Google Play Protect

Unpacking the Packed Unpacker

Reversing an Android Anti-Analysis Native Library

Maddie Stone @maddiestone Virus Bulletin 2018



Who am I? - Maddie Stone

- Reverse Engineer on Google's Android Security Team
- 5+ years hardware & firmware reversing of embedded devices
- BS in Computer Science, Russian, & Applied Math
- MS in Computer Science
- Presented at conferences, such as REcon, OffensiveCon, DerbyCon, and BlackHat USA.



@maddiestone

Malware Analysts vs Malware Authors

striving for the asymmetric advantage

Anti-analysis: techniques to frustrate analysis and make reverse engineering malware more difficult

Objective: Determine if an app is malware. Quickly.

- Have an app that looks suspicious, but need evidence to determine if it's malware
 - App won't run in dynamic analysis
 - Most code is native
 - Many similar apps

Introduction - Target of Analysis



Target of Analysis: Android App Native Library (*.so in ELF format)

Introducing WeddingCake!

...because it has lots of layers

WeddingCake Anti-Analysis Techniques



Check for Xposed Framework

Characteristics of WeddingCake

- Android native library included in APKs as .so/ELF
- Different name in each sample lib[3-8 random lowercase characters].so
- Java classes that interface with library have random names -> different for each sample ses.fdkxxcr.udayjfrgxp.ojoyqmosj.xien.xmdowmbkdgfgk
- Two strings under the .comment section in the ELF: Android clang version 3.8.275480 (based on LLVM 3.8.275480) GCC: (GNU) 4.9.x 20150123 (prerelease)

Characteristics of WeddingCake

- Two Java-declared native methods with the following signatures
 - **The names of the methods change in each sample**

public native int vxeg(Object[] p0); Performs run-time environment checks and the ELF's main functionality public static native String quaqrd(int p0); Returns string at index p0 in a hard-coded array

 Samples often have a 3rd method declared: public native Object ixkjwu(Object[] p0);

CPU Variants of WeddingCake

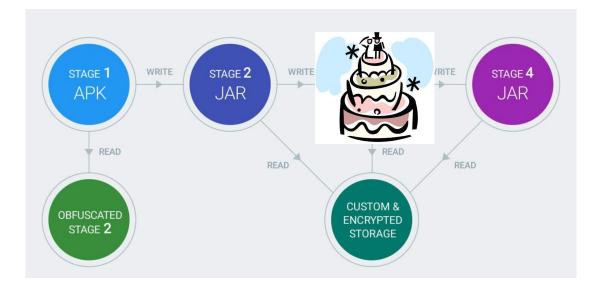
- 32-bit "generic" ARM is most common (armeabi)
- Also seen other versions of the library:
 - 32-bit ARMv7 (armeabi-v7a)
 - ARM64 (arm64-v8a)
 - x86 (x86)

92e80872cfd49f33c63993d52290afd2e87cbef5db4adff1bfa97297340f23e0 <u>https://bit.ly/2L18eT0</u>

Anti-Analysis Lib File Paths	Anti-Analysis Library "Type"
lib/armeabi/librxovdx.so	32-bit "generic" ARM
lib/armeabi-v7a/librxovdx.so	32-bit ARMv7
lib/x86/libaojjp.so	x86

Purpose of WeddingCake

- 5000+ distinct APK samples containing WeddingCake
- Used by newer variants of <u>Chamois</u> family
- Protects functionality that authors want to hide by "wrapping" it in anti-analysis protections



Analyzing WeddingCake

Sample: e8e1bc048ef123a9757a9b27d1bf53c092352a26bdbf9fbdc10109415b5cadac https://bit.ly/2Nkc4ZS

- JNI allows developers to declare Java native methods that run in other languages (C/C++) in the application
- Native methods are declared in Java

public static native String quaqrd(int p0);

public native Object ixkjwu(Object[] p0);

public native int vxeg(Object[] p0);

 The declared Java native method is implemented in C/C++

"Getting Started with the NDK", Android, <u>https://developer.android.com/ndk/guides/</u>

"JNI Tips", Android, <u>https://developer.android.com/training/articles/perf-jni</u>

When load() or loadLibrary() is called in Java, the ELF is "loaded" and JNI_OnLoad() is run in the ELF

- "Registering" native methods: pair the Java method declaration to the correct subroutine in the native library
 - "Discovery": the function names and function signatures matching in both Java and the .so Java_<mangled class name>_<mangled method_name>
 - RegisterNatives JNI function
 - Requires string of the method name and the string of the method signature

"Resolving Native Method Names", Oracle, <u>https://docs.oracle.com/javase/6/docs/technotes/guides/jni/spec/design.html#wp615</u>

"Registering Native Methods in JNI", Stack Overflow, https://stackoverflow.com/questions/1010645/what-does-the-registernatives-method-do

jint RegisterNatives(JNIEnv *env, jclass clazz, const JNINativeMethod *methods, jint nMethods);

typedef struct {

char *name; char *signature;

void *fnPtr;

} JNINativeMethod;

• Signatures

First Look

- None of the native method names exist in native lib (as funcs or strings)
- JNI_OnLoad is exported, but not defined in IDA
 - The bytes at +0x24,
 +0x28, and +0x44 are defined as data
- No strings
 - Including method names and signatures

text:00001B20	;	
text:00001B20		
text:00001B20		EXPORT JNI_OnLoad
text:00001B20	JNI OnLoad	
text:00001B20 F0 B5	-	PUSH {R4-R7, LR}
text:00001B22 03 AF		ADD R7, SP, #0xC
text:00001B24 9D B0		SUB SP, SP, #0x74
text:00001B26 07 49		LDR R1, = (stack chk guard ptr - 0x1B2C)
text:00001B28 79 44		ADD R1, PC ;stack_chk_guard_ptr
text:00001B2A 09 68		LDR R1, [R1] ;stack_chk_guard
text:00001B2C 09 68		LDR R1, [R1]
text:00001B2E 1C 91		STR R1, [SP,#0x70]
text:00001B30 00 25		MOVS R5, #0
text:00001B32 1B 95		STR R5, [SP, #0x6C]
text:00001B34 EE 43		MVNS R6, R5
text:00001B36 04 49		LDR R1, =(byte A450 - 0x1B3C)
text:00001B38 79 44		ADD R1, PC; byte A450
text:00001B3A 09 78		LDRB R1, [R1]
text:00001B3C 00 29		CMP R1, #0
text:00001B3E 05 D0		BEQ loc 1B4C
text:00001B40 00 F0 93 FF		BL sub_2A6A
text:00001B40	:	
text:00001B44 80 73 00 00	off 1B44	DCDstack_chk_guard_ptr - 0x1B2C
text:00001B44		; DATA XREF: .text:00001B26 [†] r
text:00001B48 14 89 00 00	off 1B48	DCD byte_A450 - 0x1B3C ; DATA XREF: .text:00001B36fr
text:00001B4C		
	1	
text:00001B4C		
text:00001B4C text:00001B4C	loc 1B4C	; CODE XREF: .text:00001B3E ¹ j
	loc_1B4C	; CODE XREF: .text:00001B3E [†] j STR R0, [SP,#0x14]
text:00001B4C	loc_1B4C	
text:00001B4C text:00001B4C 05 90	loc_1B4C	STR R0, [SP, #0x14]
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48	loc_1B4C	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54)
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44	loc_1B4C	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21	loc_1B4C	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B54 01 70	loc_1B4C	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC; byte_A450 MOVS R1, #1 STRB R1, [R0]
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B54 01 70 text:00001B56 13 91		STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRB R1, [R0] STR R1, [SP,#0x4C]
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B54 01 70 text:00001B56 13 91 text:00001B58 0C 02		STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC; byte_A450 MOVS R1, #1 STRB R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 13 91 text:00001B58 0C 02 text:00001B58 10 B4		STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRB R1, [R0] STR R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH {R4} POP {R0}
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 11 91 text:00001B56 13 91 text:00001B58 0C 02 text:00001B58 10 B4 text:00001B5C 01 BC		STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRB R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH (R4) POP (R0) BL j_jmalloc
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B54 01 70 text:00001B56 13 91 text:00001B56 10 B4 text:00001B5A 10 B4 text:00001B5E 05 F0 33 F8	;	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STR R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH [R4] POP (R0) BL j_jmalloc B loc_1B68
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 13 91 text:00001B58 0C 02 text:00001B58 10 B4 text:00001B5C 01 BC text:00001B5C 01 BC text:00001B5C 01 E0	;	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STR R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH [R4] POP (R0) BL j_jmalloc B loc_1B68
text:00001B4C text:00001B4C 05 90 text:00001B50 78 44 text:00001B50 78 44 text:00001B50 78 44 text:0001B54 01 70 text:00001B54 01 70 text:00001B56 13 91 text:00001B56 10 64 text:00001B50 10 84 text:00001B50 10 84 text:00001B50 10 84 text:00001B52 05 F0 33 F8 text:00001B62 01 E0 text:00001B62 01 E0	;	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRB R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH (R4) POP (R0) BL j_jmalloc
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 13 91 text:00001B56 05 C0 22 text:00001B5C 01 BC text:00001B5C 05 F0 33 F8 text:00001B62 01 E0 text:00001B62 FC 88 00 00	;	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STR R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH [R4] POP (R0) BL j_jmalloc B loc_1B68
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 13 91 text:00001B56 10 B4 text:00001B5C 01 BC text:00001B5C 01 BC text:00001B62 01 E0 text:00001B62 text:00001B62 text:00001B62	;	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STR R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH [R4] POP (R0) BL j_jmalloc B loc_1B68
<pre>text:00001B4C text:00001B4E 05 90 text:00001B50 78 44 text:00001B50 78 44 text:00001B52 01 21 text:00001B54 01 70 text:00001B56 13 91 text:00001B58 05 C02 text:00001B52 01 BC text:00001B52 05 F0 33 F8 text:00001B62 01 E0 text:00001B64 FC 88 00 00 text:00001B68</pre>	; off_1B64 ;	STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STR R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 POSH [R4] POP [R0] BL j_jmalloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E ^î r
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 13 91 text:00001B5C 01 BC text:00001B5C 01 BC text:00001B62 01 E0 text:00001B62 FC 88 00 00 text:00001B68 text:00001B68	; off_1B64 ;	<pre>STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRE R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH [R4} POP {R0} BL j_jmalloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E^îr ; CODE XREF: .text:00001B42^ĵj</pre>
<pre>text:00001B4C text:00001B4E 05 90 text:00001B50 78 44 text:00001B50 78 44 text:00001B50 121 text:00001B54 01 70 text:00001B56 13 91 text:00001B56 10 B4 text:00001B52 05 F0 33 F8 text:00001B52 05 F0 33 F8 text:00001B62 text:00001B64 text:00001B64 text:00001B68 text:00001B68 text:00001B68</pre>	; off_1B64 ;	<pre>STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRB R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH [R4] POP {R0} BL j_j_malloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E[↑]r ; CODE XREF: .text:00001B62[↑]j ; .text:00001B6E↓j</pre>
<pre>text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B54 01 70 text:00001B56 13 91 text:00001B58 10 B4 text:00001B52 01 EC text:00001B52 05 F0 33 F8 text:00001B62 text:00001B62 text:00001B64 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68</pre>	; off_1B64 ;	<pre>STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRE R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH {R4} POP {R0} BL j_j_malloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E^îr ; CODE XREF: .text:00001B4E^îj ; .text:00001B62^ĵj ; .text:00001B62^ĵj</pre>
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 13 91 text:00001B5A 10 B4 text:00001B5C 01 BC text:00001B5C 01 BC text:00001B62 01 E0 text:00001B62 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 45 55 text:00001B68 01 35	; off_1B64 ;	<pre>STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRE R1, [R0] STRE R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH (R4) POP {R0} BL j_jmalloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E¹r ; CODE XREF: .text:00001B62¹j ; .text:00001B6E₄j STRB R5, [R0,R5] ADDS R5, #1</pre>
<pre>text:00001B4C text:00001B4C 05 90 text:00001B50 78 44 text:00001B50 78 44 text:00001B50 121 text:00001B54 01 70 text:00001B56 13 91 text:00001B56 05 F0 33 F8 text:00001B52 05 F0 33 F8 text:00001B62 01 E0 text:00001B64 text:00001B64 text:00001B68 text:00001B68</pre>	; off_1B64 ;	<pre>STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRE R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH {R4} POP {R0} BL j_j_malloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E[†]r ; CODE XREF: .text:00001B4E[†]r ; .text:00001B6E_j STRE R5, [R0,R5] ADDS R5, #1 CMP R4, R5 BNE loc_1B68</pre>
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 10 B4 text:00001B50 10 BC text:00001B50 01 BC text:00001B62 01 E0 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 45 55 text:00001B6A 01 35 text:00001B6A C 42 text:00001B6A FF D1	; off_1B64 ;	<pre>STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRE R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH {R4} POP {R0} BL j_j_malloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E[†]r ; CODE XREF: .text:00001B4E[†]r ; .text:00001B6E_j STRE R5, [R0,R5] ADDS R5, #1 CMP R4, R5 BNE loc_1B68</pre>
text:00001B4C text:00001B4C 05 90 text:00001B4E 05 48 text:00001B50 78 44 text:00001B52 01 21 text:00001B56 13 91 text:00001B56 13 91 text:00001B58 0C 02 text:00001B52 01 BC text:00001B52 01 BC text:00001B62 01 E0 text:00001B62 text:00001B68 text:00001B68 text:00001B68 text:00001B68 text:00001B68 155 text:00001B68 01 35 text:00001B6C AC 42 text:00001B6C FB D1 text:00001B70 06 4D	; off_1B64 ;	<pre>STR R0, [SP,#0x14] LDR R0, = (byte_A450 - 0x1B54) ADD R0, PC ; byte_A450 MOVS R1, #1 STRB R1, [R0] STR R1, [SP,#0x4C] LSLS R4, R1, #8 PUSH (R4) POP (R0) BL j_j_malloc B loc_1B68 DCD byte_A450 - 0x1B54 ; DATA XREF: .text:00001B4E[†]r ; CODE XREF: .text:00001B62[†]j ; .text:00001B6E_↓j STRB R5, [R0,R5] ADDS R5, #1 CMP R4, R5 BNE loc_1B68 LDR R5, = (off_2C08+1)</pre>

Beginning Analysis

- Start with JNI_OnLoad
- Repetitive calls to same function over different blocks of memory → Encryption

sub_2F30: Decryption Subroutine

ſ	00001C62	E 8	48			LDR	$R0, = (unk_{90EC} - 0x1C68)$
	00001C64	78	44			ADD	R0, PC ; unk_90EC
	00001C66	37	21			MOVS	R1, #0x37
	00001C68	04	91			STR	R1, [SP,#0x80+var_70]
	00001C6A	10	в4			PUSH	{R4}
	00001C6C	04	BC			POP	{R2}
	00001C6E	20	в4			PUSH	{R5}
	00001c70	08	BC			POP	{R3}
	00001C72	01	FO	5D	F9	BL	sub_2F30
L	00001076	-	10				DA (1 0102 0 1070)

	00001c0A	20	в4			PUSH	{R5}	
	00001C0C					POP	{R3}	
	00001C0E	01	FO	8F	F9		sub_2F30	
	00001c12						$R0, = (unk_{907F} - 0x1C18)$	
	00001c14						R0, PC ; unk_907F	
	00001c16					MOVS	R1, #0x18	
	00001C18					STR	R1, [SP,#0x80+var_30]	
	00001C1A 00001C1C					PUSH	{R4}	
	0000101	20	DA.			PUSH	{R2} {R5}	
	00001C1E	0.8	BC			POP	(R3)	
	00001C22	01	FO	85	F9		sub_2F30	
	00001C26						R0, = (unk_9097 - 0x1C2C)	
	00001c28						R0, PC ; unk_9097	
	00001c2A					PUSH	{R6}	
	00001C2C					POP	{R1}	
	00001C2E	10	в4			PUSH	{R4}	
	00001C30					POP	{R2}	
	00001C32					PUSH	{R5}	
	00001C34						{R3}	
	00001C36	01	FO	7B	F9	BL	sub_2F30	
	00001C3A	FU	48			LDR	R0, =(unk_90B6 - 0x1C40) R0, PC ; unk_90B6	
	00001C3C 00001C3E	10				ADD	{R6}	
	00001C3E					POP	(R1)	
	00001C40					PUSH	{R4}	
	00001C44					POP	{R2}	
	00001C46					PUSH	{R5}	
	00001C48					POP	(R3)	
	00001C4A	01	FO	71	F9	BL	sub_2F30	
	00001C4E	EC	48			LDR	$R0, = (unk_{90D5} - 0x1C54)$	
	00001c50 00001c52	78	44			ADD	R0, PC ; unk_90D5	
	00001C52	17	21				R1, #0x17	
	00001c54						R1, [SP,#0x80+var_34]	
	00001c56						{R4}	
	00001C58 00001C5A					POP	{R2}	
	00001C5A					PUSH	{R5} {R3}	
_	00001055			67			sub_2F30	
F	00001C62					LDR	R0, = (unk_90EC - 0x1C68)	
	00001C64					ADD	R0, PC ; unk_90EC	
	00001C66	37	21			MOVS	R1, #0x37	
	00001C68					STR	R1, [SP,#0x80+var_70]	
	00001C6A					PUSH	{R4}	
	00001C6C					POP	{R2}	
	00001C6E 00001C70					PUSH	{R5}	
	00001C70				FO	POP	{R3} sub_2F30	
/ L	00001072	101		50		LUK	R0, = (unk_9125 = 0x10/0)	
	00001C78					ADD	R0, PC ; unk_9123	
	00001C7A					MOVS	R1, #0x14	
	00001c7c					STR	R1, [SP,#0x80+var_44]	
	00001C7E	10				PUSH	{R4}	
	00001C80	04				POP	{R2}	
	00001C82	04 20	в4			PUSH	(R2) (R5)	
	00001C82 00001C84	04 20 08	B4 BC			PUSH	<pre>{R2} {R5} {R3}</pre>	
	00001C82 00001C84 00001C86	04 20 08 01	B4 BC F0	53	F9	PUSH POP BL	{R2} {R5} {R3} sub_2F30	
	00001C82 00001C84 00001C86 00001C8A	04 20 08 01 E0	B4 BC F0 48	53	F9	PUSH POP BL LDR	{R2} {R5} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90)	
	00001C82 00001C84 00001C86 00001C8A 00001C8A	04 20 08 01 E0 78	B4 BC F0 48 44	53	F9	PUSH POP BL LDR ADD	<pre>{R2} {R5} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90) R0, PC; unk_9137</pre>	
	00001C82 00001C84 00001C86 00001C8A	04 20 08 01 E0 78 1A	B4 BC F0 48 44 21	53	F9	PUSH POP BL LDR	<pre>{R2} {R5} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90) R0, PC ; unk_9137 R1, #0x1A</pre>	
	00001C82 00001C84 00001C86 00001C8A 00001C8C 00001C8E	04 20 08 01 E0 78 1A 02	B4 BC F0 48 44 21 91	53	F9	PUSH POP BL LDR ADD MOVS	<pre>{R2} {R5} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90) R0, PC; unk_9137</pre>	
	00001C82 00001C84 00001C86 00001C8A 00001C8C 00001C8E 00001C90	04 20 08 01 E0 78 1A 02 10	B4 BC F0 48 44 21 91 B4	53	F9	PUSH POP BL LDR ADD MOVS STR	<pre>{R2} {R5} {R3} sub_2730 R0, = (unk_9137 - 0x1C90) R0, PC; unk_9137 R1, #0x1A R1, [Sp,#0x80+var_78]</pre>	
	00001c82 00001c84 00001c86 00001c8A 00001c8C 00001c8E 00001c90 00001c92 00001c94 00001c96	04 20 08 01 E0 78 1A 02 10 04 20	B4 BC F0 48 44 21 91 B4 BC B4	53	F9	PUSH POP BL LDR ADD MOVS STR PUSH POP PUSH	<pre>{R2} {R2} {R3} sub_2730 R0, =(unk_9137 - 0x1C90) R0, PC ; unk_9137 R1, #0x1A R1, [SP,#0x80+var_78] {R4} {R2} {R5}</pre>	
	00001c82 00001c84 00001c86 00001c8c 00001c8c 00001c90 00001c92 00001c94 00001c98	04 20 08 01 E0 78 1A 02 10 04 20 08	B4 BC F0 48 44 21 91 B4 BC B4 BC	53	F9	PUSH POP BL LDR ADD MOVS STR PUSH POP PUSH POP	<pre>{R2} {R5} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90) R0, PC; unk_9137 R1, #0x1A R1, [SP,#0x80+var_78] {R4} {R2} {R5} {R3}</pre>	
	00001C82 00001C84 00001C86 00001C8C 00001C8C 00001C8C 00001C90 00001C92 00001C96 00001C98 00001C98	04 20 08 01 E0 78 1A 02 10 04 20 08 01	B4 BC F0 48 44 21 91 B4 BC B4 BC F0	53	F9	PUSH POP BL LDR ADD MOVS STR PUSH POP PUSH POP BL	<pre>{R2} {R2} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90) R0, PC; unk_9137 R1, #0x1A R1, [SP,#0x80+var_78] {R4} {R2} {R5} {R3} sub_2F30</pre>	
	00001C82 00001C84 00001C86 00001C8C 00001C8C 00001C8C 00001C92 00001C92 00001C94 00001C96 00001C98 00001C9A	04 20 08 01 E0 78 1A 02 10 04 20 04 20 08 01 DC	B4 BC F0 48 44 21 91 B4 BC B4 BC F0 48	53	F9	PUSH POP BL LDR ADD MOVS STR PUSH POP PUSH POP BL LDR	<pre>{R2} {R3} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90) R0, PC ; unk_9137 R1, #0x1A R1, [SP, #0x80+var_78] {R4} {R2} {R4} {R2} {R3} sub_2F30 R0, = (unk_9151 - 0x1CA4)</pre>	
	00001C82 00001C84 00001C86 00001C8C 00001C8C 00001C8C 00001C90 00001C92 00001C96 00001C98 00001C98	04 20 08 01 E0 78 1A 02 10 04 20 04 20 08 01 DC	B4 BC F0 48 44 21 91 B4 BC B4 BC F0 48	53	F9	PUSH POP BL LDR ADD MOVS STR PUSH POP PUSH POP BL	<pre>{R2} {R2} {R3} sub_2F30 R0, = (unk_9137 - 0x1C90) R0, PC; unk_9137 R1, #0x1A R1, [SP,#0x80+var_78] {R4} {R2} {R5} {R3} sub_2F30</pre>	

sub_2F30(Byte[] encrypted_array, int length, Word[]
word_seed_array, Byte[] byte_seed_array)

- encrypted_array: Pointer to the encrypted byte array (bytes to be decrypted)
- length: Length of the encrypted byte array
- word_seed_array: Word (each value in array is 4 bytes) seed array
- byte_seed_array: Byte (each value in array is 1 byte) seed array

Generating the Seed Arrays

```
byte_seed_array = malloc(0 \times 100 u);
                                            while ( curr_count );
index = 0;
                                            word_seed_array = malloc(0x400u);
 do
                                            index = 0:
   byte_seed_array[index] = index;
                                            do
 {
    ++index; }
 while ( 256 != index );
                                              word_seed_array[byte_seed_array[index]] =
 v4 = 0x2C09;
                                                 index;
                                              ++index;
 curr_count = 256;
 copy_byte_seed_array = byte_seed_array
                                            }
 do
                                            while ( 256 != index );
 {
     v6 = 0x41C64E6D * v4 + 0x3039;
     v7 = v6;
     v8 = copy_byte_seed_array[v6];
     v9 = 0x41C64E6D * (v6 & 0x7FFFFFFF) + 0x3039;
     copy_byte_seed_array[v7] = copy_byte_seed_array[v9];
     copy_byte_seed_array[v9] = v8;
     --curr_count;
     v4 = v9 \& 0x7FFFFFF;
```

Generating the Seed Array: Anti-Reversing

- Output of the Seed Array Generation Algorithms:
 - Byte Seed Array byte array from 0 to 0xFF
 - Word Seed Array word (4 bytes) array from 0 to 0xFF
- Anti-Reversing Technique
 - Complex algorithm instead of simple algorithm

Maybe not purposeful?

Chris Eng noticed the constants match the glibc's implementation of rand (simple linear congruential generator). The developers may have accidentally reduced the period size.

- Bypass:
 - Run dynamically and capture arrays

Decryption Algorithm

- The overall framework of the in-place decryption process is:
 - 1) Decryption function is called on an array of encrypted bytes
 - 2) Decryption is performed
 - 3) The encrypted bytes are overwritten by the decryption bytes

• Not identified as any known encryption/decryption algorithm

Decrypting the Library

- Need to decrypt the native library quickly for further analysis
 - Don't need to understand the decryption → just need to build a solution to decrypt it
- Want any solution to be applicable to the multitude of samples
 - Different memory address, registers

IDAPython Decryption Script:

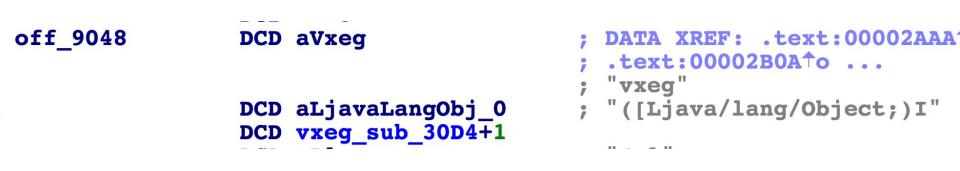
http://www.github.com/maddiestone/IDAPythonEmbeddedToolkit/And roid/WeddingCake_decrypt.py

Decrypted Contents

Each of the encrypted arrays decrypts to a string

00009480																	·
00009400	01	F5	FO	81	88	94	F1	C6	29	18	2F	DD	0C	34	AE	32)./4.2
00009490	EA	8E	53	58	0C	52	EE	BE	2F	05	F5	OF	C2	FC	18	BA	
000094A0	B 3	6E	36	39	C7	D2	FD	D5	FE	73	4 B	3A	A3	06	FE	D3	.n69sK:
000094B0	F5	88	46	0A	DC	14	28	D8	CB	5D	59	44	EB	2E	FD	A2	F(]YD
000094C0	FO	8C	56	61	E1	F2	36	E2	1A	91	2C	65	ED	31	3E	EE	Va6,e.1>.
000094D0	17	A6	34	48	F5	47	42	00	20	75	0D	CD	C2	56	98	9E	4H.GBuV
000094E0	54	7E	FB	08	59	47	3E	E4	CC	0C	FB	90	DE	5C	FE	9B	T~YG>\
000094F0	5F	AA	AF	63	67	D7	EF	E5	C6	80	99	9B	94	1B	6F	24	cg
00009500	AB	22	DF	38	3A	OF	3D	84	A 8	5E	94	7E	D2	D0	6B	8F	.".8:.=^.~k.
00009510	4E	C8	D5	75	A4	89	D8	DF	3B	78	98	DD	E4	76	A 8	A2	Nu;xv
00009520	C6	DA	89	AE	9F	EF	DF	8D	7F	38	15	5A	5A	FA	22	05	8.ZZ.".
00009530	8B	F9	88	E0	8A	00	4C	E9	0B	9B	7D	91	8F	BE	05	A2	L}
00009540	DE	71	DE	3F	8E	25	67	25	CC	DA	81	95	2B	44	33	OF	.q.?.%g%+D3.
00009550	0D	52	3B	2E	AA	B6	E8	3C	AE	33	FB	4D	EF	14	6E	2A	.R;<.3.Mn*
00009560	11	D1	65	B2	E 8	D6	44	B0	5F	A2	49	48	EC	0E	E3	29	eDIH)
00009570					E3						C3		77		A9		.2/*w7
00009580	E9	EC	8A	01	7D	61	F7	03	8B	0E	BB	4F	B2	E3	92	07	}a0
00009480	01	F 5	FO	01	00	94	F1	CG	29	18	25	DD	00	20	50	12	
									100)./([B
00009490	29	5B	42	00	0C	52	EE	BE	2F	05	28	4C	6A	61	76	61)[BR/.(Ljava
00009490 000094A0	29 2F	5B 6C	42 61	00 6E	0C 67	52 2F	EE 53	BE 74	2F 72	05 69	28 6E	4C 67	6A 3B	61 5B	76 4C	61 6A)[BR/.(Ljava /lang/String;[Lj
00009490 000094A0 000094B0	29 2F 61	5B 6C 76	42 61 61	00 6E 2F	0C 67 6C	52 2F 61	EE 53 6E	BE 74 67	2F 72 2F	05 69 43	28 6E 6C	4C 67 61	6A 3B 73	61 5B 73	76 4C 3B	61 6A 29) [BR/.(Ljava /lang/String; [Lj ava/lang/Class;)
00009490 000094A0 000094B0 000094C0	29 2F 61 4C	5B 6C 76 6A	42 61 61 61	00 6E 2F 76	0C 67 6C 61	52 2F 61 2F	EE 53 6E 6C	BE 74 67 61	2F 72 2F 6E	05 69 43 67	28 6E 6C 2F	4C 67 61 72	6A 3B 73 65	61 5B 73 66	76 4C 3B 6C	61 6A 29 65) [BR/.(Ljava /lang/String;[Lj ava/lang/Class;) Ljava/lang/refle
00009490 000094A0 000094B0 000094C0 000094D0	29 2F 61 4C 63	5B 6C 76 6A 74	42 61 61 61 2F	00 6E 2F 76 4D	0C 67 6C 61 65	52 2F 61 2F 74	EE 53 6E 6C 68	BE 74 67 61 6F	2F 72 2F 6E 64	05 69 43 67 3B	28 6E 6C 2F 00	4C 67 61 72 CD	6A 3B 73 65 C2	61 5B 73 66 56	76 4C 3B 6C 98	61 6A 29 65 9E) [BR/.(Ljava /lang/String;[Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V
00009490 000094A0 000094B0 000094C0 000094D0 000094E0	29 2F 61 4C 63 54	5B 6C 76 6A 74 7E	42 61 61 2F FB	00 6E 2F 76 4D 08	0C 67 6C 61 65 6A	52 2F 61 2F 74 61	EE 53 6E 6C 68 76	BE 74 67 61 6F 61	2F 72 2F 6E 64 2F	05 69 43 67 3B 6C	28 6E 6C 2F 00 61	4C 67 61 72 CD 6E	6A 3B 73 65 C2 67	61 5B 73 66 56 2F	76 4C 3B 6C 98 49	61 6A 29 65 9E 6E) [BR/.(Ljava /lang/String;[Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In
00009490 000094A0 000094B0 000094C0 000094D0 000094E0 000094F0	29 2F 61 4C 63 54 74	5B 6C 76 6A 74 7E 65	42 61 61 2F FB 67	00 6E 2F 76 4D 08 65	0C 67 61 65 6A 72	52 2F 61 2F 74 61 00	EE 53 6E 6C 68 76 EF	BE 74 67 61 6F 61 E5	2F 72 2F 6E 64 2F C6	05 69 43 67 3B 6C 80	28 6E 2F 00 61 99	4C 67 61 72 CD 6E 9B	6A 3B 73 65 C2 67 94	61 5B 73 66 56 2F 1B	76 4C 3B 6C 98 49 6F	61 6A 29 65 9E 6E 24) [BR/.(Ljava /lang/String;[Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$
00009490 000094A0 000094B0 000094C0 000094D0 000094E0 000094F0 00009500	29 2F 61 4C 63 54 74 AB	5B 6C 76 6A 74 7E 65 22	42 61 61 2F FB 67 DF	00 6E 2F 76 4D 08 65 38	0C 67 61 65 6A 72 3A	52 2F 61 2F 74 61 00 0F	EE 53 6E 6C 68 76 EF 28	BE 74 67 61 6F 61 E5 29	2F 72 2F 6E 64 2F C6 4C	05 69 43 67 3B 6C 80 61	28 6E 2F 00 61 99 6E	4C 67 61 72 CD 6E 9B 64	6A 3B 73 65 C2 67 94 72	61 5B 73 66 56 2F 1B 6F	76 4C 3B 6C 98 49 6F 69	61 6A 29 65 9E 6E 24 64) [BR/.(Ljava /lang/String;[Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid
00009490 000094A0 000094B0 000094C0 000094D0 000094E0 000094F0 00009500 00009510	29 2F 61 4C 63 54 74 AB 2F	5B 6C 76 6A 74 7E 65 22 63	42 61 61 2F FB 67 DF 6F	00 6E 2F 76 4D 08 65 38 6E	0C 67 61 65 6A 72 3A 74	52 2F 61 2F 74 61 00 0F 65	EE 53 6E 6C 68 76 EF 28 6E	BE 74 67 61 6F 61 E5 29 74	2F 72 2F 6E 64 2F C6 4C 2F	05 69 43 67 3B 6C 80 61 43	28 6E 2F 00 61 99 6E 6F	4C 67 61 72 CD 6E 9B 64 6E	6A 3B 73 65 C2 67 94 72 74	61 5B 73 66 56 2F 1B 6F 65	76 4C 3B 6C 98 49 6F 69 6E	61 6A 29 65 9E 6E 24 64 74) [BR/. (Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid /content/Content
00009490 000094A0 000094B0 000094C0 000094D0 000094E0 000094F0 00009500 00009510 00009520	29 2F 61 4C 63 54 74 AB 2F 52	5B 6C 76 6A 74 7E 65 22 63 65	42 61 61 2F FB 67 DF 6F 73	00 6E 2F 76 4D 08 65 38 6E 6F	0C 67 61 65 6A 72 3A 74 6C	52 2F 61 2F 74 61 00 0F 65 76	EE 53 6E 6C 68 76 EF 28 6E 65	BE 74 67 61 6F 61 E5 29 74 72	2F 72 2F 6E 64 2F C6 4C 2F 3B	05 69 43 67 3B 6C 80 61 43 00	28 6E 2F 00 61 99 6E 6F	4C 67 61 72 CD 6E 9B 64 6E 5A	6A 3B 73 65 C2 67 94 72 74 5A	61 5B 73 66 56 2F 1B 6F 65 FA	76 4C 3B 6C 98 49 6F 69 6E 22	61 6A 29 65 9E 6E 24 64 74 05) [BR/. (Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid /content/Content Resolver;ZZ.".
00009490 000094A0 000094C0 000094C0 000094D0 000094E0 000094F0 00009500 00009510 00009520 00009530	29 2F 61 4C 63 54 74 AB 2F 52 4C	5B 6C 76 6A 74 7E 65 22 63 65 6A	42 61 61 2F FB 67 DF 6F 73 61	00 6E 2F 76 4D 08 65 38 6E 6F 76	0C 67 6C 61 65 6A 72 3A 74 6C 61	52 2F 61 2F 74 61 00 0F 65 76 2F	EE 53 6E 6C 68 76 EF 28 6E 65 6C	BE 74 67 61 6F 61 E5 29 74 72 61	2F 72 2F 6E 64 2F C6 4C 2F 3B 6E	05 69 43 67 3B 6C 80 61 43 00 67	28 6E 2F 00 61 99 6E 6F 15 2F	4C 67 61 72 6E 9B 64 6E 5A 53	6A 3B 73 65 C2 67 94 72 74 5A 74	61 5B 73 66 56 2F 1B 6F 65 FA 72	76 4C 3B 6C 98 49 6F 69 6E 22 69	61 6A 29 65 9E 6E 24 64 74 05 6E) [BR/. (Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid /content/Content Resolver;ZZ.". Ljava/lang/Strin
00009490 000094A0 000094C0 000094C0 000094D0 000094E0 000094F0 00009500 00009510 00009520 00009530	29 2F 61 4C 63 54 74 AB 2F 52 4C 67	5B 6C 76 6A 74 7E 65 22 63 65 6A 3B	42 61 61 2F FB 67 DF 6F 73 61 00	00 6E 2F 76 4D 08 65 38 6E 6F 76 3F	0C 67 61 65 6A 72 3A 74 6C 61 8E	52 2F 61 2F 74 61 00 0F 65 76 2F 25	EE 53 6E 68 76 EF 28 6E 65 6C 67	BE 74 67 61 6F 61 E5 29 74 72 61 25	2F 72 2F 6E 64 2F C6 4C 2F 3B 6E CC	05 69 43 67 3B 6C 80 61 43 00 67 DA	28 6E 2F 00 61 99 6E 6F 15 2F 81	4C 67 61 72 6E 9B 64 6E 5A 53 95	6A 3B 73 65 C2 67 94 72 74 5A 74 2B	61 5B 73 66 56 2F 1B 6F 65 FA 72 44	76 4C 3B 6C 98 49 6F 69 6E 22 69 33	61 6A 29 65 9E 6E 24 64 74 05 6E 0F	<pre>) [BR/.(Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid /content/Content Resolver;ZZ.". Ljava/lang/Strin g;.?.%g%+D3.</pre>
00009490 000094A0 000094C0 000094C0 000094D0 000094E0 00009500 00009510 00009510 00009530 00009530	29 2F 61 4C 63 54 74 AB 2F 52 4C 67 0D	5B 6C 76 6A 74 7E 65 22 63 65 6A 3B 28	42 61 61 2F FB 67 DF 6F 73 61 00 29	00 6E 2F 76 4D 08 65 38 6E 6F 76 3F 5B	0C 67 61 65 6A 72 3A 74 6C 61 8E 42	52 2F 61 2F 74 61 00 0F 65 76 2F 25 00	EE 53 6E 6C 68 76 EF 28 6E 65 6C 67 FB	BE 74 67 61 6F 61 E5 29 74 72 61 25 60	2F 72 2F 6E 64 2F C6 4C 2F 3B 6E CC 63	05 69 43 67 3B 6C 80 61 43 00 67 DA 7B	28 6E 2F 00 61 99 6E 6F 15 2F 81 93	4C 67 61 72 6E 9B 64 6E 5A 53 95 A1	6A 3B 73 65 C2 67 94 72 74 5A 74 2B 9B	61 5B 73 66 56 2F 1B 6F 65 FA 72 44 C0	76 4C 3B 6C 98 49 6F 69 6E 22 69 33 75	61 6A 29 65 9E 6E 24 64 74 05 6E 0F 2A	<pre>) [BR/.(Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid /content/Content Resolver;ZZ.". Ljava/lang/Strin g; ?.%g%+D3. .() [B`c{u*</pre>
00009490 000094A0 000094C0 000094C0 000094D0 000094E0 00009500 00009510 00009520 00009530 00009540 00009550	29 2F 61 4C 63 54 74 AB 2F 52 4C 67 0D 11	5B 6C 76 6A 74 65 22 63 65 63 65 6A 3B 28 D1	42 61 61 2F FB 67 DF 6F 73 61 00 29 65	00 6E 2F 76 4D 08 65 38 6E 6F 76 3F 5B 82	0C 67 61 65 6A 72 3A 74 6C 61 8E 42 E8	52 2F 61 2F 74 61 00 0F 65 76 2F 25 00 D6	EE 53 6E 68 76 EF 28 65 65 67 FB 44	BE 74 67 61 6F 61 E5 29 74 72 61 25 60 B0	2F 72 2F 6E 64 2F C6 4C 2F 3B 6E CC 63 5F	05 69 43 67 3B 6C 80 61 43 00 61 43 00 67 DA 7B A2	28 6E 2F 00 61 99 6E 6F 15 2F 81 93 49	4C 67 61 72 CD 6E 9B 64 6E 53 95 A1 48	6A 3B 73 65 C2 67 94 72 74 5A 74 2B 9B EC	61 5B 73 66 56 2F 1B 6F 65 FA 72 44 C0 0E	76 4C 3B 6C 98 49 6F 69 6E 22 69 33 75 41	61 6A 29 65 9E 6E 24 64 74 05 6E 0F 2A 45	<pre>) [BR/.(Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid /content/Content Resolver;.ZZ.". Ljava/lang/Strin g;?.%g%+D3. .() [B`c{u* eDIHAE</pre>
00009490 000094A0 000094C0 000094C0 000094C0 000094F0 00009500 00009510 00009530 00009530 00009540 00009550 00009560 00009570	29 2F 61 63 54 74 AB 2F 52 4C 67 0D 11 53	5B 6C 76 6A 74 7E 65 22 63 65 63 65 28 28 00	42 61 61 2F FB 67 DF 6F 73 61 00 29 65 21	00 6E 2F 76 4D 08 65 38 6E 6F 76 3F 5B 2 A2	0C 67 6C 61 65 6A 72 3A 74 6C 61 8E 42 E8 E3	52 2F 61 2F 74 61 00 0F 65 76 2F 25 00 D6 C7	EE 53 6E 6C 68 76 EF 28 6E 65 6C 67 FB 44 2F	BE 74 67 61 6F 61 E5 29 74 72 61 25 60 B0 F7	2F 72 2F 6E 64 2F C6 4C 2F 3B 6E CC 63 5F 05	05 69 43 67 3B 6C 80 61 43 00 67 DA 7B A2 28	28 6E 6C 2F 00 61 99 6E 6F 15 2F 81 93 49 4C	4C 67 61 72 CD 6E 9B 64 6E 53 95 A1 48 6A	6A 3B 73 65 C2 67 94 72 74 5A 74 2B 9B EC 61	61 5B 73 66 56 2F 1B 6F 65 FA 72 44 C0 0E 76	76 4C 3B 6C 98 49 6F 69 6E 22 69 33 75 41 61	61 6A 29 65 9E 6E 24 64 74 05 6E 0F 2A 45 2F	<pre>) [BR/.(Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In teger</pre>
00009490 000094A0 000094C0 000094C0 000094D0 000094E0 00009500 00009510 00009520 00009530 00009540 00009550	29 2F 61 63 54 74 AB 2F 52 4C 67 0D 11 53 6C	5B 6C 76 6A 74 65 22 63 65 63 65 6A 3B 28 D1	42 61 61 2F FB 67 6F 73 61 00 29 65 21 6E	00 6E 2F 76 4D 08 65 38 6E 6F 76 3F 5B B2	0C 67 6C 61 65 6A 72 3A 74 6C 61 8E 42 E8 E3 2F	52 2F 61 2F 74 61 00 0F 65 76 2F 25 00 D6 C7 53	EE 53 6E 68 76 EF 28 65 65 67 FB 44	BE 74 67 61 E5 29 74 72 61 25 60 B0 F7 72	2F 72 2F 6E 64 2F C6 4C 2F 3B 6E CC 63 5F	05 69 43 67 3B 6C 80 61 43 00 61 43 00 67 DA 7B A2	28 6E 6C 2F 00 61 99 6E 6F 15 2F 81 93 49 4C 67	4C 67 61 72 CD 6E 9B 64 6E 53 95 A1 48	6A 3B 73 65 C2 67 94 72 74 5A 74 2B 9B EC	61 5B 73 66 56 2F 1B 6F 65 FA 72 44 C0 0E 76 4C	76 4C 3B 6C 98 49 6F 69 6E 22 69 33 75 41	61 6A 29 65 9E 6E 24 64 74 05 6E 0F 2A 45 2F 61	<pre>) [BR/.(Ljava /lang/String; [Lj ava/lang/Class;) Ljava/lang/refle ct/Method;V T~java/lang/In tegero\$.".8:.()Landroid /content/Content Resolver;.ZZ.". Ljava/lang/Strin g;?.%g%+D3. .() [B`c{u* eDIHAE</pre>

Decrypted Contents



Decrypted Contents

	Native Function Name	Native Subroutine Address	Signature	Human-Readable Signature
1	vxeg	0x30D4	([Ljava/lang/Object;)I	<pre>public native int vxeg(Object[] p0);</pre>
2	quaqrd	0x4814	(I)Ljava/lang/String;	<pre>public static native String quaqrd(int p0);</pre>
3	ixkjwu		([Ljava/lang/Object;)Ljava /lang/Object;	<pre>public native Object ixkjwu(Object[] p0);</pre>

The method numbers in the left most column are used to identify the identical method in other samples where the method name is different, but the signature is the same

Run-Time Environment Checks

Goal: Detect if application is being dynamically analyzed, debugged, or emulated

The developers would rather limit the number of potential targets than risk being detected

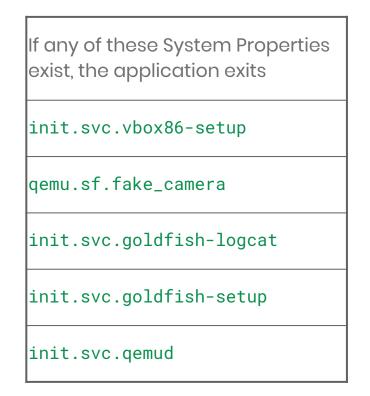
Run-Time Environment Checks

- Function #1 (vxeg) performs the run-time environment checks
- 45+ run-time checks:
 - Checking system properties
 - Verifying CPU architecture by reading the /system/lib/libc.so ELF header
 - Looking for Monkey by iterating through all PIDs in /proc
 - Ensuring the Xposed Framework is not mapped to the application process memory
- If any one of these conditions is detected, the Linux exit(0) function is called, which terminates the Android application

System Property Checks

Goal: Check if system properties show that the "hardware" is an emulator or being debugged

- 37 system properties are checked for specific values
 - Mostly debugging and emulator based
 - All at https://bit.ly/20rgz6l
- 5 system properties are checked for existence



Verifying CPU Architecture

Goal: Ensure the application is running on ARM

- Read 0x14 bytes from /system/lib/libc.so
 - Reading the ELF header
- Verify one of the following conditions is true:

e_ident[EI_CLASS] == 0x01(32-bit) AND e_machine == 0x0028(ARM)
e_ident[EI_CLASS] == 0x02(64-bit) AND e_machine == 0x00B7(AArch64)

Identifying if Monkey is Running

Goal: Determine if application is run in emulator with "fake" user

- "The Monkey is a program that runs on your emulator or device and generates pseudo-random streams of user events such as clicks, touches, or gestures, as well as a number of system-level events."
- Iterates through every PID directory under /proc/ to determine if com.android.commands.monkey is running
 - Note that this no longer works on Android N+

Identifying if Monkey is Running

- 1. Verify d_type from the dirent struct == DT_DIR
- 2. Verify d_name from the dirent struct is an integer
- 3. Construct path strings: /proc/[pid]/comm and /proc/[pid]/cmdline where [pid] is the directory entry name that has been verified to be an integer
- 4. Attempts to read 0x7F bytes from both comm and cmdline constructed path strings
- 5. Stores the data from whichever attempt (comm or cmdline) read more data
- 6. Checks if the read data equals com.android.commands.monkey, meaning that package is running

Current Process not Hooked with Xposed Framework

Goal: Confirm the application is not being analyzed and hooked with the Xposed Framework

- The Xposed Framework allows hooking and modifying of the system code running on an Android device
- Checks if LIBXPOSED_ART.SO or XPOSEDBRIDGE.JAR exist in /proc/self/maps
- Tries to find either of the following two classes using the JNI FindClass() method
 - o XC_MethodHook: de/robv/android/xposed/XC_MethodHook
 - XposedBridge: de/robv/android/xposed/XposedBridge

Summary of WeddingCake Checks



What's happened since I presented at BlackHat?

Recent Modifications

- The ELF is no longer included in the APK statically
 - Dynamically downloaded through a couple different techniques
- Within the ELF, just "cosmetic" changes
 - Moving the decryption initialization constants around
 - Decryption script still works
 - Same run-time environment checks

Conclusion

Conclusion

Malware authors are willing to miss-out on potential targets if that means not being detected

- Layered Anti-Analysis Techniques:
 - Techniques that deter human analysis (anti-RE, decryption)
 - Techniques that prevent dynamic analysis (decryption)
 - Techniques that detect dynamic analysis, debugging, & emulation



@maddiestone github.com/maddiestone/IDAPythonEmbeddedToolkit

