Bachelor's thesis

Information Technology

Identity Management

2013

Joni Helle

MULTI-TENANT ACTIVE DIRECTORY

– Organizational Units

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR´S THESIS | ABSTRACT
TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Identity Management

2013 | Pages 23

Instructor: Ossi Väänänen

Joni Helle

MULTI-TENANT ACTIVE DIRECTORY

The purpose of this research is to find out if Microsoft Active Directory could offer authentication services for multiple tenants without cross-tenant visibility. The focus of this research is Active Directory Organizational Units and those objects that commonly reside in them. This thesis approaches Multi-Tenancy by implementing Active Directory lock-down rules to default schema, effectively modifying default access and visibility permissions. These modifications prove to be very effective in blocking cross-tenant visibility and are the basis for Active Directory Multi-Tenancy.

KEYWORDS:

Microsoft Active Directory, Multi-Tenant

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ
TURUN AMMATTIKORKEAKOULU

Information Technology | Identity Management

2013 | Sivuja 23

Ohjaaja: Ossi Väänänen

Joni Helle

MULTI-TENANT ACTIVE DIRECTORY

Tämän tutkimuksen tarkoitus on selvittää voidaanko yhteen Microsoft Active Directory toteutukseen liittää monia asiakkuuksia ilman että asiakkaat tietävät toistensa olemassaolosta, ja näin ollen tarjota useille asiakkuuksille autentikointipalvelua. Tutkimuksen kohteena ovat Active Directoryn Organizational Unitit, sekä näissä yleisimmin sijaitsevat oliot. Tutkimus keskittyy lukitun Active Directoryn ajatukseen ja näiden ajatusten toteuttamiseen default schema objektien käyttöoikeusasetuksiin. Nämä muutokset osoittautuvat erittäin tehokkaiksi estämään asiakkuuksia näkemästä toisiaan ja ovat tutkimuksen lähtökohta.

ASIASANAT:

Microsoft Active Directory, Multi-Tenant

CONTENTS

LIST OF ABBREVIATIONS **5**
1 INTRODUCTION **6**
2 ACTIVE DIRECTORY MULTI-TENANCY **8**
2.1 Directory Structure 8
2.2 Object Security 9
2.3 Practical Tests 12
3 ANALYSING RESULTS **21**
4 CONCLUSION **22**
REFERENCES **23**

APPENDICES

Appendix 1: Forest Creation Answer File

LIST OF ABBREVIATIONS

ADUC	Active Directory Users and Computers tool. A systems administrator performs user account operations with this tool e.g. resets user passwords.
LDAP	Lightweight Directory Access Protocol that is used to access Directory Services.
LDP	Lightweight Directory Protocol. An old tool created by Microsoft to directly access Active Directory and to create custom LDAP queries.
Locked-Down	A Locked Down Active Directory is a Directory Service where Directory objects are stripped from unnecessary access control rules.
OU	Organizational Unit. A placeholder for objects in Active Directory.
SaaS	Software as a service. Effectively this means: one system with multiple clients without cross client visibility.
Tenant	A tenant is a synonym for a 3rd party or client. When there are multiple tenants it means multiple 3rd parties that should be unaware of each other.

1 INTRODUCTION

Microsoft Directory Services is a server side software made by Microsoft for identity management that is included as an optional component in all Microsoft Windows Server Standard, Enterprise, and Datacenter operating systems. The main use of identity management is authorization. Some examples of authorization could be illustrated by the following questions: is the user authorized to log in to this computer? Are these credentials authorized to access this document? Microsoft Directory Services identity management solution is established on the idea of a user name that everyone might know and a password that only the authorized user knows. If one of these is lost, one can not gain access to the service and must ask help from a Domain Administrator to gain the lost user name or a new password. Once a Domain Services is installed, it is given a name and is called a forest and inside this forest there is at least one domain. In most cases, like the one in this thesis, the forest contains one domain and can be thought as the same thing. Once a domain is created, an authenticated user can read

The Directory Services is like a file system with folders and files, except that these folders are called Organizational Units and the files are called objects. These objects vary like files and some are called user objects, some are group objects, and others are computer objects. These three objects are the most important objects for this thesis and these objects store user, or computer, information like passwords, group membership information, and unique identifiers. A computer, user, or a group can be a member of a group and usually a group is given the right to access a certain object. These permissions can be given directly or by inheriting them from a parent object like from the organizational unit where the object is located. These permissions are usually given to groups with descriptive names, for ease of administration, and users are joined to a group to gain access to the desired object.

When a new object is created in the Directory Service, it is created from the information located in the schema. The schema is a storage location containing all templates of all objects and can be modified as pleased. One can even break the whole Directory Services beyond repair by meddling with the schema too much. The templates located in the schema specify all information like a group object is a group object and not an user object, that is given a specific access control list, which are the most important part of this thesis, and by modifying these access control lists we can restrict user access from the beginning, meaning we are able to block users from parts of the Directory Service. These permissions can be blocked or allowed at any time after the object creation by a Domain Administrator or similar entity with the required permissions.

The idea for a multi-tenant Microsoft Active Directory came with questions ‘What are these access control rules in Directory objects?’, ‘Can we block access to Active Directory objects with access control rules?’ and ‘Do these work like NTFS file permissions?’. In order to answer these questions, a new Active Directory was created, modified and thoroughly tested in order to find out whether multiple clients can coexist in one Active Directory implementation, without cross-client visibility. System Administrator can restrict Active Directory object access and visibility by removing unnecessary entries from objects, and even create new objects with a premade access control list. Using these principles, the Active Directory was modified and thoroughly tested for object access breaches while simulating separated clients.

2 ACTIVE DIRECTORY MULTI-TENANCY

This research answers the question ‘Can Microsoft Active Directory identity services be offered as SaaS (Software As a Service) for multiple clients?’. Microsoft Active Directory does not natively support multi-tenancy without cross-client object visibility, but one can remove object visibility with access control lists by denying read permissions or by removing permissions that grant read access to the object in question thus creating separated Organizational Units (OU’s). However these changes may break important functionality of the directory services with poorly planned access restrictions, which could lead to a situation where the Directory can not read itself.

Due to the large amount of different functionalities in Microsoft Active Directory, the research scope is narrowed down to Active Directory Organizational Units and those objects that commonly reside in them, as they are the founding stones in creating separated entities for multi-tenancy in Active Directory. This research focuses on user, group and computer objects in addition to Organizational Units because they are needed for basic authentication services.

2.1 Directory Structure

A new Active Directory implementation was created and two new Organizational Units were created to the forest root ‘Clients’ Organizational Unit and ‘IAAS Provider’ Organizational Unit. The ‘Computers’ and ‘Users’ containers were not taken into use and all new objects were redirected to a new container inside IAAS Provider: “OU=_NewObjects,OU=IAAS Provider,DC=test,DC=turkuamk,DC=local” .

Each client is given their own personal Organizational Unit inside the ‘Clients’ Organizational Unit. Permissions to modify objects inside their personal Organizational Unit are given to the client administrator group. As a general

rule, all client specific data is located only inside the clients' client specific Organizational Unit.

2.2 Object Security

Microsoft Active Directory object permissions are 'Like file permissions in NTFS' (Sanderson & Rice, 2000, 25) By default, 'All users who log on with a valid username and password combination that is stored in the Active Directory database belong to the special identity group Authenticated Users' (Kane 2009, 112). Because the Authenticated Users group has read permissions to all objects in the domain by default, all clients can view all objects by default. There are two ways this access to an object could be removed:

Firstly, all undesired group permissions could be removed from objects. E.g. all permissions could be removed from all Comapny1's objects except Company1_Users. This way Company2_Users would not have access to the Company1's objects.

Secondly, it is possible to specifically deny access from Company2_Users to Company1's Organizational Unit. With a deny statement, a user can not access the resource, because an explicit deny statement always overrules an explicit allow rule. On the other hand an explicit allow rule overrides an inherited deny rule, which creates a very complicated situation because of many overlapping deny and allow statements are very hard to read correctly.

Unfortunately, the deny rule would have to be added to each client every time a new client would be brought to the domain. This leaves us with the option to remove all unnecessary permissions from all objects as the only realistic choice so that all new clients brought to the domain are not automatically allowed to see all objects.

In order to remove all unnecessary permissions, the necessary permissions need to be defined. Permissions can be categorized in three groups by access need: system access, administrative access, and user access. System Access includes object operations where the system accesses the object because of a

system-initiated reason. Administrative Access includes all object operations initiated from within Active Directory by a user. User Access includes all object operations initiated from outside Active Directory by a user.

A Schema of a new unmodified Windows 2008 R2 Directory Services has the following abbreviations (also known as ‘well known names’) in the default security description of Organizational Units: SY, DA, AO, PO, AU, and ED. These abbreviations of the well known names translate in the security tab of a new object to the following group names:

SYSTEM (SY) which is ‘used by the operating system and any processes that run using LocalSystem’ (Kane 2009, 112) LocalSystem is a server side service account used to run operating system processes.

Domain Admins (DA) which is the default group of the domain for workstation administrative tasks (Kane 2009, 109) and for domain-wide upkeep tasks.

Account Operators (AO) whose purpose is to ‘administer domain user and group accounts. By default, members can create, modify, and delete accounts for users, groups, and computers in all containers and Organizational Units (OU’s) of Active Directory, except the Built-in folder and the Domain Controllers OU.’ (Kane 2009, 106). This means daily tasks such as creating user accounts and resetting passwords.

Print Operators (PO) group ‘members can manage printers and document queues.’ (Kane 2009, 108). This means Active Directory Printer objects and workstation specific administrative tasks including document queues.

Authenticated Users (AU) which is a group for ’all users who log on with a valid username and password combination that is stored in the Active Directory database.’ (Kane 2009, 112). This group has read access to all Directory objects.

ENTERPRISE DOMAIN CONTROLLERS (ED) which is ‘used to facilitate access to forest-wide Active Directory services.’ (Kane 2009, 112) ‘All domain controllers are members of this group’ (Kane 2009, 112).

In addition to these explicit permissions, the object has inherited permissions for Pre-Windows 2000 Compatible Access which is a backwards compatibility group for Windows NT. ‘Members have read access on all users and groups in the domain. This group is provided for backward compatibility for computers running Microsoft Windows NT 4.0 and earlier.’ (Kane 2009, 108).

SELF which is ‘used as a placeholder for the current user.’ (Kane 2009, 113). This statement means the object itself and not the user that is trying to access it.

Enterprise Admins, which is used for the following ‘This group is added to the Administrator group on all domain controllers in the forest. This allows the global administrative privileges associated with this group, such as the ability to create and delete domains.’ (Kane 2009, 110).

Administrators which is a group whose ’members have complete and unrestricted access to the computer or domain controller locally, including the right to change their own permissions.’ (Kane 2009, 107).

These permissions can be categorized in the following way:

System Access: SYSTEM, and ENTERPRISE DOMAIN CONTROLLERS

Administrative Access: Domain Admins, Account Operators, Print Operators, Enterprise Admins, and Administrators

Client Access: Authenticated User, Pre-Windows 2000 Compatible Access, and SELF

Of these default permissions, the following are absolutely necessary for domain functionality:

SYSTEM, because it means all processes running on the domain controller. Without SYSTEM, the computer can not access the process it is trying to run.

ENTERPRISE DOMAIN CONTROLLERS, because this default permission is used to grant access for domain controllers for domain wide active directory services, such as performing queries and Access Control List parsing.

In order to secure domain functionality and allow Enterprise Admins to perform their designated tasks they need to be added to the necessary permissions list.

The necessary permissions list has been defined:

System Access: SYSTEM, and ENTERPRISE DOMAIN CONTROLLERS

Administrative Access: Enterprise Admins

Client Access: [none]

In order to gain access in to resources, a client needs user groups that have been delegated necessary permissions. These groups should be Domain Local Security Groups, for security purposes, which are created for each client, are located in their respective Organizational Unit, and have been delegated read permission to client's Organizational Unit. This allows client specific access to their own part of the shared infrastructure without compromising other clients.

Further permission delegation can be made for client specific administrative groups. As stated in Hack Proofing Windows 2000 Server Security 'security can be administrated with much more granularity and flexibility. One example is the ability to delegate administrative authority at OU level.' (Todd & Johnson, 2001). Effectively, this means that one can delegate 'modify' or 'full control' permissions for each client into their own Organizational Unit and thus grant them the required permissions to perform user access control operations and administrative operations inside their Organizational Unit.

2.3 Practical Tests

Multiple tests were performed in order to find out if the theory about client invisibility holds true in different common scenarios. When a user tries to access an object without the necessary read permissions the user will receive an access denied or not found error message. In a normal Active Directory

implementation, all authenticated users have read access to all objects in the domain. In the domain described in this thesis, the regular functionality has been heavily modified and multiple tests were performed in order to see whether the regular functionality of Directory Services is working correctly.

Four tests were performed in order to find out if the Organizational Unit structure can be made invisible, five tests were performed to delegate permissions for client self-administration, and three tests were carried out about joining computers to domain.

The testing platform was a brand new Windows Server 2008 R2 with Domain Services installed in Windows 2008 R2 domain level with removed authenticated users permissions as described in this thesis. The following organizational unit directory structure was created OU=1,OU=Clients,DC=testi,DC=turkuamk,DC=local and OU=2,OU=Clients,DC=testi,DC=turkuamk,DC=local. Two client groups were created in to their respective Organizational Units, 1_Users and 2_Users. Read permissions were delegated to client-specific Organizational Units. Two users were created in their respective Organizational Units, user1 and user2 and they were added to their client specific user groups.

In the first test, a blocked Organizational Unit object was accessed with ADUC (Active Directory Users and Computers tool). The test was made by logging in as user1 and opening the client OU with ADUC. The user saw two objects in it, 1 and 2 of which 2 was unknown. Of these objects user1 could not access 2 and could access 1. While object 1 could be opened and its properties be looked at, object 2's properties could not be opened.

The first test revealed a flaw in enumeration because we can see the names of all child objects of an object we have permissions to read even if we do not have permissions to read the child object. As to the date of writing this thesis Microsoft has not released Access Based Enumeration for Directory Objects as they have done for NTFS file systems. This flaw can be circumvented by lacing

the actual client name inside an Organizational Unit without a meaningful name e.g OU=2,OU=1234,OU=Clients,DC=testi,DC=turkuamk,DC=local.

In the second test, the dsquery tool was used to access a blocked Organizational Unit object through Global Catalog. A dsquery was run against client 2's Organizational Unit 2 as user1:

Dsquery ou "OU=2,OU=Clients,DC=testi,DC=turkuamk,DC=local"

The result user1 obtained from this query was "dsqueryfailed:The specific directory service attribute or value does not exist.". From this error message it can be determined that the test was a success. The object was completely inaccessible.

In the third test, a blocked Organizational Unit object was accessed with an LDAP (Lightweight Directory Access Protocol used to access Directory Services) query, with LDP (Lightweight Directory Protocol) tool. The LDP tool was opened as user1 and used to access a blocked. A connection to the local server was made and it was binded to user1's account. A queryobjectclass=* query was made against 'OU=2,OU=Clients,DC=testi,DC=turkuamk,DC=local l' This query lists everything in the specified location. User1 received one result from this query and that was DN: OU=2,OU=Clients,DC=testi,DC=turkuamk,DC=local. When user1 ran a query to directly access group 2_Users with LDP tool, user1 received a message: 'Getting 0 entries:'.

The result from the third test was a success. User1 could not receive any information that was not supposed to be received, meaning the other client's Organizational Unit. The same flaw that the first test revealed was present in this test, too.

In the fourth test, a blocked Organizational Unit security tab was accessed, with ADUC, in order to see the security groups and their members. When OU=2,OU=Clients,DC=testi,DC=turkuamk,DC=local object properties was

opened with ADUC as user1 and security tab was selected the user received an error message stating “not enough permissions to view the permissions tab”.

The fourth test was a success. User1 could not obtain any security information from a blocked object.

By analyzing these four test results, one can determine that it is possible to have two clients in the same Directory without them knowing about each other. This statement is based on the fact that the objects can be made invisible and inaccessible to other objects such as users that have not explicitly been granted read permissions. In addition, one limitation was found: each client Organizational Unit needs to be placed inside another Organizational Unit with a name that has no actual meaning e.g. a number sequence, or else everyone can read the name.

After these four tests to determine object invisibility, delegation tests were conducted for certain administrative permissions in order to test other basic Directory functionality, such as resetting passwords and joining workstations to domain. All tests were compared to commonly known results from these operations, e.g., resetting a password resets the user’s password.

The next series of tests began with ‘User and Group object Administration’ as it is one of the most important parts of every IT infrastructure. User and Group objects are the user’s identity, and with identity the owner of the system can manage authorization. With authorization the owner can manage access to information and without proper authorization management the information is hard to get or even inaccessible.

In theory, the client could want to obtain their identity provider and identity management services from two different providers. In this case, the identity management provider would not have any access to the entire directory but only to their client’s portion. This possibility was tested with Active Directory Delegation Model, and multiple tests were carried out to find out if the schema default permissions modifications broke something.

ADUC (Active Directory Users and Computers tool) delegate permissions ability was used to delegate user and group modification and creation permissions to a client administrative group in to the client specific OU. A test user account was added to this group and user upkeep operations were attempted. A total of five tests were performed.

As the first test, a telephone number was added with the Active Directory command line tool dsmod to every Active Directory user. A new test user was created for client1 called user1_2. The excepted results were that user1 and user1_2 phone number should have changed and every other users phone numbers should be unchanged. ADUC was used to check the results of the dsmod operation.

Using the Active Directory command line tool dsmod requires much care because it is easy to make mistakes that affect the whole domain, possibly overwriting object information. The command consisted of two parts: the query and the modification. The query part was: dsquery user –limit 0. The parameter limit 0 needs to be used or else the query does not find all user objects of the domain but only the first 100 (Microsoft Corporation, 2009). The results of this query were then passed to: dsmod user –tel “01001” –c. The –tel parameter changes the objects telephone number to the designated one (Microsoft Corporation, 2009) and the –c parameter means continue on error (Microsoft Corporation, 2009). The complete command was: dsquery user –limit 0 | dsmod user –tel “01001” –c

Running this test as user1 showed two dsmod succeeded messages and many failed messages. These failed messages were from known system users which could not be modified by user1. When looking at user1’s telephone number with ADUC it showed the number was modified. User1_2 was also modified, but system accounts and other clients user accounts were unmodified.

The test was a success. It was possible to modify a client’s users attributes without accidentally modifying other clients user accounts.

As the second test, user1 was given premissions to reset password in client1's Organizational Unit. User1 reset their own password with ADUC, logged out and back in with the new password. User1 could log in with the new password and was able to reset user1_2's password with ADUC. User1 tried to reset a password outside client1's Organizational Unit, but it was not possible to find other users except Active Directory default users, and their passwords could not be reset.

This test was a success, because a user password could be reset. Passwords outside the client's Organizational Unit could not be reset.

For the third test new groups were created with ADUC as user1. A group was created into the client's own Organizational Unit and another group in the 'clients' Organizational Unit. The group was successfully created in the client's own Organizational Unit but user1 was unable to create a new group anywhere else. The error message was: 'The user does not have the required permissions.'

This test was a success. Clients can create groups inside their Organizational Unit.

As the fourth test, user1 was added to a test group with ADUC. Using ADUC user1 could add theirself to a newly created 'Test1_group' inside the client's own Organizational Unit. User1 then proceeded to find another group, but was unable to find any except the Active Directory default groups. User1 tried to add itself to the domain users group and failed with the error: 'Not enough permissions.'

This test was a success. Clients can add users to groups in their Organizational Unit, as expected.

In the fifth test, a new user object was created in the client1 Organizational Unit with ADUC and another user object in the clients OU.

For this test ADUC was opened as user1 and a third test user user1_3 was created inside the client1's Organizational Unit. The user creation completed

without errors, but user1 was unable to remove the user from the group Domain Users, because the user did not have access rights to the group. All new user objects are automatically members of the domain users group.

A new permission delegation was made with ADUC using a Domain Administrator account and the permission ‘remove member’ was added to Domain Users group for the test client’s administrative group, and the test was repeated. The fifth test was once again performed, but this time successfully. User1 tried to create a new user into clients Organizational Unit but failed with the error message: ‘not enough permissions’.

The test was a success. Although the necessary permissions were not originally delegated for user removal from Domain Users group, the issue was corrected and user1 was successful in creating a new user account.

All administrative tests were successfully implemented with desired results. All user account and group operations could be done inside the client OU, and user accounts or groups could not be modified outside the client OU, with the exception of Domain Users group that was specifically given the permission to remove users from this group, because the groups were either not found or the user did not have enough permissions to modify them.

These tests show that the modifications of Active Directory default permissions were successful and permissions can be delegated to administrative tasks.

Another important basic functionality of Directory Services is joining computers to a domain. This was tested with three tests. In order to perform these tests, the client1 administrative group was delegated permissions to create computer objects into their Organizational Unit and the permissions to designate which user accounts or groups can link a computer with a pre-made computer account into the domain. Certain automated functions were also tested, like joining a computer to domain. The excepted result was that the test would fail if the computer did not have a pre-made computer account, because the user does not have access to the ’_NewObjects’ OU by default.

In the first test, it was tested if user1 can join a workstation to the domain without a pre-made computer account. A laptop was joined to the domain as user1. A computer object was automatically created into the new objects Organizational Unit, but the domain join failed, because the computer object did not have enough permissions to read itself and thus was not found.

The test was a partial success, because domain join failed but the computer object was created. The only way around of this ‘computer object was created in the default computer location’ problem would be to pre create the computer object in to the desired pace as tested in the next test.

For the second test, it was tested if a workstation can be joined to the domain with a pre-made computer account. A computer object was created as user1 using ADUC and the permission to link a computer to the object was given to the client administrative group. Then a laptop was joined to the pre-made object and a ‘successfully joined a domain’ message was received.

This test was a success. It is possible to join workstations to premade domain computer accounts.

The third test was performed to see if automated operations can be made, e.g., form a two server cluster. Forming a cluster automatically creates a new computer account for the cluster name in the default new objects Organizational Unit. Forming a cluster was attempted from two Windows Server 2008 R2 Enterprise Edition servers. The cluster was not formed, because of a time out. The cluster object was created in the new objects Organizational Unit, but the cluster could not find it due to insufficient read permissions. This test was repeated as a Domain Admin user and the cluster formed correctly.

The test was repeated using Windows Server 2012 Standard Edition servers, because these servers have enchanced clustering services. The cluster formed correctly with the credentials of user1. The reason for this was that the cluster virtual name computer account resided in the same Organizational Unit as the cluster node computer accounts which means that the cluster has read access to its virtual object immediately.

This test was a partial fail as the cluster could not form with user1's credentials and the client would need Directory Services providers help for forming clusters. This problem only exists for servers older than Windows Server 2012, as Windows Server 2012 forms cluster virtual names automatically to the same Organizational Unit where the cluster nodes reside.

The tests were successful. It was possible to perform all basic administrative tasks without problems and work around certain other problems, such as forming clusters with the help of a Domain Administrator.

3 ANALYSING RESULTS

In this thesis, a theory was made about object access and visibility blocking in Microsoft Active Directory, while retaining original functionality for the user's own objects. These modifications required extensive testing because of the complexity of access control rules. All basic object operations and basic domain functionality worked as they should, with limitations that can be circumvented. These limitations are first level object enumeration, which means that when an object is selected the child object names can be read even if the user does not have permissions to them, and cluster forming with Windows 2008 R2 servers or older that can not be done without help from a Domain Administrator, which can not be delegated as the virtual server object is created in to the new objects container that is a blocked Organizational Unit. The tests show that all basic functionality was available in an Active Directory with reduced object permissions. Effectively this means that it is possible to make very large Directories with possibly hundreds of clients, all pinned to their own Organizational Units without cross-client visibility.

However, this thesis does not cover all aspects of Microsoft Active Directory. Instead, this thesis provides a basis for future research on the field of Microsoft Active Directory Multi-Tenancy. Important aspects that were not researched include DNS, subnets, and forest trusts.

4 CONCLUSION

Multi-tenant Active Directory is possible. The limitations that were found can be circumvented. These limitations were Access-Based Enumeration and cluster forming. Forming clusters does not work because it is an automated process that places objects inside the default new object container and tries to modify the object, which is prevented. All allowed object operations work as intended even with limited access control lists, meaning that clients can access their own objects and do not have access to, or even visibility of, other clients objects.

In this thesis, a new Active Directory service was created and its schema was modified from each new object's access control list. All unnecessary static access rules were removed. Whenever new objects are created, Active Directory permission delegation grants them inherited access control rules for directory objects residing in client OU's, thus allowing client-specific object access. This boiled down to restricted access to all new parts of Active Directory retaining almost all original functionality and restricting access to all new parts of the domain.

REFERENCES

Kane, J. 2009. Microsoft Official Academic Course Windows Server 2008 Active Directory Configuration. USA: John Wiley & Sons Inc.

Microsoft Corporation 2009. Active Directory tools ‘dsmod user’ and ‘dsquery user’ command internal help.

Sanderson, M. J. & Rice, D. C. 2000. Guide to Securing Microsoft Windows 2000 Active Directory. NSA of USA. Referred 19.11.2013 http://www.nsa.gov/ia/_files/os/win2k/w2k_active_dir.pdf.

Todd, C. & Johnson, N. L. 2001. Hack Proofing Windows 2000 Server Security. Rockland MA USA: Syngress Publishing.

Appendix 1: Forest Creation Answer File

The following answer file was used to create the test forest:

[DCInstall]

InstallDNS=Yes

NewDomain=Forest

NewDomainDNSName=test.local

DomainNetBiosName=test

ReplicaOrNewDomain=domain

ForestLevel=4

DomainLevel=4

DatabasePath="D:\NTDS"

LogPath="D:\Log"

SYSVOLPath="D:\SYSVol"

ConfirmGc=Yes

RebootOnSuccess=Yes