IP-Based Actions

IP-based actions are similar to BIND 9's RPZ-IP triggers and corresponding actions. They are similar to the existing local-zone or other tag-based actions, but defined for particular IP-netblocks. If the IP(v6/v4) address of an AAAA or A in the answer section matches one of the specified netblocks, the corresponding action will apply to th nse. For example, it may change the address to a diffe e (redirect) or make the resolution result in NXDOMAIN.

This note describes details of prop mplementation of IP-ba ions in Unbound.

Configuration Sy

IP-based actions requ ew (op Unbound module plac ve r (see the implementation note b So th -c fig optio sp explicitly:

```
m d le-c  f    e    "
```

If DNSSEC-validation osed it should h

```
m d le-c  f    e  f
```

We also introduce thr configura e -da a
e e-i - ag: options a **ser** se

response-ip: <IP- e bl
If the IP address in an AAAA or A RR answ atch specified IP-netblock, the specified ac semantics as that for acce -c g-ac are exc (see below).

response-ip-data: <IP- e bl ck> < ce ec g >
This specifies the action data for e e- ith acti edi ec ."r record string" is similar to that of acce -c but it must b either AAAA, A, or CNAME. If the IP ock is an IPv6/IPv4 AAAA/A, respectively unless it's CN which can be used for bo netblocks). If it's CNAME, there mu e more than one e e-i -da a fo the same IP-netblock. Also, CNAME ther types of records must not coexist for the same IP-netblock. The textual doma e for the CNAME does not have to be explicitly terminated with a dot (.); th name is assumed to be the origin for the name.

response-ip-tag: <IP- e bl ck> < f ag >

Assign tags to respon cks. If the IP s AAA RR in t answer sectio spec k, th fied ta assigned to th cce g is d for t n and it includes onse c onding acce -c ly. Tag c e is the sam acce -c . Unlike e- ag, e e-i fine i is defined for tha

e e-i and e a c so be specified for a view, just like l cal- e and l cal-

If multiple e e-i - for the same IP-netblock (in different lines), all but the one that appear ll be d (following the similar case for acce -c l- ag). f it's r figuration error, it's quite unlikely that the result is the intend a nd would be meaningless in practice.

The e e-i - opt require th i module; if any of these options are specified but the e i module nbound will refus)load the configuration. In fact, some part modifying c answers) would work without the e i module, but such an i r wou h us even harmful. It should be bette olici figura ir

Exceptions on respons

Actions specified for e e dif m for lo i the former there is no point o n as " nes it ca Because of this differenc s follows:

- a ic, ef e d for e e-i y s ecifie ic ie
- de is non-condit i.e., it a sults resolution result before a ed n be used for queries.

On the other hand, actions specified in a -c l ac that has a matching tag with e e-i - ag hose that are d" for res ip listed above, since acce -c l- ag-a be share n local zo actions, if they behave differently depen or not in case of local zones, the behavior for response-ip-tags e e of no corresponding e e-i -da a w nerally result in NOERROR/NODATA instead of NXDOMAIN, since the response-ip data are inh type specific, and non-existence of data doesn't indicate anything about the existence o the qname itself. For example, if the matching tag action is static but there is no data for the corresponding response-ip

configuration, then the result will be NOERRO DA e only case where NXDOMAIN is returned is when an `al a _ d mai` actio lies.

Notes on Detailed Behavior

No AA Bit

If the original answer is tweaked due to a response-ip action, the final response sent to the client will never have the AA (authoritative answer) bit on in the header section. This is different from some cases of local-zone and access l-tag actions (faked NXDOMAIN and local/redirect data), where the final respons nsidered "authoritative" and has the AA bit on. This behavior is consistent with BIND 9 RP in that case it's not specific to RPZ-IP triggers).

Answer Records Only

IP-based actions apply only to records in t wer section. Even if the IP address of an AAAA or A record in the additional (or, unli practice, authority) section matches an IP-netblock of some `e e-i` option e records (or the response itself) won't be modified.

Multi-RR Case

If an AAAA or A RRset in the r section of a query contains multipl an r more of the addresses are subject onse-ip processing, the action for th ma address in the answer message will b sively used. For example, if an ans AA t contains the following IPv6 a in this order:

```
2001:db8::1
2001:db8::2
2001:db8::3
```

and the following IP-based a are configured:

```
e     e-i : 200   ::1/128 al a  _  d mai
e     e-i : 200   /128  edi ec
e     e-i -da a:          A 2001:db8::bad
```

then the first matching action (alway y and the response will be converted to NXDOMAIN (with no answer reco re are multiple matching redirect actions for different IP-netblocks of the same answer RRset, only the first matching redirect data will be used, and the resulting response will contain only one address (which is the matching redirect data). Since it appears that Unbound caches an RRset in the order it receives from the upstream server, the same action will be applied to subsequent responses to the same query when they are directly answered from the cache.

This behavior largely follows the behavior of BIND 9's RPZ-IP triggers. Note, however, that the order of the answer RRs from remote servers is not always predictable, so it is also

unpredictable which e e-i action is d when there are multiple candidates. This is different from BIND 9 RPZ-IP e the triggers a lied to sorted RRs by default and the matching rule is generally pre e for the sam

CNAME Chasing

If a CNAME is specified as th or an IP-based action, Unbound will automatically chase the CNAME target unti l answer (whether positive or negative, or some other error), and return the co E chain to the end client. For example, for the above original AAAA answer efine a redirect response-ip action as follows:

```
e     e-i : 200          8  edi ec
e     e-i -da a.         :1/128 CNAME  a ge .e am le.
```

then Unbound will resolve AAA etrieve it from the cache), and include the result in the final answer at n example final answer is:

```
<  igi al   ame>          e .e am
 a ge .e am le.
```

Unlike “CNAME-based redire zone actions, AME chasing will take place even if the redirect data is used for ontrol-tag-act t is not redirect but can use local data (e.g. static). For examp to the above uration, if there’s a following tag setup:

```
acce  -c     l-          : 192.0.2.1/      "m  ag"   a ic
 e     e-i - ag:  :db8          128 "m  ag"
```

then, if the client at 192.0.2.1 sends redirect data for the corresponding response-ip-da target will be chased (in retrospect, this would have been more intuitive for the local zone case, too).

Even if the CNAME target RRset has a valid RRSIG, the RRSIG won’t be included in the final response to the client, nor the AD bit will be set in the response, regardless of whether or not the original query sets the DO or CD bit (see also “DNSSEC implications” below). This behavior is consistent with BIND 9.

Type-ANY query suppresses this chasing. For instance, in the above example configuration if the query type is ANY, the answer will only contain the CNAME RR. This behavior is consistent with BIND 9.

Note: Chasing CNAME targets for an IP-based action may be especially expensive in terms of performance (see implementation notes below). It’s probably advisable to avoid this configuration whenever possible.

No Recursive Application

This proposed implementation does not try to apply IP-based actions “recursively”; that is, it does not apply an IP-based action to the IP address specified as a result of data of a redirect

response-ip action. For examp[redacted]e apply a redirect action for `2001:db8::3/128` used in the previous example a[redacted]o `2001:db8::bad`. Then, even if there is another IP-based action which `2001:d[redacted]tches, that action won't apply. (This is the same as BIND 9's RPZ-IP.)

The same restriction applies to[redacted]dress of a CNAME when the CNAME is the data of an IP-based action. Fo[redacted]the example of the CNAME chain shown a[redacted]

```
<  igi al   ame> CN[redacted]e .e am le.
 a ge .e am le. AAAA 2[redacted]ffff
```

even if there is another IP-based actio[redacted]matches, that action apply. In this case, only the CNAME will be returned to the client in order to avoid inclu IP address in the answer (in this example, `2001:db8::ffff`) that would otherwise be to an IP-based action. It will also help avoid having a weird corner cases like a CNAME (e.g., consider the case where `2001:db8::ffff` is redirect to `<  igi al   ame>` [redacted]. Note that a sophisticated client (such as a local caching server using this Unbound as a "forwarder") could re-query for the CNAME target when it gets the incomplete CNAME chain. this case the intended IP-based action will apply and that client will get an answer that the administrator of this action would probably envision. For traditi[redacted]b resolve[redacted]n an incomplete CNAME chain effe[redacted]r means r[redacted]n failure. [redacted]not ideal, [redacted]would be acceptable in practice as in m[redacted]ses the pr[redacted]ntent of su[redacted]ction wou[redacted]o avoid returning a specific IP addres[redacted]

In any case, such a situation [redacted]idered a l[redacted]configurati[redacted]r, and Un[redacted]eaves an informational level of log mes[redacted]hen it det[redacted]e situation:

```
CNAME  a ge   f [redacted]ec   e [redacted]-i  ac i[redacted] ld be [redacted]c
 e     e-i  ac i[redacted]  ;   [redacted]
```

This is also consistent with BIND 9 R[redacted]to CNAME targets if the CNAME comes from an RPZ.

Similar to the above cases, even if there is a local-zone action to which `a ge .e am le` would be subject, it won't apply. In this case the above AAAA RR will be included in the answer to the client (it's no different from how a redirected CNAME target in local-zone or tag-based actions works in general).

This is different from BIND 9 RPZ, which never applies RPZ rules more than once. In this implementation we defer from it to keep the implementation simpler, but we may want to revisit it (this is also different from what's written in draft-vixie-dns-rpz).

No Override for Other Loca

Similar to the previous subse sed actions will not apply t er
local-data or tag-based actio mple, if we have the follow t

```
l cal-   e: e am      edi ec
l cal-da a:  e a  c m    A 192.0.2.1
 e     e-i : 192.0  0/2
```

then a query for example.com/A will wered from t l-z h RD f
192.0.2.1. Even if it would match the -i c
answer. This is consistent with BIND 9 RPZ.

Authority and Additional Sections

If an AAAA or A RR of the answer section of a response ifie to an IP-based action, the authority and additional sections of the resulting fina nse client will be cleared (an EDNS OPT RR may still be added to the additional s). T the case regardless of the action type, even if it's `edi ec` or `d mai` varian This is ferent from BIND 9 RPZ: it adds an SOA of the corresponding RPZ to the authority tion fo itive answers and to the additional section for NXDOMAIN.

DNSSEC Implications

If an original response is modified due to an IP-based action, the resulting final response will never have the AD bit on even if the original response was DNSSEC-validated. Any RRSIG RRsets for the modified RRset will be removed from the answer section; however, if the original response is a CNAME chain and some of the CNAMEs have RRSIGs, these RRSIG will be kept in the final response (note tha MEs can never be subject to an IP-based action). This behavior is compatible with B RPZ.

Multi-Level Netblock Match

We will (eventually) try to imp multi-level matching: if the best (longest) matching IP-prefix does not have a matching tag client but a less-specific matching IP-netblock has a matching tag, the action for th specific IP-netblock should apply. For example, if we have the following two response-ip nfigurations:

```
e     e-i - ag:  0.2.128/28 " ag1  ag2"
e     e-i - ag:  55/32 " ag3"
```

and if an A RR in the answer section 2 255 but tags for the client include "tag1" but not "tag3", then the action 28 should apply even if the best matching netblock is `192.0.2.255/32`.

This ideal behavior is consistent with how tags for local-zones match. But this may need non-trivial extensions to existing Unbound utilities, while such a multi-level setup is supposed to be quite rare. So we may skip this behavior and always consider the best matching netblock (if there is no matching tag, treat it as there is no matching netblock) in the initial implementation.

This is a **TODO** item as of this writing (as of February 10, 2017, this “ideal behavior” is not implemented).

Type ANY Query

The primarily intende[redacted] to apply them to type AAAA or A queries. But it should also wor[redacted] answer contains an AAAA or A record. This is consistent with BIN[redacted]. [redacted] of the original answer will be removed except the one that t[redacted] or which it may be replaced (in case of a `edi ec` action) o[redacted] types of actions including[redacted] variants). This beha[redacted]

Inform Log

If an `i f m` or `i f` [redacted] a message will be logge[redacted] information about the [redacted] similar to log messages [redacted] and `i f m_de` [redacted] the m[redacted] netblock instead of the [redacted] message is as follows:

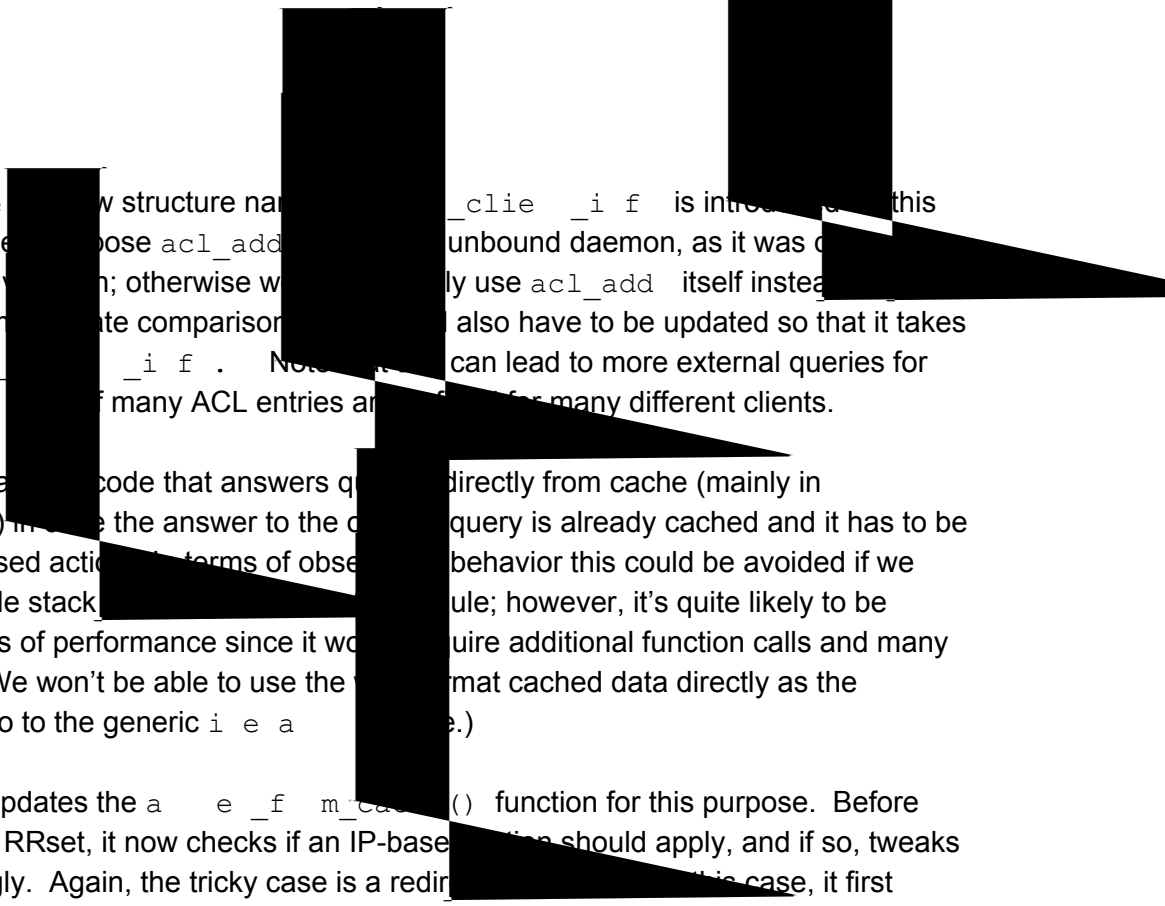
```
i f : 192.0.2.[redacted]5005 i f.e am le.c [redacted]A
IN
```

Overview of Implementation Design

Unlike other local-zone and tag-based actions, whether to apply an IP-based action cannot be determined completely locally since it depends on the result of normal resolution. Also, since the “CNAME chasing” behavior will require additional sub-queries, we cannot just tweak the answer immediately before sending it to the client.

So we chose to implement the main functionality as a separate Unbound module named `e i` (meaning “response IP”). It’s intended to be placed on top of the module stack (usually immediately abo[redacted]iterator module) and works as a filter for resolution results. Its `e a e()` fun[redacted]st passes the state to the lower module to resolve the original query. When it’s done, [redacted]ks the answer section for AAAA or A records that may require IP-based action processin[redacted]ome action needs to be applied, the module modifies the answer accordingly (cha[redacted]he IP address to a different one, converting the whole answer to NXDOMAIN, etc[redacted]usually completes the module. The exception is the case where a redirect to CNAM[redacted]which c[redacted]ub-c[redacted] triggered to resolve the CNAME target, and the original query st[redacted]eld unt[redacted]mplet[redacted] completion of the sub-query, the module’s `i f m_ e()` [redacted]ction is [redacted]. It a[redacted]s the resolved target records to the (already tweaked) answer sectio[redacted]al a[redacted]. Finally, the control comes back to the `e a e()` function, w[redacted]e module processing.

The `e i` module will need [redacted]acce[redacted]part of `acl_add` of the client that triggers a particular query so that it can a[redacted]g-ba[redacted] per-view actions. So we’ll need to extend the existing `me h_ e _clie ()` [redacted]tion so [redacted]kes the sub part of `acl_add` and stores it in



the m d le_ a e w structure nar _clie _i f is intro this purpose (we didn't like pose acl_add unbound daemon, as it was c be module-boundary w n; otherwise w ly use acl_add itself instea new structure). The m ate comparison also have to be updated so that it takes into account e i _ _i f . Note can lead to more external queries for the same qname and many ACL entries ar for many different clients.

We also need to upda code that answers q directly from cache (mainly in daem / ke .c) in the answer to the c query is already cached and it has to be modified by an IP-based acti terms of obse behavior this could be avoided if we always call the module stack ule; however, it's quite likely to be unacceptable in terms of performance since it wo uire additional function calls and many more data copies. (We won't be able to use the mat cached data directly as the resolution needs to go to the generic i e a e.)

We implementation updates the a e _f m_ca () function for this purpose. Before encoding the answer RRset, it now checks if an IP-base tion should apply, and if so, tweaks the answer accordingly. Again, the tricky case is a redir t is case, it first needs to see if the CNAME target is cached, but to do so it will first make a local copy of the original RRsets, tweaks the answer, and releases acquired locks (it needs to release the lock for the additional cache lookup, and so it needs to make a copy of the cache data as the reference to the cache is not protected by the cache). Also, if the CNAME target is not cached, it will behave as if the original cache lookup failed to trigger the usual resolution (and processing by the e i module). These are complicated and inefficient, but seem to be unavoidable costs.