Code Review

141627

Gjøvik University College

Faculty of Computer Science and Media Technology

IMT 3501 Software Security, Hanno Langweg

Assignment 04

Autumn semester 2014

Introduction

The task involved a code review a on given code with documentation. Since we should also document our process, I decided to split this assignment in two parts. The Review and the Documentation. The Review part includes the used strategy, the found vulnerabilities and suggestions for removal of those. In the Documentation part includes a short description how I could apply the techniques mentioned in the Review part.

1 Code Review

My strategy involved the following steps:

- Define scope
- Collect information
- Review: design, code; test
- Document findings
- Analyze findings: severity, remediation effort and suggestions

1.1 Define scope

The inspection will cover TOCTTOU condition findings as well as input data vulnerabilities.
Since source code is provided, the review method will be white box.

1.2 Collect information

A documentation paper was provided with the source code. It covered the implementation and a control flow chart of the software to get an overview of the program.
I found a static Delphi code analyzer tool which I will use but besides, do a manual inspection of the code, for which I had to get familiar with the Delphi Language. [1]
Compiling [2] the code throws errors on Windows 7 of missing units, so the review will be static and without tests.

The tools in use where

grep

vim

Pascal Analyzer [3]

Lazarus [2]

1.3 Review

First was the Pascal Analyzer run to find errors. The results are in the appendix A and analyzed in the next section.

Arguments passed to the program are defined as an array called ParamStr [1]. Every user input should be sanitized, so I started the manual lookup with it.

A grep -n ParamStr * resulted, the ParamStr is used in the following lines:

fmSDAppInfo.pas : Line 123

SDCommon.pas : Lines 396, 397, 398, 399, 400

SDSecureDisplaySvcUnit.pas : Lines 73, 207, 244, 245, 246, 310

SDUserSessionManager.dpr : Lines 194, 197, 200, 203

I further looked up if any hard coded paths are included in the source with

grep -n '[A-Z]:\\' *

and found results:

SDCommon.pas: Lines 68, 111

SDInfoSecurity.pas: Lines 333, 335

SDSecureDisplaySvcUnit.pas: Lines 253, 254, 307, 361, 362, 370, 493, 494, 204, 205

1.4 Analysis of findings

1.4.1 Pascal Analyzer

There were no strong warnings in neither of the delphi projects defined by the .dpr files.

The normal warnings included minor issues like public defined variables which only find private usage, same with interfaces.

The analyzing was done with the evaluation version, so some output was omitted.

1.4.2 Lookup for ParamStr

fmSDAppInfo.pas :

Line 123

The input data is sanitized here. The procedure has a default value if StrToIntDef fails to parse the argument. If there are more arguments passed, it will simply use the default.

SDCommon.pas :

Line 396, 397, 398, 399, 400

The argument here is in the initialization section of the unit. It fills a variable with the log file name and writes its actions to the log file. There are no actions needed here.

SDSecureDisplaySvcUnit.pas :

Line 73

A half hard coded path to an executable file is passed to StartAsLocalSystemInSession function.
It makes Use of the ParamStr(0) and extracts the path of the directory to get a hard coded path to the SDUserSessionManager.exe executable file.
Following the function call, the path is next passed to a Windows library call CreateProcessAsUser.
The hard coded path to the SDUserSessionManager.exe file is a flaw, because an adversary could imitate or altered this file.
A Solution would be an dynamic implementation or some sort of a key negotiation between the Cage Service and the Cage Manager or other forms of integrity checks.

Line 207, 244, 245, 246, 310
All commented out.

SDUserSessionManager.dpr :
Lines 194, 197, 200, 203
Every line contains the same method for paths to executables like mentioned above. Line 194 is just for logging purpose and line 203 is commented out but in 197 and 200 are processes created from said executables. The vulnerability is the same as above, since the file could have been altered or imitated.

1.4.3 Lookup for hard coded paths

Every hard coded path should be looked up for earlier mentioned vulnerability reasons.

1.5 Conclusion

The found vulnerabilities aren't major but should be considered in an future patch. A major problem in this source is the lack of documentation or comments. The Shark Cage document provided a general idea of the software architecture but it was hard to copy these ideas on the source code to get a view of the bigger picture.

2 Documentation

The next time I do a code review, I will exchange the points “Define scope” and “Collect information”. Defining a scope before anything of the software is known messes up the time estimation. Time management is easier with more information. It took me a lot of time to get a picture of the software, but since not all units where provided and no test runs could be done, it would eat up too much time to check for logical or design errors.
It was also hard to find logical errors since I am not familiar with the delphi language.

References:

[1] Delphi Basic [Internet]. Neil Moffatt. Available from http://www.delphibasics.co.uk/

[2] Lazarus [Internet]. Available from http://www.lazarus.freepascal.org/

[3] Pascal Analyzer [Internet]. Peganza. Available from

http://www.peganza.com/products_pal.htm

Appendix A

```
****************************************************************************
*                    Warnings Report for                    *
*              C:\SECUREDESKTOP-SOURCE\SDAPPINFO.DPR                  *
*                    10/8/2014 7:33:23 AM                       *
****************************************************************************

In this evaluation version, identifiers starting with J,K,L are excluded.
Also some report lines are displayed as "xxxxxx".

Interfaced identifiers that are used, but not outside of unit (13, was unknown):
--------------------------------------------------------------------------------


ApplicationDirectoryName : UnicodeString        Typed const, Interfaced SDCommon (15)
BackgroundBitmapFileName : UnicodeString        Typed const, Interfaced SDCommon (23)

The entire list is not shown in this evaluation version

Interfaced class identifiers that are public/published, but not used outside of unit (7, was unknown):
------------------------------------------------------------------------------------------------------
GetBitmap                      Func, Method          dmScreenshot\TScreenshot (15)

imgApplicationLogo : Simple (unknown) ClassField            fmSDAppInfo\TfmAppInfo (16)

The entire list is not shown in this evaluation version



Functions called as procedures (4, was 4):
----------------------------------------------------------------------------


fmSDAppInfo.TfmAppInfo.DisplayAppInfo at fmSDAppInfo (134)
fmSDAppInfo.TfmAppInfo.RemoveAppBar at fmSDAppInfo (148)
```

```
Redeclared identifiers from System unit (1, was unknown):
-----------------------------------------------------------------------------

PI : Simple (unknown)       Var, Local          fmSDAppInfo\TfmAppInfo\StartApp (199)
```

```
*************************************************************************
*                    Warnings Report for                    *
*            C:\SECUREDESKTOP-SOURCE\SDSECUREDISPLAYSVC.DPR            *
*                    10/8/2014 7:31:32 AM                    *
*************************************************************************

In this evaluation version, identifiers starting with J,K,L are excluded.
Also some report lines are displayed as "xxxxxx".

Interfaced identifiers that are used, but not outside of unit (22, was unknown):
--------------------------------------------------------------------------------
ApplicationDirectoryName : UnicodeString  Typed const, Interfaced SDCommon (15)
BackgroundBitmapFileName : UnicodeString  Typed const, Interfaced SDCommon (23)


The entire list is not shown in this evaluation version


Interfaced class identifiers that are public/published, but not used outside of unit (1, was unknown):
-----------------------------------------------------------------------------------------------------
StartAsLocalSystemInSession    Proc, Method          SDSecureDisplaySvcUnit\TSDSecureDisplayService (29)


Variables that are set, but never referenced (2, was unknown):
----------------------------------------------------------------------------
Name : UnicodeString          RecField             SDInfoProcesses\SDDumpAccessMask\TAccessRight (626)
Value : Cardinal             RecField             SDInfoProcesses\SDDumpAccessMask\TAccessRight (625)



Empty begin/end-blocks (1, was unknown):
----------------------------------------------------------------------------
SDInfoProcesses (405)
```

Empty finally-block (1, was unknown):

```
---------------------------------------------------------------------------
SDInfoProcesses (414)
```

Functions called as procedures (29, was 29):

```
---------------------------------------------------------------------------
SDCommon.SDWriteToMailslot at SDSecureDisplaySvcUnit (247)
SDCommon.SDWriteToMailslot at SDSecureDisplaySvcUnit (254)
```

The entire list is not shown in this evaluation version

Identifier with same name as keyword/directive (1, was unknown):

```
---------------------------------------------------------------------------
Name : UnicodeString     RecField      SDInfoProcesses\SDDumpAccessMask\TAccessRight (626)
```

```
****************************************************************************
*                    Warnings Report for                  *
*          C:\SECUREDESKTOP-SOURCE\SDUSERSESSIONMANAGER.DPR          *
*                    10/8/2014 7:33:57 AM                  *
****************************************************************************

In this evaluation version, identifiers starting with J,K,L are excluded.
Also some report lines are displayed as "xxxxxx".

Interfaced identifiers that are used, but not outside of unit (19, was unknown):
--------------------------------------------------------------------------------
ApplicationDirectoryName : UnicodeString  Typed const, Interfaced SDCommon (15)
BackgroundBitmapFileName : UnicodeString  Typed const, Interfaced SDCommon (23)

The entire list is not shown in this evaluation version



Variables that are set, but never referenced (2, was unknown):
----------------------------------------------------------------------------
Name : UnicodeString        RecField            SDInfoProcesses\SDDumpAccessMask\TAccessRight (626)
Value : Cardinal            RecField            SDInfoProcesses\SDDumpAccessMask\TAccessRight (625)



Empty begin/end-blocks (1, was unknown):
----------------------------------------------------------------------------
xxxxxxxxxxxxxxx xxxxx



Empty finally-block (1, was unknown):
----------------------------------------------------------------------------
SDInfoProcesses (414)
```

```
Functions called as procedures (21, was 21):
----------------------------------------------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxx xx xxxxxxxxxxxxxxxxxxxxx xxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxx xx xxxxxxxxxxxxxxxxxxxxx xxxxx

The entire list is not shown in this evaluation version



Duplicate lines (1, was unknown):
----------------------------------------------------------------------------
SDUserSessionManager(285) Log('')



Redeclared identifiers from System unit (1, was unknown):
----------------------------------------------------------------------------
PUnicodeString = PUNICODE_STRING Type, Global       SDUserSessionManager (34)



Identifier with same name as keyword/directive (1, was unknown):
----------------------------------------------------------------------------
Name : UnicodeString       RecField          SDInfoProcesses\SDDumpAccessMask\TAccessRight (626)
```