



# The complexity of reversing Flutter applications

Axelle Apvrille, Fortinet

Nullcon, March 2024



# Who am I?



**Axelle Apvrille**

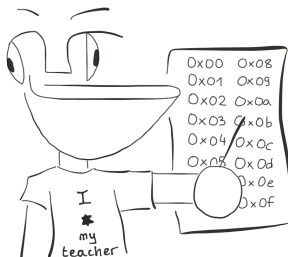
Principal Security Researcher at **Fortinet**, @cryptax

Lead organizer of **Ph0wn CTF**

I analyze **Android malware** and **IoT** malware



# Goal of this talk



Understand how to reverse Flutter applications  
with a special focus on Android malware

sub-goal: solve GreHack CTF 2023 Dart challenge



**Dart** is an **object-oriented** programming language with a **C-style** syntax

```
class Hello {  
  void sayHello() {  
    print("Hello Nullcon!");  
  }  
}  
  
void main() {  
  var hello = Hello();  
  hello.sayHello();  
}
```



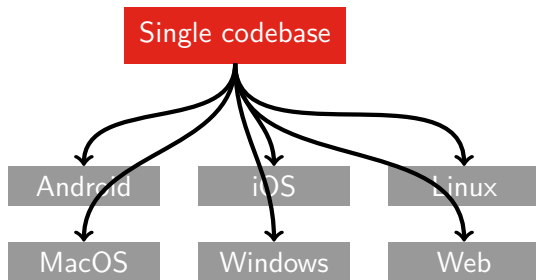
## Dart: 4 output formats

Output format	Size	Time
Self contained	5,919,728	0,006s
AOT snapshot	873,440 <b>14%</b>	0,012s <b>2x</b>
JIT snapshot	4,924,048 <b>83%</b>	0,333s <b>55x</b>
Kernel snapshot	1968 <b>0.03%</b>	0,411s <b>68x</b>

```
dart compile exe|aot-snapshot|jit-snapshot|kernel file.dart
```



# Dart can be natively compiled for multiple platforms



## Dart

Native machine code

Android and iOS: apps bundled with a Dart VM runtime

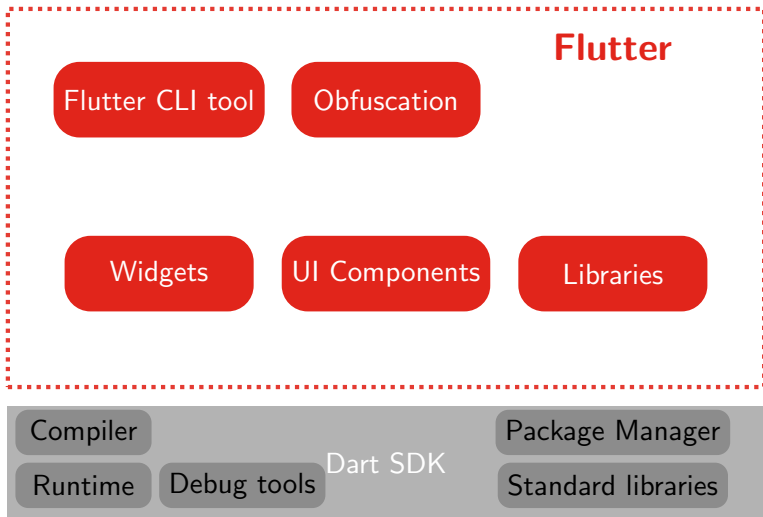
## Java

Byte code

Android and iOS: JVM for mobile exists but primarily for dev and testing.



# Flutter uses the Dart language and SDK



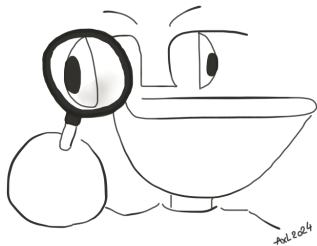
# Flutter output types

Output type	Speed	Comments
Kernel snapshot	Slow	Flutter <b>Debug</b> builds. Portable. Easy to reverse
<del>JIT snapshot</del>		<i>Not used in Flutter</i>
AOT snapshot	Fast	Flutter <b>Release</b> builds. <b>Compiled</b> <b>Natively.</b> Difficult to reverse
<del>Self-contained</del>		<i>Not used in Flutter</i>





# Focus



- 1 Understand how to reverse **Flutter** applications for Android, especially malware. **Release** applications → **Dart AOT snapshot**.
- 2 Solve **GreHack CTF 2023 Dart challenge** → It's a **Dart AOT snapshot**

Let's focus on **Dart AOT snapshots**



# Disassemblers do not support AOT snapshots

libapp.so — Binary Ninja Free 4.0.4911 free

File Edit View Analysis Debugger Plugins Window Help

libapp.so X +

Symbols Q 458130

Name	Address	Section
sub_458130	0x00458130	.text

sub\_458130

```
int64_t sub_458130(void* arg1 @ r14, void* arg2 @ r15, int64_t arg3)

sub_458130:
0 @ 00458131 int64_t __saved_rbp
1 @ 00458131 int64_t* rbp = &__saved_rbp
2 @ 0045813f int64_t var_10 = *(arg1 + 0xc8)
3 @ 00458155 int64_t var_20
4 @ 00458155 if (&var_20 u<= *(arg1 + 0x38))

5 @ 004581d2 (*(arg1 + 0x270))() {"p_instructions no-asserts x64-sy..."}

6 @ 0045815b *(arg2 + 0xc897)
7 @ 00458162 int64_t rsi
8 @ 00458162 int64_t rdi
9 @ 00458162 rsi, rdi = sub_20bca4(rbp, arg2)
10 @ 0045816e int64_t rax_3 = (*(arg1 + 0x80) + 0x1728)
11 @ 00458179 if (rax_3.d == *(arg2 + 0x27))

12 @ 00458186 rax_3, rdi = sub_593b38(rdi, rsi, *(arg2 + 0xf04f), arg1, arg2)

13 @ 0045818b var_20 = rax_3
14 @ 0045818f int64_t var_28 = arg3
15 @ 00458192 void* rax_4 = sub_457f32(arg1, rdi)
16 @ 00458198 int64_t var_28_1 = var_20
17 @ 004581a2 int64_t var_30 = *(arg2 + 0xf9d7)
18 @ 004581a2 ...
```

Cross References

Filter (3)

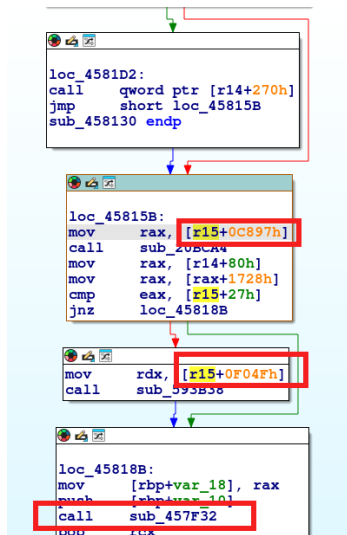
- Code References (2)
- sub\_458130 (2)
- 00458155 if (&var\_20 u<= \*
- 004581d9 jmp 0x45815b
- Variable References (1)
- void\* arg2 (1)
- 0045815b \*(arg2 + 0xc897)

linux-x86\_64 0x45815b-0x458162 (0x7 bytes)

No function name, wrong arguments for the function



# Disassemblers do not support AOT snapshots



No strings, no literals, no function names



# Disassemblers do not support AOT snapshots

```
0x00458131 4889e5 mov rbp, rsp
0x00458134 4883ec18 sub rsp, 0x18
0x00458138 498b86c80000. mov rax, qword [r14 + 0xc8]
0x0045813f 488945f8 mov qword [var_8h], rax
0x00458143 33c0 xor eax, eax
0x00458145 4863c0 movsxd rax, eax
0x00458148 488b4c8510 mov rcx, qword [rbp + rax*4 + 0]
0x0045814d 48894df0 mov qword [var_10h], rcx
0x00458151 493b6638 cmp rsp, qword [r14 + 0x38]
< 0x00458155 0f8677000000 jbe 0x4581d2
; CODE XREF from fcn.00458130 @ 0x4581d9(x)
0x0045815b 498b8797c800. mov rax, qword [r15 + 0xc897]
0x00458162 e83d3bdbff call fcn.0020bca4
0x00458167 498b86800000. mov rax, qword [r14 + 0x80]
0x0045816e 488b80281700. mov rax, qword [rax + 0x1728]
0x00458175 413b4727 cmp eax, dword [r15 + 0x27]
< 0x00458179 0f850c000000 jne 0x45818b
0x0045817f 498b974ff000. mov rdx, qword [r15 + 0xf04f]
0x00458186 e8adb91300 call fcn.00593b38
; CODE XREF from fcn.00458130 @ 0x458179(x)
> 0x0045818b 488945e8 mov qword [var_18h], rax
```

No strings, no literals, no function names



# Disassemblers do not support AOT snapshots

CodeBrowser: Ghidra disassemblies/libapp.so

File Edit Analysis Graph Navigation Search Select Tools Window Help

Program Trees

- libapp.so
  - dynamic
  - bss
  - .text
  - .hash
  - .dynsym
  - .dynstr
  - .eh\_frame
  - .rodata
  - .note.gnu.build-id
  - segment\_1.1
  - .shstrtab
  - \_elfSectionHeaders

Symbol Tree

- Imports
- Exports
  - \_kDartIsolateSnapshotData
  - \_kDartIsolateSnapshotInstructions
  - \_kDartSnapshotBuildId
  - \_kDartVmSnapshotData
  - \_kDartVmSnapshotInstructions
- Functions
- Labels
- Classes
- Namespaces

Filter:

Data Type Manager

- Data Types
  - BuiltinTypes
  - libapp.so
  - generic\_clib\_64

Listing: libapp.so

```
// .text
// SHT_PROGBITS [0x200000 - 0x59ace0f]
// raw:003000000-ran:0069ace0f
//
_kDartVmSnapshotInstructions
XREF[4]: Entry Point(*
002fe72c(*)
_elfSectionHe

00300000 f0 51 00    undefined...
00 00 00
00 00 10 ...
00300000 f0                undefined1F0h                [0]

00300001 51                undefined151h                [1]
00300002 00                undefined100h                [2]
00300003 00                undefined100h                [3]
00300004 00                undefined100h                [4]
00300005 00                undefined100h                [5]
00300006 00                undefined100h                [6]
00300007 00                undefined100h                [7]
00300008 10                undefined110h                [8]
00300009 00                undefined100h                [9]
0030000a 00                undefined100h                [10]
0030000b 00                undefined100h                [11]
0030000c 00                undefined100h                [12]
0030000d 00                undefined100h                [13]
0030000e 00                undefined100h                [14]
0030000f 00                undefined100h                [15]
00300010 30                undefined130h                [16]
00300011 00                undefined100h                [17]
00300012 13                undefined113h                [18]
00300013 00                undefined100h                [19]
00300014 00                undefined100h                [20]
00300015 00                undefined100h                [21]
00300016 00                undefined100h                [22]
00300017 00                undefined100h                [23]
00300018 b0                undefined180h                [24]
00300019 51                undefined151h                [25]
0030001a 00                undefined100h                [26]
0030001b 00                undefined100h                [27]
0030001c 00                undefined100h                [28]
0030001d 00                undefined100h                [29]
0030001e 00                undefined100h                [30]
0030001f 00                undefined100h                [31]
```

Bad entry point, Completely lost



# Dart assembly defines its own registers!



	x86_64	Aarch32	Aarch64
Stack Pointer (SP)	R4	R14	<b>X15</b> (custom)
<b>Object Pool (PP)</b>	<b>R15</b>	<b>R5</b>	<b>X27</b>
<b>VM Thread (THR)</b>	<b>R14</b>	<b>R10</b>	<b>X26</b>



# Documentation is ... the code

<https://github.com/dart-lang/sdk/>

```
enum Register {  
  ...  
  R5 = 5,    // PP  
  R6 = 6,    // CODE_REG  
  R7 = 7,    // FP on iOS, DISPATCH_TABLE_REG on non-iOS (AOT only)  
  R8 = 8,  
  R9 = 9,  
  R10 = 10,   // THR  
  R11 = 11,   // FP on non-iOS, DISPATCH_TABLE_REG on iOS (AOT only)  
  R12 = 12,   // IP aka TMP  
  R13 = 13,   // SP  
  R14 = 14,   // LR  
  R15 = 15,   // PC
```

[https://github.com/dart-lang/sdk/blob/main/runtime/vm/constants\\_arm.h](https://github.com/dart-lang/sdk/blob/main/runtime/vm/constants_arm.h)



## Example of Function Prologue for Aarch64

```
; push frame pointer and link register on the stack
STP      X29, X30, [X15, #FFFFFFF0h]!
; update frame pointer
MOV      X29, X15
; allocate 16 bytes on the stack
SUB      X15, X15, #10h
; stack overflow check
LDR      X16, [X26, #38h]
CMP      X15, X16
B.LS     loc_3D75DC
```


- **X15**: custom stack pointer for AAarch64
- **X26**: holds a pointer to the current thread





# Dart Object Pool

Index	Value
0	True
1	Address to user object
2	9223372036854775807
3	In type cast
...	...
???	Password:
???	The door is locked
...	...
1456	Out of Memory
1457	Address to user object
...	...



# Dart Object Pool

Index	Value
0	True
1	Address to user object
2	9223372036854775807
3	In type cast
...	...
???	Password:
???	The door is locked
...	...
1456	Out of Memory
1457	Address to user object
...	...

Diagram illustrating the Dart Object Pool structure and address calculation:

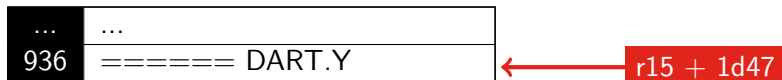
- The pool is represented as a table with **Index** and **Value** columns.
- Indices 0 through 1457 are shown, with values ranging from `True` to `Out of Memory` and `Address to user object`.
- Red arrows indicate the base address `r15` for indices 0 and 1457.
- Offset calculations are shown for indices 7 and 1457:
  - For index 7:  $r15 + (\text{index} * 8) + 0-7$
  - For index 1457:  $r15 + (1457 * 8) + 0-7$
- Double-headed arrows labeled "offset" indicate the distance from the base address to the target address.

$$\text{offset} = (\text{index} * 8) + (0-7)$$
$$\text{index} = \text{offset} // 8$$



# Examples of access to the Object Pool

```
; x86-64  
mov     r11, qword ptr ds:[r15+1D47h]
```



## Big indexes are computed - example on Aarch64

```
; loads object pool + 0xF038  
ADD     X17, X27, #Fh, LSL #12  
LDR     X17, [X17, #38h]
```

Loads object pool (X27) + 0xF000 (0xF LSL 12) + 0x38 = 0x0F038



# Dart's representation of integers: SMI/MINT

```
mov qword [rbp - 0x18], rax
mov r11d, 0x75e ; decimal value = 943
mov qword [rax + 0x17], r11
mov r11d, 0x760 ; 944
mov qword [rax + 0x1f], r11
mov r11d, 0x422 ; 529
```

Dart has 2 different representations of integers:

- 1 **Small Integers (SMI)**. They fit on 31 bits. **Least significant bit set to 0.**
- 2 **Medium Integers (Mint)**. Bigger.

Most significant bit



Small integer value

Least significant bit



SMI indicator

<https://cryptax.medium.com/>



[reversing-flutter-apps-darts-small-integers-b922d7fae7d9](https://medium.com/@apvrille/reversing-flutter-apps-darts-small-integers-b922d7fae7d9)

# DART.Y CTF challenge

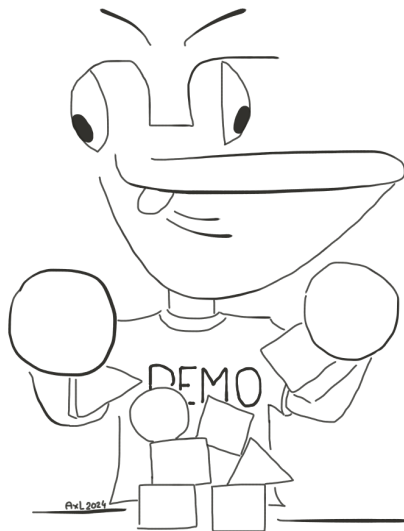
*Pico bought a connected fridge at DART.Y. It locks up his favorite caviar from predators, except Pico is hungry and can't remember the password to open his fridge...*

```
===== DART.Y - Your Secure & Smart Fridge =====  
Password:
```

- Dart AOT snapshot, not stripped
- Flag format is GH23{.....}
- The challenge was renamed in GreHack CTF 2023



# Demo



## When you enter the wrong password

Content	Index
-----	-----
deli	943
ph0wn{	944
{	529
pico	945
le	946
croco	947
GH23{	948
caviar	949

champagne	950
drink	951
chocolate	952
yacht	953
-	555
@	231
++	954
+	535
...	
The door is locked	
=====	

It's an Object Pool!



# Disassembling the AOT snapshot

Fortunately, it's *not* stripped

```
[0x00058000]> afl~main
0x000afb90    6    201 main
0x000b297c    3    33  sym.main_1
```

We don't have the entire Object Pool but we can guess some

```
mov r11, qword [r15 + 0x1d47] <-- guess: ===== DART.Y ...
mov qword [rsp], r11
call sym.printToConsole
call sym.stdout
mov qword [var_8h], rax
mov r11, qword [r15 + 0x1d4f] <-- guess: Password:
mov qword [rsp], r11
call sym._StdSink.write
call sym.stdin                <-- wait for user input
```





# Create Flag

```
mov qword [rsp], rax
call sym.Stdin.readLineSync
mov qword [var_bp_8h], rax
call sym.createFlag      <-- 0oooooooooh! createFlag
```

## In createFlag

```
[0x000afb90]> s sym.createFlag
[0x000afc5c]> pif
...
mov r11, qword [r15 + 0x1d6f] <-- Guess: Content | Index
mov qword [rsp], r11
call sym.printToConsole
mov r11, qword [r15 + 0x1d77] <-- Guess: ----- | -----
mov qword [rsp], r11
call sym.printToConsole
```



# Many objects are loaded from the Object Pool

```
call sym.stub__iso_stub_AllocateArrayStub
mov qword [var_8h], rax
mov r11, qword [r15 + 0x1d7f]
mov qword [rax + 0x17], r11
mov r11, qword [r15 + 0x1d87]
mov qword [rax + 0x1f], r11
mov r11, qword [r15 + 0x108f]
mov qword [rax + 0x27], r11
mov r11, qword [r15 + 0x1d8f]
mov qword [rax + 0x2f], r11
mov r11, qword [r15 + 0x1d97]
mov qword [rax + 0x37], r11
mov r11, qword [r15 + 0x1d9f]
mov qword [rax + 0x3f], r11
mov r11, qword [r15 + 0x1da7]
mov qword [rax + 0x47], r11
mov r11, qword [r15 + 0x1daf]
mov qword [rax + 0x4f], r11
mov r11, qword [r15 + 0x1db7]
mov qword [rax + 0x57], r11
...
```

Index	Value
0x1d7f // 8	deli
0x1d87 // 8	ph0wn{
0x108f // 8	{

$0x1d7f // 8 = 943$

$0x1d87 // 8 = 944$

Content	Index
-----	-----
deli	943
ph0wn{	944



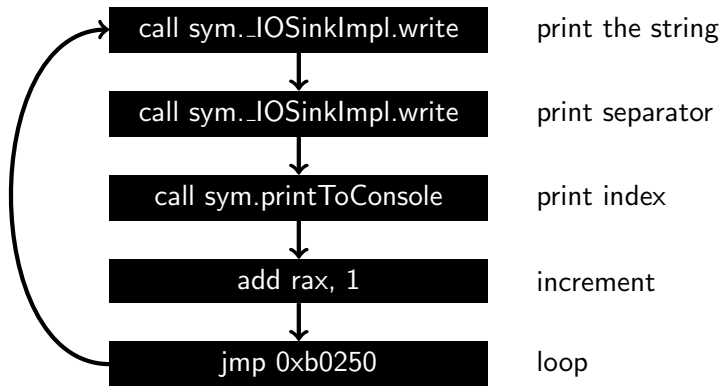
## Those are the indexes of the supplied Object Pool

```
mov r10d, 0x70
call sym.stub__iso_stub_AllocateArrayStub
mov qword [rbp - 0x18], rax
mov r11d, 0x75e
mov qword [rax + 0x17], r11
mov r11d, 0x760
mov qword [rax + 0x1f], r11
mov r11d, 0x422
```

Index	SMI	Value
0x1d7f // 8 = 943 <sub>10</sub>	943 <sub>10</sub> << 1 = 0x75e	deli
0x1d87 // 8 = 944 <sub>10</sub>	944 <sub>10</sub> << 1 = 0x760	ph0wn{
0x108f // 8 = 529 <sub>10</sub>	529 <sub>10</sub> << 1 = 0x422	{



## Loop to print the Object Pool



## Final part of createFlag

```
mov r11, qword [r15 + 0x115f]
mov qword [rsp], r11
call fcn.0007880c
mov qword [var_sp_8h], rax
mov r11, qword [r15 + 0x1e5f]
mov qword [rsp], r11
call fcn.0007880c
mov qword [var_sp_8h], rax
mov r11, qword [r15 + 0x1e77]
...
mov rsp, rbp
pop rbp
ret
```

- Access to many objects of the Object Pool
- fcn.0007880c is string concatenation
- The result is returned by createFlag, so it's the flag!



## createFlag summary

Print table loop

Concatenate parts of flag

Return

Content	Index
-----	-----
deli	943
ph0wn{	944
{	529
pico	945
le	946
croco	947
GH23{	948
caviar	949



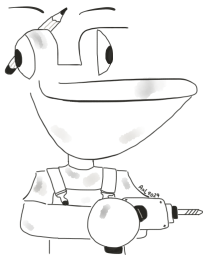
# Recover the flag

0x1da7	GH23{
0x115f	-
0x1e5f	s
0x1e77	lurp
0x115f	-
0x1e9f	it
0x115f	-
0x1e5f	s
0x115f	-
0x1d7f	deli
0x1ea7	cious



GH23{ \_slurp\_it\_s\_delicious\_with\_some\_lobster! }

## Next Goal



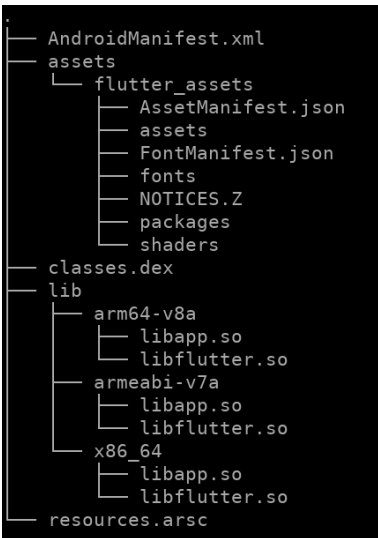
- 1 Understand how to reverse **Flutter** applications for Android, especially malware. **Release** applications → **Dart AOT snapshot**.
- 2 Solve ~~GreHack CTF 2023 Dart challenge~~ → It's a ~~Dart AOT snapshot~~. **DONE**

Let's focus on Flutter applications for Android





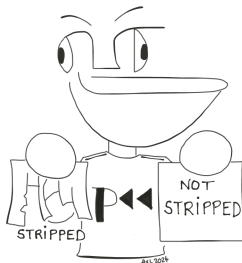
# Flutter application on Android: locating the payload



- `classes.dex`: contains Java code, and Dalvik to Flutter glue
- `./lib/xxx/libflutter.so`: Flutter implementation
- `./lib/xxx/libapp.so`: payload!



# Reversing Flutter applications: what's different?



## Releases are stripped

```
dart compile aot-snapshot -S ./debuginfo file.dart
```

```
[0x00170000]> afl
0x003f7b80    9    212 fcn.003f7b80 <-- no function names
0x003db6c0   28    464 fcn.003db6c0
0x0036f024   14    232 fcn.0036f024
0x00332678   14    232 fcn.00332678
```

## Example: Android/SpyLoan (2023)

- Attract victims for easy loans
- Complete a loan application, enter personal information.
- Malware blackmails victims to repay more quickly.



# Example: Android/SpyLoan (2023)

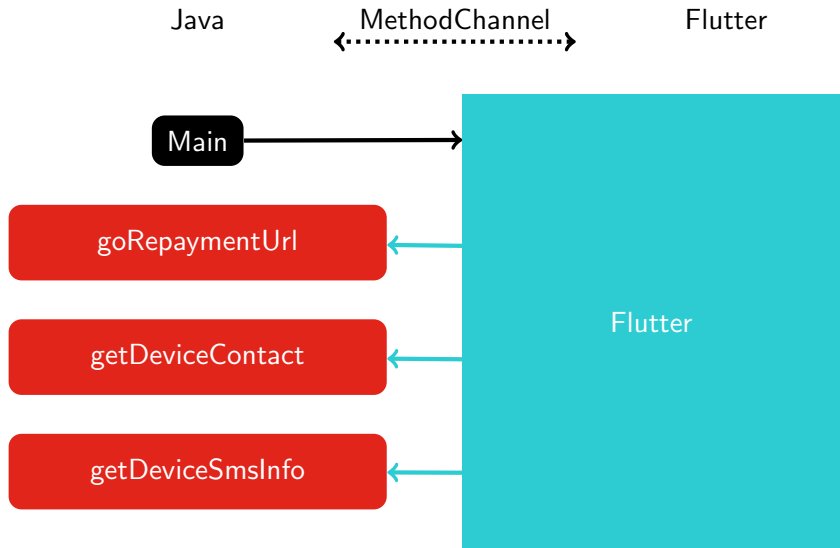
- Attract victims for easy loans
- Complete a loan application, enter personal information.
- Malware blackmails victims to repay more quickly.



Source:

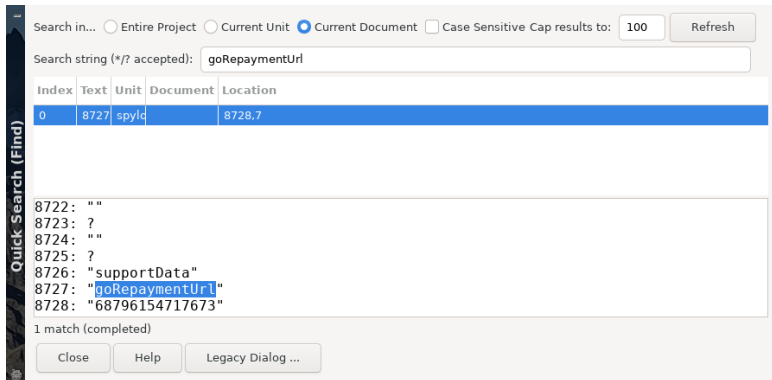
<https://www.dailyradar.in/aa-kredit-loan-app-review/>

# Android/SpyLoan: implementation



# Where is goRepaymentUrl called?

- 1 ‘goRepaymentUrl’ is provided to MethodChannel
- 2 Search ‘goRepaymentUrl’ in the Object Pool
- 3 Index is **8727=0x110b8**



Search in... ☐ Entire Project ☐ Current Unit ☒ Current Document ☐ Case Sensitive Cap results to: 100 Refresh

Search string (\*/? accepted): goRepaymentUrl

Index	Text	Unit	Document	Location
0	8727	spyd		8728,7

Quick Search (Find)

8722: ""  
8723: ?  
8724: ""  
8725: ?  
8726: "supportData"  
8727: "goRepaymentUrl"  
8728: "68796154717673"

1 match (completed)

Close Help Legacy Dialog ...



# Where is goRepaymentUrl called?

- 1 ‘goRepaymentUrl’ is provided to MethodChannel
- 2 Search ‘goRepaymentUrl’ in the Object Pool
- 3 Index is **8727=0x110b8**
- 4 Search assembly loading the index:  
ADD REGISTER, X27, #11h, LSL #12  
LDR REGISTER, [REGISTER, #B8h]

Search in... ☐ Entire Project ☐ Current Unit ☒ Current Document ☐ Case Sensitive Cap results to: 100 Refresh

Search string (\*/\* accepted): [X\*, #b8h]

Index	Text	Unit	Document	Location
61	LOAL spylc			sub_39824C4+74h
62	LOAL spylc			sub_39FDC4+200h
63	LOAL spylc			sub_3A6B3C+1DCh
64	LOAL spylc			__CompactLinkedCustomHashMap6_HashFieldBase6MapMixin5_HashBase6_CustomEqualsAndHashCode6_LinkedHashMapMixin@3220832___regenerat
65	LOAL spylc			__CompactLinkedIdentityHashMap6_HashFieldBase6MapMixin5_HashBase6_IdentialAndIdentityHashCode6_LinkedHashMapMixin@3220832___regenerab
66	LOAL spylc			__CompactLinkedIdentityHashMap6_HashFieldBase6MapMixin5_HashBase6_IdentialAndIdentityHashCode6_LinkedHashMapMixin@3220832___regenerab
67	LOAL spylc			sub_3CB1E4+4h
68	LOAL spylc			sub_3CD0F4+38h
69	LOAL spylc			sub_3DBB0C+558h
70	LOAL spylc			sub_3DBB0C+558h
71	LOAL spylc			sub_3DBB0C+558h
72	LOAL spylc			sub_3DBB0C+558h
73	LOAL spylc			sub_3DBB0C+558h
74	LOAL spylc			sub_3DBB0C+558h
75	LOAL spylc			sub_3DBB0C+558h
76	LOAL spylc			sub_3DBB0C+558h
77	LOAL spylc			sub_3DBB0C+558h
78	LOAL spylc			sub_3DBB0C+558h
79	LOAL spylc			sub_3DBB0C+558h
80	LOAL spylc			sub_3DBB0C+558h
81	LOAL spylc			sub_3DBB0C+558h
82	LOAL spylc			sub_3DBB0C+558h
83	LOAL spylc			sub_3DBB0C+558h
84	LOAL spylc			sub_3DBB0C+558h
85	LOAL spylc			sub_3DBB0C+558h
86	LOAL spylc			sub_3DBB0C+558h
87	LOAL spylc			sub_3DBB0C+558h
88	LOAL spylc			sub_3DBB0C+558h
89	LOAL spylc			sub_3DBB0C+558h
90	LOAL spylc			sub_3DBB0C+558h
91	LOAL spylc			sub_3DBB0C+558h
92	LOAL spylc			sub_3DBB0C+558h
93	LOAL spylc			sub_3DBB0C+558h
94	LOAL spylc			sub_3DBB0C+558h
95	LOAL spylc			sub_3DBB0C+558h
96	LOAL spylc			sub_3DBB0C+558h
97	LOAL spylc			sub_3DBB0C+558h
98	LOAL spylc			sub_3DBB0C+558h
99	LOAL spylc			sub_3DBB0C+558h
100	LOAL spylc			sub_3DBB0C+558h

75 matches (completed)

Close Help Legacy Dialog ...

# What is the name of this Flutter function?

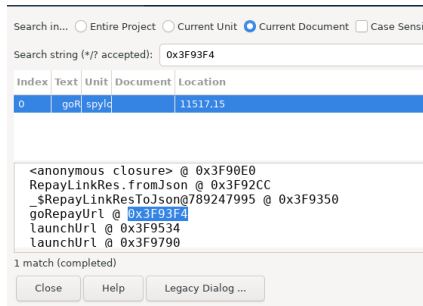
```
sub_4093F4      proc
...
STUR          X0, [X29, #FFFFFFE8h]
ADD           X17, X27, #Bh, LSL #12
LDR           X17, [X17, #C88h]
STUR          W17, [X0, #Fh]
LDUR          X1, [X29, #FFFFFFF0h]
LDUR          W2, [X1, #7]
ADD           X2, X2, X28, LSL #32
STUR          W2, [X0, #13h]
ADD           X17, X27, #11h, LSL #12 ; goRepaymentUrl loaded from ObjectPool
LDR           X17, [X17, #B8h]
```

- JEB relocation base for zero based relocatable objects:  
**0x10000** (Options / General / Back-end properties: root, parsers, native, disas)
- The function is at  $0x4093F4 - 0x10000 = 0x3F93F4$
- Search Code Information for **0x3F93F4**





# Function name found



Java

MethodChannel

Flutter

goRepaymentUrl()

goRepayUrl()



# Flutter apps for Android AArch64: status

## With JEB

- Read the Object Pool: **Yes** (strings only)
- Find function names: **Yes** via Code Information.
- Find string cross references: **Yes** via Search.

Are we lost *without* JEB?



# There is still hope



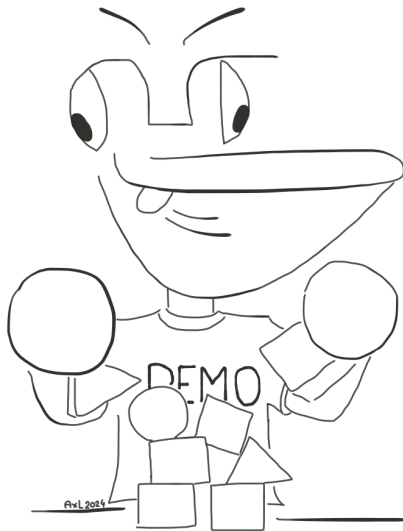
- Blutter: <https://github.com/worawit/blutter>
- Only works for **recent Android Flutter AAarch64**
- Requires **GCC 13**
- `python3 blutter.py ./malware/spyloan/arm64-v8a outputdir`

## Output

- `pp.txt`: all Dart objects in the Object Pool
- `asm/`: assembly code



# Demo



# Finding goRepaymentUrl with Blutter

```
grep -C 3 goRepaymentUrl pp.txt
```

```
[pp+0x110a8] String: ""  
[pp+0x110b0] List(7) [0, 0x2, 0x2, 0x1, "mode", 0x1, Null]  
[pp+0x110b8] String: "supportData"  
[pp+0x110c0] String: "goRepaymentUrl"  
[pp+0x110c8] String: "68796154717673"  
[pp+0x110d0] Null  
[pp+0x110d8] String: " in type cast"
```

- Blutter finds offset **0x110c0**
- In reality, assembly loads **0x110b8** (0x110c0-8)



# Finding function name with Blutter

Search for function at **0x3F93F4**

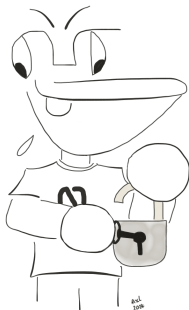
```
$ grep -ri 3F93F4 ./asm/  
./asm/flutter_project/plugin/Plugin.dart:    // ** addr: 0x3f93f4  
...
```

**./asm/flutter\_project/plugin/Plugin.dart**

```
static _ goRepayUrl(/* No info */) async {  
    // ** addr: 0x3f93f4, size: 0x140  
    // 0x3f93f4: EnterFrame  
    //    0x3f93f4: stp                fp, lr, [SP, #-0x10]!  
    //    0x3f93f8: mov                fp, SP  
    // 0x3f93fc: AllocStack(0x18)  
    //    0x3f93fc: sub                SP, SP, #0x18  
    // 0x3f9400: SetupParameters (dynamic _ /* r1, fp-0x10 */) }
```



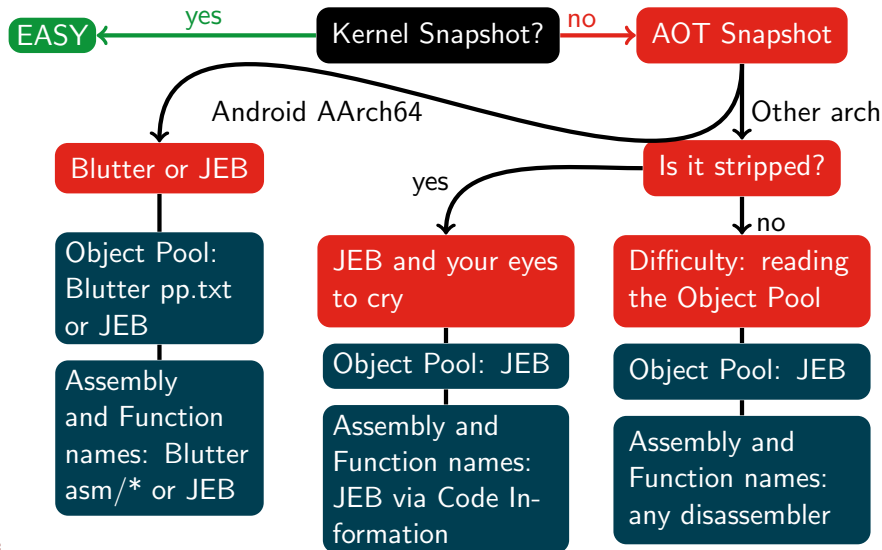
# Goals unlocked



- 1 Understand how to reverse ~~Flutter~~ applications for Android, especially malware. ~~Release~~ applications → ~~Dart AOT snapshot.~~ **DONE**
- 2 Solve ~~GreHack CTF 2023 Dart challenge~~ → It's a ~~Dart AOT snapshot.~~ **DONE**

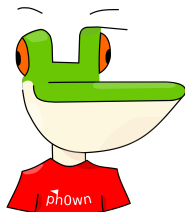


# Reversing Dart / Flutter





# Thanks for your attention!



- <https://github.com/cryptax/talks>
- @cryptax (X, Mastodon.social)
- <https://ph0wn.org> CTF - November 29-30, 2024

