v6ops
Internet-Draft
Intended status: Informational
Expires: September 4, 2012

I. Gashinsky
Yahoo!
J. Jaeggli
Zynga
W. Kumari
Google Inc

March 03, 2012

Operational Neighbor Discovery Problems
draft-ietf-v6ops-v6nd-problems-05

Abstract

In IPv4, subnets are generally small, made just large enough to cover the actual number of machines on the subnet. In contrast, the default IPv6 subnet size is a /64, a number so large it covers trillions of addresses, the overwhelming number of which will be unassigned. Consequently, simplistic implementations of Neighbor Discovery (ND) can be vulnerable to deliberate or accidental denial of service, whereby they attempt to perform address resolution for large numbers of unassigned addresses. Such denial of attacks can be launched intentionally (by an attacker), or result from legitimate operational tools or accident conditions. As a result of these vulnerabilities, new devices may not be able to "join" a network, it may be impossible to establish new IPv6 flows, and existing IPv6 transported flows may be interrupted.

This document describes the potential for DOS in detail and suggests possible implementation improvements as well as operational mitigation techniques that can in some cases be used to protect against or at least alleviate the impact of such attacks.

Status of this Memo

This Internet-Draft will expire on September 4, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document describes implementation issues with IPv6's Neighbor Discovery protocol that can result in vulnerabilities when a network is scanned, either by an intruder or through the use of scanning tools that perform network inventory, security audits, etc. (e.g. "nmap").

This document describes the problem in detail, suggests possible implementation improvements, as well as operational mitigation techniques, that can in some cases protect against such attacks.

The RFC series documents generally describe the behavior of protocols, that is, "what" is to be done by a protocol, but not exactly "how" it is to be implemented. The exact details of how best to implement a protocol will depend on the overall hardware and software architecture of a particular device. The actual "how" decisions are (correctly) left in the hands of implementers, so long as implementations differences will generally produce proper on-the-wire behavior.

While reading this document, it is important to keep in mind that discussions of how things have been implemented beyond basic compliance with the specification is not within the scope of the neighbor discovery RFCs.

1.1. Applicability

This document is primarily intended for operators of IPV6 networks and implementors of [RFC4861]. The Document provides some operational considerations as well as recommendations to increase the resilience of the Neighbor Discovery protocol.

2. The Problem

In IPv4, subnets are generally small, made just large enough to cover the actual number of machines on the subnet. For example, an IPv4 /20 contains only 4096 address. In contrast, the default IPv6 subnet size is a /64, a number so large it covers literally billions of billions of addresses, the overwhelming majority of which will be unassigned. Consequently, simplistic implementations of Neighbor Discovery may fail to perform as desired when they perform address resolution of large numbers of unassigned addresses. Such failures can be triggered either intentionally by an attacker launching a Denial of Service attack (DoS)[RFC4732] to exploit this vulnerability, or unintentionally due to the use of legitimate operational tools that scan networks for inventory and other

purposes. As a result of these failures, new devices may not be able to "join" a network, it may be impossible to establish new IPv6 flows, and existing IPv6 transport flows may be interrupted.

Network scans attempt to find and probe devices on a network. Typically, scans are performed on a range of target addresses, or all the addresses on a particular subnet. When such probes are directed via a router, and the target addresses are on a directly attached network, the router will attempt to perform address resolution on a large number of destinations (i.e., some fraction of the 2^64 addresses on the subnet). The router's process of testing for the (non)existence of neighbors can induce a denial of service condition, where the number of necessary Neighbor Discovery requests overwhelms the implementation's capacity to process them, exhausts available memoryand replaces existing in-use mappings with incomplete entries that will never be completed. A directed DoS attack may seek to intentionally create similar conditions to that created unintentionally by a network scan. The resulting network disruption may impact existing traffic, and devices that join the network may find that address resolution attempts fail. The DOS as a consequence of network scanning was previously described in [RFC5157]

In order to mitigate risk associated with this DoS threat, some router implementations have taken steps to rate-limit the processing rate of Neighbor Solicitations (NS). While these mitigations do help, they do not fully address the issue and may introduce their own set of issues to the neighbor discovery process.

3. Terminology

Address Resolution Address resolution is the process through which a node determines the link-layer address of a neighbor given only its IP address. In IPv6, address resolution is performed as part of Neighbor Discovery [RFC4861], p60

Forwarding Plane That part of a router responsible for forwarding packets. In higher-end routers, the forwarding plane is typically implemented in specialized hardware optimized for performance. Steps in the forwarding process include determining the correct outgoing interface for a packet, decrementing its Time To Live (TTL), verifying and updating the checksum, placing the correct link-layer header on the packet, and forwarding it.

Control Plane That part of the router implementation that maintains the data structures that determine where packets should be forwarded. The control plane is typically implemented as a "slower" software process running on a general purpose processor

and is responsible for such functions as communicating network status changes via routing protocols, maintaining the forwarding table, performing management, and resolving the correct link-layer address for adjacent neighbors. The control plane "controls" the forwarding plane by programming it with the information needed for packet forwarding.

Neighbor Cache As described in [RFC4861], the data structure that holds the cache of (amongst other things) IP address to link-layer address mappings for connected nodes. As the information in the Neighbor Cache is needed by the forwarding plane every time it forwards a packet, it is usually implemented in an ASIC.

Neighbor Discovery Process The Neighbor Discovery Process (NDP) is that part of the control plane that implements the Neighbor

Discovery protocol. NDP is responsible for performing address resolution and maintaining the Neighbor Cache. When forwarding packets, the forwarding plane accesses entries within the Neighbor Cache. When the forwarding plane processes a packet for which the corresponding Neighbor Cache Entry is missing or incomplete, it notifies NDP to take appropriate action (typically via a shared queue). NDP picks up requests from the shared queue and performs any necessary discovery action. In many implementations the NDP is also responsible for responding to router solicitation messages, Neighbor Unreachability Detection (NUD), etc.

4. Background

Modern router architectures separate the forwarding of packets (forwarding plane) from the decisions needed to decide where the packets should go (control plane). In order to deal with the high number of packets per second, the forwarding plane is generally implemented in hardware and is highly optimized for the task of forwarding packets. In contrast, the NDP control plane is mostly implemented in software processes running on a general purpose processor.

When a router needs to forward an IP packet, the forwarding plane logic performs the longest match lookup to determine where to send the packet and what outgoing interface to use. To deliver the packet to an adjacent node, the forwarding plane encapsulates the packet in a link-layer frame (which contains a header with the link-layer destination address). The forwarding plane logic checks the Neighbor Cache to see if it already has a suitable link-layer destination, and if not, places the request for the required information into a queue, and signals the control plane (i.e., NDP) that it needs the link-layer address resolved.

In order to protect NDP specifically and the control plane generally from being overwhelmed with these requests, appropriate steps must be taken. For example, the size and fill rate of the queue might be limited. NDP running in the control plane of the router dequeues requests and performs the address resolution function (by performing a neighbor solicitation and listening for a neighbor advertisement). This process is usually also responsible for other activities needed to maintain link-layer information, such as Neighbor Unreachability

Detection (NUD).

By sending appropriate packets to addresses on a given subnet, an attacker can cause the router to queue attempts to resolve so many addresses that it crowds out attempts to resolve "legitimate" addresses (and in many cases becomes unable to perform maintenance of existing entries in the neighbor cache, and unable to answer Neighbor Solicitation). This condition can result in the inability to resolve new neighbors and loss of reachability to neighbors with existing ND-Cache entries. During testing it was concluded that 4 simultaneous nmap sessions from a low-end computer was sufficient to make a router's neighbor discovery process unusable and therefore forwarding became unavailable to the destination subnets.

The failure to maintain proper NDP behavior whilst under attack has been observed across multiple platforms and implementations, including the largest modern router platforms available (at the inception of work on this document).

5. Neighbor Discovery Overview

When a packet arrives at (or is generated by) a router for a destination on an attached link, the router needs to determine the correct link-layer address to use in the destination field of the layer 2 encapsulation. The router checks the Neighbor Cache for an existing Neighbor Cache Entry for the neighbor, and if none exists, invokes the address resolution portions of the IPv6 Neighbor Discovery [RFC4861] protocol to determine the link-layer address of the neighbor.

[RFC4861] Section 5.2 (Conceptual Sending Algorithm) outlines how this process works. A very high level summary is that the device creates a new Neighbor Cache Entry for the neighbor, sets the state to INCOMPLETE, queues the packet and initiates the actual address resolution process. The device then sends out one or more Neighbor Solicitations, and when it receives a corresponding Neighbor Advertisement, completes the Neighbor Cache Entry and sends the queued packet.

6. Operational Mitigation Options

This section provides some feasible mitigation options that can be employed today by network operators in order to protect network availability while vendors implement more effective protection measures. It can be stated that some of these options are "kludges", and can be operationally difficult to manage. They are presented, as they represent options we currently have. It is each operator's responsibility to evaluate and understand the impact of changes to their network due to these measures.

6.1. Filtering of unused address space.

The DoS condition is induced by making a router try to resolve addresses on the subnet at a high rate. By carefully addressing machines into a small portion of a subnet (such as the lowest numbered addresses), it is possible to filter access to addresses not in that assigned portion of address space using Access Control Lists (ACLs), or by null routing, features which are available on most existing platforms. This will prevent the attacker from making the router attempt to resolve unused addresses. For example if there are only 50 hosts connected to an interface, you may be able to filter any address above the first 64 addresses of that subnet by null-routing the subnet carrying a more specific /122 route or by applying ACLs on the WAN link to prevent the attack traffic reaching the vulnerable device.

As mentioned at the beginning of this section, it is fully understood that this is ugly (and difficult to manage); but failing other options, it may be a useful technique especially when responding to an attack.

This solution requires that the hosts be statically or statefully addressed (as is often done in a datacenter) and may not interact well with networks using [RFC4862]

6.2. Minimal Subnet Sizing.

By sizing subnets to reflect the number of addresses actually in use, the problem can be avoided. For example, [RFC6164] recommends sizing the subnets for inter-router links to only have 2 addresses (a /127). It is worth noting that this practice is common in IPv4 networks, in part to protect against the harmful effects of ARP request flooding.

Subnet prefixes longer than a /64 are not able to use stateless auto-configuration [RFC4862] so this approach is not suitable for use with hosts that are not statically configured.

6.3. Routing Mitigation.

One very effective technique is to route the subnet to a discard interface (most modern router platforms can discard traffic in hardware / the forwarding plane) and then have individual hosts announce routes for their IP addresses into the network (or use some method to inject much more specific addresses into the local routing domain). For example the network 2001:db8:1:2:3::/64 could be routed to a discard interface on "border" routers, and then individual hosts could announce 2001:db8:1:2:3::10/128, 2001:db8:1:2:3::66/128 into the IGP. This is typically done by having the IP address bound to a virtual interface on the host (for example the loopback interface), enabling IP forwarding on the host and having it run a routing daemon. For obvious reasons, host participation in the IGP makes many operators uncomfortable, but can be a very powerful technique if used in a disciplined and controlled manner. One method to help address these concerns is to have the hosts participate in a different IGP (or difference instance of the same IGP) and carefully redistribute into the main IGP.

6.4. Tuning of the NDP Queue Rate Limit.

Many implementations provide a means to control the rate of resolution of unknown addresses. By tuning this rate, it may be possible to ameliorate the issue, as with most tuning knobs (especially those that deal with rate limiting), the attack may be completed more quickly due to the lower threshold. By excessively lowering this rate you may negatively impact how long the device takes to learn new addresses under normal conditions (for example, after clearing the neighbor cache or when the router first boots). Under attack conditions you may be unable to resolve "legitimate" addresses sooner than if you had just left the parameter untouched.

It is worth noting that this technique is worth investigating only if the device has separate queues for resolution of unknown addresses and the maintenance of existing entries.

7. Recommendations for Implementors.

The section provides some recommendations to implementors of IPv6 Neighbor Discovery.

At a high-level, implementors should program defensively. That is, they should assume that attackers will attempt to exploit implementation weaknesses, and should ensure that implementations are robust to various attacks. In the case of Neighbor Discovery, the following general considerations apply:

Manage Resources Explicitly Resources such as processor cycles, memory, etc. are never infinite, yet with IPv6's large subnets it is easy to cause NDP to generate large numbers of address resolution requests for non-existent destinations. Implementations need to limit resources devoted to processing Neighbor Discovery requests in a thoughtful manner.

Prioritize Some NDP requests are more important than others. For example, when resources are limited, responding to Neighbor Solicitations for one's own address is more important than initiating address resolution requests that create new entries. Likewise, performing Neighbor Unreachability Detection, which by definition is only invoked on destinations that are actively being used, is more important than creating new entries for possibly non-existent neighbors.

7.1. Prioritize NDP Activities

Not all Neighbor Discovery activities are equally important. Specifically, requests to perform large numbers of address resolutions on non-existent Neighbor Cache Entries should not come at the expense of servicing requests related to keeping existing, in-use entries properly up-to-date. Thus, implementations should divide work activities into categories having different priorities. The following gives examples of different activities and their importance in rough priority order. If implmented, the operation and priority of these should be configurable by the operator.

1. It is critical to respond to Neighbor Solicitations for one's own address, especially for a router. Whether for address resolution or Neighbor Unreachability Detection, failure to respond to Neighbor Solicitations results in immediate problems. Failure to respond to NS requests that are part of NUD can cause neighbors to delete the NCE for that address, and will result in followup NS messages using multicast. Once an entry has been flushed, existing traffic for destinations using that entry can no longer be forwarded until address resolution completes successfully. In other words, not responding to NS messages further increases the NDP load, and causes on-going communication to fail.

2. It is critical to revalidate one's own existing NCEs in need of refresh. As part of NUD, ND is required to frequently revalidate existing, in-use entries. Failure to do so can result in the entry being discarded. For in-use entries, discarding the entry will almost certainly result in a subsequent request to perform address resolution on the entry, but this time using multicast. As above, once the entry has been flushed, existing traffic for destinations using that entry can no longer be forwarded until address resolution

completes successfully.

3. To maintain the stability of the control plane, Neighbor Discovery activity related to traffic sourced by the router (as opposed to traffic being forwarded by the router) should be given high priority. Whenever network problems occur, debugging and making other operational changes requires being able to query and access the router. In addition, routing protocols dependent on Neighbor Discovery for connectivity may begin to react (negatively) to perceived connectivity problems, causing additional undesirable ripple effects.

4. Traffic to unknown addresses should be given lowest priority. Indeed, it may be useful to distinguish between "never seen" addresses and those that have been seen before, but that do not have a corresponding NCE. Specifically, the conceptual processing algorithm in IPv6 Neighbor Discovery [RFC4861] calls for deleting NCEs under certain conditions. Rather than delete them completely, however, it might be useful to at least keep track of the fact that an entry at one time existed, in order to prioritize address resolution requests for such neighbors compared with neighbors that have never been seen before.

7.2. Queue Tuning.

On implementations in which requests to NDP are submitted via a single queue, router vendors should provide operators with means to control both the rate of link-layer address resolution requests placed into the queue and the size of the queue. This will allow operators to tune Neighbour Discovery for their specific environment. The ability to set, or have per interface or per prefix queue limits at a rate below that of the global queue limit might limit the damage to the neighbor discovery processing to the network targeted by the attack.

Setting those values must be a very careful balancing act - the lower the rate of entry into the queue, the less load there will be on the ND process, however, it will take the router longer to learn legitimate destinations as a result. In a datacenter with 6,000 hosts attached to a single router, setting that value to be under 1000 would mean that resolving all of the addresses from an initial state (or something that invalidates the address cache, such as a STP TCN) may take over 6 seconds. Similarly, the lower the size of the queue, the higher the likelihood of an attack being able to knock out legitimate traffic (but less memory utilization on the router).

8. IANA Considerations

No IANA resources or consideration are requested in this draft.

9. Security Considerations

This document outlines mitigation options that operators can use to protect themselves from Denial of Service attacks. Implementation advice to router vendors aimed at ameliorating known problems carries the risk of previously unforeseen consequences. It is not believed that these mitigation techniques or the implementation of finer-grained queuing of NDP activity create additional security risks or DOS exposure.

10. Acknowledgements

The authors would like to thank Ron Bonica, Troy Bonin, John Jason Brzozowski, Randy Bush, Vint Cerf, Tassos Chatzithomaoglou, Jason Fesler, Wes George, Erik Kline, Jared Mauch, Chris Morrow and Suran De Silva. Special thanks to Thomas Narten and Ray Hunter for detailed review and (even more so) for providing text!

Apologies for anyone we may have missed; it was not intentional.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, April 2011.

11.2. Informative References

[RFC4732] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[RFC5157] Chown, T., "IPv6 Implications for Network Scanning", RFC 5157, March 2008.

Authors' Addresses

Igor Gashinsky
Yahoo!
45 W 18th St
New York, NY
USA

Email: igor@yahoo-inc.com

Joel Jaeggli
Zynga
111 Evelyn
Sunnyvale, CA
USA

Email: jjaeggli@zynga.com

Warren Kumari
Google Inc
1600 Amphitheatre Parkway
Mountain View, CA
USA

Email: warren@kumari.net