# PLIB Bugs

**Brought to you by: sjbaker**

## #55 integer overflow for maliciously crafted tga file

**Status:** open **Owner:** nobody **Labels:** None
**Priority:** 5
**Updated:** 2021-04-06 **Created:** 2021-04-03 **Creator:** Wooseok Kang **Private:** No

In plib, there is an integer overflow vulnerability that may cause arbitrary code execution in the victim's system with a maliciously crafted input.

The vulnerability resides in ssgLoadTGA() function in src/ssg/ssgLoadTGA.cxx file. In line 91, the program reads data from given tga file using fread.

```
if ( fread(header, 18, 1, f) != 1 )
```

Then, it stores the value to xsize and ysize and bits without sanitizing.

```
// image info
int type = header[2];
int xsize = get16u(header + 12);
int ysize = get16u(header + 14);
int bits  = header[16];
```

If xsize and ysize are enough large to cause integer overflow the small heap block is allocated when the new image is created. It leads to buffer overrun when reads data to this buffer.

```
GLubyte *image = new GLubyte [ (bits / 8) * xsize * ysize ];
```

I attach the maliciously crafted tga file which crashes program like below.

```
$ apt source plib
$ cd plib-1.8.5
$ ./configure && make
$ cd src/ssg
$ gcc -I../../src/sg -I../../src/util test.cxx -lplibssg
```

where text.cxx is as follows

```
#include <stdlib.h>
#include "ssg.h"

int main(int argc, char **argv) {
    ssgLoadTGA (argv[1], NULL);
}
```

```
$ ./a.out poc.tga
DEBUG: ssgLoadTGA: Loading 'poc.tga', colormap 65535x65535-8.
terminate called after throwing an instance of 'std::bad_alloc'
  what():  std::bad_alloc
Aborted (core dumped)
```

Thank you.

**1 Attachments**

poc.tga

**Related**

Bugs: #55

## Discussion

Steve Baker - *2021-04-06*

PLIB has been obsolete and unmaintained for at LEAST 15 years!!

Good catch...but it's not ever getting fixed.

Steve Baker - *2021-04-06*

PLIB has been obsolete and unmaintained for at LEAST 15 years!!

Good catch...but it's not ever getting fixed.

On 2021-04-03 06:16, Wooseok Kang wrote:

Log in to post a comment.

**[bugs:#55] integer overflow for maliciously crafted tga file**

**Status:** open
**Group:**
**Created:** Sat Apr 03, 2021 12:16 PM UTC by Wooseok Kang
**Last Updated:** Sat Apr 03, 2021 12:16 PM UTC
**Owner:** nobody
**Attachments:**

- poc.tga
  (24 Bytes; application/octet-stream)

In plib, there is an integer overflow vulnerability that may cause arbitrary code execution in the victim's system with a maliciously crafted input.

The vulnerability resides in ssgLoadTGA() function in src/ssg/ssgLoadTGA.cxx file. In line 91, the program reads data from given tga file using fread.

```
if ( fread(header, 18, 1, f) != 1 )
```

Then, it stores the value to xsize and ysize and bits without sanitizing.

~~~
// image info
int type = header[2];
int xsize = get16u(header + 12);
int ysize = get16u(header + 14);
int bits = header[16];
~~~

If xsize and ysize are enough large to cause integer overflow the small heap block is allocated when the new image is created. It leads to buffer overrun when reads data to this buffer.

```
GLubyte *image = new GLubyte [ (bits / 8) * xsize * ysize ];
```

I attach the maliciously crafted tga file which crashes program like below.
~~~
$ apt source plib
$ cd plib-1.8.5
$ ./configure && make
$ cd src/ssg
$ gcc -I../../src/sg -I../../src/util test.cxx -lplibssg
~~~

where text.cxx is as follows

~~~

# include <stdlib.h></stdlib.h>

# include "ssg.h"

int main(int argc, char **argv) {
ssgLoadTGA (argv[1], NULL);
}
~~~

~~~
$ ./a.out poc.tga
DEBUG: ssgLoadTGA: Loading 'poc.tga', colormap 65535x65535-8.
terminate called after throwing an instance of 'std::bad_alloc'
what(): std::bad_alloc
Aborted (core dumped)
~~~

Thank you.

Sent from sourceforge.net because you indicated interest in https://sourceforge.net/p/plib/bugs/55/

To unsubscribe from further messages, please visit https://sourceforge.net/auth/subscriptions/

**Related**

Bugs: #55

SourceForge Headquarters
225 Broadway Suite 1600
San Diego, CA 92101
+1 (858) 454-5900

## Resources