UDP Magic Numbers

draft-herbert-udp-magic-numbers-01

Tom Herbert <tom@herbertland.com>

Problem statement

- Middleboxes are motivated to parse UDP payloads to provide services. e.g.:
    - Firewalls for tunnels
    - SPUD
    - BIER
- Matching port numbers at middle boxes for this purpose is not robust

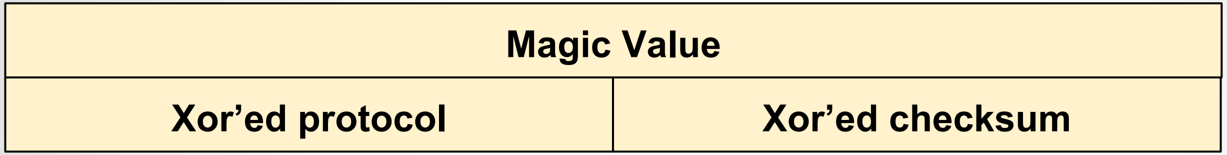
Port number matching issue

- RFC7605: Meaning of transport ports is shared agreement between source and destination only
- Misinterpreting a packet could have bad effects, e.g. packet loss of otherwise acceptable traffic
- There is no concept of application binding to a destination port, apps can send packets to UDP ports without needing privileges
- UDP encapsulations use source port for entropy so that does not help

UDP magic numbers

- Insert a 64 bit magic number at beginning of UDP payload for identification
- Generalizes concepts of SPUD and STUN (RFC5389)
- Magic number is unique per application protocols
- No change to UDP protocol

Proposed format



Magic Value

Xor’ed protocol

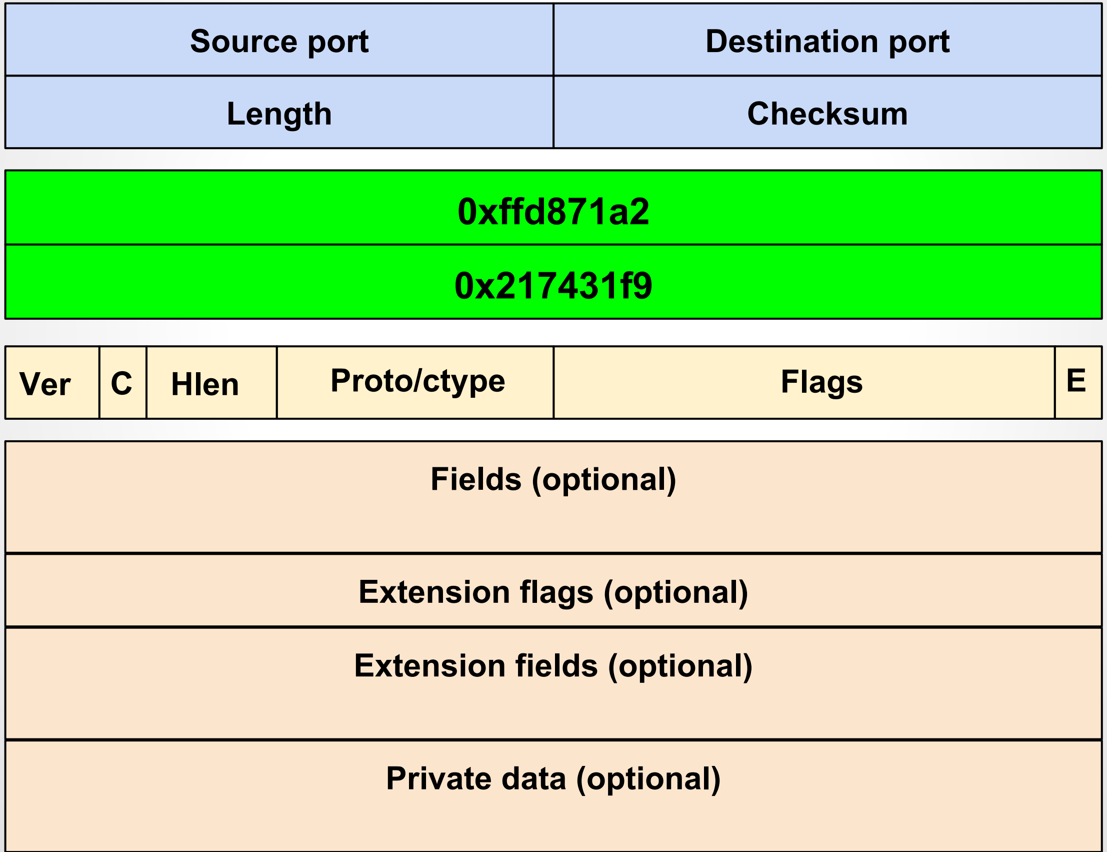
Xor’ed checksum

- *Magic Value*: 0xffd871a2
- *Xor’ed protocol*: A protocol number for payload XOR 0x36b4
- *Xor’ed checksum*: Checksum over Magic Value and Xor’ed protocol fields XOR 0x5ce9

Requirements

- Magic number must be distinguishable from application layer data
- Protocol numbers can be assigned ports for convenience
- Application protocols may allow magic number use to be optional or required

GUE example (port=6080)



Source port
Destination port
Length
Checksum
0xffd871a2
0x217431f9
Ver
C
Hlen
Proto/ctype
Flags
E
Fields (optional)
Extension flags (optional)
Extension fields (optional)
Private data (optional)

Thankyou!