

## Re: How do fix a good solution against spam..

*Source:* <http://www.derkeiler.com/Mailing-Lists/FreeBSD-Security/2004-05/0063.html>

---

**From:** Anton Alin-Adrian (*aanton\_at\_reversedhell.net*)

**Date:** 05/16/04

Date: Sun, 16 May 2004 01:46:53 +0300

To: [freebsd-security@freebsd.org](mailto:freebsd-security@freebsd.org)

It's half off topic, half not. Something has to be done, and it takes technical skills and knowleged ppl to handle the issues. At least this is how I rationate when deciding where to ask.

I started an anti-spam project on my own. At some point others offered to help, but we all know boring real-life shuts down all the enthusiasm.

M.Jessa> Not only it's way faster than perl based messagewall, amavisd and  
M.Jessa> mailscanner etc but it also has neat stuff like making connections  
M.Jessa> back to the sender's MX checking for validity of the sender's  
M.Jessa> email.

So far I can only release this code. It implements exactly what was mentioned about exim. I use it with qmail because qmail I have, but can be used with postfix/sendmail with ease. So now not only exim can do that hack.

I just wanted to make the code available so users can benefit from it (hopefully).

PS – this is how i use it:

.qmail-file:

```
| /usr/local/bin/check /usr/local/bin/safecat /path/to/Maildir/tmp  
/path/to/Maildir/new
```

#the above after | is on a single line.

Hope there are not many bugs.

Yours Sincerely,

--

Alin-Adrian Anton

Reversed Hell Networks

GPG keyID 0x1E2FFF2E (2963 0C11 1AF1 96F6 0030 6EE9 D323 639D 1E2F FF2E)

gpg --keyserver pgp.mit.edu --recv-keys 1E2FFF2E

FreeBSD–Security: Re: How do fix a good solution against spam..

```
/*
 * The MX query routines are Copyrighted (C) 2004 by HL Combrinck and are licensed under GPL (see
 below),
 * and they provide "Sample C code to resolve MX records for an address".
 *
 *
 * This program is derivative work based on his original functions, and is distributed under the following
 terms:
 *
 * LICENSE:
 *
 * The program provides functions for testing if an e–mail address was faked by a spammer or it's real, and it's
 part of the L.A.U.R.A anti–spam project and campaign.
 *
 * Copyright (C) 2004 Anton Alin–Adrian aanton()reversedhell.net
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111–1307 USA
 *
 * END OF LICENSE
 */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/nameser.h>
#include <resolv.h>
```

```
#define PORT 25 /* SMTP default port */
#define MAXDATASIZE 1024 /* we don't need more */
```

```
/* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! REPLACE WITH YOUR *REAL* DOMAIN & *FAKE* E–MAIL USER
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
```

## FreeBSD–Security: Re: How do fix a good solution against spam..

```
#define MY_VALID_MAIL_DOMAIN "INEXISTENT–USER–HERE@reversedhell.net" /* replace user
with something decent like 'antispamrobot' */
#define MY_VALID_DOMAIN "reversedhell.net" /* must be your real domain you are connecting from */

struct mx
{
    int pref;
    char host[1024];
};

#ifndef HFIXEDSZ
# define HFIXEDSZ 12
#endif
#ifndef INT16SZ
# define INT16SZ sizeof(cit_int16_t)
#endif
#ifndef INT32SZ
# define INT32SZ sizeof(cit_int32_t)
#endif

int totalsize=0;

/*
 * Compare the preference of two MX records. Check the actual
 * number listed in the MX record – if they're the same, randomize.
 */
int mxcomp(int p1, int p2)
{
    if (p1 > p2) return(1);
    else if (p1 < p2) return(0);
    else return(rand() % 2);
}

/*
 * sort_mxrecs()
 *
 * Sort MX records
 *
 */
void sort_mxrecs (struct mx *mxrecs, int nmx)
{
    int a, b;
    struct mx t1, t2;

    if (nmx < 2) return;

    for (a = nmx - 2; a >= 0; --a)
    {
        for (b = 0; b <= a; ++b)
        {
            if (mxcomp(mxrecs[b].pref, mxrecs[b+1].pref))
```

## FreeBSD–Security: Re: How do fix a good solution against spam..

```
        {
            memcpy(&t1, &mxrecs[b], sizeof(struct mx));
            memcpy(&t2, &mxrecs[b+1], sizeof(struct mx));
            memcpy(&mxrecs[b], &t2, sizeof(struct mx));
            memcpy(&mxrecs[b+1], &t1, sizeof(struct mx));
        }
    }
}

/*
 * getmx()
 *
 * Get MX recs for an address.
 *
 * Upon success, it fills 'mxbuff' with one or more MX hosts, delimited by
 * ':' chars, and returns the number of hosts. 0 if none found.
 */
int getmx(char *mxbuff, char *dest, int maxbuffsz)
{
    union
    {
        u_char bytes[1024];
        HEADER header;
    } ans;

    int ret;
    unsigned char *startptr, *endptr, *ptr;
    char expanded_buf[1024];
    unsigned short pref, type;
    int n = 0;
    int qdcount;

    struct mx *mxrecs = NULL;
    int nmx = 0;

    ret = res_query (dest, C_IN, T_MX, (unsigned char *)ans.bytes,
                    sizeof(ans));

    if (ret < 0)
    {
        mxrecs = malloc(sizeof(struct mx));
        mxrecs[0].pref = 0;
        strcpy(mxrecs[0].host, dest);
        nmx = 0;
    }
    else
    {
        if (ret > sizeof(ans)) ret = sizeof(ans);
    }
}
```

## FreeBSD–Security: Re: How do fix a good solution against spam..

```
startptr = &ans.bytes[0];
endptr = &ans.bytes[ret];
ptr = startptr + HFIXEDSZ; /* skip header */

for (qdcnt = ntohs(ans.header.qdcnt); qdcnt--;)
    ptr += ret + QFIXEDSZ
{
    if ((ret = dn_skipname(ptr, endptr)) < 0) return(0);
}

while(1)
{
    memset (expanded_buf, 0, sizeof(expanded_buf));
    ret = dn_expand (startptr, endptr, ptr, expanded_buf,
        sizeof(expanded_buf));
    if (ret < 0) break;
    ptr += ret;

    GETSHORT (type, ptr);
    ptr += INT16SZ + INT32SZ;
    GETSHORT (n, ptr);

    if (type != T_MX) ptr += n;
    else
    {
        GETSHORT(pref, ptr);
        ret = dn_expand(startptr, endptr, ptr, expanded_buf,
            sizeof(expanded_buf));
        ptr += ret;

        ++nmx;
        if (mxrecs == NULL)
            mxrecs = malloc(sizeof(struct mx));
        else
            mxrecs = realloc (mxrecs, (sizeof(struct mx) * nmx));

        mxrecs[nmx - 1].pref = pref;
        strcpy(mxrecs[nmx - 1].host, expanded_buf);
    }
}

/* sort by MX pref */
sort_mxrecs(mxrecs, nmx);

strcpy(mxbuff, "");
for (n=0; n<nmx; ++n)
{
    if (strlen(mxbuff)+strlen(mxrecs[n].host) < maxbuffsz)
        strcat(mxbuff, mxrecs[n].host);
    else
```

## FreeBSD–Security: Re: How do fix a good solution against spam..

```
        break;
        strcat(mxbuff, ":");
    }
    /* kill last ':' */
    if (mxbuff[strlen(mxbuff)-1] == ':') mxbuff[strlen(mxbuff)-1] = 0;
    free(mxrecs);
    return(nmx);
}
```

```
int checkmail(char *addy, char *myhost)
```

```
{
    int sockfd, numbytes;
    char buf[MAXDATASIZE];
    struct hostent *he;
    struct sockaddr_in their_addr;
    fd_set readfds;

    if ((he=gethostbyname(myhost)) == NULL) {
        perror("gethostbyname");
        return -2;
    }

    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socket");
        return -1;
    }

    their_addr.sin_family = AF_INET; // host byte order
    their_addr.sin_port = htons(PORT); // short, network byte order
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    memset(&(their_addr.sin_zero), '\0', 8); // zero the rest of the struct

    if (connect(sockfd, (struct sockaddr *)&their_addr, sizeof(struct sockaddr)) == -1) {
        perror("connect");
        close(sockfd);
        return -2;
    }

    if ((numbytes=recv(sockfd, buf, MAXDATASIZE-1, 0)) == -1) {
        perror("recv");
        close(sockfd);
        return -1;
    }

    buf[3]='\0';
    if (atoi(buf)!=220) {
        close(sockfd);
        return -1;
    }
}
```

## FreeBSD–Security: Re: How do fix a good solution against spam..

```
memset(buf,0x0,sizeof(buf));
snprintf(buf,sizeof(buf),"helo %s\r\n",MY_VALID_DOMAIN);
if (send(sockfd,buf,strlen(buf),0)==-1)
{
    perror("send");
    close(sockfd);
    return -1;
}

memset(buf,0x0,sizeof(buf));

if ((numbytes=recv(sockfd, buf, MAXDATASIZE-1, 0)) == -1) {
    perror("recv");
    close(sockfd);
    return -1;
}
buf[3]='\0';
if (atoi(buf)!=250) {
    close(sockfd);
    return -1;
}
memset(buf,0x0,sizeof(buf));

snprintf(buf,sizeof(buf),"MAIL FROM:<%s>\r\n",MY_VALID_MAIL_DOMAIN);
if (send(sockfd,buf,strlen(buf),0)==-1)
{
    perror("send");
    close(sockfd);
    return -1;
}

memset(buf,0x0,sizeof(buf));
if ((numbytes=recv(sockfd, buf, MAXDATASIZE-1, 0)) == -1) {
    perror("recv");
    close(sockfd);
    return -1;
}
buf[3]='\0';
if (atoi(buf)!=250) {
    close(sockfd);
    return -1;
}

memset(buf,0x0,sizeof(buf));
snprintf(buf,sizeof(buf),"RCPT TO:<%s>\r\n",addy);
if (send(sockfd,buf,strlen(buf),0)==-1)
{
    perror("send");
    close(sockfd);
    return -1;
}
```

## FreeBSD–Security: Re: How do fix a good solution against spam..

```
memset(buf,0x0,sizeof(buf));
if ((numbytes=recv(sockfd, buf, MAXDATASIZE-1, 0)) == -1) {
    perror("recv");
    close(sockfd);
    return -1;
}
buf[3]='\0';
if (atoi(buf)!=250) {
    close(sockfd);
    return -2;
}

return 0;

} // checkmail

int loopcheckmail(char *addy)
{
    int n,ret;
    char buf[1024], *ptr;
    char *myhost;

    myhost=(char *) malloc(strlen(addy)+1);
    myhost=strchr(addy,'@')+1;

    n = getmx (buf, myhost, sizeof(buf)-1);

    if (!n)
    {
        ret=checkmail(addy,myhost);
    }
    else
    {
        ptr=strchr(buf,':');
        if (ptr!=NULL) *ptr='\0';
        ret=checkmail(addy,buf);
    }
    return ret;
}

char *read_mail_buffer(FILE *fp)
{
    char c='\0';
    int i=0;
    long int size=1024+1;
    int padder=1024;

    char *ptr,*s;
```

```

if ((s=(char *) malloc((size_t)size))==NULL)
{
    perror("malloc");
    exit(EXIT_FAILURE);
}
memset(s,(char)0x0,(size_t) size);
ptr=s;

while ((c!=(char)EOF)){
    c=(char) getc(fp);

    if (i>=size-1)
    {
        size+=padder;
        if ( (s=(char *)realloc(s,(size_t)size) ) == NULL) {
            perror("realloc");
            exit(EXIT_FAILURE);
        }
        ptr=s+i*sizeof(char);
        if (totalsize > 700000) padder=padder*2;
    }
    i++;

    *(ptr++)=c;

}
*(--ptr)='\0';

totalsize=size;

return (char *) s;
}

int filtervalidmail(char *s)
{
    char *ptr;
    char *addy;
    char *left,*right;
    int i,j,stop=0;
    char c;

    ptr = strcasestr(s,"From:");
    if (ptr==NULL) return -1;
    ptr+=5;
    ptr=strchr(ptr,'@');
    left=ptr;
    right=ptr;
    while (isalnum(*(--left)) )
    {

```

## FreeBSD–Security: Re: How do fix a good solution against spam..

```
c=*(--left);

ptr=strchr(ptr,'<')+1;
for (i=0;*(ptr++)!='>';i++);
addy=(char *) malloc((i+1)*sizeof(char));
memset(addy,0x0,i+1);

}

int main (int argc,char *argv[])
{
    int ret;
    char *bigbuf;
    /*
    if (argc < 2) {
        fprintf(stderr,"What to check? Give me valid e–mail format.\n");
        exit(EXIT_FAILURE);
    }
    ret=loopcheckmail(argv[1]);
    switch (ret) {
        case -1:
            fprintf(stderr,"IRRELEVANT: Error..\n");
            break;
        case -2:
            fprintf(stderr,"BLOCK!\n");
            break;
        case 0:
            fprintf(stderr,"IRRELEVANT\n");
            break;
    }

    */
    bigbuf=read_mail_buffer(stdin);
    filtervalidmail(bigbuf);
    return 0;
}
```

- 
- application/pgp–signature attachment: [OpenPGP digital signature](#)