

android

안드로이드 어플리케이션 리버싱 방지 기법

BoB 2기 교육생 박민건(KaiEn)

목차

- 1. 프로젝트 소개
- 2. DEX 파일 구조
- 3. Anti Analysis 기법
- 4. Dalvik VS ART
- 5. DEMO
- 6. Q & A



프로젝트 소개



<u>안드로이드 시큐어코딩 자동화 검증 도구 제작.</u>

목적 : 국내외 앱들의 보안성 및 **안정성을** 높이기 위함.

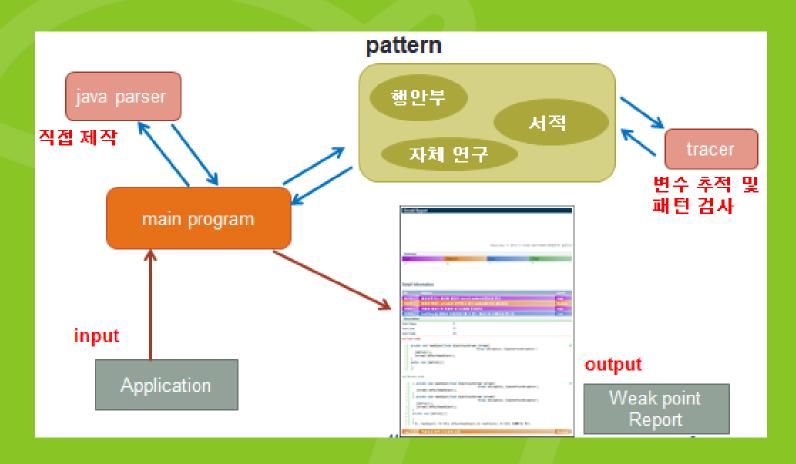
범위: 국내외 모든 안드로이드 앱

예상결과/기대효과:

- **개인 개발자도** 보안성을 갖춘 안전한 안드로이드 앱 개발을 지원.
- 앱의 **안정성까지 고려하여** 점검하기 때문에 사용자들이 더욱 **완성도 높** 은 앱을 사용할 수 있음.
- **현재 무료로** 제공되는 소스코드 자동화 **진단 도구가 없으므로** 이에 대한 연구 및 개발에 의미
- **역공학 방지 솔루션이 도입**되어 악의적인 사용자의 **역공학 분석을 방어** 할 수 있음.



프로그램 Flowchart



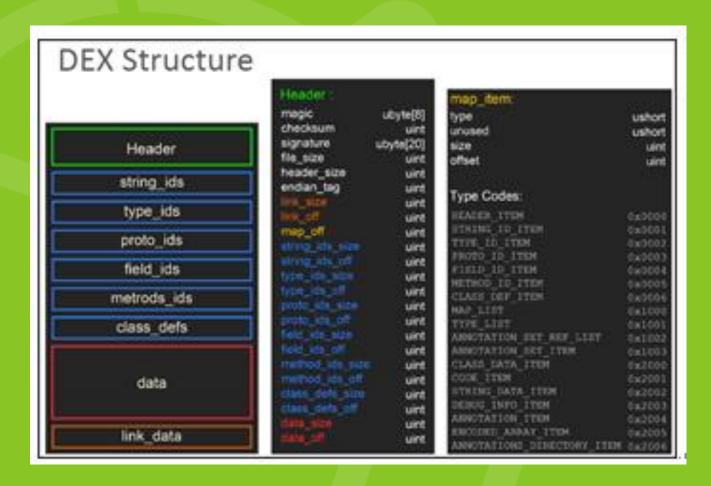


버그패턴 연구 및 수집

번호	내용
ORR61	KeyguardManager.KeyguardLock 사용
ORR63	Android.provider.contacts 클래스 사용
ORR65	listenUsingInsecureRtcommWithServiceRecord 함수 사용
ORR66	getinsecure 함수 사용
ORR67	createInsecureRfcommSocketToServiceRecord 함수 사용
ORR68	SecureRandom의 디폴트 생성자를 사용하지 않는다.
ORR70	Camera 사용 완료 후 Camera object를 릴리즈하지 않는다.
ORR71	Camera.open() 사용 시 예외를 검사하지 않는다.
ORR72	Display 클래스의 비권장 함수 사용
ORR74	암시적 인텐트 사용
ORR76	setJavaScriptEnabled() 함수 호출
ORR77	dearView() 함수 사용
ORR78	findAll 함수 사용
ORR79	PrivilegedAction, PrivilegedExceptionAction함수 사용
ORR83	canZoomIn(), canZoomOut(), getScale() 함수 사용
ORR84	freeMemory()
ORR85	onGlobalFocusChanged 함수 사용
ORR86	savePassword 함수 사용
ORR88 ORR89	setCertificate 함수 사용
	showFindDialog 함수 사용
ORR91 ORR92	capturePicture() 함수 사용
ORR94	onChildViewAdded, onChildViewRemoved함수 사용 com.google.android.gcm에서 사용이 장려되지 않는 classes 사용
ORR95	Android.util.config 클래스의 상수 값 사용
ORR96	Android.util.comig 클래스의 영우 값 사용 Android.app.notification의 FLAG_HIGH_PRIORITY 상수 사용
ORR100	Thread.suspend 함수 사용
ORR101	TextToSpeech.OnUtteranceCompletedListener 인터페이스 사용
ORR103	android.app.TabActivity클래스 사용
ORR104	android.telephony.PhoneStateListener.LISTEN SIGNAL STRENGTH 상수 사용
ORR106	android.provider.Settings.System.ADB_ENABLED_ALB
ORR107	android.provider.Settings.System.AIRPLANE_MODE_ON 사용
ORR108	android.provider.Settings.System.AIRPLANE MODE RADIOS ALB
ORR109	android.provider.Settings.System.ALWAYS FINISH ACTIVITIES 사용
ORR110	android.providerSettings.SystemANDROID ID 사용
ORR111	android.provider.Settings.System.ANIMATOR_DURATION_SCALE 사용
ORR112	android.provider.Settings.System.AUTO_TIME 사용
ORR113	android.provider.Settings.System.AUTO TIME ZONE 사용
ORR114	android.provider.Settings.System.BLUETOOTH_ON 사용



안드로이드 역공학 방지 기법 연구





프로젝트 결과물



Summary								
High	Medum	Low	Total					
9	12	12	33					
Detail Information								
No	Subject			Level				
				and the same of th				
ORREIT	Information Leak of System Data			Low Medium				
ORIRITS	Declare data members as private and provide accessible wrapper methods							
06930	Relative Path Traversal							
ORR49	Detection of Error Condition Without Action							
06932	Object Model Violation: Just one of equals() and hashCode() Defined							
OREGES	Use of Insufficiently Random Values							
000006	Files under Global Access			Medium				
CHORAG	Use of a throken or Riscky Cryptographic Alg	orithm		Medium				
ORR13	Do not use deprecated or obsolete classes of	e methods		High				
OKSR21	Do not use public static nonfinal variables			Modium				
ORR22	Enable serialization compatibility during cla	ass evolution		Low				
OR923	Do not deviate from the proper signatures of serialization methods							
ORREN	Do not invoke overridable methods from the	readObject[] method		Low				

안드로이드 어플리케이션 역공학 방지 기술 분석

확민건*, 오정희**, 류환실*** 김수민**** *인하대학교 접퓨터정보공하과 **부산대학교 정보전류터공학부 **경찰대학 법학과 ***경찰대학 법학과 ***경찰대학 법학과 ***경찰대학 법학과

An analysis on Technics for Preventing Android Reverse Engineering

Min-Gun Pak *, Jeoughui Oh **, Hwahn-Il Lyoo ***, Soo Min Kim ****
** Dept. of Computer Science & Information Technology, Inha University
** Dept. of Computer Science & Engineering, Pusua N'atonal University
*** Dept. of Law, Korean National Police University
**** Dept. of Mathematical Sciences, Korea Advanced Institute of Science and Technology

요 약

1. 서론

스마트폰 시대가 도래함에 따라 일상의 여러 일등 이 스마트폰을 통하여 이루어기고 있다. 스마트폰에 는 다양한 어름리케이션들이 설치되어 있으며, 사용 자가 원하고 필요되 하는 어플리케이션도 손쉽게 설 치하여 이용할 수가 있다. 그래서 사망들은 스마트폰

전도 침해될 수 있다. 이 같은 내용들은 그저 가정이 아니라 현실에서 현저하게 나타나는 일들이다. 그러 므로 우리는 어떻게 대저하고 해결할 것인지를 고민 해야 한다.

본 논문에서는 안드로이드 어플리케이션 역공학을 방지하기 위한 기법들을 설명한다. 역공학을 방지하기







DEX Structure

Header

string_ids

type_ids

proto_ids

field_ids

metrods_ids

class defs

data

link data

Header:

magic ubyte[8] checksum uint signature ubyte[20] file size uint header_size uint endian_tag uint uint uint map off uint uint

uint

uint

map_ite<u>m:</u>

type ushort unused ushort size uint offset uint

Type Codes:

 HEADER_ITEM
 0x0000

 STRING_ID_ITEM
 0x0001

 TYPE_ID_ITEM
 0x0003

 PROTO_ID_ITEM
 0x0004

 METHOD_ID_ITEM
 0x0005

 CLASS_DEF_ITEM
 0x0006

 MAP_LIST
 0x1000

 TYPE_LIST
 0x1001

 ANNOTATION_SET_REF_LIST
 0x1002

 ANNOTATION_SET_ITEM
 0x2000

 CCLASS_DATA_ITEM
 0x2000

 STRING_DATA_ITEM
 0x2002

 DEBUG_INFO_ITEM
 0x2003

 ANNOTATION_ITEM
 0x2004

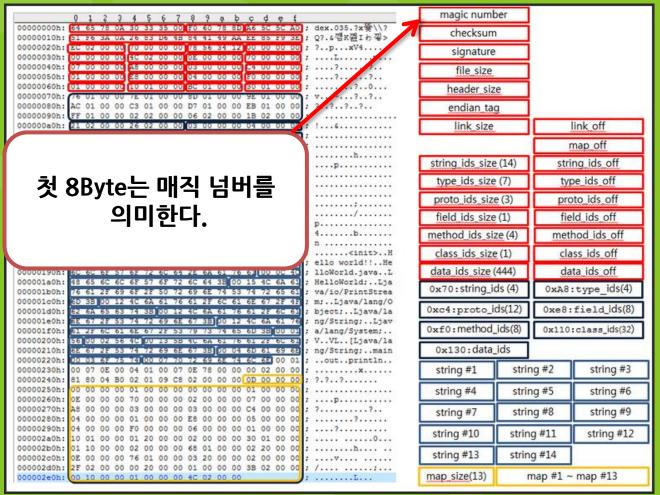
 ENCODED_ARRAY_ITEM
 0x2005

 ANNOTATIONS_DIRECTORY_ITEM
 0x2006

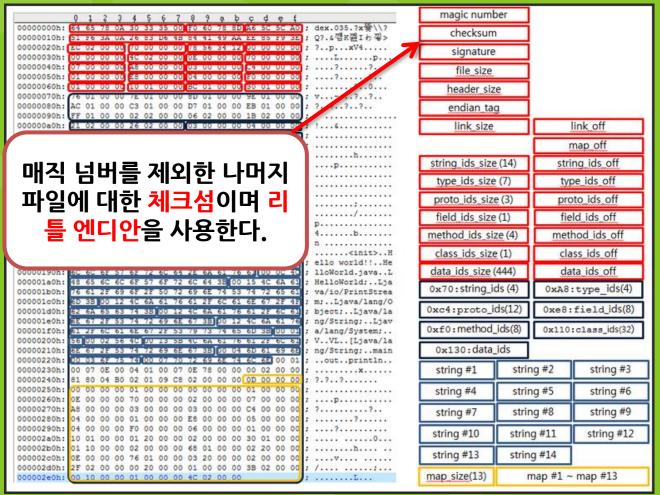


0 1 2 3 4 5 6 7 8 9 a b c d e f	magic numb	er		
00000000h: 64 65 78 0A 30 33 35 00 F0 60 78 8D A6 5C 5C A0 ; dex.035.?x衰\\?	checksum			
00000010h: SI F6 3A UA 26 83 D6 4B 84 41 49 AA EE 85 F9 3E ; Q?.6깸K꼞Ih낗> 00000020h: EC 02 00 00 70 00 00 00 78 56 34 12 00 00 00 00 ; ?.pxV4		_		
00000030h: 00 00 00 00 4C 02 00 00 0E 00 00 00 70 00 00 0 ;Lp	signature			
00000040h: 07 00 00 00 08 00 00 00 03 00 00 00 00 00 ;??	file size			
00000050h: 01 00 00 00 E8 00 00 00 04 00 00 00 F0 00 00 00 ;??	JANUAR PROPERTY.	_		
00000060h: 01 00 00 00 10 01 00 00 BC 01 00 00 30 01 00 00 ;?0 00000070h: 78 01 00 00 72 01 00 00 80 01 00 00 92 01 00 00 ; v~??	header_siz	e		
00000080h: AC 01 00 00 C3 01 00 00 D7 01 00 00 EB 01 00 00; ???	endian tag			
00000090h: FF 01 00 00 02 02 00 00 06 02 00 00 1B 02 00 00 ;				
000000a0h: 21 02 00 00 26 02 00 00 03 00 00 00 00 00 00 00 ; !	link size		link off	
000000b0h: 05 00 00 00 06 00 00 00 07 00 00 00 00 00 00 00;		_=		
000000c0h: 0A 00 00 00 08 00 00 00 00 00 00 00 00 00			map_off	
000000e0h: 05 00 00 00 70 01 00 00 04 00 01 00 00 00 00 00;p	string ids size	(14)	string ids off	
000000f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00				
00000100h: 01 00 01 00 0D 00 00 00 02 00 00 00 00 00 00 00;	type ids size (7)		type_ids_off	
00000110h: 00 00 00 01 00 00 00 02 00 00 00 00 00 00;	proto ids size (3)		proto ids off	
00000120h: 02 00 00 00 00 00 00 03B 02 00 00 00 00 00 00;	Accordant Constant Co		hannandhann dhann	
00000130h: 01 00 01 00 01 00 00 02 02 00 00 04 00 00 05;/	field_ids_size (1)		field_ids_off	
00000150h: 34 02 00 00 08 00 00 00 62 00 00 00 1A 01 01 00 : 4b	method ids size (4)		nethod ids off	
00000160h: 6E 20 02 00 10 00 0E 00 01 00 00 03 00 00 00; n	method_ids_size (4)		ictilod_ids_off	
00000170h: 01 00 00 00 06 00 06 30 69 6E 69 74 3B 00 0D 49 ; <init>H</init>	class ids size (1)		class ids off	
00000180h: 65 6C 6C 6F 20 77 6F 72 6C 64 21 21 00 0F 48 65 ; ello world!!He			The second secon	
00000190h: 6C 6C 6F 57 6F 72 6C 64 2E 6A 61 76 6 00 0C 40 ; lloWorld.javaL	data_ids_size (444)		data_ids_off	
000001a0h: 48 65 6C 6C 6F 57 6F 72 6C 64 3B 00 15 4C 6A 61 ; MelloWorld;Lja 000001b0h: 76 61 2F 69 6F 2F 50 72 69 6E 74 53 74 72 65 61 ; va/io/PrintStrea	0x70:string_ids (4)		A8:type ids(4)	
00000160h: 76 61 2F 69 6F 2F 50 72 69 62 74 53 74 72 65 61; Va/16/FFIREStrea			omic.olbe_co()	
000001d0h: 62 6A 65 63 74 35 00 12 4C 6A 61 76 61 2F 6C 61; bject;Ljava/la	0xc4:proto_ids(12)		0xe8:field_ids(8)	
000001e0h; SE 67 2F 53 74 72 69 6E 67 3B 00 12 4C 6A 61 76; ng/String:Ljav				
000001f0h: 61 2F 6C 61 6E 67 2F 53 79 73 74 65 6D 3B 00 01; a/lang/System;	0xf0:method_ids(8)		0x110:class_ids(32)	
00000200h: 56 00 02 56 4C DU 13 5B 4C 6A 61 /6 61 2F 6C 6B; VVL[Ljava/la	0 400 dete	tal.		
00000210h: 6E 67 2F 53 74 72 69 6E 67 3B 00 04 6D 61 69 6B; ng/String;main 00000220h: 00 03 6F 75 74 00 07 70 72 69 6F 74 6C 6F 00 01;outprintln	0x130:data_	ias		
00000220h: 00 03 6F 75 74 00 07 70 72 69 6E 74 6C 6E 00 01;outprintln	string #1	string #2	string #3	
00000240h; 81 80 04 B0 02 01 09 C8 02 00 00 00 00 00 00; ?.??	String #1	Juling #2	samy "5	
00000250h: 00 00 00 00 01 00 00 00 00 00 00 01 00 00	string #4	string #5	string #6	
00000260h: 0E 00 00 00 70 00 00 00 02 00 00 07 00 00 00 ;p	Sumg - 1	Junia	samg #0	
00000270h: A8 00 00 00 03 00 00 03 00 00 00 C4 00 00 00; ?	string #7	string #8	string #9	
00000280h: 04 00 00 00 01 00 00 00 E8 00 00 05 00 00 00 ;				
00000290h: 04 00 00 00 F0 00 00 00 06 00 00 01 00 00 00;?	string #10	string #11	string #12	
000002a0h: 10 01 00 00 01 20 00 00 02 00 00 03 01 00 00;				
00000250h: 0E 00 00 00 76 01 00 00 03 20 00 00 02 20 00 00;v	string #13	string #14		
000002d0h: 2F 02 00 00 00 20 00 00 01 00 00 00 3B 02 00 00; /		-		
000002e0h: 00 10 00 00 01 00 00 00 4C 02 00 00 ;L	map_size(13)	map #	map #1 ~ map #13	

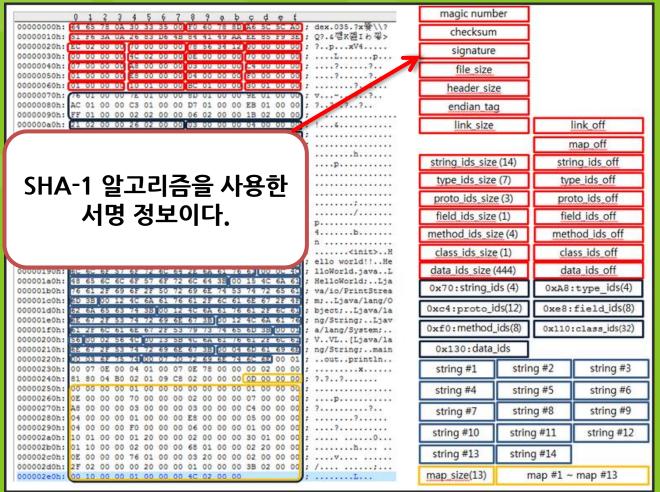












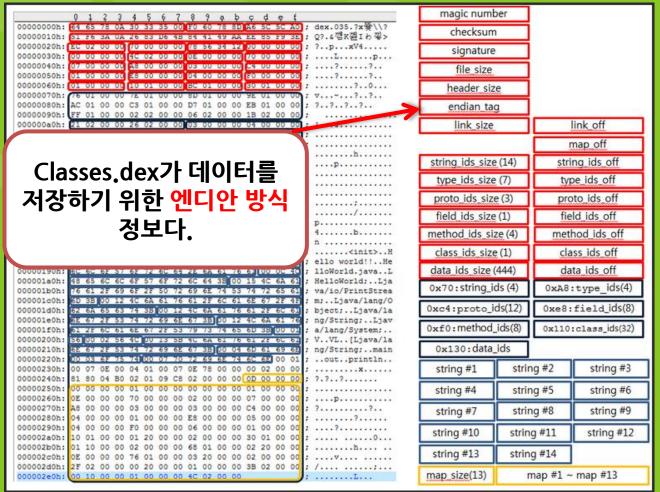




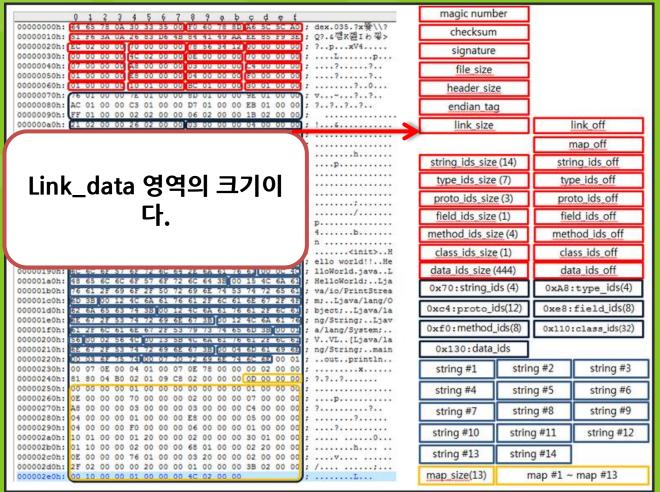




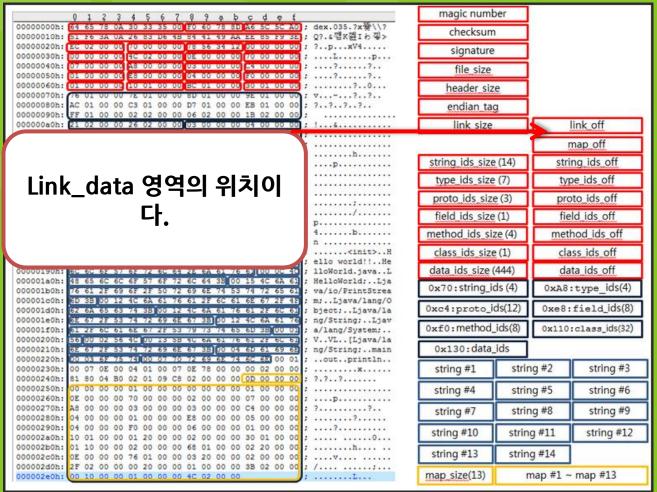




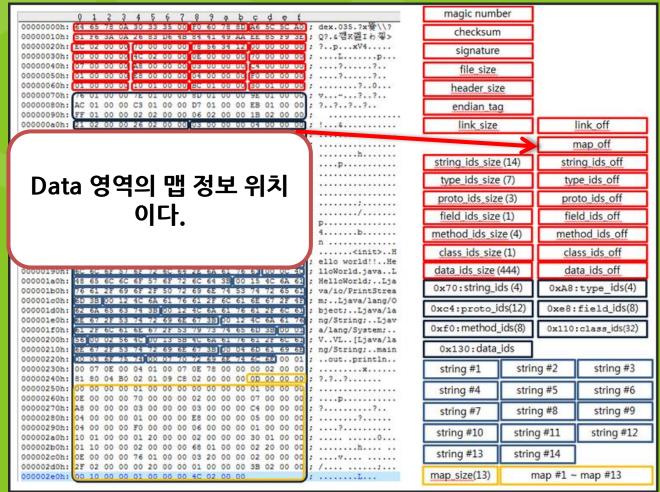








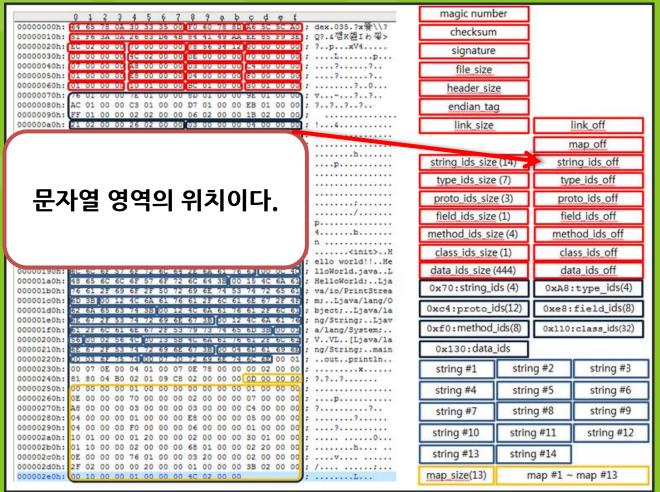




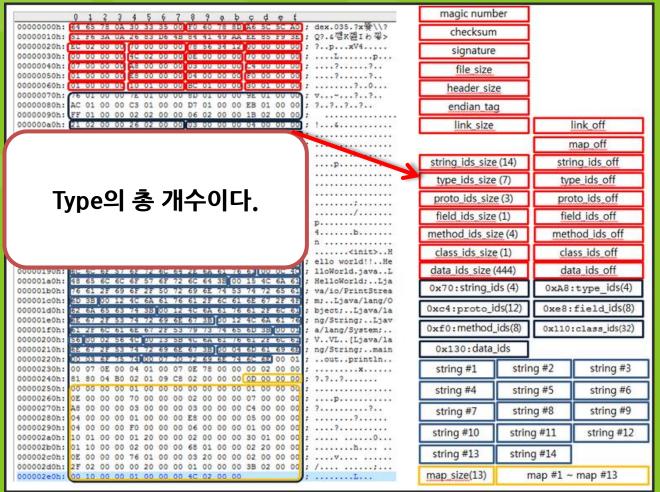








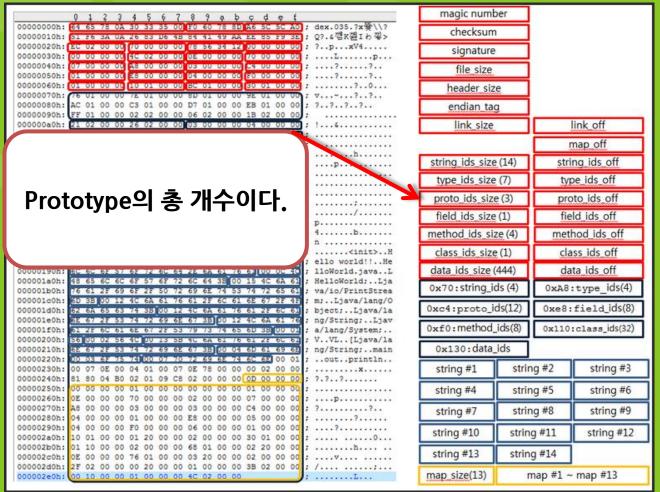




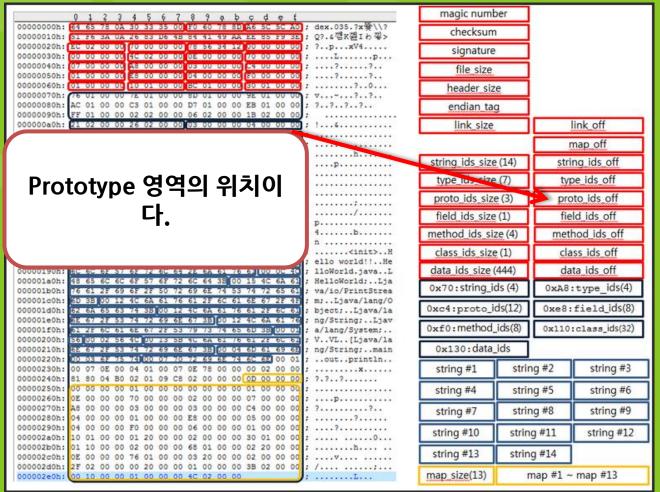








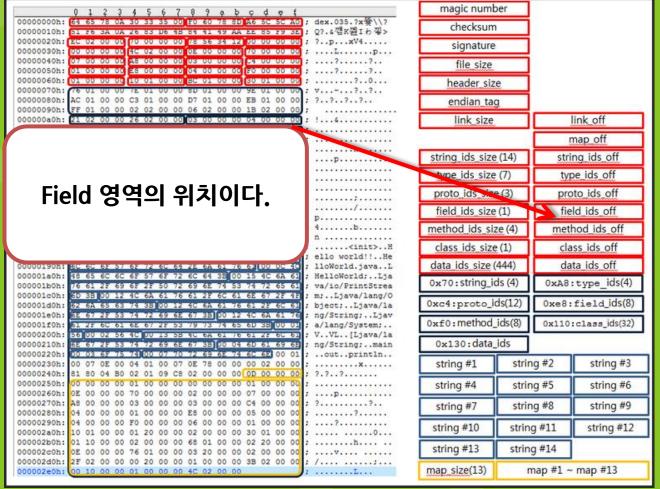




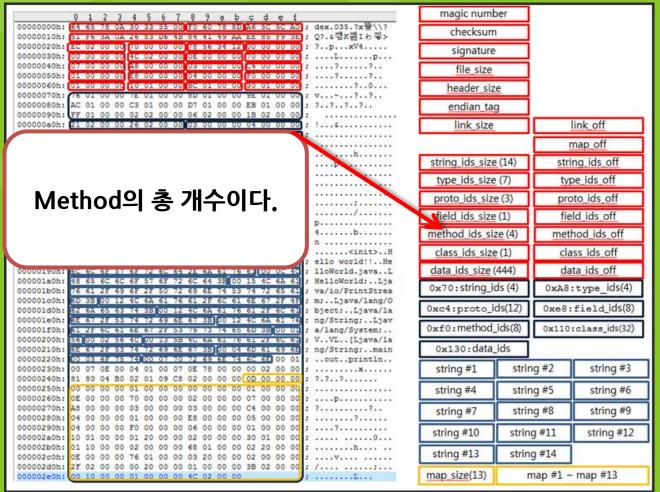




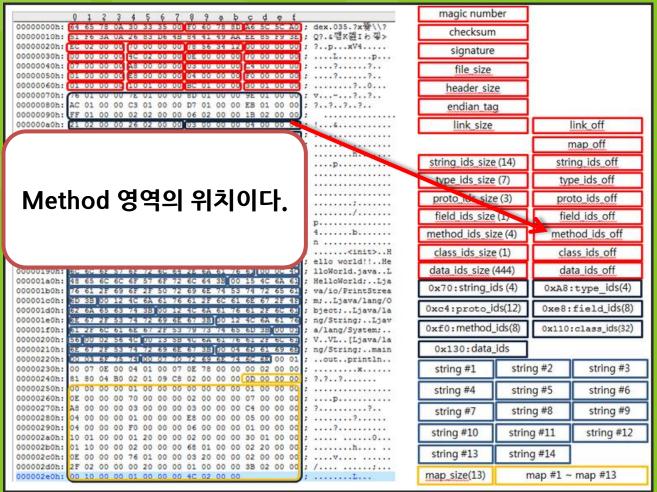








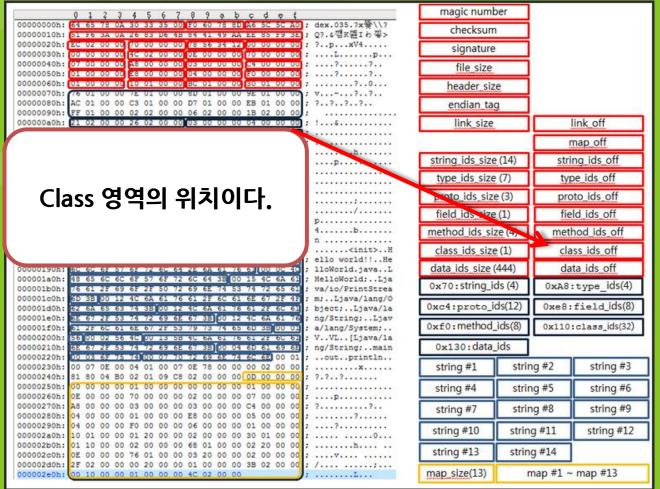




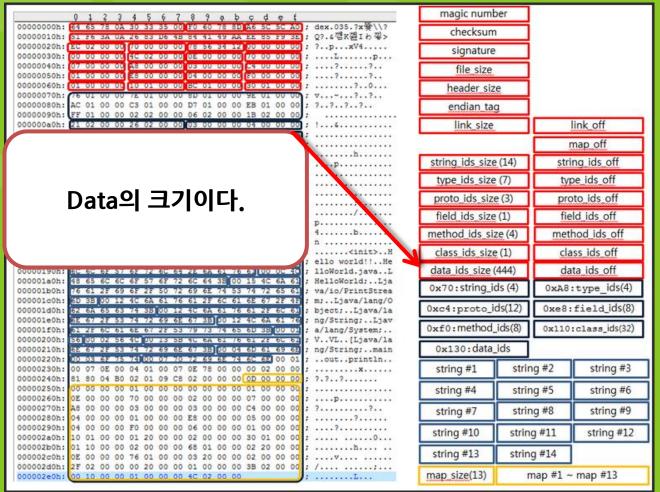




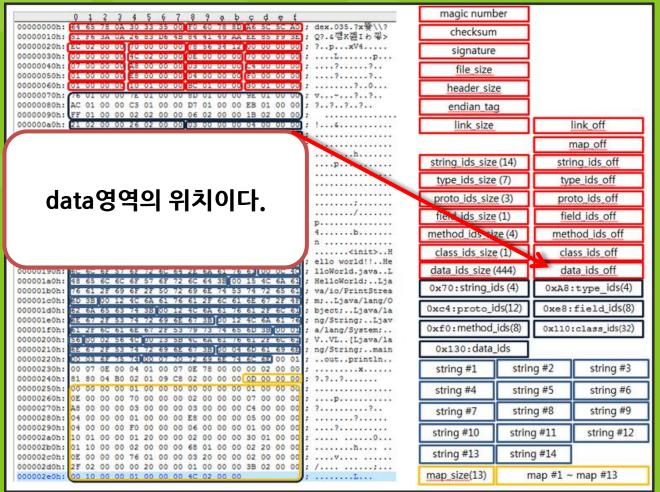








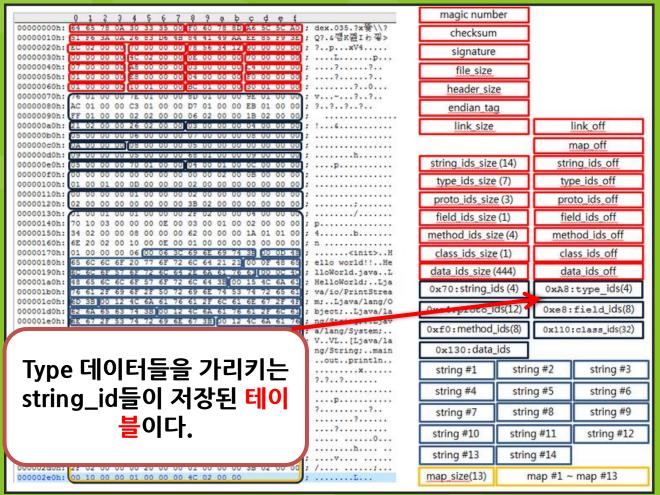




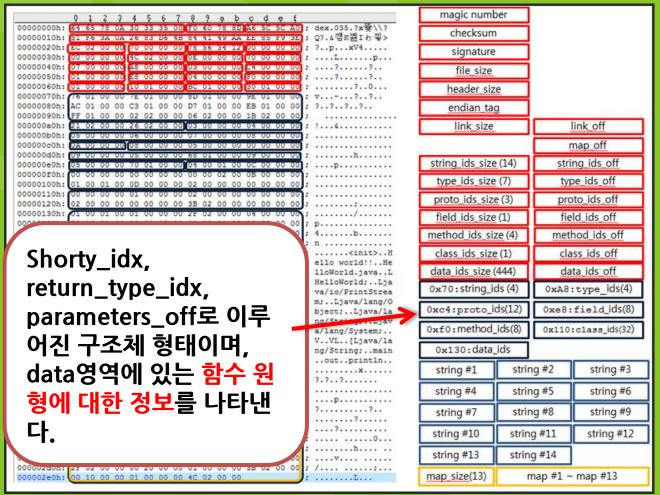




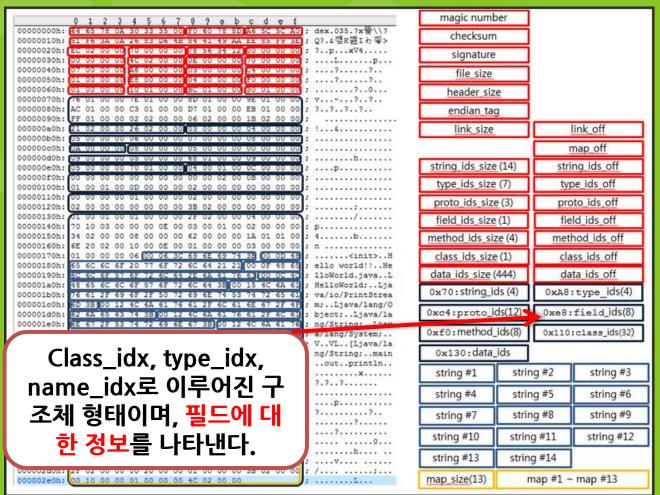




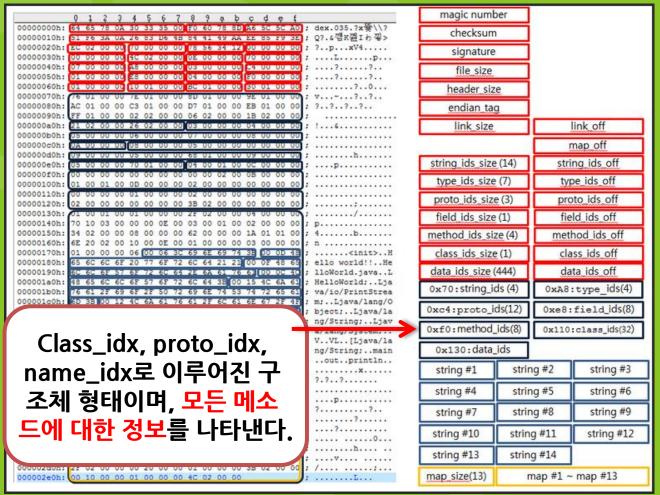




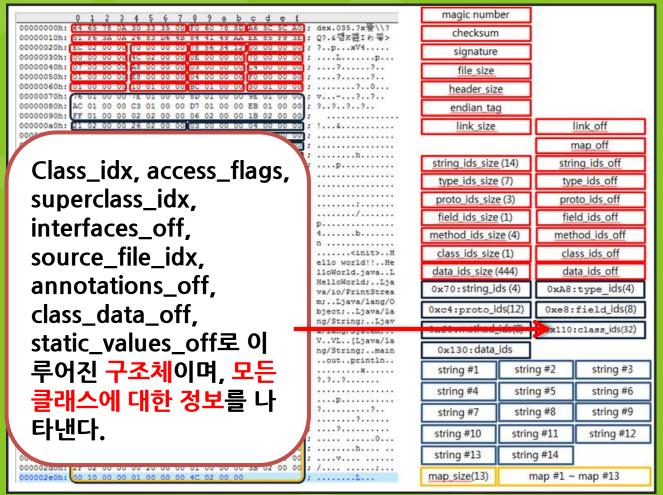




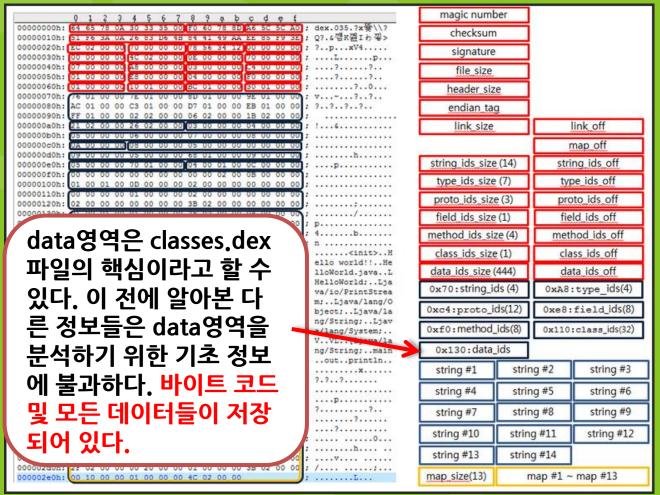








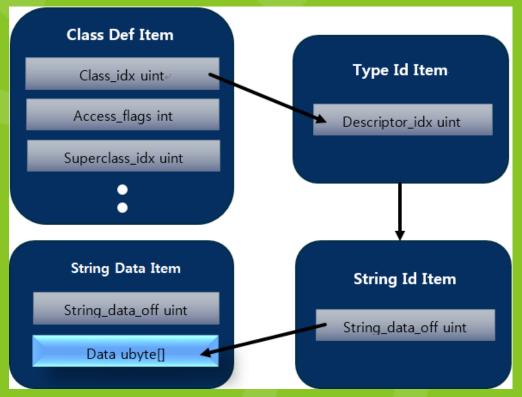






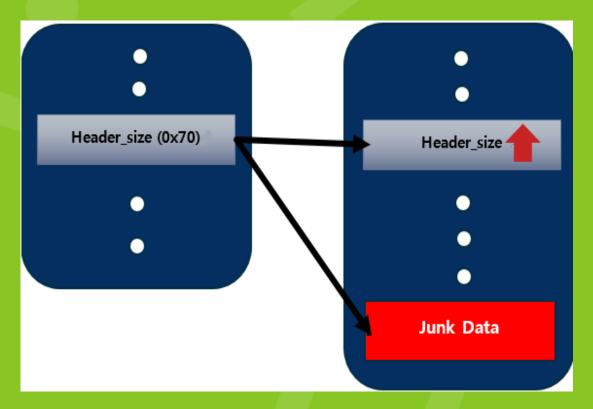


• 클래스명 길이 변경





• 헤더 크기 변경



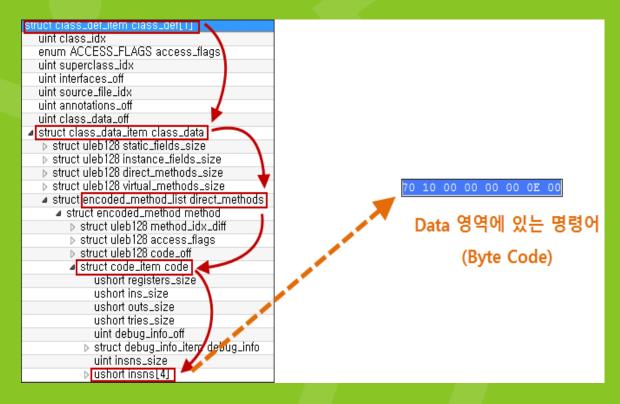


• Invalid OP 코드 삽입

```
public interface Opcodes {
     * IMPORTANT NOTE: The contents of this file are mostly generated
    * automatically by the opcode-gen tool, Any edits to the generated
    * sections will get wiped out the next time the tool is run,
    // BEGIN(libcore-opcodes): GENERATED AUTOMATICALLY BY opcode-gen
    int OP_NOP
                                         = 0x00000;
    int OP_MOVE
                                         = 0x0001;
    int OP_MOVE_FROM16
                                         = 0x0002;
    int OP_MOVE_16
                                         = 0x0003;
    int OP_MOVE_WIDE
                                         = 0x0004;
    int OP_MOVE_WIDE_FROM16
                                         = 0x0005;
    int OP_MOVE_WIDE_16
                                         = 0x0006;
    int OP_MOVE_OBJECT
                                         = 0x0007;
    int OP_MOVE_OBJECT_FROM16
                                         = 0x0008;
    int OP_MOVE_OBJECT_16
                                         = 0x0009;
    int OP_MOVE_RESULT
                                         = 0x000a;
    int OP_MOVE_RESULT_WIDE
                                         = 0x000b;
    int OP_MOVE_RESULT_OBJECT
                                         = 0x000c;
    int OP_MOVE_EXCEPTION
                                         = 0x000d;
    int OP_RETURN_VOID
                                         = 0x000e;
    int OP_RETURN
                                         = 0 \times 0000 f;
```



• Invalid OP 코드 삽입





• Invalid OP 코드 삽입

12 01 38 01 03 00 FE FE

12 01₽	V1(Register) = 0₽
38 01 03 00₽	If V1 == 0, (항상 TRUE)↔ jump to current offset + 0x0003↔
FE FE	Dummy code₽



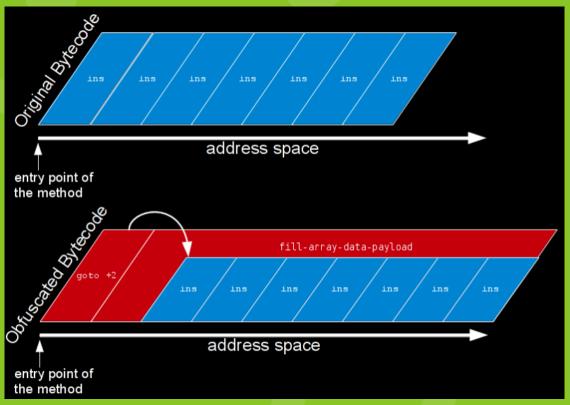
• Invalid Offset 참조

12	01	38	01	03	00	1A	00	70	17	12	01	38	01	03	00
1A	00	71	12	12	01	38	01	03	00	1A	00	70	17	00	00

12 01₽	V1(Register) = 0₽
38 01 03 00₽	If V1 == 0, (항상 TRUE)↔ jump to current offset + 0x0003↔
1A 00 [2h]₽	Const String을 가리키는 String_id (= [2h])↔ 를 V0(=this)에 집어넣는다.↔



Using Fill-array-data-payload





• 난독화

- Data명 조작
- 연산 난독화
- 코드 집적
- 재배치
- 저장소 및 인코딩
- 데이터 집적
- 재배치



• 난독화 도구 - proguard

```
com.android
                       CCOIoll.class
                                   CCIIOcc.class ×
internal.telephony
                       package com.android.system.admin;
🚊 🖶 system.admin
  ⊞ CCOIoll
                        final class CC1IOcc
  ⊞ CCIIOcc
                         implements Runnable
  private static final byte[] oIlclcIc
  CC1IOcc(CICoICCo paramCICoICCo)
    COOLOII
  CcOCoIcO
  private static String oCIIC11 int para

    CoccOIo

     CoooOIIO
                           byte[] arrayOfByte1 = oIlclcIc;
```



• 난독화 도구 - dexguard



• ZIP Format Header 조작

Offset	Bytes	Description
0	4	Signature (= 0x02014b50)
4	2	Version made by
6	2	Version needed to extract
8	2	General purpose bit flag
10	2	Compressed method
12	2	File last modification time

General Purpose Bit Flag: Bit 00: encrypted file Bit 01: compression option Bit 02: compression option Bit 03: data descriptor Bit 04: enhanced deflation Bit 05: compressed patched data Bit 06: strong encryption Bit 07-10: unused Bit 11: language encoding Bit 12: reserved Bit 13: mask header values Bit 14-15: reserved



• ZIP Format Header 조작

																â#)>šâ‡RÜHōŸ <mark>PK</mark>
14	00	14	00	09	08	80	00	OA	AC	28	44	14	9F	4E	70	
6D	02	00	00	54	06	00	00	13	00	08	00	00	00	00	0.0	mT
01	00	20	00	00	00	00	00	00	00	41	6E	64	72	6F	69	Androi
64	4D	61	6E	69	66	65	73	74	2E	78	6D	60	7A	E5	04	Androi dManifest.xml <mark>zå.</mark>

파일당	# 5
 M€	TA-INF
🎉 res	<u> </u>
An	droidManifest.xml*
cla	sses.dex*
res	ources.arsc



• Smali 조작

```
methods
rotected onCreate Landroid/og/Bundle;)V
s 4
neter "savedInstanceState"
Full path를 보여준다.

ogue
14
r-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/og/Bundle;)V

15
high16 v1, 0x7f03
r-virtual {p0, v1}, Lcom/example/library_making/LibActivity;->setContentView(I)V

17
rstance v0, Lcom/example/library/MainActivity;
string v1, "testing"
string v2, "http"
r-direct {v0, v1, v2}, Lcom/example/library/MainActivity;-><init>(Ljava/lang/String;Ljava/lang/String;)V
```



• Smali 조작

```
if-nez v4, :cond_0
move v2, p2
move v3, p0
:goto 0
add-int/lit8 p0, p0, 0x1
add-int/2addr v2, v3
add-int/lit8 p1, v2, -0x2
:cond 0
int-to-byte v2, p1
aput-byte v2, v1, v5
add-int/lit8 v5, v5, 0x1
if-lt v5, p2, :cond_1
const/4 v2, 0x0
invoke-direct {v0, v1, v2}, Liava lang/String; -><init>([BI)V
return-object v0
:cond_1
move v2, p1
                            Make Decompilers
aget-byte v3, v4, p0
                                 Confused
```



Exception Recursion

```
public class Zzzzzzzzzzzzzzzzzzzz extends Activity
   private String download throws MalformedURLException{
       Toast.makeText(getApplicationContext(), "HI", Toast.LENGTH LONG);
        String ccc = "abc";
        String ddd = "dce";
        return ccc + ddd:
   protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try{
            String a = download();
        catch (MalformedURLException e)
            System.out.println(e);
```



Exception Recursion

```
.method public ImNeverUsed()V
    registers 4.
   # Exception recursion of doom!
    new-instance v0, Ljava/lang/RuntimeException;
   invoke-direct {v0}, Ljava/lang/RuntimeException;-><in t>()V
   throw v0 # Thro
    .catch_Ljava/lang/Exception; {:try_start_8 .. :try_end_6} :catch_6
      -instance v0, Ljava/lang/RuntimeException;
   invoke-direct {v0}, Ljava/lang/RuntimeException:-
    try end c
    catch Ljava/leng/Exception: (:try_start_6 ...try_end_c} :catch_0
   const-string v0, "useless"
   new-instance v1, Ljava/lang/StringBuilder;
   invoke-static {v0}, Ljava/lang/String;->valueOf(Ljava/lang/Object;)Ljava/lang/String;
   invoke-direct {v1, v2}, Ljava/lang/StringBuilder;-><init>(Ljava/lang/String;)V
   const-string v2, "and stuff"
   invoke-virtual {v1, v2}, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;
   invoke-virtual {v1}, Ljava/lang/StringBuilder;->toString()Ljava/lang/String;
end method
```



Exception Recursion

```
v0, <t: RuntimeException>
                new-instance
                                                 ∠v0}, <void RuntimeException.<init>() imp. @ def RuntimeException
                invoke-direct
                throw
# tru 0x68C-0x698:
# catch Exception:
                                                 v0, √t: RuntimeException>
                new-instance
                                                 {νθ∕, γυοίd RuntimeException.<init>() imp. @ def RuntimeException
                invoke-direct
                throw
                .byte 0x1A
                 .byte 0x22 # "
                 .byte
                 .byte 0x19
                .byte 0x7/ # q
                .bate 0x10
                 .byte
```



• Endian_Tag 변경

```
/** API level to target in order to suppress extended opcode usage */
public static final int API_NO_EXTENDED_OPCODES = 13;
* file name of the primary (@code .dex) file inside an
* application or library {@code .jar} file
public static final String DEX IN JAR NAME = "classes.dex";
/** common prefix for all dex file "magic numbers" */
public static final String MAGIC PREFIX = "dex\n";
/** common suffix for all dex file "magic numbers" */
public static final String MAGIC SUFFIX = "\0";
/** dex file version number for the current format variant */
public static final String VERSION CURRENT = "036";
/** dex file version number for API level 13 and earlier */
public static final String VERSION FOR API 13 = "035";
* value used to indicate endianness of file contents
public static final int ENDIAN_TAG = 0x12345678;
* Maximum addressable field or method index.
* The largest addressable member is Oxffff, in the "instruction formats" spec as field@CCCC or
public static final int MAX_MEMBER_IDX = 0xFFFF;
* Maximum addressable type index.
 * The largest addressable type is 0xfffff, in the "instruction formats" spec as type@CCCC.
public static final int MAX TYPE IDX = 0xFFFF;
```

Header: magic ubyte[8] checksum uint signature ubyte[20] file size uint header size uint endian tag uint link size uint link off uint map off uint uint uint type ids size uint type ids off uint uint proto ids off uint uint uint method ids size uint method ids off uint uint uint uint uint



Fake Dex

```
C:\WINDOWS\system32\cmd.exe - \square \times \chinq C:\WIsers\Hwahn\Desktop\test>\attrib +s +h classes.dex
```

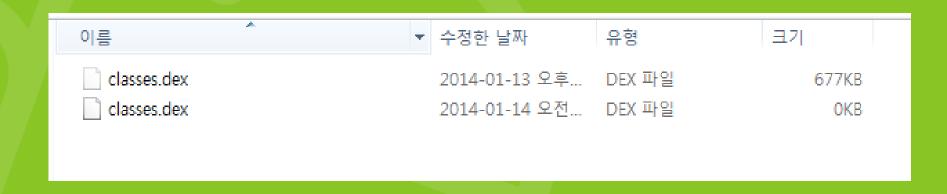


Fake Dex

a)										문기	计丑							-	-		×
글꼴	글꼴(F): O Adobe Arabic V														~		도움말(H)				
	÷	‡	-	M	M			M	M							M	%	M		"	^
	M	M	M	M	M	€	>	*	!!	M	_	M	M	M	M	M	/	M	M	M	
	M	M	M	M	M	N	107	101	107	M	M	M	M	M	M	M	M	M	M	M	
	M	M	M	M	M					M	M	M	M	M							
	M	M	M	M	4	12				M	+	-	M	M	×	n	M	1	2	3	
	4	M	M	M	M	M	+	-	M	M	M	M	M	M	M	M	M	M	M	M	
	M	M	M	M	M	M	M	M	M	M	M	₹`	€	M	M	₽	M	₩	M	M	
	€	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	
	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	~
	복사할 문자(A): 선택(S) 복사(C))								
문자 집합(T): 유니코드 > 유니										유니	크드	로	이동	(U):			20	20			
분류	방남	≝(G)	Ξ		5	무						~									
검색														원래	대로	(R)					
U+2	063:	Inv	isible	e Se	para	tor															



Fake Dex

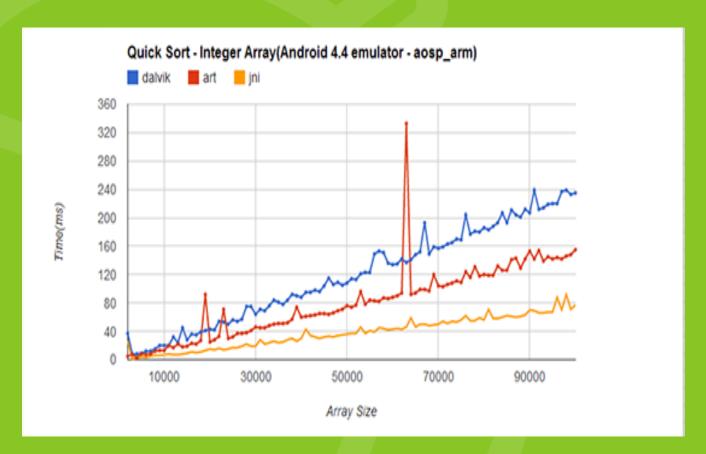


















DEMO



Q&A

