost cijelog sustava, poboljšavajući mehanizme autentifikacije i mehanizme koji kontroliraju korisničke privilegije.

Napredna autentifikacijska rješenja su bazirana na oznakama (engl. *tokenima*). Jednom kada se korisnik autentificira stvara se jedinstvena i jednom iskoristiva oznaka kojom se korisnik predstavlja resursu za kojeg je autoriziran. Najpoznatiji sustav temeljen na oznakama je Kerberos razvijen na MIT-u. Skoro sva raspoloživa SSO rješenja su bazirana na Kerberosu.

IX. SSO U KONTEKSTU WEB-a

Upravo su u kontekstu Web aplikacija neki elementi, poput portala, sigurnosnih provjera, važni za kvalitetu i troškovno povoljnu proširivost ponuđenih usluga. Sukladno tomu, i SSO sustavi u Web kontekstu imaju posebno značenje kao nadopuna središnjih mehanizama zaštite pristupa. Budući da je Web tehnologija sasvim prikladna za to da se u okviru višeslojne arhitekture i postojeći sustavi i usluge stave na raspolaganje iza zajedničkoga korisničkog i pristupnog sučelja, SSO rješenje u Web kontekstu je isto tako u stanju po potrebi zajamčiti sigurnu autentifikaciju i autorizaciju nad svim sustavima.

Dok se SSO rješenja u pojedinom poduzeću okreću zahtjevima smanjenja troškova i većoj sigurnosti, na Webu su ti zahtjevi u drugom planu. Web je više okrenut korisnicima usluga. Mnoge stranice primjenjuju SSO, a da korisnik toga nije ni svjestan. Komercijalne Web stranice koje zahtijevaju više od jednog prijavljivanja (unosa imena i lozinke) ispituju strpljenje korisnika. Stoga takve stranice nisu komercijalno isplative, jer većina korisnika ih zaobilazi.

Najprihvatljivije ostvarenje SSO-a na Webu je uporabom malih količina informacije koje poslužitelj pohranjuje na klijentsko računalo preko preglednika i kojima kasnije može pristupati. Jedna takva informacija se naziva kolačić (engl. *cookie*, u daljnjem tekstu se koristi engleski naziv). Kod prve prijave na poslužitelj, jedan *cookie* sa zapisanim korisničkim imenim, se pohranjuje na klijentsko računalo. Prilikom svakog sljedećeg zahtjeva, poslužitelj može dohvatiti korisničko ime iz pohranjenog *cookie-a* i odlučiti o daljnjim akcijama. Danas je *cookie* podržan od strane svih internet preglednika, pa je njihova uporaba u ostvarivanju SSO-a velika.

Korištenjem *cookie-a* kao ulaznice u SSO-u, postavlja se pitanje njihove sigurnosti. Uvijek je moguće prisluškivati pakete koje preglednik šalje poslužitelju i uhvatiti *cookie*.

Kako preglednici koriste DNS da bi utvrdili koji *cookie* pripada kojem poslužitelju, moguće je privremeno onesposobiti DNS i prevariti preglednik tako da pošalje *cookie* lažnom poslužitelju. Ako se *cookie* trajno pohranjuje na korisnikov disk, moguće je pročitati ga direktno s diska. No zadnji slučaj nije moguć kod SSO-a, jer se u primjeni *cookie* čuva isključivo u radnoj memoriji (engl. *in-memory cookie*).

Arhitekti sustava u kojima se *cookie* koristi za autentifikaciju trebaju uzimati u obzir mogućnost krađe *cookie-a*. Svaki takav *cookie* treba sadržavati što manje informacija i nije pogodno da pohranjuje korisničko ime i lozinku u čitkom tekstualnom obliku. Ako je moguće, *cookie* bi trebao sadržavati informacije prema kojima sustav može potvrditi da je korisnik autoriziran za njegovo korištenje. Preporučaju se sljedeće informacije:

1. sjednički identifikator (ID)
2. vrijeme i datum kada je *cookie* kreiran
3. vrijeme isteka valjanosti
4. IP adresa preglednika kojemu je *cookie* izdan
5. poruka za potvrdu autentičnosti (engl. *Message Authenticity Check* - MAC)

Ukoliko se *cookie* ukrade neće se moći trajno koristiti zbog ograničenja trajanja valjanosti. Uz to IP adresa točno definira adresu s koje je poslan. MAC poruka potvrđuje da se ostala polja nisu neovlašteno mijenjala. Najčešće se izračunava jednim od hash algoritama, npr. MD5 ili SHA. *Cookie* se može zaštititi kriptiranjem, npr. DES algoritmom. Za iznimno osjetljive aplikacije, preporuča se kriptiranje cijelog komunikacijskog kanala između preglednika i poslužitelja. To se postiže SSL protokolom. *Cookie* će biti kriptiran s ostatkom podataka koji se šalju kroz kanal, pa se ne može presresti bez da se razbije cijeli kanal.

X. CENTRALNI AUTENTIFIKACIJSKI SUSTAV ZA WEB

Centralni autentifikacijski sustav (engl. *Central Authentication Service* – u daljnjem tekstu piše skraćeno CAS) za Web je napravljen da radi s aplikacijama kojima se pristupa isključivo preko Web preglednika. Svaki zahtjev se presreće od strane poslužitelja ili aplikacije, a neautentificirani korisnici se ne propuštaju do traženog resursa.

CAS je autentifikacijski sustav razvijen na Sveučilištu Yale koji pruža pouzdan način autentifikacije korisnika. CAS protokol je dostupan svima, a programski kod se može skinuti s službenih stranica.

Usluge koje pruža CAS su:

- Olakšava SSO na višestrukim Web aplikacijama, kao i jezgru za servise koji nisu nužno bazirani na Webu ali posjeduju grafičko sučelje.
- Omogućava nepouzdanim aplikacijama autentifikaciju korisnika bez pristupa lozinkama.
- Pojednostavnjuje procedure koje aplikacije moraju slijediti da bi autentificirali korisnika.
- Autentifikacija se lokalizira na jednoj Web aplikaciji, čime se olakšava čuvanje i promjena lozinki, bez izmjena ostalih aplikacija.

CAS arhitektura

Centralni autentifikacijski sustav je dizajniran kao samostalna Web aplikacija. Ostvaren je kao nekoliko Java servleta i koristi HTTPS protokol. Pristupa mu se kroz tri URL-a:

a) URL za prijavu,
b) URL za validaciju,
c) opcionalni URL za odjavu.

Korisnik uobičajeno dolazi na jednu od Web aplikacija koje za autnetifikaciju i SSO koriste CAS. U tom slučaju aplikacija preusmjerava korisnika na URL za prijavu. Tom URL-u se može pristupiti i direktno, ako se želi autentificirati bez da se pozove bilo koja aplikacija. URL za prijavu obrađuje "primarnu" autentifikaciju. Zapravo, zahtijeva od korisnika unos imena i lozinke, te provjerava njihovu valjanost. CAS sustav se može konfigurirati da vrši provjeru korisničkog imena i lozine pomoću bilo kojeg poznatog sustava kao što su:

1. obična datoteka s lozinkama
2. kriptirana datoteka s lozinkama
3. baza podataka
4. LDAP server
5. ostalo

Kako bi se omogućila automatska ponovna autentifikacija, CAS šalje pregledniku *in-memory cookie* (koji se briše čim se preglednik ugasi). Taj *cookie* se naziva *cookie* za dodjelu ulaznica (engl. *ticket-granting cookie*), i identificira korisnika koji se već uspješno prijavio. Ako se on izostavi, korisnik će morati unijeti ime i lozinku svaki put kada ga aplikacija preusmjeri na CAS. Ako se ne izostavi, dobiva se *Single Sign-On* za više Web aplikacija. Odnosno, korisnik unosi ime i lozinku samo jednom i dobiva pristup svim aplikacijama koje koriste CAS. Prilikom odjave, *cookie* za dodjelu ulaznica se uništava čime je potrebna ponovna prijava (odjava se izvršava odlaskom na URL za odjavu – *logout URL*).

Prilikom primarne autentifikacije, CAS pamti i aplikaciju kojoj je korisnik htio pristupiti i s koje je preusmjeren. To je moguće jer su aplikacije prilikom preusmjeravanja korisnika dužne predati i svoj identifikator. Ako autentifikacija uspije, CAS kreira ulaznicu – veliki slučajni broj. Tada se ta ulaznica povezuje s korisnikom koji se uspješno autentificirao i aplikacijom kojoj želi pristupiti. Drugim riječima, ako je korisnik K preusmjeren s aplikacije A, CAS kreira ulaznicu T kojom se dozvoljava korisniku K da uđe u aplikaciju A. Kreirana ulaznica je namijenjena samo za jednu uporabu – jednom iskoristiva, valjana je samo za korisnika K i samo za aplikaciju A.

Po završetku primarne autentifikacije, CAS usmjerava korisnika na aplikaciju s koje je došao, odnosno kojoj je htio pristupiti. To je moguće jer već spomenuti identifikator aplikacije (engl. *serviceID*) djeluje kao URL

(engl. *callback URL*). To jest, identifikator mora predstavljati URL same aplikacije. CAS usmjerava korisnika na taj URL, dodajući kreiranu ulaznicu kao jedan parametar.

XI. FEDERATIVNI PRISTUP SSO-U

Noviji pristup ostvarenja SSO-a za Web aplikacije je federativni pristup, odnosno federacija (savez) Web aplikacija (engl. *federation*). Federacija koristi standardizirane protokole kako bi jedna aplikacija dokazala identitet korisnika drugoj aplikaciji, čime se izbjegava nepotrebna zalihost. Istovremeno svaka aplikacija ne mora znati način ostvarenja autentifikacije i autorizacije u ostalim aplikacijama. Ključ je u standardnim mehanizmima i formatima razmjene korisničkih informacija, odnosno standardu komunikacije. *Security Assertion Markup Language* (SAML) predstavlja takav standard.

SAML je razvijen od strane Organizacije za napredak standarda strukturiranih podataka (Organization for the Advancement of Structured Information Standards – OASIS). Predstavlja prijenos autentifikacijskih i autorizacijskih podataka, te atributa u XML obliku. Omogućava entitetima u komunikaciji stvaranje izjava (o identitetu, atributima i pravima subjekta (principala, korisnika)) namijenjenih drugim entitetima. SAML je fleksibilan i proširljiv protokol dizajniran da ga koriste i mijenjaju drugi protokoli.

SAML i Web Single Sign-On

SAML je najviše u upotrebu upravo u razvijanju SSO-a. Korisnik se prijavljuje na jednu stranicu i nakon toga može pristupati drugim stranicama bez dodatne prijave. SAML omogućava SSO kroz komunikaciju izjavama (engl. *assertions*). Stranica na kojoj se korisnik prijavio, šalje izjavu drugoj stranici koja tada može korisnika propustiti kao da se direktno prijavio.

Autorizacija temeljena na atributima

Slično SSO-u, i ovdje komuniciraju dvije Web stranice samo što se ovog puta prenose drugi podaci – atributi o ulozi koja je korisniku dodijeljena prilikom prijave. Autorizacija temeljena na atributima se koristi kada nisu potrebni podaci o tome kada i gdje se korisnik prijavio, ili se ne smiju slati preko mreže iz sigurnosnih razloga.

Sigurnost Web servisa

SAML izjave se mogu koristiti unutar SOAP poruka kako bi se prenijeli sigurnosni podaci i podaci o identitetu između sudionika.

XII. ZAKLJUČAK

SSO je tradicionalno smatran sigurnosnom tehnologijom, koja ima osnovnu zadaću autentifikaciju i autorizaciju korisnika. U kombinaciji s drugim sigurnosnim tehnologijama pruža mogućnost izrade različitih sigurnosnih rješenja. No danas se više ne smije gledati na SSO samo kroz autorizaciju i autentifikaciju. Zbog sve veće količine informacija dostupnih oko nas, potrebno ih je razvrstati i organizirati na način koji pruža bolju preglednost pojedinom korisniku. Dakle, prikaz podataka se prilagođava pojedincu, prilagođava korisniku prema njegovim željama. SSO se razvija u smjeru da će se aplikacije pouzdati u SSO, da način na koji se podaci prezentiraju ovisi o tome tko ih gleda. To se najviše odnosi na Internet, gdje je pojam personalizacije dugo poznat.

SSO ima svoje prednosti kao što su povećana produktivnost korisnika, jedna sigurnosna baza, lakša administracija, ali i mane poput jedne točke napada, prilagodbe gotovih aplikacija. Ovisno o željama korisnika potrebno je proučiti njihov omjer i ovisno o željama upustiti se u ostvarenje SSO-a. Integracija u gotove aplikacije može biti skup i dugotrajan proces, jer je potrebno mijenjati aplikacije u određenoj mjeri. Mnogo je bolje razvijati SSO paralelno s razvojem novih aplikacija.

LITERATURA

[1] T. Pavić, “Diplomski rad: Autentifikacija i autorizacija korisnika na jednom mjestu”, Fakultet elektrotehnike i računarstva, Zagreb, 2006.

[2] The Open Group: Single Sign-On, http://www.opengroup.org/security/sso/

[3] Central Authentication Service, http://www.ja-sig.org/products/cas/index.html

[4] OASIS Security Services (SAML) TC http://www.oasis-open.org/committees/security/

(Dostupnost literature na Internet stranicama provjerena u siječnju 2007.)