Software Security

Introduction

Author: Prof Bill Buchanan

Software Security

Software Flaws

Author: Prof Bill Buchanan

Non-intentional. These can be:

- **Validation flaws**. The code fails to check for valid input data.
- **Domain flaws**. This is where data leaks from one program to another.
- **Serialisation flaws**. This is where data changes while being passed from one program to another.
- **Identification/Authentication flaws**. This is where there is a lack of identification for processes or users.
- **Boundary condition flaws**. This is where resource access is not checked, and can thus allow an external hacker to use up resources.
- **Logic flaws**.

Bug

Trap-door

Intentional. This can either be caused by malious code (such as a Trojan or back-door programs).

Author: Prof Bill Buchanan

Software Security

Best Practice

Author: Prof Bill Buchanan

Best practice

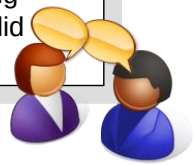
Software Security

Principle of least privilege

Processes and scripts should run with the least privilege possible, to minimize damage

Never trust user input

All user input check be checked before it is used. This includes checking for correct number/string format, including valid characters.



Defence-in-depth

Checkpoints should be added for authentication and authorization at software interfaces, and interfaces within modules.

Use secure defaults

Sometimes developers encounter security problems in running and applications. It is important that these are fully tested before reducing the security.

Never rely on obfuscation

Obfuscation of code just makes it more difficult to determine its operation. If an intruder wants to "crack" a program, then normally can, so other methods of securing the code should be employed.

If it's not used … disable it!

Any services which can be accessed can be compromised, thus, if they are not needed, they should be disabled.

Authenticate at the front-end

In terms of resources, it is often better to authentication at the front-end rather than the back-end

Never trust external systems

External systems should always be seen as a potential risk, and should never be fully trusted.

Reduce Surface Area

This should minimize the information that can be accessed from outside, and to handle errors in a graceful way.

System is only as secure as the weakest link

The overall security of a system is only as strong as its weakest link.

Based on Microsoft ASP.NET Good Practice Guide

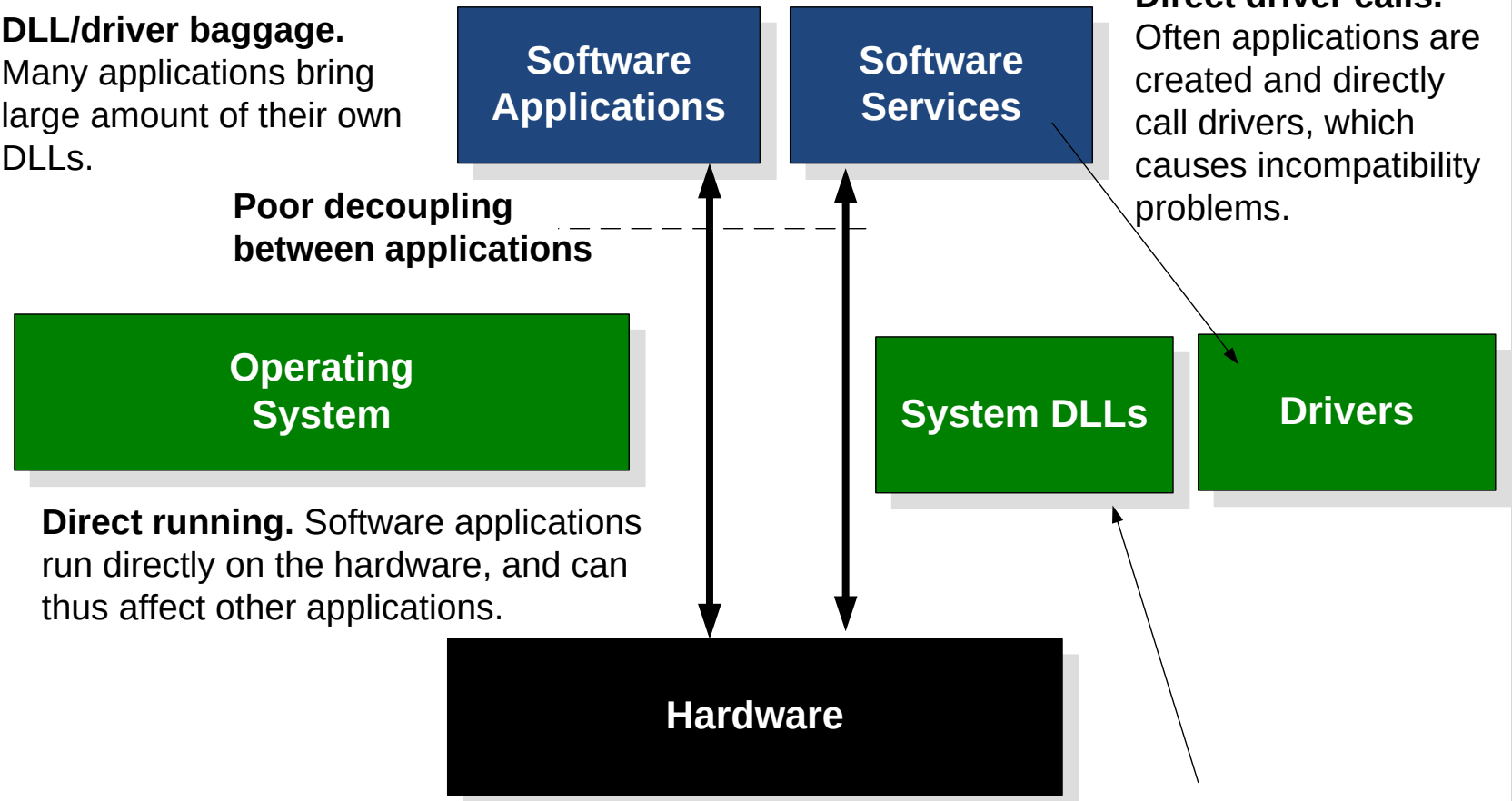
Author: Prof Bill Buchanan

Binding to OS. Some applications are bound to the OS version, and do not work with other OSs or versions.

Bind to hardware. Often application programs are compiled to a certain hardware/architecture.

DLL/driver baggage. Many applications bring large amount of their own DLLs.

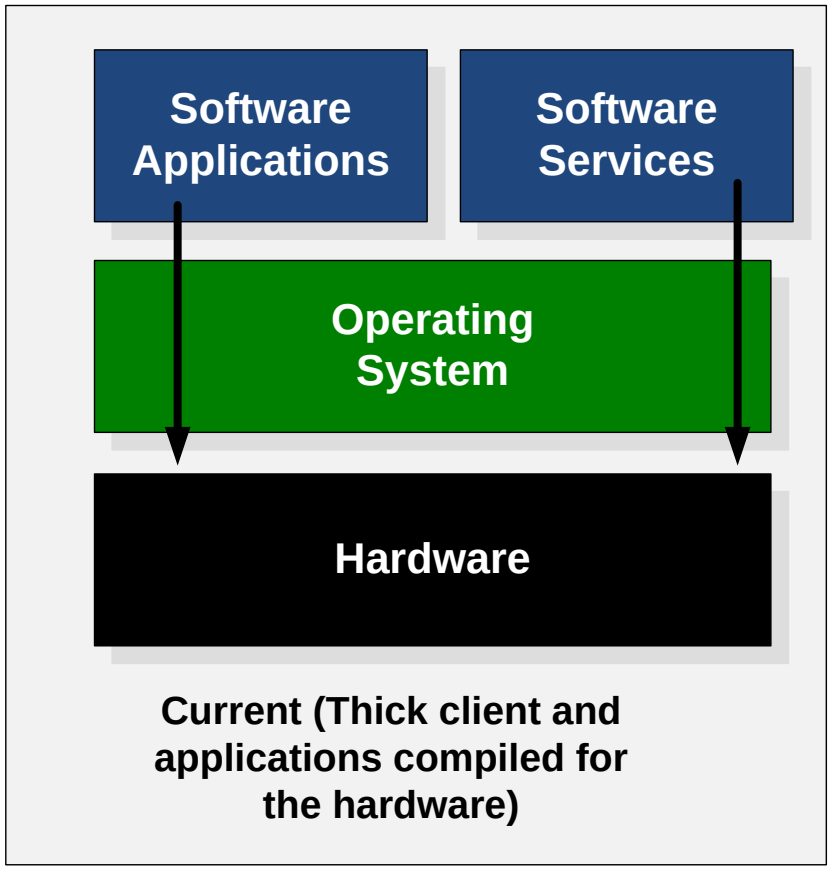
Direct driver calls. Often applications are created and directly call drivers, which causes incompatibility problems.



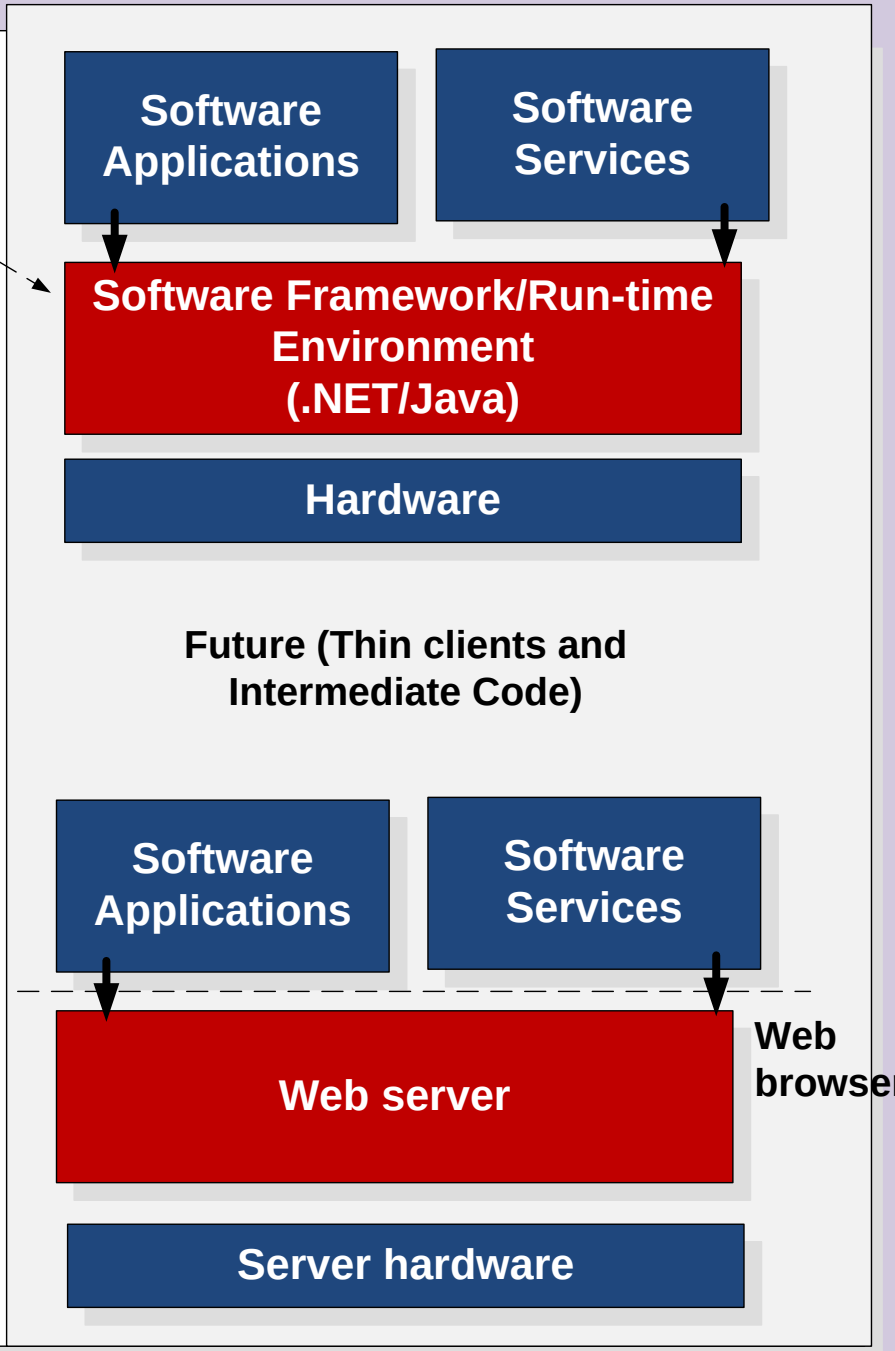
Direct running. Software applications run directly on the hardware, and can thus affect other applications.

DLL Hell. This is where DLL can be replaced with incorrect ones, such as being overwritten with malious ones or wrong version

Applications are independent of the hardware and run within the framework. The framework manages their operation, to make sure they do not damage the system or other applications



Current (Thick client and applications compiled for the hardware)



Future (Thin clients and Intermediate Code)

Author: Prof Bill Buchanan

Software Security

Keeping Code Secure

Author: Prof Bill Buchanan

```
using System;

namespace simple
{
    class Class1
    {
        static void Main(string[] args)
        {
            string name;

            System.Console.Write("What is your name?");
            name=System.Console.ReadLine();
            System.Console.WriteLine("Hello " + name);
        }
    }
}
```

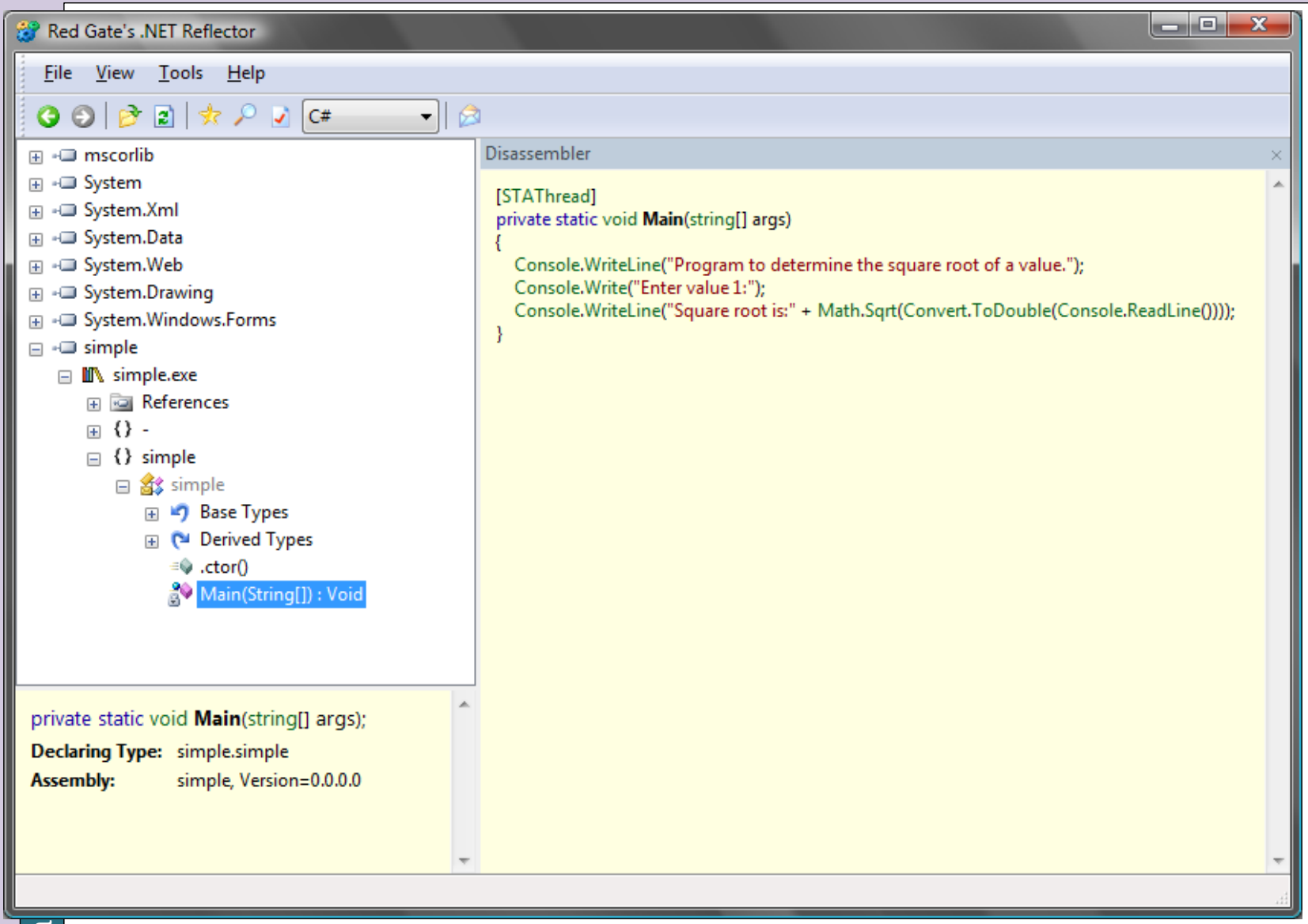
EXE

```
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

```
F:\docs\src\simple\test>dir
 Volume in drive F has no label.
 Volume Serial Number is 2886-0553

 Directory of F:\docs\src\simple\test

25/07/2008  01:20    <DIR>          .
25/07/2008  01:20    <DIR>          ..
28/01/2007  17:06               437 simple.cs
               1 File(s)            437 bytes
               2 Dir(s)  113,418,530,816 bytes free

F:\docs\src\simple\test>csc simple.cs
Microsoft (R) Visual C# 2008 Compiler version 3.5.21022.8
for Microsoft (R) .NET Framework version 3.5
Copyright (C) Microsoft Corporation. All rights reserved.


F:\docs\src\simple\test>dir
 Volume in drive F has no label.
 Volume Serial Number is 2886-0553

 Directory of F:\docs\src\simple\test

15/09/2008  16:37    <DIR>          .
15/09/2008  16:37    <DIR>          ..
28/01/2007  17:06               437 simple.cs
15/09/2008  16:37             4,096 simple.exe
               2 File(s)          4,533 bytes
               2 Dir(s)  113,418,526,720 bytes free

F:\docs\src\simple\test>simple
Program to determine the square root of a value.
Enter value 1:9
Square root is:3
```

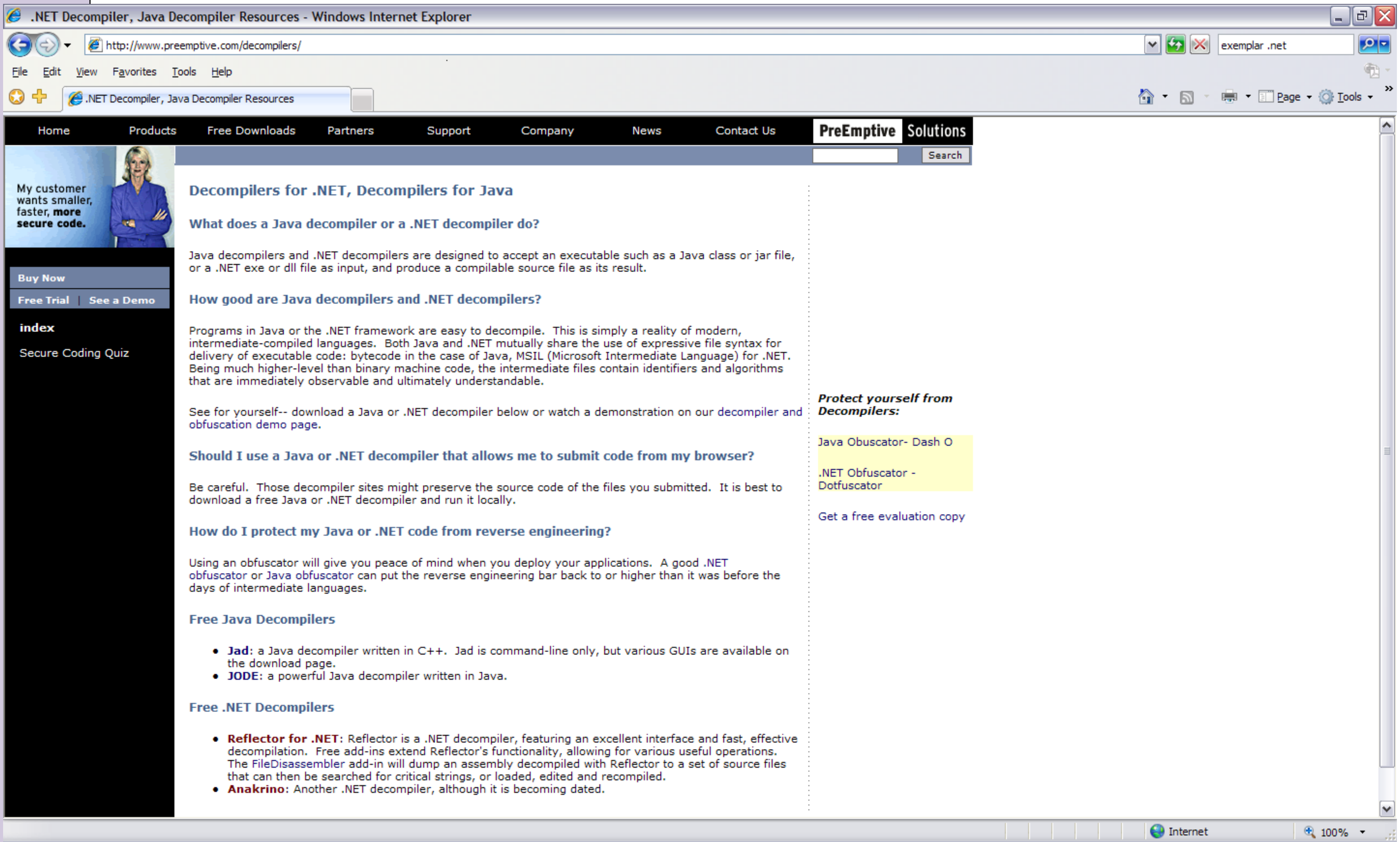
Run program

```
F:\docs\src\simple\test>exemplar simple.exe > list.cs

F:\docs\src\simple\test>dir
 Volume in drive F has no label.
 Volume Serial Number is 2886-0553

 Directory of F:\docs\src\simple\test

15/09/2008  16:38    <DIR>          .
15/09/2008  16:38    <DIR>          ..
15/09/2008  16:38               451 list.cs
28/01/2007  17:06               437 simple.cs
15/09/2008  16:37             4,096 simple.exe

F:\docs\src\simple\test>type list.cs
namespace simple {
        class simple {

                [STAThread]
                private static void Main(string[] args) {
                        double local0;
                        double local1;

                        Console.WriteLine("Program to determine the
square root of a value.");
                        Console.Write("Enter value 1:");
                        local0 = Convert.ToDouble(Console.ReadLine());
                        local1 = Math.Sqrt(local0);
                        Console.WriteLine("Square root is:" + local1);
                }

                public simple() : base() {
                }
        }
}
```



Softwa

Disassembler



.NET Decompiler, Java Decompiler Resources - Windows Internet Explorer

http://www.preemptive.com/decompilers/

Home | Products | Free Downloads | Partners | Support | Company | News | Contact Us | PreEmptive Solutions

My customer wants smaller, faster, **more secure code.**

Buy Now

Free Trial | See a Demo

index

Secure Coding Quiz

Decompilers for .NET, Decompilers for Java

What does a Java decompiler or a .NET decompiler do?

Java decompilers and .NET decompilers are designed to accept an executable such as a Java class or jar file, or a .NET exe or dll file as input, and produce a compilable source file as its result.

How good are Java decompilers and .NET decompilers?

Programs in Java or the .NET framework are easy to decompile. This is simply a reality of modern, intermediate-compiled languages. Both Java and .NET mutually share the use of expressive file syntax for delivery of executable code: bytecode in the case of Java, MSIL (Microsoft Intermediate Language) for .NET. Being much higher-level than binary machine code, the intermediate files contain identifiers and algorithms that are immediately observable and ultimately understandable.

See for yourself-- download a Java or .NET decompiler below or watch a demonstration on our decompiler and obfuscation demo page.

Should I use a Java or .NET decompiler that allows me to submit code from my browser?

Be careful. Those decompiler sites might preserve the source code of the files you submitted. It is best to download a free Java or .NET decompiler and run it locally.

How do I protect my Java or .NET code from reverse engineering?

Using an obfuscator will give you peace of mind when you deploy your applications. A good .NET obfuscator or Java obfuscator can put the reverse engineering bar back to or higher than it was before the days of intermediate languages.

Free Java Decompilers

- **Jad**: a Java decompiler written in C++. Jad is command-line only, but various GUIs are available on the download page.
- **JODE**: a powerful Java decompiler written in Java.

Free .NET Decompilers

- **Reflector for .NET**: Reflector is a .NET decompiler, featuring an excellent interface and fast, effective decompilation. Free add-ins extend Reflector's functionality, allowing for various useful operations. The FileDisassembler add-in will dump an assembly decompiled with Reflector to a set of source files that can then be searched for critical strings, or loaded, edited and recompiled.
- **Anakrino**: Another .NET decompiler, although it is becoming dated.

Protect yourself from Decompilers:

Java Obuscator- Dash O

.NET Obfuscator - Dotfuscator

Get a free evaluation copy

Softw

Dotfuscator

Software Security

Obfuscation

Author: Prof Bill Buchanan

```
#include <stdio.h> main(t,_,a)char
*a;{return!0<t?t<3?main(-79,-13,a+main(-87,1-_,
main(-86,0,a+1)+a)):1,t<_?main(t+1,_,a):3,main(-
94,-27+t,a)&&t==2?_<13? main(2,_+1,"%s %d %d\
n"):9:16:t<0?t<-72?main(_,t, "@n'+,#'/*{}w+/
w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/
n{n+,/+#n+,/#\ ;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K
w'K:'+}e#';dq#'l \ q#'+d'K#!/
+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n'){)#}w'){){nl]'/
+#n';d}rw' i;# \ ){nl]!/n{n#'; r{#w'r nc{nl]'/
#{l,+'K {rw' iK{;[{nl]'/w#q#n'wk nw' \
iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/
*{q#'ld;r'}{nlwb!/*de}'c \ ;;{nl'-{}rw]'/
+,}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/ ')
}+}{rl#'{n' ')# \ }'+}##(!!/") :t<-
50?_==*a?putchar(31[a]):main(-
65,_,a+1):main((*a=='/')+t,_,a+1)
:0<t?main(2,2,"%s"):*a=='/'||main(0,main(-61,*a,
"!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m
.vpbks,fxntdCeghiry"),a+1);}
```

What does this program do?

On the first day of Christmas,
my true love sent to me
A partridge in a pear tree.
On the second day of Christmas,
my true love sent to me
Two turtle doves,
And a partridge in a pear tree….

Elimination of all whitespace.
Use of conditional and list expression instead of the more familiar if-then-else statement and statement blocks.
A simple encoding of the poem's strings.
Encoding of multiple "functions" into the single function main

Identifier Renaming

This involves renaming all the classes, methods, and fields to short names, or even non-printing names

Before

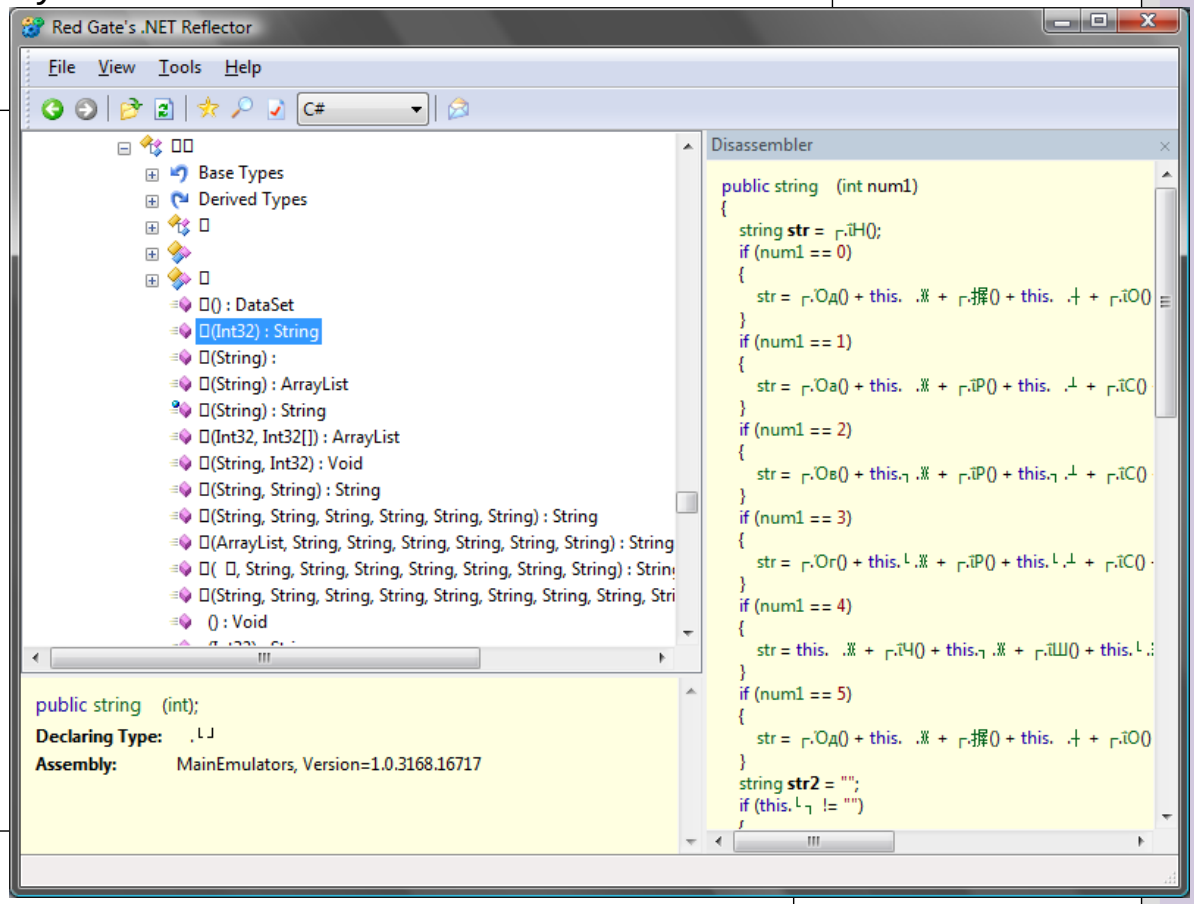
```
namespace Emulator {
        public class gen_switch {
          public ArrayList32logging59
          public ArrayList32level5Commands59
          public string32level5Name59
          public ArrayList32level69Commands59
          public string32level69Name59
          public ArrayList32level39Commands59
          public string32level39Name59
          public ArrayList32level16Commands59
          public string32level16Name59
          public ArrayList32level17Commands59
          public string32level17Name59
          public ArrayList32level25Commands59
          public string32level25Name59
```

After

```
C:\netwsims>exemplar mainemulators.exe
namespace ☺ {
        public class ☺ {
                public ArrayList32☺59
                public ArrayList32☻59
                public string32☺59
                public ArrayList32♥59
                public string32☻59
                public ArrayList32♦59
                public string32♥59
                public ArrayList32♣59
                public string32♦59
                public ArrayList32♠59
                public string32♣59
                public ArrayList32♫59
```

Author: Prof Bill Buchanan

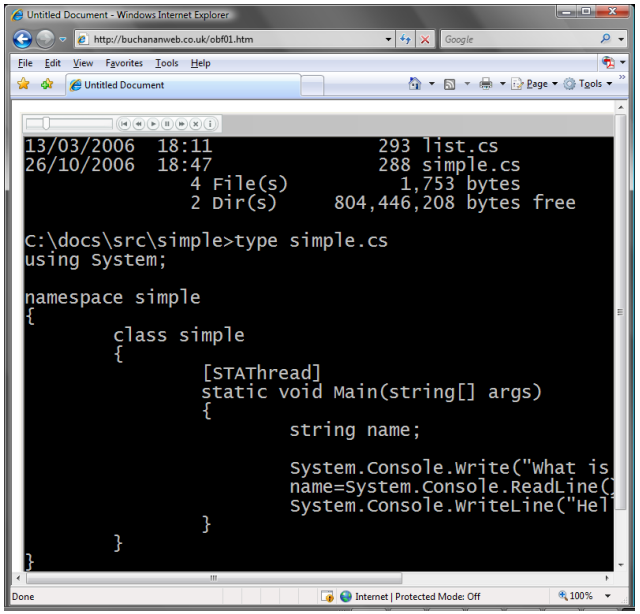
String Encryption
A standard way that many “crackers” work is to search for key strings within a program. String encryption is used to encrypt strings through an encryption method, so that they cannot be search for.

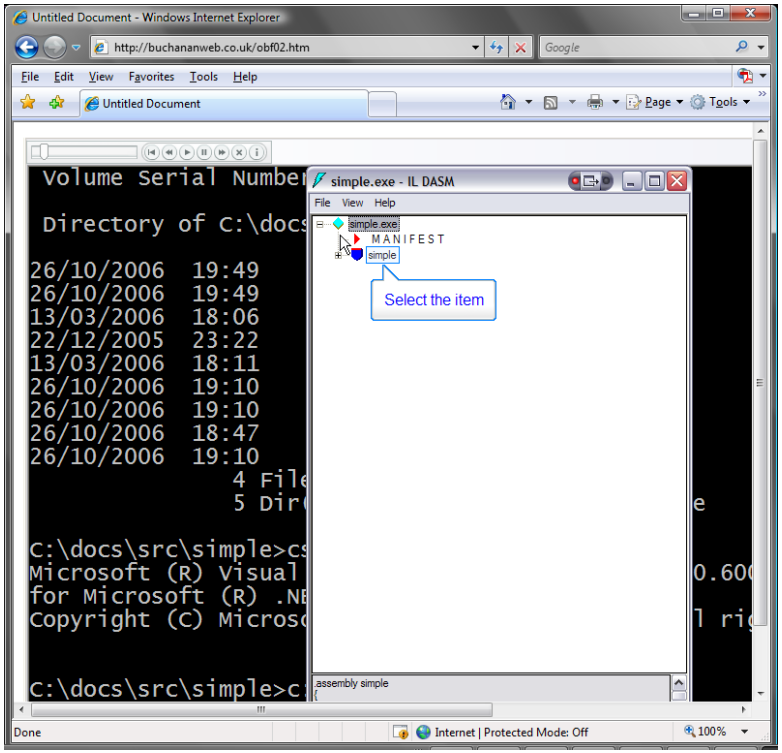


Flow obfuscation
This involves scrambling the flow of the program, so that it is difficult to determine its actual operation.

View demo of Obfuscation

http://buchananweb.co.uk/obf01.htm





View demo of Obfuscation2

http://buchananweb.co.uk/obf02.htm

```
using System;

namespace simple
{
    class Class1
    {
        static void Main(string[] args)
        {
            string name;

            System.Console.Write("What is your name?");
            name=System.Console.ReadLine();
            System.Console.WriteLine("Hello " + name);
        }
    }
}
```

EXE

```
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

Class1::Main : void(string[])

```
.method private hidebysig static void  Main(string[] args) cil managed
{
  .entrypoint
  .custom instance void [mscorlib]System.STAThreadAttribute::.ctor() = ( 01
  // Code size       33 (0x21)
  .maxstack  2
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

Variables have been renamed

```
) cil managed

te::.ctor() = ( 01
  .maxstack  2
  .locals init (string V_0)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void class [mscorlib]System.Console::Write(string)
  IL_000a:  call       string class [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string string::Concat(string,
                                             string)
  IL_001b:  call       void class [mscorlib]System.Console::WriteLine(string
  IL_0020:  ret
} // end of method a::Main
```

Classes have been renamed

Class1::Main : void(string[])

```
.method private hidebysig static void  Main(string[] args) cil managed
{
  .entrypoint
  .custom instance void [mscorlib]System.STAThreadAttribute::.ctor() = ( 01
  // Code size       33 (0x21)
  .maxstack  2
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

Variables have been renamed

Microsoft Visual C++ Runtime Library

Runtime Error!

Program: c:\exemplar.exe

This application has requested the Runtime to terminate it in an unusual way.
Please contact the application's support team for more information.

OK

```
  .maxstack
  .locals i
  IL_0000:
  IL_0005:                                                    string)
  IL_000a:                                                    Line()
  IL_000f:
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string string::Concat(string,
                                             string)
  IL_001b:  call       void class [mscorlib]System.Console::WriteLine(string
  IL_0020:  ret
} // end of method a::Main
```

Classes have been renamed