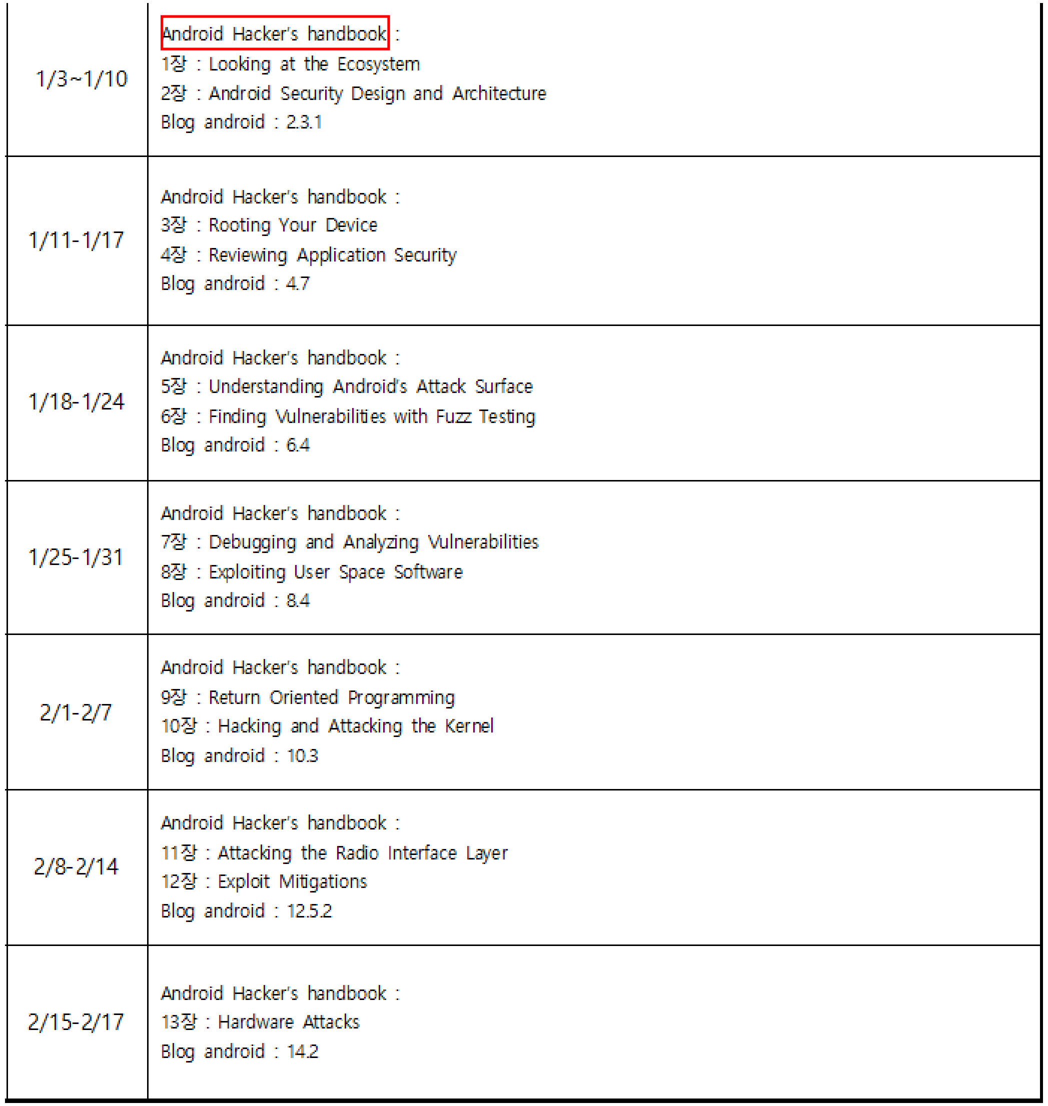
겨울방학 스터디 계획서

소프트웨어 공학부 91913622

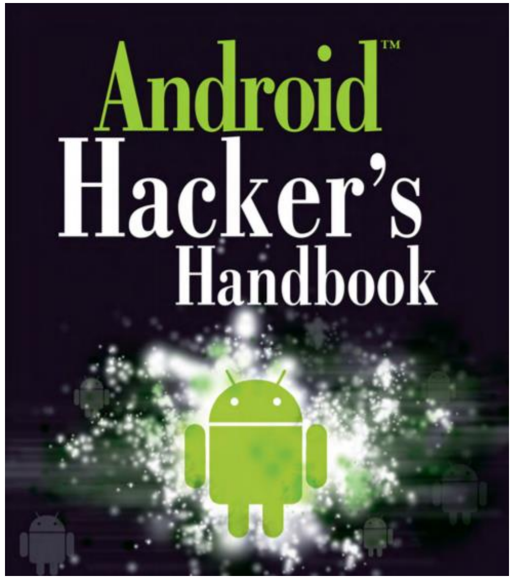
Android study

송태현

Android Study



1/3~1/10	Android Hacker's handbook : 1장 : Looking at the Ecosystem 2장 : Android Security Design and Architecture Blog android : 2.3.1
1/11-1/17	Android Hacker's handbook : 3장 : Rooting Your Device 4장 : Reviewing Application Security Blog android : 4.7
1/18-1/24	Android Hacker's handbook : 5장 : Understanding Android's Attack Surface 6장 : Finding Vulnerabilities with Fuzz Testing Blog android : 6.4
1/25-1/31	Android Hacker's handbook : 7장 : Debugging and Analyzing Vulnerabilities 8장 : Exploiting User Space Software Blog android : 8.4
2/1-2/7	Android Hacker's handbook : 9장 : Return Oriented Programming 10장 : Hacking and Attacking the Kernel Blog android : 10.3
2/8-2/14	Android Hacker's handbook : 11장 : Attacking the Radio Interface Layer 12장 : Exploit Mitigations Blog android : 12.5.2
2/15-2/17	Android Hacker's handbook : 13장 : Hardware Attacks Blog android : 14.2



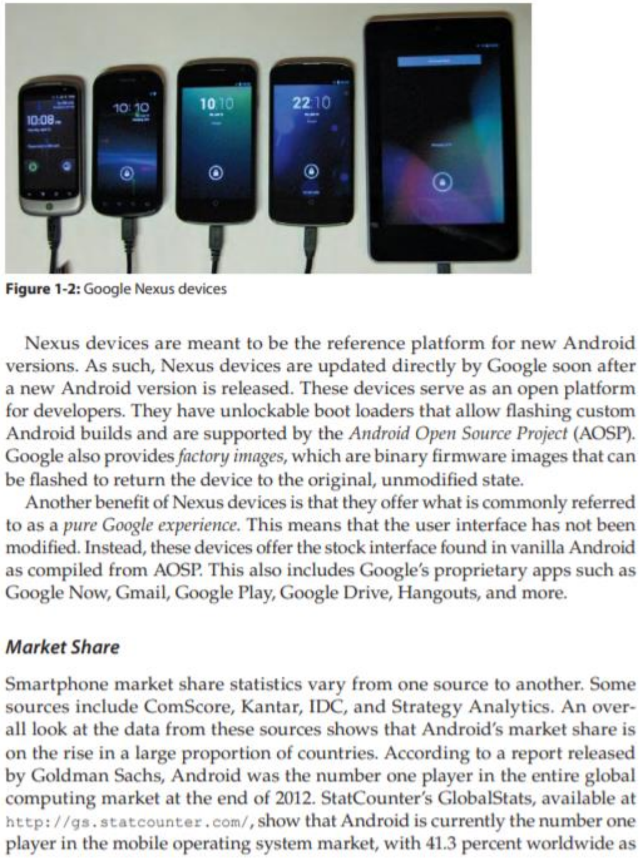


Figure 1-2: Google Nexus devices

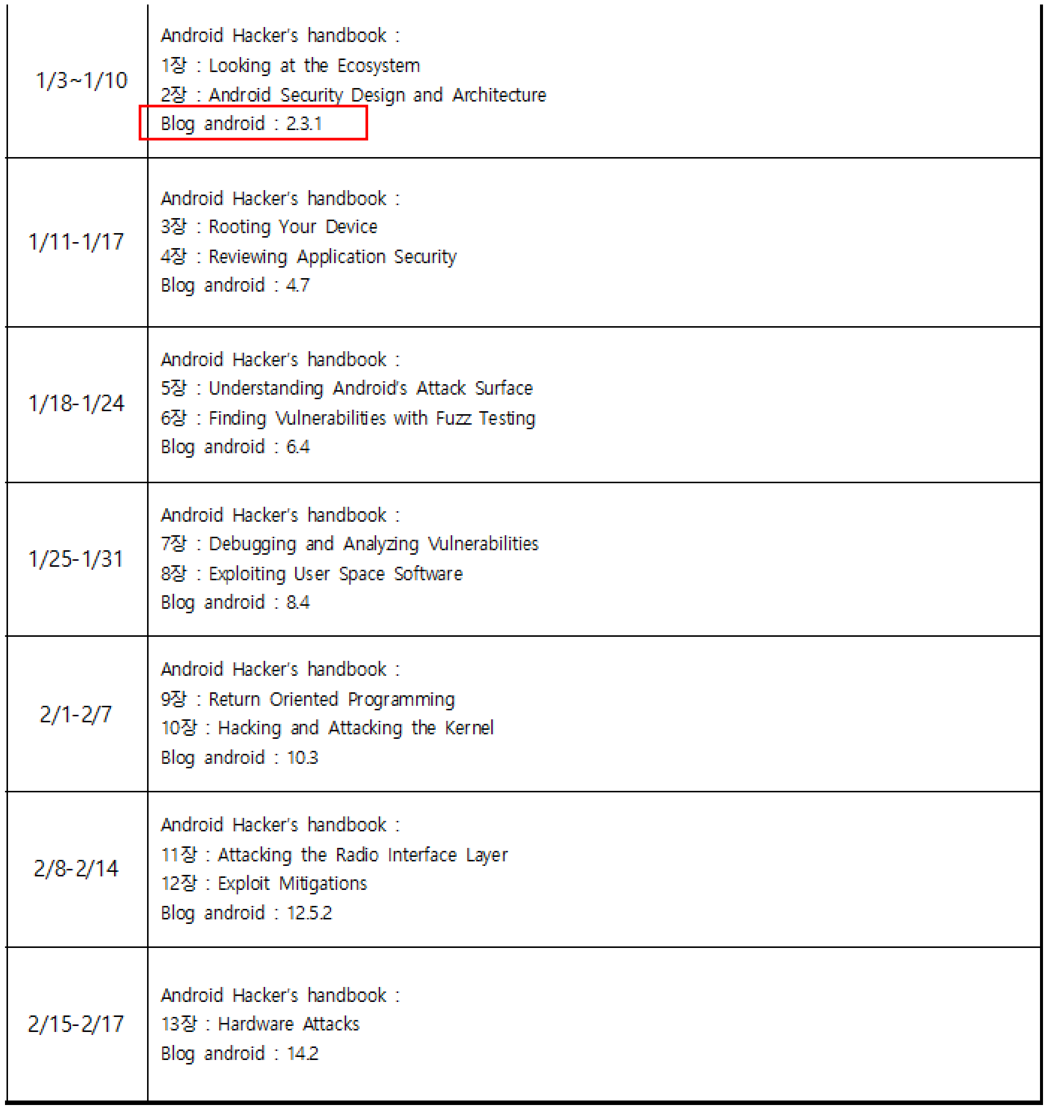
Nexus devices are meant to be the reference platform for new Android versions. As such, Nexus devices are updated directly by Google soon after a new Android version is released. These devices serve as an open platform for developers. They have unlockable boot loaders that allow flashing custom Android builds and are supported by the *Android Open Source Project* (AOSP). Google also provides *factory images*, which are binary firmware images that can be flashed to return the device to the original, unmodified state.

Another benefit of Nexus devices is that they offer what is commonly referred to as a *pure Google experience*. This means that the user interface has not been modified. Instead, these devices offer the stock interface found in vanilla Android as compiled from AOSP. This also includes Google's proprietary apps such as Google Now, Gmail, Google Play, Google Drive, Hangouts, and more.

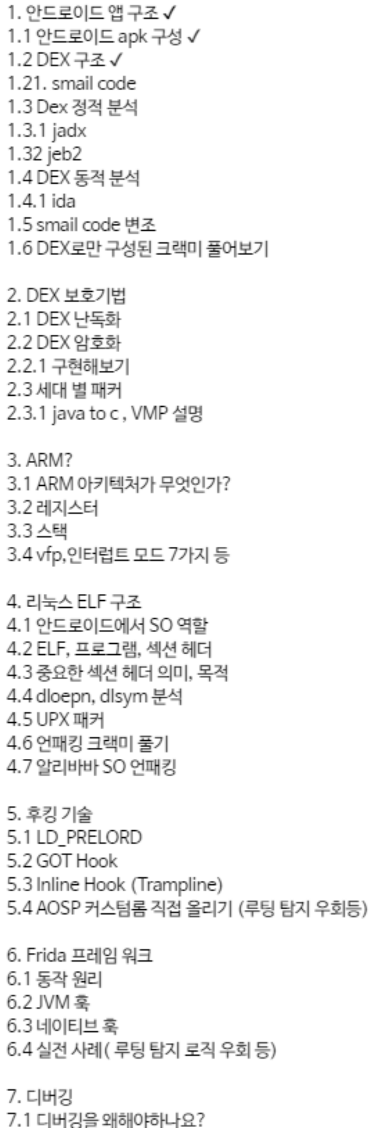
Market Share

Smartphone market share statistics vary from one source to another. Some sources include ComScore, Kantar, IDC, and Strategy Analytics. An overall look at the data from these sources shows that Android's market share is on the rise in a large proportion of countries. According to a report released by Goldman Sachs, Android was the number one player in the entire global computing market at the end of 2012. StatCounter's GlobalStats, available at `http://gs.statcounter.com/`, show that Android is currently the number one player in the mobile operating system market, with 41.3 percent worldwide as

Android Study



1/3~1/10	Android Hacker's handbook : 1장 : Looking at the Ecosystem 2장 : Android Security Design and Architecture Blog android : 2.3.1
1/11-1/17	Android Hacker's handbook : 3장 : Rooting Your Device 4장 : Reviewing Application Security Blog android : 4.7
1/18-1/24	Android Hacker's handbook : 5장 : Understanding Android's Attack Surface 6장 : Finding Vulnerabilities with Fuzz Testing Blog android : 6.4
1/25-1/31	Android Hacker's handbook : 7장 : Debugging and Analyzing Vulnerabilities 8장 : Exploiting User Space Software Blog android : 8.4
2/1-2/7	Android Hacker's handbook : 9장 : Return Oriented Programming 10장 : Hacking and Attacking the Kernel Blog android : 10.3
2/8-2/14	Android Hacker's handbook : 11장 : Attacking the Radio Interface Layer 12장 : Exploit Mitigations Blog android : 12.5.2
2/15-2/17	Android Hacker's handbook : 13장 : Hardware Attacks Blog android : 14.2



1. 안드로이드 앱 구조 ✓
1.1 안드로이드 apk 구성 ✓
1.2 DEX 구조 ✓
1.21. smail code
1.3 Dex 정적 분석
1.3.1 jadx
1.32 jeb2
1.4 DEX 동적 분석
1.4.1 ida
1.5 smail code 변조
1.6 DEX로만 구성된 크랙미 풀어보기

2. DEX 보호기법
2.1 DEX 난독화
2.2 DEX 암호화
2.2.1 구현해보기
2.3 세대 별 패커
2.3.1 java to c , VMP 설명

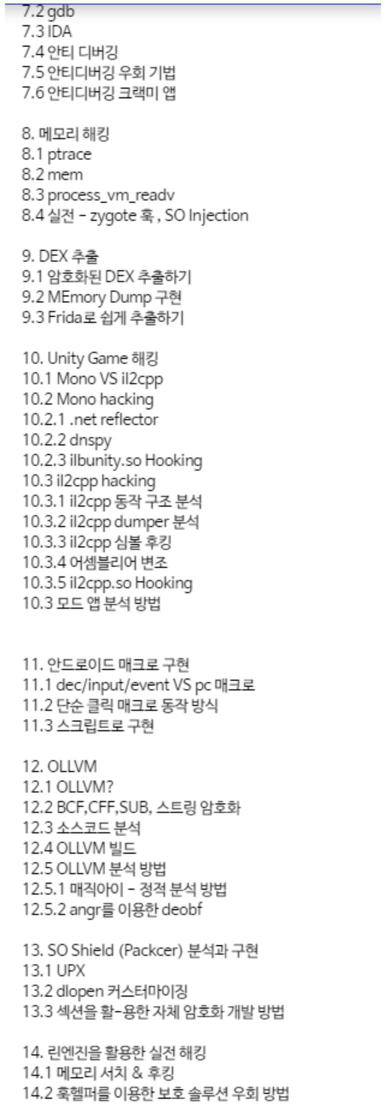
3. ARM?
3.1 ARM 아키텍처가 무엇인가?
3.2 레지스터
3.3 스택
3.4 vfp,인터럽트 모드 7가지 등

4. 리눅스 ELF 구조
4.1 안드로이드에서 SO 역할
4.2 ELF, 프로그램, 섹션 헤더
4.3 중요한 섹션 헤더 의미, 목적
4.4 dloepn, dlsym 분석
4.5 UPX 패커
4.6 언패킹 크랙미 풀기
4.7 알리바바 SO 언패킹

5. 후킹 기술
5.1 LD_PRELORD
5.2 GOT Hook
5.3 Inline Hook (Trampline)
5.4 AOSP 커스텀롬 직접 올리기 (루팅 탐지 우회등)

6. Frida 프레임 워크
6.1 동작 원리
6.2 JVM 훅
6.3 네이티브 훅
6.4 실전 사례(루팅 탐지 로직 우회 등)

7. 디버깅
7.1 디버깅을 왜해야하나요?



7.2 gdb
7.3 IDA
7.4 안티 디버깅
7.5 안티디버깅 우회 기법
7.6 안티디버깅 크랙미 앱

8. 메모리 해킹
8.1 ptrace
8.2 mem
8.3 process_vm_readv
8.4 실전 – zygote 훅 , SO Injection

9. DEX 추출
9.1 암호화된 DEX 추출하기
9.2 MEmory Dump 구현
9.3 Frida로 쉽게 추출하기

10. Unity Game 해킹
10.1 Mono VS il2cpp
10.2 Mono hacking
10.2.1 .net reflector
10.2.2 dnspy
10.2.3 ilbunity.so Hooking
10.3 il2cpp hacking
10.3.1 il2cpp 동작 구조 분석
10.3.2 il2cpp dumper 분석
10.3.3 il2cpp 심볼 후킹
10.3.4 어셈블리어 변조
10.3.5 il2cpp.so Hooking
10.3 모드 앱 분석 방법

11. 안드로이드 매크로 구현
11.1 dec/input/event VS pc 매크로
11.2 단순 클릭 매크로 동작 방식
11.3 스크립트로 구현

12. OLLVM
12.1 OLLVM?
12.2 BCF,CFF,SUB, 스트링 암호화
12.3 소스코드 분석
12.4 OLLVM 빌드
12.5 OLLVM 분석 방법
12.5.1 매직아이 – 정적 분석 방법
12.5.2 angr를 이용한 deobf

13. SO Shield (Packcer) 분석과 구현
13.1 UPX
13.2 dlopen 커스터마이징
13.3 섹션을 활–용한 자체 암호화 개발 방법

14. 린엔진을 활용한 실전 해킹
14.1 메모리 서치 & 후킹
14.2 훅헬퍼를 이용한 보호 솔루션 우회 방법