IMT3501, software security, excercise #04

Jan Samuelsen Lindemann, 120926

October 8, 2014

Abstract

In this excercise, the task was to find vulnerabilities in the SecureDesktop application and describe the process / strategy used.

1 Strategy

1. Find tools
2. Use tools
3. Suggest solutions to findings.

2 Finding tools

The source code used in the SecureDesktop appears to be Pascal/Delphi and finding great tools for the job proved to be difficult.

2.1 Tools found

The most relevant tool i found was Pascal Analyzer[1]
Some of the other tools i found and tried out was:

- Code Healer [2]
- SourceMonitor [3]

I was unable to get Code Healer to work with the sourcecode, not sure why. SourceMonitor seemed to be more of a code metric analysis tool, did not seem particularly relevant at the time.

3 Results

The tool i used, Pascal Analyzer allowed me to analyze each of the delphi project files by itself. The output was sectioned into different groups, depending on what type of vulnerability/error/improvement it found. I have included a summary of the relevant findings in section 5.

While none of the results seemed particularly critical, but mostly where suggestions on improvement to make things run better, or code that wasnt used anywhere else, I have not gone into any detail on these results.

I instead attempted to look through the code manually or without any complex tools.

The first thing that is evident is the lack of documentation/comments in most of the code. This makes it at times very hard to understand the authors intention. Also the code appears to be incomplete.

Looking for hardcoded variable-values by just using grep turns up some results.

```
grep -E [a-zA-Z0-9]*" := "[0-9]+ *
```

the ones worth mentioning:

- Several cases in "SDInfoProcesses.pas" where variables like "nPrivilegeNameSize" or "nPrivilegeDisplayNameSize" are hardcoded to be 255, if for some reason this might need to be changed in the future, it might be better to have this as a global constant incase of refactoring.
- Several cases in "SDInfoProcesses.pas" where variables like "cbName" are hardcoded to 2049. For the same reason as previous case, it might be better to change this incase of refactoring.

However, lack of familiarity with the programming language may void these arguments.

I noticed in some cases that the code will attempt to interact with outside files.

```
grep .exe *
```

This lists many cases where the code attempts to interact with files such as notepad.exe, calc.exe, bds.exe and possibly more. I imagine it might pose a security risk interacting with these files since one typically does not have any guarantee that these files have not been modified in any way.

4 Remediation

The solution I've come up with to the results in this article are mainly to document and comment the code in a better way, such as to allow a better understanding of what is actually happening vs what was intended to happen.

References

[1] Pascal Analyzer,
http://www.peganza.com/

[2] Code Healer,
http://www.socksoftware.com/codehealer.php

[3] SourceMonitor ,
http://www.campwoodsw.com/

5 Output

```
results: SDAppInfo.dpr
Interfaced identifiers that are used, but not outside of unit (13, was 13):
-------------------------------------------------------------------------------

ApplicationDirectoryName : AnsiString
Typed const, Interfaced SDCommon (15)

BackgroundBitmapFileName : AnsiString
Typed const, Interfaced SDCommon (23)


Interfaced class identifiers that are public/published,
but not used outside of unit (7, was 7):
-------------------------------------------------------------------------------

GetBitmap                                   Func, Method
dmScreenshot\TScreenshot (15)

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


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 1):
-------------------------------------------------------------------------------

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

```
results: SDSecureDisplaySVC.dpr
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)


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):
-------------------------------------------------------------------------------

xxxxxxxxxxxxxxx xxxxx


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)

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

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

==========

Results: SDUserSessionManager.dpr

Interfaced identifiers that are used, but not outside of unit (19, was 19):

ApplicationDirectoryName : UnicodeString
Typed const, Interfaced SDCommon (15)

Variables that are set, but never referenced (2, was 2):

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

Value : Cardinal RecField
SDInfoProcesses\SDDumpAccessMask\TAccessRight (625)

BackgroundBitmapFileName : UnicodeString
Typed const, Interfaced SDCommon (23)

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

SDInfoProcesses (405)

Empty finally-block (1, was 1):

SDInfoProcesses (414)

Functions called as procedures (21, was 21):

---

xxxxxxxxxxxxxxxxxxxxxxxxx xx xxxxxxxxxxxxxxxxxxx xxxxx
SDCommon.SDWriteToMailslot at SDUserSessionManager (162)

The entire list is not shown in this evaluation version

Duplicate lines (1, was 1):

---

xxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxx

Redeclared identifiers from System unit (1, was 1):

PUnicodeString = PUNICODE_STRING
Type, Global SDUserSessionManager (34)

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

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