DNS Interception: Protecting the Client

Brett Smith, 2015-08-04

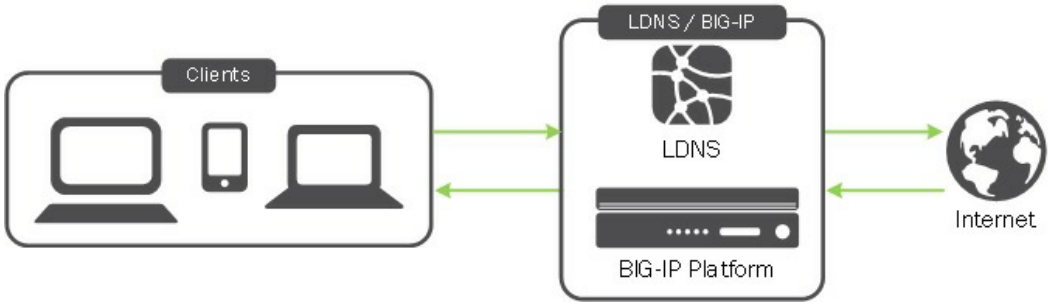
Introduction

Everything starts with a DNS request. So why not use it to protect the client? With the recent addition of Secure Web Gateway Services to the F5 line up of modules in TMOS 11.5, it provided the ability to access a URL Categorization database via iRules that contains 150 URL categories and identifies over 60 million URLs. Pair that with the IP Intelligence Services that was introduced in TMOS 11.2 and some DNS iRules, you now have a solution to filter all DNS Requests and Reponses originating from your network. This simple but powerful tool gives you the ability to protect clients which you may not have control over by sending back a Non-Existent Domain in the response to prevent the client from connecting to the malicious server or undesirable content.

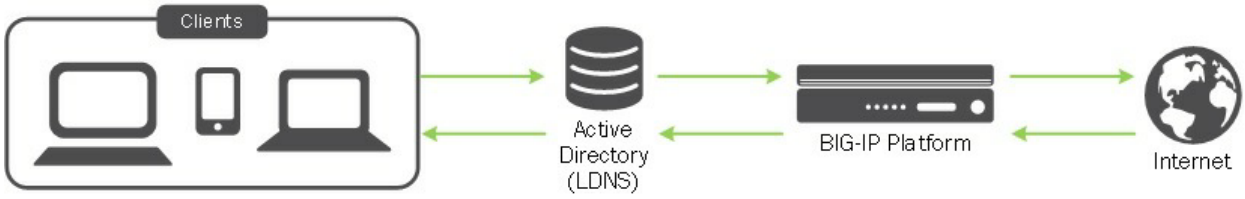
This iRule solution is applied to a DNS resolver or a catch all (0.0.0.0/0:53) DNS virtual server where the BIG-IP is a default gateway. It allows the BIG-IP to explicitly or transparently intercept all DNS Requests and Responses from the client and apply security filtering controls. This solution is suited to almost any outbound DNS scenario where you need to protect the client from accessing malicious threats or undesirable content intentionally or unintentionally. One example where you may find this solution handy, is on a Guest or BYOD network where you need a transparent method of adding security when you don't have control of the client.

How the solution works

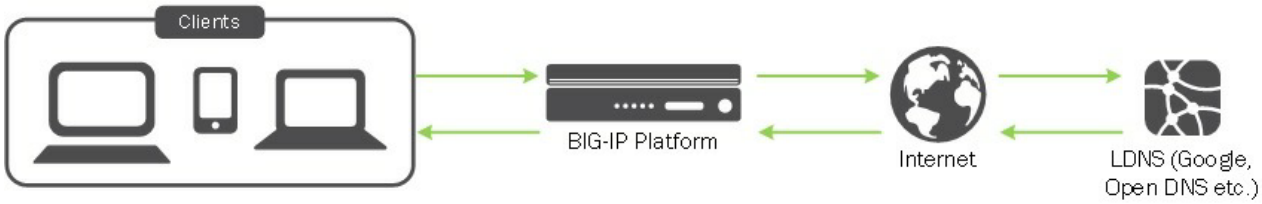
Scenario 1: The BIG-IP is configured as a DNS resolver and the client's DNS settings have been configured via DHCP with the IP of the BIG-IP as the LDNS.



Scenario 2: The BIG-IP is a default gateway on the network and a catch all (0.0.0.0/0:53) DNS virtual server transparently intercepts all DNS Requests an Responses. The client's DNS settings have been configured via DHCP with the IP of a LDNS that has to traverse the BIG-IP.



Scenario 3: The BIG-IP is a default gateway on the network and a catch all (0.0.0.0/0:53) DNS virtual server transparently intercepts all DNS Requests an Responses. The client's DNS settings have been configured via DHCP or manually with a public DNS service (e.g Google, Open DNS, etc.) an the client has to traverse the BIG-IP.



The “client” in the above scenarios doesn’t have to be an end user device such as a tablet, it could be a forward proxy server for example.

The iRule is applied to the virtual server (VS) with a DNS profile and the events DNS_REQUEST and DNS_RESPONSE are triggered. When the DNS_REQUEST is triggered, the DNS Question Name is passed to the URL Categorization database using CATEGORY::lookup. When the DNS_RESPONSE is triggered, the IP address in the DNS Response can be passed to IP Intelligence database using IP::reputation.

Solution Features

DNS Request Filtering configurable items:

- **DNS Request Filtering** - Enable or Disable all DNS_REQUEST filtering.
- **URL Categories** e.g. Adult_Content, Hacking. If the DNS Question (FQDN - e.g. playboy.com) matches a category in the data group (default: dns_request_url_categories_dg), NXDOMAIN will be returned in the response.
- **DNS Question Type** e.g. A, AAAA, ANY etc. Only the Question Types configured in the data group (default: dns_request_question_type_dg) will be filtered.
- **FQDN/TLD Whitelist** e.g. f5.com or .gov. Any FQDN/TLD in the whitelist data group (default: dns_request_fqdn_whitelist_dg) will bypass DNS_REQUEST filtering regardless of the Question Type or URL Category.

DNS Response Filtering configurable items:

- **DNS Response Filtering** - Enable or Disable all DNS_RESPONSE filtering.
- **IP/Subnet Whitelist** e.g 192.168.0.0/16 or 1.1.1.1. Any IP or IP Subnet in the whitelist data group will bypass DNS_RESPONSE filtering.
- **IPI Threat Categories** e.g. Spam Sources, Phishing. If the DNS RDATA (A & AAAA only) matches a category in the data group, NXDOMAIN will be returned in the response.

Global Parameters

- **Logging Control** - Off, Level 1 (NXDOMAIN and Whitelist Matching) and Level 2 (All DNS Requests & Responses)

Requirements - BIG-IP Version / Licensing

- BIG-IP 11.2+ for IPI Subscription (DNS Response filtering)
- BIG-IP 11.5+ for URL Categorization or SWG Subscription (DNS Request filtering)
- Licensing:
    - GTM/DNS or DNS Services add-on to LTM
    - URL Categorization Subscription or SWG Subscription for DNS Request filtering
    - IPI Subscription for DNS Response filtering

Configuration

1. Data Groups

Multiple data groups are used throughout the solution to make it easy for the administrator to make changes on the fly without having to change the iRule. By default, the following data groups need to be created. The values can be modified to your liking.

1.1 Data Group Name: *dns_request_url_categories_dg*
Purpose: URL Category Names

If the DNS Question Name (e.g. playboy.com) matches a category (Adult_Content) in the data group, NXDOMAIN will be returned in the response. To obtain a list of possible URL Categories and their descriptions, run: ***tmsh list sys url-db url-category { description }***. Example categories are included below.

TMSH:

```
create ltm data-group internal dns_request_url_categories_dg type string
modify ltm data-group internal dns_request_url_categories_dg records add {"Adult_Content"}
modify ltm data-group internal dns_request_url_categories_dg records add
{"Advanced_Malware_Command_and_Control"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Advanced_Malware_Payloads"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Bot_Networks"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Compromised_Websites"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Elevated_Exposure"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Emerging_Exploits"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Hacking"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Keyloggers"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Malicious_Embedded_Link"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Malicious_Embedded_iFrame"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Malicious_Web_Sites"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Militancy_and_Extremist"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Mobile_Malware"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Newly_Registered_Websites"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Nudity"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Peer-to-Peer_File_Sharing"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Phishing_and_Other_Frauds"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Proxy_Avoidance"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Sex"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Spyware"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Tasteless"}
modify ltm data-group internal dns_request_url_categories_dg records add {"Web_and_Email_Spam"}
```

1.2 Data Group Name: *dns_request_question_type_dg*
Purpose: DNS Question Types

Only the Question Types (e.g. A, AAAA) configured in the data group will be filtered. Example Question Types are included below.

TMSH:

```
create ltm data-group internal dns_request_question_type_dg type string
modify ltm data-group internal dns_request_question_type_dg records add {"A"}
modify ltm data-group internal dns_request_question_type_dg records add {"AAAA"}
modify ltm data-group internal dns_request_question_type_dg records add {"ANY"}
modify ltm data-group internal dns_request_question_type_dg records add {"CNAME"}
modify ltm data-group internal dns_request_question_type_dg records add {"MX"}
```

1.3 Data Group Name: *dns_request_fqdn_whitelist_dg*
Purpose: FQDN / TLD Whitelisting

Any FQDN/TLD (e.g. f5.com or .gov) in the whitelist data group will bypass DNS_REQUEST filtering regardless of the Question Type or URL Category. Example Question Types are included below.

TMSH:

```
create ltm data-group internal dns_request_fqdn_whitelist_dg type string
modify ltm data-group internal dns_request_fqdn_whitelist_dg records add {"f5.com"}
```

1.4 Data Group Name: *dns_response_ip_whitelist_dg*
Purpose: IP / Subnet Whitelisting

Any IP or IP Subnet in the whitelist data group will bypass DNS_RESPONSE filtering regardless of the IP Reputation.

TMSH:

```
create ltm data-group internal dns_response_ip_whitelist_dg type ip
modify ltm data-group internal dns_response_ip_whitelist_dg records add {"10.0.0.0/8"}
modify ltm data-group internal dns_response_ip_whitelist_dg records add {"172.16.0.0/12"}
modify ltm data-group internal dns_response_ip_whitelist_dg records add {"192.168.0.0/16"}
```

1.5 Data Group Name: *dns_response_ipi_categories_dg*
Purpose: IP Intelligence Category Names

TMSH:

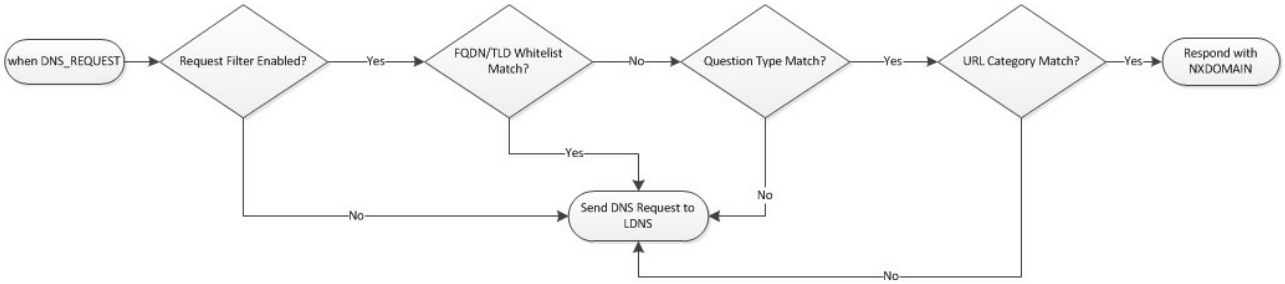
```
create ltm data-group internal dns_response_ipi_categories_dg type string
modify ltm data-group internal dns_response_ipi_categories_dg records add {"BotNets"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Networks"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Denial of Service"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Illegal"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Infected Sources"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Phishing"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Scanners"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Spam Sources"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Web Attacks"}
modify ltm data-group internal dns_response_ipi_categories_dg records add {"Windows Exploits"}
```

2. iRule

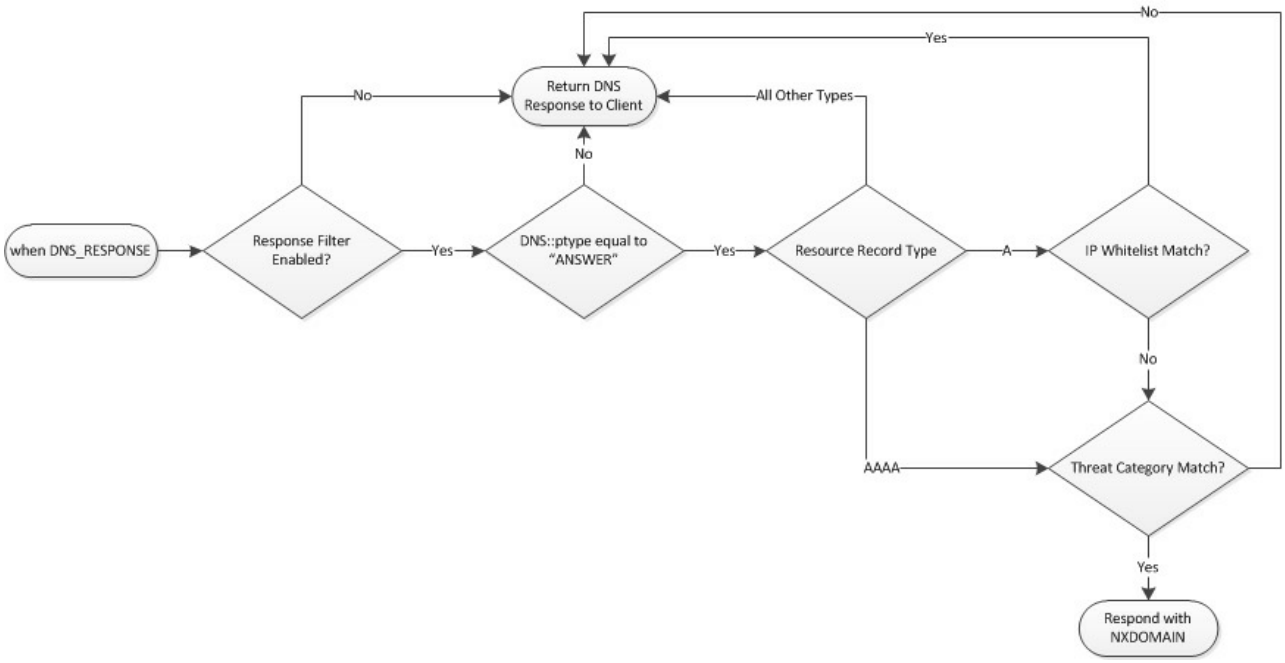
iRule Name: *dns_request_response_filter_irule -> https://devcentral.f5.com/codeshare/dns-request-and-response-filtering-using-url-db-and-ipi-subscriptions*

The iRule is applied to the DNS resolver or a catch all (0.0.0.0/0:53) DNS virtual server. The logic is explained below and has been built to cater for most situations. Simply save the iRule to the BIG-IP using the Codeshare link above.

DNS Request iRule logic:



DNS Response iRule logic:



3. Virtual Servers, Pools, Nodes and DNS Profile

3.1 BIG-IP as a DNS Resolver (Scenario 1)

When the BIG-IP is configured as a DNS Resolver, you need to configure a LDNS for the BIG-IP to resolve the requests. In my example I'm using Google's public DNS servers. Make sure you change the Virtual Server IP and/or any other settings to suit your environment.

TMSH:

```
create ltm node google-public-dns-a { address 8.8.8.8 }
create ltm node google-public-dns-b { address 8.8.4.4 }


create ltm pool google_dns_pool { members replace-all-with { google-public-dns-a:domain google-public-dns-b:domain } }


create ltm profile dns dns_interception { cache none defaults-from dns dns64 disabled dns-security none enable-cache no enable-dns-express no enable-dns-firewall no enable-dnssec no enable-gtm no enable-logging no process-rd yes process-xfr no unhandled-query-action allow use-local-bind no }


create ltm virtual dns_resolver_udp_vs { destination 10.1.1.1:domain ip-protocol udp profiles replace-all-with { udp_gtm_dns dns_interception } source-address-translation { type automap } rules { dns_request_response_filter_irule } pool google_dns_pool }
```

3.2 BIG-IP is a default gateway (Scenario 2 and 3)

When the BIG-IP is configured as a default GW, you only need to configure the catch all (0.0.0.0/0:53) DNS virtual server and the DNS Profile. Make sure you change the VLAN and/or any other settings to suit your environment.

TMSH:

```
create ltm profile dns dns_interception { cache none defaults-from dns dns64 disabled dns-security none enable-cache no enable-dns-express no enable-dns-firewall no enable-dnssec no enable-gtm no enable-logging no process-rd yes process-xfr no unhandled-query-action allow use-local-bind no }
```

```
create ltm virtual catch_all_dns_udp_vs { destination 0.0.0.0:domain ip-protocol udp profiles replace-all-with { udp_gtm_dns dns_interception } vlans-enabled vlans replace-all-with { vlan1 } rules { dns_request_response_filter_irule } translate-address disabled }
```

Conclusion

By combining threat intelligence services with DNS, produces a simple and effective protection in a IoT world.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com