IPv4 Address Sharing: Problem, Solutions, and Test results

draft-boucadair-intarea-nat-reveal-analysis-04

draft-abdo-hostid-tcpopt-implementation-01

BEHAVE WG

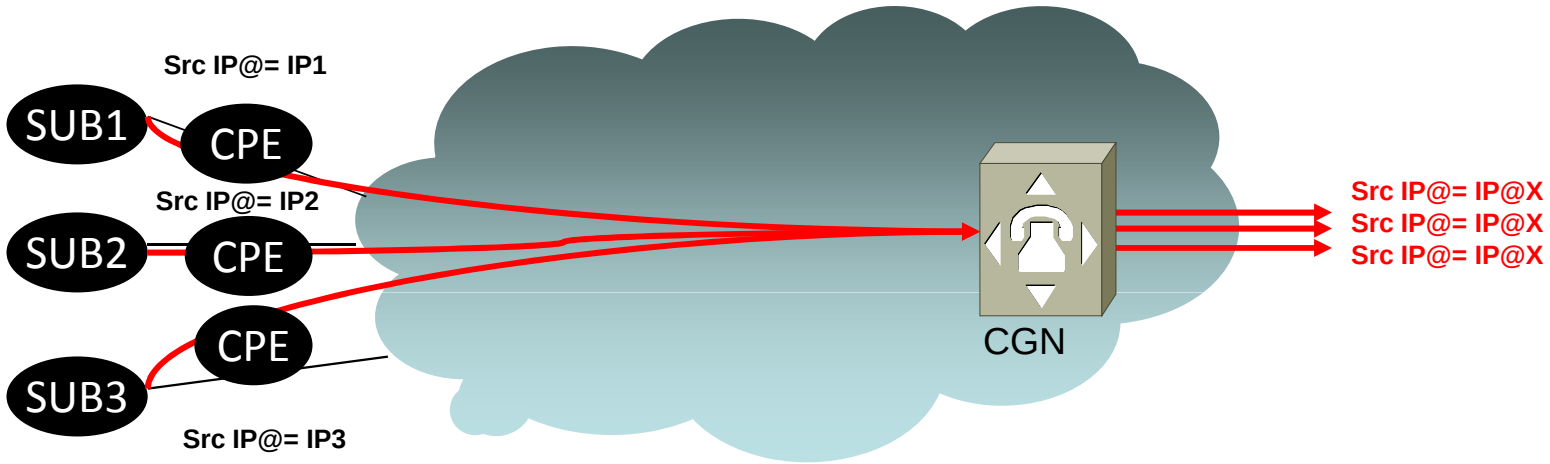
IETF 82-Taipei, November 2011

S. Sivakumar, E. Abdo, M. Boucadair and **J. Queiroz**

Address sharing – Problem statement

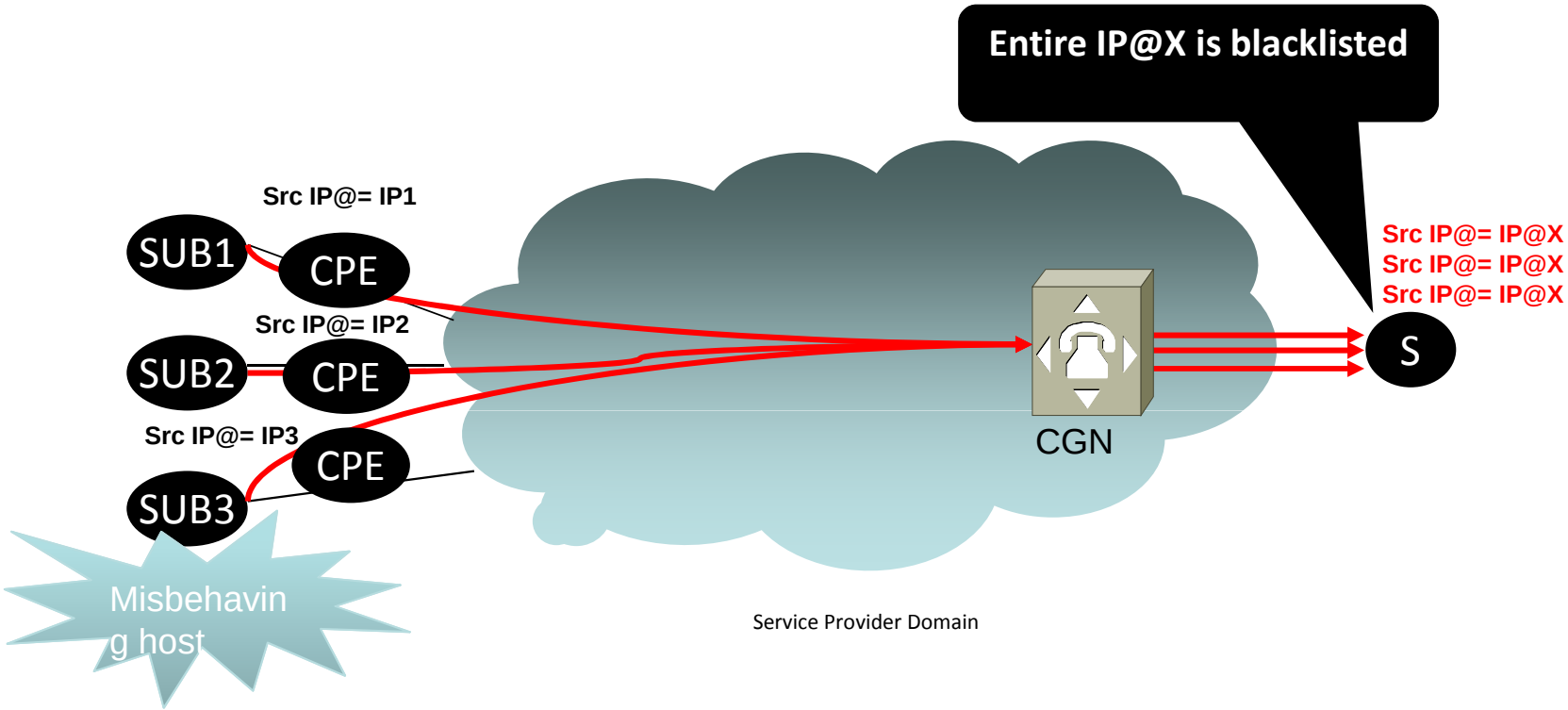
- Documented extensively
  - RFC 6269
  - Several I-Ds
- Applies to all address sharing entities
  - CGN/NAT64/DS-Lite/A+P/4rd/DIVI
  - Application proxies (e.g., HTTP proxies)
- Specific use case that causes denial of service

Address Sharing



The internal and the external IP addresses may be of distinct address families (e.g., IPv4, IPv6):
NAT44 or NAT64

Implicit Identification



Blacklisting a misbehaving user:
The server relies on the source IP address

All subscribers using the same address will be impacted:
Unhappy customers, calls to the hotline for the IP Network Provider ($$/mn, OPEX loss for the ISP)

A Solution is Needed

- Malicious host/user disrupt services
- Need generic solution across all address sharing mechanisms
  - CGN,NAT64, DS-Lite, A+P 4rd, dIVI, Application proxies
- I-D.boucadair-intarea-nat-reveal-analysis

	UDP	TCP	HTTP	Encrypted traffic	Success Ratio	Possible performance impact	Modify OS TCP/IP stack is needed (*)	Deployable	Notes
IP Option	Yes	Yes	Yes	Yes	30%	High	Yes	Yes	
TCP Option	No	Yes	Yes	Yes	99%	Med to High	Yes	Yes	
IP-ID	Yes	Yes	Yes	Yes	100%	Low to Med	Yes	Yes	1
HTTP Header (XFF)	No	No	Yes	No	100%	Med to High	No	Yes	2
Proxy Protocol	No	Yes	Yes	Yes	Low	High	No	No	
Port Set	Yes	Yes	Yes	Yes	100%	NA	No	Yes	1,3
HIP					Low	NA	--	No	4,5

HOST_ID as a TCP OPTION

- Original idea is documented in I-D.wing-nat-reveal-option
  - Denoted as HOST_ID_WING

- An additional TCP option format to convey a HOST_ID is also considered
  - ***Motivation***: cover also the load-balancer use case and provide richer functionality as Forwarded-For HTTP header
  - Denoted as HOST_ID_BOUCADAIR

Linux Kernel Modifications

- Support HOST_ID WING and HOST_ID BOUCADAIR
- Enable/Disable injecting HOST_ID TCP Option
- When HOST_ID TCP option is supported, the information to be injected is configurable:
  - Source IPv6 address or the first 64 bits of the address
  - Source IPv4 address
  - Source port number
  - Source IPv4 address and Source port
  - IPv6 address or the first 64 bits of the B4 when DS-Lite is activated
- When the HOST_ID TCP option is enabled, stripping any existing HOST_ID TCP option is enabled by default

I-D.abdo-hostid-tcpopt-implementation

- Methodology
  - A local server has been configured to ***verify*** HOST_ID TCP options are correctly injected
    - TCP options are injected by a remote host connected to Internet
  - TCP packets are issued ***simultaneously*** from a host supporting HOST_ID TCP Options and a “legacy” host
  - Tests are ***repeated*** several times…
  - A ***robot*** is used to issue TCP packets and to aggregate results
  - Testing has been conducted under ***several configurations***
    - Hosts behind managed CPEs from two ISPs
    - Hosts behind a firewall without any CPE in the path
    - Connected to an enterprise network
    - Hosts behind a DS-Lite CGN

I-D.abdo-hostid-tcpopt-implementation

- Various combinations of the HOST_ID TCP options have been tested
  - HOST_ID_WING
    - HOST_ID_WING has also been adapted to include 32 bits and 64 bits values
    - No particular impact on session establishment has been observed
  - HOST_ID_BOUCADAIR (source port)
  - HOST_ID_BOUCADAIR (IPv4 address)
  - HOST_ID_BOUCADAIR (source port:IPv4 address)
  - HOST_ID_BOUCADAIR (IPv6 Prefix)

Results: HTTP

	NOPT	OPT_WING	Diff
Top10	100,00000%	100,00000%	0,00000%
Top100	100,00000%	100,00000%	0,00000%
Top200	100,00000%	100,00000%	0,00000%
Top300	99,66667%	99,66667%	0,00000%
Top400	99,50000%	99,50000%	0,00000%
Top500	99,40000%	99,40000%	0,00000%
Top600	99,33333%	99,33333%	0,00000%
Top700	99,42857%	99,42857%	0,00000%
Top800	99,37500%	99,37500%	0,00000%
Top900	99,33333%	99,33333%	0,00000%
Top1000	99,40000%	99,40000%	0,00000%
Top2000	99,25000%	99,20000%	0,05000%
Top3000	99,13333%	99,10000%	0,03333%
Top4000	99,10000%	99,05000%	0,05000%
Top5000	99,08000%	99,04000%	0,04000%
Top6000	99,18333%	99,15000%	0,03333%
Top7000	99,21429%	99,15714%	0,05714%
Top8000	99,11250%	99,05000%	0,06250%
Top9000	99,11111%	99,05556%	0,05556%
Top10000	99,12000%	99,07000%	0,05000%

	NOPT	OPT_BOUCADAIR	Diff
Top10	100,00000%	100,00000%	0,00000%
Top100	100,00000%	100,00000%	0,00000%
Top200	100,00000%	100,00000%	0,00000%
Top300	99,66667%	99,66667%	0,00000%
Top400	99,50000%	99,50000%	0,00000%
Top500	99,40000%	99,40000%	0,00000%
Top600	99,33333%	99,33333%	0,00000%
Top700	99,42857%	99,42857%	0,00000%
Top800	99,37500%	99,37500%	0,00000%
Top900	99,33333%	99,33333%	0,00000%
Top1000	99,40000%	99,40000%	0,00000%
Top2000	99,25000%	99,20000%	0,05000%
Top3000	99,13333%	99,10000%	0,03333%
Top4000	99,10000%	99,05000%	0,05000%
Top5000	99,08000%	99,04000%	0,04000%
Top6000	99,18333%	99,15000%	0,03333%
Top7000	99,21429%	99,15714%	0,05714%
Top8000	99,11250%	99,05000%	0,06250%
Top9000	99,11111%	99,04444%	0,06667%
Top10000	99,12000%	99,06000%	0,06000%

No Impact for the Top1000 websites

Connection problems only with ***5 HTTP servers***

delay(HOST_ID_WING) < delay(NO_OPTION): 47,85 %
delay(HOST_ID_BOUCADAIR (source port:IPv4 address)) < delay(NO_OPTION): 47,06 %
delay(HOST_ID_BOUCADAIR (source port)) < delay(NO_OPTION): 54,9 %

Results: FTP

	NOPT	HOST_ID	Diff
first 10	100,00000%	100,00000%	0,00000%
first 100	100,00000%	100,00000%	0,00000%
first 200	100,00000%	99,50000%	0,50000%
first 300	100,00000%	99,33333%	0,66667%
first 400	99,75000%	99,25000%	0,50000%
first 500	99,80000%	99,40000%	0,40000%
first 600	99,83333%	99,50000%	0,33333%
first 700	99,71429%	99,42857%	0,28571%
first 800	99,75000%	99,50000%	0,25000%
first 900	99,77778%	99,44444%	0,33333%
first 1000	99,80000%	99,40000%	0,40000%
first 2000	99,75000%	99,30000%	0,45000%
first 2050	99,75610%	99,31707%	0,43902%

- A list of **5591** FTP servers has been used to conduct these testings
- Among this list, only **2050** was reachable:
  - Failure to reach 937 FTP servers due to connection timeout.
  - Failure to reach 1286 FTP servers due to DNS errors.
  - Failure to reach 717 FTP servers because access was denied.
  - Could not connect to 500 FTP servers
  - Etc.

Problems are encountered with 9 servers (from the 2050 servers list)

- Connection is frozen after "227 Entering passive mode..)"

Based upon the average of the session establishment with the 2050 FTP servers:

•delay(HOST_ID_WING) < delay(NO_OPTION): 48,43902 %
•delay(HOST_ID_BOUCADAIR (source port:IPv4 address))<delay(NO_OPTION):47,41463 %
•delay(HOST_ID_BOUCADAIR (source port)) < delay(NO_OPTION): 48,43902 %

Misc

- One "managed" CPE ***discard*** all SYN packets conveyed "badly" coded TCP options while another "managed" CPE forwards those packets to Internet
- Our testing demonstrated that ***2,6%*** of HTTP servers enforce some parsing validation for TCP options
- SSH and Telnet services have been tested locally

Next Steps

- Support the HOST_ID Injection in ***ACK mode***
- Support TCP options ***injection by the CGN*** and drive the appropriate testing to conclude about impact of using these options on the CGN performances
- ***Update the iptables*** module to enforce policies based upon the content of the HOST_ID TCP option

Appendix

I-D.boucadair-intarea-nat-reveal-analysis

	UDP	TCP	HTTP	Encrypted traffic	Success Ratio	Possible performance impact	Modify OS TCP/IP stack is needed (*)	Deployable	Notes
IP Option	Yes	Yes	Yes	Yes	30%	High	Yes	Yes	
TCP Option	No	Yes	Yes	Yes	99%	Med to High	Yes	Yes	
IP-ID	Yes	Yes	Yes	Yes	100%	Low to Med	Yes	Yes	1
HTTP Header (XFF)	No	No	Yes	No	100%	Med to High	No	Yes	2
Proxy Protocol	No	Yes	Yes	Yes	Low	High	No	No	
Port Set	Yes	Yes	Yes	Yes	100%	NA	No	Yes	1,3
HIP					Low	NA	--	No	4,5

(1) Requires mechanism to advertise NAT is participating in this scheme (e.g., DNS PTR record)
(2) This solution is widely deployed
(3) When the port set is not advertised, the solution is less efficient.
(4) Requires the client and the server to be HIP-compliant and HIP infrastructure to be deployed
(5) If the client and the server are HIP-enabled, the address sharing function does not need to insert a user-hint. If the client is not HIP-enabled, designing the device that performs address sharing to act as a UDP/TCP-HIP relay is not viable.

(*) Server side

IP option, IP ID and Proxy Protocol are **broken**

HIP is not “widely” **deployed**

Port Set requires **coordination**

XFF is **largely deployed** in operational networks but still the address sharing function **needs to parse all applications messages**

TCP Option is superior to XFF since it is not specific to HTTP but what about **UDP**? Update the Servers OS **TCP/IP is required**

HOST_ID_WING

HOST_ID_WING is sent in the SYN packet

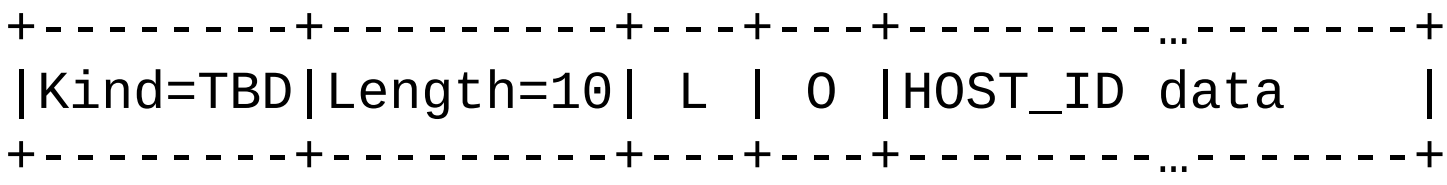
```
+---------+--------+---------------+
|Kind=TBD |Length=4| HOST_ID data  |
+---------+--------+---------------+
```

HOST_ID data: 16 bits

HOST_ID data can be:

- lower 16 bits of the IP address
- VLAN ID
- VRF ID…

HOST_ID_BOUCADAIR



L: Lifetime (value=validity time; RFC6250)

- 0: Permanent

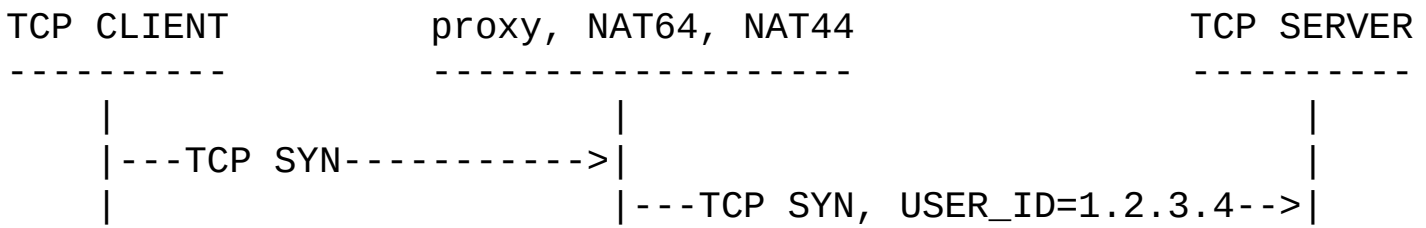
Origin:

- 0: Internal Port
- 1: Internal IPv4 address
- 2: Internal Port:Internal IPv4 address
- 3: IPv6 Prefix
- Else: No particular semantic;

USER_ID: depends on the content of the Origin field; padding is required

HOST_ID_BOUCADAIR

1. SYN Mode: the option is sent in the SYN packet



2. ACK Mode:
   1) Send HOST_ID_ENABLED in SYN
   2) If the remote TCP server supports that option, it must return it in SYNACK
   3) Then the TCP Client sends HOST_ID_BOUCADAIR in ACK