



University of Arizona, Department of Computer Science

CSc 620 — Assignment 1 — Due Mon Sep 8, 23:59 — 20%

Christian Collberg
September 2, 2008

1 Introduction

The purpose of this assignment is to get you familiar with the idea of *code transformations* by studying relevant tools. Some of these tools perform code obfuscation or software watermarking, others can be used to *build* such tools. You will be using some of these tools in future assignments.

This assignment is worth 20% of your final grade.

There are two outcomes of this assignment:

1. A web-page that describes what you learnt about the tool. (10%)
2. A 15-minute presentation where you teach the class about the tool. (10%)

You don't have to prepare separate slides for the presentation, you can lecture straight from the web-page.

You'll be evaluated on the thoroughness of your research, the quality of your prose, and the fluency of your presentation.

Assuming you do a good job, we'll add the web-page to the *Surreptitious Software* web-site!

2 The tools

Tables 1 and 2 list the tools I want you to review. You should look for additional resources: papers about the tools, reviews, other tools/applications that make use of the tool, etc.

3 The review

You should submit a zipfile *toolname.zip* containing a directory *toolname* which contains a file *index.html* plus any other files (such as *jpg* files of screenshots, code samples, etc.) that *index.html* references.

The review should be written in the style of a research paper. Avoid value-judgments. Use correct and formal English. This isn't your Facebook page. Don't use lists unless absolutely necessary; write readable prose.

Figure 1 lists the *index.html* template file that you should download and modify.

#	tool	description	student	presentation
1	SANDMARK	Java code obfuscation, watermarking, birthmarking tool <ul style="list-style-type: none"> • http://sandmark.cs.arizona.edu • http://ieeexplore.ieee.org/iel5/8013/27399/01219058.pdf?arnumber=1219058 	ricarlos	Thu Sep 18
2	Zelix classmaster	Commercial Java obfuscation tool <ul style="list-style-type: none"> • http://www.zelix.com/klassmaster/download1.html 	ghigliom	Tue Oct 7
3	SourceAgain	Commercial Java decompiler <ul style="list-style-type: none"> • http://www.ahpah.com/product.html 	ghigliom	Tue Oct 7
4	preemptive	Commercial Java/.NET obfuscator <ul style="list-style-type: none"> • http://www.preemptive.com/dasho-java-obfuscator.html • http://www.freepatentsonline.com/6102966.html 	tpatki	Tue Oct 7
5&6	SOOT	Java bytecode manipulation tool <ul style="list-style-type: none"> • http://www.sable.mcgill.ca/soot 	robackja & sushanth	Thu Sep 18
7	jmark	Java bytecode watermarking tool <ul style="list-style-type: none"> • http://se.naist.jp/jmark 	qingju	Tue Oct 7

Table 1: Java/.Net tools

#	tool	description	student	presentation
8	LOCO	x86 code obfuscation tool <ul style="list-style-type: none"> • http://diablo.elis.ugent.be/obf_download 	anandp	Thu Sep 18
9	STILO	<i>SToring Information in Linked Object-files</i> — x86 watermarking tool <ul style="list-style-type: none"> • http://diablo.elis.ugent.be/stilo 	mbi	Thu Sep 18
10	burneye	Unix executable encryption tool <ul style="list-style-type: none"> • http://www.securiteam.com/tools/5BPOHOU7PQ.html • http://bismark.extracon.it/linux/teso • http://www.woodmann.com/collaborative/tools/index.php/Burndump • http://www.woodmann.net/collaborative/tools/images/Bin_Burneye_2007-12-7_9.51_burneye-1.0.1-src.tar.zip 	bhandari	Tue Oct 7
11	boomerang	x86 binary decompiler <ul style="list-style-type: none"> • http://boomerang.sourceforge.net 	marshall	Tue Oct 7
12&13	CIL&O'Caml	Tool to translate C into an intermediate code that allows you to manipulate it in the O'Caml language. <ul style="list-style-type: none"> • http://sourceforge.net/projects/cil • http://manju.cs.berkeley.edu/cil • http://caml.inria.fr/ 	kpcoogan & jamyers	Thu Sep 18

Table 2: Binary and C source code tools.

```

<html>
<body>

<title > NAME OF TOOL </title >
<center>
  <h1> NAME OF TOOL </h1>
  <br>
  <h3>Reviewed by: YOUR NAME</h3>
</center>

<h2>Abstract</h2>
<em>Short paragraph describing the tool</em>

<h2>Introduction</h2>
<em>What does the tool do, who wrote it, is it
open source/commercial, is it currently maintained,
when was it written, etc </em>

<h2>Installation</h2>
<em>How is the tool downloaded/built/installed /...? </em>

<h2>Usage</h2>
<em>How is the tool used? If appropriate,
give code examples. </em>

<h2>Internals</h2>
<em>If you've got access to source code, papers, etc,
describe algorithms used. </em>

<h2>Evaluation</h2>
<em>Give examples where the tool fails and succeeds.
Can you make it crash, generate the wrong code?
Does it work for small programs, large programs,
simple programs, complex programs?
If appropriate, give code examples. If appropriate,
evaluate performance. Be specific in your
evaluation. Don't use value-judgements like
"it 's good/bad". </em>

<h2>Screenshots</h2>
<em>If appropriate, give screenshots </em>

<h2>References</h2>
<ul>
  <li> <em> Where to download </em>
  <li> <em> Papers about the tool </em>
  <li> <em> Anything else relevant </em>
</ul>
</body>
</html>

```

Figure 1: The `index.html` template file.

4 The presentation

Follow the outline of the web-page in your presentation. Focus on what the tool does, how it's used, what it does well/poorly. Some of these tools you might be using in your projects, so the presentation is like an advertisement for the tool, and you want your fellow classmates to know when and how they can use it. If you have the time, a small demo is a good idea. Remember that you only have 15 minutes, so plan and rehearse the talk well!

5 OK, what should I do???

Here's a rough guide to how you should go about this assignment:

1. Start by downloading and installing the tool. Do this *immediately* so that you can be sure that the tool exists, can be built, and actually does something reasonable!
2. Download and read any documentation.
3. Search for additional information about the tool: wikipedia, google scholar, citeseer, google, etc.
4. Learn how to use the tool. Depending on the nature of the tool, this may mean you have to write little test programs to submit to the tool, code snippets which invoke the tool, etc.
5. Download the `toolname.zip` template directory from <http://www.cs.arizona.edu/~collberg/Teaching/620/2008/Assignments>.
6. Write the evaluation.
7. A week before the due-date you should come and show me what you have. You are, of course, welcome to come and talk to me at any time about any problems you might have!
8. Make sure that all files are present, that you're only using relative references to files in your directory, and that you're not using any weird html that doesn't display well in all browsers.
9. Submit `toolname.zip` by emailing to `collberg@gmail.com`. Make sure that the zip-file contains exactly one directory, with all the relevant files.
10. Prepare your presentation. Prepare a demo if appropriate. Rehearse!

```

-
\'*-
) _'-
. : ' .
: - ' \
; *' - ' *- -
' - -)
; ' ' .
:. . \
. \ . : '-) .
) '+; ; ) :
: ' | ; ;-
; ' : '-: - '* ;
.*' / .*' ; .*'- +' '*'
'**- '** '*-*
```