

IPv6 and Security

Kostya Kortchinsky

CERT Renater

DRAFT

Agenda

IPv4 & IPv6 Nodes

- **IPv4 & IPv6 Nodes**

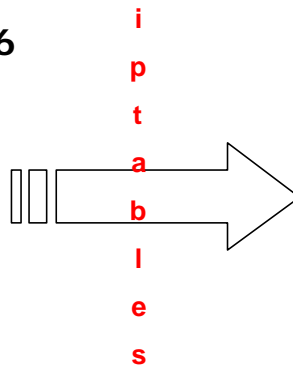
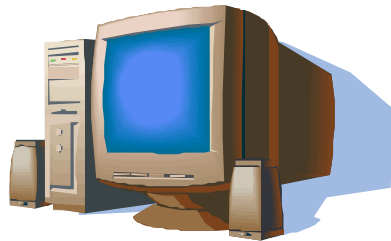
- The essential requirement for a successful transition from IPv4 to IPv6 is to maintain compatibility with IPv4 while implementing IPv6. The following mechanisms are employed by the IPv6 hosts and routers that need to interoperate with IPv4 hosts and utilise IPv4 routing infrastructures.
 - 1. Dual IP layer - Providing complete support for both IPv4 and IPv6 in hosts and routers.
 - 2. IPv6 over IPv4 tunneling - Encapsulating IPv6 packets within IPv4 headers to carry them over IPv4 routing infrastructures. Two types of tunneling are employed: configured and automatic.
- Unfortunately the previous mechanisms are likely to introduce on existing (and yet secured) networks new access vulnerabilities.
- One can't content himself with installing an IPv6 stack on a node, he must ensure that the newly configured host complies with an appropriate security policy.

- **Demonstration**

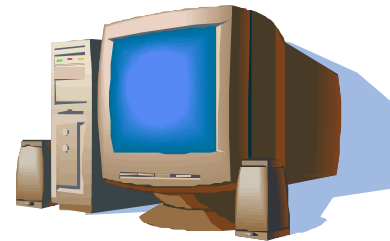
- 1. We will study briefly the case of two Linux workstations with newly configured IPv6 and IPv4 network security provided by iptables.
- 2. We will then study the case of a Linux and a FreeBSD workstation, with network security provided by some router's access list.

IPv6 Example: Host Based Filtering

Debian SID – IPv4 & IPv6



Debian SID – IPv4 & IPv6



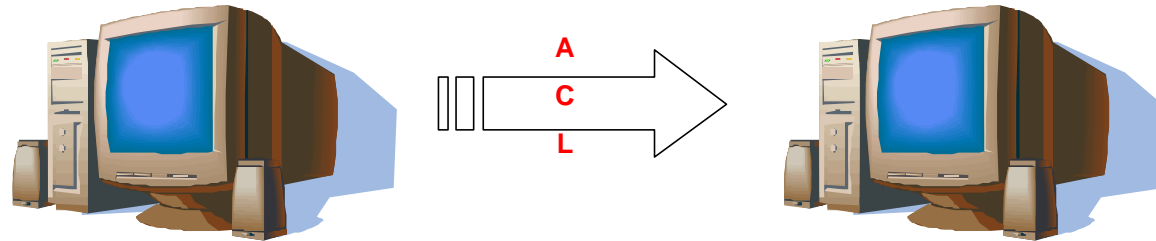
```
cert@Sisyphe:~$ nmap -sT -P0 195.98.xxx.xxx
Starting nmap V. 3.10ALPHA3 ( www.insecure.org/nmap )
Interesting ports on 195.98.xxx.xxx:
(The 1601 ports scanned but not shown below are in state: filtered)
Port      State      Service
21/tcp    open      ftp
22/tcp    open      ssh
113/tcp   closed    auth

cert@Sisyphe:~$ nmap -6 -sT -P0 2001:660:xxx:xxx::xxx
Starting nmap V. 3.10ALPHA3 ( www.insecure.org/nmap )
Interesting ports on 2001:660:xxx:xxx::xxx:
(The 1601 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open      ftp
22/tcp    open      ssh
23/tcp    open      telnet
```

IPv6 Example: Router Based Filtering

Debian SID – IPv4 & IPv6

FreeBSD 4.5 – IPv4 & IPv6



```
cert@Sisyphe:~$ nmap -sT -P0 193.49.xxx.xxx
Starting nmap V. 3.10ALPHA3 ( www.insecure.org/nmap )
Interesting ports on 193.49.xxx.xxx:
(The 1602 ports scanned but not shown below are in state: filtered)
Port      State      Service
41/tcp    closed    graphics
53/tcp    closed    domain

cert@Sisyphe:~$ nmap -6 -sT -P0 2001:660:xxx:xxx::xxx
Starting nmap V. 3.10ALPHA3 ( www.insecure.org/nmap )
Interesting ports on 2001:660:xxx:xxx::xxx:
(The 1602 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open       ssh
111/tcp   open       sunrpc
```

IPv6 Example: Solutions

Linux: iptables

```
iptables -A INPUT -i eth0 -p tcp -s 2001:660:xxx:xxx::/64 --dport 22 -j ACCEPT
```

```
iptables -A OUTPUT -o eth0 -p tcp -d 2001:660:xxx:xxx::/64 --sport 22 ! --syn -j  
ACCEPT
```

FreeBSD: ip6fw

```
ip6fw add pass tcp 2001:660:xxx:xxx::/64 any to ${ip} 22 setup in via ${if}
```

```
ip6fw add pass tcp from any to any established
```

```
ip6fw add pass all from ${ip} to any out via ${if}
```

tcp_wrapper

```
hosts.allow: sshd: [2001:660:xxx:xxx::]/64
```

```
hosts.deny: ALL: ALL
```

Logs

```
Oct 15 10:31:20 Icare sshd[4503]:
```

```
Refused connect from 2001:660:yyy:yyy::1
```

```
Oct 15 10:55:37 Icare sshd[4560]:
```

```
Connection from 2001:660:xxx:xxx::1 port 53435
```

```
Oct 15 10:55:38 Icare sshd[4560]:
```

```
Accepted password for root from 2001:660:xxx:xxx::1 port 53435 ssh2
```

Possible Abuse Against IPv6 Transition Technologies

- **IPv4 compatible address (::/96)**
 - To implement automatic tunneling, IPv4 compatible address (::127.0.0.1) are used. From IPv6 stack point of view, an IPv4 compatible address is considered as a normal unicast address.
 - If an IPv6 packet has IPv4 compatible address in the header, the packet will be encapsulated automatically into an IPv4 packet, with IPv4 address taken from lowermost 4 bytes of the IPv4 compatible address. Since there is no good way to check if embedded IPv4 address is sane, improper IPv4 packet can be generated as a result.
 - Malicious party can abuse it, by injecting IPv6 packets to an IPv4/v6 dual stack node with certain IPv6 source address, to cause unexpected IPv4 packets.
- **IPv4 mapped address (::ffff:0.0.0.0/96)**
 - IPv4 mapped address is used to handle inbound IPv4 traffic toward AF_INET6 sockets, and outbound IPv4 traffic from AF_INET6 sockets.
 - When we have an AF_INET6 socket bound to IPv6 unspecified address (:::), IPv4 traffic, as well as IPv6 traffic will be captured by the socket. The kernel will present the address of the IPv4 peer to the user land program by using IPv4 mapped address. The user land program can manipulate IPv4 mapped address just like it would do against normal IPv6 unicast address.
 - Malicious party may be able to use the IPv6 packets with IPv4 mapped address, to bypass access control, or generate unexpected IPv4 traffic.