Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp277-282)

Testing the Tools for IPv6 Traffic Tunneling

Drago Žagar, Goran Martinović and Snježana Rimac-Drlje
University of Osijek
Faculty of Electrical Engineering
Kneza Trpimira 2b , HR-31000 Osijek,
CROATIA

Abstract--. Through the years of existence of IPv4 protocol, security was a major problem. Therefore, with the development of new IPv6 protocol, lots of attention was given to security of communication that is built in protocol itself. IPv6 provides many new possibilities and features and also enables significant improvements in confidentiality of information transmitted through the network. During the IPv4/IPv6 transition period we are using mechanism of encapsulation. To connect isolated IPv6 islands, we must use 6to4 mechanism; tunnelling of IPv6 packets through IPv4 network. Characteristics of such communications open a lot of security threats; mainly DDoS attacks on IPv4 and IPv6. This paper analyses mechanisms for transition from Ipv4 to Ipv6. The tools for traffic tunnelling were tested, and some integral transition solutions were given.

1. INTRODUCTION

A new version of the Internet protocol IPv6 will very soon replaces an old IPv4 protocol. IPv6 brings many new improvements especially in simplicity, routing speed, quality of service and security. A new network protocol, IPv6 enables flexible use of extended headers, as well as an IPSec protocol. In spite of these improvements, it is still necessary to take care of network security [5][6][7].

A transition period from IPv4 to IPv6 require some coexistence mechanisms. Dual stack is one of the simplest methods for introducing the IPv6 in Internet [1]. Dual stack protocol maintains both IPv4/IPv6 addresses and can communicate with all IPv4/IPv6 network nodes. The second transition mechanism is traffic encapsulation. In the first stage of IPv6 implementation this mechanism enables encapsulation of IPv6 traffic through the IPv4 Internet. In an advanced stage of the transition process the encapsulation will interconnect the remained IPv4 networks over the IPv6 Internet.

The new network protocol introduces some improved security mechanisms, but it is still possible to misuse it. That is often a consequence of inadequate knowledge of network administrators concerning the new protocol and ignorance of possible misuses. The security issues are especially emphasized in a period of coexistence of IPv4 and IPv6, while the transition mechanisms open new and till now unknown possibilities of an unauthorized invasion. Usually, IPv6 traffic is tunneled through the IPv4 network without knowledge of the network administrator. It is therefore extremely important to take care of network security, and to properly implement the protecting mechanisms.

2. TRAFFIC TUNNELING

In a period of coexistence of both IPv4 and IPv6 protocols we use a mechanism of traffic tunneling. Tunneling techniques are usually classified according to a mechanism by which the encapsulating node detects an address of the end node in the tunnel. In the first two methods of tunneling *router-to-router* (Figure 1.) and *host-to-router* (Figure 2.) IPv6 packets are tunneled to a router. The end point of the tunnel is proxy router that has to encapsulate IPv6 packets and forward them to a final destination. Therefore, the address of the IPv6 packet that is tunneled cannot provide an IPv4 address of a tunnel end point.

Instead of that, the end point of the tunnel should be assessed by the information from a configuration node performing the tunneling. The last two methods for tunneling *host-to-host* (Figure 3.) and *router-to-host* perform IPv6 packet tunneling for a whole end-to-end data path. In this case the destination address of the IPv6 packet and an encapsulated IPv4 header define the same node. An automated tunneling technique uses a special format of the IPv6 address with built in IPv4 address to enable the tunneling nodes to obtain IPv4 address of the tunnel end point [1][2][3][4].

This paper is a part of the CAR6Net project 100-503/04-01/5

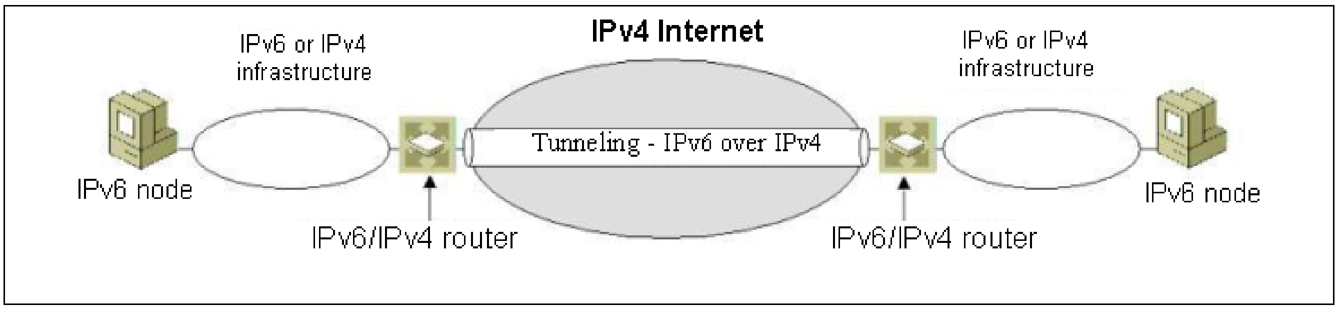
IPv4 Internet
IPv6 or IPv4 infrastructure
IPv6 or IPv4 infrastructure
Tunneling - IPv6 over IPv4
IPv6 node
IPv6 node
IPv6/IPv4 router
IPv6/IPv4 router

Figure 1. Router-to-Router tunneling

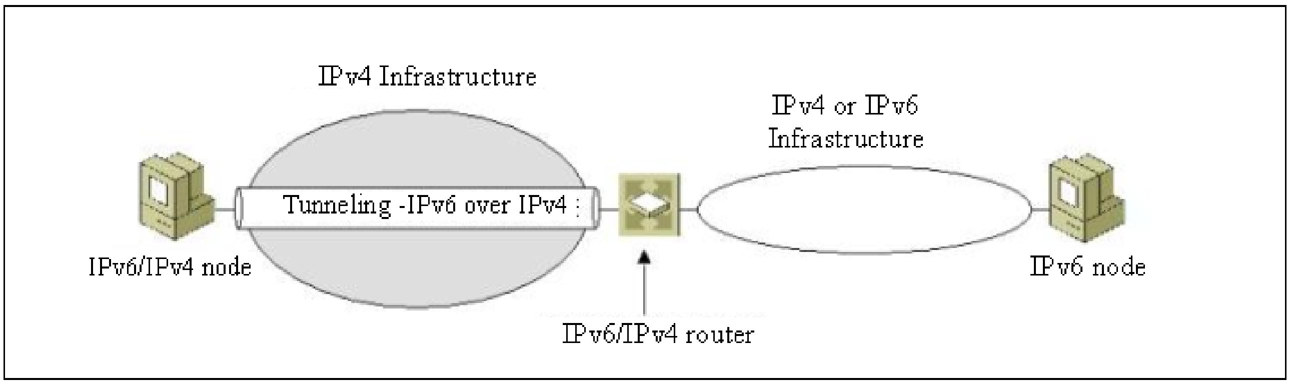

Figure 2. Host-to-Router and Router-to-Host tunneling

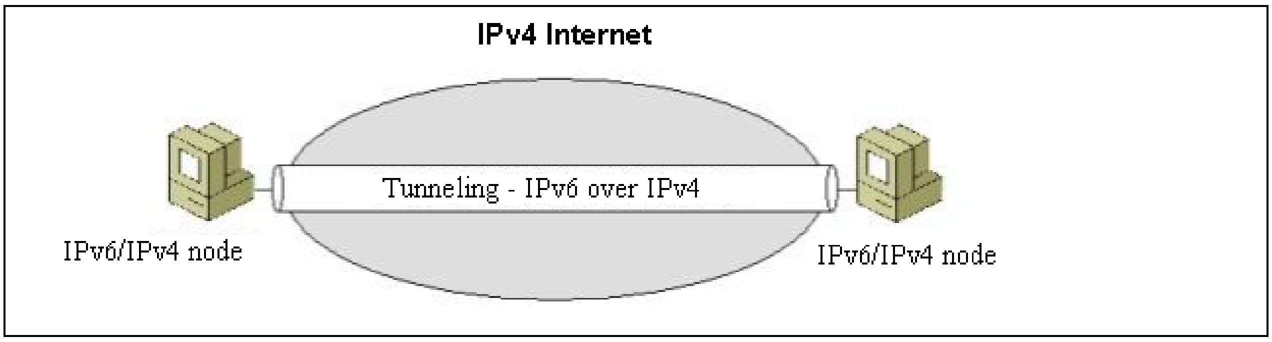

Figure 3. Host-to-Host tunneling

By this technique it is not necessary to explicitely configure the end-point addresses, and therefore significantly simplifies the whole configuration.

3. TESTING NETWORK

For an experimental setup at the Faculty of Electrical Engineering, University of Osijek, a new IPv6 network was implemented, consisting of three computers (built on Intel P4 CPU), one driven by Linux Gentoo OS (Kernel 2.4.25), and two by WinXP OS SP 1 (Figure 4.). Besides that, we have made testing on one computer in SRCE (University Computing Center) in Zagreb, driven by Linux Debian OS Kernel 2.4.26. A connection between IPv4 and IPv6 networks was implemented by a router device with IPv6 characteristics. All computers in a local IPv6 network were configured as “dual stack” devices. The implemented IPv6 router connected the local IPv6 network with the IPv4 Internet and the IPv6 network (CAR6Net – Carnet IPv6 network) [8].

In the experimental IPv6 network we have tested the available freeware (open source) transition tools for the IPv4/IPv6 protocol.

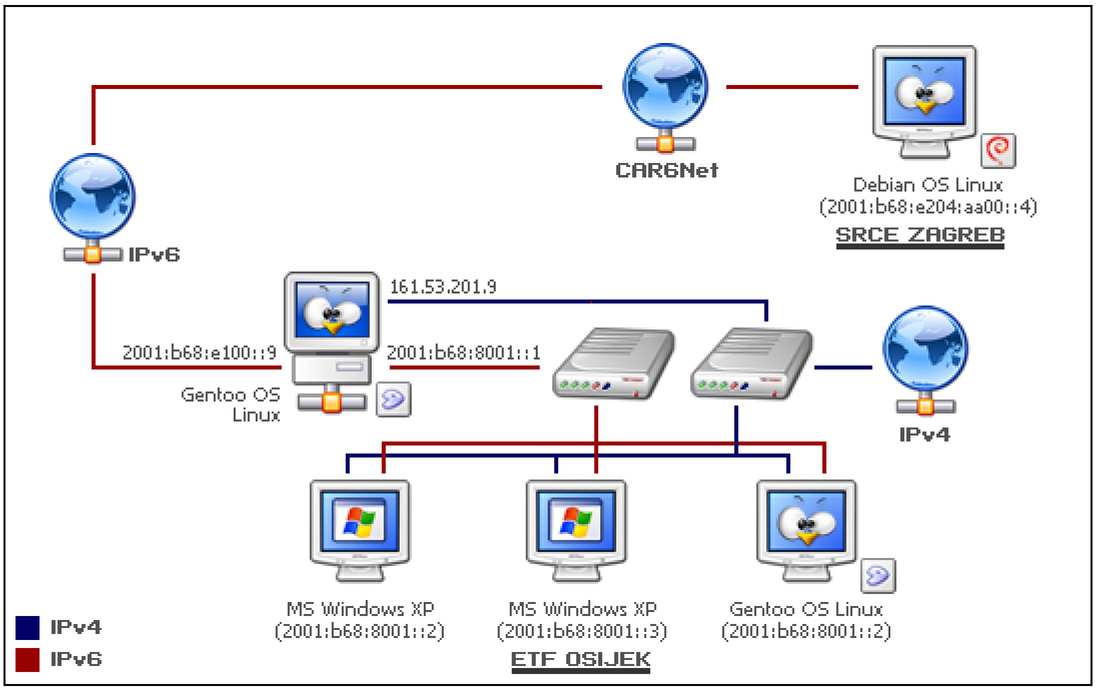

Figure 4. Experimental IPv6 network

4. TOOLS FOR IPv6 TUNNELING

From a set of tools appropriate for the traffic tunneling we have tested the available freeware tools: ***asybo***, ***relay6*** and ***nt6tunnel***.

nt6tunnel is a tool that enables conversion of the IPv4 packets into IPv6 and vice versa. *nt6tunnel* does not have a graphical interface and is a shell application. This tool uses some basic parameters, as *localport* and *remotehost*, while the optional parameters include one specific parameter for this tool, *proxy* configuration, by which we could use the password. One of the disadvantages of the *nt6tunnel* is that a user is not informed about tunnel establishment and the only way to check it is to tunnel some traffic through the network.

AsyBoV6 is a *bouncer* from IPv4 to IPv6 (listen to IPv4 and for every opened connection creates the data tunnel from IPv4 to IPv6). It is used for the old IPv4 applications over the IPv6 network (when we want an output with IPv6 address).
Old apllication (IPv4) → Asybov6 6to4 (IPv6) → *Gateway* (4(6)-tunneling) → *Relay router* (IPv6) → 6bone → Destination

Relay6 is also a *bouncer,* the application that forwards incoming connections to a defined destination:

Application → relay6 → *remote service*

Moreover, *Relay6* enables conversion from IPv4 to IPv6 packets. That means that we can use the applications created only for IPv4 also in the IPv6 network. *Relay6* operates with any type of the TCP application, actually a packet content is not changed, it makes a conversion from IPv* to IPv*.

Accordingly, we could perform not only IPv4→IPv6, but also IPv4→IPv4, IPv6→IPv4 and IPv6→IPv6.

To establish the connection we have to configure the basic parameters, as *Local Address*, *Local Port, Remote Address* i *Remote Port*. IPv6 address 2001:b68:8001::2 and IPv4 address 161.53.201.163 are addreses configured by *dual stack* computer, driven by Windows XP OS, while 2001:b68:8001::4 is an address of the computer driven by Linux Gentoo OS.

The basic characteristics of the tools used for testing is that all support basic functions, and differ by some additional options. The basic parameters of the tested tools are *local_port*, ie. the port waiting for connection (listening), *remote_host* and *remote_port* parametar.

5. THE TEST RESULTS

For the purpose of testing the tools for traffic tunneling we have used the *Nmap* software. When the tunnel was established, the tools aborted the connection as soon as they were scanned. While the tool listens the specific port waiting for the connection we could assess the opened port. This could become a potential security problem. The active connections are not exposed to this type of attack.

On addition to the basic operation set, the *nt6tunnel* provides an additional option, a possibility to use a password when the tool acts as an IRC *proxy*. The main disadvantage of the *nt6tunel* is its vulnerability to DDOS (*Distributed Denial of Service*) attack. When a connection is interrupted irregularly, a server does not disconnect clearly. Even when the connection is disconnected regularly, the used ports remain opened for a short time. That could enable an attacker to flood the server with a huge number of connection requests that could bring an application to fall down.

Asybo is a tool with a user-friendly graphical interface. It offers a control over the established connections and opened ports, i.e. listening port. By choosing an IANA well-known ports as listening port AsyBoV6 will notice the user. The main disadvantage of this tool is that it cannot work behind the firewall and it can tunnel only IPv4 packets over the IPv6 network, while the other combinations are currently not possible.

The Relay6 comprises some combination of the former tools. On addition to the basic parameters, it has some additional options, one of important ones being a possibility to create its own firewall. Beside the basic tool i.e. "Shell" application, we can use a Frontend version with very articulated and user friendly graphical interface which could relieve the work and owerview of the established connections. Outside the basic parameters: *local_port* which determinates the port on which Relay6 listens, *remote_host* which determinates *remote-server* on which the Relay6 application reroutes the requests for connection and *remote-port*; we could have several optional parameters. The main optional parameter is possibility to create of a Firewall which consist of some simple rules. Basically, it is an ASCII file in which every single line represents the rule consisting of three parts, devided by one or more spaces:

incoming IPv* *bind* IPv* *bind port*

Waching the IPv* the Firewall decides which client could be connected to Relay6. We could, also, assess the output port which was allocated to accepted client, however the preferred output IPv*. If the Relay6 listens for all ports it is very important to decide who is allowed to connect on which port(s). Besides that, the Firewall could bind IPv*/port for every accepted client. This is defined by the Firewall rules: when the client connects to Relay6, Relay6 asks its IPv*. If its address is allowed, Relay6 accepts the client and defines the connecting rules. Otherwise, Relay6 disconnect the connection.

For example:

```
*firewall.txt*
161.53.201.163
189.230.21.3
client01.foo.net
192.168.1.3
151.251.4.67
```

client 192.168.0.2 will be rejected,
client 161.53.201.163 will be accepted.

The incoming IPv* could be determinated numerically or by the name (including DNS). It is recommended to use numerical form.

TABLE I. CHARACTERISTICS OF IPV6 TESTING TOOLS

Criterion/Tool	*AsyBoV6*	*Relay6*	*nt6tunnel*
Platform	Win XP	Win XP	Win XP
Interface	Frontend	Shell/Frontend	Shell
User friendly	High	Middle/High	Low
Type	Bouncer	Only TCP bouncer	Bouncer
Tunneling	IPv4→IPv6	IPv*→IPv*	IPv*→IPv*
Work behind firewall	No	Yes	Yes
Immunity to attacks	Middle	Middle/High	Low/Middle
Options	-	Own firewall	Authentication

The Table I. sumarises the the characteristics of the tested tools.

According to the authors knowledge there exist no freeware tools that could detect IPv6 traffic encapsulated in the IPv4 network. We have used a tool Ethereal (Open source packet sniffer and analyzer) with complete IPv6 support and did not detect the exact source of the packets neither that IPv6 was encapsulated in the IPv4 packet. In the test network we could encapsulate anything and thus potentially attack a destination host. To prevent possible incidents it is desirable to implement some of the commercial tools that could detect the encapsulation (e.g. RealSecure).

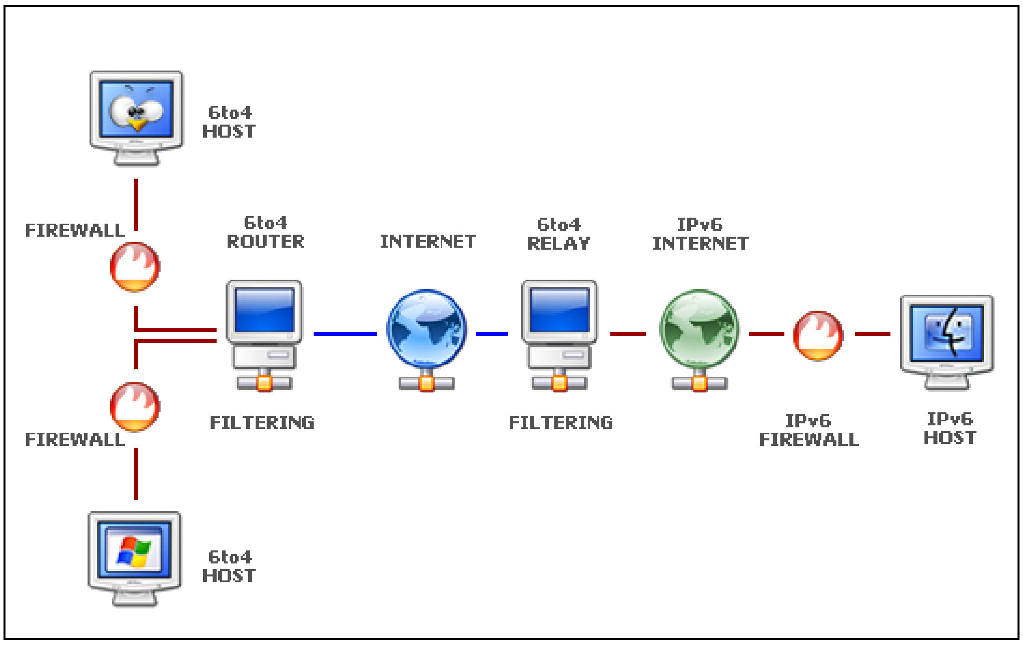

Figure 5. The transition network and possible protection

6. NETWORK PROTECTION IN TRANSITION PERIOD

The most proper solution for the period of transition IPv4→IPv6 will be the use of ***dual stack*** mechanism**.** By using dual stack devices, the attack could be directed to two different protocols and therefore it is much harder to assert different attacks. If some device is configured as a dual stack and connected to the IPv6 network it does not represent some security problem. The most important rule by the dual stack is that IPv6 security setup should practically be a mirrored security setup of IPv4. For the applications that should be secure we should use some cryptography methods to disable packet spoofing. The positions where some kinds of protection should be implemented are shown in Figure 5.

Outside the dual stack we could use traffic tunneling performed by available tools. On the basis of testing results we can conclude that the most appropriate tunneling tool is Realy6, which could set its own firewall. By the tunneling IPv6 over the IPv4 networks problem could be opened security. Most of the firewalls will register that as the normal IPv4 traffic, and so an arbitrary malicious load could seriously harm the network. To disable such type of traffic we have to filter packets with protocol label 41 in an IPv4 header. To increase a security level it is also desirable to put the IPv6 firewall at the end of the tunnel to check the type of load after decapsulation. By introducing a new IPv6 protocol we use some new types of ICMP messages. To protect the network from DDOS attacks (or to decrease the probability of the attack) we have to decide which types of ICMP messages are necessary and then filter the others by using **ip6tables** tool.

7. CONCLUSION

The new network protocol, IPv6, will directly or indirectly improve security surrounding. However, an implementation of the IPv6 protocol cannot be a solution to all security problems, primarily because of that it brings some new own security problems. Most of them are connected with transition from IPv4 to IPv6.

By transition from IPv4 to IPv6 it is necessary to use dual stack devices as well as static tunneling. However, that brings some additional security threats and the network administrators have to establish a secure connection between tunnel end points as well as to implement different interior and exterior security measures. On the basis of testing results the available tunneling tools we can conclude that the appropriate solution could be Realy6. To prevent possible security incidents it is desirable to implement some of the commercial tools that could detect the encapsulation.

By using *dual stack* device the attacker could attack two different protocols, and it is harder to defend to different types of attacks. Therefore, dual stack is more wulnerable than one using only IPv4 or IPv6 protocol, and currently the support to native IPv6 accordingb to security is not satisfactory. The fact that some network device is configured as dual stack and connected to IPv6 network is not security problem, but we shall use to separated IP structures by which we have to synchronise the security measures. The most important rule, by using the dual stack devices, is to mirror security parameters and security setup used by IPv4. Every Firewall rule and every access list constraining the access to some users must be translated to the appropriate rule and access list by IPv6. It is very important to avoid the implementation of dual stack device offering different characteristics for IPv4 and IPv6. For applications that should be secure we have to use some of cryptographic methods, thus disabling the packet spoofing.

REFERENCES

[1] RFC 1933: Transition Mechanisms for IPv6 Hosts and Routers

[2] RFC 2529: Transmission of IPv6 preko IPv4 Domains without Explicit Tunnels

[3] RFC 1853: IP in IP Tunneling

[4] RFC 3056: Connection of IPv6 Domains via IPv4 Clouds

[5] RFC 2402: IP Authentication Header (AH)

[6] RFC 2406: IP Encapsualtion Security Payload (ESP)

[7] RFC 2460: Internet Protocol, Version 6 (IPv6) Specification

[8] Žagar D., S. Vidaković, IPv6 Security: Improvements and Implementation Aspects, *Proceedings of the 8th International Conference on Telecommunications, Contel 2005.,*Zagreb, Croatia, June 15.-17. 2005.