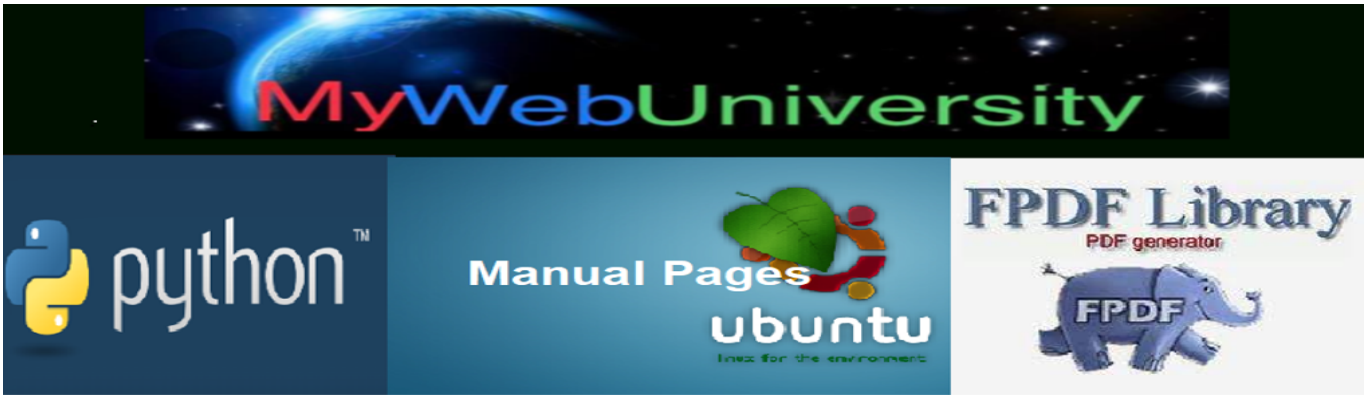

Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'resolved.conf.d.5'

$ man resolved.conf.d.5

RESOLVED.CONF(5) resolved.conf RESOLVED.CONF(5)

NAME

resolved.conf, resolved.conf.d - Network Name Resolution configuration files

SYNOPSIS

/etc/systemd/resolved.conf

/etc/systemd/resolved.conf.d/*.conf

/run/systemd/resolved.conf.d/*.conf

/usr/lib/systemd/resolved.conf.d/*.conf

DESCRIPTION

These configuration files control local DNS and LLMNR name resolution.

CONFIGURATION DIRECTORIES AND PRECEDENCE

The default configuration is set during compilation, so configuration is only needed when it is necessary to deviate from those defaults. Initially, the main configuration file in /etc/systemd/ contains commented out entries showing the defaults as a guide to the administrator. Local overrides can be created by editing this file or by creating drop-ins, as described below. Using drop-ins for local configuration is recommended over modifications to the main configuration file.

In addition to the "main" configuration file, drop-in configuration snippets are read from /usr/lib/systemd/*.conf.d/, /usr/local/lib/systemd/*.conf.d/, and /etc/systemd/*.conf.d/. Those drop-ins have higher precedence and override the main configuration file. Files in the *.conf.d/ configuration subdirectories are sorted by their filename in lexicographic order, regardless of in which of the subdirectories they reside. When multiple files specify the same option, for options which accept just a single value, the entry in the

file sorted last takes precedence, and for options which accept a list of values, entries are collected as they occur in the sorted files.

When packages need to customize the configuration, they can install drop-ins under /usr/. Files in /etc/ are reserved for the local administrator, who may use this logic to override the configuration files installed by vendor packages. Drop-ins have to be used to override package drop-ins, since the main configuration file has lower precedence. It is recommended to prefix all filenames in those subdirectories with a two-digit number and a dash, to simplify the ordering of the files.

To disable a configuration file supplied by the vendor, the recommended way is to place a symlink to /dev/null in the configuration directory in /etc/, with the same filename as the vendor configuration file.

OPTIONS

The following options are available in the [Resolve] section:

DNS=

A space-separated list of IPv4 and IPv6 addresses to use as system DNS servers. Each address can optionally take a port number separated with ":", a network interface name or index separated with "%", and a Server Name Indication (SNI) separated with "#". When IPv6 address is specified with a port number, then the address must be in the square brackets. That is, the acceptable full formats are "111.222.333.444:9953%ifname#example.com" for IPv4 and "[1111:2222::3333]:9953%ifname#example.com" for IPv6. DNS requests are sent to one of the listed DNS servers in parallel to suitable per-link DNS servers acquired from systemd-networkd.service(8) or set at runtime by external applications. For compatibility reasons, if this setting is not specified, the DNS servers listed in /etc/resolv.conf are used instead, if that file exists and any servers are configured in it. This setting defaults to the empty list.

FallbackDNS=

A space-separated list of IPv4 and IPv6 addresses to use as the fallback DNS servers. Please see DNS= for acceptable format of addresses. Any per-link DNS servers obtained from systemd-networkd.service(8) take precedence over this setting, as do any servers set via DNS= above or /etc/resolv.conf. This setting is hence only used if no other DNS server information is known. If this option is not given, a compiled-in list of DNS servers is used instead.

Domains=

A space-separated list of domains optionally prefixed with "~", used for two distinct purposes described below. Defaults to the empty list.

Any domains not prefixed with "~" are used as search suffixes when resolving single-label hostnames (domain names which contain no dot), in order to qualify them into fully-qualified domain names (FQDNs). These "search domains" are strictly processed in the order they are specified in, until the name with the suffix appended is found. For compatibility reasons, if this setting is not specified, the search domains listed in /etc/resolv.conf with the search keyword are used instead, if that file exists and any domains are configured in it.

The domains prefixed with "~" are called "routing domains". All domains listed here (both search domains and routing domains after removing the "~" prefix) define a search path that preferably directs DNS queries to this interface. This search path has an effect only when suitable per-link DNS servers are known. Such servers may be defined through the DNS= setting (see above) and dynamically at run time, for example from DHCP leases. If no per-link DNS servers are known, routing domains have no effect.

Use the construct "~." (which is composed from "~" to indicate a routing domain and "." to indicate the DNS root domain that is the implied suffix of all DNS domains) to use the DNS servers defined for this link preferably for all domains.

LLMNR=

Takes a boolean argument or "resolve". Controls Link-Local Multicast Name Resolution support (RFC 4795[1]) on the local host. If true, enables full LLMNR responder and resolver support. If false, disables both. If set to "resolve", only resolution support is enabled, but responding is disabled. Note that systemd-networkd.service(8) also maintains per-link LLMNR settings. LLMNR will be enabled on a link only if the per-link and the global setting is on.

MulticastDNS=

Takes a boolean argument or "resolve". Controls Multicast DNS support (RFC 6762[2]) on the local host. If true, enables full Multicast DNS responder and resolver support. If false, disables both. If set to "resolve", only resolution support is enabled, but responding is disabled. Note that systemd-networkd.service(8) also maintains per-link Multicast DNS settings. Multicast DNS will be enabled on a link only if the per-link

and the global setting is on.

DNSSEC=

Takes a boolean argument or "allow-downgrade". If true all DNS lookups are DNSSEC-validated locally (excluding LLMNR and Multicast DNS). If the response to a lookup request is detected to be invalid a lookup failure is returned to applications.

Note that this mode requires a DNS server that supports DNSSEC. If the DNS server does not properly support DNSSEC all validations will fail. If set to "allow-downgrade" DNSSEC validation is attempted, but if the server does not support DNSSEC properly, DNSSEC mode is automatically disabled. Note that this mode makes DNSSEC validation vulnerable to "downgrade" attacks, where an attacker might be able to trigger a downgrade to non-DNSSEC mode by synthesizing a DNS response that suggests DNSSEC was not supported. If set to false, DNS lookups are not DNSSEC validated.

Note that DNSSEC validation requires retrieval of additional DNS data, and thus results in a small DNS look-up time penalty.

DNSSEC requires knowledge of "trust anchors" to prove data integrity. The trust anchor for the Internet root domain is built into the resolver, additional trust anchors may be defined with dnssec-trust-anchors.d(5). Trust anchors may change at regular intervals, and old trust anchors may be revoked. In such a case DNSSEC validation is not possible until new trust anchors are configured locally or the resolver software package is updated with the new root trust anchor. In effect, when the built-in trust anchor is revoked and DNSSEC= is true, all further lookups will fail, as it cannot be proved anymore whether lookups are correctly signed, or validly unsigned. If DNSSEC= is set to "allow-downgrade" the resolver will automatically turn off DNSSEC validation in such a case.

Client programs looking up DNS data will be informed whether lookups could be verified using DNSSEC, or whether the returned data could not be verified (either because the data was found unsigned in the DNS, or the DNS server did not support DNSSEC or no appropriate trust anchors were known). In the latter case it is assumed that client programs employ a secondary scheme to validate the returned DNS data, should this be required.

It is recommended to set DNSSEC= to true on systems where it is known that the DNS server supports DNSSEC correctly, and where software or trust anchor updates happen regularly. On other systems it is recommended to set DNSSEC= to "allow-downgrade".

In addition to this global DNSSEC setting systemd-networkd.service(8) also maintains per-link DNSSEC settings. For system DNS servers (see above), only the global DNSSEC setting is in effect. For per-link DNS servers the per-link setting is in effect, unless it is unset in which case the global setting is used instead.

Site-private DNS zones generally conflict with DNSSEC operation, unless a negative (if the private zone is not signed) or positive (if the private zone is signed) trust anchor is configured for them. If "allow-downgrade" mode is selected, it is attempted to detect site-private DNS zones using top-level domains (TLDs) that are not known by the DNS root server. This logic does not work in all private zone setups.

Defaults to "no".

DNSOverTLS=

Takes a boolean argument or "opportunistic". If true all connections to the server will be encrypted. Note that this mode requires a DNS server that supports DNS-over-TLS and has a valid certificate. If the hostname was specified in DNS= by using the format format "address#server_name" it is used to validate its certificate and also to enable Server Name Indication (SNI) when opening a TLS connection. Otherwise the certificate is checked against the server's IP. If the DNS server does not support DNS-over-TLS all DNS requests will fail.

When set to "opportunistic" DNS request are attempted to send encrypted with DNS-over-TLS. If the DNS server does not support TLS, DNS-over-TLS is disabled. Note that this mode makes DNS-over-TLS vulnerable to "downgrade" attacks, where an attacker might be able to trigger a downgrade to non-encrypted mode by synthesizing a response that suggests DNS-over-TLS was not supported. If set to false, DNS lookups are send over UDP.

Note that DNS-over-TLS requires additional data to be send for setting up an encrypted connection, and thus results in a small DNS look-up time penalty.

Note that in "opportunistic" mode the resolver is not capable of authenticating the server, so it is vulnerable to "man-in-the-middle" attacks.

In addition to this global DNSOverTLS= setting systemd-networkd.service(8) also maintains per-link DNSOverTLS= settings. For system DNS servers (see above), only the global DNSOverTLS= setting is in effect. For per-link DNS servers the per-link setting is in effect, unless it is unset in which case the global setting is used instead.

Defaults to "no".

Cache=

Takes a boolean or "no-negative" as argument. If "yes", resolving a domain name which already got queried earlier will return the previous result as long as it is still valid, and thus does not result in a new network request. Be aware that turning off caching comes at a performance penalty, which is particularly high when DNSSEC is used. If "no-negative" (the default), only positive answers are cached.

Note that caching is turned off by default for host-local DNS servers. See CacheFromLocalhost= for details.

CacheFromLocalhost=

Takes a boolean as argument. If "no" (the default), and response cames from host-local IP address (such as 127.0.0.1 or ::1), the result wouldn't be cached in order to avoid potential duplicate local caching.

DNSStubListener=

Takes a boolean argument or one of "udp" and "tcp". If "udp", a DNS stub resolver will listen for UDP requests on address 127.0.0.53 port 53. If "tcp", the stub will listen for TCP requests on the same address and port. If "yes" (the default), the stub listens for both UDP and TCP requests. If "no", the stub listener is disabled.

Note that the DNS stub listener is turned off implicitly when its listening address and port are already in use.

DNSStubListenerExtra=

Takes an IPv4 or IPv6 address to listen on. The address may be optionally prefixed with a protocol name ("udp" or "tcp") separated with ":". If the protocol is not specified, the service will listen on both UDP and TCP. It may be also optionally suffixed by a numeric port number with separator ":". When an IPv6 address is specified with a port number, then the address must be in the square brackets. If the port is not specified, then the service uses port 53. Note that this is independent of the primary DNS stub configured with DNSStubListener=, and only configures additional sockets to listen on. This option can be specified multiple times. If an empty string is assigned, then the all previous assignments are cleared. Defaults to unset.

Examples:

DNSStubListenerExtra=192.168.10.10

DNSStubListenerExtra=2001:db8:0:f102::10

DNSStubListenerExtra=192.168.10.11:9953

DNSStubListenerExtra=[2001:db8:0:f102::11]:9953

DNSStubListenerExtra=tcp:192.168.10.12

DNSStubListenerExtra=udp:2001:db8:0:f102::12

DNSStubListenerExtra=tcp:192.168.10.13:9953

DNSStubListenerExtra=udp:[2001:db8:0:f102::13]:9953

ReadEtcHosts=

Takes a boolean argument. If "yes" (the default), systemd-resolved will read /etc/hosts, and try to resolve hosts or address by using the entries in the file before sending query to DNS servers.

ResolveUnicastSingleLabel=

Takes a boolean argument. When false (the default), systemd-resolved will not resolve A and AAAA queries for single-label names over classic DNS. Note that such names may still be resolved if search domains are specified (see Domains= above), or using other mechanisms, in particular via LLMNR or from /etc/hosts. When true, queries for single-label names will be forwarded to global DNS servers even if no search domains are defined.

This option is provided for compatibility with configurations where public DNS servers are not used. Forwarding single-label names to servers not under your control is not standard-conformant, see IAB Statement[3], and may create a privacy and security risk.

SEE ALSO

systemd(1), systemd-resolved.service(8), systemd-networkd.service(8), dnssec-trust-anchors.d(5), resolv.conf(5)

NOTES

1. RFC 4795

   https://tools.ietf.org/html/rfc4795

2. RFC 6762

   https://tools.ietf.org/html/rfc6762

3. IAB Statement

https://www.iab.org/documents/correspondence-reports-documents/2013-2/iab-statement-dotless-domains-considered-harmful/