ipsets fqdn resolving

Concept

- Use a Sqlite3 database to store the list of entries that need to be updated.
- A script run on an interval will perform lookups for each fqdn, no more frequent than the DNS record's TTL
  - `next_lookup_dt` is set after a lookup to the current time plus the lookup's ttl in seconds
    - `UPDATE entries SET last_lookup_dt = datetime('now'), next_lookup_dt = datetime('now', '+$ttl seconds') WHERE fqdn=$fqdn and ipset=$ipset;`
  - Query for a list of entries that need to be checked due to expired ttl of last lookup
    - `SELECT * FROM entries WHERE next_lookup_dt <= datetime('now');`

We have two options for dealing with changed IPs.

The first involves creating a new ipset, swapping it with the current in-memory set, and then destroying the old ipset. After lookups are performed, the current list of IP addresses in the database would be used to create a new ipset list which would then be swapped and the old list destroyed. This would be effective only if this lookup script is the only thing manipulating the ipset list.

The second involves locating and deleting the entry that needs to be updated. This allows for additional processes to manipulate the list. The output of ipset list would be searched for an entry where the command matches the fqdn lookup. That IP address would be removed from the list using an ipset del command and then the new IP addresses added with the appropriate comment set.

Code

```
# CREATE the sqlite3 database and table(s)
sqlite3 /opt/ipsets-dynamic/ipsets-dynamic.db <<EOF
CREATE TABLE entries (
    fqdn TEXT NOT NULL,
    ipset TEXT NOT NULL,
    last_ip_address TEXT,
    last_lookup_dt DATETIME DEFAULT (datetime('now')),
    next_lookup_dt DATETIME DEFAULT (datetime('now')),
    last_change_dt DATETIME DEFAULT (datetime('now')),
    added DATETIME DEFAULT (datetime('now')),
    comment TEXT,
    PRIMARY KEY (fqdn, ipset)
);
EOF
```

Example

Assuming the ipset list exists as below:

```
ipset list unifisiteallowlist -output json | grep -v initval
```

```
# OUTPUT
[
  {
    "name" : "unifisiteallowlist",
    "type" : "hash:net",
    "revision" : 7,
    "header" : {
      "family" : "inet",
      "hashsize" : 1024,
      "maxelem" : 65536,
      "comment" : true,
      "bucketsize" : 12,
      "memsize" : 1800,
      "references" : 1,
      "numentries" : 10
```

```
    },
    "members" : [
      {
        "elem" : "1.2.3.4",
        "comment" : "resolve:some-fqdn.domain.com"
      },
   ]
  }
]
```

```
result=$(ipset list unifisiteallowlist -output json | grep -v initval | jq -r '.[].members[] |
select(.comment | contains("resolve:some-fqdn.domain.com")) | .elem')
echo $result
```

```
# OUTPUT
1.2.3.4
```

The value of result would be empty if there was no match found.

```
#!/bin/bash

ENTRIESDB='/opt/ipsets-dynamic/ipsets-dynamic.db'

# PROCESS ALL RECORDS
sqlite3 -separator '|' $ENTRIESDB "SELECT fqdn, ipset, last_ip_address FROM entries;" |
while IFS='|' read -r fqdn ipset ip; do
  RECORD_LINE=$(dig +noall +answer "$fqdn" | grep "IN\s*A")
  if [ -n "$RECORD_LINE" ]; then
    TTL=$(echo "$RECORD_LINE" | awk '{print $2}')
    ADDRESS=$(echo "$RECORD_LINE" | awk '{print $5}')

    if [[ "$ip" != "$ADDRESS" ]]; then
      echo "Entry: $fqdn / $ipset / $ip / $ADDRESS / $TTL - Update needed"
      # FIND AND DELETE EXISTING IPSET ENTRY
      echo "ipset del $ipset $ip"
      # CREATE NEW IPSET ENTRY
      echo "ipset add $ipset $ADDRESS comment \"resolve:$fqdn\""
      # UPDATE DATABASE
```

```
      echo "sqlite3 $ENTRIESDB \"UPDATE entries SET
last_ip_address='$ip',next_lookup_dt=datetime('now','+$TTL seconds') WHERE fqdn='$fqdn' and
ipset='$ipset';\""
    else
      echo "Entry: $fqdn / $ipset / $ip / $ADDRESS / $TTL - NO CHANGE"
    fi
  else
    echo "Entry: $fqdn / $ipset / $ip / NO A RECORD FOUND"
  fi
done
```

```
#!/bin/bash

ENTRIESDB='/opt/ipsets-dynamic/ipsets-dynamic.db'

# PROCESS ONLY TTL EXPIRED RECORDS
sqlite3 -separator '|' $ENTRIESDB "SELECT fqdn, ipset, last_ip_address FROM entries WHERE
next_lookup_dt < datetime('now');" |
while IFS='|' read -r fqdn ipset ip; do
  RECORD_LINE=$(dig +noall +answer "$fqdn" | grep "IN\s*A")
  if [ -n "$RECORD_LINE" ]; then
    TTL=$(echo "$RECORD_LINE" | awk '{print $2}')
    ADDRESS=$(echo "$RECORD_LINE" | awk '{print $5}')

    if [[ "$ip" != "$ADDRESS" ]]; then
      echo "Entry: $fqdn / $ipset / $ip / $ADDRESS / $TTL - Update needed"
      # FIND AND DELETE EXISTING IPSET ENTRY
      echo "ipset del $ipset $ip"
      # CREATE NEW IPSET ENTRY
      echo "ipset add $ipset $ADDRESS comment \"resolve:$fqdn\""
      # UPDATE DATABASE
      echo "sqlite3 $ENTRIESDB \"UPDATE entries SET
last_ip_address='$ip',next_lookup_dt=datetime('now','+$TTL seconds') WHERE fqdn='$fqdn' and
ipset='$ipset';\""
    else
      echo "Entry: $fqdn / $ipset / $ip / $ADDRESS / $TTL - NO CHANGE"
    fi
  else
```

```
    echo "Entry: $fqdn / $ipset / $ip / NO A RECORD FOUND"
  fi
done
```

Revision #5
Created 2026-02-04 22:11:14 UTC by bluecrow76
Updated 2026-02-04 23:28:57 UTC by bluecrow76