Install BIND for DNS and ad-blocking on CentOS 7

This post is for installing BIND on CentOS 7 to use as a local DNS server, and including a pre-configured zone file containing over 2000 of the most common ad server domain names. Basically what will happen is when you visit a webpage that sends a request to one of these ad servers, it will redirect that request to a webserver running on this DNS server, and serve up a transparent 1×1 pixel gif file. That means the DNS request prevents the HTTP/S request from ever leaving your network, causing webpages to load a tad faster, and provide a 99.9% ad-free browsing experience!

In my setup I have SELinux disabled (sed -i /etc/selinux/config -r -e 's/^SELINUX=.*/SELINUX=disabled/g'). If you disable selinux (or firewalld), you do so at your own risk. It is not advisable to do either if your server is directly accessible on the internet (this one should not be), so please do so at your own risk! At the bottom of this post I've included the simple rules needed for firewalld.

Let's Get Started!

Installation von Updates und Voraussetzungen

```
# yum -y update && yum -y install bind httpd wget
```

Konfiguration des bind Service

```
# mv /etc/named.conf /etc/named.conf.orig
# vim /etc/named.conf
```

```
acl "trusted" {
    172.16.1.0/24;
};

options {
    listen-on port 53 { any; };
    listen-on-v6 port 53 { ::1; };
    directory   "/var/named";
    dump-file   "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query     { any; };

    recursion yes;
    allow-recursion { trusted; };
    listen-on { 172.16.1.101; };
    allow-transfer { none; };
```

```
    dnssec-enable no;
    dnssec-validation no;
    dnssec-lookaside auto;

    bindkeys-file "/etc/named.iscdlv.key";
    managed-keys-directory "/var/named/dynamic";
    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";

    forwarders {
    8.8.8.8;
    8.8.4.4;
    };
    forward first;

};

logging {
        channel default_debug {
                 file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
    type hint;
    file "named.ca";
};

zone "domain.local" IN {
    type master;
    file "domain.local.zone";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
include "/var/named/blacklists/ads";
```

There are a few things in this config that you will need to customize to your environment.

1. **First**, at the top is an **`acl config`** with currently shows 172.16.1.0/24. You will need to update this to your network.
2. **Second**, you will need to find and **`change the listen-on IP`**, and set that to the IP of the server you're installing on.
3. **Third**, the above config has the **`forwarders`** server set to Google's DNS servers. If you're fine with that, no change is needed. If you want to change that, just replace the two IP's there with your ISP's DNS servers, or others.
4. **Lastly**, you will need to **`update the “zone” config`**, which is set to domain.local. Using

blitz.home as an example, it should look like this:

```
zone "blackgate.home" IN {
    type master;
    file "blackgate.home.zone";
};
```

Erstellen der ad-block Zone

```
# vim /var/named/null.zone.file
```

```
$TTL 86400
@   IN  SOA dns01.domain.local. hostmaster.domain.local. (
            2016010100  ; serial
            21600       ; refresh after 6 hours
            3600        ; retry after 1 hour
            604800      ; expire after 1 week
            86400   )   ; minimum TTL of 1 day

        IN  NS  dns01.domain.local.

@       IN  A   127.0.0.1
*       IN  A   127.0.0.1
```

Above is the zone file that is used to redirect ad requests. **Again, you need to make changes for your environment.**

1. In the second line, you will need to update to your domain. As example, you would want to set it to: `@ IN SOA dns01.blackgate.home. hostmaster.blackgate.home. (` Leave hostmaster as is!
2. Then you will need to change dns01.domain.local to the domain name of this server (dns01, in my example) you will need to update line 9 as well, to someting like: `IN NS dns01.blackgate.home.`

Erstellen der DNS lokalen Forward Zone

When creating the next file, **replace domain.local.zone with your domain name as specified in the named config in step #2;** i.e. blackgate.home.zone. Otherwise, named will not be able to find this file and start the service.

```
# vim /var/named/domain.local.zone    # CHANGE THE FILE NAME AS DIRECTED
ABOVE
```

```
$TTL 86400
@   IN  SOA dns01.domain.local. hostmaster.domain.local. (
            2015122100  ; serial
```

```
            21600       ; refresh after 6 hours
            3600        ; retry after 1 hour
            604800      ; expire after 1 week
            86400   )   ; minimum TTL of 1 day

        IN  NS  dns01.domain.local.

dns01       IN  A   172.16.1.101
fw01        IN  A   172.16.1.254
dns         IN  CNAME   dns01
```

Just like in the ad-block zone file, you will need to **update line 2** and **line 9** with your relevant domain info.

Down toward the end of the file I have two examples of how to manually add DNS records (A records). Just use this same format if you want to add any of your own. The bottom line is for creating a CNAME record for dns, and pointing it to my server named dns01. If this server is not named dns01 for you, replace dns01 with the correct name.

Now if you ever go back and update this file (which is normal), update the serial (line 3) so named will know to reread this file since it's been updated.

The serial is 10 digits, and is best used in a date format (i.e., YYYYMMDDVV, where Y=Year, M=Month, D=Day, V=Version). Just restart the named service after changes have bene made.

Herunterladen des AD-Blacklist Zonen Files

Here we are just making a directory, and downloading a pre-configured zone file with over 2000 ad domains listed, and renamining the file to ads.

```
# mkdir -p /var/named/blacklists
# wget -O /var/named/blacklists/ads
'http://pgl.yoyo.org/adservers/serverlist.php?hostformat=bindconfig&showintro=0&mimetype=plaintext'
```

Konfiguration des DNS lokalen HTTPD

```
# wget -O /var/www/html/a.gif
'http://probablyprogramming.com/wp-content/uploads/2009/03/tinytrans.gif'
```

Above we downloaded the 1×1 transparent gif file that we'll serve up instead of ads. Next we need to update the httpd config with rewrite rules to know when/how to serve the file. We need to open the file for editing, and just add in a few lines inside the <Directory “/var/www”> section.

```
# vim /etc/httpd/conf/httpd.conf
```

```
<FilesMatch "a.gif$">
```

```
    Header set Cache-Control "max-age=290304000, public"
</FilesMatch>

    RewriteEngine On
    RewriteBase /
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_FILENAME} !-l
    RewriteRule ^(.*)$ http://localhost/a.gif
```

Konfiguration Firewalld, sowie Starten und enablen der Services

```
# systemctl enable named.service && systemctl enable httpd.service   #
Enable services
# firewall-cmd --permanent --add-port=53/tcp
# firewall-cmd --permanent --add-port=53/udp
# firewall-cmd --permanent --add-service=http
# firewall-cmd --reload

# systemctl reboot
```

From here you just need to configure your client computer to use this server as it's DNS server, and you should then be ad free! If it's not working right, or you're having problem, just me know and I'd be glad to help out!

Last update: **2017/09/29 15:47**