CYGNA LABS

Protection against malware with DNS firewalls

By Timothy Rooney

About Cygna Labs

Cygna Labs is a software developer and one of the top three global DDI vendors. Many Fortune 100 customers rely on Cygna Labs' DDI products and services, in addition to its industry-leading security and compliance solutions to detect and proactively mitigate data security threats, affordably pass compliance audits, and increase the productivity of their IT departments. For more information, visit https://cygnalabs.com.

Introduction

Securing your network requires disciplined systems, processes and intelligence. Most security strategies rightly focus on data flows into and out of an organization in an attempt to prevent illicit intrusion and infiltration. However, you may not be protecting your network optimally if you are not considering attacks originating from within your network. For example, an attacker may attempt to install malware on devices within your network to enlist such devices as subject to the control of the attacker.

Such malware may be installed via phishing or spear phishing attacks that bait users into opening executable email attachments or installing software from an attacker website. Whether a device is attacked while inside the enterprise network or a user device is physically brought onto the network, if it is trusted within the confines of an enterprise network it may have access to sensitive information. The installed malware may perform data collection, locating internal resources using DNS reconnaissance. In addition, DNS could be used to identify the current IP address of the attacker's external destination for exfiltration of the information.

A DNS firewall can be deployed to protect your network and to help identify such infected devices. As malware attempts to locate its command and control center for instructions or software updates using DNS, the DNS firewall can stifle the attempt and prevent the communication. This white paper describes the mechanics of a DNS firewall and basic DNS deployment models with respect to functionally separating DNS resolution roles onto discrete DNS server implementations. This paper focuses specifically on the implementation of DNS firewall technology which provides the ability to apply special handling such as dropping certain DNS queries to known bad actor domains. DNS firewall technology enables enterprises to reduce the probability of malware proliferation within their networks.

What is a DNS firewall?

When you think of an Internet firewall, you likely think of a gateway device which examines IP packets flowing through it and which selectively blocks or redirects those packets meeting certain criteria. Such criteria may include filtering parameters such as IP addresses or ports such that when an IP packet under inspection matches such parameter settings, the packet is blocked or otherwise handled according to policy settings. A DNS firewall performs similar examination and policy handling functions for DNS queries to prevent unwelcome DNS and subsequent data traffic.

Another common assumption associated with Internet firewalls is that they are deployed on the perimeter of a network with the intention of protecting the network from attacks originating external to the network. DNS firewalls however protect the network against attacks that originate within the network. Why worry about internal attacks if morale is sky high and IP firewalls are seemingly impervious? Sophisticated phishing or targeted spear phishing attacks may trick unsuspecting associates into clicking on an attachment or a link to a disguised attacker website to download and install the malware. Plus with the proliferation of smart phones and "bring your own device" (BYOD) initiatives intentionally or unintentionally established, it's quite possible that devices physically leaving the domain of a perfectly firewalled network may elsewhere become infected with malware when operated on less secure networks such at as the coffee shop wifi or at home.

Certain forms of malware infiltrate a device as a remote agent or robot ("bot") which, along with several other similarly infected devices, forms a "botnet" where an attacker can command several bots to perform attacks such as distributed denial of service attacks. Recent high profile attacks using Mirai, Hajime, and Persirai malware strains attack Internet of Things (IoT) devices like cameras to serve as botnet participants. And additional attacks on Microsoft® Windows™ devices featuring the wannacry and petya ransomware variants also highlight the potential wide scale threat posed by malware.

Malware installed on an infected device will typically attempt to contact the attacker's command and control (C&C) center to receive its marching orders, and the means of contacting the C&C starts with a DNS lookup. The primary goal of a DNS firewall is to identify such C&C contact attempts, to block such attempts and to identify the infected device.

A leading DNS reference implementation, BIND from the Internet Systems Consortium (ISC) supports the establishment of DNS firewall policies via its response policy zones (RPZ) feature. Other vendors like PowerDNS, Microsoft, KnotDNS and NLNet also support or will soon support RPZ functions. RPZ enables a DNS administrator to define policies in standard DNS resource record format to enable filtering of DNS queries.

Filtering triggers can be defined based on the queried name (QNAME), querier IP address, resolved IP address (IP address within A or AAAA query response), resolving name server domain name (NSDNAME) as resolved within the query response Authoritative section, and resolving name server IP (NSIP) as provided within the query response Additional section. Thus throughout the resolution process for a particular query, the recursive DNS server can logically filter at multiple points along the way, then enact the corresponding policy action prior to sending the response to the client. Such action can be defined as responding with NXDOMAIN, NODATA, pass through, drop, truncation, or inclusion of predefined response data, such as directing the session to a walled garden.

The beauty of this technique is in defining policies as resource records within a zone or zones which enables DNS administrators to create their own policies and/or to subscribe to a provider or providers of malicious domain (filtering) information, which can simply zone transfer such domain information to the corresponding recursive DNS servers. Updates of this zone information of course should be secured via the use of access control lists (ACLs) as well as transaction signatures (TSIG) to sign incremental or full zone updates.

DNS Firewall Scenario

Let's walk through an example of how the DNS firewall works. Figure 1 illustrates the standard DNS query flow where a client issues a query to a local recursive DNS server which resides on your enterprise network, within your broadband service provider's network or your coffee shop. The recursive DNS server seeks the answer for your query by querying down the domain tree, from the root, to the .com domain in this case and so on down the tree.

In this particular scenario, let's say the querying device has been infected by malware. The malware bot will attempt to contact its botnet headquarters to request instructions or to transmit sensitive information. If the malware author's site is located at mal.bad-example.com, the laptop will issue such a query using standard DNS protocol. Upon locating the DNS server responsible (authoritative) for the bad-example.com domain thanks to a referral from the .com DNS server, the recursive server queries and obtains an answer from the malware author's DNS server, which maps the mal.bad-example.com site to IP address 192.0.2.24.

Under normal operation, the recursive DNS server would happily relay this answer back to the requesting malware and enable it to connect to the C&C center located at IP address 192.0.2.24. But here's where the DNS firewall comes into play. Before sending the response to the client, the recursive server looks up its configured response policies. In this case, it finds a policy that indicates for address (A record) query responses containing a question name (Qname) within the "bad-example.com" domain branch, revise the response to point to 172.16.200.1 (instead of 192.0.2.54 in this case).

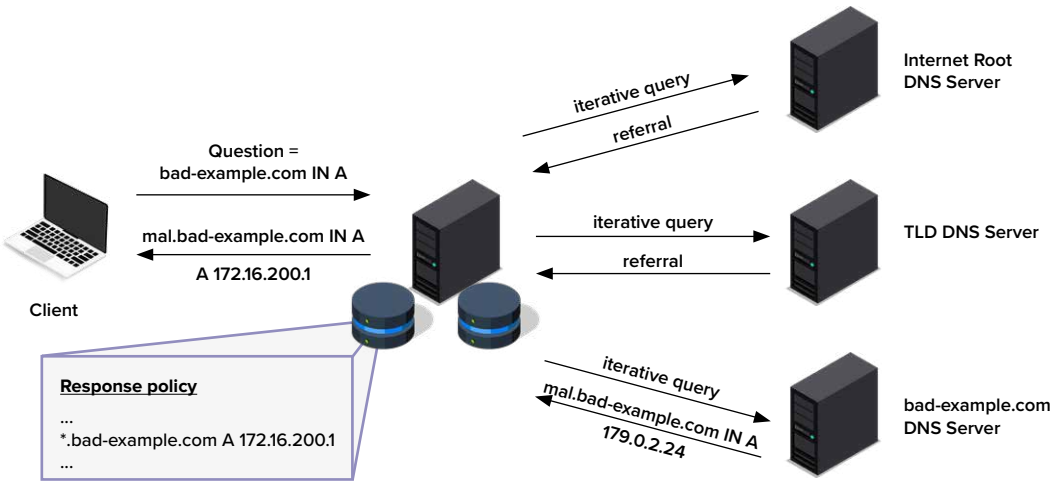

Figure 1: DNS firewall scenario

In this example, we see how the DNS server modifies the response provided to the client. In this case, the response directs the querier to a private IP address, where you can setup a remediation system or web captive portal. Other policies enable configuration of the DNS server to drop the response, send "not found" or "no data", and more as we'll discuss next.

DNS Firewall Deployment

Deployment of DNS servers within an enterprise must account for availability and performance requirements. In addition, basic security practices recommend deploying sets of DNS servers in support of particular roles in resolving domain names. This can reduce the breadth of the impact should a given server be attacked. In this case, we'll associate roles with deploying DNS servers to address the following query types for your name space. Each type of query is labeled by letter in Figure 2 for reference.

A. Queries from internal clients for internal hosts

B. Queries from internal clients for external or Internet hosts

C. Queries from external or Internet clients for internal hosts

Let's briefly review these scenarios to identify where deploying DNS firewalls applies.

A. Internal Authoritative Queries

Starting on the left side of Figure 2, queries from internal hosts for destinations within the internal name space, e.g., internal email servers, intranet servers, printers, should be resolved authoritatively by internal DNS servers configured with internal zone and resource record information. These queries (labeled A) are authoritatively resolved by the internal DNS servers for internal name space. Figure 3 illustrates this more clearly with those servers not impacted by these types of queries are "grayed out."

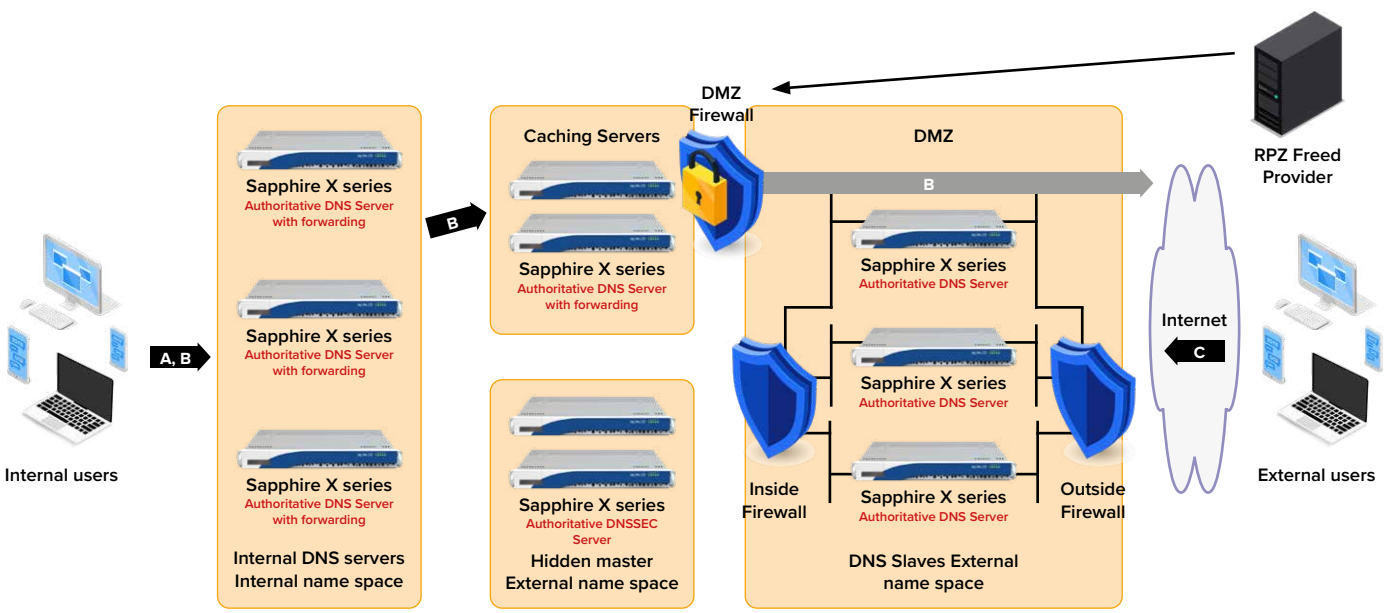

Figure 2: DNS Deployment Roles

B. Recursive Queries

Starting on the left side of Figure 2, queries from internal hosts for destinations within the internal name space, e.g., internal email servers, intranet servers, printers, should be resolved authoritatively by internal DNS servers configured with internal zone and resource record information. These queries (labeled A) are authoritatively resolved by the internal DNS servers for internal name space. Figure 3 illustrates this more clearly with those servers not impacted by these types of queries are "grayed out."

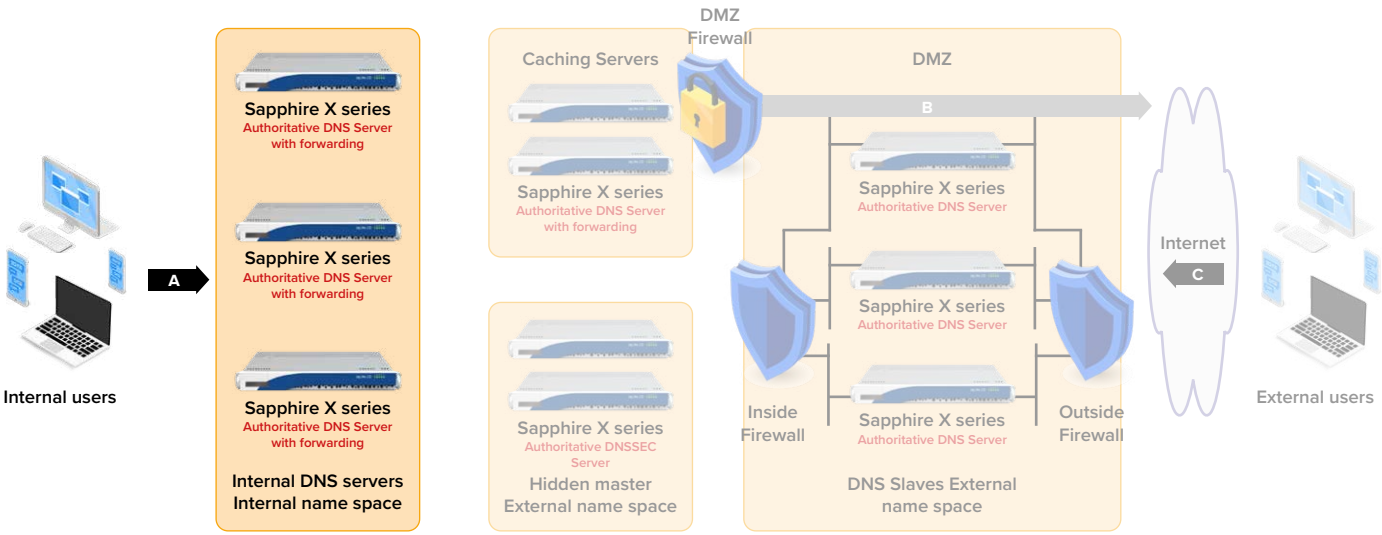

Figure 3: Scenario A - Internal queries for internal name space

Some organizations funnel outbound queries though a set of caching servers as depicted in Figures 2 and 4 in order to reduce the number of sources for outbound DNS queries (and inbound answers) and to build up a richer cache of queried data over time. This is illustrated by the forwarding of queries from internal DNS servers to the caching servers located in the middle of the figure. These caching servers then leverage cached resolution data or resort to the root hints file to traverse the domain tree in search of the query answer. This two-stage outbound resolution path traverses the DMZ and is shown as the "B" arrow in the figure.

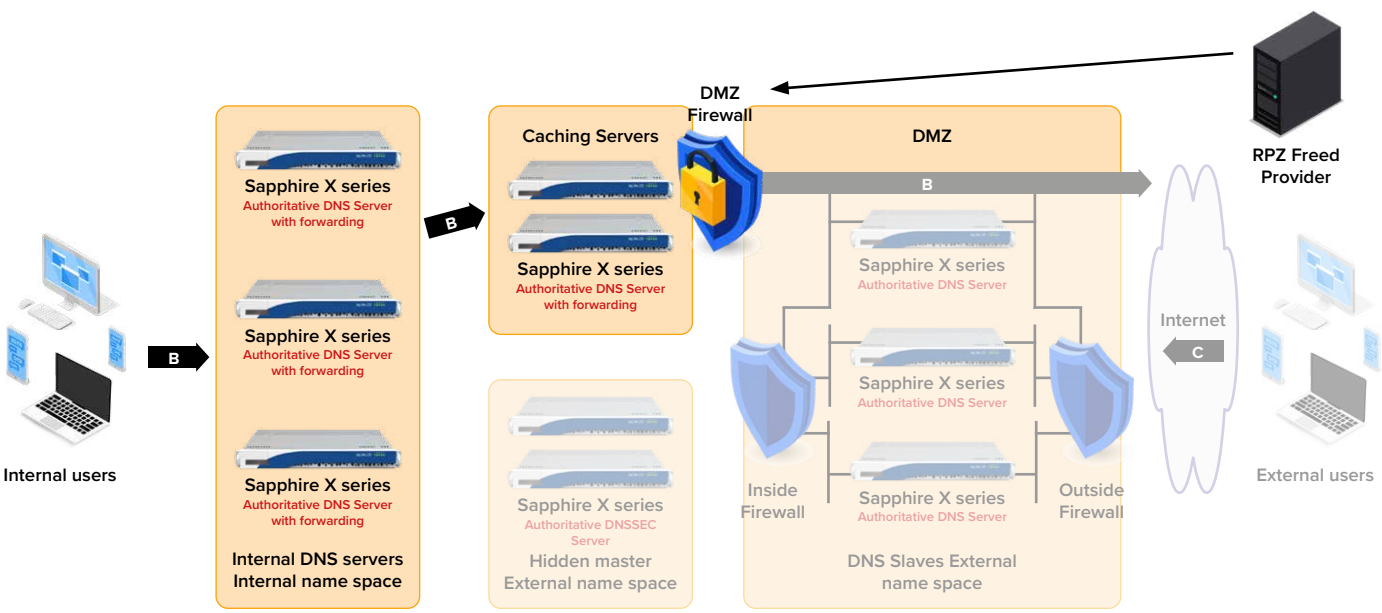

Figure 4: Scenario B - Internal queries for external name space

As depicted in Figure 4, a DNS firewall can be implemented at the caching server level to apply response policies to Internet-bound queries and responses. Alternatively, the DNS firewall function can be deployed at the first recursive server layer, i.e., the servers shown in the left box of Figure 4. Deploying the DNS firewall function closer to end clients affords the ability to detect the end client device which made a given query, e.g., which triggered the firewall policy. Such detection can be configured via logging setup as we'll discuss later. Detection of the end device provides the necessary detail to mitigate the potentially infected device. The DNS firewall at the caching server level only enables logging of the IP address of the forwarding recursive server in the absence of additional information.

C. External Queries

External DNS servers are configured with resolution information for hosts that are accessible from the Internet, e.g., your web, email, and other public applications. DNS queries from the Internet should not be permitted beyond the DMZ. The slave DNS servers in the DMZ should be authoritative for all externally published destinations. The master server should be “hidden” and inside the interior firewall (not queried by Internet servers). The master server is hidden by not publishing it among the NS records in this zone and the parent zone.

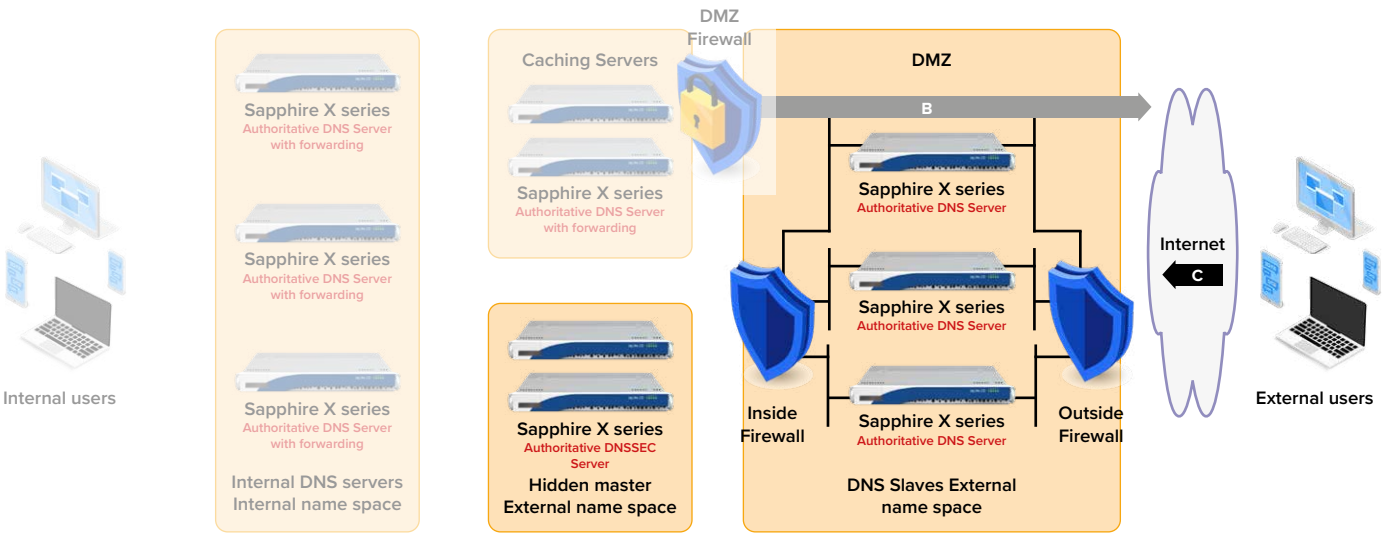

Figure 3: Scenario A - Internal queries for internal name space

When Diamond IP Sapphire Sx20 or Sx10D appliance(s) are deployed as external authoritative master DNS servers, zones can be automatically signed to enable DNSSEC signing of your public name space. The master Sx appliance supplies signed zone information to authoritative slave DNS servers deployed within the DMZ for querying by Internet clients.

Summary

Of these basic deployment scenarios, DNS firewall functionality applies to case B. The firewall is applied after completion of the query recursion process to determine the disposition of the query response to the client.

DNS Firewall Configuration Details

DNS firewall policies are configured on recursive DNS servers to apply policies to query responses to known bad domains. Response Policy Zones (RPZ) functionality, which provides the DNS firewall functionality, is supported in Cygna Labs Diamond IP's Sapphire appliances as illustrated in Figure 3 or on stock BIND servers. RPZ information can be updated periodically using an RPZ data provider such as Cygna Labs Diamond IP via signed zone transfers. RPZ feeds can be generated and managed internally and/or by leveraging Cygna Labs Diamond IP's DNS Firewall subscription service.

DNS Firewall Rules

DNS firewall rules are defined in the language of DNS servers, that is, resource records within zone files. Specially coded resource records each indicate a trigger and a corresponding policy. In the example of Figure 1, the trigger was an IPv4 address (A record type) question (Qname) of *.bad-example.com, which interpreting the * wildcard, indicates any Qnames within the bad-example.com domain branch, i.e., any Qname suffixed with bad-example.com. The corresponding policy applied for responses matching this trigger indicate setting the answered IP address to 172.16.200.1.

The full set of currently defined trigger types are:

- Qname match - the queried domain name (or within a domain branch if specified with a wildcard prefix) matches the owner field of the RPZ resource record. Specifically, the name field and resource record type fields corresponding to the question section of the query/response are matched.
- RPZ-IP - query answer IP address (or block) matches the IP address in the general format: <prefix length>.<reversed IP address>.rpz-ip. For example, to match an answer with an IP address from the 192.0.2.0/24 block, this would be encoded as 24.0.2.0.192.rpz-ip. An IPv6 address can be similarly encoded with the abbreviation of "zz" for a double colon abbreviation; for example one could match an IPv6 address in the answer from the 2001:db8:4b30::f0/128 as 128.f0.zz.4b30.db8.2001.rpz-ip.
- RPZ-NSIP - authoritative name server IP address, encoded similarly as rpz-ip matches but with the rpz.nsip suffix.
- RPZ-NSDNAME - authoritative name server domain name, e.g., bad.example.com.rpz-nsdname.
- RPZ-CLIENT-IP - client IP address formatted as above with prefix length and reversed IP address and with the rpz-client-ip suffix.

Triggers are matched on a best fit basis. That is all records with triggers matching the query or answer are sorted by the best fit (e.g., longest prefix match for IP addresses) or most granular match; the policy corresponding to such a best match trigger is then applied.

For each defined trigger, a policy can be defined to answer the query with either:

- NXDOMAIN – indicates an answer for the queried name was not found
- NODATA – signified by a NOERROR response with no query answers (a zero "answer count"); this indicates a queried name was valid but no data for the queried type was found
- pass-through - no alteration of the response
- drop – provide no answer to the query
- TCP-only – respond with the truncated (TC) bit set in the DNS header to stimulate the client to requery over TCP
- local policy such as directing the querier to portal or walled garden to initiate remediation or to display a web page indicating an invalid query or possible infection.

Each of these policies and corresponding triggers can be defined within response policy zone files. The BIND response-policy statement enables association of these zone files as RPZs. This statement also enables policy overrides per file (e.g., disable all policies defined in the file for troubleshooting for example, respond to all triggers with NODATA, etc.). This statement also enables the specification of overriding parameters for RPZs including:

- **recursive-only** – this policy indicates PRZ processing shall only apply to recursive queries; this policy may be applied to an individual zone or all response policy zones.
- **max-policy-ttl** – this statement enables setting of the time to live (TTL) of the response resource record. The TTL dictates how long the client should cache this response. Once the cache expires, the client will issue a query once again should the client request this information. This enables rapid refreshing of policies if desired. This policy may be applied to an individual zone or all response policy zones.
- **log** – This parameter applies only to individual zones and indicates whether RPZ policy hits should be logged. Logging must also be configured per the Logging Configuration section.
- **break-dnssec** – by default, RPZ is not applied to queries where DNSSEC validation has been requested. Given that DNSSEC authenticates the publisher and data integrity of the DNS response, any change to the response per RPZ policy would by definition destroy the message integrity and invalidate the DNSSEC signature. Setting this parameter to "yes" instructs the server to perform RPZ processing even on DNSSEC queries, permitting the server to "break DNSSEC."
- **min-ns-dots** – This parameter applies globally and stipulates the number of "dots" in the queried domain name that must exist to apply RPZ processing. A value of "1", the default, would apply to example.com for example
- **qname-wait-recurse** – Normally, the recursive server performs full recursion to ascertain the query answer prior to seeking a response policy action. Setting this parameter to "no" configures the server not to await recursion, since it receives the Qname with the query from the client, and to apply any defined response policy for the Qname immediately.
- **nsip-wait-recurse** – Normally, the recursive server performs full recursion to ascertain the query answer prior to seeking a response policy action. Setting this parameter to "no" instructs the server to access its cache for the Qname and, if it exists, the corresponding answering DNS server IP address. If the IP address is defined as a RPZ-NSIP trigger, apply the corresponding policy immediately, bypassing full recursion.

DNS Firewall Policy Precedence

When an ISC DNS server loads a zone file, including response policy zones, it reorders the RRSets in canonical order. While a "longest prefix" style trigger match is applied to each query response, the first detected such longest match will dictate the corresponding policy to be applied. You may desire to block most records in a domain but define pass through records for known reputable records with that domain or a child domain.

While the ordering within a file is canonical and not the order in which you enter the data, you can specify multiple zone files, up to thirty-two, to which to apply response policies. The order in which you define these zones is honored such that the policy is applied corresponding to the longest match in the first file detected. Thus you can list pass through policies first, then list blocking policies in separate zone files.

Logging Configuration

It's important to configure DNS logging to receive notification of RPZ policy triggers. Such notification can inform your security team of a potentially malware-infected device for rapid remediation. ISC BIND supports logging configuration by specifying a logging channel(s) and then directing pre-defined logging category events to respective channel(s). Logging channels may be in the form of server log files, syslog, standard error or null. The "rpz" logging category applies to DNS queries that trigger an RPZ policy. Thus configuring rpz category events to direct output to a syslog channel would provide the ability to collect RPZ policy event notifications via syslog. Log processing logic could optionally be applied using a third party log collector such as Splunk to perform additional actions such as alerting.

A Simpler Approach

This plethora of parameters affords great flexibility in defining DNS firewall policies and overrides, though they may be challenging to configure properly. Configuring these response policy zones, resource records and logging configuration properly requires precision entry of policies for each trigger element you identify as worthy of processing. Cygna Labs' Diamond IP product portfolio can streamline your DNS firewall implementation so you can more efficiently secure your network from malware. Our web graphical interface supports simple entry of response policy zone parameters, file specifications and logging configuration. Our DNS firewall subscription service keeps your firewall configuration refreshed and up-to-date with the latest detected malware domains.

Additional Security Considerations

A DNS firewall is an important component of not only your DNS security strategy but also your overall network security strategy. It adds a strong defensive layer within your defense in depth approach. In addition to DNS firewall configuration, recursive DNS servers should also be configured to perform DNSSEC validation to enable validation of signed resource record sets. This requires configuration of the current Internet root zone public key (DNSKEY KSK record data). Recursive servers can also be configured to automatically maintain this trust anchor public key as the root zone rolls.

Other recursive security settings should also be configured to provide maximum availability and reliability to the DNS recursive service. Such settings include setting ACLs to constrain from what IP networks or addresses recursive queries may be made for example, configuring denial of service rate limiting, restricting the number of outstanding queries per client and limiting the query depth to minimize the chasing of bogus queries. These settings are easily configured using Cygna Labs Diamond IP products.

Summary

Cygna Labs Diamond IP's IP address management products support configuration of DNS firewall functions via its web user interface for our Sapphire appliances as well as stock ISC BIND servers you may already operate. We also provide a DNS firewall subscription service comprising a regularly updated feed of bad domain information which can be easily configured with our systems though customers are free to implement their own policies using our systems or use other or additional bad domain providers.

Toll Free: **(844) 442-9462**
International: **+1 (305) 501-2430**
Fax: **+1 (305) 501-2370**

Sales: **sales@cygnalabs.com**
Support: **support@cygnalabs.com**
Billing: **finance@cygnalabs.com**

cygnalabs.com

CYGNA LABS