DNS Real-time Black hole List System

Zeno. O. Popovici and Remus Brad, *Member IEEE*

Abstract — **This paper describes a DNS Real-time Black hole List system. The RBL allows easy blacklisting of SPAM and can be shared between multiple mail servers. The application allows complete online processing of the list and has also an integrated notification system for easy e-mail notifications. The RBL server used BIND and the management software is built in PHP with a MYSQL backend.**

***Keywords* — DNS, RBL, DNSBL, PHP, MYSQL, SPAM**

I. Introduction

SPAM has become a part of our every day life, a cat and mouse game, between a billion dollar underground industry and the IT staff of every organization.

There are many anti-spam solutions out there, but sometimes one needs a customized solution for fighting spam. This solution comes as a custom DNS RBL (Real-time Black-hole List system).

II. Implementation

We started with defining the initial requirements of the application: a simple online blacklisting interface, possibility to define custom DNS responses (error messages, delisting interface. Extra features are introduced in the later in the development process: integration with our current authentication system, e-mail notifications, and dynamic DNS updates.

One of the base requirements was reliability of DNS updates and failsafe procedures to catch malformed input, which prevent DNS data file corruption. This was important because it could cause the e-mail server to reject all mail in certain situations.

For the development of the frontend a PHP application with a MYSQL backend was chosen because both platforms are Open Source and therefore do not generate additional costs and also provide easy programming and maintenance. Such a system would be 100% web-based accessible anytime, anywhere.

For the backend we needed a DNS server as most email servers have standard functions for managing DNS blacklists. This implementation is very robust, as we only need to properly configure a DNS server for our purposes.

In the end we choose BIND, because of our previous experience with the system and because BIND supports dynamic DNS updates.

A. Proposed Functionality

To summarize, the proposed components of such a system were:

- Visitor – Domain Lookup (Database Search)
- Visitor – Domain Delisting Request
- Visitor – Notification System
- Admin – Blacklist Domain (Reasons, Header)
- Admin – Domain Lookup
- Admin – Delist Domain
- Admin – Permanent Blacklist Domain
- Admin – Add Domain Info (Owner, Contact)
- Admin – Notification System
- System – BIND Server
- System – Dynamic DNS Updates
- System – Error Management

B. User Roles and Application Life Cycles

There are two kinds of users of the RBL System: Administrators who administer the RBL, list and delist domains, site Visitors who submit delist requests. The activities of a user in each of these roles over the lifecycle of this conference are summarized below.

Administrator Lifecycle:

1. Login is managed by our university standard authentication system. Our datacenter personnel has admin rights in the system, once they login to our website.
2. *Inserts* (Lists) a domain into the database providing full headers, domain name and listing reason. Entering domain owner data is optional. If owner data is inserted, he can choose to notify the owner of the blacklisting.
3. *Lists* the status of the most 20 entries in the RBL database. *Domain name, listed date, status, reason* and *listed by* fields are shown.
4. *Searches* the RBL database for a specific domain. All domain information stored in the database is shown.
5. *Delists* or *Permanently Blacklists* a domain, providing a reason.
6. *Deletes* (and Delists) a domain from the database.

Visitor (or Blacklisted Domain Owner) Lifecycle:

1. Receives a customized "Service Unavailable" 554 e-mail reject messages, which directs him to our RBL web application on blacklist.ulbsibiu.ro.
   Alternatively, he receives an e-mail containing a notice about the RBL listing and a link, which

Zeno O. Popovici is with the Data Communication Department, "Lucian Blaga" University of Sibiu, Romania (e-mail: zeno.popovici@ulbsibiu.ro).

Remus Brad is with the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania (e-mail: remus.brad@ulbsibiu.ro).

directs him to our RBL web application on blacklist.ulbsibiu.ro.

2. A page with all information stored in the database about his domain is shown.
   Alternatively, he visits the site and *Searches* the RBL database for a specific domain. All domain information stored in the database is shown.
3. *Sends a delisting request* that contains name, e-mail, domain and delist reason.
4. Receives notifications about the status of the domain when an administrator updates the RBL entry.

C. *Data Model*

The Data Model for this application is a basic one, only one table being used. There are not relational.
"*rbl_id*" is used to track entries in notifications and RBL website application.

	Name	Type	Length
1	rbl_id	smallint	6
	rbl_date	datetime	0
	rbl_domain_owner	varchar	255
	rbl_domain_owner_email	varchar	255
	rbl_domain	varchar	255
	rbl_spam_mail	mediumtext	0
	rbl_short_reason	mediumtext	0
	rbl_delist_request	enum	0
	rbl_delist_reason	mediumtext	0
	rbl_status	enum	0
	rbl_last_modified	datetime	0
	rbl_updated_by	varchar	255
	rbl_entered_by	varchar	255

Figure 1. Data Model.

III. System Modules

A. *Web Application – Backend*

The web application design is minimal and manages all RBL entries, inserts the information provided into the database, updates the DNS Server and notifies the Visitor/Administrator via e-mail.

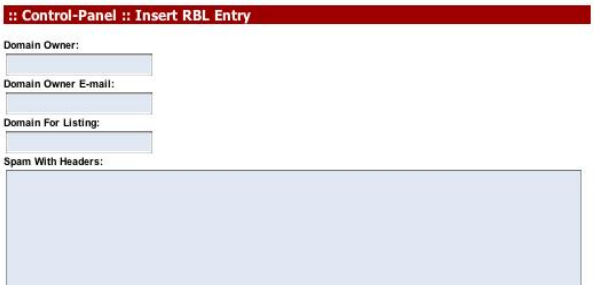

Figure 2. Insert RBL Entry.

Administrators can easily follow up on delist requests, edit entries, and view recent actions.

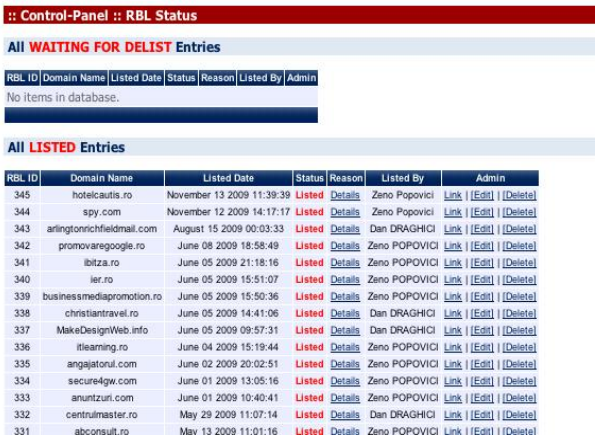

Figure 3. View Recent Actions.

B. *Web Application – Frontend*

The frontend is a simple website where the domain owner or any visitor are given information about the RBL system, statistics and where they can search for a specific RBL entries. A delisting form is also available.

Figure 4. Frontend View.

C. *Web / Shell Application - NSUpdate*

NSUpdate is a C++ program that is run from a standard Unix shell. As PHP cannot directly access NSUpdate, we generate a shell script using *fwrite()* and run it via the *system()* function. We use a secure key to authenticate to BIND.

This is how we generate and run the update script:

```
function rblnsupdate($rbl_action,$rbl_domain,$rbl_id)
{
// Update NS Records for RBL
// Set Files
$rbl_path = .'rbl/named';
$rbl_keyfilename = 'rbl.ulbsibiu.ro.private';
$rbl_nsaction = 'named/nsaction.sh';
$rbl_ip = '127.0.0.2';
$rbl_zone = 'rbl.ulbsibiu.ro';
$rbl_message = '"Blacklisted by RoEduNet Sibiu - See:
http://blacklist.ulbsibiu.ro/?rbl_id='.$rbl_id.'"';

//Open action file
$fp = fopen($rbl_nsaction,'w');
//Write Delete or Update action
fwrite($fp,'#!/bin/sh cd '.$rbl_path.' nsupdate -k
rbl.ulbsibiu.ro.private <<EOF');

//Check if we're deleting
if ($rbl_action == 'DELETE') {
```

```
fwrite($fp, 'server hercules.ulbsibiu.ro
prereq yxdomain '.$rbl_domain.'.'.$rbl_zone.'
update delete '.$rbl_domain.'.'.$rbl_zone.' A
update delete '.$rbl_domain.'.'.$rbl_zone.' TXT
send');
} elseif ($rbl_action == 'UPDATE') {
fwrite($fp, 'server hercules.ulbsibiu.ro
prereq nxdomain '.$rbl_domain.'.'.$rbl_zone.'
update add '.$rbl_domain.'.'.$rbl_zone.' 43200 A
'.$rbl_ip.'
update add '.$rbl_domain.'.'.$rbl_zone.' 43200 TXT
'.$rbl_message.'
send');
}
fwrite($fp,'  quit   EOF');
fclose($fp);
$run = 'sh '.$rbl_path.'/nsaction.sh';
$output = system($run, $retval);
}
```

D. Web Application – Form Validation (VDaemon)

The authors have not developed this module. VDaemon is a purchased, highly configurable shareware module that does Server-Side validations for all user forms. If errors are found, the specific fields are colored in red and notifications messages are displayed.

E. DNS Servers – Configuration

We configured an instance of the BIND DNS Server as a regular DNS Service, containing our rbl.ulbsibiu.ro master zone. We removed our private key below for security reasons.

Code from *named.conf* file of our RBL BIND Server:

```
// Private key allowed to update this service
key "rbl.ulbsibiu.ro." {
algorithm hmac-md5;
secret "string_containing_private_key};

// Master Zone
zone "rbl.ulbsibiu.ro" {
type master;
 file "master/rbl.ulbsibiu.ro";

// Allowing transfer of this zone only for these IPs
allow-transfer { 193.226.5.32/27; 217.73.173.0/24;
194.169.191.0/24; 94.52.190.83; 88.158.220.187;};
allow-update { key rbl.ulbsibiu.ro.; };
;
```

We also included the new sub domain and DNS service in our main DNS server configuration so that the rest of the world can find it. Please note that the 2 BIND server s are situated on 2 separate physical systems with their own real IP address.

Code from named.conf file of our main DNS Server:

```
;---------RBL---------------
rbl         IN     NS     hercules.ulbsibiu.ro.      ;
blacklist   IN     CNAME  hera    ;
```

F. Mail Server – Configuration

We are using postfix for our e-mail service. Postifx has full support for RBLs and configuring it to accept our new service, is made as follows.

Code from main.conf file of our postfix Mail Server:

```
#Recipient Restrictions
smtpd_recipient_restrictions =
reject_invalid_hostname,
reject_unknown_recipient_domain,
reject_unauth_pipelining,
reject_rhsbl_sender rbl.ulbsibiu.ro
```

A. Web Service – Statistics (MRTG)

The idea of generating usage statistics came short after bringing the system online. We wanted to test the efficiency of the system.

The data was gathered using command line "*cat*" utitly with several arguments to parse the postfix mail logs.

We built a bash script to generate a text file with the number of RBL rejections "*mrtg_generator*":

```
#!/usr/bin/bash
// Parse maillog
cat /etc/mrtg/maillog | grep "rbl.ulbsibiu.ro" -c
>/etc/mrtg/stats/rbl.log
// Delete maillog
cat /dev/null > /etc/mrtg/maillog
```

The scripts runs every 5 minutes and then deletes the *maillog* file. The *maillog* file is generated by postfix automatically.

For the visual representation of the data we choose MRTG (Multi Router Traffic Grapher). In order for the data to be passed to mrtg we created yet another script, this time in Perl, a language MRTG understands "*rbl.pl*":

```
#!/usr/bin/perl
$str=`cat /etc/mrtg/stats/rbl.log`;
$val=int($str);
print "$val\n";
print "0\n";
print "0\n";
print "0\n";
```

Because we wanted to run the scripts before MRTG runs we've created also a "*mrtg_starter.sh*" script:

```
#!/usr/bin/bash
MRTG Starter
echo "Generating Stats Files"
echo ""
/etc/mrtg/mrtg_generator.sh
echo ""
echo "Running MRTG"
echo ""
/usr/bin/mrtg /etc/mrtg/mrtg.cfg
```

And finally, we configured the MRTG config file "*mrtg.cfg"* as follows:

```
##################################################
# Description: Postfix Mail Queue
# Contact: ccom@ulbsibiu.ro
# System Name: ULBS Mail System
# Location: Communication Center
#.........................................

Target[rbl]: `/etc/mrtg/scripts/rbl.pl`
Extension[rbl]: php
Colours[rbl]: BrightBlue#7794C9,DarkBlue#011F5B,DARK
GREEN#006600,VIOLET#ff00ff
YLegend[rbl]: Emails Blocked by blacklist.ulbsibiu.ro
ShortLegend[rbl]: emails blocked by
```

```
blacklist.ulbsibiu.ro
Legend1[rbl]: Emails Blocked by blacklist.ulbsibiu.ro
LegendI[rbl]:  RBL Blocked Mail:
PNGTitle[rbl]: RBL Blocked Mail
(blacklist.ulbsibiu.ro)
Options[rbl]: growright, noinfo, nopercent, integer,
nobanner, printrouter, gauge, perhour, noo
TimeStrPos[rbl]: RU
MaxBytes[rbl]: 9999999
Title[rbl]: RBL Blocked Mail(blacklist.ulbsibiu.ro)
#--------------------------------------------
```

The graphic output of MRTG is listed in section V.

IV. RBL Implementation Method

All implementations of RBLs are working on the same basic principle: as the system receives an e-mail a DNS query is sent in a standard format to the main DNS server for the sender domain. The response is either *NXDomain*, or an IP address. If the response is an IP address, the mail server generates a "554 Service Unavailable" reject message.

There is an informal protocol for the addresses returned by RBL queries, which match. Most RBLS return an address in the 127.0.0.0/8 IP loopback network. The address 127.0.0.2 indicates a generic listing. Other addresses in this block may indicate something specific about the listing — that it indicates an open relay, proxy, spammer-owned host, etc.

We're only using 127.0.0.2 in our listings, indicating a generic listing.

A. RBL Lookup Examples

This is an example RBL lookup trough our system using *nslookup* command line utility:

```
[root@ares ~]# nslookup example.com.rbl.ulbsibiu.ro
Server:   194.169.191.2
Address:  194.169.191.2#53
** server can't find example.com.rbl.ulbsibiu.ro:
NXDOMAIN
```

As you can see the domain above is not listed.

```
[root@ares ~]# nslookup zeno.ro.rbl.ulbsibiu.ro
Server:   194.169.191.2
Address:  194.169.191.2#53
Non-authoritative answer:
Name:   zeno.ro.rbl.ulbsibiu.ro
Address: 127.0.0.2
```

However, this other one is listed and will will bounce back to the sender:

```
Nov 13 18:04:19 ares postfix/smtpd[25968]: NOQUEUE:
reject: RCPT from mail-gx0-
f218.google.com[209.85.217.218]: 554 5.7.1 Service
unavailable; Sender address [myself@zeno.ro] blocked
using rbl.ulbsibiu.ro; Blacklisted by RoEduNet Sibiu -
See: http://blacklist.ulbsibiu.ro/?rbl_id=346;
from=<myself@zeno.ro> to=<zeno.popovici@ulbsibiu.ro>
proto=ESMTP helo=<mail-gx0-f218.google.com>
```

V. Statistics & Conclusion

Our database has reached 330 blocked domains and is still growing. MRTG statistics show that the system is indeed catching Spam, regular filters would otherwise miss.

The numbers may not be impressive, but during a few periods in the year, if the system is maintained properly can block a lot of unwanted "region specific" Spam. It also helps educating online marketers, to play fair, as we had till now around 20 delisting requests.

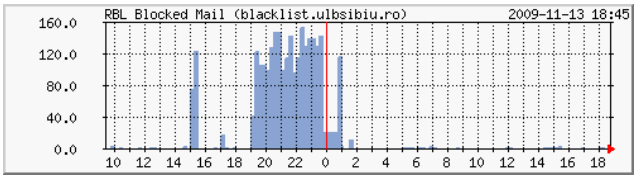

Figure 5. MRTG - Daily Blocked Spam.

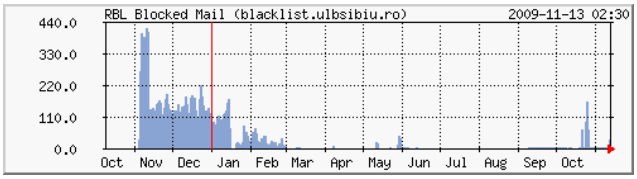

Figure 5. MRTG - Yearly Blocked Spam.

VI. Appendix

A. PHP

PHP is a widely used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

B. MYSQL

MySQL is a relational database management system (RDBMS). MySQL stands for "My Structured Query Language". The program runs as a server providing multi-user access to a number of databases.

C. BIND

BIND is by far the most widely used DNS software on the Internet. It provides a robust and stable platform on top of which organizations can build distributed computing systems with the knowledge that those systems are fully compliant with published DNS standards.

NSLookup and NSUpdate are command line tools provided with the BIND package.

D. Postfix

Postfix is a free and open source mail transfer agent (MTA), a computer program for the routing and delivery of email. It is intended as a fast, easy-to-administer, and secure alternative to the widely used Sendmail MTA.

E. Spam

Spam is the abuse of electronic messaging systems (including most broadcast media, digital delivery systems) to send unsolicited bulk messages indiscriminately.

F. RBL

A DNSBL (DNS-based Blackhole List, Block List, or Blacklist; see below) is list of IP addresses published through the Internet Domain Name Service in a particular format. DNSBLs are most often used to publish the addresses of computers or networks linked to spamming; most mail server software can be configured to reject or flag messages which have been sent from a site listed on one or more such lists.

G. MRTG

The Multi Router Traffic Grapher, or just simply MRTG, is free software for monitoring and measuring the traffic load on network links. It allows the user to see traffic load on a network over time in graphical form.

H. BASH

Bash is the shell for the GNU operating system from the GNU Project. It can be run on most Unix-like operating systems. It is the default shell on most systems built on top of the Linux kernel.

I. PERL

Perl is a high-level, general-purpose, interpreted, dynamic programming language.

REFERENCES

[1] Bakken S.S, Aulbach A, Schmid E, Winstead J, Wilson L.T., Lerdorf R, Zmievski A, Ahto J, PHP Manual,http://www.php.net/manual/en/

[2] HTML - http://www.w3.org/MarkUp/

[3] O. Reilly Media, Inc., “MySQL Cookbook” – http://www.oreilly.com

[4] O. Reilly Media, Inc., “The PHP Cookbook” – http://www.oreilly.com

[5] Eric A. Meyer: “Cascading Style Sheets 2.0 Programmer's Reference”, Editura Osborne/McGraw-Hill, 2001.

[6] Marc Johnson, “Javascript Manual of Style”, 1998

[7] MySQL Reference Manual, http://www.mysql.com/documentation/index.html

[8] Wikipedia, http://www.wikipedia.org

[9] Postfix, Reference Manual, http://www.postfix.org/documentation.html

[10] MRTG, Reference Manual, http://oss.oetiker.ch/mrtg/doc/index.en.html

[11] BIND, Reference Manual, http://www.isc.org/software/bind/

[12] KLOTH.Net, How to Build an own DNSBL, http://www.kloth.net/internet/dnsbl-howto.php