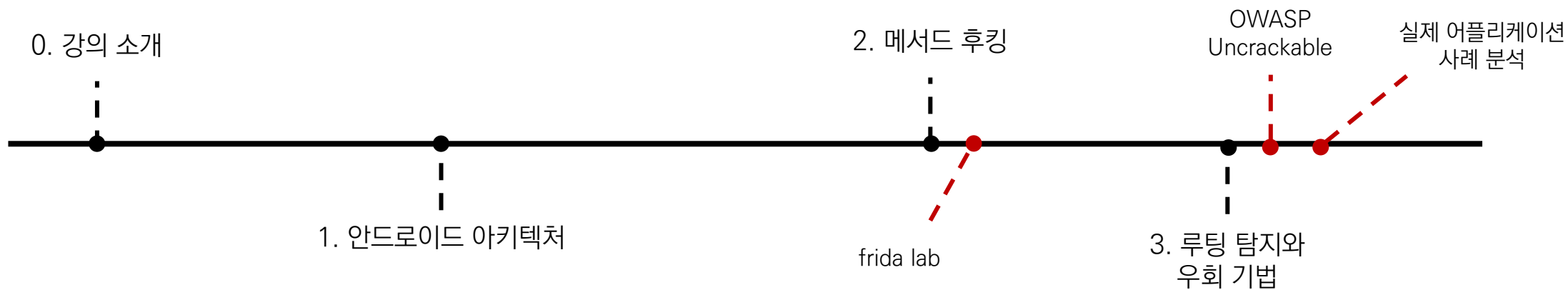


2020 안드로이드 앱 해킹 교육

PART 1. 교육 소개 및 메서드 후킹 기법

금일 교육 강의 구성

강의 타임라인



2020 안드로이드 앱 해킹 교육

강의 소개

일시 : 8/10~8/23 (화, 목) | 오후 7시 ~ 오후 9시 | 공5410, **공5412**

교육자 : 18 박민

주 교육 내용 : 안드로이드 리버싱과 후킹 테크닉을 이용한 어플리케이션 해킹

기본 지식 : java, javascript

* 몇몇 실습 및 과제는 실제 서비스 되는 앱을 대상으로 진행하므로 해당 부분 유튜브 영상 제공 불가능

0 회차 (교육전) : Nox, frida, adb, jadx, ida 설치

1 회차 : 메서드 후킹과 루팅 탐지 우회 기법

2 회차 : 객체(instance) 후킹과 메서드 강제 호출

3 회차 : JNI 소개 및 네이티브 함수 후킹

4 회차 : JNI 후킹을 이용한 게임 해킹 기법

강의구성



강의 소개

후킹(Hooking)?

운영 체제나 응용 소프트웨어 등의 각종 컴퓨터 프로그램에서 소프트웨어 구성 요소 간에 발생하는 함수 호출, 메시지, 이벤트 등을 **중간에서 바꾸거나 가로채는 명령, 방법, 기술이나 행위**를 말한다.

〈냉장고에서 우유를 꺼내 먹는 프로세스 예시〉

냉장고 문을 연다 -> ~~우유를 꺼낸다~~ -> 뚜껑을 연다 -> 마신다.



hook

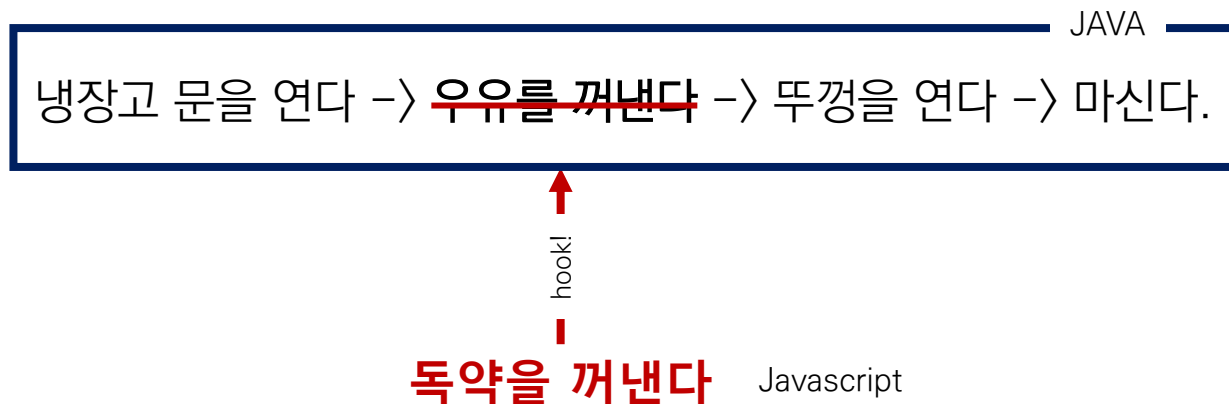
독약을 꺼낸다

강의 소개

후킹(Hooking)?

운영 체제나 응용 소프트웨어 등의 각종 컴퓨터 프로그램에서 소프트웨어 구성 요소 간에 발생하는 함수 호출, 메시지, 이벤트 등을 **중간에서 바꾸거나 가로채는 명령, 방법, 기술이나 행위**를 말한다.

〈냉장고에서 우유를 꺼내 먹는 프로세스 예시〉



안드로이드 해킹이니깐 안드로이드를 알아보자!

안드로이드 아키텍처

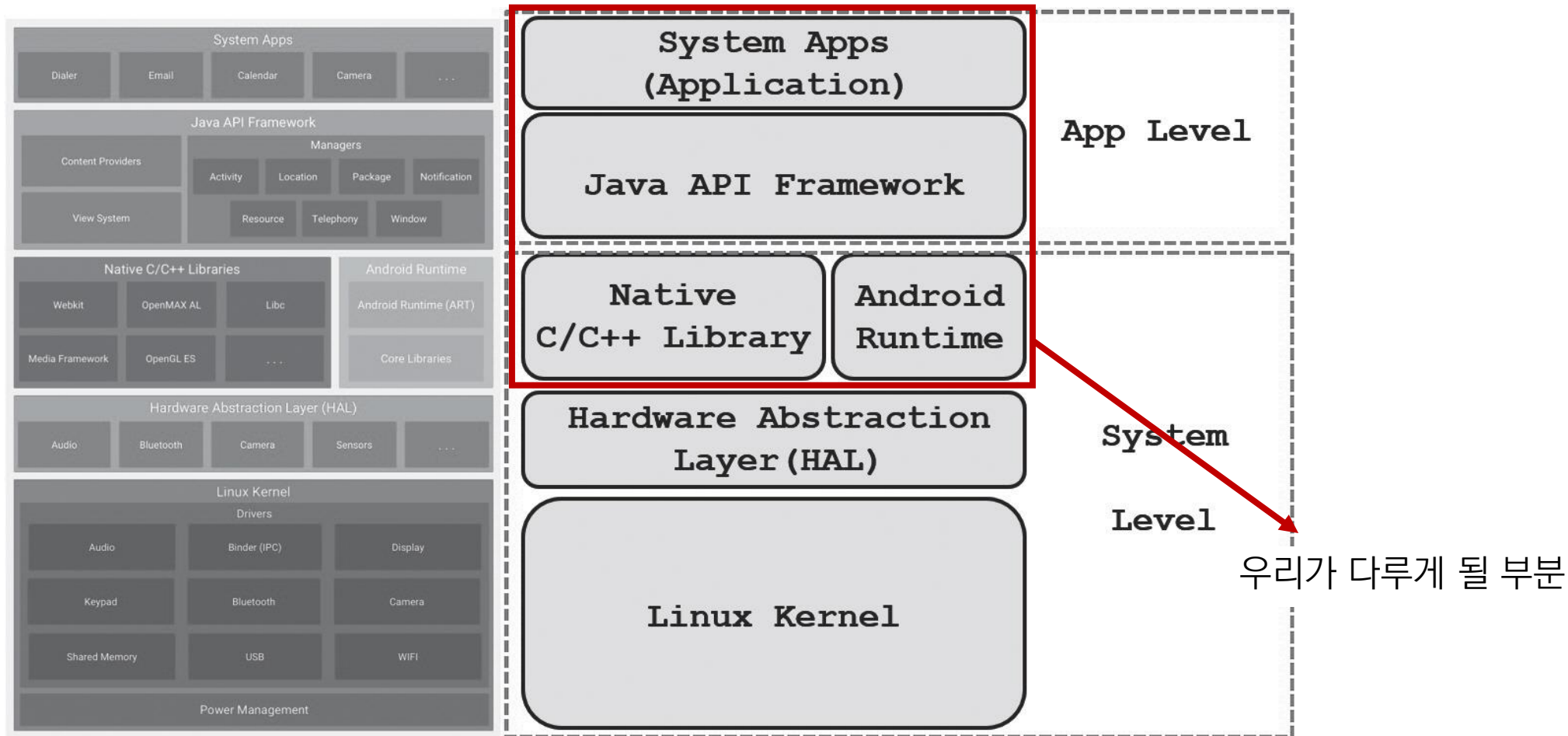


안드로이드(Android OS)

- 구글(google)에서 만든 모바일 운영체제
- 리눅스(!!) 커널 기반
- 오픈소스 플랫폼
- 무려 APK가 JVM(Java Virtual Machine) 기반으로 동작
: DVM(Dalvik Virtual Machine)

안드로이드 해킹이니깐 안드로이드를 알아보자!

안드로이드 아키텍처



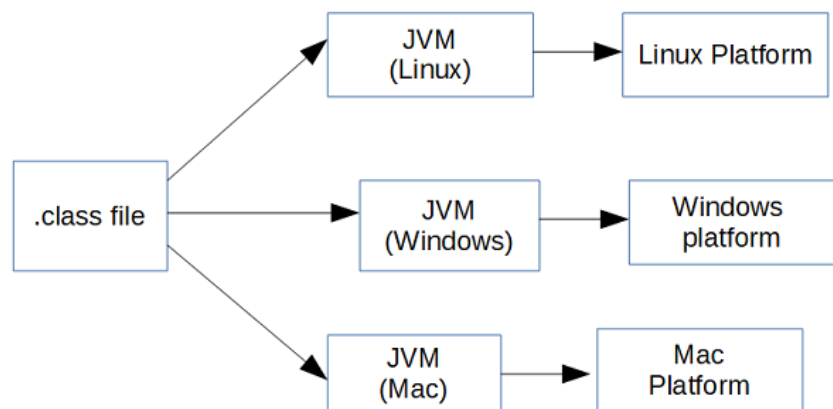
안드로이드 해킹이니깐 안드로이드를 알아보자!

안드로이드 아키텍처

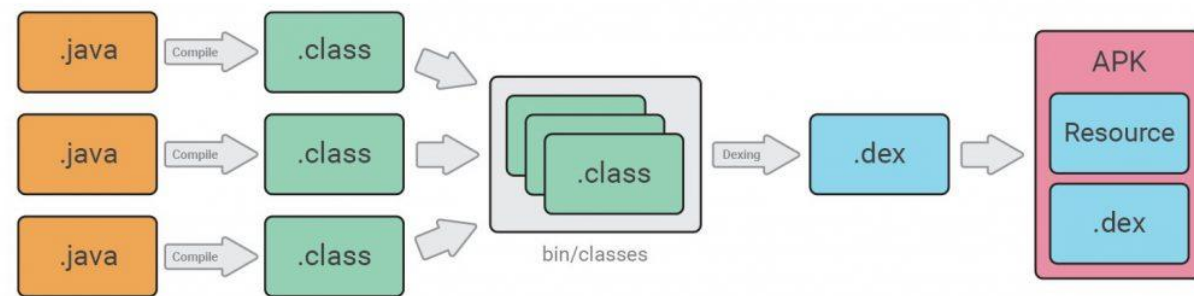
JVM? DVM?

Java Virtual Machine? Dalvik Virtual Machine?

JVM



DVM

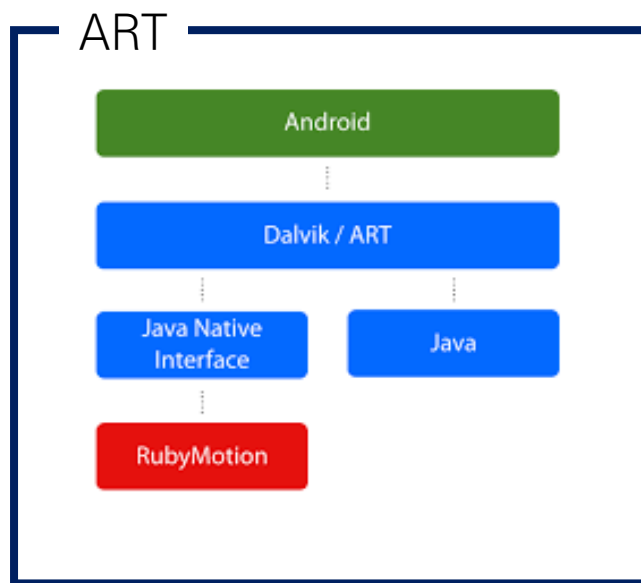


안드로이드 해킹이니깐 안드로이드를 알아보자!

안드로이드 아키텍처

ART

Android Runtime



안드로이드 해킹이니깐 안드로이드를 알아보자!

안드로이드 아키텍처

안드로이드 기기

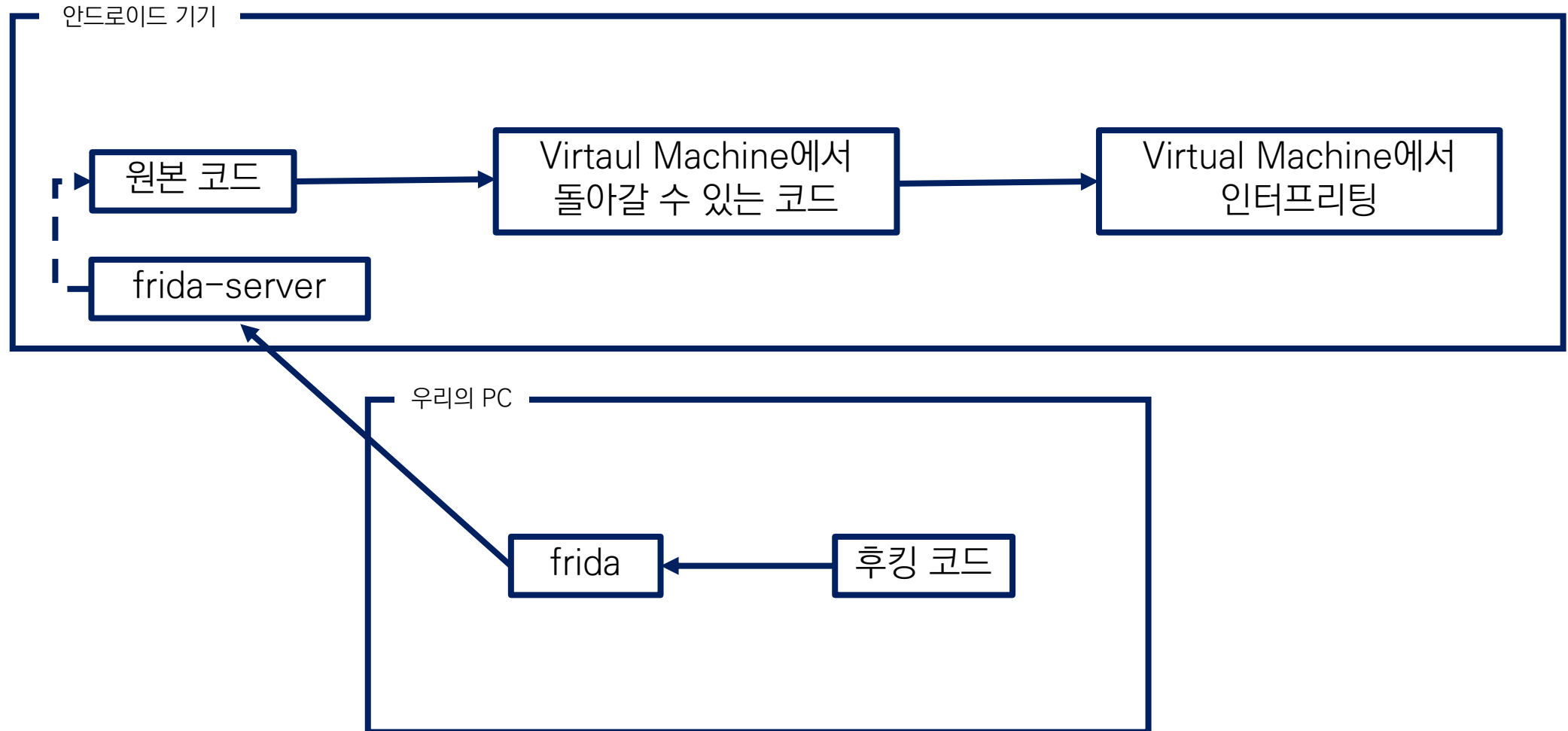
원본 코드

Virtual Machine에서
돌아갈 수 있는 코드

Virtual Machine에서
인터프리팅

안드로이드 해킹이니깐 안드로이드를 알아보자!

안드로이드 아키텍처



안드로이드 APK에 존재하는 메서드를 후킹해보자

메서드 후킹

```
Java.perform(function() {  
    var _class = Java.use(클래스 이름);  
    _class.[메서드 이름].implementation = function() {  
        변경할 루틴  
    }  
});
```

안드로이드 APK에 존재하는 메서드를 후킹해보자

메서드 후킹

android.hello.practice

```
class practice {  
    public int hooktarget() {  
        return 1;  
    }  
}
```

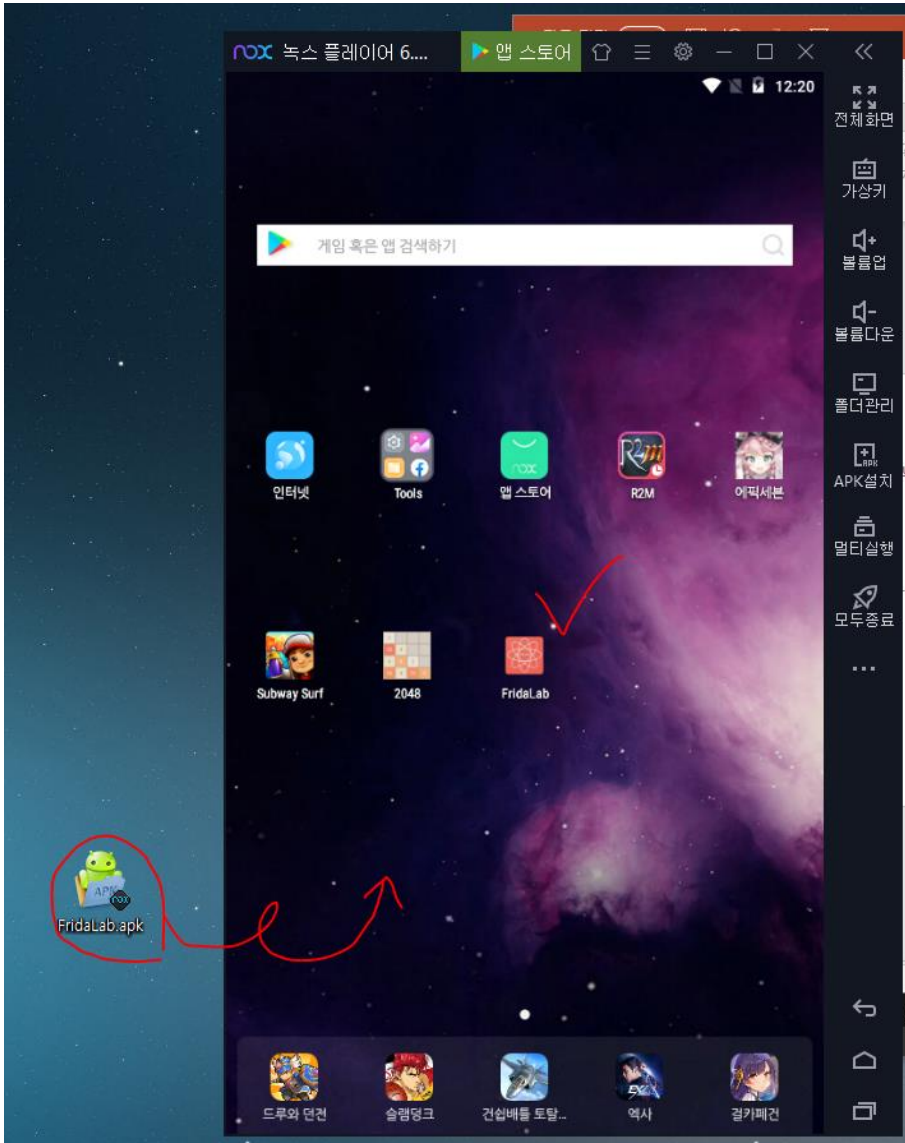
```
class practice {  
    public int hooktarget() {  
        return 0;  
    }  
}
```

후킹 코드

```
Java.perform(function() {  
    var _class = Java.use("android.hello.practice");  
    _class.hooktarget.implementation = function() {  
        return 0;  
    }  
});
```

method hooking

실습 1 – fridalab 01

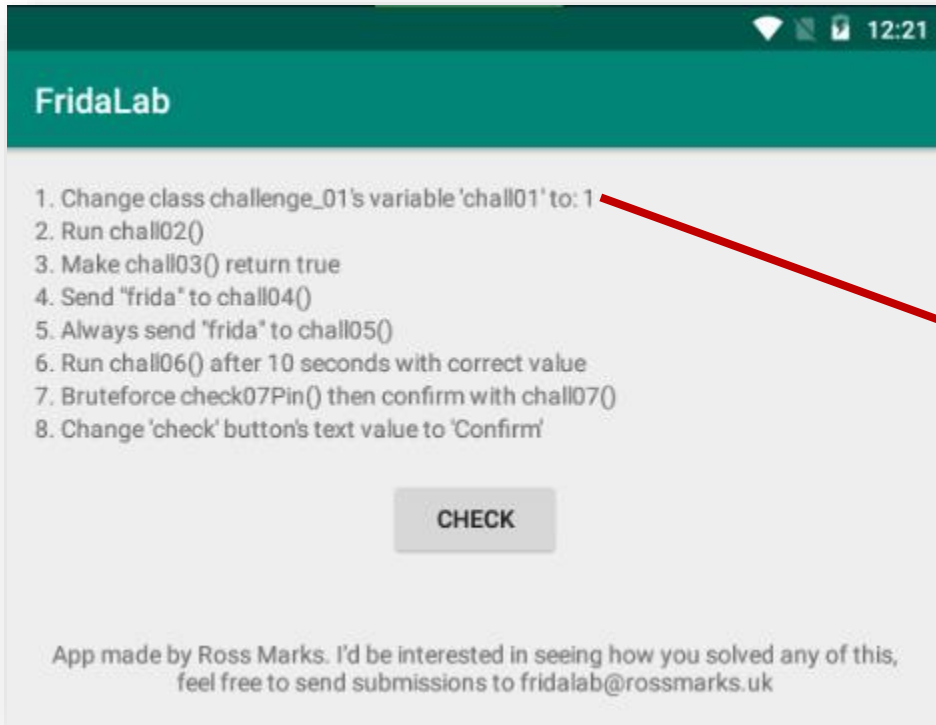


1. APK 다운로드 후 Nox에 drag&drop하여 설치
2. 실행
3. 루팅 모드 ON

method hooking

실습 1 – fridalab 01

frida lab : frida를 이용한 후킹을 연습하기 위해 만들어진 워게임



1번 문제는 특정 클래스의
멤버 변수를 변경하는 것이다

method hooking

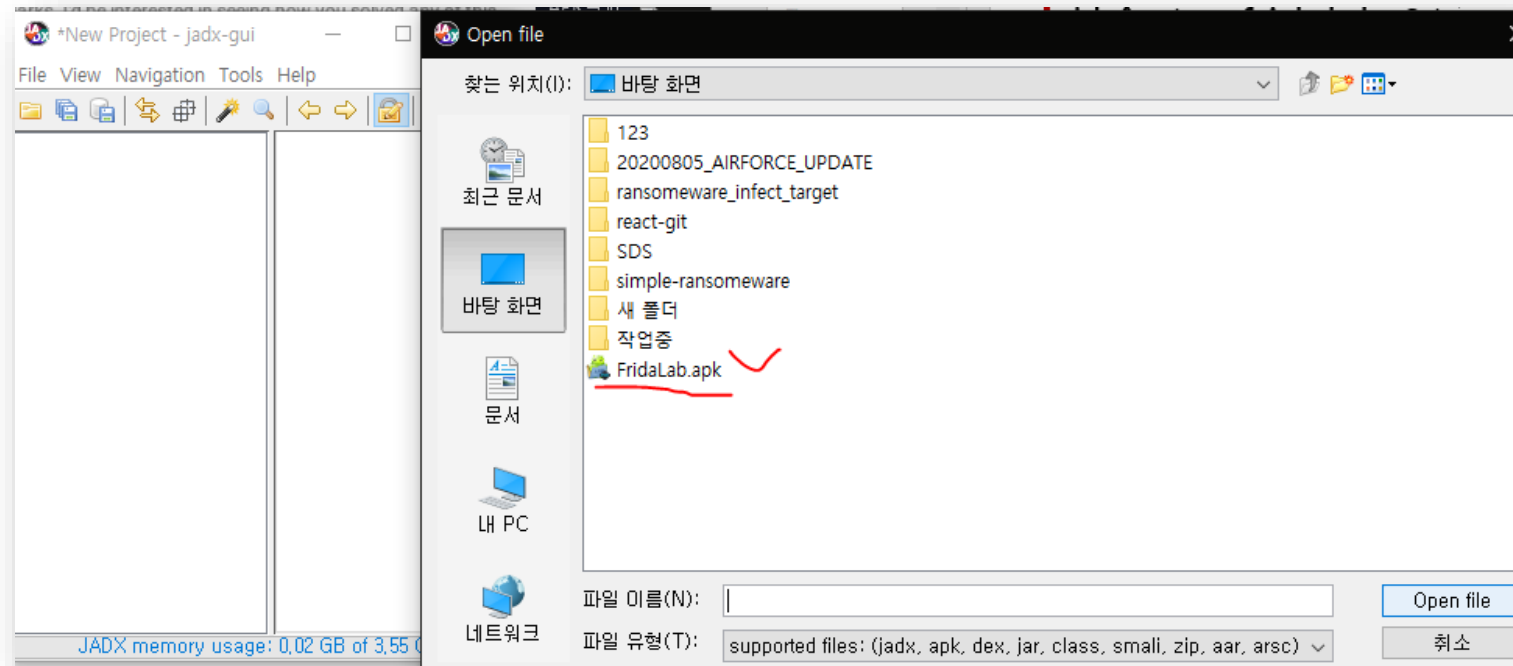
실습 1 – fridalab 01

1. `adb connect 127.0.0.1:62001`
2. `adb shell`
3. `./data/local/tmp/frida(tab) &`

3번의 tab은 자동완성

method hooking

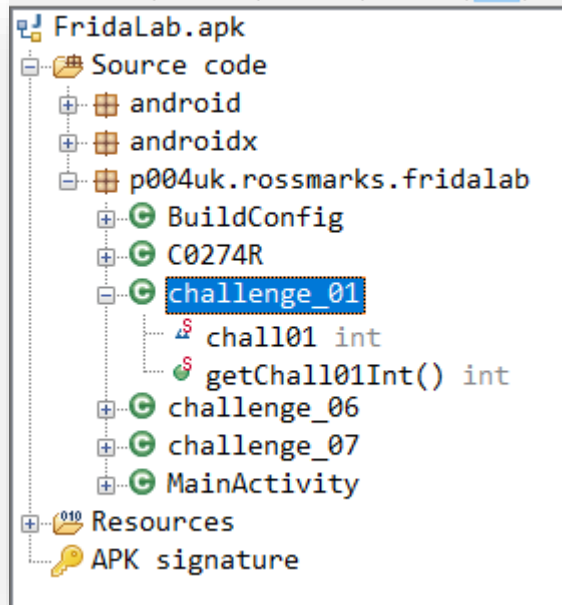
실습 1 – fridalab 01



jadx 실행 후 FridaLab.apk 열기 -> APK를 동적 분석하기 위함

method hooking

실습 1 – fridalab 01



좌측 메뉴 바에서 challenge_01로 이동

```
1 package p004uk.rossmarks.fridalab;
2
3 /* renamed from: uk.rossmarks.fridalab.challenge_01 */
4 public class challenge_01 {
5     static int chall01;
6
7     public static int getChall01Int() {
8         return chall01;
9     }
10 }
```

challenge_01의 구현 코드가 나와있음
어디를 후킹하는 것이 좋을까?

method hooking

실습 1 – fridalab 01

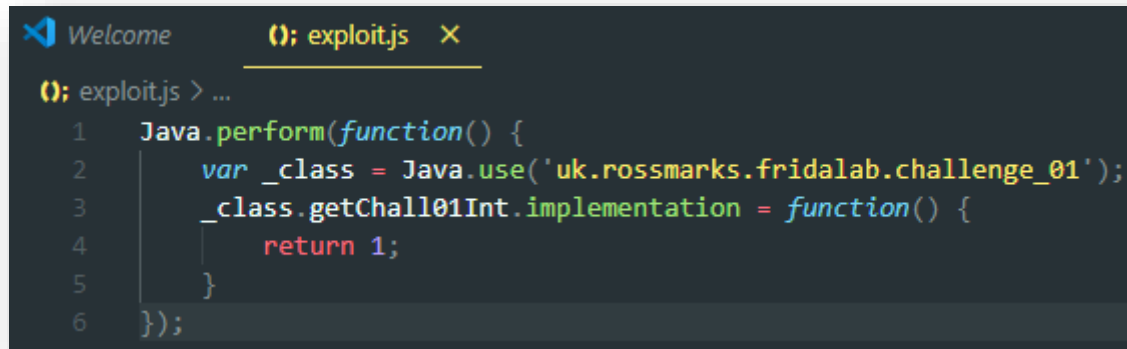
```
1 package p004uk.rossmarks.fridalab;
2
3 /* renamed from: uk.rossmarks.fridalab.challenge_01 */
4 public class challenge_01 {
5     static int chall01;
6
7     public static int getChall01Int() {
8         return chall01;
9     }
10 }
```

앞서 배운 메서드 후킹(implementation)을 이용하여

getChall01Int()의 리턴 값을 바꾼다면?

method hooking

실습 1 – fridalab 01



```
0; exploit.js > ...
1  Java.perform(function() {
2      var _class = Java.use('uk.rossmarks.fridalab.challenge_01');
3      _class.getChall01Int.implementation = function() {
4          return 1;
5      }
6  });
```

exploit.js라는 이름의 파일을 만들고 코드를 위와 같이 작성

visual studio code가 없다면 메모장도 가능!

method hooking

실습 1 – fridablab 01

frida 후킹 스크립트 실행 방법

Frida -U -i [후킹 스크립트 이름] [대상 프로세스 이름]

우리가 작성한 js파일

frida-ps -U

method hooking

실습 1 – fridalab 01

```
1416  sdcard
1470  servicemanager
3620  su
1472  surfaceflinger
1772  system_server
1016  ueventd
3845  uk.rossmarks.fridalab
1475  vinput
1471  vold
2056  wpa_supplicant
1490  zygote

C:\Users\miny7>
```

프로세스 이름은 uk.rossmarks.fridalab

method hooking

실습 1 – fridalab 01

```
C:\Users\miny7\Desktop\123>frida -U -l exploit.js uk.rossmarks.fridalab

  ____
 /  _ \
| (|_) |
|  _ < |
|_| \_|_|

Frida 12.8.20 - A world-class dynamic instrumentation toolkit

Commands:
  help      -> Displays the help system
  object?   -> Display information about 'object'
  exit/quit -> Exit

... More info at https://www.frida.re/docs/home/

[SM G965N::uk.rossmarks.fridalab]-> _
```

FridaLab

1. Change class challenge_01's variable 'chall01' to: 1
2. Run chall02()
3. Make chall03() return true
4. Send "frida" to chall04()
5. Always send "frida" to chall05()
6. Run chall06() after 10 seconds with correct value
7. Bruteforce check07Pin() then confirm with chall07()
8. Change 'check' button's text value to 'Confirm'

CHECK

해킹에서 매우매우 필수적인 요소 루팅!

루팅 탐지와 우회기법

루팅(Rooting)?

모바일 기기에서 구동되는 안드로이드 운영 체제 상에서 최상위 권한(루트 권한)을 얻음으로 해당 기기의 생산자 또는 판매자 측에서 걸어 놓은 제약을 해제하는 행위를 가리키는 말이다.

해킹의 위험때문에 금융권, 게임, ..과 같이 돈과 관련된 어플 에서는 대부분 루팅을 **금지**하고 있음.

해킹에서 매우매우 필수적인 요소 루팅!

루팅 탐지와 우회기법

어떻게 탐지할까?

```
root@shamu:/ # ls /system/bin/ | grep su
su
surfaceflinger
```

루팅된 기기에는 특정 경로에 특정 파일이 존재

/system/bin/su
/system/xbin/su
.
.

해당 파일을 open해서 열리는지 안열리는지 true/false로 판단

이외에도 프로세스를 이용한 탐지, 삼성 Knox, ...

해킹에서 매우매우 필수적인 요소 루팅!

루팅 탐지와 우회기법

우회하는 기법? -> 루팅 탐지하는 함수를 후킹한다

```
public Boolean isRooted() {  
    if(루팅) return true;  
    else    return false;  
}
```

```
public Boolean isRooted() {  
    return false;  
}
```

Rooting Detection Bypass~

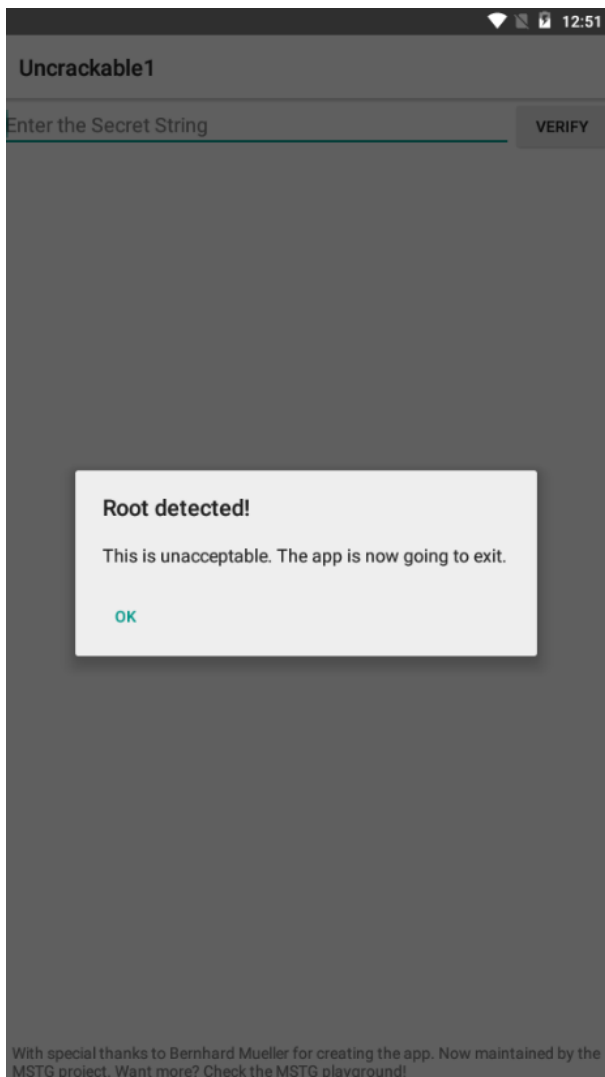
실습 2 – Uncrackable Level 1

https://github.com/OWASP/owasp-mstg/raw/master/Crackmes/Android/Level_01/UnCrackable-Level1.apk

설치후 Nox에 Drag&Drop

frida server를 키는 것까지!

실습 2 – Uncrackable Level 1

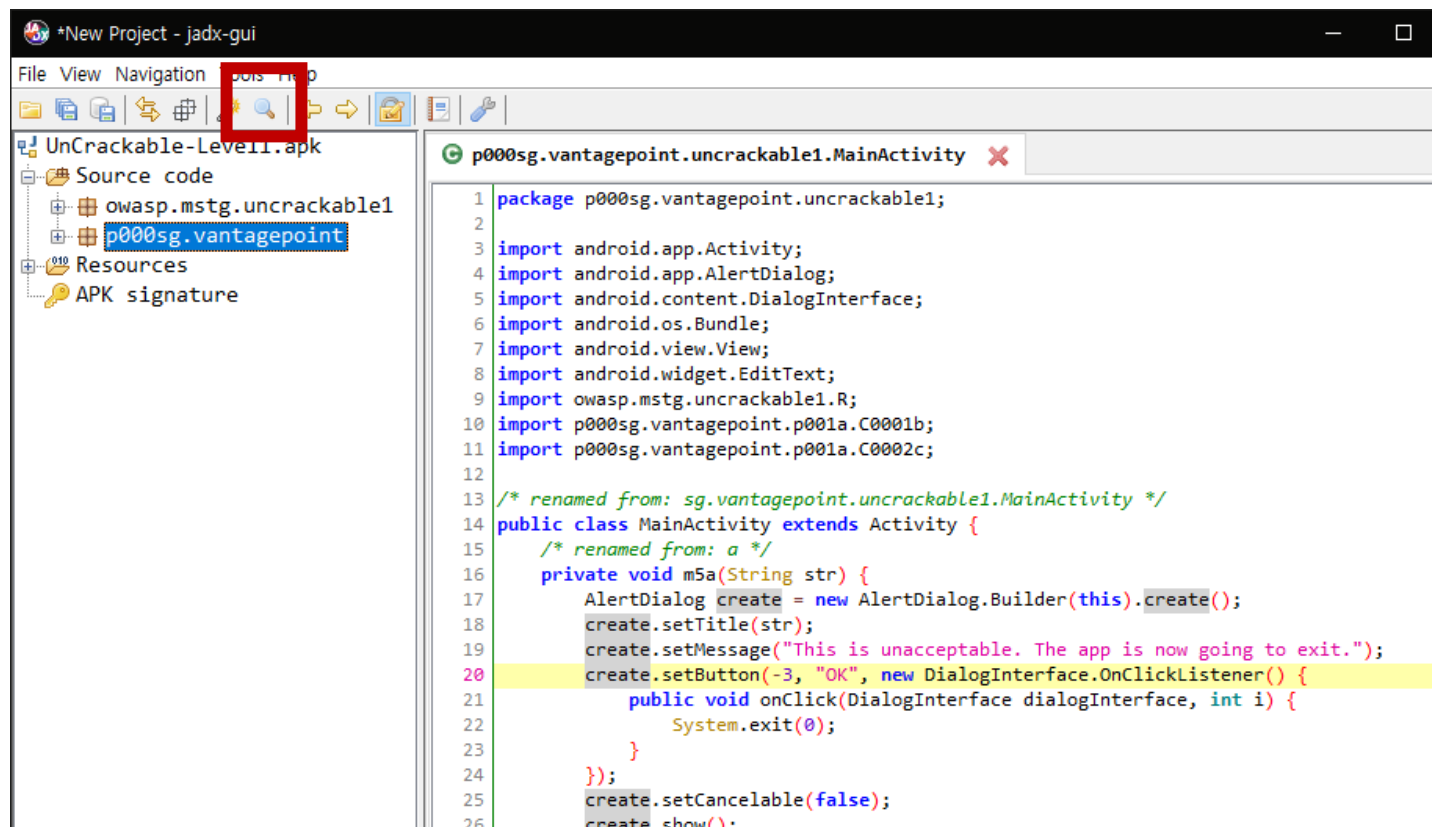


루팅이 된 기기라며 실행이 되지 않는다.

우리는 이것을 정적 분석한 후 후킹할 것이다!

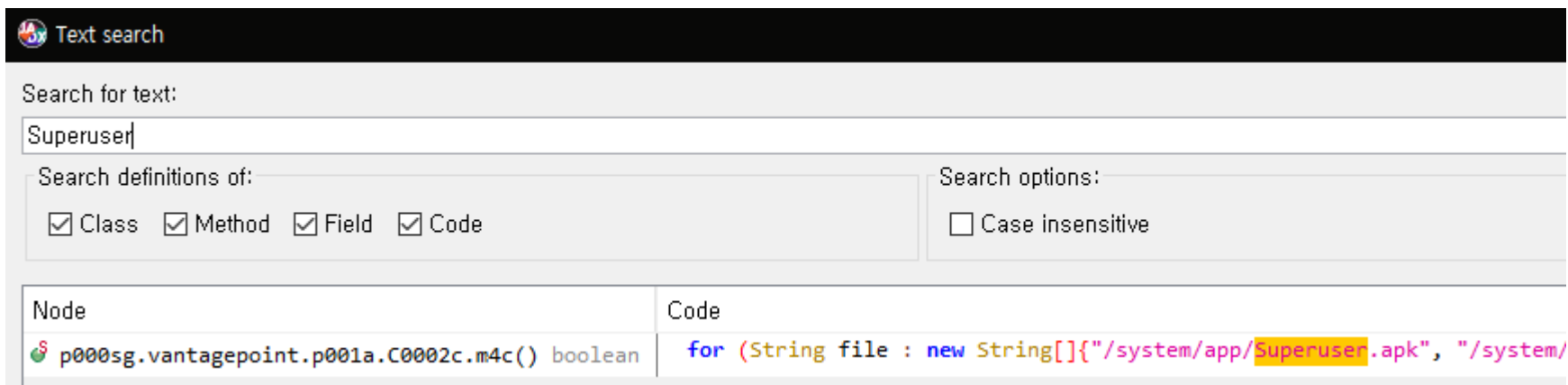
루팅을 탐지하는 함수는 어떻게 찾을까?
→ 팝업 메시지/루팅 파일을 JADX에서 검색

실습 2 – Uncrackable Level 1



상단에 돋보기 클릭

실습 2 – Uncrackable Level 1



루팅 파일인 Superuser.apk를 검색했을 때 어떤 메서드가 잡히는 것을 확인

→ 더블클릭하여 이동!

실습 2 – Uncrackable Level 1

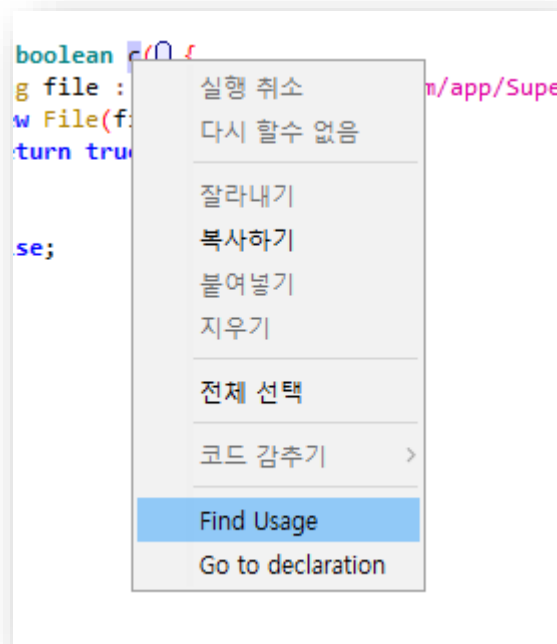
```
public static boolean c() {  
    for (String file : new String[]{"system/app/Superuser.apk",  
        if (new File(file).exists()) {  
            return true;  
        }  
    }  
    return false;  
}
```

루팅 파일이 존재하면 true, 아니면 false 리턴! 그렇다면..?

해당 메서드가 무조건 false를 리턴하도록 후킹..?

메서드명이 난독화(a, b, c)되어있어 후킹할 수 없다.

실습 2 – Uncrackable Level 1



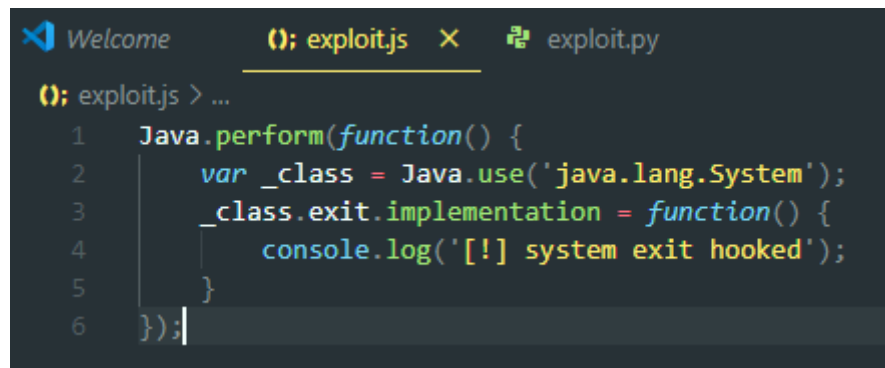
```
/* access modifiers changed from: protected */  
public void onCreate(Bundle bundle) {  
    if (c.a() || c.b() || c.c()) {  
        a("Root detected!");  
    }  
    if (b.a(getApplicationContext())) {  
        a("App is debuggable!");  
    }  
    super.onCreate(bundle);  
    setContentView(R.layout.activity_main);  
}
```


실습 2 – Uncrackable Level 1

```
private void a(String str) {  
    AlertDialog create = new AlertDialog.Builder(this).create();  
    create.setTitle(str);  
    create.setMessage("This is unacceptable. The app is now going to exit.");  
    create.setButton(-3, "OK", new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialogInterface, int i) {  
            System.exit(0);  
        }  
    });  
    create.setCancelable(false);  
    create.show();  
}
```

exit을 후킹하면 어떨까?

실습 2 – Uncrackable Level 1



```

Welcome    (); exploit.js  ×  exploit.py
(); exploit.js > ...
1  Java.perform(function() {
2      var _class = Java.use('java.lang.System');
3      _class.exit.implementation = function() {
4          console.log('[!] system exit hooked');
5      }
6  });

```

어떻게 실행시킬 것인가?

Rooting Detection Bypass~

실습 2 – Uncrackable Level 1

앱 실행 -> 후킹코드 실행

앱 실행 -> 루팅 탐지로 인해 종료 -> 후킹코드 실행불가

Rooting Detection Bypass~

실습 2 – Uncrackable Level 1

앱 실행 -> 실행과 동시에 후킹 코드 실행 -> 루팅 탐지전에 후킹됨 -> 정상 동작

실습 2 – Uncrackable Level 1

```
1  import frida, sys
2
3  def on_message(message, data):
4      if message['type'] == 'send':
5          print("[*] {0}".format(message['payload']))
6      else:
7          print(message)
8
9  PACKAGE_NAME = "owasp.mstg.uncrackable1"
10
11  jscode = """
12  Java.perform(function() {
13      var _class = Java.use('java.lang.System');
14      _class.exit.implementation = function() {
15          console.log('[!] system exit hooked');
16      }
17  });
18  """
19
20  try:
21      device = frida.get_usb_device(timeout=10)
22      pid = device.spawn([PACKAGE_NAME])
23      print("App is starting ... pid : {}".format(pid))
24      process = device.attach(pid)
25      device.resume(pid)
26      script = process.create_script(jscode)
27      script.on('message', on_message)
28      print("[*] Running Frida")
29      script.load()
30      sys.stdin.read()
31  except Exception as e:
32      print(e)
```

```
import frida, sys

def on_message(message, data):
    if message['type'] == 'send':
        print("[*] {0}".format(message['payload']))
    else:
        print(message)

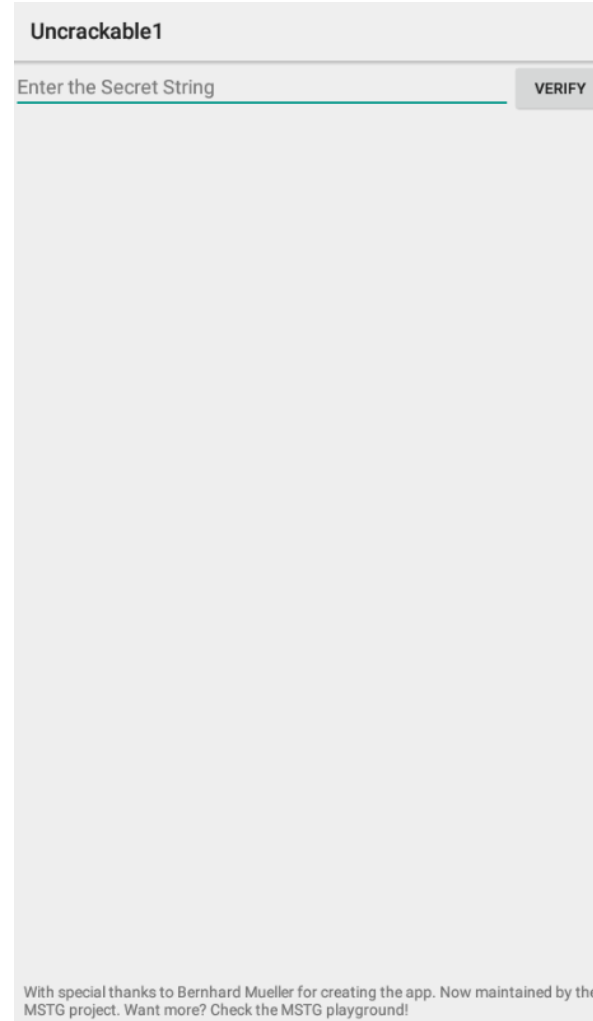
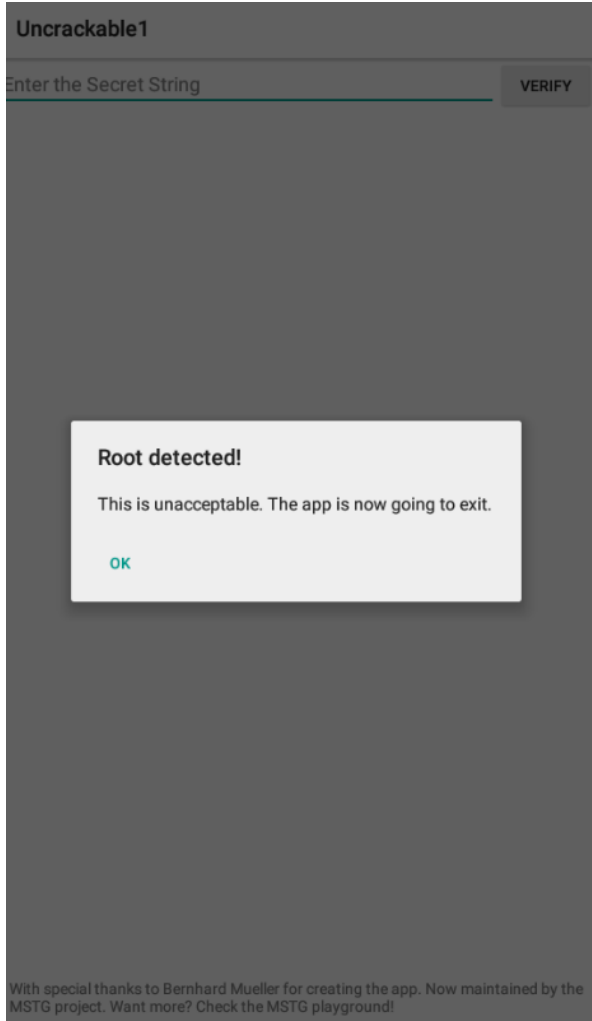
PACKAGE_NAME = "owasp.mstg.uncrackable1"

jscode = """
Java.perform(function() {
    var _class = Java.use('java.lang.System');
    _class.exit.implementation = function() {
        console.log('[!] system exit hooked');
    }
});
"""

try:
    device = frida.get_usb_device(timeout=10)
    pid = device.spawn([PACKAGE_NAME])
    print("App is starting ... pid : {}".format(pid))
    process = device.attach(pid)
    device.resume(pid)
    script = process.create_script(jscode)
    script.on('message', on_message)
    print("[*] Running Frida")
    script.load()
    sys.stdin.read()
except Exception as e:
    print(e)
```

Rooting Detection Bypass~

실습 2 – Uncrackable Level 1



```
App is starting ... pid : 4617
[*] Running Frida
[!] system exit hooked
[]
```

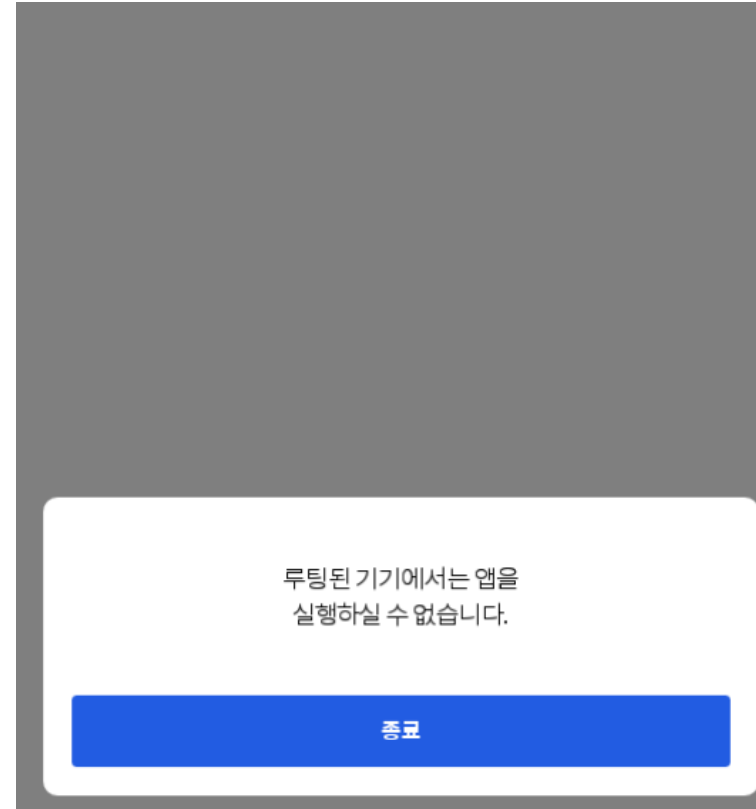
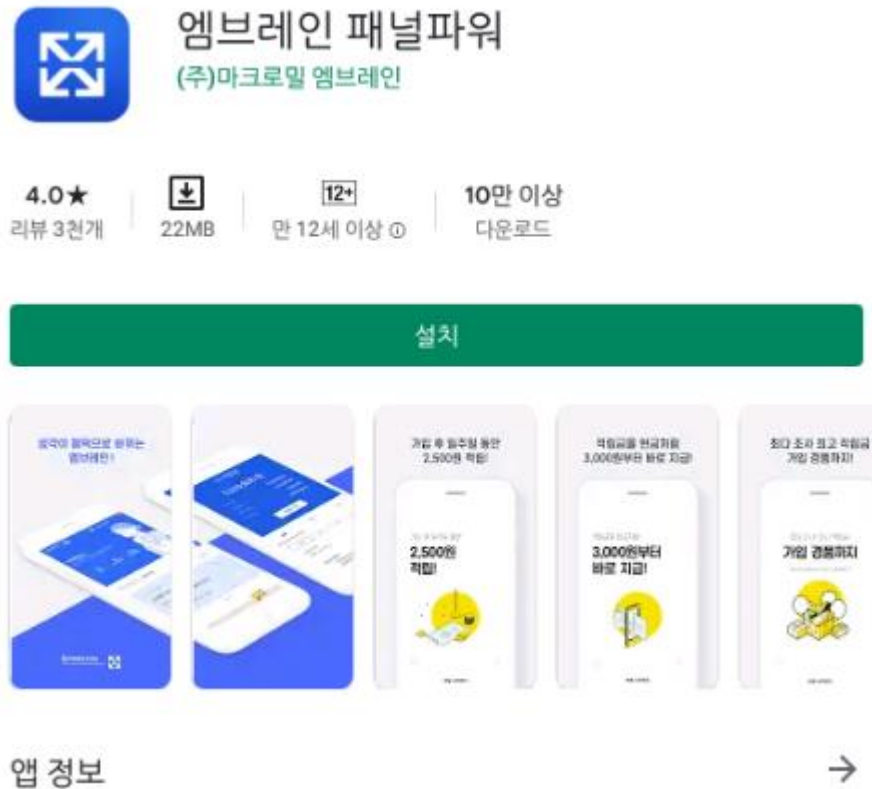
Rooting Detection Bypass~

실습 3 – 실제 사례 분석

CGV를 뚫어보자 부승 빠승~

하나 더 뚫어보자 부승빠승~

과제 – 엠xx인 패x파워



*Hint : 해당 회사에서 사용중인 보안 모듈이 두개 이상일 경우 루팅 탐지 루틴도 두개 이상 존재할 수 있다.