Internet Area
Internet-Draft
Intended status: Standards Track
Expires: 3 December 2026

L. He
L. Gai
Z. Jia
Y. Liu
Tsinghua University

1 June 2026

Security Requirements for IP Tunnel Nodes
draft-gai-intarea-ip-tunnel-node-security-00

Abstract

IP tunnel nodes are widely deployed for IPv4/IPv6 transition, overlay connectivity, traffic engineering, and inter-domain services. A tunnel node that accepts tunneled packets from unauthorized sources or forwards decapsulated packets without validating the inner packet can become an open relay, a source-address-spoofing enabler, or a path around filtering policy.

This document specifies security requirements for IP tunnel nodes. It defines requirements for tunnel enablement, peer authorization, source address validation, forwarding-scope validation, recursive encapsulation limits, IPv6 extension header handling, ICMP behavior, logging, and operational management. The requirements apply to IP-in-IP, GRE, IPv6 tunnel mechanisms, and IPv4/IPv6 transition tunnels.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 December 2026.

Copyright Notice

Table of Contents

1. Introduction

IP tunneling carries an inner IP packet as the payload of an outer IP packet. The outer header is routed across a delivery network. The tunnel egress removes the outer encapsulation and then forwards or locally consumes the inner packet. This model is used by IP-in-IP [RFC2003], generic packet tunneling in IPv6 [RFC2473], Generic Routing Encapsulation (GRE) [RFC2784], GRE-in-IPv6 [RFC7676], and

IPv4/IPv6 transition mechanisms [RFC4213].

A tunnel node is not merely another router on the native forwarding path. Decapsulation creates a second forwarding decision, often in a different address family or forwarding domain. If the tunnel node accepts traffic from arbitrary outer sources, or forwards arbitrary inner packets after decapsulation, it can bypass filtering that would normally have applied on the native path.

This document specifies requirements that make tunnel nodes explicit, bounded, and accountable. The intent is not to deprecate IP tunneling. The intent is to prevent tunnel nodes from operating as open relays, source spoofing enablers, or uncontrolled recursive forwarding points.

1.1. Scope

This document applies to devices and software that encapsulate, decapsulate, relay, or forward IP tunnel packets, including routers, hosts, firewalls, tunnel brokers, customer edge devices, provider edge devices, and middleboxes.

The requirements apply to configured tunnels and to automatic or transition mechanisms when those mechanisms are enabled. This document does not define a new tunnel encapsulation and does not change the packet format of any existing tunnel protocol.

1.2. Terminology

IP tunnel:
: A mechanism that carries an inner IP packet as payload of an outer IP packet. The delivery protocol can be IPv4 or IPv6, and the passenger protocol can be IPv4 or IPv6.

Tunnel node:
: A node that performs IP tunnel encapsulation, decapsulation, or tunnel relay processing. "Tunnel node" is the generic term used in this document. The following terms describe functional roles of a tunnel node. A single device can act as an ingress tunnel node for one tunnel, an egress tunnel node for another tunnel, and a relay tunnel node for a third tunnel.

Ingress tunnel node:
: A tunnel node that receives a native IP packet and encapsulates it into an outer IP packet for delivery across a tunnel.

Egress tunnel node:
A tunnel node that receives an encapsulated IP packet, removes the outer tunnel encapsulation, and locally delivers or forwards the resulting inner IP packet according to tunnel policy.

Relay tunnel node:
A tunnel node that decapsulates a packet and then forwards or re-encapsulates the resulting inner packet into another routing domain, address family, tunnel, or native network.

Open tunnel node:
A tunnel node that accepts tunneled packets from sources that are not administratively configured, authenticated, or otherwise authorized.

Tunnel peer:
A remote endpoint or relay authorized to send tunneled packets to a tunnel node.

Inner source prefix set:
The set of inner source prefixes that a tunnel peer is authorized to inject through a tunnel.

Recursive encapsulation:
A packet structure in which the inner packet exposed by one decapsulation step is itself another tunneled packet.

1.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Tunnel Node Processing Model

A tunnel node compliant with this document applies the following logical processing stages:

1. Determine whether the addressed tunnel mechanism is enabled.

2. Identify the candidate tunnel policy from the outer packet and local configuration.

3. Validate the outer source address and tunnel peer.

4. Authorize decapsulation for the selected tunnel policy.

5. Decapsulate the packet.

6. Validate the exposed inner packet, including the inner source address and destination forwarding scope.

7. Apply recursive encapsulation limits and IPv6 extension header policy.

8. Forward, locally deliver, re-encapsulate, or discard the packet.

9. Update counters and emit rate-limited logs or control-plane responses as configured.

An implementation MAY combine these stages internally, but it MUST preserve the externally visible security properties described in this document.

A packet that fails any validation stage MUST be discarded unless a later section explicitly allows a different action. The tunnel node MUST NOT forward a packet merely because a previous stage accepted the outer packet.

3. Default Configuration Requirements

Tunnel decapsulation MUST be disabled unless a tunnel mode is explicitly enabled by configuration, authenticated control-plane state, or a standards-defined automatic tunnel mechanism.

Implementations MUST NOT enable open decapsulation by default. A node that supports a permissive or automatic mode MUST require explicit administrative action to enable that mode.

A tunnel node MUST provide a per-tunnel policy object containing at least:

* the enabled encapsulation type;

* the authorized outer peer address or peer address set, except where a standards-defined automatic mode derives this value from the packet;

* the inner source prefix set authorized for each peer;

* the forwarding domain or interface into which validated inner packets may be forwarded;

* the maximum recursive decapsulation depth;

* the IPv6 extension header policy applied after decapsulation;

* logging and rate-limiting parameters.

The default maximum recursive decapsulation depth MUST be one. A higher value MUST require explicit configuration.

4. Peer Authorization and Authentication

Configured tunnels SHOULD authenticate tunnel peers. Authentication MAY be provided by a secure control plane, IPsec, authenticated keying, cryptographic tunnel mechanisms, or another operator-approved mechanism.

GRE keys, IPv6 flow labels, interface identifiers, and predictable address formats MUST NOT be treated as peer authentication by themselves. They MAY be used as selectors after the peer has already been authorized by other means.

If authentication is not available for a tunnel mode, the implementation MUST provide address-based peer authorization and inner source prefix validation.

Operators SHOULD prefer authenticated tunnels when the tunnel crosses an untrusted network. Operators SHOULD treat unauthenticated tunnels as explicitly exposed attack surfaces and SHOULD apply stricter rate limits and logging to them.

5. Source Address Validation Requirements

Source Address Validation (SAV) prevents packets with invalid or unauthorized source addresses from entering or leaving a network. Tunnel processing MUST NOT bypass SAV for either the outer packet or the inner packet exposed after decapsulation.

5.1. Outer Source Address Validation

A tunnel node MUST validate the outer source address before decapsulation. For configured tunnels, the outer source address MUST match the configured tunnel peer or a configured peer address set.

For automatic tunnel mechanisms, the tunnel node MUST perform the source-address consistency checks specified for that mechanism. 6to4 implementations and relays MUST implement the security checks described in [RFC3964].

Packets that fail outer source address validation MUST be discarded before decapsulation. A tunnel node MUST NOT generate an ICMP error in response to a packet discarded for failed outer source address validation unless local policy explicitly permits it and the response is rate limited.

5.2. Inner Source Address Validation

After decapsulation, an egress or relay tunnel node MUST validate the inner source address against the inner source prefix set associated with the tunnel peer and tunnel mode.

If the inner source address is not covered by the authorized inner source prefix set, the packet MUST be discarded. The packet MUST NOT be forwarded, re-encapsulated, reflected, or locally answered.

For ingress encapsulation, a tunnel node MUST apply SAV to native packets before encapsulating them into a tunnel. SAV SHOULD be implemented using BCP 38 and BCP 84 techniques, including strict or feasible-path unicast reverse path forwarding where appropriate [RFC2827] [RFC3704] [RFC8704].

For multihomed or asymmetric networks, strict uRPF can cause false drops. In those deployments, operators SHOULD use feasible-path, enhanced feasible-path, access-list, or source-prefix-list policies that preserve SAV while accounting for routing asymmetry.

5.3. Destination and Forwarding-Scope Validation

A tunnel node MUST verify that the inner destination is permitted by the tunnel policy before forwarding the decapsulated packet.

A tunnel node MUST NOT act as an open relay that accepts arbitrary tunneled traffic and forwards arbitrary inner destinations into the global Internet.

If a tunnel is intended to provide transit service, the node MUST define the permitted source prefixes, destination scope, and forwarding domain for that service. Transit behavior MUST be disabled by default.

6. Recursive Encapsulation Requirements

A tunnel node MUST detect when a decapsulated inner packet is itself an IP tunnel packet supported by the same node.

Recursive decapsulation MUST be bounded by a configurable maximum depth. The default maximum depth MUST be one. Packets that exceed the maximum depth MUST be discarded.

A tunnel node MUST apply the same peer authorization, outer source validation, inner source validation, IPv6 extension header policy, and forwarding-scope validation at each decapsulation layer. Validation at the outermost layer MUST NOT be treated as authorization for deeper layers.

Automatic re-encapsulation of a decapsulated packet into another tunnel MUST be disabled by default. If re-encapsulation is enabled, it MUST be controlled by explicit routing or policy state and MUST preserve the SAV requirements in this document.

7. IPv6 Extension Header Requirements

IPv6 extension headers are specified by [RFC8200]. Operational considerations for forwarding packets that contain IPv6 extension headers are described in [RFC7045] and [RFC9098]. Tunnel nodes need explicit policy because decapsulation can expose an IPv6 packet that was not visible to previous filtering devices.

After each decapsulation step, a tunnel node MUST apply its IPv6 extension header policy to the exposed inner IPv6 packet before forwarding or re-encapsulation.

A tunnel node MUST be able to parse enough of the IPv6 header chain to enforce:

* source and destination address validation;

* fragment policy;

* routing header policy;

* upper-layer protocol policy where such policy is configured;

* local control-plane protection.

If the tunnel node cannot parse enough of the IPv6 header chain to enforce configured policy, it MUST discard the packet or send it to a rate-limited inspection path. It MUST NOT blindly forward such a packet on the fast path.

A tunnel node MUST provide configurable limits for:

* the number of IPv6 extension headers;

* the total length of the IPv6 extension header chain;

* the number of Fragment Headers;

* the use of Routing Headers;

* the use of Hop-by-Hop Options.

Routing Header Type 0 packets MUST be discarded, consistent with [RFC5095]. Segment Routing Headers and other routing headers SHOULD be accepted only when the tunnel node is part of an explicitly configured trusted domain for that routing header type.

IPv6 fragments that prevent the tunnel node from enforcing configured SAV, forwarding, or upper-layer policy MUST be discarded. Implementations SHOULD provide counters that distinguish these drops from ordinary forwarding drops.

8. Protocol-Specific Requirements

8.1. IP-in-IP and IP6IP6

For IP-in-IP [RFC2003] and IP6IP6 [RFC2473], tunnel nodes MUST enforce configured outer peer authorization and inner source prefix validation. A packet with protocol number 4 or 41 addressed to a node MUST NOT be decapsulated unless it matches an enabled tunnel policy.

An IP-in-IP or IP6IP6 tunnel node MUST NOT forward a decapsulated inner packet whose source address is outside the inner source prefix set authorized for that tunnel peer.

8.2. GRE and GRE-in-IPv6

For GRE [RFC2784] and GRE-in-IPv6 [RFC7676], the tunnel node MUST validate the outer peer before using GRE header fields to select a tunnel.

GRE keys MAY select a tenant, virtual routing table, or tunnel policy, but a GRE key MUST NOT authorize traffic by itself.

If a GRE tunnel carries multiple inner protocol families, the tunnel policy MUST define inner source prefix sets for each enabled passenger protocol family.

8.3. IPv4/IPv6 Transition Tunnels

For configured IPv6-over-IPv4 and IPv4-over-IPv6 transition tunnels [RFC4213], the node MUST bind each tunnel peer to the inner prefixes that peer may inject.

Cross-protocol encapsulation MUST NOT bypass SAV for either address family. Operators SHOULD audit transition tunnels more frequently than single-family tunnels because IPv4 and IPv6 SAV policies are often configured by different teams, devices, or routing processes.

8.4. 6to4

6to4 [RFC3056] has known operational and security problems, including the deprecation of the 6to4 anycast relay prefix described in [RFC7526]. A node that continues to operate 6to4 MUST implement the filtering checks in [RFC3964] and the additional requirements in this document.

A 6to4 relay or border function MUST discard packets when the embedded IPv4 address in a 2002::/16 address is inconsistent with the outer IPv4 source or destination according to the direction of processing and the relay role.

A 6to4 relay MUST NOT provide unrestricted transit for spoofed inner sources. Relay behavior MUST be explicitly enabled, logged, and rate limited.

9. ICMP, TTL, Hop Limit, MTU, and Fragmentation

When a tunnel node forwards a decapsulated packet, it MUST apply the normal forwarding rules for the inner protocol, including TTL or Hop Limit processing.

If a packet is dropped because TTL or Hop Limit expires after decapsulation, any generated ICMP error MUST be rate limited. ICMP errors MUST NOT be generated for packets that fail source validation, peer authorization, or extension header policy unless explicitly enabled by local policy.

Tunnel nodes SHOULD rate limit ICMP Echo processing for packets exposed by decapsulation. Tunnel nodes MUST NOT allow decapsulated ICMP traffic to create amplification materially larger than the received packet stream.

Tunnel nodes MUST provide MTU handling consistent with the underlying tunnel specifications. A tunnel node SHOULD avoid fragmentation where Path MTU Discovery can be used.

Fragmentation behavior MUST NOT bypass validation. If a tunnel node cannot validate the inner packet because required information is split across fragments or unavailable without reassembly, the node MUST either perform safe reassembly within configured resource limits or discard the packet.

Reassembly state created for tunnel validation MUST be resource limited. When resource limits are exceeded, the node MUST fail closed for packets that require reassembly for validation.

10. Operational Management and Telemetry

Operators deploying tunnel nodes SHOULD:

* inventory all enabled tunnel modes and endpoints;
* disable unused tunnel protocols at borders and hosts;
* bind every configured peer to authorized inner source prefixes;
* align IPv4 and IPv6 SAV policy for cross-protocol tunnels;
* explicitly configure recursive encapsulation depth;
* apply a conservative IPv6 extension header policy at tunnel egress;
* rate limit ICMP generated after decapsulation;
* monitor tunnel drop counters;
* periodically test that tunnel nodes are not open relays.

A tunnel node MUST provide counters for at least:

* packets accepted by tunnel policy;
* packets discarded for unknown tunnel mode;
* packets discarded for unauthorized outer peer;
* packets discarded for failed inner SAV;
* packets discarded for exceeded recursive encapsulation depth;
* packets discarded for IPv6 extension header policy;
* packets discarded for fragment policy;

* generated ICMP errors after decapsulation.

Operators SHOULD be able to export these counters through existing telemetry mechanisms. Implementations SHOULD support sampled logging for discarded packets, subject to rate limits and privacy controls.

Logs SHOULD include the tunnel identifier, outer peer, tunnel mode, drop reason, and address family. Logs SHOULD NOT include payload beyond the headers required for diagnosis.

Operators SHOULD treat open tunnel nodes as security defects unless they are part of a deliberate, documented, and filtered relay service.

11. IANA Considerations

This document has no IANA actions.

12. Security Considerations

This entire document specifies security requirements for IP tunnel nodes. The main threats are:

* unauthorized use of a tunnel node as an open relay;

* injection of spoofed inner source addresses;

* bypass of IPv4 or IPv6 SAV through cross-protocol tunnels;

* recursive encapsulation used to construct unintended forwarding paths;

* IPv6 extension header chains used to evade filtering or induce expensive processing;

* ICMP reflection or amplification after decapsulation;

* inconsistent policy between native forwarding and tunnel forwarding.

The requirements in this document reduce these risks by requiring peer authorization, inner source prefix validation, bounded recursive decapsulation, explicit IPv6 extension header policy, and operational telemetry.

These requirements do not protect against a compromised authorized peer that sends traffic from authorized source prefixes. Operators needing stronger guarantees SHOULD use authenticated and encrypted tunnel mechanisms and SHOULD apply traffic anomaly detection at the tunnel ingress and egress.

Tunnel telemetry can reveal customer prefixes, tunnel peers, and traffic policy. Implementations and operators SHOULD limit log retention and SHOULD aggregate or anonymize exported telemetry when fine-grained data is not required for operations or incident response.

Measurement data collected through tunnel nodes can reveal SAV gaps in specific networks. Such data SHOULD be handled as sensitive operational security information.

13. Contributors

The following individual contributed significantly to the technical content, review, and analysis presented in this document:

Daguo Cheng
Tsinghua University
Beijing
China
Email: cdg22@mails.tsinghua.edu.cn

Chentian Wei
Tsinghua University
Beijing
China
Email: wct24@mails.tsinghua.edu.cn

14. Acknowledgments

The author thanks the IETF community for prior work on IP tunneling, source address validation, IPv6 extension header processing, and operational security guidance.

15. References

15.1. Normative References

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <https://www.rfc-editor.org/info/rfc2003>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/info/rfc3704>.

[RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, DOI 10.17487/RFC3964, December 2004, <https://www.rfc-editor.org/info/rfc3964>.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <https://www.rfc-editor.org/info/rfc4213>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/info/rfc7045>.

[RFC7676] Pignataro, C., Bonica, R., and S. Krishnan, "IPv6 Support for Generic Routing Encapsulation (GRE)", RFC 7676, DOI 10.17487/RFC7676, October 2015, <https://www.rfc-editor.org/info/rfc7676>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8704] Sriram, K., Montgomery, D., and J. Haas, "Enhanced Feasible-Path Unicast Reverse Path Forwarding", BCP 84, RFC 8704, DOI 10.17487/RFC8704, February 2020, <https://www.rfc-editor.org/info/rfc8704>.

[RFC9098] Gont, F., Hilliard, N., Doering, G., Kumari, W., Huston, G., and W. Liu, "Operational Implications of IPv6 Packets with Extension Headers", RFC 9098, DOI 10.17487/RFC9098, September 2021, <https://www.rfc-editor.org/info/rfc9098>.

15.2. Informative References

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <https://www.rfc-editor.org/info/rfc3056>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <https://www.rfc-editor.org/info/rfc5095>.

[RFC7526] Troan, O. and B. Carpenter, Ed., "Deprecating the Anycast Prefix for 6to4 Relay Routers", BCP 196, RFC 7526, DOI 10.17487/RFC7526, May 2015, <https://www.rfc-editor.org/info/rfc7526>.

Authors' Addresses

Lin He
Tsinghua University
Beijing
China
Email: he-lin@tsinghua.edu.cn

Le Gai
Tsinghua University
Beijing
China

Email: gl25@mails.tsinghua.edu.cn

Zedong Jia
Tsinghua University
Beijing
China
Email: jzd25@mails.tsinghua.edu.cn

Ying Liu
Tsinghua University
Beijing
China
Email: liuying@cernet.edu.cn