

모바일 앱과 클라우드 서비스 연동 에서 발생하는 보안 취약점 2

BCG 장건희

02/28

목차

- 하드코딩
- 어플리케이션 해킹 유형
 - o ios생태계에서의 방법
- 앱 패키지 분석으로 기술적 정보 파악하기
- 크리덴셜 정보 관리
- 모니터링 비용 감축의 위험성

참조

- NHN Cloud On 웨비나 10 | 모바일 해킹에 대응하는 강력한 앱 보안 솔루션 <https://youtu.be/69vFnVR3Nu0?si=Jc-WjlAXeQynbJDy>
- 2020 공개SW 페스티벌 Track1_Session7 "모바일 앱의 클라우드 보안취약점과 대응방안" 박현준(네이버) <https://youtu.be/gS6ErctP8hQ?si=ZBpqwcdq-2v490ev>
- Zimperium(짐페리움) 모바일 앱 보호와 소스코드 난독화 <https://youtu.be/VM2bbfVccRg?si=t1yXcFhHsVacZ7su>
- 모바일 취약점 진단을 할 때 꼭 봐야 할 가이드 (영상 자막) <https://youtu.be/LmsSDylsPN8?si=0f3nlDZOazp4CZK9>

하드코딩

- 상수나 변수에 들어가는 값을 소스코드에 직접 쓰는 방식
- 모바일 앱 실행 시 사용자에게 받을정보를 소스코드에 입력하거나 변수, 아이디, 비밀번호, 대칭키 등 중요 정보를 주석 처리하는 것도 하드코딩.

하드코딩

- 상수나
- 모바일 변수, 하드코딩

InjectorPCA (injector404) 95 Reputation -1.00 Rank 49th Signal 10.00 Impact 75th Percentile

4209223 Open S3 Bucket Writeable To Any Aws User

State: Resolved (Closed) Severity: High (7 ~ 8.9)

Disclosed: March 30, 2017 8:22am +0900 Participants: 1

Reported To: Ruby Visibility: Disclosed (Full)

Weakness: Improper Authentication - Generic

Bounty: \$500

TIMELINE

injector404 submitted a report to Ruby. Feb 27th (2 years ago)

Hi All,

I know that <http://rubyci.s3.amazonaws.com> is used for file uploads on reports and so when i open your s3 bucket i able see all of your public/private files i already see you fix this vulnerability but it not completely fixed

```
root@injector:~# aws s3 ls s3://rubyci
PRE aix71_ppc/
PRE amazon/
PRE arch/
PRE archive/
PRE armv8b/
PRE c64b/
PRE centos5-32/
PRE centos5-64/
PRE centos7/
```

사진공유 앱의
클라우드 저장소
루트키 하드코딩

클라우드 저장소의
잘못된 권한
부여(읽기/쓰기 가능)

jarvanxing / skypixel_lottery

Branch: master skypixel_lottery / www / public / main.js

jarvanxing 3d8e106f204c InputFile

1 contributor

655 lines (355 stmt) 36.3 KB

```
1 function s3(tablename, tabledata, successCallback, errorCallback) {
2     var bucketName = "static-skypixel-beta-ne";
3     AWS.config.update({
4         accessKeyId: 'AKIAI90N1F29H532C0TA',
5         secretAccessKey: 'SdV8Zsu/4D8nbykaBhCBQC6Pvta7LD8u5u7Snp',
6         region: 'us-west-2',
7         bucket: bucketName
8     });
9
10    // var url = "http://" + bucketName + ".s3.amazonaws.com/skypixel_lottery/" + tablename + ".json";
11}
```

정적분석

- 소프트웨어를 실행할 필요가 없음
- 소스코드를 분석
- 언패키징 - 분석 - 리패키징 과정을 거침

동적분석

- 디버깅
- 실제 프로그램을 분석
- 스트레스 테스트
- 모의 해킹
- 리버스 엔지니어링

• 두 가지 분석 방법

- 정적 분석
- 동적 분석

Frida(Dynamic Binary Instrumentation)동적 바이너리
계측 도구

```
.text:00081BF0      public gets
.text:00081BF0      gets                proc near
.text:00081BF0
.text:00081BF0      arg_0                = dword ptr 8
.text:00081BF0
.text:00081BF0      ; __unwind {
.text:00081BF0      push                ebp
.text:00081BF1      mov                 ebp, esp
.text:00081BF3      push                ebx
.text:00081BF4      push                edi
.text:00081BF5      push                esi
.text:00081BF6      and                 esp, 0FFFFFFFh
.text:00081BF9      sub                 esp, 10h
.text:00081BFC      call                $+5
; DATA XREF: gets+124c
.text:00081C01      loc_81C01:
.text:00081C01      pop                 ebx
.text:00081C02      add                 ebx, (offset_GLOBAL_OFFSET_TABLE_ - offset loc_81C01)
.text:00081C08      mov                 eax, ds:(stdin_ptr - 0F9754h)[ebx]
.text:00081C0E      mov                 [esp+8], eax
.text:00081C12      mov                 eax, [eax]
.text:00081C14      mov                 ecx, [eax+30h]
.text:00081C17      cmp                 byte ptr [ecx+20h], 0
.text:00081C18      jnz                 short loc_81C25
.text:00081C1D      mov                 [esp], eax ; stream
.text:00081C20      call                _flockfile
; CODE XREF: gets+2B7j
.text:00081C25      loc_81C25:
.text:00081C25      mov                 esi, [ebp+arg_0]
.text:00081C28      xor                 edi, edi
.text:00081C2A      jmp                 short loc_81C34
```

FRIDA

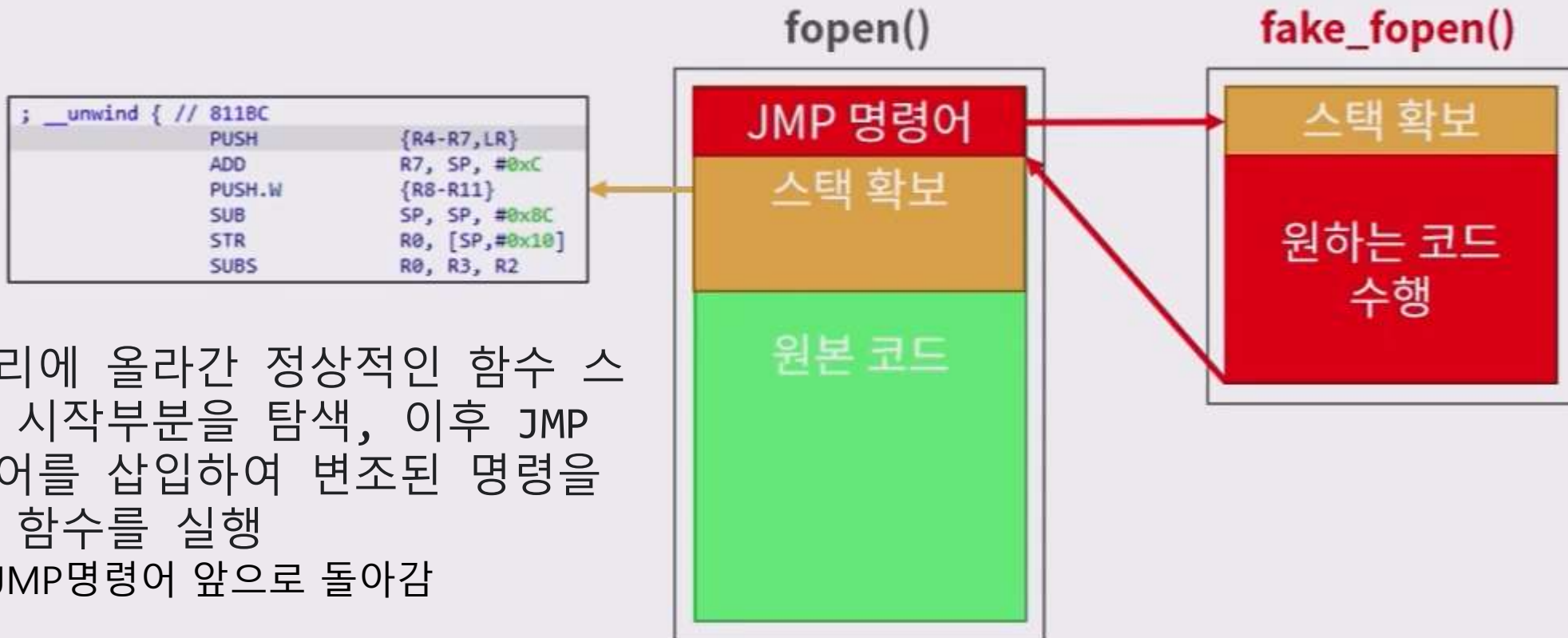
[OVERVIEW](#)
[DOCS](#)
[NEWS](#)
[CODE](#)
[CONTACT](#)

Dynamic instrumentation
toolkit for developers,
engineers, and
researchers.

```
Breakpoint 1, 0x0008054 in _start ()
-----[ registers ]-----
r0 : 0x00000000
r1 : 0x00000000
r2 : 0x00000000
r3 : 0x00000000
r4 : 0x00000000
r5 : 0x00000000
r6 : 0x00000000
r7 : 0x00000000
r8 : 0x00000000
r9 : 0x00000000
r10 : 0x00000000
r11 : 0x00000000
r12 : 0x00000000
rap : 0x00000000 -> 0x00000001
sfr : 0x00000000
apc : 0x00000054 -> = start=0; push (r12, %r)
pcsr : [thumb fast interrupt overflow carry zero negative]
-----[ stack ]-----
0xbefff950[+0x00: 0x00000001 <- $sp
0xbefff954[+0x04: 0xbefff94e -> "/home/pi/lab/gdb-example"
0xbefff958[+0x08: 0x00000000
0xbefff95c[+0x0c: 0xbefff967 -> "TERM=vt100"
0xbefff960[+0x10: 0xbefff972 -> "SHELL=/bin/bash"
0xbefff964[+0x14: 0xbefff982 -> 0x5f474458
0xbefff968[+0x18: 0xbefff9d1 -> "LC_ALL=en_US.UTF-8"
0xbefff96c[+0x1c: 0xbefff9e4 -> "USER=pi"
-----[ code:armv7 ]-----
0x801c      andeq    r8, r0, r0
0x8040      andeq    r9, r0, r0
0x8044      muleq    r9, r4, r0
0x8048      muleq    r9, r4, r0
0x804c      andeq    r9, r0, r1
0x8050      andeq    r9, r0, r0
-> 0x8054 = start=0; push (r12, %r)
0x8058 < start+4>      add    r11, sp, #0
0x805c < start+8>      sub    sp, sp, #16
0x8060 < start+12>     mov    r0, #1
0x8064 < start+16>     mov    r1, #2
0x8068 < start+20>     bl     0x8074 <max>
-----[ threads ]-----
[#0] Id 1, Name: "gdb-example", stopped, reason: BREAKPOINT
-----[ trace ]-----
[#0] 0x8054->Name: _start()
gef>
```


동적 분석 원리 - 인라인 후킹

- 후킹 : 갈고리로 가로채는 이미지



메모리에 올라간 정상적인 함수 스택의 시작부분을 탐색, 이후 JMP 명령어를 삽입하여 변조된 명령을 담은 함수를 실행
이후 JMP명령어 앞으로 돌아감

iOS 생태계 유형1 - 탈옥



ios - tweak

Tweak : 비틀다

Cydia 라고 하는 탈옥
앱스토어 에서 설치,

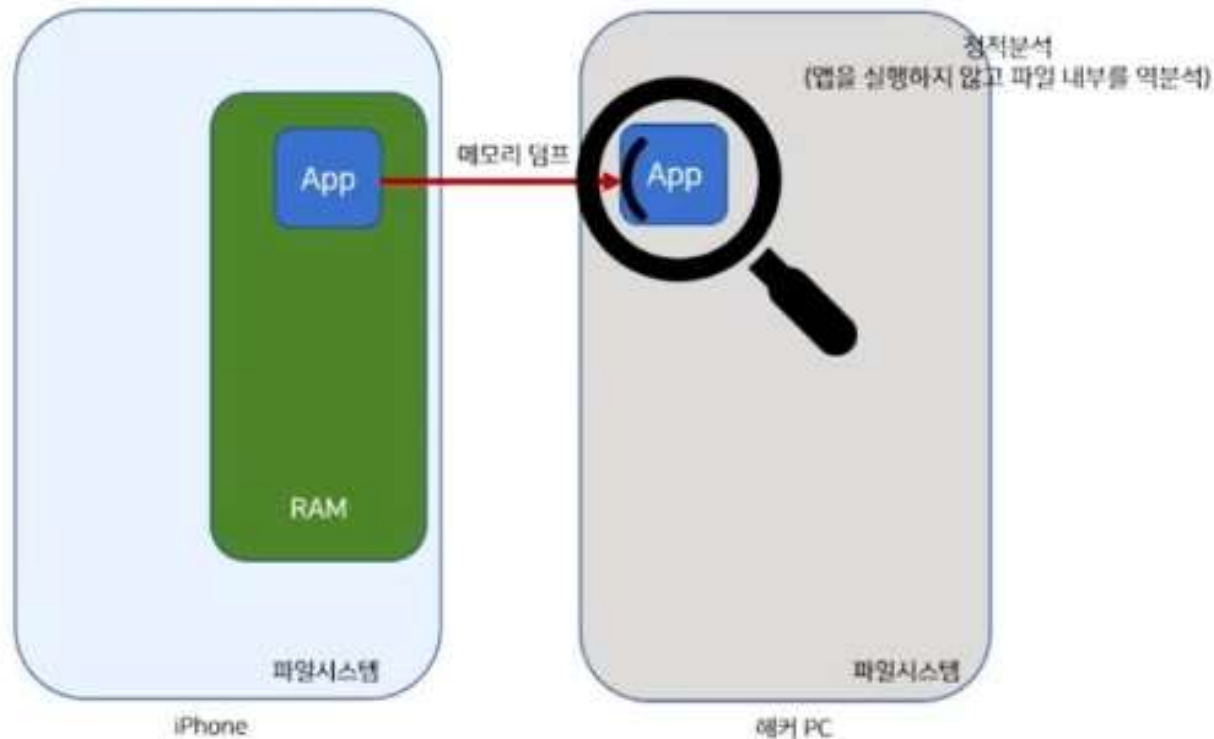
정상 프로그램 실행시
메모리에 tweak을 삽입
하여 메모리 어뷰징등
의 허가되지 않은 동작
을 실행

iOS 모바일 해킹 유형: tweak (dylib injection)



iOS 생태계 유형 2 - 앱 위변조

iOS 모바일 해킹 유형: 앱 위변조



모바일 앱 분석

- Back-End 아키텍처 구성
 - 내장된 클라우드 SDK를 통해 서비스를 호출(REST API)
 - 사용자인증에 Credential (Access/Secret Key) 필요

모

- B



- 클라우드 SDK가 포함된 앱 : 클라우드 back-end 서비스가 연동
- 앱에서 클라우드 서비스 접근 시 Credential 정보를 사용

☞ Credential을 훔칠 수 있다면 클라우드 자원에 접근 가능

모니터링 비용 절감의 위험성

- 소규모 서비스일때는 큰문제가 없지만 서비스가 커지면서 모니터링 비용이 커질수가 있음.
- 대략적으로 중-대 규모의 서비스에서 월간 발생하는 로그 저장비용, 로그 수집비용, 모니터링 서비스 사용료등을 합치면 \$500 - \$2,000 정도의 비용이 나감.
- 하지만 기술장애나 보안 사고시에 기록이 없으면 사건 규명도 하지 못하므로 로그 저장비용을 절약하는 행위를 지양할 필요가 있음.

* AWS CloudWatch, Azure Monitor, Google Cloud Monitoring 등의 기본 적인 로그 서비스

보안을 생각하는 클라우드 사용

클라우드 키에 대한 적절한 권한 통제
루트키 사용 금지
키 하드코딩 금지

- 클라우드 보안에 대한 비용 투자
 로그 + 모니터링
- 내부 토론, 보안 컨설팅을 통한 리스크 관리
 버그바운티 프로그램 참여