DNS

https://campus.barracuda.com/doc/96026352/

CloudGen Firewall and DNS

The Domain Name System (DNS) is organized as a hierarchical tree. This tree contains information for mapping Internet hostnames to IP addresses and vice versa. The structure is stored as information in a distributed database that is managed by multiple DNS servers. The role of the CloudGen Firewall as a DNS server (authoritative, primary, secondary, caching-only) depends on how and what part of the tree structure is managed by a name server and how the servers exchange information to keep the tree's information up to date.

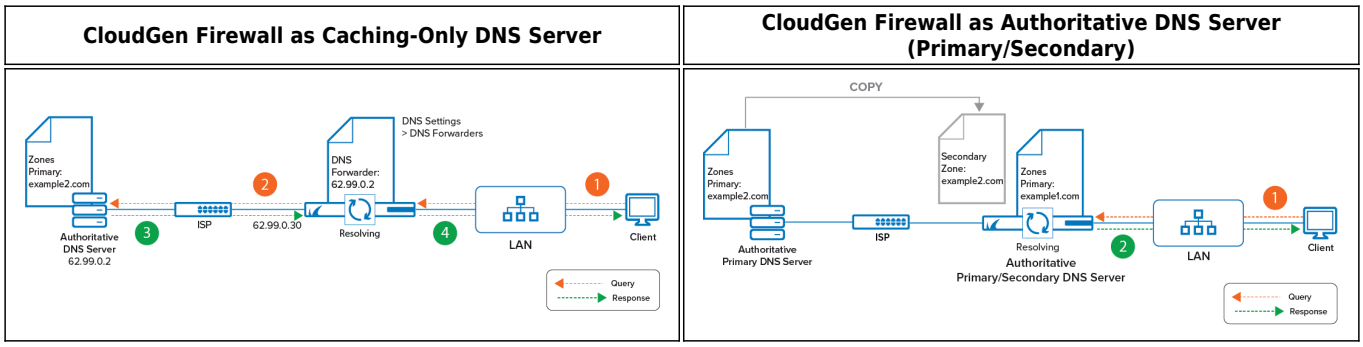
Barracuda Networks' implementation of the DNS extends the basic functionality of the BIND-based DNS system by additional features. Health probes provide a means to check the availability of certain target hosts. Listeners can be explicitly configured to snoop for incoming DNS resolving requests on selected interfaces. These two main features can then be combined to allow DNS load balancing depending also on the availability of certain targets.

Authoritative DNS Server vs. Caching-Only Server

The Barracuda CloudGen Firewall can act both as an authoritative (primary and/or secondary) and a caching-only DNS server, but it cannot run both at the same time.

Based on its factory settings, the CloudGen Firewall is operating as a caching-only DNS server and takes complete care of resolving the query for a client on the LAN. For more information, see How to Configure DNS Settings and How to Configure a Caching DNS Service.
The resolving strategy for queries originating from remote networks depends on further configurations. Because of this initial configuration as a caching-only DNS server, the CloudGen Firewall can query the related DNS server(s) on the Internet and do the whole license activation after being powered up for the first time.

If the firewall operates as an authoritative server (primary/secondary), it will return definitive answers to DNS queries about domains and domain hosts specified in its configuration (primary zone / secondary zone). A firewall can operate simultaneously both as a primary and secondary DNS server. For example, it can manage configured primary zones for its own domain and secondary zones for a foreign domain as copies from another primary DNS server. As an authoritative DNS server, the CloudGen Firewall can also answer DNS queries from remote networks.



DNS Zones and DNS Records

DNS Records

The CloudGen DNS service manages domains and hosts. All information about hosts and associated IP addresses is defined in resource records where each record holds a key-value (record type identifier) that classifies the rest of the information, e.g., information about whether the record describes a zone (SOA), a name server (NS), a web server (A / AAAA), or a mail exchanger (MX). For an easier configuration, you can select the requested record type out of a set of predefined record classes during the configuration process. In all resource records, additional information like time-to-live (TTL) or serial numbers define whether and in which periods zone information needs to be synchronized between multiple name servers in order to keep the DNS tree consistent in the distributed DNS database.

DNS Zones

A zone is a well-defined point of delegation in the DNS tree and contains contiguous parts of that tree for which a name server manages the respective information. If the CloudGen Firewall holds the primary copy of the zone data (primary zone), it operates as a primary server. The following image shows the record for a primary zone with two subordinated resource records on the CloudGen Firewall:

Domain	Type	Owner	TTL	Email	Record Data	Description
example.com (Master)			86400	root@example.com		
	NS	@	3600		ns1.example.com	Automatically generated
	A	ns1	3600		62.99.0.30	Automatically generated

If the CloudGen Firewall loads the information from another server and holds it as a copy (secondary zone), it operates as a secondary server. If the CloudGen Firewall is configured as a primary server, it is classified as an authoritative server. If the CloudGen Firewall is configured as a secondary server, the secondary role is indicated by a simple secondary entry without the possibility of viewing the hosted information because changes to this information can be made only on the responsible primary

server. The following image shows the record for a secondary zone on the firewall:

Domain	Type	Owner	TTL	Email	Record Data	Autocreate	Description
example.com (Slave)			1d			No	

If the CloudGen Firewall also holds information about which IP address maps to which Internet hostname, this is a reverse zone. If a reverse zone is configured, feeding the DNS server with an IP address in the query (reverse lookup) returns the associated hostname of a specific host within the respective domain:

30.0.99.62.in-addr.arpa (Reverse)			86400	root@example.com		No	
	PTR	11.0.99.62.in-addr.arpa	3600		www.example.com	No	Web server

For more information, see [How to Configure a Zone](#).

Zone Transfers

Zone transfers are essential for keeping configured primary zones synchronized to all secondary DNS servers. In order for a CloudGen Firewall-based primary DNS server to transfer a zone to a secondary DNS server, two entries must be added to the primary DNS server for the secondary DNS server. The following example assumes that the secondary DNS server can be reached on IP addresses like 5.5.5.239 and 6.6.6.239:

Record Type	Example for Entry
An A-record	`A  ns2 5.5.5.239, 6.6.6.239`
An NS-record	`NS @ ns2.standalone.org`

On the CloudGen Firewall, synchronization is basically achieved by updating the zone configuration on the primary DNS server. When clicking **Send Changes / Activate**, the serial number of the zone record is incremented by one. Because the primary zone record now has a higher serial number than the version on the secondary DNS server, the secondary server will take over new zone data from the primary one.

Synchronization of Zone Transfers

Sometimes, however, the serial number of the primary server is lower than the serial of a secondary DNS server. For instance, if your current primary and secondary DNS servers are non-Barracuda Networks products and you want to use your CloudGen Firewall as your new primary DNS server, the serial number of the secondary DNS server might still be higher because it most probably uses a different numbering scheme for the serial number than the CloudGen Firewall. In such a case, you must add a serial number offset to the CloudGen Firewall's serial number so that the sum of the serial and the offset is at least one higher than the old serial number of the secondary DNS server.

For converting, use a time-stamp converter, e.g., https://www.unixtimestamp.com/.

The usage of the Serial Number Offset might become clearer in the following example:

DNS Server Type	BIND Numbering Scheme	Example Serial	Serial Offset
Secondary DNS Server	serial-update-method 'date'	Assumed time of last zone transfer from old primary to secondary DNS server: 2019-04-30, 00:00 YYYYMMDDnn, e.g., `2019043000`	-
New DNS Primary Server e.g., CloudGen Firewall	serial-update-method 'unixtime'	Assumed time for zone transfer from new DNS primary to old secondary DNS server: 2019-05-01, 00:00 Number of seconds since the Unix epoch January 1st 1970, e.g., equivalent to 2019050100 (YYYYMMDDhh) = `1556668800` `(unixtime)`	On the old secondary server, the Unix timestamp representation of the real time 2019-05-01-00 of the zone-transfer will be received as a numerical value of 1556668800. However, in order to accept new zone information, the secondary DNS server must receive a serial which is effectively 2019043001, so the difference between 2019043001 and 1556668800 must be transmitted as the serial offset. In this example, the value will be 462374201. `2019043001 - 1556668800 =` **`462374201`**

If you want to run your Barracuda CloudGen Firewall as a DNS primary server with non-Barracuda DNS secondary servers, you must set the serial number offset so that it is always higher than the serial of a secondary DNS serial at the moment when you click Send Changes /

Activate. Therefore, it is recommended to increase the difference by adding enough seconds to the serial number offset. This must only be done when doing a zone transfer for the first time.

Zone-to-Host Record Relation

When a primary zone is created, the CloudGen Firewall offers the option to automatically create the corresponding first two entries that are essential for setting up a base configuration. If this option is selected, two DNS records will be created automatically for the configured zone:

Domain	Type	Owner	TTL	Email	Record Data	Description
example.com (Master)			86400	root@example.com		
	NS	@	3600		ns1.example.com	Automatically generated
	A	ns1	3600		62.99.0.30	Automatically generated

The first record is of type NS and delegates the zone `example.com` to the name server with the hostname `ns1`. The second record is of type A and maps the hostname `ns1` to the DNS service's IP address.

When automatically creating these two records, the first manually configured service IP address is always used.

For some DNS records, the configuration window offers the option to automatically create reverse entries within the corresponding reverse zone.

Before a reverse DNS record can be created automatically, the reverse zone must be configured first. Otherwise, the reverse DNS record will be discarded.

For more information, see How to Configure a Zone.

Processing of Hostnames within Zones

For the new DNS implementation, Barracuda Networks has focused on an optimal compromise to address a correct handling of DNS-specific configuration data and a high level of usability.

Example:

- `host1.example.com` in the zone `example.com` will be expanded to be `host1.example.com.` (note the trailing colon) because the hostname `host1.example.com` conforms to the ending of the domain name `example.com`. where both have `example.com` in common.
- `host1.example.com` in the zone `example.at` will not be expanded but rather prepended to `example.at` resulting in `host1.example.com.example.at.` (note the trailing colon)

because the hostname and the domain name end with different partial names (`at` and `com`).

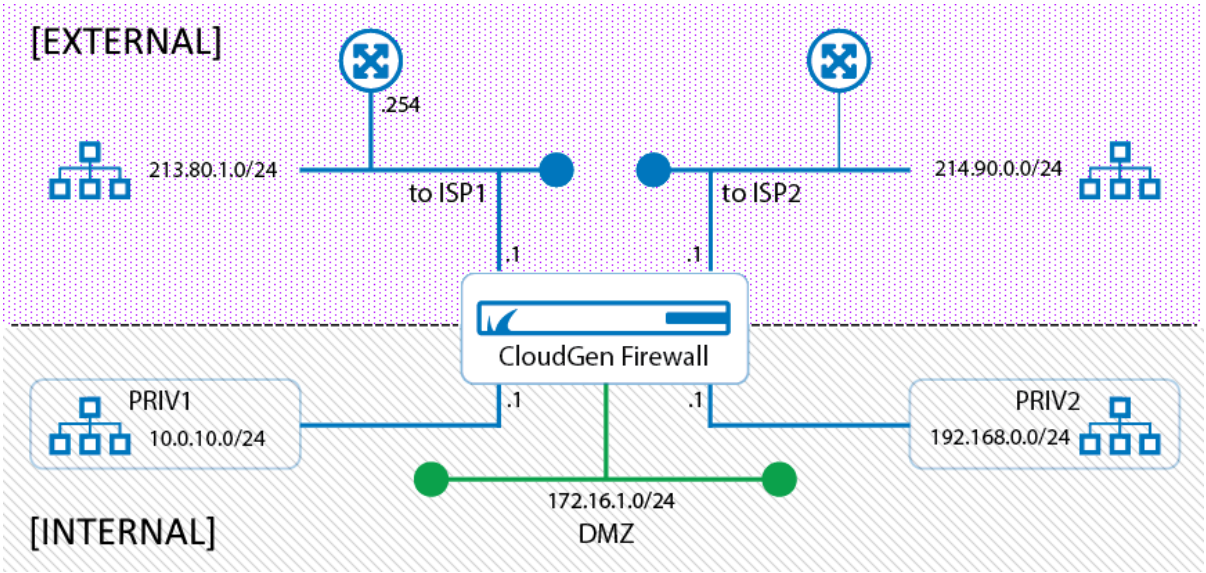
Split DNS: The Classification of DNS Service IPs / Listeners

The DNS service listens to incoming requests on all configured service IP addresses and handles them with the same priority. Sometimes, however, it is advantageous to respond to resolving requests from a LAN differently than from a WAN.

When creating an explicit **DNS Listener**, your selected IP address can be either an internal or an external IP address. You must therefore assign either the tag **INTERNAL** or **EXTERNAL** to such a special DNS service IP address. Such a listener is identified by its **Listener Name** and can later be used in a **DNS Record** together with its corresponding assigned options (EXTERNAL, INTERNAL, ALL, <user defined>):

- **EXTERNAL** – The set of DNS-served IP addresses is limited to public IP addresses.
- **INTERNAL** – The set of DNS-served IP addresses is limited to private IP addresses.
- **ALL** – EXTERNAL + INTERNAL.
- ***<User defined>*** – Choose any name that best describes the selected DNS service IP address.

The following figure shows a network setup with multiple IP addresses connected both to the WAN and the LAN. The subsequent table shows how the DNS listener considers special service IP addresses depending on the classification and **Listener Name**.



Configuration Tree...	Listener Name	Listener Classification	Example Service IPs Used by DNS Listener

...Assigned Services > DNS > Service Properties	-	-	213.80.1.1, 214.90.0.1, 10.0.10.1, 172.16.1.1, 192.168.0.1
...Assigned Services > DNS > DNS Service > DNS Settings > DNS Listeners Classification	INTERNAL	INTERNAL	10.0.10.1, 172.16.1.1, 192.168.0.1
	EXTERNAL	EXTERNAL	213.80.1.1, 214.90.0.1
	toISP1	EXTERNAL	213.80.1.1
	toISP2	EXTERNAL	214.90.0.1
	PRIV1	INTERNAL	10.0.10.1
	PRIV2	INTERNAL	192.168.0.1
	DMZ	INTERNAL	172.16.1.1

As you can see in the table above, the **Listener Classification** of any configured service IP address (e.g., toISP1, PRIV1, DMZ, ...) can only be either **INTERNAL** or **EXTERNAL**.

It is recommended to configure listeners prior to zones because they will be required during the configuration of DNS records within the zone.

Using DNS Listeners for Recursive Lookups from Remote Networks

The new DNS service is based on the commonly known BIND standard. In case a recursive DNS server is configured, the DNS service automatically configures empty zones. This prevents the firewall from sending meaningless queries to Internet servers that cannot handle them.

Note that this option is BIND-specific and therefore cannot be disabled when the firewall is configured to operate in recursive mode!

By default, the DNS service is listening to requests on all configured DNS service IP addresses. However, recursive DNS lookups are only answered for requests originating from a network that is directly configured on a local interface. DNS requests from a remote network (reachable only via a gateway) are not answered by default.

In case you want to permit answering requests from a remote network, you must configure an explicit listener for that network and allow recursive lookups:

Queries originating from a...	...direct-attached network	... remote network
Requires listener	• NO (recursive lookups allowed by default setup)	• YES
Response for hosted zones	• YES	• YES

Response for foreign (=non-hosted) zones	• YES	• NO (default) • YES: requires explicit listener + recursive lookups allowed

For more information, see How to Configure a DNS Listener.

Availability of Target Hosts (Health Probes)

On the CloudGen Firewall, health probes can be configured to test the reachability of certain targets. Health probes can be attached to a DNS resource record to confirm the validity of a configured IP address. In case the health probe confirms the reachability of a certain target, the associated IP address in the DNS resource record will be returned as a valid IP address for the query. If the configured target does not respond, the configured IP address will not be sent as a valid answer for the probed target.

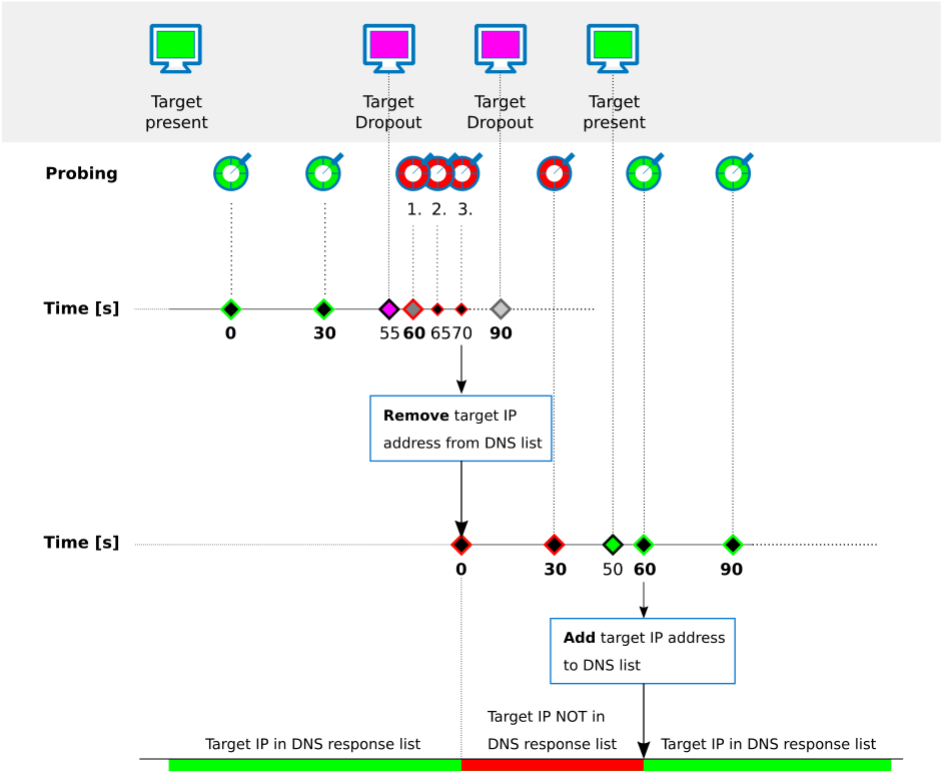
Probing provides the following options:

Probe Type	Annotation
ICMP	Probes a target using the ICMP protocol (ping). Suitable for general probing if only physical presence must be verified.
HTTP/S	Probes a target using the HTTP and HTTPS protocol. Suitable for web servers that either respond on port 80 or 443.
TCP + Port Number	Probes a target using the TCP protocol and a special port number. Suitable for any arbitrary combination of TCP protocol and port.

For more information, see How to Configure a Split DNS Setup and How to Configure a DNS Health Probe.

Probe Pattern and Timing

By default, probing is done every 30 seconds. If the target is reachable, the next probing is done in 30 seconds. If a probing fails, two subsequent probes will follow within the period of a few seconds. If the second probing in sequence (which effectively is the third probing within 30 seconds) also fails, the associated IP address will be removed from the list of valid IP addresses and will therefore not be sent in the response of valid IP addresses for the last resolving query. Probing will be continued for a period of 30 seconds until the target is available again. From this time on, probing will start over according to this pattern.



Responding Queries with Multiple IP Addresses, Simple Load Balancing, and Failover

If you are running multiple servers, e.g., web servers at different ISPs and/or redundant web servers in your DMZ, you must associate multiple IP addresses with the same hostname, e.g., www.example.com, 62.99.0.11, 212.86.0.11, 213.47.0.11. In case a query asks for the resolution of the web server's hostname (www.example.com), all IP addresses (62.99.0.11, 212.86.0.11, 213.47.0.11) will be sent to the querying client if no other options limit this rule. However, the order of the IP addresses will alter each time the same query is made. By default, this order applies to a cyclic rotation:

# Query	Query	Resolving Result Returned by DNS Server (first | second | third | fourth position in list)	Effective Load Balancing IP Address
1.	nslookup www.example.com	**62.99.0.11** | 212.86.0.11 | 213.47.0.11	62.99.0.11
2.	nslookup www.example.com	212.86.0.11 | 213.47.0.11 | **62.99.0.11**	212.86.0.11

3.	nslookup www.example.com	213.47.0.11 | **62.99.0.11** | 212.86.0.11	213.47.0.11
4. = 1.	nslookup www.example.com	**62.99.0.11** | 212.86.0.11 | 213.47.0.11	62.99.0.11

This cyclic rotation will cause the querying clients to contact another target each time after a resolution, thus distributing requests between multiple targets. When attaching a health probe to a set of configured IP addresses, failover scenarios can be handled in case a configured host becomes unavailable.

For more information, see How to Configure Simple DNS Load Balancing with Failover.

DNS Interception

DNS Interception allows redirection or blocking of DNS queries for specific domains. This is achieved by applying policies. When creating a policy, you can also specify allow-listing for certain domains.

For more information, see How to Configure DNS Interception.

DNS Default SOA Values

This new DNS system works with the following DNS default settings:

SOA Parameter	SOA Default Preset Values
SERIAL	Unix Timestamp + configurable offset
REFRESH	86400
RETRY	7200
EXPIRE	3600000
MINIMUM	7200

When migrating from a 7.x to a 8.x firmware version, all values configured as part of firmware 7.x will continue to be used instead.

Figures