The Year In Flash

The year in Flash bugs, exploits and mitigations

Natalie Silvanovich
@natashenka
About me

- Natalie Silvanovich  
  AKA natashenka AKA Flashtasha
- Project Zero member
- Previously did mobile security on Android and BlackBerry
- Flash enthusiast
- Reporter of $\frac{1}{3}$ of Flash vulnerabilities
My goal
My goal

- Bug finding is my top priority
  - Mostly code review
  - Some fuzzing (with Mateusz Jurczyk AKA j00ru)
  - 1 bug per day -> 1 bug per week
  - Flash bugs stay gone

- Analyze external bugs and exploits
My goal

- Occasionally exploit bugs to answer questions
  - Is exploitation possible?
  - Is exploitation reliable?
  - How does X impact exploitability

- Work on mitigations (with James Forshaw and Mark Brand)
This talk

- Attack surface
- The year in Flash
  - New bugs and bug classes
  - 0-days, 1-days and other exploits
  - Mitigations
- The future?
Flash is ...

- **AS2 -- ActionScript 2**
  - Interpreted legacy Flash Scripts with own VM
  - Reduced API set
  - Generally more bugs with lower exploitability
  - Blurry boundaries between VM and APIs
Flash is ...

● AS3 -- ActionScript 3
  ○ Modern VM with JIT and interpreter
    ■ Extendible
    ■ GC Heap / Fixed Heap
    ■ Optimized for Flash
  ○ Open source VM
  ○ Open and closed source APIs
  ○ Bugs are less dense but more exploitable
Flash is ...

- Anticorpus
  - Functionality outside of script
  - MP4 parser, zlib, regex, image decoders, etc
Warning

OBJECTS IN MIRROR ARE LARGER THAN THEY APPEAR
July 2015

● Large update (36 bugs)
● Hacking Team dump
● Mitigations
July 2015

- Hacking Team dump contained two 0-days and two fixed bugs
  - ByteArray/OpaqueBackground -- 0-day UaFs due to valueOf redefinition (CVE-2015-0349 and CVE-2015-05122)
  - ConvolutionFilter issue shown earlier (CVE-2015-3039/CVE-2015-0349)
  - Integer overflow in Function.apply -- reported via Chromium VRP before use (CVE-2015-0387)
  - NULL pointer in BitmapData, not exploitable (CVE-2015-05123)
CVE-2015-3039

- Redefinition issue in ConvolutionFilter (also reported by bilou)
- AS2 allows any method to be redefined in script (monkey-patching)
- Generally native methods accept any type and convert objects with `valueOf`, `toString`, `object constructor`, etc.
var filter = new ConvolutionFilter(...);
var n = { valueOf : ts };  
var a = [];
    a[0] = n;
filter.matrix = a;
function ts(){
    filter.matrix = [1];
}
July 2015

- `valueOf/toString` bugs receive increased attention
  - Many similar bugs reported in next few months
  - Adobe starts efforts to pre-emptively fix similar bugs

- 33 bugs in regular update

- Vector mitigations implemented
Vector Mitigation

"I don't afraid Adobe analysts at all" -- Vitaly Toropov

- Adds checksums to Vectors that are checked before doing sensitive functions
- Some Vectors are also on their own heap page
- Reduced the reusability of exploit code
- Generally increases the quality of bug needed for an exploit
- Substitution of ByteArray or BitmapData is possible, but not as good
CVE-2015-3130

- Redefinition issue involving valueOf
var s = 1;
var rec_array:Array = new Array();
rec_array.push({name: "john", city: "omaha"});
rec_array.push({name: "bob", city: "omaha"});
rec_array.length = {valueOf : gl};

rec_array.sortOn(['name', 'city']);

function gl(){
    if(s< 3){
        s++;
        return 100000;
    }else{
        return 2;
    }
}

if (array->getLength() == 0){ return; }
int length = array->getLength();
char** s = new char*[array->getLength()];
memcpy(s, array->items, length);
August 2016

- Many more bugs similar to HT bugs
- MC UaFs pour in
CVE-2015-5550 (MovieClip UaFs)

- Very common AS2 bug, 100+ reported this year
  - Small variety of freed object
- Also works with TextFields
- Root cause is that display fields are freed outside of garbage collection
  - Always, for real, even if there are references (in AS2)
CVE-2015-5550 (MovieClip UaFs)

- Happens when function parameters are converted after local variables are initialized, but before they are used.
- Fixed by enforcing convert -> initialize -> use order.
```javascript
var clip1 = this.createEmptyMovieClip("clip1", 1);
var clip2 = this.createEmptyMovieClip("clip2", 2);
var n = {toString: func};
clip1.swapDepths(n);

function func(){
    clip1.removeMovieClip();
    return "clip2";
}
```

```
SO *s = GetObject();
MC *m = native_data[10];
```
September/October 2015

- 23 bugs in September updates and 20 in October
  - Mostly UaFs and other redefinition bugs
- 0-day immediately after October update (reported by TrendMicro, NATO targets)
CVE-2015-7645

- Reported two weeks before it was found in the wild
- Type confusion in serializations, due to weird AVM behaviour
- Two other variants also reported and fixed in emergency patch
- None of these bugs compile
CVE-2015-7645

From the AVM:

```c
// In theory we should reject duplicate slots here;
// in practice we don't, as it causes problems with some existing content
//if (basetb->findBinding(name, ns) != BIND_NONE)
//  toplevel->throwVerifyError(kIllegalOverrideError, toplevel->core()->
toErrorString(qn), toplevel->core()->toErrorString(this));
```

tl;dr a method can be overridden by a var

Most natives don’t make assumptions, but some do. Especially interfaces.
class superclass{
    ...
    public function writeExternal(){
        return 1;
    }
}

class subclass extends superclass{
    public var writeExternal:uint = 7;
    ...
}

CVE-2015-7645

From the AVM:

Multiname mn(core-&gt;getPublicNamespace(t-&gt;pool),
    core-&gt;internConstantStringLatin1(kWriteExternal));
m_functionBinding = toplevel-&gt;getBinding(t, &mn);

and later:

MethodEnv* method =
    obj-&gt;vtable-&gt;methods[AvmCore::bindingToMethodId(info-&gt;get_functionBinding()] ;
method-&gt;coerceEnter(argc, argv);
How was this bug exploited?

- Traits property array is variable-sized
- Corrupted ByteArray to get R/W access to entire memory space
November and December 2015

- Huge Dec update, 79 bugs, mostly MC UaF
  - Structural changes to AS2 to make broad fixes
- New mitigations
  - Checksumming on byteArray
  - Isolated Heap
  - NOP slide mitigations
- Exploit kit 1-day and 0-day
CVE-2015-8446

- 1-day in Angler
- Similar to CVE-2015-5560
- Integer overflow in ID3 allocation
  - Controllable size
  - Controllable overwrite
- Exploited using BitmapData
CVE-2015-8651

- Integer overflow leading to heap overflow in JIT (reported by Huawei)
CVE-2015-8651

● SWF contained two exploits
  ○ Typical vector exploit
  ○ Post Isolated Heap exploit including such elements as
    ■ Long if statements nested almost 100 times
    ■ Using both a media file and an image to fill heap slots at different points in the exploit
    ■ Triggering the bug ~600 times
    ■ Final results was memory space access via ByteArray
Timeline

- 7/8 APSB15-16
  - 36 bugs
- 8/11 APSB15-19
  - 35 bugs
- 7/18 APSA15-03
  - Emergency
  - Vector Mitigation
- 8/11 APSB15-23
  - 23 bugs
- 10/13 APSB15-25
  - 20 bugs
- 10/16 APSB15-27
  - Emergency
  - 3 bugs
- 11/10 APSB15-28
  - 17 bugs
- 12/8 APSB15-32
  - 78 bugs
- 9/21 APSB15-23
  - 23 bugs
- 10/16 APSB15-27
  - Emergency
  - 3 bugs
- 12/8 APSB15-32
  - 78 bugs
- 10/14 APSB15-7645
  - 0-day
- 12/23 APSB15-8651
  - 0-day
- 7/10
  - 0-day
  - CVE-2015-5122
  - CVE-2015-5123
- 12/28 APSB16-01
  - Emergency
  - 19 bugs
- 2/9 APSB16-04
  - 22 bugs

Google
February 2016

- 22 bugs fixed
  - UaF straglers
  - Fuzz bugs, mostly media
  - Exception-related bugs
CVE-2016-0985

- Unusual type confusion in AS3
- Occurs due to catching an exception
  - Should have been fatal error
try{
    var t = new TextField();
} catch(e:Error){
    var t2 = new TextField();
}
CVE-2016-0984

- Read-only UaF in Sound class
- Exception-related
var s = new Sound();
var b = new ByteArray(); // 1000 bytes
s.loadPCMFromByteArray(b, 100, "float", false, 2.0);
var c = new ByteArray(); // size 2
try{
    s.loadPCMFromByteArray(c, 1, "float", false, 2.0);
}catch(e:Error){}
s.extract(b, 1, 0);
March 2016

- 22 bugs fixed
  - More fuzz bugs
    - Exploited one
  - UaFs
- pwn2own
  - 4 MC UaFs, one parsing bug in JPG
- 0-day
  - Overflow in Bitmap in platform-specific code
CVE-2016-0998

- Exploitable fuzz bug
- Caused by UaF check
- Required ASLR bypass
- Got around IsoHeap with JIT pages
var o = {};
o.unwatch();

void* args = alloca(numArgs);
...
convertToStdString(args[0])
April 2016

- 24 bugs fixed
  - Fuzzed images
  - The UaFs continue
- 0-day
  - Reported by TrendMicro in Magnitude EK
  - Type confusion in FileReference
  - Did not work on current mitigation set (0 on free)
CVE-2016-1019

- 0-day vulnerability
- Reported some variants
- Type confusion in AS2
var o = { toString : f };  
var t = new TextField(o);  

function f(){  
    var fr = FileReference  
    fr.call(this);  
}
June 2016

- 36 bugs fixed
  - MC UaFs (echo from pwn2own)
- 0-day
  - Reported by Kaspersky
  - OOB memory access in open source parser
void AbcParser::parseExecPolicyAttributes(const uint8_t* metadata, MethodInfo* m)
{
    ...
    for (int q = 0; q < values_count; ++q)
    {
        Stringp key = pool->getString(readU30(p));
        Stringp val = pool->getString(readU30(p));
Stringp PoolObject::getString(int32_t index) const
{
    ConstantStringData* dataP = _abcStrings->data + index;
    if (dataP->abcPtr >= _abcStringStart && dataP->abcPtr < _abcStringEnd)
    {
        uint32_t len = AvmCore::readU32(dataP->abcPtr);
        Stringp s = core->internStringUTF8((const char*) dataP->abcPtr, len, true, false);
        s->Stick();
        dataP->abcPtr = NULL;
        WBRC(core->gc, _abcStrings, &dataP->str, s);
    }
    return dataP->str;
}
Conclusions

- Finding bugs in Flash is generally getting harder
  - 1 bug per day versus 1 per week
- Certain bug classes are drying up, but others are taking their places
- Flash mitigations are making it more difficult to exploit bugs, especially with low-quality bugs
The Future (What’s left?)

- MC UaFs (and AS2) probably still exist, but getting hard to exploit
  - Eventually similar bugs will have marginal utility
  - Display UaFs in AS3?
- Redefinition bugs are no longer ‘deep’
- More AS3 bugs?
The Future (What’s left?)

- More anticorpus bugs / use of anti-corpus?
  - Media (MP4, FLV)
- Open source AVM?
- Platform-specific code
- Flash deprecation
  - Browsers?
Thank You

- Adobe
Questions?

http://googleprojectzero.blogspot.com/
@natashenka
natalie@natashenka.ca