PowerDNS

PROXYv2 Protocol for DNS

Passing source information from loadbalancers to DNS backends

Pieter Lexis **Peter van Dijk**

OARC 36, 29th of November 2021

POWERDNS

The Problem

- DNS set-ups can be fronted by loadbalancers/proxies
- Backend servers require true client IP address for ACL, views, or other purposes
- Proxies may not want to do extensive packet parsing and processing
- Backends may also want to know about transports used, including port numbers

POWERDNS

Existing Solutions

- EDNS Client Subnet
- X-Proxied-For (XPF)
- Private/bespoke EDNS option

POWERDNS

Drawbacks of EDNS0 Client Subnet

- Squatting existing EDNS options is **a bad idea**
- Requires parsing and modifying DNS packets in-flight
- No way to pass ECS from recursor to proxied auth
- No port information
- No transport protocol information

POWERDNS

Drawbacks of XPF and Bespoke Options

XPF

- Draft expired in the IETF
- Requires parsing and modifying DNS packets in-flight
- Requires special handling to not break TSIG

Private EDNS option

- Requires changes to all software in the chain
- Hard to debug with standard tools

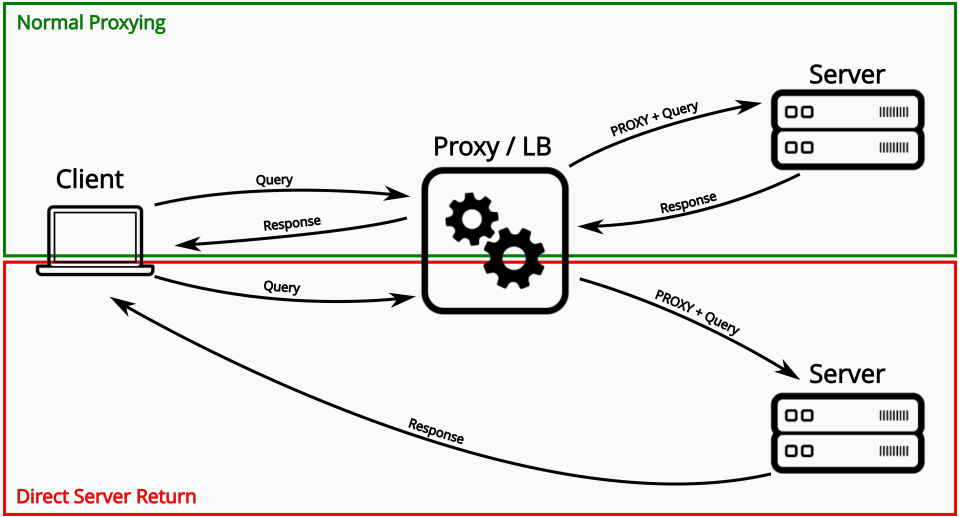
POWERDNS

The PROXYv2 Protocol

- Binary protocol
- Prefixes proxied data
- Passes v4, v6 addresses, ports, and protocol (TCP/UDP)
- Extensible with any number of arbitrary TLV fields
- Already supported by several loadbalancer vendors
- Goes well together with ECS used as intended

POWERDNS

PROXY Header Placement



Implementations

- Loadbalancers (HAProxy, f5) [TCP only]
- Webservers (nginx, Apache) [TCP only]
- dnsdist 1.5.0
- PowerDNS Authoritative Server 4.6.0
- PowerDNS Recursor 4.4.0
- Roadmapped for BIND9 and Unbound
- Experimental patch for Knot DNS

POWERDNS

PROXY Protocol description

- Header, followed by
  - Proxied address: src, dst, including ports
  - Arbitrary data as tag, length, value
  - Original data
- Header prefixed to every query (UDP)
- Sent first in connection (TCP)
- No header in replies!
- No TCP reuse to backend for different clients

```
+-----------------+
| Header          |
+-----------------+
| Proxied address |
+-----------------+
/ TLVs            /
+-----------------+
| Original data   /
/          ...    |
+-----------------+
```

POWERDNS

PROXY Header

```
struct proxy_hdr_v2 {
    /*
        \r \n \r \n \x00 \r \n Q  U  I  T  \n
    hex 0D 0A 0D 0A 00   0D 0A 51 55 49 54 0A
    */
    uint8_t sig[12];
    /* protocol version (4 bits, \x2)
       and command (4 bits) */
    uint8_t ver_cmd;
    uint8_t fam;      /* protocol family and socket family */
    uint16_t len;     /* number of following header bytes */
};
```

POWERDNS

Proxied Address

```
union proxy_addr {
    struct {        /* for TCP/UDP over IPv4, len = 12 */
        uint32_t src_addr;
        uint32_t dst_addr;
        uint16_t src_port;
        uint16_t dst_port;
    } ipv4_addr;
    struct {        /* for TCP/UDP over IPv6, len = 36 */
         uint8_t  src_addr[16];
         uint8_t  dst_addr[16];
         uint16_t src_port;
         uint16_t dst_port;
    } ipv6_addr;
    struct {        /* for AF_UNIX sockets, len = 216 */
         uint8_t src_addr[108];
         uint8_t dst_addr[108];
    } unix_addr;
};
```

POWERDNS

TLV

```
struct pp2_tlv {
    uint8_t type;
    uint8_t length_hi;
    uint8_t length_lo;
    uint8_t value[0];
};
```

POWERDNS

Further Reading

PROXY protocol specification

dnsdist documentation on using the PROXY protocol

ISC BIND9 issue for implementing PROXY protocol

Unbound feature request to add Proxy Protocol support

Knot feature request and patch

POWERDNS

Questions?