NHN Cloud On

모바일 해킹 사례와 NHN AppGuard를 통한 대응 방안

이범수, 이준영 / 클라이언트개발팀

모바일 해킹 유형들과 앱 보안 방법

모바일 해킹이란?

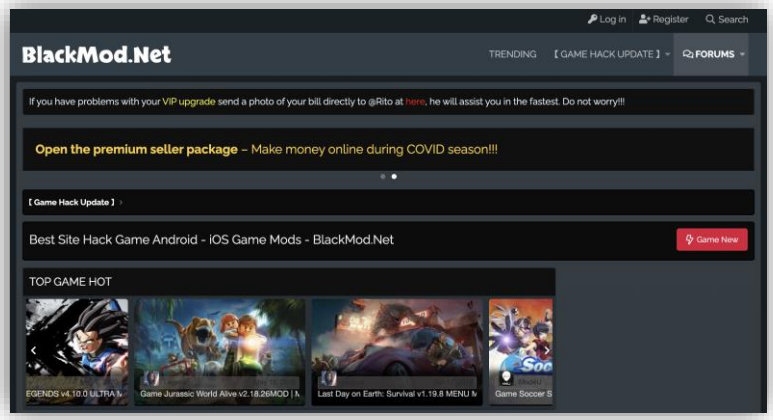
- **해킹**
  - 해킹은 전자 회로나 컴퓨터의 하드웨어, 소프트웨어, 네트워크, 웹사이트 등 각종 정보 체계가 본래의 설계자나 관리자, **운영자가 의도하지 않은 동작**을 일으키도록 하거나 체계 내에서 **주어진 권한 이상으로 정보를 열람, 복제, 변경 가능하게 하는 행위**를 광범위하게 이르는 말

모바일 해킹이란?

- **모바일 해킹**
  - 의도하지 않은 동작
  - 주어진 권한 이상으로 정보를 열람, 복제, 변경을 가능하게 하는 행위

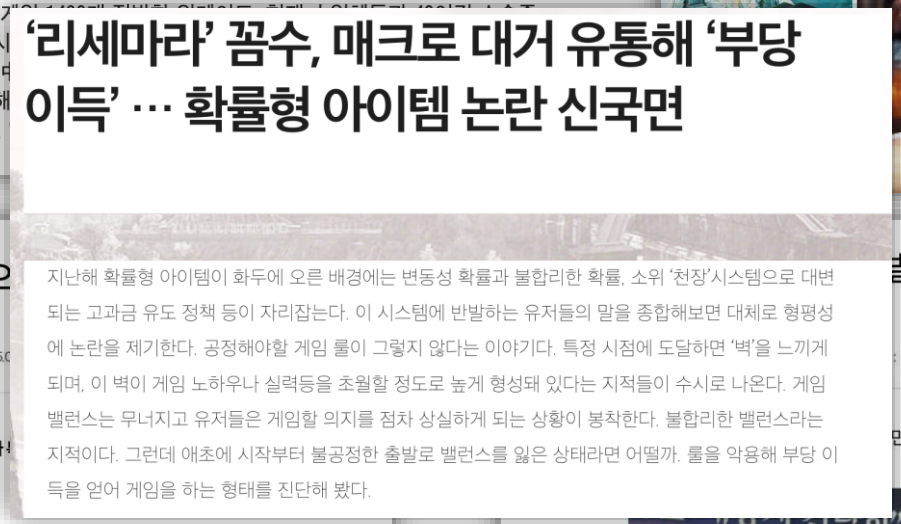
모바일 해킹이란?

- 모바일 해킹
  - 왜?



모바일 해킹이란?

- 해킹 사례들
  - 복제 앱, 리패키징 앱, 치팅, 리세마라



모바일 해킹이란?

- **해킹사고 발생에 따른 영향**
  - 사용자 수 급감
  - 매출 하락
  - 기업 신뢰도 하락

Android 모바일 해킹 유형 및 대응

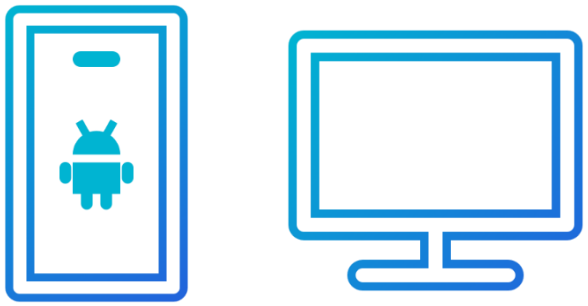
- 게임 앱 해킹
- 대응 방안

Android 모바일 해킹 유형 및 대응

- **게임 앱 해킹**
  - 실제로 게임 앱을 해킹해 봅시다!

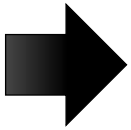
Android 모바일 해킹 유형 및 대응

- **환경 구성**



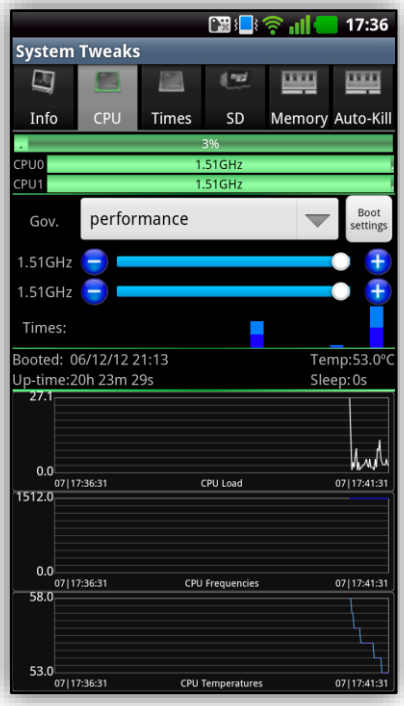
앱을 실행할 기기(실제 기기, 에뮬레이터)

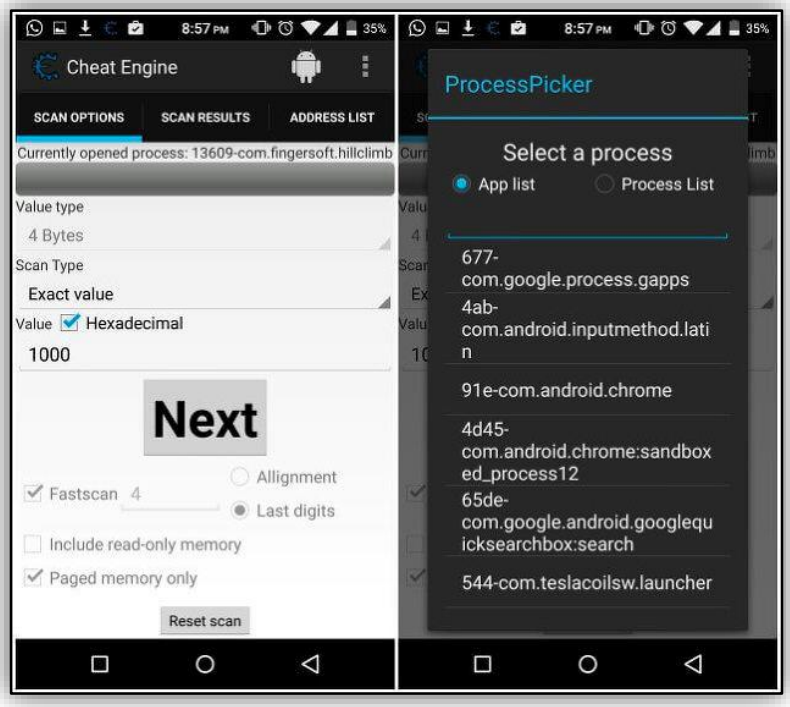
루팅



Android 모바일 해킹 유형 및 대응

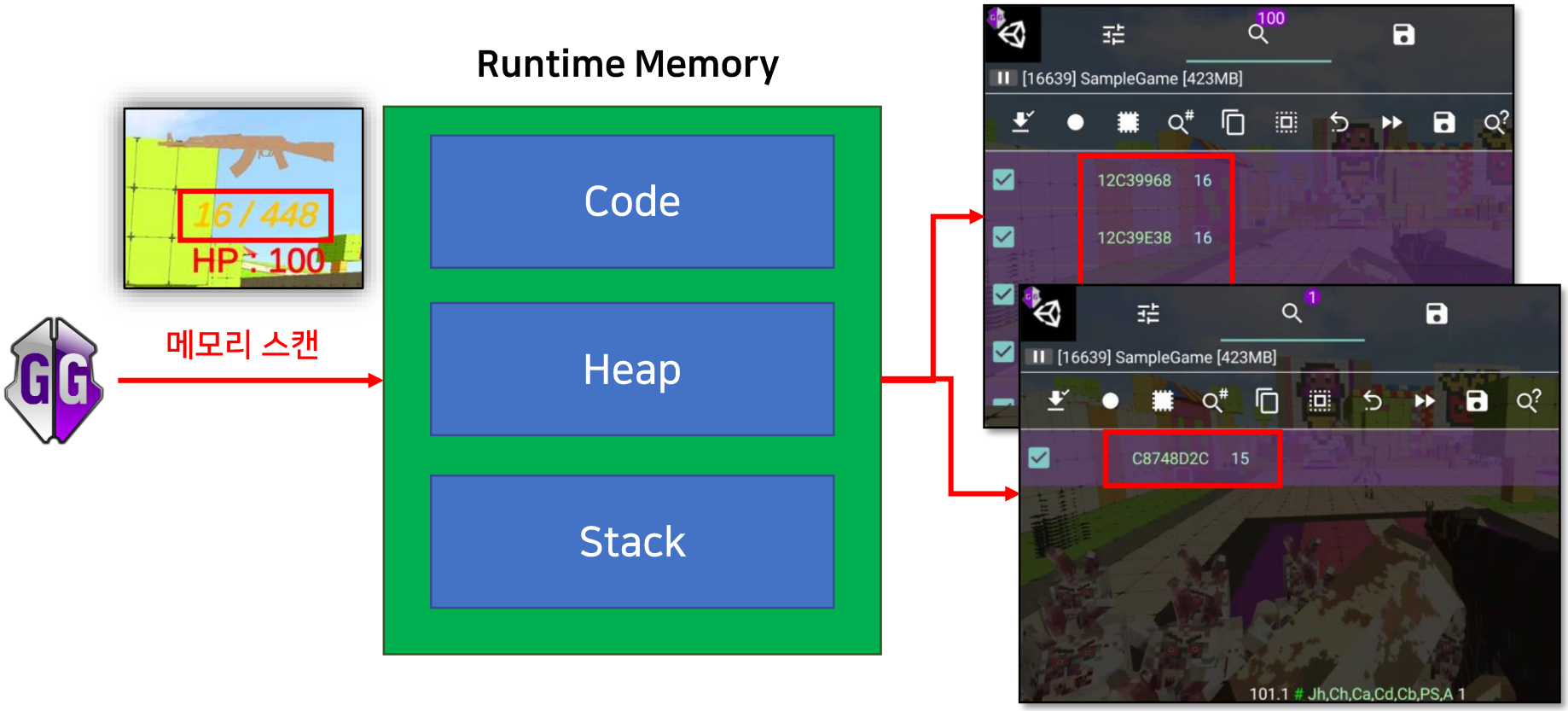
- **루팅의 목적?**
  - 주어진 권한 이상으로 뭔가를 하기 위함





Android 모바일 해킹 유형 및 대응

- **치팅툴**
  - 어떻게 동작할까?

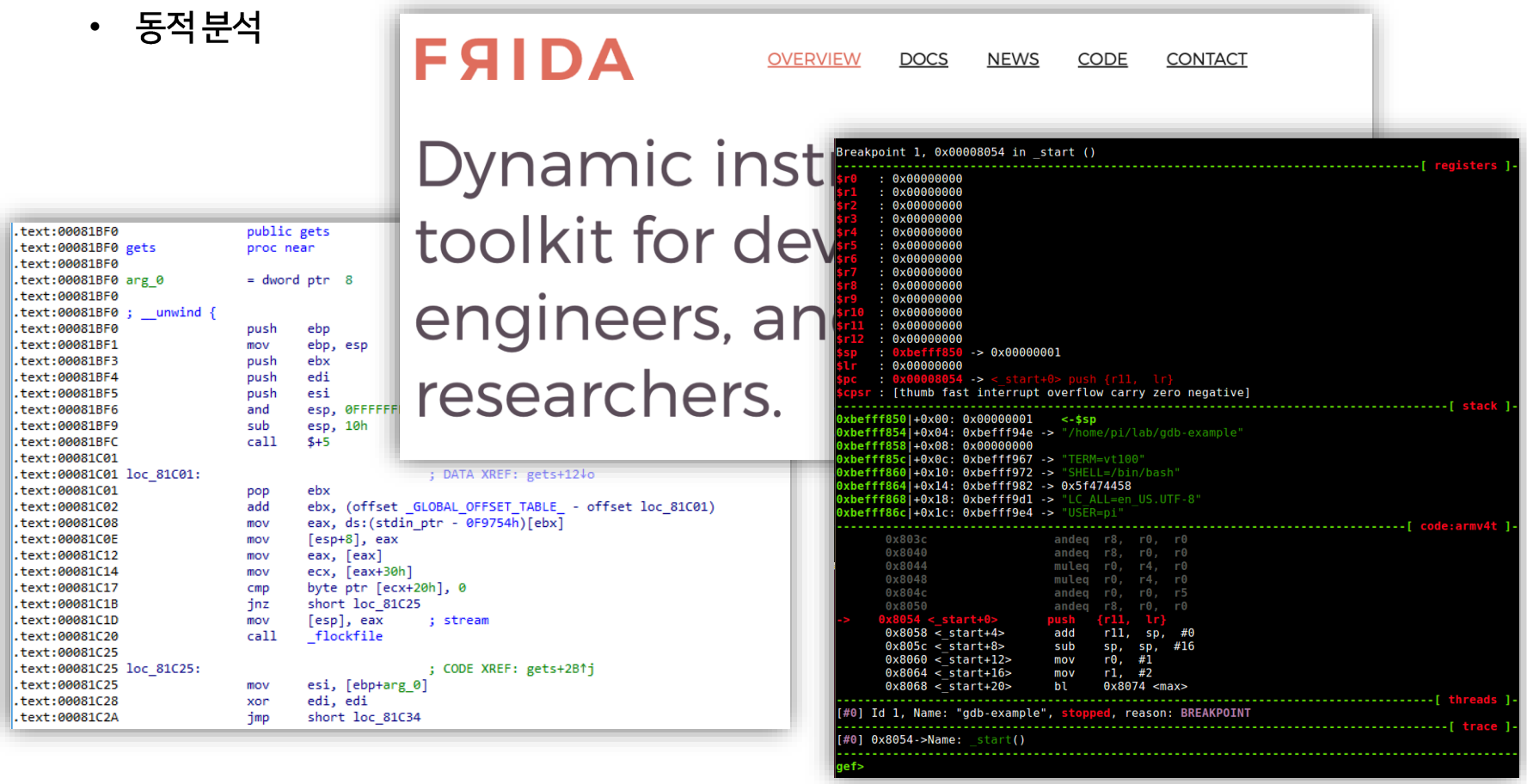


Android 모바일 해킹 유형 및 대응

- **메모리 변조**

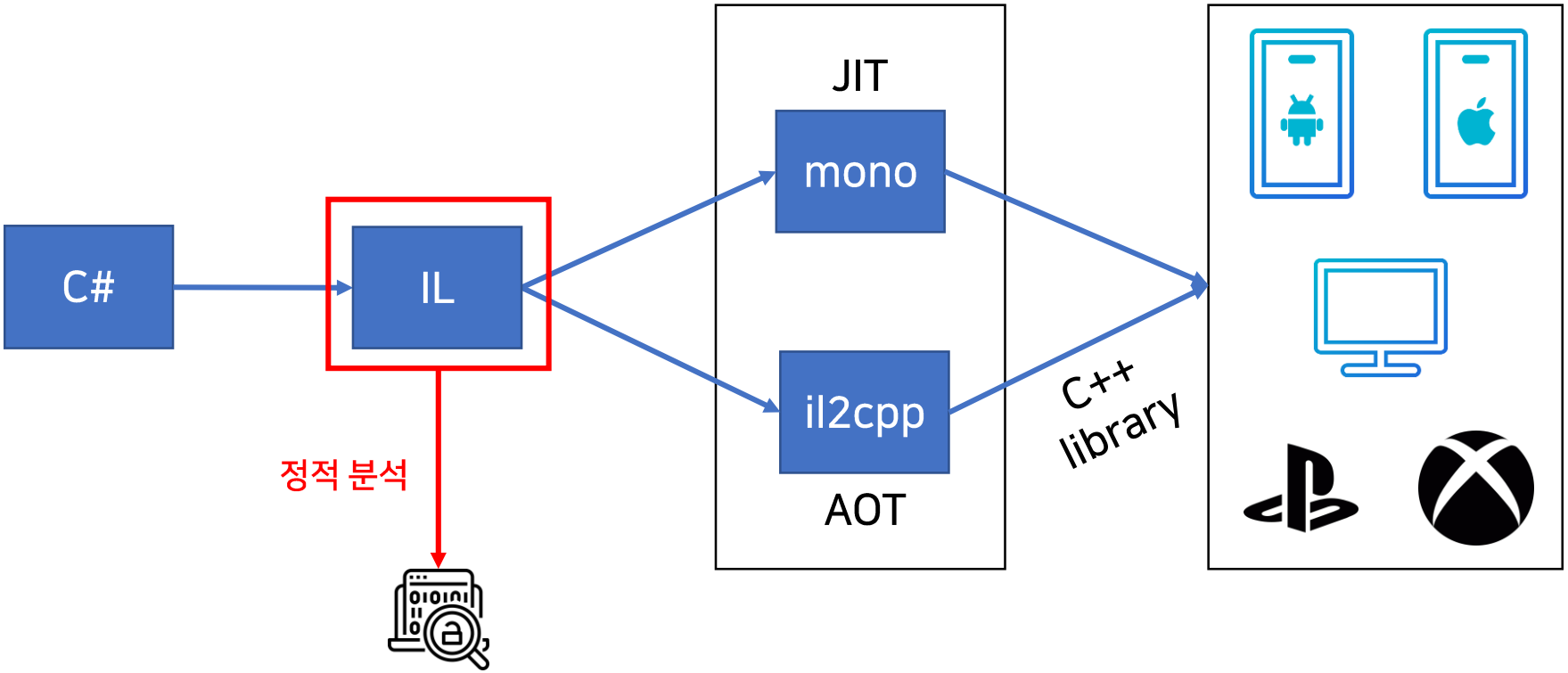
Android 모바일 해킹 유형 및 대응

- **두 가지 분석 방법**
  - 정적 분석
  - 동적 분석



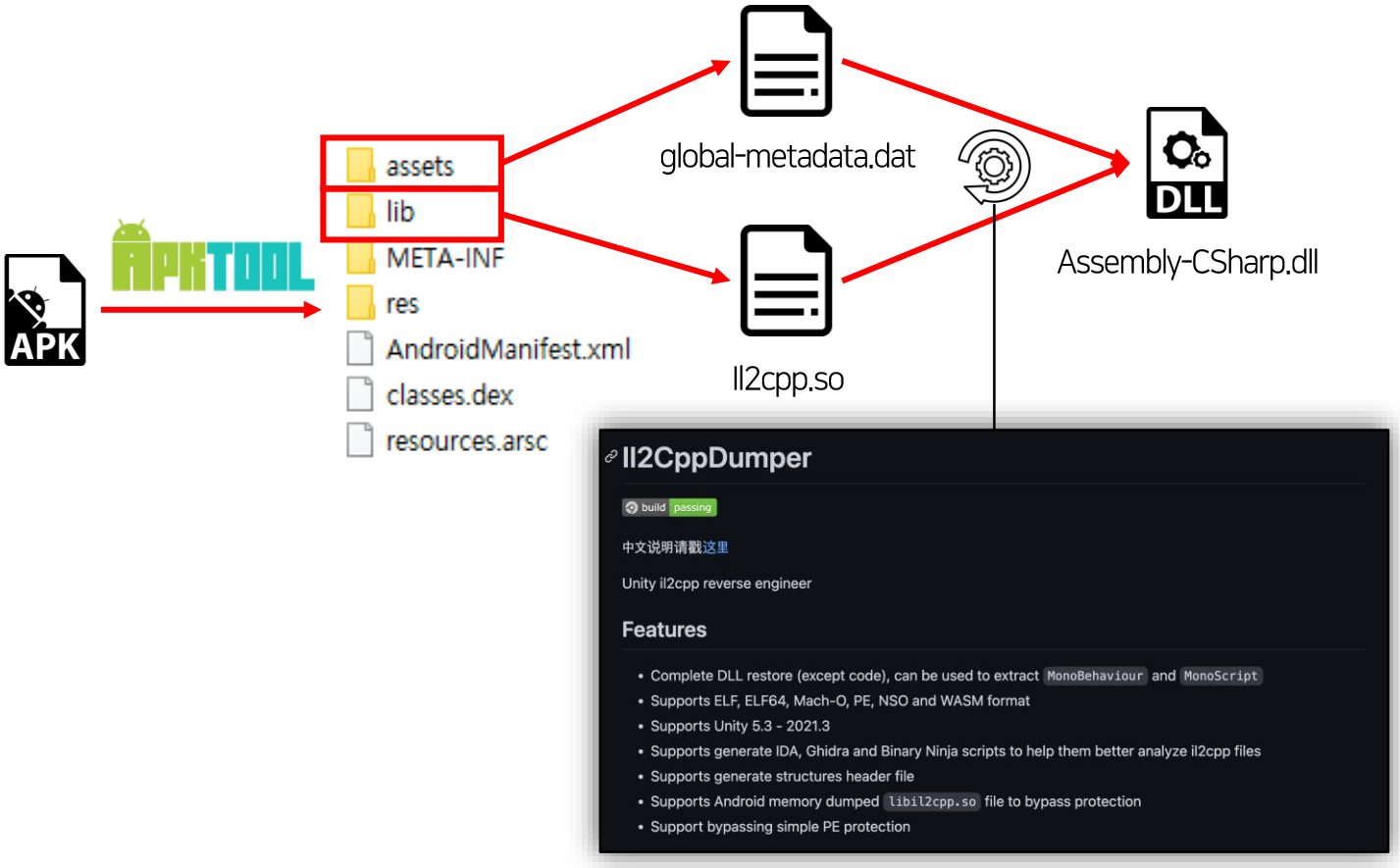
Android 모바일 해킹 유형 및 대응

- 유니티 엔진 구조



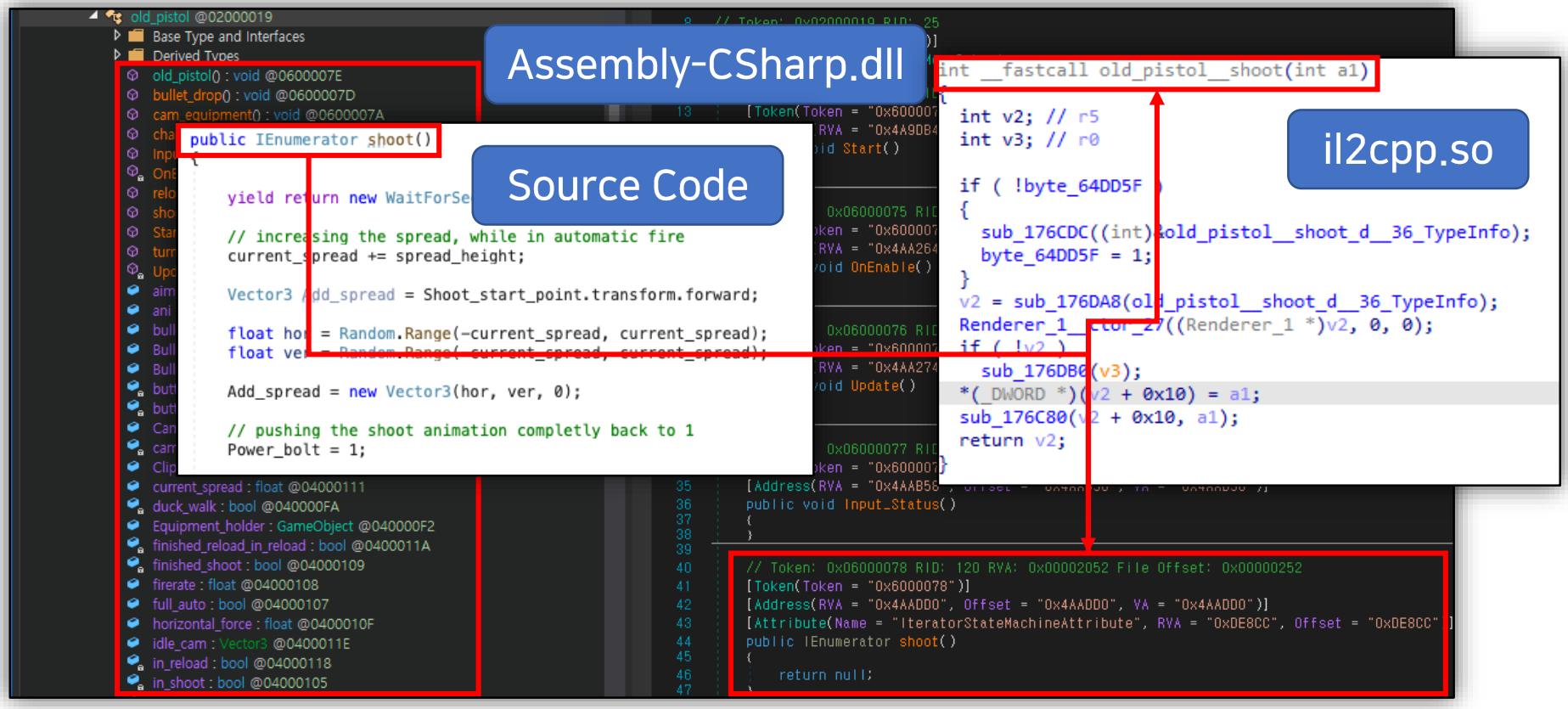
Android 모바일 해킹 유형 및 대응

- **정적 분석**
  - Assembly-CSharp.dll 추출



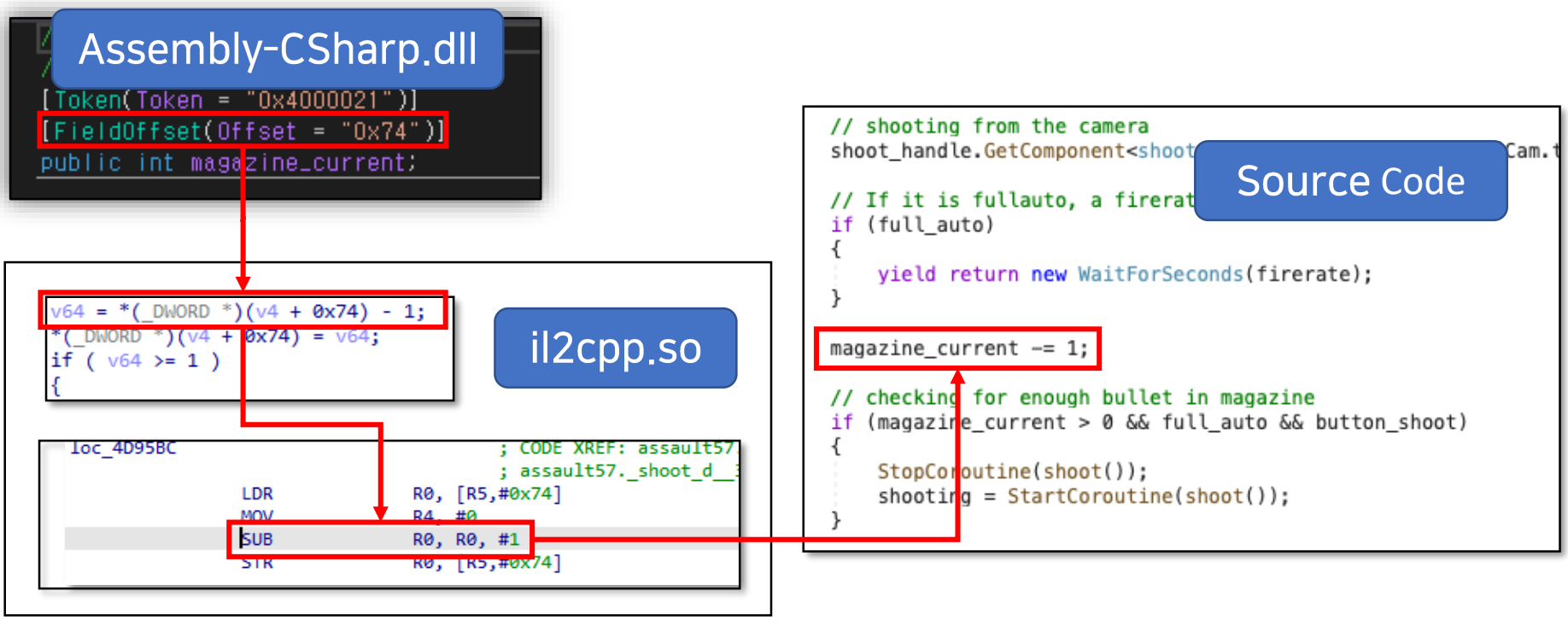
Android 모바일 해킹 유형 및 대응

- **정적 분석**
  - Assembly-CSharp.dll 디컴파일



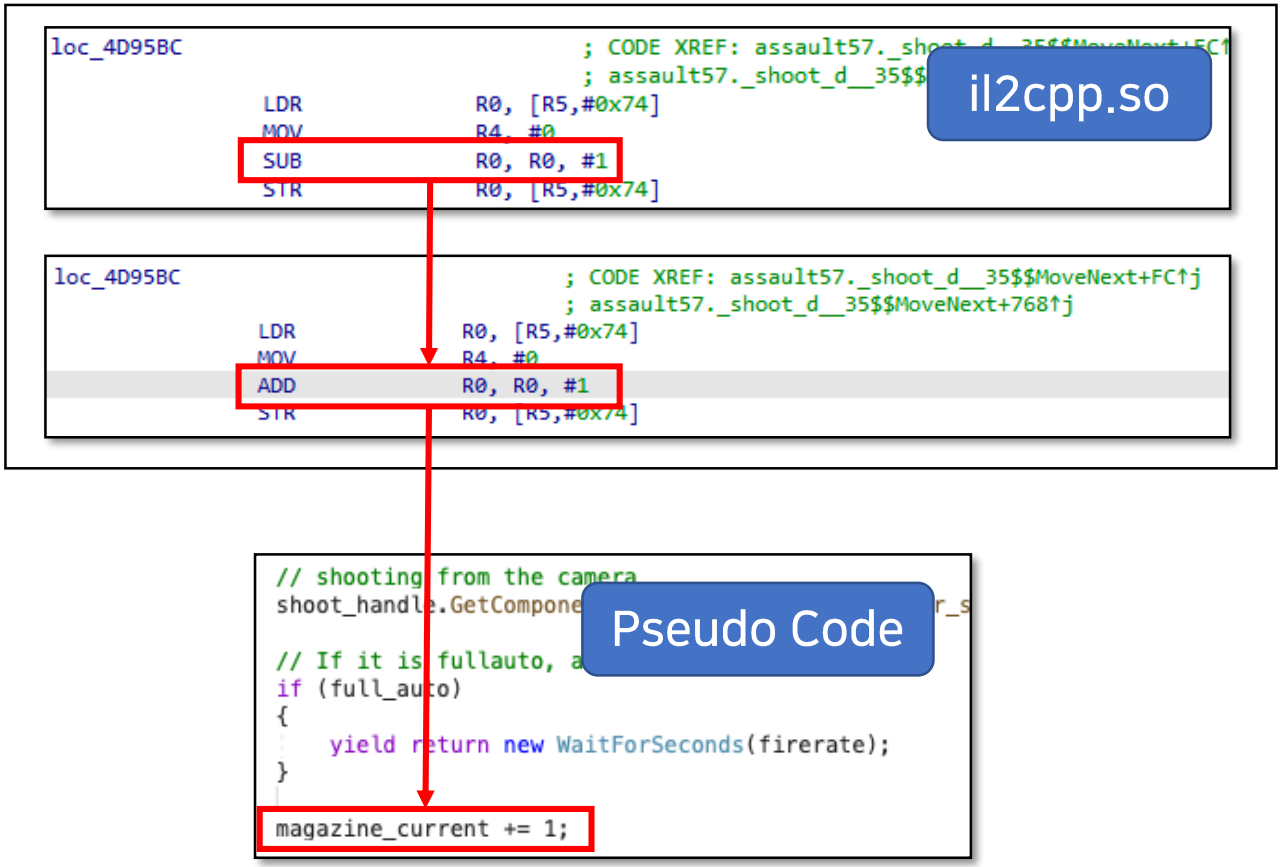
Android 모바일 해킹 유형 및 대응

- **라이브러리 변조**
  - 탄창의 총알을 발사할수록 증가하게 해볼까?



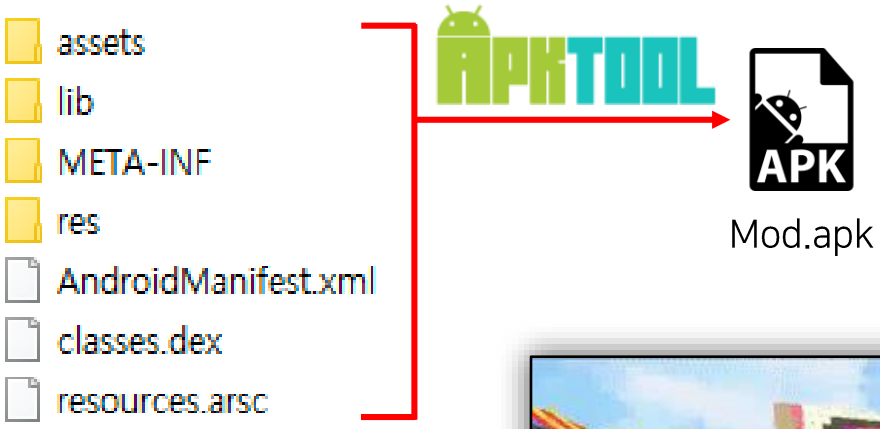
Android 모바일 해킹 유형 및 대응

- **라이브러리 변조**
  - 실행 코드 변조



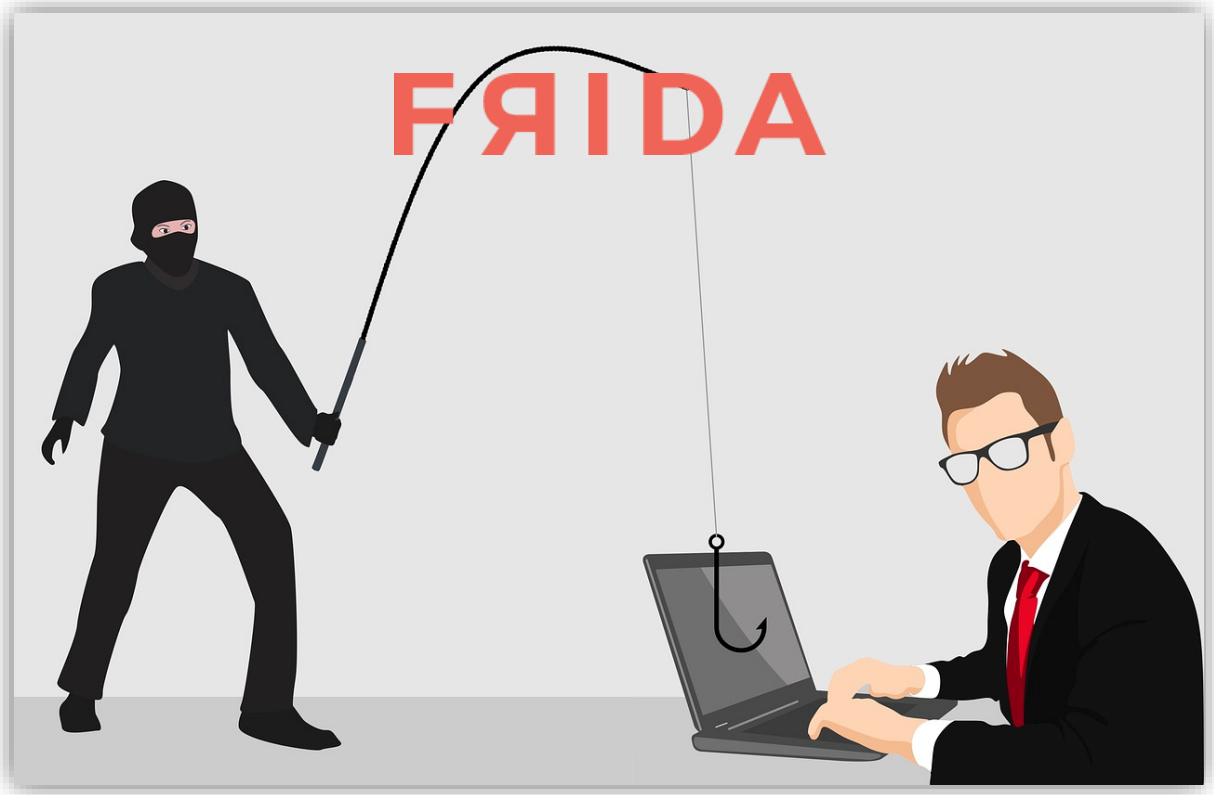
Android 모바일 해킹 유형 및 대응

- **리패키징**



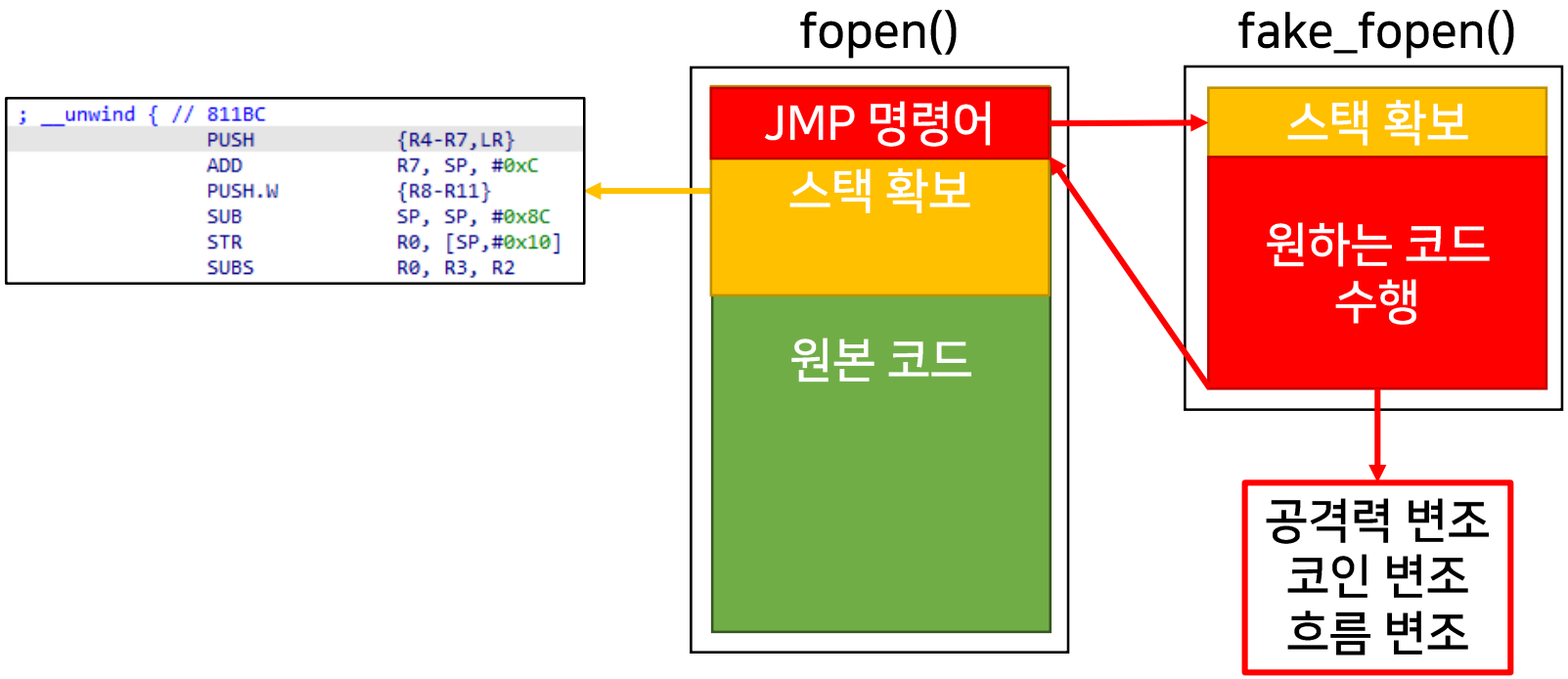
Android 모바일 해킹 유형 및 대응

- **동적 분석**
  - Frida
    - Dynamic Binary Instrumentation tookit



Android 모바일 해킹 유형 및 대응

- **동적 분석**
  - 프리다 후킹 동작 원리
    - 인라인 후킹



Android 모바일 해킹 유형 및 대응

- 동적 분석

```
const base = Module.findBaseAddress("libil2cpp.so")
Interceptor.attach(base.add(0x4d95c4), {
    onEnter(args) {
        console.log(JSON.stringify(this.context));
    },
    onLeave(retval) {
    }
});
```

```
il2cpp:004D95BC
il2cpp:004D95BC loc_4D95BC
il2cpp:004D95BC LDR             R0, [R5,#0x74]
il2cpp:004D95C0 MOV             R4, #0
il2cpp:004D95C4 SUB             R0, R0, #1
il2cpp:004D95C8 STR             R0, [R5,#0x74]
```

```
{"pc":"0xcd5005c4","sp":"0xcd0252f8","r0":"0x1b","r1":"0xbb914e48",
{"pc":"0xcd5005c4","sp":"0xcd0252f8","r0":"0x1a","r1":"0xbb914e48",
{"pc":"0xcd5005c4","sp":"0xcd0252f8","r0":"0x19","r1":"0xbb914e48",
```

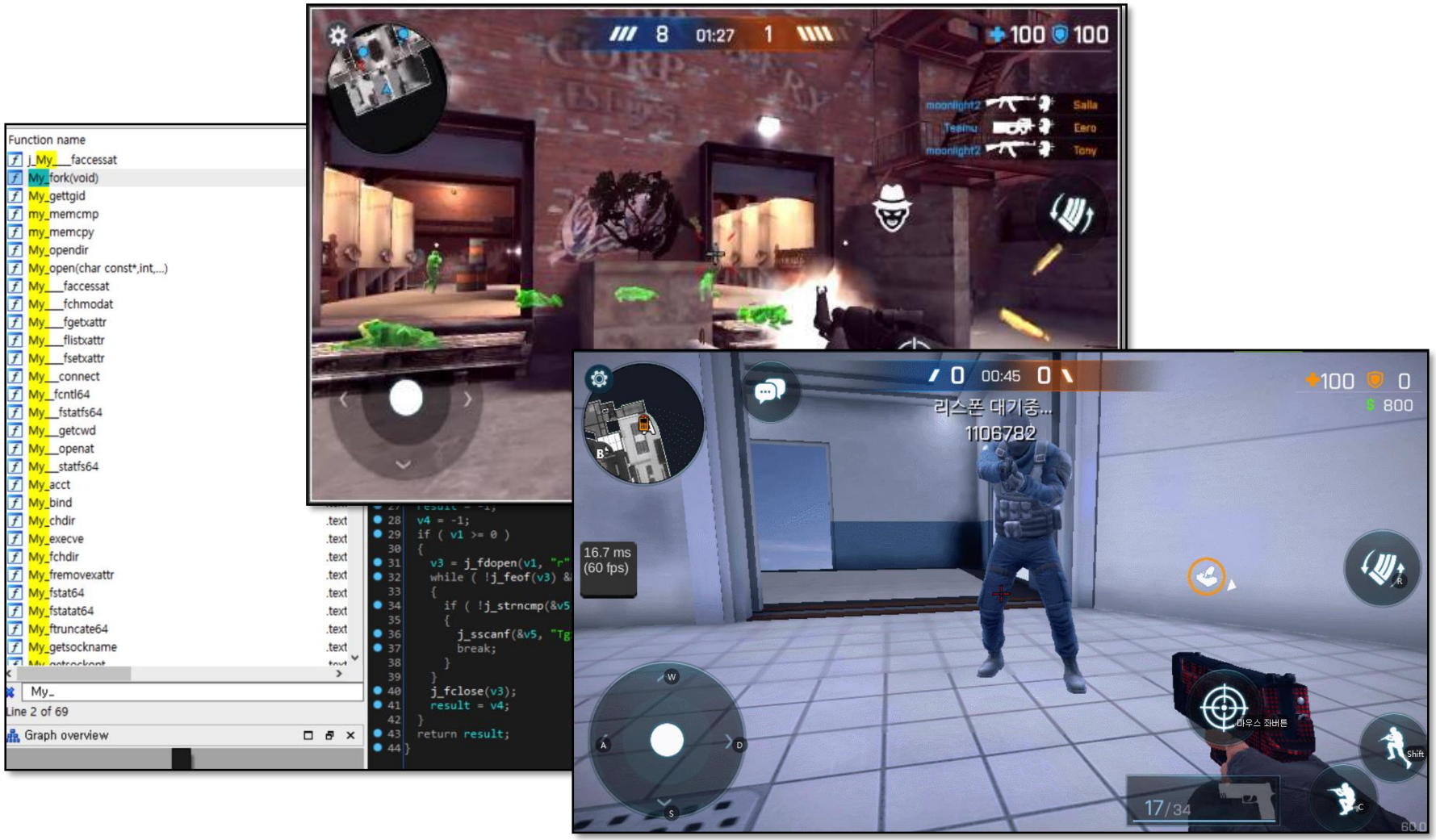
Android 모바일 해킹 유형 및 대응

- 동적 분석

```
const base = Module.findBaseAddress("libil2cpp.so")
Interceptor.attach(base.add(0x4d95c4), {
    onEnter(args) {
        this.context.r0 = 1000;
    },
    onLeave(retval) {
    }
});
```

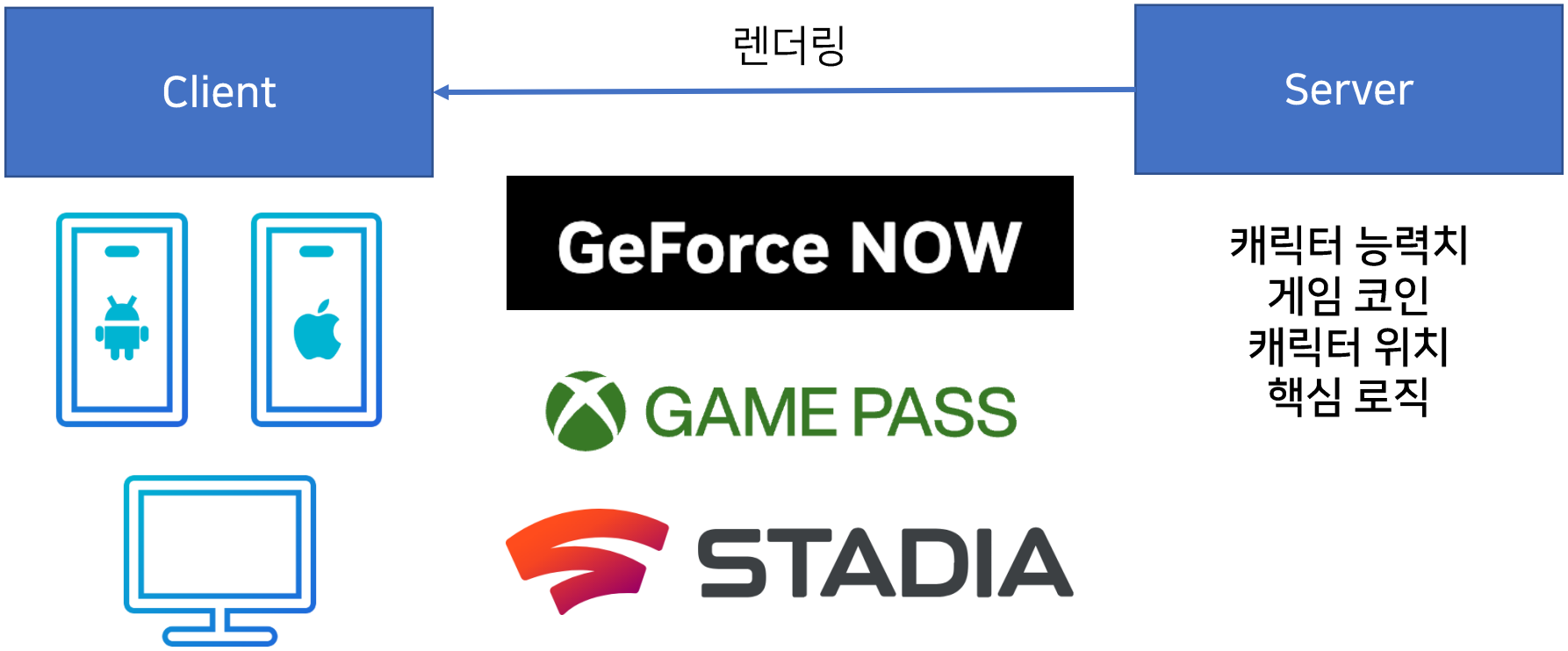
Android 모바일 해킹 유형 및 대응

- 이외에도 다양한 악의적인 행위들



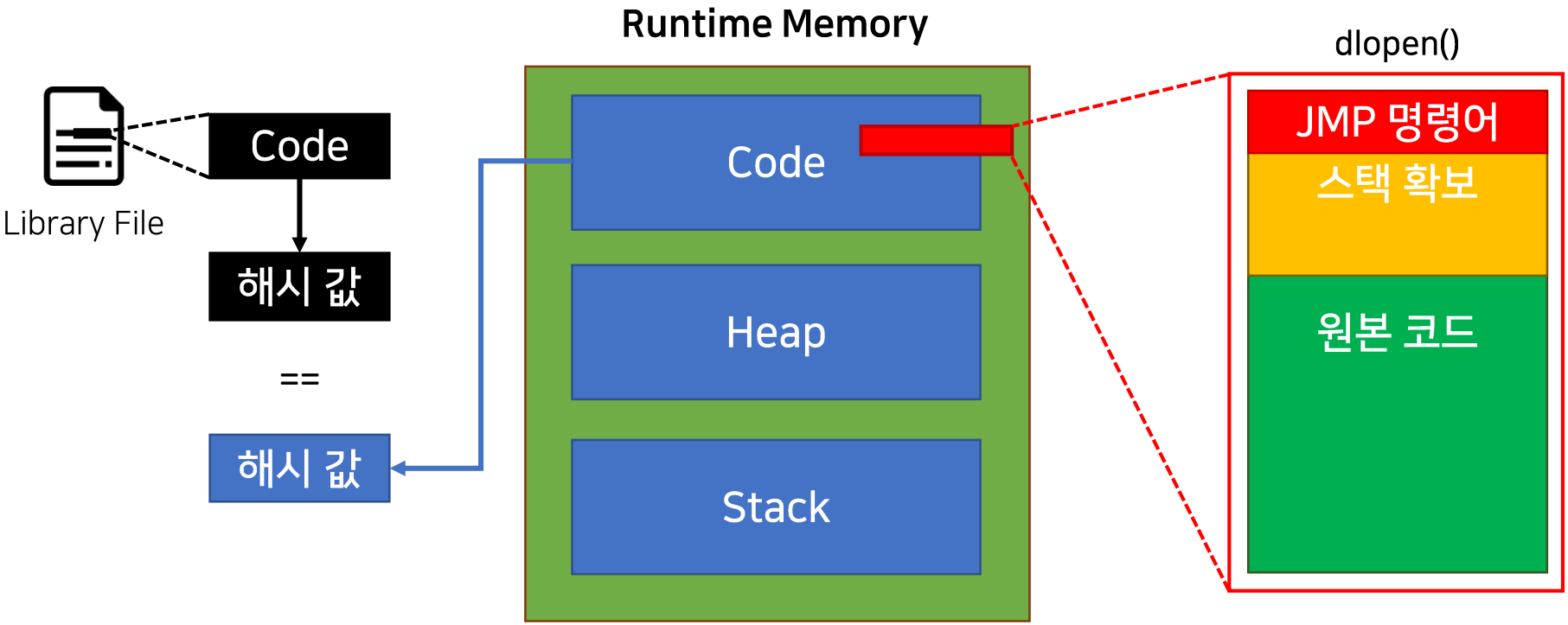
Android 모바일 해킹 유형 및 대응

- **대응 방안**
  - 서버 검증



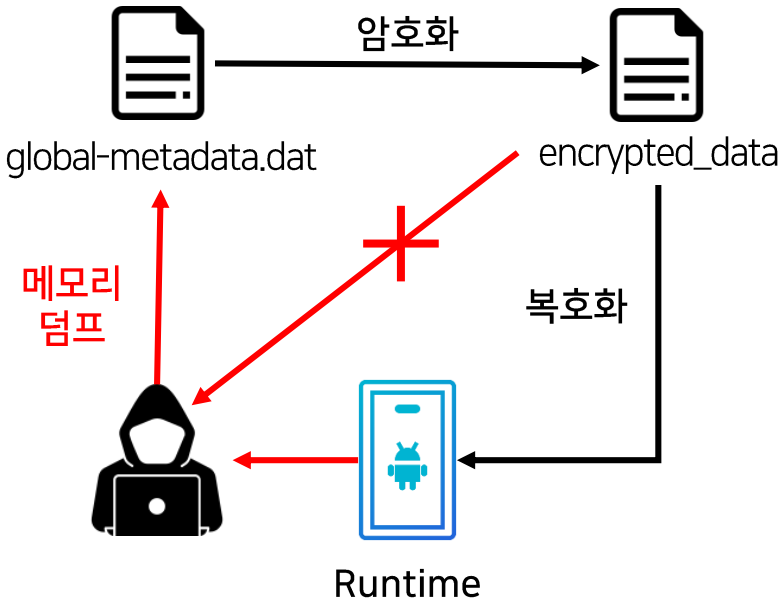
Android 모바일 해킹 유형 및 대응

- **대응 방안**
  - 라이브러리 코드 변조 & 인라인 후킹 탐지



Android 모바일 해킹 유형 및 대응

- **대응 방안**
  - 유니티 엔진 보호



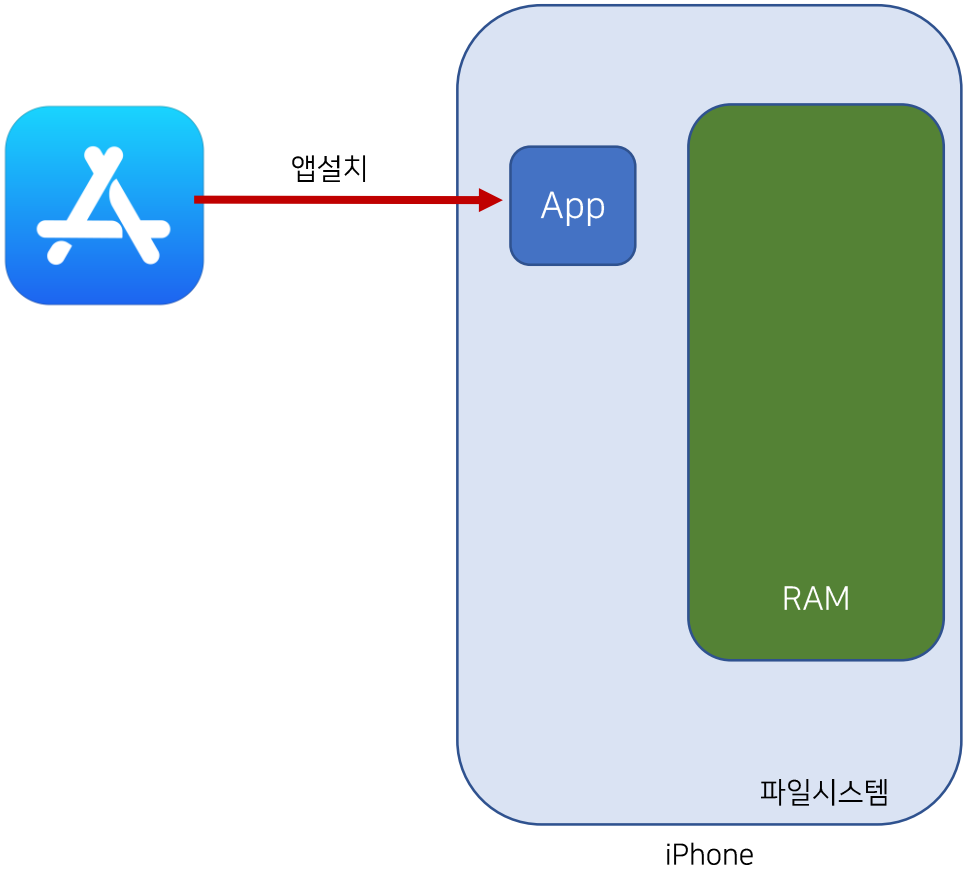

```
// Namespace:
public class AutomaticGunScriptLPFP : MonoBehaviour // TypeDefIndex: 3388
{
    // Fields
    private Animator anim; // 0x18
    [Header("Gun Camera")]
    public Camera gunCamera; // 0x20
    [Tooltip("How fast the camera field of view changes when aiming.")]
```

```
// Namespace:
public class oVkFrPXxWudEoOkXUOqEZm : MonoBehaviour // TypeDefIndex: 3388
{
    // Fields
    private Animator kMvQ; // 0xC
    [Header("Gun Camera")]
    public Camera haptFYeug; // 0x10
    [Tooltip("How fast the camera field of view changes when aiming.")]
    [Header("Gun Camera Options")]
    public float hOGOanpb; // 0x14
    [Tooltip("Default value for camera field of view (40 is recommended).")]
    public float EGnBBTAXGd; // 0x18
    public float NghVps; // 0x1C
    [Tooltip("Name of the current weapon, shown in the game UI.")]
    [Header("UI Weapon Name")]
    public string FhgbpmHlpb; // 0x20
    private string bahrAcKwSsZWOzMx; // 0x24
    [Tooltip("Toggle weapon sway.")]
    [Header("Weapon Sway")]
    public bool guAfdGmtxN; // 0x28
    public float qhhbOqgpha; // 0x2C
    public float MLstBtmPGrzGT; // 0x30
    public float JQJuCiDbYxYyZmr; // 0x34
    private Vector3 roeRFroLenlnpuLipkT; // 0x38
```

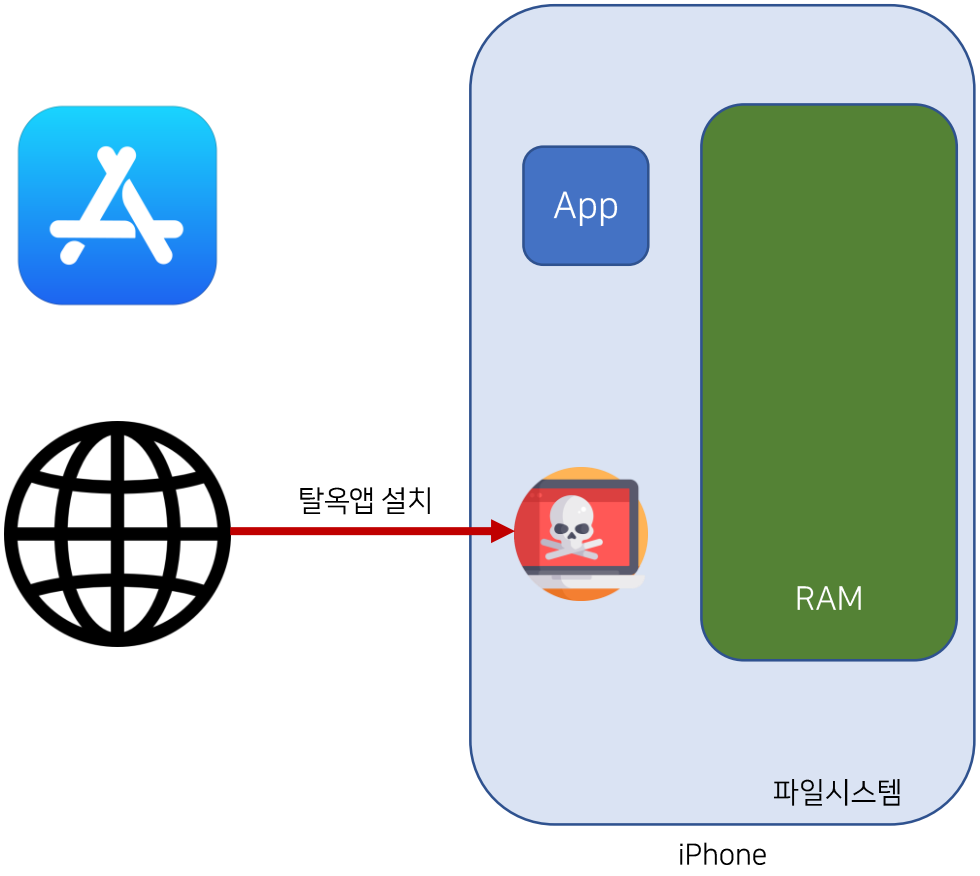
iOS 모바일 해킹 유형 및 대응

- 해킹 유형
- 대응 방안

iOS 모바일 해킹 유형: 탈옥



iOS 모바일 해킹 유형: 탈옥



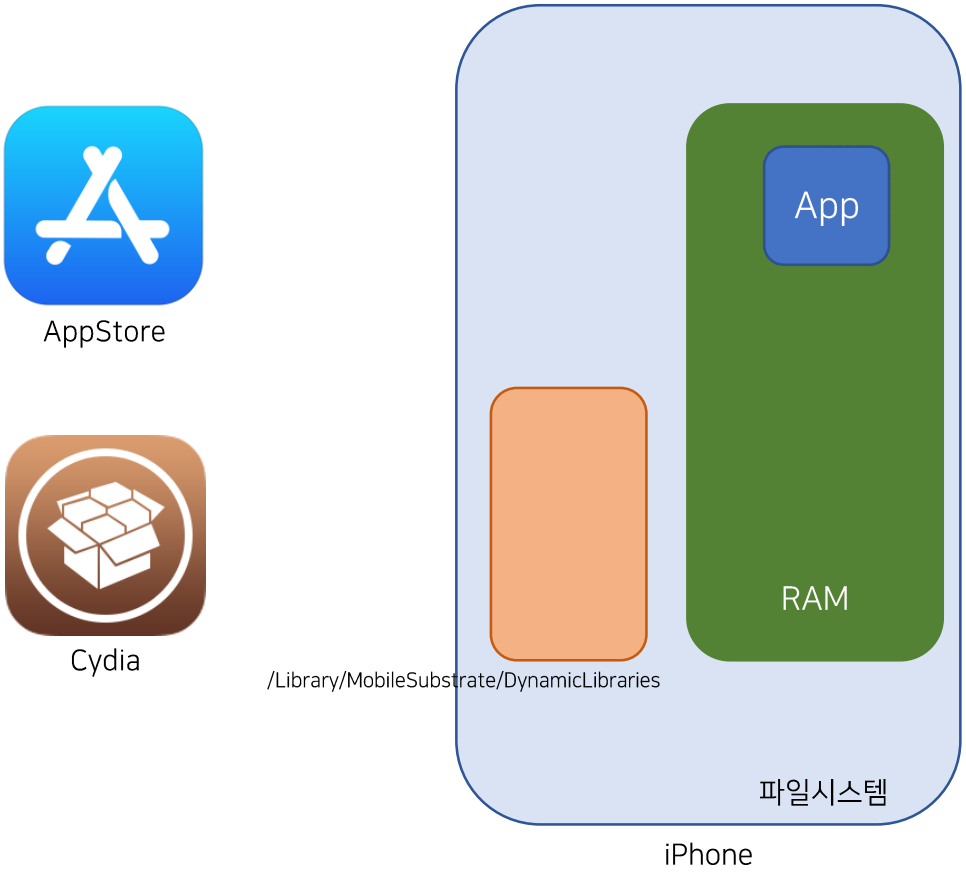
iOS 모바일 해킹 유형: 탈옥

iOS 모바일 해킹 유형: 탈옥

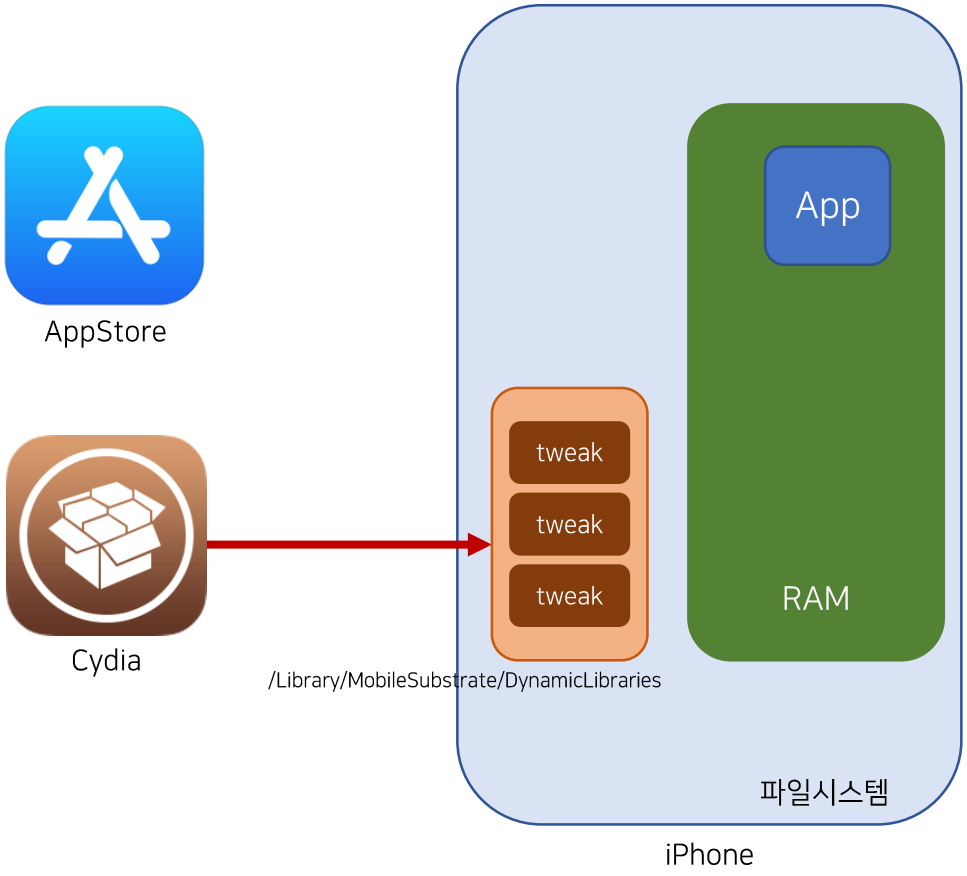
WHY ?

- ROOT 권한을 얻기 위해
- 제한된 곳에 파일 read/write 가능
- 앱 실행 시 임의의 동적 라이브러리를 load
- 앱 해킹을 위한 디버깅이 용이함
- 탈옥은 iOS 앱 해킹을 위한 기본적인 절차

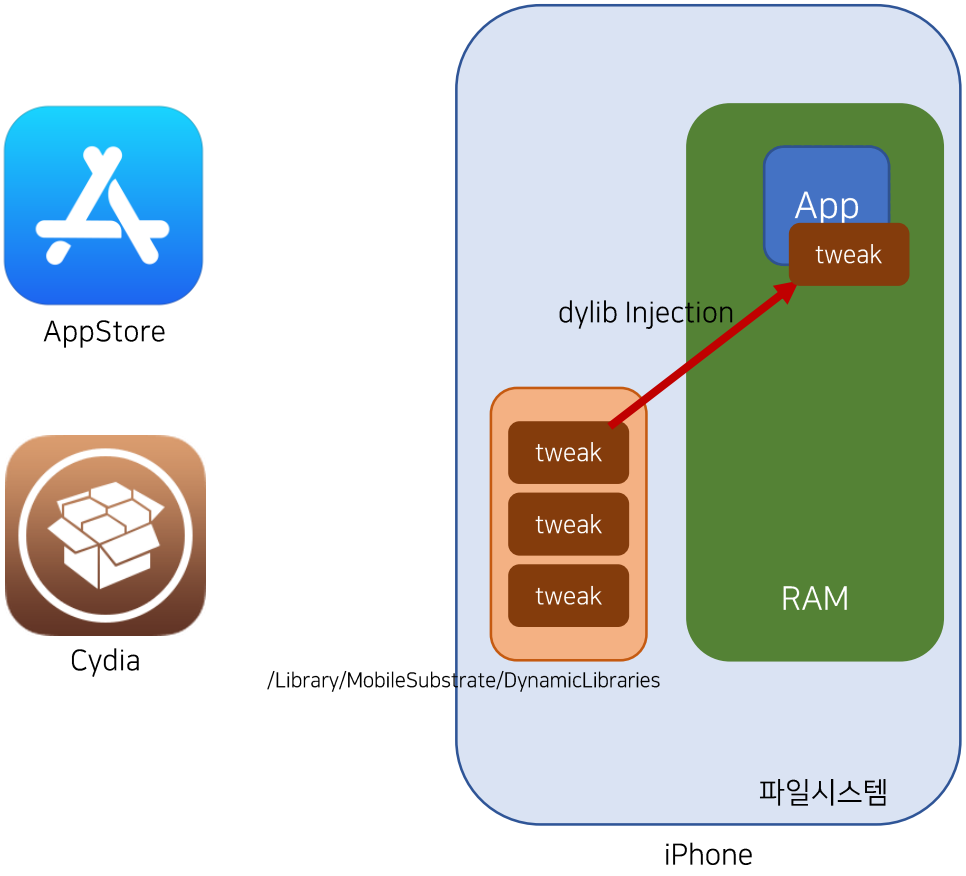
iOS 모바일 해킹 유형: tweak (dylib injection)



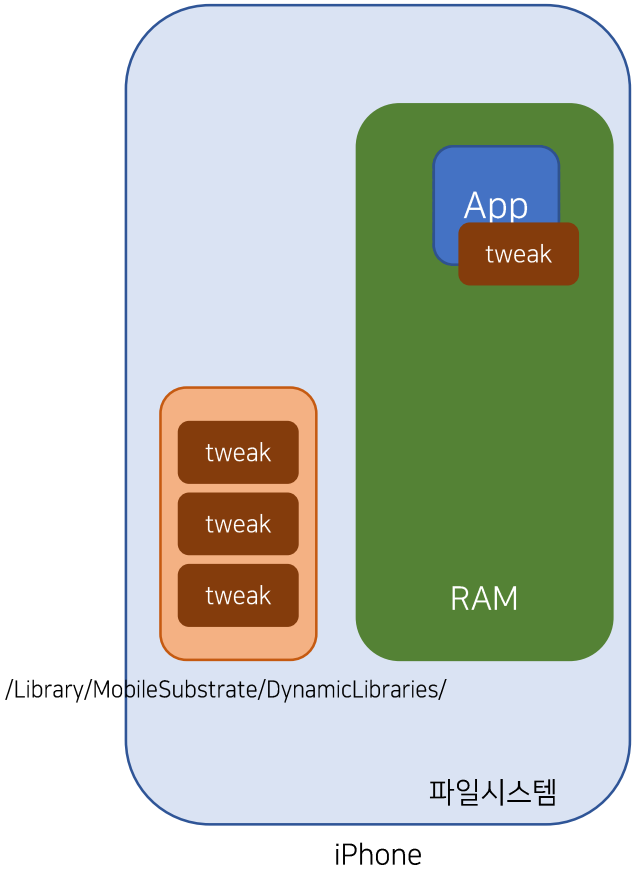
iOS 모바일 해킹 유형: tweak (dylib injection)



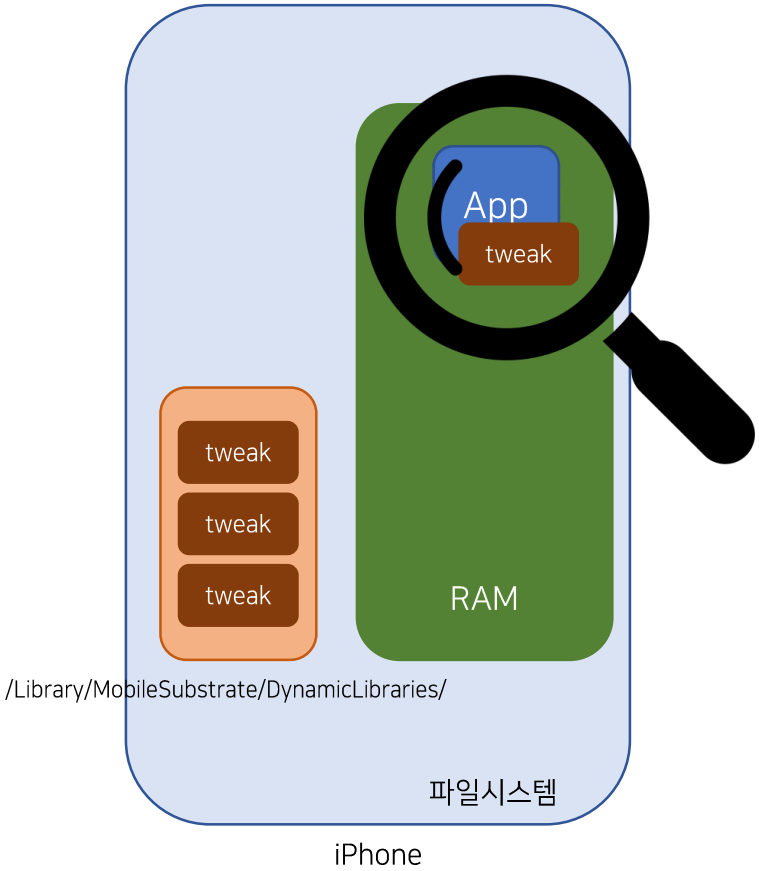
iOS 모바일 해킹 유형: tweak (dylib injection)



iOS 모바일 해킹 유형: tweak (dylib injection)

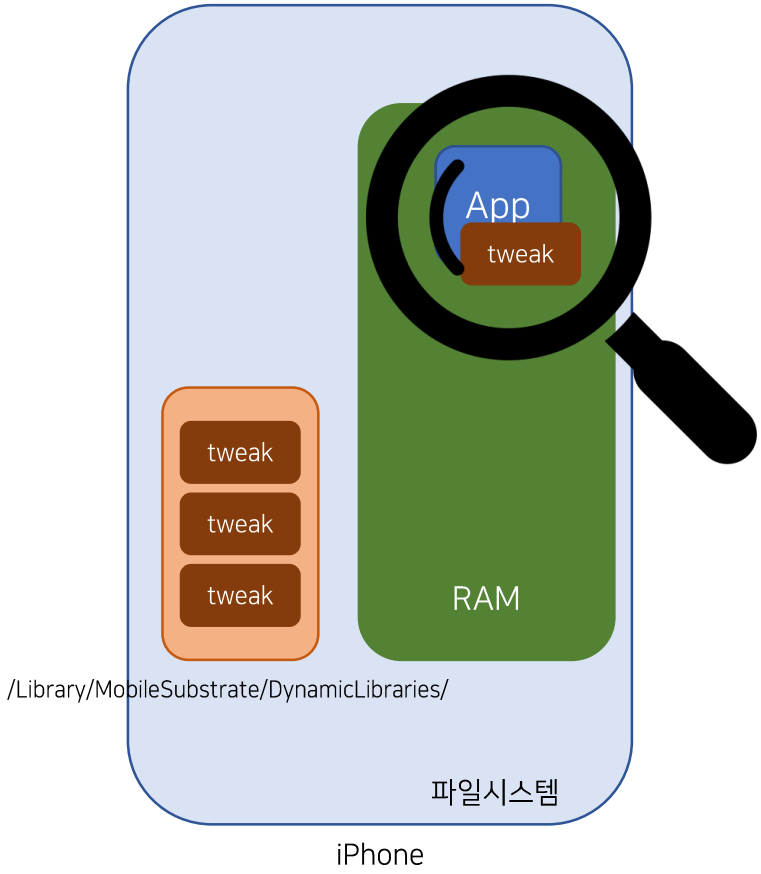


iOS 모바일 해킹 유형: tweak (dylib injection)




```
void attack(int damage) {
    enemy.reduceHP(damage)
    ...
    ...
}
```

iOS 모바일 해킹 유형: tweak (dylib injection)

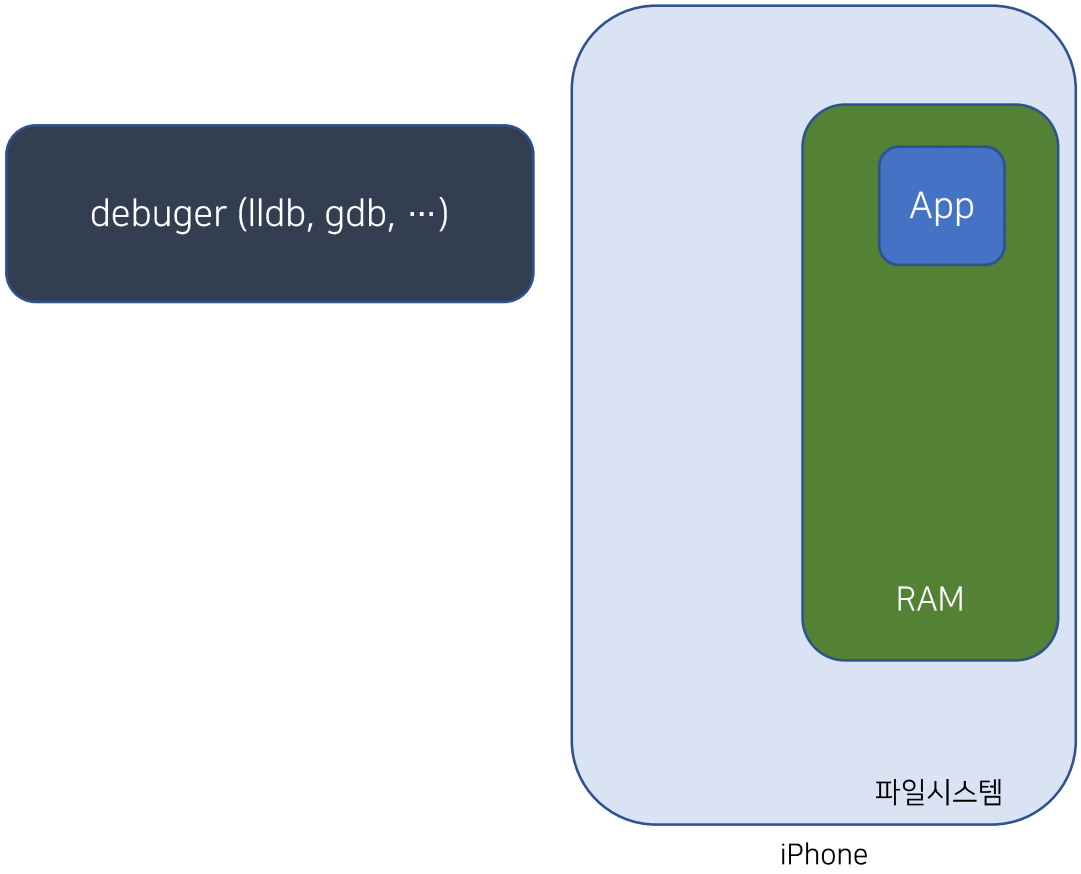



```
void attack(int damage) {
    enemy.reduceHP(damage)
    ...
    ...
}
```

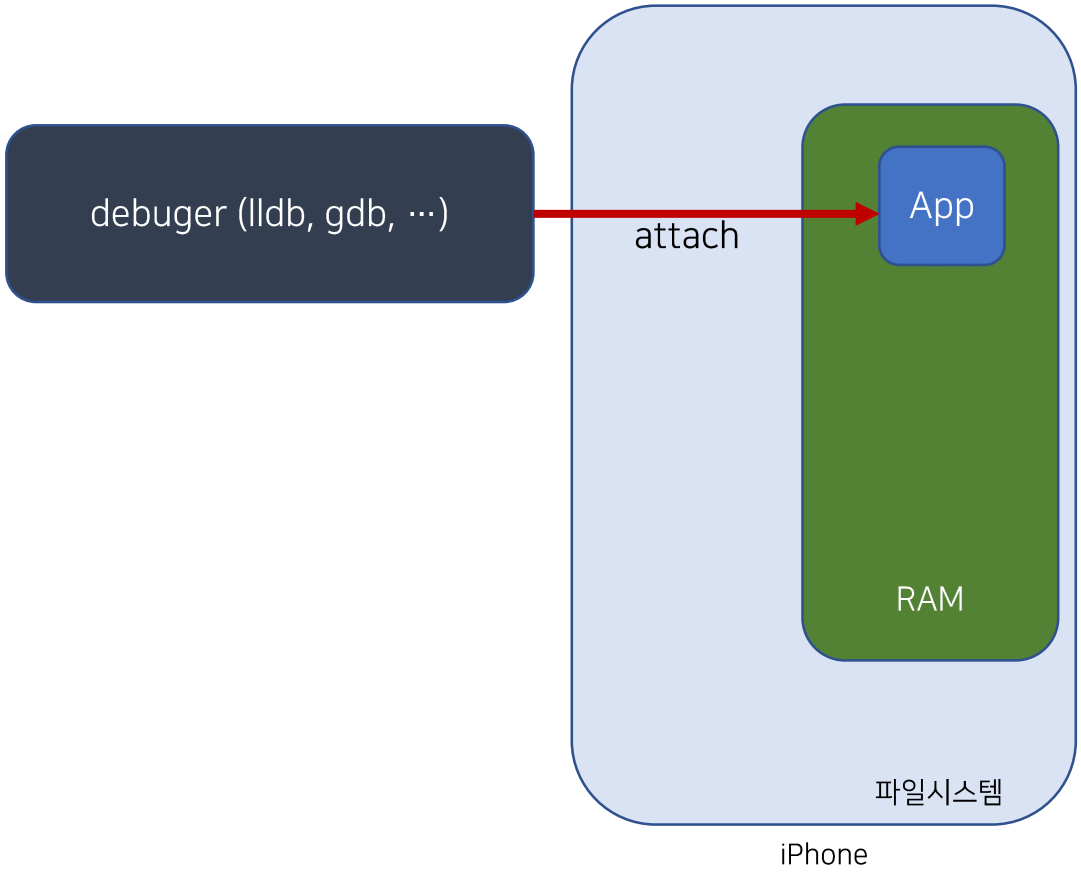
함수 후킹을 통해 파라미터를 변조

```
void attack(int damage) {
    damage = 99999999
    enemy.reduceHP(damage)
    ...
    ...
}
```

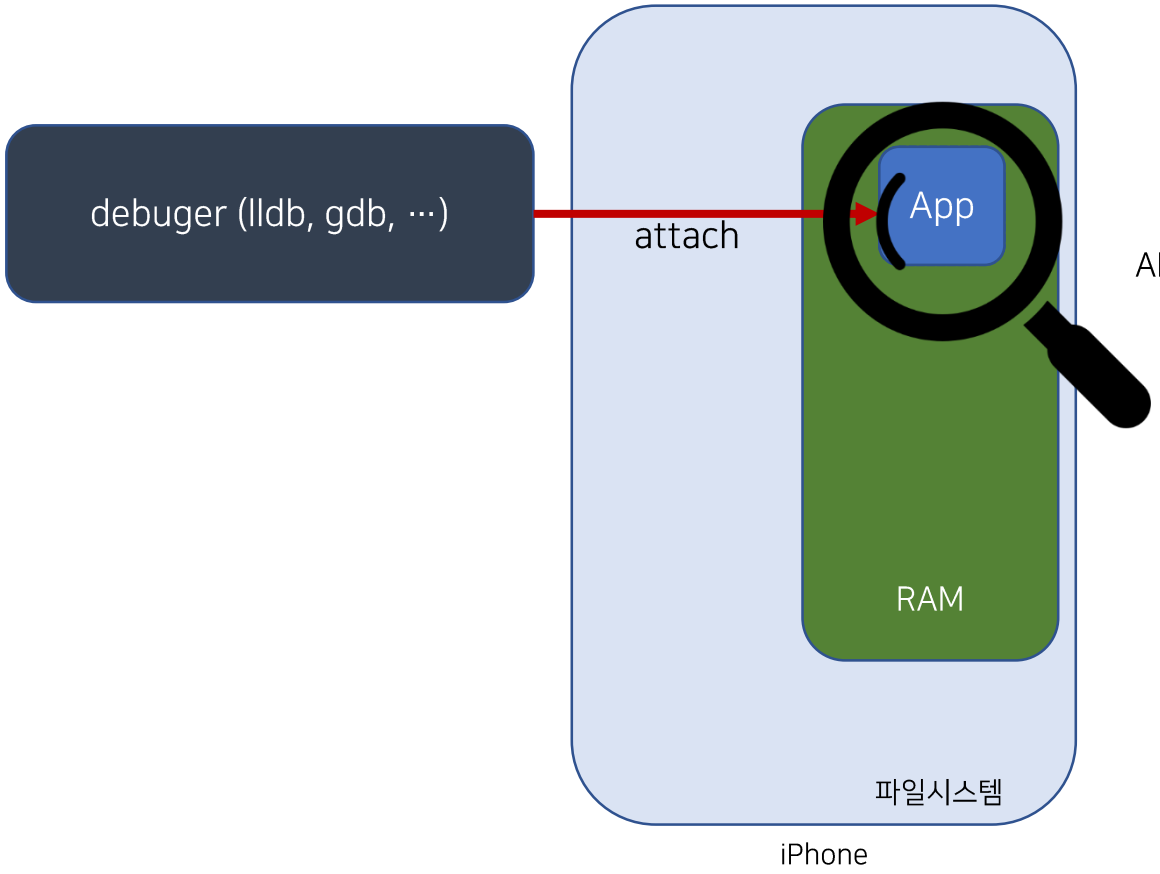
iOS 모바일 해킹 유형: 디버깅



iOS 모바일 해킹 유형: 디버깅

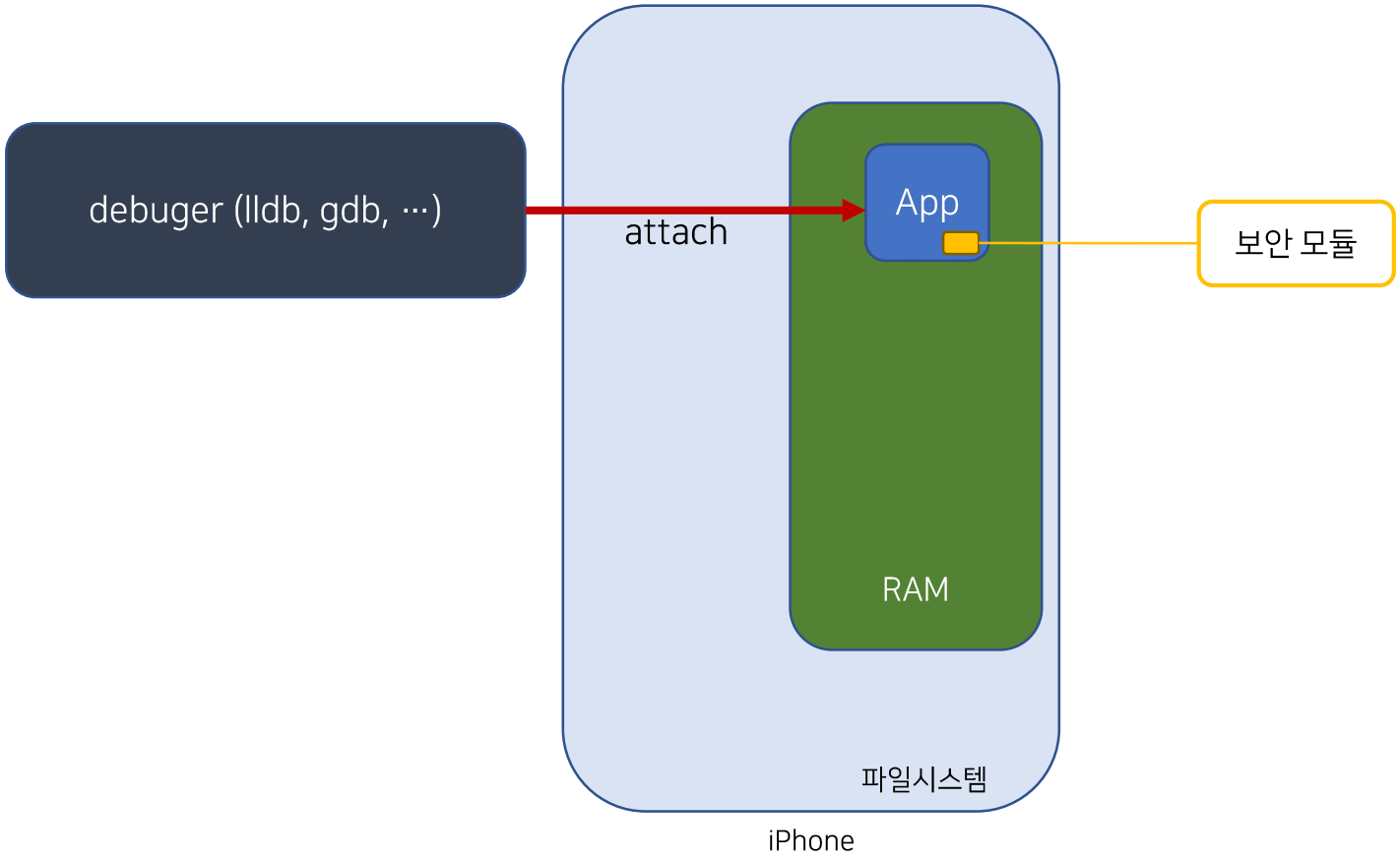


iOS 모바일 해킹 유형: 디버깅

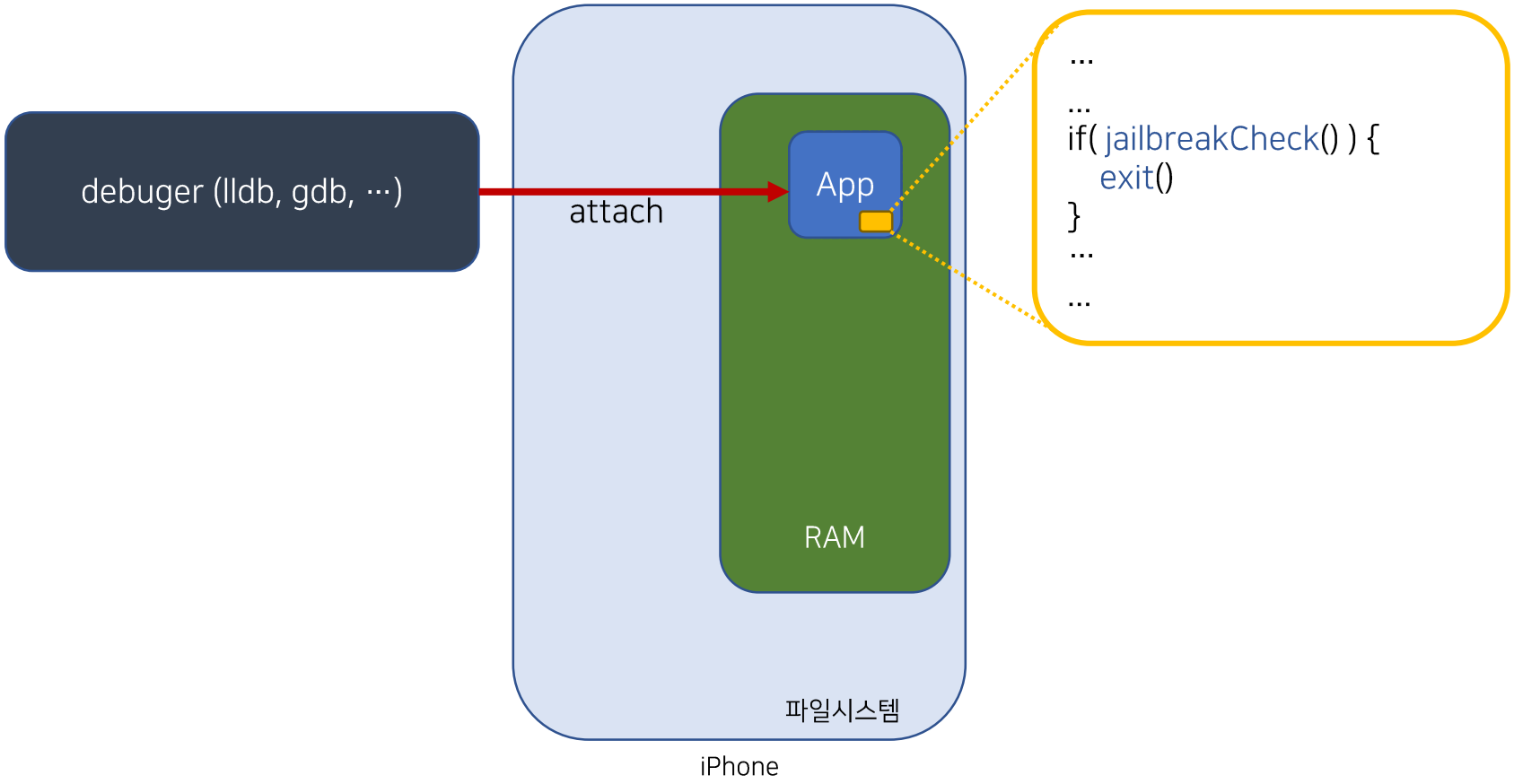


API에 break point를 설정하며 분석

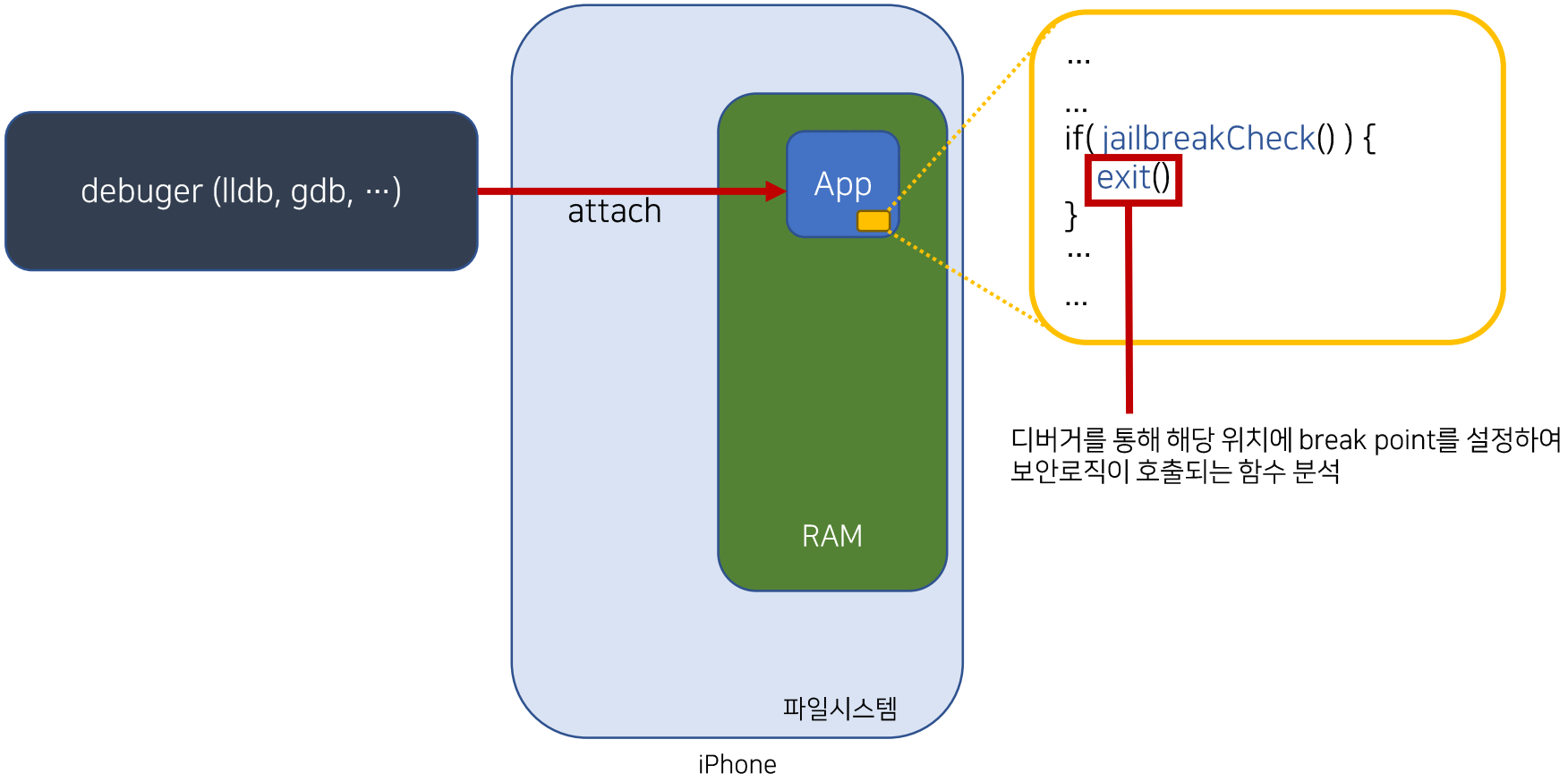
iOS 모바일 해킹 유형: 디버깅



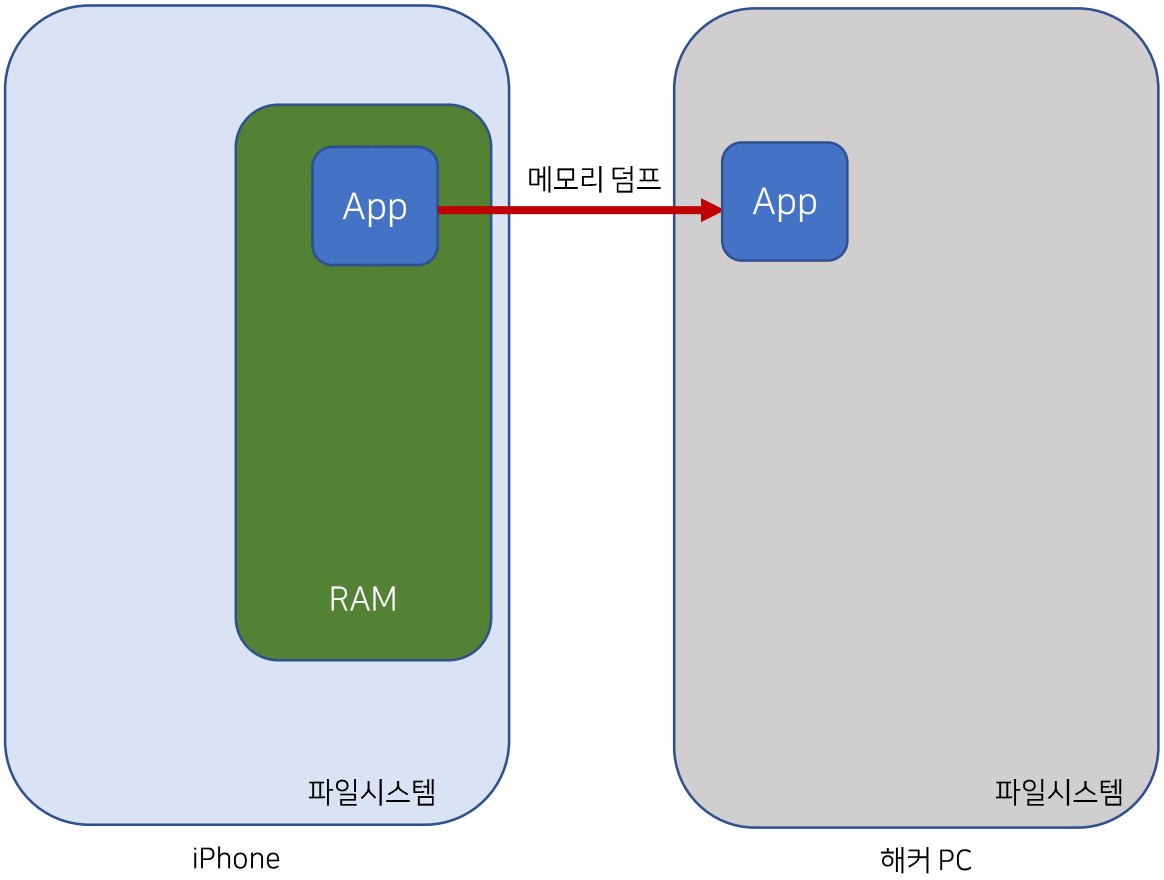
iOS 모바일 해킹 유형: 디버깅



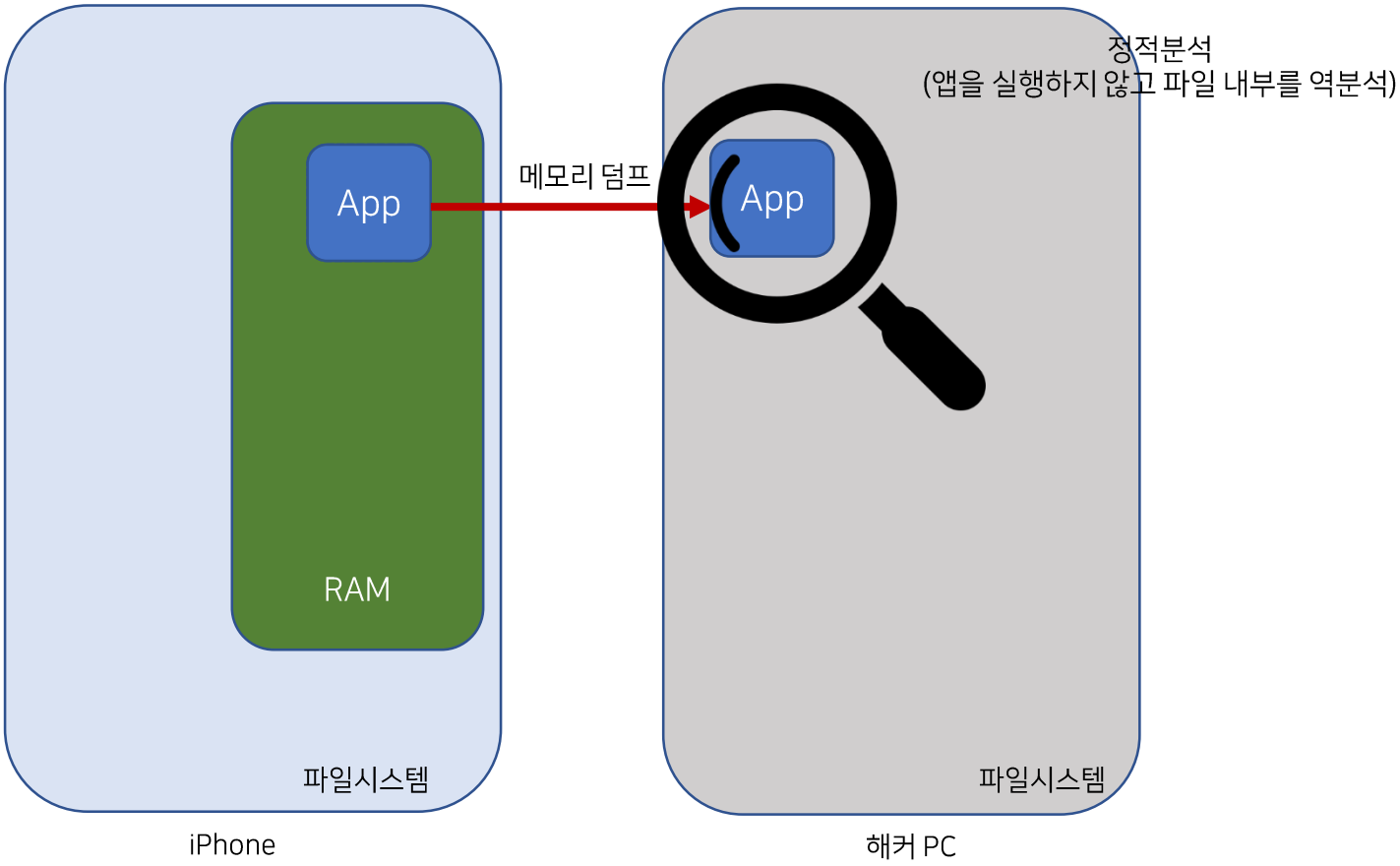
iOS 모바일 해킹 유형: 디버깅



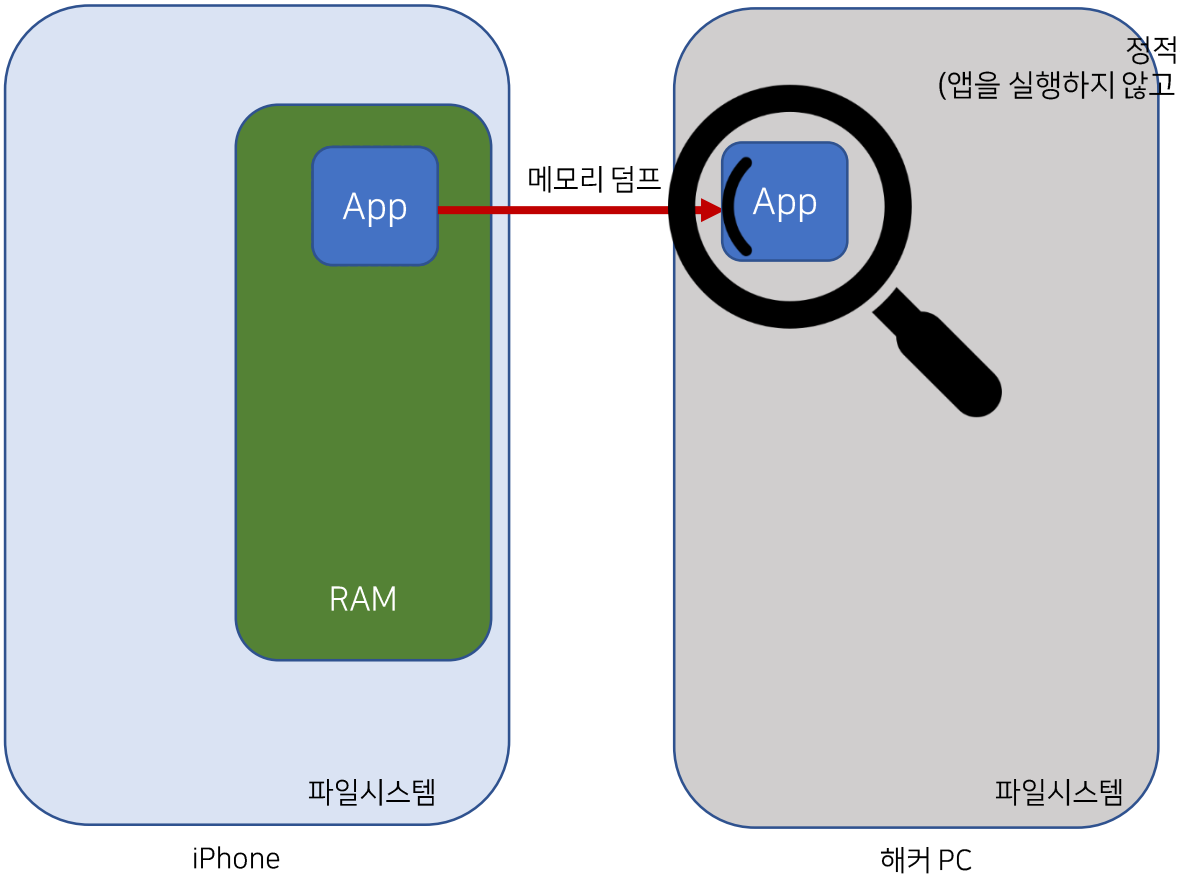
iOS 모바일 해킹 유형: 앱 위변조



iOS 모바일 해킹 유형: 앱 위변조



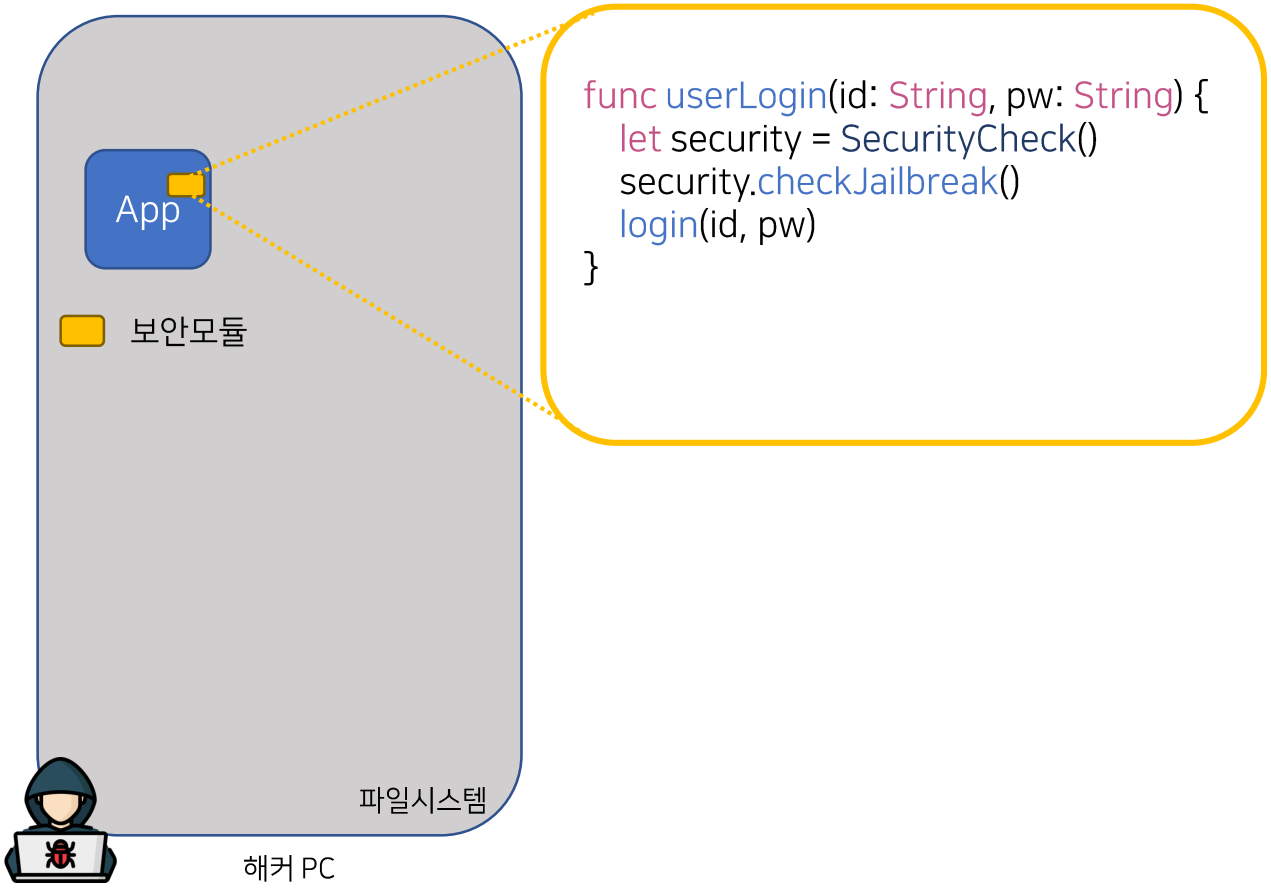
iOS 모바일 해킹 유형: 앱 위변조



1. 보안 모듈 우회 방안 분석
2. 변조를 통해 이득을 얻을 수 있는 포인트를 분석
3. 변조 방안 구체화
   - 어셈블리를 직접 패치
   - dylib injection

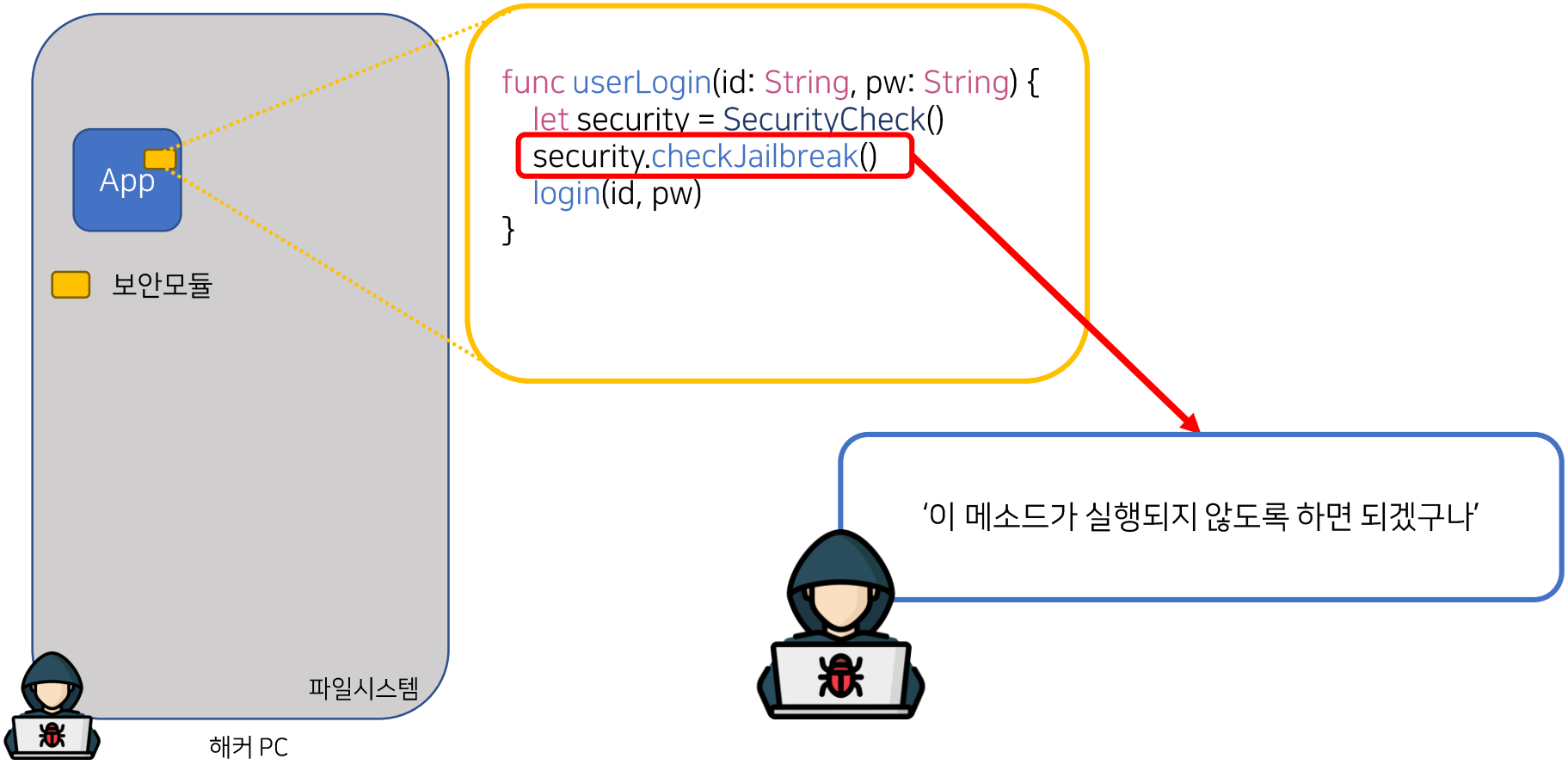
iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 1. 심볼 확인



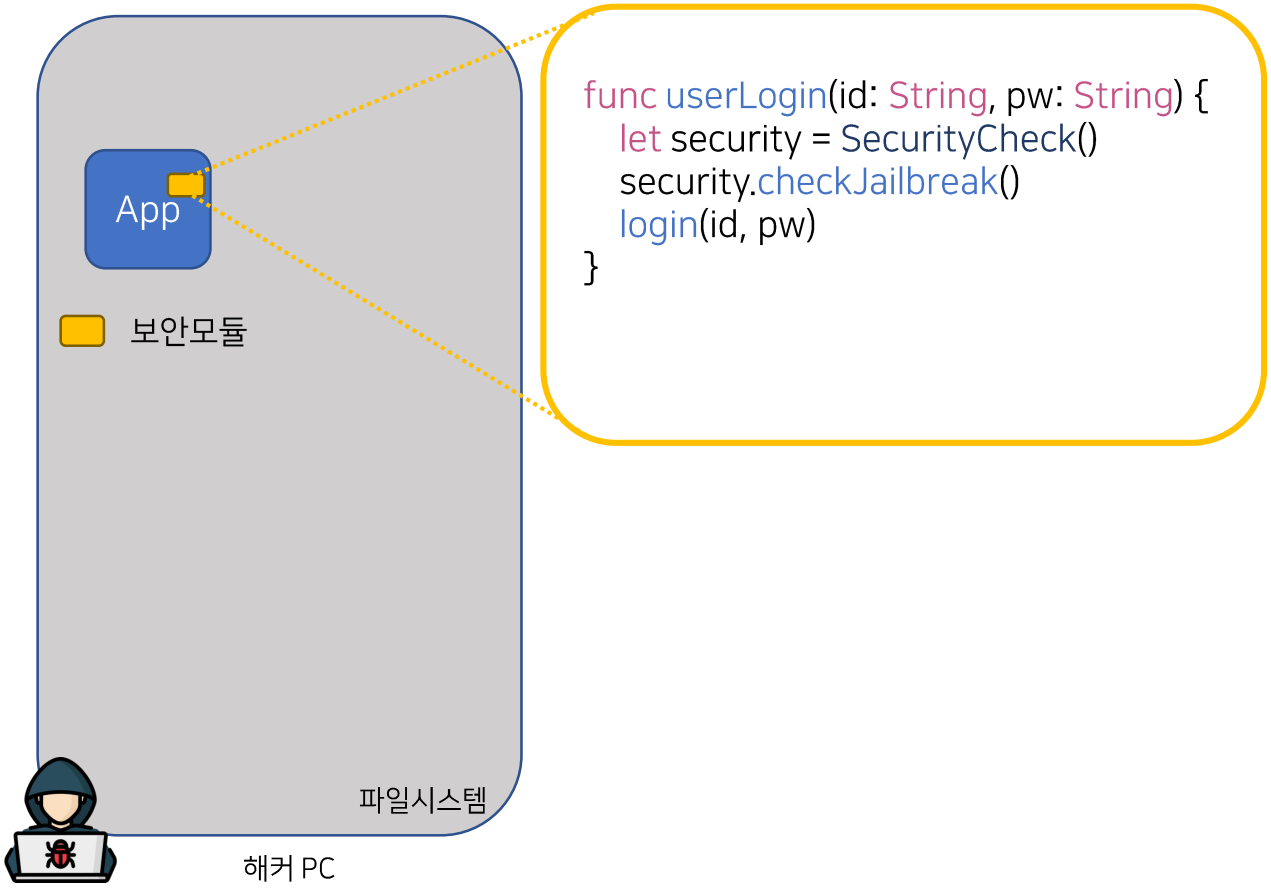
iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 1. 심볼 확인



iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 2. 메소드 스위즐링 수행



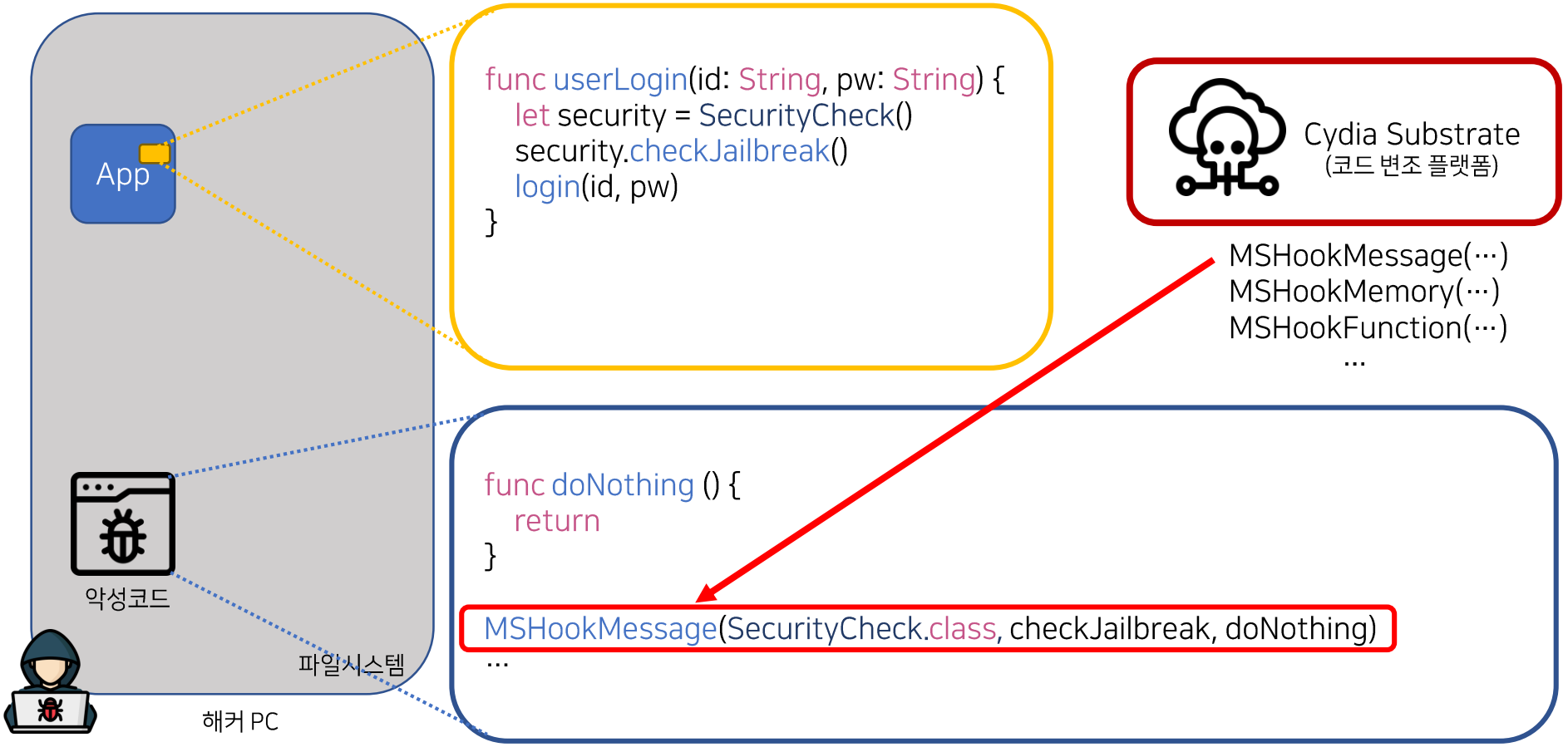

```
func userLogin(id: String, pw: String) {
    let security = SecurityCheck()
    security.checkJailbreak()
    login(id, pw)
}
```

Cydia Substrate
(코드 변조 플랫폼)

MSHookMessage(…)
MSHookMemory(…)
MSHookFunction(…)
…

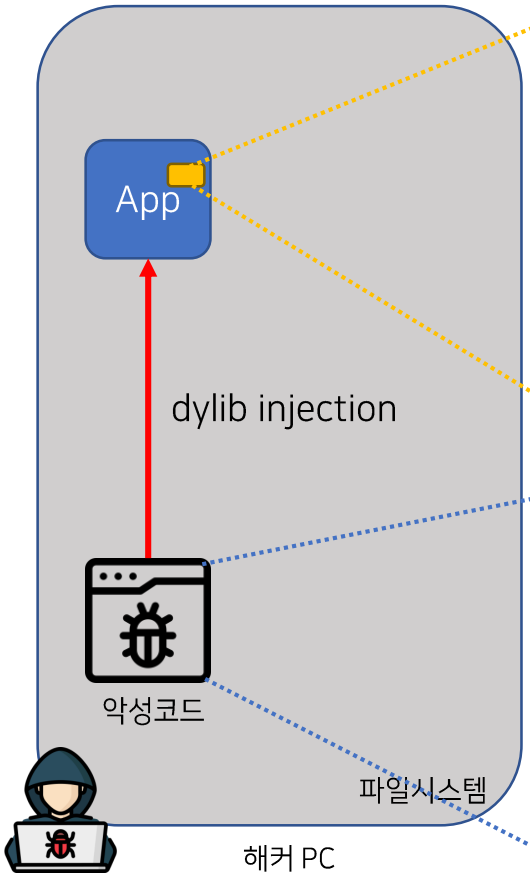
iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 2. 메소드 스위즐링 수행



iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 3. dylib injection




```
func userLogin(id: String, pw: String) {
    let security = SecurityCheck()
    security.checkJailbreak()
    login(id, pw)
}
```

Cydia Substrate
(코드 변조 플랫폼)

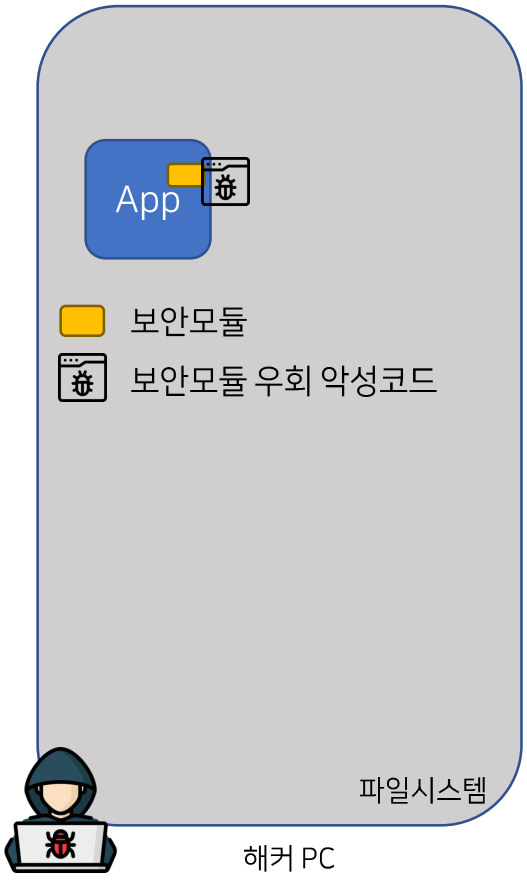
MSHookMessage(…)
MSHookMemory(…)
MSHookFunction(…)
…

```
func doNothing () {
    return
}

MSHookMessage(SecurityCheck.class, checkJailbreak, doNothing)
…
```

iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 4. 보안모듈 우회 완료




```
func userLogin(id: String, pw: String) {
    let security = SecurityCheck()
    security.checkJailbreak()
    login(id, pw)
}
```

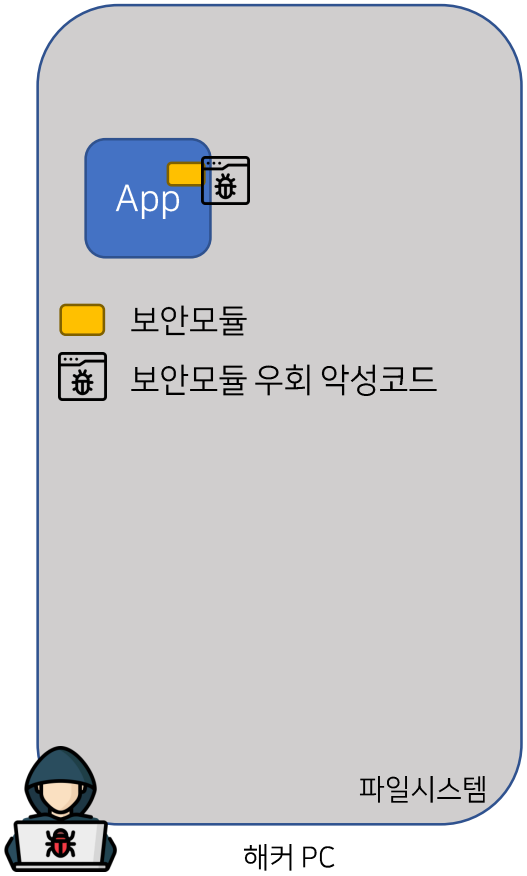
MSHookMessage(…)
MSHookMemory(…)
MSHookFunction(…)
…

```
func doNothing () {
    return
}

MSHookMessage(SecurityCheck.class, checkJailbreak, doNothing)
…
```

iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 1. 심볼 확인




```
// RVA: 0x17949CC Offset: 0x17949CC VA: 0x17949CC Slot: 36
public virtual void ChangeStart() { }

// RVA: 0x1794A6C Offset: 0x1794A6C VA: 0x1794A6C Slot: 37
public virtual void ChangePosition() { }

// RVA: 0x1794B00 Offset: 0x1794B00 VA: 0x1794B00 Slot: 38
public void GetDamage(bool ABCBBCCAACCBACDADDDAACC = True) { }

// RVA: 0x1794C74 Offset: 0x1794C74 VA: 0x1794C74
public float GetCriticalDamage() { }
```

유니티 심볼 분석 프로그램:
il2CppDumper

'해당 로직이 정상적으로 실행되지 않도록 하면 데미지를 받지 않을 수도 있겠구나'

iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 1. 심볼 확인


```
// RVA: 0x17949CC Offset: 0x17949CC VA: 0x17949CC Slot: 36
public virtual void ChangeStart() { }

// RVA: 0x1794A6C Offset: 0x1794A6C VA: 0x1794A6C Slot: 37
public virtual void ChangePosition() { }

// RVA: 0x1794B00 Offset: 0x1794B00 VA: 0x1794B00 Slot: 38
public void GetDamage(bool ABCBBCCAACCBACDADDDAACC = True) { }

// RVA: 0x1794C74 Offset: 0x1794C74 VA: 0x1794C74
public float GetCriticalDamage() { }
```

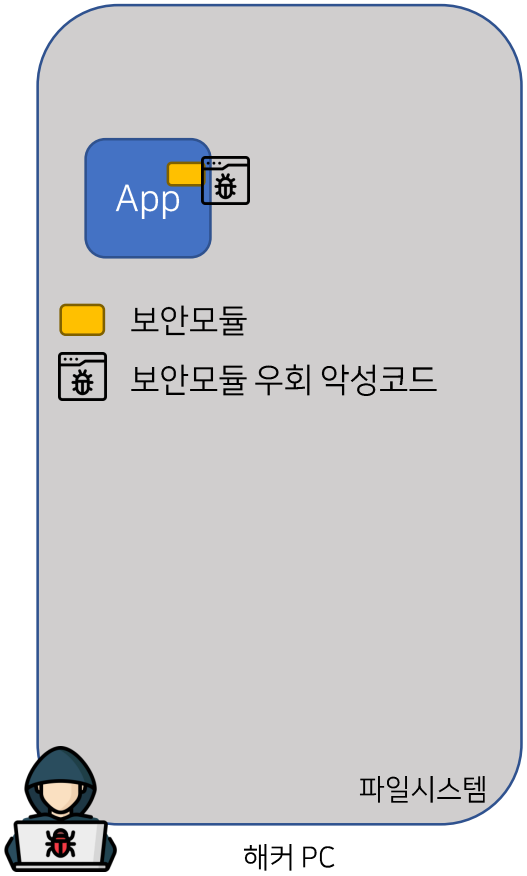
유니티 심볼 분석 프로그램:
il2CppDumper

```
void __cdecl GetDamage(Object_o *this, bool ABCBBCCAACCBACDADDDAACC, const MethodInfo *method)
{
  _BOOL4 v3; // w20
  __int64 v5; // x21
  float v6; // s8
  float v7; // s9
  float v8; // s8
  bool v9; // zf
  __int128 v10; // q0
  __int128 v11; // q1
  float v12; // s0
  float v13; // s2
  ADDDAABADCCCBDCDBBCCDAC_c *v14; // x0
  struct ADDDAABADCCCBDCDBBCCDAC_StaticFields *v15; // x8
  __int128 v16; // q0
```

정적분석 프로그램: IDA pro

iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 2. 바이너리 패치

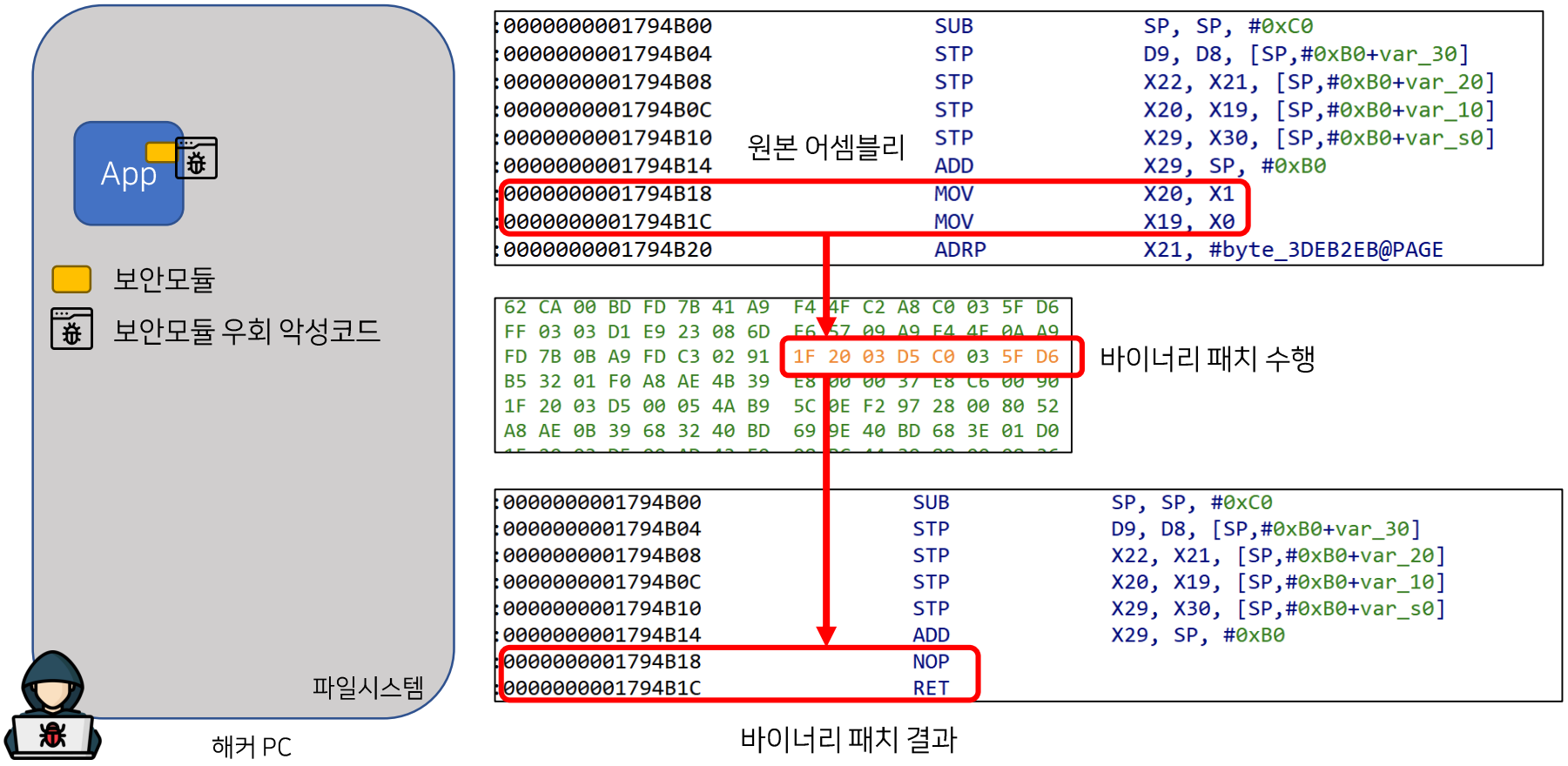


원본 어셈블리

```
0000000001794B00    SUB     SP, SP, #0xC0
0000000001794B04    STP     D9, D8, [SP,#0xB0+var_30]
0000000001794B08    STP     X22, X21, [SP,#0xB0+var_20]
0000000001794B0C    STP     X20, X19, [SP,#0xB0+var_10]
0000000001794B10    STP     X29, X30, [SP,#0xB0+var_s0]
0000000001794B14    ADD     X29, SP, #0xB0
0000000001794B18    MOV     X20, X1
0000000001794B1C    MOV     X19, X0
0000000001794B20    ADRP    X21, #byte_3DEB2EB@PAGE
```

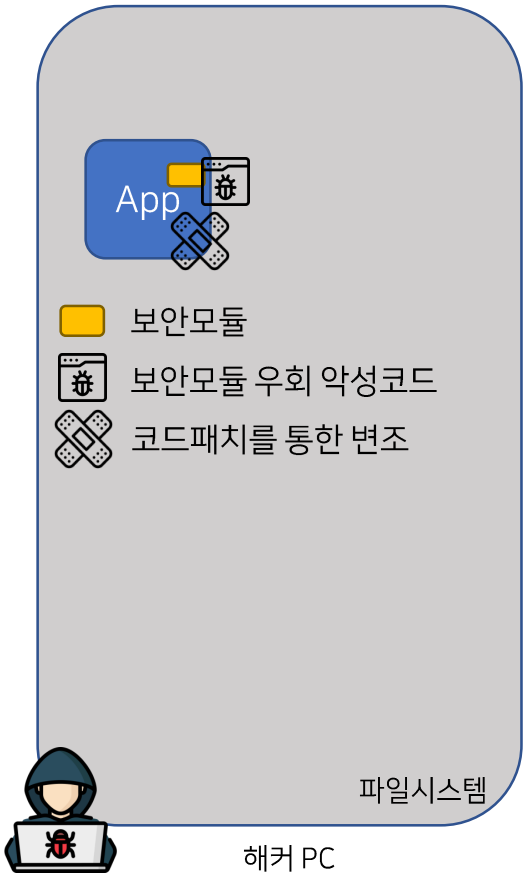
iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 2. 바이너리 패치



iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 2. 바이너리 패치



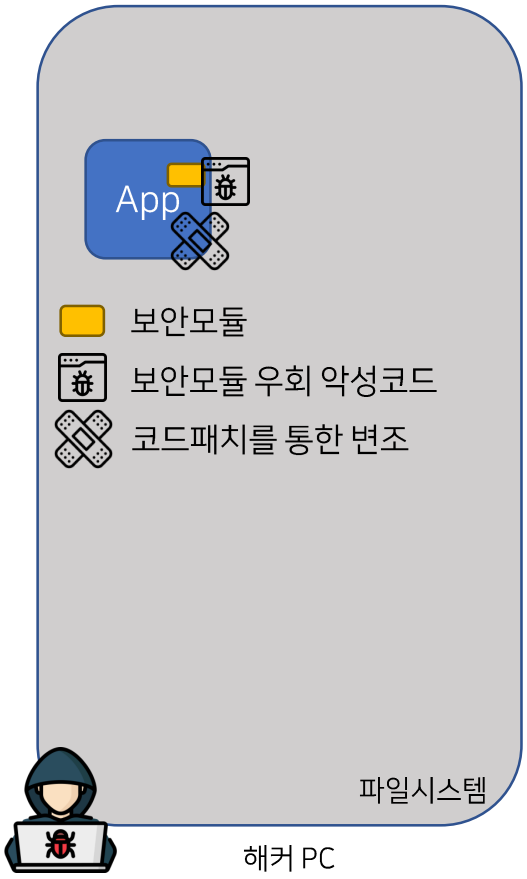

```
void __cdecl GetDamage(Object_o *this, bool ABCBBCCAACCBACDADDDAACC, const MethodInfo *method)
{
  _BOOL4 v3; // w20
  __int64 v5; // x21
  float v6; // s8
  float v7; // s9
  float v8; // s8
  bool v9; // zf
  __int128 v10; // q0
  __int128 v11; // q1
  float v12; // s0
  float v13; // s2
  ADDDAABADCCCBDCDBBCCDAC_c *v14; // x0
  struct ADDDAABADCCCBDCDBBCCDAC_StaticFields *v15; // x8
  __int128 v16; // q0
```

바이너리 코드패치를 통한 변조 결과

```
void __cdecl GetDamage(Object_o *this, bool ABCBBCCAACCBACDADDDAACC, const MethodInfo *method)
{
  return;
}
```

iOS 모바일 해킹 유형: 앱 위변조

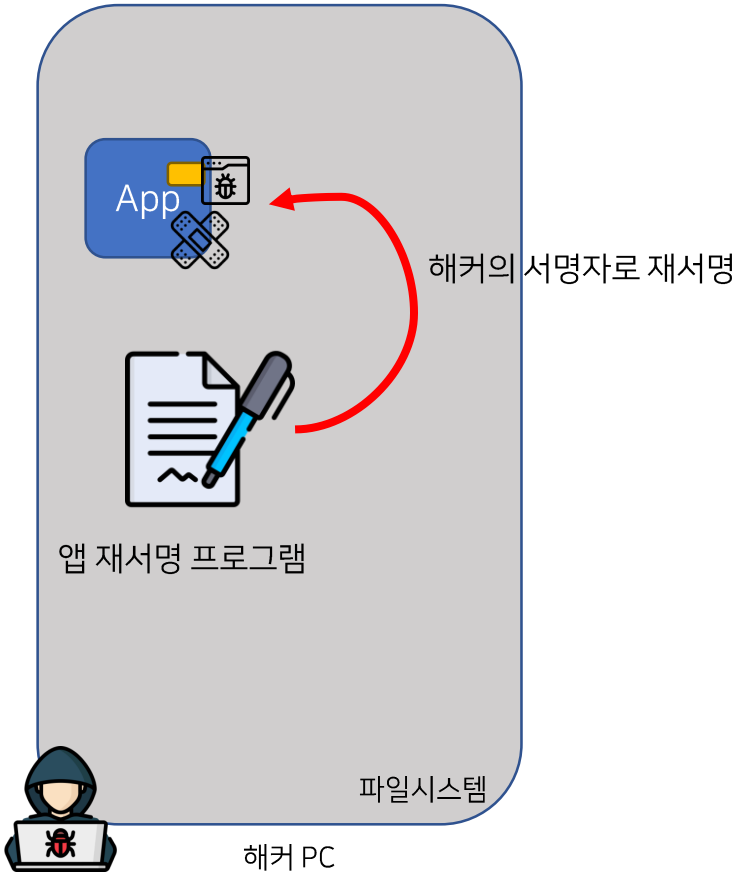
- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 3. 변조 완료



- 메소드 스위즐링을 통해 jailbreakCheck() 함수 우회 (탈옥 탐지 우회)
- 바이너리 패치를 통해 getDamage() 함수 변조 (데미지를 입지 않음)
- 앱의 무결성이 훼손되어 있음

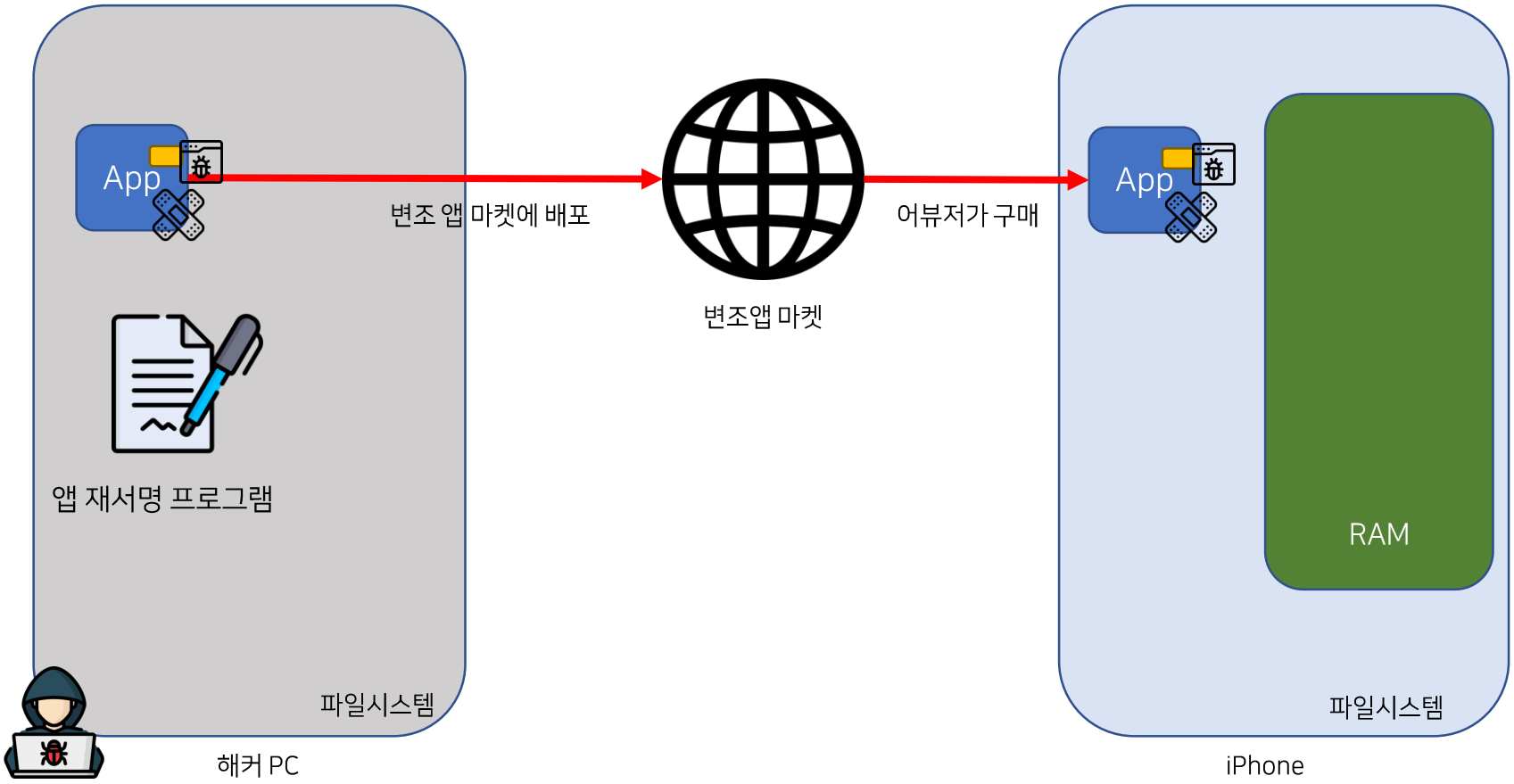
iOS 모바일 해킹 유형: 앱 위변조

- 해커의 서명자로 재서명

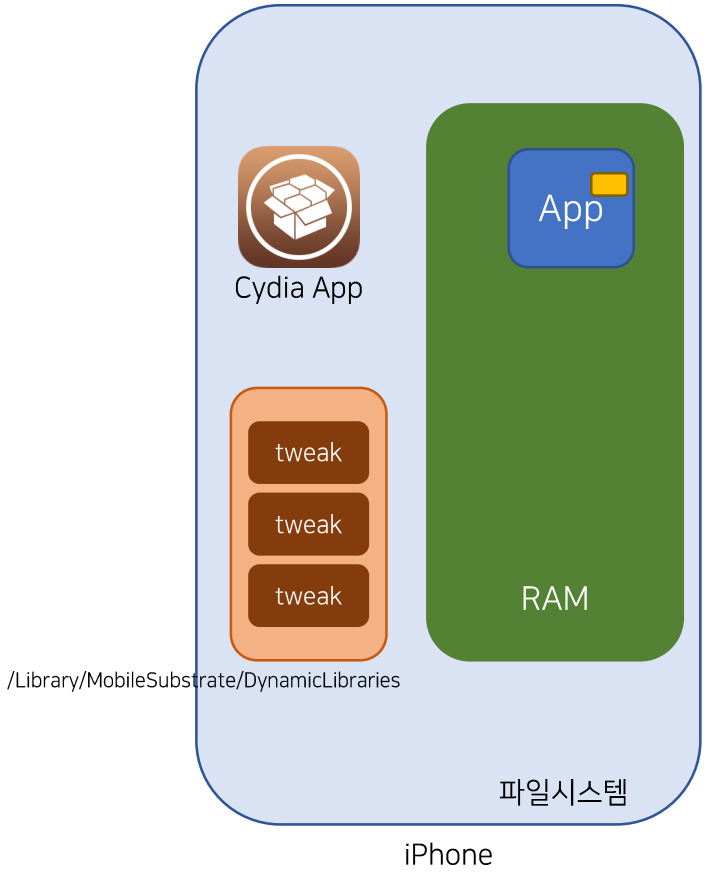


iOS 모바일 해킹 유형: 앱 위변조

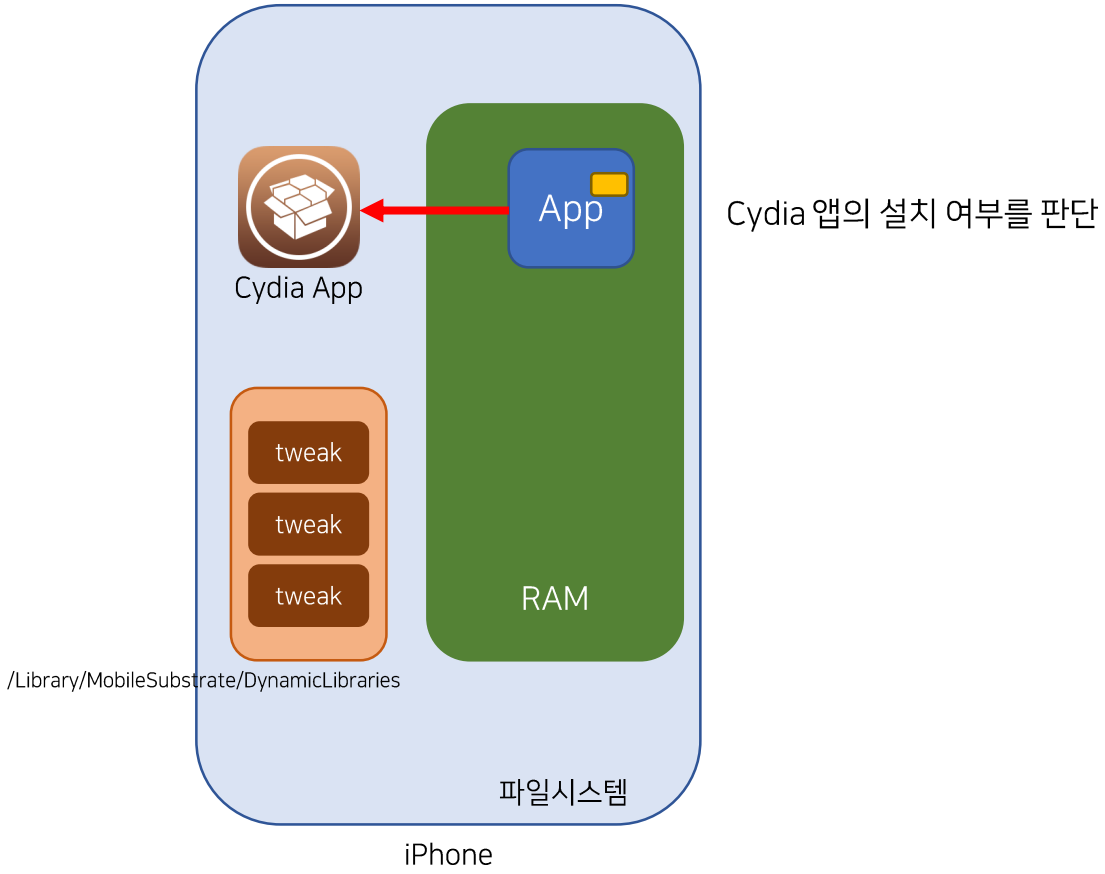
- 변조앱 배포



iOS 모바일 해킹 대응: 탈옥 탐지

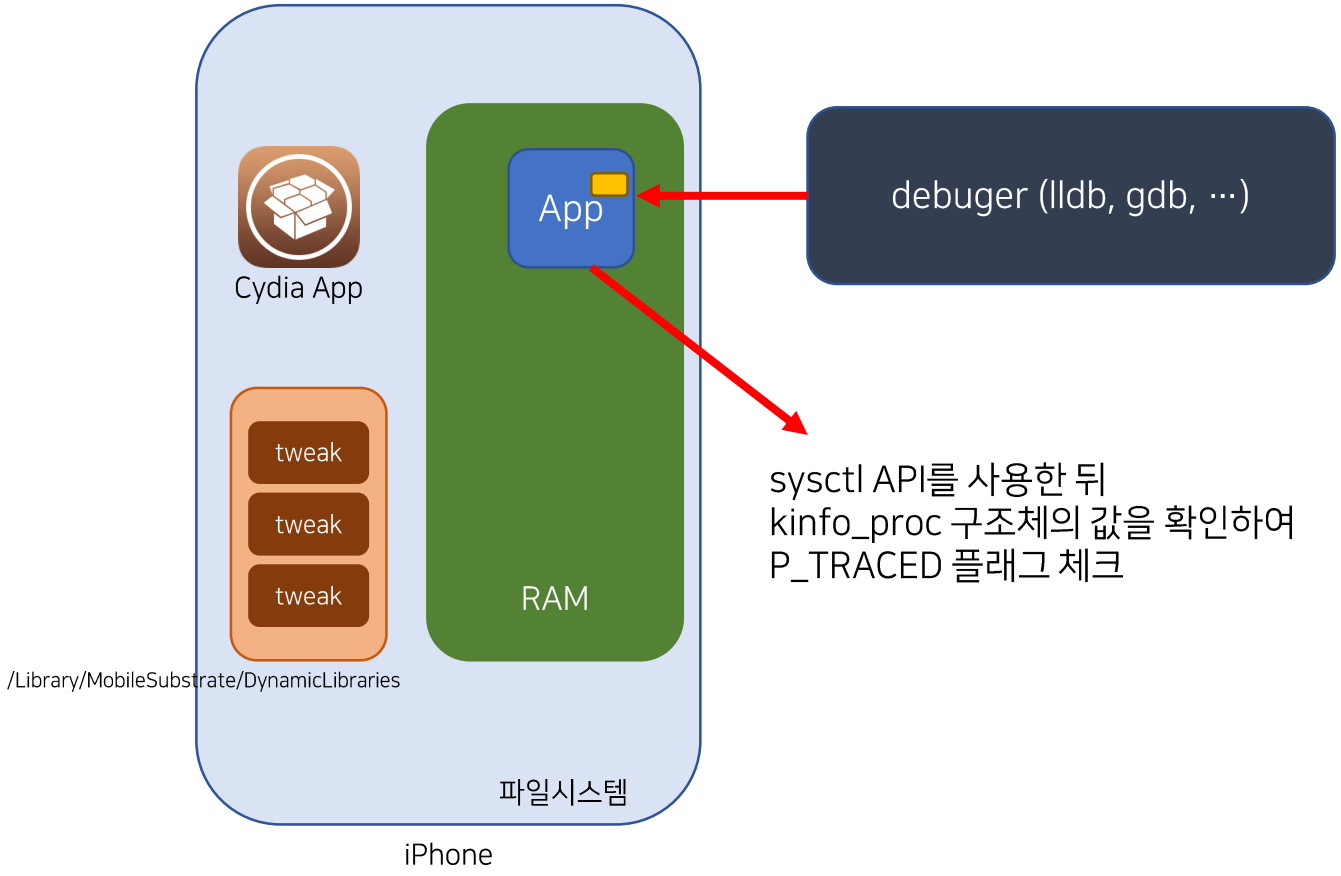


iOS 모바일 해킹 대응: 탈옥 탐지

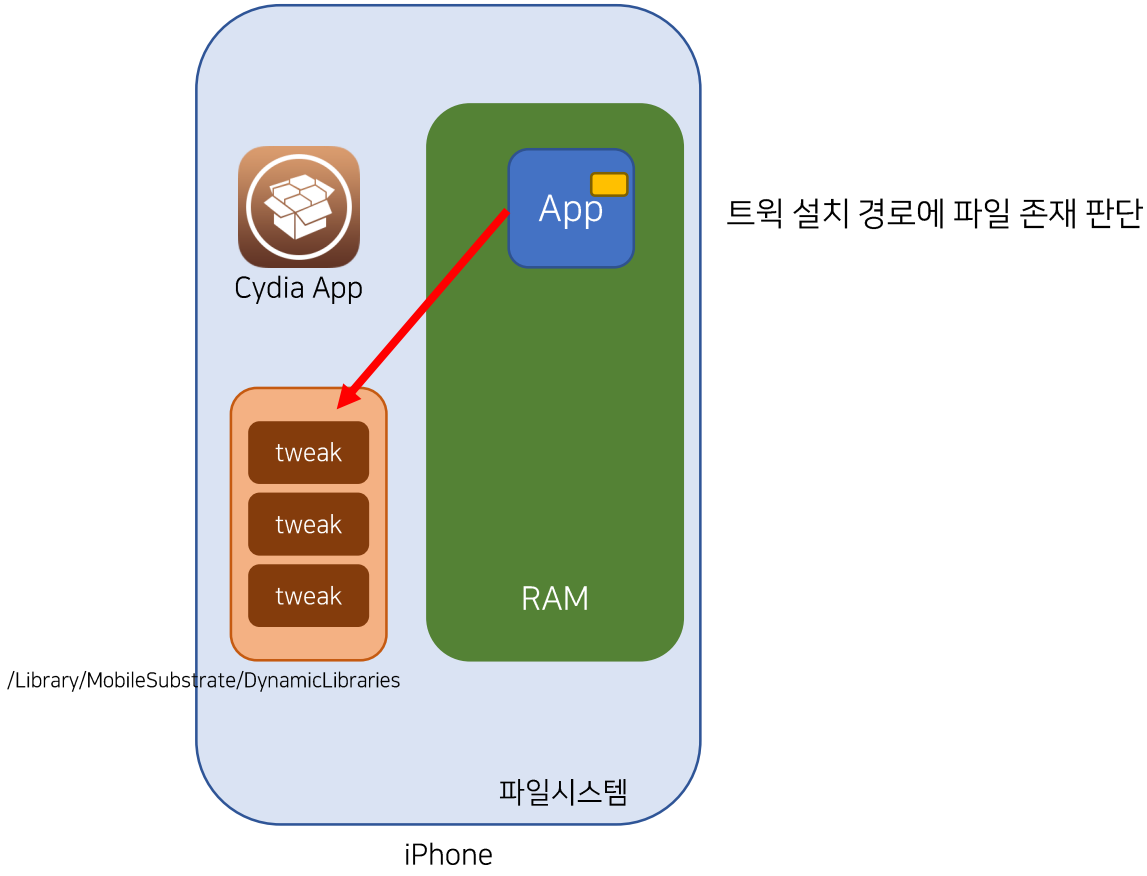


Cydia 앱의 설치 여부를 판단

iOS 모바일 해킹 대응: 디버깅 탐지

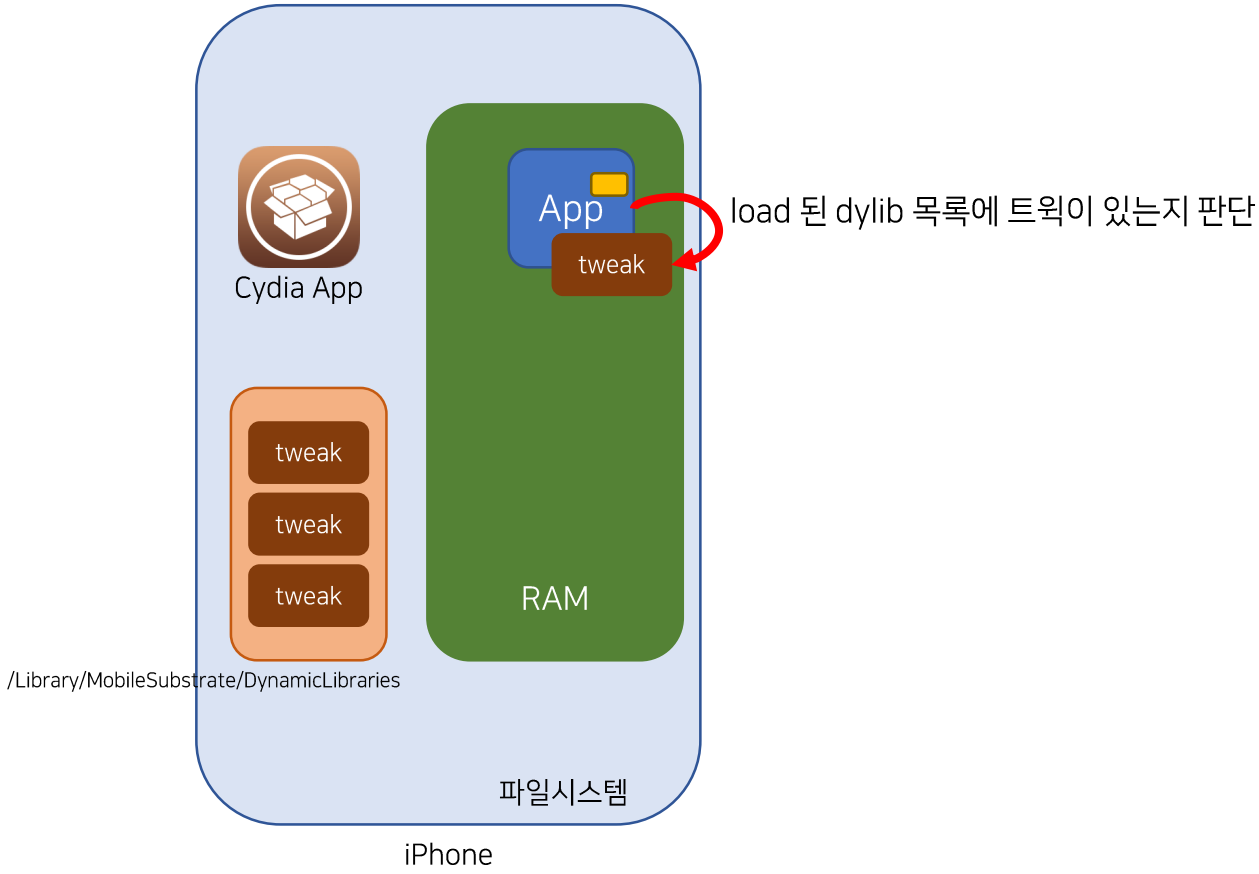


iOS 모바일 해킹 대응: tweak 탐지



트윅 설치 경로에 파일 존재 판단

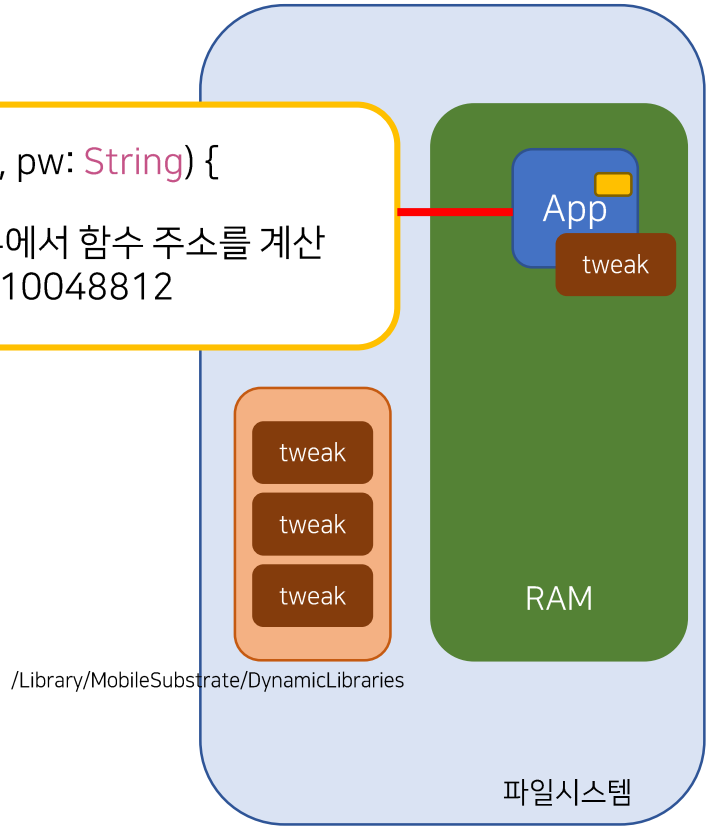
iOS 모바일 해킹 대응: tweak 탐지



iOS 모바일 해킹 대응: 메소드 스위즐링 탐지

```
func userLogin(id: String, pw: String) {
    ...
}
```

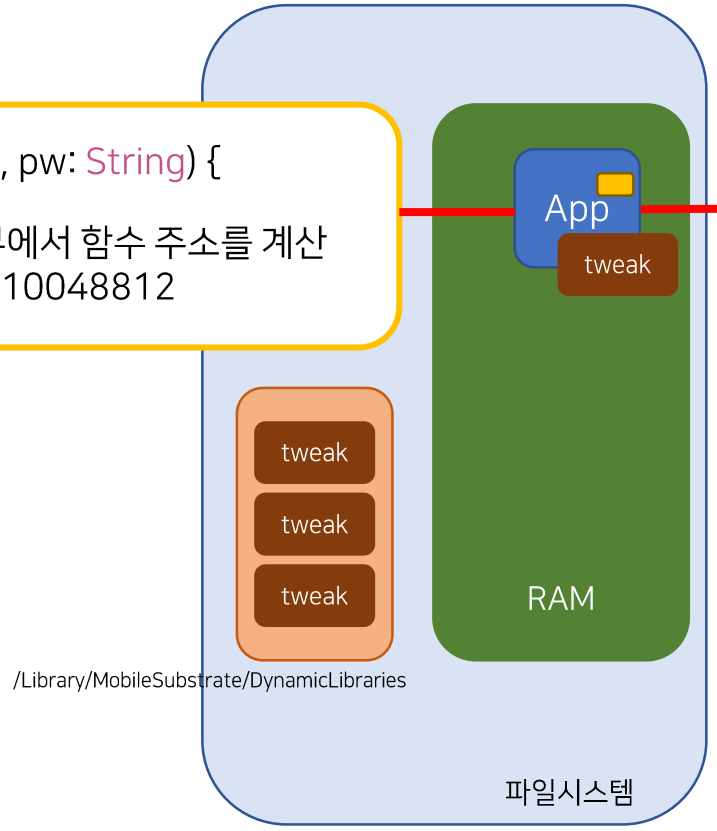
앱 내부에서 함수 주소를 계산
=> 0x10048812



iOS 모바일 해킹 대응: 메소드 스위즐링 탐지

```
func userLogin(id: String, pw: String) {
    ...
}
```

앱 내부에서 함수 주소를 계산
=> 0x10048812



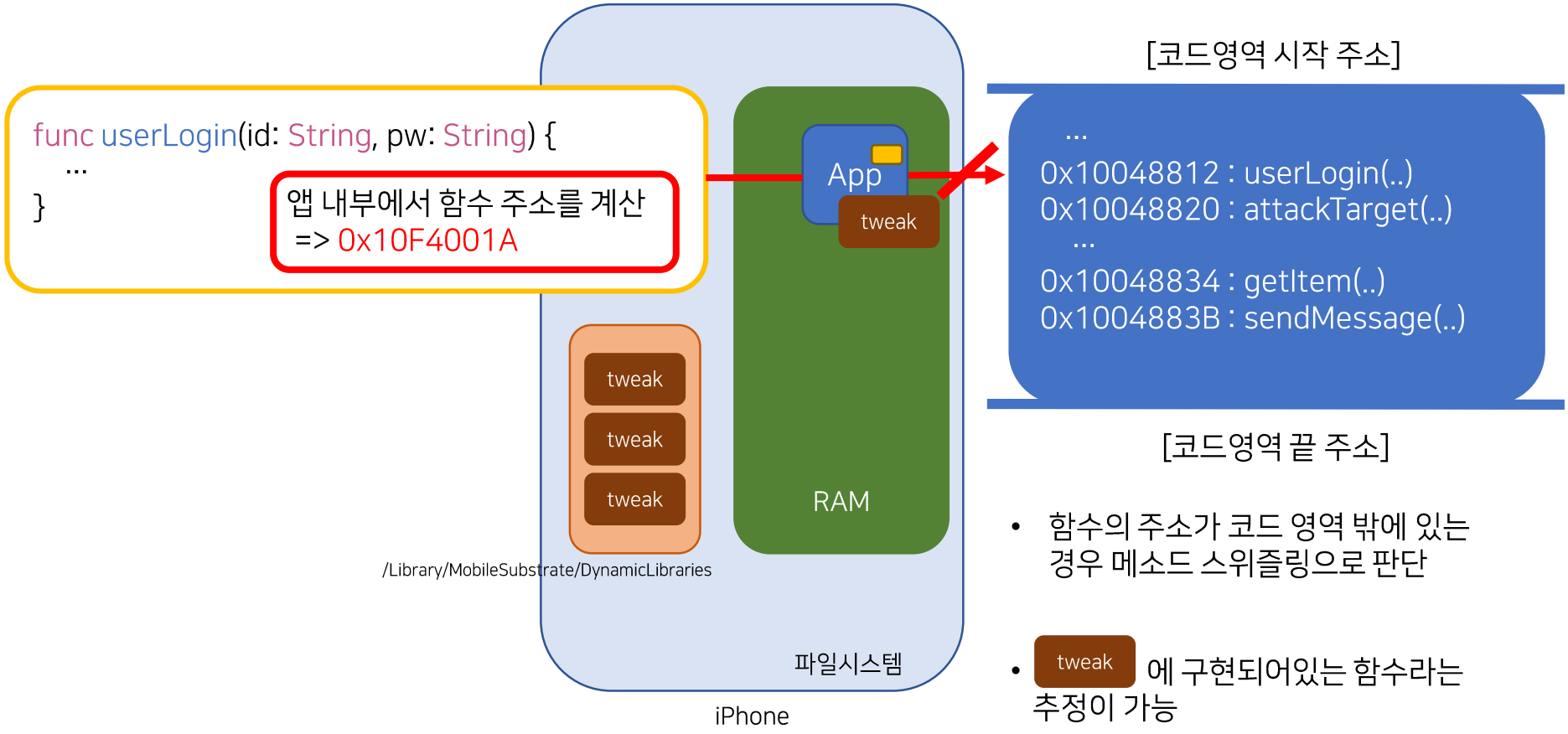
[코드영역 시작 주소]

```
    ...
0x10048812 : userLogin(..)
0x10048820 : attackTarget(..)
    ...
0x10048834 : getItem(..)
0x1004883B : sendMessage(..)
```

[코드영역 끝 주소]

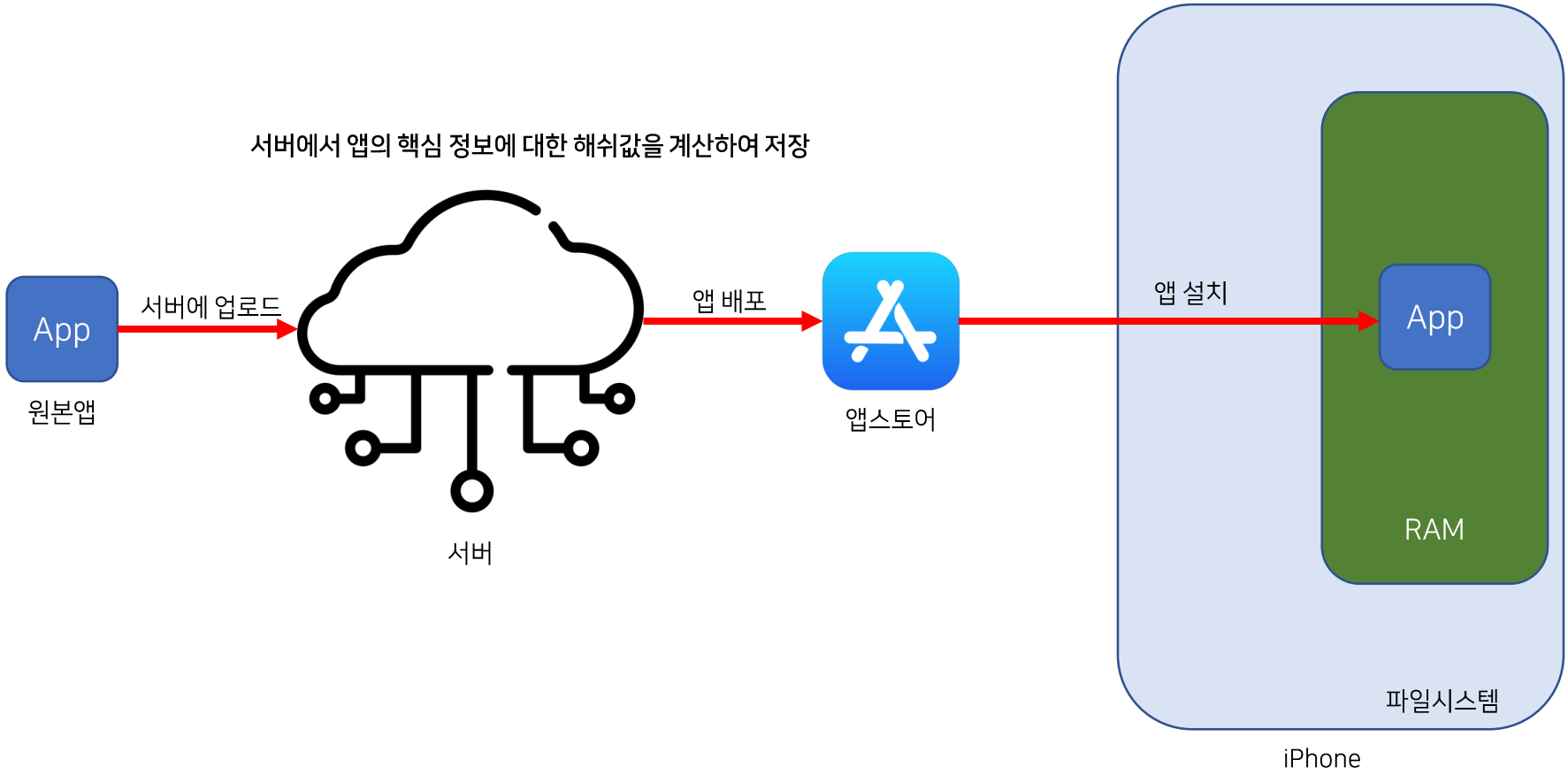
- 정상인 경우, 해당 함수 주소가 메인 앱 코드 영역 내에 존재

iOS 모바일 해킹 대응: 메소드 스위즐링 탐지

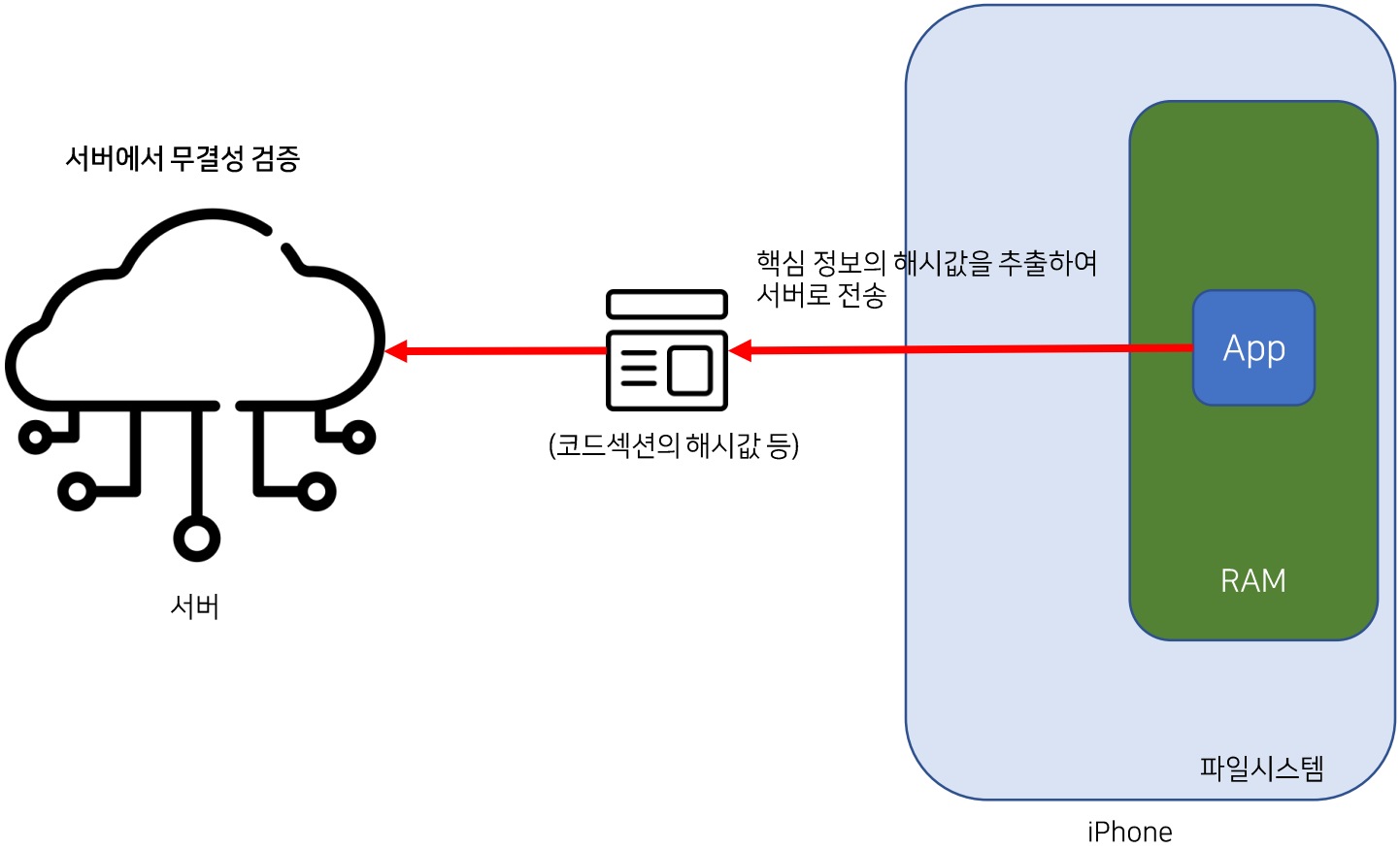


- 함수의 주소가 코드 영역 밖에 있는 경우 메소드 스위즐링으로 판단
- tweak 에 구현되어있는 함수라는 추정이 가능

iOS 모바일 해킹 대응: 앱 위변조 탐지



iOS 모바일 해킹 대응: 앱 위변조 탐지



NHN AppGuard 도입사례

NHN

NHN LINK

NHN wetoo

NHN PAYCO

NHN Bugs!

NHN PLAYART

대외 고객

LOAD COMPLETE

SUPER PLANET

gamepub

millie

CLOVERGAMES

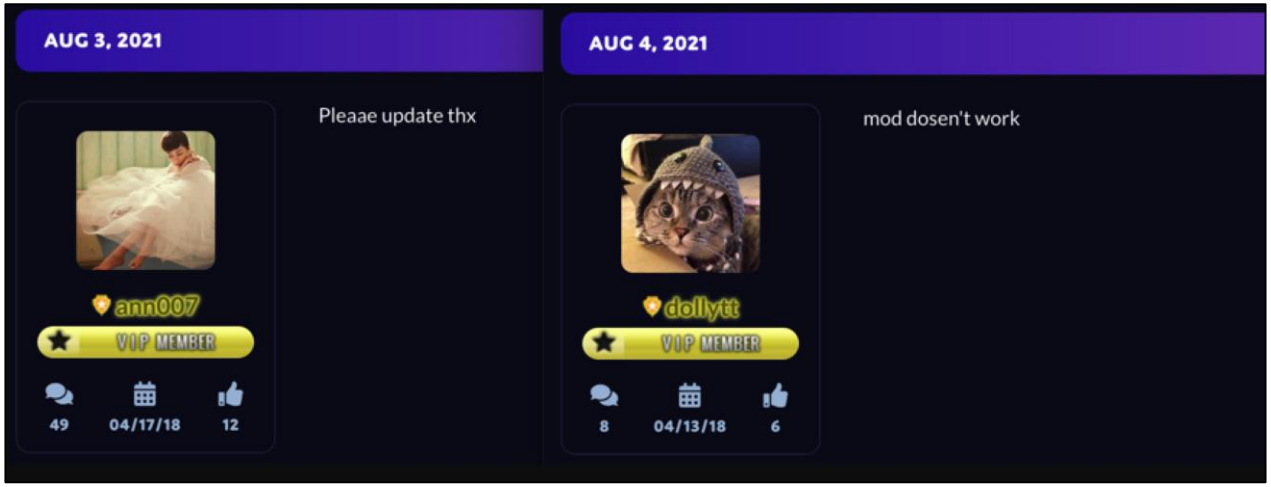
PERCENT

iamSchool

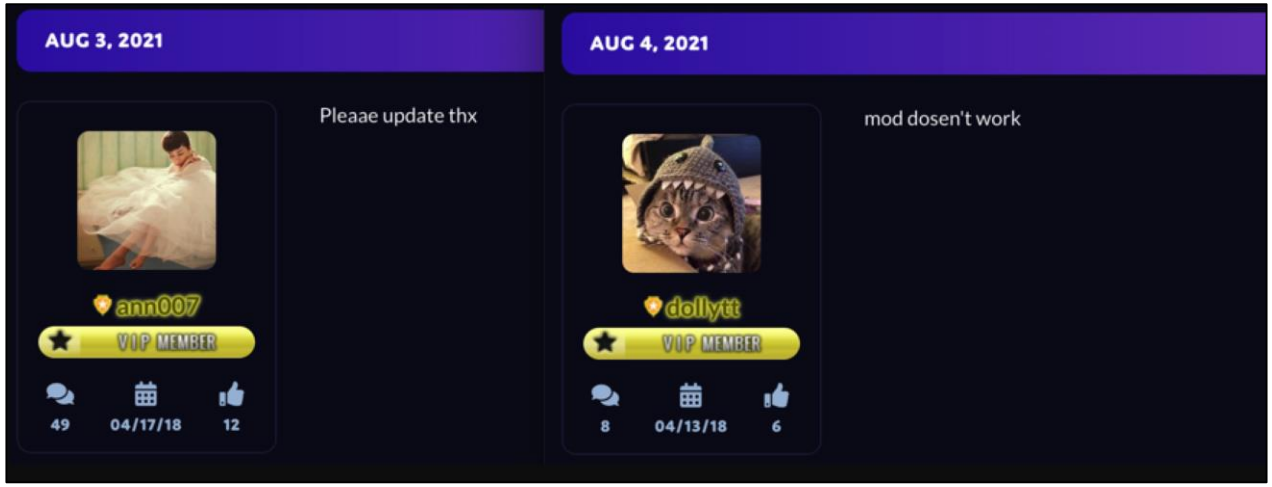
mobirix

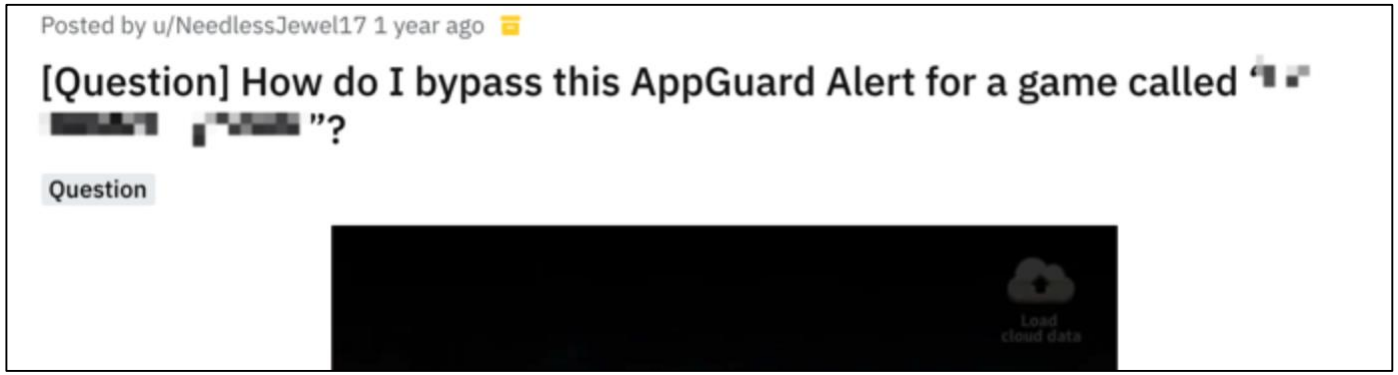
RISING WINGS

NHN AppGuard 도입사례

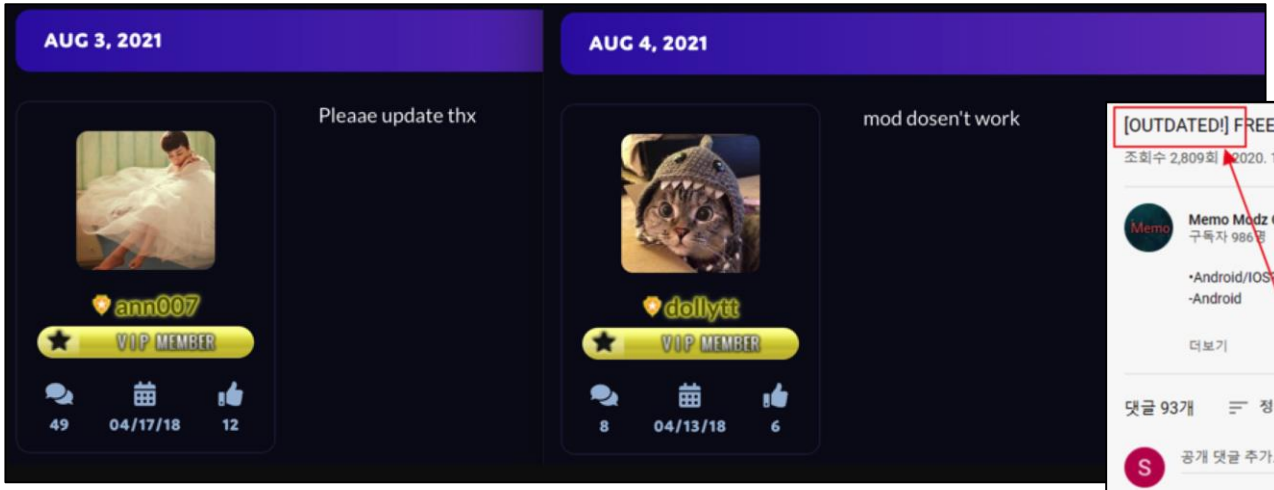


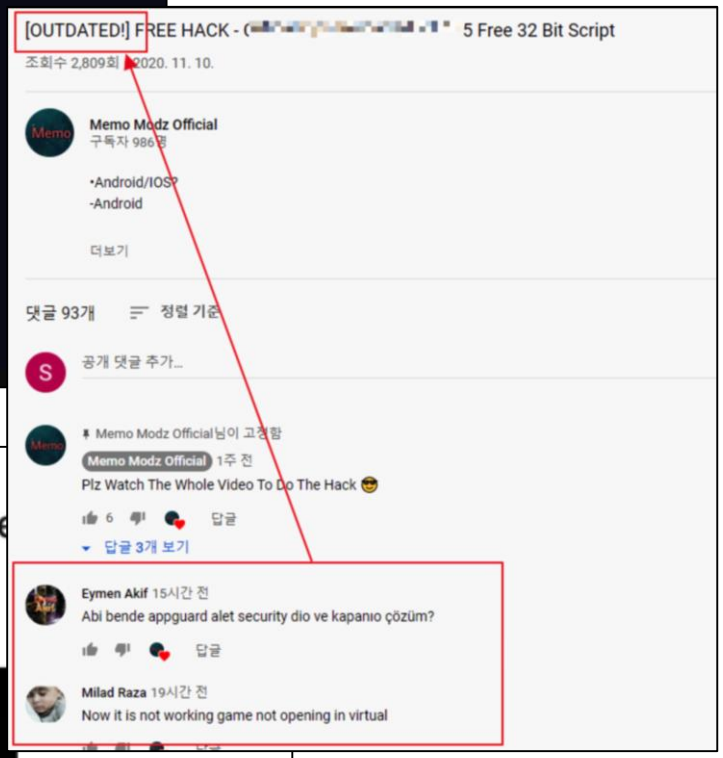
NHN AppGuard 도입사례

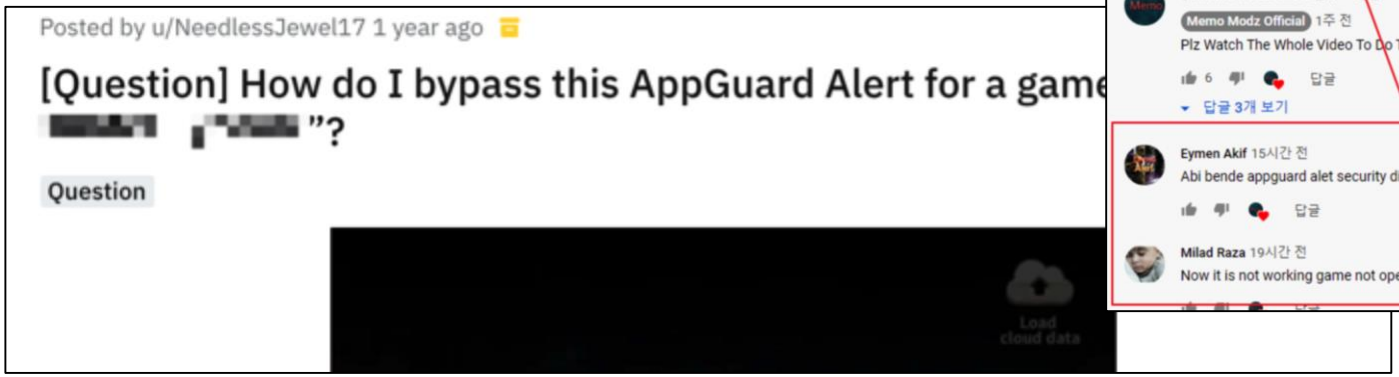




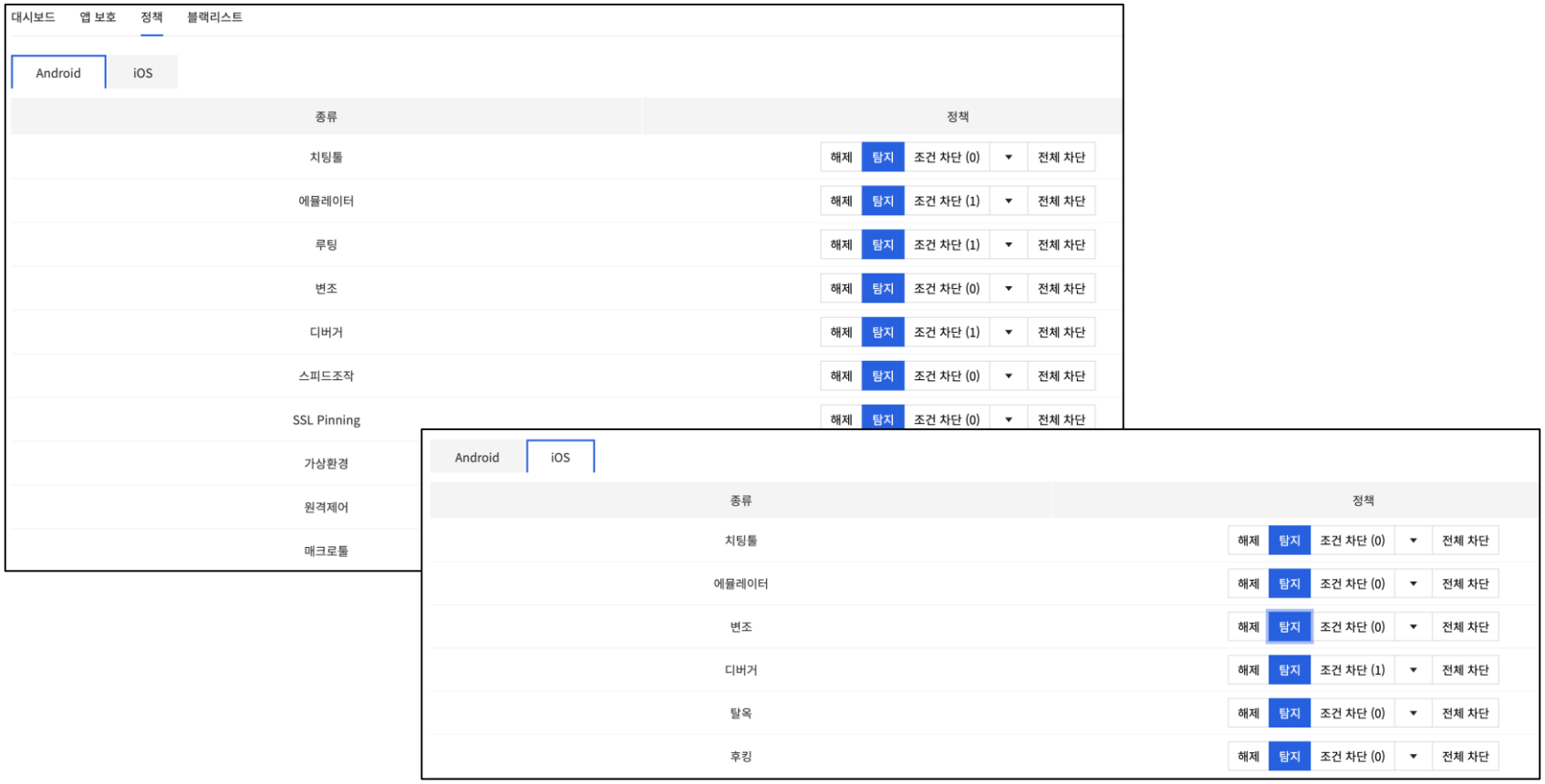
NHN AppGuard 도입사례



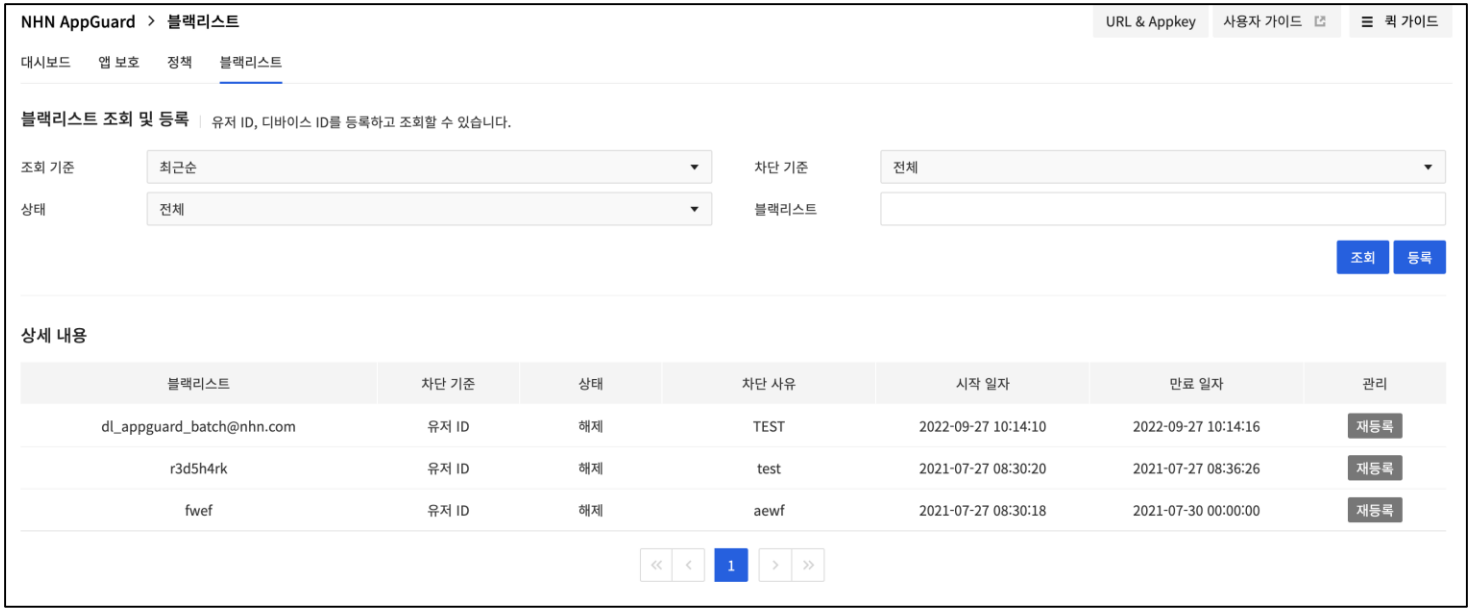




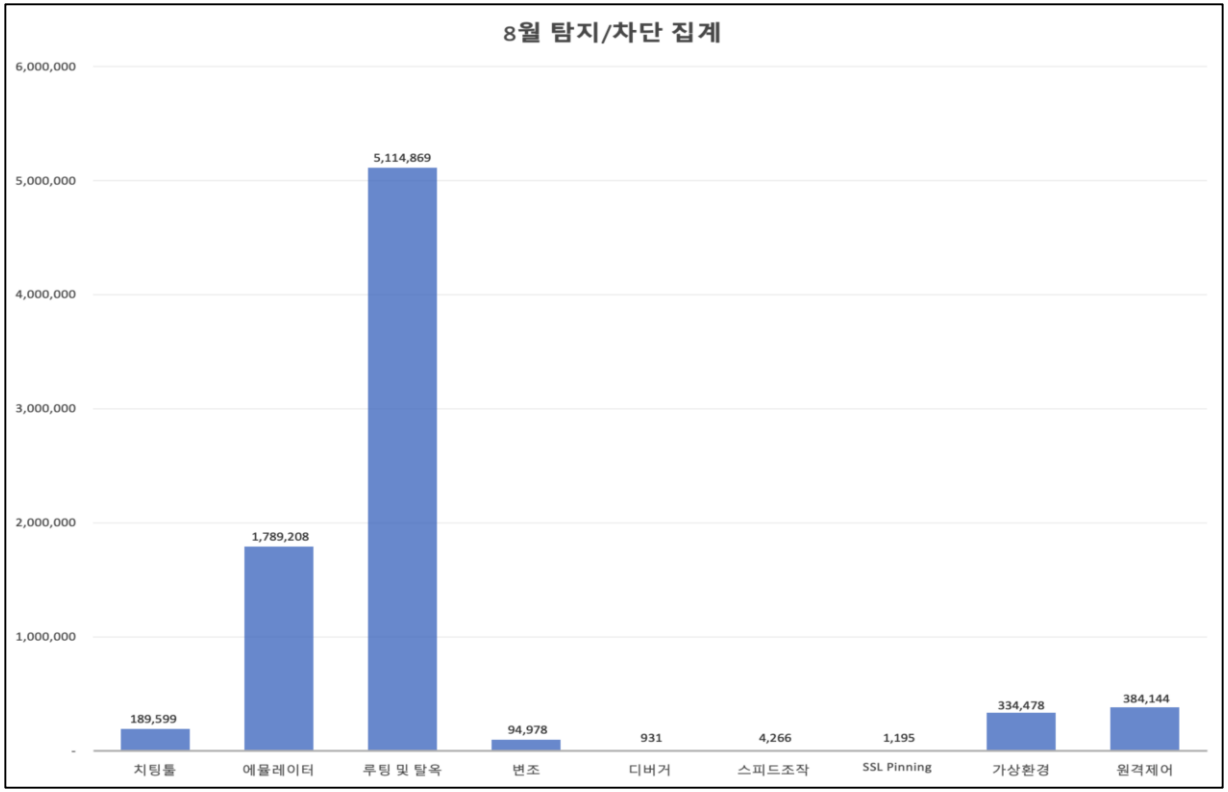
NHN AppGuard 도입사례



NHN AppGuard 도입사례



NHN AppGuard 도입사례



출처

- https://pixabay.com/ko/photos/%ed%95%b4%ec%bb%a4-%ec%8b%a4%eb%a3%a8%ec%97%a3-%ed%95%b4%ed%82%b9-3342696/
- https://pixabay.com/ko/illustrations/%ec%82%ac%ec%9d%b4%eb%b2%84-%ea%b3%b5%ea%b2%a9-%ec%95%94%ed%98%b8%ed%99%94-%ec%8a%a4%eb%a7%88%ed%8a%b8%ed%8f%b0-4444448/
- https://pixabay.com/ko/photos/%eb%8f%88-%eb%8f%99%ec%a0%84-%ec%8a%a4%ed%83%9d-%ec%8c%93%ec%9d%b8-%eb%8f%99%ec%a0%84-2180330/
- https://www.etoland.co.kr/plugin/mobile/board.php?bo_table=etohumor02&wr_id=1336540&cpage=
- https://zdnet.co.kr/view/?no=20130806162314
- https://www.asiae.co.kr/article/2015091910120097310
- https://gadfactory.tistory.com/95
- https://www.khgames.co.kr/news/articleView.html?idxno=133357
- https://www.hankyung.com/it/article/201806211682v
- https://pixabay.com/ko/vectors/%ec%9e%ac%ec%a0%95%ec%a0%81-%ec%9d%b8-%ec%9c%84%ea%b8%b0-%ec%83%81%ec%8b%a4-%ec%82%ac%ec%97%85-7249222/
- https://github.com/topjohnwu/Magisk
- https://apkvision.com/app/tools/cf-auto-root-47044/
- https://www.facebook.com/SuperSUofficial?ref=br_rs
- http://www.hungryapp.co.kr/bbs/bbs_view.php?durl=YmNvZGU9ZWxnYXJkJnBpZD0xMDIzMDM0JnVzZXI9
- https://ko.101-help.com/ef3cd1e2e0-android-phoneeseo-geulggoleul-byeongyeonghaneun-bangbeob-eungweon-eobseum/
- https://kgezzang.tistory.com/818
- https://gameguardian.net/forum/
- https://www.cheatengine.org/
- https://sbgamehacker.net/
- https://frida.re/

출처

- https://azeria-labs.com/debugging-with-gdb-introduction/
- https://www.youtube.com/watch?v=-9X965jXrn8
- https://www.flaticon.com/free-icon/vulnerability_1995751?term=vulnerable&page=1&position=1&page=1&position=1&related_id=1995751&origin=search
- https://www.flaticon.com/free-icon/playstation-logotype_1443?term=play%20station&page=1&position=1&page=1&position=1&related_id=1443&origin=search
- https://github.com/Perfare/Il2CppDumper
- https://www.flaticon.com/free-icon/files_569800?term=file&page=1&position=4&page=1&position=4&related_id=569800&origin=search
- https://www.flaticon.com/free-icon/dll-file-format-variant_29510?term=dll%20file&page=1&position=1&page=1&position=1&related_id=29510&origin=search
- https://www.flaticon.com/free-icon/apk-file-format_28869?term=apk%20file&page=1&position=1&page=1&position=1&related_id=28869&origin=search
- https://www.flaticon.com/free-icon/reverse-engineering_8265828?term=reverse%20engineer&page=1&position=21&page=1&position=21&related_id=8265828&origin=search
- https://pixabay.com/ko/illustrations/%ed%94%bc%ec%8b%b1-%ec%82%ac%ea%b8%b0-%ec%82%ac%ec%9d%b4%eb%b2%84-%eb%b3%b4%ec%95%88-%ed%95%b4%ed%82%b9-3390518/
- https://www.nvidia.com/ko-kr/geforce-now/
- https://www.xbox.com/ko-KR/xbox-game-pass
- https://namu.wiki/w/Stadia
- https://www.flaticon.com/free-icon/hacker_3518775?term=hacker&page=1&position=2&page=1&position=2&related_id=3518775&origin=search
- https://www.flaticon.com/free-icon/virus_4029725?related_id=4029725&origin=search
- https://www.flaticon.com/free-icon/hacking_2040305?term=hacking&page=1&position=54&page=1&position=54&related_id=2040305&origin=search
- https://www.flaticon.com/free-icon/band-aid_6703938?term=patch&page=1&position=3&page=1&position=3&related_id=6703938&origin=search
- https://www.flaticon.com/free-icon/hacker_1320457?term=hacker&page=1&position=3&page=1&position=3&related_id=1320457&origin=search
- https://www.flaticon.com/free-icon/malware_1059629?term=malware&page=1&position=11&page=1&position=11&related_id=1059629&origin=search
- https://www.flaticon.com/free-icon/world-wide-web_1006771?term=website&page=1&position=1&page=1&position=1&related_id=1006771&origin=search

출처

- https://www.flaticon.com/free-icon/contract_913393?term=sign&page=1&position=23&page=1&position=23&related_id=913393&origin=search
- https://www.flaticon.com/free-icon/cloud_2818793?term=cloud%20server&page=1&position=9&page=1&position=9&related_id=2818793
- https://www.flaticon.com/free-icon/post_3596965?term=post&page=1&position=63&page=1&position=63&related_id=3596965&origin=search
- https://www.flaticon.com/free-icon/search_7079548?related_id=7079548&origin=search

NHN Cloud On

NHN Cloud

유연하게, 안전하게
비즈니스에 힘이 되다.