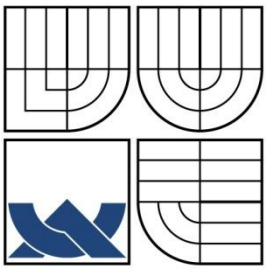
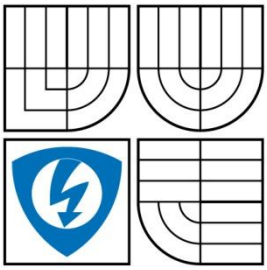


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF

IDENTITY MANAGEMENT

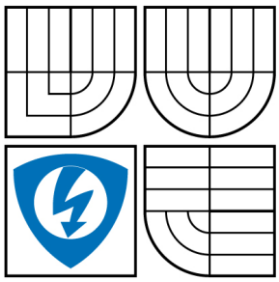
IDENTITY MANAGEMENT

DIPLOMOVÁ PRÁCE
MASTER´S THESIS

AUTOR PRÁCE BC. DANIEL KEFER
AUTHOR

VEDOUCÍ PRÁCE ING. TOMÁŠ PELKA
SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Daniel Kefer	***ID:***	88509
Ročník:	2	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Identity management

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je v prvé řadě zmapovat možnosti centralizované správy identit a SSO (Single SignOn). Student prozkoumá tuto oblast se zaměřením na open source projekty. Druhou část projektu představuje analýza, návrh implementace a konfigurace funkčního propojení fakultní databáze uživatelů, jenž je postavena na Novell Netware, s adresářovým serverem LDAP v kombinaci s autentizačním serverem Kerberos. Součástí práce bude podrobný popis a zdůvodnění navržené konfigurace.

DOPORUČENÁ LITERATURA:

[1] TODOROV , Dobromir . Mechanics of User Identification and Authentication : Fundamentals of Identity Management. 1st edition. [s.l.] : Auerbach Publications, 2007. 760 s. ISBN 978-1420052190.

[2] WINDLEY, Phillip . Digital Identity. [s.l.] : O'Reilly Media, Inc., 2005. 254 s. ISBN 978-0596008789.

[3] BIRCH, David . Digital Identi Management. [s.l.] : Ashgate Publishing, 2007. 220 s. ISBN 978-0566086793.

Termín zadání:	9.2.2009	***Termín odevzdání:***	26.5.2009

Vedoucí práce: Ing. Tomáš Pelka

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práve tretích osob, zejména nesmí zasahovat nedovoleným zpusobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Abstrakt

Diplomová práce je rozdělena do dvou částí. V první z nich je na teoretické úrovni zpracována oblast Identity Managementu. Jsou zde popsány jednotlivé oblasti Identity Managementu, kam patří autentizace, autorizace a audit. Dále se tato část zabývá problematikou Single Sign-On, tedy využití jednoho přihlášení k přístupu k více nezávislým systémům či službám.

Ve druhé části, která je stěžejním prvkem této diplomové práce, bylo přistoupeno k praktické realizaci projektu v prostředí infrastruktury Ústavu telekomunikací na Fakultě elektrotechniky a komunikačních technologií VUT v Brně, jehož cílem bylo implementovat v rámci počítačové učebny sloužící k výuce OS Linux prostředí pro centrální autentizaci a Single Sign-On pro studenty a vyučující za využití výhradně open source technologií. Hlavními prvky tohoto řešení je OS Linux Debian, protokol pro bezpečnou autentizaci Kerberos a LDAP server OpenLDAP. Pro demonstraci Single Sign-On byla zvolena služba NFS, která slouží k přístupu k souborům po síti a díky které jsou uživatelé nezávislí na použití konkrétní stanice pro práci. Správa uživatelů a jejich import z centrální databáze FEKT VUT je řešena pomocí skriptů vyvinutých v jazyce Python.

Dále bylo za pomocí technologií Apache, PHP a MySQL vytvořeno front-endové rozhraní pro administrátora systému, kde má možnost snadno zkoumat a vyhodnocovat podezřelé bezpečnostní události v síti, o kterých je rovněž zpravován v reálném čase pomocí emailu.

Vytvořený projekt je koncipován jako bezpečnostní platforma pro správu sítě, lze tedy na základě vyplynuvších potřeb například zpřístupnit další služby pro Single Sign-On či přidávat další mechanismy pro vyhodnocování podezřelých událostí v síti.

Klíčová slova

Identity management, Single Sign-On, Linux, Kerberos, LDAP, audit, bezpečnost

Abstract

The master thesis is divided into two parts. In the first part, identity management is described on theoretical basis. Particular domains of identity management including authentication, authorization and audit are explained as well as Single Sign-On concept, i.e. using single credentials and entering them just once for access to multiple independent systems or services.

In the second part, which forms the main part of this thesis, a practical project was implemented on the infrastructure of the Department of Telecommunications within the Faculty of Electrical Engineering and Communication, Brno University of Technology. The goal of this project was to create an environment for central

authentication and Single Sign-On using only open source technologies within a computer laboratory used for teaching OS Linux. The project is based on OS Linux Debian, Kerberos as a protocol for secure authentication and LDAP server OpenLDAP. For the Single Sign-On demonstration, NFS services for accessing data on the network were chosen. Using NFS services, users can sign-on to any workstation and access all their data. Administration of users and their import from central FEEC databases was implemented using scripts developed in Python. Next, using Apache, PHP and MySQL, a front-end audit interface for the network administrator was developed in order to inspect and evaluate security events in the network. Messages about suspicious events are delivered to administrator's mailbox in real time. The project is intended as a security platform which means that other services can be implemented for Single Sign-On as well as new mechanisms for evaluation of suspicious events.

Keywords

Identity Management, Single Sign-On, Linux, Kerberos, LDAP, Audit, Security

Citace

KEFER, D. *Identity management.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 68 s. Vedoucí diplomové práce Ing. Tomáš Pelka.

Prohlášení o původnosti práce

Prohlašuji, že svou diplomovou práci na téma „Identity Management“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení *§* 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....
podpis autora

Poděkování

Děkuji vedoucímu své diplomové práce Ing. Tomáši Pelkovi, doktorandovi na Fakultě elektrotechniky a komunikačních technologií Vysokého učení technického v Brně, za velmi kvalifikované vedení diplomové práce a cenné rady a odborné konzultace při řešení praktického projektu.

V Brně dne ....

podpis autora

Obsah

Seznam obrázků

Úvod

Mohutný přenos služeb různých institucí pro přístup on-line s sebou přinesl závažný problém, kterým je potřeba spárování fyzické identity s identitou virtuální. Že tento problém není snadno řešitelný, dokazují četné statistiky pojednávající o krádeži identity v prostředí internetu a informačních systémů.

Smyslem této práce je představení identity managementu, oblastí, které řeší, protokolů a systémů, které používá, a následně praktické realizace projektu v prostředí infrastruktury VUT, jehož cílem je vytvořit prostředí pro centrální autentizaci uživatelů a Single Sign-On. Dále je kladen důraz na možnosti auditu bezpečnostních událostí v síti.

1. Identity Management – teoretické základy

1.1. Definice a zaměření identity managementu

Termín „Identity Management“ bývá definován různými způsoby, zde jsou některé konkrétní definice:

„Identity Management spočívá v identifikaci autorizovaných uživatelů a jejich zavedení do systému, který je používán ke spravování informací o jejich identitě.“[1]

„Identity Management je sada procesů, které umožňují autentizaci procesů náležících k identitě.“ [2]

“Vytvoření flexibilních definicí pro jednotlivce a skupiny, pomocí kterých je možné autentizovat uživatele a povolit různé úrovně oprávnění v závislosti na používané službě.“ [3]

.„Identity Management je management životního cyklu identity entit.“ [4]

Ať už si vybereme kteroukoli z těchto definicí, je zřejmé, že budeme věnovat problematice spárovaní reálné a elektronické identity, přiřazování uživatelských práv k těmto identitám a dalšími souvisejícími oblastmi.

S touto problematikou jsou spojeny následující pojmy:

Za **identitu** lze považovat unikátní soubor vlastností, na základě kterých je jednoznačně definována konkrétní entita systému (tedy například uživatel).

Identifikace je proces rozpoznání deklarované identity autoritou.

Autorita je pak prvek systému nadřazený ostatním prvkům, který má v rámci tohoto systému rozhodovací pravomoce v definované oblasti.

Po formální stránce lze definovat tři oblasti identity managementu, kterými jsou:

- autentizace – ověření deklarované identity
- autorizace – stanovení práv k různým akcím pro danou identitu
- audit – vytváření a udržování záznamů o akcích provedených v kontextu identit

Jednotlivé oblasti budou podrobněji představeny v následujících kapitolách.

1.2. Autentizace

Jak již bylo řečeno, autentizace se řeší problém ověření identity uživatele vzhledem k použitému systému. Forma autentizace se rozlišuje podle tzv. autentizačních faktorů, neboli podle informací, za pomocí kterých uživatel deklaruje svoji identitu.

Existují tři druhy těchto faktorů:

- faktor vlastnictví, neboli něco, co uživatel má, čímž může být např. mobilní telefon, GRID karta [18], Smart karta [19], USB token [20], certifikát a podobně.
- faktor znalosti, neboli to, co uživatel ví. Tím jen nejčastěji heslo, fráze, nebo PIN (Personal Identification Number) [21].

- faktor existence, neboli to, čím uživatel je, případně to, co uživatel dělá. Obvykle se v rámci této oblasti používají otisky prstů, struktura duhovky, DNA, podpis nebo hlas.

Podle povahy informací, ke kterým uživatel v rámci systému přistupuje, se uplatňuje buď jednofaktorová, nebo vícefaktorová autentizace (příkladem vícefaktorové autentizace může být aplikace elektronického bankovnictví, kam se uživatel přihlašuje nejdříve zadáním uživatelského jména a hesla a dále tokenem, který mu přijde pomocí SMS na jeho mobilní telefon).

Nejčastěji je autentizace řešena pomocí uživatelského jména a hesla, kdy platnost těchto údajů je ověřena oproti cílovému systému. Přenos těchto údajů mezi klientem a systémem lze řešit více způsoby, mezi které patří:

- Plain text authetication: Nejjednodušší způsob, kdy jméno a heslo je přímo odesláno bez jakýchkoli úprav k cílovému systému. Tento způsob je například použit ve velmi rozšířené službě FTP, ale v nově implementovaných systémech již není příliš používán ze zjevných bezpečnostních důvodů.
- Přenos hashovaných údajů: S tímto přístupem se lze často setkat zejména u webových aplikací, kdy je heslo zadané klientem nejprve zahashováno pomocí některého z rozšířených hashovacích algoritmů (MD5, SHA-1, SHA-2). Tento přístup má výhodu, že ani samotná aplikace nezná heslo uživatele, pro ukládání hesel se navíc často využívají tzv. salted hashes, tzn. k samotnému heslu je zřetězen další údaj, např. příjmení uživatele a hash je pak vytvořen z celého tohoto řetězce. Pokud útočník získá přístup k úložišti hesel, neidentifikuje tak ani fakt, že např. dva různí uživatelé mají stejné heslo. Nevýhodou je ovšem nulová odolnost proti útoku známému jako „replay attack“, kdy útočník nejdříve odchytí data posílaná legitimním uživatelem a potom je sám využije k autentizaci a krádeži identity.
- Výzva-odpověď (challenge-response): Po úvodním požadavku na autentizaci od klienta server zašle náhodně vygenerovaný řetězec, který klient zašifruje pomocí svých autentizačních údajů a pošle ho zpět na server. Ten si mezitím spočítá očekávanou odpověď a po obdržení od klienta jeho verzi s očekávanou porovná. Pokud se neliší, považuje klienta za autentizovaného. Tento typ autentizace je již odolný proti útoku „replay attack“.
- Kerberos: Aktuálně se používá jeho verze 5, která je podrobně popsána v kapitole s popisem řešení.

1.3. Autorizace

V okamžiku, kdy víme, kým uživatel je, nastává problém, jaká oprávnění mu přiřadíme. Tento problém řeší autorizace. Obecně by se tedy dala autorizace definovat jako proces přiřazení přístupových práv uživateli. Ta jsou následně uplatňována na základě systému řízení přístupu.

Oprávnění ve většině informačních systémů se dělí do tří základní části:

- Read - povolení přístupu pro čtení, tzn. uživatel nemůže nijak modifikovat data, ke kterým přistupuje
- Write – povolení přístupu pro zápis, tzn. uživatel může měnit data ke kterým přistupuje.
- Execute – povolení přístupu pro spuštění, což se týká spustitelných souborů, skriptů a podobně.

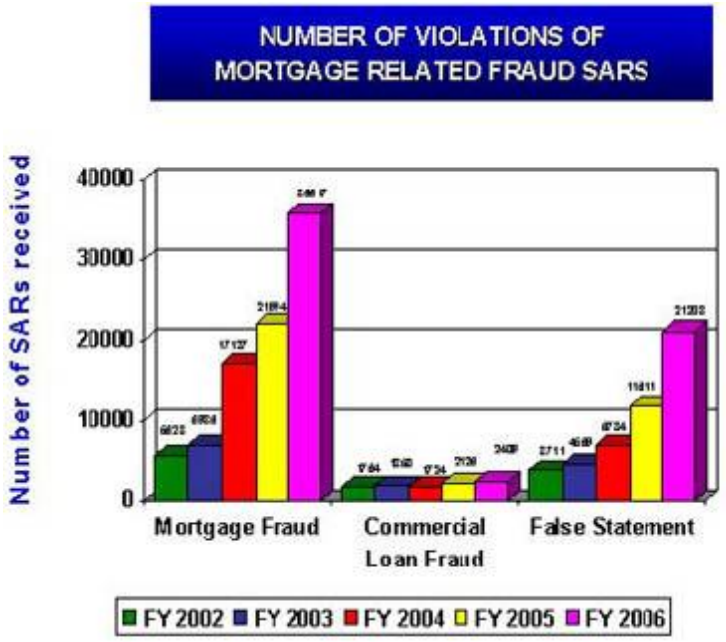
Nejčastěji se využívají tři koncepty kontroly přístupu, kterými jsou:

- DAC (Discretionary Access Control, lze přeložit jako volná kontrola přístupu [22]) – zde veškerá oprávnění pro různé skupiny či uživatele definuje vlastník obsahu, z čehož vyplývá, že každá entita v systému musí mít svého vlastníka.
- MAC (Mandatory Access Control, lze přeložit jako závazná kontrola přístupu [23]) – zde je přístup k veškerému obsahu systému definován pomocí jednotné politiky systému, kterou definuje jeho administrátor. Tento systém je například implementován ve známé nadstavbě Linuxu s názvem SELinux (Security Enhanced Linux) a obecně se využívá častěji u systémů, které drží data citlivé povahy.
- RBAC (Role-Based Access Control, kontrola přístupu na základě rolí [24]) – tento přístup spočívá v tom, že každý uživatel musí mít v systému jednu nebo více rolí, tato jeho role musí být schválena administrátorem systému a veškerý obsah v systému má nadefinována uživatelská práva pro jednotlivé role.

1.4. Audit

Zatímco v mnoha běžně používaných systémech (zejména proprietárních aplikacích) je integrovaný identity management řešen pouze v oblastech autentizace a autorizace, význam oblasti auditu byl zohledněn až u novějších systémů a to zejména proto, že elektronické podvody se staly celosvětově nejrychleji rostoucí trestnou činností. Některé uveřejněné statistiky jsou vidět na obrázcích 1 a 2.

Obr. 1: Statistiky vývoje online podvodů podle U.S. Department of Justice pro USA [13]



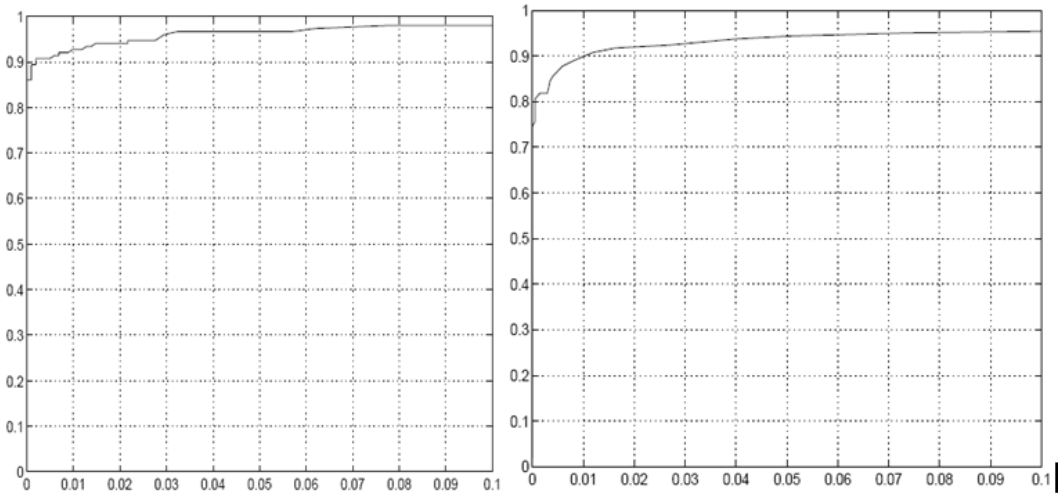
Obr. 2: Statistiky vývoje řešených událostí v bankovní oblasti (zde oblast hypoték, kde se obvykle využívá krádeže identity) společností ACFE (Association of Certified Fraud Examiners) [14]

V oblasti auditu se lze setkat s různými přístupy:

- Úplná absence nebo nedostatečná úroveň logování – s tou se lze často setkat u malých jednoduchých proprietárních aplikací. Z hlediska bezpečnosti je tento stav nevyhovující.
- Ruční analýza logů z jednotlivých systémů a zařízení – dostatečné pro malé organizace, ne však příliš spolehlivé z hlediska přítomnosti lidského faktoru a možnosti opomenutí některého zařízení.
- Centralizovaná správa logů – v tomto případě jsou logy z jednotlivých zařízení posílány na centrální server a administrátorovi se tak zjednodušuje jeho práce. Často je tento přístup kombinovaný s automatickým vyhodnocováním záznamů a administrátor tak může být zatěžován pouze se záznamy, které vyžadují jeho pozornost. Problémem u takového systému však může být jeho konfigurace, kdy administrátor nemusí být upozorněn na všechny potencionálně nebezpečné události.
- Inteligentní centralizované vyhodnocování záznamů – zejména v oblasti finančního průmyslu (nicméně začíná se rozvíjet i v oblasti telekomunikací a průmyslu) se relativně nově využívají systémy, které nejen záznamy vyhodnocují a zprávy posílají obsluze, ale jsou schopny samy vyhodnotit s pomocí technologií data miningu nebezpečné situace a předat zpět systému pokyn k zablokování události. Jedná se o systémy SIEM (Security Event Management System) [37] a Fraud Detection Systems (systémy detekce podvodů) [36], které zásadně zvyšují úroveň bezpečnosti v rámci

infastruktury, je však potřeba počítat s vysokou cenou těchto systémů. Pro detekci bezpečnostních incidentů využívají těchto metod:

- detekce na základě pravidel – nejjednodušší varianta, kdy jsou v systému předdefinována pravidla, kterými se systém řídí. Příkladem může být pravidlo pro elekronické bankovnictví, kdy např. pokud se klient pokouší převést peníze do rizikové země, pak je vyžadována např. paralelní autentizace transakce pomocí zaslání SMS s tokenem.
- detekce na základě samoučících mechanismů – nejčastěji se používá Bayesiánských metod nebo neuronových sítí. Na základě prvotního naučení systém průběžně vyhodnocuje jednotlivé události a na základě výsledků se sám průběžně učí události vyhodnocovat. Porovnání výkonnosti FDS s využitím Bayesiánských metod a neuronových sítí lze vidět na obrázku č. 3, kde byly vyhodnoceny ROC (Receiver Operation Characteristic – křivky sloužící ke grafickému znázornění vlastností systému při klasifikaci do dvou tříd) křivky pro oba typy přístupu. Na ose x je vždy poměrný počet false positives (korektní události vyhodnocené jako nebezpečné) a na ose y poměrná úspěšnost správného vyhodnocení událostí. Z grafu je vidět, že systémy využívající neuronové sítě mají vyšší úspěšnost při nulovém počtu false positives a také to, že u systémů využívajících Bayesiánských metod při zvyšování citlivosti dochází k významnému zvyšování poměru false positives bez adekvátního zvýšení počtu korektně ohodnocených transakcí. Výzkum provedla společnost Siemens AG [5].



Obr. 3: Porovnání ROC křivek pro systém pracující s neuronovými sítěmi (vlevo) a Bayesiánskými metodami (vpravo).

1.5. Single sign-on

Single sign-on [25] je koncept kontroly přístupu k více nezávislým systémům. Za normálních podmínek je potřeba, aby se klient autentizoval ke každému z těchto systémů samostatně. Filozofie single sign-on je taková, že uživatel se autentizuje

pouze k prvnímu systému v řadě, k ostatním systémům již bude autentizován automaticky na základě této první autentizace.

Výhodou toho systému je značné zjednodušení práce s používanými systémy. Klient si navíc snáze zapamatuje jediné přihlašovací údaje. Na druhou stranu, pokud budou tyto údaje vyzrazeny, útočník získá automaticky přístup ke všem používaným systémům.

Obecně lze pro implementaci single sign-on využít dvou přístupů. Prvním z nich je využití autentizační autority nezávislé na používaných systémech, se kterou jsou však tyto systémy schopné komunikovat. V tomto případě se po první autentizaci do této autority uloží informace o jejím provedení a klient je vybaven univerzálním tokenem pro přístup ke všem do konceptu zařazeným systémům. Všechny systémy se tak snaží nejdříve pracovat s tímto tokenem, jehož platnost ověří proti zmíněné autentizační autoritě, než po klientovi vyžadují znovuzadání přihlašovacích údajů. Toto je případ v projektu použitého protokolu Kerberos, který je popsán ve zvláštní kapitole.

Druhou možností je pak uložení přístupových údajů přímo v samotných systémech. V případě přechodu z jednoho systému do druhého tak autentizační token nedrží klient, ale přímo používaný systém. Toto je případ například protokolu OAuth, který je využíván především v prostředí internetu a který bude blíže představen v následující kapitole.

2. Open source single sign-on systémy

Cílem této kapitoly je představit vybrané volně dostupné konkrétní systémy sloužící pro správu identit a single sign-on. Budou představeny jednak systémy používané v rámci lokálních sítí a jednak protokoly určené pro užití v síti internet.

2.1. Samba

Samba [26] původně vznikla jako volná implementace protokolu pro sdílení souborů a tiskáren SMB používaného společností Microsoft, díky čemuž i stanice postavené na OS Linux mohly přistupovat ke zdrojům poskytovaným servery v doménách Windows NT. Vývoj Samby probíhá od roku 1992 a její aktuální stabilní verze má číslo 3, verze 4 je však již dostupná v alfa verzi.

Aktuálně tento projekt implementuje následující mechanismy:

- NetBIOS – jmenný protokol
- SMB – protokol pro sdílení souborů a tiskáren v sítích Microsoft
- CIFS – rozšíření protokolu SMB
- DCE/RPC (Distributed Computing Environment / Remote Procedure Call) – protokol pro vzdálené volání funkcí umožňující pracovat se vzdálenými systémy
- WINS Server (Windows Internet Name Service) – server přidělující NetBios jména
- NT Domain Suite – sada protokolů pro autentizaci a autorizaci

Připravovaná verze 4 řeší nepříliš vysokou úroveň bezpečnosti stávajícího řešení za pomocí plné podpory protokolu Kerberos.

2.2. FreeIPA

FreeIPA (Free Identity, Policy and Audit) [27] je SSO řešení implementované pomocí následujících technologií a prostředků:

- Fedora Linux
- Fedora Directory Server
- MIT Kerberos
- NTP
- DNS
- Webové a command-line nástroje pro správu uživatelů

Aktuální verze tohoto projektu má číslo 1.2.1, jedná se tedy o relativně nový projekt (aktivní vývoj probíhá od roku 2007). Jeho cílem do budoucna je konkurovat

komerčnímu projektu Active Directory. Již nyní jsou představeny hlavní vlastnosti připravované verze 2, která bude mimo jiné obsahovat tyto nové komponenty:

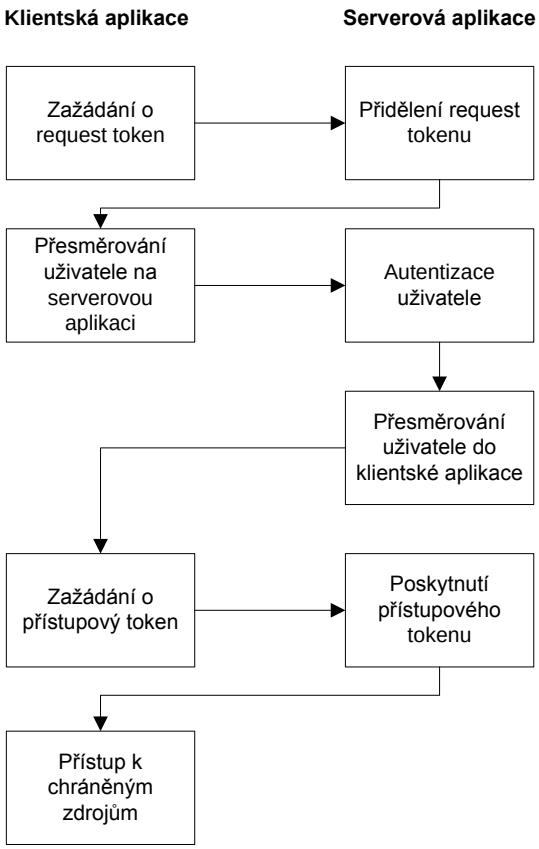
- Certifikační autorita
- podpora protokolu FreeRadius
- centrální správa Kerberos principalů služeb
- centrální auditování bezpečnostních událostí
- centrální správa bezpečnostních politik

2.3. Oauth

Oauth [28] je SSO protokol používaný především webovými aplikacemi. Oproti běžně využívaným řešením nevyužívá tento protokol centrálně spravovanou databázi uživatelů, ale uživatelské přihlašovací údaje (přesněji tokeny) jsou ukládány přímo v jednotlivých webových aplikacích. Protokol funguje následujícím způsobem:

1. Uživatel se naloguje do aplikace, která v tomto případě funguje jako klient, a ze které chce mít přístup k prostředkům poskytovaným jinou (serverovou) aplikací. V této klientské aplikaci zadá požadavek pro získání přístupového tokenu k serverové aplikaci.
2. Klientská aplikace přesměruje uživatele na OAuth rozhraní serverové aplikace, kde se uživatel autentizuje a na základě své autentizace povolí přístup klientské aplikaci po specifikovanou dobu a se specifikovanými právy.
3. Po potvrzení těchto údajů je klient přesměrován zpět do klientské aplikace, která si uloží získaný přístupový token.
4. Od této chvíle, kdykoli uživatel chce přístup k serverové aplikaci, je automaticky autentizován přímo z klientské aplikace pomocí vystaveného tokenu.

Podrobný diagram funkce protokolu je uveden na následujícím obrázku:



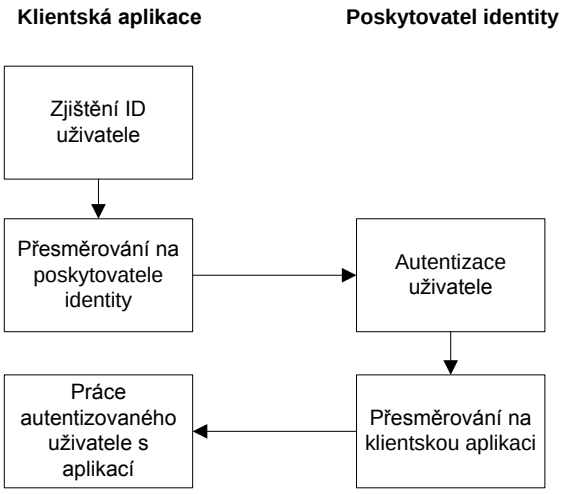
Obr. 4: Schéma funkce protokolu OAuth [16]

2.4. OpenID

OpenID [29] je opět protokol vyvinutý pro prostředí internetu. Nefunguje zcela jako single sign-on, přesto tak bývá někdy označován [30] a je vhodné jej na tomto místě zmínit. Z konceptu single sign-on využívá princip jednotné databáze uživatelů. Princip autentizace je následující a předpokládá vlastnictví OpenID jména a hesla u jeho libovolného poskytovatele a předchozí správné nastavení navštívené aplikace:

1. Uživatel navštíví aplikaci, která podporuje OpenID. Do této aplikace zadá své OpenID uživatelské jméno ve tvaru username@server.
2. Aplikace uživatele přesměruje na daný server, kde uživatel zadá své heslo.
3. OpenID server vyhodnotí, zda uživatel zadal správné heslo. Pokud ano, přesměruje jej zpět do původní aplikace, které dodá pouze informaci, že uživatel je opravdu nositelem udané identity. Aplikace se dozví pouze jeho uživatelské jméno, ale nikoli heslo.

Schéma funkce OpenID je zobrazeno na obrázku č. 5:



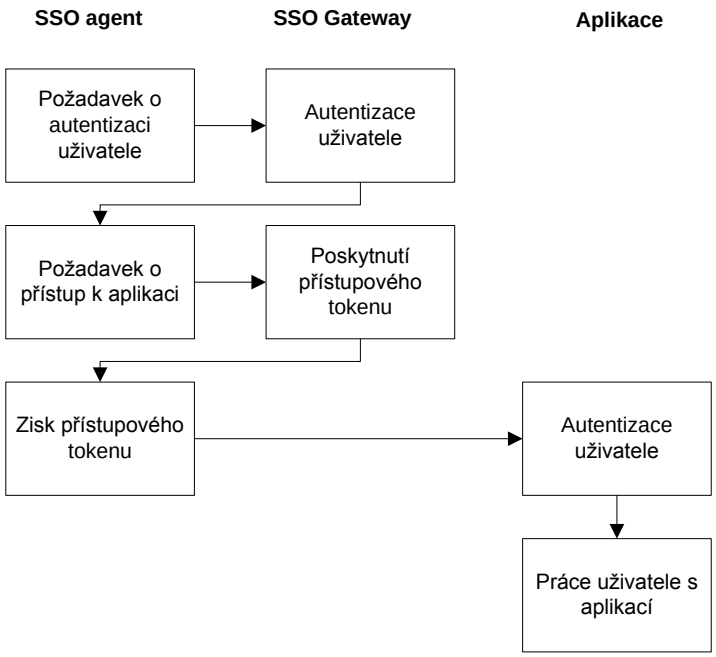
Obr. 5: Schéma funkce protokolu OpenID [17]

2.5. JOSSO

JOSSO, neboli Java Open Single Sign-On [31], je single sign-on řešení vyvinuté v jazyce Java, které je zaměřené rovněž především na webové aplikace. Sestává se ze tří komponent:

- SSO Gateway – webová služba držící autentizační token uživatele, která v procesu funguje jako brána mezi uživatelem a aplikací
- SSO Agent – klientský software zprostředkující komunikaci s bránou
- Partner Application – cílová aplikace, která podporuje SSO systém JOSSO.

Schéma komunikace je znázorněno na obrázku č. 6:



Obr. 6: Schéma funkce systému JOSSO [31]

3. Popis realizovaného projektu

3.1. Prostředí a cíle projektu

Cílem projektu bylo v prostředí počítačové učebny na ústavu telekomunikací FEKT VUT v Brně zavést autentizaci studentů tak, aby každý pracoval ve svém vlastním prostředí. Zároveň by vytvořený systém měl být platformou pro single sign-on, a studenti měli používat stejná přihlašovací jména, jaká používají již pro přístup do fakultního systému Novell eDirectory. Na základě auditních záznamů by potom mělo být pomocí frontendového rozhraní vysledovatelné, kdy a jak se student přihlašoval, zda opravdu byl fyzicky přítomen v učebně a podobně. Administrátor by měl být okamžitě zpravován o potencionálních bezpečnostních incidentech v síti.

3.2. Vybrané technologie

Při výběru konkrétních technologií, na kterých řešení postavit, bylo pracováno s těmito předpoklady:

- Veškeré použité prostředky musí být dostupné zdarma.
- Studenti budou v hodinách pracovat pod OS Linux Debian, je ovšem žádoucí, aby systém podporoval i jiné distribuce OS Linux.
- Autentizace musí probíhat bezpečnou formou, zejména heslo nesmí putovat po síti v čitelné nebo dostupnými prostředky rozluštitelné podobě.
- Podpora single sign-on.
- Pro použitá řešení musí existovat dostatečná dokumentace
- Všechny použité prostředky musí být v aktivním stadiu vývoje a podpory
- Je žádoucí, aby zvolené řešení bylo kompatibilní prostředky, které se v budoucnu mohou v dané síti používat.

Jednotlivé vybrané komponenty jsou popsané v následujících kapitolách.

3.2.1. Linux Debian 5.0 „Lenny“

V první řadě bylo třeba vybrat operační systém pro serverovou část řešení. Zejména s ohledem na požadavek licence systému zdarma bylo vybíráno z distribucí operačního systému Linux. Na počátku prací na projektu se stala vybraným systémem aktuální verze Debianu 4.0 (Etch), v průběhu prací pak vyšla nová verze 5.0 – Lenny, na kterou byl projekt poté zmigrován.

Debian GNU/Linux [12] je distribuce Linuxu, která se používá především pro serverová řešení. Tato distribuce je známa skrz svoji konzervativnost, kdy výhodou je zejména stabilita systému, který používá pouze dostatečně vyzkoušená řešení. Daní potom je, že jsou použité komponenty většinou značně zastaralé.

Debian využívá propracovaný systém balíčků APT (Advanced Packaging Tool), který umožňuje snadnou správu nainstalovaných komponent, vyhledávání mezi existujícími balíčky, jejich instalaci a update/upgrade z různých zdrojů.

3.2.2. MIT Kerberos 1.6

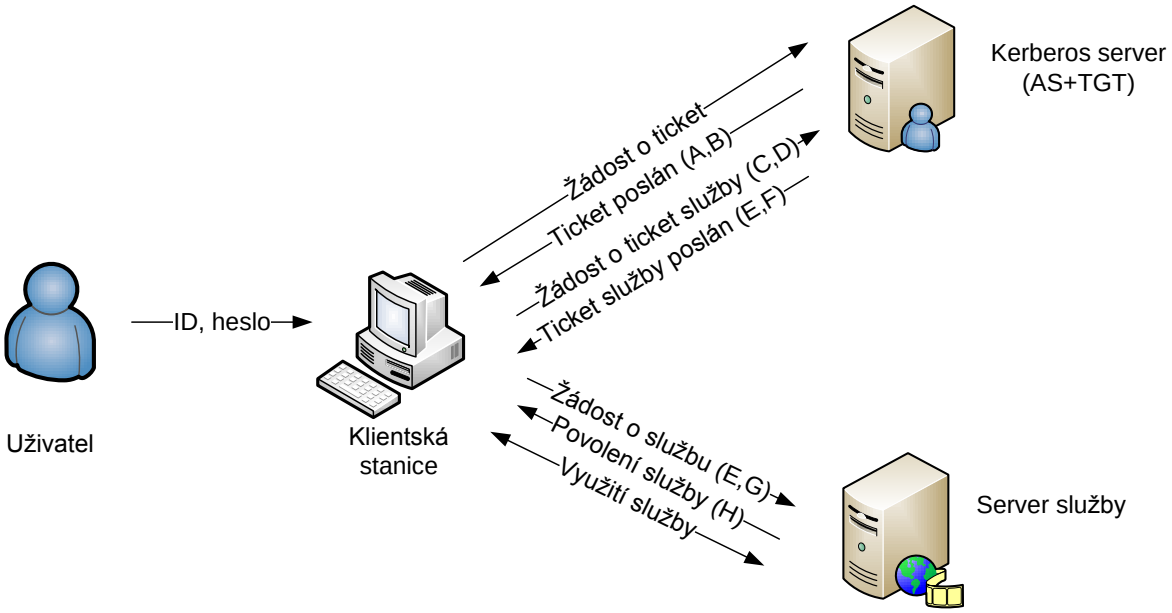
Pro autentizaci uživatelů a single sign-on byl vybrán protokol Kerberos, konkrétně jeho implementace MIT (Massachusetts Institute of Technology) [9], jakožto bezpečný, široce podporovaný a moderní autentizační prostředek.

- Kerberos je protokol, který byl vyvinut za účelem autentizace klienta v nezabezpečené síti a jeho aktuální verze je Kerberos V. Protokol využívá symetrické kryptografie a jde využít k implementaci SSO v rámci sítě, k tomu využívá tzv. důvěryhodné třetí strany (KDC – key distribution center). Princip jeho funkce je následující:
    1. Uživatel zadá uživatelské jméno a heslo.
    2. Klientská stanice provede hash hesla, tento hash se stane tajným klíčem klienta.
    3. Klient odešle na autentizační server žádost v otevřeném formátu typu „Uživatel XYZ požaduje přístup ke službě."
    4. Pokud autentizační server nalezne uživatele ve své databázi, odešle na klienta:
        - Zprávu A: Client/TGS (Ticket Granting Server) Session Key (klíč opravňující uživatele k použití Ticket Granting Serveru pro další práci) zaheslovaný tajným klíčem klienta
        - Zprávu B: Ticket Granting Ticket (TGT – obsahuje ID klienta, jeho síťovou adresu, dobu platnosti ticketu a Client/TGS Session Key) zašifrovaný pomocí tajného klíče TGS.
    5. Klient rozšifruje Client/TGS Session Key za pomocí svého tajného klíče a TGT ponechá zašifrovaný.
    6. Poté, když klient požaduje přístup k vybrané službě, odešle na TGS opět dvě zprávy:
        - Zprávu C: Sestávající se z TGT a ID požadované služby
        - Zprávu D: Autentifikátor, který je složen z ID klienta a časové značky, zašifrovaný pomocí Client/TGS Session Key.
    7. Server vezme ze zprávy C TGT, který rozšifruje pomocí svého tajného klíče, a získá tak Client/TGS Session Key. S jeho pomocí potom rozšifruje zprávu D a získá tak autentifikátor. Poté pošle klientovi další dvě zprávy:
        - Zprávu E: Client-to-server Ticket (sestávající se z ID klienta, síťové adresy klienta, doby platnosti a Client/TGS Session Key) zašifrovaný pomocí tajného klíče služby
        - Zprávu F: Client/Server Session Key zašifrovaný pomocí Client/TGS Session Key
    8. Na základě obdržených zpráv se klient připojí k serveru služby a odešle mu tyto dvě zprávy:
        - Zprávu E, kterou obdržel od TGS v předešlém kroku.

- Zprávu G: Nový autentifikátor, který obsahuje ID klienta a časovou značku a je zašifrován pomocí Client/Server Session Key.

9. Server služby dešifruje Client-to-server Ticket pomocí tajného klíče služby a získá tak Client/Server Session key, který použije k rozšifrování zprávy G. Následně klientovi odešle:
   - Zprávu H: Časová značka ze zprávy G zvýšená o 1, zašifrovaná pomocí Client/Server Session Key.
10. Klient rozšifruje zprávu H a zjistí, zda obsahuje očekávaný obsah. Pokud ano, pozná, že byl autentizován a může začít využívat danou službu.

Schéma komunikace je znázorněno na Obrázku 7:



Obr. 7: Schéma komunikace protokolu Kerberos

3.2.3. Bind 9.5

Kerberos ke své správné funkci vyžaduje korektní překlad použitých doménových jmen na IP adresy a naopak. K tomuto účelu je možné použít dvou metod:

- překlad na základě lokální databáze
- použití centralizované databáze

Lokální databáze je v případě platformy Linux realizována souborem /etc/hosts, který obsahuje dvojice IP adresa a její doménové jméno. Tento přístup je sice pricipiálně jednoduchý, ale v případě jakýchkoli změn či větších sítí velmi obtížně udržovatelný. Proto bylo v projektu využito centrální databáze DNS (Domain Name System), v rámci které je možné veškeré pravidla pro překlad IP adres na doménová jména spravovat na jednom místě.

Bind (Berkeley Internet Name Domain) [7] je v prostředí internetu nejvíce užívaným DNS serverem, který je kontinuálně vyvíjen již od 80. let minulého století. Je snadno konfigurovatelný pomocí textových souborů a velmi dobře dokumentovaný, takže volba tohoto konkrétního produktu pro účely diplomové práce byla jednoznačnou záležitostí.

3.2.4. OpenLDAP 2.4.

Samotný Kerberos pracuje pouze s přihlašovacím jménem a heslem uživatele, je však potřeba centrálně spravovat další údaje o uživateli, zejména:

- Křestní jméno, příjmení
- Unikátní identifikační číslo
- Skupiny, do nichž je uživatel v rámci systému zařazen
- Shell uživatele (typicky /bin/bash)
- Cestu k domovskému adresáři

Vhodnou bází pro tato data je LDAP (Lightweight Directory Access Protocol) databáze, která splňuje všechny uvedené požadavky, je snadno škálovatelná a široce podporovaná. Jako konkrétní implementace tohoto protokolu pro účely diplomové práce byl vybrán projekt OpenLDAP [10].

OpenLDAP je řešení, které je v linuxové distribuci Debian dobře podporováno a sestává se z těchto komponent:

- slapd: démon, který zajišťuje funkci serverové strany řešení
- systémové knihovny, které implementují LDAP protokol
- klientská strana: nástroje ldapsearch (vyhledávání na LDAP serveru), ldapadd (přidávání nových položek), ldapdelete (mazání položek) a další.

3.2.5. Apache 2.2

Pro účely auditování bezpečnostních událostí v rámci systému bylo třeba navrhnout rozhraní mezi auditorem a systémem. Jako ideální se pro auditora jeví použití tenkého klienta, zejména kvůli jednoduchému a rychlému přístupu z jakékoli lokace. Toto řešení na serverové straně vyžaduje použití webového serveru, za který byl vybrán HTTP server Apache.

Apache [32] je distribuován pro více platforem (ačkoli nejvíce je využíván na –NIX systémech) a opět se jedná o nejrozšířenější webový server na internetu. Jeho aktuální verze 2.2 podporuje velmi široké spektrum modulů a je běžně používána i v rámci enterprise sféry komerčních řešení.

3.2.6. MySQL 5.0

Bezpečnostní události jsou logovány do textového logu v adresáři /var/log/kerberos. Z hlediska zpracovávání, třídění a filtrování logů je však práce s textovými záznamy značně neefektivní. Proto si řešení vyžádalo použití RDBMS (Relational DataBase Management System), neboli relačního databázového systému. Zde je možnost

velmi širokého výběru i mezi volně dostupnými řešeními (např. PostgreSQL, HSQLDB a další), vybrán byl systém MySQL [33] zejména z hlediska jednoduchosti správy a předchozím zkušenostem autora.

Tento relační databázový systém pro projekt diplomové práce představuje více než dostatečnou dostupnost potřebných funkcionalit stejně jako vyhovující rychlost zpracování dat.

3.2.7. PHP 5.2

PHP (původně zkratka znamenala Personal Home Page, dnes se používá oficiální název PHP: Hypertext Preprocessor) [34] je skriptovací jazyk, který je určen zejména pro tvorbu jednoduchých webových aplikací. Obsahuje však i command line interface, který umožňuje tvorbu a provádění skriptů v rámci operačního systému bez návaznosti na webový server.

V rámci projektu diplomové práce je PHP použito pro auditovací rozhraní systému a pro dolování logů z textových logů a plnění MySQL databáze. Za tímto účelem je v projektu PHP použito spolu se samostatným modulem pro komunikaci s MySQL.

3.2.8. Python 2.5

Za účelem správy uživatelů systému byly naprogramovány skripty v jazyce Python [11], což je vysokoúrovňový programovací jazyk obdobně jako PHP. Používá minimalistickou syntaxi a dobře se hodí pro hromadné zpracování dat. Výhoda rovněž je, že je pro něj opět dostupné velké množství modulů, díky čemuž je programátor odstíněn od řady problémů.

V rámci projektu byly pro Python použity tyto moduly:

- sys – modul pro přístup k funkcím a proměnným operačního systému
- os – další modul pro přístup k funkcím a proměnným operačního systému
- base64 – zajišťuje kódování a dekódování řetězců uložených v Base64
- random – generátor náhodných čísel
- string – zpracování řetězců
- csv – parsování textových souborů typu csv
- codecs – rozhraní přístupu k interním kodekům Pythonu pro zpracování chyb, znakových sad a podobně

Dále zvlášť je potřeba nainstalovat další moduly, které nejsou součástí standardní instalace:

- mySqlDb – modul pro komunikaci s mySQL databází
- configobj – modul pro parsování konfiguračních souborů
- ldap – modul pro komunikaci s LDAP serverem
- pexpect – modul pro realizaci interaktivní komunikace s libovolnou službou

3.2.9. NFS 4

NFS (Networking File System) [35] je síťový souborový systém používaný ke sdílení dat po síti (obdobnou funkci na platformě Windows zajišťuje protokol SMB – Server Message Block).

V rámci projektu je tento protokol využit jednak pro centrální uchovávání domovských adresářů uživatelů, takže budou mít vždy přístup ke svým souborům z libovolné stanice zařazené v systému, jednak pro sdílení dat mezi uživateli, ale také jako praktická demonstrace (proof of concept) funkčnosti single sign-on v síti, kdy je implementována bezpečná autentizace k síťovému úložišti pomocí protokolu Kerberos.

3.3. Popis řešení

Řešení je rozděleno mezi serverovou a klientskou část.

Na serveru běží tento SW:

- Debian 5.0
- Kerberos 5, implementace MIT 1.4.4
- OpenLDAP 2.3.0
- Python 2.4.4
- PHP 5.2
- skripty pro správu uživatelů vyvíjené v Pythonu
- NFS4 - serverová část

Na klientovi pak běží pouze Debian 4.0 s nainstalovanými PAM (Pluggable Auhentication Module) balíčky pro LDAP a Kerberos a NFS klient.

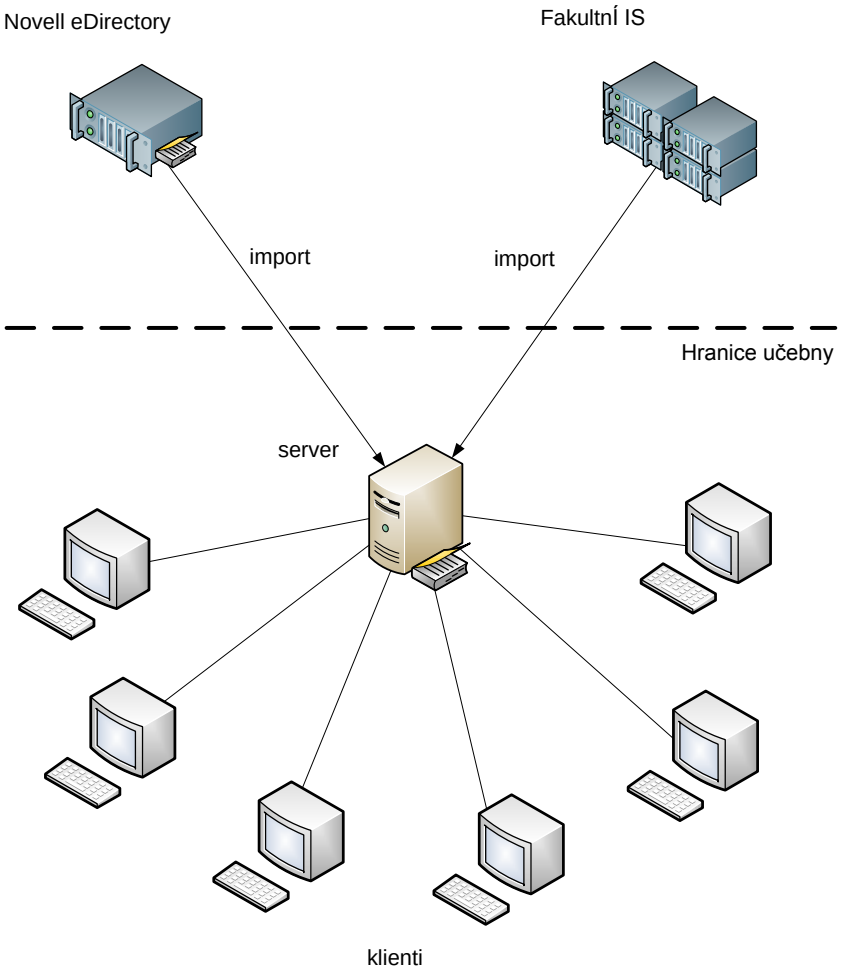
Serverová část zajišťuje tyto úlohy:

- Ověřování identity uživatelů
- Poskytování informací z LDAPu
- Import uživatelů z fakultní databáze Novell
- Mazání uživatelů
- Držení auditních záznamů

Klientská část pak zajišťuje následující:

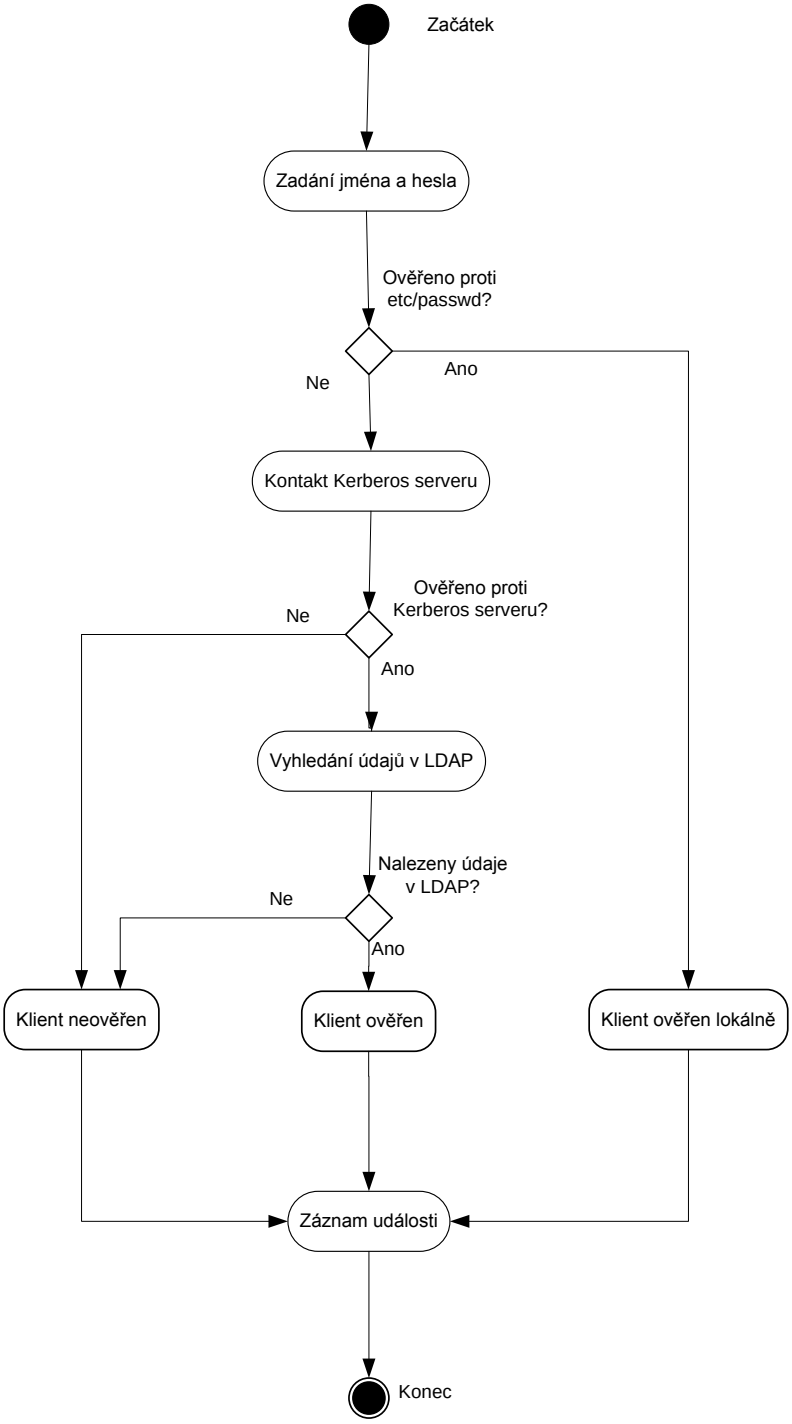
- front-end pro autentizaci uživatele
- uživatelské rozhraní pro práci

Struktura sítě je uvedena na Obrázku č. 5. Serverová i klientská část řešení je umístěna v učebně. Server pak pomocí importu získává data z univerzitního IS a systému Novell eDirectory.



Obr. 8: Schéma sítě

Postup autentizace je pak zachycen na Obrázku č. 9. V první řadě je proveden pokus o ověření uživatele lokálně (např. přihlášení lokálního administrátora apod.). Pokud se nepovede lokální ověření, systém se dotáže autentizačního serveru Kerberos. Ten se pokusí klienta ověřit, pokud se to povede, vydá mu TGT a klient se pokusí stáhnout údaje o uživateli z LDAP serveru. Pokud se některá z událostí nepodaří, klientovi je odmítnuta autentizace.



Obr. 9: průběh autentizace klienta

3.4. Postup implementace

Předpokládáme následující fakta (na základě prostředí, do kterého byl projekt implementován):

- použitá doména má název utko.feec.vutbr.cz
- server má doménové jméno „im-server.utko.feec.vutbr.cz“ a adresu 192.168.1.97
- klient má doménové jméno „im-client.utko.feec.vutbr.cz“ a adresu 192.168.1.42

- na obou serverech je pracováno pod uživatelem root.

Kde jsou v následujícím textu tyto údaje, při instalaci do jiného prostředí je potřeba je změnit na skutečné údaje.

3.4.1. Prekonfigurační kroky společné pro server i klient

Na počátku je třeba implicitní instalaci uvést do definovaného výchozího stavu pro bezproblémovou instalaci řešení.

a) Úprava hostname

Nejdříve je vhodné zkontrolovat a případně upravit hostname stroje. Podle uživatelských zkušeností na internetu se kerberos může chovat nepředvídatelně, pokud není hostname nastaveno jako FQDN (Fully Qualified Domain Name), tedy hostname musí obsahovat jméno stroje i domény.

vim /etc/hostname

Zde by v případě serveru mělo být napsáno:

```
im-server.utko.feec.vutbr.cz
```

A v případě klienta:

```
im-client.utko.feec.vutbr.cz
```

Pokud zde je z instalace nastaveno pouze samotné jméno stanice (např. im-server), upravíme hodnotu na FQDN, uložíme soubor a pomocí příkazu

hostname –F /etc/hostname

aktualizujeme hostname z upraveného souboru. Následně pomocí samotného příkazu

hostname

zkontrolujeme, zda se hostname aktualizoval korektně.

b) Statické vlastnosti síťového rozhraní

V implicitní instalaci je v Debianu nainstalovaný a spuštěný DHCP klient, který si v pravidelných intervalech aktualizuje vlastnosti z DHCP serveru i v případě, že je příslušné síťové rozhraní nakonfigurováno staticky s pevnou IP adresou. Toto činí problémy především u nastavení DNS serveru (soubor /etc/resolv.conf se neustále mění na hodnoty získané z DHCP serveru), který je pro chod řešení nezbytný. Proto je potřeba podniknout následující kroky.

Nejdříve nastavíme vlastnosti síťového rozhraní:

vim /etc/network/interfaces

U aktivního síťového rozhraní upravíme vlastnosti na reálné údaje. V případě stroje im-server bude konfigurace vypadat následovně:

```
iface eth1 inet static
   address 192.168.122.97
```

```
netmask 255.255.255.0
gateway 192.168.122.1
dns-nameservers 192.168.122.97 192.168.122.1
```

V případě DNS serveru tedy bude primární DNS server nastaven u všech strojů na adresu stroje im-server a sekundární na adresu doposud používaného DNS serveru, ostatní údaje ponecháme na základě konfigurace přidělené DHCP serverem.

Následně vyrestartujeme síťové rozhraní, čímž načteme novou konfiguraci:

/etc/init.d/networking restart

Dále odstraníme balík dhcp klienta, jelikož nebyl nalezen jiný způsob, jak zamezit jeho neustálému pollování:

aptitude remove dhcp3-common

A upravíme konfiguraci DNS v souboru /etc/resolv.conf:

```
nameserver 192.168.122.97
nameserver 192.168.122.1
```

Vzhledem k tomu, že při překládání DNS jmen je nejdříve kontrolován lokální hosts soubor, je potřeba zamezit tomu, aby bylo lokální doménové jméno překládáno nesprávně. To vyřešíme tak, v souboru /etc/hosts zakomentujeme následující řádek

```
127.0.1.1 im-server.utko.feec.vutbr.cz im-server
```

tak, že na jeho začátek umístíme znak „#“.

c) Úprava vlastností konfiguračního rozhraní

Při instalaci některých balíků se spouští rozhraní debconf, do kterého jsou interaktivně zadávány konfigurační hodnoty. Je tedy potřeba nakonfigurovat debconf tak, aby se ptal na všechny dotazy:

dpkg-reconfigure debconf

Spustí se konfigurační rozhraní, kde zadáme postupně na jednotlivých stranách tyto hodnoty:

```
interface: dialog
```

```
priority: low
```

Následně se konfigurační rozhraní samo ukončí.

d) Nastavení přesného času a časové zóny

Kerberos vyžaduje synchronizaci času mezi klientskou a serverovou částí. Toto zajistíme pomocí NTP (Network Time Protocol) klienta synchronizovanému oproti NTP serveru. Nejdříve tedy stáhneme klientský balík ntpdate:

aptitude install ntpdate

A následně sesynchronizujeme čas například proti NTP serveru cesnet.cz:

ntpdate ntp.cesnet.cz

Dále nakonfigurujeme správnou časovou zónu pomocí příkazu:

dpkg-reconfigure tzdata

Ve spuštěném rozhraní nakonfigurujeme lokální časovou zónu:

```
Geographic Area: Europe
Time zone: Prague
```

3.4.2. Serverová část

e) Instalace a konfigurace DNS serveru

Nejdříve je potřeba nainstalovat výše popsaný DNS server BIND:

aptitude install bind9 dnsutils

Tím nainstalujeme následující balíčky:

- bind9 – serverová strana DNS serveru
- dnsutils – obsahuje klientské utility pro práci s DNS serverem

Následně nakonfigurujeme jednotlivé DNS zóny a jejich používané soubory v souboru /etc/bind/named.conf. Otevřeme si tedy tento soubor pro editaci:

vim /etc/bind/named.conf

a do tohoto souboru připíšeme následující dva odstavce pro dopředný a reverzní překlad:

```
zone „utko.feec.vutbr.cz“ {
   type master;
   file “/etc/bind/zones/utko.feec.vutbr.cz.db”;
};

zone “122.168.192.in-addr.arpa” {
   type master;
   file “/etc/bind/zones/122.168.192.in-addr.arpa.db”;
};
```

Samozřejmě jednotlivé hodnoty upravíme na základě skutečných parametrů sítě.

Dále vytvoříme konfigurační soubor dopředného doménového překladu:

vim /etc/bind/zones/utko.feec.vutbr.cz.db

a do tohoto souboru zapíšeme následující obsah:

```
$TTL 1h
@   IN SOAim-server.utko.feec.vutbr.cz.
```

```
root.utko.feec.vutbr.cz. (
                    0000000001
                    1h
                    15m
                    2w
                    1h
)
@          IN NS  im-server.utko.feec.vutbr.cz.
@          IN A   192.168.122.97
im-server  IN A   192.168.122.97
im-client  IN A   192.168.122.42
```

Tím definujeme následující skutečnosti:

- výchozí nastavení „time to live“, tedy jak dlouho budou klienti ukládat přeložené adresy do cache, je 1 hodina
- primární server pro tuto zónu je im-server.utko.feec.vutbr.cz
- kontakt na správce zóny je root@utko.feec.vutbr.cz
- sériové číslo zóny je 0000000001, tuto hodnotu je potřeba navyšovat při úpravách v záznamech, zejména pokud budou použity “slave” DNS servery, které si na základě změny sériového čísla aktualizují svůj záznam
- podřízené DNS servery budou své uložené sériové číslo zóny s uvedeným jednou za hodinu
- pokud se nezdaří spojení podřízeného DNS serveru za účelem porovnání sériového čísla, bude se o to pokoušet po patnáctiminutových intervalech
- pokud se nezdaří připojení podřízeného DNS serveru za účelem porovnání sériového čísla, po dvou týdnech od prvního pokusu tento zneplatní své uložené záznamy pro tuto zónu
- v případě dotazu na neexistující záznam si podřízený server tuto skutečnost bude pamatovat jednu hodinu
- adresa domény bude překládána na adresu 192.168.122.97
- im-server vlastní adresu rovněž 192.168.122.97
- im-client vlastní adresu 192.168.122.42

Na tomto místě je nutné uvést, že pro korektní funkci Kerbera by každý klient měl mít korektní DNS záznam, při přidávání klientů je třeba je tudíž zanést do tohoto souboru a zároveň zvýšit sériové číslo záznamu.

Stejně tak vytvoříme konfigurační soubor pro reverzní překlad

`vim /etc/bind/zones/122.168.192.in-addr.arpa.db`

a do tohoto souboru zapíšeme následující obsah:

```
$TTL 1h
@  IN SOAim-server.utko.feec.vutbr.cz.
root.utko.feec.vutbr.cz. (
                    0000000001
                    1h
                    15m
                    2w
                    1h
)
@         IN NS im-server.utko.feec.vutbr.cz.
97        IN PTRim-server.utko.feec.vutbr.cz.
42        IN PTRim-client.utko.feec.vutbr.cz.
```

Uvedené položky mají stejný význam jako u předchozího konfiguračního souboru s výjimkou posledních dvou řádků, které udávají:

- IP adresa 192.168.122.97 bude překládána na doménové jméno im-server.utko.feec.vutbr.cz
- IP adresa 192.168.122.42 bude překládána na doménové jméno im-client.utko.feec.vutbr.cz

Protože BIND historicky obsahoval značné množství zranitelností a takto nainstalovaný běží s právy roota, je vhodné ho uzavřít do tzv. chroot prostředí, aby neměl přístup k celému systému, což je pro něj zbytečné. Uzavřeme ho tedy do adresáře /var/lib/named.

Nejdříve vytvoříme potřebnou adresářovou strukturu:

```
mkdir -p /var/lib/named/etc
mkdir -p /var/lib/named/dev
mkdir -p /var/lib/named/var/cache/bind
mkdir -p /var/lib/named/var/run/bind/run
```

Do této struktury přesuneme stávající konfigurační soubory BINDu:

```
mv /etc/bind /var/lib/named/etc
```

A z důvodu zachování konvenčního uspořádání vytvoříme symbolický odkaz v adresáři /etc do nově vytvořeného adresáře:

```
ln –s /var/lib/named/etc/bind /etc/bind
```

Pro takto uzavřený BIND vytvoříme zařízení null a random:

```
mknod /var/lib/named/dev/null c 1 3
mknod /var/lib/named/dev/random c 1 8
```

Na nově vytvořené adresáře nastavíme práva a vlastníka na Bind:

```
chmod 666 /var/lib/named/dev/null
```

```
chmod 666 /var/lib/named/dev/random
chown -R bind:bind /var/lib/named/var/*
chown -R bind:bind /var/lib/named/etc/bind
```

Dále je třeba zařídit logování DNS serveru. Proto nejdříve nainstalujem logovací démon, který BIND používá:

```
aptitude install sysklogd
```

a vytvoříme socket v chrootovaném adresáři tak, že upravíme startovací skript démona syslogd:

```
vim /etc/init.d/sysklogd
```

kde přidáme socket démona pro sbírání zpráv BINDu uzavřeného v chrootovaném prostředí:

```
PATH=/bin:/usr/bin:/sbin:/usr/sbin
SYSLOGD="-a /var/lib/named/dev/log"
```

Nakonec vyrestartujeme služby s novými nastaveními:

```
/etc/init.d/sysklogd restart
/etc/init.d/bind9 restart
```

a ověříme funkci DNS serveru na serveru i klientovi např. takto:

```
nslookup im-client.utko.feec.vutbr.cz
nslookup 192.168.122.42
```

Pokud je vše nakonfigurováno správně, dostaneme na tyto příkazy následující odezvy:

```
Server:   192.168.122.97
Address:  192.168.122.97#53

Name:     im-client.utko.feec.vutbr.cz
Address:  192.168.122.42
```

```
Server:   192.168.122.97
Address:  192.168.122.97#53

42.122.168.192.in-addr.arpa  name = im-
client.utko.feec.vutbr.cz.
```

Tímto je instalace DNS serveru úspěšně ukončena.

f) Instalace a konfigurace Kerbera

Nejdříve nainstalujeme serverovou část implementace MIT protokolu Kerberos. Ta se sestává jednak z Key Distribution Centra, tedy komponenty, která vystavuje

tickety klientům, a jednak z administračního serveru, který slouží ke správě principalů:

```
sudo apt-get install krb5-{admin-server,kdc}
```

Po stažení balíků se spustí rozhraní Debconf, kde zadáme odpovědi na tyto dotazy:

```
Default Kerberos version 5 realm: UTKO.FEEC.VUTBR.CZ
Does DNS contain pointers to your realm's Kerberos Servers? No
Create Kerberos KDC Configuration with debconf? Yes
Kerberos4 compatibility mode to use? none
Run a krb524d? No
Should the data be purged as well as the package files? No
Run the Kerberos5 administration daemon (kadmind)? Yes
What are the Kerberos servers for your realm? im-server.utko.feec.vutbr.cz
 What is the administrative server for your realm? im-server.utko.feec.vutbr.cz
```

Význam provedených nastavení je následující:

- Kerberos realm je nastavený jako UTKO.FEEC.VUTBR.CZ (odpovídá použité doméně, jen je potřeba jej zadat velkými písmeny)
- DNS server neobsahuje záznamy ohledně KDC serveru Kerbera, tzn. klienti budou používat KDC server specifikovaný v jejich nastaveních
- Konfigurace KDC serveru bude vytvořena pomocí konfiguračního rozhraní debconf (budou zobrazeny následující dotazy, které budou uloženy do konfiguračních souborů)
- Nebude povolena autentizace pomocí starší verze protokolu Kerberos 4
- Nebude spouštěn démon pro kompatibilitu s tickety používanými starší verzí Kerberos 4
- V případě smazání balíku kerbera nebudou odstraněny konfigurační soubory
- Bude spuštěno administrační rozhraní Kerbera, které slouží ke správě principalů
- KDC server má adresu im-server.utko.feec.vutbr.cz
- Administrační server má rovněž adresu im-server.utko.feec.vutbr.cz

Dále je třeba uvedený realm fyzicky vytvořit:

```
krb5_newrealm
```

Zde zadáme master password, kterým budou šifrovány jednotlivé principaly. Je vhodné toto heslo vybrat silné.

Ve výchozím nastavení Kerberos loguje do souboru /etc/log/syslog. Toto by bylo jednak značně nepřehledné a dále by bylo náročnější extrahování záznamů pro front-endové auditorské rozhraní. Proto vytvoříme adresář pro logování Kerbera:

```
mkdir /var/log/kerberos
```

Následně otevřeme pro editaci konfigurační soubor Kerbera:

```
vim /etc/krb5.conf
```

Najdeme část označenou `[realms]`, která udává, kde hledat KDC a administrační server pro každý realm. Mezi již uvedenými realmy zkontrolujeme a případně doplníme záznam o našem realmu:

```
UTKO.FEEC.VUTBR.CZ = {
     kdc = im-server.utko.feec.vutbr.cz
     admin _server = im-server.utko.feec.vutbr.cz
     default_domain = utko.feec.vutbr.cz
}
```

Údaj default_domain je nepovinný, slouží pro autentizaci v případě, že klienti používají tickety Kerberos 4, které oproti Kerberu 5 neobsahují FQDN názvy.

Dále najdeme část `[domain_realm]`, která definuje mapování z hostnames na jednotlivé realmy Kerbera. Pokud používáme pouze jeden Kerberos realm, který odpovídá doménovému názvu, není tato úprava nezbytná. Pro případ možných budoucích úprav je však lepší ji vyplnit, jelikož ničemu vadit nebude.

```
[domain_realm]
     .utko.feec.vutbr.cz = UTKO.FEEC.VUTBR.CZ
     utko.feec.vutbr.cz = UTKO.FEEC.VUTBR.CZ
```

Nakonec je vhodné nastavit logování Kerbera do samostatných souborů v adresáři, který jsme si pro tento účel vytvořili. To provedeme pomocí přidání těchto řádků:

```
[logging]
     kdc = FILE:/var/log/kerberos/krb5kdc.log
     admin_server = FILE:/var/log/kerberos/kadmin.log
     default = FILE:/var/log/kerberos/krb5lib.log
```

Tím definujeme, že KDC (tedy informace o vydávání či zamítání ticketů) budeme logovat do souboru krb5kdc.log, informace z administračního rozhraní do souboru kadmin.log, a vše ostatní do souboru krb5.log.

Ukončíme vim a provedeme restart kerbera, aby se načetla nová nastavení:

```
sudo invoke-rc.d krb5-admin-server restart
sudo invoke-rc.d krb5-kdc restart
```

Následovně je potřeba vytvořit principal pro roota, aby mohl sám spravovat ostatní principaly. Nejdříve tedy spustíme rozhraní kadmin.local (lokální rozhraní administračního serveru, ke kterému má přístup pouze root):

```
kadmin.local
```

Zde vytvoříme zmiňovaný principal pro roota root/admin. „Admin“ za lomítkem zajistí, že root bude mít přístup do rozhraní kadmin, se kterým se pracuje stejně jako s kadmin.local, ovšem toto rozhraní již autentizuje uživatele pomocí protokolu Kerberos a tudíž jej lze spouštět i z jiných počítačů:

```
addprinc root/admin
```

Následně budeme vyzváni pro zadání hesla k principalu (celkově dvakrát – jednou pro zadání a podruhé pro ověření).

Dále příkazem

```
exit
```

opustíme administrační konzoli kadmin.local a upravíme access list definovaný v souboru /etc/krb5kdc/kadm5.acl tak, aby všichni uživatelé, kteří mají „admin“ za lomítkem, měli přístup do konzole kadmin. Nejdříve tedy otevřeme tento soubor pro editaci:

```
vim /etc/krb5kdc/kadm5.acl
```

a zde odkomentujeme následující řádek:

```
*/admin *
```

První položka znamená skupinu, které se pravidlo týká, a hvězdička za mezerou znamená, že uvedená skupina má přidělena plná práva, tedy výpis, editaci, přidávání i mazání principalů.

Poté je potřeba vyrestartovat službu administračního rozhraní pomocí příkazu

```
/etc/init.d/krb5-admin-server restart
```

Pokud vše proběhlo v pořádku, budeme schopni se pomocí nově vytvořeného principalu autentizovat ke konzoli kadmin. Tu zavoláme příkazem:

```
kadmin
```

Zde se autentizujeme pomocí hesla nastaveného při vytváření principalu a měli bychom být schopni vylistovat seznam vytvořených principalů. Pomocí příkazu

```
listprincs
```

ověříme, že přístupová práva jsou přidělena v pořádku:

```
K/M@UTKO.FEEC.VUTBR.CZ
kadmin/admin@UTKO.FEEC.VUTBR.CZ
kadmin/changepw@UTKO.FEEC.VUTBR.CZ
kadmin/history@UTKO.FEEC.VUTBR.CZ
krbtgt/UTKO.FEEC.VUTBR.CZ@UTKO.FEEC.VUTBR.CZ
root/admin@UTKO.FEEC.VUTBR.CZ
```

Volitelně je možné přidat politiku hesel. Tu můžeme přidat například následovně:

```
add_policy -maxlife 180days -minlife 2days -minlength 8 -minclasses 2 -history 10 default
```

Význam jednotlivých parametrů je následující:

- maxlife: doba expirace hesla
- minlife: nejkratší doba, po jaké systém dovolí změnit heslo
- minlength: minimální délka hesla
- minclasses: minimální počet typů znaků, které musí heslo obsahovat (velká písmena, malá písmena, číslice, nealfanumerické znaky)
- history: historie hesel, kterou si bude Kerberos pamatovat, a v rámci které nedovolí zadat dvakrát stejné heslo
- default: jméno politiky. „Default" znamená, že tato politika bude platit pro všechny principaly.

V rámci implementovaného projektu není politika hesel vyžadována, toto nastavení je ponecháno na posouzení administrátora systému.

Po ukončení práce v kadmin rozhraní ho opustíme příkazem

```
exit
```

Tímto je hotova počáteční konfigurace Kerbera.

g) Instalace a konfigurace OpenLDAP serveru

Dalším krokem je instalace LDAP serveru OpenLDAP:

```
aptitude install slapd ldap-utils libldap-2.4-2 libdb4.6
```

Tímto nainstalujeme následující balíčky:

- slapd: démon OpenLDAP serveru
- ldap-utils: klientská strana pro práci s OpenLDAP serverem
- libldap-2.4-2: knihovny pro OpenLDAP
- libdb4.6: databázové knihovny

Po stažení balíčků se otevře rozhraní debconf, kde odpovíme na následující dotazy:

```
Omit OpenLDAP server configuration? No
DNS domain name: utko.feec.vutbr.cz
Name of your organization? utko.feec.vutbr.cz
Administrator password: (heslo)
Confirm password: (heslo)
Database backend to use: HDB
Do you want the database to be removed when slapd is purged? No
Allow LDAPv2 protocol? No
```

Tyto položky mají následující význam:

- Zadání, že chceme OpenLDAP nakonfigurovat pomocí rozhraní debconf
- jméno domény, kterou bude LDAP obsluhovat
- jméno organizace (lze ponechat jméno domény)
- zadání a potvrzení hesla administrátora OpenLDAP serveru
- vybrání databázového backendu, který bude OpenLDAP využívat (lze vybírat mezi Berkeley databázemi BDB a HDB, pro účely projektu lze vybrat libovolnou databázi, ale HDB je v rozhraní doporučená)
- Při smazání OpenLDAP serveru nebude smazán obsah databáze
- Nepovolíme starší verzi protokolu LDAPv2, která není pro projekt potřebná

Dále upravíme konfigurační soubor serverové části LDAP serveru:

```
vim /etc/ldap/slapd.conf
```

Zde zapneme logování připojení, operací a výsledků pro účely potencionálního ladění serveru:

```
loglevel 256
```

Nejčastěji bude probíhat vyhledávání pomocí uid, takže přidáme indexování podle této hodny pro rychlejší vyhledávání:

```
index ObjectClass eq
index uid   eq
```

Opustíme vim a zastavíme OpenLDAP server

```
/etc/init.d/slapd stop
```

a provedeme přeindexování s novým nastavením (uid)

```
slapindex
```

Dále změníme vlastníka souborů spravovaných dat na službu openldap

```
chown openldap:openldap /var/lib/ldap/*
```

A nakonec znovu spustíme OpenLDAP server s novými nastaveními.

```
/etc/init.d/slapd start
```

h) Instalace a konfigurace NFS 4

Nejdříve vytvoříme systematickou strukturu pro aplikaci NFS:

```
mkdir –p /mnt/data/home
mkdir /mnt/share
```

Tato struktura má následující význam:

- adresář /mnt/data bude sloužit jako mount point pro klienty
- v adresáři /mnt/data/home budou mít studenti umístěné své domovské adresáře

- adresář /mnt/data/share bude sloužit jako sdílené úložiště pro studenty (např. pro distribuci zadání úloh, programů a podobně)

Pokud již byly vytvořeny nějaké domovské adresáře adresáři /home, je třeba nakopírovat do adresáře /mnt/data/home, protože z tento adresář budeme mountovat i do adresáře /home na serveru, aby všichni uživatelé měli svá data dostupná po celé síti. To zajistíme úpravou souboru /etc/fstab:

```
vim /etc/fstab
```

Zde přidáme tento řádek:

```
/mnt/data/home  /home  none   rw,bind   0   0
```

Významy položek jsou následující:

- /mnt/data/home: zdroj mountování
- /home: cíl mountování
- none: zajistí použití originálního filesystému
- rw: povolit čtení i zápis na svazek
- bind: obsah bude dostupný ve zdrojovém i cílovém adresáři
- první „0“: namountovaný adresář nebude zálohován
- druhá „0“: namountovaný adresář nebude kontrolován při startu systému

Dále nainstalujeme balíčky NFS:

```
aptitude install nfs-kernel-server nfs-common portmap
```

Ty obsahují následující komponenty:

- nfs-kernel-server: serverová strana NFS
- nfs-common: klientská strana NFS
- portmap: služba pro podporu RPC (Remote Procedure Call)

Dále je potřeba vytvořit principal pro server a přidat jej do systémového keytabu. Spustíme tedy rozhraní administrační rozhraní kerbera příkazem

```
kadmin
```

kde se autentizujeme heslem roota a pomocí příkazu

```
addprinc –randkey nfs/im-server.utko.feec.vutbr.cz
```

vytvoříme principal serveru pro službu NFS s náhodně vygenerovaným heslem. Ten poté vyexportujeme do systémového keytabu pomocí příkazu

```
ktadd nfs/im-server.utko.feec.vutbr.cz
```

a opustíme konzoli kadmin:

```
exit
```

Následně je potřeba upravit konfigurační soubory NFS pro podporu Kerbera. Začneme s konfigurací serverové části:

```
vim /etc/default/nfs-kernel-server
```

Zde upravíme následující položku:

```
NEED_SVCGSSD=yes
```

Tím zajistíme při startu služby spuštění démona svcgssd, který zajišťuje interface mezi Kerberem a NFS.

Dále otevřeme konfigurační soubor klientské části:

vim /etc/default/nfs-common

A zde upravíme následující položky:

```
NEED_IDMAPD=yes
NEED_GSSD=yes
```

Tato nastavení způsobí, že démon nfs-common bude spouštěn zároveň s démony idmapd, který slouží pro vzdálené mapování ID uživatele, a gssd, který zajišťuje interface mezi Kerberem a NFS na klientské části.

Samotný démon idmapd ještě potřebuje pro svou korektní funkci nastavit správně místní doménu. To zařídíme úpravou konfiguračního souboru /etc/idmapd.conf

vim /etc/idmapd.conf

Kde upravíme parametr domény v sekci General:

```
[General]
Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = utko.feec.vutbr.cz
```

Dále specifikujeme adresář pro exportování pomocí úpravy souboru /etc/exports:

vim /etc/exports

Zde zapíšeme následující řádek:

```
/mnt/data gss/krb5i(rw,fsid=0,insecure,no_subtree_check)
```

Významy jednotlivých položek jsou následující:

- mnt/data: adresář určený k exportu
- gss/krb5i: cílové stanice se budou autentizovat pomocí Kerbera s kontrolou integrity dat
- rw: adresář bude zpřístupněn pro zápis
- fsid=0: udává root exportované struktury
- insecure: povolení připojení z libovolného portu pro případ např. úpravy pravidel na firewallu
- no_subtree_check: vypnutí kontroly, zda požadovaný soubor opravdu patří do exportovaného adresáře. Jedná se o implicitní volbu, která nesnižuje spolehlivost přístupu k exportovaným souborům.

Následně je třeba vyrestartovat služby NFS s novým nastavením:

```
/etc/init.d/nfs-kernel-server restart
/etc/init.d/nfs-common restart
```

Nakonec vyexportujeme adresář /mnt/data pomocí načtení aktuální konfigurace souboru /etc/exports:

```
exportfs –r
```

a zkontrolujeme úspěšný export pomocí příkazu

```
showmount –e
```

kdy bychom měli získat tento výpis:

```
Export list for im-server.utko.feec.vutbr.cz:
/mnt/data gss/krb5i
```

i) Instalace a konfigurace klientských balíčků na serveru

Nejdříve nainstalujeme PAM modul pro Kerberos:

```
aptitude install libpam-krb5
```

Dále nainstalujeme potřebné klientské balíčky pro komunikaci s ldap serverem:

```
aptitude install libnss-ldap nscd
```

Tyto balíčky mají následující účel:

- libnss-ldap: modul, který umožňuje LDAP serveru vystupovat jako jmenná služba
- nscd: Name Service Cache Daemon, služba, která slouží ke cachování výsledků z LDAP serveru

Spustí se rozhraní debconf, kde zodpovíme následující otázky:

```
LDAP server Uniform Resource Identifier: ldap://192.168.122.97/
Distinguished name of the search base:
dc=utko,dc=feec,dc=vutbr,dc=cz
LDAP version to use: 3
Does the LDAP database require login? No
Special LDAP privileges for root? No
Make the configuration file readable/writeable by its owner only?
No
Make local root Database admin. No
Does the LDAP database require login? No
Local crypt to use when changing passwords. crypt
```

Zadané konfigurační parametry mají následující význam:

- adresa LDAP serveru
- vyhledávací báze na LDAP serveru

- verze LDAP protokolu
- LDAP databáze pro přečtení hodnot nepotřebuje login
- root na localhostu nebude mít žádná zvláštní privilegia na LDAP serveru
- konfigurační soubor bude vytvořen se standardními právy (např. pro potřebu jeho čtení různými službami) – lze případně následně upravit s využitím standardních linuxových mechanismů
- při změně přístupového hesla k LDAP serveru bude na stanici toto heslo šifrováno

Dále otevřeme pro editaci konfigurační soubor ldap.conf, který slouží k nastavení klienta ldap:

```
vim /etc/ldap/ldap.conf
```

a zapíšeme následující dva řádky:

```
BASE  dc=utko,dc=feec,dc=vutbr,cd=cz
URI ldap://192.168.122.97/
```

Tyto parametry určují vyhledávací bázi na LDAP serveru a jeho adresu.

Nakonec aktivujeme čtení uživatelských jmen a skupin z LDAP serveru upravením souboru /etc/nsswitch.conf

```
vim /etc/nsswitch.conf
```

kde upravíme následující dva řádky:

```
passwd:         compat ldap
group:          compat ldap
```

Debian obsahuje popsaný bug, který se v případě, že je v souboru nsswitch.conf definovaný ldap, snaží hledat skupiny zařízení na tomto serveru, přestože je uveden až na druhém místě. To výrazně prodlužuje dobu bootování a výpis velkého množství chybových hlášek. Tento efekt lze výrazně zmírnit úpravou politiky vyhledávání položek. To učiníme tak, že otevřeme soubor /etc/libnss-ldap.conf:

```
vim /etc/libnss-ldap.conf
```

V tomto souboru najdeme řádek

```
#bind_policy hard
```

Tento řádek odkomentujeme a upravíme následujícím způsobem:

```
bind_policy soft
```

Následovně nainstalujeme PAM modul pro přístup k LDAPu:

```
aptitude install libpam-ldap
```

Nakonec upravíme konfigurační soubory PAM:

Soubor common-account, který definuje autorizační pravidla:

```
vim /etc/pam.d/common-account
```

Do tohoto souboru zapíšeme následující řádky:

```
account sufficient      pam_unix.so
account required        pam_ldap.so
account required        pam_krb5.so
```

Tato konfigurace, znamená, že pokud bude účet nalezen v databázi /etc/passwd, budou pro něj použity odpovídající pravidla. Pokud zde nebude nalezen, budou autorizační pravidla definována na základě připojení k LDAP a Kerberos serveru.

Soubor common-auth, který definuje autentizační pravidla:

```
vim /etc/pam.d/common-auth
```

Do tohoto souboru zapíšeme následující řádky:

```
auth    sufficient        pam_unix.so nullok_secure
auth    sufficient        pam_krb5.so use_first_pass
auth    required          pam_deny.so
```

Tato konfigurace znamená, že nejdříve bude uživatel autentizován oproti lokální databázi (/etc/passwd), přičemž je povoleno přihlášení bezpečných služeb bez hesla. V případě neúspěchu bude proveden pokus o autentizaci proti Kerberos serveru s použitím původního hesla. Pokud ani zde autentizace neprojde, bude uživatel odmítnut.

Soubor common-password, který definuje zpracování hesel:

```
vim /etc/pam.d/common-password
```

Do tohoto souboru zapíšeme následující řádky:

```
password  sufficient   pam_unix.so nullok obscure md5
password  sufficient   pam_krb5.so use_first_pass
password  required     pam_deny.so
```

Z této konfigurace vyplývá, že uživatelovo zadané heslo bude nejprve porovnáno s lokální databází, kde má právo změnit prázdné heslo, to bude šifrováno pomocí algoritmu MD5 a budou umožněny další kontroly (délka hesla apod.). Následně bude totéž heslo porovnáno s databází Kerbera. Pokud ani jeden pokus neuspěje, bude heslo odmítnuto.

Soubor common-session, který definuje vlastnosti uživatelovy relace

```
sudo vim /etc/pam.d/common-session
```

Do tohoto souboru zapíšeme následující řádky:

```
session required     pam_unix.so
session required     pam_mkhomedir.so skel=/etc/skel/ umask=0077
```

```
session optional     pam_krb5.so minimum_uid=1000
```

Tímto definujeme, že je vyžadováno spojení k systému, v případě nenalezení domovského adresáře bude uživateli vytvořen jeho domovský adresář s veškerými přístupovými právy pouze pro něj samotného, a nepovinně může vytvořit relaci k serveru Kerbera, pokud jeho UID je vyšší než 1000 (tzn. tuto relaci nemohou vytvořit služby).

j) Instalace a konfigurace dalších podpůrných prostředků

Tato kapitola popisuje instalaci webového serveru a skriptovacích enginů s potřebnými moduly.

Nejprve doinstalujeme potřebné moduly pro Python:

```
aptitude install python-{mysqldb,ldap,pexpect}
```

Dále nainstalujeme php5 včetně modulů a webového serveru Apache:

```
aptitude install php5
aptitude install php5-{cli,common,mysql}
```

Na webovém serveru zapneme autentizaci pro kořenový adresář, kde bude umístěn skript sloužící k procházení záznamů, pomocí úpravy souboru /etc/apache2/sites-enabled/000-default

```
vim /etc/apache2/sites-enabled/000-default
```

Zde najdeme sekci `<Directory /var/www>`, kterou upravíme následujícím způsobem:

```
<Directory /var/www>
        AuthType Basic
        AuthName „Audit"
        AuthUserFile /etc/apache2/audit.acl
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
        Require valid-user
</Directory>
```

Tím zapneme pro kořenový adresář autentizaci proti souboru /etc/apache2/audit.acl. Ten vytvoříme např. pro uživatele „User“ s heslem „password“ tímto způsobem:

```
htpasswd –n –b User password > /etc/apache2/audit.acl
```

Následně nainstalujeme MySQL klienta pro připojení k databázi:

```
aptitude install mysql-client
```

Pomocí nainstalovaného klienta se připojíme k MySQL serveru (v našem případě s adresou 192.168.122.1, uživatelem „User“, heslem „heslo“ a jménem databáze „IM“):

```
mysql –h 192.168.122.1 –u User –pheslo IM
```

V této konzoli vytvoříme novou tabulku s názvem „udalosti“:

```
create table udalosti (id int(12) not null auto_increment, timestamp int(12), username char(25), ip_addr char(15), event char(12), skupina char(20), primary key(id));
```

Tato tabulka bude obsahovat následující sloupce:

- `id`: unikátní ID události, které se bude plnit automaticky
- `timestamp`: UNIXový timestamp události
- `username`: uživatelské jméno spojené s událostí
- `ip_addr`: IP adresa, ze které byla událost generována
- `event`: typ události
- `skupina`: název skupiny, kam patří uživatel, který událost vygeneroval

k) Kopírování a konfigurace vytvořených skriptů a souborů

Vhodným způsobem, například pomocí SCP klienta na server nakopírujeme jednotlivé složky do těchto umístění:

- adresář „im-server“ s konfiguračními soubory do adresáře /etc
- adresář „skripty“ do adresáře /root
- adresář „ldif“ do adresáře /root
- soubor index.php do adresáře /var/www
- soubor audit.css do adresáře /var/www/css

Nově vytvořenému adresáři im-server v /etc přidělíme přístupová práva pouze pro roota:

```
chmod 700 /etc/im-server
chmod 600 /etc/im-server/*
```

Upravíme základní konfigurační soubor systému im-server.conf na základě skutečnosti:

```
vim /etc/im-server/im-server.conf
```

Významy jednotlivých konfiguračních položek jsou následující:

- `myAddress`: emailová adresa, pod kterou bude systém rozesílat emaily administrátorovi, vyučujícím a studentům
- `SMTPServer`: IP adresa SMTP serveru, který bude použit pro rozesílání emailů
- `adminAddress`: emailová adresa administrátora systému
- `mySqlServer`: IP adresa použitého MySQL serveru
- `mySqlDb`: jméno použité databáze

- `mySqlUsername`: uživatelské jméno na MySQL serveru (tento účet musí mít právo pro zápis do tabulky "udalosti")
- `mySqlPassword`: heslo uživatele do MySQL
- `ldapServer`: IP adresa použitého LDAP serveru
- `ldapUsername`: uživatelské jméno administrátora LDAP serveru, je nutné jej uvést jako distinguished name, například "cn=admin,dc=utko,dc=feec,dc=vutbr,dc=cz"
- `ldapPassword`: heslo administrátora LDAP serveru
- `passwordLength`: délka hesla, které se bude automaticky generovat a posílat na email nově vytvářeným uživatelům systému
- `studMailDomain`: doména, na kterým budou odesílány emaily studentům
- `badLoginTreshold`: Počet po sobě jdoucích chybných loginů jednoho uživatele, po kterých bude odeslán email administrátorovi
- `krbRealm`: název použitého Kerberos realmu

Dále je na začátku skriptu /var/www/index.php třeba upravit tyto proměnné:

- `$mySqlServer` – adresa MySQL serveru
- `$mySqlDB` – jméno použité databáze
- `$mySqlUsername` – přihlašovací jméno
- `$mySqlPassword` – heslo
- `$tolerancePred` – jak dlouho před začátkem hodiny bude přihlášení studenta zobrazeno jako bezproblémové (celé číslo v sekundách, lze zadat i zápornou hodnotu). Výchozí hodnota je 900, tedy čtvrt hodiny.
- `$tolerancePo` – jak dlouho po konci hodiny bude přihlášení studenta zobrazeno jako bezproblémové (celé číslo v sekundách, lze zadat i zápornou hodnotu). Výchozí hodnota je 0.

Následně pomocí souboru struktura.ldif vytvoříme základní strukturu LDAP serveru:

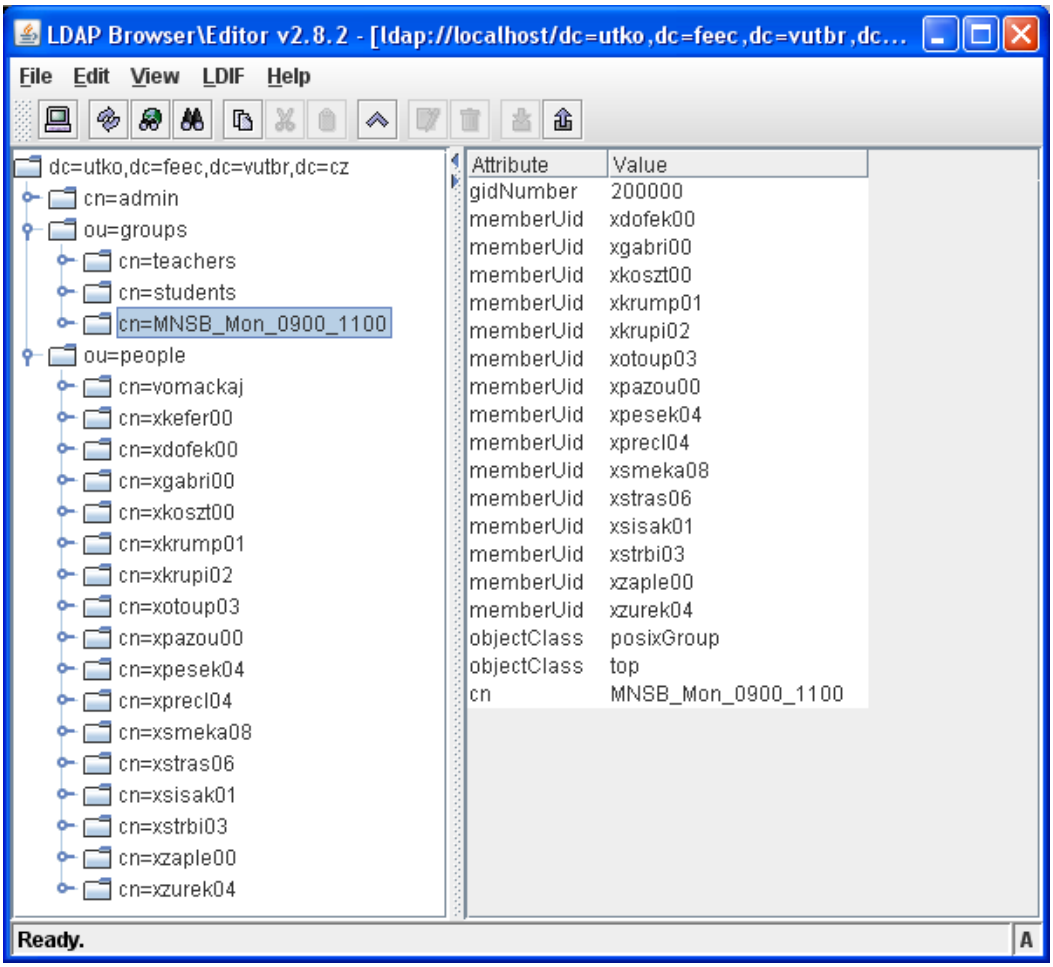
```
ldapadd -x -D -W „cn=admin,dc=utko,dc=feec,dc=vutbr,dc=cz“ -f /root/ldif/struktura.ldif
```

Použitá struktura má následující podobu:

- ou=groups: do této organizační jednotky jsou ukládány všechny skupiny
  - cn=teachers: skupina, do které jsou přidáváni učitelé
  - cn=students: výchozí skupina všech studentů
  - dále jsou zde vytvářeny jednotlivé vyučovací skupiny při jejich zadávání do systému

- ou=people: do této organizační jednotky jsou ukládáni všichni uživatelé systému (vyučující i studenti)

Příklad vytvořené struktury již s importovanými uživateli a jednou skupinou je zobrazen na obrázku:



Obr. 10: příklad funkční vytvořené struktury na LDAP serveru

Dále vytvoříme záznam v crontabu, pomocí kterého budeme periodicky spouštět skript dolující logy z textového souboru, odesílající je do databáze a posílající emaily administrátorovi systému:

```
crontab -e
```

Zde doplníme následující řádek:

```
* * * * * /usr/bin/php5 /root/skripty/dolujLogy.php > /dev/null
```

Prvních pět hvězdiček znamená, že skript se bude spouštět každou minutu, každou hodinu, každý den v měsíci, každý měsíc v roce, každý den v týdnu. Následuje příkaz pro spuštění skriptu a přesměrování jeho výstupu do /dev/null, tedy jeho zahazování. V případě problémů lze výstup přesměrovat do textového souboru.

3.4.3. Instalace klienta

Instalace klienta je v mnohých krocích podobná instalaci klientské části na serveru, proto budou komentovány pouze kroky, které se od ní liší.

Instalace PAM modulu pro Kerberos:

```
aptitude install libpam-krb5
```

spustí se rozhraní debconf, kde nakonfigurujeme tyto údaje:

```
Default Kerberos Version 5 realm: UTKO.FEEC.VUTBR.CZ
Does DNS contain pointers to your realm's Kerberos Servers? No
Kerberos servers for your realm: im-server.utko.feec.vutbr.cz
Administrative server for your Kerberos realm: im-server.utko.feec.vutbr.cz
```

Význam jednotlivých položek je stejný jako při instalaci serveru Kerbera.

Dále nainstalujeme a nakonfigurujeme balíčky pro LDAP:

```
sudo apt-get install libnss-ldap nscd
```

```
LDAP server Uniform Resource Identifier: ldap://192.168.122.97/
Distinguished name of the search base: dc=utko,dc=feec,dc=vutbr,dc=cz
LDAP version to use: 3
Does the LDAP database require login? No
Special LDAP privileges for root? No
Make the configuration file readable/writeable by its owner only? No
Make local root Database admin. No
Does the LDAP database require login? No
Local crypt to use when changing passwords. crypt
```

Do souboru /etc/ldap/ldap.conf zapíšeme následující dva řádky:

```
BASE  dc=utko,dc=feec,dc=vutbr,dc=cz
URI ldap://192.168.122.97/
```

V souboru /etc/nsswitch.conf upravíme následující dva řádky:

```
passwd:         compat ldap
group:          compat ldap
```

Upravíme soubor /etc/libnss-ldap.conf

```
bind_policy soft
```

Nainstalujeme knihovny pro PAM LDAPu:

```
aptitude install libpam-ldap
```

Upravíme konfigurační soubory PAM:

- /etc/pam.d/common-account:

```
account sufficient      pam_unix.so
account required        pam_ldap.so
account required        pam_krb5.so
```

- **/etc/pam.d/common-auth:**

```
auth    sufficient        pam_unix.so nullok_secure
auth    sufficient        pam_krb5.so use_first_pass
auth    required          pam_deny.so
```

- **/etc/pam.d/common-password**

```
password  sufficient   pam_unix.so nullok obscure md5
password  sufficient   pam_krb5.so use_first_pass
password  required     pam_deny.so
```

- **/etc/pam.d/common-session**

```
session required     pam_unix.so
session required     pam_mkhomedir.so skel=/etc/skel/ umask=0077
session optional     pam_krb5.so minimum_uid=1000
```

Následně nainstalujeme balíčky pro klientskou stranu NFS:

aptitude install nfs-common portmap

V případě NFS se musí Kerberos principalem prokazovat i klientský stroj. Proto nainstalujeme klientské nástroje s administrační konzoli kadmin:

aptitude install krb5-user

Dále spustíme konzoli kadmin, ke které se autentizujeme pomocí hesla k principalu root:

kadmin

Zde vytvoříme principal stanice s náhodným heslem:

addprinc – randkey nfs/im-client.utko.feec.vutbr.cz

Tento principal uložíme do systémového keytabu, je však nutné použít pouze jediný typ šifrování, které NFS podporuje:

ktadd -e des-cbc-crc:normal nfs/im-client.utko.feec.vutbr.cz

Administrační konzoli kadmin opustíme pomocí příkazu

exit

a upravíme soubor /etc/default/nfs-common:

```
NEED_IDMAPD=yes
NEED_GSSD=yes
```

Do souboru /etc/idmapd.conf zadáme doménové jméno:

```
Domain = utko.feec.vutbr.cz
```

Dále vytvoříme adresář, do kterého budeme mountovat obsah adresáře share pro sdílená data:

```
mkdir /mnt/share
```

Otevřeme pro editaci konfigurační soubor fstab, který definuje jednotlivé mounty:

```
vim /etc/fstab
```

Zde dopíšeme následující 2 řádky:

```
im-server.utko.feec.vutbr.cz:/home          /home          nfs4
rw,soft,sec=krb5i  0  0
im-server.utko.feec.vutbr.cz:/share      /mnt/share      nfs4
   rw,soft,sec=krb5i  0  0
```

Tím definujeme následující skutečnosti:

- domovské adresáře ze serveru budou mountovány přímo do adresáře /home pomocí protokolu NFS4, bude zde povolen zápis, v případě jeho nenalezení při startu systém korektně nabootuje, autentizace proběhne pomocí protokolu Kerberos s kontrolou integrity, adresář nebude lokálně zálohován ani kontrolován
- Obsah serverového adresáře pro sdílení dat bude mountován do adresáře /mnt/share se stejnými parametry jako předchozí adresář.

3.4.4. Popis vytvořených skriptů a práce s uživateli

Správa uživatelů i další konfigurace již probíhá výlučně na straně serveru. Vytvořené skripty a konfigurační soubory jsou umístěny v celkem třech umístěních.

V adresáři /root/skripty jsou umístěny tyto soubory:

- addadmin.py – skript pro přidání nového vyučujícího
- addusers.py – skript pro přidání nové skupiny studentů
- deleteadmin.py – skript pro smazání vyučujícího
- deleteusers.py – skript pro smazání existující skupiny studentů
- dolujLogy.php – skript pro dolování událostí z textových souborů a notifikaci administrátora o podezřelých událostech
- sendmail.py – skript pro odesílání emailů

Dále v adresáři /var/www jsou umístěny následující soubory:

- index.php – skript tvořící webové rozhraní pro audit událostí
- /css/audit.css – nastavení CSS (Cascading Style Sheets) pro skript index.php

Nakonec v adresáři /etc/im-server jsou umístěny tyto konfigurační soubory:

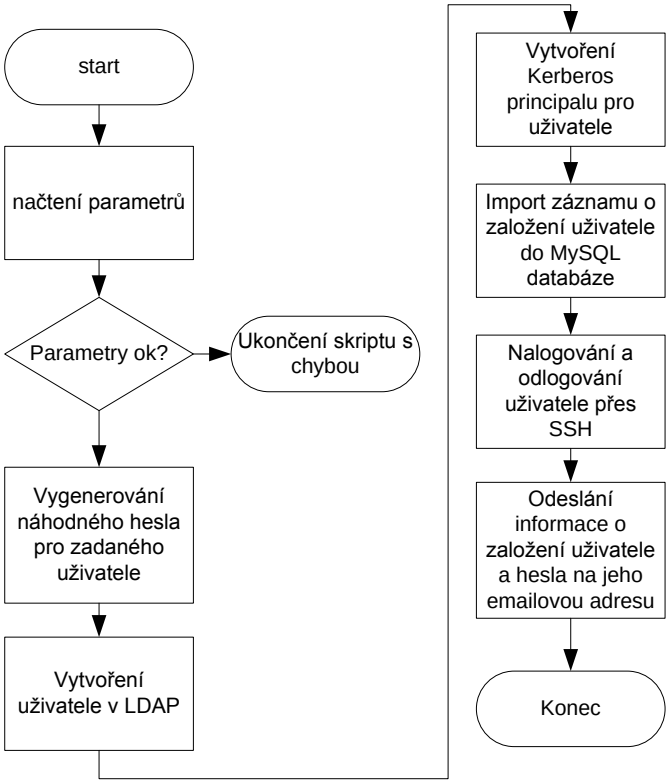
- badlogin.tpl – šablona pro odesílání emailů popisující nekorektní pokusy o přihlášení

- im-server.conf – základní konfigurační soubor vytvořeného systému, který již byl popsán v předchozí kapitole
- registration.tpl – šablona pro odesílání emailů nově zavedeným uživatelům

Důležité soubory a práce s nimi jsou popsány v následujících podkapitolách.

l) addadmin.py

Tento skript slouží k importu vyučujících do systému. Funkce skriptu je vysvětlena na následujícím vývojovém diagramu:



Obr. 11: Funkce skriptu addadmin.py

Za zmínku zde stojí automatizované SSH připojení k serveru, které je realizované pomocí modulu pexpect pro realizaci interaktivní komunikace. Toto připojení slouží k vytvoření počátečního prostředí uživatele (pokud by se rovnou připojil na klienta, neměl by potřebná práva k jeho vytvoření).

Skript očekává následující parametry:

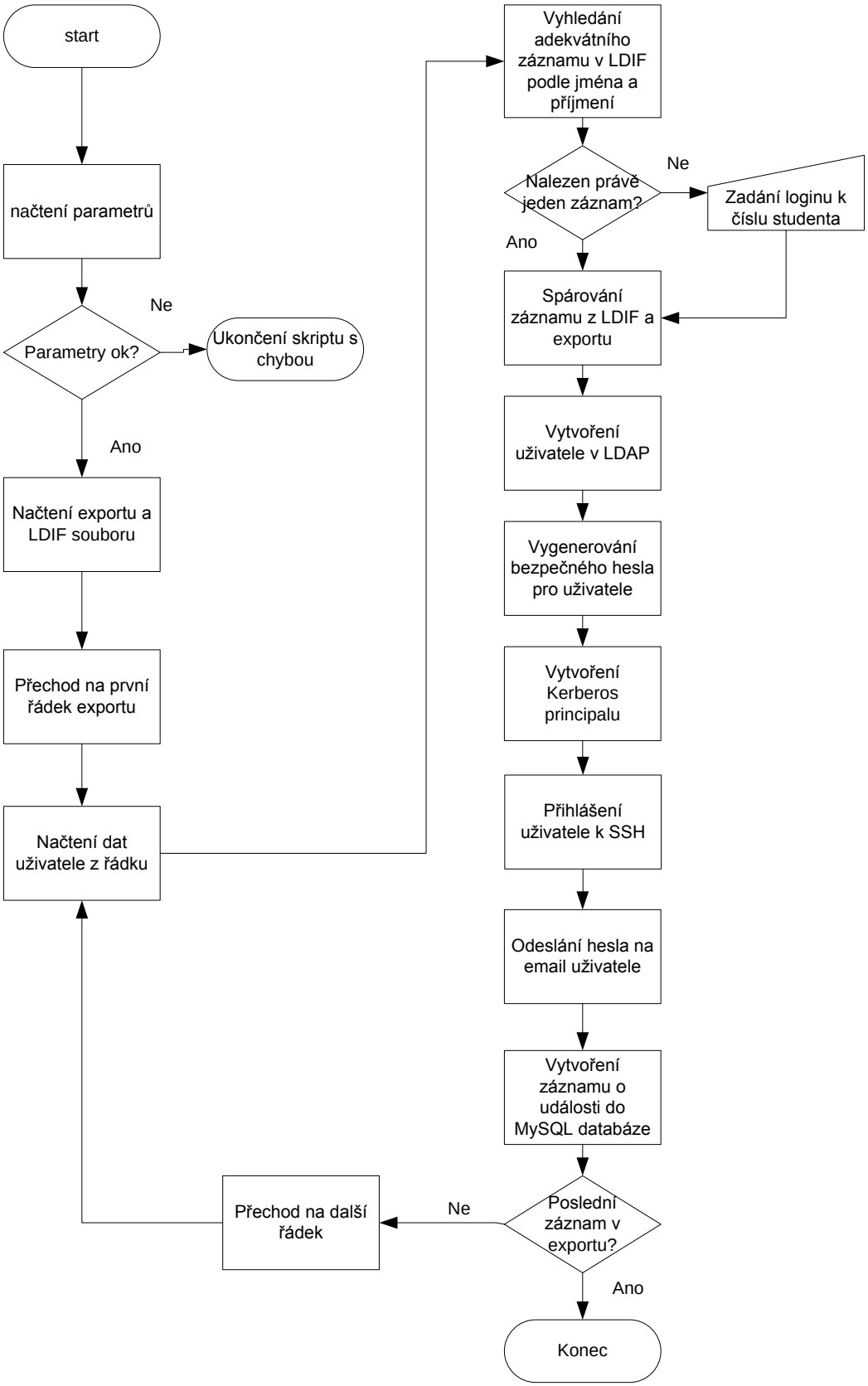
- -h: zobrazení nápovědy
- -u: uživatelské jméno zakládaného uživatele
- -g: jeho křestní jméno
- -s: příjmení
- -m: emailová adresa, na kterou bude uživatelovi odesláno oznámení o založení jeho účtu a vygenerované heslo

Přiklad spuštění skriptu:

```
python addadmin.py –u novakj –g Josef –s Novák –m novak@feec.vutbr.cz
```

m)addusers.py

Pro import je třeba mít aktuální vyexportovaný LDIF soubor z novellovské databáze pomocí utility ice.exe z MS Windows či ldapsearch z Linuxu a dále export studijní skupiny z univerzitního IS Apollo ve formátu csv, kterou chceme naimportovat. Skript vždy pro každý záznam z IS nalezne na základě jména a přijmení odpovídající záznam v LDIF souboru. V případě nalezení více záznamů požádá o ruční spárování identifikačního čísla a loginu. V rámci jednoho spuštění skriptu je možné naimportovat jednu studijní skupinu, která je identifikována pomocí předmětu a dne a času začátku a konce výuky. Funkce skriptu je znázorněna pomocí vývojového diagramu:



Obr. 12: Vývojový diagram skriptu addusers.py

Parametry skriptu:

- -h: zobrazí nápovědu

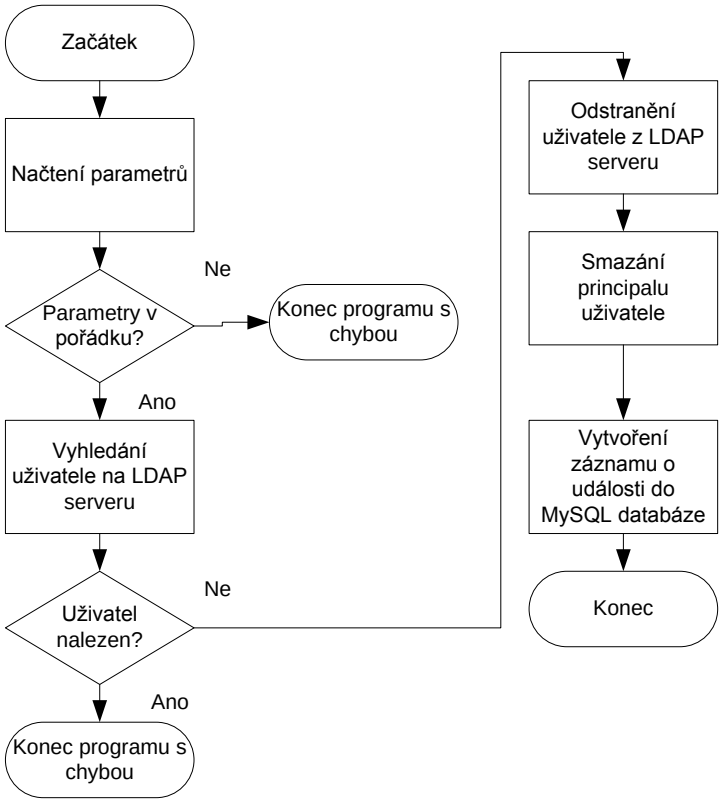
- -l: určuje cestu k LDIF souboru z Novellovské db
- -e: určuje cestu k souboru s exportem skupiny z univerzitního IS
- -g: identifikátor studijní skupiny ve formátu predmet_D_hhmm_hhmm, který specifikuje předmět, den v týdnu (anglický zkrácený formát názvu dne) a čas začátku a konce výuky, př. MBIS_Tue_1400_1600

Přiklad spuštění:

```
python addusers.py -l feec.ldif -e export.txt -g MBIS_Tue_1400_1600
```

n) deleteadmin.py

Tento skript slouží k odebrání vyučujícího ze systému. Jeho funkce je opět vysvětlena pomocí vývojového diagramu:



Obr. 13: Vývojový diagram skriptu deleteadmin.py

Skript očekává následující parametry:

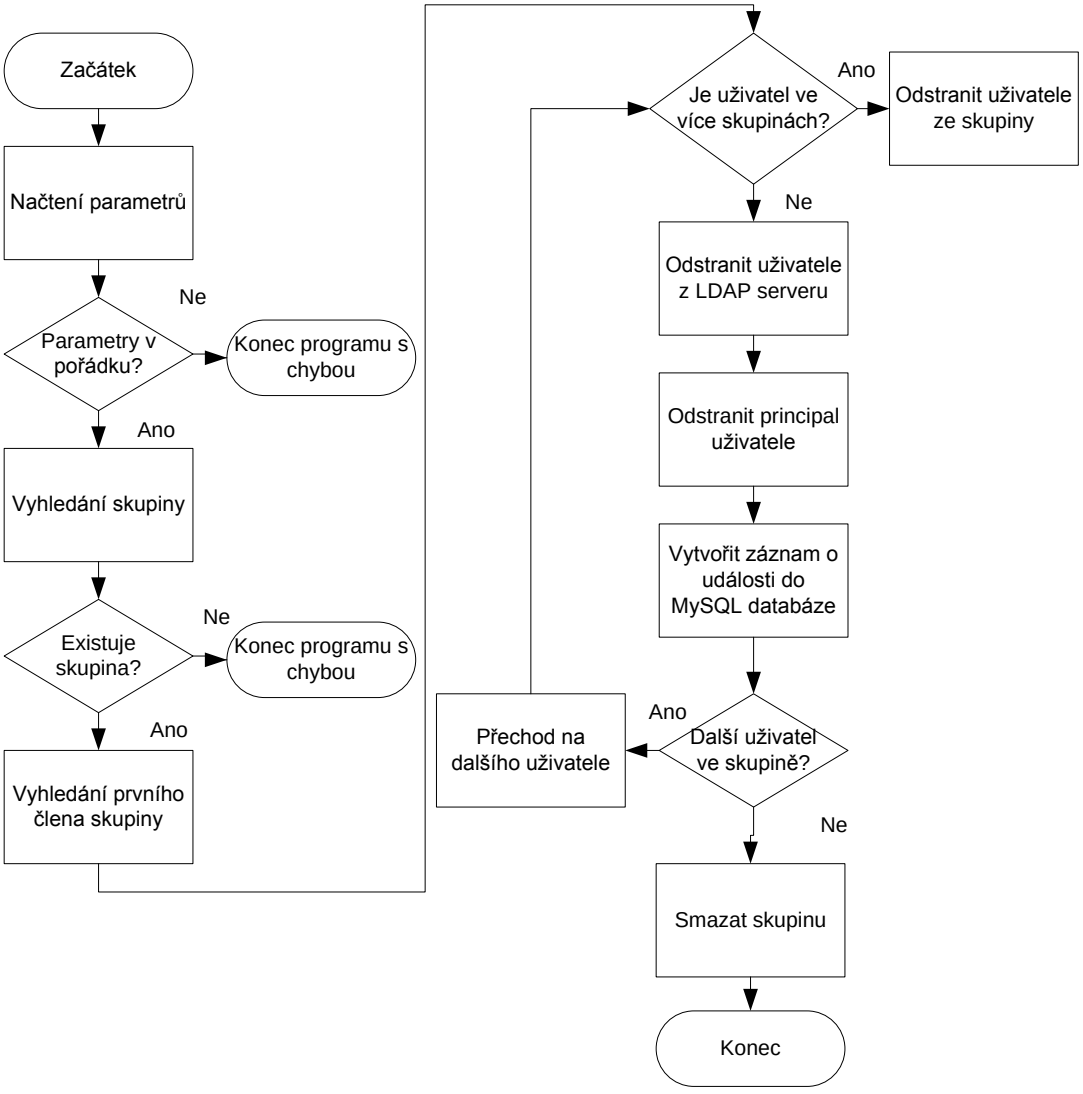
- -h: zobrazení nápovědy
- -u: uživatelské jméno uživatele, který má být odstraněn ze systému

Příklad spuštění skriptu:

```
python deleteadmin.py -u novakj
```

o) deleteusers.py

Tento skript odstraňuje vybrané skupiny, pokud je uživatel ve více studijních skupinách zároveň, potom pouze zruší jeho členství ve skupině, ale záznam samotného uživatele neodstraní. Funkce skriptu je znázorněna na následujícím vývojovém diagramu:



Obr. 14: vývojový diagram skriptu deleteusers.py

Parametry skriptu:

- -h: zobrazí nápovědu
- -g: identifikátor studijní skupiny určené ke smazání, ve formátu predmet_D_hhmm_hhmm, který specifikuje předmět, den v týdnu (anglický zkrácený formát názvu dne) a čas začátku a konce výuky, př. MBIS_Tue_1400_1600

Příklad spuštění:

```
python deleteusers.py -g MBIS_Tue_1400_1600
```

p) dolujLogy.php

Tento skript je periodicky spouštěn jednou za minutu pomocí cronu. Slouží k parsování logů Kerbera a import událostí do MySQL databáze. Zároveň vyhodnocuje podezřelé události v síti a informuje o nich administrátora. Aktuálně je sledován počet po sobě jdoucích chybných přihlášení každého uživatele následujícím způsobem:

- vytvoření pole všech uživatelů, kteří se v poslední minutě pokusili přihlásit se špatným heslem
- pro každého takového uživatele je z databáze extrahována historie s ním spojených událostí
- záznamy jsou pozpátku procházeny, dokud není nalezeno úspěšné přihlášení, nebo není dosažena hranice pro informování administrátora (ve výchozím nastavení 5 pokusů o přihlášení s chybným heslem po sobě)
- Pokud je dosaženo této hranice, odešle se administrátorovi informační email o podezřelé aktivitě v síti

Pokud praxe ukáže, že je vhodné sledovat další podezřelé události v síti, lze je do tohoto skriptu snadno doprogramovat.

q) sendmail.py

Tento skript je volán ostatními skripty vždy, pokud je třeba odeslat jakýkoli email. Je volán s následujícími parametry:

- -h: zobrazí nápovědu
- -m: emailová adresa, na kterou bude email odeslán. Pokud zde bude místo emailové adresy uveden parametr „admin“, bude email odeslán na adresu administrátora, která je definována v souboru /etc/im-server/im-server.conf
- -t: název souboru šablony z adresáře /etc/im-server (bez přípony „.tpl“), která bude použita pro odeslání emailu.
- -p: jednotlivá spojení, která budou nahrazena v uvedené šabloně.

Pokud tedy tento skript bude zavolán následujícím způsobem:

```
python sendmail.py –m admin –t badlogin –p
„username=xkefer00,treshold=5“
```

administrátorovi systému bude odeslán mail podle šablony v souboru /etc/im-server/badlogin.tpl, kde proměnná „username“ bude nahrazena za „xkefer00“ a proměnná „treshold“ za hodnotu „5“.

r) badlogin.tpl

Tento konfigurační soubor představuje šablonu pro odeslání emailu administrátorovi v případě detekce překročení „povoleného“ počtu chybných pokusů o přihlášení. Implicitně má tuto podobu:

```
Subject: Podezrela aktivita uzivatele $USERNAME

Uzivatel $USERNAME zadal $TRESHOLD-krat po sobe spatne heslo.
```

Syntaxe tohoto souboru je následující. Na prvním řádku je za označením „Subject:“ uveden předmět zprávy. Tělo je pak od předmětu odděleno prázdným řádkem. Řetězce, které jsou uvozeny znakem dolaru a uvedeny velkými písmeny, budou skriptem sendmail.py nahrazeny za poskytnuté hodnoty.

s) registration.tpl

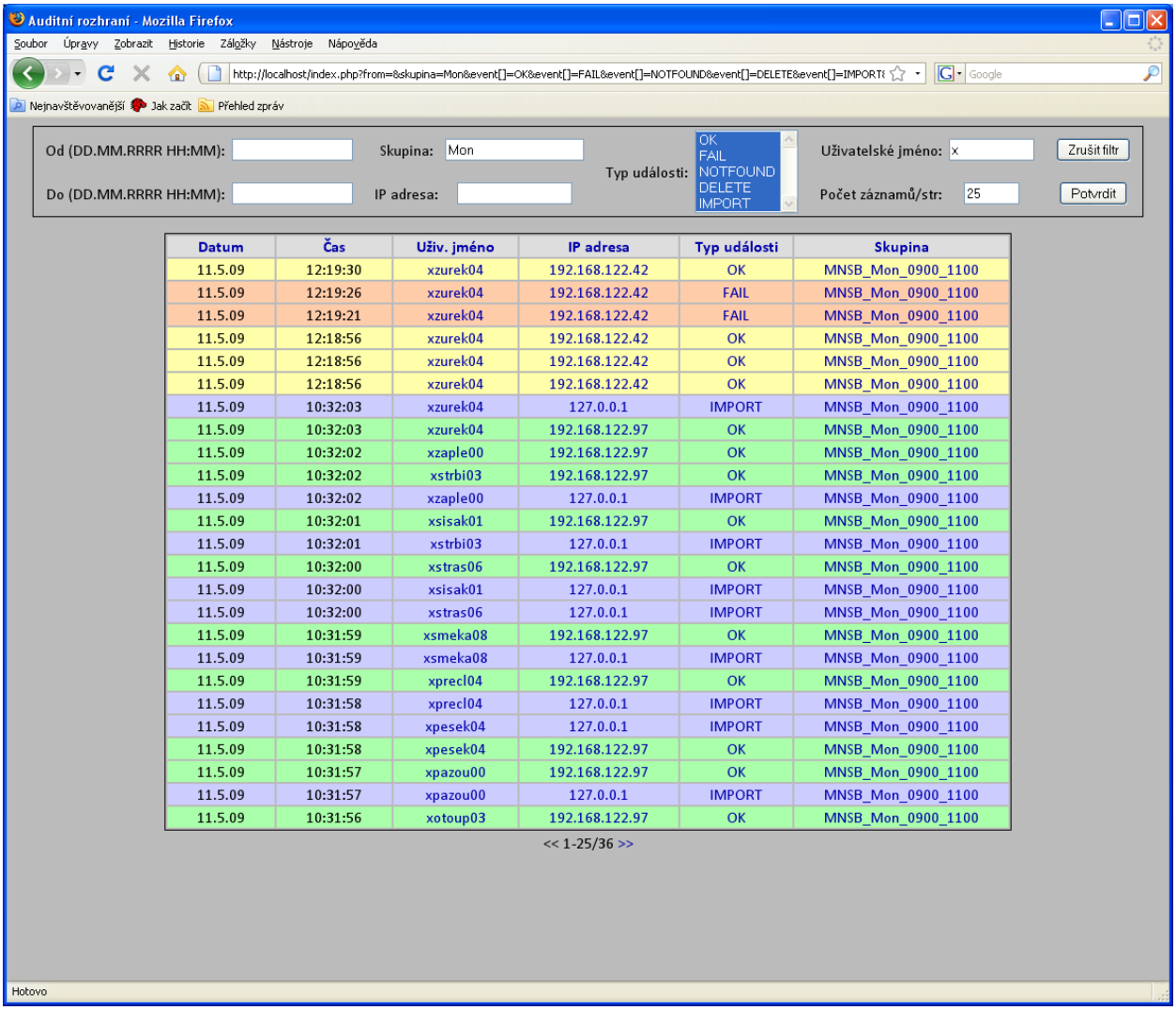
Tato šablona je využita při vytváření nových uživatelů, kdy je pomocí ní odesílán email s uživatelským jménem a vytvořeným heslem. Syntaxe je stejná jako u předešlého souboru a v implicitním nastavení má tuto podobu:

```
Subject: Registrace do systemu

Dobry den,
byl Vam zrizen pristup do systemu Debian.
login: $USERNAME
heslo: $PASSWORD

S pozdravem administrator systemu
```

Obě uvedené šablony je vhodné upravit na základě konkrétních potřeb/požadavků.

t) index.php

Jak již bylo zmíněno, tento skript slouží ve spojení s webovým serverem jako front-end k procházení bezpečnostních událostí v síti. Na obrázku je znázorněn příklad zobrazení událostí:



Obr. 15: Front-endové rozhraní pro kontrolu událostí v síti

V horní části je formulář pro filtrování záznamů s následujícími položkami:

- Od (DD.MM.RRRR HH:MM): budou zobrazeny pouze události uskutečněné po zadaném datu a čase.
- Do (DD.MM.RRR HH:MM): zobrazení událostí uskutečněných před zadaným datem a časem.
- Skupina: zobrazení událostí uživatelů ze zadané skupiny. Lze zadat jen část řetězce a tak vyfiltrovat například události uživatelů všech skupin jednoho předmětu či zadaného dne. Na obrázku jsou filtrovány událostí všech skupin mající výuku v pondělí.
- IP adresa: lze sledovat pouze události ze specifikované IP adresy. Opět lze zadat jen část řetězce a vyfiltrovat tak například události z jednoho adresního rozsahu.
- Typ události: budou zobrazeny pouze události s uvedenými typy.

- Uživatelské jméno: zobrazení událostí ke specifikovanému uživateli. Lze zadat jen část řetězce, čímž lze například pomocí filtru „x“ vyfiltrovat pouze události studentů.
- Počet záznamů/str: tato hodnota udává počet událostí zobrazených na jedné straně.

Uvedená filtrovací pravidla lze libovolně kombinovat.

Ve spodní části je pak tabulka s výpisem samotných událostí. Ty jsou rozlišeny pomocí barvy následovně:

- modrá barva označuje import a mazání uživatelů
- zelená barva označuje úspěšné přihlášení uživatele
- žlutá barva označuje rovněž úspěšné přihlášení uživatele, ovšem mimo časové rozpětí výuky své skupiny
- červená barva s oranžovým odstínem označuje neúspěšný pokus o přihlášení
- červená barva s růžovým odstínem označuje pokus o přihlášení uživatele, který není veden v databázi

Při kliknutí na odkaz v záhlaví je provedeno seřazení událostí podle daného kritéria. Při každém následujícím kliknutí je změněn směr řazení (vzestupně/sestupně). Samotné položky jednotlivých událostí „uživatelské jméno“, „IP adresa“, „Typ události“ a „skupina“ jsou zobrazeny jako odkazy. Kliknutím na konkrétní položku lze rychle vyfiltrovat všechny události týkající se této položky.

4. Závěr

V rámci diplomové práce byla nejprve zmapována a popsána oblast identity managementu na teoretické bázi, následně bylo přistoupeno k realizaci praktického projektu centrálni autentizace uživatelů pro počítačovou učebnu. Ta je postavena na OS Linux Debian (serverová i klientská část), volné implementaci MIT protokolu Kerberos a LDAP serveru OpenLDAP. Import uživatelů je řešený pomocí skriptů vyvinutých v jazyce Python. Dolování záznamů z textových souborů, upozorňování administrátora na podezřelé události a front-endové rozhraní pro audit událostí je realizováno pomocí jazyka PHP. Jako vzorová služba pro demonstraci single sign-on byla vybrána služba NFS, díky čemuž mají uživatelé možnost pracovat nezávisle na konkrétních stanicích a sdílet soubory.

Systém je navržen tak, aby fungoval především jako platforma pro bezpečnou autentizaci uživatelů a bylo možné jej dále rozvíjet na základě konkrétních potřeb. Lze tak postupně implementovat další služby využívající mechanismus single-sign on, či přidávat mechanismy pro upozorňování administrátora na nové typy bezpečnostních událostí. Všech uvedených výsledků bylo dosaženo za použití výhradně volně dostupných technologií.

Seznam použité literatury

[1] *Glossary of Terms* [online]. 2008 [cit. 2008-12-15]. Dostupný z WWW: <http://www.corestreet.com/glossary/>.

[2] CLARKE, Roger. *Identification and Authentication Glossary* [online]. 2004 [cit. 2008-12-15]. Dostupný z WWW: <http://www.anu.edu.au/people/Roger.Clarke/EC/IdAuthGloss.html>.

[3] *Information and Communications Technology Strategic Plan : Appendix F: Glossary* [online]. 2005 [cit. 2008-12-15]. Dostupný z WWW: <http://www.ict.ox.ac.uk/strategy/plan/plan.xml.ID=appF>.

[4] *Wikipedia : Identity management* [online]. 2008 [cit. 2008-12-15]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Identity_management>

[5] *Fraud Detection in Communications Networks Using Neural and Probabilistic Methods* [online]. 1998 [cit. 2008-12-15]. Dostupný z WWW: <http://www.cis.hut.fi/jhollmen/Publications/icassp98.pdf>.

[6] *The Definitive Guide to Identity Management* [online]. 2004 [cit. 2008-12-15]. Dostupný z WWW: <http://nexus.realtimepublishers.com/content/DGIMFinal.pdf>.

[7] *Bind9 - DNS, BIND, DHCP, LDAP and Directory Services* [online]. 2002-2009 [cit. 2009-05-17]. Dostupný z WWW: <http://www.bind9.net/>.

[8] *Debian Administration* [online]. 2004-2008 [cit. 2008-12-15]. Dostupný z WWW: <http://www.debian-administration.org>.

[9] *Kerberos: The Network Authentication Protocol* [online]. 2007 [cit. 2008-12-15]. Dostupný z WWW: <http://web.mit.edu/Kerberos/>.

[10] *OpenLDAP : Project Home Page* [online]. 2008 [cit. 2008-12-15]. Dostupný z WWW: <http://www.openldap.org/>.

[11] *Python Programming Language : Official Website* [online]. 1990-2008 [cit. 2008-12-15]. Dostupný z WWW: <http://www.python.org/>.

[12] *Debian : Univerzální operační systém* [online]. 1997-2008 [cit. 2008-12-15]. Dostupný z WWW: <http://www.debian.org/>.

[13] *Project America : Crime: Crime Rates: Internet Crime* [online]. 2008 [cit. 2008-12-15]. Dostupný z WWW: <http://www.project.org/info.php?recordID=166>.

[14] *Association of Certified Fraud Examiners : Organisation Website* [online]. 2007 [cit. 2008-12-15]. Dostupný z WWW: <http://www.acfe.com/>.

[15] GLASS, Eric. *The NTLM Authentication Protocol and Security Support Provider* [online]. 2003-2006 [cit. 2008-12-15]. Dostupný z WWW: <http://davenport.sourceforge.net/ntlm.html>.

[16] OAuth Core 1.0 Draft 4 [online]. 2007 [cit. 2009-05-17]. Dostupný z WWW: <http://oauth.googlecode.com/svn/spec/branches/1.0/drafts/4/spec.html>.

[17] *OpenID and Rails: Authentication 2.0* [online]. 2008 [cit. 2009-05-17]. Dostupný z WWW: <http://www.devx.com/opensource/Article/37692>.

[18] *Čo je to GRID Card?* [online]. 2009 [cit. 2009-05-17]. Dostupný z WWW: <http://www.tatrabanka.sk/cms/page/sk/fyzicke_osoby/elektronicke_bankovnictvo/internet_banking/grid_card.html>.

[19] *Smart card* [online]. 2009 [cit. 2009-05-17]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Smart_card>.

[20] *Security token* [online]. 2008 [cit. 2009-05-17]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Security_token>.

[21] *Personal identification number* [online]. 2008 [cit. 2009-05-17]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Personal_identification_number>.

[22] *Discretionary access control (DAC)* [online]. 2004 [cit. 2009-05-17]. Dostupný z WWW: <http://ou800doc.caldera.com/en/SEC_admin/IS_DiscretionaryAccCntlDAC.html>.

[23] *Mandatory Access Control* [online]. 2002 [cit. 2009-05-17]. Dostupný z WWW: <http://www.cgisecurity.com/owasp/html/ch08s02.html>.

[24] *Role-Based Access Controls* [online]. 1992 [cit. 2009-05-17]. Dostupný z WWW: <http://csrc.nist.gov/groups/SNS/rbac/documents/ferraiolo-kuhn-92.pdf>.

[25] *Single sign-on* [online]. 2006 [cit. 2009-05-17]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Single_sign-on>.

[26] *Samba - Opening Windows to a Wider World* [online]. 2009 [cit. 2009-05-17]. Dostupný z WWW: <http://us3.samba.org/samba/>.

[27] *FreeIPA* [online]. 2007 [cit. 2009-05-17]. Dostupný z WWW: <http://www.freeipa.org/page/Main_Page>.

[28] *OAuth* [online]. 2007 [cit. 2009-05-17]. Dostupný z WWW: <http://oauth.net/>.

[29] *OpenID Homepage* [online]. 2007 [cit. 2009-05-17]. Dostupný z WWW: <http://openid.net/>.

[30] *OpenID: Decentralised Single Sign-on for the Web* [online]. 2007 [cit. 2009-05-17]. Dostupný z WWW: <http://www.ariadne.ac.uk/issue51/powell-recordon/>.

[31] *PKI applications (C2)* [online]. 2007 [cit. 2009-05-17]. Dostupný z WWW: <http://pst.libre.lu/mssi-luxmbg/p1/04_auth-art.html>.

[32] *The Apache Software Foundation* [online]. 2009 [cit. 2009-05-17]. Dostupný z WWW: <http://www.apache.org/>.

[33] *MySQL :: The world's most popular open source database* [online]. 1995-2008 [cit. 2009-05-17]. Dostupný z WWW: <http://www.mysql.com/>.

[34] *PHP: Hypertext Preprocessor* [online]. 2001-2009 [cit. 2009-05-17]. Dostupný z WWW: <http://www.php.net/>.

[35] *Network File System Version 4 (nfsv4)* [online]. 2009 [cit. 2009-05-17]. Dostupný z WWW: <http://www.ietf.org/html.charters/nfsv4-charter.html>.

[36] HAWES, Timothy K.. *Fraud Detection Systems : Application of Familiar Technology to Minimize Losses* [online]. 1999 [cit. 2009-05-17]. Dostupný z WWW: <http://www.information-management.com/issues/19990201/160-1.html>.

[37] STEVENS, Matt. *Security Information and Event Management (SIEM)* [online]. 2005 [cit. 2009-05-17]. Dostupný z WWW: <http://www.certconf.org/presentations/2005/files/WC4.pdf>.

Sznam použitých zkratek

- APT – Advanced Package Tool
- CIFS – Common Internet File System
- CSS – Cascading Style Sheets
- DAC – Discretionary Access Control
- DB - Database
- DCE/RPC – Distributed Computing Environment / Remote Procedure Call
- DHCP – Dynamic Host Control Protocol
- DNA – Deoxyribonucleic Acid
- DNS – Domain Name System
- FDS – Fraud Detection System
- FEEC – Faculty of Electrical Engineering and Communication
- FEKT – Fakulta Elektrotechniky a komunikačních technologií
- FQDN – Fully Qualified Domain Name
- FTP – File Transfer Protocol
- GNU – Gnu's Not Unix
- HTTP – Hypertext Transfer Protocol
- ID - Identifier
- IP – Internet Protocol
- IS – Informační Systém
- JOSSO – Java Open Single Sign-On
- KDC – Key Distribution Center
- LDAP – Leightweight Directory Access Protocol
- LDIF – LDAP Data Interchange Format
- MAC – Mandatory Access Control
- MD5 – Message Digest 5
- MIT – Massachusetts Institute of Technology
- MS - Microsoft
- NFS – Network File System
- NS – Name Server
- NT – New Technology
- NTP – Network Time Protocol
- OS – Operating System

- PAM – Pluggable Authentication Module
- PHP – PHP: Hypertext Preprocessor
- PIN – Personal Identification Number
- PTR - Pointer
- RBAC – Role-Based Access Control
- RDBMS – Relational Database Management System
- ROC – Receiver Operating Characteristics
- RPC – Remote Procedure Call
- SCP – Secure Copy
- SELinux – Security Enhanced Linux
- SHA – Secure Hash Algorithm
- SMB – Server Message Block
- SMS – Short Message System
- SMTP – Simple Mail Transfer Protocol
- SOA – Start of Authority
- SSH – Secure Shell
- SSO – Single Sign-On
- SW - Software
- TGS – Ticket Granting Server
- TGT – Ticket Granting Ticket
- USB – Universal Serial Bus
- VUT – Vysoké učení technické
- WINS – Windows Internet Naming Service

Obsah přiloženého CD:

- DP.pdf (elektronická verze diplomové práce)
- DEVEL:
  - IM-SERVER:
    - badlogin.tpl – šablona emailu oznámení o podezřelé události
    - im-server.conf – konfigurační soubor
    - registration.tpl – šablona emailu oznámení o registraci uživatele
  - LDIF:
    - struktura.ldif – struktura LDAP serveru
  - SKRIPTY:
    - addadmin.py – přidání vyučujícího
    - addusers.py – přidání skupiny studentů
    - deleteadmin.py – smazání vyučujícího
    - deleteusers.py – smazání skupiny studentů
    - dolujLogy.php – plnění MySQL, odesílání alertů
    - sendmail.py – odesílání emailů
  - WWW:
    - CSS:
      - audit.css – CSS soubor pro auditní rozhraní
    - index.php – auditní rozhraní