dnsdist DNS proxy

dnsdist is a highly configurable DNS-, DoS- and abuse-aware loadbalancer.

Here's an example configuration file:

```
-- File: /etc/dnsdist/dnsdist.conf

-- listen for console connection with the given secret key
controlSocket("0.0.0.0:53530")
setKey("supersecretAPIkey")
setConsoleACL({"172.16.16.0/24","192.168.168.0/24","10.10.20.0/24"})

-- start and configure the web server
webserver("0.0.0.0:8053")
setWebserverConfig({password="supersecretpassword", apiKey="supersecretAPIkey"},
acl="172.16.16.0/24,192.168.168.0/24,10.10.20.0/24")

-- accept DNS queries on UDP/53 and TCP/53
addLocal("0.0.0.0:53")
-- accept DNS queries on UDP/5200 and TCP/5200
-- addLocal("0.0.0.0:5200")

-- fix up possibly badly truncated answers from pdns 2.9.22
truncateTC(true)

-- Log message
warnlog(string.format("Script starting %s", "up!"))

-- define the server pools
-- public-google
newServer({address="8.8.8.8", pool="public-google", checkInterval=300})
newServer({address="8.8.4.4", pool="public-google", checkInterval=300})

-- public-cloudflare
newServer({address="1.1.1.1", pool="public-cloudflare", checkInterval=300})
```

```
-- public-quad9
newServer({address="9.9.9.9", pool="public-quad9", checkInterval=300})
newServer({address="149.112.112.112", pool="public-quad9", checkInterval=300})

-- internal pools
newServer({address="192.168.1.53",  pool="company1-auth", checkInterval=300})
newServer({address="192.168.2.53",  pool="company2-auth", checkInterval=300})
newServer({address="192.168.3.53",  pool="company3-auth", checkInterval=300})

-- local router
newServer({address="172.16.16.254", pool="router", checkInterval=300})

newServer({address="127.0.0.1:53531", pool="nodeapp1", checkInterval=300})

-- switch the server balancing policy to round robin,
-- the default being least outstanding queries
setServerPolicy(roundrobin)

addAction({"camera.project1.loc.", "device.project1.loc."}, PoolAction("nodeapp1"))
addAction({"company1.loc"}, PoolAction("company1-auth"))
addAction({"company2.loc"}, PoolAction("company2-auth"))
addAction({"company3.loc"}, PoolAction("company3-auth"))

addAction(AllRule(), PoolAction("public-google"))
-- addAction(AllRule(), PoolAction("public-cloudflare"))
-- addAction(AllRule(), PoolAction("public-quad9"))
-- addAction(AllRule(), PoolAction("router"))

-- refuse all queries not having exactly one question
-- addAction(NotRule(RecordsCountRule(DNSSection.Question, 1, 1)), RCodeAction(DNSRCode.REFUSED))

-- return 'refused' for domains matching the regex evil[0-9]{4,}.powerdns.com$
-- addAction(RegexRule("evil[0-9]{4,}\\.powerdns\\.com$"), RCodeAction(DNSRCode.REFUSED))

-- spoof responses for A, AAAA and ANY for spoof.powerdns.com.
-- A queries will get 192.0.2.1, AAAA 2001:DB8::1 and ANY both
-- addAction("spoof.powerdns.com.", SpoofAction({"192.0.2.1", "2001:DB8::1"}))

-- spoof responses will multiple records
-- A will get 192.0.2.1 and 192.0.2.2, AAAA 20B8::1 and 2001:DB8::2
```

```
-- ANY all of that
-- addAction("spoof.powerdns.com", SpoofAction({"192.0.2.1", "192.0.2.2", "20B8::1", "2001:DB8::2"}))

-- spoof responses with a CNAME
-- addAction("cnamespoof.powerdns.com.", SpoofCNAMEAction("cname.powerdns.com."))

-- spoof responses in Lua
--[[
    function spoof1rule(dq)
        if(dq.qtype==1) -- A
        then
                return DNSAction.Spoof, "192.0.2.1"
        elseif(dq.qtype == 28) -- AAAA
        then
                return DNSAction.Spoof, "2001:DB8::1"
        else
                return DNSAction.None, ""
        end
    end
    function spoof2rule(dq)
        return DNSAction.Spoof, "spoofed.powerdns.com."
    end
    addAction("luaspoof1.powerdns.com.", LuaAction(spoof1rule))
    addAction("luaspoof2.powerdns.com.", LuaAction(spoof2rule))

--]]

-- alter a protobuf response for anonymization purposes
--[[
function alterProtobuf(dq, protobuf)
    requestor = newCA(dq.remoteaddr:toString())
    if requestor:isIPv4() then
        requestor:truncate(24)
    else
        requestor:truncate(56)
    end
    protobuf:setRequestor(requestor)
end

rl = newRemoteLogger("127.0.0.1:4242")
```

```
addAction(AllRule(), RemoteLogAction(rl, alterProtobuf))
--]]
```

-end

Revision #2
Created 26 July 2024 02:31:37 by bluecrow76
Updated 26 July 2024 02:50:27 by bluecrow76