FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITY KOMENSKÉHO
BRATISLAVA

Jednotná autentifikácia používateľov webových aplikácií na UK

Diplomová práca

Peter Kopáč **2007**

Jednotná autentifikácia používateľov webových aplikácií na Univerzite Komenského

DIPLOMOVÁ PRÁCA

Peter Kopáč

UNIVERZITA KOMENSKÉHO

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

KATEDRA INFORMATIKY

Informatika

Vedúci diplomovej práce:

Mgr. Pavol Mederly

BRATISLAVA 2007

Čestne vyhlasujem, že túto diplomovú prácu som vypracoval samostatne len s použitím uvedených zdrojov.

V Bratislave 3.5.2007

Peter Kopáč

Abstrakt

Autor:	Peter Kopáč
Názov práce:	Jednotná autentifikácia používateľov webových aplikácií na UK
Škola:	Univerzia Komenského v Bratislave
Fakulta:	Fakulta matematiky, fyziky a informatiky
Katedra:	Katedra informatiky
Vedúci diplomovej práce:	Mgr. Pavol Mederly
Rok:	2007
Počet strán:	61

Predkladaná diplomová práca sa zaoberá problematikou autentifikácie používateľov webových aplikácií, poskytovaných univerzitou. Obsahuje zhrnutie problematiky, prieskum súčasného stavu oblasti autentifikácie používateľov webových aplikácií a technický návrh na implementáciu riešenia single sign-on v tomto prostredí.

Prínos práce spočíva vo vykonaní prieskumu, analýzy a návrhu riešenia autentifikácie používateľov webových aplikácií pre projekt Jednotného autentifikačného systému UK.

Pri spracovávaní zvolenej témy sme vychádzali z uvedenej literatúry, osobných prieskumov a skúseností, ako aj spolupráce s ostatnými členmi riešiteľského kolektívu projektu.

Kľúčové slová: single sign-on, webové aplikácie, správa používateľskej identity, autentifikácia

Predhovor

Táto práca sa venuje problematike autentifikácie vo sfére webových aplikácií a tomu, ako odľahčiť používateľov, administrátorov aj systémy samotné od záťaže vznikajúcej ako dôsledok opakovaného riešenia tejto problematiky v jednotlivých aplikáciách. Jej cieľom je navrhnúť technické riešenie webového single sign-on, uspokojivo implementovateľné v rámci projektu Jednotného autentifikačného systému pre Univerzitu Komenského (ďalej JAS).

Pri písaní práce som vychádzal zo štúdia dostupnej dokumentácie, ale aj osobných prieskumov situácie na UK, práce s analyzovanými produktami a spolupráce s ostatnými členmi riešiteľského tímu projektu. Prínos práce v rámci projektu JAS pozostáva z prieskumu súčasnej situácie na UK, vytvorenia prehľadu dostupných technológií a ich preskúmania, a navrhnutia konkrétnych technických alternatív riešenia.

Predmet diplomovej práce bol zvolený na základe môjho záujmu o oblasť a zo želania pomôcť odstrániť problémy súčasného stavu, s ktorým som bol ako študent UK neraz konfrontovaný.

Týmto sa chcem poďakovať môjmu diplomovému vedúcemu, Mgr. Pavlovi Mederlymu za cenné pripomienky, rady a trpezlivosť počas celej tvorby práce. Ďalej sa chcem poďakovať administrátorom jednotlivých systémov, ktorých ústretovosť pri prieskume súčasnej situácie napomohla vzniku práce.

Obsah

Zoznam obrázkov a tabuliek

Úvod

Problematika digitálnej identity si vo svete informačných technológii vyžaduje čoraz väčšiu pozornosť. S rastom miery, v ktorej výpočtová technika vstupuje do bežného života ľudí sa zväčšujú aj nároky, ktoré na nich kladie. Bežný používateľ prichádza pri každodennej práci do kontaktu s desiatkami rozličných systémov. Často pritom ide o systémy, ktoré neposkytujú svoje služby každému, ale iba presne vymedzenej skupine používateľov. Zo svojej podstaty musia preto od používateľa vyžadovať dôkaz identity. Každý systém potrebuje časť, ktorá bude túto funkcionalitu zabezpečovať.

Tieto rozličné bezpečnostné časti si vyžadujú nielen pozornosť administrátorov, ktorí musia zabezpečovať ich spoľahlivé a bezproblémové fungovanie, kladú požiadavky aj na používateľov. Nutnosť nejakým spôsobom preukázať svoju identitu systému sa môže zdať triviálna, pri dennodennej práci s mnohými rozličnými systémami sa však nároky rýchlo začínajú hromadiť. Jednotlivé systémy navzájom často vôbec nekomunikujú, preto je na prístup do každého z nich treba samostatné používateľské meno a heslo. Keďže bezpečnosť jednotlivých systémov priamo očakáva, že používateľ bude s každým z týchto hesiel pracovať so zvýšenou opatrnosťou, pri jej zanedbaní sa zvyšuje výskyt zlyhaní bezpečnosti a ich závažnosť. Medzi ďalšie následky tohto hromadenia nárokov na používateľa patrí zníženie používateľského pohodia a v neposlednom rade aj globálny pokles efektivity práce.

Tieto problémy sú v súčasnosti aktuálne aj na UK - ich riešením sa zaoberá projekt Jednotného autentifikačného systému (JAS). Ide o iniciatívu Centra informačných technológií UK, zameranú na zlepšenie podpory používateľov. “Cieľom projektu je vytvoriť systém, ktorý odbremení používateľov od nutnosti pamätania si veľkého množstva prístupových hesiel. Umožní centralizovanú správu používateľov a ich hesiel, s možnosťou delegovania rutinných úloh na fakulty, resp. ďalšie súčasti univerzity.”([1]). Táto práca je súčasťou projektu, zameriava sa na podproblém

1 Mederly. *Jednotná autentifikácia používateľov informačných technológií*. 2006

autentifikácie používateľov webových aplikácií.

V prvej kapitole uvedieme používanú terminológiu. Výhodám a nevýhodám centralizovanej architektúry pre autentifikáciu sa budeme venovať v druhej kapitole. Tretia kapitola poskytne prehľad o súčasnom stave na UK a sformuluje kritériá, ktorými sa pri výbere riešenia budeme riadiť. Štvrtá kapitola uvedie prehľad technológií charakteristických pre problematiku. Piata kapitola porovná alternatívy riešenia. V šiestej, záverečnej kapitole zhrnieme výsledky práce a predostrieme možnosti daľšieho rozpracovania.

1 Terminológia

V tejto časti zavedieme dôležité pojmy, súvisiace s problematikou. Pretože ide o oblasť, v ktorej prebieha aktívny vývoj, je dôležité zadefinovať, ktoré z mnohých významov jednotlivých termínov budeme v práci používať.

1.1 Digitálna identita

Digitálna identita je zbierka údajov o jej vlastníkovi. Môže ísť o rodné číslo, minulé akcie používateľa v systéme či jeho preferencie práce s ním. Systém pracuje s týmito údajmi, keď potrebuje vykonať rozhodnutie, súvisiace s používateľom([2]).

1.2 Autentifikácia

Autentifikácia je proces, v ktorom používateľ systému dokazuje vlastníctvo konkrétnej digitálnej identity. Po úspešnej autentifikácii vzniká prepojenie medzi používateľom (a jeho prácou v systéme) a danou digitálnou identitou. Existujú tri základné spôsoby, ako sa môže používateľ preukázať:

- čo vie (heslo, vedomosť),
- čo má (cookie, fyzický token),
- čo je (biometrický údaj).

V prípade, že sa od používateľa vyžaduje, aby sa autentifikoval viac ako jedným spôsobom súčasne, hovoríme o *multifaktorovej* resp. s*ilnej autentifikácii* ([3]).

2 Windley. *Digital Identity*. 2005
3 Guide to CISSP. *Access Control Concepts* 2007

1.3 Autorizácia

Pod autorizáciou budeme rozumieť proces určenia činností, ktoré má vlastník danej digitálnej identity právo v systéme vykonávať. V priebehu vývoja technológii tejto oblasti postupne vzniklo niekoľko spôsobov riešenia ([2]):

- *individuálne prideľovanie práv* - každá identita má vlastný zoznam povolených činností,
- *prístupové zoznamy* - každá činnosť má vlastný zoznam identít, ktorým je povolané ju vykonávať,
- *kontrola prístupu na základe rol* – každá identita má zoznam rol, ktoré sú jej pridelené. Každá rola má vlastný zoznam povolených činností.

Použitie najnovších riešení (teda kontroly prístupu na základe rol) výrazne znižuje mieru náročnosti autorizačného procesu, ako aj pracovnej náročnosti správy používateľských účtov (viď nižšie).

1.4 Správa používateľských účtov

Pod týmto termínom (v angličtine *provisioning*) sa skrýva skupina činností, súvisiacich so životným cyklom používateľských účtov. V prvom rade je tu zahrnutá problematika ich vytvárania a počiatočného nastavovania. Ďalej ide o udržiavanie životného cyklu identity, teda vykonávanie nutných zmien v jej údajoch či činnostiach povolených jej vlastníkovi. V neposlednom rade termín zahŕňa aj spoľahlivé rušenie digitálnych identít, ktoré už nie sú potrebné či žiadané ([2]).

V kombinácii s vhodne zvoleným modelom autorizácie je možné práve v tejto oblasti ušetriť veľké množstvo zbytočnej námahy a potenciálnych bezpečnostných rizík. Existuje výrazný rozdiel medzi manuálnym odoberaním prístupových práv daného používateľa v každom podsystéme zvlášť, a globálnym odobraním roly, na ktorú už

2 Windley. *Digital Identity*. 2005

vlastník danej identity nemá oprávnenie.

1.5 Federovaná identita

Vzájomnému prepojeniu autentifikácie viacerých organizácií a odovzdávaní si informácie o používateľovi (resp. jeho úspešnom autentifikovaní) sa venuje problematika *federovanej identity* ([4]). Ťažisko problematiky leží v ustanovení dôvery medzi jednotlivými účastníkmi výmeny bez poskytnutia dôverných informácií o používateľovi ([5]). Federovaná identita nie je v súčasnej situácii na UK podstatná, preto sa jej v tejto práci bližšie venovať nebudeme. Po úspešnom zavedení jednotnej autentifikácie však poskytuje potenciál na ďalší rozvoj projektu, napríklad v spolupráci s partnerskými univerzitami.

1.6 Single Sign-On

Single sign-on ("jednotné prihlasovanie") je globálna autentifikačná schéma, pri ktorej je úspešná autentifikácia pri prístupe k jednej službe platná aj pri prístupoch k iným službám. Po prvom úspešnom prihlásení ostatné overenia prebiehajú v ideálnom prípade pre používateľa transparentne; používateľ by vôbec nemal vnímať prítomnosť kontrolného mechanizmu ([5]).

Prvotným cieľom je odľahčiť používateľa od nutnosti pracovať s veľkým množstvom rozličných autentifikačných údajov a tým zvýšiť jeho pracovné pohodlie. Obvyklé metódy dosahovania tohto cieľa poskytujú aj ďalšie výhody, ktorým sa (spolu s nevýhodami) budeme bližšie venovať v druhej kapitole tejto práce.

Single sign-on (ďalej SSO) riešenia sú štandardne zamerané na rozsah jednej organizácie, ktorá má viacero poskytovaných služieb a skupinu používateľov, ktorí ku všetkým z nich pristupujú. Nemá zmysel uvažovať SSO riešenia v prostredí, kde

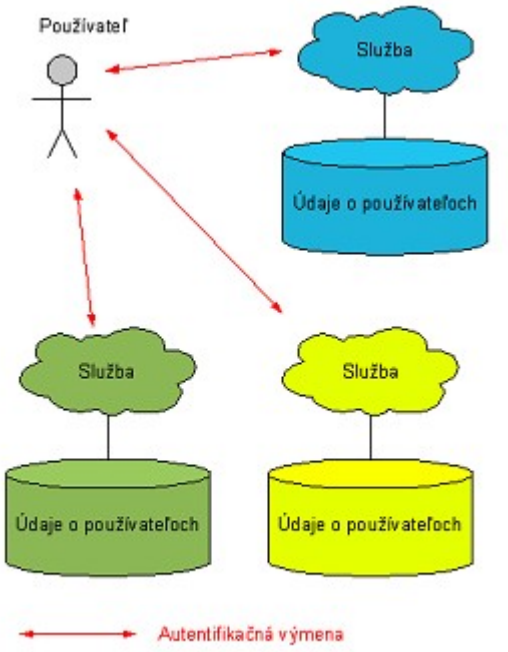
4 Pickard. *Identity Federation.* 2006
5 Milgate. *The Identity Dictionary.* 2006

rozličnú používatelia pracujú iba s veľmi presne vymedzenými službami bez nutnosti používať ostatné – náklady na implementáciu riešenia by totiž prevýšili výsledný zisk.

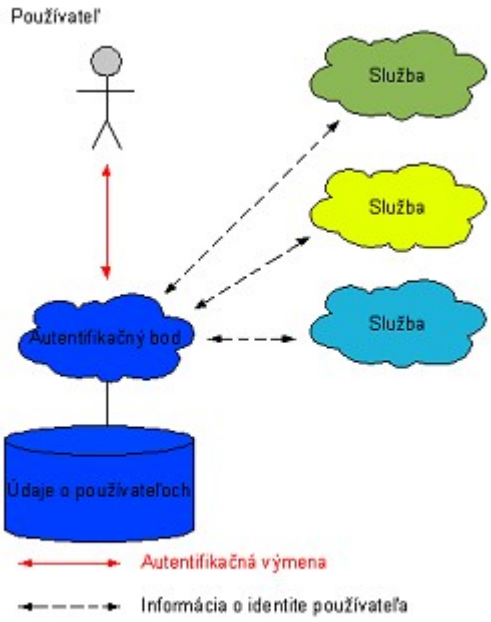
V nasledujúcej kapitole sa bližšie pozrieme na výhody a nevýhody centralizovaného autentifikačného riešenia v porovnaní s obvyklou, decentralizovanou architektúrou.

2 Porovnanie centralizovaného a decentralizovaného riešenia

Táto kapitola sa bude venovať vysvetleniu rozdielov medzi obvyklým prístupom k problematike digitálnej identity a prístupom centralizovaným. Ukážeme výhody a nevýhody oboch, spolu so spôsobom ich fungovania. Pri písaní kapitoly sme vychádzali z osobnej skúsenosti a rozličných zdrojov, z ktorých sa najpodstatnejšie tematike venujú ([2]), ([6]) a ([7]).



Obr. 1: Decentralizovaná architektúra



Obr. 2: Centralizovaná architektúra

2.1. Decentralizovaný model

Najbežnejšie sa v praxi vyskytujúcim modelom je decentralizovaná architektúra, kde si každá služba samostatne zabezpečuje autentifikáciu a autorizáciu. Tieto architektúry vznikajú postupne, pridávaním služieb – vždy je jednoduchšie novú službu zaviesť nezávisle od existujúcich komponentov ako podniknúť koncentrovanú snahu o globálne riešenie. Postupne tak vzniká spleť služieb, ktorých používatelia si musia

2 Windley. *Digital Identity*. 2005
6 Dunne. *Build and Implement a Single Sign On Solution.* 2004
7 Risto, Ritter. *Single Sign On – The main focus is on the customer.* 2002

pamätať desiatky hesiel či iných autentifikačných údajov (viď obrázok 1).

2.1.1. Decentralizovaný model: pohľad používateľa

Pre používateľa je práca s takouto architektúrou veľmi náročná. Silne pociťuje hranice medzi jednotlivými službami, pri práci je často obťažovaný žiadosťami o autentifikovanie sa. Už základné bezpečnostné požiadavky, na ktorých trvajú administrátori jednotlivých služieb (pravidlá na bezpečné heslo, jeho pravidelné obmieňanie atď) sa stávajú závažnou prekážkou. Pamätanie si množstva hesiel je nereálne, preto si používatelia heslá zapisujú alebo používajú iba jedno heslo (prípadne veľmi malý počet rozličných hesiel) na prístup ku všetkým službám. Často heslá zabúdajú, čo zaťažuje administrátorov a znižuje produktivitu.

2.1.2. Decentralizovaný model: pohľad administrátora

Každá služba vyžaduje samostatnú správu používateľských účtov. Príchod nového používateľa vyžaduje opakované zadávanie údajov do jednotlivých služieb, často s vysokou redundanciou. Existuje nebezpečie vzniku nekonzistentnosti medzi jednotlivými údajmi. Každá zmena používateľových údajov (telefónne číslo, adresa) vyžaduje rozsiahle a pracné menenie tej istej položky na mnohých rozličných miestach. Pri rušení používateľských účtov hrozí pozabudnutie na niektorú službu a následný bezpečnostný prienik.

Celá architektúra je neprehľadná, zisťovanie globálnych informácií o používateľoch je zbytočne náročné. Používatelia vytvárajú problémami so zabudnutými či nesprávnymi heslami dodatočnú pracovnú záťaž na administrátorov, resp. pracovníkov technickej podpory (help desk).

2.1.3. Decentralizovaný model: bezpečnostný pohľad

Nie je možné hovoriť o všeobecných bezpečnostých vlastnostiach architektúry, keďže jej jednotlivé časti sú navzájom nezávislé. Môžu používať rozličné metódy

autentifikácie či bezpečnostné protokoly. Zabezpečenie každej služby a jej komunikácie s používateľom prebieha individuálne.

Všetky bezpečnostné nevýhody modelu vyplývajú z ľudského faktoru. Hrozí síce zvýšené riziko zmocnenia sa hesla útočníkom, samotný prienik je však obmedzený na jednu službu. Architektúra je pomerne odolná voči útokom typu *denial of service* (pokus o odopretie služby), výpadok žiadnej služby totiž neovplyvní funkčnosť ostatných častí.

2.1.4. Decentralizovaný model: technický pohľad

Decentralizovaný prístup poskytuje architektúre robustnosť proti výpadkom – nedostupnosť časti služieb neznamená nemožnosť práce. Pridávanie nových služieb môže prebiehať samovoľne, bez nutnosti sledovať interoperabilitu medzi jednotlivými časťami architektúry. Z technického hľadiska je triviálne, aj keď vyžaduje prácu vo forme vybudovania skladu dát o používateľoch pre každú pridávanú službu.

V architektúre existuje vysoká redundancia dát o používateľoch, čo má výhody aj nevýhody. Mnohonásobne sa zvyšujú nároky na úložný priestor týchto dát, nedostatok koordinácie môže mať za následok ich nekonzistenciu. Množstvo kópií údajov však zároveň slúži ako záloha v prípade straty údajov na niektorej službe. Taktiež nevznikajú kolízie pri práci s týmito údajmi či prípadné problémy s preťažením úložného bodu – architektúra je ľahko škálovateľná vzhľadom na počet služieb.

2.2 Centralizovaný model

Vznik centralizovaného modelu riešenia problematiky digitálnej identity vychádza zo snahy odstrániť nedostatky decentralizovaného modelu. Kľúčovým konceptom je existencia jedného centrálneho bodu, ktorý zabezpečuje autentifikáciu (prípadne autorizáciu) pre všetky ostatné služby v architektúre. Každý pokus o prístup ku chránenému zdroju, na ktorý by v decentralizovanej architektúre služba odpovedala

žiadosťou o autentifikáciu, je tu konzultovaný s centrálnou autoritou. Tá je zodpovedná za prebehnutie autentifikačného procesu, ako aj upovedomenie pôvodnej služby o jeho výsledku.

Väčšina riešení single sign-on využíva centralizovaný model – autentifikáciu zabezpečuje jedine centrálny bod, bez ohľadu na službu, ku ktorej sa používateľ snaží pristupovať. Je preto jednoduché skontrolovať, či sa už tento používateľ úspešne pokúšal o prístup k inej službe. Ak takýto záznam nájde a používateľ medzitým neukončil pracovné sedenie odhlásením sa, pokus o prístup automaticky schváli bez nutnosti interakcie s používateľom (viď obrázok 2).

Projekt OpenID je zaujímavou výnimkou – ide o riešenie webového single sign-on, ktorý nepoužíva centralizovanú architektúru. Podrobnejšie sa ním budeme zaoberať vo štvrtej kapitole.

2.2.1. Centralizovaný model: pohľad používateľa

Pre používateľa je centralizovaný model omnoho pohodlnejší. Autentifikácia prebieha vždy na tom istom mieste, tými istými autentifikačnými informáciami. Od používateľa sa nevyžaduje zapamätanie si veľkého množstva prístupových údajov. V prípade nasadenia riešenia single sign-on je pracovné pohodlie ešte zvýšené absenciou nutnosti opakovane sa autentifikovať. Používateľ si však na svoje prístupové údaje musí dávať oveľa väčší pozor – v prípade ich odcudzenia je vážne ohrozená celková digitálna identita používateľa, nie iba jej jednotlivé časti ako v predošlom prípade.

2.2.2. Centralizovaný model: pohľad administrátora

Z administrátorského hľadiska je centralizovaný model veľmi výhodný. Všetky údaje o používateľoch sú uložené na jednom mieste. Znižuje sa riziko chyby pri aktualizácii či rušení používateľských účtov. Pridávanie novej služby vyžaduje zvýšenú administrátorskú pozornosť - je nutné dosiahnuť bezproblémovú spoluprácu medzi pridávaným serverom a centrálnym bodom. Po ukončení úvodnej fázy už ale nie je

nutné serverom služieb venovať toľko pozornosti, keďže priamo nepracujú s používateľskými údajmi.

2.2.3. Centralizovaný model: bezpečnostný pohľad

Existencia centrálneho bodu je zároveň výhodou aj nevýhodou architektúry. Keďže všetka autentifikácia (prípadne autorizácia) prebieha cezeň, v prípade správne navrhnutého riešenia nie je vôbec potrebné aby servery jednotlivých služieb prichádzali do kontaktu s údajmi používateľa. V prípade prieniku na server služby teda útočník nezíska prístup k údajom, spojeným s používateľovou digitálnou identitou. Stále však získa prístup k tým údajom, s ktorými služba pracuje.

Na druhú stranu, každý prienik bezpečnosti na centrálnom bode je oveľa závažnejší, pretože útočníkovi poskytuje prístup do všetkých pokrývaných služieb. Je preto nutné centrálny bod zabezpečiť čo najlepšie. Taktiež je nutné zabezpečiť, aby mal server služby istotu, že komunikuje s centrálnym bodom a nie s útočníkom, ktorý sa zaň vydáva. Opačná istota nie je vo vhodne navrhnutom, čisto autentifikačnom riešení podstatná, pretože server služby neprichádza do kontaktu s údajmi používateľa – jediné, čo prípadný útočník dosiahne vydávaním sa za server služby je, že mu centrálny bod oznámi prípadnú úspešnosť pokusu o prihlásenie sa. V prípade centralizovaného riešenia, ktoré službám poskytuje aj autorizáciu, je samozrejme situácia iná – jednotlivé servery majú prístup k autorizačným údajom používateľa. Preto je v tomto prípade potrebné venovať väčšiu pozornosť nastoleniu dôvery medzi jednotlivými servermi.

Pri používaní jednotného hesla z klientskej stanice považovanej za nedôveryhodnú vzniká bezpečnostné riziko. Preto je v prípadoch, kedy sa používatelia riešenia nedokážu vyhnúť používaniu takýchto klientských staníc vhodné vybudovať hierarchiu dôveryhodnosti. Jej cieľom je rozdeliť služby na dôležité a menej dôležité, na prístup k týmto dvom kategóriám využívať rozličné heslá, a z nedôveryhodných klientských staníc povoľovať iba prístup k menej dôležitým službám. Tým sa zmierni závažnosť prípadného prieniku, spôsobeného použitím takejto stanice.

2.2.4. Centralizovaný model: technický pohľad

Veľkou nevýhodou je nutnosť kompatibility – jednotlivé služby musia byť schopné komunikovať s centrálnym bodom. Toto v závislosti od konkrétneho riešenia buď priamo vymedzuje, ktoré služby je vôbec možné do architektúry zapojiť, alebo si vyžaduje dodatočnú námahu pri zapájaní mnohých z nich.

V prípade nefungovania centrálneho bodu nie je dostupná žiadna služba. Preto je potrebné venovať zvýšenú pozornosť jeho dostupnosti, spoľahlivosti a procedúram na zotavenie sa z poruchy. Ostatné servery neukladajú informácie o používateľoch, čím sa znižujú hardvérové nároky na ne.

2.3 Prečo centralizovaný systém?

Na prvý pohľad je decentralizovaný systém oveľa výhodnejší nielen z hľadiska bezpečnosti, ale aj čo sa týka technických hľadísk. Jeho výhody sú žiaľ v praxi zmenšované chybou ľudského faktoru, voči ktorému je zraniteľný. Preto je výhodný centralizovaný systém napriek tomu, že zdanlivo ide o krok naspäť z technického aj bezpečnostného hľadiska. Zvýšenie používateľského aj administrátorského pohodlia vedie k redukcii chybovosti, čo zvyšuje spoľahlivosť a bezpečnosť architektúry.

Kým centralizované riešenie nepokrýva väčšinu poskytovaných služieb, sú náklady na jeho implementácie väčšie ako jeho prínos. Pretožev budúcnosti predpokladáme nárast počtu služieb poskytovaných univerzitou, rozsah týchto prvotných nákladov bude s postupom času rásť. V prípade existencie centralizovanej architektúry môže jej podpora byť jedným z kritérii pri výbere nových služieb. Tým by bolo možné predísť zbytočnej námahe pri neskoršom zapájaní do architektúry. Preto je vhodné uskutočniť prechod na centralizovanú architektúru čo najskôr.

V nasledujúcej kapitole sa budeme venovať rozboru súčasného stavu na UK a kritériám, ktoré budeme používať pri rozhodovaní o vhodnostiach riešenia.

3 Situácia na UK

V tejto kapitole najprv popíšeme súčasný stav problematiky autentifikácie na UK, potom sa budeme venovať jednotlivým službám, ktoré sa budeme snažiť v riešení pokryť. Okrajovo spomenieme aj služby, ktoré nespadajú do rozsahu tejto práce, sú však podstatné v celkovom kontexte projektu JAS. Na záver kapitoly sformulujeme požiadavky, ktorými sa budeme riadiť pri hľadaní a hodnotení vhodných alternatív riešenia.

3.1 Súčasný stav

Problém autentifikácie na univerzite nikdy nebol jednotne riešený. Ako rástli možnosti informačných technológií, zväčšoval sa aj rozsah služieb, poskytovaných univerzitou. Prirodzene tak vznikla decentralizovaná architektúra so všetkými pre ňu typickými vlastnosťami, ako sme ju popísali v kapitole 2.

Každá služba má vlastný dátový sklad, obsahujúci potrebné informácie. Na vytvorenie konta u služby je potrebný zásah administrátora, ktorý má danú službu na starosti. Rôzne služby majú na starosti rôzni administrátori (aj na fakultnej úrovni), komunikácia medzi nimi je často minimálna. Od študenta, ktorý chce využívať služby univerzitou poskytované, sa vyžaduje osobná návšteva administrátorov jednotlivých služieb, častokrát spojená s opakovaným udávaním tých istých údajov. Nastáva tak obrovská redundancia a zbytočná záťaž pre študentov aj zamestnancov. Absencia jednotného postupu pri rušení používateľských účtov absolventov vytvára potenciálne bezpečnostné riziká.

Univerzita má okolo 30 000 študentov ([8], s. 25) a 5 000 zamestnancov ([8], s. 57). Ak priemerný používateľ využíva viacero služieb, počet používateľských účtov sa stáva neúnosným. Táto situácia sa bude s postupom času iba zhoršovať, ako stúpa počet

8 *Výročná správa Univerzity Komenského v Bratislave za rok 2005*. Máj 2006 (pozn. autora – v dobe písania práce ešte nebola k dispozícii výročná správa za rok 2006, preto bol použitý odhad založený na správe z predchádzajúceho roku)

používateľov a objavujú sa stále nové služby. Preto je vhodné tieto problémy riešiť čo najskôr.

Nasleduje prehľad služieb, ktoré sa na univerzite nachádzajú.

3.1.1 Univerzitný web

V januári 2007 bol na univerzite spustený "nový web", založený na Typo3. Jedná sa tu o systém na spravovanie obsahu (ang. *content management system*), umožňujúci ľahké publikovanie informácií aj používateľom bez skúseností s tvorbou webstránok. Samotné Typo3 je aplikácia s otvoreným zdrojovým kódom, napísaná v skriptovacom jazyku PHP s využitím funkcionalityMySQL.

V systéme Typo3 existujú dva druhy používateľov: *koncový* ("front-end") a *prispievateľský* ("back-end").Koncový užívateľ si môže iba pozerať stránky, prípadne pracovať s elementami, ktoré mu autor pripravil na interakciu. Hosť (neprihlásený používateľ) je základným koncovým používateľom. Prispievatelia majú prístup k rozšírenému používateľskému rozhraniu, v ktorom môžu sebe vyhradené stránky meniť. Systém taktiež podporuje skupiny – jednotlivec má jednak vlastné práva, jednak práva zdedené zo všetkých skupín, do ktorých patrí. Nie je potrebné, aby jeden používateľ mal účty oboch typov – prispievateľský účet sa mimo oblasti svojej právomoci správa ako účet bežného koncového používateľa.

Typo3 si autentifikáciu môže zabezpečovať samostatne, prenechať ju na webserver, na ktorom beží, alebo používať niektorý z mnohých existujúcich *zásuvných autentifikačných modulov* (pluggable authentication module – technológia výmenných modulov zaručujúcich autentifikáciu, ktorej použitím sa dosahuje pružnosť a prispôsobiteľnosť systému)([9]). Autorizácia je zabezpečená priraďovaním používateľov do skupín, a prípadným jednotlivým modifikovaním ich prístupových práv.

9 Typo3 CMS: *Feature List*

3.1.2 Virtuálna knižnica

Virtuálna knižnica UK má dve hlavné funkcie – jednak automatizáciu knižničných procesov (výpožičky, katalogizácia, prehľady, štatistiky), a ďalej zabezpečenie prístupu k externým informačným zdrojom (elektronické knihy, scientometrické databázy, národné bibliografie, časopisecké články). Systém virtuálnej knižnice je prevádzkovaný na softvérovom balíku Virtua. Ide o komerčný produkt napísaný v skriptovacom jazyku Perl. V systéme sa nachádzajú dva druhy používateľov – *čitatelia* a *zamestnanci knižníc*. Čitateľ smie vykonávať iba veľmi obmedzené činnosti (zarezervovať si knihu, atď), právomoci zamestnanca knižnice sú väčšie (vkladá do systému nové knihy, pracuje s rezerváciami, atď).

Autentifikačné možnosti softwareového balíka sú obmedzené – program vie pracovať s vlastnou internou databázou, napojiť sa na SQL databázu (podporované sú MS SQL, Oracle a PostgreSQL), alebo preberať autentifikačné informácie cez LDAP([10]). Predpokladáme nemožnosť modifikácie programu.

3.1.3 Centrálna databáza osôb

CDO je celouniverzitná databáza študentov a zamestnancov UK. Ide o aplikáciu vyvinutú na UK, prevádzkovanú na platforme Sun Java System Application Server. Používateľ môže mať pridelených viacero rol, medzi ktorými sa prepína – informácie, ktoré sú mu dostupné, sa pri prepnutí roly primerane menia. Skupiny používateľov, ako aj jednotlivci, môžu byť pridelení do rozličných *Access Control Realms* (zoskupenia autentifikačných a autorizačných informácií).

Application Server má širokú skálu podporovaných autentifikačných možností - plný zoznam je k dispozícii v dokumentácii, pre nás podstatné podporované štandardy sú Kerberos ([11]), LDAP a prístup do externej databázy pomocou SQL.

10 Virtua Profiler *Global Settings.* 2005
11 *GSS-API/Kerberos5 Authentication*

3.1.4 E-learning

Na UK sa v súčasnosti na elektronickú výučbu používa takmer výhradne produkt Moodle. Ide o systém, napísaný v skriptovacom jazyku PHP (a poskytovaný ako aplikácia s otvoreným zdrojovým kódom). V systéme existujú dva druhy používateľov, *učitelia* a *študenti*. Učiteľ sa od študenta líši tým, že má právo vytvárať nové kurzy a spravovať ich. V prípade, že má záujem o kurz, poskytovaný v rámci systému iným učiteľom, môže sa doň prihlásiť ako bežný študent. Nie je teda potrebné, aby jeden používateľ mal viac účtov.

Vďaka aktívnej komunite a modulárnosti systému existujú rôzne autentifikačné moduly. Plný zoznam podporovaných možností je k dispozícii v dokumentácii projektu ([12]). Existuje aj možnosť autentifikáciu prenechať na webserver, na ktorom aplikácia beží (týmto spôsobom je možné dosiahnuť integráciu s autentifikačnými možnosťami, ktoré Moodle priamo nepodporuje, ale hosťujúci webserver áno).

3.1.5 Ostatné webové aplikácie

V univerzitnom prostredí taktiež existuje veľké množstvo rozličných samostatných webových aplikácií. Najbežnejšími reprezentantami tejto kategórie sú systémy online prihlasovania sa na skúšky, používané rozličnými katedrami. Takmer výlučne sa jedná o vlastné produkty. Takouto je aj pripravovaná aplikácia na "billing", čiže účtovanie hovorov vykonaných cez VoIP (viď nižšie). Vzhľadom na možnosť voľne do týchto aplikácií zasahovať nepredpokladáme problémy s ich začlenením do jednotného systému.

Ostatné služby, ktoré bude projekt JAS pokrývať síce nespadajú priamo do oblasti tejto práce, ľahkosť ich prepojenia s pripravovaným riešením pre autentifikáciu používateľov webových aplikácií je však pre projekt podstatná. Preto ich spomenieme aspoň okrajovo - bližšie sa im venuje M. Zagiba v interných materiáloch projektu ([13]). Z

12 *MoodleDocs*
13 Zagiba. *Návrh riešenia jednotnej autentifikácie a autorizácie na Univerzite Komenského.* 2007

technologického hľadiska sa na ich zapojenie javí najvhodnejším využitie protokolu Kerberos, preto mu aj my budeme venovať zvýšenú pozornosť.

3.1.6 Voice over IP

UK poskytuje aj *Voice over IP*, telefonovanie cez internet. Na samotné smerovanie hovorov sa používa softvérový balík s otvoreným zdrojovým kódom SIP Express Router. Jeho prvoradou úlohou je autentifikovať jednotlivé VoIP účastnícke stanice a spájať hovory, ako aj viesť záznamy o dĺžke hovorov. Autentifikuje buď voči vlastnej databáze, externej databáze, alebo voči RADIUS serveru. Jednotlivé zariadenia (telefónne prístroje) sú už nakonfigurované, autentifikujú sa samostatne (kombináciou IP adresy a prístupových kódov v prístroji), bez zásahu používateľa.

3.1.7 Dial-up, Virtual Private Networks, WiFi

UK poskytuje svojim študentom a zamestnancom aj možnosti vzdialeného pripojenia sa na internet cez univerzitné PPP servery. Taktiež poskytuje možnosť pripojenia do bezdrôtových sietí či vybudovanie VPN tunelov. V súčasnosti sa na autentifikáciu a autorizáciu vo všetkých prípadoch používa protokol RADIUS. Možnosti serverových balíkov, implementujúcich tento protokol, sú široké – ponúkajú autentifikáciu voči vlastnej databáze, externej databáze, ale aj Kerberos a LDAP. Pri začleňovaní do jednotného riešenia nepredpokladáme závažnejšie problémy.

Univerzita používa FreeRADIUS, ktorý podporuje širokú škálu autentifikačných metód ([14]) Pre nás je podstatná existencia podpory LDAP, MIT Kerberos a Active Directory.

3.1.8 Windows PC, siete, fileservery, mailservery

Možnosti strojov a služieb, bežiacich na platforme Windows sú obmedzené –

14 FreeRADIUS Wiki. *Authentication*

Spoločnosť Microsoft podporuje iba vlastné technológie, navyše interoperabilita medzi staršou a novšou generáciou často vedie k problémom. Ak nebude použité riešenie zamerané na jednu z týchto technológií, môže sa ukázať nevyhnutnosťou použiť nezávislé nástroje, aby bolo možné služby platformy Windows zapojiť.

3.1.9 Linux PC, siete, fileservery, mailservery

Na rozdiel od klientských staníc, bežiacich na rozličných verziach MS Windows, Linux podporuje širšie možnosti, vďaka využitiu systému zásuvných autentifikačných modulov. Tieto umožňujú použiť na lokálne prihlasovanie centralizovaný mechanizmus, ktorý sa používa aj na prístup k ostatným službám**.** Kompletné začlenenie všetkých linuxových služieb do systému pravdepodobne bude pracné a zdĺhavé, vďaka modulárnosti a širokej komunite podporujúcej Linux by sa však nemalo ukázať ako neriešiteľné bez ohľadu na technológiu použitého riešenia.

3.2 Ciele

Ako vidíme, služby poskytované univerzitou sú rôznorodé a pomerne rozsiahle. Je preto treba potrebné určiť si ciele, ktoré chceme projektom dosiahnuť. Nasleduje popis požiadaviek, ktoré boli v rámci prípravnej fázy identifikované ako podstatné:

3.2.1 Jednoduchosť používania

Pretože jednou z hlavných motivácií projektu je odbremeniť používateľov, budeme od riešenia vyžadovať, aby bolo jednoduché na používanie. V ideálnom prípade by jeho práca mala byť pre používateľa plne transparentná. Pretože to však asi nebude možné (na to by bolo nutné JAS previazať s lokálnym prihlasovaním sa na používateľov stroj, čo so sebou nesie celé spektrum problémov), uspokojíme sa aj s malým, jednorázovým zásahom do priebehu požívateľovej práce (prihlasovací dialóg do JASu).

3.2.2 Minimálna prípravná fáza

Vzhľadom na množstvo používateľov, ktoré bude JAS pokrývať, nie je realistické očakávať zapájanie klientských staníc špecializovaným personálom. Je teda dôležité, aby prípravná fáza pred tým, ako môže používať začať využívať výhody jednotnej autentifikácie, bola čo najjednoduchšia, zvládnuteľná aj laikom. V tomto ohľade by bola ideálnym riešením architektúra bez použitia agentov, odstraňujúca nutnosť zásahu do klientskej stanice. Ak toto nebude možné, cieľom bude jednorázové stiahnutie konfiguračného súboru, prípadne malého automatizovaného inštalátora. Špecifickou podtriedou problému sú prístupy z počítačov, kde nemôže používateľ nič inštalovať ani nastavovať (napríklad internetové kaviarne). Ak nebude možné týmto používateľom zaručiť plnú funkcionalitu, musí systém poskytovať alternatívy. Nie je prijateľné vylúčiť ich úplne.

3.2.3 Mobilita riešenia

Používatelia univerzitných systémov k nim pristupujú nielen z rozličných lokalít (prenosné počítače, pripojenie z domu atď), ale aj z rozličných klientských staníc (napr. Internet café). Preto pri výbere riešenia nebudú prichádzať do úvahy technológie založené na hardvérových úpravách klientskej stanice. Toto kritérium do istej miery súvisí s predošlým - technológie ako napríklad štruktúra založená na PKI nie sú vhodné kvôli problémom s distribúciou privátnych kľúčov a ich následným prenosom medzi klientskými stanicami. Problematike PKI v kontexte práce sa budeme bližšie venovať v 4. kapitole.

3.2.4 Dobrá podpora operačných systémov

Od použitej technológie budeme vyžadovať, aby bola multiplatformová – musíme autentifikačné služby poskytovať každému používateľovi, bez ohľadu na jeho operačný systém. Taktiež zo strany serverov a poskytovaných služieb musíme byť schopní do riešenia začleniť zdroje založené na viacerých OS (minimálne Windows a Linux).

3.2.5 Jednoduchosť rozširovania

Je podstatné, aby riešenie bolo dobre škálovateľné a rozširovateľné bez zásahov do už existujúcich a napojených služieb. Toto umožní nielen postupné zapájanie jednotlivých existujúcich služieb (s dôrazom na dôkladné testovanie funkčnosti), ale aj neskoršie bezproblémové rozširovanie o nové služby bez toho, aby bola už existujúca štruktúra znefunkčnená.

3.2.6 Maximálne pokrytie súčasných zdrojov

Použité riešenie by malo poskytovať čo najjednoduchšie zapojenie existujúcich služieb, podľa možnosti s čo najmenšími zásahmi do nich. Taktiež v prípade výberu medzi viacerými alternatívami, ktoré sa budú javiť rovnako vyhovujúcimi, bude mať prednosť tá, pri ktorej vieme využiť viac z už existujúcich zdrojov a vybavenia.

3.2.7 Odolnosť voči zlyhaniu

Jedným z najvýraznejších nedostatkov centrálneho riešenia autentifikácie je tzv. *single point of failure* – ak zlyhá služba, poskytujúca autentifikáciu, všetky zdroje sú znefunkčnené. Tomuto je nutné v bežnej prevádzke zabrániť, a preto budú pri výbere riešenia mať prednosť technológie, ktoré sú robustnejšie a ľahšie sa v nich dá tejto situácii zabrániť.

3.2.8 Hierarchia dôveryhodnosti

Nie všetky služby, ktoré potrebujeme jednotnou autentifikáciou pokryť, majú rovnaké bezpečnostné vlastnosti. Z bezpečnostného hľadiska môžeme služby rozdeliť na dôveryhodné a nedôveryhodné (od toho bude závisieť, či im povolíme v prípade nutnosti vidieť používateľovo heslo). Ďalej ich môžeme rozdeliť na dôležité a menej dôležité, čím určíme, aké silné heslo sa bude vyžadovať (viď tiež kapitolu 2, odsek 2.3).

Od riešenia budeme vyžadovať, aby umožnilo rozličné správanie sa voči týmto kategóriám služieb – je napríklad neprípustné, aby služba s minimálnym zabezpečením a odolnosťou voči útokom mala prístup k centrálnemu heslu používateľa. Prinajmenšom vyžadujeme možnosť rozdelenia služieb na nedôveryhodné a dôveryhodné, a tomu zodpovedajúce používanie dvoch rozličných hesiel pri prístupe k nim.

3.2.9 Odolnosť voči útokom

Centralizácia so sebou nesie zvýšenú zraniteľnosť voči útokom na centrálnom bode. Toto treba pri návrhu riešenia zohľadniť, aby výsledkom projektu nebola zvýšená zraniteľnosť univerzitných systémov. Vybrané riešenie by malo poskytovať zabezpečenie nielen voči útokom na server, ale aj útokom na prebiehajúcu komunikáciu či útokom na klientskú stanicu (v prípadoch, kedy má o tom zmysel uvažovať). Ideálne riešenie by malo poskytnúť odpoveď na všetky tieto druhy útokov. V prípade nedostatkov v tomto kritériu od riešenia vyžadujeme bezproblémovú spoluprácu s dodatočnými komponentami riešenia, nasadenými na odstránenie týchto nedostatkov (napríklad použitie SSL na zabezpečenie komunikácie medzi klientom a centrálnym bodom).

Toľko k súčasnej situácii na UK a požiadavkách, ktoré kladieme na hľadané riešenie. V nasledujúcej kapitole sa pozrieme na technológie a nástroje, prichádzajúce do úvahy ako riešenie.

4 Možnosti riešenia

Problém jednotnej autentifikácie patrí k intenzívne skúmaným témam súčasnosti. Množstvo rozličných autentifikačných protokolov, ako aj produktov, ktoré ich implementujú, je preto veľmi veľké. Nie všetky sú však vhodné pre riešenie situácie na UK. Preto nasleduje krátky prehľad technológií a produktov, ktoré sú potenciálne vhodné na riešenie autentifikácie používateľov webových aplikácií na UK. Z rozpočtových dôvodov boli z prieskumu vylúčené komerčné produkty (s výnimkou riešenia spoločnosti Sun – jedná sa síce o komerčný produkt, je však poskytovaný zdarma).

4.1 Web SSO – riešenia založené na cookies

Webový single sign-on je najjednoduchší na poskytnutie, pretože pokrývame iba služby jedného druhu – webservery. Používatelia sa k nim pripájajú pomocou webových prehliadačov, ktoré spĺňajú isté technologické štandardy bez ohľadu na operačný systém, odpadajú teda aj problémy s interoperabilitou. Najjednoduchšie single sign-on riešenia v tejto oblasti využívajú http cookies. Tieto sú podporované všetkými modernými prehliadačmi, nie je teda vôbec potrebné modifikovať (ani nijako nastavovať) klientov, kým tieto majú povolené používanie technológie. Použitím cookies však samozrejme SSO riešenie zdedí aj ich nedostatky, ktoré musí ošetriť. Ide hlavne o problémy krádeže či zmeny cookie, ale aj o problém nedostatočne presnej identifikácie používateľa ([15]).

Z veľkého množstva dostupných single sign-on riešení založených na cookies sme vybrali troch reprezentantov – produkty Pubcookie, CoSign a Sun Java System Access Manager (ďalej AM). Pubcookie bol vybraný pre svoju jednoduchosť, CoSign pre rozsiahlosť ponúkaných autentifikačných metód. AM bol vybraný pre dobrú podporu zapojenia rozličných služieb, ako aj jednoduchosť spolupráce s produktom Sun

15 Stein, Stewart. *The WWW Security FAQ – Client Side Security*. 2003

Java System Identity Manager, ktorého použitie je zvažované v projekte JAS([13]).

4.1.1 Pubcookie

Pubcookie je projekt s otvoreným zdrojovým kódom, zameraný na riešenie problému webového SSO v rámci jednej inštitúcie. Projekt pôvodne vznikol v roku 1998 na University of Washington, v roku 2001 sa jeho autori začlenili do iniciatívy Internet2 a otvorili zdrojový kód projektu.

Projekt rieši problém webovej autentifikácie pomocou cookies. Obsahuje tri základné komponenty: *aplikačný server, prihlasovací server* a *autentifikačný mechanizmus*. Pod aplikačným serverom sa rozumie server (s nainštalovaným kontrolným modulom), na ktorom beží pokrývaná služba. Tento modul kontroluje, či sa používateľ autentifikoval – v prípade chýbajúcej informácie o autentifikácii ho presmeruje na prihlasovací server. Ten je zodpovedný za vygenerovanie prihlasovacieho formulára a spracovanie pokusu o prihlásenie. Na to využíva služby externého autentifikačného mechanizmu([16]). Po úspešnom autentifikovaní používateľa nastaví prihlasovací server cookie, ktoré o tejto skutočnosti informuje ďalšie služby, ku ktorým bude používateľ v priebehu pracovného sedenia pristupovať.

Pôvodným zámerom projektu bolo riešiť problém SSO iba v rámci jednej organizácie, preto používa aj cookies s platnosťou na doménu. Preto je na jeho fungovanie nutné, aby aplikačné servery boli v rovnakej doméne ako prihlasovací server. Vedľajším efektom riešenia je taktiež znížená miera bezpečnosti – ľubovoľný server v doméne má prístup ku *granting cookie* (cookie, vytvorený prihlasovacím serverom, obsahujúci informáciu o používateľovom oprávnení) a jeho obsahu.

Projekt v súčasnosti poskytuje moduly pre Apache a IIS. Ako externé autentifikačné služby sú overené Kerberos, LDAP a NIS. Vzhľadom na technické a bezpečnostné obmedzenia však pred Pubcookie v projekte JAS uprednostníme pokročilejšie alternatívy.

13 Zagiba. *Návrh riešenia jednotnej autentifikácie a autorizácie na Univerzite Komenského.* 2007
16 *How Pubcookie works.* 2003

4.1.2 CoSign

CoSign je riešenie problému webového SSO, pochádzajúce z University of Michigan. Ide o projekt s otvoreným zdrojovým kódom, používajúci cookies. Autori však odstránili niektoré slabiny technológie, čím zvýšili jej bezpečnosť aj použiteľnosť.

Základná architektúra je zhodná s inými SSO riešeniami – jednotlivé služby odkazujú používateľa na centrálny server, kde sa autentifikuje. V prípade, že sa už počas pracovného sedenia autentifikoval, táto informácia je poskytnutá službe spôsobom transparentným pre používateľa. Systém nepoužíva doménové cookies, používa iba host cookies. Týmto sa dosahuje nielen zvýšenie bezpečnosti (v prípade použitia doménových cookies má ľubovoľný server v doméne možnosť práce s cookie, obsahujúcim autentifikačné informácie), ale aj zvýšenie použiteľnosti – jednotlivé služby totiž nemusia byť v jednej doméne.

Systém používa dve triedy cookies – *cookie služby* ("service cookie") a *prihlasovací cookie* ("login cookie"). V rámci sedenia má používateľ resp. jeho prehliadač iba jeden prihlasovací cookie – tento je vytvorený centrálnym serverom po úspešnej autentifikácii. Každá služba, ku ktorej používateľ pristupuje, si vytvorí vlastný cookie služby a asociuje ho s používateľovým prihlasovacím cookie. Sedenie môže byť ukončené tromi spôsobmi: navštívením URL pre odhlásenie, dlhšou dobou nečinnosti ("idle timeout") a uplynutím dopredu stanoveného času od začiatku sedenia ("hard timeout"). Pre zvýšenie bezpečnosti je možné službe nastaviť, aby požadovala autentifikáciu používateľa pri jeho prvom prístupe k nej, bez ohľadu na predošlú úspešnú autentifikáciu pri prístupe k inej službe ([17]).

Zaujímavou funkcionalitou je *CoSign Friend* – ide o modul, umožňujúci vytvárať dočasné hosťovské účty pomocou e-mailových "pozvánok". Pozvánku môže zaslať ľubovoľný autentifikovaný používateľ, administrátorský zásah nie je potrebný. Tento produkt je možné nasadiť aj samostatne (bez použitia CoSign) do iných SSO ([18]).

17 Craig et al. *The Cosign Web Single Sign-On Scheme (draft 4)*. 2006
18 Craig, Craig. *Leveraging Guest Accounts for Ubiqutous Web Sign-On System Acceptance.* 2005

V súčasnosti sú k dispozícii agenti (v projekte nazývaní "filtre") pre Apache a IIS. Taktiež je k dispozícii JavaCoSign agent, ktorý umožňuje pokrytie webových aplikácii založených na technológii J2EE. Možnosti na autentifikáciu používateľa zahŕňajú BasicAuth (meno a heslo), certifikáty štandardu X.509 a Kerberos. Od verzie 2.0 vyššie existuje taktiež podpora pre externé autentifikačné mechanizmy. Systém umožňuje multifaktorovú autentifikáciu ([19]).

4.1.3 Sun Java System Access Manager

Spoločnosť Sun poskytuje sadu štyroch produktov *Sun Java System*, zameraných na prácu s digitálnou identitou: *Access Manager* (web SSO, autorizácia), *Identity Manager* (provisioning), *Federation Manager* (federovanie identít) a *Directory Server* (LDAP server). Každý z nich je samostatný, nie je potrebné ich nasadzovať spolu. My sa bližšie zameriame na Access Manager, Identity Manager spomenieme aspoň okrajovo vzhľadom na zvažovanie jeho použitia v projekte JAS. V súčasnom stave nemá zmysel uvažovať o federovaní identity a problematika prípadnej zmeny univerzitných LDAP serverov (v súčasnosti používajú OpenLDAP) je mimo oblasti záujmu tejto práce, preto sa produktami Federation Manager a Directory Server nebudeme zaoberať.

Access Manager je nástroj, poskytujúci web SSO. Jedná sa o štandardnú architektúru klient-server, k dispozícii sú *policy agents* (moduly na začlenenie služieb) pre široké spektrum webserverov (Apache, IIS a Sun Java System Application Server sú samozrejmosťou, ponuka jednotlivých agentov je k dispozícii na stránkach produktu). Vždy keď sa neautentifikovaný používateľ snaží pristupovať k chránenému zdroju, agent požiadavku odchytí a presmeruje na centrálny prihlasovací server. Ak sa používateľ v rámci sedenia už autentifikoval, je transparentne presmerovaný naspäť a môže pristúpiť k zdroju. V opačnom prípade sa musí u centrálneho serveru autentifikovať. Existuje možnosť používateľa autentifikovať iba jednorázovo, na prístup k jednému konkrétnemu zdroju. Produkt ponúka rozličné možnosti autentifikácie, pre

19 Craig, Craig. *Cosign Multi-Factor Specification (draft 6)*. 2006

nás podstatné sú Kerberos ([20]), RADIUS či meno a heslo (úplny zoznam je k dispozícii na stránke produktu). Informácie o používateľoch sú štandardne uložené v adresárovej službe, ku ktorej AM pristupuje cez LDAP, je však možné použiť aj relačnú databázu.

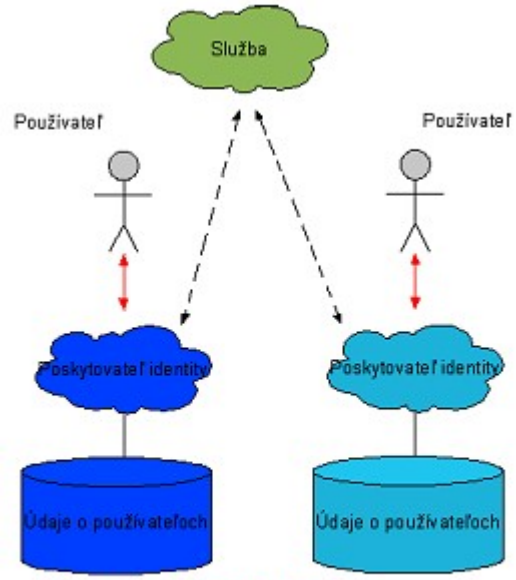
Identity Manager je produkt, zameraný na problematiku autorizácie a správu používateľských účtov ("provisioning") v iných ako webových aplikáciach. Ide o architektúru bez použitia agentov – nie je teda potrebné nič instalovať na spravovaných službách. Vytváranie používateľských účtov a zmeny ich právomocí je možné automatizovať skriptovaním. Identity Manager pristupuje k jednotlivým službám ako administrátor (je preň teda nutné vytvoriť účet s primeranými právomocami) cez Telnet či SSH. Jednotlivým používateľom môžu byť priraďované rozličné roly, ktoré ďalej určujú, ku ktorým zdrojom bude mať prístup. Informácie o používateľoch môžu byť uložené v relačnej databáze alebo adresári s prístupom cez LDAP. Hlavným cieľom produktu je uľahčenie práce administrátora, prínos pre radového používateľa je minimálny (samoobslužná zóna na zmenu hesiel v jednotlivých službách).

Riešenie spoločnosti Sun teda ponúka zaujímavú alternatívu, poskytujúcu nielen autentifikáciu a webový single sign-on, ale aj autorizáciu a provisioning používateľských účtov. Za hlavnú nevýhodu sa dá považovať naviazanosť riešenia na technológie spoločnosti (Java, operačný systém Solaris).

4.2 OpenID

OpenID je iniciatíva, poskytujúca zaujímavú alternatívu ku systémom webových single sign-on na báze cookies. Základnou myšlienkou je, že používateľ sa preukazuje pomocou URL, ktoré vlastní. Toto URL odkazuje na dokument HTML, ktorý v hlavičke obsahuje referenciu na *poskytovateľa identity* - server, ktorý potvrdzuje vlastníctvo tohoto URL. Tento môže byť niektorým z verejných serverov, ale aj súkromným serverom, prevádzkovaným používateľom. Voľba konkrétneho poskytovateľa identity je v rukách vlastníka URL – osoby, ktorá naň umiestnila dokument. Ide teda o decentralizovaný systém (viď obrázok 3).

20 Nichols. *Kerberos and Access Manager Single Sign-On.* 2005



Obr. 3: OpenID

Praktická prevádzka vyzerá nasledovne: Používateľ pri prístupe k serveru, podporujúcemu OpenID, ako svoje používateľské meno uvedie svoje OpenID URL. Server z neho vyextrahuje referenciu na poskytovateľa identity. Ak sa používateľ v priebehu tohoto sedenia ešte u serveru neautentifikoval, presmeruje používateľa. Ten sa autentifikuje u poskytovateľa identity, ktorý potvrdí vlastníctvo URL a vráti používateľa na server, ku ktorému pristupoval. V prípade, že sa už používateľ v rámci tohoto sedenia u poskytovateľa identity autentifikoval, tento vlastníctvo priamo potvrdí (proces prebieha transparentne). S používateľovými autentifikačnými údajmi teda prichádza do styku iba poskytovateľ identity, ktorého si sám vybral (a ktorý môže byť pod jeho plnou kontrolou, ak si to želá).

Na prvý pohľad nemusí byť zrejmé, prečo sa v prípade OpenID naozaj jedná o autentifikáciu – koniec koncov, služba sa na identitu používateľa pýta nejakého poskytovateľa identity, ktorého vôbec nepozná a nemá dôvod mu dôverovať. Tu je však podstatné si uvedomiť, čo tvrdí používateľ tým, že zadá svoje OpenID – hovorí tým "Toto URL patrí mne, mám nad ním kontrolu." Po otvorení URL sa služba dozvie, ktorý poskytovateľ identity potvrdzuje alebo vyvracia vlastníctvo URL. Túto informáciu dokáže zmeniť iba vlastník URL, nehrozí preto krádež identity zmenou tu

uvedeného poskytovateľa. Po autentifikovaní sa používateľa u poskytovateľa identity tento informuje službu o úspešnom priebehu. Z tejto informácie služba vie, že používateľ je oprávnený (podľa poskytovateľa, vybraného vlastníkom URL) uvádzať toto URL ako svoju identitu.

O tom, či je OpenID v skutočnosti single sign-on by sa dalo diskutovať – pri návšteve každého nového zdroja musí používateľ znovu uviesť svoju OpenID identitu. Samotná autentifikačná výmena medzi používateľom a poskytovateľom identity však prebieha iba raz za pracovné sedenie, preto sme sa ho sem rozhodli zaradiť.

Technológia je zaujímavá v širokorozsiahlom nasadení, s priemerne zdatnými používateľmi. V prípade laikov, ktorí nechcú alebo nevedia prevádzkovať vlastného poskytovateľa identity a namiesto neho by všetci používali centrálneho, univerzitou poskytovaného, sa však výhoda distribuovanosti systému stráca. Ak sa však aj nepoužije v rámci jednotného autentifikačného systému ako kľúčový prvok, stojí za úvahu poskytovať našim používateľom identity OpenID ako súčasť služieb.

Nasleduje popis technológií použiteľných nielen na autentifikáciu používateľov webových aplikácií, ale aj v ostatných častiach projektu JAS.

4.3 Kerberos

Kerberos je sieťový autentifikačný protokol, založený na princípoch symetrickej kryptografie a zdieľaného tajomstva. Základným princípom fungovania je zavedenie tretej strany (serveru Kerberos) do komunikácie medzi účastníkmi ľubovoľnej výmeny. Tento server každému z účastníkov potvrdí identitu jeho partnera a poskytne šifrovacie možnosti pre ich vzájomnú komunikáciu. Skutočnosť, že používateľ sa neautentifikuje priamo u cieľovej služby, ale u tretej strany, ktorá je vždy rovnaká, bez ohľadu na cieľovú službu, zaručuje, že používateľ potrebuje svoju identitu dokazovať iba raz v rámci konkrétneho pracovného sedenia. Sila protokolu ďalej spočíva v tom, že

používateľove heslo neopúšťa jeho pracovnú stanicu, vďaka použitým kryptografickým metódam ([21]).

Protokol ma viacero existujúcich implementácií, sú pokryté všetky bežné platformy. Komponent *Integrated Windows Authentication* (štandardná súčasť Windows od verzie 2000 a vyššie) je založený hlavne na tomto protokole. *Active Directory* ho používa na vybudovanie vzťahu dôvery, keď používateľ v jednej doméne žiada o prístup ku zdrojom inej domény. Vo svete UNIX taktiež pre protokol existuje dobrá podpora, vrátane autentifikačných modulov (PAM), ktoré ho využívajú. Pri vhodne nakonfigurovaných klientoch a serveroch je ho možné použiť nielen na zdieľanie zdrojov na sieti, ale aj na webový single sign-on či poskytovanie autentifikácie pre rozličné služby (za predpokladu, že server, na ktorom služba beží, podporuje takúto autentifikáciu).

Základným problémom je interoperabilita – implementácie protokolu sa líšia. Pri použití MIT Kerberosu na UNIXových systémoch je však možné vhodnou aplikáciou vzťahu dôvery medzi Windows doménou a MIT realmom (a s pomocou mapovania používateľských účtov) dosiahnuť bezproblémovú spoluprácu ([22]).

Použitie protokolu Kerberos na web SSO sa zakladá na využití protokolu SPNEGO (resp. jeho implementácie v autentifikačnom rozšírení HTTP Negotiate), bližšie sa budeme problematike venovať v kapitole 5.

Nevýhodami protokolu sú: nutnosť obšírneho nastavovania klientskych staníc, problematická podpora u starších operačných systémov (Windows NT), ako aj neexistujúca podpora v niektorých aplikáciách, ktoré potrebuje JAS pokryť. Na druhú stranu, za veľkú výhodu môžeme považovať, že bežné distribúcie cieľových operačných systémov už obsahujú potrebný klientsky software, a teda nie je potrebné nič inštalovať.

21 Kohl, Neuman. *RFC 1510* . 1993.

22 Shinder. *How to use Kerberos Authentication in a Mixed (Windows and UNIX) Environment*. 2006

Miera podpory protokolu v iných produktoch nie je vždy úplne zrejmá. Niektoré z nich totiž pod "podporou Kerberosu" myslia iba overenie si mena a hesla u Kerberos serveru, ktoré im používateľ prezradí (čím sa strácajú prakticky všetky bezpečnostné vlastnosti), namiesto plnej podpory práce s ticketmi.

4.4 Public Key Infrastructure

Jednou z možností riešenia, na ktorú sa vďaka spojeniu s elektronickým obchodom upriamuje pozornosť, je použitie asymetrického šifrovania v procese autentifikácie. Každý účastník má dva *kľúče – verejný*, ktorý je voľné dostupný, a *súkromný*, ktorý je známy iba jemu. Informáciu zašifrovanú jedným z nich je možné rozšifrovať iba použitím druhého. Ak je informácia zašifrovaná verejným kľúčom (čo môže urobiť každý, pretože je teoreticky verejne dostupný), bude ju schopný rozšifrovať iba adresát (nedozvie sa však, kto ju zašifroval). Ak je naopak zašifrovaná súkromným kľúčom, rozšifrovať ju môže s použitím verejného kľúča hocikto. V tomto prípade bude mať adresát istotu o identite odosielateľa (pretože nikto iný nemá prístup k zodpovedajúcemu súkromnému kľúču).

Jednou z otázok, ktoré treba vyriešiť, je spoľahlivá distribúcia verejných kľúčov. Ak by totiž útočník úspešne zamenil niečí verejný kľúč za svoj, stáva sa v očiach technológie touto osobou. Preto sa na šírenie verejných kľúčov používaju *certifikáty*, podpísané *certifikačnou autoritou* (ďalej iba CA). Tá sa svojim podpisom zaručuje, že verejný kľúč obsiahnutý v certifikáte naozaj patrí danej entite. Akýkoľvek pokus o manipuláciu s informáciami obsiahnutými v certifikáte by narušil podpis CA, a tým zneplatnil certifikát. Ak teda účastník výmeny dostane certifikát podpísaný vierohodnou CA, môže si byť istý, že verejný kľúč v ňom je pravý.

Druhým problémom pri použití tejto technológie je, ako udržať v bezpečí súkromný kľúč. Najbežnejším riešením je jeho uloženie na hardvérové zariadenie, aby sa nemusel ukladať na pracovnej stanici (ak by sa mal ukladať na stanici, musíme tejto stanici absolútne dôverovať). Toto má za následok štandardne vysoké finančné náklady

pri implementovaní technológie na účely bezpečnej autentifikácie. Najbežnejší spôsob využitia technológie PKI na dosiahnutie web SSO spočíva práve v tomto riešení, spolu s nasadením softvérového produktu, ktorý zabezpečuje jeho využitie na autentifikovanie používateľa.

Riešenie problému autentifikácie cez PKI by malo nesporné výhody. S výnimkou kontroly *zoznamu zamietnutých certifikátov* ("certificate revocation list" - certifikačnou autoritou udržiavaný zoznam certifikátov, ktoré boli zneplatnené) nie je pri použití technológie potrebná komunikácia s nejakou centrálnou autoritou. Ďalej pre technológiu existuje široká podpora, pretože je bežne využívaná (elektronický podpis, zabezpečovanie komunikácie). Samozrejme, technológia má aj svoje nedostatky ([24]). Hlavnými nevýhodami pre nás sú finančná náročnosť riešenia, ako aj obtiažnosť fyzickej distribúcie prvkov medzi používateľov.

Nasledujúce dva protokoly nie sú pre použitie na autentifikáciu používateľov webových aplikácií v prostredí UK vhodné. Pre úplnosť ich však uvádzame, spolu s odôvodnením ich zamietnutia.

4.5 Lightweight Directory Access Protocol

Jedná sa o sieťový protokol, určený na pristupovanie k adresárom a ich modifikovanie cez TCP/IP. Vznikol ako alternatívna implementácia protokolu na prístup k adresárom štandardu X.500. Pôvodný protokol X.500 vyžadoval prácu s protokolmi referenčného modelu OSI a navyše bol príliš rozsiahly a ťažkopádny. Preto vzniká jeho odľahčená ("lightweight") verzia, založená na TCP/IP.

Hlavnou výhodou protokolu je jeho jednoduchosť. Vďaka tomu je takmer univerzálne akceptovaný a široko podporovaný, patrí k štandardom v oblasti ([25]). Protokol samotný popisuje operácie, vykonanie ktorých môže klient vyžadovať od adresárového serveru. Taktiež poskytuje možnosti na zabezpečenie komunikácie s

24 Ellison, Schneier. *Ten Risks of PKI:What You're Not Being Told About Public Key Infrastructure,* 2000
25 Findlay. *Security with LDAP*. 2002

klientom (TLS, mechanizmy založené na SASL). Existuje viacero implementácii, pre nás zaujímavé sú OpenLDAP a Java Directory Server.

Všetky webové aplikácie, ktoré bude projekt JAS pokrývať, implementujú autentifikáciu pomocou tohto protokolu – majú teda schopnosť komunikovať pomocou protokolu so serverom LDAP za účelom overenia autentifikačných údajov, ktoré používateľ zadal. Aj v súčasnosti existujú na univerzite LDAP servery, používané na ukladanie informácií o používateľoch.

Napriek tomu, že protokol sa bežne využíva priamo ako autentifikačný mechanizmus, toto riešenie je pre nás nevyhovujúce – predpokladá totiž, že všetkým službám, ktoré sa budú na adresárovú službu obracať, je možné zveriť autentifikačné údaje používateľa. V architektúrach obmedzeného rozsahu môže byť prijateľné dovoliť všetkým pokrývaným službám manipulovať s týmito údajmi, v našom prostredí však musí riešenie poskytovať viacero úrovní dôveryhodnosti ([1]). Toto jednoduchá autentifikačná schéma s použitím LDAP nedokáže, preto pre nás ako možnosť riešenia neprichádza do úvahy.

4.6 Radius

Radius je autentifikačný a autorizačný protokol, určený na riadenie vzdialeného prístupu. Bežnými klientskými stanicami Radius serveru sú prístupové body na vzdialený prístup do siete (routery, switche, modemové servery). Tieto si u Radius serveru overujú platnosť autentifikačných údajov, ktoré im poskytol klient, keď sa k nim pripájal.

Protokol vznikol, aby umožnil centralizovanú správu používateľských údajov klientov – nie je realistické očakávať, že pri každej zmene budú aktualizované údaje vo všetkých kusoch hardvéru, s ktorými klient môže pracovať. Taktiež tu hrajú rolu technické obmedzenia týchto zariadení - bežný poskytovateľ pripojenia má tisíce klientov, hardvérové zariadenia nemajú dostatočnú pamäť na to, aby ich autentifikačné

1 Mederly. *Jednotná autentifikácia používateľov informačných technológií*. 2006

údaje uložili lokálne.

Protokol je de facto štandardom v oblasti, podporuje ho každý výrobca hardwaru. Radius server je možné použiť aj ako autentifikačnú autoritu pre iné služby, na čom sa zakladá jeho využitie ako web SSO. Existujú zásuvné moduly pre jednotlivé webservery Apache a IIS, ktoré im umožňujú využiť služby Radius serveru ako autority na overenie mena a hesla, dodaného používateľom.

Protokol samotný má isté bezpečnostné slabiny ([25]), vyplývajúce hlavne z jeho využitia hašovania šifrou MD5. Tieto sú pri bežnom používaní zanedbateľné, pri použití Radius serverov ako autentifikačnej autority však závažne oslabujú celkovú bezpečnosť systému. Navyše takéto riešenie neumožňuje implementáciu hierarchie dôveryhodnosti, preto je pre nás nevyhovujúce.

Protokol má mnoho implementácii, komerčných aj voľne dostupných. Na UK sa v súčasnosti používa FreeRADIUS, implementácia s otvoreným zdrojovým kódom. Poskytovanými službami je vyhovujúca a poskytuje dobré možnosti na začlenenie do jednotného autentifikačného systému (podrobnejšie viď 3.1.7.), preto nepredpokladáme nutnosť meniť používanú implementáciu za inú.

Vzhľadom na služby, poskytované univerzitou, je začlenenie Radius serverov do architektúry nutnosťou. Kvôli bezpečnostným problémom spojeným s použitím protokolu ako autentifikačného mechanizmu však budú z hľadiska riešenia iba daľšími službami.

V nasledujúcej kapitole vyberieme možnosti riešenia prichádzajúce do úvahy pre projekt, a podrobíme ich bližšej analýze.

25 Hill. *An Analysis of the RADIUS Authentication Protocol.* 2001

5 Výber alternatívy

V tejto kapitole sa budeme venovať podrobnejšej analýze troch možností, ktoré zvažujeme vo finálnom výbere riešenia – protokol Kerberos za použitia mechanizmu SPNEGO, produkt CoSign, a produkt Sun Access Manager. Všetky možnosti buď priamo obsahujú Kerberos ako integrálnu súčasť, alebo majú dobrú podporu pre jeho použitie.Tým dosiahneme ľahké prepojenie riešenia problematiky webového single sign-on s ostatnými časťami projektu JAS, kde bude Kerberos pravdepodobne použitý ako hlavný autentifikačný mechanizmus ([13]).

SPNEGO sme vybrali, pretože ide o spôsob, ako na webovú autentifikáciu použiť protokol Kerberos bez nasadenia daľších nástrojov. CoSign bol vybraný ako reprezentant triedy webových single sign-on produktov s najrozsiahlejšími možnosťami implementácie autentifikačnej hierarchie. Sun Access Manager je atraktívny kvôli ponúkanej autorizačnej funkcionalite, ako aj bezproblémovej spolupráci s Identity Managerom, zvažovanom v ostatných častiach projektu.

Postupne budeme tieto možnosti bližšie skúmať z hľadiska požiadaviek, ktoré sme identifikovali v rámci kapitoly 3.

5.1 Kerberos a Simple Protected Negotiation Protocol

Vďaka použitiu protokolu Kerberos v *Integrated Windows Authentication* ([26]) a existencii vhodných zásuvných autentifikačných modulov pre Unix je možné spojiť lokálne a vzdialené prihlasovanie. Dochádza tak k pravému single sign-on: používateľ sa prihlási iba raz, na svoju pracovnú stanicu. Po zvyšok pracovného sedenia už nemusí opakovať autentifikačný proces, bez ohľadu na zdroj, ku ktorému pristupuje.

13 Zagiba. *Návrh riešenia jednotnej autentifikácie a autorizácie na Univerzite Komenského.* 2007
26 *Integrated Windows Authentication (IIS 6.0)*

Protokol SPNEGO ([27]) implementuje Kerberos ako jeden z možných spôsobov autentifikácie prehliadača voči webserveru. V produktoch spoločnosti Microsoft je protokol SPNEGO implementovaný v autentifikačnom rozšírení HTTP Negotiate ([28]). Zo serverov tento protokol podporujú IIS (od verzie 5.0 vyššie) aj Apache (vďaka modulu *mod_auth_kerb*), z prehliadačov *Internet Explorer* od verzie 5.01, *Mozilla* 1.7, *Firefox* 0.9 (a vyššie), ako aj *Konqueror* 3.3.1. Vzhľadom na nebezpečie zneužitia však vyžaduje autentifikácia pomocou mechanizmu *Negotiate* pred použitím v každom browseri konfiguráciu (zadanie zoznamu adries, ktorým bude dovolené v komunikácii s browserom použiť *Negotiate*).

Toto riešenie bolo pôvodne navrhované pre použitie v intranetových sieťach, z čoho plynú isté obmedzenia. Klient a server musia byť buď súčasťou tej istej *domény* (Kerberos realm) alebo rozličných domén, medzi ktorými bol zadefinovaný vzťah dôvery([22]). Protokol SPNEGO sa navyše nezaoberá problematikou integrity a dôvernosti údajov, ktorú sú pomocou neho prenášané. Preto je na zaručenie integrity a dôvernosti vhodné použiť ho v kombinácii so zabezpečeným HTTP prenosom (https).

5.1.1 SPNEGO: Jednoduchosť používania

Miera transparentnosti tohto riešenia pre používateľa je jednou z jeho hlavných výhod. Vďaka možnosti previazania lokálneho prihlasovania (pri platformách UNIX a Windows 2000 a vyššie) so single sign-on riešeným touto technológiou môže riešenie pracovať kompletne na pozadí. Ticket používateľa sa inicializuje počas lokálneho prihlasovania, bez nutnosti akéhokoľvek ďalšieho zásahu používateľa. Prehliadač tento ticket sám v prípade potreby vyberie z úložiska ticketov a pošle ho serveru na spracovanie. Proces je teda v ideálnom prípade úplne transparentný. Ak chýba naviazanie na lokálne prihlasovanie, je nutná akcia používateľa (prihlásenie sa cez Kerberos, čím obdrží ticket, ktorý sa bude naďalej používať), stále je však jednoduchá.

27 Zhu et al. *RFC 4178*. 2005.
28 Jaganathan et al. *RFC 4559*. 2006.
22 Shinder. *How to use Kerberos Authentication in a Mixed (Windows and UNIX) Environment*. 2006

5.1.2 SPNEGO: Minimálna prípravná fáza

Prípravná fáza pre toto riešenie spočíva z dvoch krokov: uvedenie Kerberos klienta do prevádzky a povolenia protokolu SPNEGO. Windows 2000 a vyššie majú zabudovaného Kerberos klienta vo forme *Windows Desktop SSO* – uvedenie do prevádzky bude spočívať z nastavenia správnej domény. Príslušnosť k doméne je nutnou podmienkou zapojenia klienta. Pre staršie verzie Windows existujú samostatní klienti, čím pribúda dodatočný krok inštalácie. V prípade operačných systémov na platforme Unix pôjde o nainštalovanie a vhodné nastavenie zásuvného autentifikačného modulu PAM-KRB5, čo je síce náročnejšia operácia, stále však zvládnuteľná pre priemerného používateľa.

Postup pri povolení protokolu SPNEGO sa líší s jednotlivými prehliadačmi, nie je však príliš zložitý. V prípade, že jednotlivé servery, pokrývané riešením nepatria do jednej domény, je nutné každú doménu nastaviť zvlášť.

5.1.3 SPNEGO: Mobilita riešenia

Nedostatočná mobilita je najväčšou slabinou tohto riešenia. Pred jeho použitím je nutné vykonať úpravy nastavení, ktoré jednoducho neprichádzajú do úvahy v prostredí internetovej kaviarne či klientskej stanice, na ktorej je používateľ iba hosťom. Táto slabina je závažnou prekážkou, pretože môže potenciálne celkom zabrániť veľkému percentu používateľov využívania služby JASu. Preto, ak má byť použité toto riešenie, je nutné nasadiť ešte doplnkové riešenie na zastrešenie týchto prípadov.

5.1.4 SPNEGO: Podpora OS

Protokol Kerberos má implementácie vo všetkých pre projekt dôležitých operačných systémoch. Jednotlivé implementácie sa často líšia, ich rozdiely a riešenia pre vznikajúce problémy sú však dobre zdokumentované. Nie je preto problém dosiahnuť interoperabilitu medzi klientom a serverom, ani keď každý z nich pracuje pod iným operačným systémom.

5.1.5 SPNEGO: Jednoduchosť rozširovania

Z hľadiska serverov, ktoré poskytujú zastrešované služby, nevyžaduje rozširovanie systému o ďalšie služby žiadne zmeny. Každý z jednotlivých serverov potrebuje iba informáciu o distribučnom centre, na ktoré sa má obrátiť so žiadosťou o overenie ticketu. Počet a identita iných serverov, ktoré sa na toto centrum tiež obracajú, sú preň nezaujímavé.

V prípade klientských staníc a prehliadačov na nich je situácia trochu odlišná. V prípade, že prírastok sa nachádza v rovnakej doméne ako nejaká už pokrývaná služba, nie sú potrebné žiadne zásahy. Ak ide o novú doménu je potrebné pre ňu vykonať rovnakú konfiguráciu ako pri pôvodnom uvedení do prevádzky.

5.1.6 SPNEGO: Maximálne pokrytie súčasných zdrojov

Na pokrytie služby týmto riešením je potrebné, aby buď podporovala autentifikáciu cez Kerberos, alebo bola schopná (v prípade webových služieb) preberať autentifikačné informácie od serveru, na ktorom beží. Toto platí pre takmer všetky služby, ktoré univerzita v súčasnosti poskytuje (viď tabuľka 1 na strane 55).

5.1.7 SPNEGO: Odolnosť voči zlyhaniu

Kerberos server je kľúčovým bodom, bez neho je riešenie kompletne znefunkčnené. Je preto nutné omedziť jeho výpadky na minimum a mať k dispozícii záložný server pre prípady závažnejších technických problémov. Výpadok na strane klienta môže znamenať dokonca neschopnosť prihlásenia sa na pracovnú stanicu, prípadne prístup iba k lokálnym zdrojom.

5.1.8 SPNEGO: Hierarchia dôveryhodnosti

Riešenie nepodporuje možnosť správať sa k rozličným službám odlišne, nie je to však potrebné. Jeho výhodou totiž je, že servery jednotlivých služieb vôbec

neprichádzajú do kontaktu s autentifikačnými údajmi používateľa (*credentials*). Preto za bežných okolností nie je potrebné zvažovať dôveryhodnosť jednotlivých serverov. Veľký problém však predstavuje otázka nedôveryhodných klientskych staníc. Jej riešenie by muselo zahŕňať existenciu dvoch navzájom samostatných účtov pre každého používateľa. Vhodným zavedením vzťahov dôvery medzi jednotlivými doménami je možné dosiahnuť použiteľnosť jedného z nich iba na prístup k menej dôležitým službám a druhého na prístup k obidvom, toto riešenie však predstavuje stratu transparentnosti a môže potenciálne viesť k bezpečnostným rizikám.

Jednou z možností, ako do riešenia zapojiť aj služby, ktoré Kerberos štandardne nepodporujú, je nasledujúce improvizované riešenie. Služba si od používateľa vyžiada jeho autentifikačné údaje a následne sa pokúsi u Kerberos serveru autentifikovať ako používateľ. V prípade úspechu prijíma údaje ako platné. Pri použití tohto riešenia je však nutné veľmi dobre zvážiť dôveryhodnosť serveru služby, ako aj dôležitosť jej zapojenia do riešenia. Ide totiž o závažné narušenie viacerých konceptov, na ktorých je bezpečnosť riešenia založená (prenos hesla po sieti, sprístupnenie používateľových údajov) a potenciálne bezpečnostné riziko.

5.1.9 SPNEGO: Odolnosť voči útokom

Opäť, Kerberos server je zraniteľným bodom (*single point of failure*), bezpečnostný prienik na ňom je veľmi závažný. Je preto nutné ho dobre zabezpečiť. Protokol samotný je navrhovaný s veľkým dôrazom na sťažovanie útokov na prebiehajúcu autentifikáciu, používateľovo heslo neputuje po sieti v žiadnej forme.

Protokol Kerberos má silné zabezpečenie proti útokom na prebiehajúcu komunikáciu ([21]). Bezpečnostným problémom však môže byť SPNEGO. Tento protokol totiž implementuje pôvodne obojsmernú autentifikáciu protokolu Kerberos iba jednosmerne, od používateľa k serveru. Používateľ teda nemá žiadnu istotu o identite serveru, s ktorým komunikuje. Tento potenciálne zneužiteľný moment je dôvodom, prečo je protokol v prehliadačoch štandardne vypnutý a vyžaduje explicitné povolenie

21 Kohl, Neuman. *RFC 1510* . 1993.

použitia pre každú doménu. Používateľ však stále nemá istotu o identite serveru, a teda táto implementácia ostáva zraniteľná voči nahradeniu servera útočníkom (napríklad zmenou informácií poskytovaných prostredníctvom DNS).

5.1.10 SPNEGO: zhrnutie

Výhodou tohto riešenia je jeho relatívna jednoduchosť – po splnení základných podmienok (správne nastavenie klientov a serverov, zaradenie do domény) nie sú potrebné ďalšie špecializované servery či inštalácia dodatočného softvéru. Problémom však je skutočnosť, že nastavenie klientov a zaradenie do domény je v mnohých prípadoch, ktoré potrebujeme zohľadniť, nerealistická požiadavka. JAS totiž nesmie vylučovať ani používateľov, pristupujúcich k univerzitným zdrojom z rozličných internetových kaviarní a podobných lokalít, kde nie je možné vykonať potrebné nastavenie. Navyše funkcionalita poskytovaná riešením je iba veľmi obmedzená, preto sa budeme sústreďovať na iné alternatívy.

5.2 Sun Java System Access Manager

Spoločnosť Sun zameriava značnú pozornosť na riešenie problému identity. Aspekt autentifikácie a autorizácie používateľov webových aplikácii zabezpečuje aplikácia *Access Manager* (ďalej AM). Jedná sa o aplikáciu s centralizovanou architektúrou, používajúcu agentov na serveroch pokrývaných služieb a cookies na prehliadačoch používateľov. Druh autorizačných možností závisí od pokrývanej služby, štandarne je však možné regulovať prístup podľa URL. Jednotlivé zdroje sú rozdelené do skupín a chránené *dozornými bodmi* (policy enforcement point), ktoré v spolupráci s centrálnym bodom povoľujú alebo odopierajú prístup. Skupinám používateľom sú prideľované oprávnenia na prístup ku jednotlivým zdrojom. Jedná sa teda o kontrolu prístupu na základe rol.

AM ponúka na výber rozličné možnosti autentifikačných mechanizmov. Kompletný zoznam je k dispozícii v dokumentácii projektu ([29]), my spomenieme hlavne

29 *Sun Java System Access Manager 7 2005Q4 Technical Overview*. 2005.

pre nás podstatné metódy Active Directory (teda Kerberos), HTTP Basic (teda meno a heslo), prípadne autentifikácia založená na certifikátoch. Informácie o používateľoch má štandardne uložené pomocou LDAP – ideálne je použitie produktu Sun Directory Server, pre ktorý je AM optimalizovaný. AM však dokáže však spolupracovať s ľubovoľnou implementáciou LDAP či databázy, ku ktorej sa pristupuje použitím SQL.

5.2.1 Sun: Jednoduchosť používania

Prvý pokus používateľa o prístup ku chránenému zdroju v rámci pracovného sedenia je odchytený agentom, nainštalovanom na serveri služby *(policy agent)*. Tento skontroluje prítomnosť cookie sedenia. Ak existuje a je platný, agent transparentne povolí prístup. V opačnom prípade používateľa presmeruje na centrálny server, ktorý mu vydá autentifikačný formulár.

Používateľ teda prichádza do kontaktu s AM aspoň raz – v prípade použitia rozličných druhov sedení, ktoré AM poskytuje, môže byť v ďalšom priebehu pracovného sedenia vyžadovaná re-autentifikácia.

5.2.2 Sun: Minimálna prípravná fáza

Prípravná fáza na strane serveru novozapájanej služby spočíva z inštalácie AM agenta a nastavenia používaných autentifikačných metód. Riešenie samotné nevyžaduje žiadne zásahy do klientskej stanice (za predpokladu, že používateľov prehliadač má povolené používanie cookies). Z výberu autentifikačného mechanizmu môže vyplynúť potreba dodatočného nastavovania či úpravy klientskej stanice (napr. v prípade protokolu Kerberos konfigurácia klienta a jeho pridanie do domény).

5.2.3 Sun: Mobilita riešenia

Jediná požiadavka, ktorú na klientskú stanicu kladie samotný Access Manager, je povolené používanie cookies. Toto v kombinácii s možnosťou nastaviť použitie náhradnej autentifikačnej metódy ak prvá nie je druhou stranou podporovaná umožňuje

do riešenia začleniť aj počítače bez možnosti úpravy klienta.

5.2.4 Sun: Podpora OS

Access Manager je dostupný vo verziách pre Solaris, Linux a Windows (Server 2003, XP). Verzia pre Windows je spoločnosťou označená ako "vo vývoji" (development only), preto ju neodporúčame zvažovať. Verzia pre Linux je autormi uvádzaná ako stabilná, pre ideálne výsledky sa však odporúča nasadenie verzie pre Solaris, ktorá je najstabilnejšia. Na stránkach spoločnosti Sun je k dispozícii množstvo agentov pre rozličné služby, s možnosťou tvorby vlastných. Na strane klienta je samotné riešenie nezávislé od operačného systému, niektoré autentifikačné metódy ale nemusia byť vždy k dispozícii.

5.2.5 Sun: Jednoduchosť rozširovania

Rovnako ako pri iných centralizovaných riešeniach, pridávanie nových služieb nevyžaduje zmenu existujúcich služieb. Ak pre službu neexistuje AM agent, je nutné ho dotvoriť s využitím štandardizovaných volaní.

Access Manager poskytuje zaujímavú možnosť nasadiť viacero autentifikačných serverov súčasne a použiť medzi nimi vyrovnávanie pracovnej záťaže *(load balancing)*. Toto dáva riešeniu nielen lepšiu škálovateľnosť, ale aj vyššiu odolnosť voči výpadkom.

5.2.6 Sun: Maximálne pokrytie súčasných zdrojov

Vďaka architektúre riešenia a množstvu dostupných agentov by malo byť zapojenie súčasných zdrojov do riešenia nenáročné. Niektoré z univerzitou používaných aplikácii patria k menej rozšíreným, preto by pre ne bolo nutné použiť improvizované riešenie zapojenia alebo vytvoriť agentov svojpomocne (viď tabuľka x na strane xx). Nepredpokladáme však zásadnejšie potiaže.

5.2.7 Sun: Odolnosť voči zlyhaniu

Na zabránenie vzniku preťaženia je možné použiť funkcie na vyrovnávanie záťaže – Access Manager má zabudovanú podporu pre súčasné použitie viacerých paralelných serverov. Tieto funkcie je zároveň možné použiť ako ochranu proti výpadkom. Pri ich použití je ale nutné na serveroch AM použiť tzv. "sticky sessions" - nastavenie zaručujúce, že ak sa v priebehu pracovného sedenia už používateľ autentifikoval, každý ďalší dotaz bude nasmerovaný k tomu serveru, u ktorého prebehla prvotná autentifikácia. V prípade väčšieho pracovného vyťaženia tohto serveru by totiž mohol byť dotaz presmerovaný na iný server, na ktorom neexistuje záznam o používateľovom sedení. Tým by vznikla nutnosť opäť sa autentifikovať.

5.2.8 Sun: Hierarchia dôveryhodnosti

Access Manager poskytuje možnosť udeľovať niekoľko druhov používateľských sedení – *základné* (používateľ je autentifikovaný iba pre jednu službu), *single sign-on* (používateľ je autentifikovaný pre všetky služby v jednej doméne) a *cross-domain* (používateľ je autentifikovaný pre všetky služby vo viacerých doménach). S využitím týchto rozličných druhov sedenia v závislosti od druhu služby a spôsobu autentifikácie používateľa môžeme vytvoriť jednoduchú hierarchiu služieb – rozdelením služieb do domén podľa dôležitosti. Do domény nekritických služieb potom môžeme povoliť vstup aj používateľom, ktorí sa nedokážu autentifikovať štandardným mechanizmom a boli presmerovaní na alternatívnu, slabšiu metódu (príklad: chýbajúca podpora pre Kerberos na klientskej stanici s následnou autentifikáciou sa pomocou mena a hesla). Dostanú však iba sedenie platné na danú doménu – pri pokuse o prístup ku službám mimo nej bude od nich opäť požadovaná autentifikačná výmena, ktorá môže mať nastavené iné správanie (napríklad po zlyhaní výmeny Kerberos zamietnuť prístup a neposkytovať alternatívu).

5.2.9 Sun: Odolnosť voči útokom

Odolnosť riešenia závisí hlavne od použitých autentifikačných metód, ich

nastavenia a zabezpečenia autentifikačného serveru. Prípadný prienik bezpečnosti môže byť veľmi závažný, pretože riešenie zabezpečuje nielen autentifikáciu, ale aj autorizáciu. Je nutné zabezpečiť servery, ako aj komunikáciu medzi nimi. Taktiež je vhodné dobre zabezpečiť sklad informácií AM v prípade, že sa nenachádza na tom istom fyzickom serveri ako AM.

5.2.10 Sun: zhrnutie

Riešenie od spoločnosti Sun má oproti ostatným zvažovaným alternatívam výhodu v tom, že navyše k autentifikácii poskytuje aj autorizáciu, zjednodušenie správy používateľských účtov, a prípadné jednoduché rozšírenie o federované identity v budúcnosti. Ďalšou výhodou je bezproblémová spolupráca s produktom Sun Identity Manager, ktorý by sme mohli v projekte nasadiť na zabezpečenie správy používateľských účtov v službách mimo sféru webového SSO.

Nevýhodami sú naopak preferencia menej obvyklého operačného systému (Solaris), ako aj neprístupnosť komunity vývojárov ako voľne dostupného zdroja pomoci (podpora je k dispozícii ako platená služba). Žiaľ, tieto produkty sú zamerané hlavne s dôrazom na iné technológie spoločnosti Sun. Pre optimálne fungovanie je vhodné, aby údaje o používateľoch boli uložené na LDAP serveri (najlepšie Java Directory Server).

5.3 CoSign

Projekt CoSign vystupuje do popredia medzi riešeniami, založenými na cookies hlavne vďaka šírke podporovaných autentifikačných možností**.** Umožňuje používať súčasne viacero spôsobov (meno a heslo, certifikát, Kerberos). Jednotlivé služby môžu na prístup ku zdrojom vyžadovať niektorý konkrétny, alebo udeliť práva podľa toho, ktorý spôsob autentifikácie používateľ použil. Taktiež môžu vyžadovať *re-autentifikáciu* (opätovné autentifikovanie sa bez ohľadu na predošlý úspešný priebeh) či multifaktorovú *autentifikáciu* (autentifikovanie sa dvomi či viacerými spôsobmi

súčasne)([19]).

Tieto vlastnosti riešeniu umožňujú jemnejšiu granularitu kontroly prístupu a tým aj zvýšenú bezpečnosť riešenia – môžeme totiž z rizikových prístupových bodov (internetové kaviarne) používateľom poskytnúť autentifikáciu menom a heslom, ale sprístupniť im iba menej kritické služby. Na prístup ku kľúčovým zdrojom môže systém vyžadovať bezpečnejšiu metódu autentifikácie, napríklad cez Kerberos.

Ďalšiu zaujímavú funkcionalitu poskytuje modul CoSign Friend. Automatizované vytváranie dočasných hosťovských účtov a ich autentifikovanie cez e-mail poskytuje možnosť bližšie monitorovať činnosť jednotlivých hosťov v systéme bez zvyšovania pracovnej záťaže na administrátorov.

5.3.1 CoSign: Jednoduchosť používania

CoSign nie je pre používateľa plne transparentný – pri prvom pokuse o prístup k chráneným zdrojom prichádza do kontaktu s autentifikačným serverom. Tento mu ponúkne formulár s možnosťami autentifikácie, ktoré sú systémom podporované. Používateľ si vyberie, ktorou metódou sa chce autentifikovať (môže použiť aj viac spôsobov naraz). Pri ďalšej práci s chránenými zdrojmi už obvykle nie je potrebné autentifikačný proces opakovať.

Filtre jednotlivých chránených zdrojov môžu byť nastavené tak, aby vyžadovali konkrétne (jednu či viac) z ponúkaných možností autentifikácie, prípadne konkrétny počet rôznych metód. Ak sa používateľ snaží pristupovať k zdroju, ktorého požiadavky nespĺňa, prichádza opäť do kontaktu s autentifikačným formulárom, podobne ako na začiatku práce. Toto taktiež platí o zdrojoch, ktoré majú nastavenú požiadavku re-autentifikácie.

19 Craig, Craig. *Cosign Multi-Factor Specification (draft 6)*. 2006

5.3.2 CoSign: Minimálna prípravná fáza

Náročnosť prípravnej fázy pre klientskú stanicu záleží od konkrétnych použitých autentifikačných mechanizmoch. Ak plánujeme v priebehu pracovného sedenia použiť iba autentifikáciu menom a heslom, nie je potrebné vykonávať na klientskej stanici žiadne úpravy (za predpokladu, že prehliadač má zapnuté používanie cookies). Použitie certifikátov či autentifikácie cez Kerberos je náročnejšie na vykonané zmeny, vyžaduje totiž správne nastavenia klientských staníc.

5.3.3 CoSign: Mobilita riešenia

Miera mobility závisí od použitých (prípadne preferovaných, najslabších nutných) autentifikačných mechanizmov – pri použití mena a hesla je mobilita ideálna, v prípade komplexnejších (bezpečnejších) mechanizmov vzniká rovnaký problém ako pri predošlom kritériu – nutnosť úpravy a nastavovania klientskej stanice znižuje mobilitu.

Ako nástroj na zvýšenie mobility riešenia je možné chápať aj vyššie spomínaný modul CoSign Friend – vytváranie dočasných hosťovských účtov pomovou e-mailových pozvánok umožňuje odstránenie anonymity hosťovského prístupu bez dodatočného pracovného zaťaženia administrátorov ([18]). Toto je možné využiť na to, aby si používateľ sám vytvoril dočasný účet, ktorý použije pri práci z nedôveryhodnej klientskej stanice.

5.3.4 CoSign: Podpora OS

Projekt CoSign poskytuje agentov pre servery Apache (1.3 a 2.0) a IIS (5.0 a vyššie). Taktiež existuje balík JavaCoSign, ktorý obsahuje nástroje na previazanie JAAS (*Java Authentication and Authorization Service*) a projektu CoSign - teda jednak použitie JAAS ako autentifikačného mechanizmu pre CoSign, ale aj naopak, prihlásenie sa do JAAS pomocou CoSign. Jednotlivé autentifikačné mechanizmy, použiteľné v

18 Craig, Craig. *Leveraging Guest Accounts for Ubiqutous Web Sign-On System Acceptance.* 2005

kombinácii s projektom, majú taktiež dobré zastúpenie na rozličných platformách (viď kapitola 3).

5.3.5 CoSign: Jednoduchosť rozširovania

Ako pri ostatných centralizovaných riešeniach nie je na zastrešenie novej služby potrebná modifikácia ostatných, už zastrešených služieb. Bližšie podrobnosti závisia od použitých autentifikačných mechanizmov. Pri najbežnejších alternatívach (Kerberos, meno a heslo) stačí na serveri, prevádzkujúcom službu, nainštalovať a nastaviť filter systému CoSign.

5.3.6 CoSign: Maximálne pokrytie súčasných zdrojov

CoSign vznikol na University of Michigan (ďalej UM) rozvojom predošlého miestneho riešenia problematiky single sign-on, preto sa autori pri jeho tvorbe zamerali na pokrytie zdrojov, vyskytujúcich sa na UM. Vďaka podobnostiam s UK, čo sa rozsahu a používaných technológii týka, dokáže projekt pokryť takmer všetky z nami používaných služieb. V niektorých oblastiach sa však na UM používajú iné softvérové riešenia – napríklad *systém na správu obsahu* (content management system). Využívajú produkt Drupal, pre ktorý v spolupráci s jeho autorským tímom vytvorili modul na uľahčenie interoperability. Z tohoto dôvodu niektoré z našich služieb nie sú priamo podporované (viď tabuľka 1 na strane xx), nepredpokladáme však zásadné problémy pri ich zastrešovaní.

5.3.7 CoSign: Odolnosť voči zlyhaniu

Jedná sa o centralizované riešenie, trpí teda problémom, ktoré majú všetky riešenia tohto druhu spoločné – ak zlyhá centrálna autorita, je riešenie znefunkčnené. CoSign poskytuje nástroje na replikáciu centrálneho serveru. Na druhú stranu, zlyhanie komponentov riešenia zabezpečujúcich jeden autentifikačný mechanizmus je menej závažné, vzhľadom na možnosť použitia viacerých mechanizmov.

5.3.8 CoSign: Hierarchia dôveryhodnosti

Ľahkosť implementácie hierarchie dôveryhodnosti je jednou z hlavných devíz riešenia CoSign. Autentifikačný server umožňuje používateľom autentifikovať sa viacerými rozličnými metódami naraz. Súčasťou informácie o používateľovom pracovnom sedení je aj údaj o tom, ktorými metódami sa autentifikoval. Filter na chránenom zdroji umožňuje rozličné nastavenia. Na prístup ku konkrétnemu zdroju môže vyžadovať autentifikáciu aspoň jednou ľubovoľnou metódou, jednou konkrétnou metódou, alebo viacerými metódami ([19]). V prípade, že používateľ nastavené požiadavky nespĺňa, je automaticky presmerovaný na autentifikačný server, aby sa mohol dodatočne autentifikovať. Filter môže taktiež vyžadovať *re-autentifikáciu* (prijímať za platné iba požiadavky od používateľov, ktorí sú čerstvo autentifikovaní).

Tieto vlastnosti nám umožňujú vybudovať solídnu hierarchiu dôveryhodnosti, kde je na prístup k nekritickým zdrojom postačujúca relatívne slabá autentifikačná metóda (meno a heslo), ale s pribúdajúcou dôležitosťou služby sa zvyšujú aj požiadavky kladené na používateľa.

5.3.9 CoSign: Odolnosť voči útokom

Flexibilnosť riešenia, ktorá je veľkou výhodou pri predošlom kritériu, sa tu ukazuje ako komplikácia. Odolnosť systému voči útokom totiž závisí nielen od použitých mechanizmov, ale aj od vhodného návrhu hierarchie. Narušenie bezpečnosti pri autentifikácii jedným mechanizmom vôbec nemusí znamenať prienik do všetkých služieb, pokrývaných riešením.

5.3.10 CoSign: zhrnutie

V prípade tohto produktu sa jedná o riešenie, založené na použití cookies, je nutné, aby klient mal povolené ich používanie. Toto by však nemal byť problém ani vo

19 Craig, Craig. *Cosign Multi-Factor Specification (draft 6)*. 2006

veľmi obmedzenom prostredí, keďže cookies patria ku štandardom dnešného webu. Nevýhodou riešenia je, že sa jedná iba o čistú autentifikáciu – jeho nasadením teda nezískame žiadne uľahčenie správy používateľských účtov či autorizačné možnosti.

5.4 Vzájomné porovnanie

Riešenie SPNEGO je jednoduchým, základným využitím protokolu Kerberos na dosiahnutie single sign-on. Všetka funkcionalita, ktorú poskytuje, pochádza z tohoto protokolu, zároveň však riešenie neponúka všetky vlastnosti (obojsmernú autentifikáciu, šifrovanie komunikácie). Vo zvyšných dvoch alternatívach riešenia sa taktiež ponúka možnosť tento protokol využiť ako autentifikačný mechanizmus, čo im umožňuje pridať výhody riešenia SPNEGO navyše k vlastným. Preto je toto riešenie pre nás zaujímavé z akademického hľadiska, na praktické nasadenie však pre nás neprichádza do úvahy.

Pri vzájomnom porovnávaní riešení Sun a CoSign je rozhodovanie oveľa náročnejšie, každé z nich má oproti "súperovi" výhody aj nevýhody. Obidva ponúkajú riešenia pre tú istú sadu problémov, ťažisko každého z nich však leží niekde inde.

CoSign sa upriamuje výlučne na problematiku autentifikácie. Podporuje široké spektrum rozličných možností a alternatív zaujímavých hlavne z bezpečnostného hľadiska. Problematike autorizácie sa nevenuje vôbec, prenecháva ju na jednotlivé služby. Jeho ďalšou výhodou pre nás je jeho pôvod v akademickom prostredí – ide o projekt s otvoreným zdrojovým kódom s aktívnou komunitou, ktorá sa venuje jeho vývoju a nasadeniu v prostredí, ktoré je podobné našim podmienkam.

Naproti tomu Sun Access Manager vníma autentifikáciu ako úkon, ktorý je nutné vykonať pred činnosťami, na ktorých leží jeho hlavná pozornosť (autorizácii, resp. správe používateľských kont). Preto neponúka natoľko rozsiahle možnosti ako CoSign, sú však stále pomerne široké. Jeho hlavná sila však leží v autorizácii - či už v rolami riadenej kontrole prístupu, ktorú ponúka, alebo v jednoduchej spolupráci s

ostatnými produktami spoločnosti Sun, zameranými na túto oblasť (Identity Manager, Federation Manager) a funkcionalitou, ktorú tieto poskytujú.

	SPNEGO	Sun	CoSign
Webové služby			
Typo3	Áno*	Áno*	Áno*
Virtua	Úpravy	Úpravy	Úpravy
Moodle	Áno	Áno*	Áno*
VoIP Billing	Áno**	Áno**	Áno**
Java Application Server	Áno*	Áno	Áno
Iné webové aplikácie	Áno	Áno	Áno
Ostatné služby			
RADIUS	Áno	Áno	Áno
Windows siete	Áno	Áno***	Áno***
Windows mail	Áno	Áno***	Áno***
Linux siete	Áno	Áno***	Áno***
Linux mail	Áno	Áno***	Áno***

Tabuľka 1 – Zapojenie existujúcich služieb

Legenda pre tabuľku 1:

Áno	Služba je jednoducho zapojiteľná do riešenia
Nie	Služba neumožňuje úpravy, nutné pre jej zapojenie
Úpravy	Služba je zapojiteľná iba po rozsiahlych úpravách (zmena samotnej služby)
*	Po drobných úpravách služby alebo použití dodatočných komponentov riešenia (použitie zásuvného modulu, vytvorenie agenta)
**	Aplikácia je vo vývoji, spolupráca s kolektívom autorov by mala zaručovať kompatibilitu
***	V závislosti od použitého autentifikačného mechanizmu

Tabuľka 1 sumarizuje mieru zložitosti zapojenia jednotlivých služieb do zvažovaných riešení. Hlavnú pozornosť venujeme kategórii "webové služby", kategóriu "ostatné služby" uvádzame iba pre orientáciu v kontexte projektu JAS. Napojenie služieb Typo3 a Moodle na jednotnú autentifikáciu pozostáva z nasadenia zásuvného autentifikačného modulu na hostiteľský server a jeho následného nastavenia.

Sun Java System Application Server bude zapojený nasadením agenta (Policy Agent 2.2 for Sun Java System Application Server) resp. spustením aplikácie v jazyku Java (JavaCoSign), ktorá zabezpečí komunikáciu s centrálnym bodom. Ostatné webové

aplikácie budú zapojené prenechaním autentifikácie na hostiteľský server a jeho následnou konfiguráciou (nainštalovaním agenta vybraného riešenia).

Presný spôsob zapojenia aplikácie na VoIP Billing nie je v tomto momente možné uviesť, najpravdepodobnejším riešením ale je prenechanie autentifikácie na hostiteľský server.

Najväčší problém spôsobuje zapojenie aplikácie Virtua – bude pravdepodobne nutné vykonať rozsiahle úpravy samotnej aplikácie za účelom dosiahnutia podpory vybraného riešenia. Čo sa náročnosti uvedenia do prevádzky týka, riešenia CoSign a Sun sú ekvivalentné – Sun poskytuje širší výber existujúcich agentov, v prípade nutnosti dotvorenia agenta však nie je k dispozícii podpora komunity.

Finálne rozhodnutie o tom, ktoré z týchto dvoch riešení nasadiť bude záležať hlavne od toho, či považujeme za dôležitejšiu robustnosť a bezpečnosť ponúkanej autentifikácie, alebo poskytovanie ďalšej funkcionality. Pre administrátorov ponúka jedno riešenie lepšiu bezpečnosť a možnosť lepšie vrstvenej hierarchie, druhé zjednodušenie práce. Pre bežného používateľa sú obidve alternatívy v podstate identické, nie je medzi nimi pre neho podstatnejší rozdiel, čo sa spôsobu používania týka.

6 Záver

Cieľom projektu JAS UK je nasadenie jednotného autentifikačného riešenia na Univerzite Komenského za účelom zvýšenia používateľského a administrátorského pohodlia. Pri vhodnej implementácii sa taktiež dosiahne zvýšenie bezpečnosti. Táto práca, ktorá je jeho súčasťou, sa venovala prieskumu súčasnej situácie v oblasti autentifikácie používateľov webových aplikácií na UK, prieskumu možností riešenia a ich následnej analýze. Navrhli sme Sun Java System Access Manager a CoSign ako vhodné alternatívy riešenia a zhrnuli kritériá pre finálne rozhodnutie medzi nimi.

Po úspešnom zakončení implementácie riešenia ako súčasti projektu JAS sa ako možnosť ďalšieho rozpracovania ponúka oblasť federovanej identity. Zdieľanie autentifikačných údajov s partnerskými organizáciami môže byť pre používateľov univerzitných služieb ďalším zjednodušením práce. Taktiež v prípade voľby riešenia CoSign sa ako ďalšie pokračovanie projektu javí problematika autorizácie a automatizovanej správy používateľských účtov.

Veríme, že táto práca ako aj celý projekt JAS prispeje ku skvalitneniu služieb, poskytovaných študentom aj zamestnancom univerzity. Dúfame, že ani v budúcnosti nezaostane UK za pokrokom v oblastiach správy identity a počítačovej bezpečnosti.

Zoznam bibliografických odkazov

1. Mederly, P. *Jednotná autentifikácia používateľov informačných technológií*. Návrh projektu rozvoja používania informaačných technológií na UK č. CIT 10/2006. Interný materiál UK, marec 2006.
2. Windley, P. *Digital Identity*. O'Reilly, 2005. 254 s. ISBN 0-596-00878-3.
3. Guide to CISSP, Information Security Certification. *Access Control Concepts* [online]. 2007. Článok. Formát HTML. Dostupné na internete: <http://www.guidetocissp.com/2007/03/access-control-concepts-aaa-access.html>
4. Pickard, T. *Identity Federation* [online]. 2006. Článok. Formát HTML. Dostupné na internete: <http://www.ebcvg.com/articles.php?id=1015>
5. Milgate, A. *The Identity Dictionary* [online]. 2006. Slovník. Formát HTML. Dostupné na internete: <http://identityaccessman.blogspot.com/>
6. Dunne, C. *Build and Implement a Single Sign-On Solution* [online]. 2004. Článok. Formát HTML. Dostupné na internete: <http://www.developertutorials.com/tutorials/java/single-sign-on/page2.html>
7. Risto, S., Ritter, H: *Single Sign On – The main focus is on the customer* [online]. 2002. Článok. Formát PDF. Dostupné na internete: <http://www.novosec.com/documents/eCommerce_SingleSignOn_en.pdf>
8. *Výročná správa Univerzity Komenského v Bratislave za rok 2005*. Máj 2006. Dostupné aj na internete (formát PDF): <http://www.uniba.sk/fileadmin/user_upload/editors/subory/spravy/sprava_2005.pdf>
9. Typo3 Content Management System: *Feature List* [online]. Dokumentácia produktu. Formát HTML. Dostupné na internete: <http://typo3.com/Feature_list.1243.0.html>
10. *Virtua Profiler* : *Global Settings*. 2005. Dokumentácia produktu (verzia 46). Formát PDF.
11. *GSS API/Kerberos5 Authentication* [online]. Dokumentácia produktu (Java 2). Formát HTML. Dostupné na internete: <http://java.sun.com/products/jndi/tutorial/ldap/security/gssapi.html>
12. *MoodleDocs* [online]. Dokumentácia k produktu. Formát HTML. Dostupné na internete: <http://docs.moodle.org/en/Main_Page>

13. Zagiba, M. *Návrh riešenia jednotnej autentifikácie a autorizácie na Univerzite Komenského*. Interný materiál Centra informačných technológií UK. 2007 Formát PDF
14. FreeRADIUS Wiki. *Authentication* [online]. Dokumentácia produktu. Formát HTML. Dostupné na internete: <http://wiki.freeradius.org/Authentication>
15. Stein, L., Stewart, J. *The WWW Security FAQ – Client Side Security* [online]. 2003. Článok. Formát HTML. Dostupné na internete: <http://www.w3.org/Security/Faq/wwwsf2.html>
16. *How Pubcookie Works* [online]. 2003. Dokumentácia produktu. Formát HTML. Dostupné na internete: <http://pubcookie.org/docs/how-pubcookie-works.html>
17. Craig et al. *The Cosign Web Single Sign-On Scheme (draft 4)*. 2006. Dokumentácia produktu. Formát RTF. Dostupné na internete: <http://www.umich.edu/~umweb/software/cosign/media/cosignscheme2006a.rtf>
18. Craig, W., Craig, J. *Leveraging Guest Accounts for Ubiqutous Web Sign-On System Acceptance.* 2005. Dokumentácia produktu. Formát PDF. Dostupné na internete: <http://www.educause.edu/ir/library/pdf/EDU05135.pdf>
19. Craig, W., Craig, J. *Cosign Multi-Factor Specification (draft 6)*. 2006. Dokumentácia produktu. Formát PDF. Dostupné na internete: <http://www.umich.edu/~umweb/software/cosign/media/multifactor_spec_draft6.pdf>
20. Nichols, A. *Kerberos and Access Manager Single Sign-On* [online]. 2005. Článok. Formát HTML. Dostupné na internete: <http://blogs.sun.com/hippie/entry/kerberos_and_access_manager_single>
21. Kohl, J., Neuman, C. *RFC 1510 : The Kerberos Network Authentication Service (V5).* 1993. Špecifikácia. Formát TXT. Dostupné na Internete: <http://www.faqs.org/ftp/rfc/rfc1510.txt>
22. Shinder, D. *How to use Kerberos Authentication in a Mixed (Windows and UNIX) Environment* [online]. 2006. Článok. Formát HTML. Dostupné na internete: <http://www.windowsecurity.com/articles/Kerberos-Authentication-Mixed-Windows-UNIX-Environment.html>
23. Ellison, C., Schneier, B. *Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure* In *Computer Security Journal*. 2000, vol. 16, no. 1, p. 1-8. Dostupné aj na internete: <http://www.schneier.com/paper-pki.pdf>
24. Findlay, A. *Security with LDAP* [online]. 2002. Článok. Formát HTML.

Dostupné na internete: <http://www.skills-1st.co.uk/papers/security-with-ldap-jan-2002/security-with-ldap.html>

25. Hill, J. *An Analysis of the RADIUS Authentication Protocol* [online]. 2001. Článok. Formát HTML. Dostupné na internete: <http://www.untruth.org/~josh/security/radius/radius-auth.html>
26. *Integrated Windows Authentication (IIS 6.0)* [online] In *Microsoft TechNet.* Článok. Formát HTML. Dostupné na internete: <http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/523ae943-5e6a-4200-9103-9808baa00157.mspx?mfr=true>
27. Zhu, L. et al. *RFC 4178 : The Simple and Protected Generic Security Service Application Program Interface(GSS-API) Negotiation Mechanism.* 2005. Špecifikácia. Formát TXT. Dostupné na internete: <http://tools.ietf.org/rfc/rfc4178.txt>
28. Jaganathan, K. et al. *RFC 4559 : SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows.* 2006. Špecifikácia. Formát TXT. Dostupné na internete: <http://tools.ietf.org/rfc/rfc4559.txt>
29. *Sun Java System Access Manager 7 2005Q4 Technical Overview.* 2005. Dokumentácia produktu. Formát PDF. Dostupné na internete: <http://docs.sun.com/app/docs/doc/819-2135>