Guy Bruneau – guybruneau@outlook.com Twitter : @GuyBruneau

Build Securely a DNS Sinkhole
Step-by-Step Powered by Slackware Linux

By Guy Bruneau, GSE
Version 2.1 – 23 October 2016

1. DNS Sinkhole Overview

This configuration process is used to deploy DNS sinkhole powered by the Slackware Linux (GNU) operating system. This streamline installation was developed to easily setup either a DNS Bind or PowerDNS forwarder and act as local DNS sinkhole when the requested site is held in the local tables or database. The full installation using this setup is ~800 MB in size and provides no remote services except through Secure Shell and Webmin for remote management of the sensor and the server.

This installation has a web management interface called Webmin which is used to remotely manage the server via a SSL enabled web browser. Additional information about Webmin is available at: http://www.webmin.com/.

A **minimum of 2 GB of RAM** is recommended but if your DNS Sinkhole are serving a large enterprise, more is likely necessarythe better.

Important: Before you start, make sure you are disconnected from the network until the sensor has been securely configured.

1.1 *Installation, Configuration and Partitioning the Drive*

Drive partitioning can be done in multiple ways. This is an example for a server setup.

1.1.1 DNS Sinkhole Server Installation

Boot on the system using the Slackware CD-ROM. To partition the drive, login as root and run *cfdisk /dev/hda* (IDE drive), *cfdisk /dev/sda* (SCSI drive) or *cfdisk /dev/cciss/c0d0* (Raid drive). If this isn't a new drive, delete the old partitions before starting. Before you create the partitions, you must decide if you are going to use Bind or PowerDNS for your Sinkhole.

The custom interface provided for PowerDNS, doesn't require any command line knowledge to manage and update the Sinkhole. The other advantage, it keeps statistics when a site, domain or TLD was added to the sinkhole and by whom.

Suggested Drive Configuration

/	25 GB (Recommended minimum)
swap	5 GB
/var/log/named	10+ GB (If planning to log Bind DNS logs) **OR**
/usr/local	10+ GB (If planning to use PowerDNS, for MySQL)
/LOG	Remainder of the drive (If planning to collect sinkhole packets)

- hda1: / = 25 GB (Select new, select primary, size is 50000, beginning, bootable)
- hda2: SWAP = 5 GB or same amount as the RAM (Select Pri/Log Free Space, new, primary, size is 5000, beginning)
- Change hda2 to swap by selecting *type 82*
- hda3: / = 10 GB (Select new, select primary, size is 10000, beginning)
- hda4: Remainder *(Select Pri/Log Free Space, new, primary, remainder of disk for DNS database)*
- Select Write to save the new settings to disk
- Select *Quit* to exit

1.1.2 Install the Software

Now that you have partitioned the drive according to your specifications, and saved your setting, you are ready to setup the Operating System. This is a basic setup with Bind and running the Socat application to capture the redirected DNS Sinkhole web traffic queries.

- Run setup
- Select addswap
- Continue with installation: yes
- Check swap partitions for bad blocks (It's a choice here): yes or no
- Swap partition configured
- Select Linux installation partition
        - /dev/hda1 (format, ext4 - default)
        - /dev/hda3 (format, ext4 - default)
        - /dev/hda4 (format, ext4 - default)
- Select mount point for /dev/hda3: /var/log/named → **Needed to collect Bind logs**
                                   /usr/local → **If using PowerDNS for DB table**
- Select mount point for /dev/hda4: /LOG → **Needed to collect packet**
        - Select add none and continue with setup
- Select *continue* to go to the SOURCE section
- Select *1* to install from a Slackware CD-ROM (Note: It is searching for the CD)
- Select *OK* on continue

- Scanning for CD or DVD Drive, Select *auto* and *Ok* to continue

- Install DNS Sinkhole from the installation CD which only shows 8 packages:

A, AP, D, L, N, TCL, X, Z

- Select *OK* to continue and to go the *INSTALL* section
- Select install everything (full) and *Ok* to start the installation
- Insert Next Disk, select *Quit*
- *Skip* making a bootable USB stick
- Install LILO and select *expert*
    - Select Begin, at the blank prompt press enter, select *no*, Select Ok for Default
      Buffer Console, install to *MBR* confirm location to install lilo (select default
      @/dev/hda, /dev/sda or /dev/cciss/c0d0) and none
    - Add Linux and choose the root partition with the ***** (i.e. /dev/hda1, /dev/sda1 or
      /dev/ccisss/c0d0p1) – That is usually the default
    - Use *Linux* as a partition name
    - Install LILO
- Configure the network with your settings with static IP (select 127.0.0.1 for DNS)
- Setup DNS Sinkhole startup services (press spacebar to enable/disable)
    BIND DNS configuration: Select rc.bind service only.
    PowerDNS configuration: select rc.pdns, rc.pdns_recursor, rc.httpd, rc.mysqld
    Select OK → *Enter*
- Setup the hardware clock (*NO* to set local time zone and pic your TZ or *YES* for UTC)
- Setup the root password
- After the setup is completed, at the Slackware Linux Setup screen, select *<Cancel>*
- Remove CD (**eject**)
- Reboot (*reboot* at the prompt)
- Manually eject the CD-ROM (if using VMware)
- Log back into the server as root
- Delete residual mail *rm /var/spool/mail/root*

1.2 *Sinkhole Configuration*

- If needed, use *netconfig* to reconfigure network (IP, netmask, gateway, DNS)
- Use 127.0.0.1 for local DNS
- Configure NTP (crontab -e and change *time-a.nist.gov* to your own timeserver)
- If you are using multiple interfaces with PowerDNS (i.e. eth0, eth1), edit
/etc/httpd/extra/http-ssl.conf and add *Listen DNS_sinkhole_IP:443* (Listen
192.168.25.5:443). This will ensure the PowerDNS webserver is listening only on the
primary interface.

1.2.1 Configure Bind as DNS Sinkhole

Guy Bruneau – guybruneau@outlook.com Twitter : @GuyBruneau

If you are planning to use only Bind to run your DNS Sinkhole, you must do the following:

- Ensure that Bind is running (rc.bind)

- Edit /etc/named.conf   (Note: **//** is a comment in this file)
        - If needed, change the allow transfer
        - If needed, change the allow recursion
        - Change the list of forwarder to your site list

- Ensure your list of **include** domains matches your site custom lists. This is important when the sinkhole_parser.sh script test the zones for errors and duplicate. Any custom lists you wish to add to your sinkhole (i.e. guy_blacklist.conf) must be included in the named.conf file to be loaded in the sinkhole. The default list is:
        site_specific_sinkhole.conf (single = match specific domain)
        entire_domain_sinkhole.conf (wildcard = match entire domain)

- Save the changes

1.2.1.1 Configure Bind RNDC with rndc-confgen

You need to create a rndc key file and configuration file. rndc provides command line tool *rndc-confgen* to generate it.

- Execute *rndc-confgen*

Copy this top section to /etc/rndc.conf

```
# Start of rndc.conf
key "rndc-key" {
        algorithm hmac-md5;
        secret "iknI44sXg3gnCuXDj5MMIg==";
};

options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
# End of rndc.conf
```

Copy this section to /etc/rndc.key

```
key "rndc-key" {
        algorithm hmac-md5;
        secret "iknI44sXg3gnCuXDj5MMIg==";
};
```

- Restart Bind - /etc/rc.d/rc.bind restart
- rndc status

1.2.1.2 BIND DNS Sinkhole Setup - Hijack and Sinkhole Domains

- Edit the `/var/named/sinkhole/client.nowhere` and change the 192.168.1.5
IP address to your site sinkhole IP address and save the change.

- Edit the `/var/named/sinkhole/domain.nowhere` which is used to wildcard
an entire domain and change the 192.168.1.5 IP address to your site sinkhole IP address
(this maybe the same as client.nowhere) and save the change.   (wildcard = *.domain.ca)

By default, the *sinkhole_parser.sh* script populates the
`site_specific_sinkhole.conf` and all domains included in this file are putting
in the sinkhole just the listed sites.

1.2.2 Testing the Bind Service

- Restart the server
- netstat -an |grep 53 should show something like this:

```
tcp         0      0 192.168.25.51:53        0.0.0.0:*                    LISTEN
tcp         0      0 127.0.0.1:53            0.0.0.0:*                    LISTEN
tcp         0      0 127.0.0.1:953           0.0.0.0:*                    LISTEN
udp         0      0 192.168.25.51:53        0.0.0.0:*
udp         0      0 127.0.0.1:53            0.0.0.0:*
```

- Test your sinkhole using nslookup
        - nslookup www.google.ca

1.3 *PowerDNS DNS Sinkhole Setup*

If you are planning to use PowerDNS to run your DNS Sinkhole, you must do the
following:

- Ensure pdns_server (rc.pdns) and pdns_recursor (rc.pdns_recursor) are running
- Change directory to /usrc/etc
        - Edit recursor.conf

- Verify the network configuration matches your network requirements
- If you want to use your ISP DNS server list or you are an enterprise that wants to use Split-DNS forwarding, enable (uncomment) and update the *forward-zone-recurse* option. If you want to use the one provided, just uncomment.
- Save the changes and exit

- If you want to monitor the DNS server statistics, change directory to /usr/etc
- Edit pdns.conf
- Configure your statistical PowerDNS webserver per 3.2.3
- Save the changes and exit

1.3.1 PowerDNS Forwarding Configuration

First of all, PowerDNS is not ISC Bind. "It is written from scratch and conforms to all relevant DNS standards documents."[1] PowerDNS has two configuration file. The first is located at */etc/powerdns/recursor.conf* and the second is located at */usr/etc/pdns.conf*

In order for PowerDNS to forward the queries to the web, you need to edit */usr/etc/recursor.conf* and enable forwarding.
Uncomment **# forward-zones-recurse=.=4.2.2.2;4.2.2.1** or change the forwarded DNS servers to your own.

1.3.2 Testing the PowerDNS Service

- Restart the PowerDNS services
  - /etc/rc.d/rc.pdns restart
  - /etc/rc.d/rc.pdns_recursor restart
- netstat -an |grep 53 should show this:

```
tcp        0        0 127.0.0.1:5300          0.0.0.0:*                   LISTEN
tcp        0        0 0.0.0.0:53              0.0.0.0:*                   LISTEN
udp        0        0 127.0.0.1:29233         127.0.0.1:5300             ESTABLISHED
udp        0        0 127.0.0.1:5300          0.0.0.0:*
udp        0        0 0.0.0.0:53              0.0.0.0:*
```

- Test your sinkhole using nslookup
   - nslookup www.google.ca

---

[1] http://www.powerdns.com

2. DNS Sinkhole Web Interface

This web interface doesn't require any use of the *sinkhole_parse.sh* script. Everything is managed via the web interface. To access the custom website, do:

https://sinkhole/

Login with the default user name and password of **admin** which can be changed after the initial login the server. The following options will show up on the top of the page
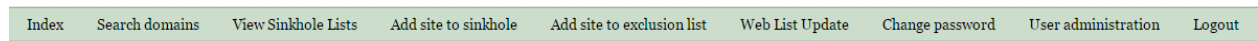
### 2.1 *Configure Local Sinkhole Address(es)*

Before being able to load any of the *Web List Update* where a number of site are pre-defined, you need to change the default *Web Sinkhole Lists* sinkhole IP address to your site web list sinkhole address.

Important: You must change under View Sinkhole Lists the default IP address of *192.168.25.6* to something you will be monitoring at your sites. If you don't care, that can be set to 127.0.0.1 → Localhost. By default, IPv6 is set to localhost (::1). Change both default Lists.

Edit sinkhole list "Web List"

	Name	Type	Content	Priority	TTL
✏️ 🗑	[ZONE]	SOA	localhost.[ZONE].com. ro		600
✏️ 🗑	[ZONE]	NS	[ZONE]		3600
✏️ 🗑	[ZONE]	AAAA	::1		3600
✏️ 🗑	[ZONE]	A	192.168.25.6		3600

Hint:

The following placeholders can be used in template records

   * [ZONE] - substituted with current zone name
   * [SERIAL] - substituted with current date and 2 numbers (YYYYMMDD + 00)

Save as new template:

Template Name [ ]
Template Description [ ]

[Commit changes] [Reset changes] [Save as template] [Update zones]

Name [Web List]

Description [Master List for Web Dom]

[Commit changes]

[Delete zone template]

The DNS Sinkhole is already populated with a list of sites. The first time you populate your DNS Sinkhole, you need to select the *List Name* and check *Sites Added Force* to load the list. You may need to select *Sites Added Force* if there is a wide gap between the previous update and the current update, the interface will let you know. To populate your sinkhole server, select *Web List Update*:

	List Name	List Url	Last Updated	Sites Added	Force
☑ 🖫 ☐	SANS ISC Low	https://isc.sans.edu/feeds/suspiciousdomains_Low.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	SANS ISC Medium	https://isc.sans.edu/feeds/suspiciousdomains_Medium.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	SANS ISC High	https://isc.sans.edu/feeds/suspiciousdomains_High.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	Malware Domain	http://mirror1.malwaredomains.com/files/justdomains	1970-01-01 00:00	0	☐
☑ 🖫 ☐	Malware Domain List	http://www.malwaredomainlist.com/hostslist/hosts.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	Zeus Tracker	https://zeustracker.abuse.ch/blocklist.php?download=baddomains	1970-01-01 00:00	0	☐
☑ 🖫 ☐	Malc0de	http://malc0de.com/bl/ZONES	1970-01-01 00:00	0	☐
☑ 🖫 ☐	Bambenek C&C	http://osint.bambenekconsulting.com/feeds/c2-dommasterlist.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	CryptoWall Ransomware C2 domain	http://ransomwaretracker.abuse.ch/downloads/CW_C2_DOMBL.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	TeslaCrypt Ransomware C2 domain	http://ransomwaretracker.abuse.ch/downloads/TC_C2_DOMBL.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	Locky Ransomware C2 domain	http://ransomwaretracker.abuse.ch/downloads/LY_C2_DOMBL.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	Locky Ransomware Payment Sites domain	http://ransomwaretracker.abuse.ch/downloads/LY_PS_DOMBL.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	TorrentLocker Ransomware C2 domain	http://ransomwaretracker.abuse.ch/downloads/TL_C2_DOMBL.txt	1970-01-01 00:00	0	☐
☑ 🖫 ☐	TorrentLocker Ransomware Payment Sites	http://ransomwaretracker.abuse.ch/downloads/TL_PS_DOMBL.txt	1970-01-01 00:00	0	☐

Update Web List   **Select All**

Add New Web List

All the domains under these lists are entered as hostname (not wildcard). This means if www.google.ca is in one of the list, it will be sinkhole as a hostname not a domain (i.e. *.google.ca).

Every time you want to update a list, select the checkbox for the *List Name* you want to update. Sometime you may get an **ERROR!** message and must check *Sites Added Force* to update that list.

This:1659
Last:1116
Processed SANS ISC Low successfully
Added: 0 new sites
Deleted: 0 existing sites
Sites in other lists: 0
Already had: 0 sites
Covered by wildcard: 543 sites
Covered by exclusion: 0 sites
Out of: 0 that were in that list

ERROR!   ✖

Quantity of sites more than doubled. List needs to be manually verified before continuing - TorrentLocker Ransomware Payment Sites Could not process TorrentLocker Ransomware Payment Sites

2.2 *Populating Site Exclusion List*

You can populate the DNS Sinkhole to never add certain sites to the DNS Sinkhole (i.e. your enterprise domain name *.google.com). By default, the DNS Sinkhole doesn't have any site loaded in the exclusion list. Sites added to this list can be viewed via the Webmin interface in MySQL, PDNS and exclusions table.

This will prevent any sites ending in *.thisdomain from being added to the sinkhole

Domain: *.google.com

Add Site

2.3 *Adding New Lists*

To add a new list that isn't part of the default, you may have to remove some unwanted data before the website is parsed. For example the Malc0de list has **zone** before the website name, to remove zone you will need to add a filter that is indicated with a forward slash "/" between the regular expression filter.

zone "192-168-0-255.com" {type master; file "/etc/namedb/blockeddomain.hosts";};
zone "airkuanta.com" {type master; file "/etc/namedb/blockeddomain.hosts";};

Edit Web List "Malc0de"

Name	Malc0de
URL	http://malc0de.com/bl/ZONE!
Parse String	/zone/	TEST

Commit changes

Another example would be to filter 127.0.0.1 before the domain name. The regex to use would look like /127\.0\.0\.1/ to filter it out. To make sure it is working, hit the TEST button to ensure the first field is the domain name.

Edit Web List "Malc0de"

Name	Malc0de
URL	http://malc0de.com/bl/ZONE!
Parse String	/zone/	TEST

Commit changes
192-168-0-255.com - ("192-168-0-255.com" {type master; file "/etc/namedb/blockeddomain.hosts";}:)
airkuanta.com - ("airkuanta.com" {type master; file "/etc/namedb/blockeddomain.hosts";}:)
buffer-control.com - ("buffer-control.com" {type master; file "/etc/namedb/blockeddomain.hosts";}:)
hwcdn.net - ("hwcdn.net" {type master; file "/etc/namedb/blockeddomain.hosts";}:)

2.4 *Adding New Sites to Sinkhole Manually*
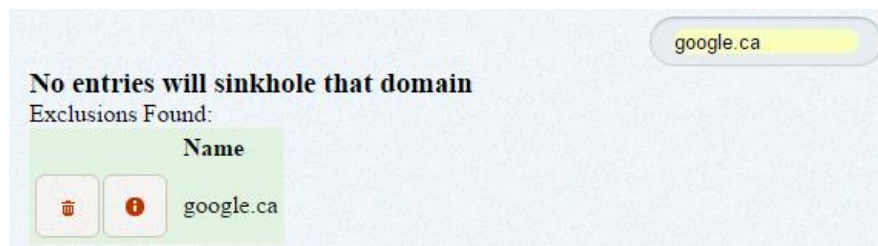
To manually add new sites to the Sinkhole manually from IDS/IPS events, CIRT data, etc, select Add site to sinkhole and paste an individual or multiple site in the box, pick the List: you want to enter it (default: Weblist), add a comment (i.e. C&C beacon) and add site. The *Clean Content* tool is to ensure the web site names are formatted properly.

Domain(s):
xiazai2.net
pzxdj.com

List: Weblist ▾
Comment: C&C beacon
Add Site

2.5 *Search Domains*

The **Search Domain** tab is used to find information about the domain entered in the Sinkhole. In this example google.ca has been added to the *Site exclusion list*. After the hostname/domain has been found, you have the option of adding further comments or deleting the site from the Sinkhole.

google.ca

No entries will sinkhole that domain
Exclusions Found:

		Name
🗑	❶	google.ca

3. Remote Access

In order for the DNS sinkhole to be remotely accessible via SSH, it will require some minor configuration changes for remote access.

3.1 *Configuring SSH TCP Wrappers*

vi /etc/hosts.allow (or use Webmin, Servers, TCP Wrappers)

Add in the TCP Wrappers file the IPC workstations allowed to connect to DNS server

sshd: 192.168.14. \
     172.16.2.

The /etc/hosts.deny has been configured to deny ALL (ALL: ALL) by default

3.2 *Webmin Configuration*

Webmin is a secure remote console manager. For example, the PowerDNS server can be managed via an SSL enabled browser to manage MySQL, restart services and view the system logs. It is quite versatile and very easy to use for those who prefer using a GUI to manage their sensor.

After you log into Webmin, to manage MySQL and Apache, go to the **Servers** section.

3.2.1 Configuring Webmin

You need to change the Webmin default account password before making this system operational. The default account is **admin** and the default password is **admin**. Change the default admin account password the following manner. At the command line console do:

/usr/local/webmin/changepass.pl /etc/webmin admin newpassword

The Webmin service can be stopped and started this way:

/etc/webmin/stop
/etc/webmin/start

3.2.2 Access is via SSL this way:

Now you can login via Webmin by entering: https://yourIPaddress:10000

4. DNS Sinkhole Configuration

This installation offers two methods to deploy a DNS sinkhole. The first method is using ISC Bind[2] and the second is using PowerDNS[3]. Both of these options use a single shell script to parse the DNS records that will be used by the sinkhole. There are two major differences between both options; with ISC Bind, everything is viewed and managed at the command line and with PowerDNS, all the records are stored in a MySQL database.

There are two ways of taking control of a domain: a system host file (in Unix /etc/hosts or in Windows C:\WINDOWS\system32\drivers\etc\hosts) and with a DNS server controlled by an organization.

To populate the DNS server with a list of sinkhole records, the script `sinkhole_parser.sh` will be used to insert the records into the PowerDNS database or Bind server in /var/named. Whether you are using PowerDNS or Bind to sinkhole malicious sites, you need to do some minimal Bind configuration in order to load the PowerDNS database with your list of malicious domains. Follow Configuring BIND as a Sinkhole to configure the various files.

4.1 *Controlling Access to Suspicious Sites*

The hosts file model is very complex and tedious to maintain. The more hosts on the network the more difficult it will be to update all the computers with a controlled DNS list. The second method is centralized. Each time an update is applied to the DNS server, any hosts accessing the Internet who will ask the corporate DNS server where is www.malware.ca and if the answer is not held locally (cached or otherwise) it will be forwarded to the next server for resolution.

However, for any domain listed in the sinkhole, the returned address will be one configured by the administrator to prevent the host from accessing known sites that "break corporate" policy (malware, spyware, etc) and redirect the client to an IP of your choice (web server, IPS/IDS) to identify and prevent the host suspected to be compromised to access that site.

4.1.1 Changing MySQL root Password

By default, the MySQL database **listens on 127.0.0.1 only** via the startup script which will prevent direct external connections to TCP port 3306. However, the MySQL database default password is blank (no password assigned) and must be changed immediately with the following command:

/usr/local/mysql/bin/mysqladmin –u root password 'your-new-password-for-sql_user-root'

---

[2] https://www.isc.org
[3] http://www.powerdns.com

The PowerDNS account is **powerdns** and password is **password**. Since all the components are running on the same computer and PowerDNS has been pre-configured to use this default password, it can be left as password since the database can only be access via 127.0.0.1.

However, if you wish to change the password, it can be changed with the following command or you can use the Webmin administrative tool but remember it must be changed as well where indicated in this document (where the powerdns account is used).

mysql –p   (root password set earlier)
\u mysql   (User mysql)

GRANT ALL PRIVILEGES ON pdns.* TO powerdns@127.0.0.1 IDENTIFIED BY \
'make_a_password_for_user_powerdns' WITH GRANT OPTION;
\q   (To quit mysql)

Note: When entering the passwords for the powerdns user, ensure it gets enclosed in single quotes or you will get an error.

If you decide to change the default password, you will also need to edit the PowerDNS configuration file and make the change there as well. To change the password, change directory to /var/www/htdocs/inc and edit *config.inc.php* and change the database password in there and save the file.

4.1.2 Configuring PowerDNS as a Sinkhole

When using the sinkhole installation CD, all the configuration files are set to have the DNS sinkhole ready to respond to DNS queries. Any queries not managed by the sinkhole will be forwarded to get a response. At the end of the installation, if you selected Apache and MySQL to start as a service, your server is now working. If you forgot to do so, please activate the startup scripts to start the services by executing at the console, pkgtool, Setup, services, select rc.httpd and rc.mysqld, select OK and exit. Restart the server to activate the service.

PowerDNS has some performance related settings that can be further tuned and are listed here at this reference.[4] These settings are configured in */usr/etc/pdns.conf*.

The `distributor-threads` setting is a choice of 1 to more backends. The default when pdns_server starts is 3. If set to 1 thread, PDNS reverts to unthreaded operation which for some systems may be a lot faster.

It is necessary for the sinkhole to keep the default setting for the CNAME (enable) to respond to sinkhole query.

---

[4] http://doc.powerdns.com/performance-settings.html

The wildcard setting (wildcards=no) cannot be used with the sinkhole. By default the sinkhole will respond to wildcard requests. For example, the sinkhole contains `dns.com` and a workstation sends a DNS request for `areyouthere.dns.com`, it will give the same answer as `dns.com`.

4.1.3 PowerDNS Monitoring Webserver

PowerDNS has a built-in web server to monitor the server. The server can be configured by editing the *pdns.conf* file and configuring the following parameters:

Start a webserver for monitoring
webserver=yes

IP Address of Webserver to listen on (configure IP address)
webserver-address=192.168.1.5

Password required for accessing the webserver (configure password)
webserver-password=password

Port of the webserver to listen on (that is the default port)
webserver-port=8081

The PowerDNS built-in server provides statistical information on the server performance. You can access the server based on the configuration you entered in the previous section. The default installation does not activate this server. Restart the PDNS service to activate the server: */etc/rc.d/rc.pdns restart*

To access the server with default ports 8081 do: http://webserver:8081

Username: admin and Password: password

PowerDNS 4.0.1

POWERDNS:::

Uptime: 3 seconds
Queries/second, 1, 5, 10 minute averages: 0, 0, 0. Max queries/second: 0
Backend query load, 1, 5, 10 minute averages: 0, 0, 0. Max queries/second: 0
Total queries: 1. Question/answer latency: 0ms

Log Messages

✖ Reset

Showing: Top 10 of 3

Resize: 10 100 500 1000 (10000) 500000

About to create 3 backend threads for UDP	1	33.3%
Creating backend connection for TCP	1	33.3%
Done launching threads, ready to distribute questions	1	33.3%
Total:	**3**	**100%**

Queries for existing records, but for type we don't have

✖ Reset

Showing: Top 10 of 0

Resize: 10 100 500 1000 (10000) 500000

Total:	**0**	**100%**

4.2   *Basic Bind DNS Configuration*

This section illustrates a basic Bind configuration needed to run a sinkhole. The sinkhole is populated by executing the `sinkhole_parser.sh`. The basic sinkhole setup is configured to run as a forwarder.

4.2.1 Configuring BIND as a Sinkhole

DNS Bind's main configuration file is located in the /etc directory and is called named.conf. This file contains the information to get the server going. It tells the named service how to start in the options section, what to log in the logging section and what zones to load.

The Bind server configuration file is `named.conf` with an example in Annex A. This file is located in /etc on the server. This file shows a standard configuration for a caching server. This configuration is shows the server is also acting as a forwarder with the forwarder option. This can contain as many DNS server as needed.

The `named.conf` file should be edited to reflect your site settings. This file can be edited with vi or via Webmin. The following settings should be reviewed and adjusted as necessary: allow-transfer, allow-recursion and forwarders.

Annex B shows an example of a single site DNS A record configuration file located in the /var/named directory. This file domain.nowhere must be configured exactly as shown in the example.

Annex C contains an example of adding a single site in a sinkhole (i.e. www.sink.ca). Our example of a single site sinkhole is called site_specific_sinkhole.conf and contains the zone information for the site that we want to resolve the IP for. This file located in the /var/named directory.

Annex D contains an example of adding an entire domain to a sinkhole (i.e. google.com). If we take the *.google.com example, this mean that anything that starts with something and ends with .google.com would be redirected to the sinkhole. This file located in the /var/named directory.

The same principle can be done with a country code such as Canada (ca), could be added to this list to sinkhole all domains ending with **.ca**. For example, if someone attempt to access google.ca or test.google.ca, it would be redirected to the sinkhole address.

rdnc Commands

rndc flush	Flushes all of the server's caches
rndc flush [view]	Flushes the server's cache for a view
rndc halt	Stop the server without saving pending updates
rndc reload	Reload configuration file and zones
rndc reconfig	Reload configuration file and new zones only

rndc status	Query the status of the server
rndc stop	Save pending updates to master files and stop the server
rndc querylog	Activate/deactivate DNS query logging

5. Populating Sinkhole with sinkhole_parse.sh

This menu shows the 5 options available to populate the DNS sinkhole. When the script is executed, the first thing it does is download a list of known bad domains from a list of sites parsed by this script. The script will download more than 20,000 domains to be used by the sinkhole.

5.1.1 Prevent a Domain from ending in the Sinkhole – checked_sites

Before starting the sinkhole_script.sh, a list of pre-populated domains that should never be in the sinkhole exist in /root/scripts/checked_sites. If any of these domains gets added to the list downloaded by the script, they will be automatically removed after you select "*D. Download sinkhole updates*" to ensure they never get blocked by the DNS Sinkhole.

Warning: Make sure you review this list and remove or add any sites that should never be in your local sinkhole. If a site has been added to the sinkhole that should not be in there, add the site to the *checked_sites* list and rerun the download to remove it from the sinkhole. This list can also be edited in Webmin under Servers → DNS Sinkhole Control → Edit checked_sites.

```
            DNS Sinkhole Menu

    A. Manually add single domain to sinkhole
    D. Download sinkhole udpates
    T. Testing new zone file for errors
    F. Empty PowerDNS database of all its records
    R. Zone check failed, restore and exit
    B. Zone file is good, load it in Bind and exit
    P. Zone file is good, load PowerDNS and exit
    E. Exit script

    What is your choice?
```

5.1.2 Manually add single domain to sinkhole (A)

This menu is used to add a single domain to the DNS sinkhole. After adding a domain to the sinkhole, make sure you test (T) the domain list to ensure the new domain does not already exist in the sinkhole and then execute (B) or (P) to load the update.

5.1.3 Download sinkhole updates (D)

This menu is used to download updates from a selected list of websites. The script contains the following list of website:

www.malwaredomains.com
https://zeustracker.abuse.ch

5.1.4 Testing new zone file for errors (T)

This menu is used for testing the DNS records for any errors. Always select this menu before loading the new list in either Bind or PowerDNS. If the test fails and you are using Bind for your sinkhole, use the "**r**" option to restore the backup file to the sinkhole.

The only reason this could happen is when the /var/named/site_specific_sinkhole.conf or /var/named/custom_domain_sinkhole.conf has been populated with a site that is now listed in site_specific_sinkhole.conf. At the console, you can use Alt-F2 to open a new terminal or open another SSH session and remove the duplicate record.

You can rerun the test and if it passes, load the updates into Bind or PowerDNS.

5.1.5 Empty PowerDNS database of all its records (F)

This menu is self explanatory. It wipes the PowerDNS database clean. Everything is removed and both servers are restarted. This probably should be done once or twice a month before reinserting the sinkhole records. When the database is wiped, all the domains that were in the sinkhole are no longer redirected to the IP address of your choice. Depending of the speed of your server, it may take less then a few minutes to load the records back into the database.

5.1.6 Zone check failed, restore and exit (R)

Important: This menu is used only if using Bind to restore the backup Bind DNS sinkhole records to its location in /var/named/entire_domain_sinkhole.conf. The backup file is stored in /tmp.

5.1.7 Zone file is good, load it in Bind and exit (B)

This menu is used to load the new sinkhole list into the Bind server. When done loading, it automatically reload the new zones and exit the script.

5.1.8 Zone file is good, load PowerDNS and exit (P)

This menu is used to delete old records from the database and load the new sinkhole list into the PowerDNS database.

In this section, the script uses a binary named zone2sql which translate the domain list into SQL text that can be imported into the database. The script compares what is in the database against what has been downloaded from the Internet and remove records that are no longer considered malicious and import the new additions. The script will parse for inclusion into the database and file with a .conf located in /var/named directory. When done loading, it automatically refresh the new zones and exit the script.

6. DNS Sinkhole Packet Capture

If you do not have an IDS to alert in realtime the clients redirected to your DNS Sinkhole address or addresses, the server has the ability to capture the data and generate a daily report based on the sinkhole clients. The report gets posted in *Webmin → Servers → DNS Reports*. In order to activate capture and report directly on the DNS server, the following steps must be done:

1. Edit /etc/rc.d/rc.local
   - Follow these steps to activate a virtual IP assigned to eth0:1
   - Uncomment and configure and eth0 virtual:
     #/sbin/ifconfig eth0:1 192.168.1.6 netmask 255.255.255.0 up
   - Uncomment the socat ports listed (80, 8000, 8080)
     (If needed, you can add more web related ports)
   - Uncomment log_packets_sinkhole.sh script
     #/usr/local/sbin/log_packets_sinkhole.sh start
   - Save changes

2. Edit /var/named/sinkhole/client.nowhere:
   - Change the IP listed (192.168.1.5) to the virtual IP address eth0:1
     (192.168.1.6 as shown in section 1)
   - Repeat the steps and edit /var/named/sinkhole/domain.nowhere
     (change both IP addresses to 192.168.1.6 as shown in section 1)
   - This will ensure the client is redirected to the virtual address
   - Save changes

3. Edit /usr/local/sbin/log_packets_sinkhole.filter
   - Make sure the IP you added for eth0:1 appears in the list. If using more than one IP, list them all. If using a single address, list only that single address. It is important the filter be listed like *\(host 192.168.1.5 or host 192.168.6\)* if using more than one address to filter only on the DNS Sinkhole address list.

4. Activate the DNS Sinkhole cronjob
   - Run command *crontab -e*
   - Uncomment the log_packets_sinkhole.sh script to run every hours.
     #Enable this to collect DNS Sinkhole Virtual IP collection to run every hour.
   - Uncomment the DNS Sinkhole report. Last line of the crontab file.
     (/g will bring you directly to the bottom of the file)
     # Creating DNS Sinkhole daily report web page
     *#15 1***/root/scripts/httpry_daily.sh /dev/null 2>1&*
   - The report will run at 1:15 each morning
   - Save changes and reboot

Note: You may need to change the script interface from eth0:1 to something else if you are using another interface.

5. Edit /usr/local/sbin/log_packets_sinkhole.sh
    - If using a virtual card other than eth0:1, this file need to be edited and modified, otherwise, no changes needed.

6. Edit /usr/etc/pdns.conf
    - This step might be necessary if you are using two NIC
    - Uncomment local-address Local IP addresses to which we bind
    - local-address=0.0.0.0 (change 0.0.0.0 to eth0 address)

6.1 *Testing your Packet Capture Configuration*

After rebooting the server, it is important to check the server changes have been configured correctly.

1. Check new sinkhole IP is active: ifconfig eth0:1
2. Check socat listeners are started on eth0:1: netstat –an |grep 80
    a. You should see TCP 80, 8000 and 8080 listening on the IP address you configured for eth0:1
3. Check log_packets_sinkhole.sh is running: ps –aef | grep daemonlogger
4. Test your sinkhole with: nslookup www.google.com
5. Test packet collection: http://eth0:1 address
6. Check /LOG/sinkhole/dailylogs/DATE/daemonlogger.pcap.xxxxxxxxxx
    a. tcpdump –nAs 0 –r daemonlogger.pcap.xxxxxxxxxx
    b. look for the connection you just attempted to be logged

6.2 *Sinkhole Web Report*

The Sinkhole Webmin report will be displayed the next day in Webmin under Servers →
DNS Reports → DNS by date. This is an example of the daily report

```
         Daily DNS Sinkhole report for the 20100816
Count    Source            Destination       HTTP      Website

1        192.168.25.28     192.168.25.6      GET       static.xvideos.com
1        192.168.25.28     192.168.25.6      GET       www.xvideos.com
2        192.168.25.26     192.168.25.6      GET       ad.adriver.ru
5        192.168.25.25     192.168.25.6      GET       finderwid.org
6        192.168.25.28     192.168.25.6      GET       bar.hub.cc
```

7. Operating System Patches

The Slackware web site should be monitored for any new patches that should be applied on the selected packages at Annex A. The site is http://www.slackware.com

The security list is available at:

http://www.slackware.com/security/list.php?l=slackware-security&y=2016

7.1 *Slackware Patch Maintenance Script*

Patches can be maintained and downloaded by running the /root/slackupdate.sh script. This script will check for any package that are available for update and saves them in /tmp/slackupdate. To install the patch updates as follow:

telinit 1
cd /tmp/slackupdate
upgradepkg <patch>.tgz
(Do not apply kernel modules or new kernels patches. System will fail if applied)
telinit 3

Note with kernels: Do not apply any kernel or modules because this system has a custom kernel and modules and if those supplied by Slackware are applied, the system will fail.

7.2 *Mounting USB Drive*

To mount a USB drive with this OS, plug in the USB drive and do the following:

dmesg |grep sda, sdb, sdc or sdd
mount /dev/sda? /mnt/hd   Where? = the partition and usually 1
cd /mnt/hd   You can copy or move files from this directory
umount /usb   When done with the USB drive

Note: the device can be sda, sdb, sdc, sdd and usually partition 1 (i.e. sda1)

8. **Customizing Server**

To capture client attempting to connect to sinkhole domains, the second NIC of the DNS sinkhole will be configured with a primary address and one to many virtual addresses linked to the primary.

eth1 Interface Setup

The secondary interface (eth1) should be configured by adding the IP address to /etc/rc.d/rc.inet1.conf . The format is straight forward, here is an example:

# Config information for eth1:
IPADDR[1]="172.16.1.1"
NETMASK[1]="255.255.255.0"
USE_DHCP[1]=""
DHCP_HOSTNAME[1]=""

To add a virtual interface to eth1, edit the /etc/rc.d/rc.local and add the following ifconfig command to activate each virtual IPs:

ifconfig eth1:0 192.168.1.10 up

9. **SUDO**

Sudo is a program used to allow users to run programs with the security privileges of another user (normally the superuser, or root). Each administrator needing access to manage the DNS sinkhole services will be provided with an account that will permit rebooting the server and restarting the PowerDNS services.

The /etc/sudoers configuration file must be managed using *visudo*.

All Sudo activity will be logged in /var/log/sudolog. Here is an example of the logs:

Mar 9 11:12:57 : security : TTY=pts/2 ; PWD=/etc ; USER=root ; COMMAND=/sbin/ifconfig
Mar 9 11:14:47 : security : command not allowed ; TTY=pts/2 ; PWD=/etc ;USER=root ;
            COMMAND=/sbin/lsmod

10. DNS Sinkhole Files and Scripts

/etc/rc.d	All system start/stop scripts
/etc/rc.d/rc.K	Kill all system script
/etc/rc.d/rc.S	Start up script for single-user mode
/etc/rc.d/rc.M	Start up script for multi-user mode
/etc/rc.d/rc.mysqld	MySQL database script
/etc/rc.d/rc.netdevice	NIC module loading script
/etc/rc.d/rc.local	Script for all other configuration
/etc/powerdns/recursor.conf	Configuration file for rc.pdns_recursor script
/etc/issue	Banner message
/etc/motd	Banner message
/etc/rc.d/rc.firewall	Setup firewall
/var/adm/messages	General log file
/var/adm/syslog	Syslog file
/var/etc/pdns.conf	Configuration file for rc.pdns
/var/run	Various server pid files
/usr/local/mysql	MySQL database

11. Annexes

The following annexes contain examples of the various configuration files used by a DNS forwarder that is also acting as a caching server.

11.1 *Annnex A: named.conf*

```
options {
      directory "/var/named";
      // version statement - inhibited for security
      version "my own";
      // optional - disables all transfers
      // slaves allowed in zone clauses
      allow-transfer {"none";};
      allow-recursion {192.168.1.0/24; localhost;};
      forwarders { 192.168.20.5; 4.2.2.1; };

    /*
     * If there is a firewall between you and nameservers you want
     * to talk to, you might need to uncomment the query-source
     * directive below.  Previous versions of BIND always asked
     * questions using port 53, but BIND 8.1 uses an unprivileged
     * port by default.
     */

    // query-source address * port 53;
};

//
// log to /var/log/named/example.log all events from
// info UP in severity (no debug)
// defaults to use 3 files in rotation
// BIND 9.x parses the whole file before using the log
// failure messages up to this point are in (syslog)
// typically /var/log/messages
//

logging {

    channel default_syslog {
          // Send most of the named messages to syslog.
          syslog local2;
          severity debug;
    };

    channel audit_log {
          // Send the security related messages to a separate file.
          file "/var/log/named/named.log";
          severity debug;
          print-time yes;
    };

    channel query_log {
```

```
            // Send the security related messages to a separate file.
            file "/var/log/named/query.log";
            severity debug;
            print-time yes;
        };

        category default { default_syslog; };
        category general { default_syslog; };
        category security { audit_log; default_syslog; };
        category config { default_syslog; };
        category resolver { audit_log; };
        category xfer-in { audit_log; };
        category xfer-out { audit_log; };
        category notify { audit_log; };
        category client { audit_log; };
        category network { audit_log; };
        category update { audit_log; };
        category queries { query_log; };
        category lame-servers { audit_log; };

};
//
// a caching only nameserver config
//
zone "." IN {
        type hint;
        file "caching-example/named.root";
};

include "/var/named/site_specific_sinkhole.conf";
include "/var/named/entire_domain_sinkhole.conf";

zone "localhost" IN {
        type master;
        file "caching-example/localhost.zone";
        allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "caching-example/named.local";
        allow-update { none; };
};
```

11.2 *Annex B: domain.nowhere*

```
$TTL  600
@               IN SOA      stars.org. root (
                            1           ; serial
                            3H          ; refresh
                            15M         ; retry
                            1W          ; expiry
                            1D )        ; minimum

                24H IN NS       @
                24H IN A        192.168.1.5
*               24H IN A        192.168.1.5
```

11.3 *Annex C: site_specific_sinkhole.conf*

```
zone "image.slidexxx.com" IN { type master; file "/var/named/sinkhole/
client.nowhere "; };
zone "our.sink.com" IN { type master; file "/var/named/sinkhole/
client.nowhere "; };
```

11.4 *Annex D: entire_domain_sinkhole.dns*

```
zone "finesse.org" IN { type master; file "/var/named/sinkhole/
domain.nowhere "; };
zone "example.com" IN { type master; file
"/var/named/sinkhole/domain.nowhere"; };
zone "ca" IN { type master; file "/var/named/sinkhole/domain.nowhere";
};
```