IP Security Maintenance and Extensions (IPsecME) WG

IETF 110, Monday, March 8, 2021

Chairs: Tero Kivinen
Yoav Nir

Responsible AD: Benjamin Kaduk

Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

• By participating in the IETF, you agree to follow IETF processes and policies.

• If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.

• As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.

• Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.

• As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:
•BCP 9 (Internet Standards Process)
•BCP 25 (Working Group processes)
•BCP 25 (Anti-Harassment Procedures)
•BCP 54 (Code of Conduct)
•BCP 78 (Copyright)
•BCP 79 (Patents, Participation)
•https://www.ietf.org/privacy-policy/ (Privacy Policy)

Administrative Tasks

Bluesheets

We need volunteers to be:

- Two note takers
- One jabber scribe

Jabber: xmpp:ipsecme@jabber.ietf.org?join

MeetEcho: https://meetings.conf.meetecho.com/ietf110/?group=ipsecme&short=&item=1

Notes: https://codimd.ietf.org/notes-ietf-110-ipsecme

Agenda (1)

- Note Well, technical difficulties and agenda bashing – Chairs (5 min) (13:00-13:05)
- Document Status – Chairs (5 min) (13:05-13:10)
  - IKEv2 Labeled IPsec – Paul Wouters (5 min) (13:10-13:15)
- Work items
  - Group Key Management using IKEv2 – Valery Smyslov (10 min) (13:15-13:25)
  - IPTFS Base Draft WGLC Changes – Christian Hopps (10 min) (13:25-13:35)
  - Management (Yang adoption and update, new SNMP draft) – Christian Hopps (10 min) (13:35-13:45)

Agenda (2)

- New items
  - IKEv2 Configuration for Encrypted DNS – Valery Smyslov (5 min) (13:45-13:50)
  - New payload format for IKEv2 – Valery Smyslov (15 min) (13:50-14:05)
  - IKEv1 graveyard – Paul Wouters (5 min) (14:05-14:10)
  - BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar (10 min) (14:10-14:20)
  - X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert (5 min) (14:20-14:25)
  - IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters (5 min) (14:25-14:30)
  - IKEv2 Multi-SA Performance – Paul Wouters (5 min) (14:30-14:35)
  - AOB + Open Mic (14:35-15:00)

WG Status Report

Published as RFC8983

- draft-ietf-ipsecme-ipv6-ipv4-codes

In WGLC:

- draft-ietf-ipsecme-ikev2-intermediate
- draft-hopps-ipsecme-iptfs

Work in progress:

- draft-fedyk-ipsecme-yang-iptfs
- draft-ietf-ipsecme-g-ikev2
- draft-ietf-ipsecme-ikev2-multiple-ke
- draft-ietf-ipsecme-labeled-ipsec
- draft-smyslov-ipsecme-rfc8229bis

redhat.

draft-iet—ipsecme-ikev2-labeled-ipsec

IETF-110, IPsecME,
March, 2021

Paul Wouters, RHEL Security

draft-ietf-ipsecme-labeled-ipsec

- https://tools.ietf.org/html/draft-ietf-ipsecme-labeled-ipsec-04
- No change since IETF-109, draft will expire soon
- Linux kernel implementation stable and unchanged for years
- Implementation for IKEv1 has existed for 10 years
  - libreswan implementation stable – via ModeCFG, no specification
- Implementation for IKEv2 (this draft)
  - libreswan published code last month, minor bugs to work out
  - No issues with draft specification
    - but might be worth specifying the SElinux implementation seperately
  - Ready for interop with another implementor
  - WGLC ?

Presentations

- **Group Key Management using IKEv2 – Valery Smyslov**
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

Group Key Management using IKEv2

draft-ietf-ipsecme-g-ikev2-02

Valery Smyslov
ELVIS-PLUS

Brian Weis
Independent

IETF 110

Securing IP Multicast

- IP multicast applications
  - Contain at least 1 sender, and N receivers
  - Take advantage of the network to route and replicate IP packets, such that the same packet reaches all N receivers
- This requires senders and receivers to share setup an IPsec SA using the same keys
  - The IPsec policy and keys are not individually negotiated, but instead of distributed by a Group Controller / Key Server (GCKS) to Group Members (GMs)
  - A GM invokes a unicast Registration protocol to authenticate to the GCKS. The GCKS then authorizes the GM, and distributes IPsec policy and keys to the GM.
  - A Rekey protocol enforces a time-based key rollover strategy

Distribution of Group Keys in IEEE 802.15

- IEEE 802.15.9 specified IKEv2 as one of KMPs for IEEE 802.15.4
  - IEEE Std 802.15.9-2015 left group keys distribution out of scope
- Draft 05 version of the IEEE Std 802.15.9 standard (March 2021) specifies that G-IKEv2 is used for group key distribution
  - GSA_INBAND_REKEY over unicast SA is used
  - SPI field in GSA payload is used to specify the type of group key

Document Status

- Has been in development for several years
  - few implementations of early draft versions exist
- Has been adopted by IPSECME WG in 2019
- Version -01 (July 2020): major rewrite
- Version -02 (January 2021): minor update
- For authors the draft looks mature
  - however, more reviews are needed

Outline of -01 Changes

- Policy representation changed
  - before: IKEv1 style, mostly using attributes
  - now: IKEv2 style – using transforms, attributes are still used for variables
- Format of GSA and KD changed
- Group key representation changed
  - before: group keys were transferred in clear inside KD payload
  - now: all keys are encrypted inside KD payload, either using SK_d derived key or using other group key
- LKH (Logical Key Hierarchy) is integrated
  - before: dedicated attributes were used to transfer LKH keys
  - now: LKH functionality is integrated into core G-IKEv2 protocol, GM semantics doesn't depend on key management method

Outline of -01 Changes (cont.)

- IANA considerations are rewritten
  - now it's more an extension to IKEv2 than a separate protocol (IKEv2 IANA registries are used)
  - many parameters have been renamed to better reflect their purpose
- A lot of clarifications
  - AUTH payload calculation for GSA_REKEY messages is described in detail
  - introduced means to indicate cross-dependency of supported algorithms in SAg payload
  - using PPK in G-IKEv2 is clarified
  - using ESN is clarified (in -02)
  - failover in situations when rekey message was missed clarified (using NEXT_SPI)
  - example of using LKH is rewtritten

GSA Payload

Contains policy necessary to participating in the group:

- Protocol (GIKE_REKEY, AH, ESP)
- Traffic Selector
- Transforms for algorithms and methods used in the policy
- Attributes for variables that change over time (like initial Message-ID)
- GSA format is now common for KEK (GIKE_REKEY) and TEK (AH, ESP)
  - GAP (Group Policy) shares the same format and is distinguished by zero protocol

KD Payload

Contains keying material necessary for the policy in the GSA payload:

- One or more keys are conveyed in the KD payload
- Security parameters are also conveyed in the KD payload
- Each key is individually wrapped in a new structure Wrapped Key
- Each Wrapped Key is encrypted using either SK_d derived key or other group key
- LKH capability is now integrated into G-IKEv2 core and is achieved by including several keys into the KD payload logically connected by encrypting next key with previous one
- Wrapped Keys may contain either group keys (common for a whole group or for subset of its members) or member keys (allows for provision keys for a member during GSA registration, needed for LKH)

IDg Payload

Contains identity of the group a GM wants to join (no changes since -00):

- has the same format as IKEv2 ID payload
- only some ID types are expected to be used
  - `ID_KEY_ID` MUST be supported
  - `ID_IPV4_ADDR`, `ID_IPV6_ADDR`, ID_FQDN, `ID_RFC822_ADDR` SHOULD be supported

Reused IKEv2 payloads

Payloads that have the same types as in IKEv2, but slightly different semantics:

- SAg (GM Supported Transforms)
  - declares which Transforms a GM is willing to accept
  - has the same format as IKEv2 SA payload, but slightly different semantics, which allow to indicate inter-dependency of supported algorithms
- D (Delete Payload)
  - used when the GCKS may want to signal to group members to delete policy (e.g., data flows finished, change of policy)
  - semantics is slightly different from IKEv2, allowing to delete all SAs

New Notifications

- INVALID_GROUP_ID (error notify)
  - GCKS informs GM that the requested Group ID in a registration protocol is invalid
- AUTHORIZATION_FAILED (error notify)
  - GCKS informs GM that it is not authorized to join the requested Group ID
- REGISTRATION_FAILED (error notify)
  - GCKS informs GM that for some reason the GM cannot join the group
  - GM sends to GCKS to unregister from the group
- SENDER (status notify)
  - GM informs the GCKS about its intention to be a sender in the group
  - requests a number of Sender-ID values, that are used as part of a counter-mode transform nonce (RFC 6054)
- REKEY_IS_NEEDED (status notify) – added in -01
  - GCKS informs GM that it must rekey IKE SA before receiving sensitive information (used in PPK scenarios)

Reused IKEv2 Notifications

- USE_TRANSPORT_MODE
  - semantics is changed, so that Protocol and SPI fields are used to indicate which SA to create in transport mode
  - multiple instances can be sent if multiple SAs are being created

Thank you!

- Comments?
- Questions?
- Please review the document

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- **IPTFS Base Draft WGLC Changes – Christian Hopps**
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

Christian Hopps
LabN Consulting, LLC

IP Traffic Flow Security

Improving IPsec Traffic Flow Confidentiality

IETF 110 – “draft-ietf-ipsecme-iptfs-07”

Update Since IETF 109

- Transport Area Review from Joe Touch Completed
  - Suggested changes incorporated from review in -04
- Pre-WGLC WG Discussion
  - Valery
    - Request more generic identifier names
    - Fix overhead numbers and update comparison data
    - Changes published in -05
  - Tero – [pre-]WGLC review
    - Changes published in -06

Update Since IETF 109

- WGLC (3 week)
  - 1/24/2021 - 2/14/2021
  - WGLC reviews with suggested changes from:
    - Tero Kivinen
    - Sean Turner
    - Paul Wouters
    - Valery Smyslov
    - Michael Richardson
- Published -07 based on WGLC mailing list discussion

Notable Changes Through WGLC

- Normative
  - No fragmenting over multiple SAs
  - P-bit in header for PLMTUD implementations (indicates probing in progress).
- Informative
  - Editorial and organizational cleanup
  - [IP]TFS_ -> AGGFRAG_ for on-wire identifiers, as well as more generic text.
  - Expanded text on ordering packet processing on receiver
  - Expanded text on how extra padding can be used to avoid fragmentation
  - Summary of receiver actions with internal references
  - Fixed IPsec overhead for comparisons in the Appendix (same conclusions)

Moving Forward

- 3-week WGLC period completed
  - All reviews addressed, with changes incorporated
- To the IESG?

Questions and Comments

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- **Management (Yang adoption and update, new SNMP draft) – Christian Hopps**
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

Donald Fedyk
Christian Hopps
LabN Consulting, LLC

YANG Model for IP Traffic Flow Security

IETF 110 – “draft-fedyk-ipsecme-yang-iptfs-02”

Changes since IETF110

- Updates based on comments received during adoption call
  - Reorganized statistics (separated inner and outer stats)
  - Added max-aggregation-time
  - Tracked name changes from I2NSF YANG model (per IESG publication review)
  - Changes included in draft-fedyk-ipsecme-yang-iptfs-02
- Adopted by WG
  - Republishing as draft-ietf-ipsecme-yang-iptfs-00

Current Tree (ike version shown)

```
module: ietf-ipsecme-iptfs
    augment /nsfike:ipsec-ike/nsfike:conn-entry/nsfike:spd
              /nsfike:spd-entry/nsfike:ipsec-policy-config
              /nsfike:processing-info/nsfike:ipsec-sa-cfg:
      +--rw traffic-flow-security
         +--rw congestion-control?     boolean
         +--rw packet-size
         |  +--rw use-path-mtu-discovery?   boolean
         |  +--rw outer-packet-size?        uint16
         +--rw (tunnel-rate)?
         |  +--:(l2-fixed-rate)
         |  |  +--rw l2-fixed-rate?    uint64
         |  +--:(l3-fixed-rate)
         |     +--rw l3-fixed-rate?    uint64
         +--rw dont-fragment?          boolean
         +--rw max-aggregation-time?   decimal64
    augment /nsfike:ipsec-ike/nsfike:conn-entry/nsfike:child-sa-info:
      +--ro traffic-flow-security
         +--ro congestion-control?     boolean
         +--ro packet-size
         |  +--ro use-path-mtu-discovery?   boolean
         |  +--ro outer-packet-size?        uint16
         +--ro (tunnel-rate)?
         |  +--:(l2-fixed-rate)
         |  |  +--ro l2-fixed-rate?    uint64
         |  +--:(l3-fixed-rate)
         |     +--ro l3-fixed-rate?    uint64
         +--ro dont-fragment?          boolean
```

```
 augment /nsfike:ipsec-ike/nsfike:conn-
entry/nsfike:child-sa-info:
       +--ro ipsec-stats {ipsec-stats}?
       |  +--ro tx-pkts?        uint64
       |  +--ro tx-octets?      uint64
       |  +--ro tx-drop-pkts?   uint64
       |  +--ro rx-pkts?        uint64
       |  +--ro rx-octets?      uint64
       |  +--ro rx-drop-pkts?   uint64
       +--ro iptfs-inner-pkt-stats {iptfs-stats}?
       |  +--ro tx-pkts?              uint64
       |  +--ro tx-octets?            uint64
       |  +--ro rx-pkts?              uint64
       |  +--ro rx-octets?            uint64
       |  +--ro rx-incomplete-pkts?   uint64
       +--ro iptfs-outer-pkt-stats {iptfs-stats}?
          +--ro tx-all-pad-pkts?       uint64
          +--ro tx-all-pad-octets?     uint64
          +--ro tx-extra-pad-pkts?     uint64
          +--ro tx-extra-pad-octets?   uint64
          +--ro rx-all-pad-pkts?       uint64
          +--ro rx-all-pad-octets?     uint64
          +--ro rx-extra-pad-pkts?     uint64
          +--ro rx-extra-pad-octets?   uint64
          +--ro rx-errored-pkts?       uint64
          +--ro rx-missed-pkts?        uint64
```

Next Steps

- Track changes in the base IPsec specification
- Solicit feedback from the WG

Definitions of Managed Objects for IP Traffic Flow Security

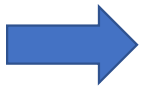
IETF 110 – “draft-fedyk-ipsecme-mib-iptfs-00”

Donald Fedyk
Eric Kinzie
LabN Consulting, LLC

Objective: Provide a read only SNMP MIB

- Some operators still require read-only SNMP support
- Mechanically derived from the YANG model

```
leaf l2-fixed-rate {
        type uint64;
        description
          "Target bandwidth/bit rate in bps for iptfs tunnel. This
           fixed rate is the nominal timing for the fixed size packet.
           If congestion control is enabled the rate may be adjusted
           down (or up if unset).";
        reference
          "draft-ietf-ipsecme-iptfs section 4.1";
      }
```



```
l2FixedRate OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "TFS bit rate may be specified at layer 2 wire rate.
          Target bandwidth/bit rate in bps for iptfs tunnel.
          This rate is the nominal timing for the fixed size
          packet. If congestion control is enabled the rate may
          be adjusted down (or up if unset)."
        ::= { iptfsConfigTableEntry 5 }
```

MIB Tree

```
---- iptfsMIB(1.3.6.1.3.500)
    +---- iptfsMIBObjects(1)
    |  +---- iptfsGroup(1)
    |  |  +---- iptfsConfigTable(1)
    |  |     +---- iptfsConfigTableEntry(1) [iptfsConfigSaIndex]
    |  |        +---- iptfsConfigSaIndex(1) Integer32
    |  |        +--r- congestionControl(2)  TruthValue
    |  |        +--r- usePathMtu(3)         TruthValue
    |  |        +--r- outerPacketSize(4)    UnsignedShort
    |  |        +--r- l2FixedRate(5)        Counter64
    |  |        +--r- l3FixedRate(6)        Counter64
    |  |        +--r- dontFragment(7)       TruthValue
    |  |        +--r- maxAggregationTime(8) NanoSeconds
    |  +---- ipsecStatsGroup(2)
    |  |  +---- ipsecStatsTable(1)
    |  |     +---- ipsecStatsTableEntry(1) [ipsecSaIndex]
    |  |        +---- ipsecSaIndex(1)  Integer32
    |  |        +--r- txPackets(2)     Counter64
    |  |        +--r- txOctets(3)      Counter64
    |  |        +--r- txDropPackets(4) Counter64
    |  |        +--r- rxPackets(5)     Counter64
    |  |        +--r- rxOctets(6)      Counter64
    |  |        +--r- rxDropPackets(7) Counter64
    |  +---- iptfsInnerStatsGroup(3)
    |  |  +---- iptfsInnerStatsTable(1)
    |  |     +---- iptfsInnerStatsTableEntry(1) [iptfsInnerSaIndex]
    |  |        +---- iptfsInnerSaIndex(1)        Integer32
    |  |        +--r- txInnerPackets(2)           Counter64
    |  |        +--r- txInnerOctets(3)            Counter64
    |  |        +--r- rxInnerPackets(4)           Counter64
    |  |        +--r- rxInnerOctets(5)            Counter64
    |  |        +--r- rxIncompleteInnerPackets(6) Counter64
    |  +---- iptfsOuterStatsGroup(4)
    |     +---- iptfsOuterStatsTable(1)
    |        +---- iptfsOuterStatsTableEntry(1) [iptfsSaIndex]
    |           +---- iptfsSaIndex(1)      Integer32
    |           +--r- txExtraPadPackets(2) Counter64
    |           +--r- txExtraPadOctets(3)  Counter64
    |           +--r- txAllPadPackets(4)   Counter64
    |           +--r- txAllPadOctets(5)    Counter64
    |           +--r- rxExtraPadPackets(6) Counter64
    |           +--r- rxExtraPadOctets(7)  Counter64
    |           +--r- rxAllPadPackets(8)   Counter64
    |           +--r- rxAllPadOctets(9)    Counter64
    |           +--r- rxErroredPackets(10) Counter64
    |           +--r- rxMissedPackets(11)  Counter64
    +---- iptfsMIBConformance(2)
       +---- iptfsMIBConformances(1)
       |  +---- iptfsMIBCompliance(1)
       +---- iptfsMIBGroups(2)
          +---- iptfsMIBConfGroup(1)
          +---- ipsecStatsConfGroup(2)
          +---- iptfsInnerStatsConfGroup(3)
          +---- iptfsOuterStatsConfGroup(4)
```

Next Steps

- Ready for WG adoption?

Comments / Questions?

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- **IKEv2 Configuration for Encrypted DNS – Valery Smyslov**
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

IKEv2 Configuration for Encrypted DNS

draft-btw-add-ipsecme-ike-02

Mohamed Boucadair (Orange)
Tirumaleswar Reddy (McAfee, Inc.)
Dan Wing (Citrix Systems, Inc.)
Valery Smyslov (ELVIS-PLUS)

March 2021, IETF#110

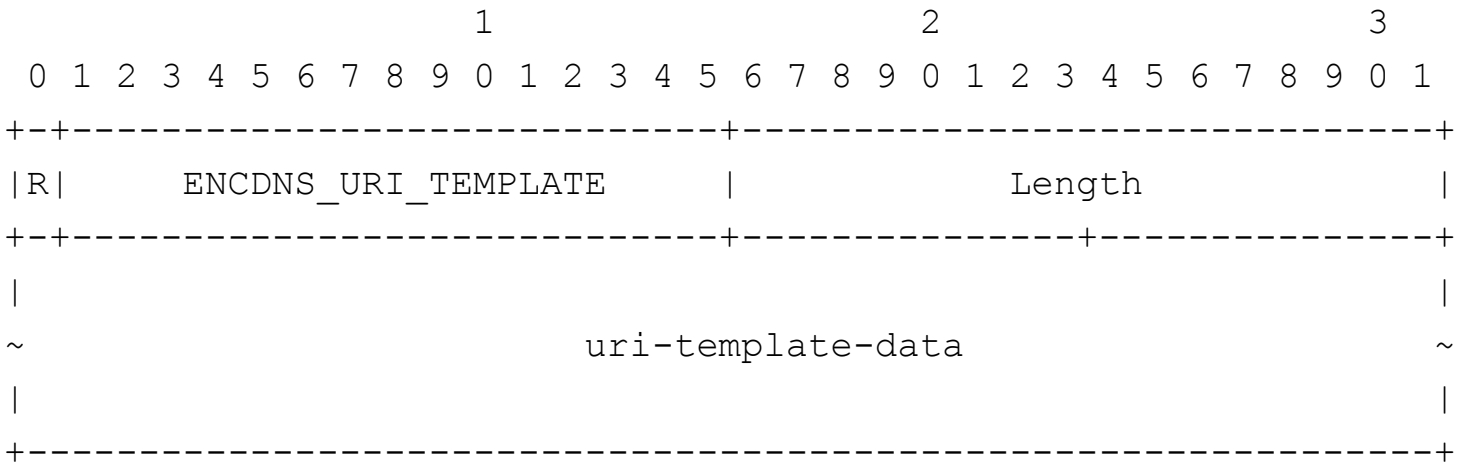
Status

- Presented at IETF#109
  - adoption of the draft was discussed
- Comments raised
  - wait for ADD WG to progress before adoption
    - ADD WG has progressed far enough to reconsider adoption of this draft
    - ADD WG adopted insecure discovery mechanisms: DHCP/RA (draft-ietf-add-dnr) and DNS (draft-ietf-add-ddr)
    - No conflict with the work in ADD WG
    - Commit to cross-review the document in ADD WG

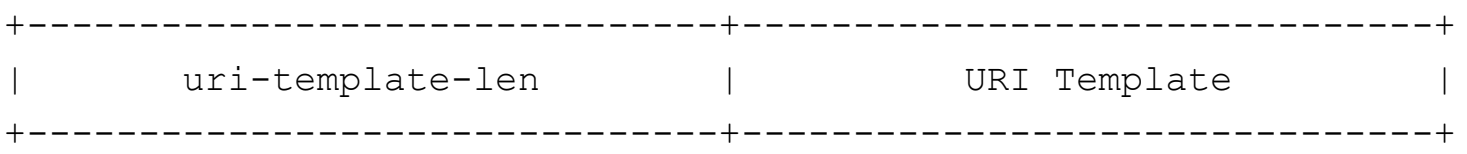
Changes from -01

- New Attribute `ENCDNS_URI_TEMPLATE` is added for conveying DOH URI Templates
- Clarification on authentication of a private encrypted DNS server
  - the encrypted DNS server hosted by VPN provider can get a domain-validate certificate from a public CA but it is only accessible to clients connected to the VPN

ENCDNS_URI_TEMPLATE Attribute Format



Each instance of the `uri-template-data` is formatted as follows:



DoH Specifics

- DoH servers may support more than one URI Template
- If DoH is requested, the initiator includes an empty `ENCDNS_URI_TEMPLATE` attribute (in addition to `ENCDNS_IP*_DOH` attributes) in the `CFG_REQUEST`
- If the responder includes `ENCDNS_IP4_DOH` or `ENCDNS_IP6_DOH` in the response, it MUST also include `ENCDNS_URI_TEMPLATE` carrying one or more URI Templates
- Avoids the need to rely on insecure discovery mechanisms (DHCP/RA), and on Do53 (which requires additional RTT and is not always secure, since using DNSSEC is not mandatory)
  - draft-schwartz-svcb-dns-01, Section 9.1

Next Steps

- Comments?
- Questions?
- Consider WG adoption

Thank you

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- **New payload format for IKEv2 – Valery Smyslov**
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

New IKEv2 Payload Format

Valery Smyslov
svan@elvis.ru

IETF 110

Existing Format Redundancy

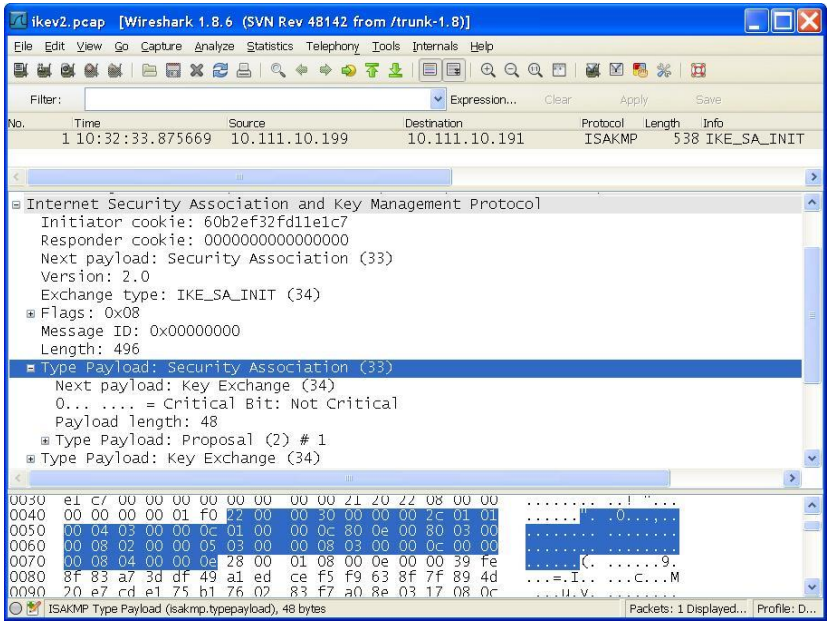
Many payloads contain substantial redundancy

- Payload Length field occupies 2 bytes, while most payloads are shorter
- most parameters occupy 2 bytes, while less than 256 values are defined
- zero-filled RESERVED fields

Example: SA Payload on the right contains one Proposal with four Transforms:

- `ENCR_AES_CBC` (128 bits)
- `PRF_HMAC_SHA2_256`
- `AUTH_HMAC_SHA2_256_128`
- `2048-bit MODP Group`

Payload size is **48** bytes, among which **24** bytes are zeroes.



Existing Format Limitations

- Payload Length field occupies 2 bytes, so payload size is limited to 64 Kbytes
  - no problem with Message size, which is limited to 4 Gbytes

Making Payloads Smaller

- Would decrease power and network bandwidth consumption (important for IoT devices)
- Would decrease chances of IP fragmentation in the IKE_SA_INIT and IKE fragmentation in the rest exchanges

Lifting 64 Kbytes Size Limit

- Would allow using PQ algorithms with long public keys and signatures
  - draft-tjhai-ikev2-beyond-64k-limit
- Would allow transferring large chunks of data (e.g. in CP payload)

New Format Requirements

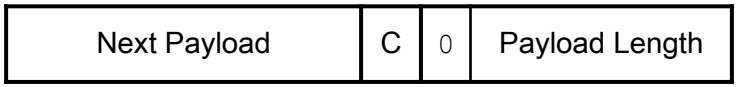
- Must be suitable for both small and large payloads
- Must be applicable to any payload type, including not yet defined ones
  - some payloads may have special format if it is justified
- The encoder/parser must remain simple and consume low resources

New Format Proposal

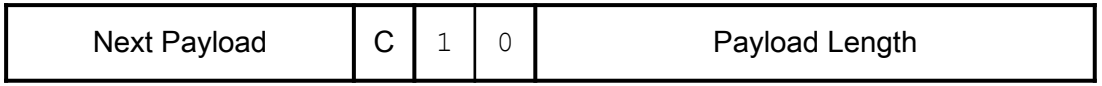
- Three possible formats for new Generic Payload Header
  - for small payloads (up to 64 bytes)
  - for medium size payloads (up to 8 Kbytes)
  - for large payloads (up to 512 Mbytes)
- No RESERVED fields
- Revise existing payloads headers to reduce their size
  - remove unnecessary fields
- Special Format for some payloads (SA, empty Status Notify)

New Generic Payload Header

1. Small payloads (2 bytes, 6 bits for Payload Length)



2. Medium size payloads (3 bytes, 13 bits for Payload Length)



3. Large payloads (5 bytes, 29 bits for Payload Length)

Revised Existing Payload Headers

The following payload headers can be revised:

- Key Exchange, Identification, Authentication, Configuration
  - remove `RESERVED` field
- Notify
  - remove `SPI Size` field (can be deducted from Protocol ID)
- Delete
  - remove `SPI Size` field (can be deducted from Protocol ID)
  - remove `Num of SPIs` field (can be deducted from Payload Length)
- Traffic Selector
  - remove `RESERVED` field
  - remove `Number of TSs` field (can be deducted from Payload Length)

Special Format

Special format (*) for:

- SA Payload
  - SA Payload grows quickly as more and more new transforms are defined and offered by initiators
- Notify Payload with some Status Type Notification and no data
  - Exchange of such payloads is a common way to negotiate support for various protocol extensions, so initial IKEv2 messages grow up as more and more extensions are defined

Both payloads contain a lot of redundancy and can be effectively compacted.

(*) Inspired by draft-smyslov-ipsecme-ikev2-compact

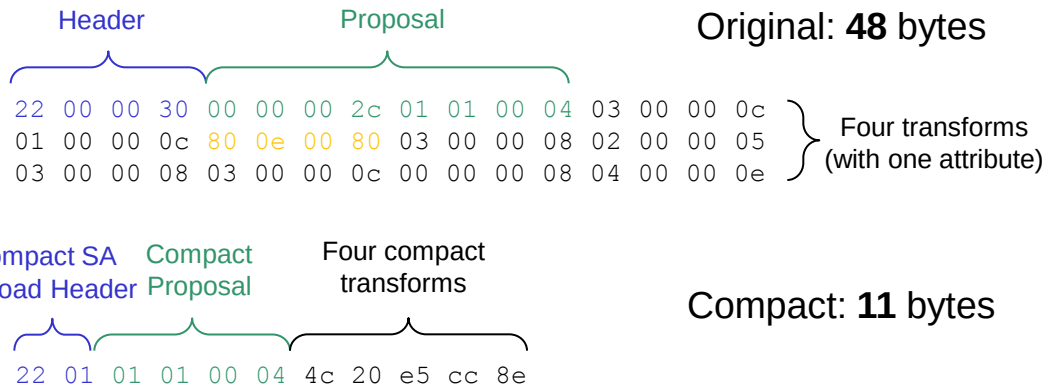
SA Payload

Outline:

- Remove all `RESERVED` fields
- Remove `Length` fields in substructures (where they are unnecessary)
- Encode all currently defined transforms w/o attributes using one octet (both `Transform Type` and `Transform ID`)
- Encode currently defined `Encryption` transforms having `Key Length` attribute using two octets
- Leave possibility to encode arbitrary (even not yet defined) `Transform Type` and `Transform ID`, as with regular format

Example: SA Payload with one Proposal and four Transforms:

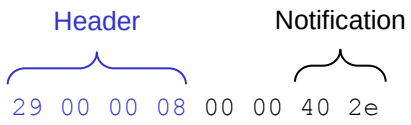
- `ENCR_AES_CBC` (128 bits)
- `PRF_HMAC_SHA2_256`
- `AUTH_HMAC_SHA2_256_128`
- `2048-bit MODP Group`



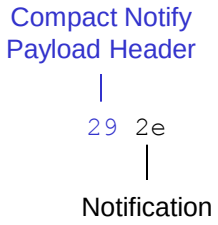
Notify Payload

Outline: encode notification in one octet (limited to first 256 status notifications) and omit all other fields from Notify Payload

Example: Notify Payload with `IKEV2_FRAGMENTATION_SUPPORTED` notification.



Original: **8** bytes



Compact: **2** bytes

Negotiation

If new format is used from the very beginning then the following options exist:

- New major IKE version (v3)
  - old responders would return `INVALID_MAJOR_VERSION`
- New type of initial exchange (e.g. `ALT_IKE_SA_INIT`)
  - old responders would return `INVALID_SYNTAX`
- New critical payload in the `IKE_SA_INIT`, followed by payloads in new format
  - old responders would return `UNSUPPORTED_CRITICAL_PAYLOAD`

Discussion

- We don’t need to assign new payload types except for special format payloads (SA and empty status Notify), do we? What about revised payloads?
- Transport issues for transferring large payloads are out of scope
  - IKE over TCP combined with IKE fragmentation (to solve limitation on 64 Kbytes on a single IKE message over TCP) can be used
  - do we need mixed mode – IKE over TCP combined with plain ESP or ESP over UDP?
- Certificates consume a lot of space, can be compressed
  - RFC 8879 is an example of certificate compression
  - in some use cases draft-mattsson-cose-cbor-cert-compress can be used

Thanks

- Comments? Questions?
- Any interest in this work?

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- **IKEv1 graveyard – Paul Wouters**
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

redhat.

draft-pwouters-ikev1-ipsec-graveyard

IETF-110, IPsecME,
March, 2021

Paul Wouters, RHEL Security

draft-pwouters-ikev1-ipsec-graveyard

- https://tools.ietf.org/html/draft-pwouters-ikev1-ipsec-graveyard-06
- No changes from -05 version other than version/date
- Tells implementors to stop implementing IKEv1
  - (implies users should not use it anymore)
- Instructs IANA to create Status column for “deprecated”
- Formally closes IKEv1 registries and reminds implementors IKEv1 is ”Historic”

Adoption call

- IETF-108 lists action item for chairs ”Perhaps go to WGLC”
  - But document was not yet adopted ?
- IETF-109 pointed this out again with request for adoption
- IETF-110 ← You are here

- If we do not adopt this document, I think we would still need another document to add a Status column to the IANA algorithm Registries. Let’s just adopt and WGLC this one.

Presentations

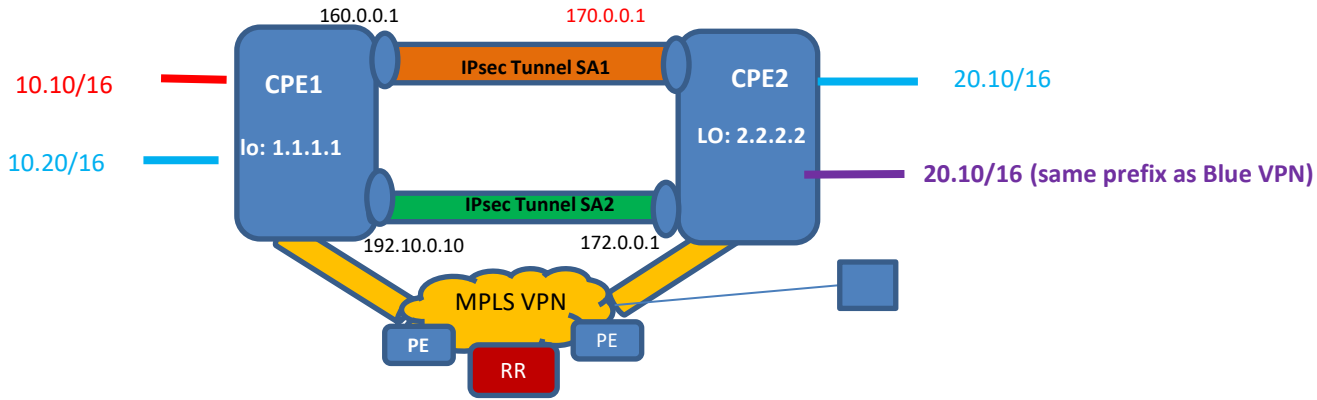
- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- **BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar**
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

SDWAN Edge Discovery

https://datatracker.ietf.org/doc/draft-dunbar-idr-sdwan-edge-discovery/

Linda Dunbar
Sue Hares
Robert Raszuk
Kausik Majumdar
March 2021

Comparing with traditional IPsec configuration



BGP Controlled SDWAN: VRF, RT, RD

- BGP Route Target and Route Distinguisher enables:
  - Simpler Traffic Selector: Import/export Route Target
    - To achieve permission of the local and remote prefixes
  - Same client prefixes reuse: Route Distinguisher
- SDWAN policies dictate if a client network (VRF) can bind with IPsec SA.

➢ **With RR to control the policy, it scale much better for multi-node VPNs**

Traditional IPsec Configuration

- ➢ IPsec IKE to authenticate with each other
- ➢ Establish IPsec SA
  - Local key configuration
  - Remote Peer address (192.10.0.10<->172.0.01)
  - IKEv2 Proposal directly sent to peer
    - Encryption method, Integrity sha512
  - Transform set
- ➢ Attached client prefixes discovery
  - By running routing protocol within each IPsec SA
- ➢ Access List or Traffic Selector
    - Permit Local-IP1, Remote-IP2

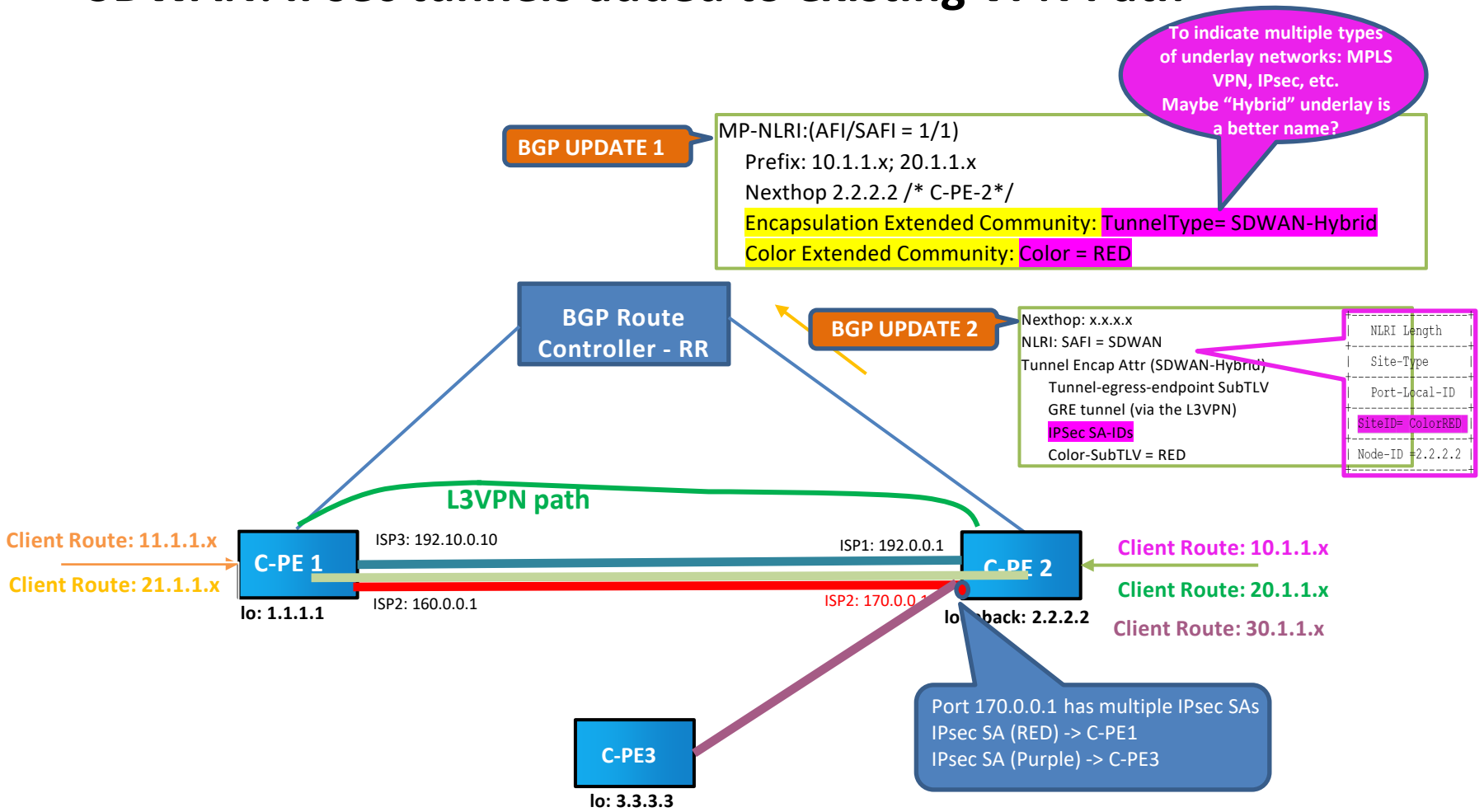
Simplification of BGP Controlled SDWAN's IPsec tunnel configuration

Traditional IPsec Configuration	BGP Controlled SDWAN's IPsec Configuration
IPsec IKE to authenticate peers	Not needed, as the SDWAN controller has authority to authenticate edges and peers
Remote Peer address configuration	Remote Peer policy is controlled by SDWAN Controller (RR)
IKEv2 Proposal directly sent to peer Encryption method, Integrity sha512	The IKEv2 proposals can be sent directly to Peer, or incorporated with BGP UPDATE
Transform set	Transform set either by BGP UPDATE or by IKEv2 message exchange
Client Prefix discovery By running separate instances of routing protocol within each IPsec SA	BGP UPDATE: Announce the client route Reachability for all parallel. No need to run multiple routing protocols in each IPsec tunnel.
Access List or Traffic Selector) Permit Local-IP1, Remote-IP2	More scalable multi-node Traffic Selector: BGP Route Target :Import/export Route Target

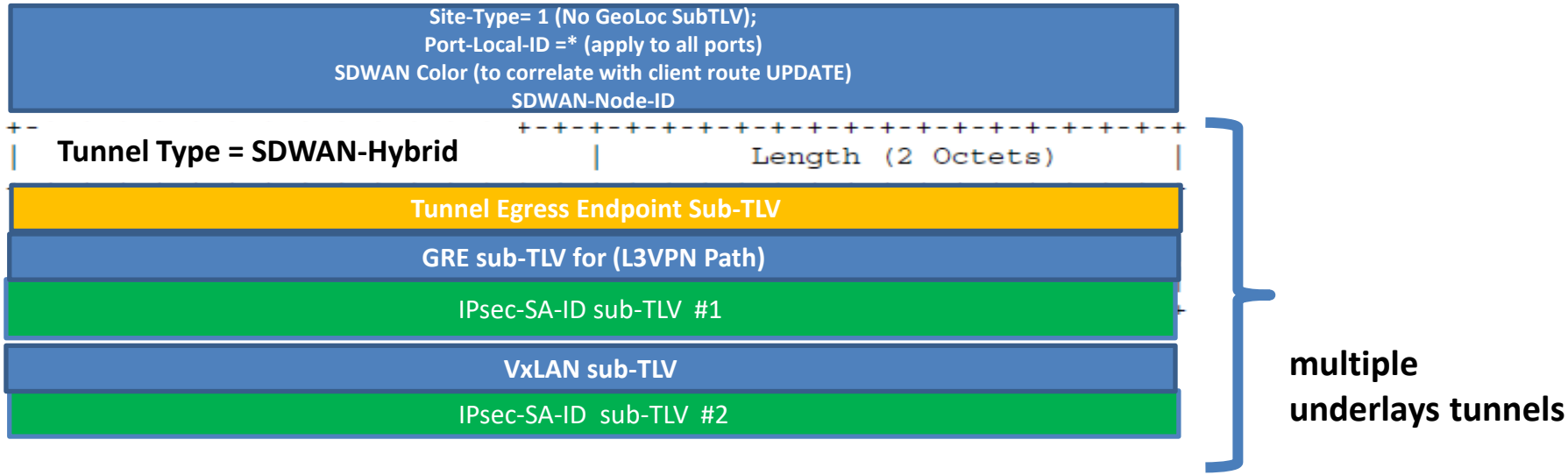
Optimization by BGP controlled SDWAN

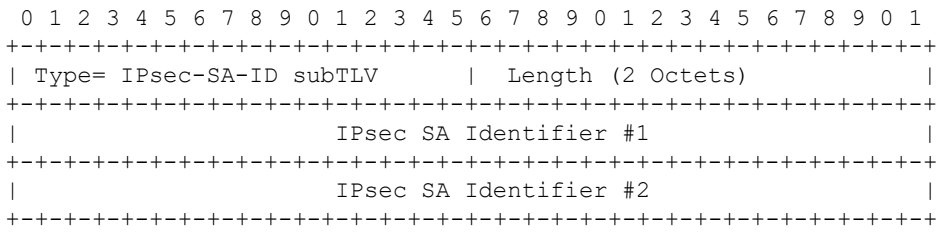
Same as traditional IPsec exchange, or can be exchanged via BGP RR

SDWAN: IPsec tunnels added to existing VPN Path

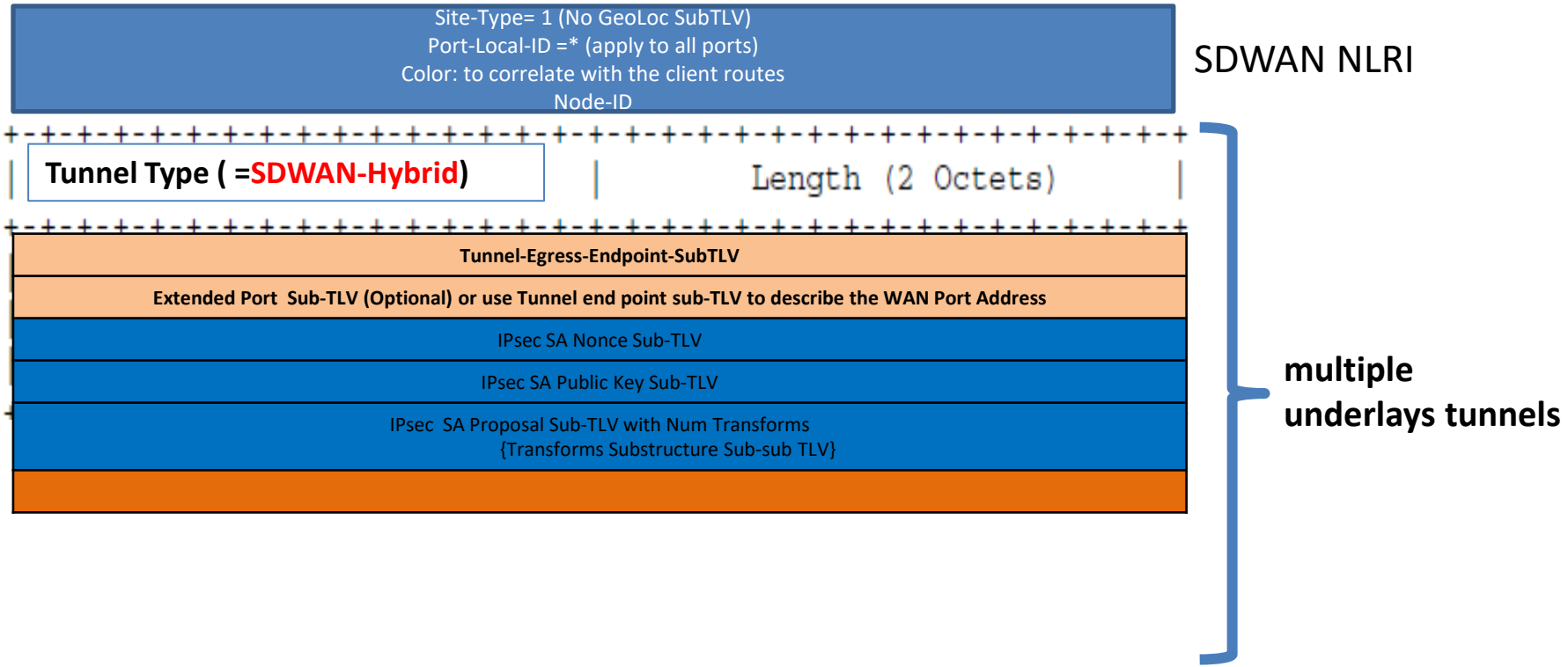


Hybrid Tunnels: with Pre-configured IPsec SA IDs





Hybrid Tunnels: with detailed IPsec SA sub-TLVs




We want
your feedback!

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- **X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert**
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- IKEv2 Multi-SA Performance – Paul Wouters

How to make use of the X.509 extensions for alternate signature schemes in IKEv2?

Leonie Bruckert (secunet AG)

Heike Hagemeier (BSI)

@IETF 110

ISO/IEC 9594-8 (2020) defines three extensions

`subjectAltPublicKeyInfo`	alternative public key information
`altSignatureAlgorithm`	alternative digital signature algorithm
`AltSignatureValue`	alternative signature

… to be used *instead* of the native fields (if present)

→ Allows for a smooth migration of a PKI to a new signature algorithm

→ Does *not* intend hybrid use of algorithms

Question: Do we need a hybrid solution for authentication with regard to PQC?

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- **IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters**
- IKEv2 Multi-SA Performance – Paul Wouters

redhat.

draft-ietf—ipsecme-ikev2-sa-ts-payloads-opts

IETF-110, IPsecME,
March, 2021

Paul Wouters, RHEL Security

History

- https://tools.ietf.org/html/draft-kampati-ipsecme-ikev2-sa-ts-payloads-opt-02
- Not my draft, work done by Kampti & Bharath of Huawei
  - Reduce bytes sent during IKE/IPsec rekey
- Libreswan is very interested in implementing this
- draft is expired but will be resubmitted this week
- IPR situation was unclear but now resolved
  - Standard IPR claim

Change from regular rekeying

- Negotiate support via Notify in IKE_AUTH
  - N(MINIMAL_REKEY_SUPPORTED)
- During REKEY
  - Omit SA and TS payloads
  - Send N(SA_UNCHANGED) with new SPI
  - Send Nonce / KE as normal if PFS
  - Calculate key from KDF as normal
  - Assume all crypto algos / traffic selector remain identical

Changes needed (according to Paul)

- Remove text about changing cryptographic algorithms
  - This is not allowed in IKEv2
- Remove text for rekey when ACL changed
  - Only way is to delete all SA's and start from scratch
- N(SA_TS_UNCHANGED) must also send old SPI for Child SA
  - This is normally sent inside SA payload
- Clarify PFS

Presentations

- Group Key Management using IKEv2 – Valery Smyslov
- IPTFS Base Draft WGLC Changes – Christian Hopps
- Management (Yang adoption and update, new SNMP draft) – Christian Hopps
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- New payload format for IKEv2 – Valery Smyslov
- IKEv1 graveyard – Paul Wouters
- BGP UPDATE for SDWAN Edge Discovery – Linda Dunbar
- X.509 extensions and alternative signature schemes in IKEv2 – Leonie Bruckert
- IKEv2 Optional SA & TS Payloads in Child Exchnage – Paul Wouters
- **IKEv2 Multi-SA Performance – Paul Wouters**

redhat.

IKEv2 Multi SA

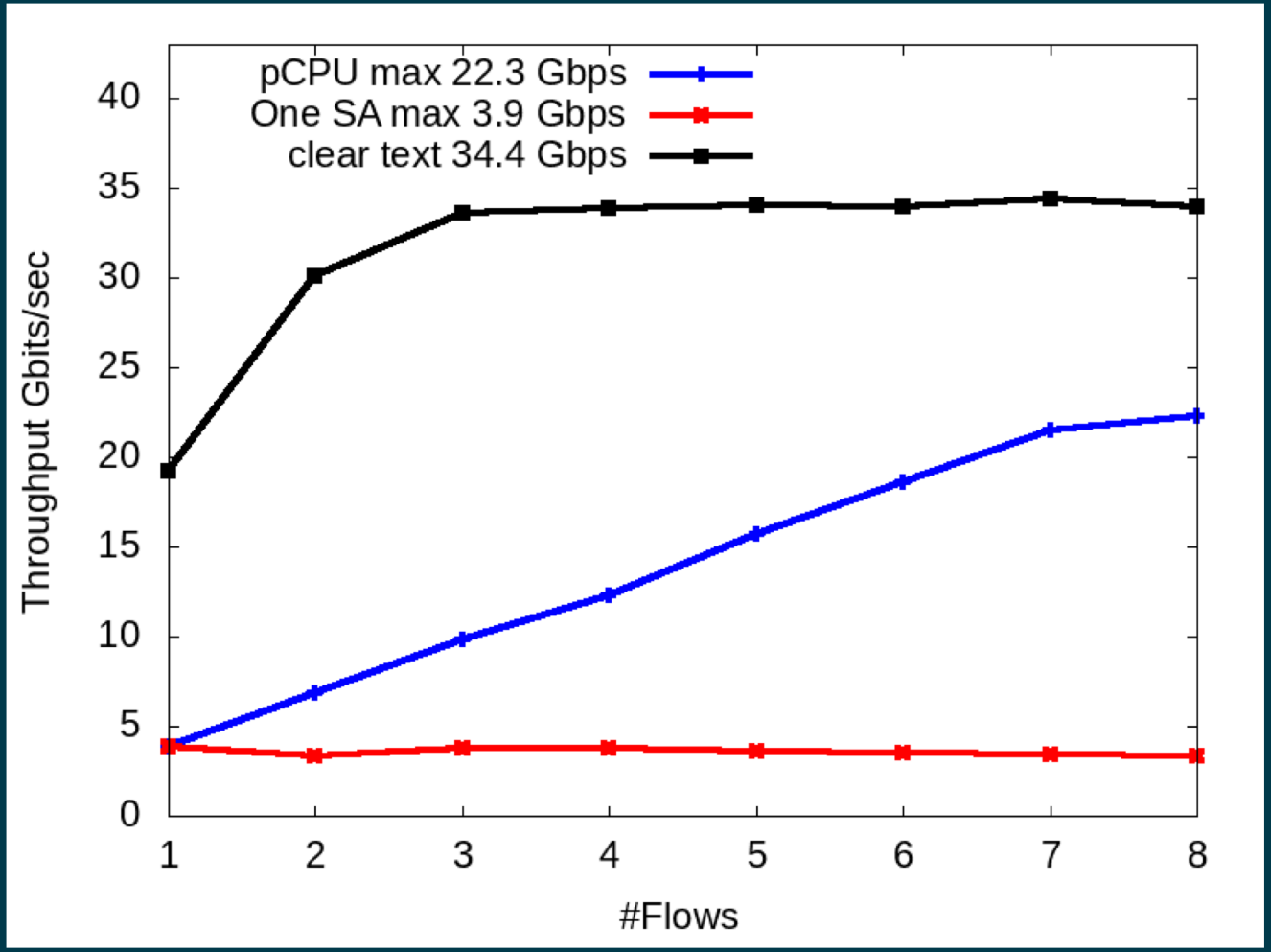
IETF-110, IPsecME,
March, 2021

Paul Wouters, RHEL Security

Resolving limitations

- Give implementation advise on how to handle multiple IPsec SA's with identical Traffic Selectors
- https://tools.ietf.org/html/draft-pwouters-multi-sa-performance-01
- Install multiple IPsec SA - one per CPU
- Two new NOTIFY payloads for IPsec SA
  - NUM_QUEUES(max)
  - QUEUE_INFO(opaque)

Benchmarks


pCPU max 22.3 Gbps
One SA max 3.9 Gbps
clear text 34.4 Gbps
Throughput Gbits/sec
#Flows
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
7
8

Changed in -01

- NUM_QUEUES(minimum) – takes one argument instead of two
- Signal CPU identifier send via QUEUE_INFO
  - Does not need to be the actual hardware CPUID
- Readability improvements

RSS for ESP is rarely supported

- RSS usually only supports UDP/TCP port hashing selector
- We want to use UDP encapsulation so RSS for UDP can be used
- But multiple Child SAs would all use the same UDP port
  - We would like Child SAs on different source ports
  - Without affecting IKEv2 channel
  - OS does not always make that easy
  - Can’t negotiate source port, because of NAT
  - How to do NAT port updates ?

Suitable as WG item ?

- Solves a real world problem (performance)
- We have running code
- We have measured positive results (with specific hardware)
- Solving the UDP source port issue would make this work on all existing (virtual and real) network cards
- Please adopt ☺

Open Discussion

- Other points of interest?