

ESIEA - CVO

---

## Test Results GostCrypt

---

*Author:*

Sebastiaan GROOT



June 27, 2014

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Static Analysis</b>	<b>1</b>
<b>3</b>	<b>Conclusion</b>	<b>9</b>

## 1 Introduction

This document contains the test results of the GostCrypt application. It is a work in progress and will be updated as more results come in. Comprehensive testing of an application the size of GostCrypt is a lengthy process. A comprehensive test of the BootLoader, setup process and windows kernel driver of TrueCrypt 7.1a, on which GostCrypt is based, was performed by the Open Crypto Audit Group[1].

Static analysis is performed with the help of the Flawfinder application. Flawfinder helps identify calls to library functions that are difficult to use safely and that are often the cause of several security vulnerabilities. Analysis using Flawfinder has been completed.

The next round of testing will focus on memory management errors using Valgrind.

## 2 Static Analysis

### Common/Dlgcode.c

Common/Dlgcode.c:2493: [4] (buffer) strcpy:

Does not check for buffer overflows when copying to destination.

Consider using strncpy or strlcpy (warning, strncpy is easily misused).

This call copies the 27-byte string from the previous call to the end of the file path. If the max. file path is not limited to (260 - 27), this could overflow.

Common/Dlgcode.c:2968: [4] (buffer) wcscat:

Does not check for buffer overflows when concatenating to destination.

Can cause a buffer overflow if the attacker has write access to the Language.xml file. It first copies a value from the language pack file into a 1024 byte buffer, appends the device number and a L':' character.

Common/Dlgcode.c:3266: [4] (buffer) strcpy:

Does not check for buffer overflows when copying to destination.

Consider using strncpy or strlcpy (warning, strncpy is easily misused).

Appends a maximum of 18 characters to the GostCrypt install path. Destination buffer is 520 bytes large (twice GST\_MAX\_PATH). If the install path can be larger than 520 - 18, this causes an overflow.

Common/Dlgcode.c:3728: [4] (buffer) strcat:

Does not check for buffer overflows when concatenating to destination.

Consider using strncat or strlcat (warning, strncat is easily misused).

If calling strcat to append a single character to a buffer and the string is already at the maximum length of the buffer (len(buffer)-1), strcat will happily replace the null terminator

with the single character and add a null terminator one position further, outside of the buffer. If the "SelectMultipleFilePath" contains a string at the beginning that isn't short enough to allow one character and one file name to be appended to it, strcat will cause a buffer overflow.

Common/Dlgcode.c:3812: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:3814: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

Common/Dlgcode.c:3819: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:3822: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

Common/Dlgcode.c:3824: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

The destination buffer is sufficiently long (8192 bytes), but an attacker with write access to the Language.xml files can cause a buffer overflow by exceeding this limit.

Common/Dlgcode.c:4553: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4557: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4561: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4565: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4569: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4573: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4577: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4581: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4585: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Dlgcode.c:4589: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Copies small (under 20 characters) messages to a 128 byte buffer. Attackers with write access to the Language.xml file can cause an overflow here.

Common/Dlgcode.c:4603: [4] (buffer) wscpy:

Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

Common/Dlgcode.c:4607: [4] (buffer) wcscpy:

Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

Copies stings from Language.xml to a 300 byte buffer. Attackers with write access to the Language.xml file can cause an overflow here.

Common/Dlgcode.c:5796: [4] (format) swprintf:

Potential format string problem. Make format string constant.

Destination buffer is 4096 bytes, which is enough for the message + drive letter. Attackers with write access to the Language.xml file can cause an overflow here.

Common/Dlgcode.c:6632: [4] (buffer) strcpy:

Does not check for buffer overflows when copying to destination.  
Consider using strncpy or strlcpy (warning, strncpy is easily misused).

Common/Dlgcode.c:6636: [4] (buffer) strcpy:

Does not check for buffer overflows when copying to destination.  
Consider using strncpy or strlcpy (warning, strncpy is easily misused).

Some callers of this function use a source buffer of twice the capacity of the destination buffer (520 bytes instead of 260). If the Windows API returns a file path this long associated with this module, it will cause an overflow.

Common/Dlgcode.c:8285: [4] (buffer) strcat:

Does not check for buffer overflows when concatenating to destination.  
Consider using strncat or strlcat (warning, strncat is easily misused).

As long as file names are not nearly close to 520 bytes long, there are no issues here. However, this function is sometimes called with a source buffer with a capacity of 520 (even if this capacity is not used).

Common/Dlgcode.c:9638: [4] (buffer) strcat:

Does not check for buffer overflows when concatenating to destination.  
Consider using strncat or strlcat (warning, strncat is easily misused).

If the SHGetFolderPathA Windows function cannot return paths larger than 507 bytes, this works fine. Otherwise, it can cause a buffer overflow.

### Common/Format.c

Common/Format.c:132: [4] (buffer) strcpy:

Does not check for buffer overflows when copying to destination.

Consider using strncpy or strlcpy (warning, strncpy is easily misused).

Destination buffer and source buffer have a size of 260 bytes. However, the next call, ToUnicode, assumes that this destination buffer of 260 bytes has double its strlen still available in size. The actual limit in the destination buffer in this strcpy call is therefore 130 bytes. This is not being checked for.

### Common/Progress.c

Common/Progress.c:77: [4] (buffer) wcscpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

An adversary with access to the Language.xml file can cause a buffer overflow.

Common/Progress.c:87: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:89: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:91: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:93: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:101: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

Common/Progress.c:110: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:112: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:114: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:116: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Common/Progress.c:118: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Small strings written to a wchar\_t buffer with a capacity of 100. Adversary with write access to the Language.xml file can cause a buffer overflow.

## Driver/Ntdriver.c

Driver/Ntdriver.c:2464: [4] (buffer) wcsncpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

A destination buffer of 200 bytes is used as the MOUNTMGR\_TARGET\_NAME structure. Seeing how the source buffer in this call has a capacity of 256 WCHAR's, this should be changed to a higher value.

## Format/Gstformat.c

Format/Gstformat.c:1207: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Format/Gstformat.c:1215: [4] (format) sprintf:

Potential format string problem. Make format string constant.

Unlikely to be vulnerable with a destination buffer of 4096 bytes. Still, an attacker with write access to Language.xml can cause an overflow.

gostcrypt\_flawfinder/Format/Gstformat.c:1444: [4] (buffer) wcsncpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1446: [4] (buffer) wcsncpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1448: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:1508: [4] (buffer) wcsncpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1510: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:1517: [4] (buffer) wcsncpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1518: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:1794: [4] (buffer) wcsncpy:

Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1797: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1800: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1803: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1806: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1809: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1812: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:1816: [4] (buffer) wscat:  
Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:2453: [4] (format) swprintf:  
Potential format string problem. Make format string constant.

gostcrypt\_flawfinder/Format/Gstformat.c:3170: [4] (format) swprintf:  
Potential format string problem. Make format string constant.

gostcrypt\_flawfinder/Format/Gstformat.c:3172: [4] (format) swprintf:  
Potential format string problem. Make format string constant.

gostcrypt\_flawfinder/Format/Gstformat.c:3198: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.  
Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:3199: [4] (buffer) wcscpy:  
Does not check for buffer overflows when copying to destination.



Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:3200: [4] (buffer) wscpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:3202: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3203: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3205: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3207: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3208: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3209: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3211: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3212: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3214: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3216: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3217: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3218: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3259: [4] (format) swprintf:

Potential format string problem. Make format string constant.

gostcrypt\_flawfinder/Format/Gstformat.c:3956: [4] (buffer) wscpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:3959: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3962: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

gostcrypt\_flawfinder/Format/Gstformat.c:3963: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:3966: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:3970: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:3971: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:4483: [4] (format) swprintf:

Potential format string problem. Make format string constant.  
gostcrypt\_flawfinder/Format/Gstformat.c:4485: [4] (format) swprintf:

Potential format string problem. Make format string constant.  
gostcrypt\_flawfinder/Format/Gstformat.c:4488: [4] (format) swprintf:

Potential format string problem. Make format string constant.  
gostcrypt\_flawfinder/Format/Gstformat.c:4728: [4] (buffer) wcscpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:4729: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:4730: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:5689: [4] (buffer) wcscpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:5690: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:5691: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:5704: [4] (buffer) wcscpy:

Does not check for buffer overflows when copying to destination.

Consider using a function version that stops copying at the end of the buffer.

gostcrypt\_flawfinder/Format/Gstformat.c:5705: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.  
gostcrypt\_flawfinder/Format/Gstformat.c:5706: [4] (buffer) wscat:

Does not check for buffer overflows when concatenating to destination.

Attacker with write access to Language.xml can cause a buffer overflow.

### 3 Conclusion

The results of the static analysis using Flawfinder identified a number of potential security vulnerabilities. These security vulnerabilities are mostly related to buffer overflows. Looking at the results, 3 groups of problems are identified.

1. The majority of potential security issues are caused by the use of buffer copy functions that do not stop at the end of the destination buffer. As in most cases the capacity of the destination buffer is known, this class of vulnerabilities can be partly mitigated.
2. Throughout the code, multiple sizes for the maximum path length are used. The sizes 260, 261, 520 and 521 were used and were sometimes used within the same function. Using the same buffer size for maximum path length and annotating the exceptions to this rule will decrease the chance on accidental errors.
3. A method of limiting the size of Language.xml strings should be introduced to reduce the risk of (accidental) buffer overflows, caused by modifications of the Language.xml file or language packs.

### References

- [1] TrueCrypt - Security Assessment (2014). Open Crypto Audit Project.  
<https://opencryptoaudit.org/reports/iSec.Final.Open.Crypto.Audit.Project.TrueCrypt.Security.Assessment.pdf>