

Operations Area Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: August 7, 2016

F. Gont
SI6 Networks / UTN-FRH
F. Baker
Cisco Systems
February 4, 2016

On Firewalls in Network Security
draft-gont-opsawg-firewalls-analysis-02

Abstract

This document analyzes the role of firewalls in network security, and recognizes their role in the internet architecture. It suggests a line of reasoning about their usage, and analyzes common kinds of firewalls and the claims made for them.

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 <http://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 August 7, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Reasoning about Firewalls	4
3.1. A Simple Model of Communication	4
3.2. The Role of Firewalls in Internet Security	5
3.3. Firewalls and The End-to-End Principle	5
4. Common kinds of firewalls	6
4.1. Perimeter security: Protection from aliens and intruders	7
4.2. Pervasive access control	8
4.3. Intrusion Management: Contract and Reputation filters . .	9
5. Firewalling Strategies	10
5.1. Blocking Traffic Unless It Is Explicitly Allowed (default deny)	11
5.2. Allow Traffic Unless It Is Explicitly Blocked (default allow)	11
6. Assumptions on IP addresses and Transport Protocol Port Numbers	12
7. State Associated with Filtering Rules	13
8. Enforcing Protocol Syntax at the Firewall	14
9. Performing Deep Packet Inspection	14
10. IANA Considerations	15
11. Security Considerations	15
12. Acknowledgements	15
13. References	16
13.1. Normative References	16
13.2. Informative References	16
Authors' Addresses	18

1. Introduction

Prophylactic perimeter security in the form of firewalls, and the proper use of them, have been a fractious sub-topic in the area of internet security. Firewalls have been largely seen by many in the IETF as a poor approach to security, and often as unnecessary and rather "evil" devices that hinder innovation and the deployment of new protocols and applications. Operationally, they are also seen by some as attack vectors, with state exhaustion attacks, side-effects of the imposition of symmetry requirements and single points of failure. This document analyzes the role of firewalls in network security, and recognizes their role in the internet architecture. It suggests a line of reasoning about their usage, and analyzes common kinds of firewalls and the claims made for them.

This document has, among others, the following goals:

- o Recognize the important role of firewalls in enterprise security architecture for providing "prophylactic" security, rather than as "evil" ad-hoc functionality/devices (see Section 3.2).
- o Analyze common kinds of firewalls and claims made for them (see Section 4).
- o Analyze implicit assumptions made by firewalls, identifying where/when some of those assumptions may not apply (see e.g. Section 6).
- o Discuss trade-offs in the possible firewalling paradigms (see Section 5).
- o Provide conceptual guidance regarding the use and deployment of .
- o Identify harmful behavior/policies commonly implemented and applied by firewalls, in the hopes of improving the state of affairs in that area.
- o Possibly trigger other work in the area of firewalls, as a result of the previous items.

2. Terminology

Firewall:

A device or software that imposes a policy whose effect is "a stated type of network traffic may or may not be allowed from A to B". The firewall may reside in the destination itself (a "host firewall"), or in any intermediate system (a "network firewall"). The firewalling functionality may be implemented in a general purpose system (e.g. an ACL in a router), or in a special purpose middleware device (e.g., a "firewall product"). The details of the policy, the granularity with which a policy can be applied, how such policy is configured, or of the firewall's implementation are just that - implementation details.

We also note that a firewall may enforce policies at different layers. Typically, the layer at which a firewall operates will impact the type of policies that a firewall will be able to apply: for example, a layer-3 firewall may be able to enforce simple policies based on layer-3 addresses and some simple layer-4 parameters such as transport protocol port numbers, while an "application firewall" may be able to enforce policies on higher-level entities such as application-request types. We note that all such firewall types essentially enforce the same role of enforcing a policy of some sort on network traffic, and hence are

referred to with the generic term "firewall" (or "firewall device" in some cases) throughout this document.

Perimeter:

The position in which the specific security policy applies. In typical deployed networks, there are usually some easy- to-define perimeters. A network connected with another network has a perimeter where the two meet, which is defined by what equipment is operated by each network. It invariably imposes a security policy at that boundary, which may be as simple as "all traffic is welcome" and as complex as matching arriving and departing traffic to ensure specific behaviors, or inspecting traffic according to various algorithms. Firewall functionality is usually implemented at or close to such network perimeters.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Reasoning about Firewalls

3.1. A Simple Model of Communication

Any communication requires at least three components:

- o a sender, someone or some thing that sends a message,
- o a receiver, someone or some thing that receives the message, and
- o a channel, which is a medium by which the message is communicated.

In the Internet, the IP network is the channel; it may traverse something as simple as a directly connected cable or as complex as a sequence of ISPs, but it is the means of communication. In normal communications, a sender sends a message via the channel to the receiver, who is willing to receive and operate on it. In contrast, attacks are a form of harassment. A receiver exists, but is unwilling to receive the message, has no application to operate on it, or is by policy unwilling to. Attacks on infrastructure occur when message volume overwhelms infrastructure or uses infrastructure but has no obvious receiver.

By that line of reasoning, a firewall operating at layer-3 primarily protects infrastructure, by preventing traffic that would attack it from it. The best prophylactic might use a procedure for the dissemination of Flow Specification Rules [RFC5575] to drop traffic sent by an unauthorized or inappropriate sender or which has no host

or application willing to receive it as close as possible to the sender.

In other words, a firewall is comparable to the human skin, and has as its primary purpose the prophylactic defense of a network. By extension, the firewall also protects a set of hosts and applications, and the bandwidth that serves them, as part of a strategy of defense in depth. Since there is no one way to prevent attacks, a firewall is not itself a security strategy; the analogy to the skin would say that a body protected only by the skin has an immune system deficiency and cannot be expected to long survive. That said, every security solution has a set of vulnerabilities; the vulnerabilities of a layered defense is the intersection of the vulnerabilities of the various layers (e.g., a successful attack has to thread each layer of defense).

3.2. The Role of Firewalls in Internet Security

One could compare the role of firewalls in prophylactic perimeter security to that of the human skin: the service that the skin performs for the rest of the body is to keep common crud out, and as a result prevent much damage and infection that could otherwise occur. The body supplies prophylactic perimeter security for itself and then presumes that the security perimeter has been breached; real defenses against attacks on the body include powerful systems that detect changes (anomalies) counterproductive to human health, and recognizable attack syndromes such as common or recently-seen diseases. One might well ask, in view of those superior defenses, whether there is any value in the skin at all; the value is easily stated, however. It is not in preventing the need for the stronger solutions, but in making their expensive invocation less needful and more focused.

3.3. Firewalls and The End-to-End Principle

One common complaint about firewalls in general is that they violate the End-to-End Principle [Saltzer]. The End-to-End Principle is often incorrectly stated as requiring that "application specific functions ought to reside in the end nodes of a network rather than in intermediary nodes, provided they can be implemented 'completely and correctly' in the end nodes" or that "there should be no state in the network." What it actually says is heavily nuanced, and is a line of reasoning applicable when considering any two communication layers.

[Saltzer] "presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that

functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level."

In other words, the End-to-End Argument is not a prohibition against lower layer retries of transmissions, which can be important in certain LAN technologies, nor of the maintenance of state, nor of consistent policies imposed for security reasons. It is, however, a plea for simplicity. Any behavior of a lower communication layer, whether found in the same system as the higher layer (and especially application) functionality or in a different one, that from the perspective of a higher layer introduces inconsistency, complexity, or coupling, extracts a cost. That cost may be in user satisfaction, difficulty of management or fault diagnosis, difficulty of future innovation, reduced performance, or something else. Such costs need to be clearly and honestly weighed against the benefits expected, and used only if the benefit outweighs the cost.

From that perspective, introduction of a policy that prevents communication under an understood set of circumstances, whether it is to prevent access to pornographic sites or to prevent traffic that can be characterized as an attack, does not fail the End-to-End Argument; there are any number of possible sites on the network that are inaccessible at any given time, and the presence of such a policy is easily explained and understood.

What does fail the End-to-End Argument is behavior that is intermittent, difficult to explain, or unpredictable. If a site can be reached sometimes and not at other times, or can be reached using this host or application but not another, one will wonder why that is the case, and may not even know where to look for the issue.

4. Common kinds of firewalls

There are at least three common kinds of firewalls:

- o Context or Zone-based firewalls, that protect systems within a perimeter from systems outside it,
- o Pervasive routing-based measures, which protect intermingled systems from each other by enforcing role-based policies, and
- o Systems that analyze network traffic behavior and trigger on events that are unusual, match a signature, or involve an untrusted peer.

Each kind of firewall addresses a different view of the network. A zone-based firewall (Section 4.1) views the network as containing

zones of trust, and deems applications inside its zone of protection to be trustworthy. A role-based firewall (Section 4.2) identifies parties on the basis of membership in groups, and prevents unauthorized communication between groups. A reputation, anomaly, or signature-based intrusion management system (Section 4.3) depends on active administration, and permits known applications to communicate while excluding unknown or known-evil applications. In each case, the host or application is its own final bastion of defense, but having a host blocking incoming traffic (so-called "host firewalls") does not defend infrastructure. That is, each type of prophylactic has a purpose, and none of them is a complete prophylactic defense.

Each type of defense, however, can be assisted by enabling an application running in a host to inform the network of what it is willing to receive. As noted in Section 4.1, a zone-based firewall, generally denies all incoming sessions and permits responses to sessions initiated outbound from the zone, but can in some cases be configured to also permit specific classes of incoming session requests, such as WWW or SMTP to an appropriate server. A simple way to enable a zone-based firewall to prevent attacks on infrastructure (traffic to an un-instantiated address or to an application that is off) while not impeding traffic that has a willing host and application would be for the application to inform the firewall of that willingness to receive incoming sessions. The Port Control Protocol [RFC6887], or PCP, is an example of a protocol designed for that purpose.

4.1. Perimeter security: Protection from aliens and intruders

As discussed in [RFC6092], the most common kind of firewall is used at the perimeter of a network. Perimeter security assumes two things: that applications and equipment inside the perimeter are under the control of the local administration and are therefore probably doing reasonable things, and that applications and equipment outside the perimeter are unknown.

For example, it may enforce simple permission rules, such as that external web clients are permitted to access a specific web server or that external SMTP MTAs are permitted to access internal SMTP MTAs. Apart from those rules, a session may be initiated from inside the perimeter, and responses from outside will be allowed through the firewall, but sessions may never be initiated from outside.

In addition, perimeter firewalls often perform some level of inspection/analysis, either as application proxies or through deep packet inspection, to verify that the protocol claimed to be being passed is in fact the protocol being passed.

In many scenarios the existence and definition of zone-based perimeter defenses is arguably a side-effect of the deployment of Network Address Translation [RFC2993]. Since e.g. a single address is shared among multiple systems, the NAT device needs to translate both the IP addresses and the transport protocol ports in order to multiplex multiple communication instances from different nodes into the same external address. Thus, the NAT device must keep a state table to know how to translate the IP addresses and transport protocol ports of incoming packets. Packets originating from the internal network will either match an existing entry in the state table, or create a new one. On the other hand, packets originating in the external network will either match an existing entry in the state table, or be dropped. Thus, as a side effect, NATs implicitly require that communication be initiated from the internal network, and only allow return traffic from the external network. We note that this is a side-effect of multiplexing traffic from multiple nodes on a single IP address, rather than a design goal of NAT devices or their associated network translation function.

Some applications make the mistake of coupling application identities to network layer addresses, and hence employ such addresses in the application protocol. Thus, Network Address Translation forces the translator to interpret packet payloads and change addresses where used by applications.

As a result, if the transport or application headers are not understood by the translator, this has the effect of damaging or preventing communication. Detection of such issues can be sold as a security feature, although it is really a side-effect of a failure. While this can have useful side-effects, such as preventing the passage of attack traffic that masquerades as some well-known protocol, it also has the nasty side-effect of making innovation difficult. This has slowed the deployment of SCTP [RFC4960], since a firewall will often not permit a protocol it does not know even if a user behind it opens the session. When a new protocol or feature is defined, the firewall needs to stop applying that rule, and that can be difficult to make happen.

4.2. Pervasive access control

Another access control model, often called "Role-based", tries to control traffic in flight regardless of the perimeter. Given a rule that equipment located in a given routing domain or with a specific characteristic (such as "student dorms") should not be able to access equipment in another domain or with a specific characteristic (such as "academic records"), it might prevent routing from announcing the second route in the domain of the first, or it might tag individual packets ("I'm from the student dorm") and filter on those tags at

enforcement points throughout network. Such rules can be applied to individuals as well as equipment; in that case, the host needs to tag the traffic, or there must be a reliable correlation between equipment and its user.

One common use of this model is in data centers, in which physical or virtual machines from one tenant (which is not necessarily an "owner" as much as it is a context in which the system is used) might be co-resident with physical or virtual machines from another. Inter-tenant attacks, espionage, and fraud are prevented by enforcing a rule that traffic from systems used by any given tenant is only delivered to other systems used by the same tenant. This might, of course have nuances; under stated circumstances, identified systems or identified users might be able to cross such a boundary.

The major impediment in deployment is complexity. The administration has the option to assign policies for individuals on the basis of their current location (e.g. as the cross-product of people, equipment, and topology), meaning that policies can multiply wildly. The administrator that applies a complex role-based access policy is probably most justly condemned to live in the world he or she has created.

4.3. Intrusion Management: Contract and Reputation filters

The model proposed in Advanced Security for IPv6 CPE [I-D.vyncke-advanced-ipv6-security] could be compared to purchasing an anti-virus software package for one's computer. The proposal is to install a set of filters, perhaps automatically updated, that identify "bad stuff" and make it inaccessible, while not impeding anything else.

It depends on four basic features:

- o A frequently-updated signature-based Intrusion Prevention System which inspects a pre-defined set of protocols at all layers (from layer-3 to layer-7) and uses a vast set of heuristics to detect attacks within one or several flows. Upon detection, the flow is terminated and an event is logged for further optional auditing.
- o A centralized reputation database that scores prefixes for degree of trust. This is unlikely to be on addresses per se, since e.g. temporary addresses [RFC4941] change regularly and frequently.
- o Local correlation of attack-related information, and
- o Global correlation of attacks seen, in a reputation database.

The proposal does not mention anomaly-based intrusion detection, which could be used to detect zero-day attacks and new applications or attacks. This would be an obvious extension.

The comparison to anti-virus software is real; anti-virus software uses similar algorithms, but on API calls or on data exchanged rather than on network traffic, and for identified threats is often effective.

The proposal also has weaknesses:

- o People do not generally maintain anti-virus packages very well, letting contracts expire,
- o Reputation databases have a bad reputation for distributing information which is incorrect, out of date, or compromised by attackers,
- o Anomaly-based analysis identifies changes but is often ineffective in determining whether new application or application behaviors are pernicious (false positives). Someone therefore has to actively decide - a workload the average homeowner might have little patience for, and
- o Signature-based analysis applies to attacks that have been previously identified, and must be updated as new attacks develop. As a result, in a world in which new attacks literally arise daily, the administrative workload can be intense, and reflexive responses like accepting https certificates that are out of date or the download and installation of unsigned software on the assumption that the site administrator is behind are themselves vectors for attack.

Security has to be maintained to be useful, because attacks are maintained.

5. Firewalling Strategies

There is a great deal of tension in firewall policies between two primary goals of networking: the security goal of "block traffic unless it is explicitly allowed" and the networking goal of "trust hosts with new protocols". The two inherently cannot coexist easily in a set of policies for a firewall.

The following subsections discuss the "default deny" and "default allow" security paradigms.

5.1. Blocking Traffic Unless It Is Explicitly Allowed (default deny)

Many networks enforce the so-called "default deny" policy, in which traffic is blocked unless it is explicitly allowed. The rationale for such policy is that it is easier to open "holes" in a firewall to allow specific protocols, than trying to block all protocols that might be employed as an attack vectors; and that a network should only support the protocols it has been explicitly meant to support.

The drawback of this approach is that the security goal of "block traffic unless it is explicitly allowed" prevents useful new applications. This problem has been seen repeatedly over the past decade: a new and useful application protocol is specified, but it cannot get wide adoption because it is blocked by firewalls. The result has been a tendency to try to run new protocols over established applications, particularly over HTTP [RFC3205]. The result is protocols that do not work as well they might if they were designed from scratch.

Worse, the same goal prevents the deployment of useful transports other than TCP, UDP, and ICMP. A conservative firewall that only knows those three transports will block new transports such as SCTP [RFC4960]; this in turn causes the Internet to not be able to grow in a healthy fashion. Many firewalls will also block TCP and UDP options they don't understand, and this has the same unfortunate result.

5.2. Allow Traffic Unless It Is Explicitly Blocked (default allow)

Some networks enforce the so-called "default allow" policy, in which traffic is allowed unless it is explicitly blocked. This policy is usually enforced at perimeters where a comprehensive security policy is not really desirable or possible, but some level of packet filtering is considered appropriate. One common example of such policy could be an ISP blocking TCP port 25 (SMTP), but allowing all other traffic.

When a strict security policy is to be enforced (e.g., at an organizational network's edge), the "default allow" policy tends to be rather inappropriate, since it is usually easier and more effective to identify the traffic that must be allowed through the firewall (and open the necessary "holes" in the firewall) than to identify and block all traffic that may be considered undesirable/inappropriate.

6. Assumptions on IP addresses and Transport Protocol Port Numbers

In a number of scenarios, simple firewall rules have traditionally been specified in terms of the associated IP addresses and transport protocol port numbers. In general, this assumes that the associated IP addresses are stable, and that there is a "well known" transport protocol port number associated with each application.

In the IPv4 world, IP addresses may be considered rather stable. However, IPv6 introduces the concept of "temporary addresses" [RFC4941] which, by definition, change over time. This may prevent the enforcement of filtering policies based on specific IPv6 addresses, or may lead to filtering based on a more coarse granularity (e.g. specific address prefixes, as opposed to specific IPv6 addresses). In some scenarios, from the point of view of enforcing filtering policies, it might be desirable to disable temporary addresses altogether.

For example, an administrator might prefer that a caching DNS server, a secondary DNS server doing zonetransfers, or an SMTP MTA, always employ the same source IPv6 address, as opposed to the temporary addresses that change over time.

The server-side transport protocol port is generally the so-called "well-known port" corresponding to the associated application. While widespread, this practice should probably be considered a kludge/short-cut rather than a "design principle" that can be relied upon for the general case. For example, use of DNS SRV records [RFC2782], or applications such as "portmapper" [Portmap] [RFC1833] might mean that the associated transport protocol port number cannot be assumed to be well-known, but rather needs to be dynamically learned. In other cases, applications may employ (by design) ephemeral port numbers, and there may be no obvious way to dynamically learn the port number being employed. FTP [RFC0959] and SIP [RFC3261] are examples of such applications.

Finally, as a result of widespread packet filtering, many protocols tend to be tunneled employing specific transport-protocol port numbers that are known to be more generally allowed by firewalls, such as TCP port 80 (HTTP). This essentially breaks the assumption that port numbers actually identify the actual application protocol using them.

Some of the so called "next generation" firewalls make fewer assumptions about port numbers, and tend to analyze the application data stream in order to infer the application protocol type, regardless of the well-known port being used. While this may prevent the circumvention of some security controls, it also implies Deep

Packet Inspection (DPI), and therefore there are a number of associated considerations, both in terms of introduced attack vectors and other possibilities for evasion of security controls (please see Section 9 for further discussion).

7. State Associated with Filtering Rules

There are two main paradigms for packet filtering:

- o Stateless filtering
- o Stateful filtering

Stateless filtering implies that the decision on whether to allow or block a specific traffic entity is based solely on the contents of such entity. One common example of such paradigm is the enforcement of network ingress filtering [RFC2827], in which packets may be blocked based on their IP addresses. Stateless filtering scales well, since there are no state requirements on the filtering device other than that associated with maintaining the filtering rules to be applied to the incoming traffic entities (e.g., packets).

On the other hand, stateful filtering implies that the decision on whether to allow or block a traffic entity is not only based on the contents of such entity, but also on the existence (or lack of) previous state associated with such entity. A common example of such paradigm is a firewall that "allows outbound connection requests and only allows return traffic from the external network" (such as the policy implicitly enforced by most NAT devices). For obvious reasons, the firewall needs to maintain state in order to be able to enforce such policies; that is, the firewall may need to keep track of all on-going communication instances, possibly applying timeouts and garbage collection on the associated state table.

Stateful filtering tends to allow more powerful packet filtering, at the expense of increased state. Thus, stateful filtering may be desirable when trying to perform deep packet inspection, but may be undesirable when the firewall is meant to block some Denial of Service attacks, since the firewall itself may become "the weakest link in the chain". Typically, the higher the firewall operates in the network stack, the more state will be required associated. For example, in order for a firewall to enforce a filtering policy based on application-layer request types, the firewall will need to enforce its filtering policy on the application-layer protocol stream, thus implying the need to perform layer-3 and layer-4 reassembly, etc.

When stateful packet filtering is warranted, its associated security implications should be considered. For example, an administrator may

want to enforce traffic filtering to mitigate denial of service attacks; however, when enforcement of such filtering implies increased state at the firewall, the firewall itself may become the easiest target for performing a denial of service attack.

8. Enforcing Protocol Syntax at the Firewall

Some firewalls try to enforce the protocol syntax by checking that only traffic complying with existing protocol definitions is allowed. While this can have useful side-effects, such as preventing the aforementioned traffic from triggering pathological behavior at the target system, it also has the nasty side-effect of making innovation difficult. For example, one of the issues in the deployment of Explicit Congestion Notification [RFC3168] has been that common firewalls often inspect reserved/unused bits and require them to be set to zero to close covert channels. Another example is the plethora of filtering rules applied to DNS traffic [DNS-FILTERING]. When a new protocol or feature is defined, the firewall needs to stop applying that rule, and that can be difficult to make happen.

NOTE:

A somewhat related concept is that of traffic normalization (or "scrubbing"), in which the filtering device can "normalize" traffic by e.g. clearing bits that are expected to be cleared, changing some protocol fields such that they are within "normal" ranges, etc. (see e.g. the discussion of "traffic normalization" in [OpenBSD-PF]). While this can have the useful effect of blocking DoS attacks to sloppy implementations that do not enforce sanity checks on the received packets, it also has the nasty side-effect of making innovation difficult, or even breaking deployed protocols. For example, some firewalls are known to enforce a default packet normalization policy that clears the TCP URG bit, as a result of the TCP urgent mechanism being associated with some popular DoS attacks. Widespread deployment of such firewalls has essentially rendered the TCP urgent mechanism unusable, leading to its eventual formal deprecation in [RFC6093].

We note that, as per our definition of "firewall" in Section 2, "traffic normalization" is not considered a firewall function.

9. Performing Deep Packet Inspection

While filtering packets based on the layer-3 protocol header fields is rather simple and straight-forward, performing enforcing a filtering policy at upper layer protocols can be a challenging task.

For example, IP fragmentation may make this task quite challenging, since even the very layer-4 protocol header could be present in a

non-first fragment. In a similar vein, IPv6 extension headers may represent a challenge for a filtering device, since they can result in long IPv6 extension header chains [RFC7112] [I-D.gont-v6ops-ipv6-ehs-packet-drops].

This problem is exacerbated as one tries to filter packets based on upper layer protocol contents, since many of such protocols implement some form of fragmentation/segmentation and reassembly. In many cases, the reassembly process could possibly lead to different results, and this may be exploited by attackers for circumventing security controls [Ptacek1998] [RFC6274].

In general, the upper in the protocol stack that a filtering policy is to be enforced, the more complex the task becomes: an attacker has more opportunities for obfuscation, ranging from e.g. ambiguities in IP and/or TCP reassembly, to e.g. application-layer obfuscation (such as HTTP URL obfuscation or JavaScript bytecode obfuscation). This usually implies that, in order to reliably enforce a filtering policy, more state is required on the firewall; and the considerations in Section 7 should be evaluated.

10. IANA Considerations

This memo asks the IANA for no new parameters. It can before publication as an RFC by the RFC Editor.

11. Security Considerations

This documents recognizes the role of firewalls in network security, and discusses a number of considerations associated with firewalls which may be of use when designing or deploying firewalls. This document, by itself, does not introduce any security implications.

12. Acknowledgements

The authors would like to thank (in alphabetical order) Fleming Andraeson, Mark Andrews, Lee Howard, Joel Jaeggli, Al Morton, Eric Vyncke and James Woodyatt, for providing valuable comments on earlier versions of this document.

This document is based on [I-D.ietf-opsawg-firewalls-00] authored by Fred Baker, and [I-D.ietf-opsawg-firewalls-01] authored by Paul Hoffman.

13. References

13.1. Normative References

- [RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, DOI 10.17487/RFC1833, August 1995, <<http://www.rfc-editor.org/info/rfc1833>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3205] Moore, K., "On the use of HTTP as a Substrate", BCP 56, RFC 3205, DOI 10.17487/RFC3205, February 2002, <<http://www.rfc-editor.org/info/rfc3205>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<http://www.rfc-editor.org/info/rfc4941>>.
- [RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <<http://www.rfc-editor.org/info/rfc6887>>.
- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <<http://www.rfc-editor.org/info/rfc7112>>.

13.2. Informative References

- [DNS-FILTERING]
Andrews, M., "On Firewalls in Internet Security (Fwd: New Version Notification for draft-gont-opsawg-firewalls-analysis-00.txt)", post to the OPSAWG mailing-list, Message-Id: <20151012002551.8F7CD3A2FFD8@rock.dv.isc.org>, 2015, <<https://mailarchive.ietf.org/arch/msg/opsawg/2YQl6xBz6jtMyIkyAx59U-oPmPQ>>.
- [I-D.gont-v6ops-ipv6-ehs-packet-drops]
Gont, F., Hilliard, N., Doering, G., LIU, S., and W. Kumari, "Operational Implications of IPv6 Packets with Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-drops-02 (work in progress), February 2016.

- [I-D.ietf-opsawg-firewalls-00]
Baker, F., "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-00 (work in progress), June 2012.
- [I-D.ietf-opsawg-firewalls-01]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-01 (work in progress), October 2012.
- [I-D.vyncke-advanced-ipv6-security]
Vyncke, E., Yourtchenko, A., and M. Townsley, "Advanced Security for IPv6 CPE", draft-vyncke-advanced-ipv6-security-03 (work in progress), October 2011.
- [OpenBSD-PF]
OpenBSD, , "pf(4) manual page: pf -- packet filter", 2015, <<http://www.openbsd.org/cgi-bin/man.cgi/OpenBSD-current/man4/pf.4&query=pf>>.
- [Portmap] Wikipedia, , "Portmap", 2014, <<https://en.wikipedia.org/wiki/Portmap>>.
- [Ptacek1998]
Ptacek, T. and T. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", 1998, <<http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps>>.
- [RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, DOI 10.17487/RFC0959, October 1985, <<http://www.rfc-editor.org/info/rfc959>>.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <<http://www.rfc-editor.org/info/rfc2782>>.
- [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, <<http://www.rfc-editor.org/info/rfc2827>>.
- [RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <<http://www.rfc-editor.org/info/rfc2993>>.

- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<http://www.rfc-editor.org/info/rfc3168>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<http://www.rfc-editor.org/info/rfc4960>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<http://www.rfc-editor.org/info/rfc5575>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<http://www.rfc-editor.org/info/rfc6092>>.
- [RFC6093] Gont, F. and A. Yourtchenko, "On the Implementation of the TCP Urgent Mechanism", RFC 6093, DOI 10.17487/RFC6093, January 2011, <<http://www.rfc-editor.org/info/rfc6093>>.
- [RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, DOI 10.17487/RFC6274, July 2011, <<http://www.rfc-editor.org/info/rfc6274>>.
- [Saltzer] Saltzer, J., Reed, D., and D. Clark, "End-to-end arguments in system design", ACM Transactions on Computer Systems (TOCS) v.2 n.4, p277-288, Nov 1984.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: <http://www.si6networks.com>

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 7, 2016

P. Liang
ICANN
A. Melnikov
Isode Ltd
D. Conrad
ICANN
July 6, 2015

Private Enterprise Number (PEN) practices and Internet Assigned Numbers
Authority (IANA) registration considerations
draft-liang-iana-pen-06

Abstract

Private Enterprise Numbers (PENs) are a technical protocol parameter frequently assigned for use in the management of network connected equipment or software via SNMP-based network management systems, LDAP, DIAMETER or GSS-API. This document discusses what a Private Enterprise Number (PEN) is, common uses of PENs, and registration procedures for IANA Considerations. The registration procedures include instructions and requirements for obtaining a new Private Enterprise Number, modifying existing numbers, and the removal of existing numbers.

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 <http://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 January 7, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Introduction to Private Enterprise Numbers	3
2.1. Various uses of PENs "in the wild"	3
3. PEN Assignment	5
3.1. Assignment of a New PEN	5
3.2. Update of an Assigned PEN	7
3.3. Removals of Private Enterprise Numbers	8
4. Registration in the Private Enterprise Number registry	8
4.1. Registration of PEN	8
4.2. Syntax for Private Enterprise Names and PENs	9
5. Acknowledgements	9
6. IANA Considerations	9
6.1. Historical Assignments	10
7. Security Considerations	10
8. References	10
8.1. Normative References	10
8.2. Informative References	11
Authors' Addresses	11

1. Introduction

A Private Enterprise Number (also known as a "PEN"), is a non-negative integer, unique within the `iso.org.dod.internet.private.enterprise (1.3.6.1.4.1)` Object Identifiers (OIDs) subtree of the ISO Object Identifier (OID) hierarchy. This hierarchy, jointly developed by ITU-T and ISO/IEC was developed to name "any type of object, concept or 'thing' with a globally unambiguous name which requires a persistent name" (See <http://www.oid-info.com/#oid>). The sub-tree for which the IETF is the Registration Authority, originally defined in [RFC1065], is used to allow any entity to obtain a globally unique identifier to reference an organization ("enterprise") in protocols.

To date, the procedures for the assignment of new PENs and the modification of assigned PENs have not been clearly documented. Private Enterprise Numbers are referenced in RFCs [RFC1157] [RFC1213]

and [RFC2578]. These documents primarily define Simple Network Management Protocol (SNMP), Management Information Base (MIB) and Structure of Management Information (SMI) structures. As such, none of these RFCs clearly describe PENs nor do they define PEN registration procedures.

As a result of the lack of documented process, updates to assigned PENs can be challenging. Given there are no clear registration requirements, it can be difficult to validate change requests, particularly in cases such as updates to organization names or legal ownership, changes to email addresses of the registered PEN owner, etc.

This document introduces PENs, how they are commonly used, and their registration and update procedures.

2. Introduction to Private Enterprise Numbers

PENs are frequently embedded in OIDs (Object Identifiers) , which are most often used in Simple Network Management Protocol (SNMP) Management Information Base (MIB) configurations. However, PENs are not designed to be used exclusively for SNMP purposes, but rather they can be and are used by a variety protocols and Data Manipulation Languages. There is no restriction for using private enterprise numbers for other protocols or data models than SNMP or MIB.

If the OID is only to be used privately, then enterprise numbers are to be used. PEN is a number under the prefix 1.3.6.1.4.1. and PEN appears as follows:

```
Prefix: iso.org.dod.internet.private.enterprise.(Your node)
1.3.6.1.4.1.xxxx
```

IANA only manages and maintain this hierarchy tree under the IESG guidelines. There are many other prefixes, such as 2.16.840.1113883, 1.2.840.113549.1.9.16.2.21, etc., under completely different arcs and managed by other repositories (which might or might not be managed by IANA). This document doesn't cover management of these other repositories.

2.1. Various uses of PENs "in the wild"

As some examples documented on Wikipedia, the most common OIDs seen "in the wild" usually belong to the private enterprise numbers allocated by IANA under the 1.3.6.1.4.1 (iso.org.dod.internet.private.enterprise) tree. Increasingly, an OID with health care and public health informatics in the United States is being used. Health Level Seven (HL7), a standards-developing

organization in the area of electronic health care data exchange is an assigning authority at the 2.16.840.1.113883 (joint-iso-itu-t.country.us.organization.hl7) tree.

It is important to note that despite the name PENs do not necessarily represent a manufacturer or Vendor ID. For example they can represent organizations and even independent developers.

The registrant of a Private Enterprise Number can create sub-trees by appending a "." along with unique numbers at the end of their PEN, i.e. to perform its own sub-allocations. For example, for LDAP, the registrant of PEN <PEN> can use:

iso.org.dod.internet.private.enterprise.<PEN>.1 for LDAP Object Classes

iso.org.dod.internet.private.enterprise.<PEN>.2 for LDAP attribute types

iso.org.dod.internet.private.enterprise.<PEN>.3 for LDAP syntaxes

A particular Object class can have OID:

iso.org.dod.internet.private.enterprise.<PEN>.1.100

iso.org.dod.internet.private.enterprise.<PEN>.1.200 for subsidiaries an/or divisions

In general any number of additional levels are permitted, for example:

iso.org.dod.internet.private.enterprise.<PEN>.1.1 can be used as a parent OID for all email related object classes, and

iso.org.dod.internet.private.enterprise.<PEN>.1.2 can be used for web related object classes.

iso.org.dod.internet.private.enterprise.<PEN>.1.3 can be used for instant messaging related object classes, etc.

Below are more example uses of PENs:

Distinguished Names and other components in X.509 certificates;

Various schema elements in X.500/LDAP directories;

GSS-API

extensions to DIAMETER

PA-TNC [RFC5792] and PB-TNC [RFC5793]

Important to note that how the numbers are used is up to the various implementers and companies building products. Neither ICANN or the IETF can police how people use the numbers out in the wild. The parties in question should resolve any inappropriate usage among themselves, and ICANN and the IETF have no role in such disputes.

3. PEN Assignment

Assignments of PENs are done by the Internet Assigned Numbers Authority (IANA). This section provides information relating to the assignment of new PENs and the requirements associated with updating already assigned PENs.

3.1. Assignment of a New PEN

PENs are assigned through a "First Come First Served" registration policy as described in [RFC5226]. They are assigned sequentially. There is no opportunity to request a particular private enterprise number.

A PEN can be requested by individuals or organizations in order to obtain a unique value for their "enterprise". Requests for new PENs can be submitted via an automated form at IANA.

In order to facilitate appropriate registration, and in particular, subsequent update of an assigned PEN, a small amount of information is required. This information includes the name and contact information of the requesting organization (or individual), the name of the contact person for the PEN, and an e-mail address of the contact.

Historically, users submit a program name, product, project, and random abbreviation as the organization name to when applying for a PEN. This practice is discouraged since multiple programs, product, and/or projects can have their own sub-trees under the PEN assigned to the organization (or individual), thus there is rarely a need for an organization to have multiple IANA-assigned PENs.

Before requesting additional OIDs, IANA encourages the identification of any existing OID assignment(s) to the requesting organization (or individual) and the creation of sub-trees where possible and appropriate. IANA may decline the allocation of new PENs to organizations that have existing registrations unless justification for multiple allocations is provided.

The following information will be requested for a new registration:

Registrant (Company/Organization) Name in ASCII (REQUIRED)

UTF-8 version of the Registrant (Company/Organization) Name
(OPTIONAL)

Registrant (Company/Organization) E-mail Address (REQUIRED)

Registrant Postal Address (REQUIRED)

Contact Name (REQUIRED)

Contact E-mail Address (REQUIRED)

Contact Postal Address (OPTIONAL)

Contact Phone Number (OPTIONAL)

Reference (OPTIONAL)

Comments (OPTIONAL)

Registrant (Company/Organization) Name: The name of the organization or individual responsible for the registration of Private Enterprise Number. If the organization is a company, it should be the full legal name including "Inc.", "Ltd.", etc.

UTF-8 version: If a UTF-8 version of the company name is available, the requester can provide the UTF-8 name. This will be listed in the registry.

Registrant (Company/Organization) E-mail Address: An e-mail address belonging to the organization that requests the PEN. This e-mail address will be publicly available in the IANA PEN Registry. The E-mail address should be a valid email address and can be a role account e-mail address.

Registrant Postal Address: The postal address/location of the organization/individual requesting the PEN. This information is only used by IANA for verification and will be kept private.

Contact Name: Name of the individual who will be responsible for the PEN on behalf of the company. This Contact person is authorized to submit changes on behalf of the Registrant (Company/Organization) described above.

Contact E-Mail Address: The e-mail address of the individual responsible for the PEN. The e-mail address must be one the Contact person can email confirmation from. This e-mail address will be publicly available in the IANA PEN Registry. The Contact E-mail Address can be the same one as the Registrant's E-mail address.

Contact Postal Address: The full postal address of the individual responsible the PEN, including state/province, zip/postal code, country, etc.

Contact Phone: The telephone number (with extension where appropriate) of the individual responsible for the PEN, including country code.

Reference: A document associated with the implementation of the OID can be referenced with the registration.

Comments: This field will contain the old Registrant/Company Name associated with a PEN if applicable.

It is recommended that a single PEN is granted per organization. IANA does not expect to allocate additional PENs to the same Registrants (Companies/Organizations) that have existing PEN records listed in the IANA PEN registry.

3.2. Update of an Assigned PEN

When a Company/Organization has been merged or acquired by another enterprise, the Registrant (Company/Organization) Name can be annotated in the registry. IANA will verify the requested changes, and, if it deems to be necessary, official letters from the existing owner might be required. It is not guarantee that the request will be granted if IANA does not have sufficient information to verify the changes, or if there is legacy use of the PEN out in the wild.

All information associated with existing PEN records, excluding the Registrant (Company/Organization) Name, shall be updated if the information is obsoleted. (See the preceding section to update the Registrant (Company/Organization) Name.) A request to update Contact information associated with an existing PEN record shall be submitted via an automated form at IANA. Requests can only be fulfilled upon verification by IANA and/or subject matter experts. Additional documentations will be required if it deems to be necessary to validate the request.

A change to the Contact Name of existing PEN records can be made to IANA in case of personnel changes, change of employment, acquisitions, etc. It would be ideal that new requests shall be

completed by the existing Contacts for the PEN records. E-mail verifications of the requested changes are required. Alternatively, supplemental documentations and/or letters issued by the Company/Organization (Registrant Name) will be required if E-mail verifications cannot be fulfilled and if it deems to be necessary.

3.3. Removals of Private Enterprise Numbers

Such request does not happen often and regularly.

Considering the fact that there might be legacy uses of any existing allocation, registrations SHOULD NOT be removed.

A Contact Name can request to remove the corresponding Contact information if the company is no longer in operation, the Contact does not wish to be listed in the IANA PEN registry and if the PEN is no longer believed to be in use. The Modification procedure described above SHOULD be followed.

Requests can only be fulfilled upon verification by IANA and/or subject matter experts if it deems to be necessary.

IF the removal request is honoured, the entry is marked as "Unassigned" and annotated as "returned on yyyy-mm-dd by xxxxxxxx". A future update to this document can allow IANA to reallocate such returned PEN, however this document doesn't allow for that.

4. Registration in the Private Enterprise Number registry

4.1. Registration of PEN

The registry table consists of a list of the following properties:

PEN number

Registrant (Company/Organization) Name (in ASCII)

UTF-8 version of the Registrant (Company/Organization) Name

Registrant (Company/Organization) E-mail Address (REQUIRED)

Contact Name

Contact E-mail Address

Date Assigned

Date Modified

Reference

Comments

NOTE: See Section 3.1 for definition of these properties.

o Values marked as "Reserved" (excluding value zero) in the registry can not be reassigned to a new company or individual without consulting IESG (or expert(s) designated by IESG). Reserved entries mark entries with unclear ownership.

o Value "Unassigned" SHOULD NOT be re-assigned unless specified otherwise, i.e. when the available pool of PENs runs out.

4.2. Syntax for Private Enterprise Names and PENs

o UTF-8 Names of Private Enterprises MUST satisfy the requirements of the NicknameFreeformClass [I-D.ietf-precis-nickname]. (Basically, this means that all ASCII letters, ASCII digits, ASCII punctuation characters, Unicode symbols are allowed.)

o Names of Private Enterprises MUST NOT begin or end with a hyphen

o Maximum value for PENs is hereby defined within 2**32-1 with 0 and 0xFFFFFFFF (in hex) marked as Reserved. (Note that while the original PEN definition has no upper bound, this document defines the upper bound, because some protocol make assumptions about how big PENs can be. For example, DIAMETER [RFC3588] assumes that this value is no bigger than 2**32-1.)

5. Acknowledgements

The authors would like to thank Dan Romascanu, Michelle Cotton, and Bert Wijnen for their contributions to this document.

6. IANA Considerations

This document requests IANA to update the PEN online template forms both NEW and Modification as defined in sections Section 3.1 and Section 3.2.

The PEN registry should be updated to include the information as defined in Section 4.1.

6.1. Historical Assignments

This document will correct the missing historical assignments that predates ICANN's management of the existing registry. These entries will be marked as "Reserved" and annotated as "Returned on yyyy-mm-dd" in the registry. These numbers MAY be re-assigned when the available pool of PENs runs out upon instructions from IESG (or IESG assigned expert(s)).

2187, 2188, 3513, 4164, 4565, 4600, 4913, 4999, 5099, 5144, 5201, 5683, 5777, 6260, 6619, 14827, 16739, 26975

The range from 11670 to 11769

7. Security Considerations

See the Security Considerations section in BCP 26 [RFC5226], and note that improper definition and application of IANA registration policies can introduce both interoperability and security issues. It is critical that registration policies be considered carefully and separately for each registry. Overly restrictive policies can result in the lack of registration of code points and parameters that need to be registered, while overly permissive policies can result in inappropriate registrations. Striking the right balance is an important part of document development.

As mentioned in a preceding section, given there are no clear registration requirements in the past, only limited information is recorded, significant out-of-date information is listed in the registry, and there is no strong authentication mechanism in place, the implications (if any) of the theft of PENs is possible. There is a possibility that the registration data can be transferred to someone else unintentionally.

8. References

8.1. Normative References

- [I-D.ietf-precis-nickname]
Saint-Andre, P., "Preparation and Comparison of Nicknames", draft-ietf-precis-nickname-09 (work in progress), January 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

8.2. Informative References

- [RFC1065] Rose, M. and K. McCloghrie, "Structure and identification of management information for TCP/IP-based internets", RFC 1065, August 1988.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", STD 15, RFC 1157, May 1990.
- [RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets:MIB-II", STD 17, RFC 1213, March 1991.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.
- [RFC5792] Sangster, P. and K. Narayan, "PA-TNC: A Posture Attribute (PA) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5792, March 2010.
- [RFC5793] Sahita, R., Hanna, S., Hurst, R., and K. Narayan, "PB-TNC: A Posture Broker (PB) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5793, March 2010.

Authors' Addresses

Pearl Liang
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90094
USA

Email: pearl.liang@icann.org

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

Email: Alexey.Melnikov@isode.com

David Conrad
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90094
US

Email: david.conrad@icann.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 8, 2017

Q. Sun
C. Xie
China Telecom
J. Bi
Tsinghua University
W. Xu
Huawei Technologies
July 7, 2016

Interface to the Address Pool Management
draft-sun-i2apm-address-pool-management-arch-01

Abstract

This document describes an mechanism for a standard, programmatic interface for address pool management. With the remaining IPv4 address becoming more and more scattered, it is complicated to manually configure the address pools on lots of Broadband Network Gateways(BNGs) for operators. By introducing SDN/NFV in BNG, the address pools can be allocated in a centralized way. It will not only simplify the address management for operators, but also improve the utilization efficiency of the address pool.

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 <http://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 January 8, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Architectural Overview	3
4. Initial Address Pool Configuration	5
5. Address Pool Status Report	7
6. Address Pool Status Query	8
7. Address Exhaustion	8
8. Address Pool Release	8
9. Control Protocol consideration	10
10. Security Considerations	10
11. Acknowledgements	10
12. References	10
12.1. Normative References	10
12.2. Informative References	10
Authors' Addresses	10

1. Introduction

The Broadband Network Gateway(BNG), which manages a routable IP address on behalf of each subscriber, should be configured with the IP address pools allocated to subscribers. However, currently operators are facing with the address shortage problem, the remaining IPv4 address pools are usually quite scattered, no more than /24 per address pool in many cases. Therefore, it is complicated to manually configure the address pools on lots of Broadband Network Gateway(BNG) for operators. For large scale MAN, the number of BNGs can be up to over one hundred. Manual configuration on all the BNGs statically will not only greatly increase the workload, but also decrease the utilization efficiency of the address pools when the number of subscribers changes in the future.

Another use case which needs to configure the address pools is IPv6 migration. For IPv6 transition mechanisms, e.g. DS-Lite, lw4over6, etc., they all need to be configured with address pools as translated routeable addresses. When high availability features, e.g. active-active/active-standby failover mechanism, etc., are enabled for these IPv6 transition mechanisms, different address pools need to be configured on each transition instance. This will further increase

the number of address pools need to be configured. Besides, the occupation of the address pools may vary during different transition periods, (e.g. at the early stage of IPv6 transition, IPv4 traffic will normally occupy a great portion of the total traffic, while in the later stage of IPv6 transition, IPv4 traffic will decrease and the amount of IPv4 address pools will decrease accordingly.

There are other devices which may need to configure address pools as well. For example, the Firewall need to configure the address pool for acl/NAT process. The VPN also needs to configure the address pools for end-users.

When SDN/NFV is introduced in the network, these devices (e.g. BNG, CGN, firewall, VPN, etc.) will run as VNFs in virtualized environment. A common centralized address management server can interact with different VNFs and allocate address pools automatically.

In this document, we propose a mechanism to manage the address pools centrally. In this way, operators do not need to configure the address pools one by one manually and it also helps to use the address pools more efficiently.

2. Terminology

The following terms are used in this document:

APMS A management system which has a centralized database manage the overall address pools and allocate address pools to the device in the devices.

DA A device agent in device, which contact with APM server to manipulate address pool.

3. Architectural Overview

In this architecture, the Address Pool Management (APM) server is a centralized address pool management server for operators to configure the overall address pools. It maintains the address pool database including the overall address pools (OAP) and the address pool status (APS). Operators can configure its remaining address pools in the OAP. They can also reserve some address pool for special-purpose usage. The address pools status is to reflect the current usage of the address pools for different devices. APM also has the interface to configure the address pools to different devices dynamically.

In each device, there is a device agent (DA) to contact with APM server. It initiates the address pools allocation requests, passes

the address pools to local instances, report the status of local address pool usage and update the address pools requests, etc. For some devices, e.g. v6transition, VPN, etc., additional routing modules needs to update the routing table accordingly.

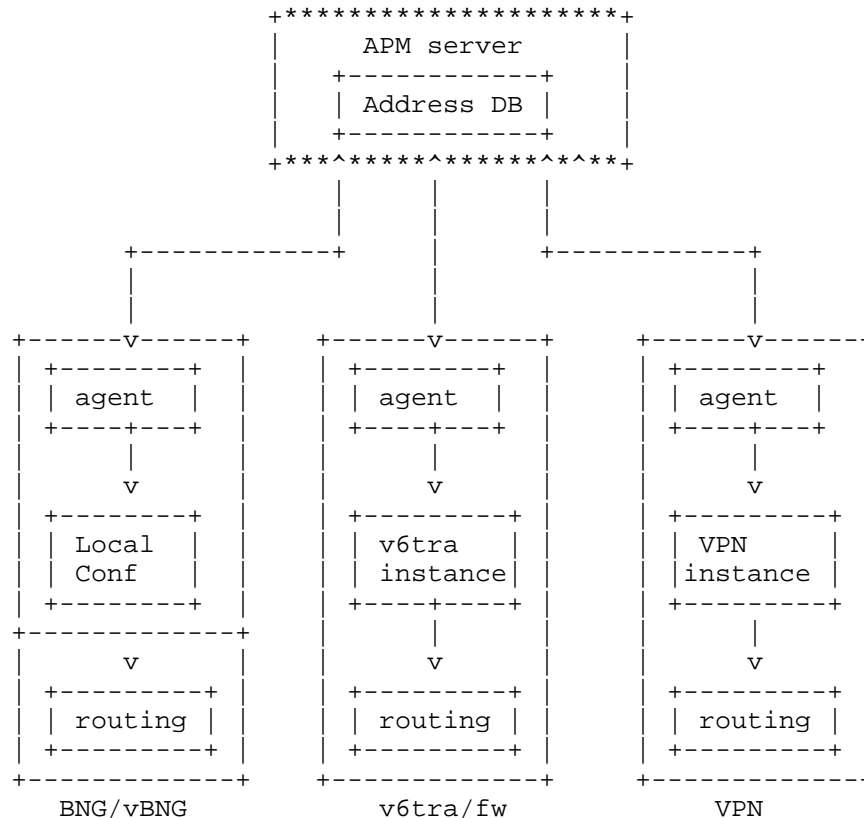


Figure 1: Interface to Address Pool Management (APM)

The overall procedure is as follows:

- o Operators will configure remaining address pools centrally in the Address Pool Management System (APMS). There are multiple address pools which can be configured centrally. The APMS server will then divide the address pools into addressing unit (AU) which will be allocated to the agent in devices by default.
- o The agent will initiate Address Pool request to the APMS. It can carry its desired size of address pool the request, or just use a default value. The address pool size in the request is only used

as a hint. The actual size of the address pool is totally determined by APMS. It will also carry the DA's identification and the type of address pool.

- o APMS looks up the remaining address pool in its local database. It will then allocate a set of address pools to the DA. Each address pool has a related lifetime.
- o DA receives the AddressPool reply and use them for their purpose.
- o If the lifetime of the address pool is going to expire, the DA should issue an AddressPoolRenew request to extend the lifetime, including the IPv4, IPv6, Ports, etc.
- o The AddressPoolReport module keeps monitoring and reports the current usage of all current address pools for each transition mechanism. if it is running out of address pools, it can renew the AddressPoolRequest for a newly allocated one. It can also release and recycle an existing address pool if the that address pool has not been used for a specific and configurable time.
- o When the connection of APMS is lost or the APMS needs the status information of certain applications, the APMS may pre-actively query the DA for the status information.

4. Initial Address Pool Configuration

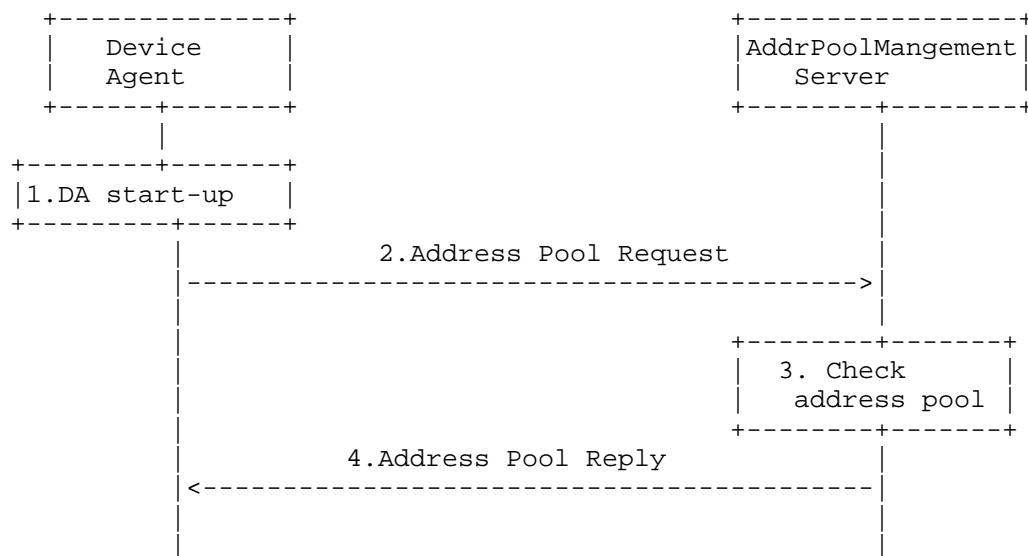


Figure 2: Initial Address Pool Configuration

Figure 2 illustrates the initial address pool configuration procedure:

1. The DA checks whether there is already address pool configured in the local site when it starts up. if no, it means the initial start-up or the address pool has been released. if yes, the address pool could be used directly.
2. The DA will initiate Address Pool request to the APMS. It can carry its desired size of address pool in the request, or just use a default value. The address pool size in the DA's request is only used as a hint. The actual size of the address pool is totally determined by APMS. It will also carry the DA's identification, the type of transition mechanism and the indication of port allocation support.
3. The APMS determines the address pool allocated for the DA based on the parameters received.
4. The APMS sends the Address Pool Reply to the DA. It will also distribute the routing entry of the address pool automatically. In particular, if the newly received address pool can be aggregated to an existing one, the routing should be aggregated accordingly.

5. Address Pool Status Report

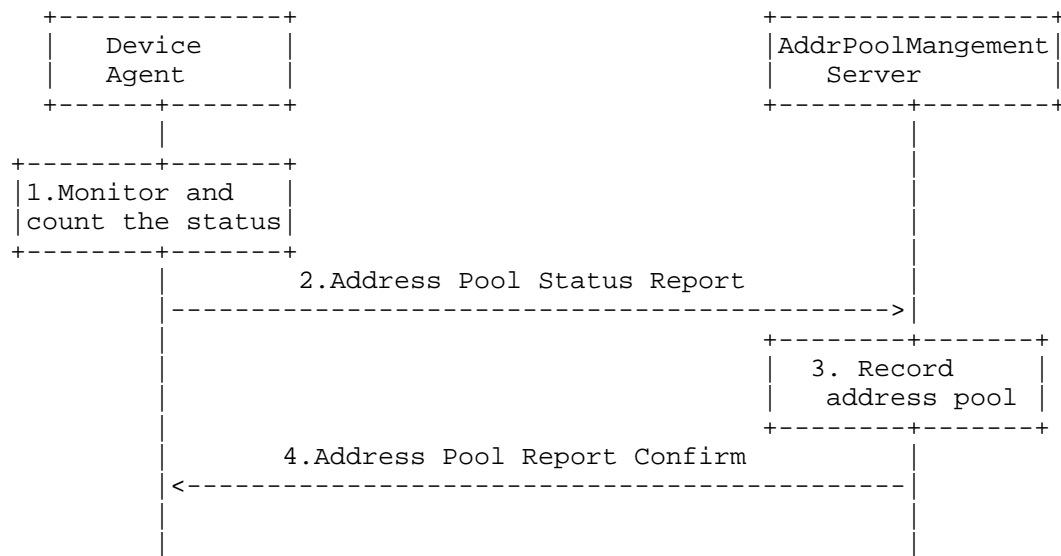


Figure 3: Address Pool Status Report

Figure 3 illustrates the active address pool status report procedure:

1. The DA will monitor and count the usage status of the local address pool. The DA counts the address usage status in one month, one week and one day, which includes the local address, address usage ratio (peak and average values), and the port usage ratio (peak and average values).
2. The DA reports the address pool usage status to the APMS. for example, it will report the address usage status in one day, which contains the IP address, NAT44, address list: 30.14.44.0/28, peak address value 14, average address usage ratio 90%, TCP port usage ratio 20%, UDP port usage ratio 30% and etc.
3. The APMS records the status and compares with the existing address information to determine whether additional address pool is needed.
4. The APMS will confirm the address pool status report request to the DA. It will keep sending the address pool status report request to the APMS if no confirm message is received.

6. Address Pool Status Query

When the status of APMS is lost or the AMS needs the status information of the DAs, the APMS may actively query the TD for the status information, as shown in step 1 of Figure 4. The following steps 2,3,4,5 are the same as the Address Pool Status Report procedure.

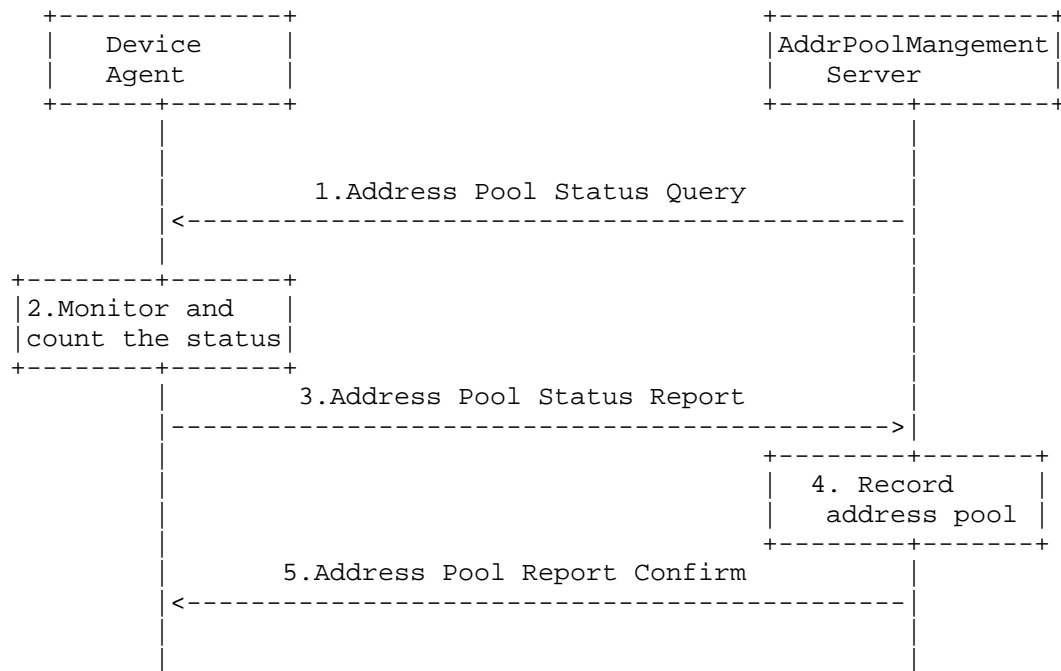


Figure 4: Address Pool Status Query

7. Address Exhaustion

When the DA uses up the addresses allocated, it will renew the address pool request to the APMS for an additional address pool. The procedure is the same as the initial address pool request.

8. Address Pool Release

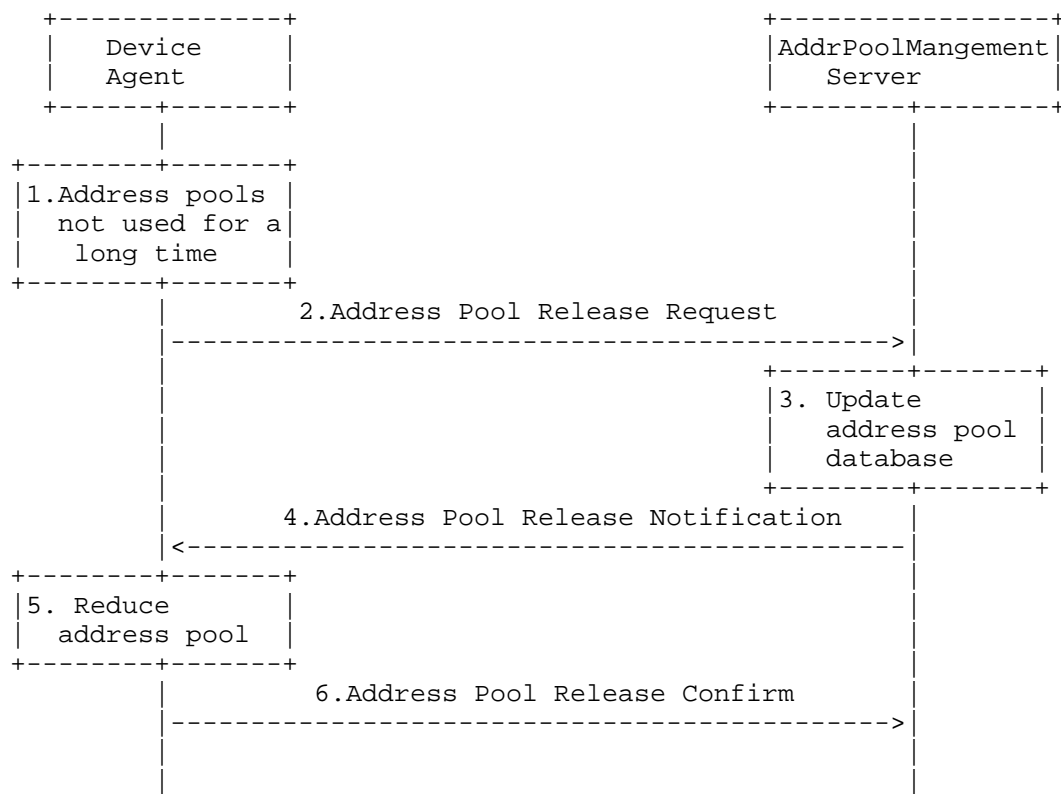


Figure 5: Address Pool Release

Figure 5 illustrates the address pool release procedure:

1. The counting module in the DA checks that there are addresses not used for a long time;
2. The DA sends the address pool release request to the APMS to ask the release of those addresses;
3. The APMS updates the local address pool information to add the new addressed released.
4. The APMS notifies the TD that the addresses have been release successfully;
5. The DA will update the local address pool. if no Address Pool Release Notification is received, the DA will repeat step 2;

6. The DA confirms with the APMS that the address pool has been released successfully.

9. Control Protocol consideration

The I2APM architecture consists of two major distinct entities: APM Server and network equipment with an APM Agent. In order to provide address pool manipulations between these two entities, the I2APM architecture calls for well-defined protocols for interfacing between them. For compatibility with legacy network equipment, the architecture reuse legacy protocol such as radius. While the IETF may also choose to define one or more specific approaches to manipulate address pool, such as NETCONF or RESTCONF with address pool YANG data model.

10. Security Considerations

11. Acknowledgements

N/A.

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, <<http://www.rfc-editor.org/info/rfc2119>>.

- 12.2. Informative References

- [RFC6674] Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway-Initiated Dual-Stack Lite Deployment", RFC 6674, DOI 10.17487/RFC6674, July 2012, <<http://www.rfc-editor.org/info/rfc6674>>.
- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<http://www.rfc-editor.org/info/rfc6888>>.

Authors' Addresses

Qiong Sun
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: sunqiong@ctbri.com.cn

Chongfeng Xie
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: xiechf@ctbri.com.cn

Jun Bi
Tsinghua University
3-212, FIT Building, Tsinghua University, Haidian District
Beijing 100084
P.R. China

Email: junbi@tsinghua.edu.cn

Weiping Xu
Huawei Technologies
Bantian, Longgang District
shenzhen 518129
P.R. China

Email: xuweiping@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 5, 2017

Q. Sun
C. Xie
China Telecom
M. Boucadair
Orange
W. Liu
Huawei Technologies
Y. Lee
Comcast
July 4, 2016

A YANG Data Model for Address Pool Management
draft-sun-i2apm-address-pool-management-yang-02

Abstract

This document specifies a YANG data model for IP address pool management. It can be used to automatically allocate, update and delete address pools in different devices of an underlying network.

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 <http://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 January 5, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. AddressPoolManagement Data Model	3
4. AddressPoolManagement YANG Module	5
5. Security Considerations	12
6. IANA Considerations	13
7. Acknowledgements	13
8. References	13
8.1. Normative References	13
8.2. Informative References	14
Authors' Addresses	14

1. Introduction

IP address pool management is one of the basic elements to configure in a network to offer connectivity services to connected devices. Concretely, pools can be provisioned to DHCP servers, IPv4 service continuity devices (e.g., DS-Lite AFTR, NAT64), Carrier Grade NAT (CGN), Broadband Network Gateway (BNG), etc. Automated means to rationalize the management of address resources and to make sure the underlying routing and forwarding capabilities are appropriately configured, are helpful for operators. This document specifies a YANG data model for that purpose.

A device can be provisioned with a pool of addresses for various reasons: service requesting hosts with addresses or prefixes (e.g., DHCP server, Delegating router), inject appropriate routing entries (e.g., PE, BNG) given that prefix assignments and routing actions must be correlated otherwise delivery of connectivity service will fail. This document does not elaborate the usage of pools provisioned to a network element.

It is worth mentioning that: (1) current practices rely on static configuration which is prone to errors, (2) the level of route aggregation cannot be driven by PE routers without any hint(s) from an entity that has the visibility on aggregation policies and the status of prefixes, etc., and (3) relying on proprietary means to trigger the injection of routing entries may lead to undesired behavior such as to increase the size of routing table and forwarding table due to injecting very specific routes.

Within this document, an address pool usually contains the address pool type, start-address, end-address, its corresponding lifetime and the identification of the usage. Each address pool is represented by an Address Pool Entry (APE).

Pools may be specific to a service offered by a network or be valid for all services.

Pools can be added and/or withdrawn.

2. Terminology

The terminology for describing YANG data models is defined in [RFC6020].

The meaning of the symbols in the tree diagrams is as follows:

Brackets "[" and "]" enclose list keys.

Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.

Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only).

Symbols after data node names: "?" means an optional node, "!" a container with presence, and "*" denotes a "list" or "leaf-list".

Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

Ellipsis ("...") stands for contents of subtrees that are not shown.

3. AddressPoolManagement Data Model

Two YANG modules are defined (Figure 1). The first module, "ietf-address-pool", defines generic address pool aspects which is common to all use cases. The second module, "ietf-address-pool-status", defines the status of the address pool.

```

module: ietf-address-pool
  +--rw address-pools
  |   +--rw address-pool* [address-pool-name]
  |   |   +--rw address-pool-name      string
  |   |   +--rw device-id?             string
  |   |   +--rw address-pool-service* [service-name]
  |   |   |   +--rw service-name      string
  |   |   +--rw address-pool-entries
  |   |   |   +--rw ipv4-address-range* [ipv4-address-range-name]
  |   |   |   |   +--rw ipv4-address-range-name      string
  |   |   |   |   +--rw ip-lower-address?            inet:ipv4-address-no-zone
  |   |   |   |   +--rw ip-upper-address?            inet:ipv4-address-no-zone
  |   |   |   |   +--rw usergateway?                 inet:ipv4-address-no-zone
  |   |   |   |   +--rw gwnetmask?                   yang:dotted-quad
  |   |   |   |   +--rw type?                         address-pool-type
  |   |   |   |   +--rw lifetime?                     yang:date-and-time
  |   |   |   |   +--rw instance?                     instance-type
  |   |   |   +--rw warning-threshold-v4? percent
  |   |   |   +--rw ipv6-prefix* [ipv6-prefix-name]
  |   |   |   |   +--rw ipv6-prefix-name      string
  |   |   |   |   +--rw ipv6-prefix?          inet:ipv6-prefix
  |   |   |   |   +--rw usergateway?          inet:ipv6-address-no-zone
  |   |   |   |   +--rw type?                 address-pool-type
  |   |   |   |   +--rw lifetime?             yang:date-and-time
  |   |   |   |   +--rw instance?             instance-type
  |   |   |   +--rw warning-threshold-v6? percent
  |   +--ro address-pool-status
  |   |   +--ro address-pool* [address-pool-name]
  |   |   |   +--ro address-pool-name      string
  |   |   |   +--ro address-pool-service* [service-name]
  |   |   |   |   +--ro service-name      string
  |   |   |   +--ro status?                enumeration
  |   |   +--ro address-pool-entries
  |   |   |   +--ro ipv4-address-range* [ipv4-address-range-name]
  |   |   |   |   +--ro ipv4-address-range-name      string
  |   |   |   |   +--ro peak-address-usage-ratio?    percent
  |   |   |   |   +--ro average-address-usage-ratio? percent
  |   |   |   +--ro ipv6-prefix* [ipv6-prefix-name]
  |   |   |   |   +--ro ipv6-prefix-name      string
  |   |   |   |   +--ro peak-prefix-usage-ratio?    percent
  |   |   |   |   +--ro average-prefix-usage-ratio? percent
  |   |   +--ro port-range* [port-range-name]
  |   |   |   +--ro port-range-name      string
  |   |   |   +--ro peak-address-usage-ratio?    percent
  |   |   |   +--ro average-address-usage-ratio? percent

```

Figure 1: Interface to Address Pool Management (APM)

4. AddressPoolManagement YANG Module

This module imports typedefs from [RFC6991] and [RFC7223].

```
<CODE BEGINS> file "ietf-address-pool@2015-10-14.yang"
module ietf-address-pool {
  namespace "urn:ietf:params:xml:ns:yang:ietf-address-pool";
  prefix address-pool;
  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  organization
    "xxx Working Group";

  contact
    "Editor:    Qiong Sun
                                     <mailto:sunqiong@ctbri.com.cn>"

    Editor:    Will(Shucheng) Liu
                                     <mailto:liushucheng@huawei.com>";

  description
    "This module contains a collection of YANG definitions for
    configuring IP address pools.

    Copyright (c) 2015 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).
    This version of this YANG module is part of RFC 7277; see
    the RFC itself for full legal notices.";

  revision 2015-10-14 {
    description
      "Initial revision.";
    reference
      "-00";
  }

  typedef percent {
```

```
    type uint8 {
        range "0 .. 100";
    }
    description
        "Percentage";
}

typedef address-pool-type{
    type enumeration{
        enum usergateway {
            description
                "The address pool has a usergateway.";
        }
        enum import-route {
            description
                "The address pool need to import a route
                to external network.";
        }
    }
    description
        "Address pool type.";
}

typedef instance-type{
    type enumeration{
        enum pppoe {
            description
                "The address pool is used for pppoe access.";
        }
        enum dhcp {
            description
                "The address pool is used for dhcp access.";
        }
        enum vpn {
            description
                "The address pool is used for vpn access.";
        }
        enum ds-lite {
            description
                "The address pool is used for ds-lite access.";
        }
        enum lw4over6 {
            description
                "The address pool is used for lw4over6 access.";
        }
    }
    enum map {
        description
            "The address pool is used for map access.";
    }
}
```



```
    }
    enum cgn {
        description
            "The address pool is used for cgn access.";
    }
    enum xlat {
        description
            "The address pool is used for xlat access.";
    }
    enum other {
        description
            "The address pool is used for others.";
    }
}
description
    "Instance type.";
}

container address-pools {
    description
        "This is a top level container for Address Pools.
        It can have one or more Address Pools. The pools may
        not be contiguous.";
    list address-pool {
        key address-pool-name;
        description
            "An Address Pool is an ordered list of
            Address Pool Entries (APE). Each Access Pool Entry has a
            list of address ranges and its associated lifetime.";
        leaf address-pool-name {
            type string;
            description
                "The name of address pool";
        }
        leaf device-id {
            type string;
            description
                "The identifier of device that using address pool";
        }
        list address-pool-service {
            key service-name;
            description
                "The services that can use these pool.";
            leaf service-name {
                type string;
                description
                    "A service name: e.g., any, voip, iptv, internet, etc.";
            }
        }
    }
}
```

```
    }

    container address-pool-entries {
      description
        "The address-pool-entries container contains
        a list of address-ranges and associated attributes.";
      list ipv4-address-range {
        key ipv4-address-range-name;
        description
          "IPv4 Address range.";
        leaf ipv4-address-range-name {
          type string;
          description
            "The name of IPv4 address range.";
        }
        leaf ip-lower-address {
          type inet:ipv4-address-no-zone;
          description
            "The lower IPv4 address of the address range.";
        }
        leaf ip-upper-address {
          type inet:ipv4-address-no-zone;
          description
            "The upper IPv4 address of the address range.";
        }
        leaf usergateway {
          type inet:ipv4-address-no-zone;
          description
            "It only exists when address pool are used for
            user addressing.";
        }
        leaf gwnetmask {
          type yang:dotted-quad;
          description
            "The netmask for usergateway.";
        }
        leaf type {
          type address-pool-type;
          description
            "The type of the address pool.";
        }
        leaf lifetime {
          type yang:date-and-time;
          description
            "The lifetime for the address pool. '0' means
            withdrawal.";
        }
        leaf instance {
```

```
        type instance-type;
        description
            "The instance of the address pool.";
    }
}
leaf warning-threshold-v4{
    type percent;
    description
        "The threshold of the ipv4 address pool.";
}

list ipv6-prefix {
    key ipv6-prefix-name;
    description
        "IPv6 prefix.";
    leaf ipv6-prefix-name {
        type string;
        description
            "The name of IPv6 prefix.";
    }
    leaf ipv6-prefix {
        type inet:ipv6-prefix;
        description
            "The IPv6 prefix.";
    }
    leaf usergateway {
        type inet:ipv6-address-no-zone;
        description
            "It only exists when address pool are used for
            user addressing.";
    }
    leaf type {
        type address-pool-type;
        description
            "The type of the address pool.";
    }
    leaf lifetime {
        type yang:date-and-time;
        description
            "The lifetime for the address pool. '0' means
            withdrawal.";
    }
    leaf instance {
        type instance-type;
        description
            "The instance of the address pool.";
    }
}
```

```
        leaf warning-threshold-v6{
            type percent;
            description
                "The threshold of the ipv6 address pool.";
        }
    }
}

/*
 * Operational state data nodes
 */

container address-pool-status {
    config false;
    description
        "This is a top level container for Address Pool Status,
        which contains the status of address pool usage.";
    list address-pool {
        key address-pool-name;
        description
            "An Address Pool is an ordered list of
            Address Pool Entries (APE). Each Access Pool Entry has a
            list of address ranges and its associated lifetime. ";
        leaf address-pool-name {
            type string;
            description
                "The name of address pool";
        }
        list address-pool-service {
            key service-name;
            description
                "The services that can use these pool.";
            leaf service-name {
                type string;
                description
                    "A service name: e.g., any, voip, iptv, internet, etc.";
            }
        }
    }
    leaf status {
        type enumeration{
            enum active {
                description
                    "The address pool is in active status.";
            }
            enum idle {
                description
                    "The address pool is in idle status.";
            }
        }
    }
}
```

```
    }
  }
  description
    "The status of address pool";
}
container address-pool-entries {
  description
    "The address-pool-entries container contains
    a list of address-ranges and associated attributes.";
  list ipv4-address-range {
    key ipv4-address-range-name;
    description
      "IPv4 Address range.";
    leaf ipv4-address-range-name {
      type string;
      description
        "The name of IPv4 address range.";
    }
    leaf peak-address-usage-ratio {
      type percent;
      description
        "The peak usage rate of the address range.";
    }
    leaf average-address-usage-ratio {
      type percent;
      description
        "The average usage rate of the address range.";
    }
  }
}
list ipv6-prefix {
  key ipv6-prefix-name;
  description
    "IPv6 prefix.";
  leaf ipv6-prefix-name {
    type string;
    description
      "The name of IPv6 prefix.";
  }
  leaf peak-prefix-usage-ratio {
    type percent;
    description
      "The peak usage rate of the prefix.";
  }
  leaf average-prefix-usage-ratio {
    type percent;
    description
      "The average usage rate of the prefix.";
  }
}
```

```

    }
    list port-range {
        key port-range-name;
        description
            "port range.";
        leaf port-range-name {
            type string;
            description
                "The name of port range.";
        }
        leaf peak-address-usage-ratio {
            type percent;
            description
                "The peak usage rate of the port range.";
        }
        leaf average-address-usage-ratio {
            type percent;
            description
                "The average usage rate of the port range.";
        }
    }
}
}
}
}
}
<CODE ENDS>

```

Figure 2: Interface to Address Pool Management (APM)

5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the support of SSH is mandatory to implement secure transport [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and contents.

All data nodes defined in the YANG module which can be created, modified and deleted (i.e., config true, which is the default). These data nodes are considered sensitive. Write operations (e.g., edit-config) applied to these data nodes without proper protection can negatively affect network operations.

6. IANA Considerations

This document requests IANA to register the following URI in the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-address-pool
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

Figure 3: namespace

This document requests IANA to register the following YANG module in the "YANG Module Names" registry [RFC6020].

```
name: ietf-address-pool
namespace: urn:ietf:params:xml:ns:yang:ietf-address-pool
prefix: address-pool
reference: RFC XXXX
```

Figure 4: IANA register

7. Acknowledgements

N/A.

8. References

8.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<http://www.rfc-editor.org/info/rfc6991>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<http://www.rfc-editor.org/info/rfc7223>>.

8.2. Informative References

- [RFC6674] Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway-Initiated Dual-Stack Lite Deployment", RFC 6674, DOI 10.17487/RFC6674, July 2012, <<http://www.rfc-editor.org/info/rfc6674>>.
- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<http://www.rfc-editor.org/info/rfc6888>>.

Authors' Addresses

Qiong Sun
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: sunqiong@ctbri.com.cn

Chongfeng Xie
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: xiechf@ctbri.com.cn

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
China

Email: liushucheng@huawei.com

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: yiul_lee@comcast.com