Table of Contents

DNS Response Spoofing

Purpose

The DNS response spoofing service allows modifying IP addresses returned by the DNS server for specific domain names. This enables overriding IP addresses in server responses for certain types of DNS queries specified in the service configuration.
This service is effective for controlling client DNS requests and redirecting them to alternative IP addresses. It is used to balance traffic to certain resources between different servers based on IP address.

Service Operation Scheme

1. The client makes a specific type of query to the DNS server (e.g., type A).
2. DPI sees the query and checks if the spoofing service is assigned for this client (IP source) for a specific resource. If blocking is configured for that query type, DPI simply drops the DNS query with that specific record type.
3. If the service is assigned, DPI drops the original client DNS query and generates a DNS server response based on the rules specified in the service.
4. DPI forwards the modified response to the client. The client is unaware of the modification and treats the response as legitimate.

Supported DNS record types:

- A — IPv4 address (length — 32 bits);
- AAAA — IPv6 address (length — 128 bits);
- MX — txt-record containing information about mail servers handling mail.

Possible actions for DNS queries:

- `ya.ru HTTPS #drop` — DPI drops the DNS query with HTTPS record type
- `ya.ru A #nxdomain` — DPI sends a response indicating the domain does not exist
- `mail.ru MX smtp.googlemail.com` — in this case, for a query to mail.ru with MX type, the response should indicate that mail.ru has a mail server at smtp.googlemail.com with a `preference` of 10.

Configuration

1. Create a text file and add DNS spoofing rules specifying the domain name, DNS record type, and IP address to be returned for that domain. Wildcard domains with * are supported.

```
vi test.txt
google.com A 192.0.2.1
test.com A #nxdomain
```

```
example.com AAAA 2001:db8:85a3::8a2e:370:7334
yahoo.com HTTPS #drop
*.fb.com A 203.0.113.5
outlook.com MX smtp.googlemail.com
```

2. Convert the text file to binary format using the `dns2dic` utility, which is readable by DPI:

```
cat test.txt|dns2dic test.bin
```

3. Place the resulting binary file in the directory where DPI will read it from:

```
cp test.bin /var/lib/dpi/dns.bin
```

4. Create a service profile:

```
fdpi_ctrl load profile --service 19 --profile.name test_193 --
profile.json '{ "dns_list" : "/var/lib/dpi/dns.bin" }'
```

`max_profiles_serv19` — setting for the maximum number of profiles. Default is 32.

By default, the DNS response is sent to the interface from which the query originated (IN interface where the query came from). Sending to the OUT interface is relevant in DPI's asymmetric mode (outbound traffic only). Configurable in `fastdpi.conf` with the parameter `emit_direction=2`

Management

Command format:

```
fdpi_ctrl [command] --service 19 [options list] [login or vchannel]
```

Enable the service:

```
fdpi_ctrl load --service 19 --profile.name test_193 --login test
#or
fdpi_ctrl load --service 19 --profile.name test_193 --vchannel 1
```

Disable the service:

```
fdpi_ctrl del --service 19 --profile.name test_193 --login test
#or
fdpi_ctrl del --service 19 --profile.name test_193 --vchannel 1
```