LWIP WG 2011-03-28

Minimal IKEv2
Tero Kivinen <kivinen@iki.fi>
AuthenTec
draft-kivinen-ipsecme-ikev2-minimal-00.txt
Example Use Case

- Garage door opener
  - Two buttons:
    - one to unlock and open door
    - another to close and lock the door
  - One led for feedback
  - Uses two-way radio communications
  - Obviously needs some kind of security
  - Battery powered
Example protocol

• Protocol can be very simple:
  • Send packet to server to start open/close door
  • Get packet back to acknowledge the command
  • Get status messages every second while door is moving
  • Get final message when operation is done
Protocol effects

- Device only wakes up when button is pressed
  - It always initiates the communication, it does not need to listen radio when it is sleeping, and it cannot reply to any messages while sleeping
- Device stays awake for some time after the button is pressed and if receives status packet blinks led and waits for more status packets.
- After certain timeout device goes back to sleep
What this means for IKEv2

- Device only needs work as IKEv2 Initiator
  - No need to work as IKEv2 Responder
- Only creates one IKEv2 SA and one IPsec SA
  - No need to support SA management operations like creating new IPsec SAs, rekeying, deleting SAs, etc.
- No need to do NAT-T, Configuration payloads, EAP authentication, Cookies, Multiple child SAs etc
- The server end would most likely be some kind of Home area network server (PC or similar).
- Pre-shared keys or RAW RSA keys authentication
  - No X.509 certificates
Authentication

- Pre-shared keys
  - Shared key printed on paper or in electronic form
  - Typed in to the home area gateway
- Raw RSA keys
  - Fingerprint of device is distributed as Pre-shared keys
  - Device imprints to first home area gateway it connects to
  - Some form of reset can be implemented to allow reimprinting
Implementation

- I created a prototype implementation of the minimal IKEv2 protocol usable for such scenarios and it took me less than a day to write the code and less than 1000 lines of perl source code.
  - I implemented sending ICMP Ping packet as didn't want to start writing server end to answer my requests...
- Implementing minimal IKEv2 is very simple compared to full implementation.
- There are some optimizations which can be done when only supporting minimal set of features.
Examples of Optimizations

- Message ID and Window code
  - In IKEv2 there is requirement to keep track of the Message IDs received and transmitted to protect replays
- Minimal implementation
  - Sends only IKE_SA_INIT and IKE_AUTH
    - No need to keep track of transmitted Message IDs
    - Does not do anything useful based on received messages (only sends empty acknowledgement or error)
  - No need to keep track of received Message IDs
$addr = sockaddr_in(0, inet_aton($$ikev2{dsthost}));

hexl_print("Sending esp packet", $packet);

$packet = pack("a4Na*a*", $$ipsec{spi_out}, 0, $iv, $packet);

if ($$ikev2{cipher} ne 'null') {
    my($packet, $pad, $i, $iv, $cipher, $addr);
    $ipsec = ikev2($ikev2);
    # Negotiate IKE
    $$ikev2{id} = 'tk@iki.fi';
    $$ikev2{cipher} = 'aes';
    $$ikev2{srchost} = "172.30.4.74";
    # $$ikev2{dsthost} = "172.30.4.74";
    $$ikev2{srchost} = "0.0.0.0";
}

use Digest::HMAC_SHA1 qw(hmac_sha1);

package IkeV2;
require 5.6.0;

# mini-ike.pl -- Minimal IKIv2 initiator
#!/usr/local/bin/perl

$pad = 0;

# $packet = generate_gen_hdr($ikev2, $next_payload, $payload_data);

# Generate generic header
$packet = $cipher->decrypt($packet);

die "Invalid SPI" if ($spi ne $$ipsec{spi_in});

die "MAC check failed" if ($mac ne substr($packet, -12));

$packet = substr($packet, 20);

hexl_print("Received esp packet", $packet);

die "Recv failed: $!"if (!defined($addr));

# Calculate keys
calculate_keys($ikev2);

# Parse IKE SA init
parse_ike_sa_init($ikev2, $$ikev2{ike_sa_init_r});

# Receive timeout
die "Receive timeout";
sub parse_notify {
    # parse_notify($ikev2, $payload_hash, $payload_str);
    # Parse notify payload
}

sub parse_nonce {
    # parse_nonce($ikev2, $payload_hash, $payload_str);
}

my($ikev2, $hash, $payload) = @_;
Conclusions

- IKEv2 is very small protocol when only minimal features are implemented
- Certificate support would multiply the code size
- Pre-shared keys or RAW RSA keys are feasible options for authentication in this kind if use scenarios
- My draft describes more of those optimizations possible:
  - draft-kivinen-ipsecme-ikev2-minimal-00.txt