CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Theseus

NETWORK ACCESS CONTROL, SINGLE COMPUTER VIEWPOINT

Jari Kuisti

Masters' thesis

May 2009

School of Technology

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JAMK UNIVERSITY OF APPLIED SCIENCES

DESCRIPTION
Date
28.5.2009

Author(s)
KUISTI, Jari

Type of Publication
Master´s thesis

Pages
81

Language
English

Confidential
☐ Until_____

Title
NETWORK ACCESS CONTROL, SINGLE COMPUTER VIEWPOINT

Degree Programme
Master's degree programme in Information Technology

Tutor(s)
HAUTAMÄKI, Jari
HUOTARI, Jouni

Assigned by
LINDROOS, Jari

Abstract
The purpose of the thesis was to develop an entirely new ideology and technique which is called a **client's NAC** (Client's Network Access Control). The objectives of the thesis were to discover methods how a single computer could make conclusions about the connected network and validate if the network is trusted or not. This is an entirely new ideology, which has not been published on the commercial markets or in academic research.

In a nutshell, the philosophy of the Network Access Control is that all devices requesting access to network's resources are untrusted until they are otherwise proved. The objective was to discover if it is possible to conduct same kind of philosophy to a single computer. A computer does not trust the network before it has done specific validations from the network and depending on the outcome of the validations; network traffic to network is allowed or denied.

The discovery in the thesis was that almost every LAN protocol has different kinds of security issues. Usually these threats are blocked in the network's outer perimeter with firewalls in such a way that the outside of the network cannot exploit these threats. This does not prevent from exploiting these security threats from inside the network. These findings supported the idea of client's NAC implementation, because if the network is trusted, the devices in the network are also trusted. The goal was to develop methods and techniques how a single computer could execute the conclusion about the connected network. This included developing the basic architecture of the client's NAC solution and discovering different authentication methods for authenticating the network. These authentication methods were analyzed with security and implementation analysis and based on these analyzes the thesis recommends certain authentication methods for client to authenticate the connected network.

Keywords
Client's NAC, untrusted network, Network Access Control, information security

Miscellaneous

JAMK UNIVERSITY OF APPLIED SCIENCES

DESCRIPTION
Date
28.5.2009

Tekijä(t)
KUISTI, Jari

Julkaisun laji
Opinnäytetyö

Sivumäärä
81

Julkaisun kieli
Englanti

Luottamuksellisuus
☐ Salainen _____saakka

Työn nimi
NETWORK ACCESS CONTROL, SINGLE COMPUTER VIEWPOINT

Koulutusohjelma
Master's degree programme in Information Technology

Työn ohjaaja(t)
HAUTAMÄKI, Jari
HUOTARI, Jouni

Toimeksiantaja(t)
LINDROOS, Jari

Tiivistelmä
Työn tavoitteena oli kehittää uutta ideologiaa ja tekniikoita (client's NAC), jossa perinteinen verkkolähtöinen näkökulma pääsynhallinnassa suunnataan yksittäiselle tietokoneelle. Tämän kaltaista tutkimusta tai konseptia ei ollut olemassa, joten kyseessä oli aivan uusi tutkimuksen aihe. Kehittämisessä lähtökohtana oli löytää malli, jonka mukaan yksittäinen tietokone pystyy päättelemään, onko verkko, johon se on kytketty, luotettu vai ei. Työssä sovellettiin ja analysointiin eri autentikointivaihtoehtoja, joiden perusteella esitettiin tiettyjä autentikointitekniikoita client's NAC -sovelluksen toteuttamiseen.

Työ osoitti, että yleisimmissä LAN-protokollissa on merkittäviä uhkia ja haavoittuvuuksia. Jos yksittäinen tietokone kykenee päättelemään verkon luottavuuden, näiden uhkien toteutumista voidaan lieventää, sillä luotettava verkko sisältää vain luotettuja laitteita. Tämä vahvisti, että client's NAC -konseptin avulla voidaan suojautua epäluotettavien laitteiden haitalliselta tietoliikenteeltä.

Eri autentikointimallit jaettiin työssä kahteen eri kategoriaan tulevan kohdeympäristön perusteella. Korkean tietoturvallisuuden ympäristöissä tietoturva ja osapuolten luottavuus on tärkein tekijä suunniteltaessa autentikointimalleja, kun taas matalamman tietoturvaluokan ympäristöihin toteutuksen helppous ja käytettävyys ratkaisee valinnassa.

Analysointi eri autentikointimallien välillä suoritettiin tietoturva-analyysillä, joka perustui tietoturvaprotokollissa oleviin yleisimpiin haavoittuvuuksiin, ja toteutusanalyysillä, jossa pyrittiin tekemään päätelmiä toteutuksen toimivuudesta ja vaikeudesta. Näiden analyysien perusteella työ esittää eri vaihtoehtoja eri ympäristöihin toteutettavaksi autentikointitavaksi client's NAC -sovellukseen.

Avainsanat (asiasanat)
pääsyn hallinta, autentikointi, tietoturva, Network Access Control, Client's NAC

Muut tiedot

CONTENTS

FIGURES

TABLES

1 INTRODUCTION

The complexity of networks has increased and also vulnerabilities in IP networks have become more hazardous. For example, a case on the vulnerability in Internet Domain Name System service (DNS cache poisoning) was published in summer 2008. This was a very serious case of vulnerability which could deceive a user to a wrong URL with keying forged information to client's resolver's cache. Fortunately, DNS cache poisoning got plenty of publicity in information security forums; therefore software were rapidly updated to a more secure version. The exploitation of this vulnerability was not extensive. Another significant network related vulnerability was recently (September 2008) noticed in TCP (Transmission Control Protocol) protocol stack. This vulnerability is based on fulfilling the destination's TCP connection queue with hostile packets. These actions cause DOS (Denial of Service) attack in the destination's network. So what makes this TCP vulnerability so hazardous? Vulnerability is a feature on TCP implementation and the utilization of TCP protocol is enormous. Almost every solution uses TCP protocol for creating and maintaining connections in IP networks.

Vulnerabilities would not be so big problem if there were not anyone who exploits these security holes. In the beginning of the Internet the threats came from random script kiddies and hackers. They were eager to prove their abilities to each other and show that they could get away with it. Today, highly motivated and well-financed organized crime is taking advantage of vulnerabilities and security holes in computer systems in the Internet. Unfortunately, vulnerabilities are not their only target. For example phishing and social engineering have become more common in these days in the Internet.

The Internet does not obey country boundaries; it spans and connects the globe. Different countries have different laws. There are no common laws or authority which could actually protect anyone or anything in the Internet; therefore it is impossible to root out organized crime. Well-designed, automated, well-funded remote attacks are constantly being launched for getting financial benefits and for the remote control computers creating botnets, which are used to infect other computers and launching distributed denial-of-service attacks.

Many organizations have lost control of their own assets, and may not have knowledge what is really happening in their networks. Organizations can not just wait for news of the new hazardous vulnerability or attack. Data security risks and threats have to be identified and managed. IT management strategy, governance, risk management and compliance are corner stones to get control of IT infrastructure.

Network Access Control (NAC) is a concept which focuses on GRC (Governance, Risk management, Compliance). As the name indicates, it concentrates on who or what is gaining access to network. Because NAC is a new technology, people have a different impression of the definition. Some might think that IEEE 802.1X port based authentication is a NAC or certain software product is NAC.

Network access control is a framework, which involves different products; NAC consists of a cluster of new and present technologies which control what device or who can get access to a protected network. NAC can restrict network access based on user authentication, device authentication and device attributes which are defined on certain policy, for example these attributes can be antivirus signature level and operating system patch level. If these attributes are not equal to policy, the device is guided to quarantine. (Reinhold, 8, 9, 13.)

NAC handles protection from the network's point of view, which means that the network does not trust devices which are connected to it, until certain validations are done and passed. How about it if your organization has several different networks which are in different security levels? Some networks can be organization's maintaining responsibility, and maintenance of some of those networks may be outsourced to a third party.

What happens if NAC is not used in every each of those networks, and somebody connects the highest security level PC to an untrusted network? What kind of data does the computer send to that network? Is there a process or a product how single computer could validate is the network trusted or not?

Network Access Control is a concept which contains different kinds of technologies and policies. The main idea is that these technologies are in the trusted network and with these technologies and policies the network validates the devices trying to access its recourses. If these devices correspond to the current policies, access is granted. In a nutshell, the philosophy of the Network Access Control is that all devices requesting

accesses to network's resources are untrusted until are otherwise proved. Could we conduct the same kind of philosophy to a single computer? A computer does not trust the network before it has done specific validations from the network. When these validations are done, depending on the outcome of the validations, network traffic to a network is allowed or denied. Information security is basically based on trust. In this case trust is the ability for a computer to be reasonably assured that a particular network has known devices and these devices meet the security requirements that the organization has agreed on. It is always safe to start a network connection from a state in which both participants do not trust each other. There are no products on the commercial markets which have certain functionality to validate networks and are based on client's mistrust. In this thesis this product is called ***client's NAC***. The actual NAC framework has been developed for different purposes, but together these solutions (NAC and the client's NAC) complement each other. Client's NAC brings one security layer more to overall security. It completes other client security products, such as personal firewall, antivirus software and most important; it completes Network Access Control concept.

Purpose of this research is to investigate is it possible to mitigate the security threats which arise when the workstation is connected to an untrusted network. Different types of network traffic are examined and the security threats are identified. The goal is to find most common local area network protocols (Windows network) and applications that generate the traffic that should be secured from untrusted devices. Security threats of these protocols are investigated and presented. In this research the objective is discover new authentication methods which are conducted from common network protocols or security protocols and introduce how these protocols could be applied in client's NAC solution.

This study presents research work concentrating in:

- Discovering Windows client's common LAN traffic/protocols and security threats of these protocols
- Techniques, methods and ideas how to control the client's joining to the network
- Comparison of different ideas of developing client's NAC authentication methods (security and implementation analysis)

- Choosing the best techniques and components how to develop client's NAC authentication and introduction of the basic architecture of a client's NAC

2 NETWORK ACCESS CONTROL

2.1 History of the term

The term NAC was launched by the Cisco Systems in year 2003. It was a part of self-defense aware networks marketing architecture. At that time, the meaning of the NAC-term was Network Admission Control. The main idea was to grant access for network equipment to network after their health had been inspected.

An accelerator for developing NAC was malware such as Blaster and SQL slammer which could independently spread in IP-networks. They were caused unpleasant problems all over the world, including many Finnish organizations. For example in October 2003, Nordea had to close 80 customer service offices because Blaster based Lovsan virus spread in their IP-networks. (Työasemat tarkastukseen 2008, 52.)

2.2 Network protection in the old days

Organizations usually protect their network's perimeter with proper firewalls and email malware scanners. This is a standard way to protect network from outside threats, but this procedure does not take into consideration inside threats such as infected laptops or other network equipments which are brought inside the network. For instance, laptop can spread malware's for hours in network before the source of the spreader has been discovered. Even though if information security programs were centrally updated, the time between logon to network and when client (workstation) update itself is creating a vulnerability window. Usually this time can be hours or even days.

Workstation health has many dimensions. Important markers are (Reinhold, 6-7.):

- Absence of malware
- Updated malware prevention tools
- Patch management
- Specific firewall settings

- Corporate security policy –based configurations

2.3 NAC fundamentals

NAC works by checking workstations' utility programs and security software that they are up to date and equal to organization's information security policy. If these components meet the information security policy and are also updated, the workstation is granted normal access to the network. Otherwise workstation is connected to separate segment in the network which is called a quarantine segment. In this quarantine segment, the workstation is updated to a proper level. After the updating process, the workstation is granted normal access to the network.

After Cisco Systems launched their Network Admission Control, the concept was so good so that many rivals came to market with their own thinner and less device manufacturer dependent systems. The NAC-term got its new meaning - Network Access Control. Heated competition in the markets produced incompatible solutions. This situation did was not win customer's confidence. NAC stayed years as a big promise, the development of which potential customers were watching from distance.

TCG group (Trusted Computing Group) which is group of a computer and network manufacturers, has made standardization work for NAC. This group includes manufactures such as Juniper Networks, Microsoft and Extreme Networks. Juniper Networks has an imposing role of NAC standardization.

TCG's vision of NAC is TNC – Trusted Network Connect, which has a full set of open standards for network access control. The main goal is to develop a set of open standards so that different manufacturers' workstation, network and server components can interoperate. (Työasemat tarkastukseen 2008, 53.)

2.4 Server and domain isolation

Microsoft has implemented their own techniques to achieve a logical security layer to the network. Microsoft calls these techniques "Server and domain isolation". The basic idea of Server and domain isolation is that there is no need for certain equipment architecture such as e.g. in 802.1X. Server and domain isolation uses Windows computers, servers, Active Directory group policies and IPSec technologies to create a layer of security to achieve logical isolation of the network traffic that moves between computers or networks. As most of the companies already have Windows environment (total market share of Windows computers 15.5.2009 is 87.9%,

Operating System Market Share 2009), the deployment of these techniques does not necessarily need further investments.

Server and domain isolation mitigates the risk of internal attackers. Even though an attacker has a physical access to an organization's internal network and uses a valid user account/password, an attacker cannot get access to servers. This is result from trusted device ideology. (Server and Domain Isolation Using IPSec and Group Policy 2006, 1.)

2.5 Security threats which NAC is not taking care of

NAC is a concept for protecting the networks from unwanted access, but there are some threats that NAC does not handle at all. This thesis concentrates on ways to mitigate these threats.

If a computer is accidentally or intentionally connected to a wrong network, computer sends information about itself to that network. In many cases this information is an IP-address, operating system version, computer name, and domain name etc. The information depends on computer's operating system. With this information it is possible to make conclusions about the computer's network infrastructure such as IP-address spaces, name services, domain controllers and operating system vulnerabilities. This kind of information can be used for hostile purposes.

What if a laptop is left in a conference room because of a lunch break and the username is locked? Someone could connect to the laptop's network interface eavesdropping equipment and could capture information from the laptop.

These examples are security threats which have to be prevented; therefore we have to implement a new solution to do this because there are no commercial solutions on the markets.

2.6 Discovering the client's outbound network traffic

Computer sends different kinds of data to a network when it is connected. The data can be basically divided into two categories: There is data which is automatically sent towards network and data which is sent after certain user actions.

The sent data depends on the operating system. This thesis concentrates on Microsoft Windows XP operating system. When Windows XP workstation is connected to a network it sends information about itself to network. The sent information depends on

the computer's network settings. For example, there can be a static IP-address or DHCP setting, NBT (NetBIOS over TCP/IP) setting and the computer can be attached to Windows domain or workgroup. The data can vary depending on network settings.

In next two chapters (3 and 4) the most interesting network protocols from security and authentication perspective are presented. A computer uses many network protocols when it sends traffic towards network, but the protocols for further examination have been chosen based on their commonness in Windows networks. In addition, the thesis examined what security threats these protocols have and how these threats can be addressed or mitigated.

The method that was to discover the interesting network traffic was first to capture computer's real traffic and analyze it, followed by orientation with literature about these protocols. The traffic was analyzed from authentication and security perspective and the objective was to find how these protocols can be utilized to authenticate the trusted network. The traffic tests were done in a test environment which had three Windows XP workstations, two Windows 2003 servers, a router and a LAN switch.

3 CLIENT'S NETBIOS OVER TCP/IP TRAFFIC

3.1 NetBIOS and NBT

NetBIOS (Network Basic Input/Output System) was developed in 1983 for IBM PC-networking. The design objectives were to build a small and fast protocol that would allow human readable names for devices, such as "OfficeComputer". It is easier to remember names than a complex numbering scheme. NetBIOS is not a protocol; it is an API (Application programming interface) for PCs to access LAN facilities. In that time there were limitations in the size of networks, size of network was no more than 72 devices on broadband access.

Because NetBIOS is an interface rather than a protocol, it requires a network protocol to carry its sessions across a network. At first network protocol was NetBEUI (NetBIOS Extended User Interface). This protocol operates over LAN using OSI layer 2, therefore it is not routed protocol. Because of this limitation and its nature to use broadcast traffic, there was a need to discover new methods to use NetBIOS in networks. (Haden 1996.)

NetBIOS over TCP/IP also called NBT (or NetBT) was published in RFC 1001 in March 1987. RFC defined NetBIOS encapsulation method in TCP and UDP packets. In other words NBT is an implementation of the NetBIOS API on top of TCP/IP. NBT provides three services:

- NetBIOS name service
- Datagram service
- Session service.

The Name Service handles NetBIOS names and is used to do name resolution; name service is at UDP port 137. The Datagram Services and Session Services are mainly used for protocols based on Server Message Block (SMB). These two communication services are used to transmit data between NetBIOS computers across the network. (Ts J, Eckstein R, & Collier-Brown D 2003, 10.) Datagram service is at UDP port 138 and Session service is at TCP port 139 (Zwicky E. D, Cooper S, Chapman B D 2000, 359).

3.2 NetBIOS name service

16 bytes long NetBIOS name identifies a computer or group of computers in the network. There are two types of NetBIOS names; name is either a unique (exclusive) or group (non-exclusive). NetBIOS applications typically use unique names when communicating with a specific process on a computer, and group names are used to communicating with multiple computers at a time. (TCP/IP Fundamentals for Microsoft Windows 2005.) Only one node can have certain unique name, but any number of nodes can have a group name. The name has to be owned by at least one node otherwise it ceases to exist. (RFC 1001 1987.)

An example of a service that uses a NetBIOS name is “The File and Printer Sharing” over Microsoft Networks component (also called Server service). For a start the Server service registers a unique NetBIOS name based on the computer name. The NetBIOS name is the 15-byte computer name plus a sixteenth byte of 0x20. Because each character is 8 bits long, it means that NetBIOS name can be only 15 characters. The last character is reserved as a special character. Windows appends spaces to computer name if it is shorter than 15 bytes long. On the contrary, the names longer than 15 bytes are truncated. The sixteenth byte of the NetBIOS name typically

identifies a specific service. The client for Microsoft Networks component (also called Workstation service) and the Messenger service uses NetBIOS. The Workstation service has sixteenth byte of 0x00 and the Messenger service has a sixteenth byte of 0x03. Figure 1 shows how NetBIOS names differ between services. (TCP/IP Fundamentals for Microsoft Windows 2005.)

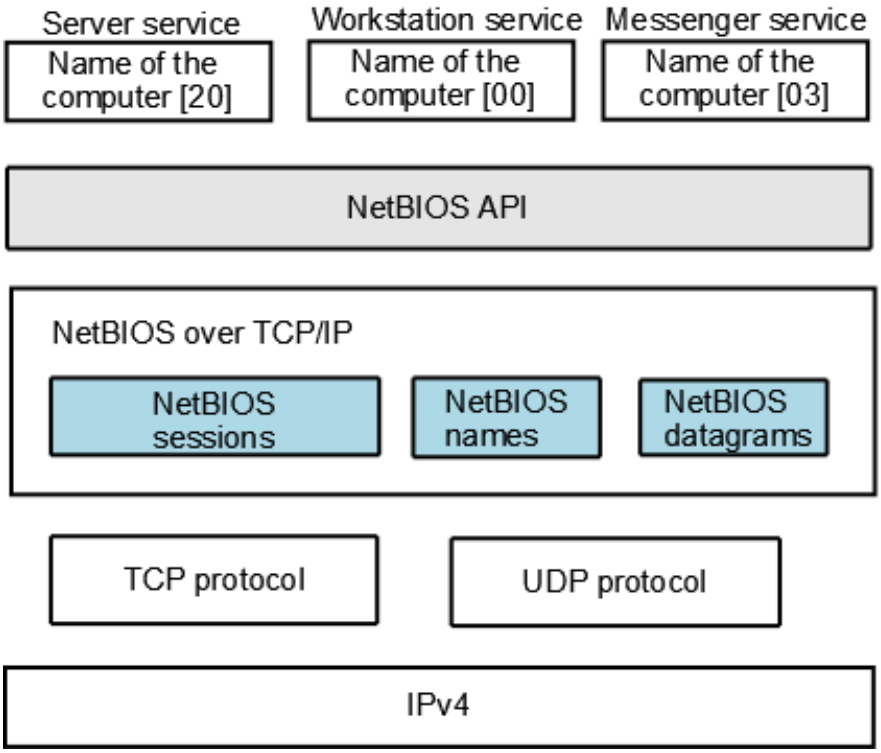


FIGURE 1. NBT and NetBIOS applications (TCP/IP Fundamentals for Microsoft Windows 2005)

When user in Windows network connects to a shared folder with a "**net use"** command or with Windows Explorer, NBT resolves the NetBIOS name for the Server service of the specified computer. After resolving the IPv4 address of the NetBIOS name, the Workstation service on the client computer can start communication with the Server service on the destination computer.

Services such as the Computer Browser, Distributed File System, and Net Logon services are dependent on Server, Workstation, and Messenger services. This means that these services also register NetBIOS names. Accessing to these services Windows network applications must use their corresponding NetBIOS names. One example is Computer Browser service which collects and distributes the list of workgroups and domains. These lists are constructed of NetBIOS names. (TCP/IP Fundamentals for Microsoft Windows 2005.)

3.3 NetBIOS end-nodes

NetBIOS implementation has three types of end-nodes. These nodes support NetBIOS interface and have applications which use NetBIOS API. Three types of end-nodes are part of NetBIOS standard (RFC 1001 1987.):

- Broadcast ("B") nodes
- Point-to-point ("P") nodes
- Mixed mode ("M") nodes

Windows Server 2003 and Windows XP support five NetBIOS node types: the node types defined in RFC 1001 and also Windows specific node types. Each node type resolves NetBIOS names differently. (TCP/IP Fundamentals for Microsoft Windows 2005.)

Broadcast node (B-node) uses broadcasts for name registration and resolution. Broadcasts are flooded inside broadcast domain, which means that NetBIOS resources that are located on different IP subnets cannot be resolved. (TCP/IP Fundamentals for Microsoft Windows 2005.) In other words, the broadcast area is a single MAC-bridged “B-LAN”. Broadcast nodes use a mix of UDP datagrams (both broadcast and directed) and TCP connections to communicate with each other. (RFC 1001 1987.)

Point-to-point nodes (P) use an NBNS (NetBIOS name server) such as WINS to resolve NetBIOS names. P-node queries the NBNS using only directed UDP datagrams and TCP sessions, this means that P-nodes can resolve NetBIOS resources located on remote subnets. P nodes cannot generate or listen for broadcast UDP packets. Therefore NetBIOS name resolution fails for all NetBIOS names if the NBNS becomes unavailable, even for NetBIOS applications that are in the same subnet. (TCP/IP Fundamentals for Microsoft Windows 2005.) However, using capabilities provided by the NBNS and NetBIOS Datagram Distribution Server (NBDD) P-nodes can use NetBIOS level broadcast and multicast services, in this case NBNS and NBDD have to be available. An end-node can query an NBDD to determine if the NBDD is willing to relay a datagram to a specific NetBIOS name. P nodes lean on NetBIOS name and datagram distribution servers. The servers’ being

local or remote does not affect the functionality of P-node; they operate the same in either case.

Mixed mode nodes (or "M") are a combination of B-node and P-node. M nodes use both broadcast and unicast. To improve response time, broadcast is a default method because of the assumption that most resources reside on the local broadcast medium rather than somewhere in an internet. If the broadcast name query fails, NBT uses an NBNS. M-nodes are also dependent on NBNS and NBDD servers. If servers become unavailable, M-nodes can continue with limited functionality. (RFC 1001 1987.)

There are also two Windows specific node types which are not described in RFC 1001. These node types are:

- **H-node** (hybrid) is a combination of P-node and B-node. This node type functions as a P-node by default. If the unicast name query fails to the NBNS, the node uses a broadcast.
- **Microsoft enhanced B-node** is a combination of B-node and the use of the local Lmhosts file. If the broadcast name query fails, the node checks the local Lmhosts file.

3.4 NetBIOS name registration, resolution, and release

NBT network resources use processes for name registration, name resolution, and name release to manage NetBIOS names.

3.4.1 Name registration

NBT hosts registers its NetBIOS names using a NetBIOS Name Registration Request message (name claim). It also maintains name information which it has been registered. This information includes:

- Whether the name is a group or unique name
- Whether the name is "in conflict"
- Whether the name is in the process of being deleted

Registration can be made via broadcast message to local subnet or a unicast message to a NetBIOS name server (NBNS). In a name conflict situation either the host that previously registered the name or the NBNS responds with a negative name

registration response. (TCP/IP Fundamentals for Microsoft Windows 2005.) Every node has a permanent unique name, which must be explicitly registered by all end-node types (RFC 1001 1987).

3.4.2 Name resolution

A NetBIOS name is a Session layer application identifier. The process of mapping a NetBIOS name to an IPv4 address is known as NetBIOS name resolution or name query. Name query is needed for example during session establishment where source and destination names must be specified. The name can be a unique or group name. If the destination name is a group name, a datagram is sent to all the members of that group. (RFC 1001 1987.)

In Windows networks NetBIOS name resolution functions same way that is described in RFC 1001. NetBIOS application running in Windows XP or Windows 2003 server broadcasts a NetBIOS Name Query Request message to local subnet or uses a direct query to NBNS. NetBIOS Name Query Request message contains the NetBIOS name of the destination host.

In NetBIOS name conflicting situation the neighboring host that has registered the same NetBIOS name or an NBNS responds by sending a negative NetBIOS Name Query Response message. (TCP/IP Fundamentals for Microsoft Windows 2005.)

3.4.3 Name release

When NetBIOS application is stopped name release occurs. B-nodes release a name by broadcasting a notice to local subnet. P-nodes send a direct notification to their NBNS. M-nodes both broadcast a notice and send notification to their NBNS. (RFC 1001 1987.) For example, if Workstation service on a Windows host is stopped, the host requests that the NBNS no longer respond to queries for the Workstation service name (TCP/IP Fundamentals for Microsoft Windows 2005). This is an explicit release. NetBIOS name release can also be silent. When end-node fails or is turned off this release typically occurs. (RFC 1001 1987.) After NetBIOS name is released, it is available for use by another host (TCP/IP Fundamentals for Microsoft Windows 2005).

3.5 NetBIOS Scope ID

The NetBIOS scope ID segments NetBIOS Names. “Scope” isolates a set of NBT nodes and ID is “tagged” to NetBIOS name as character string. If the ID on two hosts does not match, they will not be able to communicate to each other with NBT. Scope ID is a part of the full NetBIOS name. (TCP/IP Fundamentals for Microsoft Windows 2005.) By default scope ID is an empty string. This ID can be modified from Windows registry.

RFC 1001 recommends that both B and M nodes should not be used within the same scope. The scope should contain only P and M nodes or B nodes. (RFC 1001 1987.)

3.6 Security threats in NBT

NetBIOS over TCP/IP ports are UDP 137, UDP 138, and TCP 139. Unfortunately NBT provides poor security. Hosts can initiate connections to each other with NetBIOS names. NBT application could do authorization based on sender’s NetBIOS name and IP address but in practice this is very rare. Even though there is an authorization, IP addresses and NetBIOS names are easy to spoof. Attackers use port scanners or passive sensors to sniff NBT ports because of their poor security. NBT offers diagnostics tools such as *nbtstat*, this tool is very handful to an attacker, who can use the tool to begin footprinting. Once an attacker discovers an active port 139 on a device, with the *nbtstat* command, he can obtain significant information about the target computer and network. That information includes (Olzak 2007.):

- Computer name
- Contents of the remote name cache, including IP addresses
- A list of local NetBIOS names
- A list of names resolved by broadcast or via WINS
- Contents of the session table with the destination IP addresses

Mostly the attacker searches information about the OS, services, shares, user IDs and major applications running on the system. With *nbtstat* command this information is visible for a cracker. For example, IPC$ share was a few years ago common target to crackers. The following chapter discusses this share and exploits it in more detail.

3.6.1 The IPC$ share

IPC (Inter-Process Communication) share is simply a regular share with an interesting behavior added to it. On Windows NT architecture, any shared directory can be made into a null-session share. Null-session means that other computers to access the share without user-based connection data (username, password). (Configuring Null-Session Shares 1999.)

3.6.2 How crackers abused null shares?

First, they had to find out with a port scanner which computer has a TCP 139 port open. When this was done they started to footprinting the computer if the IPC$ share was available.

When successful footprinting is done, there are several Windows command line commands such as NET commands which can be used to find and map shares on remote computers. To initiate a null session to remote computers IPC share can be made with following NET command: C:\>NET USE \\TARGET\IPC$ "" /USER:"". Cracker connects to IPC share on the specified target with the password “” and the user name “”. The next step in the cracking process was to try to get access to the Windows NT hidden shares. The default hidden shares are: C$, PRINT$, ADMIN$, IPC$. The “$” sign makes these shares invisible to the average users. If a cracker gets access to remote computer’s C$ share with administrators privileges, he or she can have full control of the computer. (IPC share exploit.)

These IPC null session attacks were easy to implement because attacker did not need to have programming skills, only command line usage was needed. Most of the cases cracker needed an administrator password before he or she got access to C$ share. This only slightly delayed the attack because getting the administrator’s password from remote computer was also quite easy. On the Internet there are several software and methods such as brute force to crack a Windows administrator’s password. This null session attack is just one example of the vulnerability history of NBT. Nowadays, null session attacks are rare, because this vulnerability is well known and there are many ways to prevent null sessions. Information security threats were different when NetBIOS and NBT were implemented than nowadays. There are different methods to prevent security threats that NBT brings but the fact is that NBT is still unsecure, because security aspects were not taken consideration in the design phase of the NetBIOS.

3.7 Disabling NetBIOS

NetBIOS offers an easy way to browse network resources and share directories or files in a computer network. It has different services such as name service which for example mitigates browsing and sharing. NetBT is a very common implementation in Windows Networks, because Microsoft adopted NetBIOS for their products in the 1980s and many corporate networks can still have legacy Windows operating systems (Windows 9x, Windows NT) or other applications which need the NetBIOS API to run. That is why NBT is turned on by default in newer operating systems such as Windows 2000, Windows XP, Windows Server 2003 and Windows Vista. (Tulloch 2004.)

During the last ten years there have been many exploits in NBT. The Redbutton attack was one of the common exploits which utilized the NetBIOS session service. This technique was based on the TCP/IP connection to port 139. With this connection, a cracker could establish a null session share. The null session share made it possible to access the Windows NT hidden shares and gain administrator privileges on a target computer. (How is information enumerated through NULL session access, Remote Procedure Calls and IPC$? 1999.) It is generally noticed among the IT professionals that NBT processes are unsecure.

For better security, Microsoft had implemented DNS to be the default name resolution method for Windows 2000 and for newer operating systems. This means that on pure Windows-2000/XP/2003 networks NetBIOS is no needed. Because of the NBT's unsecure nature, what is the point of leaving NetBIOS enabled on this kind of network?

Active Directory installation to Windows 2000 server or Windows 2003 server automatically suggests and accepts NetBIOS name for Windows domain. This cannot be overridden. Even though DNS is default name resolution, Windows 2000/2003 server is forced to use NetBIOS name for the domain. The persistence of NetBIOS names is not just for the need of legacy; Windows NT domains participates fully in Windows 2000/2003 forests. NetBIOS support is required for establishing Windows domain trusts, even though there is only pure Windows 2000/2003 domain. Active Directory forests must have unique NetBIOS names in order for name resolution to work properly and support access to resources across forest boundaries. (Tulloch 2004.)

This means that NBT cannot be fully switched off from pure Windows 2000/2003 networks. Because NetBIOS has so deep integration to Windows networks and most Windows computers need it, the unsecure NBT traffic has to be handled in another way. If NBT cannot be switched off from computer's network settings, we must be sure that the network is trusted where computer sends its NBT traffic. The ***Client's NAC*** is addressing this problem.

4 DHCP AND DNS TRAFFIC AND THEIR SECURITY ISSUES

4.1 DHCP traffic

Dynamic Host Configuration Protocol (DHCP) is a common service in IP networks. DHCP is a network application protocol used by devices (DHCP clients) to obtain network configuration parameters that they need in order to operate. These parameters include e.g: IP-address, subnet mask, default-gateway, DNS server IP-address and possible other server addresses. This protocol allows devices to be connected to the network with little or no manual configuration. DHCP is based on a client-server model, where DHCP servers allocate network addresses and deliver configuration parameters to clients who are configured to use DHCP. (RFC 2131 1997.)

DHCP is defined in RFC 2131 which was released in 1997. This definition is the current DHCP definition for IPv4 networks. The extensions of DHCP for IPv6 (DHCPv6) were published as RFC 3315. DHCP consists of two components:

- Delivering protocol for configuration parameters to client (network-aware device) from a DHCP server.
- A mechanism for allocation of network settings to clients.

DHCP has three network settings allocation mechanisms:

1. In automatic allocation server assigns a permanent IP address to a client. Server has a table of past IP address assignments, so that the server can assign to a client the same network settings that the client previously had.
2. In dynamic allocation, server assigns an IP address to a client which has a limited lease period.

3. In manual allocation, IP address to a client is assigned by the network administrators, who are manually keying MAC address and IP address pairs to DHCP server. The client whose MAC address is listed in the table will be allocated corresponding IP address of the table.

DHCP uses UDP as its transport protocol. DHCP server listens for UDP port 67 and client listens for UDP port 68. The "server identifier" field in the DHCP message is used to identify a DHCP server and value of the field directs client's unicast traffic to the right server.

Successful DHCP conversation is divided into four basic messages. These messages are DHCPDISCOVERY, DHCPOFFER, DHCPREQUEST, and DHCPACK (IP lease acknowledgement). When the client has received the network settings (IP address, subnet mask, default gateway and DNS etc), it uses an address resolution protocol (ARP) query to prevent IP conflicts in case of address pool overlapping of DHCP servers. Below is a detailed description about the basic DHCP messages used in DHCP conversation:

The client starts the conversation by sending **DHCPDISCOVERY** message to destination of 255.255.255.255 or subnet broadcast address. Because this message is a broadcast message, it is not routed. Therefore if DHCP server is in another subnet, each subnet where the server is not located needs a BOOTP relay agent (also called DHCP relay agent) which forwards discover messages to the server. For instance, this relay agent can be a local router which is configured to forward DHCP packets to a DHCP server on a different subnet. (RFC 2131 1997.) A client can request its preceding IP address and lease duration from server by including in DHCP discover message its former IP address. These attributes are placed in DHCP options field in DHCP message. If the network is corresponding with the parameters that the client offered, the server might grant the request. (RFC 1533 1993.)

After the client has sent the DHCPDISCOVERY request to a server, the server reserves an IP address for client by sending a **DHCPOFFER** message. The message includes offered network address in the "yiaddr" field, client's MAC address in the "chaddr" field. Other configuration parameters such as: the subnet mask, the lease duration, DNS server addresses and NetBIOS settings etc. are located in DHCP options field.

The DHCPOFFER message can come via one or several servers to the client. This means that the client can receive multiple offers. Based on the configuration parameters offered in the DHCPOFFER message, the client chooses one server where to request configuration parameters. When the client has chosen the server, it broadcasts a **DHCPREQUEST** message which contains the "server identifier" in options field to indicate the chosen server. The "requested IP address" option value is same as the value of "yiaddr" field in the DHCPOFFER message from the server. Because the request message is broadcast, it is forwarded same way as the discovery message; through DHCP/BOOTP relay agents. The value of the "secs" field in the original discover message is copied to request message's "secs" field. This field and the same broadcast address that was in the original discover messages are the factors which direct request messages to the same set of servers that received the original DHCPDISCOVER message.

The servers withdraw offers and return the offered address to the pool of available addresses if they do not match the value in the "server identifier" field of the DHCPREQUEST message. The selected server reserves from its address pool the IP address and respond with **DHCPACK** message which includes the network parameters for the client. The combination of the "client identifier" or "chaddr" (these fields includes host's MAC address value) fields and the assigned IP address create unique identifier for the client's lease. With this identifier client and server identify a lease referred to in any DHCP messages. DHCPACK message's "yiaddr"' field is filled in with the selected IP address. (RFC 2131 1997.)

The four DHCP messages that were described above are the main messages when a client receives network parameters from the server successfully. There are also other messages which are briefly described as follows. The messages are:

DHCPNAK is message from a server to a client indicating that a network address is incorrect or the client's lease has expired. The selected server sends to client DHCPNAK message after it has received DHCPREQUEST message and is unable to satisfy the client demands.

The client indicates the server with **DHCPDECLINE** message that the network address is already in use. After the client has sent the message it restarts the configuration process. The client waits at least ten seconds before it restarts the configuration process to avoid massive network traffic in case of looping.

The client sends to server **DHCPRELEASE** message when it wants to abandon its network address and cancel the remaining lease. The released lease is identified in this message with its "client identifier" field or in "chaddr" field and network address. If the client wants to retain its network address, it will not abandon its lease during the shutdown or reboot.

If a client has manually configured network address, it can use a **DHCPINFORM** request message (clients IP-address is in messages "ciaddr" field) to obtain other configuration parameters. After the servers have received the message they send a DHCPACK message with configuration parameters. The server does not allocate a new address, it is not checking for an existing binding and it does not fill the "yiaddr" field or adds the lease time parameters. The server checks the network address in a DHCPINFORM message for consistency. (RFC 2131 1997.)

4.1.1 Security threats in DHCP

DHCP is based on BOOTP which was implemented in 1980s when the Internet was in its early stage. The Internet was only a small group of research and educational organizations using TCP/IP networks interconnecting universities and other research organizations. Because of the small amount of users there were few security threats and therefore security was not baseline for the designing of new protocols. As a result, when designing DHCP and many other protocols in 1980s and in the early 1990s, designers did not take security aspects into consideration.

DCHP traffic is unencrypted and the authentication methods are based on IP-addresses and MAC-addresses which are easy to forge. DHCP has poor security and it is not an improving fact that DHCP runs over IP and UDP which are inherently insecure. In modern networks and the Internet this causes potential security issues. DHCP messages contain information about host's network configuration parameters and therefore the protocol can be used to hostile purposes. Because both the client and the server can send DHCP messages including configuration parameters, the security issues are divided into two classes:

- Unauthorized DHCP Servers: Installing a "rogue" DHCP server, it can respond to client requests and offer them forged configuration information. This could isolate clients from a network or set them ready for further abuse later on. For example, a cracker could install a forged DHCP server which

offers forged DNS server information for DHCP client. This DNS server is under cracker's control and it can direct client computer to forged web-pages which install different kinds of malware to the computer. For instance, this computer can be then used in botnets or cover other cracking trails.

- Unauthorized DHCP Clients: Because MAC addresses are easy to forge, the client could be configured to act as a legitimate DHCP client and obtain configuration parameters from a server. With this information a cracker can compromise the network later on. Software which generates forged client requests can be used to create DOS attack by using up all the IP addresses in a DHCP server's pool. This could be also used in stealing purposes. A cracker can steal an IP address for own use.

These are serious security threats in DHCP, so how can we add security to DHCP? One solution is to put effort on security at lower layers. Controlling the physical access of the network is an important technique to prevent an unauthorized host to get access to the network. This can be done for example, with MAC address lists on LAN switches or implementing and deploying 802.1X infrastructure to network. The second alternative is IPSec which provides authentication, integrity, and confidentiality for data transfer. Because IPSec provides strong authentication methods, thereby it is a potential option to address security problems in DHCP.

These examples add security to DHCP but there are also problems in these methods. MAC address lists in LAN switches are very arduous to maintain, if hosts change their places in network topology. 802.1X needs a certain infrastructure to work and it takes time to deploy it to large networks. Hosts need IP-addresses before they can use IPSec services. This means that they have to use DHCP services first that they can receive network configuration parameters. With IPSec network administrators can prevent unauthorized clients from getting access to network servers, but it cannot prevent rogue DHCP servers. (Kozierok 2001.)

4.2 DNS

Like DHCP, Domain Name System (DNS) is based on a client-server model. It is a distributed database which contains information about domain names, host names and their IP addresses. Basically DNS is translating hostnames to IP addresses and vice versa. DNS was implemented in 1984, because host files which earlier handled

hostname - IP address mappings grew too large and maintaining of these host files was a huge problem. Maintaining and updating was difficult because there was no centralized management over host files. Each host file was updated independently. The result was that the host files were never up to date. DNS which had centralized management and distributed database was answer to these problems.

Paul Mockapetris was responsible for designing the new system, and he released in 1984 RFCs 882 and 883 which described the new system; DNS. Several updates have been published for these RFCs. RFCs 1034 and 1035 are the current specifications of Domain Name System.

Programs called nameservers are the server half of the client/server model. Nameservers have files which are called “zone files”. These files have information about certain domain or domains and their computer – IP address mappings. The information which is in zone files is called Resource Records (RR). The Client in DNS is resolver and they are used to send queries to nameservers. These queries are sent to port UDP 53. Because DNS is distributed database, nameservers can send zone files to each other. The primary nameserver informs the secondary nameserver that zone file information has changed and the secondary nameserver fetches the new information by connecting to TCP port 53. So, in Domain Name System servers use TCP traffic and the client use UDP traffic. (Albitz & Liu 2006, 9, 22.)

4.2.1 Security threats in DNS

Humans have difficulties to remember IP addresses (IPv4) because they consist of 12 digits. Names are much easier to remember and therefore DNS is one of the most important services in computer networks. Like DHCP the DNS was implemented in 1980s and the specifications did not have weight on security issues.

When DNS was implemented the trend was to use in IP based applications IP addresses and host names as a basis for allowing or disallowing access. For example UNIX programs like rlogin and rsh used host names for authentication. A user could authenticate himself without entering password from trusted hosts. Compromising organization's domain name server made a possible to masquerade an untrusted host as a trusted system and connect to UNIX servers. There were also other protocols which evolved with similar authentication methods, such as Network File System (NFS), X windows, Hypertext Transfer Protocol (HTTP), etc.

DNS protocol has not got any restrictions who can query resource records from nameservers. DNS is designed to be a public database and the concept was not to limit access to information within the DNS name space. Later versions of the BIND which is one implementation of nameserver program support certain access control features. These features include for example, restriction of zone transfers. Although there are some access control features in the nameserver programs, the main idea for the DNS protocol was that there are no restrictions who can query RRs (resource records).

Protocols like rlogin became more common and that caused pressure for the accuracy of information contained in the DNS. Forged information within DNS could lead to many security problems. It is not an improving factor that the DNS runs over IP, UDP and TCP which are insecure protocols. DNS has lack of authentication and integrity checking of the information. The vulnerabilities within the DNS can be divided into following categories: Cache poisoning, client flooding, dynamic update vulnerability, information leakage, and compromise of the DNS server's authoritative database.

In **Cache Poisoning** method, the main purpose is to input false information to nameserver's or resolver's (client) cache. The server has a cache to which it collects answers which have been queried from another nameserver. This cache can contain also negative answers. When DNS server gets a query from a resolver (client), it first looks up if the cache contains the information that the resolver queried. If the answer to a query is not in its cache, the DNS server can forward the query to another DNS server. Forwarding the query to a rogue DNS server which has forged information, the rogue server answers to query with forged information and cache poisoning occurs. Resolver's (client) cache has the same kind of analogue as the nameserver's cache. Therefore it is also vulnerable for cache poisoning attacks. Cache poisoning is also known as DNS spoofing.

Earlier versions of the BIND were very vulnerable of cache poisoning attacks. There were a few of problems which caused security issues. First, nameservers could respond to a query with information which did not include the related answer. A DNS server which received this spoofed "answer" did not perform any necessary checks to assure that the additional information was correct or even related in some way to the answer. Therefore a rogue nameserver could fill false information in the additional records section of the DNS response message. Another security problem was in earlier versions of BIND, that they did not have a feature which verified that the answer was

related to the original query. This also made nameservers vulnerable for cache poisoning. With cache poisoning, an attacker's goal is to execute a denial of service attack or masquerade as a trusted entity. For example, the attacker can forward client to forged HTTP server by injecting forged information to nameserver's or resolver's cache. Usually these forged HTTP servers include other malicious code which can infect client computers.

Client flooding is one method to execute DoS attack. The client is flooded with thousands of answers from the rogue nameserver, the client system sends out a query, but receives and accepts thousands of DNS responses. Because the client does not do any authentication for the responses, it accepts the messages.

DNS Dynamic Update Vulnerabilities are also used for denial of service attack (DoS)**.** Dynamic DNS is a modification to RFC 1035. DDNS allows dynamic updating of DNS resource records. For example DHCP can add or delete host IP address and name information to nameserver's zone files. DDNS update protocol has methods to control what applications are allowed to update zone information. These access control methods are based on IP addresses and therefore they are vulnerable to threats such as IP spoofing.

An attacker can use DNS tools such as nslookup in Windows computers to request zone transfer from nameserver and start footprinting the network. This threat is called **Information Leakage**. An intruder can automatically query IP address to hostname mappings in a domain space and discover IP addresses that are not assigned. This helps an intruder to use IP spoofing to masquerade as a host of a trusted network.

The vulnerabilities that are described above forced to develop new security features to DNS. The Internet Engineering Task Force (IETF) started working on DNS security extensions known as DNSSEC. (Davidowicz 1999.)

4.2.2 DNSSEC

RFC 2535 standardizes DNSSEC extensions. The main objective for these security enhancements was that they are interoperable with non-security aware implementations of DNS. The IETFs work group defined a new set of resource records (RRs) which interoperate with existing types of RRs. These new RRs improved security of DNS zones. Interoperability allows resolver to query DNSSEC information through non-security aware DNS servers and the security aware server to

return an answer through non-security aware DNS servers. It also allowed easier upgrading process to DNSSEC.

When DNS was implemented, one of the main principles was that DNS is a public service. For a healthy domain name system the correctness and consistency of response information is vital. This brought the need for authentication and integrity, but because DNS data is public there was no need to implement access control and confidentiality to DNSSEC. Authentication and integrity are provided with public key technology (PKI). The information that DNS zones contain (RRs) is digitally signed. With this technology, servers, resolvers and application which support DNSSEC can verify that the information received from the nameserver has not been altered by a third party. Even though there were no demands for data confidentially in DNS transactions, IETF did not prune in their DNSSEC specifications the ability to provide support for confidentiality. This made it possible for other applications to use public keys which DNSSEC provides. Security extensions for DNS also provide methods to have several keys for a given DNS name. Each key can be created from a different cryptographic algorithm. DNSSEC scope can be summarized in three services: **key distribution**, **data origin authentication, and transaction** and **request authentication**. To utilize DNSSEC features, the nameserver and the resolver have to be "security aware" (security-aware nameserver and security-aware resolver).

The **key distribution** service offers the retrieval of the public key. When the resolver queries the nameserver, it can confirm that the DNS zone data is not changed (answer for query is correct) by checking the answer's digital signature. The digital signature is verified with the correct public key which is a certain resource record (DNSKEY) in zone information. Other applications can also utilize key distribution service to distribute cryptography keys. As mentioned in previous paragraph this service supports several different types of keys and different types of key algorithms.

Data origin authentication is a service which uses the digital signature technology to confirm that zone information is trusted. This mitigates threats such as cache poisoning. Each DNS zone has a digital signature which contains encrypted hash. The hash is generated with certain hash algorithm and the hash value is encrypted with private key (digitally signed). The querying party (for example resolver) decrypts the message with nameserver zone's public key and then computes the hash value. Nameserver and the resolver have to have equal methods to compute the hash value.

After the answer is decrypted with zone's public key, the resolver compares hash values and if the values match, the data has integrity and the origin of the data is authentic.

DNS transaction and request authentication provides security for DNS requests and message headers. With this service the resolver can verify that the nameserver answers to the original query and the response came from the certain server. Nameserver adds special SIG resource record at the end of the reply and digitally signs the concatenation of the server's response and the resolver's query. This allows a security-aware resolver to verify the transaction. This service is also used in Secure DNS Dynamic Update. DNSSEC provides strong authentication for systems allowed to dynamically update DNS zone information.

Public Key Retrieval from zones can be made in two ways. Resolvers can query public key information or the key can be statically configured to the resolver. Both ways have certain problems. There is the key trusting issue when obtaining keys with DNS query. A public key must have a signature and the signature has to be reliable. To address this problem is to configure statically the resolver with the public key that authenticates the signed keys below it. This can be done by configuring the resolver with root zone's public key which is a starting point for verifying all keys found below it. The maintaining process of statically configured resolvers is arduous. When a zone's key is changed, then all resolvers have to be manually updated to correspond to change. (Davidowicz 1999.)

4.2.3 Windows and DNSSEC

Windows 7 has a DNS client which is a "non-validating security-aware stub-resolver". This means that this stub-resolver does not perform validation of queried responses. The client (stub-resolver) trusts its nameserver which performs validations on behalf of client. Because the client does not do validation "Trust Anchors" do not need to be configured. The client is said to be security aware because it expects the configured DNS server to do validation and send results in the response. When the client sends a DNS query to the server it sets the DO bit (value 1) which is situated in DNS packet header at the Z-field. The DO bit tells the server that a query came from the security-aware resolver. The client sets the DO bit only when it queries about DNSSEC information. This is a policy based mechanism where the client checks from "Name Resolution Policy Table" (NRPT table is stored in client) which domains it is to

expect DNSSEC. The client expects from the server the AD bit in the response. If the server fails in validation the AD bit is not sent in response and DNS client fails the query. Table 1 is presenting example of NRPT (Seshadri 2008.)

TABLE 1. Example of NRPT (Seshadri 2008)

Namespace	DNSSEC validation	Last hop – IPsec	IPsec encryption level
*.test.com	Set DO bit; Expect server to validate	Secure last hop with IPsec	High encryption
*.foo.test.com	Don't set DO bit; don't expect server to validate	Don't secure last hop with IPsec	n/a

Name Resolution Policy Table also defines IPSec connection to security-aware DNS server. Because the client cannot authenticate the DNS server (it trusts the server's DNSSEC validation), it has to do the authentication in some other way. The authentication can be done with IPSec and IKE (Domain Name System Security Extensions.). Authentication with IPSec is explained in chapter "5.4 IPSec and IKE".

This authentication mechanism is supported only in Windows 7 and Windows Server 2008 R2 operating systems. Windows Server 2003 partly supports DNSSEC. The server can be a secondary nameserver for DNSSEC zones and Windows XP computer can receive DNSSEC queries from the security-aware server. It cannot do validation (does not perform any cryptography, authentication, or verification), but it can store DNSSEC resource records in the stub resolver's cache. (Using DNS Security Extensions (DNSSEC) 2005.)

RCF 4035 defines that stub resolver which queries about DNSSEC RRs from the recursive nameserver will need to set the DO bit in order to receive DNSSEC RRs. Because Windows XP cannot do this, Windows 2003 server can be configured from registry to include the DNSSEC resource records in all query responses (Modify DNSSEC configuration).

When the stub resolver queries DNS information from the recursive nameserver and the response contains DNSSEC resource records, it caches them in the same way as any other resource records. After the client has received the SIG RR relating (digital signature) to the resource records, it does not perform DNSSEC authentication where the receiver sends an additional query to the server to obtain the corresponding KEY record. In other words, resolvers do not authenticate resource records by verifying the

signature information contained in the SIG resource record. The resolver does not either recognize DNS packet header flags (bits) such as CD and AD bits. (Using DNS Security Extensions (DNSSEC) 2005.) With CD bit, the security-aware resolver can disable signature validation (RFC 4035 2005). AD bit is set if all data in the response has been cryptographically verified or otherwise meets the server's local security policy (RFC 3655 2003).

5 AUTHENTICATION AND SECURITY PROTOCOLS

5.1 Security protocols, access control and authentication

The security protocols in the computer systems' are typically designed so that the system survives malicious acts such as man-in-the-middle attack, replay attack and denial of service attack. These attacks are usually the most common attack types which are used to exploit the security protocols. The protection against all possible attacks is not rational; therefore protocols are typically designed under certain assumptions about the threats.

Evaluating a protocol involves the current threat model analysis, and the analysis, does the protocol manage the threats. (Anderson 2001, 13.)

There are many ways to categorize threats. Microsoft uses the STRIDE method to categorize threat types which are: spoofing identity, tampering with data, repudiation, information disclosure, denial of service, and elevation of privilege. (Securing Windows 2000 Server 2004.) These threats can create vulnerabilities in the computer environment and these vulnerabilities can be exploited with different attacks. For example, **spoofing identity threats** contains an illegal use of another person's authentication information and obtaining illegal access to computer environment with this information. This information is usually gathered with man-in-the-middle attacks which the spoofing identity category of threat includes. (Server and Domain Isolation Using IPSec and Group Policy 2006, Appendix D.)

To address the common attacks types, the use of cryptographic authentication methods/protocols is needed. (Anderson 2001, 15) The authentication is an important part in access control function. In the client's NAC concept, the client must authenticate the network to make a conclusion about the trustworthiness of the

network. This chapter introduces the theory of different authentication methods and security protocols.

5.2 Challenge/response method

Many authentication methods use challenge/response architecture. To understand better the authentication methods which are described in this chapter, it is important to know basic functionality of challenge/response authentication architecture.

In Challenge-response (two-pass protocol) method the authenticator (server) and the supplicant (client) have a shared secret (for example password string) which is not sent through the network. The authenticator sends a challenge, consisting of a random n-bit number to the supplicant. The supplicant computes a response with algorithm F from the challenge and the shared secret. The authenticator also computes a response from the challenge and the shared secret with the same algorithm F. If the authenticator and the supplicant have the same response, the supplicant has proved to be the holder of the shared secret. (Käyttäjien tunnistaminen ja PKI, Johdanto todentamiseen 2007, 5)

5.3 DHCP authentication

To address security issues in DHCP the IETF published RFC 3118 “Authentication for DHCP Messages” in June 2001. This RFC defined a new DHCP option which provided authentication for DHCP messages. Both clients and servers check the authentication information and reject messages that come from invalid sources. RFC 3118 provides authentication using shared secret or token-based exchange of messages.

DHCP authentication extension to DHCP protocol has not been a success story. The problem was that the RFC was released relatively late. There were already millions of DHCP clients and servers which did not support this new standard, when it was released. The second problem with these extensions is that they are against the DHCP main philosophy which was to get away from having to pre-configure clients before they can access to the network. Using this standard requires additional configuration of the DHCP client, for instance shared secrets, which have to be keyed to client’s memory.

RFC 3118 provides mutual authentication and therefore it is a potential authentication alternative in a client’s NAC solution. However, it is a fact that this option is not

widely deployed, and most networks must rely on more traditional security operations. (Kozierok 2001.)

The authentication method which is described in the RFC 3118 document is called **Delayed authentication**. RFC describes the use of a new DHCP option type the **Authentication option.** This option type has a certain format for the authentication messages.

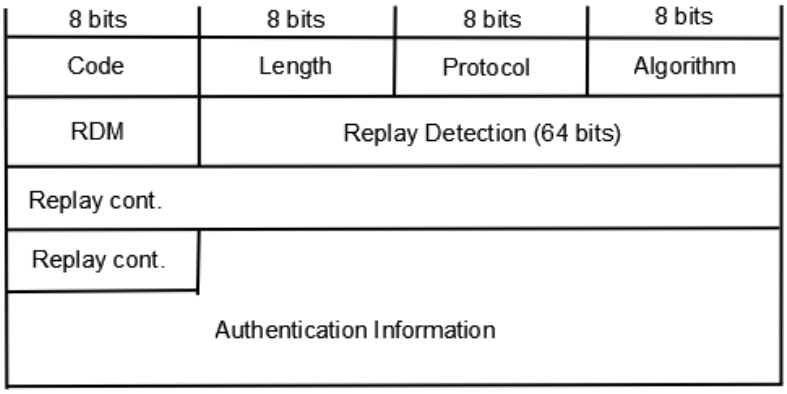


FIGURE 2. Format of the DHCP authentication option message (RFC 3118 2001)

The protocol field having value 1 informs that the delayed authentication is in question. The authentication request for server is sent in DHCPDISCOVER message. The client includes to the authentication request a client identifier option to identify itself uniquely to the server. Discover message includes just the authentication request, so it does not include the client's authentication information; therefore first DHCP message the DHCPDISCOVER message is not authenticated. The client only adds a nonce value to this message. Hostile clients could flood server with DHCPDISCOVER messages.

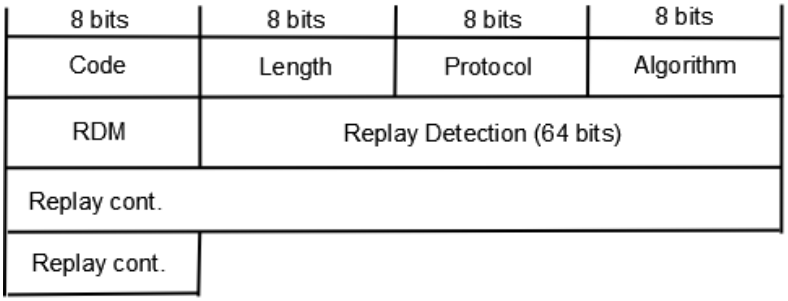


FIGURE 3. Format of DHCP authentication request in a DHCPDISCOVER or a DHCPINFORM message (RFC 3118 2001)

The server answers to request with a DHCPOFFER message which contains the authentication information. The authentication information field includes a nonce

value generated by the source as a message authentication code (MAC) to provide message authentication and entity authentication. When a client receives an offer, it can verify that the message is a reply to its DHCPDISCOVER message by computing the nonce value which was sent in the original discover message. The client identifier option was also included in it DHCPDISCOVER message to identify itself uniquely to the server. After the server has received the discover message it computes with client identifier and master key (MK) the client's key (K) and adds the secret ID value and MAC value to DHCPOFFER message. MAC value is computed from whole DHCP message including the DHCP message header and the options field. These are used as input to the HMAC-MD5 computation function. The secret ID is the identifier of the secret used to generate the MAC.

The client first checks that the value in the replay detection field is acceptable in DHCPOFFER message. This field monotonically increases counter, which mitigates replay attack threat. Next, the client computes the MAC. It uses its shared key, DHCP message information and same MAC algorithm than server used. If the MAC value is equal with the MAC value in DHCP offer message, the server is trusted. Otherwise the DHCP message is discarded. After the client has authenticated the server it accepts the offer and sends DHCPREQUEST message which includes the client's authentication information. The server uses the same methods than the client to do the authentication.

Delayed authentication is based on a shared secret. To utilize shared key techniques, it is safer to create each client their own key, because if all clients have the same key, unauthorized clients can masqueraded as authorized clients by obtaining a copy of the shared key. In this key utilization case, each server must know each client's key to authenticate the client. The key is created with MAC (message authentication code) from master key and unique identifier. The client identifier (unique identifier) can be for example subnet address or MAC address, which is unique to a certain client. The key (K) is generated with formula K = MAC(MK, unique-id). MK is a master secret key and MAC is a keyed one-way function, for example HMAC-MD5.

Without the master key MK, a client cannot create its own key K. Using the master key to create other keys has some advantages. First, the server can verify the received message by regenerating K from the client-id. This means that the server does not need to check any plain passwords. Second, verifying the MAC by computing it, the

server does not require separate authentication server. For better security, it is recommended that the MK is not stored by any clients. If MK is compromised, all clients can have new individual keys by creating a new MK.

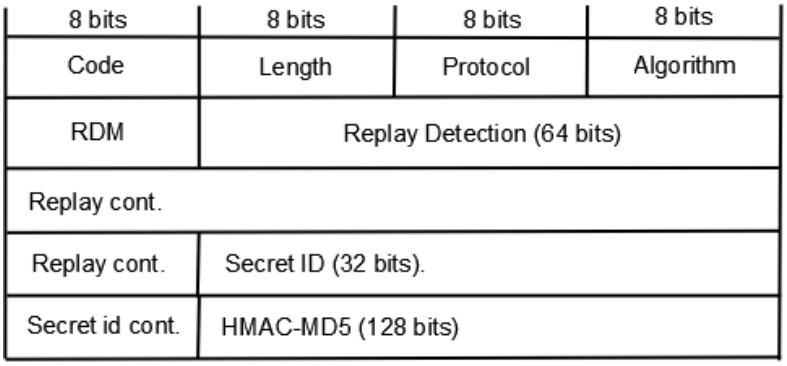


FIGURE 4. Format of the authentication information in a DHCPOFFER, DHCPREQUEST or DHCPACK message (RFC 3118 2001)

As earlier mentioned K means a shared secret value between the source and destination. Each shared secret has a secret ID value which has its own field in the message. Secret ID is a unique identifier for the shared secret value used to generate the MAC for this message. MAC can be created for example with HMAC-MD5 function. The receiver can determine from secret ID value which shared secret was used to generate the MAC in the DHCP message. (RFC 3118 2001.)

Microsoft does not support the RFC 3118 (IPv6 Security Considerations and Recommendations).

5.4 IPSec and IKE

Microsoft Windows XP has built in IPSec services, so there is no need for separate VPN (or IPSec) client. MS IPSec supports also securing remote access over Internet using the Layer Two Tunneling Protocol (L2TP). Therefore the client is called L2TP/IPSec VPN client. IPSec protocol is integrated to Windows 2000/2003 domains and the Active Directory services. IPSec policies could be distributed by AD group policy to Windows 2000\2003\XP AD domain member computers. (Szymanski, 27-28.)

IPSec (Security Architecture for the Internet Protocol) is protocols/services to provide cryptographically-based security for IPv4 and IPv6 traffic. It provides access control, data integrity, data origin authentication and confidentiality (encryption). The traffic security protocols that IPSec offers are: Authentication Header (AH) and the

Encapsulating Security Payload (ESP). Security services and protocols are agreed in IPSec security associations (IPSec SA). IKE (Internet Key Exchange Protocol) protocol provides key exchange for IPSec peers and negotiation for IPSec SAs. AH and ESP use symmetric keys to encrypt the traffic, therefore there is a need for changing keys between peers in IPSec connections. (RFC 2401 1998.)

IKE has two phases:

- IKE phase 1 which is used to authenticate and protect the identities of the IPSec peers and negotiate keys. Key negotiation is performed as an authenticated Diffie-Hellman exchange with the end result of having matching shared secret keys. IKE peers then use these keys to communicate securely during phase 2 negotiations. (IKE phase 1 has two different modes: main mode and aggressive mode)

- IKE phase 2 has one mode (quick mode). After IKE has established the secure tunnel in phase 1, quick mode occurs. Quick mode negotiates a shared IPSec policy, derives shared secret keying material used for the IPSec security algorithms, and establishes IPSec SAs. Quick mode exchanges nonces that provide replay protection and generates new shared secret key material.

When IPSec peers negotiate security associations, they also have to authenticate each other, so that the connection is established with the trusted entity. This authentication is made in IKE phase 1, and there are three options to authenticate peers:

- Pre-shared key

- Public key (PKI)

- Digital signature

Both digital signature and public key authentication require certificates. (RFC 2409. 1998)

The authentication in IKE phase 1 is computer to computer authentication with one of the techniques described above. When designing Client's NAC system, the main idea for the authentication module is that the authentication is made without user interaction. Microsoft L2TP/IPSec VPN Client implementation and Microsoft IKE

slightly differ from basic authentication methods. Windows IKE can use one of the following three methods:

- The Kerberos version 5 authentication protocol
- X.509 digital certificate with corresponding public and private Rivest, Shamir, & Adleman (RSA) key pair
- A pre-shared key

The misunderstanding that Windows IPSec requires public key infrastructure (PKI) certificates, which are often difficult to deploy, is a common reason why organizations do not deploy IPSec. Because PKI infrastructure is quite heavy to deploy, Microsoft integrated Kerberos version 5 in the IKE negotiation protocol. (Server and Domain Isolation Using IPSec and Group Policy 2006.)

5.4.1 Server and domain isolation techniques

Server and domain isolation is designed to work with the existing devices and techniques in the user's network infrastructure. The key point is that the isolation is implemented by making IPSec policies and distributing these policies via AD group policies. Traditional isolation is done with different technologies and procedures such as network segmentation with firewalls, VLANs and perimeter network access controls. Server and domain isolation can be implemented with little or no change in the existing network paths and connection methods, with little or no change to applications, and with an existing Windows 2000 or Windows Server 2003 domain infrastructure.

Server and domain isolation techniques are based on two mechanisms: **host authentication** and **host authorization**.

The host authentication mechanism inspects whether the initiator computer of the connection has valid credentials. The credentials are inspected from the Kerberos ticket, a certificate or a pre-shared key. There are two technologies that can provide this type of authentication mechanism on Windows-based computers; these technologies are the 802.1X protocol and IPSec.

After the host has determined that the communication comes from a valid (trusted) source, the host has to make a decision whether to allow or deny access. Even though

the device is authenticated, it does not guarantee that it is allowed to access a certain host. These restrictions are made with standard Windows groups to limit the users' and computers' abilities to access the resources on other computers. There is also a possibility to make user right assignments based on network, for example "Access this computer from the Network" (ALLOW) and the "Deny access to this computer from the network" (DENY).

Here is an example of a five-step process how authentication and authorization work with IPSec using Kerberos authentication and group policy.

1. User who is logged to client computer tries to get access to a share on a file server. File server is trusted host within the logical isolation. Client computer initiates the connection by connecting to server's TCP port 445. Client has IPSec policy for that server and protocol (TCP 445). Connection request to servers TCP 445 port triggers an IKE negotiation to the server. The client IKE obtains a Kerberos ticket to authenticate to the server.

2. In IKE phase 1 the server authenticates the Kerberos ticket. During the authentication process, IKE checks the ALLOW or DENY users rights from the Group Policy. With the required user right assignment, the IKE negotiation will complete, and an IPSec main mode SA will be established.

3. IPSec policy is checked to negotiate security settings for the IPSec connection.

4. After IPsec-protected communication is established, user host access permissions are checked on the server to verify that user has the required host access permissions in the Group Policy for the trusted host.

Finally, the standard Windows share and file access permissions are checked to ensure that user has the required permissions to access the data. (Server and Domain Isolation Using IPSec and Group Policy 2006.)

5.4.2 X.509 digital certificate

A certificate is a digitally signed statement which uses PKI (Public Key Infrastructure) technology.

PKI is based on a pair of encryption keys; public key and private key. The keys can be attached to identity of the person, device, or service. (Administrator's Guide to

Microsoft L2TP/IPSec VPN Client 2002.) The main idea is that a public key can be send over network and it is available for everyone. The private key is known and kept only by the owner. The usage of keys depends on whether encrypting the data or digitally signs data. A classic public key encryption example for encrypt data between two parties is presented as follows: Alice (A) and Bob (B). Alice and Bob have both public keys A_a, B_b and private keys A_i and B_j. Alice sends data D to Bob, it uses Bob's public key and known encryption algorithm to encrypt data which looks this: $B_b(D)$. When Bob receives the data it uses its private key and known decryption algorithm to decrypt the data: $B_j(B_b(D))$. In digital signatures, the method is somewhat different. The data is signed with the sender's private key and the receiver can validate the sign with sender's public key. (Kurose J & Ross K 2008, 691, 701.)

PKI has one problem; it is vulnerable for man-in-the-middle attacks, because public keys are known for everyone. For example if Cecilia (C) is eavesdropping Alice's and Bob's communication: Alice sends its public key A_a to Bob. Cecilia hijacks the public key A_a and sends its own public key C_c to Bob. Cecilia claims to be Alice. Now Bob sends a message M to Alice with Cecilia's public key $C_c(M)$. Cecilia hijacks the message and decrypts it with its own secret key C_k; $C_k(C_c(M))$. Now Cecilia can read and change the message. She can forward the message to Alice with Alice's public key which she has hijacked. Alice decrypts the message and does not notice anything about the man-in-the-middle attack. (Käyttäjän tunnistaminen ja PKI Varmennejärjestelmien perusideat 2007, 2.)

A solution for this problem is public key certification. A certificate confirms that a certain public key belongs to a specific party. The basic idea in certificates is that a third party digitally signs the subject's (owner's) public key information. The third party is called a certification authority (CA). (Käyttäjän tunnistaminen ja PKI Varmennejärjestelmien perusideat 2007, 2.) Standards for CAs are ITU X.509 and RCF 1422 which describes CA based key management for use with secure e-mail (Kurose J & Ross K 2008, 707).

CA validates identities and issues certificates, which means that Alice can trust Bob's public key and certificate if she also trusts the CA which has issued Bob's certificate. In IPSec authentication, each node on the connection validates the other node's certificate. For example, Alice must have a copy of the certificate for the issuer of Bob's certificate installed locally. If CA_{bob} issued Bob's certificate, then in order to

validate Bob's certificate, Alice must have the certificate for CA_{bob} installed. Otherwise authentication with certificates fails. In the same way, Bob must have a certificate for CA_{alice} installed.

Certificate infrastructure is hierarchical. The root CA in certificate infrastructures is the highest CA in hierarchy and the root CA can have intermediate CAs which are located below the root. If the issuing CA for a certificate is an intermediate CA, then the node performing validation must trust all CAs in the chain including the root CA. The term for this is: validating a certificate chain. The simplest certificate infrastructure has a single root CA which issues certificates to all entities requiring authentication. For example, certificates for both Alice and Bob are issued by a root CA. In installation phase both certificates Alice's and the copy of root CAs certificate are installed to Alice. Similarly Bob needs Bob's certificate and the copy of root CA's certificate.

X.509 certificate has got several fields. It contains much more information than just the CA's digital signature. Hence, when a node validates the other node's certificate it has to do several checks about the certificate. First, the **digital signature** is verified for each certificate by obtaining the public key from the issuing CA certificate. Second, a certificate must not be expired. When certificates are issued, they have a **validity period**, this field contains the start day and the end day of the period. Third, a certificate must not have been **revoked**. Issued certificates can be revoked at any time. The issuing CA maintains a list of certificates that have been revoked. This list is called certificate revocation list (CRL). Usually CRL is distributed to a dedicated computer. When checking that a certificate has not been revoked, the node has to check CRL. If the updating interval is long, there is a risk that a certificate that has been revoked can still be used because the published CRL that the node is checking is out of date. (Administrator's Guide to Microsoft L2TP/IPSec VPN Client 2002.)

5.4.3 Opportunistic Encryption (OE)

Opportunistic encryption is based on IPSec protocols, IKE and DNSSEC for key distribution. The basic idea is that two peers can create an IPSec tunnel by asking the public keys or certificates from DNSSEC server. Because DNSSEC RRs contain the digital signature and this can be validated from trust anchors, the key information (certificates or public keys) that is received from DNSSEC are trusted, therefore OE is resistant to passive attacks and active attackers as well.

OE allows secure communication between peers without any pre-arrangement specific to the pair of systems involved. This makes it possible to create secure communication with an entity that is not known in advance. Each peer's public key information has to be keyed to DNS record (KEY RR and TXT RR) to support opportunistic encryption and then enables this feature in the peers IPsec stack. After this, any two such peers can communicate securely.

Basically OE is designed for security routers, if we have two nodes A and B and these nodes have security routers A-SR and B-SR. In this kind of environment, the OE process would be following (RFC 4322 2005.):

1. A sends traffic to B, traffic travels through A-SR. A-SR checks if there is an IPSec policy for this traffic.

2. If there is an IPSec policy, A-SR check from DNS server B's reverse DNS record (which is located in the reverse DNS in-addr.arpa) and corresponding TXT record. TXT record contains B-SR's IP address and public key information.

3. A-SR initiates IKE negotiation with B-SR. B-SR checks from DNS server A-SR's key (public key) information. It queries KEY record from reverse DNS zone (A-SR IP address has corresponding KEY RR). After successful check IKE SAs are negotiated.

4. The IPSec tunnel is established.

OE utilizes IPSec protocol, IKE and DNSSEC. Windows XP supports the OE; therefore it is applicable for the client's NAC authentication process. The authentication methods that Windows XP supports in OE are pre-shared key and certificates. The pre-shred key is not rational choice if OE is used, for example in public WLANs. The best choice is to use a certificate from a universally recognized CA. The CA with the widest distribution of certificates produces the greatest opportunity for IPsec communications. There are many possibilities on the Internet to create one's own CA and certificates. Thawte (Freemail certificate) is a common CA which can be used to obtain free personal certificate, the benefit in Thawte is that the root Freemail CA certificate is already installed in Windows XP. This means that the computer trusts other computers which have certificates that are signed with Thawte CA. (Steps to turn on optional IPsec on a Windows XP computer 2006.)

5.5 Certificate based authentication using IEEE 802.1X and EAP-TLS as a authentication protocol

802.1X is an IEEE standard which describes authenticated access to IEEE 802 media such as: Ethernet, Token Ring, and 802.11 wireless LANs. 802.1X provides port based authentication and it is one of the main technology solutions in Network Access Control concept. For example, port based authentication provides authentication methods to network equipment to join the network via Ethernet switch or wireless LAN access point. Actually, some consider that the 802.1X is NAC. NAC is more than just the port based authentication, but the 802.1X is an important part of NAC which enhances security and deployment by providing support for centralized user identification, authentication, dynamic key management, and accounting. (Understanding 802.1X authentication for wireless networks 2005.)

RFC3579 defines Remote Authentication Dial In User Service (RADIUS) support for the Extensible Authentication Protocol (EAP). RADIUS is a networking protocol that provides centralized access, authorization and accounting (AAA) management for people or computers to connect and use a network service. RADIUS support is optional within IEEE 802.1X; still most of the implementations have different entities for authenticators and authentication servers (EAP server). These components can be in the same entity but generally authenticators will function as RADIUS clients and authentication servers will function as RADIUS servers and both stand in their own dedicated entity. The protocol used to relay authentication messages between the authenticator and authentication server is RADIUS. RADIUS server can either act as an authentication server or it can forward authentication messages to another authentication server (act as an authenticator).

IEEE 802.1X offers an effective framework for authenticating and controlling user traffic to a protected network. The protected network has certain components such as: supplicant (client), authenticator and authentication server. Authenticator and authentication server work together to make decisions which the supplicant can join to network. It was explained in chapter 1 (Introduction) how NAC handles protection from “network perspective”. 802.1X is a good example of this kind of perspective, the client (supplicant) cannot be sure if the network is trusted or if it is connected to network where 802.1X is not implemented. Is this a suitable authentication method for client’s NAC implementation? 802.1X ties a protocol called EAP (Extensible

Authentication Protocol) to IEEE802 media. EAP is a “carrier” protocol which can transport multiple authentication methods. The client’s NAC main requirement for authentication is that the client can authenticate the network. One of the authentication methods that EAP offers is mutual authentication. Mutual authentication means that the client can authenticate the network and vice versa. This kind of authentication is applicable to the client’s NAC implementation by forcing the client to use only this authentication method. If the network does not have 802.1X architecture and the client cannot authenticate the authentication server, the network is untrusted. The mutual authentication method which EAP provides is EAP-TLS. It provides strong security with X.509 certificates and symmetric encryption keys. EAP-TLS encapsulates the TLS/SSL protocol within EAP messages.

Extensible Authentication Protocol is described in RFC 2284. EAP is used to transport and manage authentication information between the supplicant (peer) and the authentication server. EAP messages use request/respond mode to interact between peer and EAP-server (authentication server). Messages have four different “labels”: EAP-Request, EAP-Response, EAP-Success, and EAP-Failure. EAP conversation has following phases:

- EAP client starts EAP conversation by turning itself to passive listening mode, waiting for the authenticator to initiate authentication. Authenticator is usually in wired connections a LAN switch and in wireless connections it is a wireless access point (AP).
- The authenticator sends EAP request to the peer. The Request has a type field “Identity”
- The peer responds with its identity and the authenticator forwards the EAP message to the authentication server (For example RADIUS server)
- The authentication server determines which EAP authentication mechanism to use with the peer and sends one or more requests for authentication method that has been chosen. The client and the server must use the same authentication method in order for authentication to be successful
- The peer answers each request with a response .

- The EAP conversation is terminated by the authentication server by sending the peer either an EAP-Success or EAP-Failure message. The authentication server also notifies the authenticator the result of authentication. Notify is done by RADIUS Access-Accept or Access-Reject message which contains either an EAP-Success or EAP-Failure attribute.

If authentication fails the authenticator (LAN switch or AP) closes the port or in wireless case it closes the connection which the client is connected to and the client does not get access to the network. Only EAP conversation is accepted network traffic in the authentication phase, other traffic is denied by the LAN switch or AP. EAP messages from the authenticator to the RADIUS server are encapsulated to RADIUS packets using the EAP-Message attribute. (RFC 3579 2003.)

TLS offers security services for TCP protocol such as: confidentiality, data integrity and strong end-point authentication with X.509 certificates. EAP-TLS defines EAP type for TLS and Flags field. The TLS Start flag is used by the authentication server to indicate that following EAP messages will contain EAP encapsulated TLS messages. When the client sees this flag set in a message from the server, it starts a TLS negotiation process on top of EAP.

RFC5216 “The EAP-TLS Authentication Protocol” defines EAP-Transport Layer Security. EAP-TLS supports certificate-based mutual authentication and key derivation, using encrypted negotiation, mutual authentication and key management capabilities of the TLS protocol. The EAP-TLS authentication with certificates contains following steps:

1. The Authenticator and the peer (supplicant) start the EAP-TLS conversation with negotiating EAP. The authenticator sends an EAP-Request/Identity packet to the peer.

2. The peer responds with an EAP-Response/Identity packet which contains the peer's **user-Id**. The authenticator usually works as a pass-through device (for example authenticator could be LAN switch), which forwards EAP packets to the EAP server (for example EAP server can be radius server).

3. When the EAP server has received the peer's Identity, the EAP server responds with an EAP-TLS/Start packet. This is an EAP-Request packet with EAP-Type=EAP-TLS, **the Start (S) bit** set, and **no data**.

4. The EAP-TLS conversation start with the peer sending an EAP-Response packet with EAP-Type=EAP-TLS. The data field of that packet encapsulates one or more TLS records in TLS record layer format, containing a **TLS client_hello** handshake message. TLS handshake protocol has a cipher spec item which specifies the bulk data encryption algorithm (such as null, DES, etc.) and a MAC algorithm (such as MD5 or SHA). It also defines cryptographic attributes such as the hash size. The cipher spec for the TLS records is **TLS_NULL_WITH_NULL_NULL** and null compression. This cipher spec remains the same until the **change_cipher_spec** message signals transitions in ciphering strategies. TLS version number, a sessionId, a random number, and a set of ciphersuites supported by the supplicant are TLS client hello message's attributes.

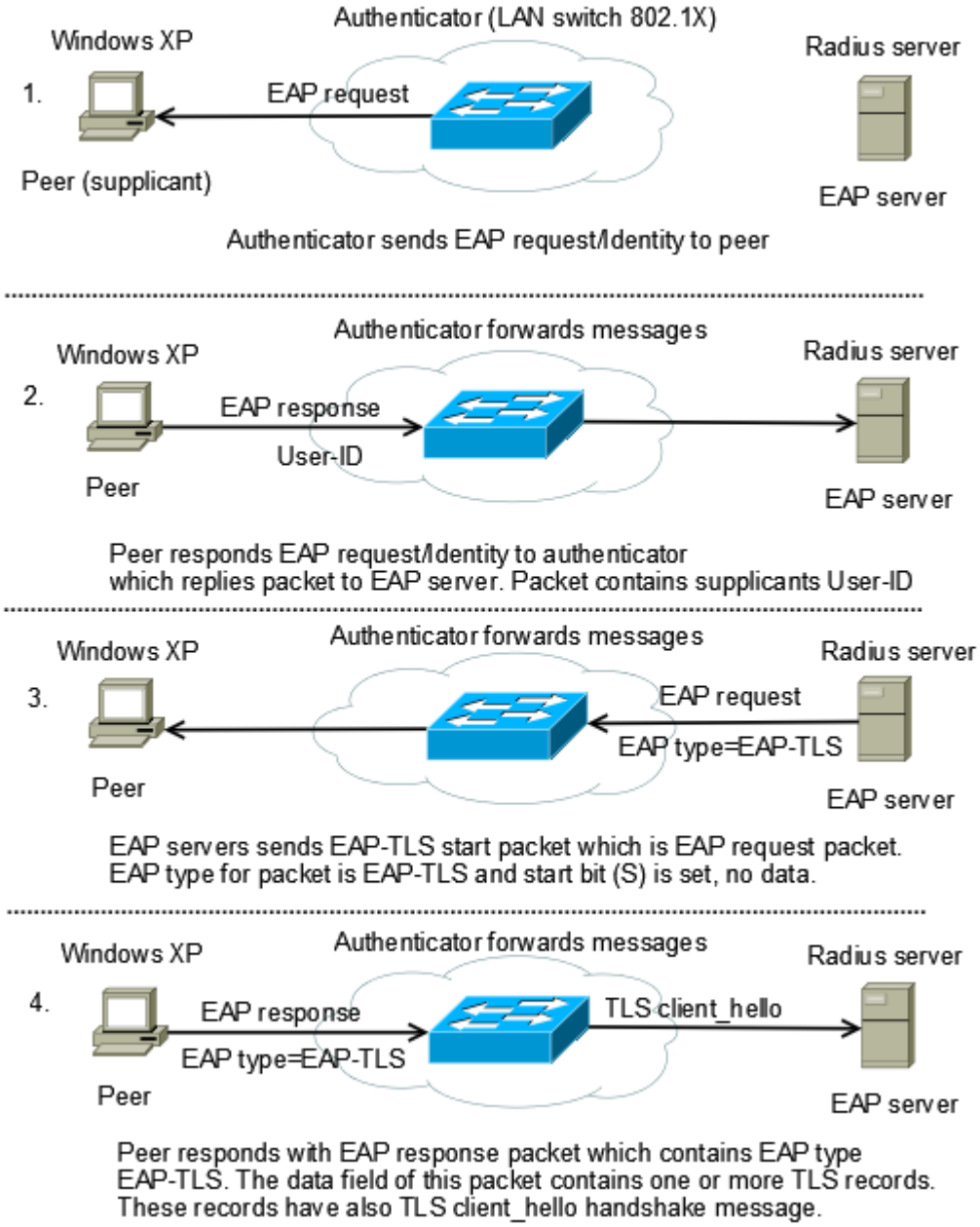


FIGURE 5. First four stages of the EAP-TLS authentication (RFC 5216 2008)

5. The EAP server responds with an EAP-Request packet with EAP-Type=EAP-TLS. The data field of this packet also encapsulates one or more TLS records. Records include: **TLS server_hello** handshake message. This hello message has the same format as client hello message has: TLS version number, random number, a sessionId, and a ciphersuite. The server chooses a ciphersuite from peer's ciphersuite set. **TLS certificate** message which contains a public key certificate chain for either a key exchange public key (such as an RSA or Diffie-Hellman key exchange public key) or a signature public key (such as an RSA or Digital Signature Standard (DSS) signature public key).

S**erver_key_exchange** message allows the key exchange to take place. C**ertificate_request** message is included when the server desires the peer to authenticate itself via public key. TLS s**erver_hello_done** which is the last handshake message encapsulated in this EAP-Request packet,

6. The peer responds with an EAP-Response packet of EAP-Type=EAP-TLS. The data field of this packet encapsulates one or more TLS records containing a **TLS client_key_exchange**, **change_cipher_spec**, **certificate** message which contains a certificate for the peer's signature public key, **certificate_verify** includes the peer's signed authentication response to the EAP server. Finally this EAP-response packet includes **TLS finished** message.

7. After receiving this packet, the EAP server responds with an EAP-Request packet with EAP-Type=EAP-TLS. Server verifies the peer's certificate and digital signature. The Packet includes, in the case of a new TLS session, one or more TLS records containing **TLS change_cipher_spec** and **TLS finished** messages. The TLS finished contains the EAP server's authentication response to the peer.

8. Peer verifies the TLS finished message in order to authenticate the EAP server. If authentication is successful, peer sends EAP-Request packet of EAP-Type=EAP-TLS and **no data**.

9. Finally server sends EAP-Success message to peer. (RFC 5216 2008.)

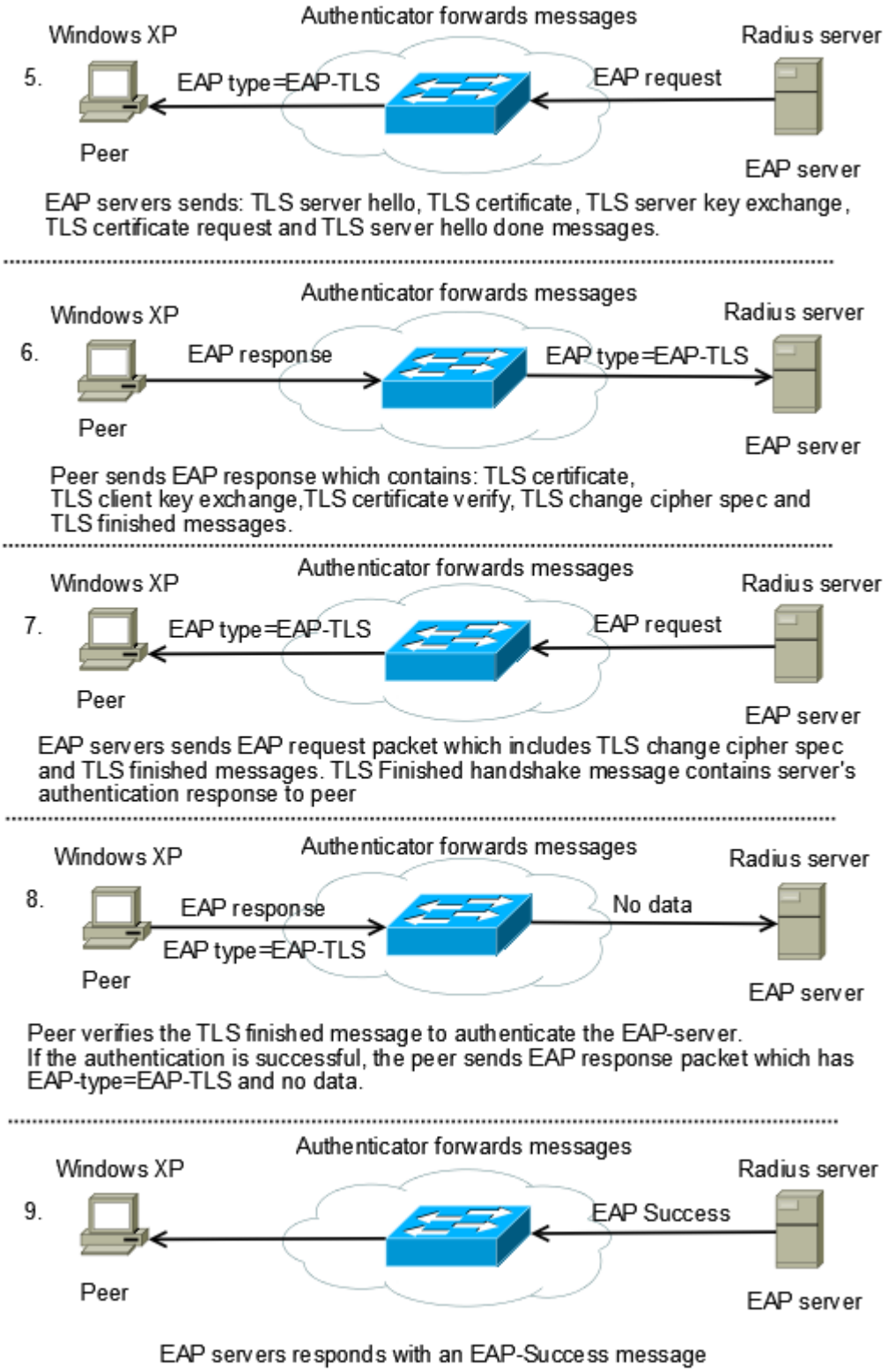


FIGURE 6. The latter five stages of the EAP-TLS authentication (RFC 5216 2008)

6 NETWORK AUTHENTICATION METHODS FOR CLIENT'S NAC IMPLEMENTATION

6.1 Access control fundamentals in client's NAC implementation

"Access control is the traditional center of gravity of computer security. It is where security engineering meets computer science". (Anderson 2001, 51)

Implementing the client's NAC, the access control concentrates on computer to computer authentication or network to computer authentication. This means that the client has to decide if it is going to join the network. This kind of access control is not traditional, because it does not give access rights to client's resources; it just controls client access to network. If the client does not trust the network, it does not join to it and it does not send information about itself to the network. The client has to execute authentication and investigations from a network so that all these functions are invisible for end users and also these functions should not significantly slow down the client's joining process to the network.

Chapter 6 introduces several ways to implement different kinds of access control methods for the client computer. This includes introduction to how different authentication methods could be applicable to a client's NAC. Some of the authentication methods are conducted from certain network protocols such as NBT, DCHP and DNS. Some authentication methods use known authentication methods such as IPSec (IKE) and EAP. Common for all these methods is that they all need some kind of information or service from the network.

The back side of the coin is that these access control methods can also restrict usability if they are badly designed. If the service is down or the information is lost from a trusted network, the client cannot get access to it. One of the main goals for client's NAC implementation is that it cannot create a single point of failure to network and it cannot restrict usability. Services or information in the network has to be duplicated.

Designing the Client's NAC implementation, one of the corner stones is that the implementation is based on mistrust. The implementation can be divided to three upper level stages, which are:

- Network connection is started from a state in which both participants do not trust each other
- Network to computer authentication and verification
- Allowing or denying access to network

6.2 The basics of client's Network Access Control

The main function in client's NAC implementation is that the client authenticates the network and determines if the network is trusted. The determination process needs an application which can carry out the determination by the result of the authentication. This application also has another significant task. It has to control the client's network traffic. First, this application has to observe the state of network interfaces, when an application notices that network interface has activated, it has to notify the authentication module that authentication has to be done. If the authentication process is unsuccessful, the network is untrusted and the network interface has to be disabled. Because this application has to observe the result of authentication and the network interface, it can be called **observing application**. Thereby, the client's NAC implementation is based on two main modules; the **authentication module (AM)** and the **observing module** (**OM** or observing application). Figure 7 describes how these modules interoperate.

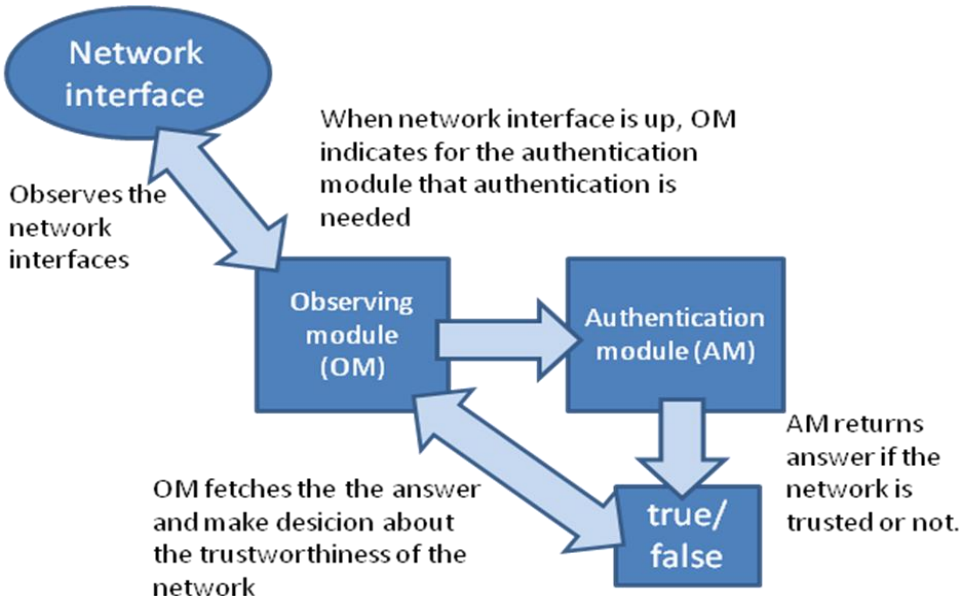


FIGURE 7. Interoperability of authentication and observing module

Figure 8 describes the access process in client's NAC implementation.

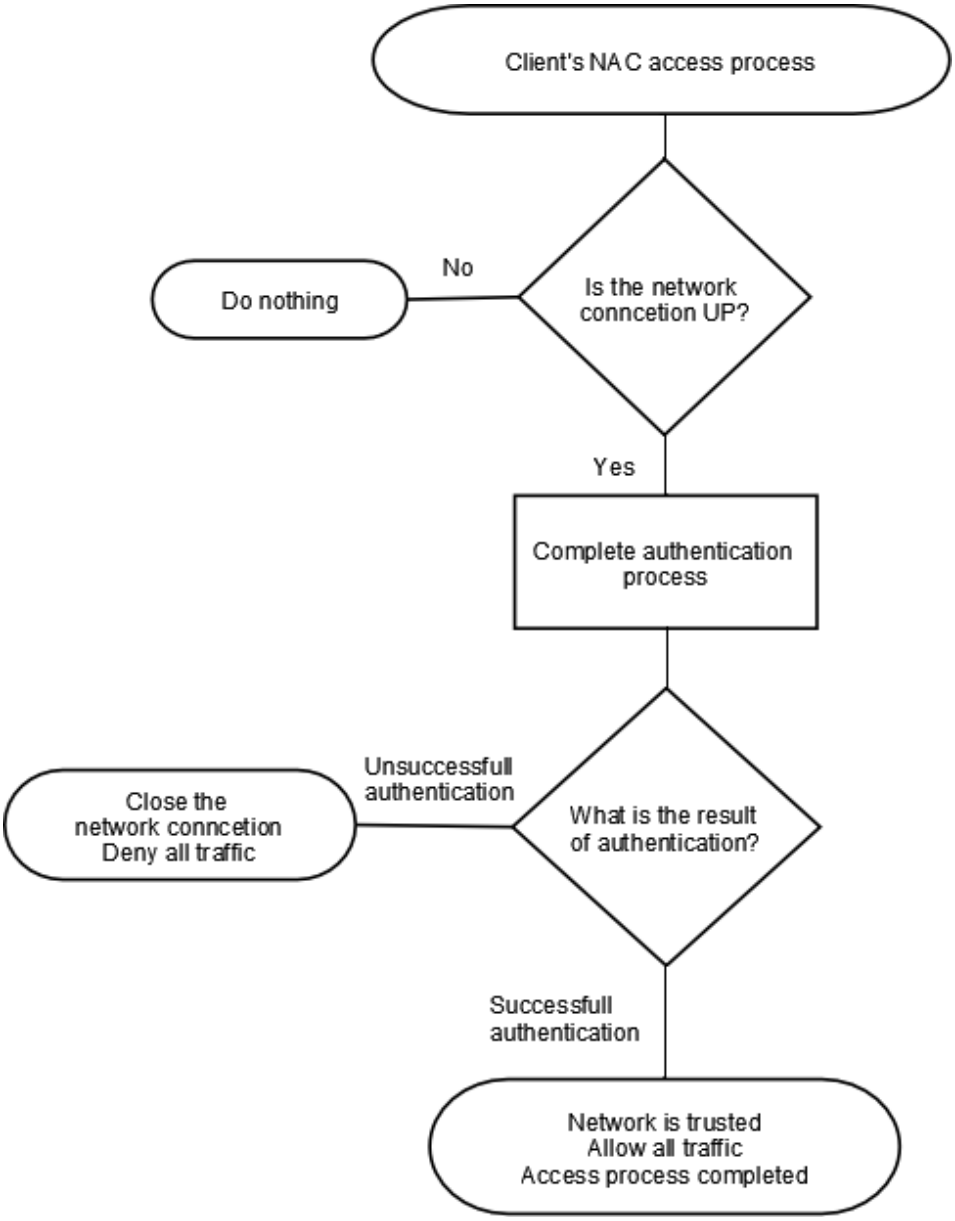


FIGURE 8. Client's NAC access process flowchart

The authentication methods that are presented in this chapter can be divided into: shared secret methods, IPSec authentication with different authentication methods and 802.1X authentication with certificates. Because the authentication methods which are conducted from the network protocols are based on basic challenge/response and shared secret methods, it is logical to start the introduction from this kind of authentication.

6.3 Challenge/response authentication method for client's NAC

In client's NAC implementation the authenticator is the client, which sends a challenge to the network, for example to certain server. The client and the network's server must have the same shared secret. Both client and server compute and create

from the challenge message digest with SHA-1 algorithm. Server sends this message digest to client and client validates the message digest. If the message digests are matching, the network is trusted. Figure 9 is describing the different stages of the challenge/response authentication.

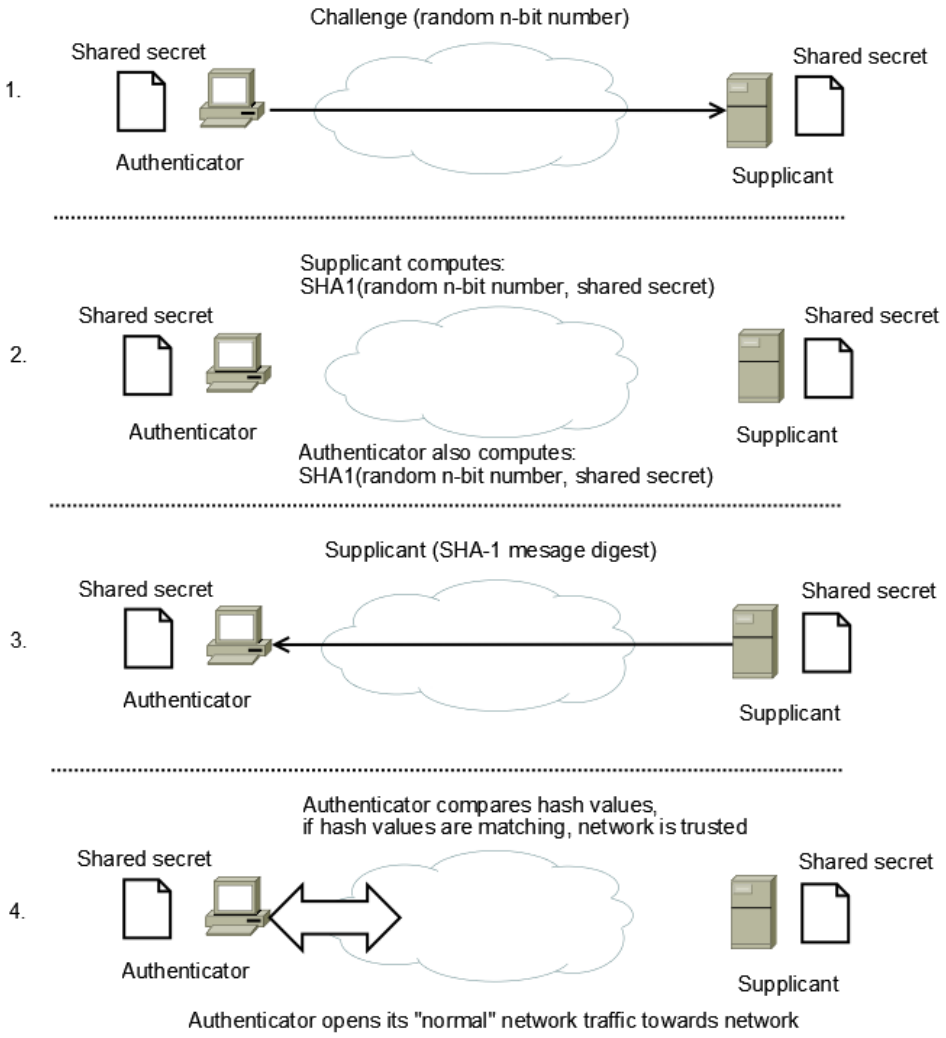


FIGURE 9. Example of challenge/response method with shared key encryption

The problems in random numbers are that they could not be random enough. This creates vulnerability to authentication if random numbers are predictable.

6.4 Using NBT for the network authentication

One way to implement network authentication is to use information which the network already has. The same in security terms; using the shared secret. With NBT there are several ways to implement network authentication. Using the name service in NBT and its name registration process, the first alternative authentication method is that the client is listening for a certain name registration broadcast from the network. At default, NBT-nodes are B-nodes so in this authentication method there has to be

“beacon” in every subnet. When “beacon” broadcasts its NetBIOS names to subnet, the client computer can make decisions based on beacons NetBIOS name if the network is a correct one. The Beacon has to have a function to notice when a new computer is connected to subnet, because otherwise it has to send broadcast traffic all the time to network subnet. The beacon should not cause broadcast flood to subnet. Figure 10 describes the authentication phases of the NBT with beacon method.

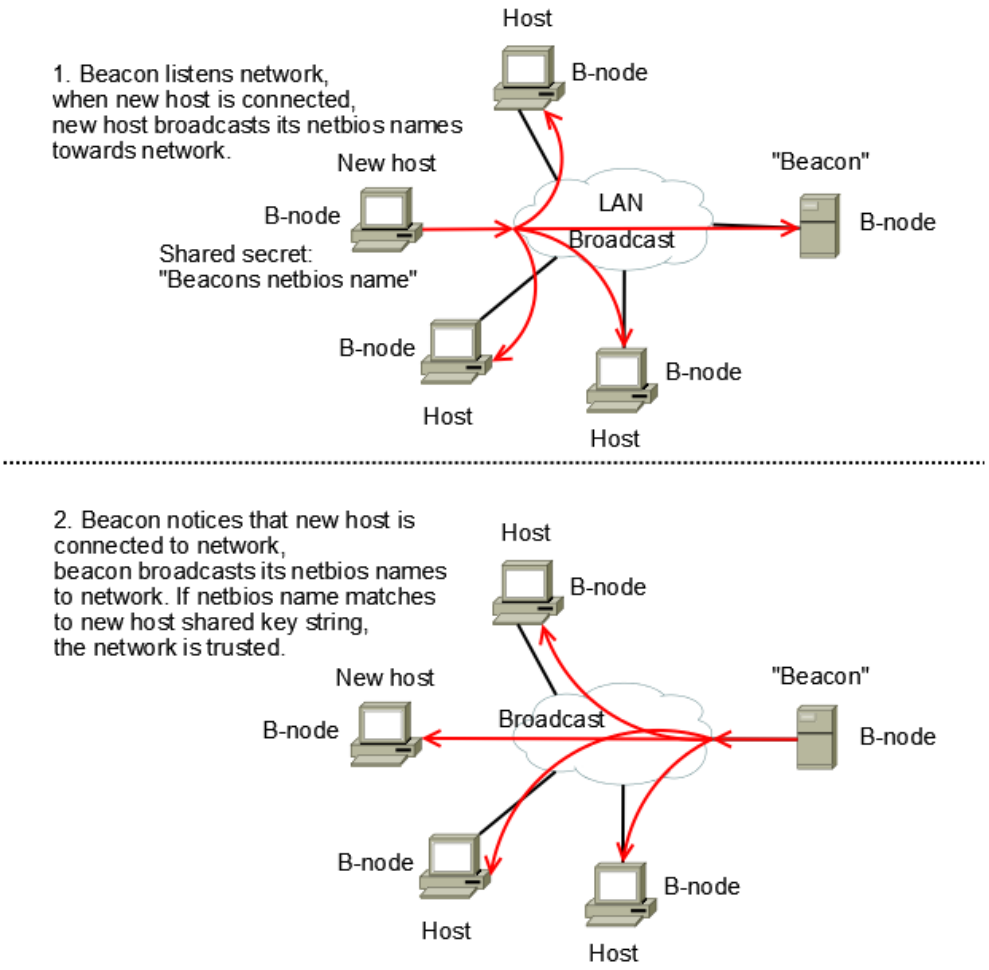


FIGURE 10. NBT with beacon authentication phases

The second alternative is that the network has WINS servers which the client is sends a certain NetBIOS name request to; if the answer is positive the client initiates NetBIOS session connection to this certain computer. If a session is established, the network is correct.

The third alternative uses NetBIOS scope ID. The client and the network’s other computers have a certain NetBIOS scopeID. ScopeID is configured from Windows registry by IT-support. If a client cannot establish a NetBIOS session to certain name/computer, the network is untrusted.

6.5 Using DHCP protocol for authenticating the network

Dynamic Host Configuration Protocol (DHCP) is a common service in IP networks. DHCP is a network application protocol used by devices (DHCP clients) to obtain IP configuration information such as: IP-address, subnet mask, default-gateway and DNS server IP-address. This protocol allows devices to be connected to the network with little or no manual configuration. From authentication perspective, the client using DHCP information for recognizing the connected network goes the same category than client using NBT information. The client authenticates the network with the information that network has.

In client's NAC solution there are a few options how to implement authentication by DHCP. First, DHCP authentication which is described in RCF 3118 can be used for authenticating the network.

Using DHCP authentication option in client's NAC solution needs clients and servers which are supporting RFC 3118. These are not commonly used because RFC 3118 was released relatively late. There were already many clients and servers in the Internet which did not support this RFC. Because authentication is based on shared keys, the shared keys have to distribute with offline method. DHCP authentication offers mutual authentication and therefore it is suitable for client's NAC. The basic function for authentication in client's NAC solution using DHCP authentication is:

- Client computer is forced to use DHCP to obtain network settings
- When client receives DHCP offer message, it can authenticate the server and if server is authenticated successfully, the network is trusted
- RFC 3118 takes into consideration also cases when network connection is broken. In renewing, init/reboot and rebinding stages the client uses the secret it used in its initial DHCPREQUEST message. If the network connection "goes down" the authentication is automatically repeated

Authentication in client's NAC solution can be also made without DHCP authentication option. There are a software tools which can detect rogue DHCP servers from a network. The basic idea of this kind of software is that they send DHCPDISCOVER messages to network and inspects if there are unknown servers answering these discover messages. The inspection is based on IP addresses.

DHCPOFFER message contains a server identification field in DHCP options. This field contains the server's IP address. Network administrators define real DHCP server IP addresses for software and software detects other than these real DHCP servers from network. The software can also display the network configuration information offered by the server.

With this knowledge we can develop a method for lighter authentication by utilizing DHCP. There are number of options to develop this method.

- Option 1: The client can use the tool which can find rogue servers to authenticate the network. Before the client sends any data to network, it executes Dhcploc tool which one example of this kind of tool. If the tool finds rogue DHCP servers network is not trusted and the client closes its network interface. The tool has to be preconfigured by administrators. Dhcploc.exe is Microsoft's command-line tool which displays the DHCP servers active on the subnet. In the case of unauthorized DHCP servers, it can create and send alert messages. Tool can also display packets that it detects from DHCP servers. Administrators can choose whether to display packets from all DHCP servers or only from unauthorized servers. The system requirements for Dhcploc are Windows XP Professional or Windows Server 2003.

- Option 2: A key is created from network settings and stored to the client by offline. This key is created from network setting which already known for network administrators. These kinds of settings are DHCP server IP address, DNS server address and subnet address. The key is computed from these attributes. The client has an application which observes DHCP messages. When the client gets DHCPOFFER message from the server, the observing application computes from the offer message the same attributes that had been computed for the key. If the key value and the computed value are the same, the observing software does not do anything. If the values are not matching, the observing application closes the network interface.

6.6 Using DNS for authenticating the network

Using DNS protocol to authenticate the network is somewhat different from for example using DHCP and NBT. The difference between these protocols is that DNS requires user interactions. Client resolver starts and sends query to DNS server when a

user is for instance using web browser and keying a web page name to the browser. One of the client's NAC demands was that the authentication is executed immediately when the computer is connected to the network and the authentication is done without user interactions. Because of this demand the client computer has to send automatically a DNS query to the network. DNS protocol does not authenticate messages and it has a poor security which was described in chapter 4.2.1 "Security threats in DNS". DNSSEC was implemented to address these security threats and it has mutual authentication which can be applied to the client's NAC implementation to authenticate the network.

The authentication and the validation mechanisms with NRPT (Name Resolution Policy Table) in DNSSEC are supported only in Windows 7 and Windows Server 2008 R2 operating systems. These mechanisms are described in chapter "4.2.3 Windows and DNSSEC". Windows XP partly supports DNSSEC. It cannot do validation but it can store DNSSEC resource records in the stub resolver's cache. Because this thesis mainly concentrates on Windows XP and Windows Server 2003, we have to implement other methods to do the authentication in client's NAC solution with DNS protocol.

Even though Windows XP does not support DNSSEC validation, the information that it stores to resolver's cache can be used to authenticate the network. Using this information to authenticate the network, the network needs a security-aware DNS server. Once again the client must have an application which observes DNS messages and it has to have functionality to send certain DNS queries to a certain DNS server. The method for authentication by using DNS protocol is following:

1. When a client is connected to network, it immediately sends a DNS query to a recursive (Windows 2003 server, which is the secondary nameserver for zone test.com) DNS server (DNS servers are defined in client's network settings). The query must be directed to DNSSEC zone (e.g. query www.test.com), so that the response contains SIG RR for the queried record. The information where to query and what has to be query is keyed for the client offline and before it is going to connect to network. In addition the client must have the test.com zone's public key information. This information is a kind of shared key.

2. After the client has queried and stored the response to stub resolver's cache, the observing application inspects the SIG RR and compares to it KEY information that has already stored to client's memory (shared key).

3. If KEY value matches with the signature (signature is trusted), the server is trusted and also the network is trusted. Otherwise the observing software closes the network connection.

Basically idea of this authentication is that the client has a "shared key" which contains the information that should be in the network which it is going to join. It compares this information to the information that it gets from DNS server. If the information matches, the network is trusted.

6.7 Microsoft VPN architecture for authenticate the network

This thesis is limited to Microsoft networks and Windows XP operating system. Microsoft has its own VPN architecture and its components are installed into XP operating systems. Therefore Microsoft's IPSec implementation is a considerable choice for authentication and encryption in a client's NAC implementation. This chapter concentrates on IPSec authentication possibilities for the client's NAC implementation.

The ability to assign IPSec policies through the group policies is the main idea for "Server and Domain Isolation", which is briefly described in chapter 2.4. This IPSec and group policy method that is described in "Server and Domain Isolation Using IPSec and Group Policy" is also a suitable option to implement authentication in client's NAC application.

For the client's NAC implementation the authentication could be following: Client is forced to make a name query when it is connected to network, or its network interface rises up. After client has noticed that the network connection is up, it performs following steps:

1. The client has an IPSec policy and group policy for DNS query. When the client is sending name query request to server's port UDP 53, it initiates IKE negotiation to server. The client IKE obtains a Kerberos ticket to authenticate to the server.

2. The server authenticates the Kerberos ticket. During the authentication process, IKE checks the ALLOW or DENY users rights from the Group Policy. With the required user right assignment, the IKE negotiation will complete, and an IPSec main mode SA will be established.

3. IPSec policy is checked to negotiate security settings for the IPSec connection.

4. After IPsec-protected communication is established, the user host access permissions are checked on the server to verify that the user has the required host access permissions in the Group Policy for the trusted host.

5. After the client computer has received the answer for the DNS query, the observing application can make decision that network is trusted. If there is a trusted computer, there has to be also a trusted network.

6.7.1 Microsoft L2TP/IPSec VPN Client and certificates

The L2TP/IPSec VPN authentication can be applied in the client's NAC system. In this case Network has to have a VPN server and a root CA. Microsoft L2TP/IPSec VPN implementation needs still some tailoring. The authentication has to be done without user interaction, so the connection to VPN server has to be done automatically. Client computer's joining process to the network with IPSec authentication is following:

1. A computer is connected to the network. Before it sends any traffic to the network, it connects to known VPN server (connection is encrypted).

2. The client computer authenticates with computer certificate to VPN server. (Assuming that certificates are distributed to all parties)

3. If authentication is successful, the client can be sure that it is connected to trusted network.

4. After successful authentication, the client can open its "normal" traffic (including the NBT traffic) towards the network. If authentication is unsuccessful, the client does not open any traffic towards the network. In another words; the client stays silent.

5. Authentication must be done every time a client computer is connected to the network. Network interface card listens for a network signal and when it gets the signal, the first thing to do is authentication

Because this authentication is done when client computers network interface card (NIC) gets the first signal from the network (NIC is raising up/LAN is plugged), the time for authentication can be few seconds. If a client's network connection is broken, it has to do the authentication every time the connection comes back. For example for network segment with 40 nodes, the LAN switch where all the 40 nodes are connected goes broken. This means that all 40 clients are disconnected from the network. When a LAN switch is changed for a new one, all 40 clients try to authenticate the network. The recommendation is that network has several VPN servers (authentication servers), in other words: authentication infrastructure has to be distributed.

6.8 Using EAP-TLS authentication in client's NAC

Implementing EAP-TLS authentication method for the client's NAC solution requires 802.1X and certificate infrastructure to the network. Because of the mutual authentication in EAP-TLS, from authentication perspective this method kills two flies with one hit; the network is authenticated and the client is authenticated in the same authentication conversation.

The only problem is that the client using EAP-TLS authentication can still send traffic to the network if the network has not the 802.1X infrastructure. To address this problem EAP-TLS client requires tailoring. When the client turns itself to passive listening mode to start an EAP conversation, there also has to be object which observes the client's outbound network traffic and the result of the authentication. This object denies all traffic except the EAP conversation until the authentication is over. If the client gets EAP-success packet from authentication server the object which is observing the traffic and authentication result allows all network traffic to the network and the client can be sure that network is trusted. In the case of EAP-failure, the observing object does not do anything; all other traffic (except EAP) is still denied.

7 COMPARING CLIENT'S NAC AUTHENTICATION ALTERNATIVES

In the previous chapter six different methods for authentication in client's NAC implementation were represented. All methods are applied from different network protocols or security architectures. In all cases the client's NAC implementation needs a certain application which uses these protocols or architectures to authenticate the network. The network authentication methods are divided into two different categories:

- Authentication by using the basic network protocols
- Authentication by using certificates and security architectures

Information security is about tradeoffs between security and usability. High security is usually meaning awkward usability and vice versa. The environment defines how secure the implementation has to be. In high secure network environment the security is the main factor when choosing the authentication method and in lower security networks (e.g. wireless LANs), the complexity of implementation is the main factor (less complex is better). Basically authentication methods which use basic network protocols (NBT, DHCP, DNS) and shared secret are directed to lower security networks and authentication by using certificates is directed to higher security networks.

Because client's NAC authenticates the network without user interactions (network to computer authentication, which the user does not see), the usability can be measured with how long the authentication is lasting and if there is a single point of failure in the authentication scene. At this point of research, we do not have any finished implementations of authentication module, so we cannot measure the authentication time. We can only set demands for the time. Single point of failure can be prevented with good designing of the authentication method. Security is also affecting usability, e.g. if authentication method is vulnerable for denial of service attacks, it causes usability problems if clients cannot connect to a trusted network. The usability is hard to analyze at this point, therefore choosing the best authentication method for client's NAC implementation is based on two different analyzes:

- Security analysis

- Implementation analysis

As explained in chapter “5.1 Security protocols, access control and authentication”, it is not rational to implement protection against all possible threats. Security protocols are typically designed under certain assumptions about the threats. The purpose of a security analysis is to evaluate if the protocol manages the threats. The security analysis is done by analyzing what threats a certain authentication method has and if it is possible to exploit these threats with certain attacks. Microsoft uses their own STRIDE method to categorize threat types, but in this security analysis three most common attack types were chosen which are used to exploit vulnerabilities in authentication or security protocols. These attack types are:

- Man in the middle attack (MitM)
- Replay attack
- Denial of service attack

If the authentication data that is sent over network is unencrypted (plaintext) the eavesdropping is quite easy, thereby the encryption of the authentication traffic is also an important factor in the security analysis.

The implementation analysis is done by analyzing how hard it is implement certain authentication alternative. How much tailoring does a certain method need?

In chapter six three different methods to use authentication were described by using the basic network protocols. These methods were:

1. Using NBT for the network verification and authentication
2. Using DHCP protocol for authenticate the network
3. Using DNS for authenticate the network

The methods to use authentication by using the certificates and security architectures were:

1. Authentication with Microsoft L2TP/IPSec VPN Client and certificates
2. Certificate based authentication using IEEE 802.1X and EAP-TLS as a authentication protocol

The authentication methods are compared to each other with these analyses.

7.1 Security and implementation analysis of NBT authentication method

In NBT case there are three alternatives how to implement authentication.

The first alternative authentication method is that a client listens for a certain name registration broadcast from the network. This can be called a NBT with beacon authentication method. The basic idea for this method is that there is a "beacon" in every subnet which broadcasts its NetBIOS names to subnet; the client computer can make decisions based on the beacon's NetBIOS name if the network is a trusted one. The method is based on shared secret. The client has a hashed key value of the name stored to its hard drive. If the beacon's name matches with the name in the client's hard drive, the network is trusted. The advantage of this method is that it is quite simple to implement. The disadvantages of this method are that shared secret is sent over network unencrypted and this is vulnerable of all exploits that were described above: MitM, replay attack and DoS. For example DoS attack is extremely easy to implement; sending a forged message to the beacon computer insisting that its name is not unique. In this situation the beacon computer becomes unavailable to other computers on the network and client computers cannot authenticate the network which means that they do not trust the network and stay unconnected.

MitM is implemented by eavesdropping the beacon's traffic. If client is connected to an untrusted network, sending the same NetBIOS name that the original beacon had to the client, the client believes that the network is trusted.

If we change the beacon's functionality in some way that the name that is broadcast to network is encrypted and hashed, this would prevent the threat of replay attack and mitigates the threat of MitM. The broadcast packet should also contain a nonce value and a time stamp. Denial of service threat can be mitigated from Windows registry by enabling the "NoNameReleaseOnDemand" value (MS00-047 2007). This does not prevent all DoS attacks. DoS attack could be created by replacing the original beacon with a rogue beacon which sends false authentication information.

With these improvements the authentication method is more secure but also it is more complex to implement. These improvements also change the whole idea of using NBT information; with improvements the authentication method can be any authentication method which uses challenge/response and shared key information. The basic

vulnerability in a shared secret is that when the shared secret end ups in wrong hands, the method is easy to exploit.

The implementation of NBT with a beacon is quite difficult to implement because it needs a beacon to every subnet. The beacon has to have certain functionality and the client has to have certain functionality to make decisions about the network. Applications have to be made for both entities. The baseline for a client's NAC solution was that only the client needs the application. The beacon also creates a single point of failure to authentication, if the beacon is down, client could not connect to the network. Therefore, every subnet needs several beacons which make the implementation very clumsy.

The second alternative for using NBT to authenticate the network utilized WINS servers. The idea was that client is sends a certain NetBIOS name request and if the answer is positive, the client initiates NetBIOS session connection to this certain computer. If a session is established, the network is correct.

The advantage of this method compared to the previous alternative is that this does not need the beacon to every subnet. If the name that the client asks is always different, for eavesdropper it is harder to figure out what is the authentication method. The client just asks for the NetBIOS names. If the client does not send any other traffic than the NetBIOS name requests to network, eventually it can be easy to make conclusions that this has to some kind of authentication method. If the name request is always different the client needs a list of NetBIOS names of the network. The name that is requested is randomly chosen from the list. The NetBIOS session connection is also chosen randomly from the list. The shared key information that the client has is WINS server IP address, the list of the network's NetBIOS names. This method is still vulnerable for DoS attacks. For example the name request can be eavesdropped and changing the answer information to false causes denial of service, because if the client does not discover the name from its list it believes that the network is not trusted. MitM attack can be done to bluff that the network is trusted, if a hostile entity has enough information from the original trusted network.

This authentication method is much simpler to implement than the first NBT alternative, only WINS server is needed and a tailored application to the client. In most networks there are several WINS servers, so this is also solving the single point of failure threat.

The third alternative is authentication using NetBIOS scope ID. The idea was that the client and the network's other computers have a certain NetBIOS scopeID. ScopeID is configured from Windows registry by IT-support. If the client cannot establish a NetBIOS session to a certain name/computer, the network is untrusted. Table 2 is presenting the summary of NBT authentication security analysis.

TABLE 2. Security analysis of NBT authentication methods

Authentication method	MitM vulnerability	Replay attack vulnerability	DoS vulnerability	Encryption
NBT with beacon	yes	yes	yes	no
NBT and WINS	yes	yes	yes	no
NBT and scope ID	yes	yes	yes	no

There is no authentication between NBT packets, and NBT traffic is unencrypted. Thereby it is very easy to eavesdrop and capture this authentication information which is built over NBT services. Methods which are described above are vulnerable for man-in-the-middle attacks (middle person attack), replay attacks and denial of service attacks (DoS).

If NBT services are chosen for authentication process, there has to be an additional component, which encrypts the authentication traffic and authenticates peer computers. For example, IPSec connection to WINS server or certain host with certain scopeID. Windows XP has a VPN client and IPSec policy ("security rules and settings that control the flow of IP traffic inbound and outbound on a host") which can be created and managed centrally in Active Directory using Group Policy objects. Windows IKE can use The Kerberos version 5 authentication protocol, so the need for a PKI could be avoided, which is often difficult to deploy. When the client computer initiates NBT session to WINS server for name request, the client has IPSec policy assigned for this action. The outbound connection request triggers an IKE negotiation to the server. The client IKE obtains a Kerberos ticket to authenticate to the server. The example of the analogue method was explained in chapter "5.4.1 Server and domain isolation techniques". Table 3 is describing the improved NBT authentication security analysis.

TABLE 3. Improved NBT authentication

Authentication method	MitM vulnerability	Replay attack vulnerability	DoS vulnerability	Encryption
Improved NBT with IPSec policy	no	no	no	yes

Using just the NBT to implement an authentication method is not a rational idea because of the lack of security in NBT implementation. Using Windows group policy and IPSec improves the security and authentication, but this can be done for any protocol, so this is not a method just for NBT.

7.2 Security and implementation analysis of DHCP authentication method

Using DHCP protocol in the client's NAC implementation as an authentication method has also three different alternatives.

The first alternative was to use RFC 3118 DHCP authentication. The authentication is based on shared secret. MAC is computed from the message and the shared key (symmetric key) with HMAC-MD5 algorithm. The advantage of this is that the shared secret is never sent over the network as plaintext. The authentication option also includes a replay detection field which is basically a counter. This field mitigates replay attack threats. As explained in chapter "5.3 DHCP authentication", the first DHCP message the DHCPDISCOVER message is not authenticated. Hostile clients could flood the servers with DHCPDISCOVER messages which cause DoS vulnerability to the DHCP authentication. DoS vulnerability is not just with discover messages; also flooding the server with authenticated messages can prevent the server's normal functionality because of huge CPU load. CPU load increases because the authentication keys for the incoming messages are computed. Delayed authentication is also vulnerable for MitM attacks. If the symmetric key is too predictable or too short, it is quite easy to compute the key.

As explained in chapter 5.3, the problem in implementation of DHCP authentication is that Microsoft does not support the RFC 3118; therefore, the DHCP client and DHCP server have to be implemented by self. There also has to be an observing application which observes the network traffic and DHCP authentication messages to make decisions, whether the network is trusted.

In the **second alternative** the client uses the tool which can find rogue servers to authenticate the network. Dhcploc sends DHCPDISCOVER messages to the network and tries to find if there are rogue servers. Dhcploc tool is based on IP addresses, the trusted server IP address is configured to a tool and the tool searches other than trusted servers from the network with discover messages. Placing a rogue server to the trusted network causes a DoS attack because clients do not trust the network.

The implementation of this method needs a Dhcploc tool which has to be preconfigured and a certain application which observes the Dhcploc tool outcomes. This is quite easy to implement because Dhcploc tool is a command line tool, so with the command line script this tool could be utilized. Script outcomes are passed to the observing application which makes the decisions if the network is trusted.

In the **third alternative** the client has a key which is created from network settings and stored to client by offline. These kind of settings are DHCP server IP address, DNS server address and subnet address. The key is computed from these attributes. It has already been mentioned many times that there is no proper authentication and encryption in the original DHCP protocol; therefore it is easy to forge, capture and alter DHCP messages. Table 4 has a summary of security analysis of the DHCP authentication methods.

TABLE 4. Security analysis of DHCP authentication method

Authentication method	MitM vulnerability	Replay attack vulnerability	DoS vulnerability	Encryption
DHCP with RFC 3118	no (yes, if shared secret is exposed)	no	yes	Yes
DHCP and Dhcploc	yes	no	yes	no
DHCP setting information as a authentication key	yes	yes	yes	no

7.3 Security and implementation analysis of DNS authentication method

Windows XP's stub resolver does not support DNSSEC validation, but it still stores the RR information to its cache. The RR information that it stores to resolver's cache can be used to authenticate the network. The client's shared key in this case is security aware DNS server's public key information (DNSKEY resource record). When the client has queried certain name information from certain DNSSEC zone, the server responds with the answer (A resource record) and its digital signature (SIG resource

record). The client compares its DNSKEY information (shared key) to the received signature (SIG) information and if the signature is the right one, the network is trusted.

This method is forgeable, because the stub resolver and Windows 2003 server cannot set the DO, AD or CD bit in DNS query/response and they cannot make validation of key information. Copying the original SIG and using this same signature information in an untrusted network, clients believe that they are in a trusted network or in MitM attack forging the SIG prevents the client to join to the trusted network creates a DoS vulnerability. Although the method that is described above is forgeable by sending forged DNS responses, it is still hard to setup a bogus DNS server which has same information (signatures, keys) than a real DNS server. An untrusted network should have the same zone information for test.com zone, because signatures are basically made by hashing the zone data and signing it with a private key. Therefore a bogus server needs exactly the same zone information as the real server and same private key. It would be easier to eavesdrop traffic than set a bogus server and change the contents of the DNS packets by MitM attack, because stub resolver and Windows 2003 server cannot do validation of messages.

The same kind of DNSSEC authentication method with Windows 7 and Windows 2008 server would very liable and secure, because Windows 7 has a security-aware stub resolver and Windows 2008 server can do DNSSEC validations. The server makes a validation of RRs and the client authenticates with IPSec/IKE the server.

In chapter “4.2.3 Windows and DNSSEC” is described how Windows 7 utilizes the DNSSEC information. The client has a “Name Resolution Policy Table” which describes what kinds of procedures are done when querying certain DNS information. From this NRPT technique, we can conduct by using “Server and Domain Isolation Using IPSec and Group Policy” techniques and DNS protocol a very secure authentication method for a client’s NAC. The authentication method is described in chapter “5.4.1 Server and domain isolation techniques”. IPSec ensures that the authentication is secure and because DNS servers are configured to client’s network settings and there can be several servers, this method also ensures that there is no single point of failure in authentication and network access process.

Using DNSSEC information to authenticate the network, the network needs a security-aware DNS server and DNSSEC infrastructure. The authentication method created with IPSec and group policy can be done with infrastructure that already exist

and is more secure than DNSSEC authentication, if the client is Windows XP computer. In all authentication methods the client must have an application which observes DNS messages and it has to be functionality to send certain DNS queries to certain DNS server (observing module). Table 5 is representing the summary of DNS authentication security analysis.

TABLE 5. Security analysis of DNS authentication methods

Authentication method	MitM vulnerability	Replay attack vulnerability	DoS vulnerability	Encryption
DNSSEC with Win XP	yes	yes	yes	no
DNS with Win 7	no	no	no	Yes, partly encrypted
IPSec, Group Policy and DNS	no	no	no	Yes

7.4 Security and implementation analysis of IPSec, EAP-TLS and OE with certificate authentication method

Security protocols IPSec and TLS have the same idea for encrypting the traffic; both protocols use symmetric keys in encryption/decryption process. Still these protocols are developed for different purposes. IPSec encapsulates and secures IP packets between source and destination IP addresses. TLS operates at the transport layer in the OSI model and secures traffic between two applications.

Both authentication methods are highly secure because of certificates and encryption of the authentication traffic. During the IKE negotiation (phase 1) IPSec peers authenticate packets with digital signature. Both peers change each other's certificates to validate the digital signatures. Peers have also the trusted CA's certificates.

In EAP-TLS authentication the RADIUS server (authentication server) initiates TLS session with client (supplicant). The server sends its digital certificate to the supplicant, which the supplicant validates and vice versa. In the same way as in IKE authentication, both entities have trust the other's certificates. Both authentication methods are not vulnerable for attacks what are presented in this chapter.

In the EAP-TLS authentication only the authentication is encrypted and secured, in IPSec the idea is that all traffic is encrypted, not just the authentication. The demand for client's NAC implementation is that the access process which includes the authentication should be secure as possible; there is no need for further encryption.

Securely composed knowledge what is the reliability of the network is enough for the client's NAC. Therefore, the EAP-TLS provides enough security to authentication.

If there is a need for further encryption between nodes, it can be made for example with Opportunistic Encryption (OE). In this technique the goal is not the authentication, it is designed for securing the traffic with encryption. The idea is that unknown peers creates IPSec tunnel if it is possible ("Server Request Security") and keys are changed with IKE which queries them from security-aware DNS server. It is already noticed that, for example Windows XP computer cannot validate DNSSEC information, but in a network where the network is authenticated which is the main idea for client's NAC, the consumption is that network is trusted and its nodes are trusted. Client's NAC solution makes the OE much more secure, because the baseline is that all clients authenticate the network and clients which are in trusted network trust each other.

Security is similarly equal in these two authentication methods; thereby the implementation solves the difference between these two methods. Both need the PKI infrastructure, which can be difficult to implement. IPSec can be easily deployed in Windows environment, because Windows XP computers have a built-in IPSec client. The EAP TLS needs the 802.1X infrastructure which means that all of the switches in the network (or at least those that clients and servers connect to) must support 802.1X. The infrastructure needs to be fairly up-to-date. Therefore, the IPSec authentication with certificates is a better choice for a client's NAC solution. The IPSec authentication with certificates could be implemented in the same way that is presented in chapter 6.7 (IPSec, Kerberos and DNS). It ensures that there is no single point of failure in authentication infrastructure, because (larger) networks usually have several DNS servers. Table 6 presents summary of all authentication methods and their security analysis.

TABLE 6. Security analysis of all presented authentication methods

Authentication method	MitM vulnerability	Replay attack vulnerability	DoS vulnerability	Encryption
NBT with beacon	yes	yes	yes	no
NBT and WINS	yes	yes	yes	no
NBT and scope ID	yes	yes	yes	no
Improved NBT with IPSec policy	no	no	no	yes
DHCP with RFC 3118	no (yes, if shared secret is exposed)	no	yes	yes
DHCP and Dhcploc	yes	no	yes	no
DHCP setting information as a authentication key	yes	yes	yes	no
DNSSEC with Win XP	yes	yes	yes	no
DNS with Win 7	no	no	no	yes, partly encrypted
IPSec, Group Policy and DNS	no	no	no	yes
IPSec with certificate	no	no	no	yes
EAP-TLS with certificate	no	no	no	yes
Opportunistic Encryption	no	no	no	yes

8 CONCLUSIONS AND FUTURE WORK

The process of designing and maintaining security in an organization's networks is a classic IT challenge. The mobility of devices and the increasing amount of vulnerabilities in applications and network protocols creates these challenges. These vulnerabilities are discovered because security aspects have changed rapidly in past years. Fortunately, there are commercial concepts such as Network Access Control and Server and Domain Isolation to address these challenges. These concepts have technical differences, but the ideology is the same; the communication is based on trust. In both concepts a network has different security layers (figure 11) to which trusted and untrusted devices are pointed out depending on the outcome of the validation process. A network has equipment and different techniques to validate the devices which try to get access to the network's recourses. These concepts have the network's perspective.

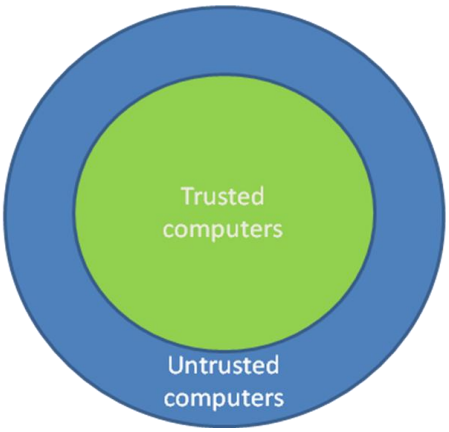


FIGURE 11. Different logical security layers

Server and Domain Isolation and Network Access control protect the network and its recourses. The protection is based on a validation process which ensures that only trusted devices get access to the network and its resources. The baseline for this research was that there are concepts and products which protect the network but there is no commercial solution which protects a single computer analogue to e.g. NAC. In this thesis, one of the main investigation issues was how it is possible to mitigate the threats directed to single a computer when it is connected to an unknown/untrusted network. This method was named **client's NAC**. The investigation was started by analyzing the client computer's network traffic.

8.1 Analysis of client's network traffic

The first logical step was to examine the client's communications flow towards the network. The purpose of this examination was to discover most common types of network traffic and find out those protocols and applications that generate traffic and should be secured from untrusted devices.

The traffic was captured in a test environment and the fundamental result was that, when Windows client is connected to a network is starts sending NBT traffic towards the network. This traffic is basically NetBIOS name registration traffic. The client also uses other protocols to communicate with other computers in Windows networks. These common protocols are: SMB protocol (Server Message Block), DNS protocol, ARP (Address Resolution Protocol) protocol and depending on IP address setup, the DHCP protocol. An example of a protocol capture can be found in appendix 1.

On a Windows LAN, it is common to have UDP ports 137, 138, and TCP port 139 enabled for NBT and TCP port 445 enabled for SMB. These ports provide NetBIOS

name resolution services, datagram services and session services in addition to other features. Unfortunately, these protocols also provide certain vulnerabilities. One example is null session exploit, which is described in chapter "3.6 Security threats in NBT ". An essential discovery was also the fact that NBT cannot be fully switched off from pure Windows 2000/2003 networks, because NetBIOS has so deep integration into Windows networks.

These findings strengthened the necessity for the solution which has analogue ideology than NAC has, but has a single computer's perspective. Because of the commonness of the DNS, DHCP and NBT protocols in Windows networks, these protocols were chosen for detailed examination and the goal was to find out if these protocols could be applicable for the client's NAC authentication process.

8.2 Controlling the client's joining to the network

Network protocols such as NBT, DHCP and DNS have vulnerabilities. When these protocols were implemented, there were not so many security threats on the Internet. Therefore, security aspects were not taken under consideration in the design phase of these protocols. Usually organizations block ports which these protocols use with a firewall on the perimeter of the network. With this traditional method organizations mitigate the threats which come from outside the organization's network – threats from the Internet. This does not prevent threats which come from the inside of the network. DNS and DHCP protocols have security extensions which have been developed because of their poor security. These extensions are DNSSEC and DHCP authentication which are described in RFCs 4033 and 3118. These methods include authentication and thereby they were also chosen to analysis if these extensions could be applicable for client's NAC authentication.

In the original Network Access Control or Server and Domain Isolation concepts the authentication is the most important function to validate if the device is trusted or not. There are also other targets which are validated such as absence of malware, updated malware prevention tools, patch management and specific firewall settings, still the authentication is most important function because it controls which devices have access to the network resources. Alike in the client's NAC solution the authentication is also the most important function to control client's joining to the network. The client has to validate if the network is trusted. The client's NAC has an opposing viewpoint to the original NAC concept. When NAC identifies trusted computers, the

client's NAC identifies a trusted network. If the client can make decisions about the connected network, it can be, for example sure that the DNS server or DHCP server are not rogue servers which offer forged information. Forged information can e.g. forward a client to a forged website, which can cause a worm infect for the client. The intelligence that offers for client the information from the network and how to make decisions is a combination of observing module (observing application) and authentication module. These are described in chapter "6.2 The basics of client's Network Access Control".

8.3 Choosing the best authentication methods for client's NAC

Information security is about tradeoffs between security and usability, therefore the environment defines how secure the implementation has to be. In this thesis the authentication methods for the client's NAC is divided based on the environment. In high secure network environment the security is the main factor when choosing the authentication method and in lower security networks the complexity of implementation is the main factor. Authentication methods which use basic network protocols (NBT, DHCP, DNS) and shared secret are directed to lower security networks and authentication by using certificates is directed to higher security networks. This has a one exception which is Opportunistic Encryption. OE includes IKE authentication with certificates, but because most client computers (operating systems) do not yet have validation functionality to validate DNSSEC information and they cannot set the DO bit to a DNS query for receive a DNSSEC resource records, the authentication is not trustworthy enough. Thereby, OE is directed to lower security networks.

This thesis introduced different authentication methods for the client's NAC solution. There were methods which were conducted from network protocols such as NBT, DHCP and DNS. Most of these authentication methods were insecure or the implementation was difficult.

By the result of security analysis and implementation analysis the best authentication method for the client's NAC implementation is **IPSec combined with Active Directory Group Policy, certificates and DNS**. This method brings security, liability and is easy to implement. Windows IKE provides Kerberos authentication and certificate authentication. With certificates the authentication process is securest, but Kerberos is secure enough. The problem in Kerberos authentication is that it needs an

Active Directory infrastructure to operate. If a network already has a PKI infrastructure (or planning to have) and Active Directory the recommendation is to use certificates, if there is not PKI infrastructure, Kerberos authentication is recommended. The implementation is easy because Windows has a built-in IPSec client. If the group policy and IPSec policy is made for DNS traffic, it ensures that there is no single point failure, because organizations usually have several DNS servers.

In the EAP-TLS method only the authentication is encrypted and secured, in the IPSec the idea is that all traffic is encrypted, not just the authentication. The demand for authentication for the client's NAC is that the client could securely compose knowledge what is the reliability of the network. The EAP-TLS brings equal security to authentication than IPSec, but the EAP-TLS authentication needs both 802.1X and the certificate infrastructure to operate; therefore it is more complicated to implement than IPSec with group policy. If an organization plans to have or have 802.1X and certificate infrastructure, then the recommendation is to use the EAP-TLS authentication in the client's NAC solution.

For the lower security networks e.g. public WLANs, the recommendation is to use Opportunistic Encryption. WLAN administrators can, for example set up a DHCP server or DNS server to a network configured to use OE. A client computer which is also configured to use OE creates IPSec connection between DHCP or DNS server. When using OE, the securest authentication method is X.509 certificates. If both peers trust each other's certificates, they also have trust the same CA. Therefore, administrators can give beforehand the CA certificate of the servers for the client.

If Thawte Freemail certificates (described in chapter "5.4.2 Opportunistic Encryption") are used in the authentication a disadvantage is that anybody can create a personal certificate from the website. At the end the trust is based only an email address, because the certificate is obtained with giving an email address; the certificate is bound an email address. This is still better than operating without any authentication. In the future when the client computers have security-aware stub resolvers, the authentication in OE is more trustful, because the clients can validate the DNSSEC information from the servers. For example when a Windows 7 becomes more common, OE is very considerable authentication method in public networks.

8.4 Future work

This thesis has research work and analyses how to develop client's NAC solution, which mitigates the threats which arise when a computer is connected to an untrusted network. The analyses concentrate on different authentication methods which is the most important function in client's NAC solution. The purpose of this thesis was to develop authentication methods from common network protocols and from known security protocols/architectures for the client's NAC. These authentication methods were analyzed from security and implementation perspective.

The best authentication method has been chosen by the result of security and implementation analysis. The basic functionality of client's NAC solution is also designed in this thesis. The next step in client's NAC development project is to implement a prototype solution and test the usability factors such as authentication time, how client's NAC is recovering from network disconnects and if the duplication of authentication servers prevents the single point of failure issue.

In future the goal is add more functionality to client's NAC solution. Examples of these functions are:

- Defense mechanism against worm infects
- Different trust levels

The worm infection in a computer usually causes abnormal behavior in the computer's data communication. The infected computer scans the Internet seeking for vulnerable applications on hosts to infect. The client's NAC solution could monitor the number of scans that a single computer sends out. If the amount of scans rapidly increases, the conclusion is that a computer has possible worm infection and data communication will be disabled.

Even better functionality would be if the computer could observe the data communication in the LAN and make decisions on ground of the LAN traffic. If the LAN has abnormal data communication, the client will disable its network interfaces.

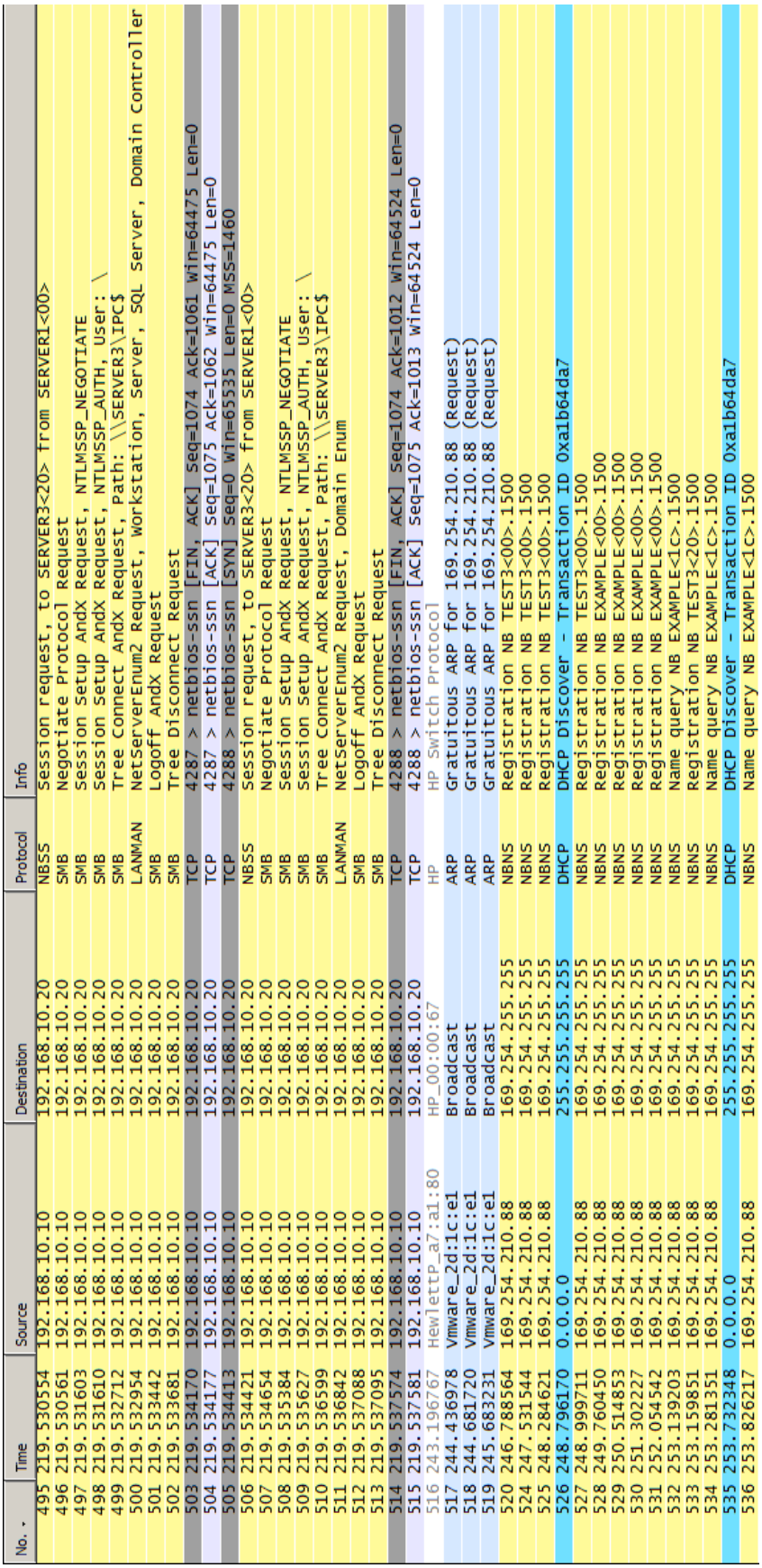
The different trust levels would be desirable functionality, when authenticating for example public WLANs. The client's NAC solution could have different trust levels to use, e.g. in the certain trust level only certain applications could communicate with the network's hosts. For instance, in this kind of case NBT communication could be

disabled or the client's NAC could, for example deploy certain firewall rules for the client's firewall. This functionality mitigates the threats which the untrusted computers and the LAN traffic create.

In future the best way to protect computers and networks is to combine NAC and client's NAC implementation. Client's NAC ensures that computers are accessed only by trusted networks and NAC ensures that the network is accessed only by trusted computers. Organizations can mitigate threats which coming from inside and outside the network, if the network is trusted from client's perspective and the network prevents unauthorized equipment from connecting to it.

APPENDICES

Appendix 1. The capture of common LAN traffic in Windows networks



No.	Time	Source	Destination	Protocol	Info
495	219.530554	192.168.10.10	192.168.10.20	NBSS	Session request, to SERVER3<20> from SERVER1<00>
496	219.530561	192.168.10.10	192.168.10.20	SMB	Negotiate Protocol Request
497	219.531603	192.168.10.10	192.168.10.20	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
498	219.531610	192.168.10.10	192.168.10.20	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: \
499	219.532712	192.168.10.10	192.168.10.20	SMB	Tree Connect AndX Request, Path: \\SERVER3\IPC$
500	219.532954	192.168.10.10	192.168.10.20	LANMAN	NetServerEnum2 Request, Workstation, Server, SQL Server, Domain Controller
501	219.533442	192.168.10.10	192.168.10.20	SMB	Logoff AndX Request
502	219.533681	192.168.10.10	192.168.10.20	SMB	Tree Disconnect Request
503	219.534170	192.168.10.10	192.168.10.20	TCP	4287 > netbios-ssn [FIN, ACK] Seq=1074 Ack=1061 Win=64475 Len=0
504	219.534177	192.168.10.10	192.168.10.20	TCP	4287 > netbios-ssn [ACK] Seq=1075 Ack=1062 Win=64475 Len=0
505	219.534413	192.168.10.10	192.168.10.20	TCP	4288 > netbios-ssn [SYN] Seq=0 Win=65535 Len=0 MSS=1460
506	219.534421	192.168.10.10	192.168.10.20	NBSS	Session request, to SERVER3<20> from SERVER1<00>
507	219.534654	192.168.10.10	192.168.10.20	SMB	Negotiate Protocol Request
508	219.535384	192.168.10.10	192.168.10.20	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
509	219.535627	192.168.10.10	192.168.10.20	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: \
510	219.536599	192.168.10.10	192.168.10.20	SMB	Tree Connect AndX Request, Path: \\SERVER3\IPC$
511	219.536842	192.168.10.10	192.168.10.20	LANMAN	NetServerEnum2 Request, Domain Enum
512	219.537088	192.168.10.10	192.168.10.20	SMB	Logoff AndX Request
513	219.537095	192.168.10.10	192.168.10.20	SMB	Tree Disconnect Request
514	219.537574	192.168.10.10	192.168.10.20	TCP	4288 > netbios-ssn [FIN, ACK] Seq=1074 Ack=1012 Win=64524 Len=0
515	219.537581	192.168.10.10	192.168.10.20	TCP	4288 > netbios-ssn [ACK] Seq=1075 Ack=1013 Win=64524 Len=0
516	243.196767	HewlettP_a7:a1:80	HP_00:00:67	HP	HP Switch Protocol
517	244.436978	Vmware_2d:1c:e1	Broadcast	ARP	Gratuitous ARP for 169.254.210.88 (Request)
518	244.681720	Vmware_2d:1c:e1	Broadcast	ARP	Gratuitous ARP for 169.254.210.88 (Request)
519	245.683231	Vmware_2d:1c:e1	Broadcast	ARP	Gratuitous ARP for 169.254.210.88 (Request)
520	246.788564	169.254.210.88	169.254.255.255	NBNS	Registration NB TEST3<00>.1500
524	247.531544	169.254.210.88	169.254.255.255	NBNS	Registration NB TEST3<00>.1500
525	248.284621	169.254.210.88	169.254.255.255	NBNS	Registration NB TEST3<00>.1500
526	248.796170	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xa1b64da7
527	248.999711	169.254.210.88	169.254.255.255	NBNS	Registration NB TEST3<00>.1500
528	249.760450	169.254.210.88	169.254.255.255	NBNS	Registration NB EXAMPLE<00>.1500
529	250.514853	169.254.210.88	169.254.255.255	NBNS	Registration NB EXAMPLE<00>.1500
530	251.302227	169.254.210.88	169.254.255.255	NBNS	Registration NB EXAMPLE<00>.1500
531	252.054542	169.254.210.88	169.254.255.255	NBNS	Registration NB EXAMPLE<00>.1500
532	253.139203	169.254.210.88	169.254.255.255	NBNS	Name query NB EXAMPLE<1c>.1500
533	253.159851	169.254.210.88	169.254.255.255	NBNS	Registration NB TEST3<20>.1500
534	253.281351	169.254.210.88	169.254.255.255	NBNS	Name query NB EXAMPLE<1c>.1500
535	253.732348	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xa1b64da7
536	253.826217	169.254.210.88	169.254.255.255	NBNS	Name query NB EXAMPLE<1c>.1500

REFERENCES

Administrator's Guide to Microsoft L2TP/IPSec VPN Client. 2002. Microsoft Corporation. 26.6.2002. Referred 20.3.2009. http://technet.microsoft.com/en-us/library/bb742553.aspx.

Albitz P & Liu C. 2006. DNS and BIND Edition: 5. O'Reilly Media, Inc.

Anderson R, 2001. Security Engineering: A Guide to Building Dependable Distributed Systems, The first edition. Wiley: Cambridge.

Configuring Null-Session Shares. 1999. LANDesk Management Suite: Referred 1.2.2009. http://download.landesk.com/support/mgtsuite6.62/shares/shares.doc.

Davidowicz D. 1999. Domain Name System (DNS) Security. Referred 29.4.2009. http://compsec101.antibozo.net/papers/dnssec/dnssec.html.

Domain Name System Security Extensions. 2009. An overview of Domain Name System (DNS) Security Extensions (DNSSEC) and information about how to deploy DNSSEC on the Windows Server 2008 R2 and Windows 7 operating systems. 5.2.2009. MS Word document downloaded 6.5.2009. http://www.microsoft.com/downloads.

Haden R. 1996. NetBIOS protocol, netbeui over TCP, server message blocks. Referred 12.1.2009. http://www.rhyshaden.com/netbios.htm.

How is information enumerated through NULL session access, Remote Procedure Calls and IPC$? 1999. SoftHeap.Com. Referred 2.2.2009. http://www.softheap.com/security/session-access.html.

IPC share exploit. Technology News, Hacking article. Referred 1.2.2009. (http://www.datastronghold.com/hacking-articles/733-ipc-share-exploit.

IPv6 Security Considerations and Recommendations. Published: February 27, 2006 | Updated: February 11, 2008. Referred 14.5.2009. http://technet.microsoft.com/en-us/library/bb726956.aspx.

Kozierok C. 2001. The TCP/IP Guide - DHCP Security Issues. Referred 18.4.2009. http://www.tcpipguide.com/free/t_DHCPSecurityIssues.htm.

Kurose J & Ross K. 2008. Computer Networking, 4th Edition. Boston: Pearson Education Inc.

Käyttäjien tunnistaminen ja PKI. 2007. Teleware's User identification and PKI course training material.

Modify DNSSEC configuration. Updated: January 21, 2005. http://technet.microsoft.com/en-us/library/cc779943(WS.10).aspx.

MS00-047. 2007. NetBIOS Vulnerability May Cause Duplicate Name on the Network Conflicts. Referred 12.5.2009. http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/q269/2/39.asp&NoWebContent=1.

Olzak T. 2007. The problem with NetBIOS. 26.3.2007. Referred 31.1.2009. http://blogs.techrepublic.com.com/security/?p=196.

Operating System Market Share. 2009. April, 2009. Referred 15.5.2009. http://marketshare.hitslink.com/report.aspx?qprid=8.

Reinhold A. 2008. Network Access Control for Dummies. Wiley Publishing Inc.

RFC 1001. 1987. Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods. March, 1987. Referred 25.1.2009. http://tools.ietf.org/html/rfc1001.

RFC 1533. 1993. DHCP Options and BOOTP Vendor Extensions. http://www.ietf.org/rfc/rfc1533.txt.

RFC 2131. 1997. Dynamic Host Configuration Protocol. March 1997. Referred 14.4.2009. http://www.ietf.org/rfc/rfc2131.txt.

RFC 2401. 1998. Security Architecture for the Internet Protocol. http://www.ietf.org/rfc/rfc2401.txt.

RFC 2409. 1998. The Internet Key Exchange (IKE). http://www.ietf.org/rfc/rfc2409.txt.

RFC 3118. 2001. Authentication for DHCP Messages. June 2001. Referred 3.5.2009. http://tools.ietf.org/html/rfc3118#ref-3.

RFC 3579. 2003. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). Septemper 2003. Referred 8.4.2009. http://www.ietf.org/rfc/rfc3579.txt.

RFC 3655. 2003. Redefinition of DNS Authenticated Data (AD) bit. November 2003. Referred 6.5.2009. http://www.ietf.org/rfc/rfc3655.txt.

RFC 4035. 2005. Protocol Modifications for the DNS Security Extensions. March 2005. Referred 6.5.2009. http://www.ietf.org/rfc/rfc4035.txt.

RFC 4322. 2005. Opportunistic Encryption using the Internet Key Exchange (IKE). http://www.ietf.org/rfc/rfc4322.txt.

RFC 5216. 2008. The EAP-TLS Authentication Protocol. March 2008. Referred 29.3.2009. http://www.faqs.org/rfcs/rfc5216.html.

Securing Windows 2000 Server. 2004. Referred 13.5.2009. http://technet.microsoft.com/fi-fi/library/cc751212(en-us).aspx.

Server and Domain Isolation Using IPSec and Group Policy. 2006. Microsoft Solutions for Security and Compliance. Downloaded 13.5.2009. http://www.microsoft.com/downloads/details.aspx?FamilyId=404FB62F-7CF7-48B5-A820-B881F63BC005&displaylang=en.

Seshadri S. 2008. DNSSEC on Windows 7 DNS client. 11.11.2008. Referred 6.5.2009. http://blogs.technet.com/sseshad/archive/2008/11/11/dnssec-on-windows-7-dns-client.aspx.

Steps to turn on optional IPsec on a Windows XP computer. 2006. Referred 17.5.2009. http://slashdot.org/~cronscript/journal/131319.

Szymanski P. Implementation of the IPSec Protocol in Microsoft Windows 2003/XP Environment. Document downloaded 13.5.2009. http://www.scribd.com/doc/8255890/Implementation-of-the-IPSec-Protocol-in-Microsoft-Windows-2003XP-Environment-.

TCP/IP Fundamentals for Microsoft Windows. 2005. Chapter 11 - NetBIOS over TCP/IP. Microsoft Corporation. 27.6.2005. Updated 18.4.2006. Referred 19.1.2009. http://technet.microsoft.com/en-us/library/bb727013.aspx.

Ts J, Eckstein R, & Collier-Brown D. 2003 Using Samba 2nd Edition. O'Reilly media.

Tulloch M. 2004. Understanding NetBIOS and Windows Server 2003. 11.5.2004. Referred 24.1.2009. http://www.windowsdevcenter.com/pub/a/windows/2004/05/11/netbios.html.

Työasemat tarkastukseen. 2008. Tietokone magazine, march 2008.

Understanding 802.1X authentication for wireless networks. 2005. Microsoft Corporation. 21.1.2005. Referred 22.3.2009. http://technet.microsoft.com/en-us/library/cc759077.aspx.

Using DNS Security Extensions (DNSSEC). 2005. Microsoft Corporation. 21.1.2005. Referred 6.5.2009. http://technet.microsoft.com/en-us/library/cc728328.aspx.

Zwicky E. D, Cooper S, Chapman B D. 2000. Building Internet Firewalls 2nd Edition. O'Reilly media.