strongSwan - Feature #1482

Allow changing init_limit_half_open etc. at runtime by reloading strongswan.conf

26.05.2016 10:18 - Danny Kulchinsky

Status:	Feedback	**Start date:**	26.05.2016
Priority:	Normal	**Due date:**	
Assignee:		**Estimated time:**	0.00 hour
Category:	configuration		
Target version:			
Resolution:			

Description

We would like to implement IKE_SA_INIT Dropping based on the number of half-open connections.

We update the parameter init_limit_half_open to 500 and reloaded ipsec (ipsec reload & ipsec update), but seems that this didn't have any effect.

Will Charon pickup this parameter during reload/update ? or we need to restart Charon ?

Hoping we can do this without affecting established connections.

```
# Options for the charon IKE daemon.
charon {

    # Accept unencrypted ID and HASH payloads in IKEv1 Main Mode.
    # accept_unencrypted_mainmode_messages = no

    # Maximum number of half-open IKE_SAs for a single peer IP.
    # block_threshold = 5

    # Whether relations in validated certificate chains should be cached in
    # memory.
    # cert_cache = yes

    # Send Cisco Unity vendor ID payload (IKEv1 only).
    # cisco_unity = no

    # Close the IKE_SA if setup of the CHILD_SA along with IKE_AUTH failed.
    close_ike_on_child_failure = yes

    # Number of half-open IKE_SAs that activate the cookie mechanism.
    cookie_threshold = 10

    # Use ANSI X9.42 DH exponent size or optimum size matched to cryptographic
    # strength.
    dh_exponent_ansi_x9_42 = no

    # Use RTLD_NOW with dlopen when loading plugins and IMV/IMCs to reveal
    # missing symbols immediately.
    # dlopen_use_rtld_now = no

    # DNS server assigned to peer via configuration payload (CP).
    # dns1 =

    # DNS server assigned to peer via configuration payload (CP).
    # dns2 =

    # Enable Denial of Service protection using cookies and aggressiveness
    # checks.
    # dos_protection = yes

    # Compliance with the errata for RFC 4753.
    # ecp_x_coordinate_only = yes
    # Free objects during authentication (might conflict with plugins).
    # flush_auth_cfg = no

    # Maximum size (complete IP datagram size in bytes) of a sent IKE fragment
    # when using proprietary IKEv1 or standardized IKEv2 fragmentation (0 for
    # address family specific        default values). If specified this limit is
    # used for both IPv4 and IPv6.
    # fragment_size = 0

    # Name of the group the daemon changes to after startup.
    group = strongswan

    # Timeout in seconds for connecting IKE_SAs (also see IKE_SA_INIT DROPPING).
    half_open_timeout = 30

    # Enable hash and URL support.
    # hash_and_url = no

    # Allow IKEv1 Aggressive Mode with pre-shared keys as responder.
    # i_dont_care_about_security_and_use_aggressive_mode_psk = no

    # Whether to ignore the traffic selectors from the kernel's acquire events
    # for IKEv2 connections (they are not used for IKEv1).
    # ignore_acquire_ts = no

    # A space-separated list of routing tables to be excluded from route
    # lookups.
    # ignore_routing_tables =

    # Maximum number of IKE_SAs that can be established at the same time before
    # new connection attempts are blocked.
    # ikesa_limit = 0

    # Number of exclusively locked segments in the hash table.
    ikesa_table_segments = 16

    # Size of the IKE_SA hash table.
    # ikesa_table_size = 1
    ikesa_table_size = 2048

    # Whether to close IKE_SA if the only CHILD_SA closed due to inactivity.
    # inactivity_close_ike = no

    # Limit new connections based on the current number of half open IKE_SAs,
    # see IKE_SA_INIT DROPPING in strongswan.conf(5).
    init_limit_half_open = 500

    # Limit new connections based on the number of queued jobs.
    # init_limit_job_load = 0

    # Causes charon daemon to ignore IKE initiation requests.
    # initiator_only = no

    # Install routes into a separate routing table for established IPsec
    # tunnels.
    # install_routes = yes

    # Install virtual IP addresses.
    # install_virtual_ip = yes

    # The name of the interface on which virtual IP addresses should be
    # installed.
    # install_virtual_ip_on =

    # Check daemon, libstrongswan and plugin integrity at startup.
    integrity_test = no
    # A comma-separated list of network interfaces that should be ignored, if
    # interfaces_use is specified this option has no effect.
    # interfaces_ignore =

    # A comma-separated list of network interfaces that should be used by
    # charon. All other interfaces are ignored.
    interfaces_use = eth1

    # NAT keep alive interval.
    # keep_alive = 20s

    # Plugins to load in the IKE daemon charon.
    # load =

    # Determine plugins to load via each plugin's load option.
    # load_modular = no

    # Initiate IKEv2 reauthentication with a make-before-break scheme.
    # make_before_break = no

    # Maximum number of IKEv1 phase 2 exchanges per IKE_SA to keep state about
    # and track concurrently.
    # max_ikev1_exchanges = 3

    # Maximum packet size accepted by charon.
    # max_packet = 10000

    # Enable multiple authentication exchanges (RFC 4739).
    # multiple_authentication = yes

    # WINS servers assigned to peer via configuration payload (CP).
    # nbns1 =

    # WINS servers assigned to peer via configuration payload (CP).
    # nbns2 =

    # UDP port used locally. If set to 0 a random port will be allocated.
    # port = 500

    # UDP port used locally in case of NAT-T. If set to 0 a random port will be
    # allocated.  Has to be different from charon.port, otherwise a random port
    # will be allocated.
    # port_nat_t = 4500

    # By default public IPv6 addresses are preferred over temporary ones (RFC
    # 4941), to make connections more stable. Enable this option to reverse
    # this.
    # prefer_temporary_addrs = no

    # Process RTM_NEWROUTE and RTM_DELROUTE events.
    # process_route = yes

    # Delay in ms for receiving packets, to simulate larger RTT.
    # receive_delay = 0

    # Delay request messages.
    receive_delay_request = no

    # Delay response messages.
    receive_delay_response = no

    # Specific IKEv2 message type to delay, 0 for any.
    # receive_delay_type = 0

    # Size of the AH/ESP replay window, in packets.
    # replay_window = 32
    # Base to use for calculating exponential back off, see IKEv2 RETRANSMISSION
    # in strongswan.conf(5).
    # retransmit_base = 1.8

    # Timeout in seconds before sending first retransmit.
    # retransmit_timeout = 4.0

    # Number of times to retransmit a packet before giving up.
    retransmit_tries = 3

    # Interval in seconds to use when retrying to initiate an IKE_SA (e.g. if
    # DNS resolution failed), 0 to disable retries.
    # retry_initiate_interval = 0

    # Initiate CHILD_SA within existing IKE_SAs.
    reuse_ikesa = no

    # Numerical routing table to install routes to.
    # routing_table =

    # Priority of the routing table.
    # routing_table_prio =

    # Delay in ms for sending packets, to simulate larger RTT.
    # send_delay = 0

    # Delay request messages.
    send_delay_request = no

    # Delay response messages.
    send_delay_response = no

    # Specific IKEv2 message type to delay, 0 for any.
    # send_delay_type = 0

    # Send strongSwan vendor ID payload
    # send_vendor_id = no

    # Whether to enable Signature Authentication as per RFC 7427.
    # signature_authentication = yes

    # Whether to enable constraints against IKEv2 signature schemes.
    # signature_authentication_constraints = yes

    # Number of worker threads in charon.
    threads = 64

    # Name of the user the daemon changes to after startup.
    user = strongswan

    crypto_test {

        # Benchmark crypto algorithms and order them by efficiency.
        # bench = no

        # Buffer size used for crypto benchmark.
        # bench_size = 1024

        # Number of iterations to test each algorithm.
        # bench_time = 50

        # Test crypto algorithms during registration (requires test vectors
        # provided by the test-vectors plugin).
        # on_add = no

        # Test crypto algorithms on each crypto primitive instantiation.
        # on_create = no

        # Strictly require at least one test vector to enable an algorithm.
        # required = no

        # Whether to test RNG with TRUE quality; requires a lot of entropy.
        # rng_true = no

    }

    host_resolver {

        # Maximum number of concurrent resolver threads (they are terminated if
        # unused).
        # max_threads = 3

        # Minimum number of resolver threads to keep around.
        # min_threads = 0

    }

    leak_detective {

        # Includes source file names and line numbers in leak detective output.
        # detailed = yes

        # Threshold in bytes for leaks to be reported (0 to report all).
        # usage_threshold = 10240

        # Threshold in number of allocations for leaks to be reported (0 to
        # report all).
        # usage_threshold_count = 0

    }

    processor {

        # Section to configure the number of reserved threads per priority class
        # see JOB PRIORITY MANAGEMENT in strongswan.conf(5).
        priority_threads {

        }

    }

    # Section containing a list of scripts (name = path) that are executed when
    # the daemon is started.
    start-scripts {

    }

    # Section containing a list of scripts (name = path) that are executed when
    # the daemon is terminated.
    stop-scripts {

    }

    tls {

        # List of TLS encryption ciphers.
        # cipher =

        # List of TLS key exchange methods.
        # key_exchange =

        # List of TLS MAC algorithms.
        # mac =
        # List of TLS cipher suites.
        # suites =

    }

    x509 {

        # Discard certificates with unsupported or unknown critical extensions.
        # enforce_critical = yes

    }

}
```

History

#1 - 26.05.2016 10:30 - Tobias Brunner

- Category set to configuration

- Status changed from New to Feedback

> We update the parameter init_limit_half_open to 500 and reloaded ipsec (ipsec reload & ipsec update), but seems that this didn't have any effect.

These commands don't reload strongswan.conf (as documented at the top of that page a SIGHUP has to be sent to charon to do so).

> Will Charon pickup this parameter during reload/update ? or we need to restart Charon ?

This particular parameter is only read when the daemon starts, it currently can't be changed at runtime (source:src/libcharon/network/receiver.c#L646). So yes, a restart is required.

#2 - 26.05.2016 11:09 - Danny Kulchinsky

Tobias Brunner wrote:

> > We update the parameter init_limit_half_open to 500 and reloaded ipsec (ipsec reload & ipsec update), but seems that this didn't have any effect.
>
> These commands don't reload strongswan.conf (as documented at the top of that page a SIGHUP has to be sent to charon to do so).
>
> > Will Charon pickup this parameter during reload/update ? or we need to restart Charon ?
>
> This particular parameter is only read when the daemon starts, it currently can't be changed at runtime (source:src/libcharon/network/receiver.c#L646). So yes, a restart is required.

Yes, you're right - we actually have a script that sends SIGHUP to Charon as well as ipsec reload/update.

Are you considering to allow this parameter to be updated during runtime ?

#3 - 26.05.2016 11:14 - Tobias Brunner

- Tracker changed from Issue to Feature

- Subject changed from No effect after changing init_limit_half_open and reloading Charon (ipsec reload & ipsec update) to Allow changing init_limit_half_open etc. at runtime by reloading strongswan.conf

> Are you considering to allow this parameter to be updated during runtime ?

There are currently no plans to do so (but I've changed this ticket to a feature request to track it).

#4 - 26.05.2016 14:49 - Danny Kulchinsky

Tobias Brunner wrote:

> > Are you considering to allow this parameter to be updated during runtime ?
>
> There are currently no plans to do so (but I've changed this ticket to a feature request to track it).

Awesome ! Thank you :)

Adding threads on the fly would also be nice ;)