Západočeská univerzita v Plzni
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

SPECIALIZOVANÁ SÍŤOVÁ VÝUKOVÁ UČEBNA S JEDNOTNOU UNIVERZITNÍ AUTENTIZACÍ

DIPLOMOVÁ PRÁCE

Bc. Ladislav Kulatý
Učitelství pro 2. stupeň ZŠ, obor Inf-Te

Vedoucí práce: Dr. Ing. Jiří Toman
Plzeň, 2014

Prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

Plzeň, 28. dubna 2014

……
vlastnoruční podpis

PODĚKOVÁNÍ

Na tomto místě bych rád poděkoval Dr. Ing. Jiřímu Tomanovi za věcné připomínky, rady a zkušenosti, kterými přispěl k vypracování této diplomové práce.

ZDE SE NACHÁZÍ ORIGINÁL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

ÚVOD ... 1
1 SYSTÉM JEDNOTNÉ AUTENTIZACE (SSO) ... 3
1.1 DŮVODY VZNIKU SSO ... 3
1.2 ČINNOST SSO ... 3
1.3 NEVÝHODY SSO ... 4
1.4 SSO NA ZČU ... 5
2 KERBEROS ... 6
2.1 PRINCIP ČINNOSTI ... 6
2.2 KERBEROS NA ZČU ... 9
3 SSO NA ZČU PODROBNĚ ... 10
3.1 PRINCIP ČINNOSTI WEBAUTH ... 11
3.2 PROJEKT OPEN ORION ... 13
3.2.1 Pilíře Open Orion ... 14
4 LDAP ... 16
4.1 PRINCIP ... 16
4.2 INFORMAČNÍ MODEL ... 17
4.3 JMENNÝ MODEL ... 18
4.4 FUNKČNÍ MODEL ... 20
4.5 BEZPEČNOSTNÍ MODEL ... 20
4.6 VÝZNAM LDAPU U PRACOVNÍCH STANIC ... 21
5 PLUGGABLE AUTHENTICATION MODULES (PAM) ... 23
5.1 KONFIGURACE PAM MODULŮ ... 24
5.2 INSTALACE PAM V OPERAČNÍM SYSTÉMU SUSE LINUX ... 27
6 KONFIGURACE STANIC PRO UNIVERZITNÍ AUTENTIZACI ... 29
6.1 VÝCHODISKA ... 29
6.2 KONFIGURACE NTP ... 30
6.3 KONFIGURACE KERBEROS ... 31
6.4 KONFIGURACE LDAP ... 35
6.4.1 Problém během konfigurace LDAP klienta v systému SUSE Linux ... 37
6.5 ODSTRANĚNÍ DOČASNÝCH DAT ... 39
6.5.1 Mazací skript ... 39
6.5.2 Konfigurace ... 41
6.6 PŘÍPRAVA UŽIVATELSKÉHO PROFILU PRO VÝUKU V KL206 ... 43
7 ZÁVĚR ... 45
8 SEZNAM OBRÁZKŮ A TABULEK ... 47
9 SEZNAM LITERATURY ... 48
10 RESUMÉ ... 51
11 PŘÍLOHY ... I

ÚVOD

Obsah této diplomové práce je zaměřen na jedno z nejdůležitějších odvětví informačních technologií a tím je bezpečnost dat. Bezpečnost dat může být míněna několika způsoby, tato práce je zaměřena na ochranu dat před cizím zpřístupněním nebo zneužitím.

Bezpečnost dat je jistě rozsáhlé téma v informačních technologiích, které se zcela jistě nikdy nepřestane probírat. Informační technologie, jak hardware, tak software, se rychle vyvíjí, s tím se vyvíjí i bezpečnostní algoritmy, které brání naše informace před dosahem nežádoucích osob. Stejně jako bezpečnostní algoritmy se vyvíjí i techniky útočníků, kvůli kterým nemůže být vývoj bezpečnostních mechanismů nikdy ukončen. Kvůli značné existenci více či méně efektivních mechanismů, různých podpor operačních systémů, rozšíření operačních systémů, začal vznikat v tomto oboru poměrně velký zmatek, který měl za následek snahy o standardizaci některých řešení, které už s konkrétními bezpečnostními algoritmy počítají.

Standardy, zejména ty, které jsou popsány v této práci, se velmi často vyvíjí na prestižních světových univerzitách, odkud se poté šíří do celého světa. Několik standardů je úspěšně implementováno i na Západočeské univerzitě, ne však zcela kompletně.

Cílem této práce je rozšířit standardizaci, využívanou na ZČU, i do operačního systému SUSE Linux Enterprise Desktop, který je nainstalovaný a využívaný v učebně KL206 a na který doposud standardizace rozšířená není. Jedná se konkrétně o chybějící jednotnou univerzitní autentizaci všech uživatelů, kteří jsou na ZČU vedeny po tzv. orion účty. Cílem tedy je umožnit uživatelům ZČU přihlásit se do operačního systému SUSE Linux pod svými univerzitními údaji.

SUSE Linux se v učebně KL206 využívá pro studijní účely. Na stanicích je aktivně vedena výuka a chybějící jednotná autentizace vnáší do výuky mnoho problémů. Kdokoli se chce do systému přihlásit, musí mít v systému zřízený lokální účet. Po ukončení výuky musí být účet zase zrušen. Před další výukou se postup opakuje. Učitel tak musí neustále dokola vytvářet a rušit uživatelská konta, obnovovat zapomenutá hesla studentů a přizpůsobovat uživatelská konta potřebám výuky. Učitel ani nemá přehled, kdo konkrétně je pod lokálním účtem přihlášen nebo kdo mu odevzdal práci. Uživatel je tedy do jisté

míry anonymní, a to jistě výukové účely není výhodné. Všechny zmíněné skutečnosti do velké míry přidělávají učiteli práci, kterou by mít nemusel.

Cílem této práce není naprogramovat software, který by měl SUSE Linux připojit do autentizační sítě ZČU. Díky zmíněné standardizaci v bezpečnostním odvětví a rozsáhlé komunitní základny je možné veškeré potřebné programy získat zpravidla v podobně Open Source a vhodně je nakonfigurovat pro potřeby univerzity. Díky tomu může tato práce posloužit i jako návod pro zprovoznění jednotné univerzitní autentizace na kterémkoli počítači se SUSE Linuxem a jemu podobným distribucím.

1 SYSTÉM JEDNOTNÉ AUTENTIZACE (SSO)

1.1 DŮVODY VZNIKU SSO

V dnešní době je běžné, že každá významnější služba, instituce, společnost, webová stránka poskytuje uživateli nějaký přístup ke svým informacím z internetu. V dnešní době je také nutné zajistit přístup k veškerým personálním informacím pouze uživateli, který je oprávněn tyto informace získat. Takový přístup je většinou realizován pomocí přístupových údajů ve formě uživatelského jména hesla, které uživatel při registraci dostane nebo zvolí.

Každý bezpečností přihlašovací systém jednotlivých serverů[1] vyžaduje různé kombinace znaků a čísel, speciálních symbolů, velkých písmen a na závěr je heslo posíleno minimálním počtem znaků. Hesla by se správně neměla opakovat, neměla by být nějakým snadno odhadnutelným řetězcem jako je „1234“ nebo „heslo“ a měla by se pravidelně obměňovat. Některá přihlášení jsou ještě doplněna o nutnost se prokázat osobním certifikátem. Všechna tato pravidla potom vedou k tomu, že je uživatel vědomě více či méně porušuje a celý bezpečnostní systém ztrácí smysl.

Výše popsané nepohodlné bezpečnostní povinnosti ve výsledku znamenají zdržení práce uživatelů (zaměstnanců), kteří se musí neustále někde přihlašovat. Spolu s rostoucím počtem přihlašovacích údajů také roste míra zapomínání těchto údajů. To zase přidělává práci systémovým administrátorům, kteří jsou tak nuceni neustále resetovat uživatelská hesla. To je náročné jak časově, tak ve výsledku i finančně.

Právě kvůli takovémuto vývoji byla snaha vymyslet nějaké bezpečností řešení, které by všechna tato bezpečnostní pravidla spojovalo a zároveň by nabízelo vysokou úroveň zabezpečení pro všechny druhy služeb. Takovéto řešení se nazývá Systém jednotné autentizace neboli Single Sign On (SSO). (1) (2)

1.2 ČINNOST SSO

V praxi činnost SSO běžný uživatel prakticky nepozná, nebo při nejmenším nevnímá, že se děje něco nestandardního, celý proces probíhá na pozadí. Přihlašovací systém SSO v principu vyžaduje po uživateli pouze jedno jediné přihlášení pro přístup

[1] Server je vzdálený počítač, který poskytuje nějakou službu klientům v síti.

do všech jinak oddělených systémů. Uživatel tak není nucen vlastnit více přihlašovacích údajů, postačí mu pouze jedny, o zbytek se postará SSO.

Výhodou i vlastností SSO je i existence jednotné správy uživatelských účtů. Uživatel má pouze jeden uživatelský účet, ve kterém se nachází veškeré údaje, které jsou nutné pro pohyb mezi systémy spadajícími pod SSO. Během prvotního přihlášení uživatel poskytuje ze svého účtu více či méně informací, které jsou nutné pro autentizaci i do jiného systému.

Mechanismus SSO je tedy navržený tak, že druhotné systémy důvěřují údajům, které jsou získány z prvotního přihlášení. Další vlastností SSO je zajištění zabezpečeného přenosu autentizačních dat mezi jednotlivými systémy proti útočníkům pomocí šifrování. Předání informací z prvotního přihlášení pro druhotná ověření uživatele může probíhat několika způsoby.

- Přímo – informace poskytnuté uživatelem poslouží i pro přístup do ostatních systému v SSO. Během následujících autentizací tak není třeba dalších údajů.
- Nepřímo – informace poskytnuté uživatelem poslouží k přístupu k ostatním údajům, které budou potřeba během další autentizace.
- Okamžitě – během prvního přihlášení je uživatel zároveň přihlášen i do jiného systému.
- Dočasně uloženo – autentizační údaje jsou dočasně uloženy do paměti pro případ navázání spojení s jiným systémem, který je může využít. (3) (1)

V současné době se přesný název normy, která kompletně popisuje SSO, nazývá X/OPEN Single Sign On (XSSO). (4) Kompletní specifikace normy XSSO je možné naleznout v příloze číslo 1.

1.3 NEVÝHODY SSO

SSO stejně jako vše ostatní v běžném životě má své nevýhody. Hlavní nevýhoda vychází přímo ze samotného principu činnosti. Útočníkovi stačí znát jeden jediný přihlašovací údaj, pomocí kterého se může dostat ke všem ostatním údajům bez vědomí uživatele. Při nasazení systému SSO by tedy měli uživatelé ještě přísněji dodržovat ta bezpečností pravidla, kvůli kterým vůbec SSO vzniklo.

Další nevýhoda je spojená s tím, že je SSO poměrně složité v praxi nasadit. Implementace tohoto systému včetně její správy je tak poměrně časově, tím pádem i finančně, náročná a ve výsledku se investice do této technologie nemusí vyplatit. (1)

1.4 SSO NA ZČU

ZČU v Plzni využívá standard SSO prostřednictvím systému WebAuth. Tato technologie byla vyvinuta na Stanfordské univerzitě a jedná se o typické webové SSO nasazení. Webové z toho důvodu, že jsou uživatelé tímto způsobem autentizováni pouze pro webové aplikace, jako je například univerzitní Webmail nebo Courseware. Autentizace tedy probíhá pouze v rámci internetového prohlížeče. WebAuth není určený pro přihlašování do klientských stanic nebo aplikací, není ani jedinou možnou implementací SSO v praxi, existují také další způsoby, jak zajistit SSO pro webové v síti, jako jsou Cosign, Shibboleth nebo CAS. (5)

Pro jednotnou autentizaci v rámci přihlašování uživatelů do stanic využívá ZČU projekt Open Orion. Open Orion je kombinace několika pokročilých síťových technologií, jako jsou Kerberos, AFS[2], Sun Java Identity Manager a Grouper[3].

Pokud je konkrétní stanice zapojena do univerzitní sítě a řádně nakonfigurována, může se do stanice přihlásit libovolný uživatel, který dostal od ZČU přihlašovací údaje. Uživatel je poté autentizován, autorizován a může mít přístup i ke svým souborům. Zároveň je tato stanice řízena a spravována z jednoho, k tomuto účelu pověřeného, místa.

V současné době má ZČU zajištěnou podporu pouze pro operační systémy Windows XP, 7 a Linux Debian. Pro SUSE Linux Enterprise Desktop, který je využíván pro výukové účely v KL206 podpora není.

SUSE Linux se využívá z několika důvodů. Jedním z důvodů je oficiální podpora Novellu, jakožto výrobce systému. Dalším důvodem je například podpora přihlašování do systému Directory Services resp. Novell eDirectory, které se využívá jako výukové prostředí na Katedře výpočetní a didaktické techniky.

[2] AFS – Andrew File System je technologie pro vzdálený přístup ke k souborům uživatele. (24)
[3] Sun Java Identity Manager a Grouper – Technologie pro centrální správu uživatelů.

2 KERBEROS

Kerberos je pokročilý autentizační protokol vyvinutý v Massachusettském technologickém institutu. Vývoj byl finančně podporován velkými společnostmi, jako jsou Microsoft, SUN Microsystems, Google, Apple a další. Tento protokol byl navržen jako univerzální autentizační mechanismus, který lze implementovat do libovolné platformy a který zajišťuje maximální možnou míru zabezpečení dat proti cizímu narušení. Zároveň je Kerberos určen pro plné nasazení do systémů využívajících SSO, kde se využije jako autentizační prvek. (5) (6) (7)

V současné době je nejnovější Kerberos ve verzi 5, která je zpětně kompatibilní s verzí 4 a je výchozím autentizačním protokolem při zapojení počítačů do domény s Active Directory[4] v systémech Windows 2000, XP a novějších. Je také využíván i zcela jinými zařízeními než jsou počítače, například Xbox[5]. (5)

Kerberos je centralizovaná služba, která spoléhá na důvěryhodnou autoritu třetí strany, jež má na starosti ověřování uživatelů. Autentizace tedy neprobíhá přímo na serveru, který zajišťuje nějakou službu, ale na počítačích, které jsou k tomu určeny. Tím je zajištěno větší bezpečnost v tom smyslu, že autentizace nebude probíhat na všech možných počítačích v síti ale pouze na jednom jediném, kterému se dá věřit. To na sebe ovšem nese několik rizik.

Hlavním problémem tedy je to, aby tyto důvěryhodné servery za každou cenu důvěryhodné zůstaly. Pokud by byly nějakým způsobem napadeny, přestávalo by mít takovéto řešení smysl. Dalším rizikem je, že případný výpadek na tomto serveru by znamenal nefunkčnost autentizace v celé síti. Pro tento případ je nutné zajistit několika stupňovou redundanci serverů. (8)

2.1 PRINCIP ČINNOSTI

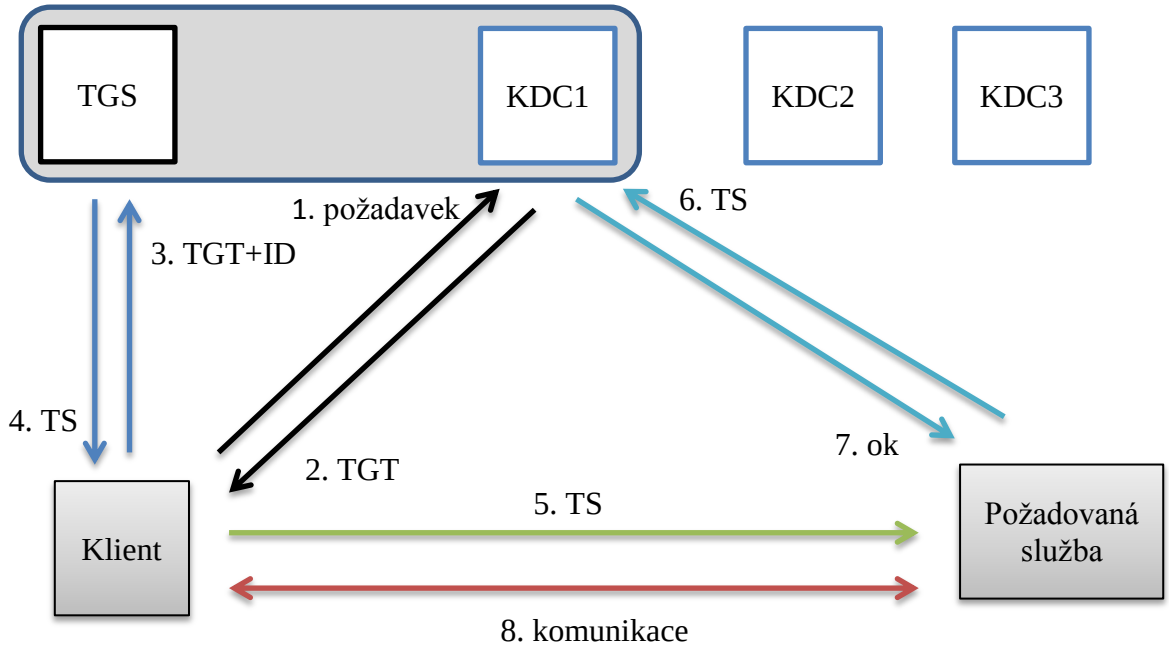
Z bezpečnostního hlediska je Kerberos navržen tak, že se během přenosu nepřenáší žádné citlivé údaje, jako jsou především uživatelská hesla. Celý proces autentizace je realizován prostřednictví lístků neboli tiketů, které mají omezenou

[4] Active Directory je systém adresářových služeb vyvinutý společností Microsoft v síťovém prostředí pro operační systémy Microsoft Windows.
[5] Xbox je herní konzole vyvinutá společností Microsoft.

platnost. Klientský počítač si o tiket musí zažádat, kdykoli chce poprvé navázat spojení se vzdáleným serverem. Pokud lístku skončí jeho platnost, musí si klient zažádat znovu. Tento lístek poté umožní klientskému počítači „projít“ skrze kontrolu na připojovaném serveru. Stejným lístkem se poté může klientský počítač autentizovat do libovolné síťové služby nebo aplikace, která je zapojena do systému zabezpečeným Kerberosem. Tím je vyřešen i systém jednotné autentizace SSO, který splňuje parametry normy XSSO.

Základním prvkem v celém procesu autentizace je služba Key Distribution Center neboli KDC. Tato služba je spuštěna v serverech, které se starají o autentizaci uživatele. V databázi této služby jsou bezpečně uloženy veškeré přihlašovací informace o uživatelích v doméně. Z důvodu zajištění chodu systému v případě výpadku jsou KDC servery, na kterých je tato služba spuštěna, v redundantním počtu. Dalším důležitým prvkem je služba TGS (Ticket granting service). Tato služba je spuštěna po boku KDC a má za úkol poskytnout lístek, který už slouží pro prokázání identity uživatele na serveru požadované služby. (8)

Konkrétní postup při autentizaci uživatele, který se snaží přihlásit do požadované služby je popsán na Obrázek 1.



Obrázek 1 Schéma činnosti Kerberos v5 Zdroj: Vlastní tvorba

1. Klient pošle požadavek na ověření nejbližšímu KDC serveru. V požadavku je pouze jméno uživatele, nikoliv heslo.
2. KDC server vytvoří lístek TGT (Ticket-Granting Ticket), který na základě uživatelova hesla, které je uloženo v databázi, zašifruje a pošle ho zpět uživateli. Tento lístek má omezenou platnost, po skončení platnosti je nutné o lístek požádat znovu.
3. Pokud klient dokáže TGT lístek dešifrovat zadaným heslem (zadaným do přihlašovacího formuláře) je jasné, že se jedná o oprávněného uživatele. Tento lístek se následně opět zašifrovaně pomocí hesla pošle službě TGS spolu s identifikačními údaji původní požadované služby. TGT lístek umožní klientovi zažádat o přístup k libovolné službě v síti Kerberos, změní se pouze ID této služby.
4. Služba TGS uživatelovo heslo zná, je součástí KDC, a tak lístek dešifruje, ověří identifikační údaje cílové služby, platnost lístku a vystaví lístek TS (Ticket service), který poslouží k prokázání identity vůči požadované službě. Lístek je opět časově omezený (na základě TGT) a šifrovaným způsobem poslán klientovi.
5. Klient má nyní k dispozici lístek TS, který pošle požadované službě. Tímto lístkem se může klient prokazovat požadované službě.
6. Požadovaná služba prozatím neví, zda je lístek platný. Lístek pošle znovu KDC serveru, který ho dokáže rozšifrovat (má v databázi uložená hesla), dokáže ověřit, zda je ještě platný a dokáže určit, zda ID požadované služby odpovídá.
7. Pokud je lístek platný, KDC pošle požadované službě potvrzení o platnosti.
8. Požadovaná služba důvěřuje KDC a autentizuje klienta, následně může probíhat komunikace. (8) (9) (10)

Pro zprovoznění autentizace v rámci přihlašování do domény pomocí Kerberos, je nutné mít v klientských počítačích nainstalovaný klient Kerberos. V současné době tyto klienty existují jak pro operační systémy Windows, tak pro Linuxové distribuce.

2.2 KERBEROS NA ZČU

Technologie Kerberos je aktivně využívána na ZČU. V rámci infrastruktury má ZČU k dispozici 4 KDC servery, 1 hlavní (Master) a 3 jeho repliky pro případ výpadku nebo jiných problémů. Podrobnější informace o jednotlivých serverech jsou zobrazeny na Obrázek 2.

Server	Typ	Poskytující služby	Porty
kerberos-adm.zcu.cz	Master	kadmin kdc krep	TCP 749 TCP/UDP 88
kerberos1.zcu.cz	Replika	kdc krb524 kx509	TCP/UDP 88 UDP 4444 UDP 9878
kerberos2.zcu.cz	Replika	kdc krb524 kx509	TCP/UDP 88 UDP 4444 UDP 9878
kerberos3.zcu.cz	Replika	kdc krb524 kx509	TCP/UDP 88 UDP 4444 UDP 9878

Obrázek 2 Kerberos KDC servery využívané na ZČU. Zdroj: http://support.zcu.cz/index.php/Kerberos

ZČU přímo neovlivňuje fungování autentizace, má ale možnost určovat parametry vystaveného lístku, tedy například jak dlouho bude lístek platit, může také určovat intervaly povinné změny hesla uživatelů.

V současné době je možné jedno heslo používat maximálně 12 měsíců. (11) Lístek, vystavený KDC, má v současné době platnost 10 hodin, to dokazuje následující obrázek, na kterém je zobrazena platnost zrovna vydaného lístku uživateli *kulda*. Důležitá jsou data *Valid starting* a *Expires*.

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: kulda@ZCU.CZ

Valid starting     Expires            Service principal
10/31/13 21:17:05  11/01/13 07:17:01  krbtgt/ZCU.CZ@ZCU.CZ
```

Obrázek 3 Platnost uživatelova TGT lístku. Zdroj: Vlastní tvorba.

3 SSO NA ZČU PODROBNĚ

SSO v podobě WebAuth je sytém, který k autentizaci využívá Kerberos. Protože se jedná o webové řešení, není nutné mít v počítači nainstalovaný žádný klient Kerberos. Kerberos tikety se v tomto případě také přenáší, ale pouze mezi servery WebAuth a Kerberos. V síti mezi klientem a WebAuth chráněnými aplikacím se přenášená data nazývají tokeny a zašifrovaně se ukládají do cookies prohlížeče nebo do URL adresy prohlížeče. Aby bylo možné se přihlašovat pomocí WebAuth, je tedy nutné mít nainstalovaný jakýkoliv internetový prohlížeč, který podporuje šifrovaný přenos SSL/TLS a cookies. Více nutných požadavků pro správnou funkci není. (12)

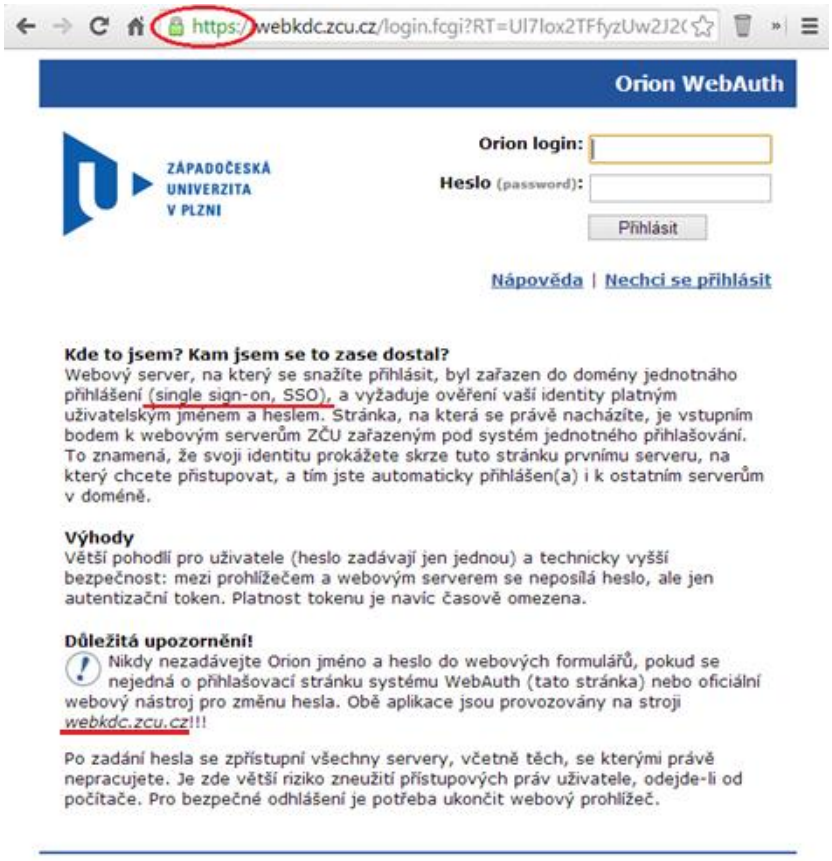
Každý aplikační server[6], který chce být chráněný systémem WebAuth, musí mít spuštěný Apache[7] server, na kterém WebAuth běží. Respektive běží na něm pouze několik modulů z celého systému WebAuth a to modul autentizační (mod_webauth) a autorizační (mod_webauthldap – není nutný, pokud webová aplikace nemusí pro svou činnost rozlišovat přístupová práva). Zatímco první jmenovaný nepřímo zajišťuje, jak je patrné, autentizaci, druhá část zajišťuje autorizaci ke svým službám. Přístupová práva uživatelů k jednotlivým službám jsou získávána pomocí LDAP[8]. Zásadní je poslední modul (mod_webkdc), který je spuštěn na samostatném serveru WebKDC. WebKDC je díky spolupráci se servery Kerberos odpovědný za autentizaci uživatele. (13) (14)

Základní princip autentizace WebAuth je podobný principu popsaném v předchozí kapitole s tím rozdílem, že v tomto případě se o vše stará prohlížeč díky cookies a SSL/TLS. Tento systém vyžaduje pro přihlášení do kterékoli internetové stránky zabezpečené WebAuth zadat uživatelské tzv. Orion jméno a heslo. Tyto údaje se pošlou na autentizační server, který má databází přístupových údajů. Pokud jsou správně, vygeneruje se pro uživatele token, který se uloží do cookies. Pokud je token uložený v cookies stále platný, je možné ho použít i do jiné aplikace a uživatel tak není nucený znovu psát své přihlašovací

[6] Aplikační server je počítač, na kterém je spuštěna nějaká webová aplikace, která je zpřístupněna vzdáleným uživatelům, například Portál ZČU.
[7] Apache server je velmi rozšířená open source aplikace pro vytvoření webového serveru na lokálním počítači.
[8] LDAP je protokol, pomocí kterého se získávají informace o uživatelích, jejich práva a adresářová struktura použitá v místní doméně. Podrobněji bude LDAP popsán v dalších kapitolách.

údaje a je automaticky přihlášen. (13) Podrobnější popis mechanismu činnosti WebAuth je popsán v následující kapitole.

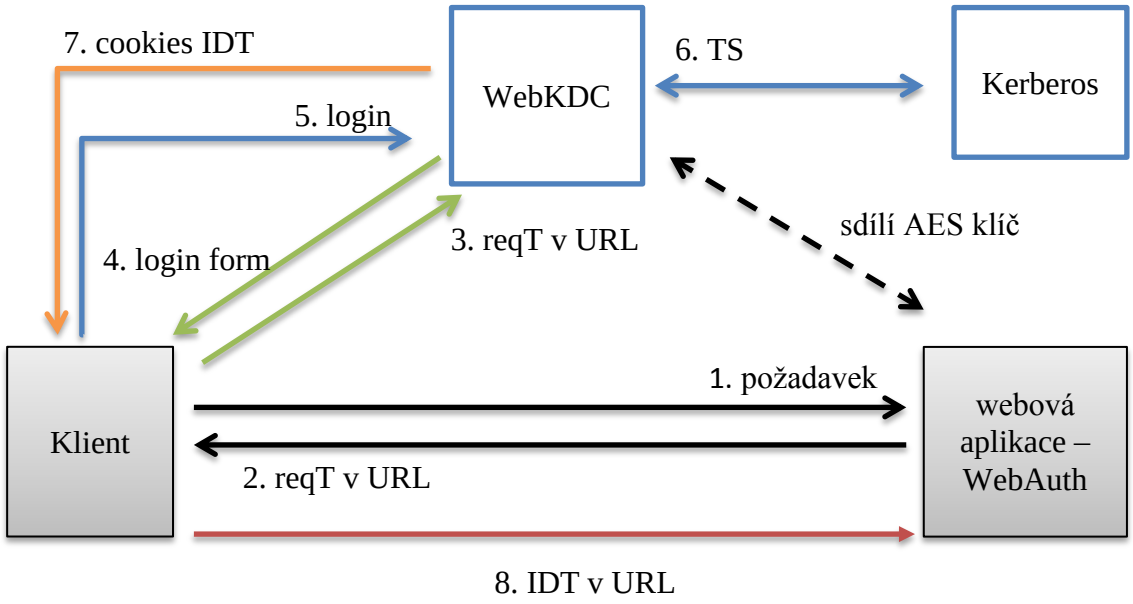
Následující Obrázek 4 upozorňuje uživatele, že je potřeba poslat své přihlašovací údaje na server WebKDC. Údaje budou poslány zašifrovanou cestou a mohou posloužit k vícenásobnému přihlašování, protože je uživatel v systému SSO.



Obrázek 4: Šifrované přihlášení SSO za ZČU pomocí WebAuth a Kerberos.

3.1 PRINCIP ČINNOSTI WEBAUTH

Činnost WebAuth probíhá v několika krocích, které vystihuje Obrázek 5. Celý proces probíhá v pozadí a uživatel si prakticky nemá možnost všimnout, co vše se na pozadí přihlašovacího procesu děje.



Obrázek 5 Princip činnosti WebAuth v síti s autentizací Kerberos. Zdroj: Vlastní tvorba.

1. Klient, respektive jeho internetový prohlížeč, se chce přihlásit do nějaké webové aplikace chráněné systémem WebAuth, například do Portálu.
2. Aplikace Portál (modul mod_webauth) si zkontroluje URL adresu, která klienta přivedla a zjistí, že se v této adrese nenachází žádný uživatelský identifikátor neboli ID Token a tudíž se jedná o doposud neověřeného uživatele. Do URL adresy uživatele tedy vloží požadavek (request ID Token) na získání ID Token. Požadavek na ID Token je zašifrován pomocí AES, dešifrovací klíč mezi s sebou sdílí server WebKDC (mod_wevkdc) i aplikace Portál (stejně jako ostatní aplikace pod WebAuth).
3. Uživatelův prohlížeč je přesměrován na autentizační server WebKDC, kterému přeposílá žádost o ID Token umístěnou v URL adrese.
4. Server WebKDC žádost v URL adrese dešifruje a ověří, zda je ještě platná a zda webová aplikace má vůbec právo vyžadovat ID Token. Pokud ano, uživateli zašle přihlašovacím formulářem jako je na Obrázek 4, do kterého skryje požadavek na ID Token.

5. Uživatel vyplní přihlašovací údaje, které se šifrovaným způsobem, odešlou do WebKDC, skrytý požadavek na ID Token zajišťuje, že vyplněný formulář přišel od správného odesilatele (vrátí se).
6. WebKDC jako klient kontaktuje Kerberos KDC server (viz předchozí kapitola). Výsledkem je lístek.
7. Pomocí lístku WebKDC vygeneruje ID Token a uloží jej do cookies prohlížeče. ID Token má také svou časovou platnost.
8. Prohlížeč uživatele je přesměrován na původně požadovanou adresu Portálu. Během přesměrování se do URL prohlížeče vloží ID Token, podle kterého může aplikace Portál autentizovat uživatele. Portál z ID Tokenu v cookies vytvoří aplikační Token, který bude sloužit pro další přístup na Portál (dokud bude mít časovou platnost).

Pokud se bude chtít uživatel přihlásit do jiné webové aplikace WebAuth neznamená to, že automaticky může. V URL adrese se opět nenachází ID Token, ze kterého by se dal vytvořit aplikační Token. Celý postup se proto musí opakovat, nicméně tentokrát jednodušeji.

Při pokusu o přihlášení do jiné webové aplikace je uživatelův prohlížeč opět přesměrován do WebKDC pro ID Token. Uživatelův prohlížeč znovu odešle žádost o ID Token, zároveň ale pošle ještě stávající ID Token, který je stále uložen v cookies prohlížeče. WebKDC poté akorát ověří, zda je ID Token ještě platný. Pokud ano, vytvoří nový ID Token určený pro novou webovou aplikaci, který opět uloží jako cookie. Do URL adresy uživatelova prohlížeče je opět umístěn tento ID Token, zároveň je prohlížeč přesměrován do původně požadované aplikace, která na základě ID Tokenu v URL adrese vytvoří cookies pro další použití. (13) (14) (15)

3.2 PROJEKT OPEN ORION

Jak již bylo zmíněno, projekt Open Orion slouží k jednotné univerzitní autentizaci uživatelů do pracovních stanic. Uživateli mohou být jak studenti, tak zaměstnanci ZČU. Cílem projektu Open Orion není pouze jednotná autentizace, ale také jednotná správa

uživatelů, centrální správa softwaru, možnost přístupu uživatelů ke svým souborům nebo do společných univerzitních aplikací.

Vzhledem i k postupnému rozšiřování ZČU je velmi vhodné, aby bylo v rámci univerzity nasazeno jednotné výpočetní prostředí. Pokud by každá fakulta využívala vlastní výpočetní prostředí, přinášelo by to mnoho komplikací, zejména finančních. Každá fakulta by tak musela mít tým svých správců, kteří by se starali o fungování počítačů, správu uživatelských účtů, technickou pomoc a další. Zároveň by uživatelé měli přístup ke svým datům pouze z budovy fakulty.

ZČU má do své sítě zapojeno několik tisíc počítačů, tyto počítače můžou být díky jednotnému výpočetnímu prostředí vzdáleně spravovány, může na ně být vzdáleně instalován systém i další software. (16) Jednotné výpočetní prostředí s centrální správou, kterou má v současné době na starosti CIV[9], se tak jeví jako výhodné.

Každé globálně použité služby mají své nevýhody, jednotné výpočetní prostředí není výjimkou. Centrální správa může leckdy na problémy reagovat velmi pomalu. Zpravidla je i centrální správa jedinou autoritou, která má povoleno do stanic cokoliv instalovat a sama se může rozhodnout, zda navrhovaný software do stanic nainstaluje či nikoliv. Pokud tedy zaměstnanec (učitel) potřebuje do stanic nainstalovat příslušný software, musí o instalaci zažádat a doufat, že mu bude vyhověno. Správci také mívají pracoviště poměrně vzdálené a k akutnímu problému nemohou dorazit včas. Avšak přes tyto problémy převyšují výhody, kvůli kterým se vyplatí jednotné prostředí využívat.

ZČU nasadila Projekt Orion, nebo také pouze Orion, do celouniverzitního provozu v roce 1996. V té době byla podpora zajištěna na systémech Windows 95/98, později byla podpora rozšířena na operační systémy Windows NT, Windows 2000 a Linux Debian. V současné době starší verze Windows nejsou podporovány, nejstarší podporovaný OS Windows je verze XP. Nicméně technologie a mechanismy, které byly zavedeny v roce 1996, zpravidla zůstaly stejné nebo prošly aktualizacemi na novější verze.

3.2.1 PILÍŘE OPEN ORION

Projekt Orion je tvořen několika stěžejními pilíři, které dohromady tvoří komplexní jednotné univerzální výpočetní prostředí. Pokud má být systém zařazen do

[9] CIV - Centrum informatizace a výpočetní techniky je hlavním střediskem výpočetního prostředí ZČU.

plnohodnotného Orionu, musí mít tyto pilíře implementovány. Zde může nastat problém, ne každý operační systém má možnost tyto technologie vůbec implementovat.

Těmito pilíři jsou autentizační mechanismus, přístup do centrálního souborového systému, databáze informací o uživatelských účtech, Systém pro správu výpočetního prostředí. Jednotné prostředí poté zahrnuje i další méně stěžejní služby, jako jsou poštovní služby, tiskové, webové služby.

1. Autentizace – Jako autentizační mechanismus je používán Kerberos verze 5, který je celosvětově uznávaný bezpečnostní standard. Pro to, aby mohla být stanice „kerberizována“, musí být v systému funkční Kerberos klient. Kerberos i v tomto případě zajišťuje SSO funkcionalitu, nejen autentizaci.
2. Přístup do centrálního souborového systému – Pro tyto účely je nasazena technologie AFS, konkrétně open source varianta OpenAFS. Každý uživatel má zřízen domovský adresář, ke kterému má přístup z kterékoli stanice.
3. Databáze informací o uživatelských účtech je získávána ze vzdáleného LDAP serveru. Mezi tyto informace patří uživatelovi přístupové údaje, práva, kontaktní údaje.
4. Systém pro správu výpočetního prostředí – Kolem roku 2005 se na ZČU začal využívat produkt Sun Java Identity Manager. Jedná se o pokročilou technologii pro řízení uživatelů, jejich zařazování do skupin a určování přístupů k programům. Pomocí této technologie lze například uživatelům hromadně přiřazovat nebo odebírat přístupy k nainstalovaným aplikacím. (17)
5. Další služby – Mezi další služby je možné zařadit poštovní služby, konkrétně vlastní IMAP, POP3 a SMTP server pro práci s emaily. Mohou to být také sdílené tiskové služby, hostingové služby nebo webové aplikace jako jsou například portal.zcu.cz nebo courseware.zcu.cz a mnoho dalších. (18)

4 LDAP

4.1 PRINCIP

Lightweight Directory Access Protocol neboli LDAP je protokol, který slouží pro přístup do adresářových služeb v doméně. Často je LDAP označován jako tenký klient, který se připojuje k serveru bez jakýchkoliv dalších technologií. LDAP je tedy technologie, která funguje na principu klient – server.

Adresářové služby nejsou v tomto případě služby, které by měly nějakou spojitost se soubory nebo složkami. Jsou to služby, které mají na starost práci s tzv. adresáři. Adresář je rozsáhlá a hierarchicky strukturovaná databáze, která obsahuje velmi mnoho informací o objektech v síti resp. v doméně.

Objekt v síti je tedy jakýkoliv prvek v síti, o kterém mohou být vedeny nějaké informace. Tyto objekty mohou být například uživatelé, uživatelské skupiny, tiskárny, pracoviště atd. Mezi informacemi, popisující tyto objekty mohou být uživatelská jména, hesla, telefonní čísla, adresy, emaily, čísla popisné atd. Je to vlastně jakási kartotéka všech možných informací. Zároveň je možné jednotlivým objektům nastavovat přístupová práva k dalším objektům nebo ke konkrétním složkám. Objekt, který je vytažen z pomyslné kartotéky, má tak přesně popsáno, kdo vlastně je a co může dělat. To je hlavní účel LDAPu.

Z výše vypsaného vyplývá, že LDAP bude nasazený nejčastěji v rozsáhlých sítích, ve kterých je z mnoha důvodů výhodné mít tato data někde logicky uspořádána a třeba rozčleněna podle přístupových práv. Takovou síť provozuje také ZČU.

Každá taková síť má tak speciální LDAP server, ve kterém jsou umístěny zmíněné databáze, respektive adresáře. Tento LDAP server má za úkol poskytovat administrátorům nebo příslušným aplikacím údaje, které jsou potřebné k běhu počítačů v síti. Protože LDAP server bývá vzdáleně přístupný počítač, je možné se k němu odkudkoli připojit a dostat tak potřebné informace například k tomu, aby mohl být uživatel přihlášen a co může na počítači dělat.

LDAP vychází ze série standardů X. 500, které určují adresářové služby. Má za úkol definovat, upravovat a přistupovat k údajům na vzdáleném adresářovém serveru. LDAP,

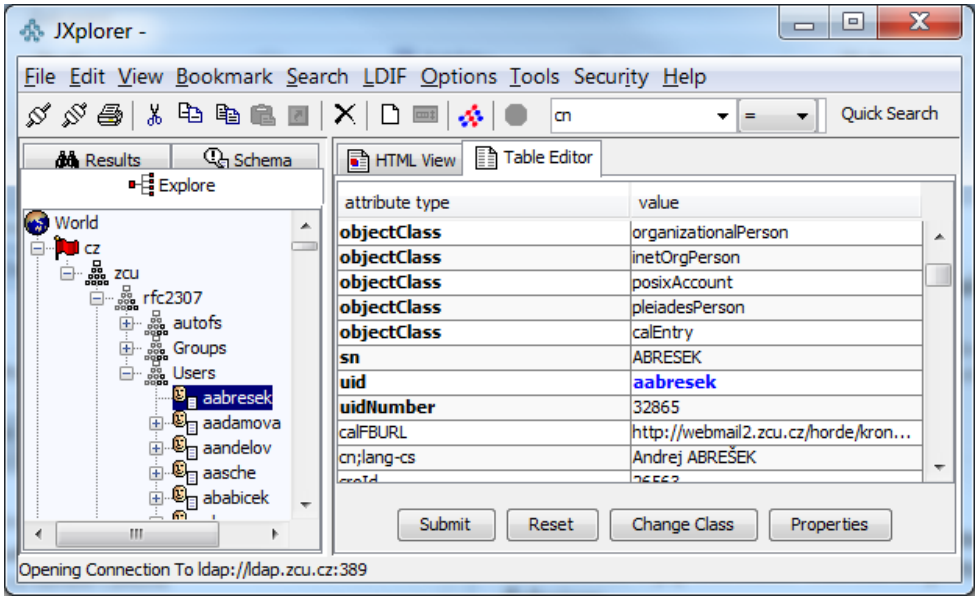
na rozdíl od mnoha jiných protokolů, není určený pro přístup k souborům a složkám uživatele, přestože název (slovo Directory) tomu může napovídat. Je to univerzální protokol, který slouží pro přístup k mnoha různým druhům adresářových služeb. Jako příklad nejčastěji jmenovaných adresářových služeb, které mají implementovaný protokol LDAP je OpenLDAP, Novell eDirectory a Windows Active Directory. ZČU využívá v centrální doméně OpenLDAP. (19) (20) (21)

Když v 80. letech vznikl ze standardu X.500 protokol DAP (Directory Access Protocol), byl založen na síťovém modelu ISO/OSI. Nicméně model ISO/OSI byl prakticky „převálcován“ modelem TCP/IP a tak byl DAP nahrazen zjednodušeným protokolem LDAP. Aktuální LDAP je ve verzi 3 (připravuje se verze 4) a je popsána v dokumentaci RFC 3377. V rámci specifikace protokolu byly definovány 4 modely, které LDAP tvoří (zároveň tedy i LDAP adresáře) a určují jeho činnost. Těmito modely jsou Informační model, Jmenný model, Funkční model a Bezpečnostní model. (22) (20)

4.2 INFORMAČNÍ MODEL

Informační model se věnuje konkrétním informacím o nějakém konkrétním objektu v síti. Takovýmto objektům se říká **záznamy** a může to být například uživatel, který má v pomyslné kartotéce vyplněno několik údajů. Informační model tedy definuje, jaké informace mohou být v „kartě“ uživatele (nejen uživatele) vyplněny.

Tomu, co popisuje informace, se říká **atribut,** každý uživatel může mít mnoho atributů. Atributy mohou být například *mail, givenName, telephoneNumber, street, sn*. Každý tento atribut má poté svoji hodnotu, kterou je už použitelná informace. Hodnotou atributu *mail* bude zcela jistě emailová adresa uživatele, hodnota atributu *street* bude zcela jistě ulice uživatele. Každému takovému atributu může být definovaná syntaxe, datový typ a další upřesnění, které musí hodnota dodržovat. (23) Na následujícím obrázku je zachycen program JXplorer, který slouží jako klient k připojení na vzdálený LDAP server. Na obrázku je vidět LDAP uživatel *aabresek* spolu s jeho atributy a hodnoty.



Obrázek 6 Atributy a hodnoty LDAP uživatele. Zdroj: Vlastní tvorba.

Jak bylo již zmíněno, jedná se o hierarchicky tvořené adresáře. Každý objekt/záznam má tedy svou úroveň. Celá LDAP struktura se nazývá strom (DIT – Directory Information Tree). Počátku struktury, nebo také kořenu stromu, se říká **root** (někdy rootDSE). Jednotlivé záznamy jsou, podobně jako větve skutečného stromu, rozvětvovány až na samé konce. Záznamy, které se už dál nevětví, jsou na konci a nazývají se **listy** (leaf). Na předchozím obrázku je vidět i struktura LDAP stromu za ZČU, která se rozvětvila až na konečného uživatele *aabresek*.

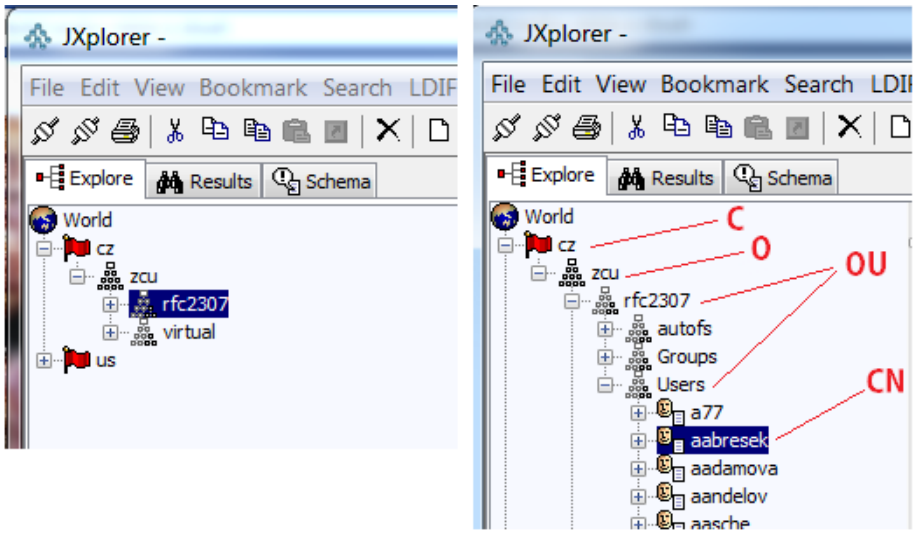
Listem bývá právě takový uživatel, který má na základě své „větve“ určité atributy. Atributy se tedy dědí. Pokud má nějaké atributy definovaný záznam na vyšší úrovni, automaticky se tyto záznamy dědí i na všechny úrovňově nižší záznamy. Tomuto procesu se říká, že potomek dědí atributy rodiče. Balíku použitých atributů, které slouží jako vzor pro záznamy ve struktuře, se říká **schéma**. Každá záznam má tedy nějaké atributy, které během vytvoření získal na základě dědičnosti a schématu.

4.3 JMENNÝ MODEL

Jmenný model definuje DN (Distinguished Name), což v překladu znamená rozlišovací jméno. DN je pro práci s LDAP adresářem zásadní. Každý záznam v LDAP databázi má své DN, díky kterému je jednoznačně identifikovatelný. Pomocí DN se také jednoznačně definují veškeré záznamy, které poté tvoří kompletní strukturu v LDAPu. Každou část této struktury můžeme poté nastavovat.

Díky DN je možné organizovat uživatele, přidávat je do skupin, přiřazovat jim nebo celým skupinám práva atd. Stejně tak je díky DN možné vyhledat konkrétního uživatele. Podle jeho umístění v různých organizačních jednotkách se dají zjistit také uživatelova práva.

V DN se rozlišuje několik druhů jmen. Těmito jmény jsou CN (Common name – běžné jméno), OU (Organization unit – organizační jednotka), O (Organization – organizace), C (Country – stát). Pomocí těchto jmen lze jednoznačně identifikovat jakýkoli záznam ve stromu. Jakým způsobem se provádí záznam a jednoznačná identifikace, je ukázáno na následujícím obrázku. Pro ukázku opět posloužit program JXplorer.



Obrázek 7 Větvení LDAP struktury na ZČU. Zdroj: Vlastní tvorba.

Na předchozím obrázku je vidět, že nejblíže kořenu stromu se nachází záznam *cz* typu *Country*. Jak je z obrázku i názvu patrné, tento záznam slouží pro rozlišení záznamů podle země. Na nižší úrovni ihned po *cz* se nachází záznam *zcu* typu *Organization*. Organizací by mohlo být samozřejmě i více, všechny by byly typu *Organization*. Záznamy typu *Organization* se dále mohou větvit na *Organization unit*, které se mohou dělit buď na koncové listy (*leaf*) nebo na další *organization unit*. Toto řazení má danou strukturu, v záznamu typu *organization unit* tedy nemůže být záznam *organization* nebo *country*. Na druhou stranu listy (*leaf*) zde být mohou.

V případě vyobrazeném na předchozím obrázku se organizace *zcu* dělí na 2 organizační jednotky *rfc2307* a na *virtual*. Organizační jednotka *rfc2307* se dále větví na 3 organizační jednotky (*autof*, *Groups*, *Users*). Organizační jednotka *Users* se už dále

nevětví, namísto toho má v sobě koncové listy, které jsou v tomto případě jednotlivý uživatelé univerzity.

Pokud bychom chtěli vyjádřit DN zmíněného uživatele *aabresek*, byla by to posloupnost záznamů, které se postupně větví až k uživateli. Tato posloupnost by byla: *sn=aabresek, ou=Users, ou=rfc2307, o=zcu, c=cz*. Jednotlivé objekty, které vedou k samotnému listu, jsou oddělné čárkou (mezera následovat nemusí) a mají stanovenou syntaxi.

4.4 FUNKČNÍ MODEL

Ve funkčním module je zahrnuto několik operací, které je možné v rámci práce v LDAP databázi provádět. Tyto operace jsou sepsány do následující tabulky. V rámci zprovoznění jednotné univerzitní autentizace nebude nutné tyto operace provádět.

Tabulka 1 Operace definované ve funkčním modelu LDAP. Zdroj: http://www.samuraj-cz.com/clanek/adresarove-sluzby-a-ldap/

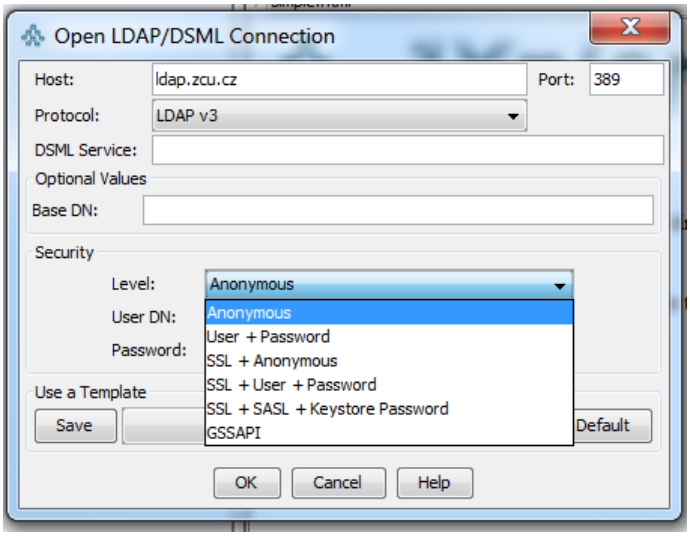
bind	inicializuje spojení, vyjednává o metodě autentizace, autentizuje
unbind	ukončí session
abandon	klient žádá o ukončení posílání výsledků na poslední dotaz
search	výběr dat z určitého regionu pomocí filtru
compare	porovná hodnotu atributu se zadanou hodnotou
add	vytvoří nový objekt
modify	upraví atributy záznamu (vytvořit, smazat, upravit)
modify RDN	slouží k přesunutí objektu v rámci stromu adresáře
delete	smazání záznamu

4.5 BEZPEČNOSTNÍ MODEL

Bezpečnostní model definuje, kdo má povoleno se k LDAP serveru připojit. V rámci standardu je možné využít anonymní přístup, který uživateli neumožňuje jakékoli změny atributů nebo hodnot. Takový přístup je využit i pro realizaci jednotné autentizace v této práce. Systém si „anonymně“ zjistí uživatelovy atributy, které využije při přihlášení.

Dále je možné se přihlásit pomocí klientských certifikátů nebo jména a hesla. Jméno uživatele je nutné uvádět v podobě uživatelova DN. Ověřování hesla může být zabezpečeno protokolem SSL/TLS. Zároveň je možné využít způsob SASL.

SASL je množina programových rozhraní, díky kterým je možné k zabezpečení využít libovolný algoritmus, který je podporován klientem i serverem. (22) Dále je možné využít i další bezpečnostní mechanismy, nicméně ty už přímo v popisu LDAP protokolu nejsou. Na následujícím obrázku jsou vidět možnosti zabezpečeného přihlášení opět v programu JXplorer.

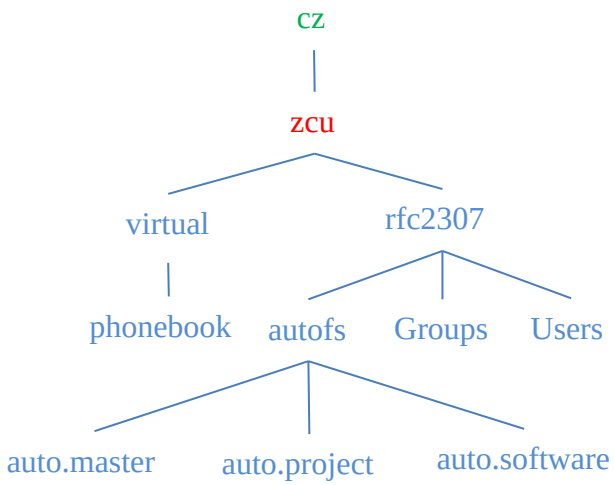


Obrázek 8 Možnosti zabezpečeného přihlášení do LDAP server v programu JXplorer. Zdroj: Vlastní tvorba.

4.6 Význam LDAPu u pracovních stanic

Kompletní schéma LDAP struktury používané na ZČU je vidět na následujícím obrázku. Zeleně zbarvené *cz* představuje stát (country), červeně zbarvené *zcu* představuje organizaci (O), zbývající modré položky jsou organizační jednotky (OU), některé jsou i vnořené. Organizační jednotky, které se už dále nevětví, obsahují už pouze jednotlivé listy (leaf), které ve schématu již nejsou.

Pro účely jednotné autentizace je nejdůležitější organizační jednotka Users, ve které je seznam všech uživatelů na ZČU a do které je nutné se během přihlašování připojit. Díky struktuře je možné zjistit DN jednotky Users a zadat jej do LDAP klienta, který bude mít na starost přihlašování vzdáleného uživatele.



Obrázek 9 Schéma LDAP struktury na ZČU. Zdroj: Vlastní tvorba.

5 PLUGGABLE AUTHENTICATION MODULES (PAM)

PAM moduly, neboli v překladu zásuvné autentizační moduly, jsou velmi často využívané komponenty ve všech novějších Linuxových distribucí. Zajišťují mnoho bezpečnostních úkonů, které vývojářům aplikací výrazně ulehčují práci. Ulehčují práci nejen vývojářům, ale také systémovým administrátorům. První PAM moduly byly vyvinuty roku 1995 programátory firmy Sun Microsystems. Následně bylo v rámci normy XSSO standardizováno i vhodné API[10] pro PAM moduly.

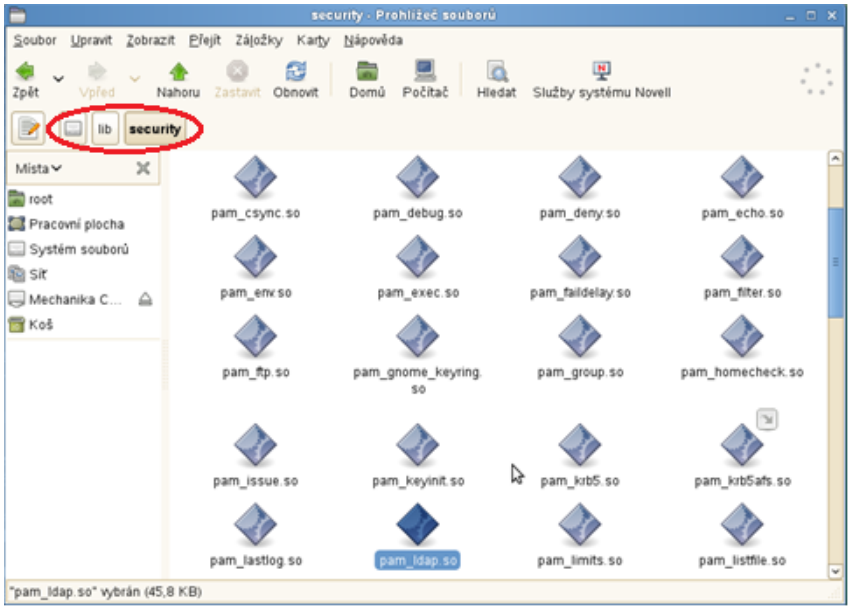
PAM modul je v podstatě knihovna, která se zavolá vždy v případě, kdy je nutné vykonat nějakou bezpečnostní proceduru. V dobách, kdy se PAM moduly nepoužívali, museli vývojáři aplikací implementovat do svých programů také určité bezpečnostní algoritmy, které zajišťovaly, že s programem, nakonec i s daty, pracovala pouze pověřená osoba například v určitou zvolenou dobu. Samozřejmě ne každý vývojář je schopný tímto způsobem kvalitně zabezpečovat své programy a tak je možné, že mnohdy byly programy zabezpečeny pouze jednoduchými algoritmy nebo třeba nebyly zabezpečeny vůbec. Z tohoto důvodu vznikly PAM.

Smysl je takový, že každý PAM modul je samostatný prvek, který se stará o jeden druh zabezpečení. Každý PAM modul tedy plní pouze jednu činnost, za to s maximální kvalitou. Pokud vývojář bude chtít svůj program zabezpečit, jednoduše k němu přiřadí některé PAM moduly podle toho, jakým způsobem chce zabezpečení provést. Zároveň má vývojář, de facto i administrátor, možnost zvolit, jak se má PAM modul chovat například při selhání ověření uživatele. Vývojáři a administrátoři mohou tedy poměrně jednoduchým způsobem nastavovat chování PAM modulu, respektive celého procesu autentizace.

PAM moduly se nepřiřazují pouze při vývoji aplikace. Pokud je již program nainstalovaný, lze k němu moduly přiřazovat během konfigurace aplikace (v praxi realizováno nejčastěji). Pokud mluvíme o „konfiguraci PAM modulů“, velmi často je tím myšleno přiřazení programu konkrétní moduly a nastavení jejich chování.

[10] API – Application Programming Interface, připravené programovatelné rozhraní, které výrazně ulehčuje programování pro danou oblast.

Samotný PAM modul lze také upravovat. V systému Linux jsou pro to vhodné nástroje, například *pam-config*, pomocí kterého lze modul upravovat v textovém režimu. Jednotlivé moduly se dají upravovat také ručně pomocí konfiguračních souborů, které jsou umístěny v SUSE Linuxu v */etc/security*. Úpravou PAM modulu ovšem způsobíme globální změnu, která se projeví u všech programů, které tento modul využívají. (24) (25) Jednotlivé PAM moduly se nacházejí v adresáři */lib/security*.

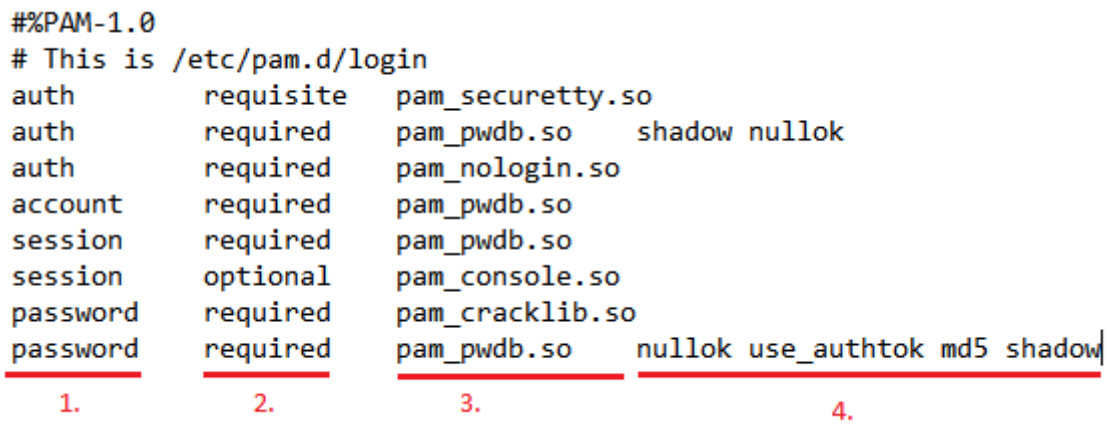


Obrázek 10 Umístění PAM modulů v SUSE Linuxu. Zdroj: Vlastní tvorba.

5.1 KONFIGURACE PAM MODULŮ

Pokud chceme nějakému programu přiřadit PAM modul, musí existovat konfigurační soubor tohoto programu. Konfigurační soubor musí mít stejný název, jako sám program a musí se nacházet v adresáři */etc/pam.d/.*

Umístění */etc/pam.d/* platí například pro distribuce SUSE Linux Desktop Enterprise nebo OpenSUSE, nicméně je možné, že jiné distribuce mají definované umístění jinde. Obsah konfiguračního souboru programu „login“ je vidět na Obrázek 11.



Obrázek 11 Obsah konfiguračního souboru. Zdroj: Vlastní tvorba

Struktura každého konfiguračního souboru je stejná. Vždy je tvořena čtyřmi sloupci, první tři sloupce (zleva) jsou povinné, čtvrtý je nepovinný. V prvním sloupci je definováno, jaký typ ověření se bude konat. V druhém sloupci je určena důležitost daného postupu, ve třetím je definováno, jaký konkrétní PAM modul bude použit, poslední sloupec je připraven pro případné parametry fungování. Přestože je struktura dána sloupci, konfigurace se čte postupně po řádcích.

Pro definování typu ověřování, který se bude provádět, slouží tedy první sloupec. Hodnoty v tomto sloupci mohou nabývat čtyř hodnot. Těmito hodnotami jsou: *auth, account, session, password*. Každý typ ověřování se zaměřuje pouze na jeden konkrétní druh ověřování. Jednotlivé typy ověřování nejsou povinné a nemají určené své pořadí, nicméně podle svých činností (viz níže) je jasné, že pořadí nemůže být zcela náhodné. Tak jako v ostatních sloupcích ani zde se nepovoluje jakýkoli překlep nebo záměna malého písmena na velké.

- auth – Typ ověřování auth znamená, že se bude ověřovat, zda je uživatel opravdu tím, za koho se vydává. Ověření zde většinou probíhá pomocí jména a hesla. Nemusí to být ovšem pouze heslo, může se jednat třeba o otisk prstu, oční test, nicméně to už má na starosti příslušný PAM modul.
- account – Typem account se ověřuje, zda se již autentizovaný uživatel může připojit, nicméně tentokrát to není ve smyslu autentizace uživatele. Ověřovat se může například to, zda se uživatel může přihlašovat v danou dobu, zda má stále

ještě platné heslo nebo zda není překročen maximální počet připojených uživatelů.

- session – Typ session znamená, že se ověřuje, zda má již autentizovaný uživatel vše připravené pro připojení. Zejména se jedná o to, zda má přiřazené dostatečné systémové prostředky (může i definovat vlastní nutné systémové prostředky), zda má připojené všechny potřebné adresáře, jednotky nebo zda má uživatel povoleny či zakázány některé konkrétní služby.
- password – Typ password určuje, že jsou použité PAM moduly, které mají v sobě algoritmy pro změnu hesla a dalších bezpečnostních údajů uživatele. Požadavek na změnu hesla může vyvolat uživatel nebo může být vyvoláno automaticky z důvodu blízkého konce platnosti.

Hodnoty v druhém sloupci značí důležitost ověření. Tyto hodnoty určují činnost programu v případě úspěchu nebo neúspěchů daného ověřovacího procesu. Hodnot může být celkem pět a to: *requisite, required, sufficient, optional, binding*.

- requisite – Requisite v překladu znamená nutný. Pokud ověřování v rámci tohoto modulu skončí neúspěchem, běh programu se okamžitě přeruší a následně ohlásí chybu ověřování (případně další doplňující informace). Přístup je poté zamítnut. Pokud ověřování skončí úspěšně, pokračuje se dalším modulem (řádkou).
- required – Required v překladu znamená potřebný. Pokud toto ověřování skončí selháním, znamená to zamítnutí celkového přístupu, nicméně před tím se postupně dokončí všechna ostatní ověřování. Pokud ověřování skončí úspěšně, pokračuje se dalším.
- sufficient – Sufficient v překladu znamená dostatečný. Pokud toto ověření skončí úspěšně, automaticky uživatel dostane přístup i v případě, že některá předchozí ověřování skončila nezdarem. Pokud skončí neúspěšně, pokračuje se dalším ověřováním.

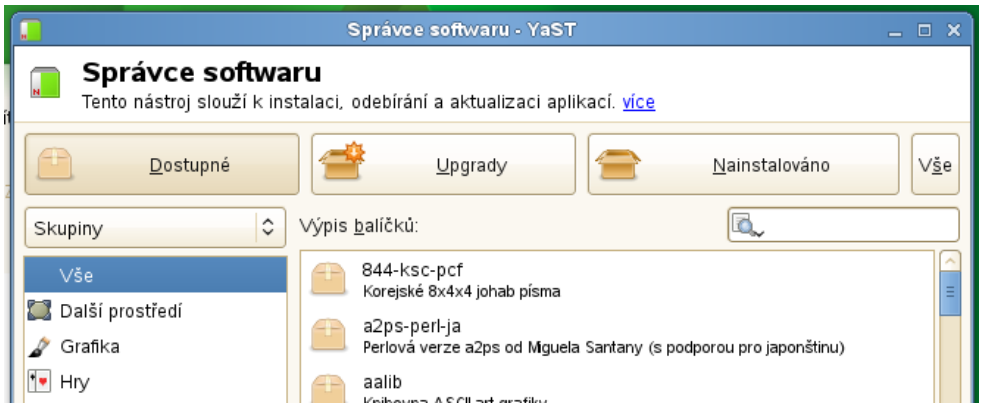
- optional – Optional je v překladu volitelný. Toto ověřování oznámí úspěch nebo neúspěch, nicméně dále je ignorováno a nijak neovlivňuje povolení nebo zamítnutí celkového přístupu.
- binding – Binding v překladu znamená vázané. Pokud ověřování skončí úspěšně a zároveň žádný předchozí modul neskončil neúspěchem, je přístup ihned povolen. Pokud ověřování skončí nezdarem, je přístup zamítnut, nicméně další ověřování se provedou. (26) (25) (24) (27)

Třetí sloupec je určen k přiřazení konkrétního PAM modulu. Pokud to PAM modul umožňuje, může být jeho činnost upravena použitím vhodného argumentu v posledním sloupci. Parametry, které je možné použít, lze najít v technické dokumentaci konkrétního modulu.

Protože nikde není dáno konkrétní pořadí při zpracování ověřování, může programátor nebo administrátor vhodnou kombinací a seřazením konfiguračních řádek přesně řídit celý proces ověřování, aniž by musel znát jednotlivé zabezpečovací mechanismy.

5.2 INSTALACE PAM V OPERAČNÍM SYSTÉMU SUSE LINUX

PAM moduly jsou zpravidla šířeny jako open source. Stáhnout se dají na internetových stránkách konkrétní organizace zodpovědné za PAM modul nebo díky repozitáři[11] pomocí správců balíčků v jednotlivých Linuxových distribucích. V SUSE Linux je to program Správce softwaru.



Obrázek 12 Výřez okna programu Správce softwaru. Zdroj: Vlastní tvorba.

[11] Repozitář je vzdálený server, který je určený k distribuci aplikací a různých balíčků pomocí internetu. K repozitáři se tak může systém připojit a stáhnout vždy aktuální verzi softwaru.

Zároveň mohou některé aplikace vyžádat nějaký konkrétní PAM modul a ihned ho automatiky stáhnout a nainstalovat, opět za pomocí repozitáře. V SUSE Linuxu se tak děje velmi často a díky tomu není nutné zjišťovat, jaké konkrétní PAM moduly potřebujeme. Okno výzvy pro nainstalování vhodného PAM modulu je zachyceno například v kapitole: 6.3 Konfigurace Kerberos.

Tyto konfigurační aplikace, zejména z balíku YAST2, který je popsaný v následující kapitole, také provedou základní konfiguraci PAM modulů k dané aplikaci. Pokud administrátorovi ovšem nevyhovují možnosti nastavení v aplikaci, musí si konfiguraci PAM modů provést ručně, dle principů popsaných v předchozí kapitole.

6 KONFIGURACE STANIC PRO UNIVERZITNÍ AUTENTIZACI

6.1 VÝCHODISKA

Konfigurace probíhala v operačním systému SUSE Linux Enterprise Desktop 11, který se využívá pro výukové účely v učebně KL206. Cílem je rozšířit jednotnou univerzitní autentizaci i na tyto stanice s tímto operačním systémem.

Univerzitou je zajišťována podpora jednotné autentizace pouze operačním systémům Windows a Debian Linux. Z toho důvodu byly na stanicích v KL206 doposud využívány pouze lokální uživatelské účty, které se musely ručně vytvořit pro každého uživatele (studenta, učitele). Po skončení výukového bloku musely být účty zase odstraňovány, případně pročišťovány nebo měněny. Dalším faktem bylo i to, že uživatelé byli v síti anonymní a v případě nějakého problému nemuselo být snadné uživatele vypátrat.

Všechny tyto postupy jsou časově náročné a nepraktické. Z praktického hlediska by nebylo možné takovýmto způsobem organizovat výuku. Učitel by velmi těžko stíhal zakládat lokální účty desítkám studentů a poté je zase odebírat. Výsledkem by tak mohl být naprosto neuspořádaný operační systém, který by neustále „bobtnal“ o soubory uživatelských účtů, které už dávno nejsou potřeba. Jednotná autentizace, kterou je cílem zprovoznit v operačním systému SUSE Linux, tak přinese výrazné usnadnění řízení výuky a umožní udržet uspořádaný operační systém nutný k dlouhodobé činnosti.

Cílem tedy je, aby ve stanicích v učebně KL206 nemuseli být žádní lokální uživatelé, krom samozřejmě uživatele root[12], aby byly Orion přihlašovací údaje funkční právě i pro přihlašování do operačního systému SUSE a aby byl po odhlášení uživatele počítač „čistý“. K tomu bylo nutné zajistit několik věcí.

Cílem není získat po přihlášení uživatele přístup ke globálnímu domovskému adresáři uživatele ani k ostatním sdíleným adresářům v doméně ZČU. Přestože by to bylo možné, díky klientu OpenAFS[13], není to žádoucí opět z praktických důvodů. Tyto služby

[12] root je uživatel v operačních systémech Linux, který má plná práva ke všem částem systému.
[13] OpenAFS je klientský software, který implementuje standard AFS do stanice, díky tomu je možné přistupovat ke vzdáleným adresářům a souborům, které jsou v doméně k dispozici.

jsou v učebně KL206 využívány z výukových účelů pomocí softwaru Novell Client[14]. Kombinace OpenAFS a Novellu by nemusela pracovat správně a stabilně a běžná výuka by tak mohla být ohrožena.

Konfigurace jednotlivých protokolů probíhá v operačních systémech zpravidla pomocí konfiguračních souborů. Výjimkou není ani SUSE Linux, nicméně v SUSE Linuxu je k dispozici několik nástrojů, které konfiguraci usnadňují. Balík těchto konfiguračních nástrojů se jmenuje YAST neboli *Yet Another System Tools*, volně přeloženo jako Pouze další systémové nástroje. V současné době je YAST ve verzi 2.

Protokoly, které je nutné konfigurovat, jsou NTP, Kerberos, LDAP. Celkový proces, který je nutné nakonfigurovat, bude ve finále fungovat následujícím způsobem:

1. Uživatel v přihlašovacím formuláři do operačního systému SUSE Linux vloží své Orion údaje.
2. Operační systém kontaktuje LDAP server, podle kterého zjistí, zda takový uživatel na ZČU existuje a zda má správné heslo.
3. Operační systém kontaktuje Kerberos KDC server a na základě uživatelských údajů získá tiket, který ho autentizuje. Po autentizaci je uživatel přihlášen a je evidován jako přihlášený ZČU uživatel.
4. Systém připraví příslušný adresář, do kterého připraví všechny nutné soubory a složky pro nového uživatele (lokální domovský adresář a další).
5. Proces přihlášení je hotov, uživatel může pracovat.

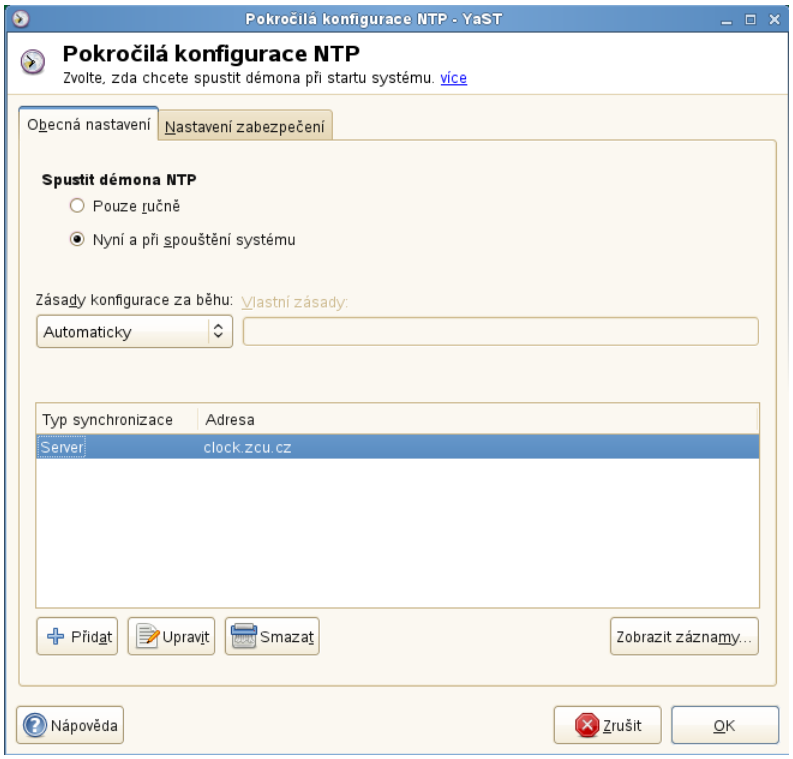
6.2 KONFIGURACE NTP

Network Time Protocol je protokol, který se stará o synchronizaci času získávaného z nějakého serveru. Jak bylo několikrát zmíněno v přechozích kapitolách, v univerzitní síti se využívají tikety nebo tokeny, které mají omezenou časovou platnost. Aby se tedy dalo přesně určit, zda je lístek nebo token ještě platný, je nutné mít ve všech počítačích zapojených do domény stejný čas. Čas se synchronizuje pomocí NTP serveru,

[14] Novell Client je software vyvinutý společností Novell, který se používá pro adresářové, autentizační a autorizační služby v síťovém prostředí Novell.

který údaje o času poskytuje. ZČU má vlastní (veřejný) NTP server, jehož adresa je clock.zcu.cz. Funkční NTP protokol je nutný pro chod protokolu Kerberos.

Pro konfiguraci NTP je použit software Konfigurace NTP z balíku YAST. Do tohoto programu je nutné pouze přidat nový NTP server a nechat ho spouštět při spuštění systému. Jedna z činností, kterou tento program provede, je to, že nastaví konfigurační soubor *ntp.conf* umístěný v */etc/.* Obrázek 13 zachycuje příslušené nastavení, které bylo provedeno.

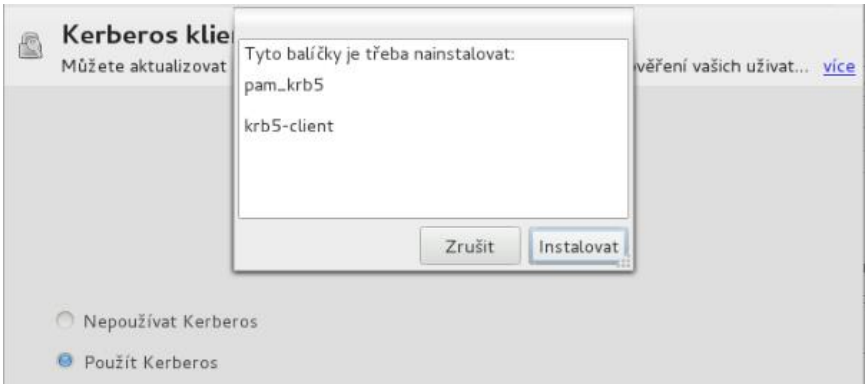


Obrázek 13 Nastavení protokolu NTP pomocí nástroje v balíku YAST. Zdroj: Vlastní tvorba.

6.3 KONFIGURACE KERBEROS

Aby bylo možné konfigurovat Kerberos, je nutné mít plně funkční synchronizaci času pomocí NTP (viz předchozí kapitola). Stejně jako v předchozí kapitole můžeme v SUSE Linuxu využít YAST a v něm připravený Kerberos klient.

Pro zprovoznění klienta Kerberos je nutné do systému doinstalovat příslušný PAM modul a další potřebný balíček. O této skutečnosti YAST ještě před dokončením konfigurace upozorní a nabídne tyto moduly automaticky stáhnout a nainstalovat. Tato situace je vyobrazena na Obrázek 14.

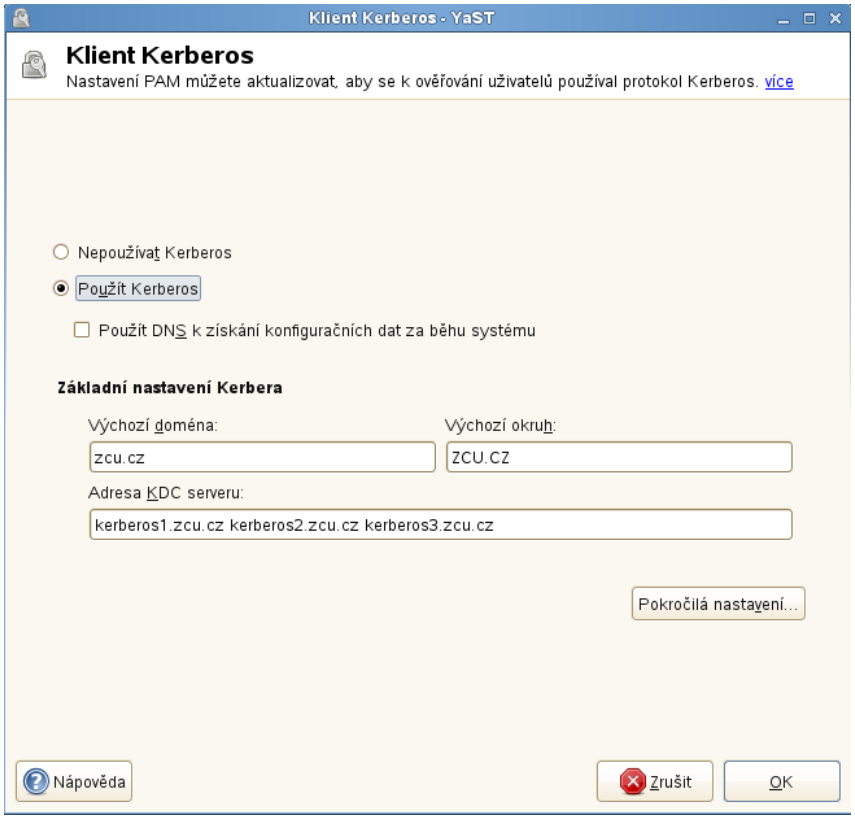


Obrázek 14 Výzva pro nainstalování PAM modulů. Zdroj: Vlastní tvorba.

Potřebný PAM modul se nazývá *pam_krb5*, dále je ještě nutné doinstalovat balíček *krb5-client*, který je opravdovým klientem sítě Kerberos. Vizuální klient Kerberos je tedy spíše jakýmsi grafickým prostředníkem mezi uživatelem a skutečným Kerberos klientem.

Dále je nutné do klienta zadat informace o Kerberos infrastruktuře na ZČU. Mezi tyto informace patří výchozí doména, ta je pro ZČU jednoduše *zcu.cz*. Výchozí okruh je *ZCU.CZ* a adresy jednotlivých KDC serverů jsou *kerberos1.zcu.cz, kerberos2.zcu.cz, kerberos3.zcu.cz*. (28)

Více nastavování není třeba. Kompletní nastavení je vidět na Obrázek 15. Po potvrzení konfigurace Kerberos klient službu spustí a především nastaví zásadní konfigurační soubor *krb5.conf* umístěný v adresáři */etc/.*



Obrázek 15 Nastavení Kerberos klienta pomocí balíku nástrojů YAST. Zdroj: Vlastní tvorba.

Tímto ovšem ještě není konfigurace dokončena, protože podstatný soubor *krb5.conf* ještě není zcela upraven pro správnou komunikaci se ZČU. Naštěstí není nutné soubor ručně poupravovat, protože ZČU tento soubor veřejně poskytuje na svých stránkách. Stačí tedy soubor stáhnout a přepsat jím původní soubor *krb5.conf*. (29) Z tohoto důvodu nebylo třeba více nastavovat grafický klient Kerberos, protože další nutné parametry jsou již nastavené v souboru stáhnutém ze stránek ZČU.

Do tohoto souboru lze nahlédnout, nachází se v něm kompletní konfigurace Kerberos na ZČU. Lze z něho vyčíst například adresy serverů, konfigurace a umístění dalších důležitých souborů. Obsah souboru *krb5.conf* je vypsán na několika následujících řádcích.

```
[libdefaults]
        default_realm = ZCU.CZ
        clockskew = 300
#       default_realm = EXAMPLE.COM

[realms]
ZCU.CZ = {
        kdc = kerberos1.zcu.cz
```

```
        kdc = kerberos2.zcu.cz
        kdc = kerberos3.zcu.cz
        default_domain = zcu.cz
        kpasswd_server = kerberos-adm.zcu.cz:464
        admin_server = kerberos-adm.zcu.cz
}
W3K-ZCU.CZ = {
        kdc = adrasteia.w3k.zcu.cz
        kdc = hermes.zcu.cz
}

[logging]
        kdc = FILE:/var/log/krb5/krb5kdc.log
        admin_server = FILE:/var/log/krb5/kadmind.log
        default = SYSLOG:NOTICE:DAEMON
[domain_realm]
        .zcu.cz = ZCU.CZ
[appdefaults]
pam = {
        ticket_lifetime = 1d
        renew_lifetime = 1d
        forwardable = true
        proxiable = false
        minimum_uid = 1
        clockskew = 300
        external = sshd
        use_shmem = sshd
}
```

Po tomto nastavení je již možné získat tiket od pověřeného KDC serveru. Příkaz na získání tohoto lístku je *kinit kulda*, slovo *kulda* je zde název uživatele. Příkazem *klist* je možné vypsat všechny Kerberos lístky, získané od KDC serveru. Možných příkazů pro práci s klientem Kerberos je daleko více. Pokud přijetí lístku funguje, je konfigurace Kerberosu hotova.

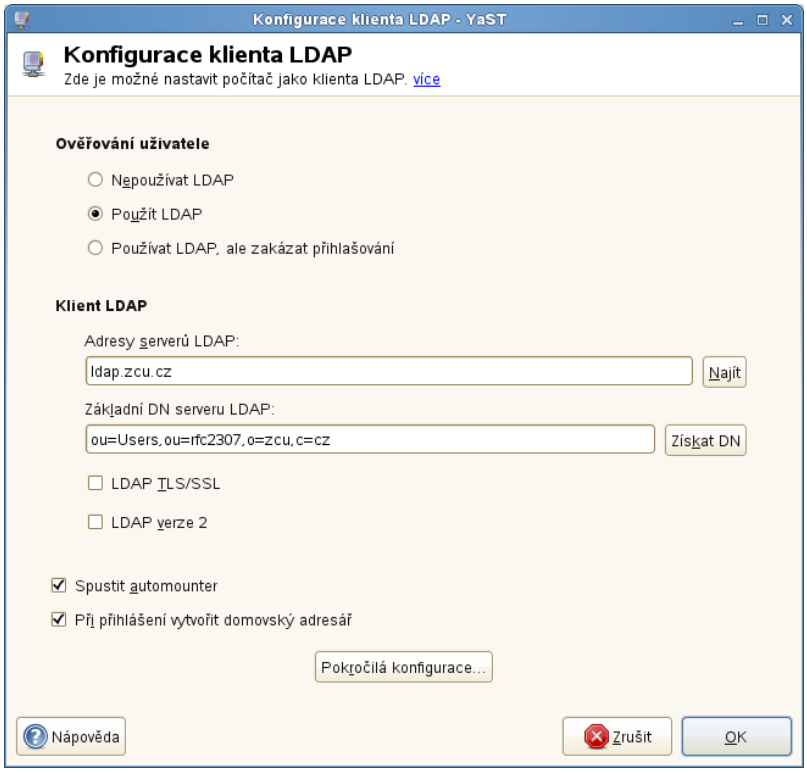
```
linux:~ # kinit kulda
Password for kulda@ZCU.CZ:
linux:~ # klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: kulda@ZCU.CZ

Valid starting     Expires            Service principal
10/31/13 21:17:05  11/01/13 07:17:01  krbtgt/ZCU.CZ@ZCU.CZ
        renew until 11/15/13 21:17:01
linux:~ #
```

Obrázek 16 Uživatel získal Kerberos ticket po příkazu kinit. Zdroj: Vlastní tvorba

6.4 KONFIGURACE LDAP

Aby bylo možné získat údaje o všech uživatelích v síti ZČU, je nutné mít funkční LDAP klient, který se připojí k LDAP serveru. Z něho získá seznam studentů, jejich práva, začlenění do organizačních jednotek atd.

Také ke zprovoznění LDAPu je možné využít balík YAST a v něm připravený LDAP klient. Správně vyplnění klient je vyobrazeno na Obrázek 17.



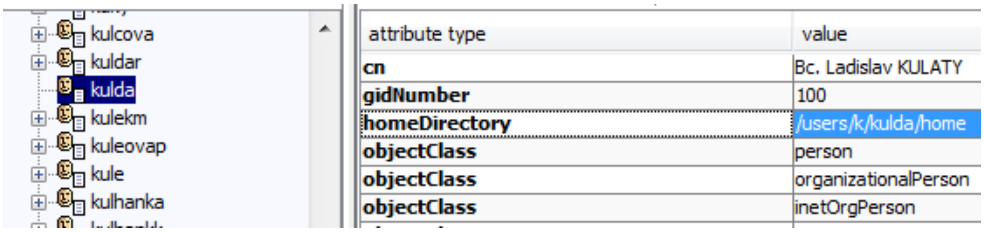
Obrázek 17 Konfigurace LDAP klienta. Zdroj: Vlastní tvorba

Tento klient umožňuje pracovat s protokolem LDAP v několika režimech. Pro nás je nutné, aby se LDAP využíval jich při přihlašování, proto volíme možnost „Použít LDAP“. ZČU má vlastní LDAP server, který má adresu *ldap.zcu.cz* (30).

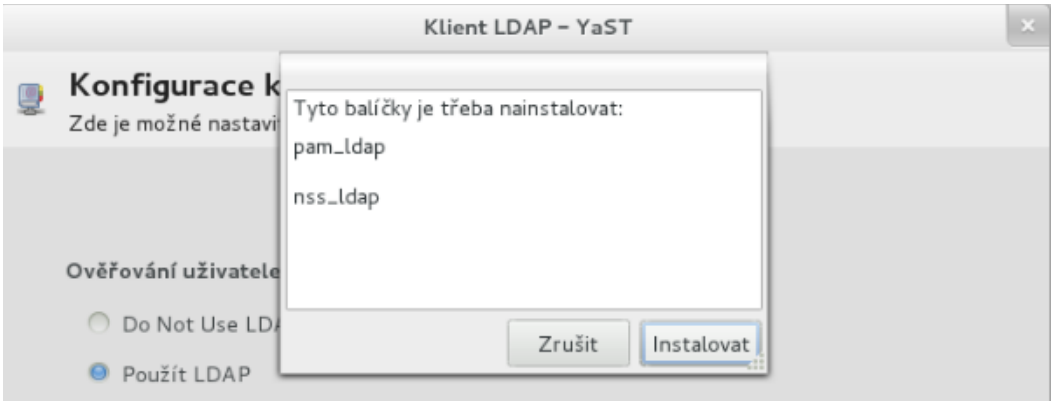
Dalším krokem je zadat základní DN serveru LDAP, kterým se určí, kde se na severu nachází seznam vzdálených uživatelů a jejich informace. Tyto DN přímo závisí na konfiguraci místního LDAP serveru. Pokud konfiguraci DN místního LDAP serveru neznáme, pomocí tlačítka „Získat DN“ můžeme LDAP server kontaktovat a potřebné údaje získat. V kapitole 4. LDAP je ukázáno, že seznam uživatelů je umístěn v organizační jednotce *Users*, která má DN: *ou=Users, ou=rfc2307, o=zcu, c=cz*.

Nakonec ještě zatrhneme možnost „Spustit automounter“. Díky této volbě operační systém uživateli automaticky připojí adresáře, které bude potřebovat pro svou činnost. Dále také zatrhneme možnost „Při přihlášení vytvořit domovský adresář“. Název vypovídá o tom, k čemu je tato možnost dobrá. Je-li výše zatrhnut automounter, je tento domovský adresář automaticky připojen.

Informace o tom, kde se má vytvořit domovský adresář je také jedním z LDAP atributů uživatele. Tento atribut se jmenuje *homeDirectory* a pro uživatele *kulda* má hodnotu */users/k/kulda/home*. Tato informace je zachycena na Obrázek 18. Konečné vytvořené složky přihlášeného LDAP uživatele jsou vyobrazeny na Obrázek 20. Tyto vytvořené složky přesně kopírují cestu převzatou z LDAPu.



Obrázek 18 Cesta k domovskému adresáři v LDAP databázi zjištěna programem JXplorer. Zdroj: Vlastní tvorba.

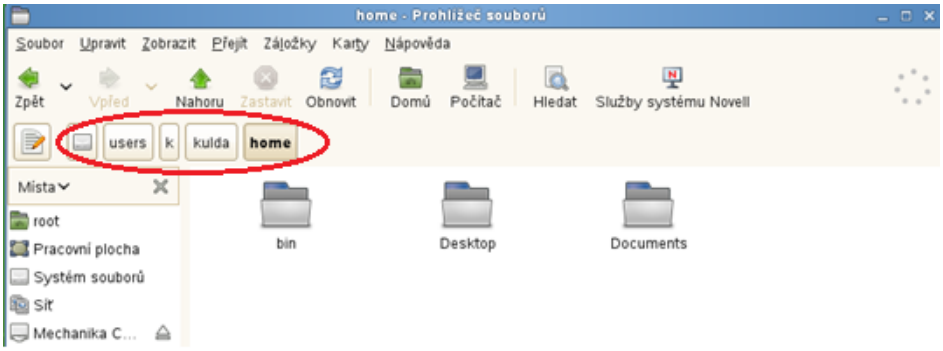


Obrázek 19 Balíčky potřebné pro konfiguraci LDAP klienta. Zdroj: Vlastní tvorba.

Po stisknutí tlačítko „OK“ klient LDAP opět detekuje chybějící potřebné balíčky. První z těchto balíčků je PAM modul *pam_ldap*, druhým balíčkem je *nss_ldap*, tato situace je vyobrazena na Obrázek 19.

Po nainstalování balíčků bude opět automaticky nakonfigurováno několik souborů. První soubor je */etc/ldap.conf* a druhý */etc/nsswitch.conf.* Tyto soubory by již nemělo být třeba dál upravovat.

Před pokusem o přihlášení uživatele je nutné restartovat počítač. Po úspěšném přihlášení vzdáleného uživatele se díky zatrhnuté volbě „Při přihlášení vytvořit domácí adresář“ a „Spustit automounter“ vytvoří nutné uživatelovy soubory. Cesta k těmto souborům je převzata od umístění uživatele na serveru LDAP, viz Obrázek 18. Složka s domácím adresářem a dalšími soubory/složky se tedy pro uživatele *kulda* vytvoří v pracovní stanici ve složce */users/k/kulda/*, pro uživatele *elgrova* se vytvoří v umístění */users/e/elgrova/*. Cesta je tedy převzata z celé univerzitní hierarchie.



Obrázek 20 Automaticky vytvořená adresářová struktura přihlášeného vzdáleného uživatele kulda převzatá z LDAP databáze. Zdroj: Vlastní tvorba

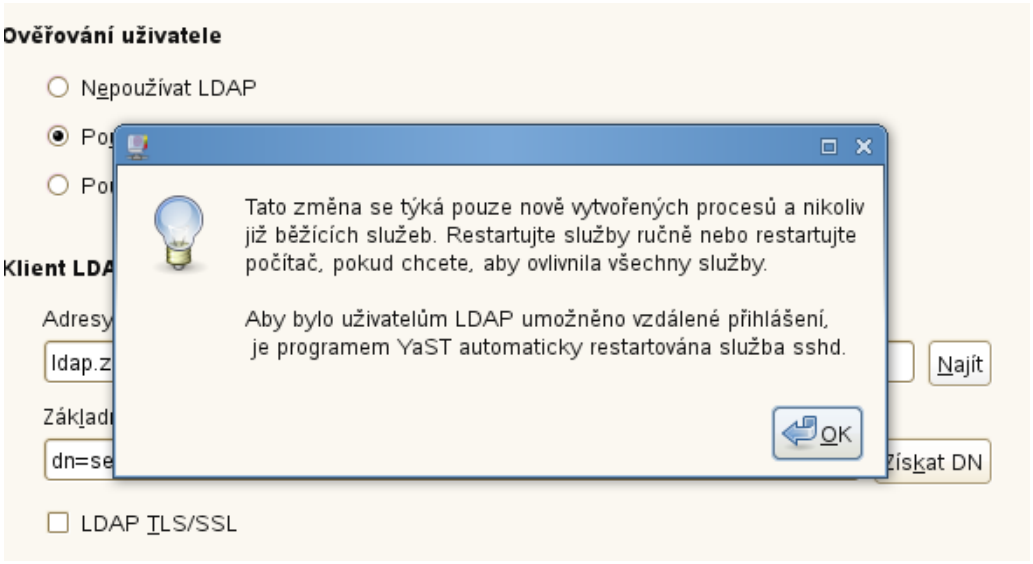
6.4.1 PROBLÉM BĚHEM KONFIGURACE LDAP KLIENTA V SYSTÉMU SUSE LINUX

Přestože výše popsaný postup by měl být funkční, během vlastního provádění konfigurace LDAP klienta se vyskytl problém. Na operačním systému OpenSUSE, na kterém byl postup také testován, problém nebyl. Nicméně na všech počítačových stanicích v učebně KL206 s operačním systémem SUSE Linux Enterprise Desktop nastal stejný problém při pokusu o dokončení konfigurace LDAP klienta.

Po stisknutí dokončovacího tlačítka „OK“, které by mělo vyvolat výše zmíněné činnosti, LDAP klient odmítal službu spustit. Na místo spuštění služby neustále zobrazoval chybovou hlášku o tom, že požadovaná změna bude funkční pouze pro nové služby, nikoli pro stávající. Tato zpráva je vyobrazena na Obrázek 21. Po doporučeném restartování počítače ovšem služba LDAP stále spuštěna nebyla. Klient LDAP byl stále v režimu „Nepoužívat“. Po dalším pokusu o spuštění, respektive dokončení konfigurace LDAP klienta, se zobrazovalo pořád stejné upozornění.

V rozsáhlé dokumentaci systému SUSE Linux, ve které je podrobně popsáno nastavení mnoha různých klientů a protokolů, není žádná zmínka o takové zprávě. (31)

Není ani jasné, zda se tento problém týká všech verzí SUSE Linuxů nebo pouze verze 11.2, instalované na místních stanicích, která je již poněkud starší. Vzhledem k tomu, že tento systém je v učebně KL206 aktivně využíván pro výukové účely, není možné systém nějakým způsobem aktualizovat, protože by to ohrozilo funkčnost jiných programů a tím i výuky.



Obrázek 21 Chybová hláška informující o nefunkčnosti LDAP nastavení. Zdroj: Vlastní tvorba.

Následovalo velmi mnoho dlouhých hodin zjišťování příčiny, pročítání technických dokumentací, psaní dotazů na česká i zahraniční linuxová fóra, vyhledávání na internetu konfiguračních souborů. Postupné ukončování spuštěných a používaných procesů nepřinášelo změnu. Jeden z mnoha pokusů o řešení se nakonec jevil jako funkční.

Problém byl zjištěn v souboru *nsswitch.conf*, který z neznámého důvodu nedokázal klient LDAP správně nakonfigurovat. Na Obrázek 22 je zachycen část tohoto souboru. Modře zvýrazněná část s povely *passwd_compat: ldap* a *group_compat: ldap* původně v souboru chyběla. Po přidání těchto dvou řádků do souboru se najednou klient LDAP sám spustil. Při následném otevření klienta LDAP z YAST byla už zatrhnuta možnost „Používat LDAP“.

Tyto informace byly zjištěny z dokumentace systému SUSE Linux Enterprise Server (verze pro server nikoli desktop) z poměrně staré verze 10. V dokumentaci se jasně píše, že tyto řádky musí být v souboru *nsswitch.conf* obsaženy pro autentizaci na LDAP serveru. (36)

Bez této „drobnosti“ by vůbec nemohlo možné celou práci dokončit, muselo by se hledat jiné, mnohem komplikovanější řešení, ať už instalaci jiné verze SUSE Linuxu nebo třeba vyhledávat jiné, neoficiální LDAP klienty s nejistou kompatibilitou.

```
# passwd: files nis
# shadow: files nis
# group:  files nis

passwd: compat
group:  compat

passwd_compat: ldap
group_compat: ldap

hosts:  files mdns4_minimal
networks:       files dns
```

Obrázek 22 Chybějící část konfigurace v souboru nsswitch.conf. Zdroj: Vlastní tvorba.

6.5 ODSTRANĚNÍ DOČASNÝCH DAT

Po úspěšném dokončení kroků popsaných v předchozích kapitolách bude možné do systému přihlásit jakéhokoli uživatele, který má platné Orion přihlašovací údaje.

V předchozí kapitole je popsáno, že se v systému automaticky vytvoří soubory a složky, které uživatel potřebuje ke kompletnímu přihlášení a své činnosti. Tyto soubory a složky ovšem v systému zůstávají i po té, co se počítač vypne, nijak se tedy nemažou. Po několika měsících používání tak v systému zůstane velmi mnoho zbytečných dat, kvůli kterým bude systém neustále bobtnat a zbytečně zabírat místo na disku. Z toho důvodu je nutné ještě zajistit automatické mazání těchto souborů.

Z předchozích kapitol vyplynulo, že po přihlášení LDAP uživatele, se v systému vytvoří určitá data v umístění */users/* (převzato z LDAP adresáře). Cílem poslední části práce je tedy zajistit, aby se tato data automaticky odstraňovala ve chvíli, kdy již nejsou potřeba, například během vypnutí počítače. K tomuto účelu byl vytvořen skript, který se spustí při vypínání systému a odstraní obsah složky */users/*. Tím je docíleno vždy čistého systému při následném spuštění počítače.

6.5.1 MAZACÍ SKRIPT

Skript byl nazván *rmusr* a musí být umístěn ve složce */etc/init.d*. V této složce jsou umístěny všechny skripty, které se automaticky spouští během vypínání nebo zapínání systému, takto je navržen systém SUSE Linux. Skript musí mít nastavená přístupová práva

rwxr-xr-x, neboli taková, díky kterým může být skript čten a spouštěn z jakéhokoli přihlášeného uživatele, ale upravován může být pouze uživatelem *root*. Algoritmus skriptu je poměrně jednoduchý. Skript je vypsán na následujících několika řádcích.

```
. /etc/rc.status
rc_reset
cd /users && rm -f -r * .*  >/dev/null 2>&1
echo done
rc_exit
```

První řádek skriptu znamená, že se použije skript *rc.status*, který má na starosti všechny spouštěcí skripty. Příkazem *rc_reset* se vyresetuje stav této služby. Skript se tváří pro systém jako služba a může mít vlastní stav, například že běží, neběží, chyba atd. Tento příkaz říká, pokud služba běží, musí se vypnout a znovu spustit, pokud služba neběží, musí se spustit. Tyto dva řádky musí vždy ve skriptu být, aby byl skript rozeznatelný a ovladatelný systémem. (32)

Následující (třetí) řádek dává systému následující instrukce: Jdi do složky */users/* a smaž veškerý obsah této složky bez jakéhokoli dotazu nebo potřebě potvrzení od uživatele, o dokončení mazání nebo případných chybách neinformuj (respektive standardní výstup nezobrazuj). O mazání souborů tedy uživatel není informován a v případě potíží se nenaruší proces vypínání. Proces mazání probíhá tedy zcela v tichosti.

Zde je třeba upozornit, že je nutné napsat příkazy pro určení cesty (*cd /users*) a tiché mazání (*rm -f -r * .* >/dev/null 2>&1*) do jednoho řádku spojené symboly *&&*. Kdyby byly příkazy napsané samostatně, byl by skript velmi nebezpečný. V situaci, ve které by z nějakého důvodu neexistovala složka */users/* (například je smazána), by příkaz pro umístění do složky */users/* skončil nezdarem. Následoval by další příkaz v pořadí, tedy tiché mazání. Systém by pak začal mazat sám sebe, což by mělo fatální následky.

```
lrwxrwxrwx   1 root    root        4  8. úno  2011 reboot -> halt*
-rwxr-xr-x   1 root    root      387 23. led 11.03 rmusr*
-rwxr-xr-x   1 root    root     2127 12. srp  2008 xfs*
-rwxr-xr-x   1 root    root     4323 21. úno  2009 xinetd*
-rwxr-xr-x   1 root    root     5363 20. úno  2009 ypbind*
linux-li9v:/etc/init.d #
```

Obrázek 23 Umístění skriptu rmusr v /etc/init.d. Zdroj: Vlastní tvorba.

6.5.2 KONFIGURACE

Další část skriptu tvoří konfigurační soubor, který se musí jmenovat stejně jako skript, tedy *rmusr*. Tento soubor se nachází ve složce */etc/sysconfig* a slouží k případné další konfiguraci skriptu. Přístupová práva postačí *rw-r--r--*, tedy taková, která povolují soubor číst kterémukoli uživateli a upravovat pouze *rootem*. V případě této práce není potřeba žádná další konfigurace a tak tělo souboru tvoří pouze jeden řádek povinný řádek:

RMUSR_OPTIONS=""

```
-rw-r--r--   1 root root   170 23. led 11.13 rmusr
-rw-r--r--   1 root root   878  8. úno  2011 words
-rw-r--r--   1 root root  1869  8. úno  2011 yast2
-rw-r--r--   1 root root  1759  8. úno  2011 ypbind
linux-li9v:/etc/sysconfig #
```

Obrázek 24 Umístění konfiguračního souboru rmusr ve složce /etc/sysconfig. Zroj: Vlastní tvorba.

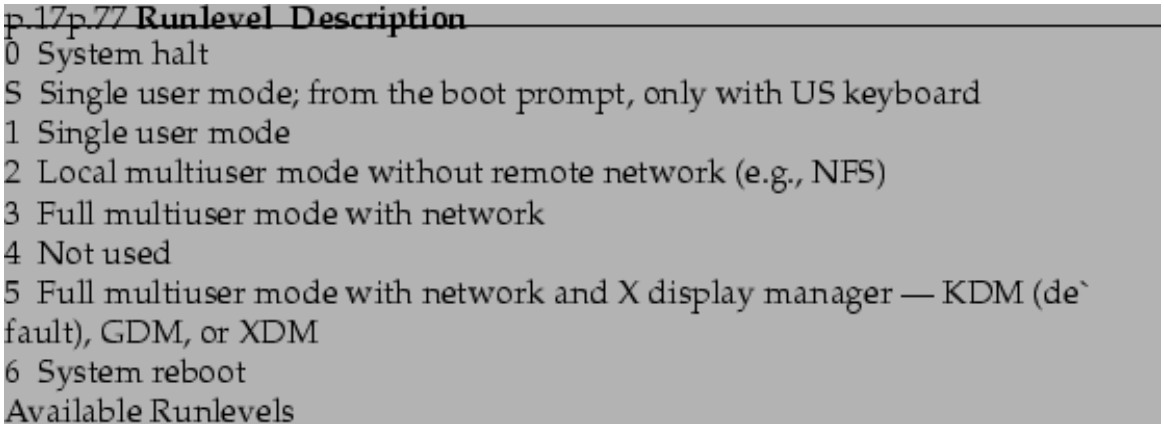
Skript je nyní jako celek zařazen mezi ostatní služby, nicméně ve výchozím stavu je vypnutý. O tom se lze snadno přesvědčit příkazem *chkconfig rmusr*, na který bude vráceno *off*, tedy vypnuto. Službu je tedy nutné zapnout příkazem *chkconfig rmusr on*. O tom, že je služba zapnutá se opět můžeme přesvědčit příkazem *chkconfig rmusr*, na který bude již vráceno *on*, neboli zapnuto. Tento postup je zachycen na následujícím obrázku.

```
linux-li9v:/etc/init.d/rc3.d # chkconfig rmusr
rmusr  off
linux-li9v:/etc/init.d/rc3.d # chkconfig rmusr on
linux-li9v:/etc/init.d/rc3.d # chkconfig rmusr
rmusr  on
linux-li9v:/etc/init.d/rc3.d #
```

Obrázek 25 Zapnutí mazacího skriptu. Zdroj: Vlastní tvorba.

Při aktivaci se skriptu se stane ještě jedna zásadní věc. Do složek */etc/init.d/rc3.d* a */etc/init.d/rc5.d* se automaticky vytvoří systémové linky[15], které vedou na skript. Jednotlivé složky *rcx.d* představují úrovně (režimy) spouštění operačního systému, je jich 8: *rcS.d, rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, rc6.d.* Podle toho, v jakém režimu se systém spustí, se začnou spouštět linky z příslušné složky. Umístěním linků do jednotlivých složek můžeme jednoduše zvolit, kdy chceme skript, umístěný v */etc/init.d/*, spouštět. Na následujícím obrázku je popsáno, jaké režimy spouštění představují jednotlivé složky. (33)

[15] Systémový link neboli symlink je soubor, který představuje odkaz na nějaký jiný soubor.


p.17p.77 **Runlevel Description**
0 System halt
S Single user mode; from the boot prompt, only with US keyboard
1 Single user mode
2 Local multiuser mode without remote network (e.g., NFS)
3 Full multiuser mode with network
4 Not used
5 Full multiuser mode with network and X display manager — KDM (de` fault), GDM, or XDM
6 System reboot
Available Runlevels

Obrázek 26 Popis jednotlivých režimů spuštění systému. Zdroj: http://www-uxsup.csx.cam.ac.uk/pub/doc/SUSE/SUSE9.0/adminguide-9.0/node17.html#tab:boot.runlevel

Pro účely jednotné autentizace v učebně KL206 jsou zajímavé složky *rc3.d* a *rc5.d*. Linky, které jsou umístěny ve složce *rc3.d*, jsou spuštěné ve chvíli, kdy je spuštěn systém v textovém režimu, respektive „plném textovém multiuserovém režimu se sítí“. Linky, které jsou umístěny ve složce *rc5.d*, jsou spuštěné ve chvíli, kdy je systém spuštěný v grafickém režimu, neboli v „plném grafickém multiuseru se sítí“. Právě tyto linky ve skutečnosti spouští skripty. Je tedy zajištěno, že se skript spustí jak v textovém režimu, tak v grafickém.

V předchozích odstavcích se vždy píše o linkách. Ke skriptu může totiž vést několik linků, stejně tak je tomu u skriptu *rmusr*. Na následujícím obrázku je patrné, že k tomuto skriptu vedou dva linky, *K01rmusr* a *S09rmusr*. Z toho důvodu, že linků ve složkách *rcx.d* nachází velmi mnoho, jsou na obrázku vypsány pouze ty, které mají v názvu *rmusr*.

```
linux-li9v:/etc/init.d/rc3.d # l |grep rmusr
lrwxrwxrwx  1 root root    8 23. led 11.10 K01rmusr -> ../rmusr*
lrwxrwxrwx  1 root root    8 23. led 11.10 S09rmusr -> ../rmusr*
```

Obrázek 27 Systémové linky, které vedou ke skriptu rmusr, umístěné ve složce rc3.d.

U těchto linků jsou zásadní jejich názvy. První písmeno v názvu označuje, zda se skript bude spouštět při vypnutí (písmeno K) nebo při zapnutí systému (písmeno S) pro daný režim spuštění (vypnutí systému neznamená vypnutí počítače, může to být pouhá změna režimu například z textového na grafický). Následující čísla znamenají, v jakém pořadí se začnou linky spouštět. Zbývající část je název skriptu. (34)

Link, který vykoná mazací skript, se tedy spustí při vypnutí systému jako první v pořadí a při zapnutí systému jako devátý v pořadí. V tomto případě, kdy je potřeba

smazat obsah složky */users/*, by stačilo, aby se skript spouštěl pouze jednou, tedy buď při startu, nebo při ukončení. Redundance linků je zde realizována pouze pro jistotu v případě, že by jeden z nějakého důvodu přestal fungovat.

6.6 PŘÍPRAVA UŽIVATELSKÉHO PROFILU PRO VÝUKU V KL206

V současné chvíli je tedy přihlášení vzdáleného LDAP uživatele plně funkční, po ukončení práce se odstraní dočasná data. K opravdovému nasazení ještě schází připravit uživatelský profil tak, jak se vyžaduje během výuky v KL206. Uživatelský účet je sice funkční, nicméně stále v něm chybí několik programů a dalších souborů, které jsou pro výukové účely nutné, například programy pro správu softwaru Novell, který se v KL206 vyučuje.

Pokud je v systému vytvořený lokální uživatel, je tomuto uživateli systémem automaticky přiřazen domovský adresář, který v sobě obsahuje několik programů a dalších souborů nutných pro výuku, nicméně vzdálenému uživateli se tak neděje. Lokální uživatel se vytváří podle předem definovatelných schémat (kostra neboli skeleton), ve kterých je definováno, jaké soubory a programy má uživatel mít k dispozici. Pro vzdáleného uživatele tomu tak není. Je tedy nutné docílit, aby vzdálený uživatel měl k dispozici stejný profil, jako uživatel lokální.

Postupů při realizaci výše zmíněného cíle může být více, nicméně v této práci byl zvolen následující postup:

1. Vytvořit nového lokálního uživatele stanice.
2. Zkopírovat soubory domovského adresáře lokálního uživatele do libovolného nového adresáře, například *usr_profile.*
3. Adresář *usr_profile* zkopírovat do systémové složky */etc/.*
4. Upravit konfigurační soubor *common-session-pc*, který pomocí vhodného PAM modulu automaticky zkopíruje soubory z adresáře *usr_profile* do domovského adresáře nově vytvořeného uživatele. Tento soubor se nachází v adresáři */etc/pam.d*.

Na následujícím obrázku je zachycena úprava soubor *common-session-pc*. Šedivě zvýrazněnou část je nutné do souboru dopsat. Jak funguje konfigurace PAM modulů a co

přesně znamenají jednotlivé řádky, je popsáno kapitole 24. Díky této úpravě se s použitím PAM modulu *pam_mkhomedir.so* zkopíruje obsah adresáře *usr_profile* do domovského adresáře každého nového LDAP uživatele.

```
session required pam_mkhomedir.so skel=/etc/usr_profile umask=0077
session  optional   pam_mkhomedir.so
session required    pam_limits.so
session required    pam_unix2.so
session optional    pam_krb5.so
session optional    pam_umask.so
session optional    pam_gnome_keyring.so    auto_start_if=gdm
```

Obrázek 28 Obsah souboru common-session-pc. Zdroj: Vlastní tvorba.

Během přihlašování LDAP uživatele se začnou soubory kopírovat. Vzhledem k tomu, že souborů je několik tisíc a celková jejich velikost dosahuje téměř 300MB, je proces přihlašování viditelně delší. Po úspěšném přihlášení je ovšem k dispozici uživatelský profil vzdáleného LDAP uživatele se stejnými parametry jako profil lokální, je tedy vše připraveno pro výuku.

Při autentizaci některých uživatelů vůči *ldap.zcu.cz* nastal další problém související s nastavením loginShell na hodnotu: *loginShell: /bin/tcsh.* Jedná se o starší nastavení uživatelských účtů, u kterých se má o zpracování přihlašovacích skriptů starat shell[16] *tcsh*. V případě nastavení *loginShell: /bin/bash*, tedy že se bude o přihlášení starat shell *bash,* problém nenastal. Shell *tcsh* totiž neprovede korektně interpretaci skriptů z nastavení uživatelského prostředí a některé aplikace (Novell client, ConsoleOne apod.) poté nefungují.

Bylo nutné obrátit se na správce LDAP serveru na CIVu s nastavením. Hodnota *loginShell: /bin/bash*, je standardní nastavení při současném způsobu vytváření uživatelů. Dle sdělení CIVu bylo z asi 15 000 účtů nastaveno *loginShell: /bin/tcsh* asi u 800 uživatelů dříve založených.

[16] Shell – Prostředí příkazového řádku, které interpretuje uživatelovy příkazy.

7 ZÁVĚR

Vývoj autentizačních metod se zcela jistě nezastavil, nicméně určitě se oproti minulosti zpomalil. Některé metody se přestaly používat, jiné se naopak ustálily. Jako důkaz ustálení slouží i tato práce, ve které se úspěšně podařilo aplikovat známé technologie, jako jsou LDAP, Kerberos či PAM, do, pro Západočeskou univerzitu, doposud neznámého prostředí operačního systému SUSE Linux Enterprise Desktop.

Cíle, které byly pro tuto práci určené, jsou splněny. Kromě rozšíření plně funkční jednotné autentizace může tato práce posloužit i jako návod pro zprovoznění jednotné autentizace v SUSE Linuxu a stejně tak může být užitečná i pro jiné operační systémy na bázi Unix. Některé detaily se zcela jistě mohou lišit, ovšem obecný postup bude vždy podobný.

SUSE Linux patří dnes k nejrozšířenějším Linuxových distribucím a to především ve firemní sféře. Právě z důvodu častého firemního nasazení má SUSE Linux vyvinuto mnoho nástrojů, které dělají tento systém robustní a dostatečně bezpečný. Jiné Linuxové distribuce samozřejmě mohou obsahovat podobné nástroje jako SUSE, nicméně se často soustředí na obyčejné koncové uživatele. Postrádají různé konfigurační nástroje, které je nutné doinstalovat a dokonfigurovat.

SUSE Linux se i z těchto bezpečnostních důvodů využívá během výuky v učebně KL206, kde se některé, v této práci zmíněné, technologie také vyučují a představují, především LDAP. To, že se autentizaci podařilo úspěšně zprovoznit, velmi výrazně usnadní učitelovu práci během výuky.

Učitel nyní nemusí ztrácet čas režijními povinnostmi, jako jsou přípravy uživatelských účtů, mazání starých účtů, obnovování zapomenutých hesel, informování studentů a další. Studenti se jednoduše přihlásí na svůj účet tak, jako kdekoli jinde na univerzitě, a mohou začít pracovat. Zároveň v případě technických komplikací bude moci učitel nahlédnout do této práce a prověřit nastavení. Díky této diplomové práci je konečně výuka Katedry výpočetní a didaktické techniky na Fakultě pedagogické plně začleněna do autentizačního systému ZČU, a tak je výuka v operačním sytému SUSE Linux plně srovnatelná jako v systému Windows.

Bude jistě zajímavé sledovat, jak se budou autentizační postupy vyvíjet a jakým tempem se budou postupně aplikovat do praxe. Je možné, že se v budoucnu objeví nějaké bezpečnostní hrozby, kvůli kterým bude muset univerzita rychle přejít na jiné modernější technologie. Nicméně dokud se tak nestane, což by se díky současným aplikovaným technologiím na ZČU stát nemělo, nebude nutné přecházet na jiné mechanismy a tato práce zůstane aktuální.

V současné době se nevídanou rychlostí rozšiřují mobilní telefony, tablety a jiná přenosná zařízení. Je jen otázkou času, kdy se bude plánovat rozšiřování jednotné autentizace a přístup do univerzitní domény právě pro tato zařízení. V blízké době nebude pravděpodobně nutné přecházet na nové technologie ani z těchto důvodů. Zcela jistě postačí aplikovat postupy, které jsou sepsány v této práci, do jiných operačních systémů, jako jsou například Android nebo iOS.

Přes všechny moderní pokročilé bezpečnostní mechanismy i nadále zůstává největší bezpečnostní riziko v uživateli. Samotný uživatel je a zřejmě ještě dlouho bude nejslabší článek bezpečnostního procesu. Dokud si uživatel nebude schopný vytvořit, zapamatovat a chránit své bezpečné heslo, nebudou nikdy data dostatečně dobře zabezpečena. Na tuto skutečnost je nutné poukazovat už během výuky na základních školách. Jedině tak je možné tuto jednoduchou myšlenku postupně dostat do povědomí všech počítačových uživatelů.

8 SEZNAM OBRÁZKŮ A TABULEK

Obrázek 1 Schéma činnosti Kerberos v5 Zdroj: Vlastní tvorba 7
Obrázek 2 Kerberos KDC servery využívané na ZČU. Zdroj: http://support.zcu.cz/index.php/Kerberos 9
Obrázek 3 Platnost uživatelova TGT lístku. Zdroj: Vlastní tvorba. 9
Obrázek 4: Šifrované přihlášení SSO za ZČU pomocí WebAuth a Kerberos. 11
Obrázek 5 Princip činnosti WebAuth v síti s autentizací Kerberos. Zdroj: Vlastní tvorba... 12
Obrázek 6 Atributy a hodnoty LDAP uživatele. Zdroj: Vlastní tvorba. 18
Obrázek 7 Větvení LDAP struktury na ZČU. Zdroj: Vlastní tvorba. 19
Obrázek 8 Možnosti zabezpečeného přihlášení do LDAP server v programu JXplorer. Zdroj: Vlastní tvorba 21
Obrázek 9 Schéma LDAP struktury na ZČU. Zdroj: Vlastní tvorba. 22
Obrázek 10 Umístění PAM modulů v SUSE Linuxu. Zdroj: Vlastní tvorba. 24
Obrázek 11 Obsah konfiguračního souboru. Zdroj: Vlastní tvorba 25
Obrázek 12 Výřez okna programu Správce softwaru. Zdroj: Vlastní tvorba. 27
Obrázek 13 Nastavení protokolu NTP pomocí nástroje v balíku YAST. Zdroj: Vlastní tvorba. 31
Obrázek 14 Výzva pro nainstalování PAM modulů. Zdroj: Vlastní tvorba. 32
Obrázek 15 Nastavení Kerberos klienta pomocí balíku nástrojů YAST. Zdroj: Vlastní tvorba. 33
Obrázek 16 Uživatel získal Kerberos ticket po příkazu kinit. Zdroj: Vlastní tvorba 34
Obrázek 17 Konfigurace LDAP klienta. Zdroj: Vlastní tvorba 35
Obrázek 18 Cesta k domovskému adresáři v LDAP databázi zjištěna programem JXplorer. Zdroj: Vlastní tvorba. 36
Obrázek 19 Balíčky potřebné pro konfiguraci LDAP klienta. Zdroj: Vlastní tvorba. 36
Obrázek 20 Automaticky vytvořená adresářová struktura přihlášeného vzdáleného uživatele kulda převzatá z LDAP databáze. Zdroj: Vlastní tvorba 37
Obrázek 21 Chybová hláška informující o nefunkčnosti LDAP nastavení. Zdroj: Vlastní tvorba. 38
Obrázek 22 Chybějící část konfigurace v souboru nsswitch.conf. Zdroj: Vlastní tvorba..... 39
Obrázek 23 Umístění skriptu rmusr v /etc/init.d. Zdroj: Vlastní tvorba. 40
Obrázek 24 Umístění konfiguračního souboru rmusr ve složce /etc/sysconfig. Zroj: Vlastní tvorba. 41
Obrázek 25 Zapnutí mazacího skriptu. Zdroj: Vlastní tvorba. 41
Obrázek 26 Popis jednotlivých režimů spuštění systému. Zdroj: http://www-uxsup.csx.cam.ac.uk/pub/doc/SUSE/SUSE9.0/adminguide-9.0/node17.html#tab:boot.runlevel 42
Obrázek 27 Systémové linky, které vedou ke skriptu rmusr, umístěné ve složce rc3.d. 42
Tabulka 1 Operace definované ve funkčním modelu LDAP. Zdroj: http://www.samuraj-cz.com/clanek/adresarove-sluzby-a-ldap/ 20

9 SEZNAM LITERATURY

1. Malý pohled za oponu Single Sign-On. PŘIBYL, Tomáš. *Computerworld: Deník pro IT profesionály*[online]. 2009 [cit. 2013-11-17]. Dostupné z: http://computerworld.cz/securityworld/maly-pohled-za-oponu-single-sign-on-47282.

2. Single Sign On. *Autenticationworld.com* [online]. © 2006 [cit. 2013-11-17]. Dostupné z: http://www.authenticationworld.com/Single-Sign-On-Authentication/.

3. Introduction to Single Sign-On. *The Open Group* [online]. © 1995-2010 [cit. 2013-11-18]. Dostupné z: http://www.opengroup.org/security/sso/sso_intro.htm.

4. Single Sign-On. *The Open Group* [online]. © 1995-2010 [cit. 2013-11-18]. Dostupné z: http://www.opengroup.org/security/sso/.

5. ABOUT: FREQUENTLY ASKED QUESTIONS ABOUT THE MIT KERBEROS CONSORTIUM. *MIT Kerberos Consorsium* [online]. 2007 [cit. 2013-11-27]. Dostupné z: http://www.kerberos.org/about/FAQ.html.

6. Kerberos: The Network Authentication Protocol. *Massachusetts Institute of Technology* [online]. [2002], 2014/03/11 [cit. 2013-11-27]. Dostupné z: http://web.mit.edu/kerberos/.

7. MIT Kerberos & Internet trust (MIT-KIT) Consortium. *MIT Kerberos Consortium* [online]. 2007 [cit. 2013-11-27]. Dostupné z: http://www.kerberos.org/.

8. Kerberos protokol a Single sign-on. BOUŠKA, Petr. *Samuraj* [online]. 2010 [cit. 2013-12-10]. Dostupné z: http://www.samuraj-cz.com/clanek/kerberos-protokol-a-single-sign-on/.

9. Ověřování prostřednictvím protokolu Kerberos V5. MICROSOFT. *Microsoft TechNet* [online]. 2014 [cit. 2013-12-13]. Dostupné z: http://technet.microsoft.com/cs-cz/library/cc783708(v=ws.10).aspx.

10. KERBEROS PROTOCOL TUTORIAL. *MIT Kerberos Consortium* [online]. 11/27/2007 [cit. 2013-12-13]. Dostupné z: http://www.kerberos.org/software/tutorial.html.

11. Kerberos. ZÁPADOČESKÁ UNIVERZITA. *Support.zcu.cz: server uživatelské podpory* [online]. [2003], 1/30/2013 [cit. 2014-12-29]. Dostupné z: http://support.zcu.cz/index.php/Kerberos.

12. WebAuth Features. *Stanford University* [online]. [2003], 5/11/2011 [cit. 2014-1-10]. Dostupné z: http://webauth.stanford.edu/features.html.

13. LPS:WebAuth. ZÁPADOČESKÁ UNIVERZITA. *Support.zcu.cz: server uživatelské podpory* [online]. [2003], 9/20/2013 [cit. 2014-1-10]. Dostupné z: http://support.zcu.cz/index.php/LPS:WebAuth#Dal.C5.A1.C3.AD_odkazy.

14. WebAuth Technical Specification. STANFORD UNIVERSITY. *INFORMATION TECHNOLOGY SERVICES* [online]. [2003], 8/29/2013 [cit. 2014-1-10]. Dostupné z: http://webauth.stanford.edu/protocol.html#compwkdc.

15. WebAuth. *Eyrie* [online]. [2002], 29/08/2013 [cit. 2014-1-10]. Dostupné z: http://www.eyrie.org/~eagle/software/webauth/

16. Kategorie:WEBnet. *Support.zcu.cz: server uživatelské podpory* [online]. [2003], 29/08/2013 [cit. 2014-1-19]. Dostupné z: http://support.zcu.cz/index.php/Kategorie:WEBnet.

17. GROUPER. *Internet2* [online]. ©2014 [cit. 2014-3-17]. Dostupné z: http://www.internet2.edu/products-services/trust-identity-middleware/grouper/#service-overview.

18. Referenční dokumentace služeb. ZÁPADOČESKÁ UNIVERZITA. *Support.zcu.cz: server uživatelské podpory* [online]. [2003], 13.3.2007 [cit. 2014-1-19]. Dostupné z: http://support.zcu.cz/index.php/Referen%C4%8Dn%C3%AD_dokumentace_slu%C5%BEeb

19. LDAP. PYSZKO, Tomas. *Fakulta informatiky Masarykovy univerzity* [online]. 2008 [cit. 2014-2-22]. Dostupné z: http://www.fi.muni.cz/~kas/p090/referaty/2008-podzim/ct/ldap.html.

20. Adresářové služby a LDAP. BOUŠKA, Petr. *Samuraj* [online]. 14.09.2007 [cit. 2014-2-22]. Dostupné z: http://www.samuraj-cz.com/clanek/adresarove-sluzby-a-ldap/

21. AD/Organizační jednotky pracovišť. ZÁPADOČESKÁ UNIVERZITA. *Support.zcu.cz: server uživatelské podpry* [online]. [2003], 14/3/2011 [cit. 2014-2-23]. Dostupné z: http://support.zcu.cz/index.php/AD/Organiza%C4%8Dn%C3%AD_jednotky_pracovi%C5%A1%C5%A5.

22. LDAP. NOVOTNÝ, Jiří. *Fakulta informatiky Masarykovy univerzity* [online]. 2011 [cit. 2014-2-23]. Dostupné z: http://www.fi.muni.cz/~kas/p090/referaty/2011-podzim/ct/ldap.html

23. Adresářová služba X.500 a LDAP. LASOŇ, Martin. *INNET | VŠB - Technická univerzita Ostrava* [online]. [2003] [cit. 2014-2-23]. Dostupné z: http://homel.vsb.cz/~las03/ldap/.

24. Chapter 2. Authentication with PAM. *Doc.openSUSE.org - Documentation Guides & Manuals* [online]. [2012] [cit. 2014-2-2]. Dostupné z: http://doc.openSUSE.org/products/draft/SLES/SLES-security_sd_draft/cha.pam.html#sec.pam.pam-config.

25. PAM - správa autentizačních mechanismů. BOBČÍK, Boleslav. *ROOT.CZ* [online]. 19. 9. 2000 [cit. 2014-2-2]. Dostupné z: http://www.root.cz/clanky/pam-sprava-autentizacnich-mechanismu/.

26. 3. PAM Essentials. *Pluggable Authentication Modules* [online]. [© 2001-2003] [cit. 2014-2-12]. Dostupné z: http://www.freebsd.org/doc/en/articles/pam/pam-essentials.html.

27. PAM (Pluggable Authentication Modules). *User Authentication HOWTO* [online]. 2000-05-02 [cit. 2014-2-12]. Dostupné z: http://tldp.org/HOWTO/User-Authentication-HOWTO/x115.html.

28. Hapter 6. Network Authentication with Kerberos. *Doc.openSUSE.org - Documentation Guides & Manuals*[online]. 2012-06-25 [cit. 2014-3-3]. Dostupné z: http://doc.openSUSE.org/products/draft/SLES/SLES-security_sd_draft/cha.net.kerberos.html.

29. Kerberos. *Support.zcu.cz: server uživatelské podpory* [online]. [2003], 30.1.2013 [cit. 2014-3-3]. Dostupné z: http://support.zcu.cz/index.php/Kerberos.

30. Kerberos. *Support.zcu.cz: server uživatelské podpory* [online]. [2003], 22.10.2013 [cit. 2014-3-3]. Dostupné z: http://support.zcu.cz/index.php/Orion_skupiny.

31. SUSE Linux Enterprise Server Security Guide. *Doc.openSUSE.org - Documentation Guides & Manuals*[online]. 25.6.2012 [cit. 2014-3-4]. Dostupné z: http://doc.openSUSE.org/products/draft/SLES/SLES-security_sd_draft/index.html.

32. Chapter 13. The SUSE LINUX Boot Concept / 13.4. Init Scripts. *SUSE LINUX – Administration Guide*[online]. [2004] [cit. 2014-3-10]. Dostupné z: http://www.novell.com/documentation/SUSE91/SUSElinux-adminguide/html/ch13s04.html.

33. The SUSE Linux Boot Concept. *UCS University Computing Service* [online]. 2003-11-05 [cit. 2014-3-10]. Dostupné z: http://www-uxsup.csx.cam.ac.uk/pub/doc/SUSE/SUSE9.0/adminguide-9.0/node17.html#tab:boot.runlevel.

34. Chapter 13. The SUSE LINUX Boot Concept / 13.3. Changing Runlevels. *SUSE LINUX – Administration Guide* [online]. [2004] [cit. 2014-3-10]. Dostupné z: http://www.novell.com/documentation/SUSE91/SUSElinux-adminguide/html/ch13s03.html.

35. Andrew File System (AFS). *Information Technology* [online]. ©2008 [cit. 2014-3-10]. Dostupné z: http://technology.pitt.edu/network-web/hosting-timesharing/afs.html.

36. Configuring an LDAP Client with YaST. *SUSE* [online]. ©2014 [cit. 2014-3-31]. Dostupné z: https://www.SUSE.com/documentation/sles10/book_sle_reference/data/sec_ldap_yast_client.html.

10 RESUMÉ

The development of authentication methods wasn't stopped yet, however the development gets slower. Some of the methods were stopped to use other are stabilized and used. As a proof of stabilization is this thesis where were successfully applied known technologies like LDAP, Kerberos or PAM to the environment of operation system SUSE Linux Enterprice Desktop that were totally unknown for the University of West Bohemia.

The aims of the thesis have been met. Besides fully working unified university authentication can this thesis be used like a guide for setting up unified authentication in other operating systems based on Unix. Some details can be different but general procedures will be similar.

SUSE Linux is used in classroom KL206 for educational purposes. Now the teacher doesn't need to waste the time with administration obligations, for example creating user accounts, deleting old accounts, restoring forgotten passwords and so on. Students can easily login to their account like anywhere in university and can start to work. Also, in case of technical problems the teacher can open this thesis and check right settings. Thanks to this thesis can be teaching of the Department of Computer Science and Educational Technology fully integrated to authentication system of University of West Bohemia so the teaching in operation system SUSE Linux is fully comparable to teaching under operating system Windows.

11 PŘÍLOHY

1. Scope of the Single Sign-On Standard

The scope of the Single Sign-On Standard ***(code-named XSSO at the present)***, is to define services in support of:

- the development of applications to provide a common, single end-user sign-on interface for an enterprise, and
-
- the development of applications for the co-ordinated management of multiple user account management information bases maintained by an enterprise.

Functional Objectives

User Sign-on Interface

The following functional objectives have been defined for the XSSO in support of a user sign-on interface:

- The interface shall be independent of the type of authentication information handled.
- Change of user controlled authentication information shall be supported. This is interpreted as initially being restricted to change of user password although capability for future extension shall not be precluded.
- Support shall be provided for a caller to establish a default user profile. User selection from a set of available user profiles is not required to be supported but shall not be precluded as a future extension.
- Support for the initiation of cleanup services on session termination, or sign-off, shall be supported.
- Provision of a service to enable a caller to notify the XSSO implementation of a change of user controlled authentication information by an application other than the XSSO implementation is an optional requirement and may be supported.
- XSSO shall not predefine the timing of secondary sign-on operations.
- Note: This means that XSSO shall not require that all sign-on operations are performed at the same time as the primary sign-on operation. This would result in the creation of user sessions with all possible services even though those services may not actually be required by the user.

User Account Management Interface

The following functional objectives have been defined for the XSSO in support of a user account management interface:

- The creation, deletion, and modification of user accounts shall be supported.
- The setting of attributes for individual user accounts shall be supported. The attributes to be supported shall include as a minimum those necessary to support the XBSS.

Non-Functional Objectives

The non-functional objectives of the XSSO are:

- The XSSO shall be authentication technology independent. The interface shall not prescribe the use of a specific authentication technology, nor preclude the use of any appropriate authentication technology.
    - ***Note:*** *Some authentication technology, for example those based upon challenge-response mechanisms of which a user held device is a component may not be appropriate for use as part of secondary sign-on functions.*
- XSSO shall be independent of platform or operating system. XSSO shall not preclude the integration of common desktops or common servers, including mainframes. There is no expectation that such desktops or servers shall be capable of integration within XSSO without modification.

Security Objectives

The security objectives to be met by an implementation of XSSO are:

- XSSO shall not adversely affect the resilience of the system within which it is deployed.
- XSSO shall not adversely impact the availability of any individual system service.
- XSSO shall not provide access by principals to User Account Information to which they would not be permitted access within the controlling security domain for that information.
- An XSSO implementation shall audit all security relevant events which occur within the context of the XSSO.
- An XSSO implementation shall protect all security relevant information supplied to or generated by the XSSO implementation such that other services may adequately trust the integrity and origin of all security information provided to them as part of a secondary sign-on operation.
- The XSSO shall provide protection to security relevant information when exchanged between its own constituent components and between those components and other services.

Out of Scope

The following aspects are not considered to be within the current scope of XSSO:

- Support for single sign-on across enterprise system boundaries.
- User initiated change of non-user configured authentication information, for example magnetic badges, smartcards, etc.
- Selection of alternative user profiles on user sign-on.
- Configuration and management of alternative sets of user profiles.
- Maintenance of the integrity of the single sign-on user account information base with underlying individual service user account information bases when those underlying user account information bases are modified by means other than XSSO provided functionality.
- Graphical and command line user interfaces to XSSO based services. These are the province of applications written to utilise the XSSO.