ipv6mon manual

ipv6mon is a tool meant for monitoring IPv6 address usage on a local network. It is meant to be particularly useful in networks that employ IPv6 Stateless Address Auto-Configuration (as opposed to DHCPv6), where address assignment is decentralized and there is no central server that records which IPv6 addresses have been assigned to which nodes during which period of time.

ipv6mon employs active probing to discover IPv6 addresses in use, and determine whether such addresses remain active.

Address discovery techniques

ipv6mon employs a number of different techniques to discover active IPv6 addresses:

- Active multicast probes
- Passive monitoring of Duplicate Address Detection (DAD) traffic
- Address generation heuristics

Multicast probes

ipv6mon regularly sends multicast probe packets to the all-nodes link-local multicast address (ff02::1), such that it can discover new attached nodes. Two different types of probe packets are used:

- Regular ICMPv6 Echo Request packets
- ICMPv6 packets including a Destination Options header with an unsupported option of type 10xxxxxx

Regular ICMPv6 Echo Request packets elicit ICMPv6 Echo Response packets from all local nodes that normally respond to such packets. On the other hand, ICMPv6 packets including an unsupported option of type 10xxxxxx elicit ICMPv6 Parameter Problem error messages, and can be effective to discover nodes that do not usually respond to multicast ICMPv6 echo requests.

Both multicast probe packets are sent with different Source Addresses, such that they elicit responses from different addresses (as a result of the default IPv6 Source Address selection policy). Hence. all (or most) addresses of each node can be discovered.

Passive monitoring of Duplicate Address Detection (DAD) traffic

ipv6mon also monitors IPv6 Duplicate Address Detection (DAD) traffic, such that all newly configured addresses are discovered as soon as possible.

Address generation heuristics

Whenever an active address is detected, heuristics are performed to discover other addresses in use by the same node. Namely, the Interface-ID of the newly discovered address is used with all other local prefixes to form "candidate" addresses (addresses that might be in use in the local network). Thus, it is possible for ipv6mon to discover active addresses even if they are not used to send traffic on the local network (e.g., discover traditional SLAAC addresses used by hosts that employ “Privacy Addresses”).

Command-line options

Most of the operation parameters of the ipv6mon tool can be configured by the system administrator through the corresponding configuration file. Some of them can be configured through command-line options.

--config-file, -c CONFIG_FILE

This option specifies the pathname of the ipv6mon configuration file. If left unspecified, it defaults to "/etc/ipv6mon.conf"

--show-config, -q

This option causes ipv6mon to read the configuration file and output the different parameters that would result from that configuration. When this option is set, ipv6mon will exit immediately after the configuration parameters have been printed on standard output.

--verbose, -v

This option causes ipv6mon to be verbose when reporting errors.

--help, -h

This option causes ipv6mon to print usage information on standard output.

Credits

The ipv6mon tool version 0.1 and related manuals were produced by Fernando Gont <fgont@si6networks.com> on behalf of the United Kingdom's Centre for the Protection of National Infrastructure (CPNI) <http://www.cpni.gov.uk>.

License

Configuration file

Most of the operation parameters of the ipv6mon tool can be configured by the system administrator through the corresponding configuration file. Except for the setting of the “NetworkInterface” variable, we advise administrator to keep the sample configuration file “as is”.

The following variables can be set in the ipv6mon configuration file:

NetworkInterface

This variable specifies the network interface that will be used for IPv6 address monitoring.

AddressLogFile

This variable specifies the pathname of the log file where address usage will be recorded. If left unspecified, the pathname defaults to "/var/log/ipv6mon.log".

LockFile

This variable is used to specify the pathname of the lockfile (PID file) to be used to easily identify running instances of ipv6mon. If left unspecified, it defaults to "/var/run/ipv6mon.pid".

UnprivilegedUser

This variable specifies the unprivileged username that ipv6mon will switch to (i.e., setuid), such that superuser privileges are released. If left unspecified, it defaults to "ipv6mon".

UnprivilegedUser

This variable specifies the unprivileged username that ipv6mon will switch to (i.e., setuid), such that superuser privileges are released. If left unspecified, it defaults to "ipv6mon".

TimestampFormat

This option specifies the format to be used for the timestamps that are included in the log file. Possible options are “date” and “epoch”. The former causes the tool to include the timestamps in the format “Sat Dec 24 05:24:50 2011”, whereas the latter causes the tool to include timestamps as the number of seconds since the Epoch. If left unspecified, this option defaults to “date”.

MaxAddressEntries

This variable specifies the maximum number of addresses that ipv6mon will keep track of at any given time. Since ipv6mon keeps the list of currently-used addresses in memory (such that they can be probed on a regular basis), the variable MaxAddressEntries indirectly enforces a limit on the amount of system memory that ipv6mon will use.

MaxCandidateEntries

This variable specifies the maximum number of "candidate addresses" that ipv6mon will create at any

given time. If left unspecified, it defaults to MaxAddressEntries/4.

AddressTimeout

This variable specifies the amount of time (in seconds) after which an IPv6 address will be considered “unused” if no response is received from that address.

CandidateAddressTimeout

This variable specifies the amount of time (in seconds) after which a “candidate” IPv6 address will be considered “unused” if no response is received from such address. Since candidate addresses are created as a result of a heuristics (rather than in response to packets originated from such addresses), CandidateAddressTimeout will typically be shorter than AddressTimeout, such that system resources are not tied to these (possibly unused) addresses for an unnecessarily long period of time.

MaxUnprobedInterval

This variable specifies the maximum amount of time (in seconds) that an address will remain “unprobed”. For obvious reasons, its value should be smaller than AddressTimeout and CandidateAddressTimeout, and should also usually account for more than one probe packet to be sent before an address is "timed out".

UnicastProbeInterval

This variable specifies the amount of time (in seconds) between unicast probes sent to an IPv6 address. Once MaxUnprobedInterval seconds have elapsed without probing an addresses, unicast probes will be sent every UnicastProbeInterval seconds until a response is received or the address is timed out.

McastEchoProbeInterval

This variable specifies the amount of time (in seconds) between ICMPv6 Echo Request probe packets sent to the all-nodes on-link multicast address (ff02::1). Multicast Echo Request packets can easily detect newly attached devices and probe existing devices. However, because they are sent to a multicast address, the interval should be long enough such that unnecessary traffic spikes are reduced.

Note: Some operating systems (notably Windows Vista and Windows 7) will not respond to ICMPv6 echo request messages sent to a multicast address. However, such systems can still be detected by means of probe packets that include an unsupported option of type 10xxxxxx.

McastUnrecProbeInterval

This variable specifies the amount of time (in seconds) between ICMPv6 Echo Request probe packets (containing an unrecognized option in a Destination Options header) sent to the all-nodes on-link multicast address (ff02::1). Multicast Echo Request packets can easily detect newly attached devices and probe existing devices. However, because they are sent to a multicast address, the interval should be long enough such that unnecessary traffic spikes are reduced.
ProbeType

This variable specifies the type of probe packets that will be used for both unicast and multicast probes.

“echo” specifies that only regular ICMPv6 Echo Request messages should be used. “unrec” specifies that only ICMPv6 Echo Request messages with an unsupported option should be used. “all” specifies that both types of probe packets should be used. If left unspecified, this option defaults to “all”.

Credits

The ipv6mon tool version 0.1 and related manuals were produced by Fernando Gont <fgont@si6networks.com> on behalf of the United Kingdom's Centre for the Protection of National Infrastructure (CPNI) <http://www.cpni.gov.uk>.

License