Communicating Proxy Configurations in Provisioning Domains

draft-ietf-intarea-proxy-config-06

Tommy Pauly, Dragana Damjanovic, & Yaroslav Rosomakho
INTAREA
IETF 123, July 2025, Madrid

Background

PvD JSON format that lets a proxy describe its configuration

- Alternate protocols (upgrade to secure types)
- Matching rules for when to use a proxy; standardized alternative to PAC files

First shared in July 2023, adopted May 2024

Authors believe the spec is complete and ready to use

Updates since IETF 122

- Updated destination match rules
  - Ordered list of rule sets
  - More examples!
- Guidance for vendor-specific keys
  - SVCB parameter hint
  - Expanded security considerations

Destination Rules

Each destination rule is a dictionary that contains:

- DNS domains & FQDNs *or* IP subnets
- Optional port ranges
- List of proxies that the rule applies to (or an empty list for to define an exception rule)

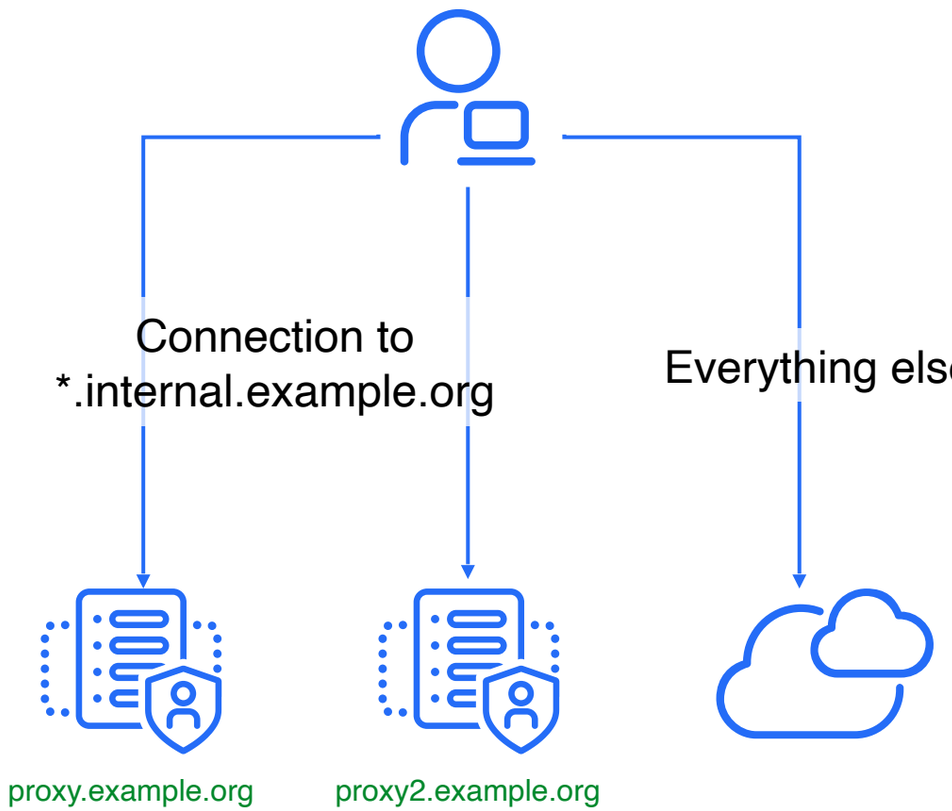
The PvD contains an ordered list of destination rules

To route traffic, evaluate the rules in order

Proxy definitions now contain an identifier string that is used as a reference from destination rules

Destination Rules

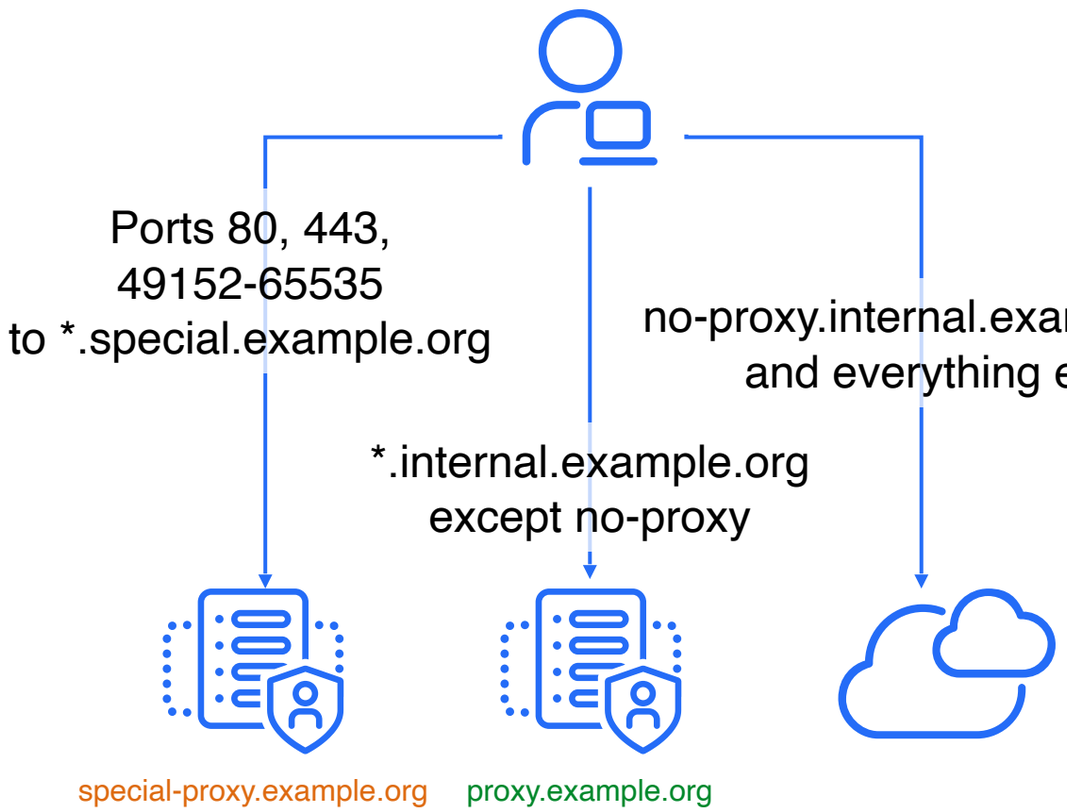
Example 1: Load Balancing




```
"proxies": [
  {
    "protocol": "http-connect",
    "proxy": "proxy.example.org:80",
    "identifier": "default_proxy"
  },
  {
    "protocol": "http-connect",
    "proxy": "proxy2.example.org:80",
    "identifier": "default_proxy"
  }
],
"proxy-match": [
  {
    "domains": [ "*.internal.example.org" ],
    "proxies": [ "default_proxy" ]
  }
]
```

Destination Rules

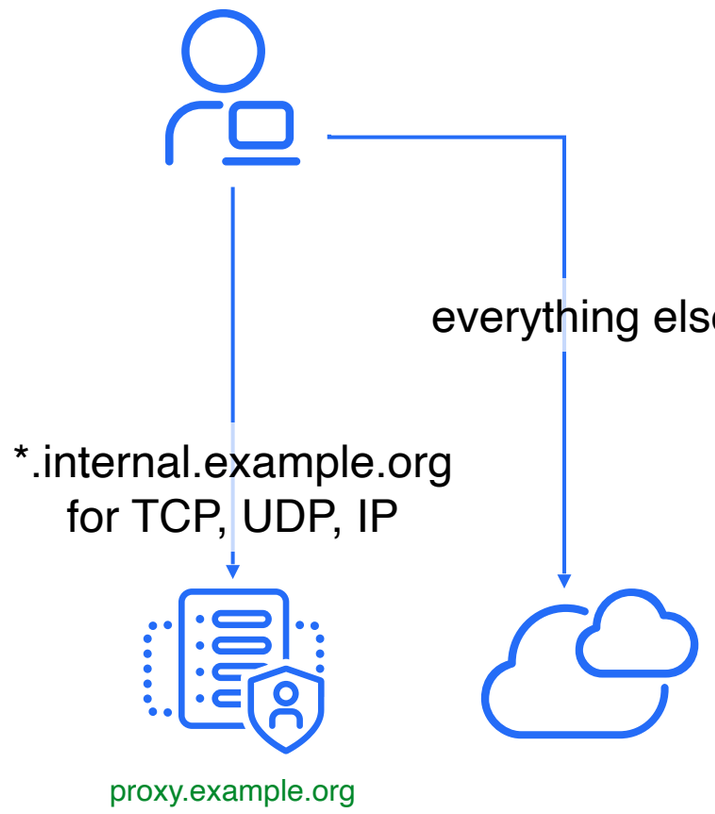
Example 2: Split routing and exceptions




```
"proxies": [
  {
    "protocol": "http-connect",
    "proxy": "proxy.example.org:80",
    "identifier": "default_proxy"
  },
  {
    "protocol": "http-connect",
    "proxy": "special-proxy.example.org:80",
    "identifier": "special_proxy"
  }
],
"proxy-match": [
  {
    "domains": [ "*.special.example.org" ],
    "ports": [ "80", "443", "49152-65535" ],
    "proxies": [ "special_proxy" ]
  },
  {
    "domains": [ "no-proxy.internal.example.org" ],
    "proxies": [ ]
  },
  {
    "domains": [ "*.internal.example.org" ],
    "proxies": [ "default_proxy" ]
  }
]
```

Destination Rules

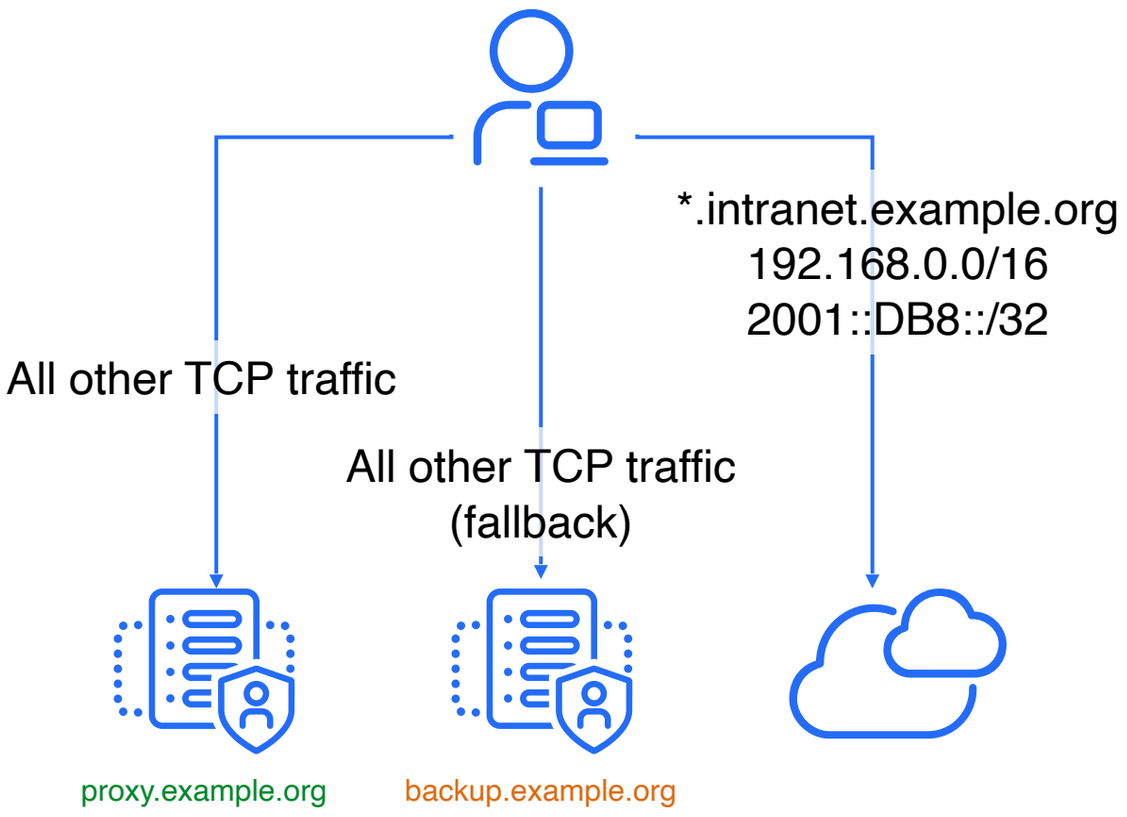
Example 3: Different protocols with shared rules




```
"proxies": [
  {
    "protocol": "http-connect",
    "proxy": "proxy.example.org:80",
    "identifier": "default_proxy"
  },
  {
    "protocol": "connect-udp",
    "proxy": "https://proxy.example.org/masque/udp/
{target_host}/{target_port}",
    "identifier": "default_proxy"
  },
  {
    "protocol": "connect-ip",
    "proxy": "https://proxy.example.org/masque/ip{?
target,ipproto}",
    "identifier": "default_proxy"
  }
],
"proxy-match": [
  {
    "domains": [ "*.internal.example.org" ],
    "proxies": [ "default_proxy" ]
  }
]
```

Destination Rules

Example 4: Default proxies with exceptions




```
"proxies": [
  {
    "protocol": "http-connect",
    "proxy": "proxy.example.org:80",
    "identifier": "default_proxy"
  },
  {
    "protocol": "http-connect",
    "proxy": "backup.example.org:80",
    "identifier": "secondary_proxy"
  }
],
"proxy-match": [
  {
    "domains": [ "*.intranet.example.org" ],
    "proxies": [ ]
  },
  {
    "subnets": [ "192.168.0.0/16", "2001:DB8::/32" ],
    "proxies": [ ]
  },
  {
   "proxies": [ "default_proxy", "secondary_proxy" ]
  }
]
```

Vendor-Specific Keys

Denoted by containing an underscore (`acme_authmode`)

Custom keys can be in both proxy definitions and destination rules

- Can be enforced with `mandatory` flag in proxy definition

Unknown keys in destination rules always cause the rule to be ignored by a client

SVCB Parameter

Added a `pvd` hint in SVCB/HTTPS DNS records

- Helps discover that a proxy has a PvD
- Useful for other cases than proxies

Not required, but helpful to know that it is worth checking

Next steps

Separate implementations (clients and proxy/ publishers) being tested at scale

Requesting to start WGLC

- Please review!