Analysis of Solution Candidates to Reveal a Host Identifier in Shared Address Deployments

draft-ietf-intarea-nat-reveal-analysis

IETF 83-Paris, March 2012

M. Boucadair, J. Touch, P. Levis and R. Penno

Status Update

- -00 (February 2012)
  - Update Privacy Section: A. Cooper provided the text
  - Add HOST_ID requirements (Comment received from A. Cooper)
    - Uniqueness of identifiers in HOST_ID: Local *vs.* Global
    - Refresh rate of HOST_ID
    - Manipulate HOST_IDs: Strip/re-write/Insert
    - Interference between HOST_IDs
  - Clarify IPv6 is also concerned
  - Re-organize the analysis section

- -01 (March 2012)
  - Add a new analysis section for the ICMP-based scheme proposed by Andrew
  - Update the analysis table
  - Cite RFC6462 as suggested by S. Brim

Synthesis

	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
ICMP	Yes	Yes	Yes	Yes	100%	High	Yes	Yes	6,7
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

(*) Server side

(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.
(6) 100% success ratio is implementation specific.
(7) The solution is inefficient in various scenarios

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. **Prototype** exists

Port Range is more suitable for **services offered by the service provider** operating the address sharing function. Otherwise, a dedicated mechanism to **advertise the port range assignment policy** is required

Tentative Conclusion?

- An exhaustive list of potential solutions to convey the HOST_ID are documented so far
  - Any missing approach to include?
  - Any missing criteria to the comparison table?

- Should we include a Recommendation?
  - If Yes, comment the following proposal:
    - Indicate XFF, TCP Option and port ranges as viable solutions
    - Recommend to pursue the TCP Option specification effort given APPSAWG adopted recently “Forwarded-For” http://tools.ietf.org/html/draft-ietf-appsawg-http-forwarded-00