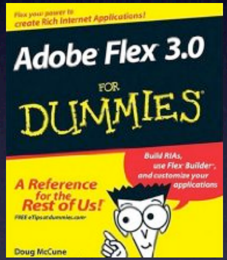
Steal this Code

Decompiling SWFs for fun and profit

Doug McCune

Who am I?

- Flex developer
- Blogger: dougmccune.com
- Author: Adobe Flex 3.0 for Dummies



- Principal Software Engineer @ UNIVERSAL MIND

SpatialKey

doug@dougmccune.com

What am I going to talk about?

- What's in a SWF?
- Decompiling tools
- Example: decompiling Photoshop Express
- What you get, what you don't
- Example: fucking with Natzke
- Was that made with _______?
- Security implications
- Example: implementing the Photoshop Express filters
- Obfuscation and Encryption
- The moral lesson

What’s a SWF?

- ActionScript Byte Code (ABC)
- Embedded graphical assets (swf, png, etc)
- Shapes drawn in Flash Authoring
- Frames, timeline nonsense, and the rest of the weird stuff you designers like

For more than you ever wanted to know about SWF files:
http://www.adobe.com/devnet/swf/
http://www.adobe.com/devnet/swf/pdf/swf_file_format_spec_v9.pdf
http://www.adobe.com/devnet/actionscript/articles/avm2overview.pdf

What’s a SWF?

the good stuff

- ActionScript Byte Code (ABC)
- ~~Embedded graphical assets (swf, png, etc)~~
- ~~Shapes drawn in Flash Authoring~~
- ~~Frames, timeline nonsense, and the rest of the weird stuff you designers like~~

I don’t care

For more than you ever wanted to know about SWF files:
http://www.adobe.com/devnet/swf/
http://www.adobe.com/devnet/swf/pdf/swf_file_format_spec_v9.pdf
http://www.adobe.com/devnet/actionscript/articles/avm2overview.pdf

Decompiling Tools

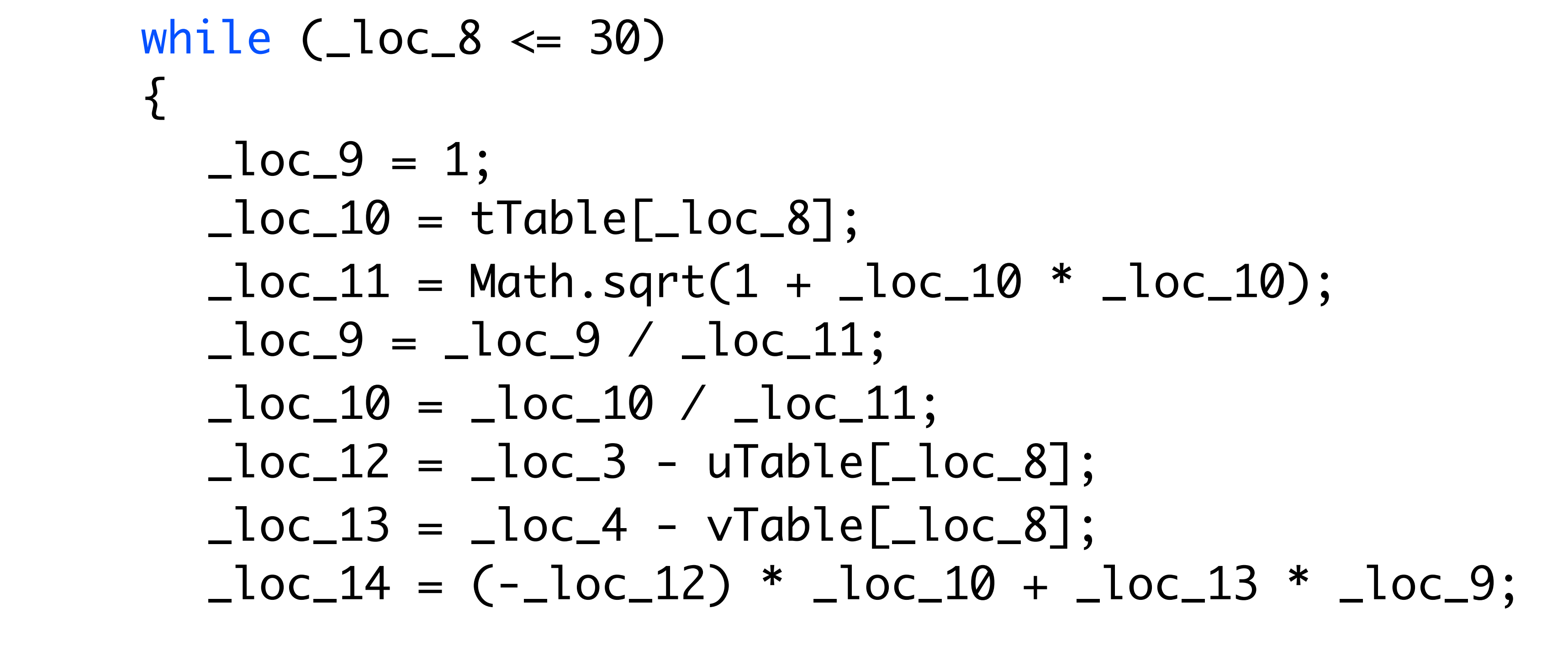
- Sothink SWF Decompiler
  - $80, Mac or PC, works great

- Nemo 440
  - free AIR app, generates ABC bytecode
- swfdump
  - free, included in Flex SDK, dumps ABC bytecode
- abcdump.as
  - free, part of the Tamarin project
- ActionScript Viewer (ASV)
  - $80 only AS3 support in “prerelease” version, PC only

How easy is it?

Let’s decompile Photoshop Express
in a few seconds

But there’s a catch



```
    while (_loc_8 <= 30)
    {
       _loc_9 = 1;
       _loc_10 = tTable[_loc_8];
       _loc_11 = Math.sqrt(1 + _loc_10 * _loc_10);
       _loc_9 = _loc_9 / _loc_11;
       _loc_10 = _loc_10 / _loc_11;
       _loc_12 = _loc_3 - uTable[_loc_8];
       _loc_13 = _loc_4 - vTable[_loc_8];
       _loc_14 = (-_loc_12) * _loc_10 + _loc_13 * _loc_9;
```

* actual code from Photoshop Express

What you get and what you don’t

You get:

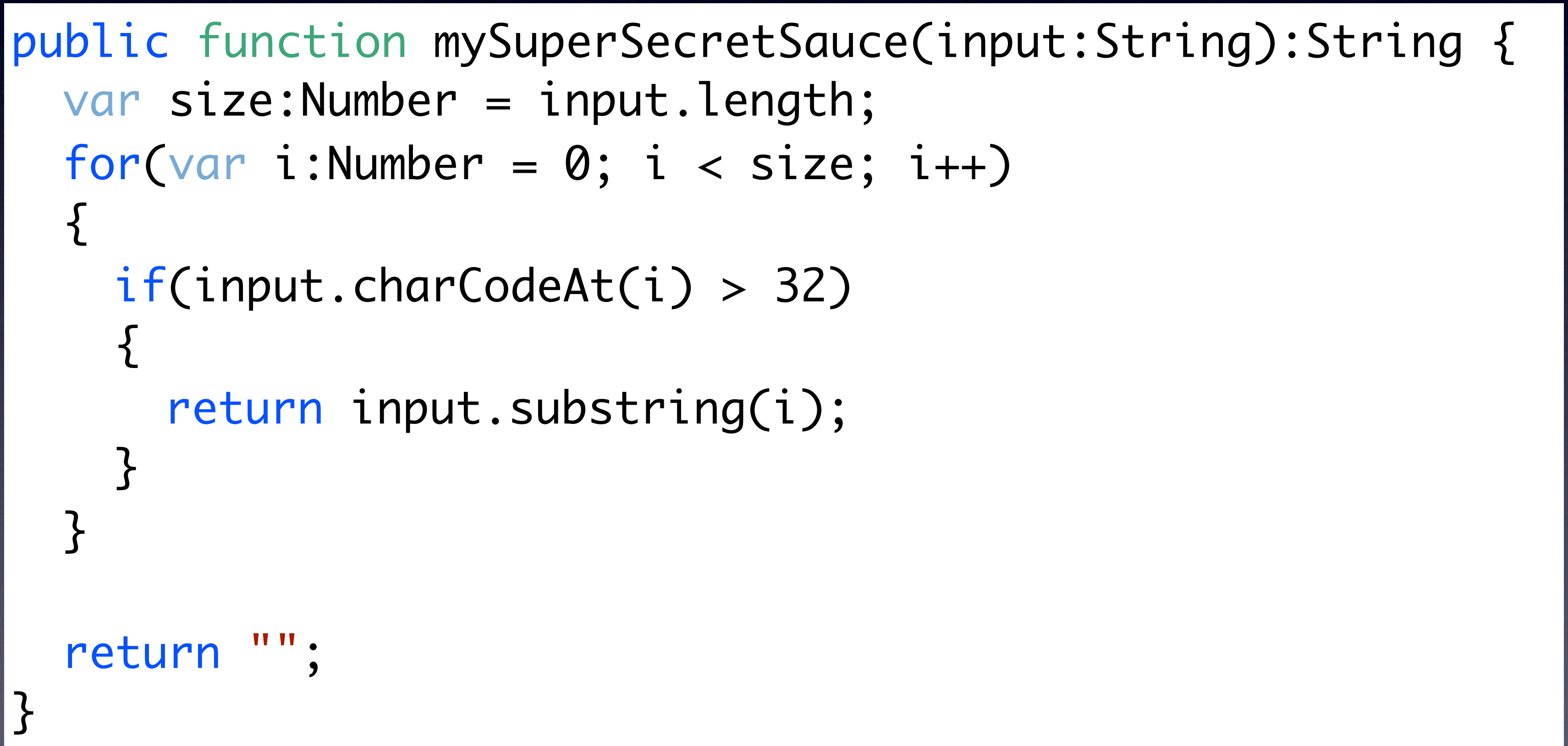
- package structure
- class names
- method names and signatures
- class-level variable names

You don’t get:

- most local variable names
- method parameter names
- for loops (turn into while loops)
- sometimes initial variable assignments

Comparing ActionScript, ABC bytecode, and decompiled source

Example ActionScript function:



```
public function mySuperSecretSauce(input:String):String {
  var size:Number = input.length;
  for(var i:Number = 0; i < size; i++)
  {
    if(input.charCodeAt(i) > 32)
    {
      return input.substring(i);
    }
  }

  return "";
}
```

Comparing ActionScript, ABC bytecode, and decompiled source

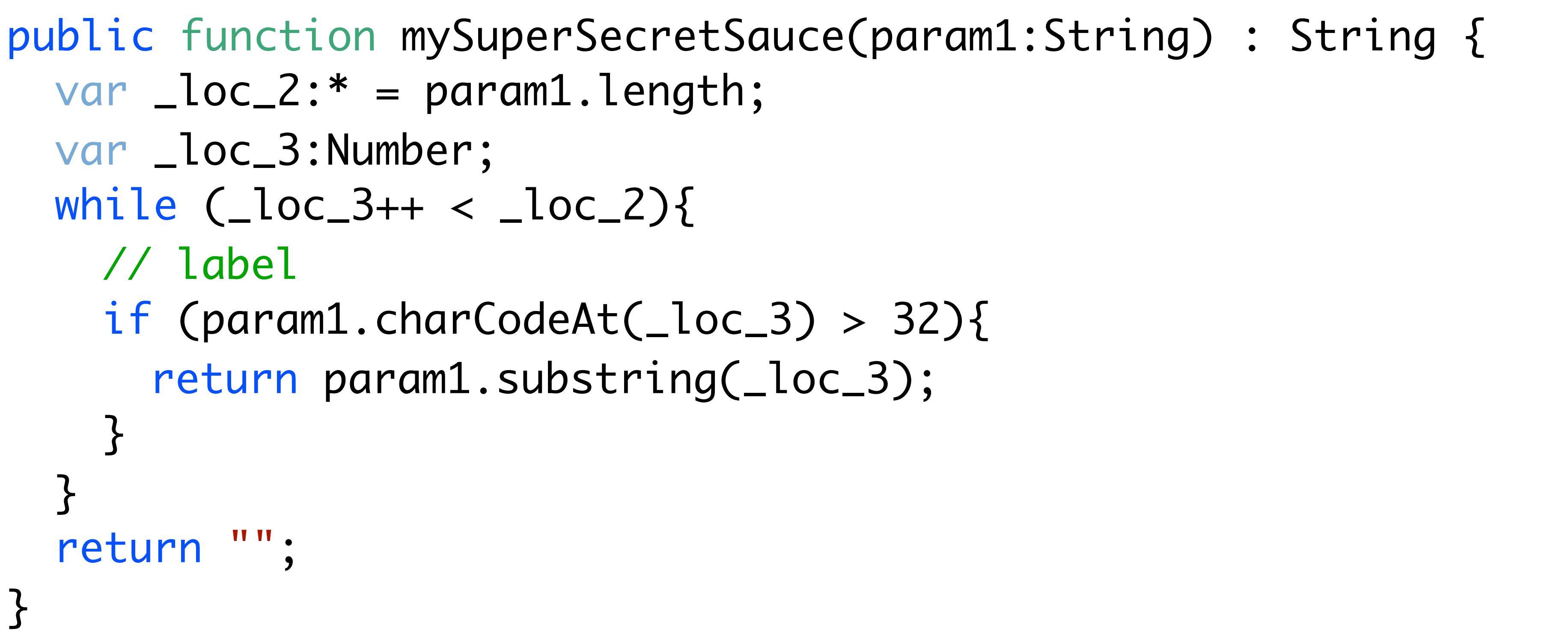
ABC bytecode:

```
    function mySuperSecretSauce(String):String   {
        0       getlocal0
        1       pushscope
        2       getlocal1
        3       getproperty     length
        5       convert_d
        6       setlocal2
        7       pushbyte        0
        9       convert_d
        10      setlocal3
        11      jump            L1


        L2:
        15      label
        16      getlocal1
        ...
```

Comparing ActionScript, ABC bytecode, and decompiled source

Decompiled source (from Sothink decompiler):



```
public function mySuperSecretSauce(param1:String) : String {
  var _loc_2:* = param1.length;
  var _loc_3:Number;
  while (_loc_3++ < _loc_2){
    // label
    if (param1.charCodeAt(_loc_3) > 32){
      return param1.substring(_loc_3);
    }
  }
  return "";
}
```

Common problems

- For loops become while loops
- Initial variable values are lost
- Duplicated new object creation

Original decompiled code:

```
public function adjustHue(param1:Number):void {
   param1 = param1 * (Math.PI / 180);
   var _loc_2:* = Math.cos(param1);
   var _loc_3:* = Math.sin(param1);
   var _loc_4:Number;
   var _loc_5:Number;
   var _loc_6:Number;
   var _loc_7:* = new Array(...);
   concat(_loc_7);
   return;
}
```

ABC (byte code)

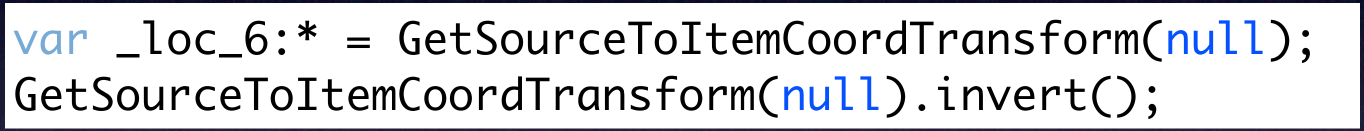
```
_as3_setlocal <3>
_as3_pushdouble 0.213
_as3_convert_d
_as3_setlocal <4>
_as3_pushdouble 0.715
_as3_convert_d
_as3_setlocal <5>
_as3_pushdouble 0.072
_as3_convert_d
_as3_setlocal <6>
```

“Correct” decompiled code:

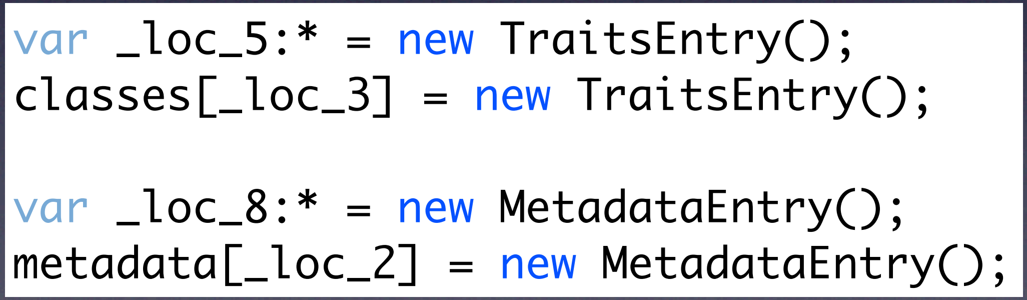
```
public function adjustHue(param1:Number):void {
   param1 = param1 * (Math.PI / 180);
   var _loc_2:* = Math.cos(param1);
   var _loc_3:* = Math.sin(param1);
   var _loc_4:Number = .213;
   var _loc_5:Number = .715;
   var _loc_6:Number = .072;
   var _loc_7:* = new Array(...);
   concat(_loc_7);
   return;
}
```

Common problems

- Duplicated new object creation



```
var _loc_6:* = GetSourceToItemCoordTransform(null);
GetSourceToItemCoordTransform(null).invert();
```



```
var _loc_5:* = new TraitsEntry();
classes[_loc_3] = new TraitsEntry();

var _loc_8:* = new MetadataEntry();
metadata[_loc_2] = new MetadataEntry();
```

Fucking with Natzke

Was that made with _________?

You can quickly check what projects were used with simple package checking:

- **Flex:** mx.*

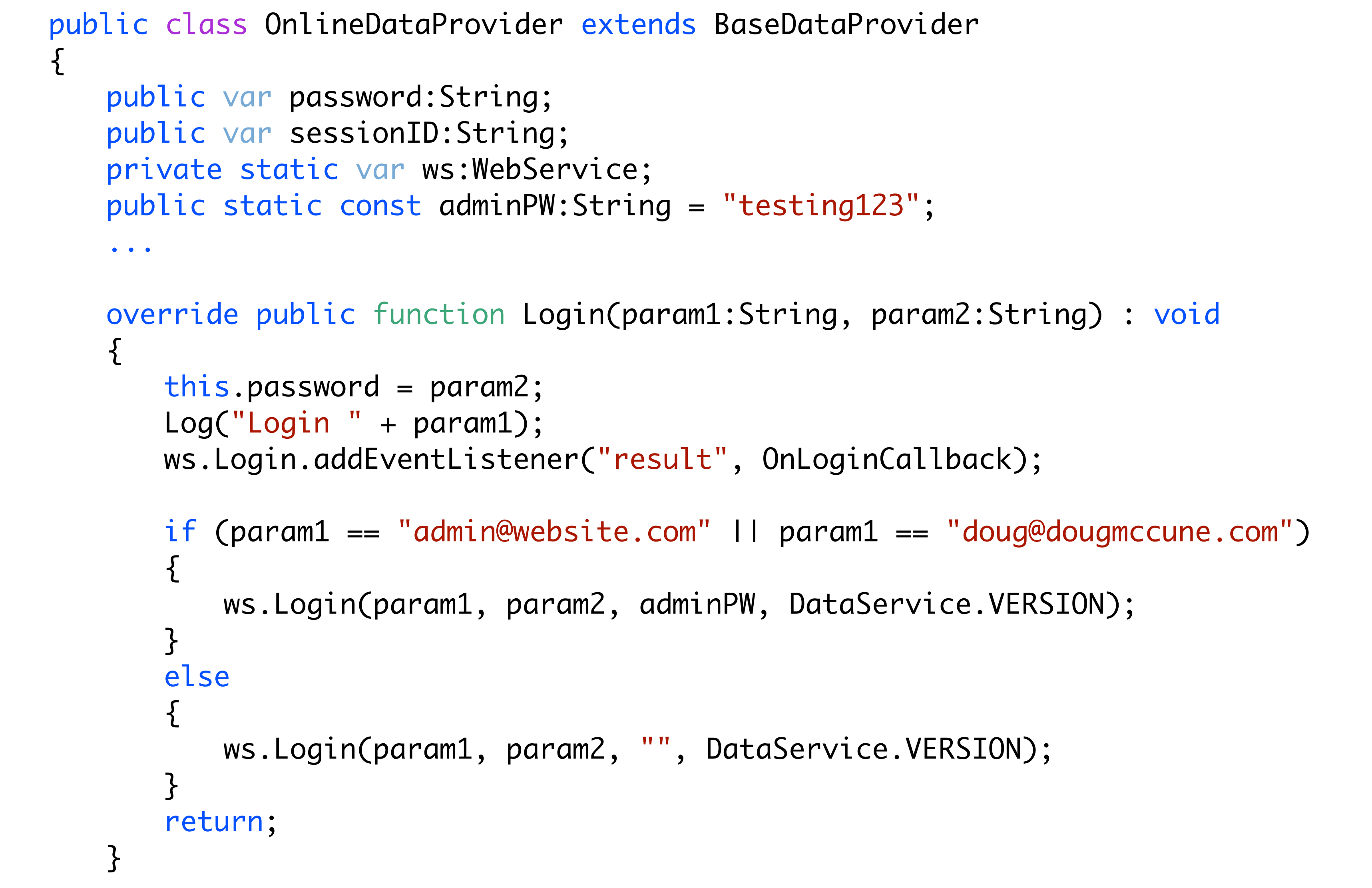
  mx.core.Application.VERSION
- **PaperVision 3D**: org.papervision3D.*

  org.papervision3D.Papervision3D.VERSION
- **Away3D**: away3d.*
- **AS3 Crypto**: com.hurlant.*
- **Box 2D**: Box2D.*

Security Implications

- NOTHING is safe in your SWF
- NEVER store passwords of any kind
- Secure your services on the web server!
- Clean out any “developer” or “debug” mode magic before publishing

Don’t do this!



```
public class OnlineDataProvider extends BaseDataProvider
{
    public var password:String;
    public var sessionID:String;
    private static var ws:WebService;
    public static const adminPW:String = "testing123";
    ...

    override public function Login(param1:String, param2:String) : void
    {
        this.password = param2;
        Log("Login " + param1);
        ws.Login.addEventListener("result", OnLoginCallback);

        if (param1 == "admin@website.com" || param1 == "doug@dougmccune.com")
        {
            ws.Login(param1, param2, adminPW, DataService.VERSION);
        }
        else
        {
            ws.Login(param1, param2, "", DataService.VERSION);
        }
        return;
    }
```

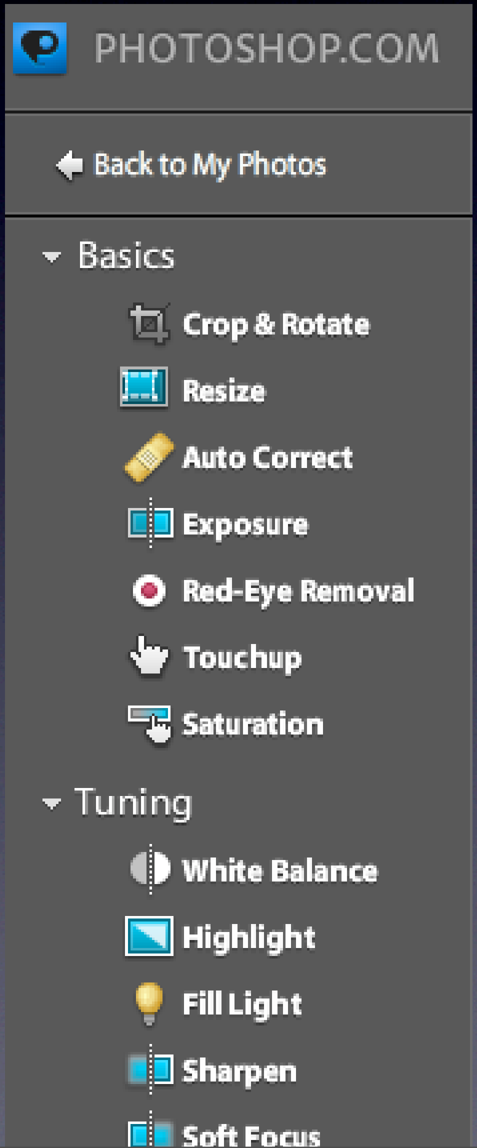
Stealing Photoshop Express filters

- Image manipulation algorithms are mostly just math

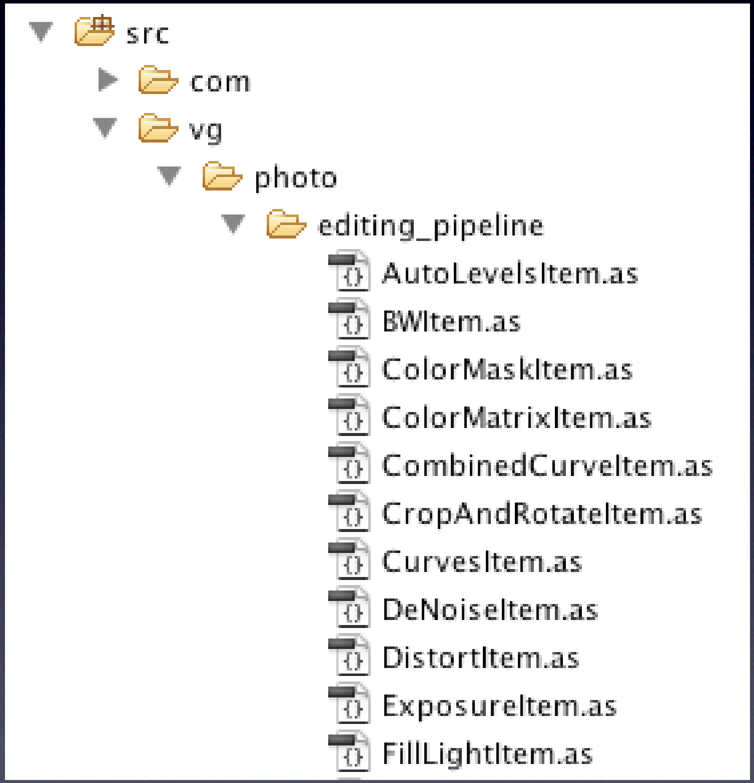
  (algorithms = easy to steal)

- I can ignore all UI components

  (UI stuff and app layout = hard to steal)

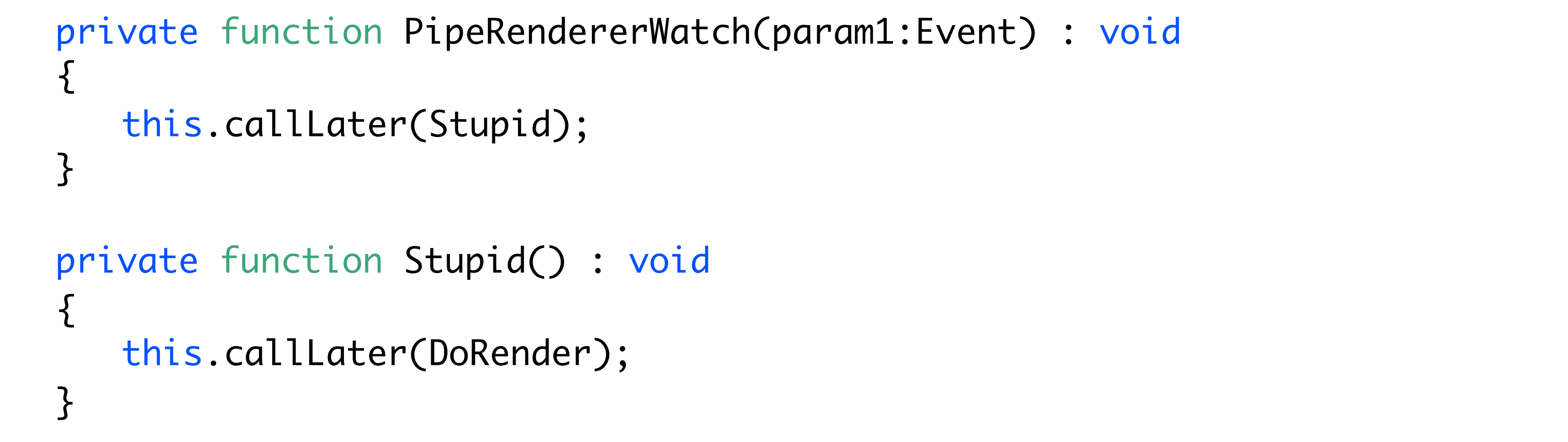


Look how beautiful it is!



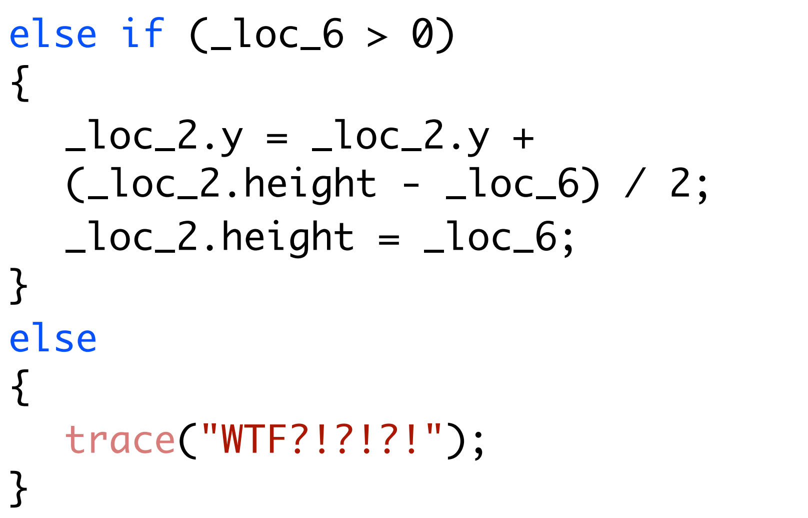
```
▼ src
    ▶ com
    ▼ vg
        ▼ photo
            ▼ editing_pipeline
                AutoLevelsItem.as
                BWItem.as
                ColorMaskItem.as
                ColorMatrixItem.as
                CombinedCurveItem.as
                CropAndRotateItem.as
                CurvesItem.as
                DeNoiseItem.as
                DistortItem.as
                ExposureItem.as
                FillLightItem.as
```

Sometimes you chuckle

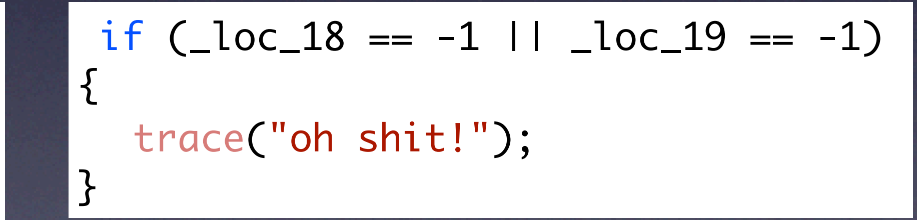


```
private function PipeRendererWatch(param1:Event) : void
{
    this.callLater(Stupid);
}

private function Stupid() : void
{
    this.callLater(DoRender);
}
```



```
else if (_loc_6 > 0)
{
   _loc_2.y = _loc_2.y +
   (_loc_2.height - _loc_6) / 2;
   _loc_2.height = _loc_6;
}
else
{
   trace("WTF?!?!?!");
}
```



```
 if (_loc_18 == -1 || _loc_19 == -1)
{
   trace("oh shit!");
}
```

Obfuscation

- Obfuscation is the process of making your source code intentionally horrible to read
- Obfuscation is a hurdle
- You are changing the bytecode that gets run

- KindiSoft secureSWF
  - $99-$400 (personal, standard, pro)
- irrFuscator
  - 69 euros

Obfuscation problems

- You’re not running the code you wrote!
- I have not found any obfuscators that work 100% with the Flex framework
- Obfuscation is a one way street and difficult to integrate into a build process
- Poor obfuscation can still lead to runnable code, it just might make it harder to read

Good obfuscation is a bitch

Example from Desktop Tower Defense SWF

```
var \x01 = -1921 + \x01\x02();
while (\x01 = eval("\x01") - 825, eval("\x01") == 612)
{
    \x01 = eval("\x01") - 523;
    break;
}
\x01 = eval("\x01") - 15;
if (eval("\x01") == 585)
{
    \x01 = eval("\x01") - 275;
}
```

Encryption

- Approach: load an encrypted SWF at runtime and decrypt
- Options: Runtime Shared Library (RSL), Flex modules
- The elephant in the room: You can’t hide the decryption key
  - (unless you transmit it over a secure server)

Encryption problems

- You have to decrypt at runtime for Flash Player to be able to run your code
    - This probably means custom loading code (modified preloader, ModuleManager, etc)
- Your decryption keys and algorithms are not secure
- Alternative commercial option: NitroLM

Legality/Morality

- You're probably violating someone's terms of use:

  From a real terms of use document:

  You can't do anything that *"attempts to decompile, disassemble, reverse engineer, or derive the source code for any software product provided by us to you in object code format only;"*

(* but actually nothing like this is included in Adobe's terms for Photoshop Express or Buzzword, etc)

Don’t be a dick

- Use your powers for good, not evil
- Learn from other people
- Don’t fuck people over, karma’s a bitch
- If you steal someone’s code they can decompile yours to prove it

Thanks!

http://dougmccune.com/blog

doug@dougmccune.com