                 IPv6 Implications for Network Scanning

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Abstract

   The much larger default 64-bit subnet address space of IPv6 should in
   principle make traditional network (port) scanning techniques used by
   certain network worms or scanning tools less effective. While
   traditional network scanning probes (whether by individuals or
   automated via network worms) may become less common, administrators
   should be aware that attackers may use other techniques to discover
   IPv6 addresses on a target network, and thus they should also be
   aware of measures that are available to mitigate them. This
   informational document discusses approaches that administrators could
   take when planning their site address allocation and management
   strategies as part of a defence-in-depth approach to network
   security.

Table of Contents

1. Introduction

   One of the key differences between IPv4 and IPv6 is the much larger
   address space for IPv6, which also goes hand-in-hand with much larger
   subnet sizes. This change has a significant impact on the
   feasibility of TCP and UDP network scanning, whereby an automated
   process is run to detect open ports (services) on systems that may
   then be subject to a subsequent attack. Today many IPv4 sites are
   subjected to such probing on a recurring basis. Such probing is
   common in part due to the relatively dense population of active hosts
   in any given chunk of IPv4 address space.

   The 128 bits of IPv6 [1] address space is considerably bigger than
   the 32 bits of address space in IPv4. In particular, the IPv6
   subnets to which hosts attach will by default have 64 bits of host
   address space [2]. As a result, traditional methods of remote TCP or
   UDP network scanning to discover open or running services on a host
   will potentially become less feasible, due to the larger search space
   in the subnet. Similarly, worms that rely on off-link network
   scanning to propagate may also potentially be more limited in impact.
   This document discusses this property of IPv6 and describes related
   issues for IPv6 site network administrators to consider, which may be
   useful when planning site address allocation and management
   strategies.

   For example, many worms, like Slammer, rely on such address scanning
   methods to propagate, whether they pick subnets numerically (and thus
   probably topologically) close to the current victim, or subnets in
   random remote networks. The nature of these worms may change, if
   detection of target hosts between sites or subnets is harder to
   achieve by traditional methods. However, there are other worms that
   propagate via methods such as email, for which the methods discussed
   in this text are not relevant.

   It must be remembered that the defence of a network must not rely
   solely on the unpredictable sparseness of the host addresses on that
   network. Such a feature or property is only one measure in a set of
   measures that may be applied. This document discusses various
   measures that can be used by a site to mitigate attacks as part of an
   overall strategy. Some of these have a lower cost to deploy than
   others. For example, if numbering hosts on a subnet, it may be as
   cheap to number hosts without any predictable pattern as it is to
   number them sequentially. In contrast, use of IPv6 privacy
   extensions [3] may complicate network management (identifying which
   hosts use which addresses).

This document complements the transition-centric discussion of the
issues that can be found in Appendix A of "IPv6 Transition/
Co-existence Security Considerations" [12], which takes a broad view
of security issues for transitioning networks. The reader is also
referred to a recent paper by Bellovin on network worm propagation
strategies in IPv6 networks [13]. This paper discusses some of the
issues included in this document, from a slightly different
perspective.

2. Target Address Space for Network Scanning

   There are significantly different considerations for the feasibility
   of plain, brute-force IPv4 and IPv6 address scanning.

2.1. IPv4

   A typical IPv4 subnet may have 8 bits reserved for host addressing.
   In such a case, a remote attacker need only probe at most 256
   addresses to determine if a particular service is running publicly on
   a host in that subnet. Even at only one probe per second, such a
   scan would take under 5 minutes to complete.

2.2. IPv6

   A typical IPv6 subnet will have 64 bits reserved for host addressing.
   In such a case, a remote attacker in principle needs to probe 2^{64}
   addresses to determine if a particular open service is running on a
   host in that subnet. At a very conservative one probe per second,
   such a scan may take some 5 billion years to complete. A more rapid
   probe will still be limited to (effectively) infinite time for the
   whole address space. However, there are ways for the attacker to
   reduce the address search space to scan against within the target
   subnet, as we discuss below.

2.3. Reducing the IPv6 Search Space

   The IPv6 host address space through which an attacker may search can
   be reduced in at least two ways.

   First, the attacker may rely on the administrator conveniently
   numbering their hosts from [prefix]::1 upward. This makes scanning
   trivial, and thus should be avoided unless the host's address is
   readily obtainable from other sources (for example, it is the site's
   published primary DNS or email Mail Exchange (MX) server).
   Alternatively, if hosts are numbered sequentially, or using any
   regular scheme, knowledge of one address may expose other available
   addresses to scan.

Second, in the case of statelessly autoconfiguring [1] hosts, the
host part of the address will usually take a well-known format that
includes the Ethernet vendor prefix and the "fffe" stuffing. For
such hosts, the search space can be reduced to 48 bits. Further, if
the Ethernet vendor is also known, the search space may be reduced to
24 bits, with a one probe per second scan then taking a less daunting
194 days. Even where the exact vendor is not known, using a set of
common vendor prefixes can reduce the search. In addition, many
nodes in a site network may be procured in batches, and thus have
sequential or near sequential Media Access Control (MAC) addresses;
if one node's autoconfigured address is known, scanning around that
address may yield results for the attacker. Again, any form of
sequential host addressing should be avoided if possible.

2.4. Dual-Stack Networks

Full advantage of the increased IPv6 address space in terms of
resilience to network scanning may not be gained until IPv6-only
networks and devices become more commonplace, given that most IPv6
hosts are currently dual stack, with (more readily scannable) IPv4
connectivity. However, many applications or services (e.g., new
peer-to-peer applications) on the (dual-stack) hosts may emerge that
are only accessible over IPv6, and that thus can only be discovered
by IPv6 address scanning.

2.5. Defensive Scanning

The problem faced by the attacker for an IPv6 network is also faced
by a site administrator looking for vulnerabilities in their own
network's systems. The administrator should have the advantage of
being on-link for scanning purposes though.

3. Alternatives for Attackers: Off-Link

If IPv6 hosts in subnets are allocated addresses 'randomly', and as a
result IPv6 network scanning becomes relatively infeasible, attackers
will need to find new methods to identify IPv6 addresses for
subsequent scanning. In this section, we discuss some possible paths
attackers may take. In these cases, the attacker will attempt to
identify specific IPv6 addresses for subsequent targeted probes.

3.1. Gleaning IPv6 Prefix Information

Note that in IPv6, an attacker would not be able to search across the
entire IPv6 address space as they might in IPv4. An attacker may
learn general prefixes to focus their efforts on by observing route
view information (e.g., from public looking-glass services) or
information on allocated address space from Regional Internet

Registries (RIRs). In general, this would only yield information at
most at the /48 prefix granularity, though some specific /64 prefixes
may be observed from route views on some parts of some networks.

3.2. DNS Advertised Hosts

Any servers that are DNS listed, e.g., MX mail relays, or web
servers, will remain open to probing from the very fact that their
IPv6 addresses will be published in the DNS.

While servers are relatively easy to find because they are DNS-
published, any systems that are not DNS-published will be much harder
to locate via traditional scanning than is the case for IPv4
networks. It is worth noting that where a site uses sequential host
numbering, publishing just one address may lead to a threat upon the
other hosts.

3.3. DNS Zone Transfers

In the IPv6 world, a DNS zone transfer is much more likely to narrow
the number of hosts an attacker needs to target. This implies that
restricting zone transfers is (more) important for IPv6, even if it
is already good practice to restrict them in the IPv4 world.

There are some projects that provide Internet mapping data from
access to such transfers. Administrators may of course agree to
provide such transfers where they choose to do so.

3.4. Log File Analysis

IPv6 addresses may be harvested from recorded logs, such as web site
logs. Anywhere else where IPv6 addresses are explicitly recorded may
prove a useful channel for an attacker, e.g., by inspection of the
(many) Received from: or other header lines in archived email or
Usenet news messages.

3.5. Application Participation

More recent peer-to-peer applications often include some centralised
server that coordinates the transfer of data between peers. The
BitTorrent application builds swarms of nodes that exchange chunks of
files, with a tracker passing information about peers with available
chunks of data between the peers. Such applications may offer an
attacker a source of peer IP addresses to probe.

3.6. Multicast Group Addresses

   Where an Embedded Rendezvous Point (RP) [7] multicast group address
   is known, the unicast address of the RP is implied by the group
   address. Where unicast-prefix-based multicast group addresses [5]
   are used, specific /64 link prefixes may also be disclosed in traffic
   that goes off-site. An administrator may thus choose to put aside
   /64 bit prefixes for multicast group addresses that are not in use
   for normal unicast routing and addressing. Alternatively, a site may
   simply use their non-specific /48 site prefix allocation to generate
   RFC3306 multicast group addresses.

3.7. Transition Methods

   Specific knowledge of the target network may be gleaned if that
   attacker knows it is using 6to4 [4], ISATAP [10], Teredo [11], or
   other techniques that derive low-order bits from IPv4 addresses
   (though in this case, unless they are using IPv4 NAT, the IPv4
   addresses may be probed anyway).

   For example, the current Microsoft 6to4 implementation uses the
   address 2002:V4ADDR::V4ADDR while older Linux and FreeBSD
   implementations default to 2002:V4ADDR::1. This leads to specific
   knowledge of specific hosts in the network. Given one host in the
   network is observed as using a given transition technique, it is
   likely that there are more.

   In the case of Teredo, the 64-bit node identifier is generated from
   the IPv4 address observed at a Teredo server along with a UDP port
   number. The Teredo specification also allows for discovery of other
   Teredo clients on the same IPv4 subnet via a well-known IPv4
   multicast address (see Section 2.17 of RFC 4380 [11]).

4. Alternatives for Attackers: On-Link

   The main thrust of this text is considerations for off-link attackers
   or probing of a network. In general, once one host on a link is
   compromised, others on the link can be very readily discovered.

4.1. General On-Link Methods

   If the attacker already has access to a system on the current subnet,
   then traffic on that subnet, be it Neighbour Discovery or
   application-based traffic, can invariably be observed, and active
   node addresses within the local subnet learnt.

In addition to making observations of traffic on the link, IPv6-
enabled hosts on local subnets may be discovered through probing the
"all hosts" link-local multicast address. Likewise, any routers on
the subnet may be found via the "all routers" link-local multicast
address. An attacker may choose to probe in a slightly more
obfuscated way by probing the solicited node multicast address of a
potential target host.

Where a host has already been compromised, its Neighbour Discovery
cache is also likely to include information about active nodes on the
current subnet, just as an ARP cache would do for IPv4.

Also, depending on the node, traffic to or from other nodes (in
particular, server systems) is likely to show up if an attacker can
gain a presence on a node in any one subnet in a site's network.

4.2. Intra-Site Multicast or Other Service Discovery

A site may also have site- or organisational-scope multicast
configured, in which case application traffic, or service discovery,
may be exposed site wide. An attacker may also choose to use any
other service discovery methods supported by the site.

5. Tools to Mitigate Scanning Attacks

There are some tools that site administrators can apply to make the
task for IPv6 network scanning attackers harder. These methods arise
from the considerations in the previous section.

The author notes that at his current (university) site, there is no
evidence of general network scanning running across subnets.
However, there is network scanning over IPv6 connections to systems
whose IPv6 addresses are advertised (DNS servers, MX relays, web
servers, etc.), which are presumably looking for other open ports on
these hosts to probe further. At the time of writing, DHCPv6 [6] is
not yet in use at the author's site, and clients use stateless
autoconfiguration. Therefore, the author's site does not yet have
sequentially numbered client hosts deployed as may typically be seen
in today's IPv4 DHCP-served networks.

5.1. IPv6 Privacy Addresses

   Hosts in a network using IPv6 privacy extensions [3] will typically
   only connect to external systems using their current (temporary)
   privacy address. The precise behaviour of a host with a stable
   global address and one or more dynamic privacy address(es) when
   selecting a source address to use may be operating-system-specific,
   or configurable, but typical behaviour when initiating a connection
   is use of a privacy address when available.

   While an attacker may be able to port scan a privacy address, if they
   do so quickly upon observing or otherwise learning of the address,
   the threat or risk is reduced due to the time-constrained value of
   the address. One implementation of RFC 4941 already deployed has
   privacy addresses active (used by the node) for one day, with such
   addresses reachable for seven days.

   Note that an RFC 4941 host will usually also have a separate static
   global IPv6 address by which it can also be reached, and that may be
   DNS-advertised if an externally reachable service is running on it.
   DHCPv6 can be used to serve normal global addresses and IPv6 privacy
   addresses.

   The implication is that while privacy addresses can mitigate the
   long-term value of harvested addresses, an attacker creating an IPv6
   application server to which clients connect will still be able to
   probe the clients by their privacy address when they visit that
   server. The duration for which privacy addresses are valid will
   impact the usefulness of such observed addresses to an external
   attacker. For example, a worm that may spread using such observed
   addresses may be less effective if it relies on harvested privacy
   addresses. The frequency with which such address get recycled could
   be increased, though this may increase the complexity of local
   network management for the administrator, since doing so will cause
   more addresses to be used over time in the site.

   A further option here may be to consider using different addresses
   for specific applications, or even each new application instance,
   which may reduce exposure to other services running on the same host
   when such an address is observed externally.

5.2. Cryptographically Generated Addresses (CGAs)

   The use of Cryptographically Generated Addresses (CGAs) [9] may also
   cause the search space to be increased from that presented by default
   use of stateless autoconfiguration. Such addresses would be seen
   where Secure Neighbour Discovery (SEND) [8] is in use.

5.3. Non-Use of MAC Addresses in EUI-64 Format

   The EUI-64 identifier format does not require the use of MAC
   addresses for identifier construction. At least one well known
   operating system currently defaults to generation of the 64-bit
   interface identifier by use of random bits, and thus does not embed
   the MAC address. Where such a method exists, an administrator may
   wish to consider using that option.

5.4. DHCP Service Configuration Options

   One option open to an administrator is to configure DHCPv6, if
   possible, so that the first addresses allocated from the pool begins
   much higher in the address space than at [prefix]::1. Further, it is
   desirable that allocated addresses are not sequential and do not have
   any predictable pattern to them. Unpredictable sparseness in the
   allocated addresses is a desirable property. DHCPv6 implementers
   could reduce the cost for administrators to deploy such 'random'
   addressing by supporting configuration options to allow such
   behaviour.

   DHCPv6 also includes an option to use privacy extension [3]
   addresses, i.e., temporary addresses, as described in Section 12 of
   the DHCPv6 [6] specification.

6. Conclusions

   Due to the much larger size of IPv6 subnets in comparison to IPv4, it
   will become less feasible for traditional network scanning methods to
   detect open services for subsequent attacks, assuming the attackers
   are off-site and services are not listed in the DNS. If
   administrators number their IPv6 subnets in 'random', non-predictable
   ways, attackers, whether they be in the form of automated network
   scanners or dynamic worm propagation, will need to make wider use of
   new methods to determine IPv6 host addresses to target (e.g., looking
   to obtain logs of activity from a site and scanning addresses around
   the ones observed). Such numbering schemes may be very low cost to
   deploy in comparison to conventional sequential numbering, and thus,
   a useful part of an overall defence-in-depth strategy. Of course, if
   those systems are dual-stack, and have open IPv4 services running,
   they will remain exposed to traditional probes over IPv4 transport.

7. Security Considerations

   There are no specific security considerations in this document
   outside of the topic of discussion itself. However, it must be noted
   that the 'security through obscurity' discussions and commentary
   within this text must be noted in their proper context. Relying

purely on obscurity of a node address is not prudent, rather the
advice here should be considered as part of a 'defence-in-depth'
approach to security for a site or network. This also implies that
these measures require coordination between network administrators
and those who maintain DNS services, though this is common in most
scenarios.

8. Acknowledgements

   Thanks are due to people in the 6NET project (www.6net.org) for
   discussion of this topic, including Pekka Savola, Christian Strauf,
   and Martin Dunmore, as well as other contributors from the IETF v6ops
   and other mailing lists, including Tony Finch, David Malone, Bernie
   Volz, Fred Baker, Andrew Sullivan, Tony Hain, Dave Thaler, and Alex
   Petrescu. Thanks are also due for editorial feedback from Brian
   Carpenter, Lars Eggert, and Jonne Soininen amongst others.

9. Informative References

   [1]   Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6)
          Specification", RFC 2460, December 1998.

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

   [3]   Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions
          for Stateless Address Autoconfiguration in IPv6", RFC 4941,
          September 2007.

   [4]   Carpenter, B. and K. Moore, "Connection of IPv6 Domains via
          IPv4 Clouds", RFC 3056, February 2001.

   [5]   Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6
          Multicast Addresses", RFC 3306, August 2002.

   [6]   Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M.
          Carney, "Dynamic Host Configuration Protocol for IPv6
          (DHCPv6)", RFC 3315, July 2003.

   [7]   Savola, P. and B. Haberman, "Embedding the Rendezvous Point
          (RP) Address in an IPv6 Multicast Address", RFC 3956,
          November 2004.

   [8]   Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure
          Neighbor Discovery (SEND)", RFC 3971, March 2005.

   [9]   Aura, T., "Cryptographically Generated Addresses (CGA)",
          RFC 3972, March 2005.

   [10] Templin, F., Gleeson, T., Talwar, M., and D. Thaler, "Intra-
         Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 4214,
         October 2005.

   [11] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network
         Address Translations (NATs)", RFC 4380, February 2006.

   [12] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/
         Co-existence Security Considerations", RFC 4942,
         September 2007.

   [13] Bellovin, S., et al, "Worm Propagation Strategies in an IPv6
         Internet", as published in ;login:, February 2006,
         <http://www.cs.columbia.edu/~smb/papers/v6worms.pdf>.

Author's Address

   Tim Chown
   University of Southampton
   Southampton, Hampshire SO17 1BJ
   United Kingdom

   EMail: tjc@ecs.soton.ac.uk