Banjo

An Android Disassembler for Binary Ninja

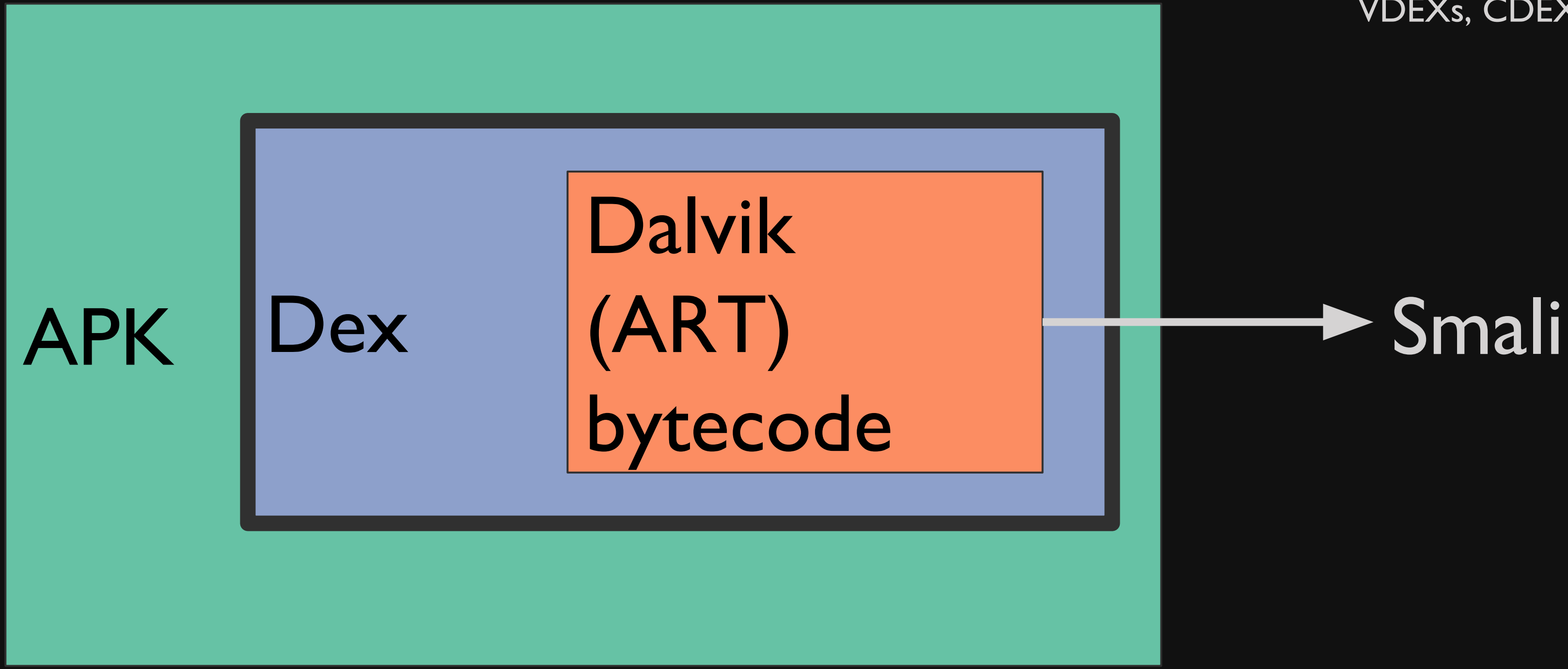
Austin Ralls

austin.ralls@carvesystems.com

github.com/carvesystems/banjo

github.com/carvesystems/presentations

We're ignoring ODEXs, VDEXs, CDEXs, etc.



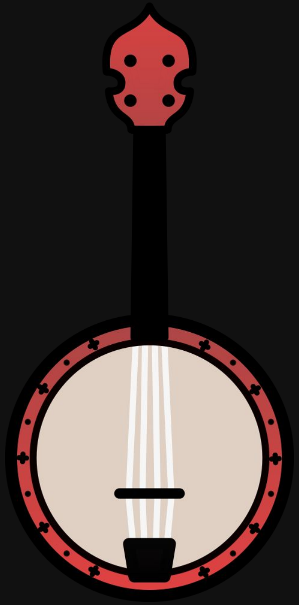
JAR .class JVM bytecode

(not intended to be comprehensive)	Text disassembly	Interactive disassembly and graph view	Intermediate representation (lifting)	Decompilation
Compiled binaries	Objdump Capstone Udis86 …	IDA Pro Binary Ninja Hopper …	LLIL (Binary Ninja) LLVM (Mcsema) Falcon (Goblin) …	Hex-Rays Ghidra RetDec
Android applications	baksmali	Graphviz PNG generators Radare2 JEB?	Jimple (Soot)	Jadx JD-GUI Dex2jar -> java decompiler JEB

Goals

1. Be better for this specific task than r2
2. Open source a complex architecture plugin for Binary Ninja
3. Have a Python disassembler library

This is a research project



Disassembler library

Binary Ninja integration

Baksmali compatible-ish CLI

Demo

x86		Smali	
90	nop	0000	nop
31 c0	xor eax,eax	b7 00	xor-int/2addr v0,v0
e8 eb ff ff ff	call <offset>	70 10 71 00 01 00	(next slide)

```
70 10 71 00 01 00 invoke-direct {v1}, Ljava/lang/Object;-><init>()V

70    invoke-direct

1_    one argument

71 00 method id 0x0071

_1    argument is register v1

__    unused arguments
```

How to find method 0x71

1. Go to file header, find method_ids_off
2. Jump to 0x71*8 bytes into this section to find method_id
3. Method_id has class, proto, and name indexes
4. Go back to file header, find type_ids_off
5. Jump to class_id_idx*4 bytes into this section to find string_idx
6. Go back to file header, find string_ids_off
7. Jump to string_idx*4 bytes into this section to find string_data_off
8. Find string_data section in map_list
9. Jump to string_data_off bytes in the string_data section to find the string_data_item
10. Parse the string_data_item to find the data field
11. The data field is a MUTF-8-encoded string representing the class name ← (⅓ done with text disassembly!)
12. Go back to file header, find proto_ids_off
13. Jump to proto_id_idx*12 bytes into this section to find a proto_id_item
14. Parse the proto_id_item
15. …

Binary Ninja concepts

Binaryview (ELF, Dex)

Architecture (x86, Dalvik/Smali)

LLIL (lifting, intermediate representation)

Hacky workarounds

How do you read bytes at a specified offset in an Architecture?

- Guess you need to precompute lookup tables for those.

Hacky workarounds

How do you read bytes at a specified offset in an Architecture?

Guess you need to precompute lookup tables for those.

How do you access BinaryView data from an Architecture?

You can't... Use a side-channel (write to disk).

Hacky workarounds

How do you read bytes at a specified offset in an Architecture?

Guess you need to precompute lookup tables for those.

How do you access BinaryView data from an Architecture?

You can't... Use a side-channel (write to disk).

How do you cache file-specific data in an Architecture instance?

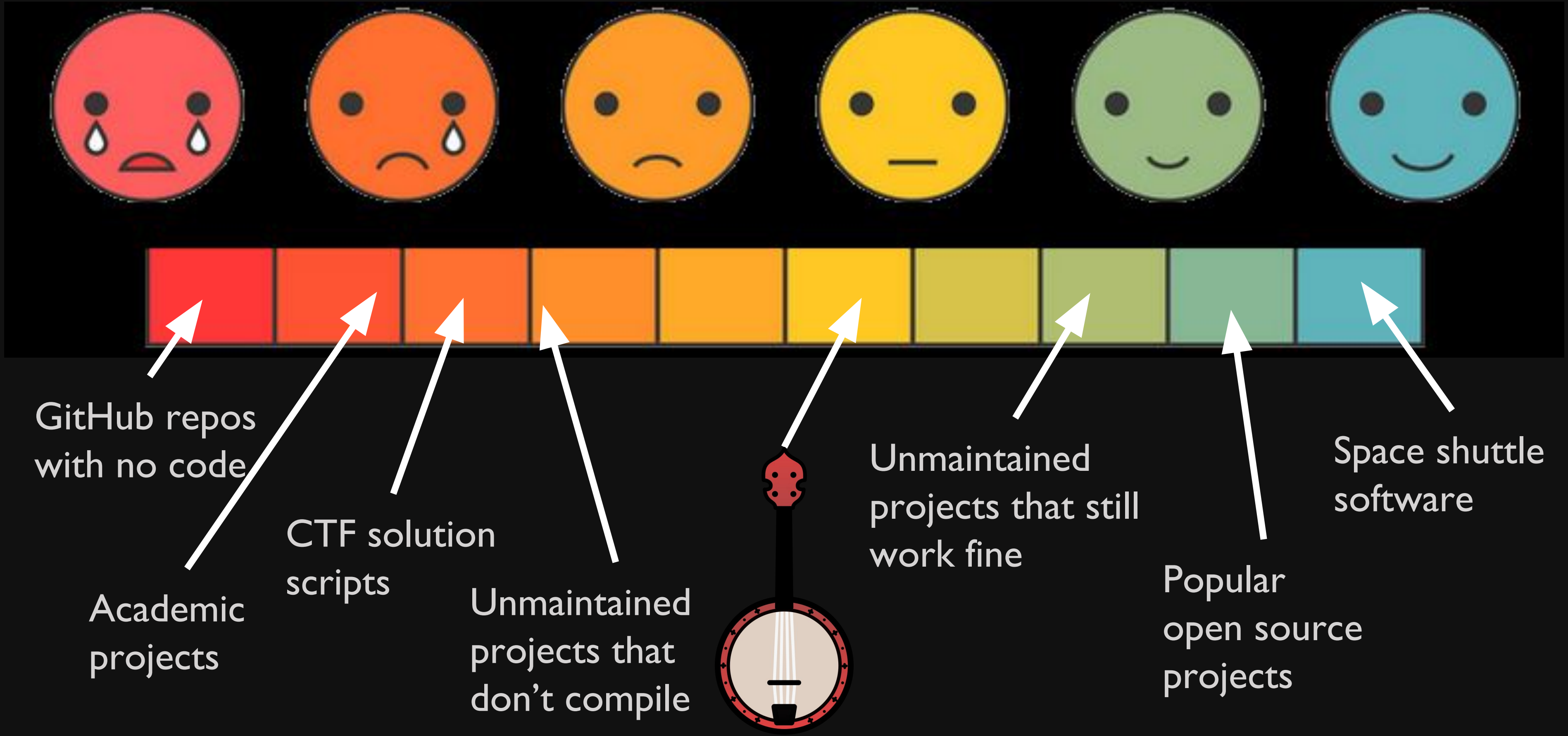
(Have not figured this one out yet, but most things work without it)

Things that were not obvious to me

What functions of the Architecture class do you actually need to implement?

How do you actually add a reference to another address?

How do you actually run background threads?


GitHub repos with no code
Academic projects
CTF solution scripts
Unmaintained projects that don’t compile
Unmaintained projects that still work fine
Popular open source projects
Space shuttle software

Status: mostly works, with rough edges

Still in development

Shoutouts

austin.ralls@carvesystems.com

github.com/carvesystems/{banjo,presentations}