Introduction to IPsec

Paul Wouters, Aiven

IETF 113, SAAG
March 2022

Who heckles the hecklers?

RFC 4303 Section 1.1 states:

> The spelling "IPsec" is preferred and used throughout this and all related IPsec standards. All other capitalizations of IPsec (e.g., IPSEC, IPSec, ipsec) are deprecated.

Note: All RFCs and drafts in this presentation are clickable links

RFC 6071: IPsec and IKE Document Roadmap

Outdated document from 2011 listing “all RFCs”

Still provides a nice overview in 63 pages

NIST SP 800 77 Rev1: “Guide to IPsec VPNs”

- Recent document from 2020
- Authors include various IETFers [*]
- Introducion to IPsec
- Various imaginary deployments
- NIST / FIPS recommendations
- Configurations for common IPsec systems
- A steal at only 149 pages
- It’s fun and it’s free

[*] including me

IPsec is more than a double edged sword

“IPsec” consists of two separate protocols

- Control channel (userland)
  - Internet Key Exchange(“IKE”)
- Data plane (kernel)
  - Encapsulated Security Payload (“ESP”)
  - ~~Authenticated Header (“AH”)~~
- Control <-> Data plane communication
  - RFC2367: PFKEYv2
  - RFC3549: Linux NetLink

Two negotiations need to happen for “IPsec”

1) The IKE protocol establishes a Security Association state with the peer (**IKE SA**)
2) The IKE SA negotiates one or more IPsec Associations (**IPsec SA**)

An IKE packet request and reply is called an **Exchange**

The IKE protocol itself is encrypted. This encryption is indepedent of the IPsec (packet) encryption.

A little bit of (confusing) terminology

The IKE SA used to be called ISAKMP SA
The IKE SA is now formally called the Parent SA
The IPsec SA is now formally called the Child SA

I personally use the terms IKE SA and IPsec SA

What does IPsec provide?

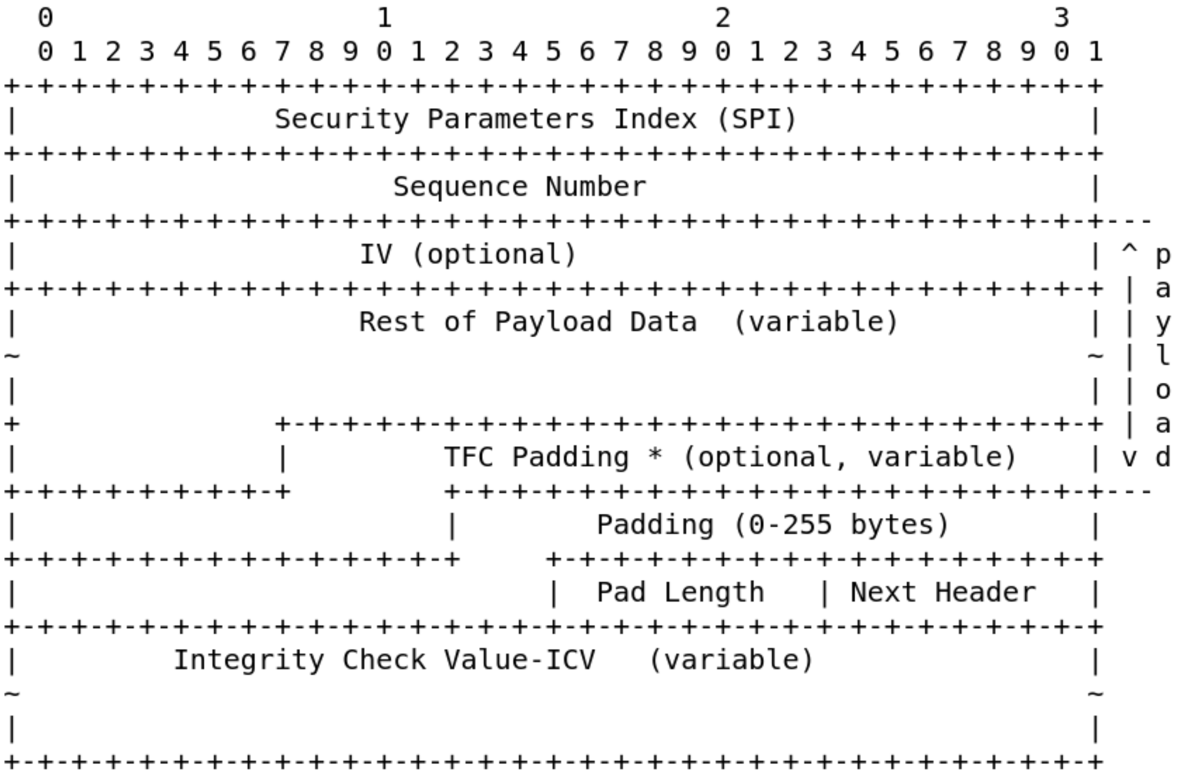
- Host to host IP packet encryption
- Network to network encrypted interconnect
- Remote Access VPN
- ~~Encrypt the entire internet by default~~

RFC4303: Encapsulated Security Payload (“ESP”)

protocol 50 (“not port 50”)

- Fairly standard “encrypted data chunk”
- Identifying nonces for streams (“SPIs”)
- Algorithm agility (eg AEAD support now)
- Replay protection via Sequence Numbers
- Traffic Selectors (IP src,dst, proto, port / type)
- Padding, obfuscation packets
- Compression (it’s complicated, just don’t do it)
- Support high speed (Extended Sequence Numbers)

RFC4303: Encapsulated Security Payload (“ESP”)



```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Security Parameters Index (SPI)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---
|                    IV (optional)                              | ^ p
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | a
|                    Rest of Payload Data  (variable)           | | y
~                                                               ~ | l
|                                                               | | o
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | a
|               |         TFC Padding * (optional, variable)    | v d
+-+-+-+-+-+-+-+-+         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---
|                         |        Padding (0-255 bytes)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |  Pad Length   | Next Header   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Integrity Check Value-ICV   (variable)                |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. Substructure of Payload Data

ESP: Transport Mode

- Can only be used for host-to-host encryption
- Authenticates part of IP packet header
- Encrypts IP packet body
- Does not encrypt source / dest IP address
- Least amount of overhead, biggest MTU
- Generally fastest lookups in kernel
- Completely fails with NAT
  - So ofcourse it was still used with NAT
  - L2TP VPN is IPsec transport mode using:
    UDP(ESP(L2TP(PPP((IP))))) [over PPPOE]

ESP: Tunnel Mode

- Can be used for network to network
- Encrypts entire IP packet in a new packet body
- More overhead, reduces available MTU
- Requires src/dst policies
- Generally slower lookups in kernel
- Using RFC3948 ESPinUDP, works well with NATs
  - Stuff the packet into a UDP 4500 packet

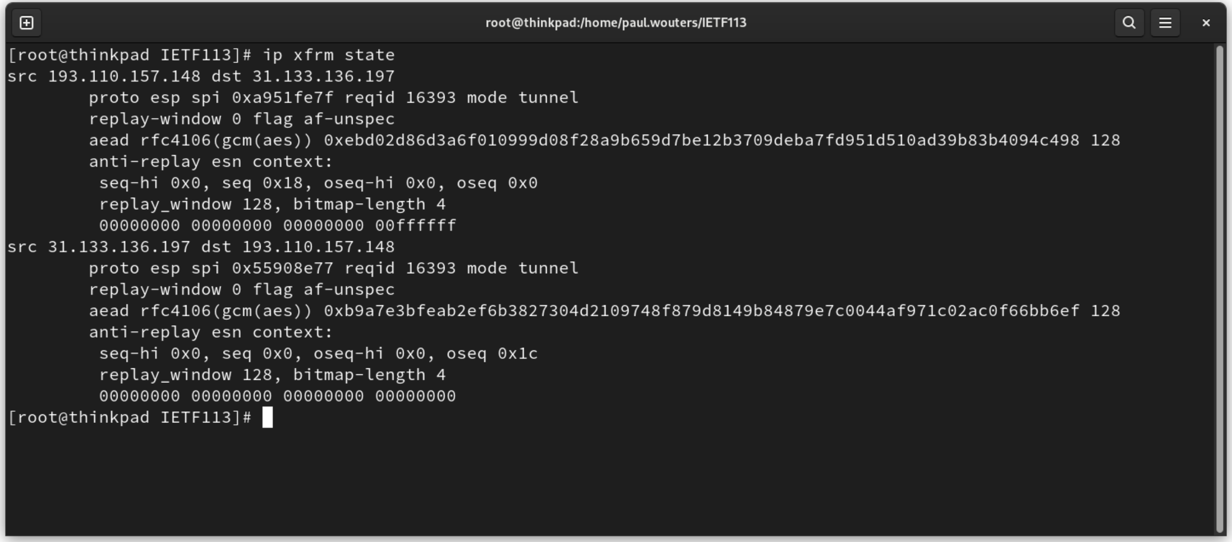
Encapsulating the ESP packets

- RFC3948 ESPinUDP for NAT-Traversal
- RFC3947 Negotiate NAT-Traversal with IKE
  - Uses UDP port 4500, easilly blocked
  - Also, often all non-DNS UDP is blocked
- Non-standard TCP encap over port 10000
- RFC8229: TCP encap for IKE and IPsec
  - provisioned port (eg could use 443)
  - ‘Happens to have’ a prefix “IKETCP”
    - One could demux HTTPS / IPsec

Managing and using ESP: SPD and SAD

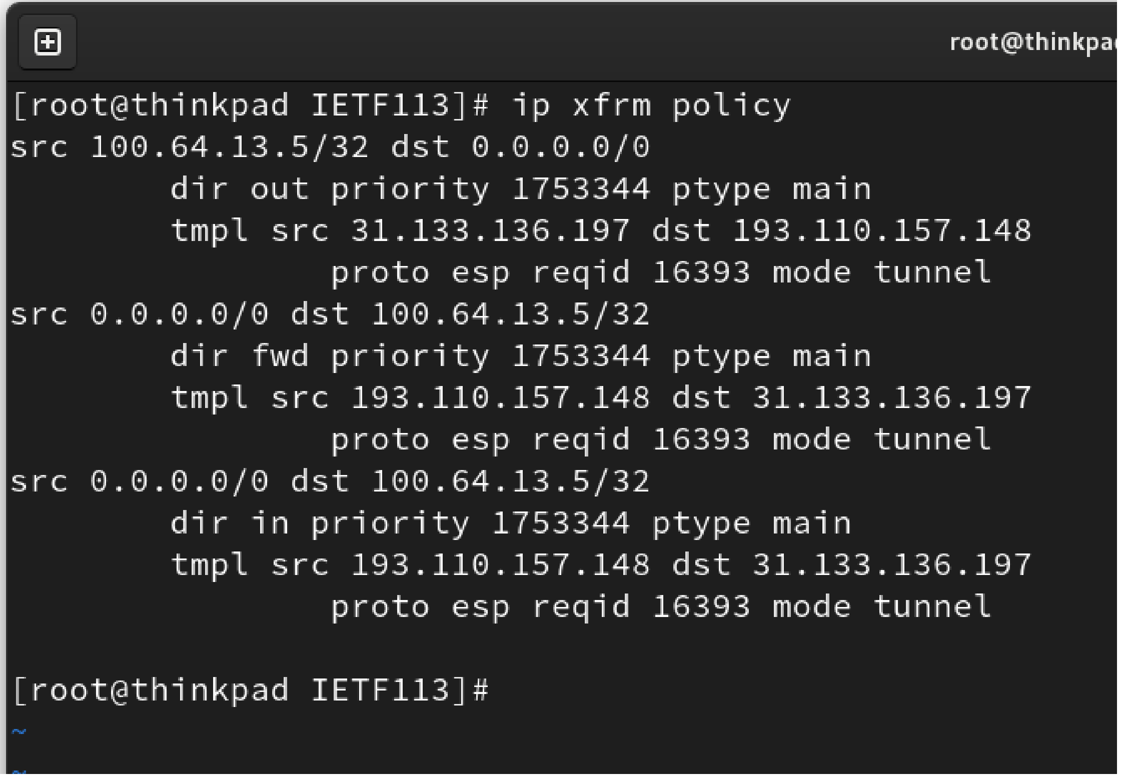
- Security Association Database (“SAD”)
  - Where to send/receive packets to/from
  - list of peers (IP, transport, crypto state, etc)
  - link to SPD entry (for tunnel mode)
  - state (crypto keys, counters, seqnum, etc)
- Security Policy Database (“SPD”)
  - Traffic Selector (src, dst, proto, port, etc)
  - Which SAD entry to use
- SAD/SPD unidirectional (inbound / outbound)

IPsec example SAD:



```
[root@thinkpad IETF113]# ip xfrm state
src 193.110.157.148 dst 31.133.136.197
        proto esp spi 0xa951fe7f reqid 16393 mode tunnel
        replay-window 0 flag af-unspec
        aead rfc4106(gcm(aes)) 0xebd02d86d3a6f010999d08f28a9b659d7be12b3709deba7fd951d510ad39b83b4094c498 128
        anti-replay esn context:
         seq-hi 0x0, seq 0x18, oseq-hi 0x0, oseq 0x0
         replay_window 128, bitmap-length 4
         00000000 00000000 00000000 00ffffff
src 31.133.136.197 dst 193.110.157.148
        proto esp spi 0x55908e77 reqid 16393 mode tunnel
        replay-window 0 flag af-unspec
        aead rfc4106(gcm(aes)) 0xb9a7e3bfeab2ef6b3827304d2109748f879d8149b84879e7c0044af971c02ac0f66bb6ef 128
        anti-replay esn context:
         seq-hi 0x0, seq 0x0, oseq-hi 0x0, oseq 0x1c
         replay_window 128, bitmap-length 4
         00000000 00000000 00000000 00000000
[root@thinkpad IETF113]# 
```

IPsec example SPD:



```
[root@thinkpad IETF113]# ip xfrm policy
src 100.64.13.5/32 dst 0.0.0.0/0
        dir out priority 1753344 ptype main
        tmpl src 31.133.136.197 dst 193.110.157.148
                proto esp reqid 16393 mode tunnel
src 0.0.0.0/0 dst 100.64.13.5/32
        dir fwd priority 1753344 ptype main
        tmpl src 193.110.157.148 dst 31.133.136.197
                proto esp reqid 16393 mode tunnel
src 0.0.0.0/0 dst 100.64.13.5/32
        dir in priority 1753344 ptype main
        tmpl src 193.110.157.148 dst 31.133.136.197
                proto esp reqid 16393 mode tunnel

[root@thinkpad IETF113]#
~
```

Internet Key Exchange (“IKE”)

Negotiate all the IKE and IPsec parameters

1) Negotiate Ephemeral DiffieHellman
2) Perform peer authentication / authorization
3) Negotiate one or more IPsec connections
4) Keep it all alive
5) Rekey both IKE and IPsec key materials - provide PFS

RFC2409 IKEv1 published in1998

(basically “historic”, but still secure!)

(simplified to avoid talking about legacy stuff)

1) Negotiate Ephemeral DiffieHellman
2) Perform peer authentication / authorization
3) Negotiate one or more IPsec connections
4) Keep it all alive (well, IKE could terminate!)
5) Rekey both IKE and IPsec key materials
6) ...
7) profit !

IKEv1: Many extensions

- NAT Traversal (RFC3947 / RFC3948)
- Dead Peer Detection (RFC 3706)
- New algorithms (remember its from 1998)
- XAUTH (draft-ietf-ipsec-isakmp-xauth)
- ModeCFG (draft-dukes-ike-mode-cfg)
- X.509 PKI (RFC4945)

- Many vendor extensions

IKEv1 deployment issues

- See ietf-ipsecme-ikev1-algo-to-historic
- Both sides retransmit
- Amplification attacks, DoS attacks
- Too made modes (Main, Aggressive, Revised, Hybrid)
- Too many round trips to establish
- Each address range needs own IPsec SA
- Authentication failure causes decrypt failure
- Missing support for Mobility
- Not all fields in IKE packet integrity protected

IKEv2 improvements

- Tie the IKE SA to all its IPsec SAs
- Only Initiator retransmits
- anti-DDoS cookies (and puzzles)
- Reduce RTT by combining IKE /1st IPsec SA
- EAP support (with many RTTs)
- Combine many Traffic Selectors into one IPsec SA
- New fancy features

IKEv2: RFC7296 (via RFC4306, RFC4718, RFC5996)

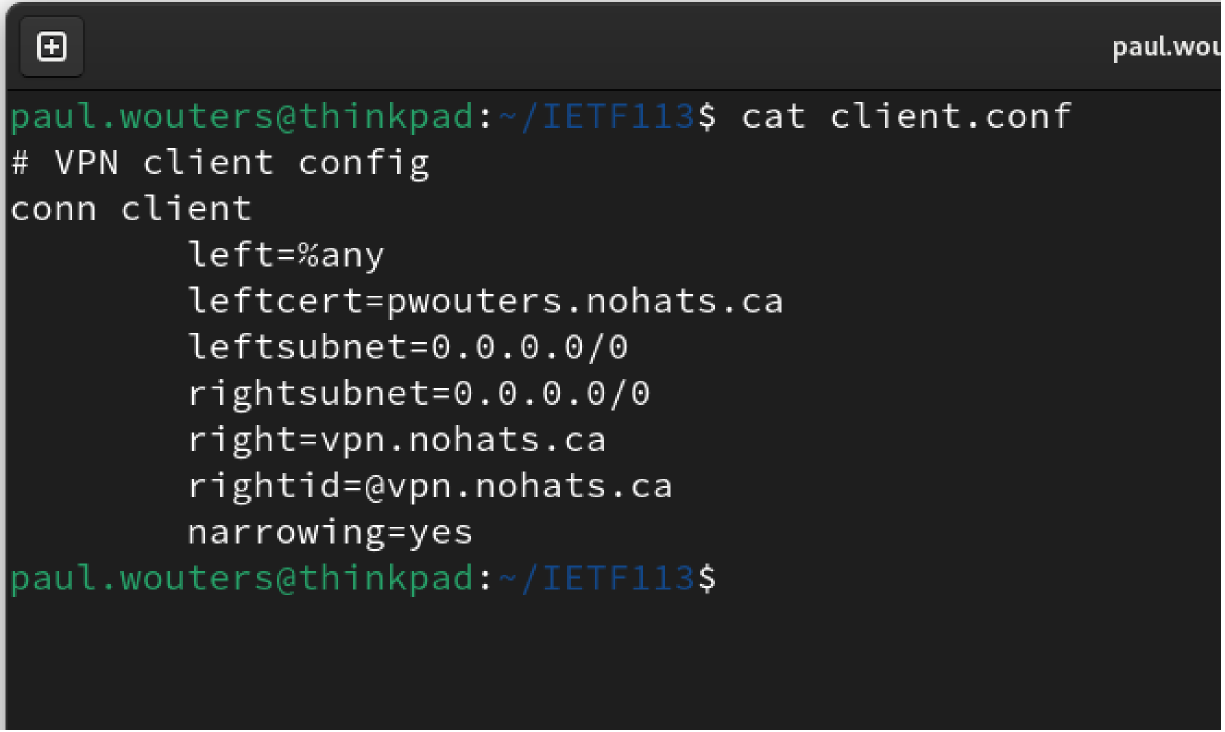
- No more IKE Modes (well, kinda sorta)

1) IKE_SA_INIT Exchange for ephemeral DH
2) IKE_AUTH Exchange for auth AND 1st IPsec SA
3) CREATE_CHILD_SA Exchange for more IPsec SA’s and IKE / IPsec rekeying
4) Informational Exchange for various things
    1) Dead Peer Detection
    2) Mobility Updates
    3) Deletes

Some interesting IKEv2 Extensions

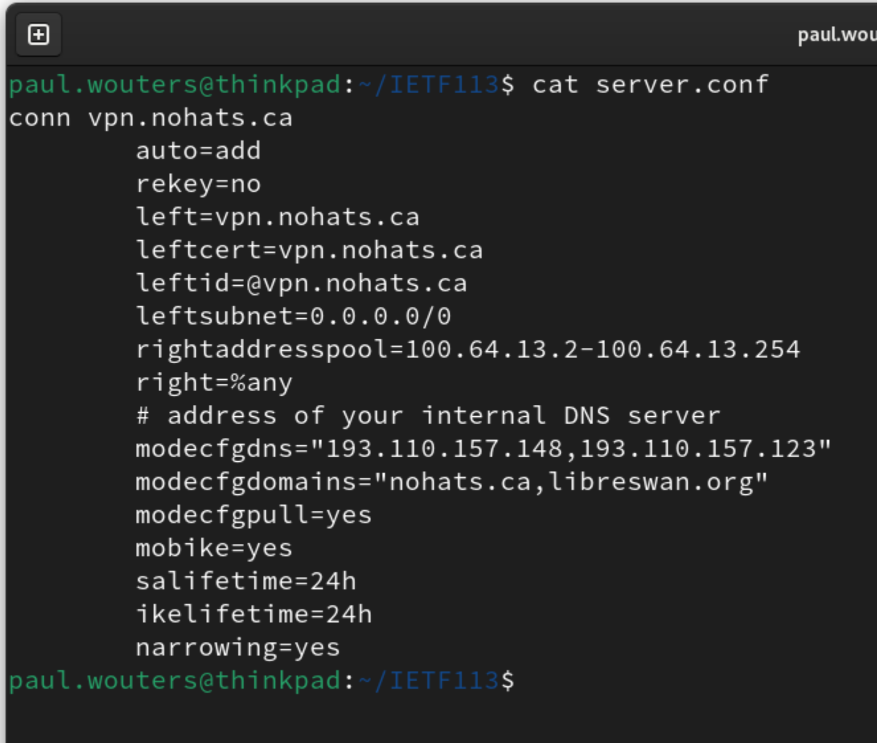
- RFC4555 MOBIKE (mobility and multihome)
- RFC5723 Session Resumption
- Algorithm and (EAP) authentication updates
- RFC5685 REDIRECT (client) support
- RFC7283 IKE Fragmentation support
- draft-ietf-ipsecme-ikev2-intermediate to support large post-quantum key blobs
- RFC8784 Postquantum PreShared Keys
- draft-ietf-ipsecme-ikev2-multiple-ke for Hybrid/Composite Key Exchange

IPsec example IKE client config:



```
paul.wouters@thinkpad:~/IETF113$ cat client.conf
# VPN client config
conn client
        left=%any
        leftcert=pwouters.nohats.ca
        leftsubnet=0.0.0.0/0
        rightsubnet=0.0.0.0/0
        right=vpn.nohats.ca
        rightid=@vpn.nohats.ca
        narrowing=yes
paul.wouters@thinkpad:~/IETF113$
```

IPsec example IKE server config:



```
paul.wouters@thinkpad:~/IETF113$ cat server.conf
conn vpn.nohats.ca
        auto=add
        rekey=no
        left=vpn.nohats.ca
        leftcert=vpn.nohats.ca
        leftid=@vpn.nohats.ca
        leftsubnet=0.0.0.0/0
        rightaddresspool=100.64.13.2-100.64.13.254
        right=%any
        # address of your internal DNS server
        modecfgdns="193.110.157.148,193.110.157.123"
        modecfgdomains="nohats.ca,libreswan.org"
        modecfgpull=yes
        mobike=yes
        salifetime=24h
        ikelifetime=24h
        narrowing=yes
paul.wouters@thinkpad:~/IETF113$
```