Internet Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: 4 September 2025

W. Kumari
Google, LLC
A. Alston
Alston Networks
. Vyncke
S. Krishnan
Cisco
D. Eastlake
Independent
3 March 2025

Safe(r) Limited Domains
draft-wkumari-intarea-safe-limited-domains-04

Abstract

Documents describing protocols that are only intended to be used within "limited domains" often do not clearly define how the boundary of the limited domain is implemented and enforced, or require that operators of these limited domains perfectly filter at all of the boundary nodes of the domain to protect the rest of the global Internet from these protocols and vice-versa.

This document discusses some design principles and offers mechanisms to allow protocols that are designed to operate in a limited domain "fail-closed" rather than "fail-open", thereby making these protocols safer to deploy on the Internet.

These mechanism are not applicable to all protocols intended for use in a limited domain, but if implemented on certain classes of protocols, they can significantly reduce the risks.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Internet Area Working Group Working Group mailing list (int-area@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/int-area/.

Source for this draft and an issue tracker can be found at https://github.com/wkumari/draft-wkumari-intarea-safe-limited-domains.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[RFC8799] discusses the concept of "limited domains", provides examples of limited domains, as well as Examples of Limited Domain Solutions, including Service Function Chaining (SFC [RFC7665]), Segment Routing, "Creative uses of IPv6 features" (including Extension headers, e.g., for in situ Operations, Administration, and maintenance [RFC9378]).

In order to provide context, this document will quote extensively from [RFC8799], but it is assumed that the reader will actually read [RFC8799] in its entirety.

[RFC8799] Section 3, notes:

> A common argument is that if a protocol is intended for limited use, the chances are very high that it will in fact be used (or misused) in other scenarios including the so-called open Internet. This is undoubtedly true and means that limited use is not an excuse for bad design or poor security. In fact, a limited use requirement potentially adds complexity to both the protocol and its security design, as discussed later.

Notably, in [RFC8799] Section 2, states:

> Domain boundaries that are defined administratively (e.g., by address filtering rules in routers) are prone to leakage caused by human error, especially if the limited domain traffic appears otherwise normal to the boundary routers. In this case, the network operator needs to take active steps to protect the boundary. This form of leakage is much less likely if nodes must be explicitly configured to handle a given limited-domain protocol, for example, by installing a specific protocol handler.

In addition, [RFC8799] Section 6, notes:

> Today, where limited domains exist, they are essentially created by careful configuration of boundary routers and firewalls. If a domain is characterized by one or more address prefixes, address assignment to hosts must also be carefully managed. This is an error-prone method, and a combination of configuration errors and default routing can lead to unwanted traffic escaping the domain. Our basic assumption is therefore that it should be possible for

domains to be created and managed automatically, with minimal human configuration. We now discuss requirements for automating domain creation and management.

This document discusses some of the mechanisms which protocol designers can use to limit the scope of their protocols to a single link. If the protocol is intended to be used in across multiple links, but should not be forwarded beyond a single administrative domain, then the protocol designer should consider making the protocol "fail-closed" rather than "fail-open", as described below.

This is primarily targeted towards protocols which are intended to primarily be used within a single layer-2 broadcast domain, or for protocols which provide a transport type service (similar to MPLS or SRv6) and are not intended to remain within a single administrative domain.

2. Conventions and Definitions

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.

3. Some types of limited domain protocols

[RFC8799] Section 3 discusses some examples of Limited Domains, based mainly on the network type (e.g. Home, Sensor Networks, Data Centers, etc).

This section instead classifies the types of limited domain protocols based more on their intended use, and technology.

Broadly speaking, there are two types of limited domain protocols:

* Layer-2 type limited domain protocols: These are protocols that are intended to be used within a single LAN segment.

* Transport type service (for example MPLS and SRv6): These protocols are intended to provide a transport service, and are intended to remain within a single administrative domain such as a Enterprise or a Service Provider network.

4. Fail-open versus Fail-closed

Protocols can be broadly classified as either "fail-open" or "fail-closed". Fail-closed protocols are those that require explicit interface or device-wide configuration to enable them to be accepted or processed when received on an interface. A classic example of a fail-closed protocol is MPLS ([RFC3031]): In order to allow MPLS to transit an interface, the operator must enable the MPLS protocol on that interface and on the device itself. This ensures that outside MPLS traffic does not leak in.

Fail-open protocols are those that require explicit configuration in order to ensure that they do not leak out of a domain, for example, through the application of filters. An example of a fail-open protocol is SRv6 - in order to ensure that SRv6 traffic does not leak out of a network, the operator must explicitly filter this traffic, and, in order to ensure that SRv6 traffic does not leak in, the operator must explicitly filter SRv6 traffic.

Fail-open protocols are inherently riskier than fail-closed protocols, as they rely on perfect configuration of filters on all interfaces at the boundary of a domain, and, if the filters are removed for any reason (for example, during troubleshooting), there is a risk of inbound or outbound leaks. In addition, some devices or interfaces may have limitations in the size and complexity of filters that can be applied, and so adding new filter entries to limit leaks of a new protocol may not be possible.

Fail-closed protocols, on the other hand, do not require any explicit filtering. In order for the protocol to be accepted and processed when received on an interface, the operator must explicitly enable the protocol on that interface and on the device itself. In addition, there is less risk of operational mistakes, as it does not rely on filters that may be limited in number and complexity. Finally, fail-closed protocols do not require that operators of networks outside of the limited domain implement filters to protect their networks from the limited domain traffic.

5. IP Hop-Limit Limiting

Some limited domain protocols are intended to only be used within a single IP subnet. In these cases, it may be possible to use the IP Hop-Limit to ensure that the protocol does not leak out of the subnet.

By specifying that the IP Hop-Limit of packets carrying the protocol be set to a value of 1, it is possible to ensure that the protocol does not leak out of the subnet. This is because routers will decrement the Hop-Limit of packets by 1 when forwarding them, and discard the packet when it reaches zero.

The approach of setting the IP Hop-Limit to 1 ensures that the protocol does leave the subnet. This is different from requiring the received IP Hop-Limit has a value of 255, as used in [RFC3682], which ensures that traffic cannot be spoofed from outside the subnet.

Which option to choose (if either) depends on the specific requirements of the protocol.

6. IPv4 Multicast Addressing

Some protocols (e.g OSPF) use addresses from the IP Local Network Control Block [RFC5771], (224.0.0/24). In addition to providing a discovery mechanism, this traffic is not forwarded off-link, providing a simple and effective way to limit the scope of the protocol.

In some (rare) cases, IPv4 "Link Local" addresses ([RFC3927] may be an appropriate mechanism to limit the scope of the protocol, but this such a niche case that it is not discussed further here.

7. IPv6 Link Local Addresses

Link-Local IPv6 Unicast Addresses ([RFC4291] Section 2.5.6) are used for communication between nodes on a single link. They are not routable and are not forwarded by routers. In cases where a limited-domain protocol is intended to be used only within a single link, the use of IPv6 Link-Local addresses can be an effective way to limit the scope of the protocol.

8. Making a layer-3 type limited-domain protocol fail-closed

One way to make a limited-domain protocol fail-closed is to assign it a unique layer-2 protocol identifier, usually an EtherType. This mechanism is used by MPLS. In modern router and hosts, if such a protocol identifier is not enabled on an interface, then the Ethernet chip-set will ignore the frame, and the node will not see or process it. Thus, it is necessary to specifically enable the layer-2 protocol identifier on all relevant interfaces inside the limited domain, and the protocol will be blocked at the domain boundary where the protocol has not been so enabled. This is a simple and effective mechanism to ensure that the protocol does not leak out of the limited domain if and when an operator makes a mistake in configuring

filters based on identifiers appearing deeper in the frame such as IP addresses or IP protocol or header options.

This layer-2 protocol identifier technique only works for transport-type limited domain protocols (i.e., protocols running at layer 3). Higher layer protocols cannot necessarily be protected in this way, and so cryptographically enforced mechanisms may need to be used instead (e.g., as done used by ANIMA in [RFC8994] and [RFC8995]).

9. Ethernet Protocol Identification

Figure 1 shows the general format of Ethernet frames. The relevant protocol identification field occurs after the destination and source MAC addresses and any tags (such a VLAN tags). The alternatives for protocol identification are discussed in Section 3 of [RFC9542].

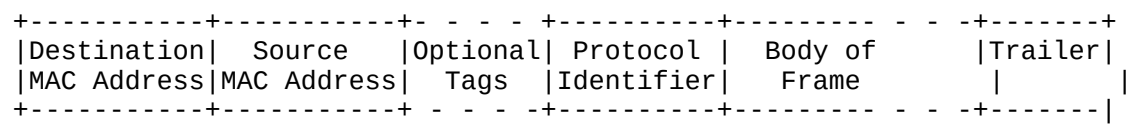
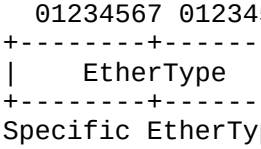


Figure 1: Ethernet Frame Format

This document considers EtherType protocol identification. An EtherType is an unsigned 16-bit field in an Ethernet frame with a value in the range of 0x0600 to 0xFFFF, and so it is a somewhat limited resource; however, there exists a special Extended EtherType (0x88B7) that can be suffixed by an Organizationally Unique Identifier (OUI) followed by a further 16-bits identifying the protocol relative to that OUI as discussed in Section 3 of [RFC9542]. These alternatives of a direct EtherType or use of the Extended EtherType for the case of the IANA OUI are illustrated in Figure 2. The following subsections discuss the factors which may influence the choice between these alternatives when use of such layer 2 protocol identification, to make the isolation of a limited domain more robust, is warranted.



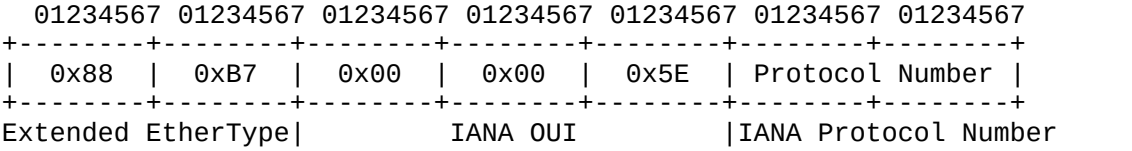


Figure 2: EtherType Based Protocol Identification

Because specific EtherTypes are a limited resource, an Extended EtherType SHOULD be used unless there is a strong reason why it will not work satisfactorily and a specific EtherType is required.

9.1. Extended EtherType Protocol Identification

The main advantage of using an Extended EtherType with an IANA Protocol Number, as shown in Figure 2, is that such a number can be allocated by IANA with Expert Review based on an Internet Draft and is thus relatively easy to obtain. The main disadvantage is that the protocol identification is 5 bytes longer than a specific dedicated EtherType.

9.2. Specific EtherType Protocol Identification

The primary disadvantage of using a specific EtherType, as opposed to an Extended EtherType, is that assignment of such an EtherType is significantly more difficult than assignment of an Extended EtherType IANA protocol number. As discussed in [RFC9542], a specific EtherType can only be assigned by the IEEE Registration Authority under the following policy: "Since EtherTypes are a fairly scarce resource, the IEEE RAC has let us know that they will not assign a new EtherType to a new IETF protocol specification until the IESG has approved the protocol specification for publication as an RFC. In exceptional cases, the IEEE RA is willing to consider "early allocation" of an EtherType for an IETF protocol that is still under development as long as the request comes from and has been vetted by the IESG." ([RFC9542] Appendix B.1, citing [IESG_EtherType])

During development and testing, a protocol can use a "Local Experimental Ethertype" (0x88b5 and 0x88b6 - [IANA_EtherType]). Once the protocol is approved for publication, the IESG can request an EtherType from the IEEE. However, there is always a risk of some implementation using a Local Experimental EtherType not getting updated causing conflicts with a later different use of that experimental EtherType.

The primary advantage of using a specific EtherType is the saving of 5 bytes relative to the use of the Extended EtherType with a protocol number under the IANA OUI.

10. Security Considerations

Protocols are designated as "limited domain" because something unexpected might happen if they leak outside of a domain with unified management. For example, VLAN or VPN or overlay identifiers may be misinterpreted resulting in the delivery of data to or the acceptance of data from unauthorized network nodes violating intended security constraints. The use of a layer-2 protocol identifier to provide a "fail closed" barrier at the domain border can significantly improve security by eliminating the opportunity for such misinterpretation.

11. IANA Considerations

This document has no IANA actions.

12. References

12.1. Normative References

[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/rfc/rfc2119>.

[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/rfc/rfc8174>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/rfc/rfc8799>.

12.2. Informative References

[IANA_EtherType] "IANA EtherType Registry", Web <https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml#ieee-802-numbers-1>.

[IESG_EtherType] "IESG Statement on EtherTypes", Web <https://www.ietf.org/about/groups/iesg/statements/ethertypes>, 1 May 2023.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/rfc/rfc3031>.

[RFC3682] Gill, V., Heasley, J., and D. Meyer, "The Generalized TTL Security Mechanism (GTSM)", RFC 3682, DOI 10.17487/RFC3682, February 2004, <https://www.rfc-editor.org/rfc/rfc3682>.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, DOI 10.17487/RFC3927, May 2005, <https://www.rfc-editor.org/rfc/rfc3927>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/rfc/rfc4291>.

[RFC5771] Cotton, M., Vegoda, L., and D. Meyer, "IANA Guidelines for IPv4 Multicast Address Assignments", BCP 51, RFC 5771, DOI 10.17487/RFC5771, March 2010, <https://www.rfc-editor.org/rfc/rfc5771>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/rfc/rfc7665>.

[RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <https://www.rfc-editor.org/rfc/rfc8994>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

[RFC9378] Brockners, F., Ed., Bhandari, S., Ed., Bernier, D., and T. Mizrahi, Ed., "In Situ Operations, Administration, and Maintenance (IOAM) Deployment", RFC 9378, DOI 10.17487/RFC9378, April 2023, <https://www.rfc-editor.org/rfc/rfc9378>.

[RFC9542] Eastlake 3rd, D., Abley, J., and Y. Li, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 9542, DOI 10.17487/RFC9542, April 2024, <https://www.rfc-editor.org/rfc/rfc9542>.

Acknowledgments

Much thanks to Deborah Brungard and Brian Carpenter, for their review and comments.

Also much thanks to everyone else with whom we have discussed this topic; I've had numerous discussions with many many people on this, and I'm sure that I've forgotten some of them. Apologies if you were one of them.

Changelog

* 01-02:
  - Add Donald Eastlake as an author.
  - Substantial re-write and expansion of material concerning specific and Extended EtherType protocol identification.
  - Add initial Security Considerations text.
* 00-01:
  - Deborah pointed out that "this only works for transport-type limited domain protocols (e.g., SRv6)" could be read as SRv6 fails-closed.

Authors' Addresses

Warren Kumari
Google, LLC
Email: warren@kumari.net

Andrew Alston
Alston Networks
Email: alston.networks@gmail.com

ric Vyncke
Cisco
Email: evyncke@cisco.com

Suresh Krishnan
Cisco
Email: suresh.krishnan@gmail.com

Donald Eastlake
Independent
Email: d3e3e3@gmail.com