Internet Engineering Task Force
INTERNET-DRAFT
Expires: Feb 22, 2003

Jun-ichiro itojun Hagino
Research Lab, IIJ
Aug 22, 2002

IPv4 mapped address considered harmful
draft-itojun-v6ops-v4mapped-harmful-00.txt

Status of this Memo

Abstract

IPv6 address architecture [Hinden, 1998] defines IPv4 mapped address. The representation is used in IPv6 basic API [Gilligan, 1999] to denote IPv4 destinations on AF_INET6 socket within the API. At the same time, there are protocol proposals that use IPv4 mapped address on wire. Therefore, IPv4 mapped address has two meanings, and they are not distinguishable from the userland applications. This draft discusses security threats due to the dual use of IPv4 mapped address. It also discusses threats due to the additional complexities introduced by IPv4 mapped address.

1. Dual meaning of IPv4 mapped address

IPv6 basic socket API [Gilligan, 1999] defines the use of IPv4 mapped address with AF_INET6 sockets. IPv4 mapped address is used as an internal identifier for IPv4 peers, on AF_INET6 sockets. The API is

designed with IPv4/v6 dual stack nodes in mind. When an IPv4 packet reaches an IPv4/v6 dual stack node, kernel IPv4 layer will handle it,

then passes it up to TCP/UDP layer. When TCP/UDP layer finds an AF_INET6 listening socket, it will pass the packet to the listening socket as if it was from the corresponding IPv4 mapped address. Let us call it "basic API behavior" in this draft.

Some of the translator technologies such as SIIT [Nordmark, 2000] uses IPv4 mapped address in header fields of actual IPv6 packet on wire. These technologies are designed with IPv6 only nodes in mind. It is assumed that IPv6 packets with IPv4 mapped address will be handled by IPv6 layer then by TCP/UDP layer, and reaches an AF_INET6 socket. Let us call it "SIIT behavior" in this draft.

2. Threats due to the use of IPv4 mapped address on wire

When userland application on top of AF_INET6 API sees peers with IPv4 mapped addresses (like by getpeername(2) or recvfrom(2)), it cannot detect if the packet actually was IPv4 (IPv4 mapped address appeared due to basic API behavior) or IPv6 (SIIT behavior).

This ambiguity creates chances to malicious party to trick victim nodes. Here are a couple of examples:

- By transmitting IPv6 packet with ::ffff:127.0.0.1 in IPv6 source address field, applications that assume basic API behavior will be tricked to believe that the packet is from the node itself (IPv4 loopback address, 127.0.0.1).

- By transmitting IPv6 packet to firewall device, with IPv4 mapped address corresponds to address inside the firewall (like ::ffff:10.1.1.1) as the IPv6 source address, malicious party could bypass IPv4 filtering rules and inject traffic inside the firewall.

- Assume that the victim node is an IPv4/v6 dual stack node. By transmitting IPv6 packet with IPv4 mapped address corresponds to IPv4 broadcast address (::ffff:10.255.255.255) in IPv6 source address field, to TCP/UDP port that swaps IPv6 source and destination address (e.g. UDP port 53, DNS), malicious node can trick the victim node to generate improper IPv4 broadcast traffic; This is because basic API on the victim node will emit transmission requests to destination IPv4 mapped address, ::ffff:10.255.255.255, into IPv4 traffic.

3. Other threats related to IPv4 mapped address

3.1. Access control complexity

RFC2553 section 3.7 adds complexity to access controls. Due to the additional complexity, it is likely that there will be many mistakes in access controls.

Due to RFC2553 section 3.7, AF_INET6 socket will accept IPv4 packets. On an IPv4/v6 dual stack node, if there is no AF_INET listening socket, normal administrators would believe that there will be no access from IPv4 peers. However, if AF_INET6 listening socket is present, IPv4 peers will be able to access the service.

To protect applications from this threat, every access control logic has to have a special case handling for IPv4 mapped address. It is impossible to enforce such a requirement to every application implementations.

4. Suggested protocol change

- In IPv4 address architecture document [Hinden, 1998] explicitly state that IPv4 mapped address is for use within basic API [Gilligan, 1999] , and basic API only. Forbid any other uses.

- Move any document that suggests the use of IPv4 mapped address on wire to historic, due to security reasons.

The above change will remove the threat due to the use of IPv4 mapped address on wire.

Another way is to deprecate RFC2553 section 3.7, however, due to the wide deployment of applications that use IPv6 basic API, the option is not feasible.

5. Suggested implementation tips

5.1. Kernel/library developers

o Do not support IPv4 mapped address on AF_INET6 API (RFC2553 section 3.7). By doing so the kernel TCP/UDP code will be greatly simplified, and will reduce the likelihood of security-sensitive kernel bugs.

o Implement 2553bis [Gilligan, 2002] IPV6_V6ONLY socket option, and make the default value to on (the default value suggested by the document is "off"). This has almost the same effect as the previous bullet. With the approach you still have to implement complex in-kernel interaction between AF_INET and AF_INET6 socket, which can lead to security-senstive kernel bugs. Also, once a userland application turns the socket option off, your system will become vulnerable. The change will make your stack incompatible with 2553bis section 3.7 and 5.3.

o Drop any IPv6 native packet with IPv4 mapped address in any of IPv6 header fields as well as IPv6 extension header fields. It will make the system incompatible with SIIT.

o Drop any IPv6 DNS response that contains IPv4 mapped address.

5.2. Application developers

o In EVERY userland application check the IPv6 source address, if it embeds bad IPv4 address. This approach is impossible in reality, as it's hard to know what is "bad" address, and there are millions of coders in different places. There is no way to enforce this rule.

o Do not try to utilize RFC2553 section 3.7 (IPv4 traffic on AF_INET6 socket). Implement server applications by using AF_INET and AF_INET6 listening socket. Explicitly set IPV6_V6ONLY socket option to on, whenever the socket option is available on the system.

> NOTE: Due to the lack of standard behavior in bind(2) semantics, this may not be possible on some systems. Some IPv6 stack does not permit bind(2) to 0.0.0.0, after bind(2) to ::. Also, there is no standard on how IPv4 traffic will be routed when both 0.0.0.0 and :: listening sockets are available on the same port.

6. Security considerations

The document talks about security issues in the use of IPv4 mapped address. Possible solutions are provided.

7. Change History

none yet.

References

Hinden, 1998.
R. Hinden and S. Deering, "IP Version 6 Addressing Architecture" in RFC2373 (July 1998). ftp://ftp.isi.edu/in-notes/rfc2373.txt.

Gilligan, 1999.
R. Gilligan, S. Thomson, J. Bound, and W. Stevens, "Basic Socket Interface Extensions for IPv6" in RFC2553 (March 1999). ftp://ftp.isi.edu/in-notes/rfc2553.txt.

Nordmark, 2000.
E. Nordmark, "Stateless IP/ICMP Translator (SIIT)" in RFC2765 (February, 2000). ftp://ftp.isi.edu/in-notes/rfc2765.txt.

Gilligan, 2002.
R. Gilligan, S. Thomson, J. Bound, J. McCann, and W. R. Stevens, "Basic Socket Interface Extensions for IPv6" in draft-ietf-ipngwg-rfc2553bis-06.txt (July 2002). work in progress material.

Author's address

Jun-ichiro itojun Hagino
Research Laboratory, Internet Initiative Japan Inc.
Takebashi Yasuda Bldg.,
3-13 Kanda Nishiki-cho,
Chiyoda-ku,Tokyo 101-0054, JAPAN
Tel: +81-3-5259-6350
Fax: +81-3-5259-6351
email: itojun@iijlab.net