

SECURITY OF IPV6: FROM FIREWALLS POINT OF VIEW

(EXTENDED ABSTRACT)

Mr. János Mohácsi , Research Associate, NIIF/HUNGARNET, H-1132 Budapest, Victor
Hugo u. 18-22.

E-mail: mohacsi@niif.hu

Tel: +36 1 4503081

Fax: +36 1 3506750

1 INTRODUCTION

The recent growth of IPv6 usage had made unavoidable to analyse whether the IPv6 without IPSec can provide enough security to communicate via IPv6. This analysis is also important since the application of IPSec on the Internet is relatively scarce, and probably will be limited, however IPSec itself providing a good, modular framework. The other security approach, the firewall, becomes building block of each IP network. This paper tries to analyse what is available and missing for IPv6 firewalling. This work is partially presented for the TF-NGN/GÉANT community. On the conference extended survey results and performance results will be also presented.

2 THE IPV6 FIREWALLS

IPv6 has quite widespread support in the level different applications and also in the level of vendors.. However the IPv6 mandates implementation of IPSec on a standard compliant IPv6 implementations, complete IPv6/IPSec solution is quite rare [KAME][USAGI][SUNIPV6]. The demand of secure using of IPv6 could be also fulfilled, with a popular 'workaround' called IP firewall, that is very widely used in IPv4, and one of the favourite tools of network/security managers.

The Internet firewall is a system, that implements and enforces the security policy between two network: usually protects and internal private network (Intranet) from external Internet.

Internet firewall can be implemented for IPv6 also, considering the following differences from IPv4:

- IPv6 does not require Network Address Translation to support address conservation
- IPv6 firewalls should support IPv6 chaining header structure
- IPv6 network scanning virtually impossible due to a 64 bit interface identifier that are almost pseudo random
- IPv6 firewalls should support IPv4/IPv6 transition and coexistence
- IPv6 firewalls should not break the IPv4 security

There also some new threats for IPv6 that has to be investigated by firewall implementers:

- Some IPv4-IPv6 transition mechanisms requires tunnelling, that can negatively impact security, as envisaged in draft of Pekka Savola [SAVOLA]
- Usage of IPv4 compatible addresses can induce security problem

3 EVALUATION OF IPV6 FIREWALLS

After looking at the available IPv6 firewalls we found to be only few are generally available: ip6fw [KAME], ip_filter [IP_FILTER], packet filter [OPENBSDPF] Cisco [CISCO], netfilter [NETFILTER]. The result of evaluation is summarised in the following table as November 2003:

	ip_filter 3.4.31	ip6fw 2003/11/15	Cisco IOS 12.3(4)T	Netfilter 2.4.x
Type of Firewall (packet filtering / application gateway / circuit gateway)	Packet filter with advanced state control	Packet filter with basic state control	Advanced packet filtering, simple since 12.2(2)T	Modular packet filter with advanced state control
Platform availability	FreeBSD / NetBSD / OpenBSD / Solaris	FreeBSD/ NetBSD / OpenBSD / BSDI	Cisco routers from 1700 to GSR with some exceptions	Linux 2.4.x/2.5.x
IPv4 and IPv6 handling	Same program with -6 switch. Separate rules and tables (same syntax)	Separate program from ipfw. Separate rules and tables (similar syntax), later ipfw	Separate tables and separate commands, standard access list	Separate program. separate rules and table, in the future unified table
Where is filtering done (input / output / both)	Both	both	Probably at input	Pre_routing (Input), Post_routing (Output), +forward, +local_in, +local_out
What attributes can be checked? (protocol, ports, etc.)	Protocol, interface, IP address, ports, ICMPv6 type, any protocol type, tos, ttl	Protocol, interface, IP address, ports, ICMPv6 type, any protocol type	Protocol, IP address, ports, dscp, flow value, TCP flags	Protocol, interface, IP address, ports, connection_tracking only on unfragmented packets, marking, MAC address
Attributes	TCP attributes	SYN, ACK, FIN, RST, URG, PSH, +ports, keep state, RFC1323 and RFC1644 not supported,	Ports, FCP flags	SYN,ACK, FIN, RST, URG, PSH, + ports, established, related
	UDP attributes	Ports, Keep-state	ports	Ports

	IPv6 attributes	Address, flow support also	Address	Address	Address, Autoconfigured EUI-64 address usage
	ICMPv6 attributes	keep-state, but only ICMPv6 ECHO handled completely,	Can be specified by number	Autoconfiguration support,	Can be specified by number
	Filter for connection setup?	yes	yes	With reflective access-list	Yes
	Connection/Rate limitation?	In the future	In the future (for IPv4 the code exist)	For ICMPv6	In IPv4 yes. IPv6 not planned yet
	Can it filter IPv6 stateless autoconfiguration?	Rudimentary support only	Rudimentary support only	yes	Basic. Autoconfigured EUI-64 address usage, but the protocol itself not.
Autoconfiguration	Can it filter RA/RS?	Keep-state possible	Simple allow/deny possible with full ICMPv6 filtering	yes	Simple accept/drop/reject possible with full ICMPv6 filtering
	Can it filter router redirect?	Nor recognised, nor handled	Simple allow/deny possible with full ICMPv6 filtering	yes	Simple accept/drop/reject possible with full ICMPv6 filtering
	Can it filter NA/NS?	Keep-state possible	Simple allow/deny possible with full ICMPv6 filtering	yes	Simple accept/drop/reject possible with full ICMPv6 filtering
	Can it filter DAD?	Simple allow/Deny possible for ICMPv6	Simple allow/deny possible with full ICMPv6 filtering	yes	Simple accept/drop/reject possible with full ICMPv6 filtering
	ICMPv6 support?	Keep-state possible	Simple allow/deny possible with full ICMPv6 filtering	Simple allow/deny	Simple accept/drop/reject possible with full ICMPv6 filtering
	How IPv6 extension header handled?	Basic handling	Recognised, check existence – negated check not working, fragments handling in the near future	Extension header handling, Fragmentation header also	hop-by-hop.ipv6-opts.ipv6-route.ipv6-frag.ah,esp.ipv6-nonxt handling
	Source routing handling? (recognition, removal, reject, etc.)	Only working for IPv4	Rudimentary, via matching every routing header, not possible to take action for certain routing header	no	yes
	Filter language?	Quite powerful, easy to understand after knowing the principles, not foolproof	Quite good, easy to understand after knowing the principle, not foolproof	good, but easy to setup, easy to change the order,	Quite powerful, not so easy to understand, not foolproof
	Default behaviour? (accept/deny)	Configuration/Compilation dependent	Configuration/Compilation dependent	Deny, with implicit autoconfiguration support	accept
	Is NAT supported (e.g. translating link/site local addresses to global addresses or IPv4/IPv6 translation)?	Not supported. Not compatible with IPv6	Not supported. Separate tool exist for IPv6/IPv4 translation in KAME stack (natptd, faith)	Not supported	Not supported.
	Stealth feature?	yes	no	no	Possible, complicated
	IPv6 over IPv4 tunnelled packet handling?	No, possible if tunnelling is done on the firewall	No, possible if tunnelling is done on the firewall	No, possible if tunnelling is done on the firewall	No, possible if tunnelling is done on the firewall
	Logging capabilities?	Via external ipmon program	Console logging, dynamically adjustable via sysctl interface	Via syslog	Via syslog
	Documentation?	Web site, Good manual page, tutorial, but lot of things not documented at all	Good manual page, basically same as ipfw (well documented), but lot of things not documented at all, FreeBSD contain some predefined rules	Good documentation, web pages	Website, hacking description, usage description outdated, poor manual pages, Some information at Bieringer's IPv6 webpages.
	Service/Support?	Via mailing list	Via KAME mailing list	Cisco commercial support	Via netfilter mailing list
	Costs?	Freely available, modification should be sent back to the author	Freely available	Available with IOS 12.3(4)T, 12.2(14)S and 12.0(23)S	Freely available

The OpenBSD Packet filter [OPENBSD] not included in the table due to lack of space. The main difference against the ipfilter, that it can properly handle the ICMPv6 messages and has lots of extension in QoS and upper layer packet handling, like Spam control etc.

4 CONCLUSION

The IPv6 firewalls are increasingly supporting more and more features. All the firewalls, we have tested are able to do the basic packet filtering with negligible performance impact (usually less than 2% in larger packets and less than 10% on small packets) We can safely claim that all the tested IPv6 firewalls work well at the basic level, but the implementers of firewalls should better support IPv6. All implementations are basically done with IPv4 in mind and the new and advanced IPv6 features are not or hardly supported.

5 ACKNOWLEDGEMENT

I would like to thank to Tim Chown from University of Southampton for their valuable comments and review of earlier version of the paper.

6 REFERENCES

[GTPv6]	GEANT Test Programme for IPv6,	http://www.ipv6.ac.uk/gtpv6/
[IP_FILTER]	IPFilter homepage	http://coombs.anu.edu.au/~avalon/ip-filter.htm
[KAME]	KAME Project	http://www.kame.net
[NETFILTER]	Firewalling, NAT and packet mangling for Linux 2.4	http://netfilter.samba.org/
[OPENBSDPF]	OpenBSD Packet Filter	http://www.benzedrine.cx/pf.html
[SAVOLA]	IPv6 Transition/Co-existence Security Considerations	http://www.netcore.fi/pekkas/ietf/draft-savola-v6ops-security-overview-00.txt
[SUNIPV6]	IPv6 for Solaris,	http://www.sun.com/solaris/ipv6/
[TF-NGN]	Task Force: Next Generation Networks,	http://www.dante.org.uk/tf-ngn/
[USAGI]	USAGI (UniverSAl playGround for IPv6)	http://www.linux-ipv6.org