Aleksandr Litreev

CYBERGAME PLATFORM ADMINISTRATIVE USER ACCESS MANAGEMENT

Bachelor Thesis

Data and Networking

November 2016

KYAMK
University of Applied Sciences

KYAMK
University of Applied Sciences

Author	Degree	Time
Aleksandr Litreev	Bachelor of Data & Networking	November 2016

Thesis Title
Cybergame platform administrative user access management 31 pages

Commissioned by
Kymenlaakso University of Applied Sciences (48220 Kotka, Pääskysentie 1)

Supervisor
Vesa Kankare, Senior Lecturer

Abstract

The main objective of this bachelor thesis is a research of feasibility of establishing an intermediate service authorization for an isolated special ICTLAB Cybergame project environment. Created interim authorization service must support one of the following protocols: LDAP, RADIUS or TACACS+.

The objective of this thesis was a research and implementation of interim authorization server, that will be able to route requests to the main server or will proceed request itself in synchronization with the main server. Another objective of this thesis is to get acquainted with Windows Server and Linux operating systems as well, as with LDAP, RADIUS or TACACS+ protocols and OpenLDAP and Active Directory software.

This paperwork explains the process of configuration of LDAP Read Only Domain Controller with Microsoft Active Directory, LDAP replication with macOS Server and Pass-Through Authentication Server with Linux.

The Author describes several implementations for LDAP protocol on different operating systems, such as Windows, Linux and macOS. He also provides guidance on initial configuration of different LDAP software (OpenLDAP, OpenDJ, Active Directory and so on).

Keywords
documentation, model, thesis, report writing, ldap, authorization, authentication, windows server

CONTENTS

ABBREVIATIONS

Abbreviation	**Explanation**
LDAP	Lightweight Directory Access Protocol
RADIUS	Remote Authentication Dial-In User Service
TACACS	Terminal Access Controller Access-Control System
AD	Active Directory (LDAP solution developed by Microsoft Corporation and implemented in Windows Server OS family)
RODC	Read-Only Domain Controller
DNS	Domain Name Server
DSRM	Directory Services Restore Mode (Special recovery mode for Windows Server OS family)
SASL	Simple Authentication and Security Layer
IETF	Internet Engineering Task Force
GUI	Graphical User Interface
DHCP	Dynamic Host Configuration Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
CN	Common Name (Part of the X.500 Directory Specification, which defines nodes in a LDAP directory)
DC	Domain Component (Part of the X.500 Directory Specification, which defines nodes in a LDAP directory)
OU	Organization Unit (Part of the X.500 Directory Specification, which defines nodes in a LDAP directory)

1 INTRODUCTION

1.1 Main objectives

The main objective of this bachelor thesis is a research of feasibility of establishing an intermediate service authorization for an isolated special ICTLAB Cybergame project environment. Created interim authorization service must support one of the following protocols: LDAP, RADIUS or TACACS+.

Security design of created solution must expect to be compromised. Hacking or compromisation of solution should not affect main server and should not compromise network of ICTLAB.

This paperwork contains overview of available authorization and authentication protocols, overview of implementations and solutions, that author has found. Author widely used websites of software vendors and books, related to topic of this bachelor thesis.

1.2 Environment

This bachelor thesis was designed for ICTLAB environment during ICTLAB CyberGame project. ICTLAB already have working Windows Server running Active Directory Domain Services (including LDAP authorization & authentication server) and operating network, build on top of Cisco hardware. For experimental purposes VMWare Workstation Pro has been used as a platform to build and test intermediate authorization services.

During this project, author used different operating systems trying to establish intermediate authorization service: Windows Server 2012 R2 Core, Windows Server 2012 R2 Standard and latest Ubuntu Linux distribution.

1.3 Author background

Author studied at Federal State Budget-Financed Educational Institution of Higher Education The Bonch-Bruevich Saint - Petersburg State University of Telecommunication for Bachelor degree in Software Engineering and at Kymenlaakso University of Applied Sciences, participating in Double Degree program in Data & Networking. Author held course “Windows Systems” by Marko Oras (Kymenlaakso University of Applied Sciences). During this course, got acquainted with Active Directory principles of operation, have learned about domain controller management.

Already had experience in Linux and BSD systems administration, Windows Server 2012 management. Familiar with networking software, such as WireShark. Moreover, author familiar with such programming languages as C/C++, JavaScript (Node.JS) and PHP.

Worked as a lead system administrator in ZonaSpace (largest coworking center in Russia), as Chief Technical Officer at Mobium LLC (software vendor), as Senior Back-End Developer at Avalank LLC (software vendor). During 2016 russian parliament elections served as Cybersecurity Expert at Open Russia Foundation (founded by Mikhail Khodorkovsky).

2 PROJECT GOALS

The objective of this thesis was a research and implementation of interim authorization server. Interim server should be able to route requests to the main server. Also, implemented solution can proceed requests by itself in synchronization with the main server.

As said previously, security design of created solution must be expected to be compromised. Hacking or compromisation of solution should not affect the main server and should not compromise network of ICTLAB.

Finally, one more objective of this thesis is to get acquainted with authentication and authorization software on Windows, Linux and macOS operating systems.

3 OVERVIEW

3.1 Overview of available authorization and authentication protocols

LDAP (Lightweight Directory Access Protocol) - is an open, vendor-neutral directory service protocol that runs on a layer above the TCP/IP stack. It provides a mechanism used to connect to Internet directories and to modify them. LDAP is widely used to store user credentials centralized and it provides common entry point for various software for user authentication and authorization. Authentication and authorization is just a part of LDAP solutions. LDAP implemented in open-source and proprietary solutions. Most popular LDAP proprietary implementation is the Active Directory Server developed by Microsoft Corporation. OpenLDAP and FreeIPA servers are popular open-source solutions.

LDAP authentication and authorization flow is simple: client-side application sends an LDAP “Bind request” with user credentials, such as username and password, then LDAP server verifying credentials and replying with LDAP “Bind response”. No additional two-step verifications are required.

LDAP is flexible. All data transferred with LDAP is encrypted, if encryption is enabled on the server-side. (Microsoft, 2015)

RADIUS (Remote Authentication Dial-In User Service) - is a networking protocol that allows software to authenticate and authorize users. RADIUS runs in the application layer using UDP as a transport protocol. The RADIUS server is usually a background process running on a UNIX, Linux or Microsoft Windows server.

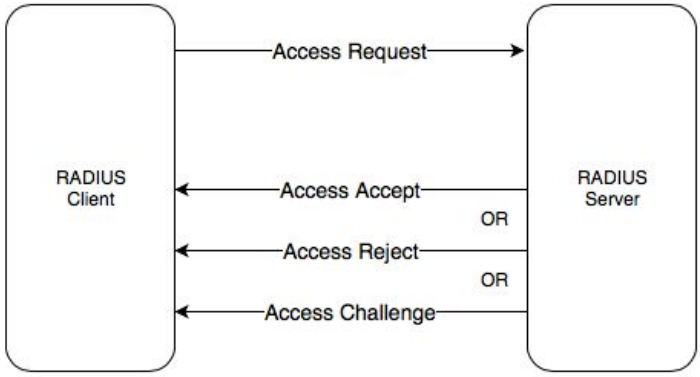

Figure 1: RADIUS workflow design

RADIUS client-side application sends an “Access Request” query to RADIUS server with specified credentials (usually, username and password). Then, RADIUS Server may send one of the following responses (as seen in Fig.1):

- “Access Accept” — access granted, user authorized and authenticated with specified credentials using RADIUS protocol.

- “Access Reject” — access denied, user is not authorized and authenticated. This error may occur when user has entered wrong username or/and password or this account does not exist.

- “Access Challenge” — there is one more step required to grant access to the system. This works like two-step verification of a user. Server requests additional information from the user, such as a secondary password, PIN, token, or card.

RADIUS only encrypts the user's password as it travels from the RADIUS client to RADIUS server. (Cisco, 2006)

TACACS+ (Terminal Access Controller Access-Control System Plus) — is a proprietary, closed protocol developed by Cisco. Based on an previous version of protocol called TACACS, TACACS+ implements authentication and authorization services. This protocol runs using TCP transfer protocol.

One of the main advantages of using TACACS+ is built-in security. It's automatically encrypts all the information that goes through it. This means, that TACACS+ is more secure and not vulnerable as RADIUS at this point.

Also, Cisco's TACACS+ solutions are able to provide logs of all commands that were made by authorized users.

Since main authorization server of ICTLAB is running Windows Server 2012 R2 with Active Directory services (LDAP-compatible software) this work expounds all features, principles of operation and details of LDAP protocol and implementations. (Cisco, 2008)

3.2 Overview of LDAP implementations

LDAP has significant amount of implementations. Open-source and proprietary solutions are available. For instance, few of such implementations are:

Active Directory (Microsoft Corporation) — database-based system for Windows domain networks, that provides authorization, authentication, policies and other services. Active Directory (AD) comes along with Windows Server. It allows to store user accounts centralized and manage them (Carter, 2003, page 193). AD is compatible with LDAP (Lightweight Directory Access Protocol). Active Directory contains Domain Services (AD DS) — core of Active Directory product, it stores data about users of the domain, it verifies user credentials and managing their permissions and access levels. Server running this service is called a domain controller. It works as LDAP server also. Also AD runs Certificate Services (a certification authority server), Federation Services (provides users with single sign-on access to systems and applications across enterprise boundaries), Lightweight Directory Services (provides directory services) and Rights Management Services (limiting access to documents and files). (Microsoft, 2014)

OpenLDAP (The OpenLDAP project) — is a free and open-source solution for LDAP protocol. OpenLDAP is written in C programming language and is available for several operating system, such as Linux, Android, OS X, Solaris, Microsoft Windows and BSD systems. Main module of OpenLDAP is a “slapd” — stand-alone daemon that serves main functionality of the system. Services as “ldapsearch”, “ldapadd”, “ldapdelete” are running on the client-side machines. OpenLDAP is also known to be able to use pass-through authentication. (ArchWiki, 2013)
Also, OpenLDAP implemented in macOS Server and known as macOS Profile Management.

389 Directory Server — Red Hat product, also known as Red Hat Directory Server. Licensed under GPL license, which means this software is open source. 389 Directory Server also compatible with latest LDAP protocol version, it has graphical interface that can be used for administration. This software mostly used in networks, where workstations runs Fedora and RHEL operating systems.

ApacheDS (Apache Directory Service) — free and open-source solution, released under Apache license. ApacheDS is completely compatible with latest LDAP protocol version and available for all popular platforms, such as Windows, OS X and Linux (Karasulu, 2013). Also, Apache provides Apache Directory Studio — software with graphical user interface, that allows to easily manage ApacheDS user accounts and domains. Apache Directory Studio built on top of Eclipse.

LDAP.JS (Joyent) — free and open-source framework for implementing LDAP client-side applications and servers in Node.JS programming language.

LDAP.JS is completely compatible with the LDAP protocol itself, moreover, it is interoperable with OpenLDAP.

Active Directory, OpenLDAP and LDAP.JS allows to create a LDAP server to authorize and authenticate users. Created LDAP server should act as a replicated LDAP domain controller. Thereby, synchronization of LDAP databases should be implemented.

Synchronization should be implemented in order to create a new LDAP server, which will act as a replica for a main one. (Joyent, 2015)

OpenDJ — open source multiprotocol software for authorization and authentication, that supports LDAP, REST and Web services for access. It also provides authentication to another LDAP directory service, such as Active Directory. (Forgerock, 2016)

4 IMPLEMENTATION

There are various ways to implement secondary (or read-only) domain server. In this work, several possible techniques of implementation are described.

During this work, network of servers was implemented in virtual environment with VMWare Workstation (as it seen in Fig. 2).

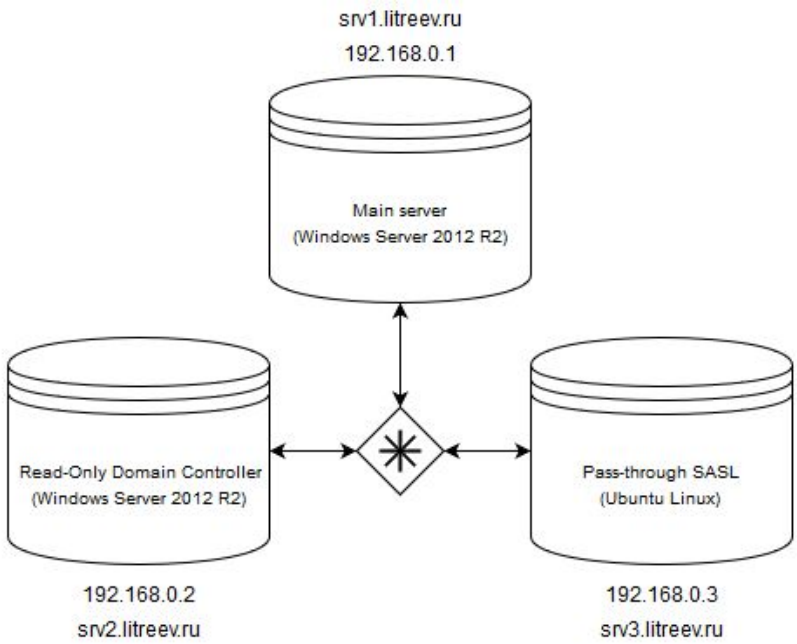

Figure 2: Active Directory Domain Services Configuration Wizard (Step 3)

- **srv1.litreev.ru** - main Windows Server 2012 R2, domain controller, running Active Directory and DNS servers with records for all the servers in the network.
- **srv2.litreev.ru** - Read-Only Domain Controller running on Windows Server 2012 R2. Automatic replication from srv1.litreev.ru is configured.
- **srv3.litreev.ru** - Ubuntu Linux server, running OpenLDAP with Pass-Through SASL feature compiled.

Devices with client-side software should be connected to the same network with Read-Only Domain Controller or Pass-through SASL server.

4.1 Configuring OpenLDAP as pass-through authentication server

The SASL (Simple Authentication and Security Layer) is a network module for user authentication. With SASL, different applications may share specific verification mechanism for the user's identity. In other words, SASL is a layer between protocols and mechanisms that makes verification mechanisms protocol-independent (IETF, 2006).

SASL is a proposed standard RFC 4422 of IETF (Internet Engineering Task Force).

LDAP servers are able to permit other LDAP servers to authenticate users with Pass-Through authentication feature. With Pass-Through authentication, user, first of all, authenticating on the intermediate server and only then connection transfers to the main LDAP resource (IETF, 2006).

This option should be compiled into OpenLDAP. To compile OpenLDAP with pass-through feature, following command should be executed on obtained source-code files:

```
./configure --enable-spasswd --with-cyrus-sasl
```

Compiled OpenLDAP will be able to store password with SASL-link. For instance:

```
userPassword: {SASL}dummy@litreev.ru
```

After following these steps "saslauthd" daemon should be installed and enabled.

Pass-through authentication workflow:

1. OpenLDAP receiving BIND request with parameters DN1 and PWD1
2. OpenLDAP reading userPassword attribute from DN1 entry

3. DN1 password matching SASL password, OpenLDAP processing an SASL authentication with USER@DOMAIN and PWD1 credentials
4. SASL authentication daemon using provided credentials to look up for the user using backend service (for instance, Active Directory) and gets the matching DN, DN2
5. SASL processing a BIND operation with DN2 and PWD1
6. The backend managing the BIND and returning response to SASL
7. SASL returning response to OpenLDAP (Succeed or Failed)
8. OpenLDAP returning response to the LDAP client-side application

Parameters to the authentication backend should be obtained before the configuration process begins. Author took following Active Directory configuration as an example:

1. Active Directory server: ldap://srv1.litreev.ru
2. BIND DN: CN=Administrator,CN=Users,DC=litreev,DC=ru
3. BIND password: SALAsana16
4. Users branch: CN=DomainUsers,DC=litreev,DC=ru

After authentication backend parameters are obtained, SASL daemon must be configured.

First of all, LDAP protocol should be enabled for SASL. To enable LDAP for SASL, change “MECH” variable to “ldap” in the SASL system configuration file “/etc/sysconfig/saslauthd”:

```
MECH=ldap
```

Then, all connection information should be entered in the SASL configuration file “/etc/saslauthd.conf”:

```
ldap_servers: ldap://srv1.litreev.ru
ldap_search_base: CN=DomainUsers,DC=litreev,DC=ru
ldap_timeout: 30
ldap_filter: sAMAccountName=%U
```

```
ldap_bind_dn:
CN=Administrator,CN=Users,DC=litreev,DC=ru
ldap_password: SALAsana16
ldap_deref: never
ldap_restart: yes
ldap_scope: sub
ldap_use_sasl: no
ldap_start_tls: no
ldap_version: 3
ldap_auth_method: bind
```

To apply changes SASL service should be restarted:

```
service saslauthd restart
```

Now, communication between OpenLDAP and SASL should be established. To do so, set OpenLDAP password-check method to SASL and enable communication between daemons using mutex:

```
pwcheck_method: saslauthd
saslauthd_path: /var/run/saslauthd/mux
```

Add OpenLDAP user to SASL group:

```
usermod -a -G sasl ldap
```

To finish configuration of LDAP SASL authentication, add SASL to OpenLDAP configuration:

```
sasl-host       localhost
sasl-secprops   none
```

Configuration is done and all the daemons should work properly now. (LTB Project, 2013)

4.2 Configuring RODC on Windows Server

RODC (Read-Only Domain Controller) - new feature of Windows Server that available in Windows Server 2008 or newer. Main purpose of Read-Only Domain Controller is a deploying of domain controller in places, where physical security cannot be guaranteed. Features of RODC are: faster logon time, credential caching, administrators roles separation.

Read-Only Domain Controller is a popular solution for small and medium businesses with small networks, poor physical security and relatively small user number. (Microsoft, 2011)

4.2.1 Configuring RODC using GUI (Graphical User Interface)

In Windows Server 2003 and earlier versions there was a feature called Secondary Domain Controller, but it was deprecated and removed from newer versions of Windows Server. Active Directory (as a part of Windows Server) can be used as a RODC — Read-Only Domain Controller. Read-Only Domain Controller allows to create a replica of a main domain controller, which will follow all the changes from the main server.

Read-Only Domain Controller can be configured with GUI (Graphical User Interface) or with Windows Powershell®. GUI interface is available in Standard versions of Windows Server, but removed from Core edition.

First of all, new server should be associated with the domain server of main server. Then, Active Directory Domain Services (ADDS) package should be installed.

After ADDS is installed, Windows Server will automatically propose you to promote this server (as seen in Fig. 3) to a domain controller.

Figure 3: Windows Post-deployment configuration notification

Promotion to a domain controller is necessary in order to create Read-Only Domain controller.

Assuming that the domain already exists, new server should be added to an existing domain. Windows also requires specifying of the domain name for this operation. Domain name of the main server must be specified.

Windows Server is a highly secured and strict server operating system that requires administrator accounts for such operations as post-deployment configuration. Credentials of administrator account for main domain controller must be supplied to perform this kind of operation.

Thus, if domain name is “litreev.ru” and username of administrator is “Administrator”, ADDS Configuration Wizard window will look like it shown of the Fig. 4.

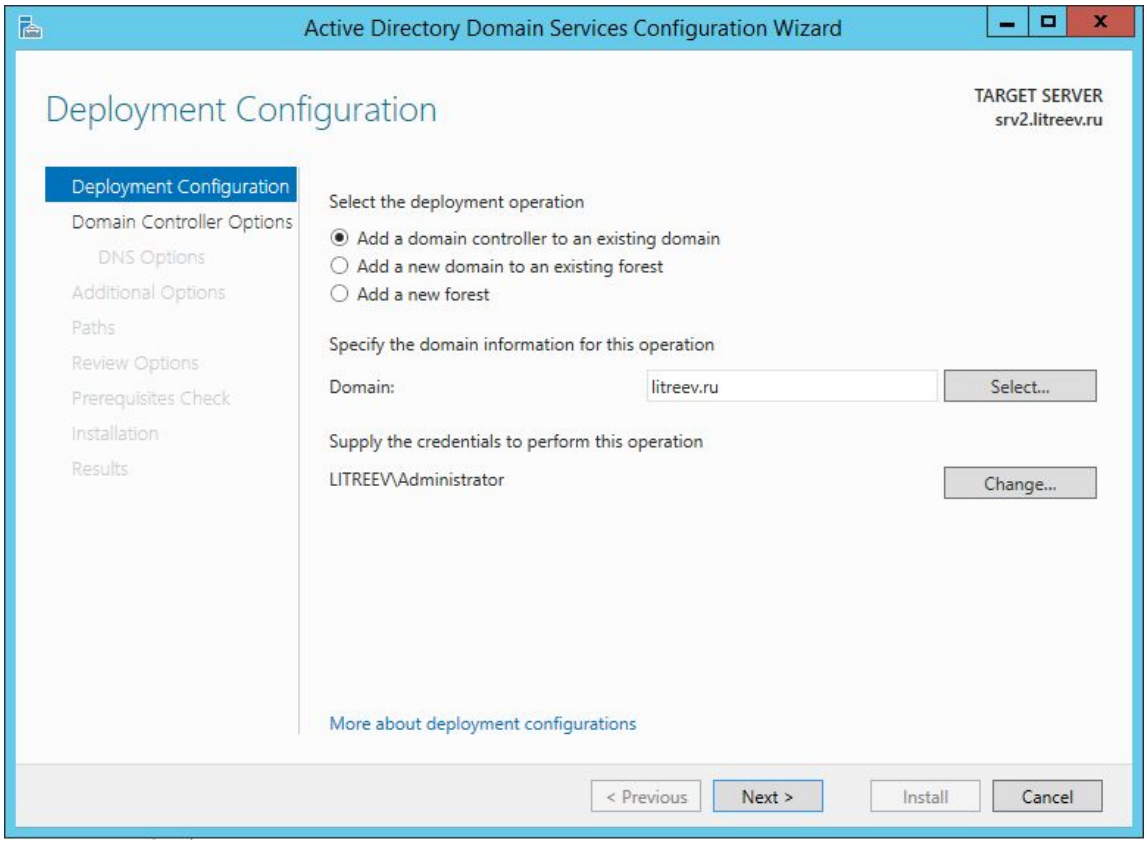

Figure 4: Active Directory Domain Services Configuration Wizard (Step 1)

After domain name is specified and administrator's credentials are provided, domain controller capabilities should be specified as well. This server is going to be a Read-Only Domain Controller, therefore, RODC option should be selected.

According to Figure 5, Windows requires Directory Services Restore Mode (DSRM) password. DSRM is a unique Windows Server feature that allows administrator to get access to Active Directory database in case if something went wrong.

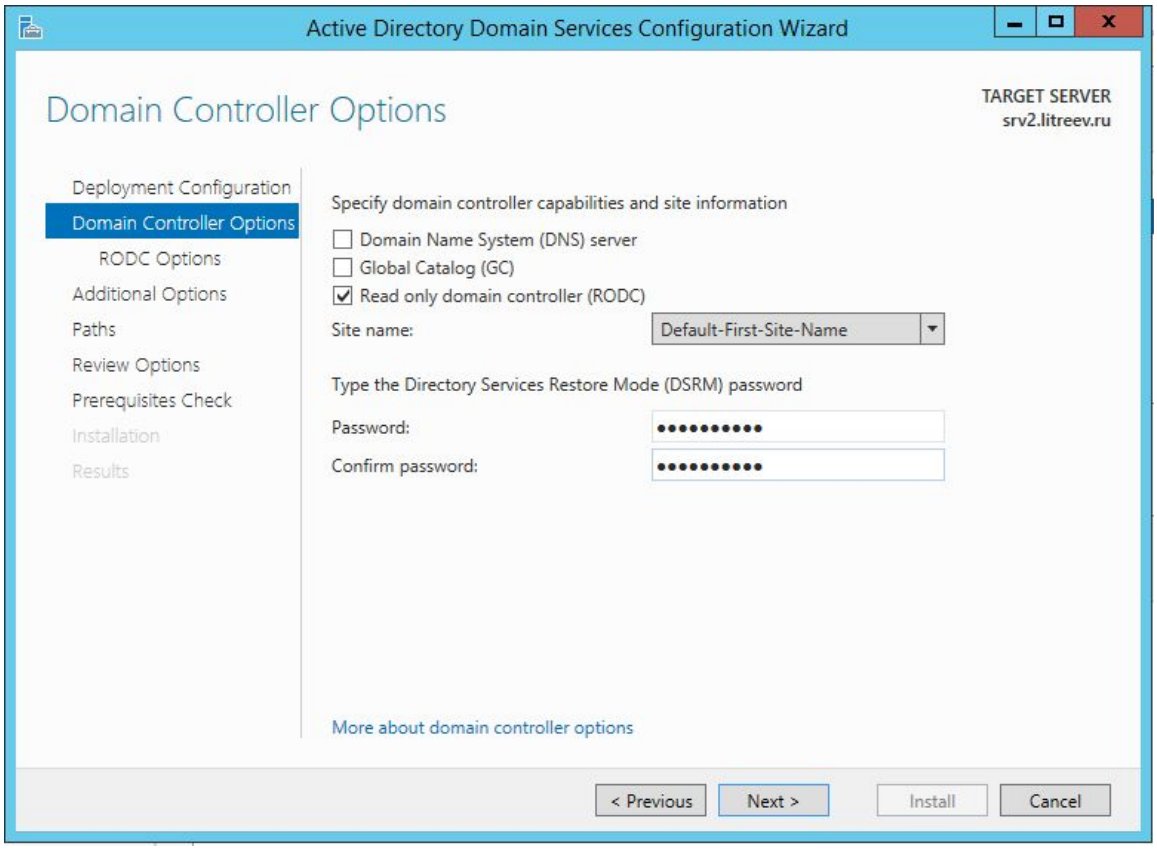

Figure 5: Active Directory Domain Services Configuration Wizard (Step 2)

Now that the configuration of RODC is almost completed, replication source server should be specified.

Windows Server is able to look up for other Windows Server machines automatically, so, replication source domain name will appear automatically on the third step of ADDS Configuration Wizard. If it is not — domain name should be specified manually. In case of this bachelor thesis, domain name of experimental main domain controller is “srv1.litreev.ru” (as seen in Fig. 6).

> ***Note:*** *If Windows is unable to resolve an IP address of specified domain name, make sure that:*
>
> 1. *DNS server is configured correctly*
> 2. *DNS server is specified in the network adapter system or provided by DHCP server.*

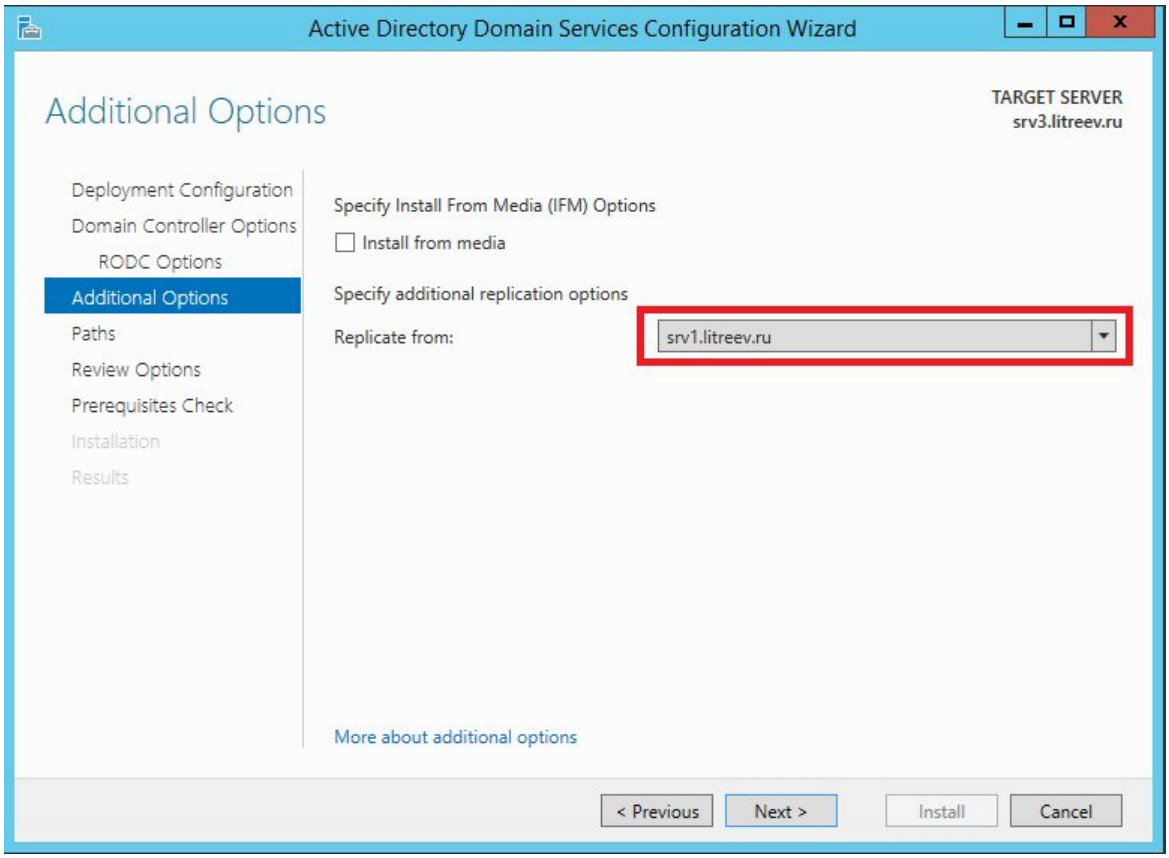

Figure 6: Active Directory Domain Services Configuration Wizard (Step 3)

After this, Windows Server will check all the prerequisites and will run generated Windows Powershell® script automatically. System will restart after finishing installation.

4.2.2 Configuring RODC with Windows Powershell®

Windows Server has a special edition called Core. It is the special version without GUI (Graphical User Interface) which allows to configure itself only using Windows Powershell® and standard Windows command prompt.

The following commands allows to configure AD RODS automatically for “litreev.ru” domain name, using “srv1.litreev.ru” server as a replication source.

```
Import-Module ADDSDeployment

Install-ADDSDomainController `
```

```
-AllowPasswordReplicationAccountName
@("LITREEV\Allowed RODC Password Replication Group")
`
-NoGlobalCatalog:$true `
-Credential (Get-Credential) `
-CriticalReplicationOnly:$false `
-DatabasePath "C:\Windows\NTDS" `
-DenyPasswordReplicationAccountName
@("BUILTIN\Administrators", "BUILTIN\Server
Operators", "BUILTIN\Backup Operators",
"BUILTIN\Account Operators", "LITREEV\Denied RODC
Password Replication Group") `
-DomainName "litreev.ru" `
-InstallDns:$false `
-LogPath "C:\Windows\NTDS" `
-NoRebootOnCompletion:$false `
-ReadOnlyReplica:$true `
-ReplicationSourceDC "srv1.litreev.ru" `
-SiteName "Default-First-Site-Name" `
-SysvolPath "C:\Windows\SYSVOL" `
-Force:$true
```

4.3 Configuring Apple macOS Server to work with Active Directory

In some networks, not all clients are bound to the same Active Directory domain as the server. Also, if enterprise network contains only computers running Apple macOS, probably, network engineers and software administrators would like to use Apple macOS Server solution to authenticate and authorize users using Apple Profile Manager.

Apple Profile Manager is designed to work with Active Directory or any other LDAP server as well as without it.

Apple Profile Manager uses Digest MD5 authentication protocol, which is supported by Active Directory also. This compatibility makes connection between AD and Apple Profile Manager possible. This feature does not require any specific setup except in the following two cases:

- The server that runs the Apple Profile Manager s on a different domain than the clients, using this network.
- Apple Profile Manager supposed to use third-party LDAP server.

In these cases, Apple Profile Manager (and Wiki software, which comes along with Apple Profile Manager) should be configured to use plain-text authentication.

Following command should be executed via macOS Terminal to enable plain-text authentications:

```
sudo serveradmin stop wiki

sudo /usr/libexec/PlistBuddy -c 'set
:Auth:Authenticator plaintext'
/Library/Server/Wiki/Config/collabd.plist

sudo serveradmin start wiki
```

Since plain-text authentication is vulnerable for Man-In-The-Middle attacks, it is highly recommended to use SSL encryption with Apple Profile Manager. (Apple, 2016)

4.4 Configuring OpenDJ for Pass-Through Authentication

Installation of OpenDJ is quite easy and can be executed by using both command prompt and graphical user interface. Following commands will automatically download and execute installation software.

```
$ cd /path/to/sun-java6/bin
```

```
$ ./javaws
http://download.forgerock.org/downloads/opendj/2.4.5/install/QuickSetup.jnlp
```

Installation process contains few steps with graphical interface, where user may choose domain settings he need (as seen in Fig. 7), set security settings and so on.

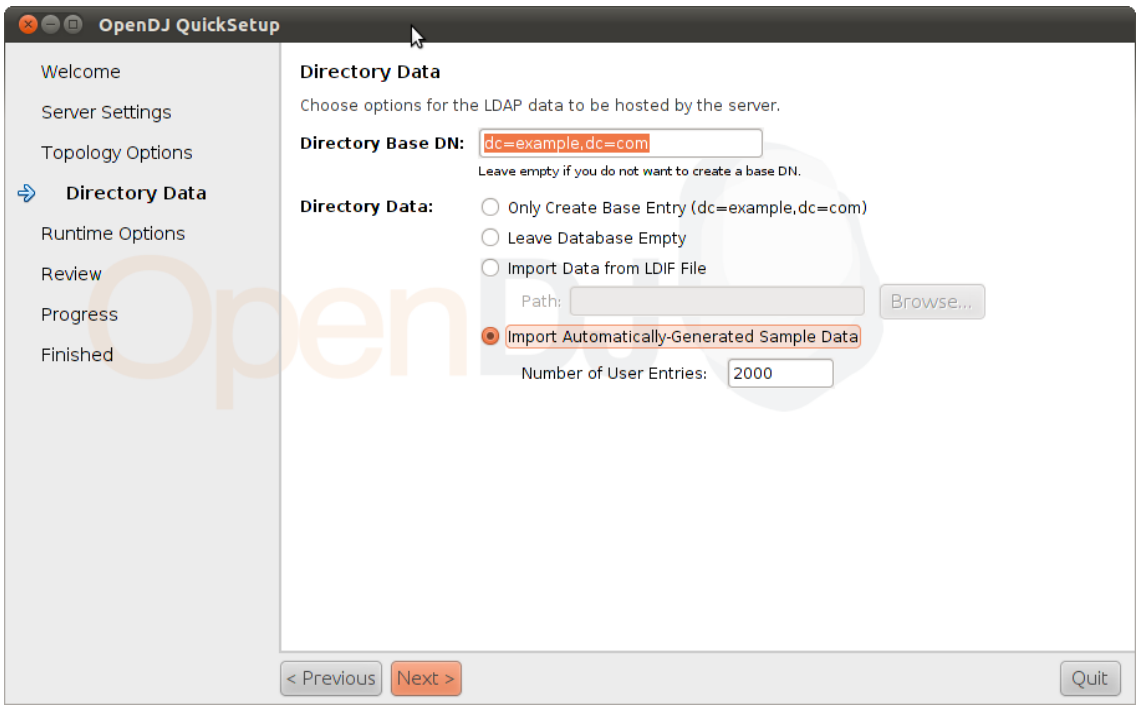

Figure 7. OpenDJ Installation on Linux

Process of installation is the same for all platforms: Windows, Linux and OS X.

After installation, OpenDJ Control Panel software starts automatically, it allows to manage entries of OpenDJ database (as seen in Fig.8). Also, it shows current server status and details of machine (such as Java version) (Poitou, 2016).

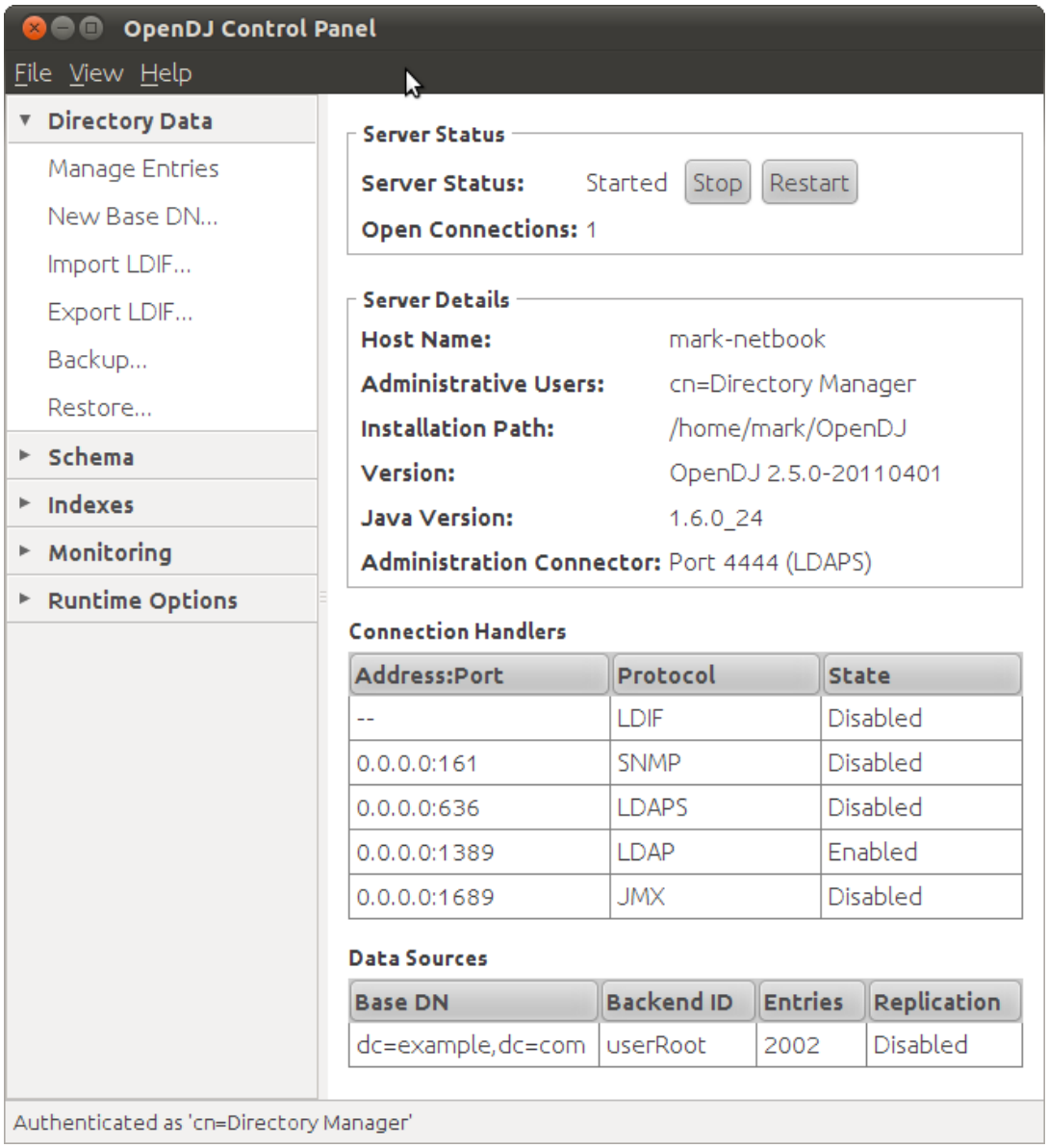

Figure 8. OpenDJ Control Panel on Linux

However, graphical interface is not always available on Linux operating systems, and many network engineers and system administrators prefer command-line prompt (Terminal) over GUI. The author prefers to use the command line as a universal method (some distributions of Linux comes without GUI). Therefore, all the next steps are done with command-line prompts.

First of all, OpenDJ should have configured Pass-Through Authentication policy. To configure such policy, administrator should use the *dsconfig* command:

```
dsconfig \
 create-password-policy \
 --port 4444 \
 --hostname opendj.example.com \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 --type ldap-pass-through \
 --policy-name "PTA Policy" \
 --set
primary-remote-ldap-server:pta-server.example.com:636
\
 --set mapped-attribute:uid \
 --set mapped-search-base-dn:"dc=PTA Server,dc=com" \
 --set mapping-policy:mapped-search \
 --set use-ssl:true \
 --set trust-manager-provider:JKS \
 --trustAll \
 --no-prompt
```

Configured policy will be used for users under *DC=PTA Server,DC=com* domain controller. Users must have the share same *uid* values on both servers.

Database of OpenDJ uses *UID* as the naming attribute, and all the records also have *CN* attribute. Active Directory entries use *CN* as the naming attribute. User accounts on both sides share the same *CN* records.

For instance, OpenDJ user profile with cn=LDAP PTA User and DN *UID=ldapptauser,OU=People,DC=example,DC=com* matches to an AD account with DN *CN=LDAP PTA User,CN=Users,DC=internal,DC=forgerock,DC=com*. To enable Pass-Through authentication via Active Directory for this user with *CN=LDAP PTA User*, following commands should be executed:

```
$ ldapsearch \
 --hostname opendj.example.com \
 --baseDN dc=example,dc=com \
 uid=ldapptauser \
 cn
dn: uid=ldapptauser,ou=People,dc=example,dc=com
cn: LDAP PTA User

$ ldapsearch \
 --hostname ad.example.com \
 --baseDN "CN=Users,DC=internal,DC=forgerock,DC=com" \
 --bindDN
"cn=administrator,cn=Users,DC=internal,DC=forgerock,DC
=com" \
 --bindPassword password \
 "(cn=LDAP PTA User)" \
 cn
dn: CN=LDAP PTA
User,CN=Users,DC=internal,DC=forgerock,DC=com
cn: LDAP PTA User
```

Now the OpenDJ server will work as Pass-Through Authentication server for selected users. To add a user to this authentication policy, user account attribute *pwdPolicySubentry* should have no value and created policy should be set as *ds-pwp-password-policy-dn* value for this user account. To set it so, the following command should be executed:

```
ldapsearch \
 --port 1389 \
 --baseDN dc=example,dc=com \
 uid=user.0 \
 pwdPolicySubentry
dn: uid=user.0,ou=People,dc=example,dc=com


$ ldapmodify \
 --port 1389 \
```

```
 --bindDN "cn=Directory Manager" \
 --bindPassword password
dn: uid=user.0,ou=People,dc=example,dc=com
changetype: modify
add: ds-pwp-password-policy-dn
ds-pwp-password-policy-dn: cn=PTA Policy,cn=Password
Policies,cn=config
```

Now, this user should be able to authenticate using OpenDJ Pass-Through Authentications with his credentials on Active Directory LDAP server (Forgerock, 2010).

If system administrator wants a group of user to use OpenDJ Pass-Through Authentication, he can implement a created policy not for just single user, but for the whole group.

Of course, following Pass-Through Authentication solution can work not only with Active Directory, but with any LDAP-compatible server.

4.5 Hardening security with firewall

Every piece of software may contain vulnerabilities, which may lead to exploitation of such vulnerabilities and, consequently, cyberattacks. To protect enterprise network, author recommends to use firewall software. Firewall (software) — is a software, that is responsible for the control of incoming and outgoing connections. Some of the operating systems are designed with embedded firewall software. The idea of firewall software is to block unwanted ports for incoming and outgoing connections. In order to setup firewall properly and keep LDAP working, firewall should be configured to keep LDAP ports 389 and 636 opened.

To protect Linux server after OpenLDAP or OpenDJ configuration, the author recommends to use UFW firewall. To install UFW on Ubuntu, system administrator should execute following command as a root:

```
apt-get install ufw
```

System will automatically download and install the latest version of UFW from software repositories. Next step is a firewall configuration.

In order to allow LDAP ports, we need to set ports 389 and 636 allowed for UFW. To do so, the following commands should be executed:

```
sudo ufw allow 389
sudo ufw allow 636
```

Also, it is possible to configure UFW to allow networking for specified application:

```
sudo ufw allow "OpenLDAP LDAP"
```

If administrator working with server via SSH tunnel, he should also allow SSH ports, otherwise, all SSH connections will be aborted by UFW firewall. Engineer may allow SSH with following command:

```
sudo ufw allow sshd
```

Now UFW configured. To start UFW with created network rules, following command should be executed:

```
sudo ufw enable
```

Other operating systems such as Windows and macOS are designed with embedded firewalls, which can be configured with graphical user interface as well, as with terminal commands. Moreover, there is more firewall software by third-party developers, such as Kaspersky Cloud EndPoint Security and others.

5 CONCLUSIONS

In the process of writing the thesis, the author became acquainted with existing authorization protocols, such as LDAP (Lightweight Directory Access Protocol), RADIUS (Remote Authentication Dial-In User Service) and TACACS+ (Terminal Access Controller Access-Control System Plus), and learned more about the principles of operation of these protocols.

During project work, enterprise network containing main server and two secondary servers were created in the virtual environment. The author has installed Windows Server 2012 R2 as a main server on a virtual machine and Active Directory Domain Services were configured on it. Also, DNS server was installed and configured.To create a replicated domain controller, another virtual machine was configured with Windows Server 2012 R2 Standard. This machine was configured to act as a read-only domain controller. Afterwards, the operating system of this machine was reinstalled to Windows Server 2012 R2 Core in order to create RODC, but with the Windows PowerShell commands. Both created servers were fully operational. Then, one more virtual machine was created with Ubuntu Linux operating system onboard. OpenLDAP was compiled and installed with Pass-Through features from the source code. Daemons of Simple Authentication and Security Layer and OpenLDAP were tweaked to work together seamlessly. Also, OpenDJ was configured the same way on the same machine as OpenLDAP.

Windows Server is easy to set up. Using graphical user interface it takes less than a five minute to configure read-only domain controller. Process of tweaking is intuitive and simple. Linux is a bit complicated, but more flexible and can be configured for any purposes of network administrators. OpenLDAP is a good alternative for proprietary Microsoft's product Active Directory. It is compatible and cross-platform solution.

LDAP protocol have been chosen, assuming that the main server of ICTLAB is running Windows Server and Active Directory, which is compatible only with this protocol.

Created solution helps main server stay secured even if secondary servers became compromised.

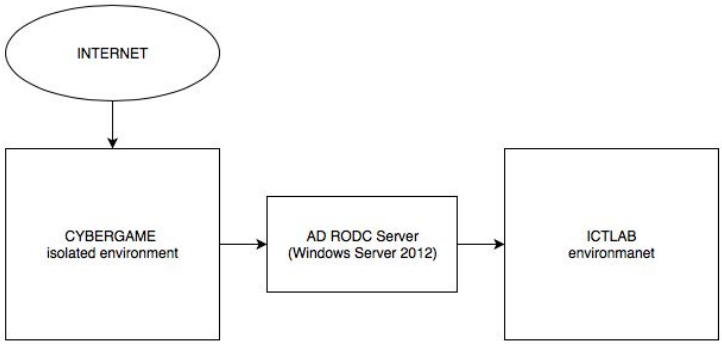

Figure 9: Cybergame & ICTLAB network architecture design

It is important to have no direct connection between Cybergame and ICTLAB (as it seen in Fig. 9). AD RODC working with two isolated networks and follow LDAP requests.

The author has succeeded with Active Directory, Apple Profile Manager, OpenLDAP and OpenDJ configuration. In fact, there are many other solutions to experiment with. All the overviewed solutions have a wide range of uses: university internal networks, small business networks and so on.

This bachelor thesis explains and describes only establishing of LDAP-based intermediate authentication and authorization server. However, there are more authorization and authentication protocols exists, such as RADIUS and TACACS. Every engineer should select such protocol, that corresponds to his needs and meets the network environment. For instance, LDAP protocol is perfect for enterprise network, that contains only computers and servers running Windows. If engineer working with Cisco devices running IOS/IOS-XR and his enterprise network contains Linux machines, he would probably try RADIUS or TACACS+ protocol. Every case is unique and authorization and authentication solution should be designed specifically for this case.

6 REFERENCES

- Apple (2016) Use Profile Manager or Wiki service with Active Directory or third-party LDAP services. Available at: https://support.apple.com/en-us/HT202285 (Accessed: 30 September 2016).
- ArchWiki (2013) OpenLDAP. Available at: https://wiki.archlinux.org/index.php/OpenLDAP (Accessed: 08 April 2016).
- Carter, G. (2003) LDAP System Administration: Putting Directories to Work. 1st Edition edn. United States: O’Reilly Media, Inc, USA.
- Cisco (2006) How Does RADIUS Work? Available at: Cisco (2006) How Does RADIUS Work. Available at: http://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/12433-32.html (Accessed: 06 April 2016). (Accessed: 06 April 2016).
- Cisco (2008) TACACS+ and RADIUS Comparison. Available at: http://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/13838-10.html (Accessed: 07 April 2016).
- Forgerock (2010) Configuring Pass-Through Authentication. Available at: https://backstage.forgerock.com/docs/opendj/3/admin-guide/chap-pta#about-pta (Accessed: 11 November 2016).
- Forgerock (2016) OpenDJ. Available at: https://forgerock.org/opendj/ (Accessed: 01 November 2016).
- IETF (2006) RFC4422 — Simple authentication and security layer (SASL). Available at: https://tools.ietf.org/html/rfc4422 (Accessed: 10 April 2016).
- Joyent (2015) Reimagining LDAP for Node.Js. Available at: http://ldapjs.org (Accessed: 08 April 2016).
- Karasulu, A. (2003) Proposal for an Apache Directory Project. Available at: http://directory.apache.org/original-project-proposal.html (Accessed: 10 November 2016).
- LTB Project (2013) Pass-Trough authentication with SASL. Available at: http://ltb-project.org/wiki/documentation/general/sasl_delegation (Accessed: 13 April 2016).

- Microsoft (2011) AD DS: Read-Only Domain Controllers. Available at: https://technet.microsoft.com/en-us/library/cc732801%28v=ws.10%29.aspx (Accessed: 11 April 2016).
- Microsoft (2014) Active Directory Domain Services Collection. Available at: https://technet.microsoft.com/en-us/library/cc780036(v=ws.10).aspx (Accessed: 07 April 2016).
- Microsoft (2015) Lightweight Directory Access Protocol. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/aa367008%28v=vs.85%29.aspx (Accessed: 06 April 2016).
- Poitou, L. (2016) OpenDJ installation guide - OpenDJ - confluence. Available at: https://wikis.forgerock.org/confluence/display/OPENDJ/OpenDJ+Installation+Guide (Accessed: 11 November 2016).
- Tcherniakhovski, A. (2012) Configuring OpenLDAP pass-through authentication to Active Directory. Available at: https://blogs.msdn.microsoft.com/alextch/2012/04/25/configuring-openldap-pass-through-authentication-to-active-directory/ (Accessed: 13 April 2016).