

# 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



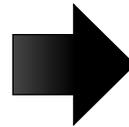
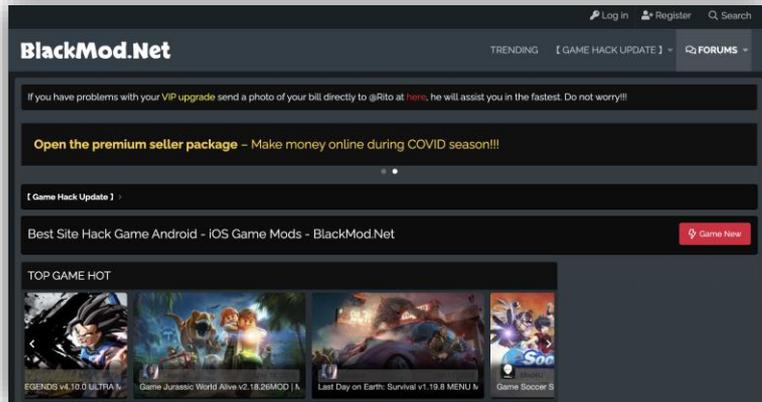
# 모바일 해킹이란?

- 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

## 왕서방, 거 너무 똑같은 거 아니오? 우리도 참을만큼 참았소

[Close-up] 中 짝퉁 게임에 손해 수십조... 국내 게임사들 잇따라 소송 대응

- 백발에 고글마저 똑같이 착용

넥슨, 캐릭터 무단 도용으로 中업체 '4399코리아' 고소 계획

배긴 기사... 4399가 배긴을 사들이고 있다... 배긴은 배긴이 배긴을 사들이고 있다...

- 中시

이젠

- 피해

"개별



## '리세마라' 꿈수, 매크로 대거 유통해 '부당 이득' ... 확률형 아이템 논란 신국면

지난해 확률형 아이템이 화두에 오른 배경에는 변동성 확률과 불합리한 확률, 소위 '천장'시스템으로 대변되는 고과금 유도 정책 등이 자리잡는다. 이 시스템에 반발하는 유저들의 말을 종합해보면 대체로 형평성에 논란을 제기한다. 공정해야할 게임 룰이 그렇지 않다는 이야기다. 특정 시점에 도달하면 '벽'을 느끼게 되며, 이 벽이 게임 노하우나 실력등을 초월할 정도로 높게 형성돼 있다는 지적들이 수시로 나온다. 게임 밸런스는 무너지고 유저들은 게임할 의지를 점차 상실하게 되는 상황이 봉착한다. 불합리한 밸런스라는 지적이다. 그런데 애초에 시작부터 불공정한 출발로 밸런스를 잃은 상태라면 어떨까. 룰을 악용해 부당 이득을 얻어 게임을 하는 형태를 진단해 봤다.

## 발진 "보안 대처 미흡 '죄송하다'"



만에 적극 대응

과학 >

### 해킹사고' 뽀뿌에서 광고앱으

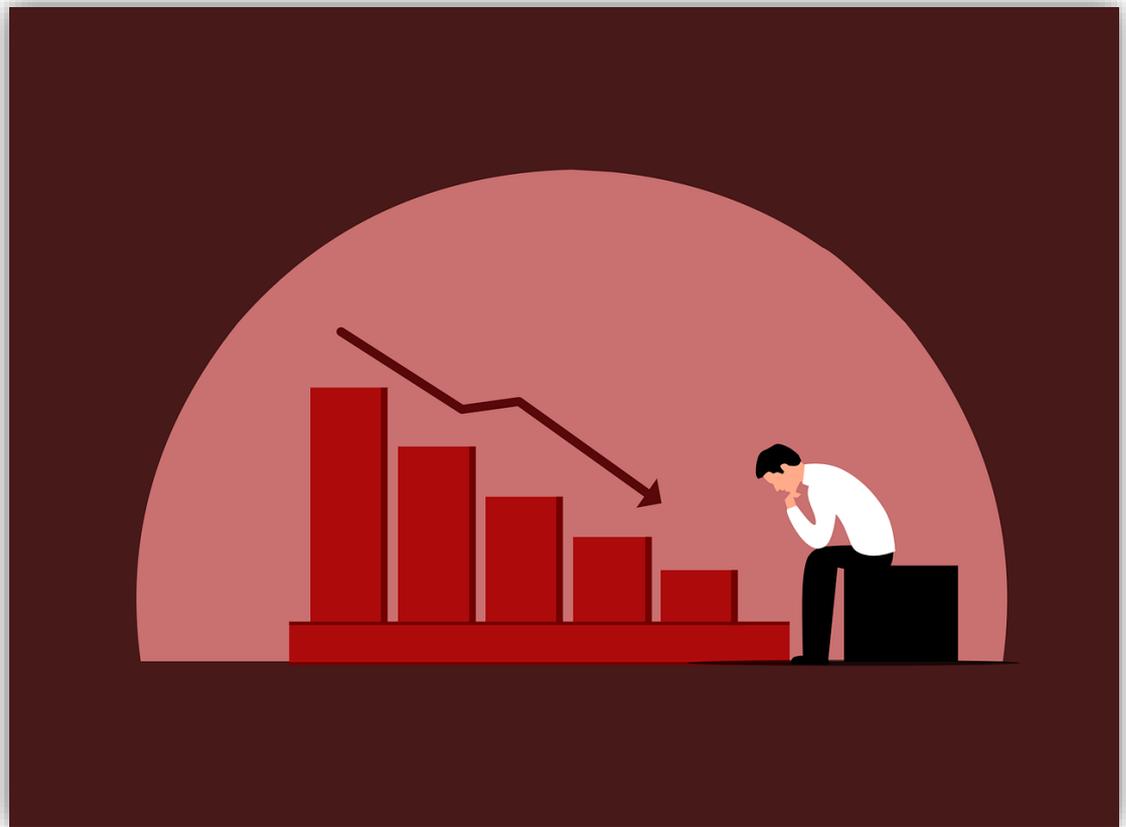
최종수정 2015.09.19 10:25 기사입력 2015.09.19 10:25

흥미 유발하는 콘텐츠 배열...광고와 연결돼있고 팝업되거나! 원격 조정 기능 갖고 있어 앱 패치 가능 수행



# 모바일 해킹이란?

- 해킹사고 발생에 따른 영향
  - 사용자수 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

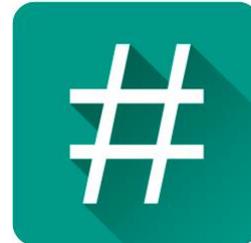


앱을 실행할 기기(실제 기기, 에뮬레이터)

루팅  
➔

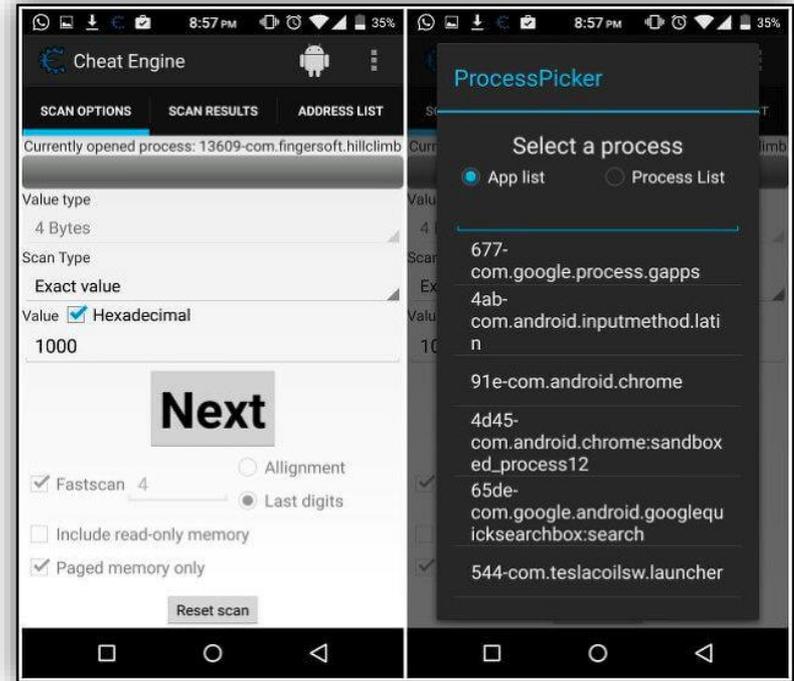
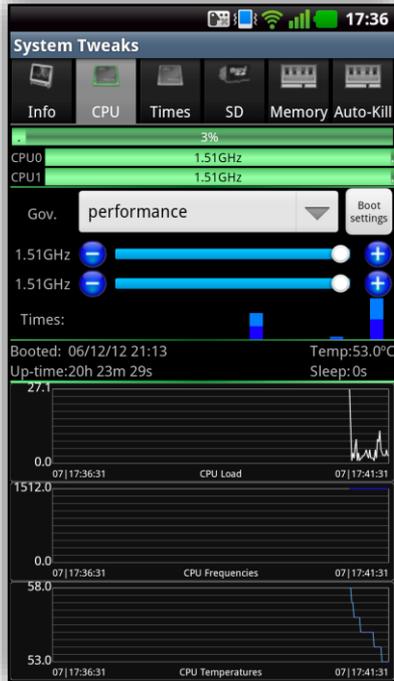


Magisk  
The Magic Mask for Android



# Android 모바일 해킹 유형 및 대응

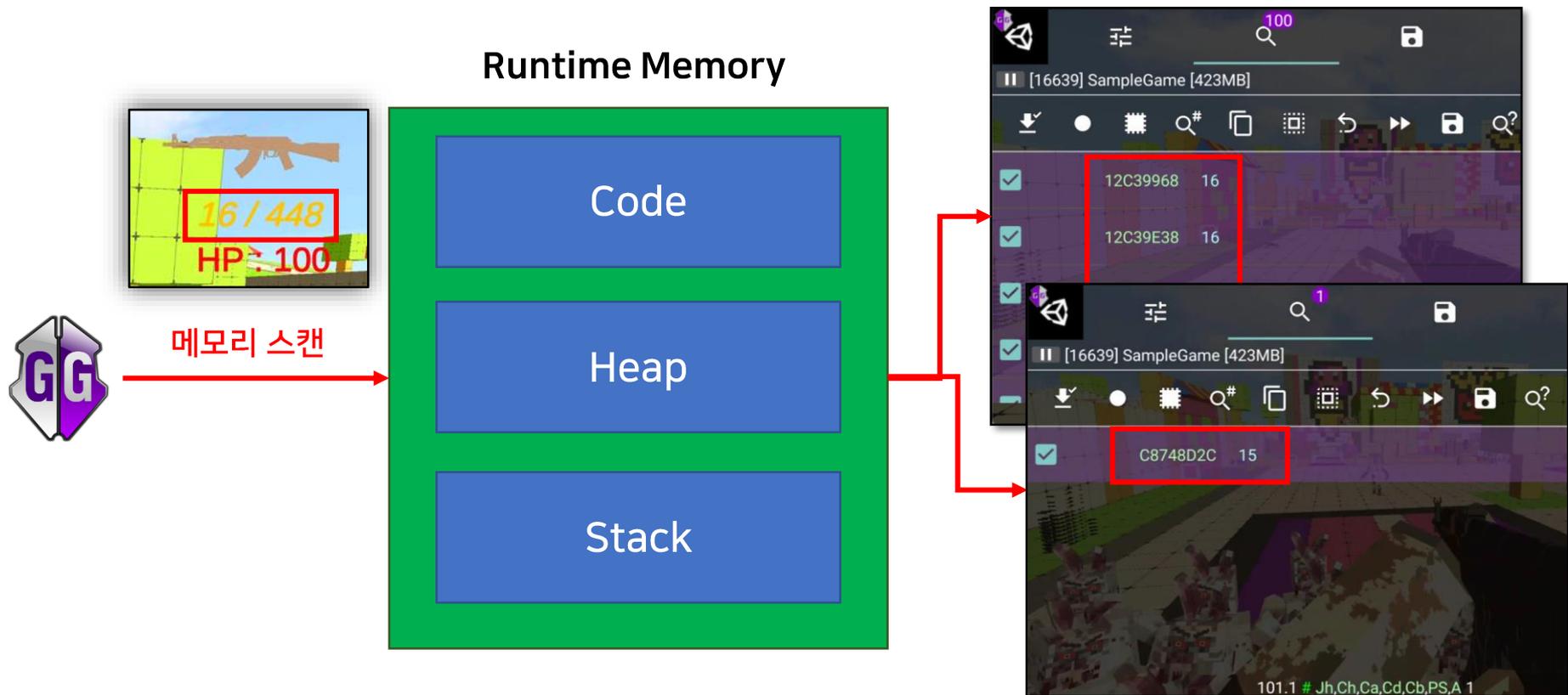
- 루팅의 목적?
  - 주어진 권한 이상으로 뭔가를 하기 위함



# Android 모바일 해킹 유형 및 대응

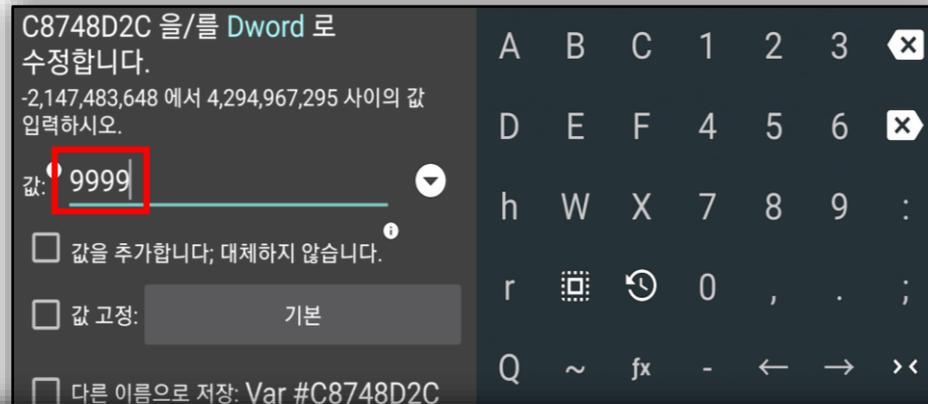
- 치팅툴

- 어떻게 동작할까?



# Android 모바일 해킹 유형 및 대응

- 메모리 변조



# Android 모바일 해킹 유형 및 대응

- 두 가지 분석 방법
  - 정적 분석
  - 동적 분석

```

.text:00081BF0      public gets
.text:00081BF0      gets
.text:00081BF0      ; arg_0 = dword ptr 8
.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
.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      byte   ptr [ecx+20h], 0
.text:00081C1B      jnz    short loc_81C25
.text:00081C1D      mov     [esp], eax
.text:00081C20      call   _flockfile
.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](#)

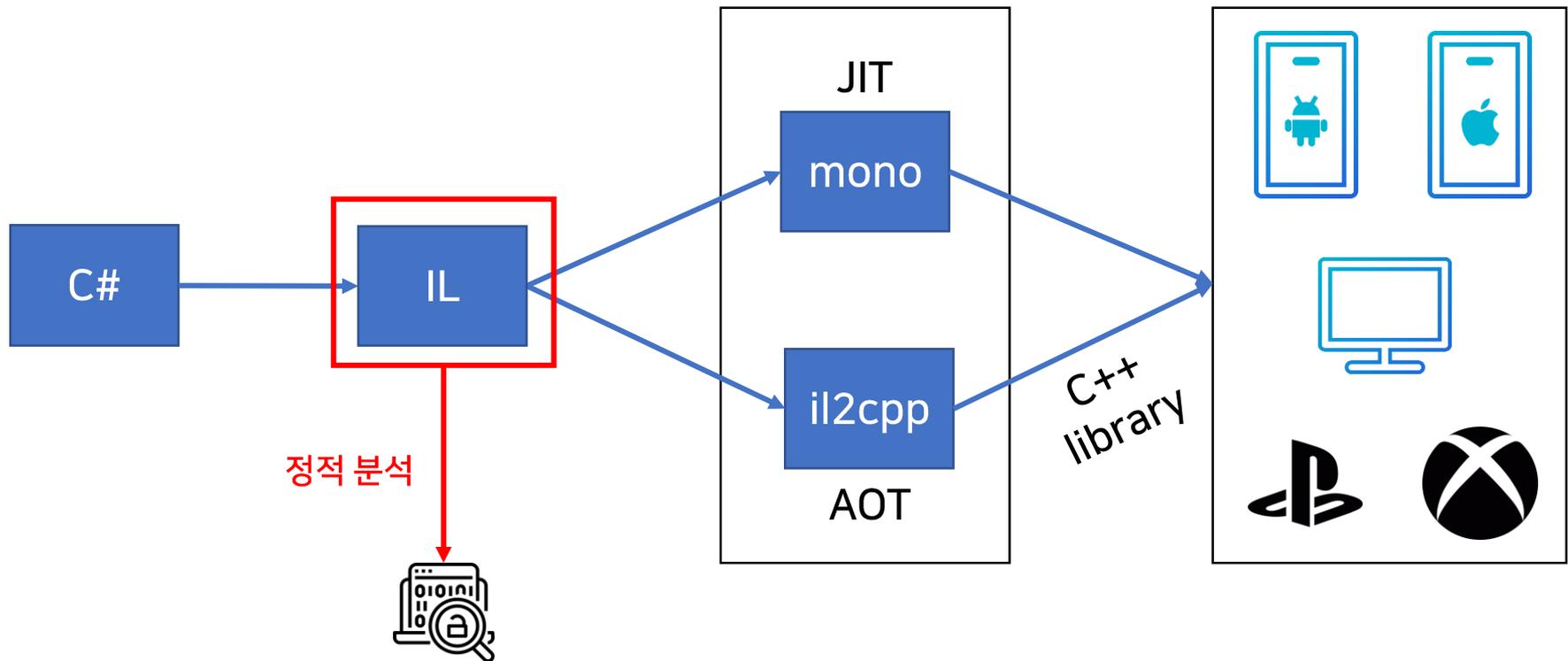
Dynamic instrumentation toolkit for developers, reverse engineers, and security researchers.

```

Breakpoint 1, 0x00008054 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
$sp : 0xbffff850 -> 0x00000001
$lr : 0x00000000
$pc : 0x00008054 -> <start+0> push {r11, lr}
$cpsr : [thumb fast interrupt overflow carry zero negative]
-----[ stack ]-----
0xbffff850+0x00: 0x00000001 <-$sp
0xbffff854+0x04: 0xbffff94e -> "/home/pi/lab/gdb-example"
0xbffff858+0x08: 0x00000000
0xbffff85c+0x0c: 0xbffff967 -> "TERM=vt100"
0xbffff860+0x10: 0xbffff972 -> "SHELL=/bin/bash"
0xbffff864+0x14: 0xbffff992 -> 0x5f474458
0xbffff868+0x18: 0xbffff9d1 -> "LC_ALL=en_US.UTF-8"
0xbffff86c+0x1c: 0xbffff9e4 -> "USER=pi"
-----[ code:armv4t ]-----
0x803c      andeq r8, r0, r0
0x8040      andeq r8, r0, r0
0x8044      muleq r0, r4, r0
0x8048      muleq r0, r4, r0
0x804c      andeq r0, r0, r5
0x8050      andeq r8, r0, r0
-> 0x8054 <start+0>      push {r11, lr}
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>
    
```

# Android 모바일 해킹 유형 및 대응

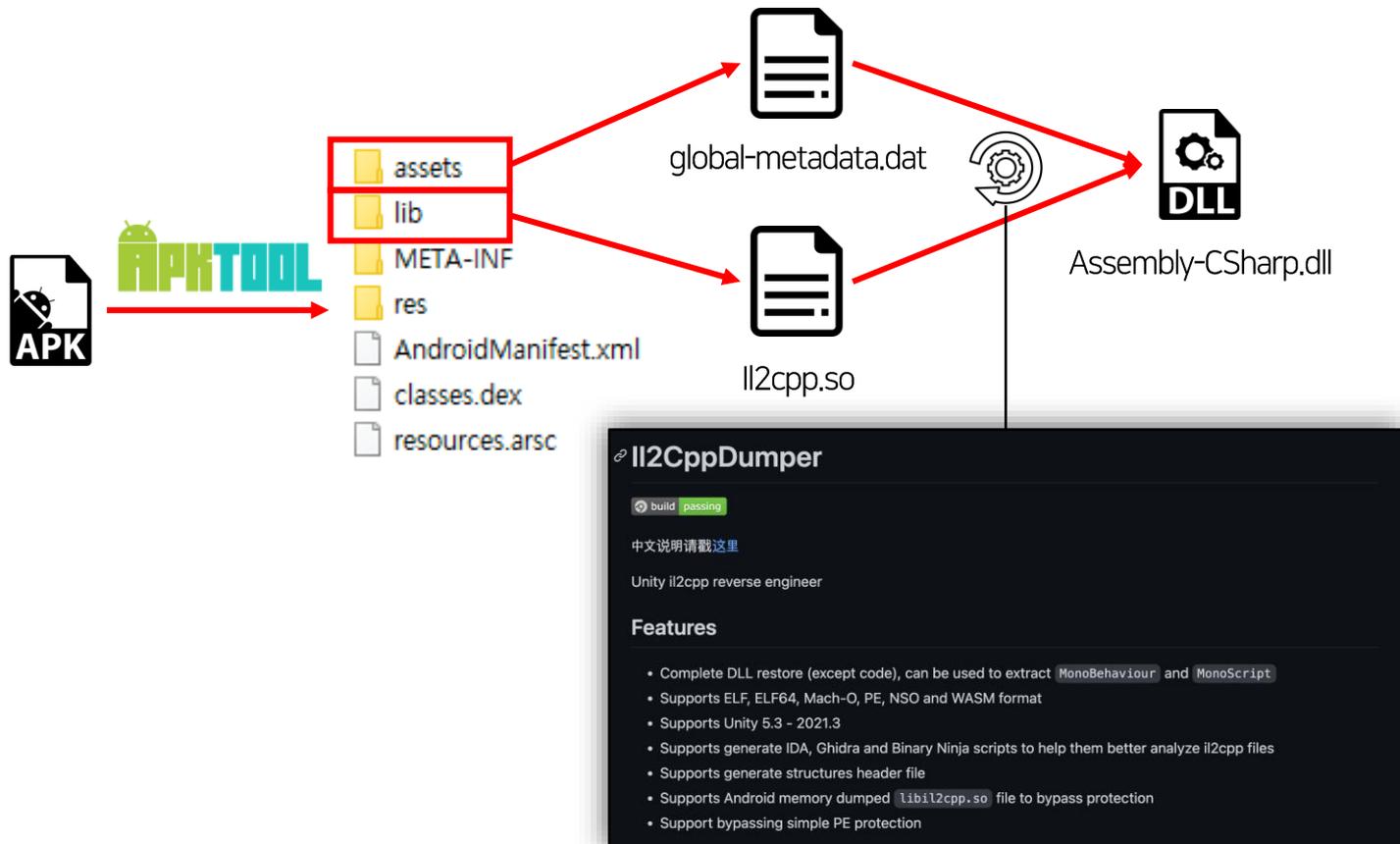
- 유니티 엔진 구조



# Android 모바일 해킹 유형 및 대응

- 정적 분석

- Assembly-CSharp.dll 추출



# Android 모바일 해킹 유형 및 대응

- 정적 분석
  - Assembly-CSharp.dll 디컴파일

The image shows a decompiler interface with the following components:

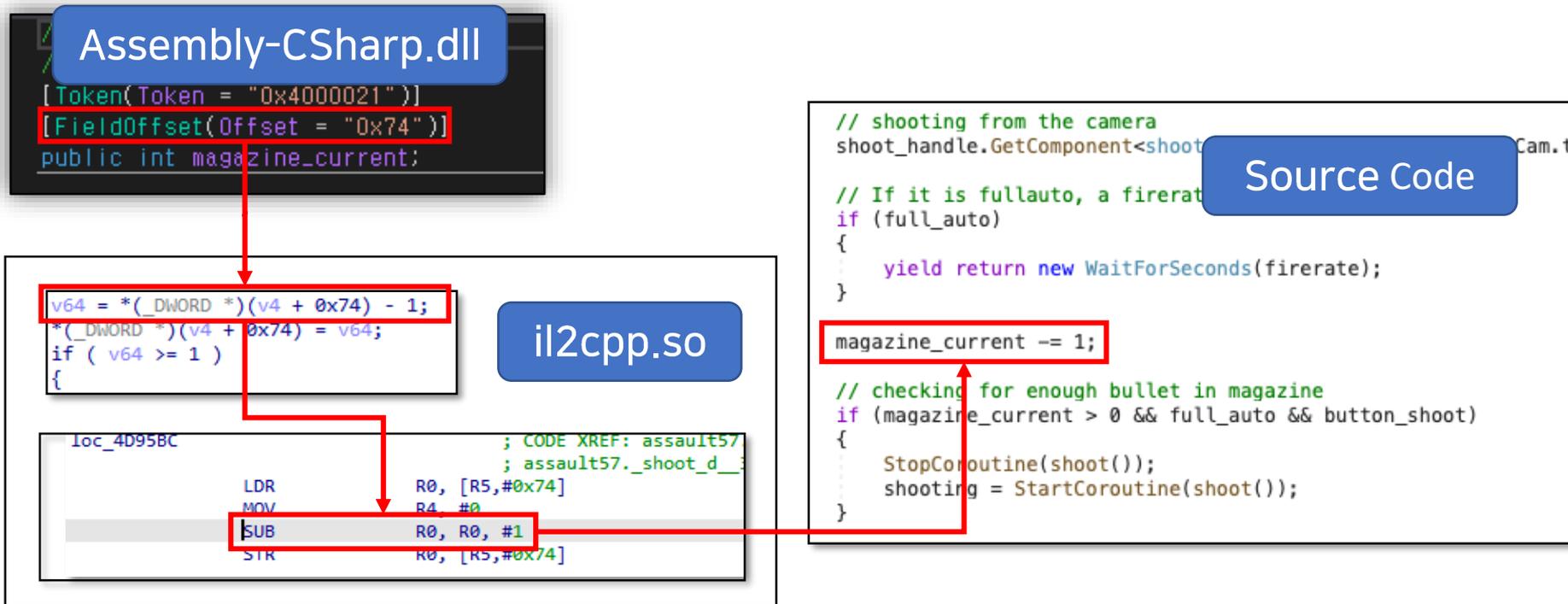
- Assembly-CSharp.dll**: A blue box pointing to the decompiled assembly code on the right.
- Source Code**: A blue box pointing to the original C# source code on the left.
- il2cpp.so**: A blue box pointing to the assembly code, indicating the decompiler used.
- Source Code**: A white box containing the C# code for the `shoot()` method:
 

```
public IEnumerator shoot()
{
    yield return new WaitForSeconds(1);
    // increasing the spread, while in automatic fire
    current_spread += spread_height;
    Vector3 Add_spread = Shoot_start_point.transform.forward;
    float hor = Random.Range(-current_spread, current_spread);
    float ver = Random.Range(-current_spread, current_spread);
    Add_spread = new Vector3(hor, ver, 0);
    // pushing the shoot animation completly back to 1
    Power_bolt = 1;
}
```
- Assembly Code**: A white box containing the assembly code for the `shoot()` method:
 

```
int __fastcall old_pistol_shoot(int a1)
{
    int v2; // r5
    int v3; // r0
    if ( !byte_64DD5F )
    {
        sub_176CDC((int)old_pistol_shoot_d_36_TypeInfo);
        byte_64DD5F = 1;
    }
    v2 = sub_176DA8(old_pistol_shoot_d_36_TypeInfo);
    Renderer_1_ctor_27((Renderer_1 *)v2, 0, 0);
    if ( !v2 )
        sub_176DB6(v3);
    *(DWORD*)(v2 + 0x10) = a1;
    sub_176C80(v2 + 0x10, a1);
    return v2;
}
```

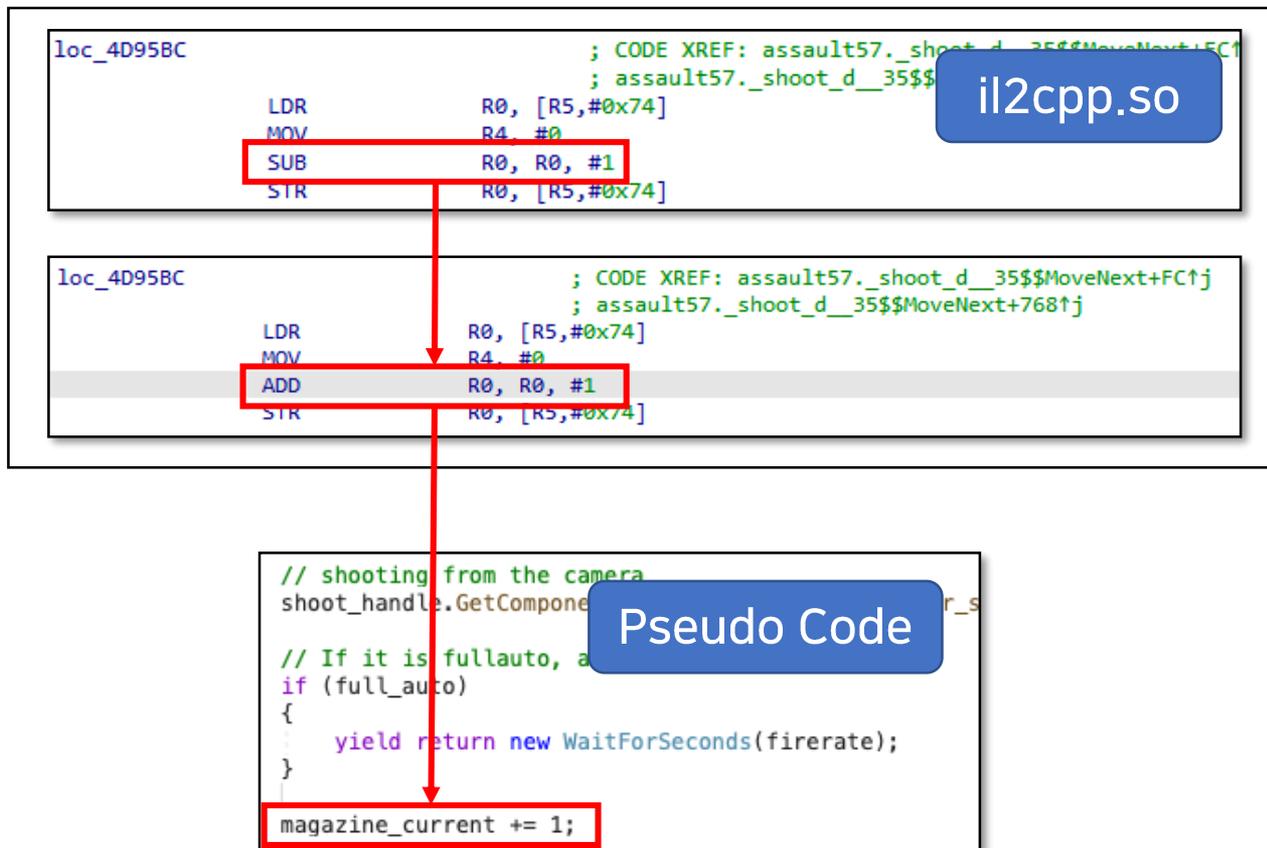
# Android 모바일 해킹 유형 및 대응

- 라이브러리 변조
  - 탄창의 총알을 발사할수록 증가하게 해볼까?



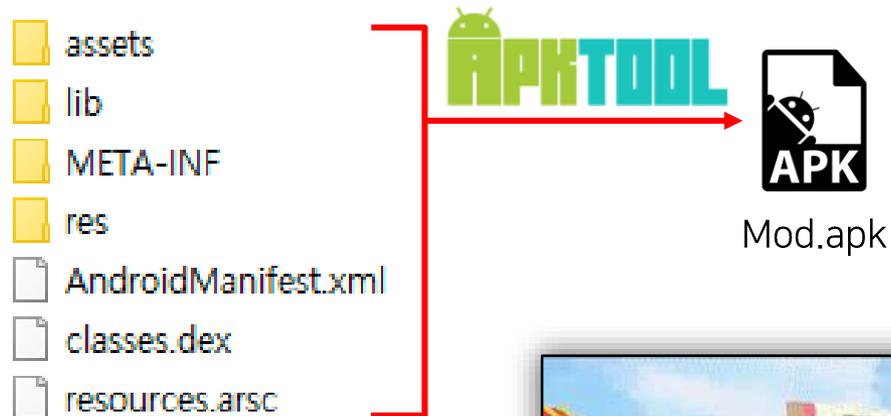
# Android 모바일 해킹 유형 및 대응

- 라이브러리 변조
  - 실행 코드 변조



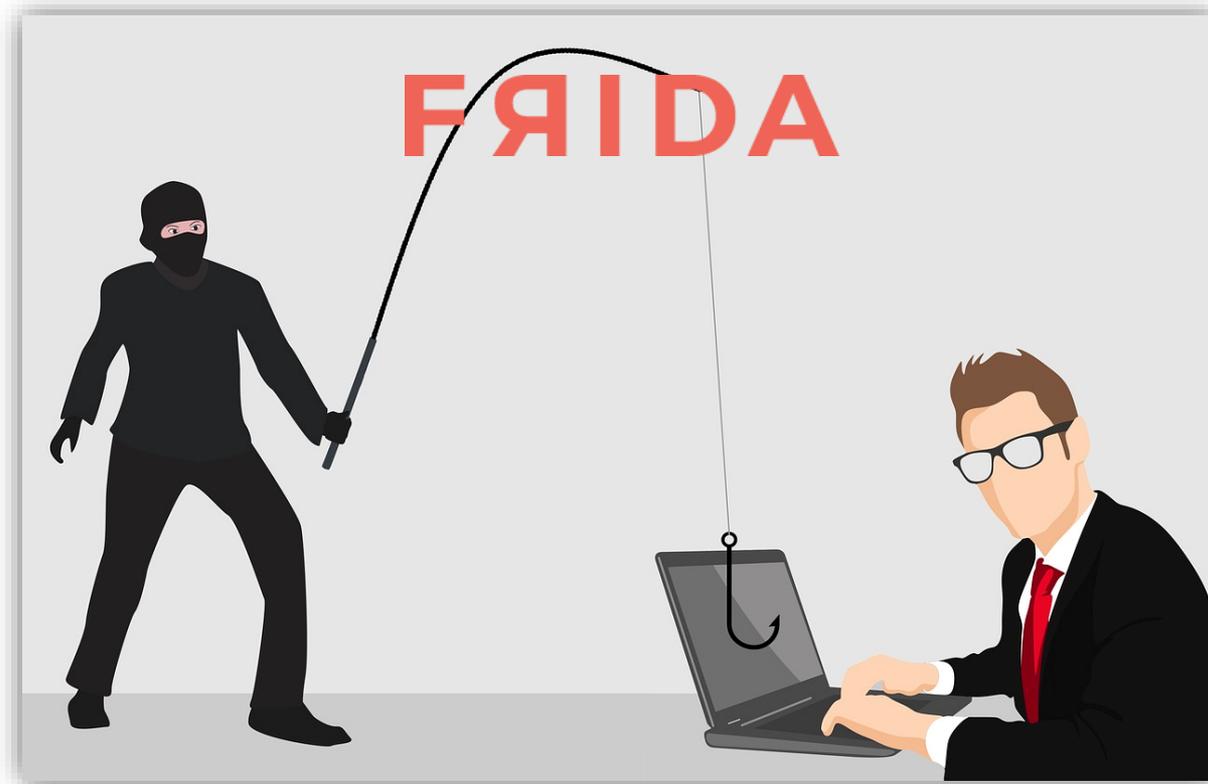
# Android 모바일 해킹 유형 및 대응

- 리패키징



# Android 모바일 해킹 유형 및 대응

- 동적 분석
  - Frida
    - Dynamic Binary Instrumentation toolkit



# Android 모바일 해킹 유형 및 대응

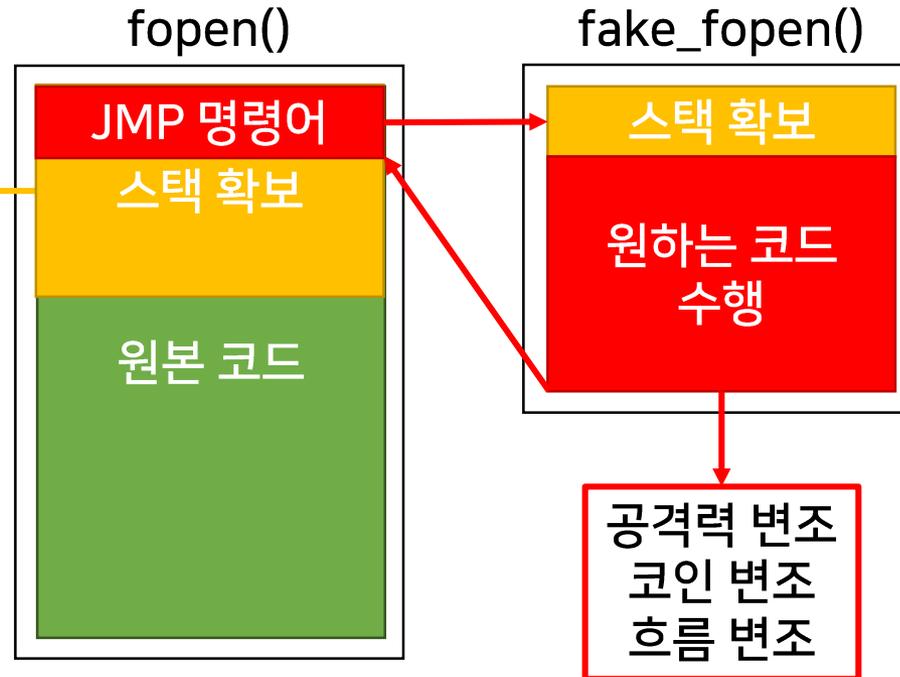
- 동적 분석

- 프리다 후킹 동작 원리

- 인라인 후킹

```

; __unwind { // 811BC
PUSH      {R4-R7,LR}
ADD       R7, SP, #0xC
PUSH.W    {R8-R11}
SUB       SP, SP, #0x8C
STR       R0, [SP,#0x10]
SUBS     R0, R3, R2
    
```



# Android 모바일 해킹 유형 및 대응

- 동적 분석

The image illustrates a dynamic analysis workflow in an Android application. It starts with a JavaScript hook in a module named 'libil2cpp.so'. The hook is triggered at address 0x4d95c4, where the context is logged. The assembly instructions show the execution flow: loading a value from memory (R5, #0x74), moving it to register R4, subtracting 1 from R0, and storing the result back to memory (R5, #0x74). This sequence of operations is linked to a log output showing the context object, where the 'r0' register value is highlighted as '0x19'.

```
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

Function name

- ✓ j\_My\_\_faccessat
- ✓ My\_fork(void)
- ✓ My\_gettid
- ✓ my\_memcmp
- ✓ my\_memcpy
- ✓ My\_opendir
- ✓ My\_open(char const\*,int,...)
- ✓ My\_\_faccessat
- ✓ My\_\_fchmodat
- ✓ My\_\_fgetxattr
- ✓ My\_\_flistxattr
- ✓ My\_\_fsetxattr
- ✓ My\_\_connect
- ✓ My\_\_fcntl64
- ✓ My\_\_fstafs64
- ✓ My\_\_getcwd
- ✓ My\_\_openat
- ✓ My\_\_stats64
- ✓ My\_\_acct
- ✓ My\_\_bind
- ✓ My\_\_chdir
- ✓ My\_\_execve
- ✓ My\_\_fchdir
- ✓ My\_\_fremovexattr
- ✓ My\_\_fstata64
- ✓ My\_\_fstata64
- ✓ My\_\_ftruncate64
- ✓ My\_\_getsockname
- ✓ My\_\_getsockopt

Line 2 of 69

Graph overview

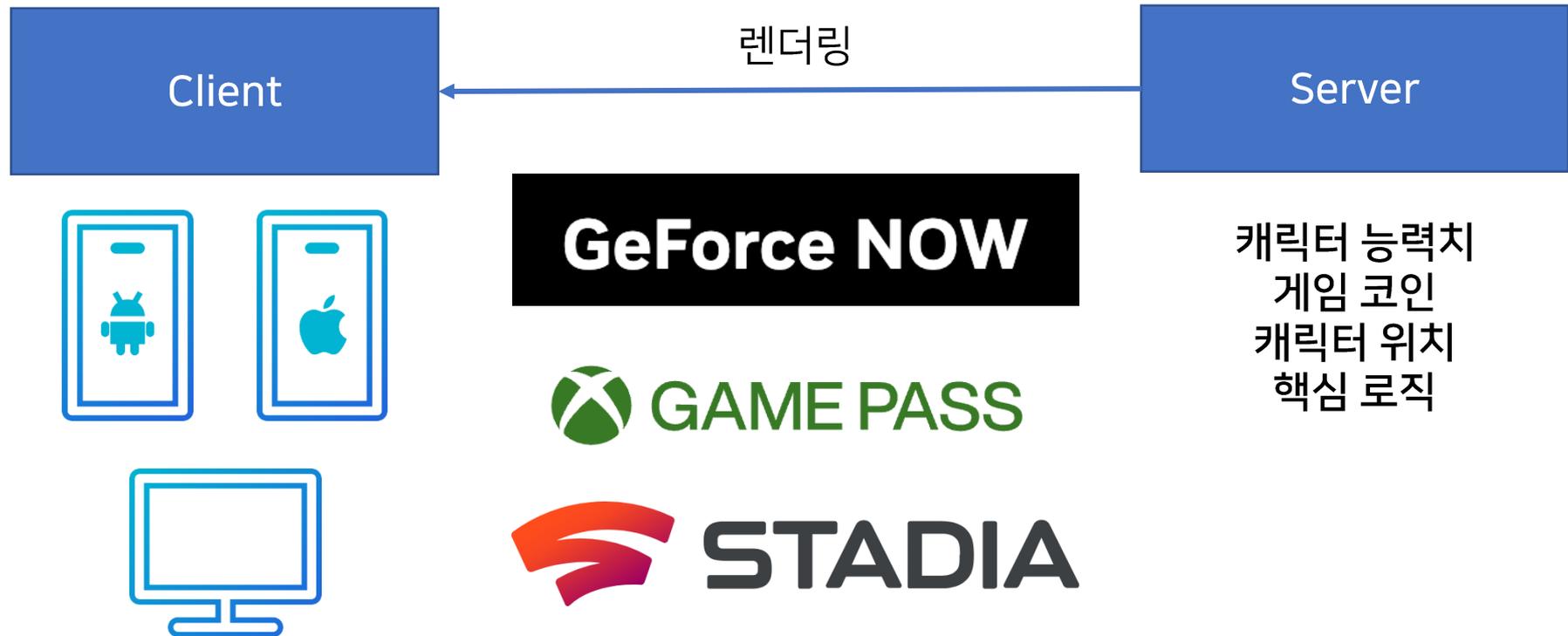



```

27 result = -1;
28 v4 = -1;
29 if ( v1 >= 0 )
30 {
31     v3 = j_fdopen(v1, "r");
32     while ( !j_feof(v3) &
33     {
34         if ( !j_strncmp(&v5
35         {
36             j_sscanf(&v5, "Tg
37             break;
38         }
39     }
40     j_fclose(v3);
41     result = v4;
42 }
43 return result;
44 }
                
```

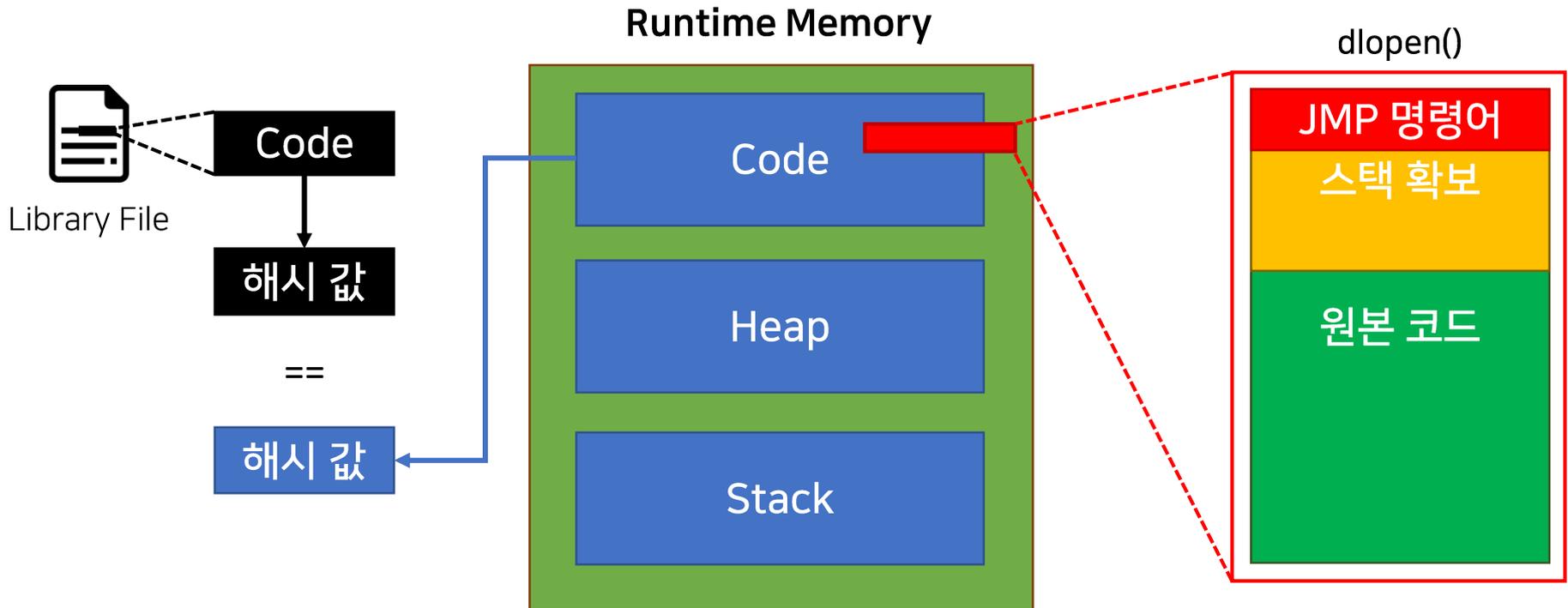
# Android 모바일 해킹 유형 및 대응

- 대응 방안
  - 서버 검증



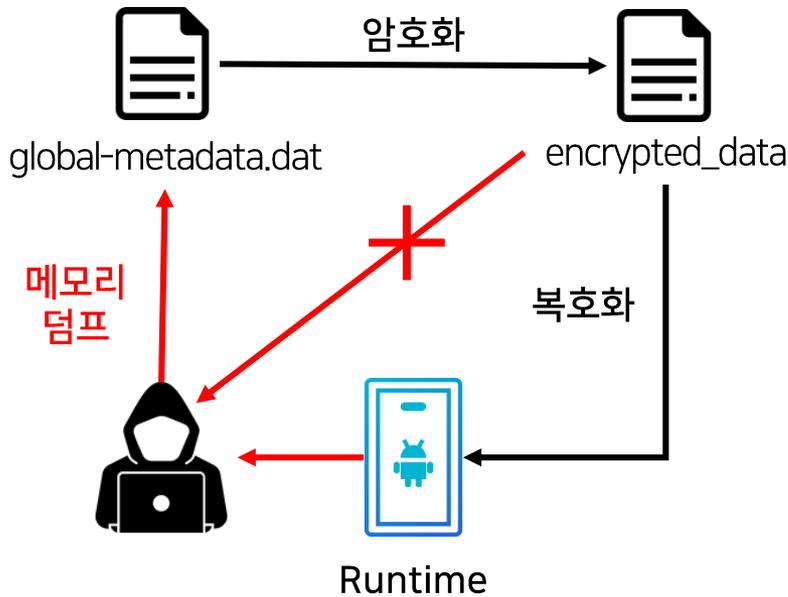
# Android 모바일 해킹 유형 및 대응

- 대응 방안
  - 라이브러리 코드 변조 & 인라인 후킹 탐지



# Android 모바일 해킹 유형 및 대응

- 대응 방안
  - 유니티 엔진 보호



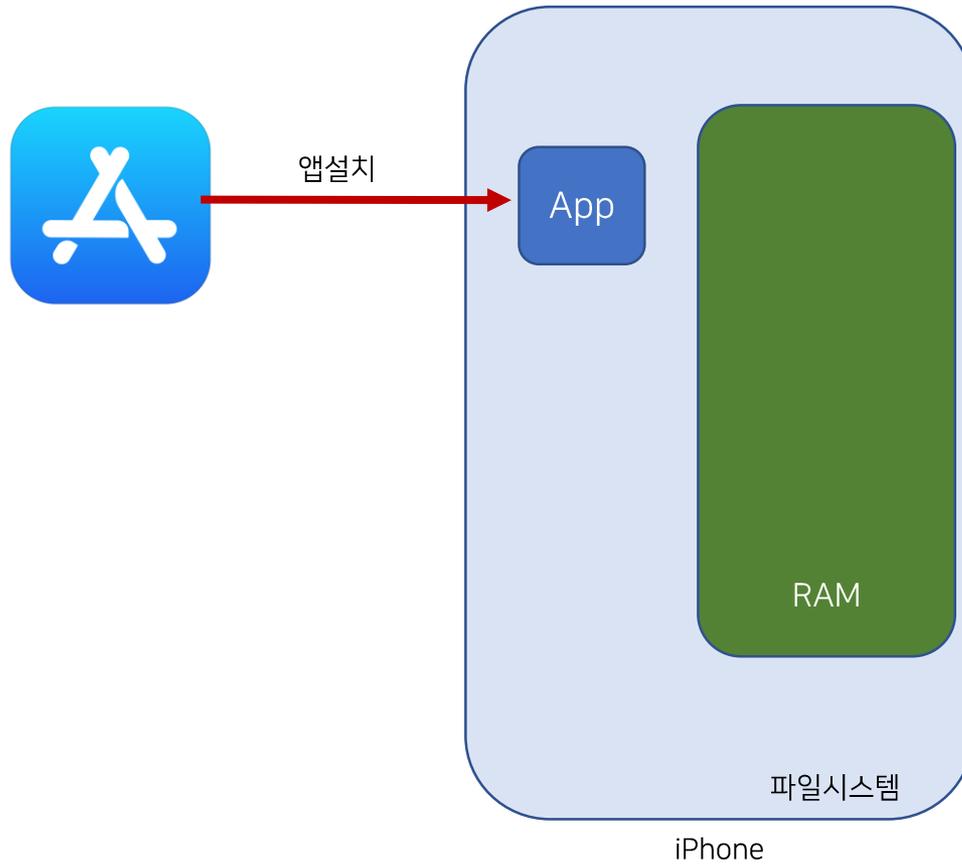
```

// 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.")]
    [Header("Gun Camera Options")]
    // Namespace:
    public class oVkJFrPXxWudEoOkXU0qEZm : MonoBehaviour // TypeDefIndex: 3388
    {
        // Fields
        private Animator kMVQ; // 0xC
        [Header("Gun Camera")]
        [Header("UI Weapon Name")]
        public Camera haptFYeug; // 0x10
        [Tooltip("How fast the camera field of view changes when aiming.")]
        [Header("Gun Camera Options")]
        public float hOG0anpb; // 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 roeRFroLenInpuLipkT; // 0x38
    }
}
    
```

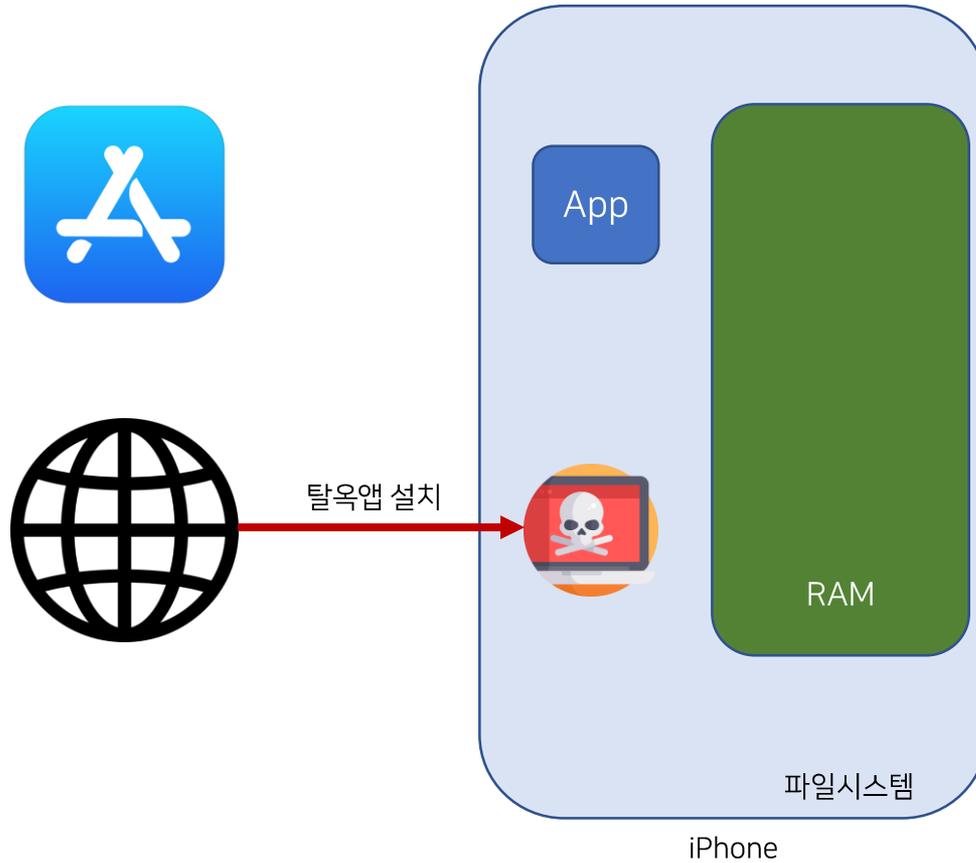
# iOS 모바일 해킹 유형 및 대응

- 해킹 유형
- 대응 방안

# iOS 모바일 해킹 유형: 탈옥



# iOS 모바일 해킹 유형: 탈옥



# iOS 모바일 해킹 유형: 탈옥



iPhone

# iOS 모바일 해킹 유형: 탈옥



iPhone

## 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

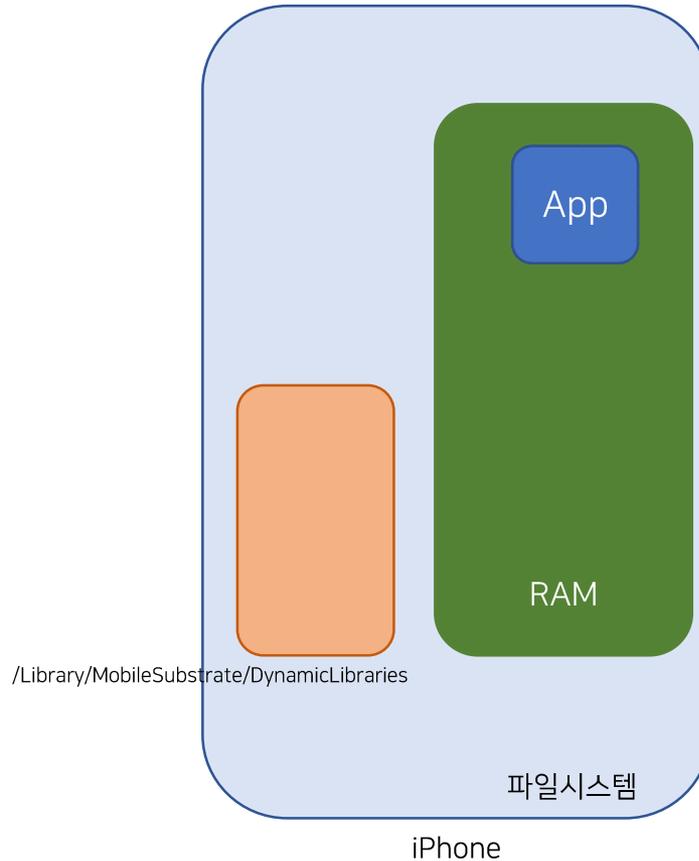
# iOS 모바일 해킹 유형: tweak (dylib injection)



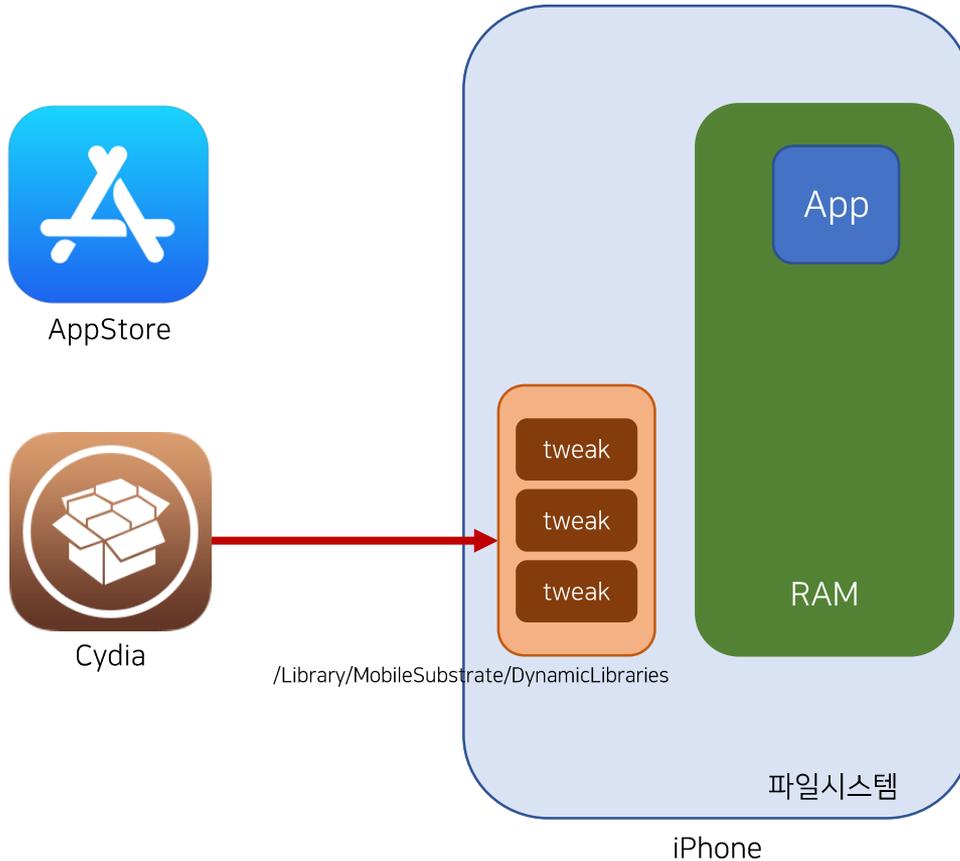
AppStore



Cydia



# iOS 모바일 해킹 유형: tweak (dylib injection)



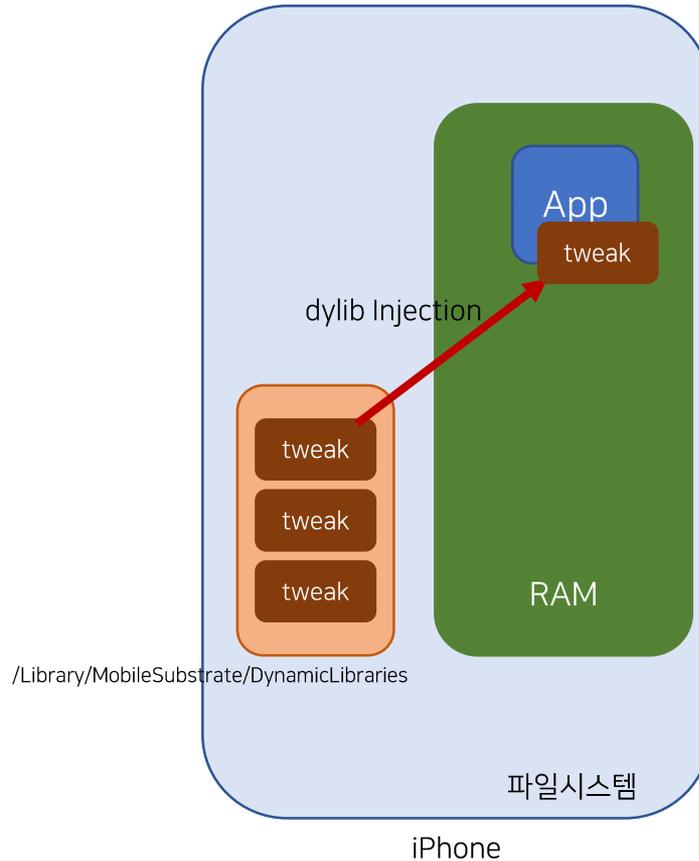
# iOS 모바일 해킹 유형: tweak (dylib injection)



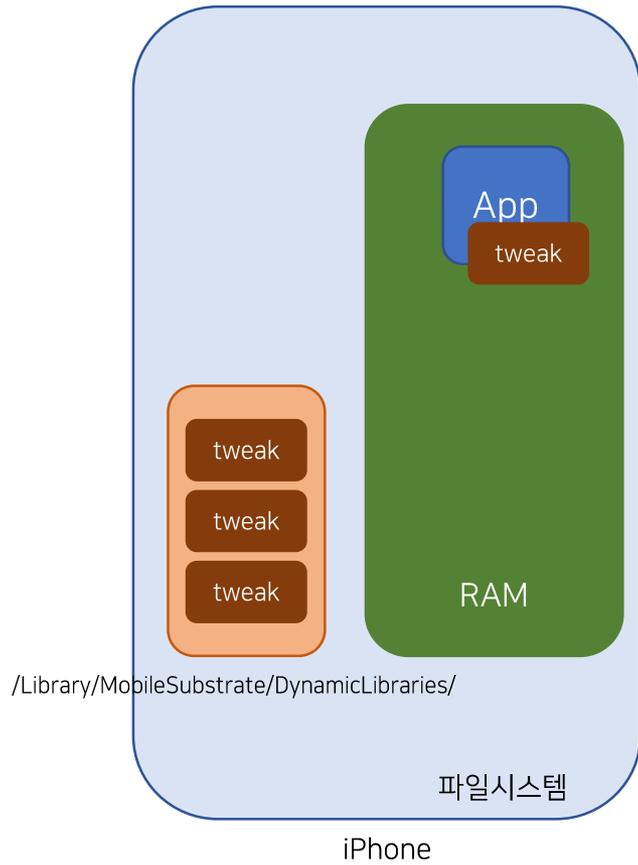
AppStore



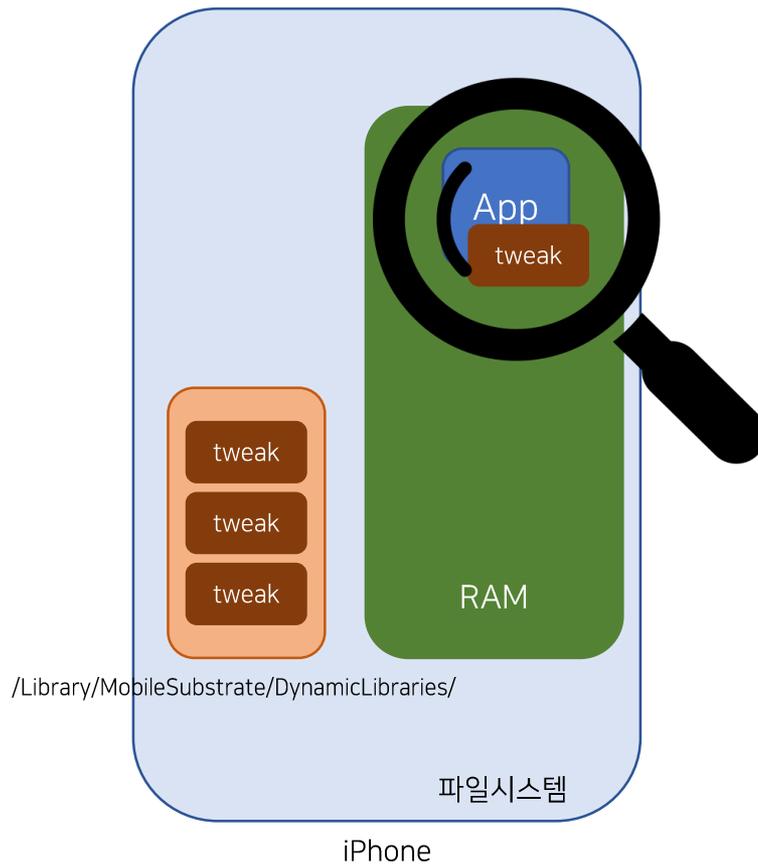
Cydia



# iOS 모바일 해킹 유형: tweak (dylib injection)

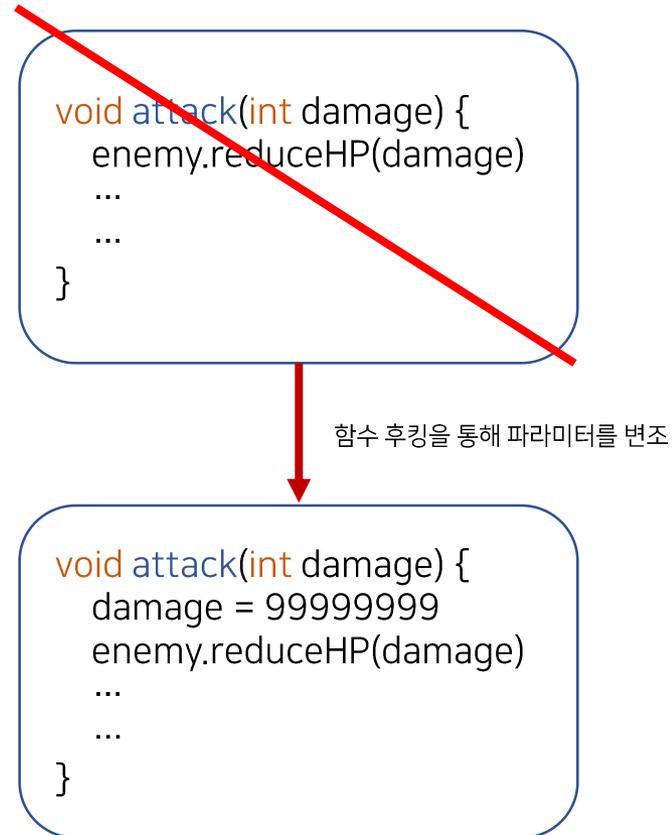
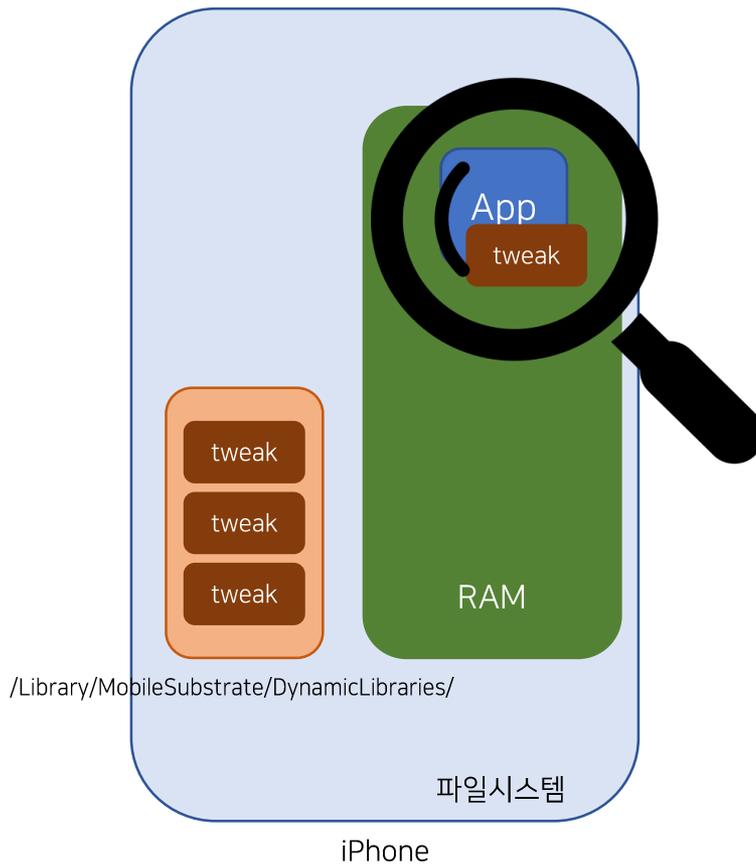


# iOS 모바일 해킹 유형: tweak (dylib injection)

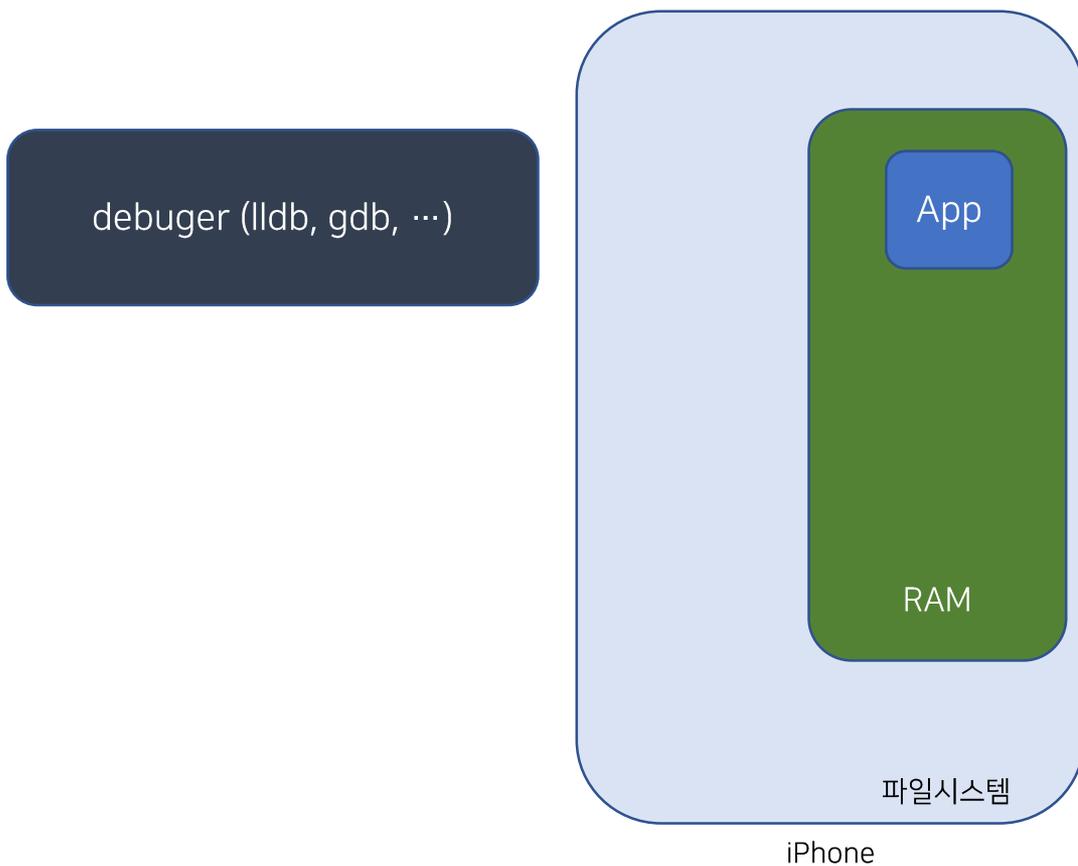


```
void attack(int damage) {  
    enemy.reduceHP(damage)  
    ...  
    ...  
}
```

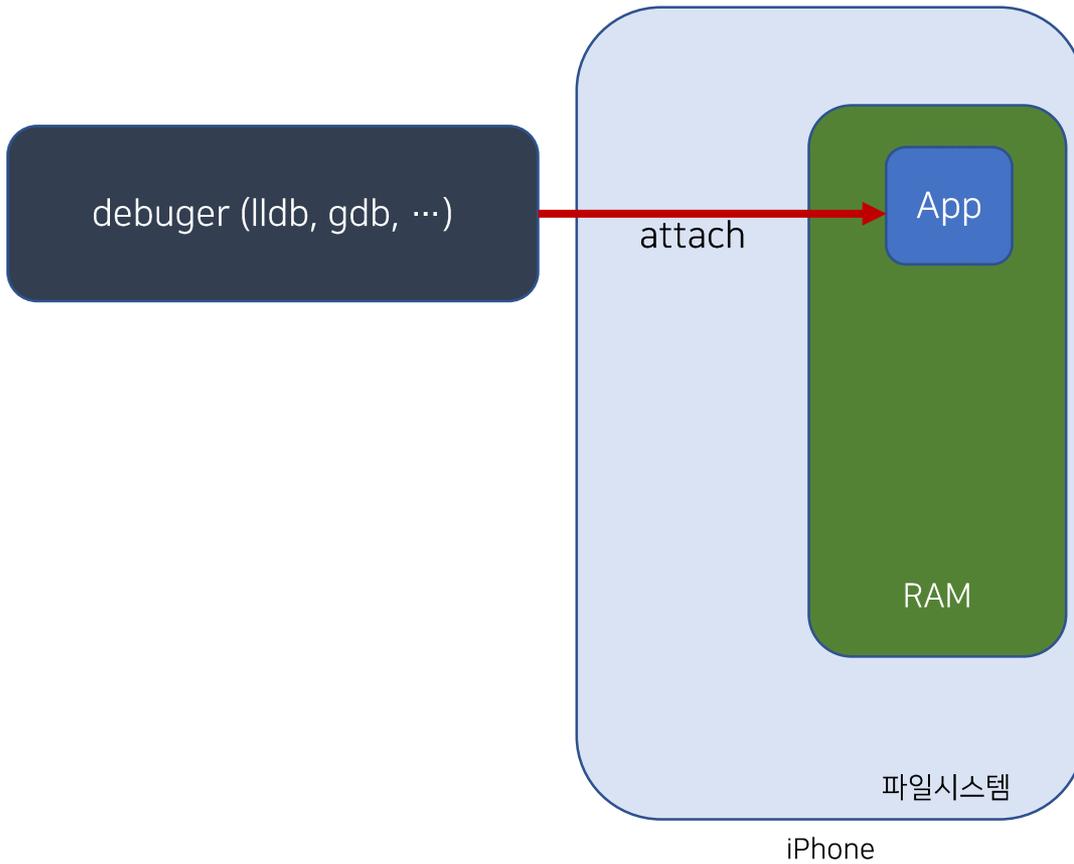
# iOS 모바일 해킹 유형: tweak (dylib injection)



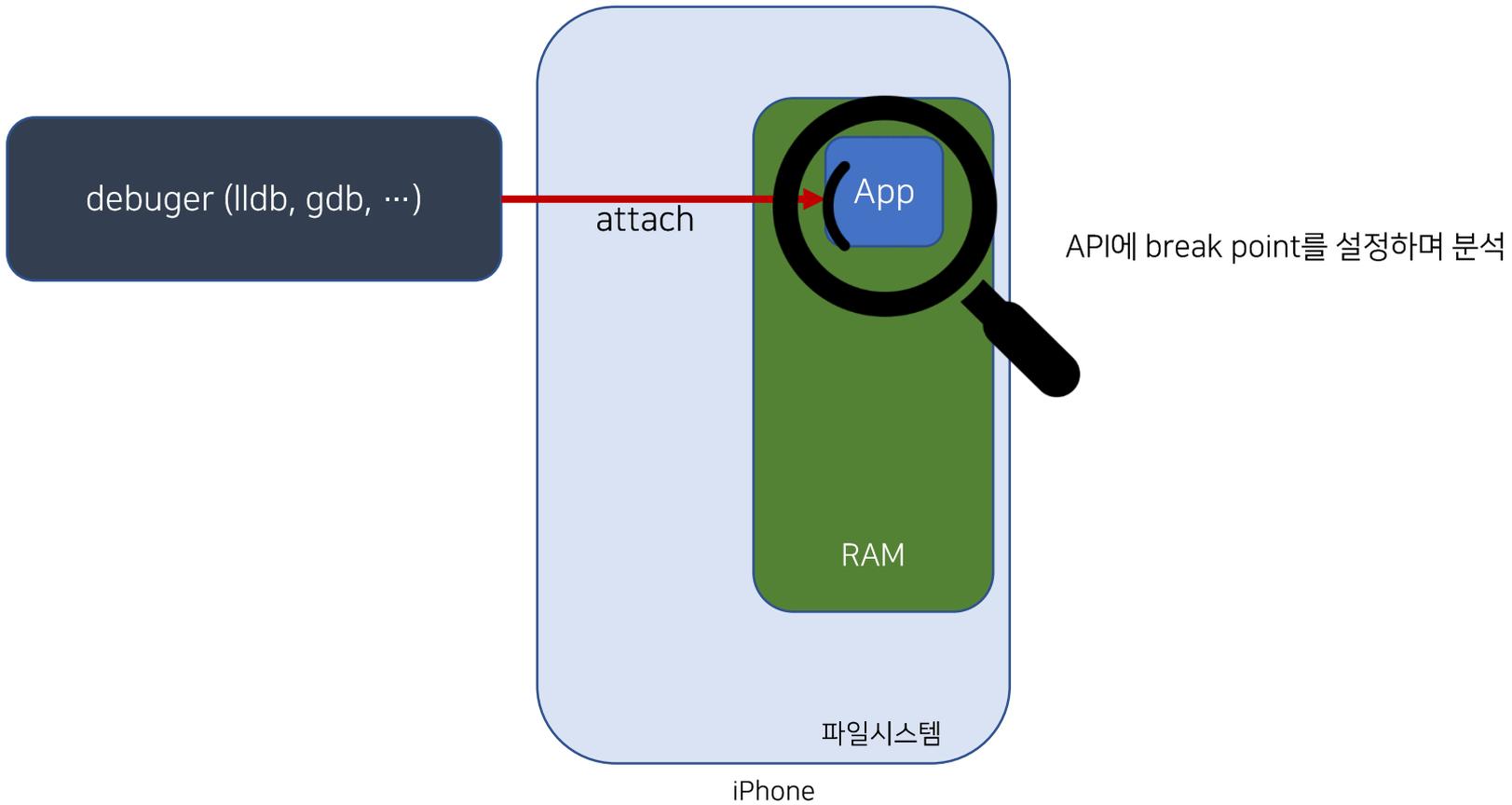
# iOS 모바일 해킹 유형: 디버깅



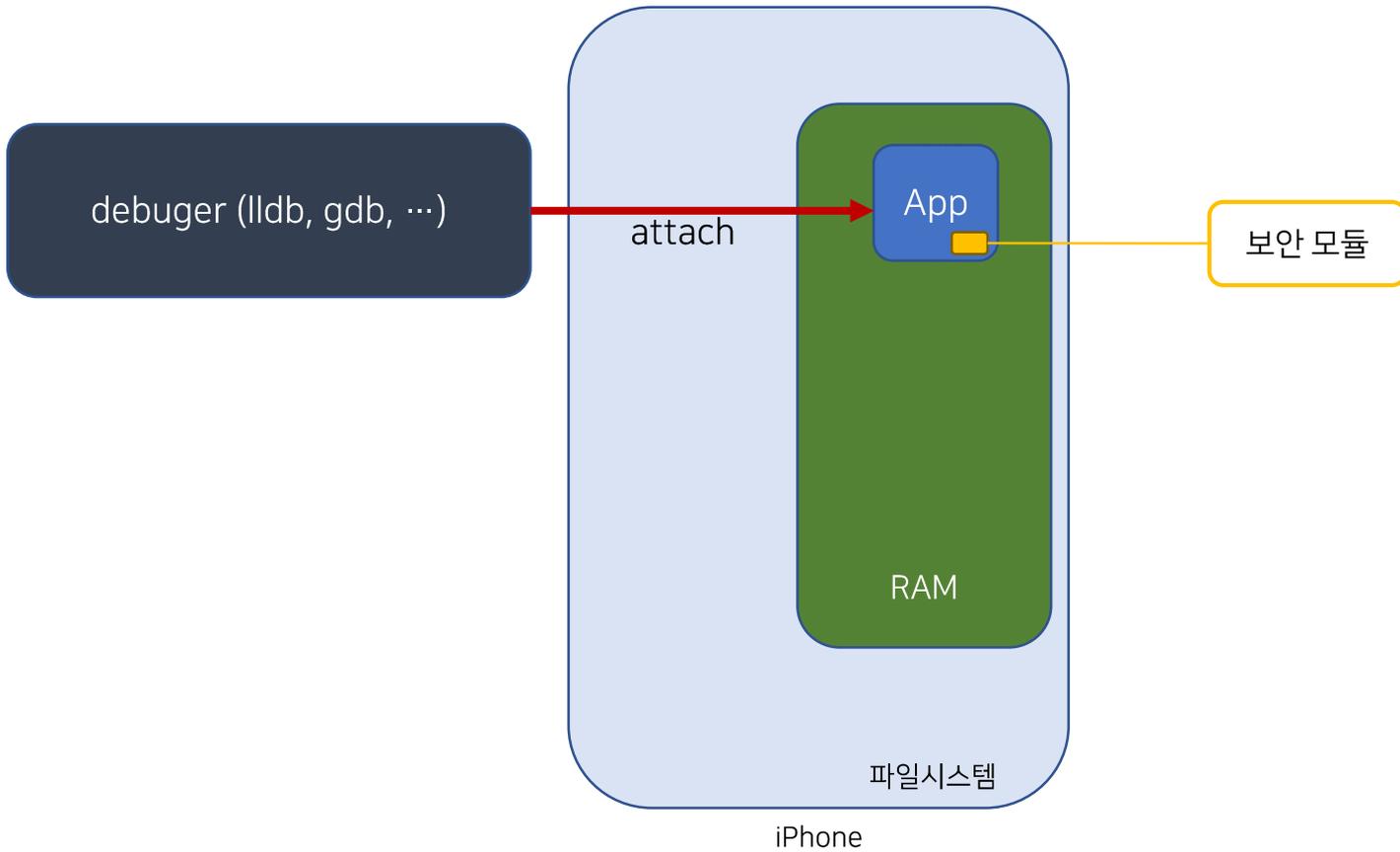
# iOS 모바일 해킹 유형: 디버깅



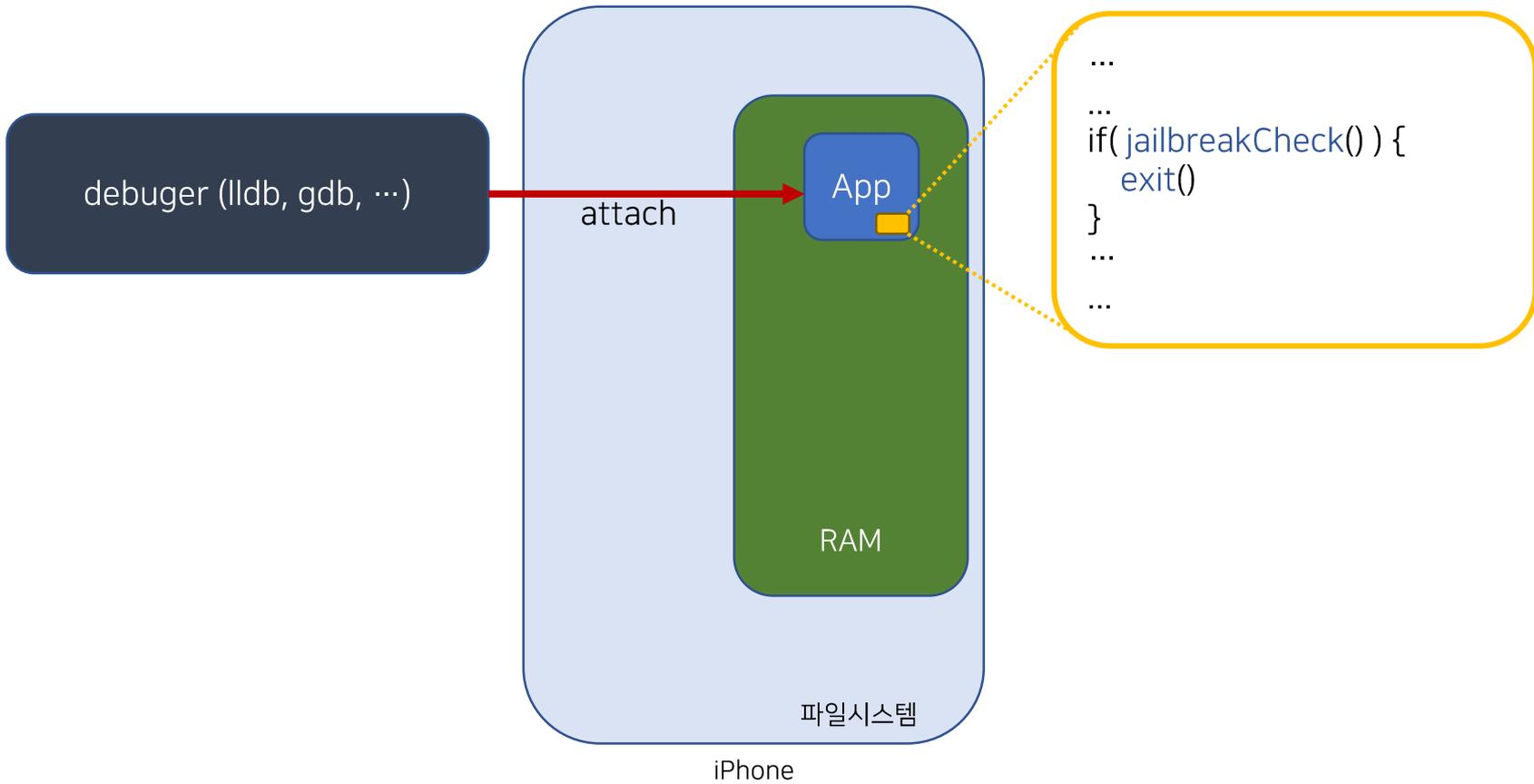
# iOS 모바일 해킹 유형: 디버깅



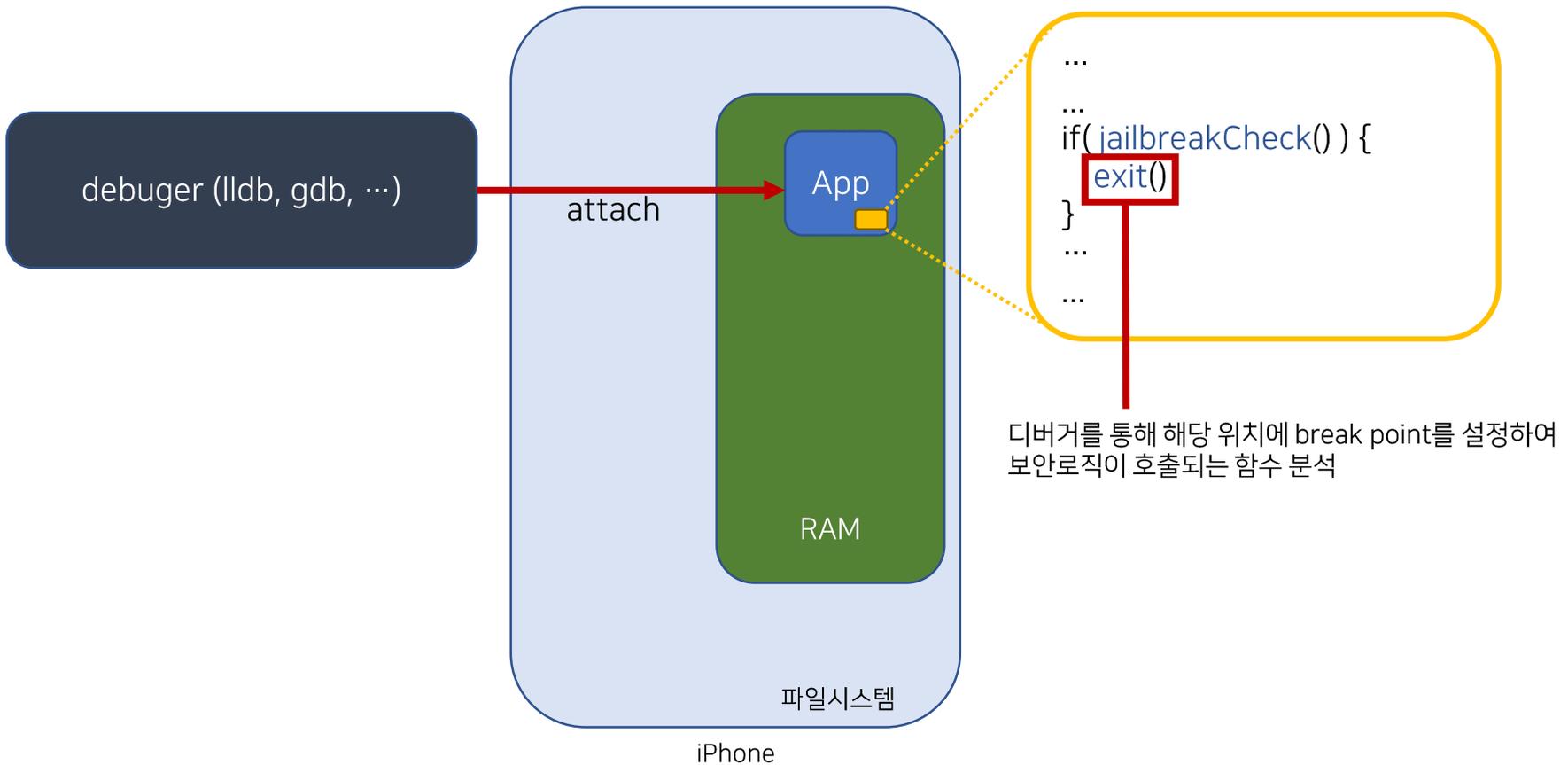
# iOS 모바일 해킹 유형: 디버깅



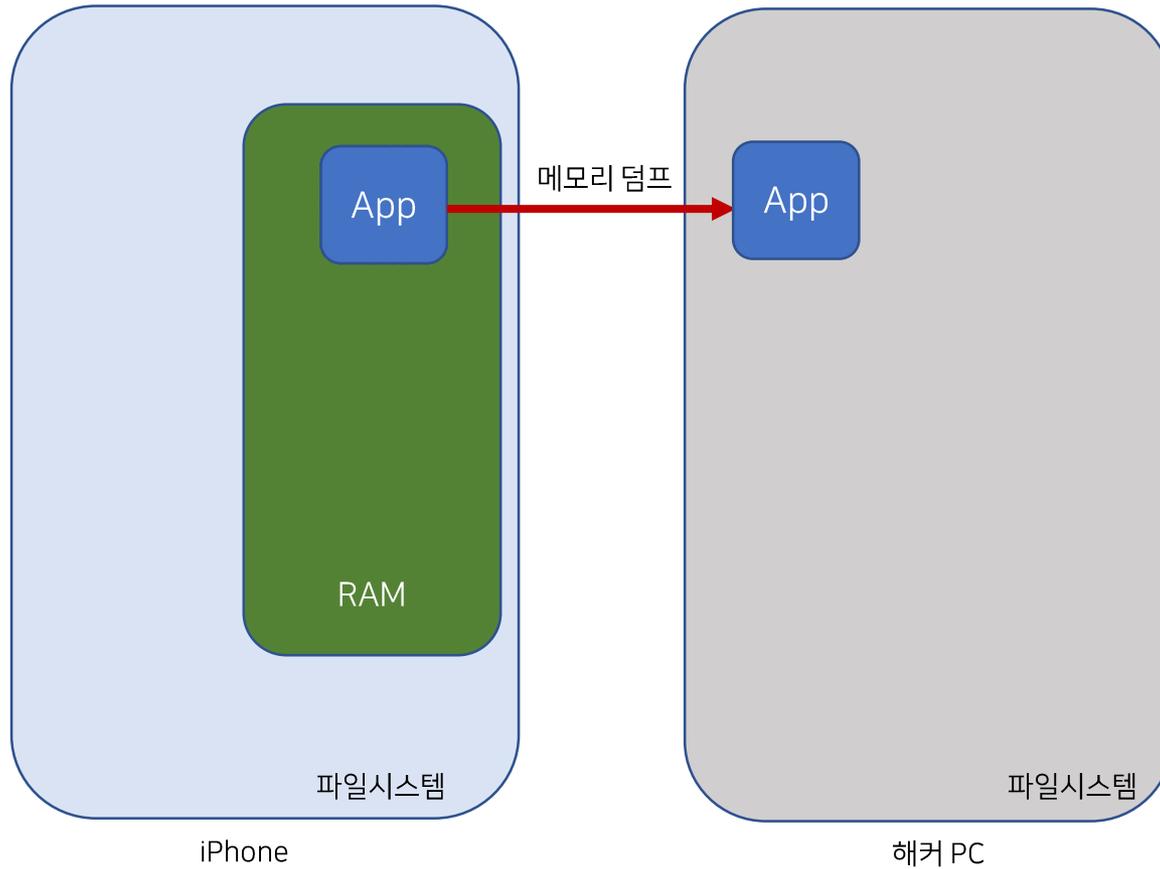
# iOS 모바일 해킹 유형: 디버깅



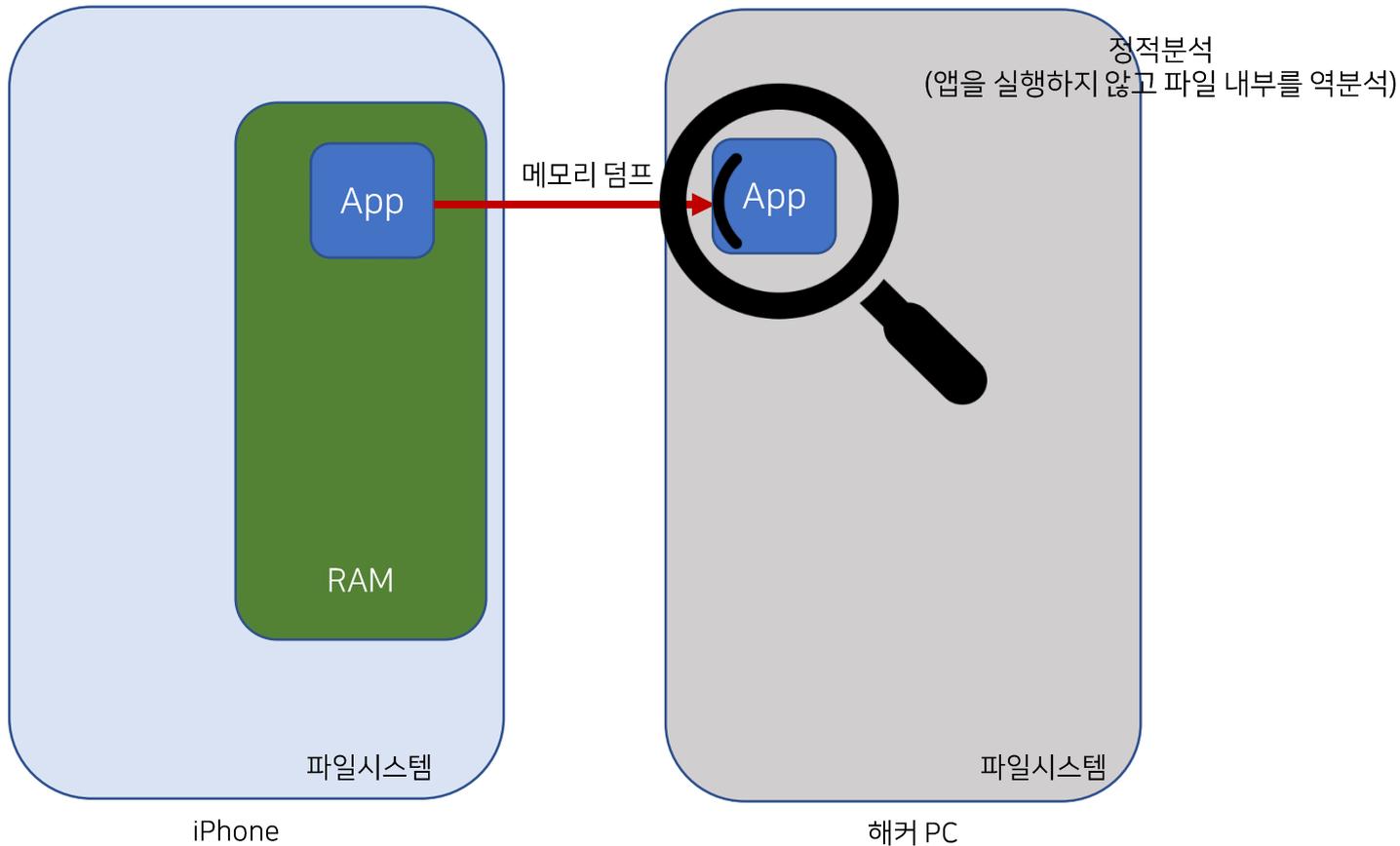
# iOS 모바일 해킹 유형: 디버깅



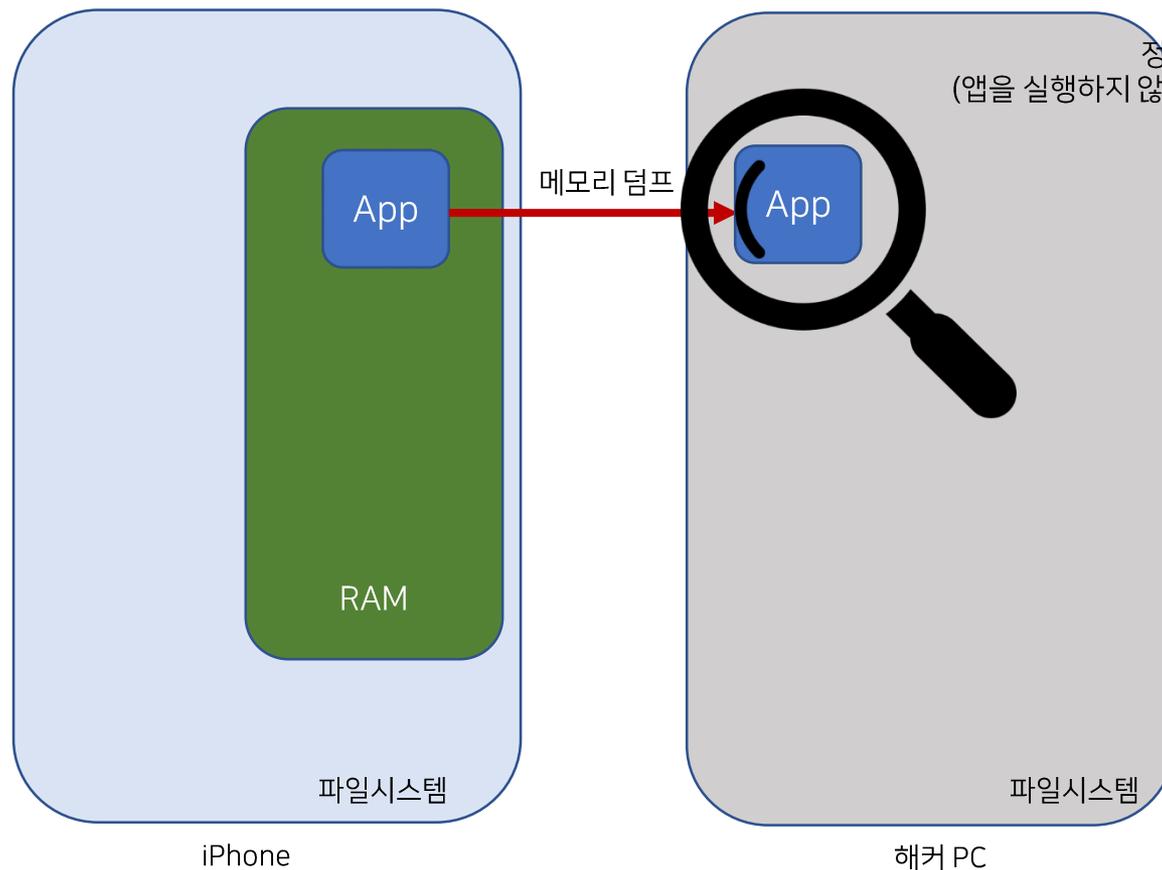
# iOS 모바일 해킹 유형: 앱 위변조



# iOS 모바일 해킹 유형: 앱 위변조



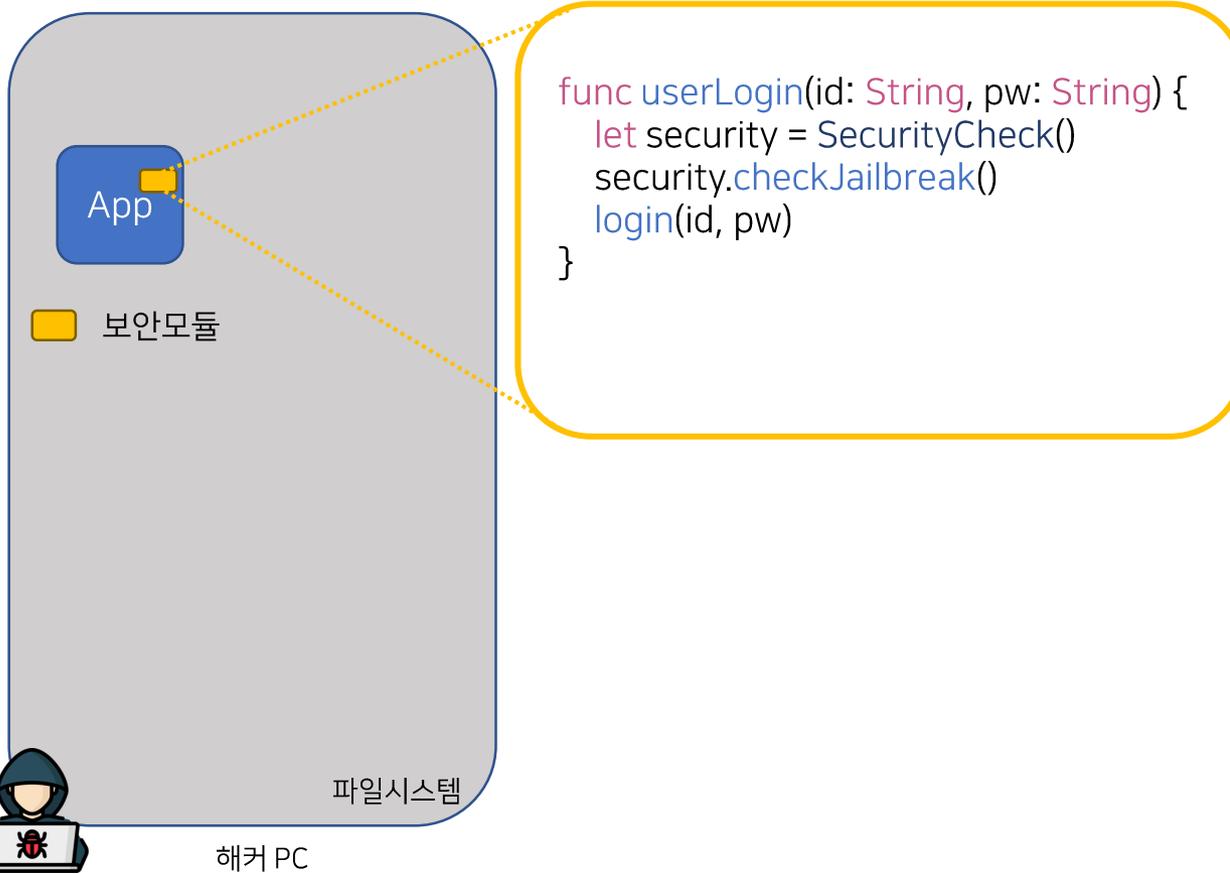
# iOS 모바일 해킹 유형: 앱 위변조



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

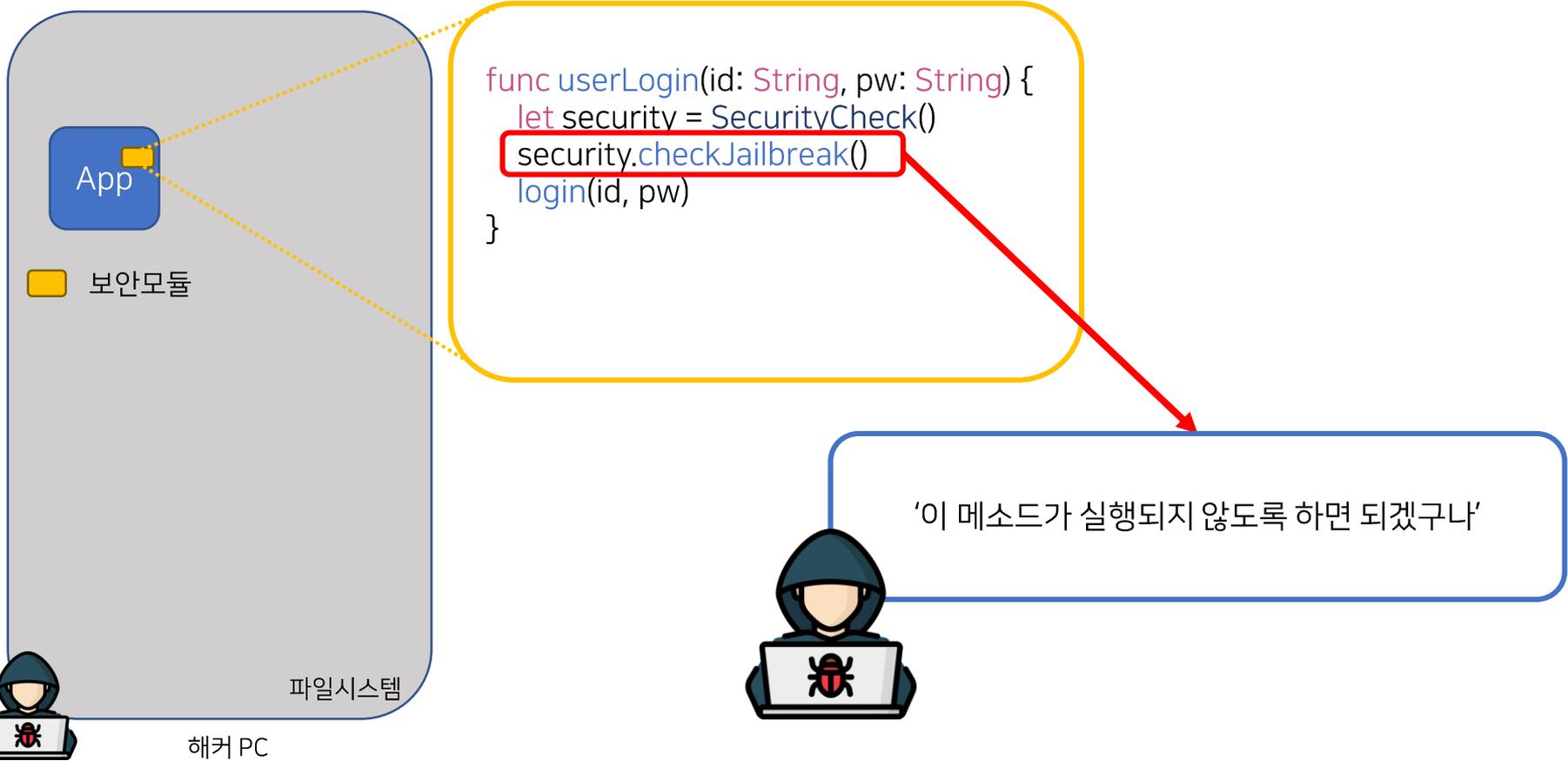
# iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 1. 심볼 확인



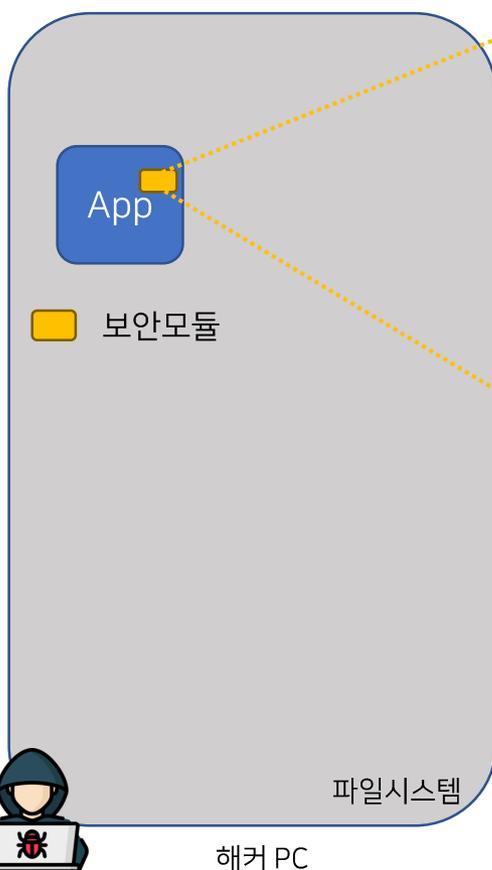
# iOS 모바일 해킹 유형: 앱 위변조

- 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)  
}
```



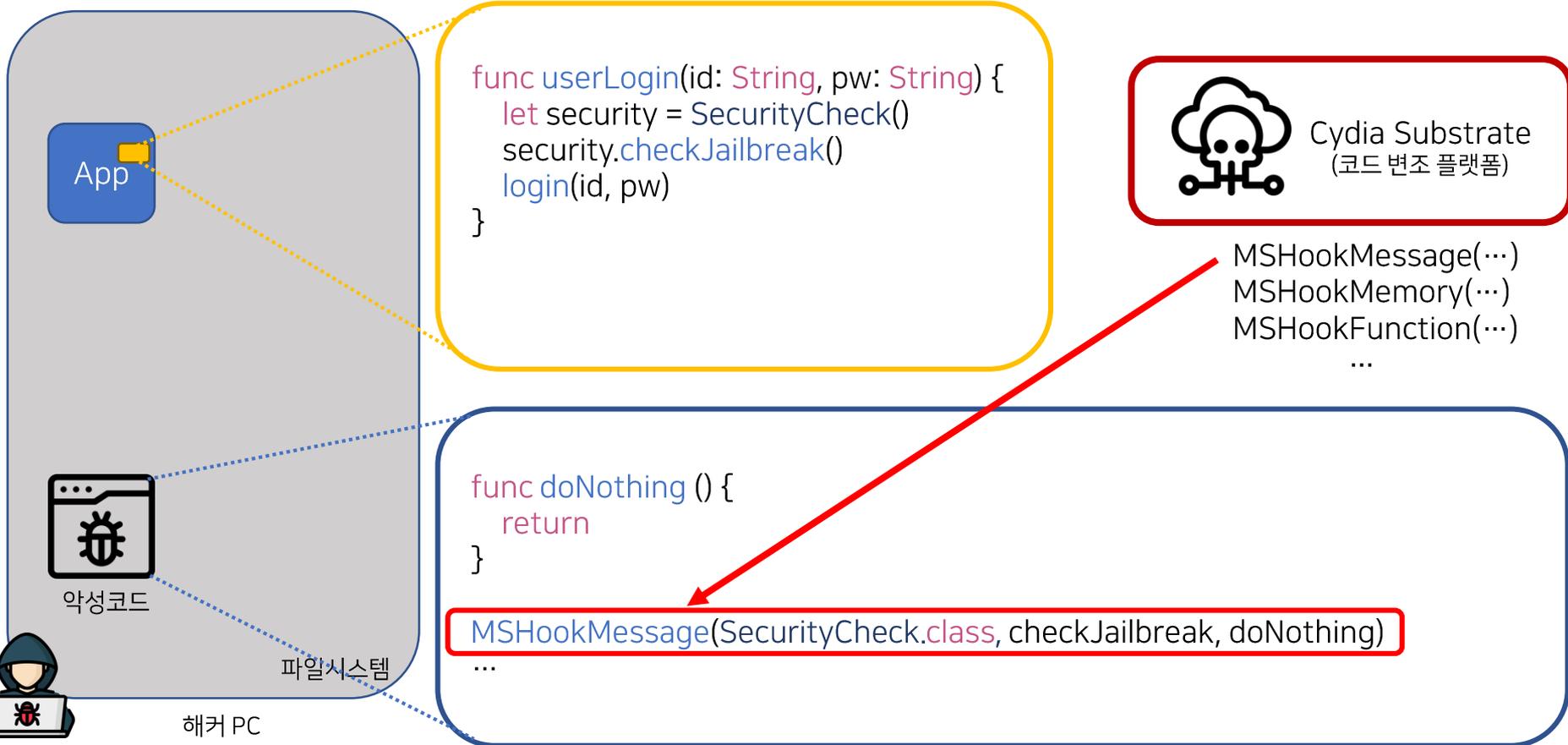
Cydia Substrate  
(코드 변조 플랫폼)

MShookMessage(...)  
MShookMemory(...)  
MShookFunction(...)

...

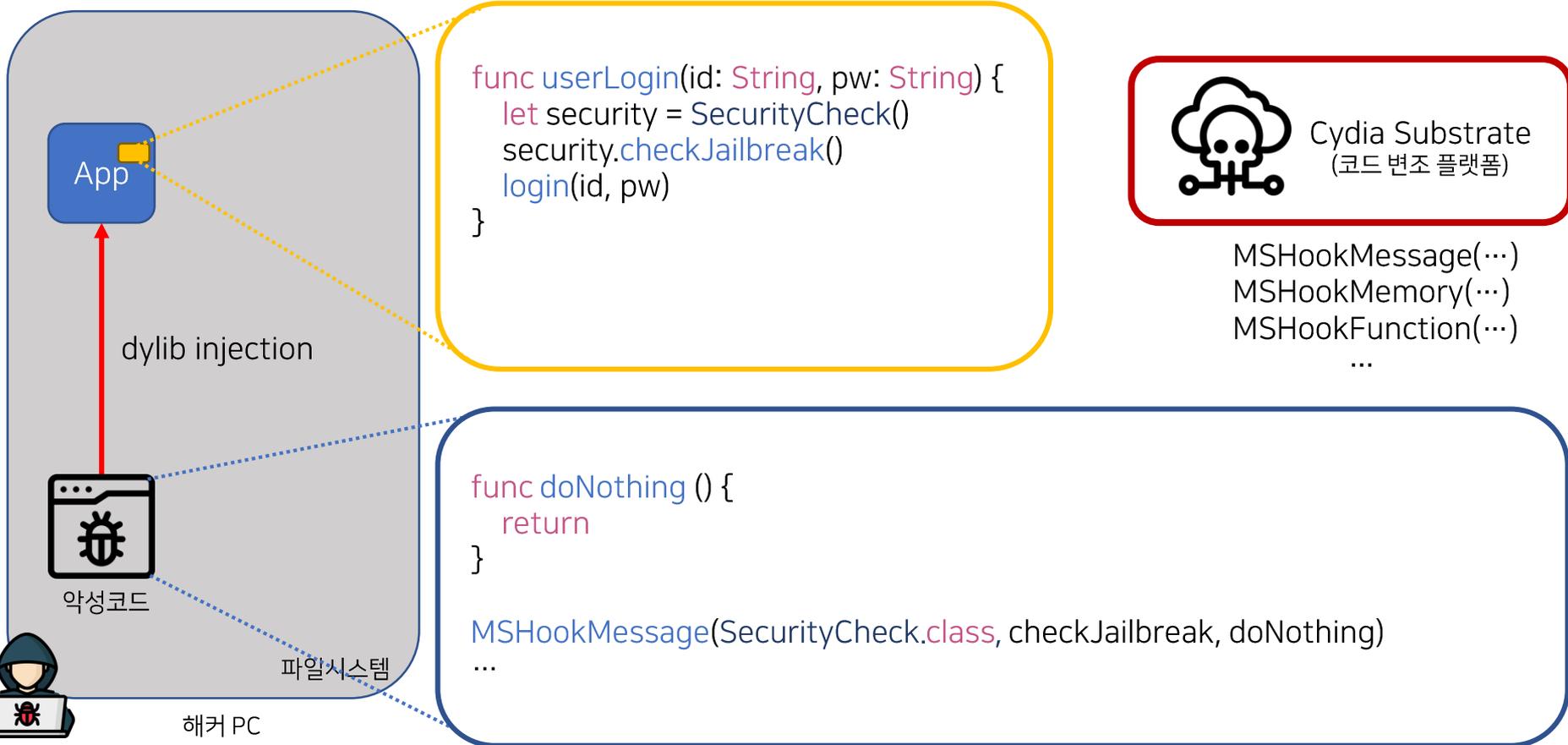
# iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 2. 메소드 스위즐링 수행



# iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 3. dylib injection



MSHookMessage(...)  
MSHookMemory(...)  
MSHookFunction(...)  
...

# iOS 모바일 해킹 유형: 앱 위변조

- 보안모듈 우회 방안 분석: 4. 보안모듈 우회 완료



해커 PC

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

```
func doNothing () {
    return
}

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



MSHookMessage(...)  
 MSHookMemory(...)  
 MSHookFunction(...)  
 ...

# iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 1. 심볼 확인



App

보안모듈



보안모듈 우회 약성코드

파일시스템

```

// 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 ABCBBCCAACCBACDADDDAACCC = True) { }

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

유니티 심볼 분석 프로그램:  
il2CppTypeDumper

‘해당 로직이 정상적으로 실행되지 않도록 하면 데미지를 받지 않을 수도 있겠구나’



해커 PC



# iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 1. 심볼 확인



보안모듈

🔒

보안모듈 우회 약성코드

파일시스템



해커 PC

```

// 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 ABCBBCCAACCBACDADDDAACCC = True) { }

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

유니티 심볼 분석 프로그램:  
il2CppDumper

```

2 void __cdecl GetDamage(Object_o *this, bool ABCBBCCAACCBACDADDDAACCC, const MethodInfo *method)
3 {
4     _BOOL4 v3; // w20
5     __int64 v5; // x21
6     float v6; // s8
7     float v7; // s9
8     float v8; // s8
9     bool v9; // zf
10    __int128 v10; // q0
11    __int128 v11; // q1
12    float v12; // s0
13    float v13; // s2
14    ADDDAABADCCCBCDBCCDAC_c *v14; // x0
15    struct ADDDAABADCCCBCDBCCDAC_StaticFields *v15; // x8
16    __int128 v16; // q0
17    __int128 v17; // q1
```

정적분석 프로그램: IDA pro

# iOS 모바일 해킹 유형: 앱 위변조

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



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	원본 어셈블리 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



해커 PC

```

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
    
```

원본 어셈블리

```

62 CA 00 BD FD 7B 41 A9 F4 4F C2 A8 C0 03 5F D6
FF 03 03 D1 E9 23 08 6D F6 57 09 A9 F4 4E 0A A9
FD 7B 0B A9 FD C3 02 91 1F 20 03 D5 C0 03 5F D6
B5 32 01 F0 A8 AE 4B 39 E8 00 00 37 E8 C6 00 90
1F 20 03 D5 00 05 4A B9 5C 0E F2 97 28 00 80 52
A8 AE 0B 39 68 32 40 BD 69 9E 40 BD 68 3E 01 D0
    
```

바이너리 패치 수행

```

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      NOP
0000000001794B1C      RET
    
```

바이너리 패치 결과

# iOS 모바일 해킹 유형: 앱 위변조

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



App

- 보안모듈
- 보안모듈 우회 약성코드
- 코드패치를 통한 변조

파일시스템

```

2 void __cdecl GetDamage(Object_o *this, bool ABCBBCCAACCBACDADDDAACCC, const MethodInfo *method)
3 {
4     _BOOL4 v3; // w20
5     __int64 v5; // x21
6     float v6; // s8
7     float v7; // s9
8     float v8; // s8
9     bool v9; // zf
10    __int128 v10; // q0
11    __int128 v11; // q1
12    float v12; // s0
13    float v13; // s2
14    ADDDAABADCCCBCDBCCDAC_c *v14; // x0
15    struct ADDDAABADCCCBCDBCCDAC_StaticFields *v15; // x8
16    __int128 v16; // q0
17    __int128 v17; // q1
    
```

↓

 바이너리 코드패치를 통한 변조 결과

```

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



해커 PC

# iOS 모바일 해킹 유형: 앱 위변조

- 변조를 통해 이득을 얻을 수 있는 부분을 분석: 3. 변조 완료



App

-  보안모듈
-  보안모듈 우회 약성코드
-  코드패치를 통한 변조

파일시스템

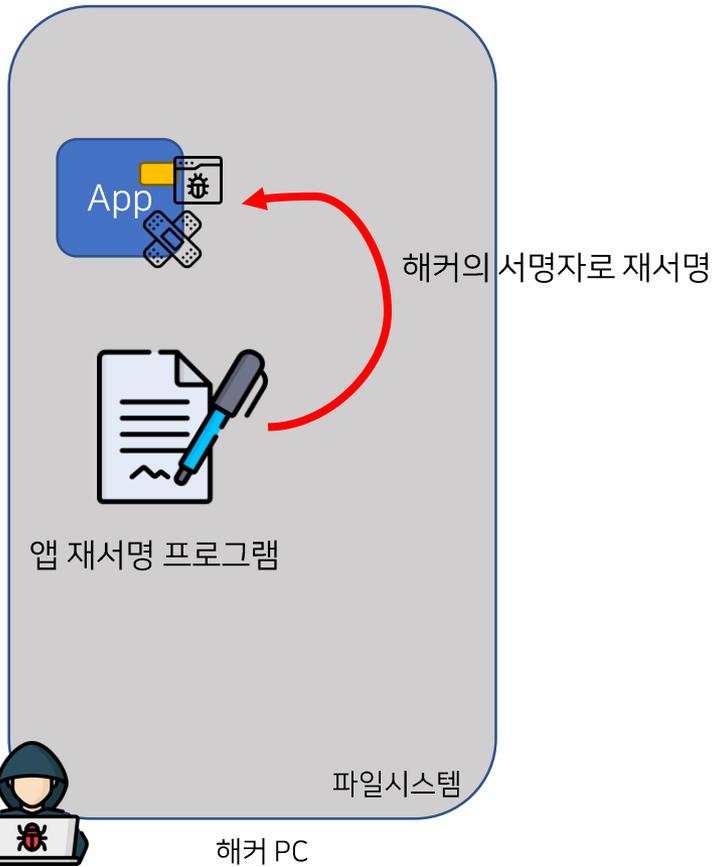


해커 PC

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

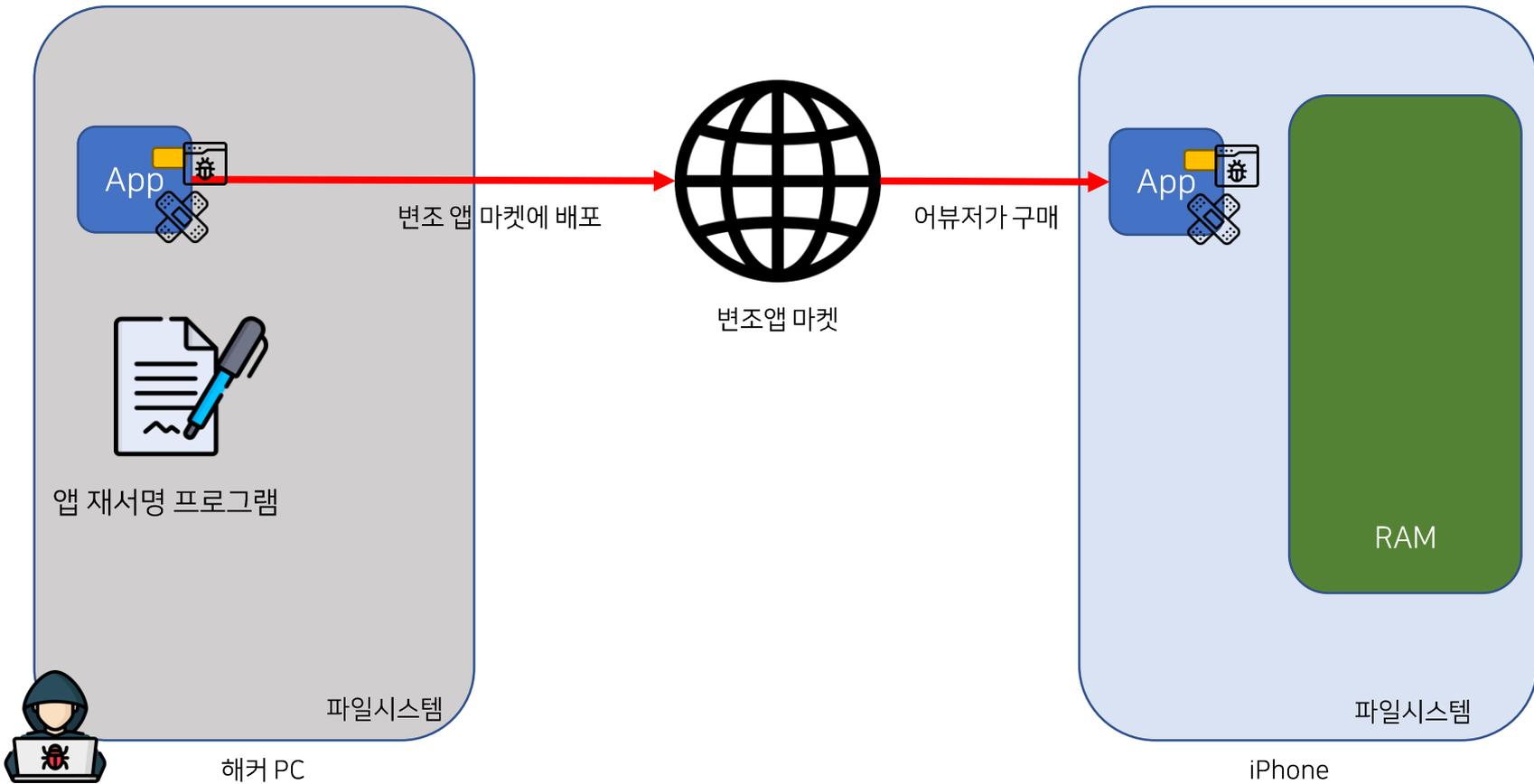
# iOS 모바일 해킹 유형: 앱 위변조

- 해커의 서명자로 재서명

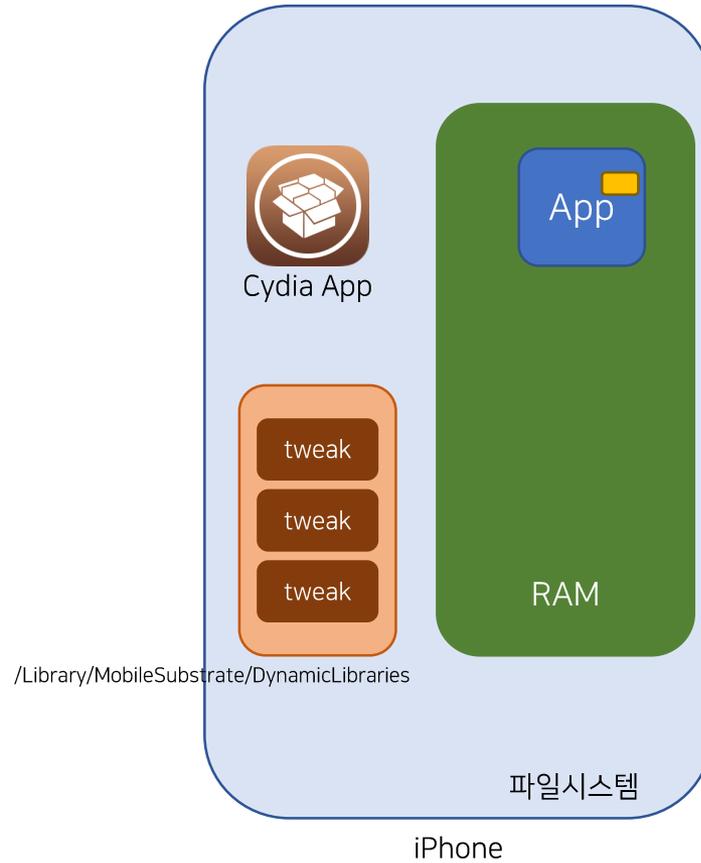


# iOS 모바일 해킹 유형: 앱 위변조

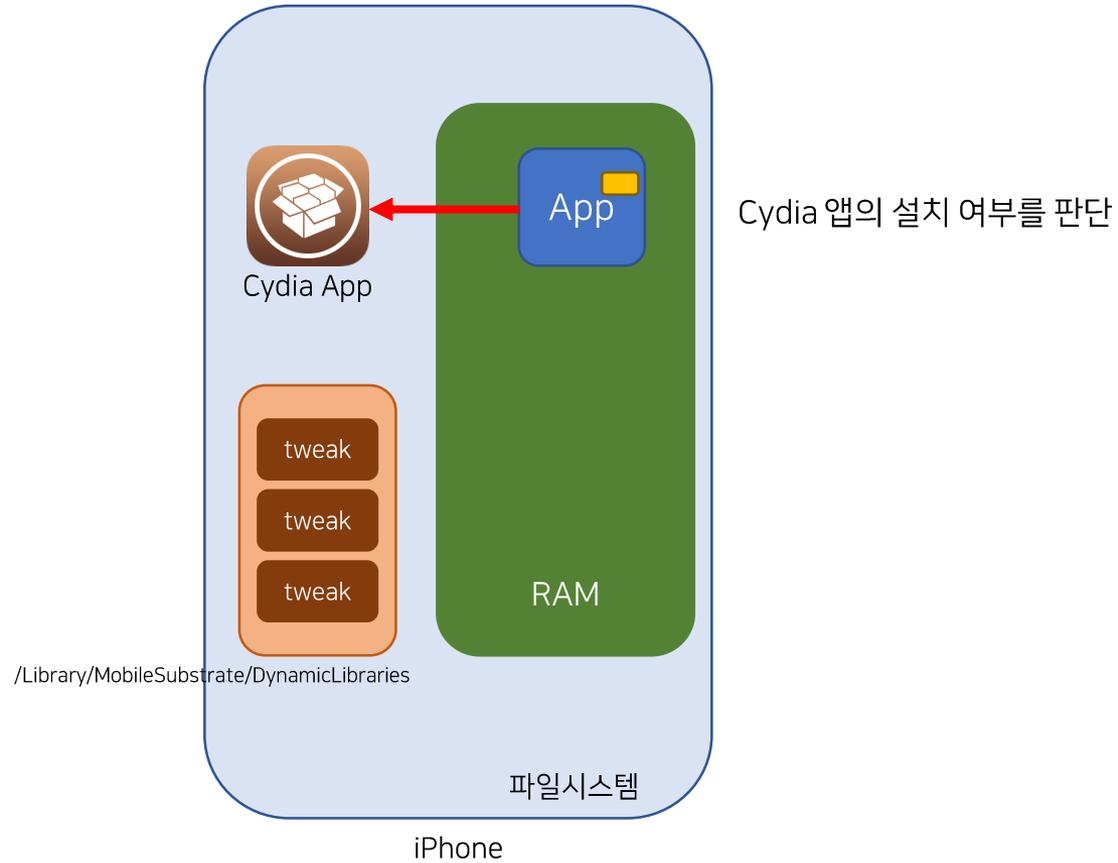
- 변조앱 배포



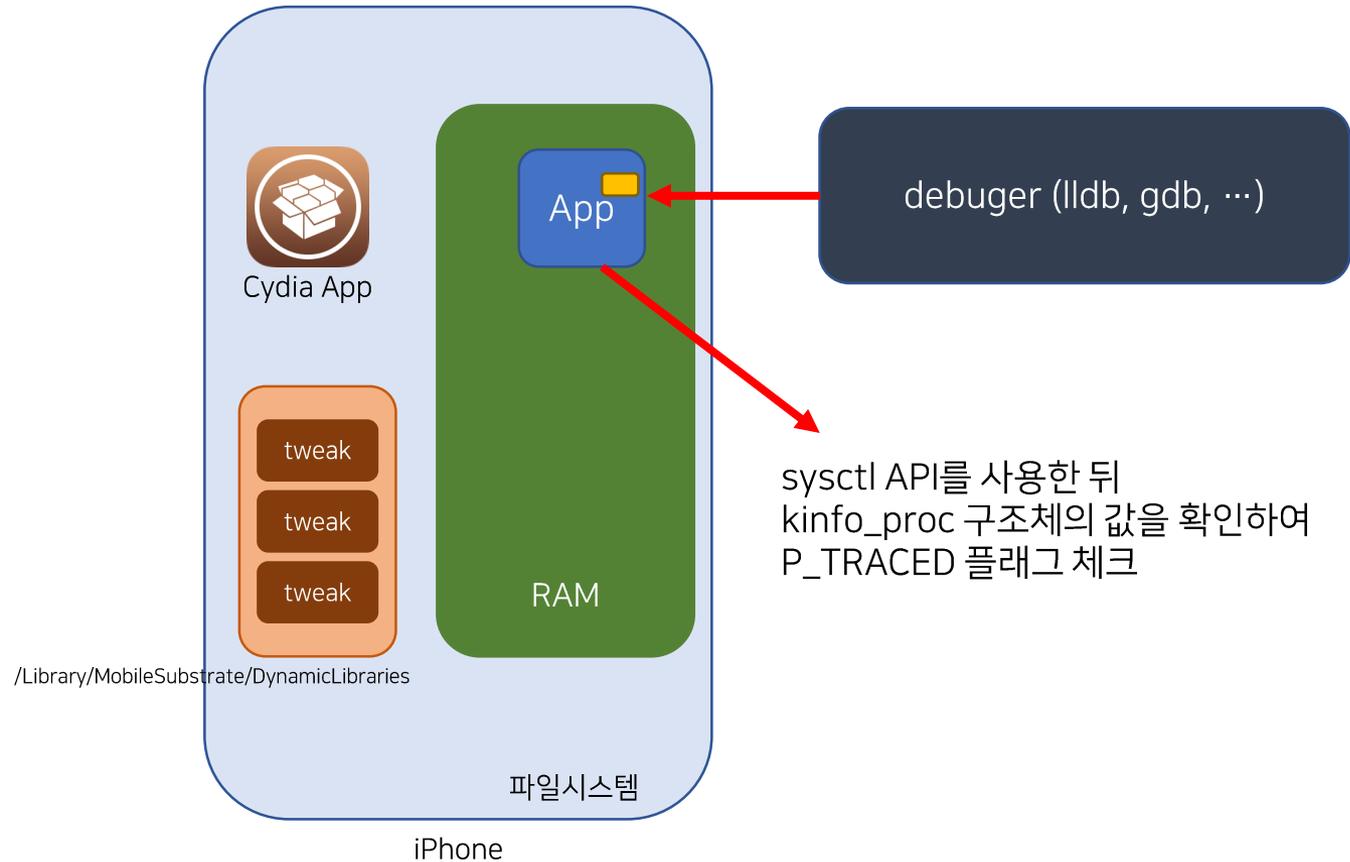
# iOS 모바일 해킹 대응: 탈옥 탐지



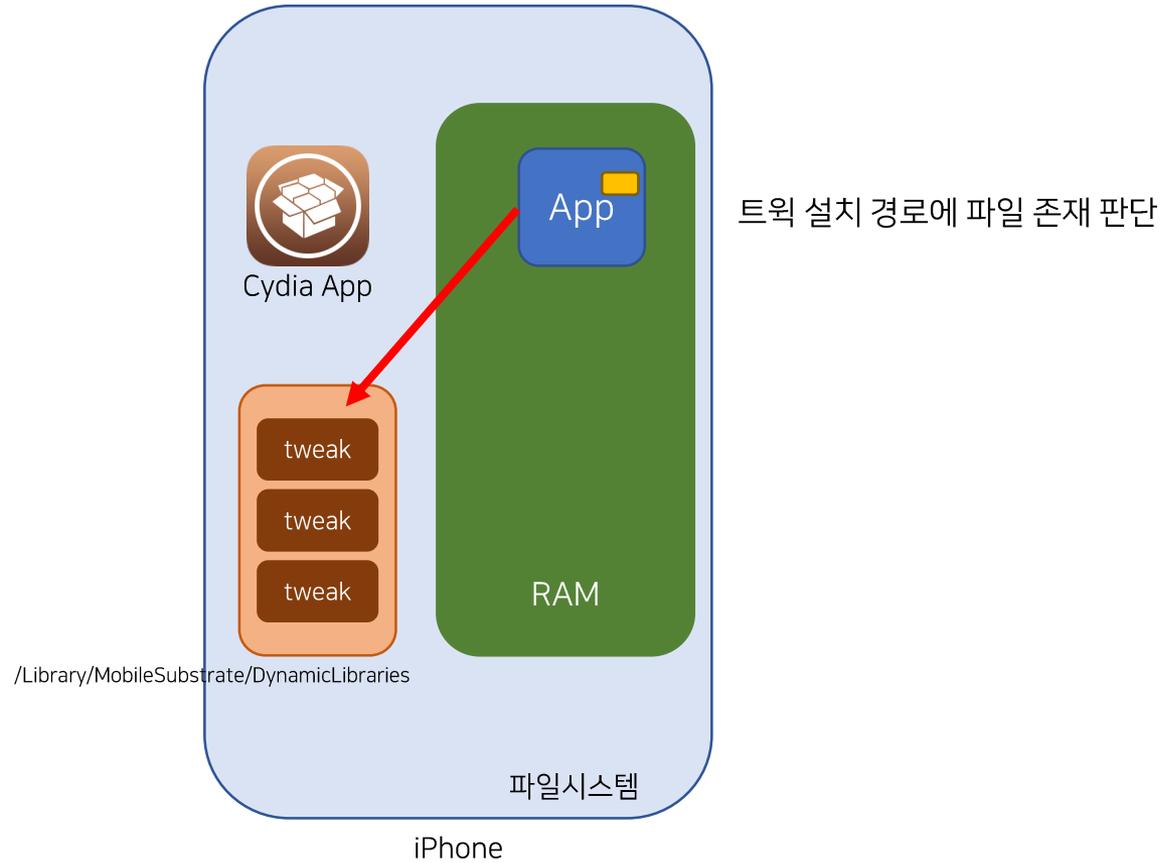
# iOS 모바일 해킹 대응: 탈옥 탐지



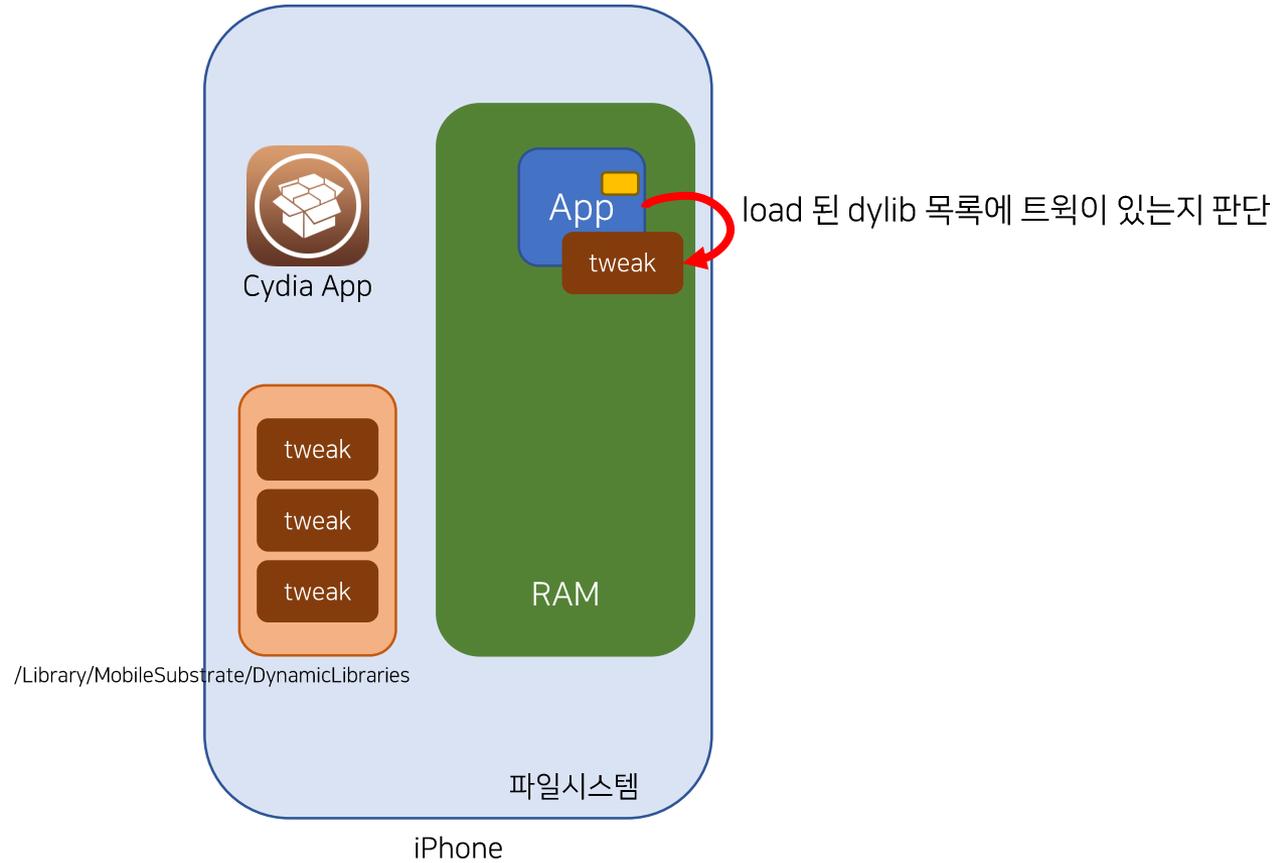
# iOS 모바일 해킹 대응: 디버깅 탐지



# iOS 모바일 해킹 대응: tweak 탐지



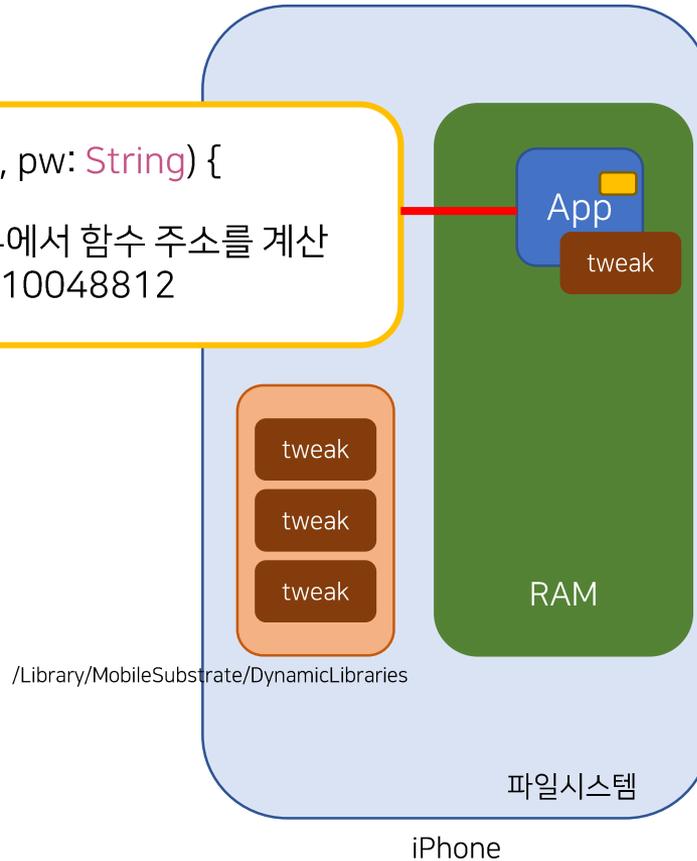
# iOS 모바일 해킹 대응: tweak 탐지



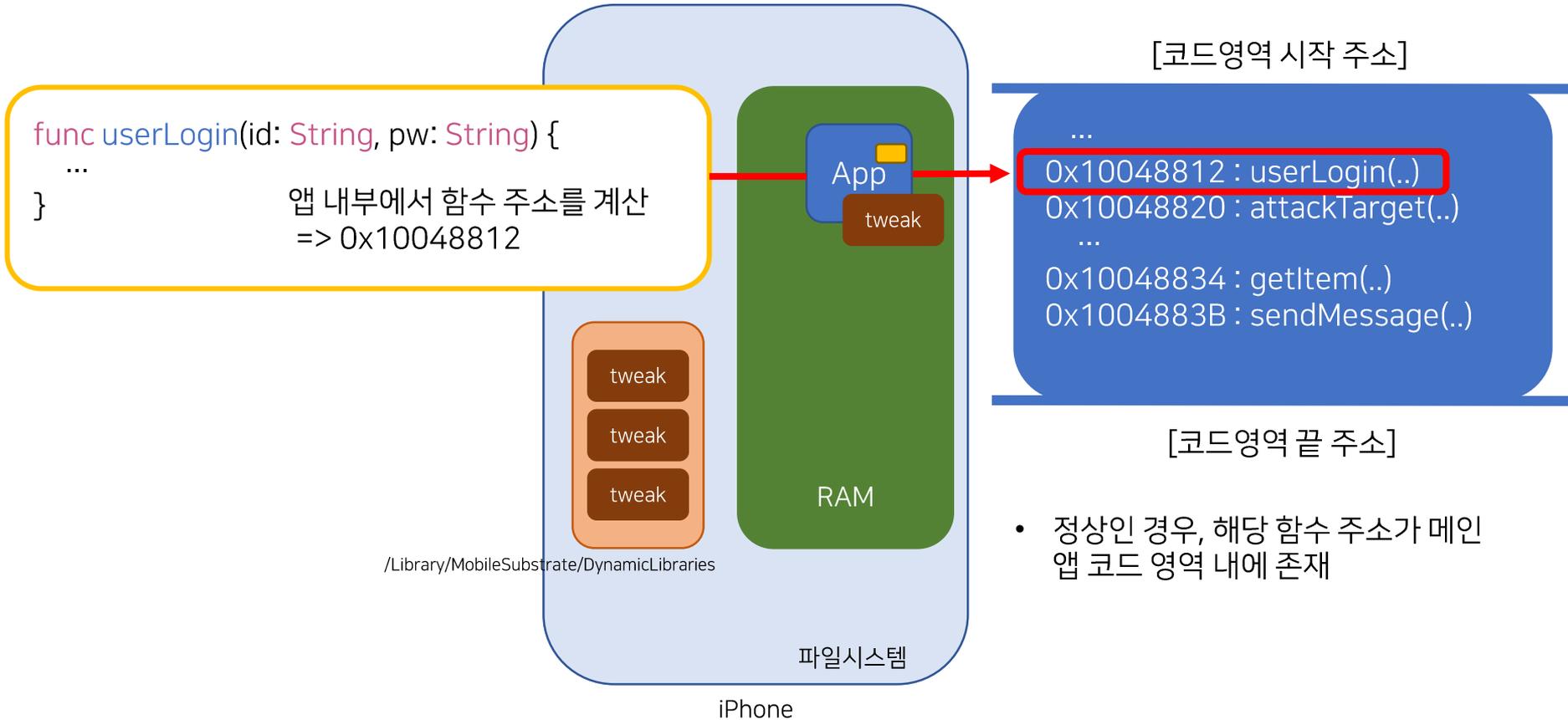
# iOS 모바일 해킹 대응: 메소드 스위즐링 탐지

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

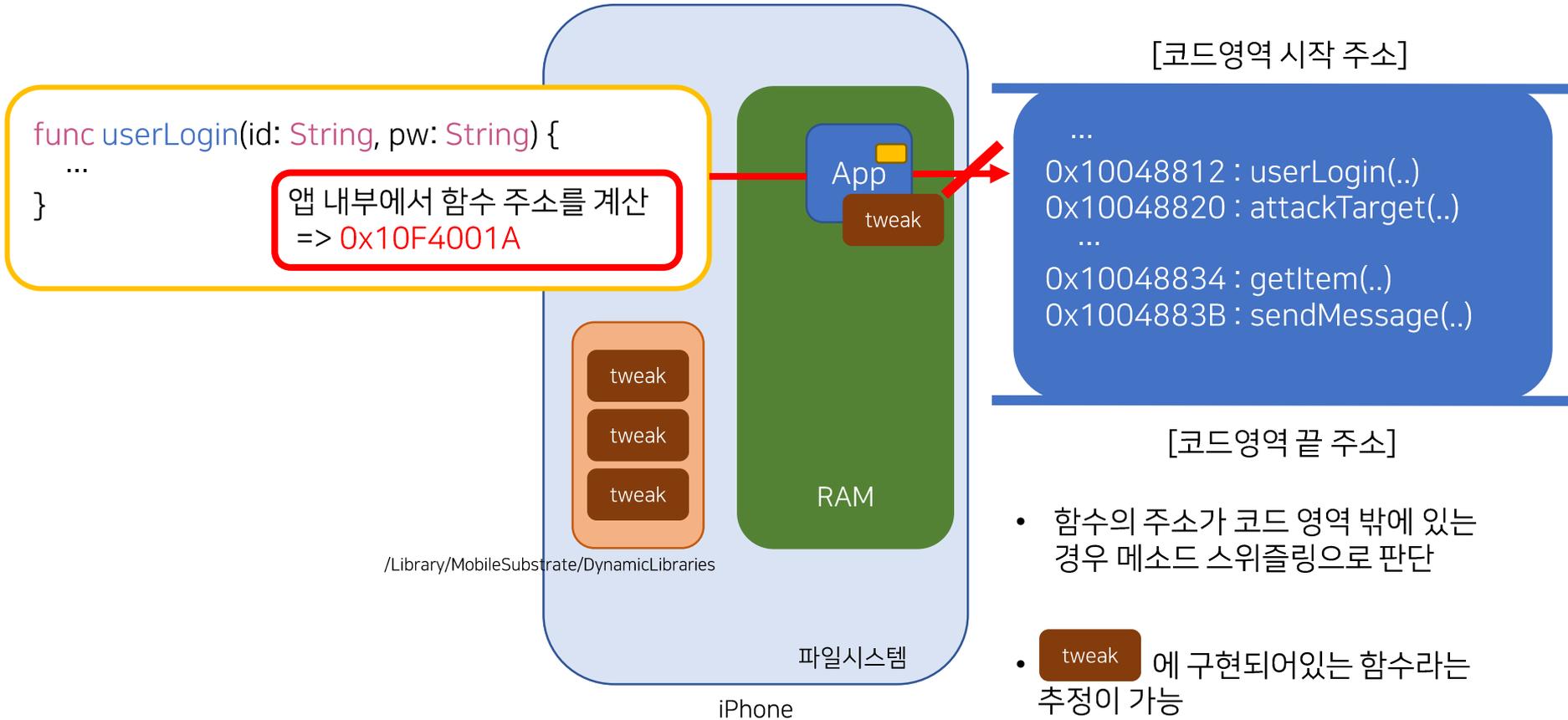
앱 내부에서 함수 주소를 계산  
=> 0x10048812



# iOS 모바일 해킹 대응: 메소드 스위즐링 탐지

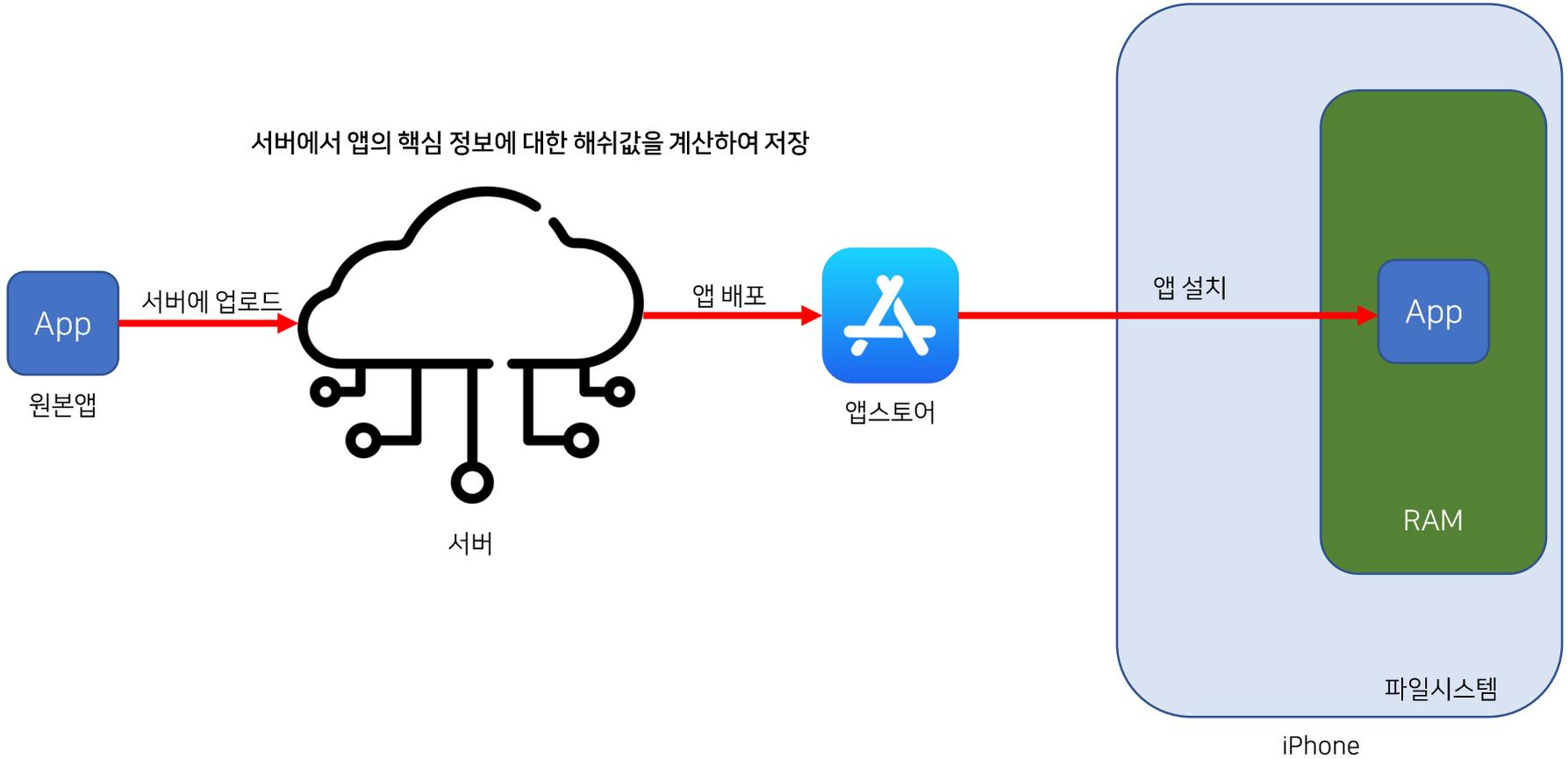


# iOS 모바일 해킹 대응: 메소드 스위즐링 탐지

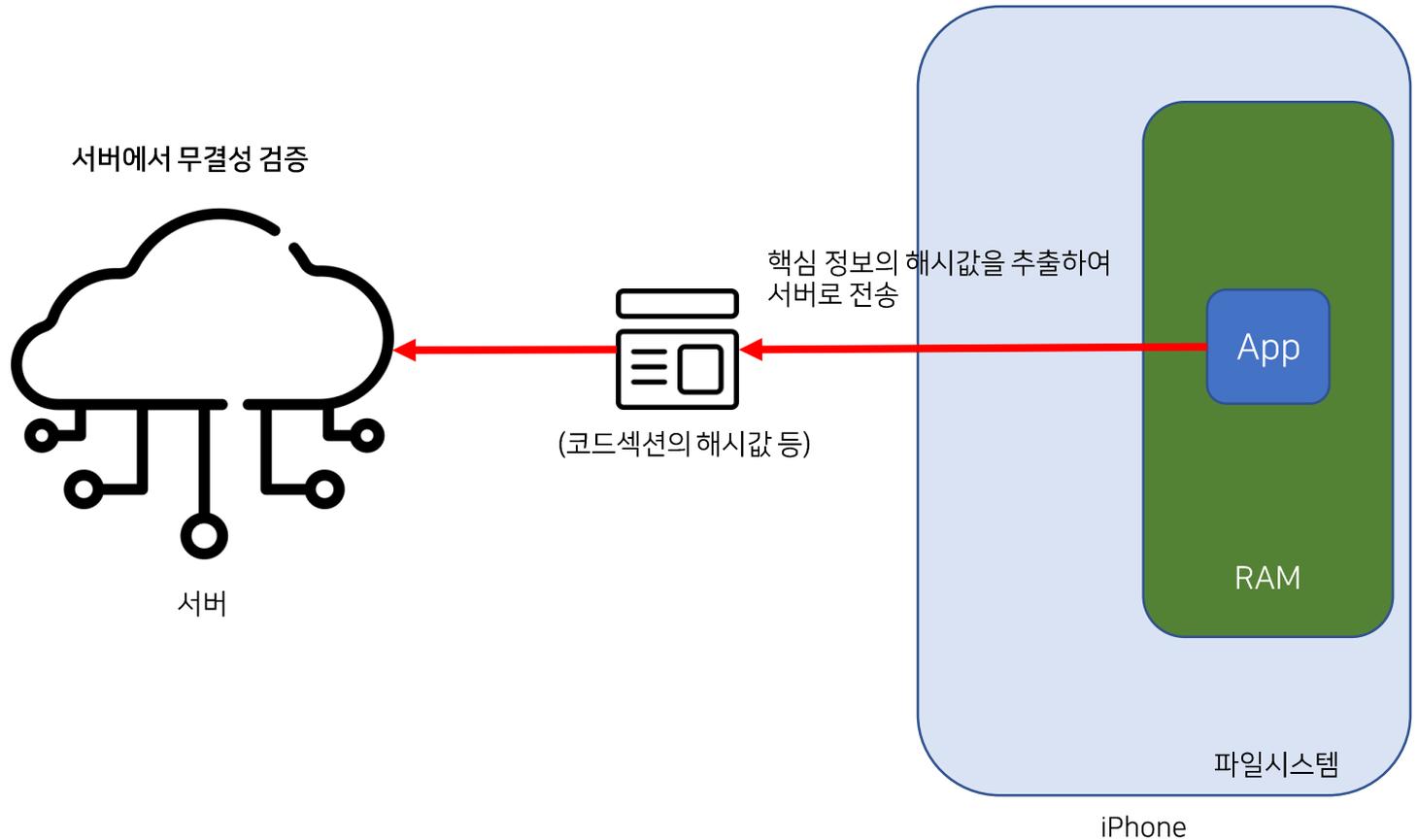


# iOS 모바일 해킹 대응: 앱 위변조 탐지

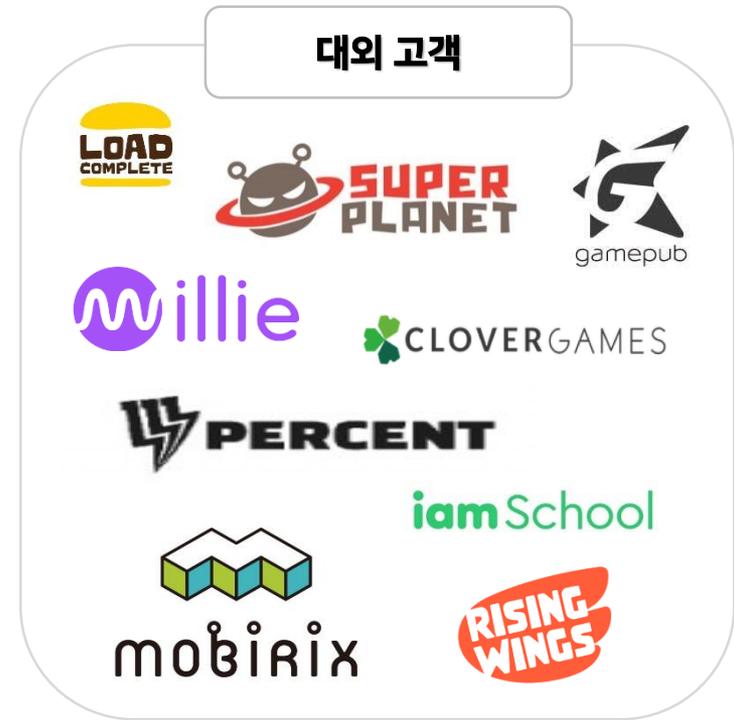
서버에서 앱의 핵심 정보에 대한 해쉬값을 계산하여 저장



# iOS 모바일 해킹 대응: 앱 위변조 탐지



# NHN AppGuard 도입사례

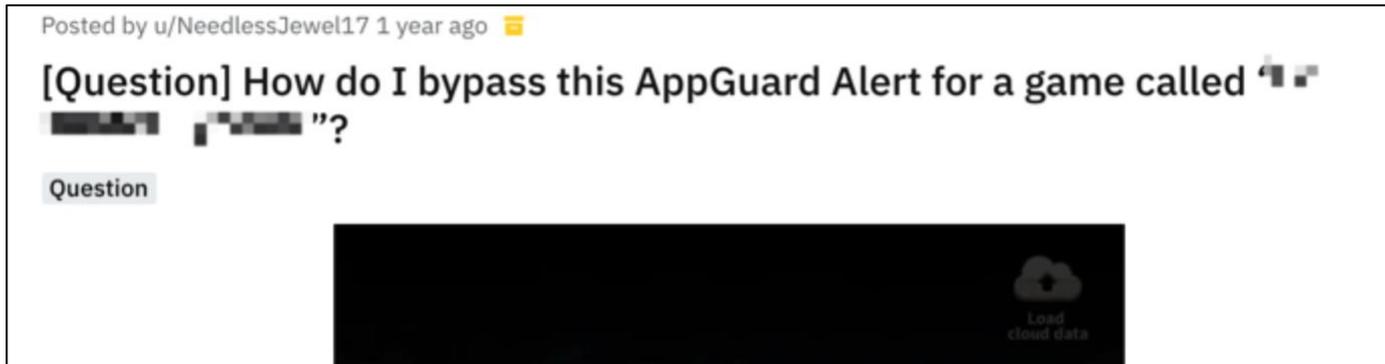
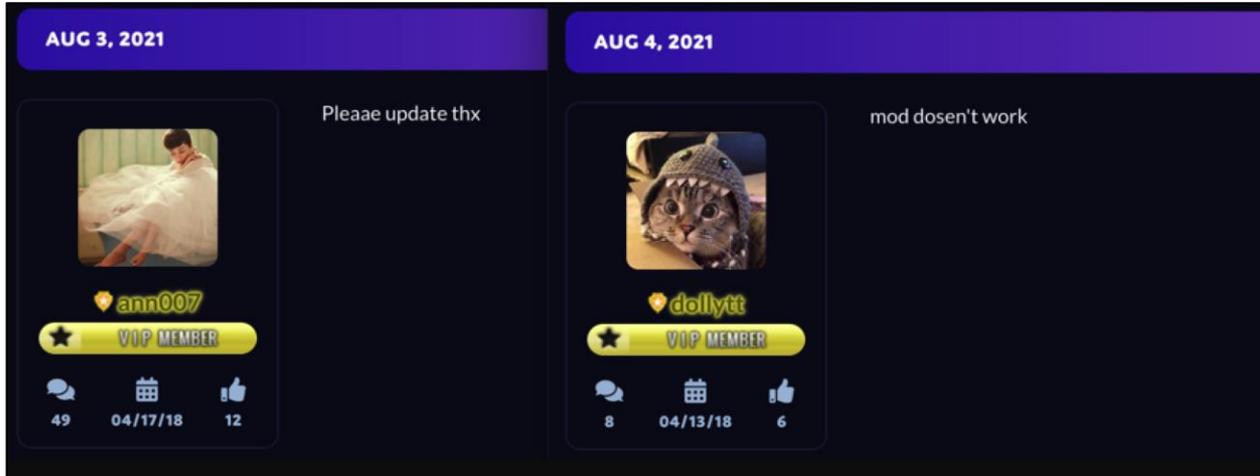


# NHN AppGuard 도입사례

The screenshot displays a social media interface with two posts. The first post, dated AUG 3, 2021, is from user **ann007** (VIP MEMBER) and contains the text "Pleaee update thx". The second post, dated AUG 4, 2021, is from user **dollytt** (VIP MEMBER) and contains the text "mod dosen't work". Both posts feature profile pictures and engagement metrics (comments, dates, likes) at the bottom.

Date	User	Text	Comments	Date	Likes
AUG 3, 2021	ann007	Pleaee update thx	49	04/17/18	12
AUG 4, 2021	dollytt	mod dosen't work	8	04/13/18	6

# NHN AppGuard 도입사례



# NHN AppGuard 도입사례

**AUG 3, 2021**

Pleaae update thx



ann007  
VIP MEMBER

49 04/17/18 12

**AUG 4, 2021**

mod dosen't work



dollytt  
VIP MEMBER

8 04/13/18 6

Posted by u/NeedlessJewel17 1 year ago

**[Question] How do I bypass this AppGuard Alert for a game**  
 [REDACTED] [REDACTED]”?

Question



**[OUTDATED!] FREE HACK - (REDACTED) 5 Free 32 Bit Script**  
 조회수 2,809회 2020. 11. 10.

**Memo Modz Official**  
 구독자 986명  
 -Android/iOS?  
 -Android

더보기

댓글 93개 정렬 기준

공개 댓글 추가...

**Memo Modz Official**님이 고정함  
**Memo Modz Official** 1주 전  
 Plz Watch The Whole Video To Do The Hack 😊  
 6 👍 1 🗨️ 1 🇺🇸 답글  
 ↓ 답글 3개 보기

**Eymen Akif** 15시간 전  
 Abi bende appguard alet security dio ve kapanio çözüm?  
 1 👍 1 🗨️ 1 🇺🇸 답글

**Milad Raza** 19시간 전  
 Now it is not working game not opening in virtual

# NHN AppGuard 도입사례

대시보드   앱 보호   정책   블랙리스트

Android   iOS

종류	정책
치팅툴	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
에뮬레이터	해제 <b>탐지</b> 조건 차단 (1)   ▼   전체 차단
루팅	해제 <b>탐지</b> 조건 차단 (1)   ▼   전체 차단
변조	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
디버거	해제 <b>탐지</b> 조건 차단 (1)   ▼   전체 차단
스피드조작	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
SSL Pinning	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
가상환경	
원격제어	
매크로틀	

Android   iOS

종류	정책
치팅툴	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
에뮬레이터	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
변조	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
디버거	해제 <b>탐지</b> 조건 차단 (1)   ▼   전체 차단
탈옥	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단
후킹	해제 <b>탐지</b> 조건 차단 (0)   ▼   전체 차단

# NHN AppGuard 도입사례

NHN AppGuard > 블랙리스트
URL & Appkey   사용자 가이드       **퀵 가이드**

대시보드   앱 보호   정책   블랙리스트

**블랙리스트 조회 및 등록** | 유저 ID, 디바이스 ID를 등록하고 조회할 수 있습니다.

조회 기준: 최근순 ▼

상태: 전체 ▼

차단 기준: 전체 ▼

블랙리스트:

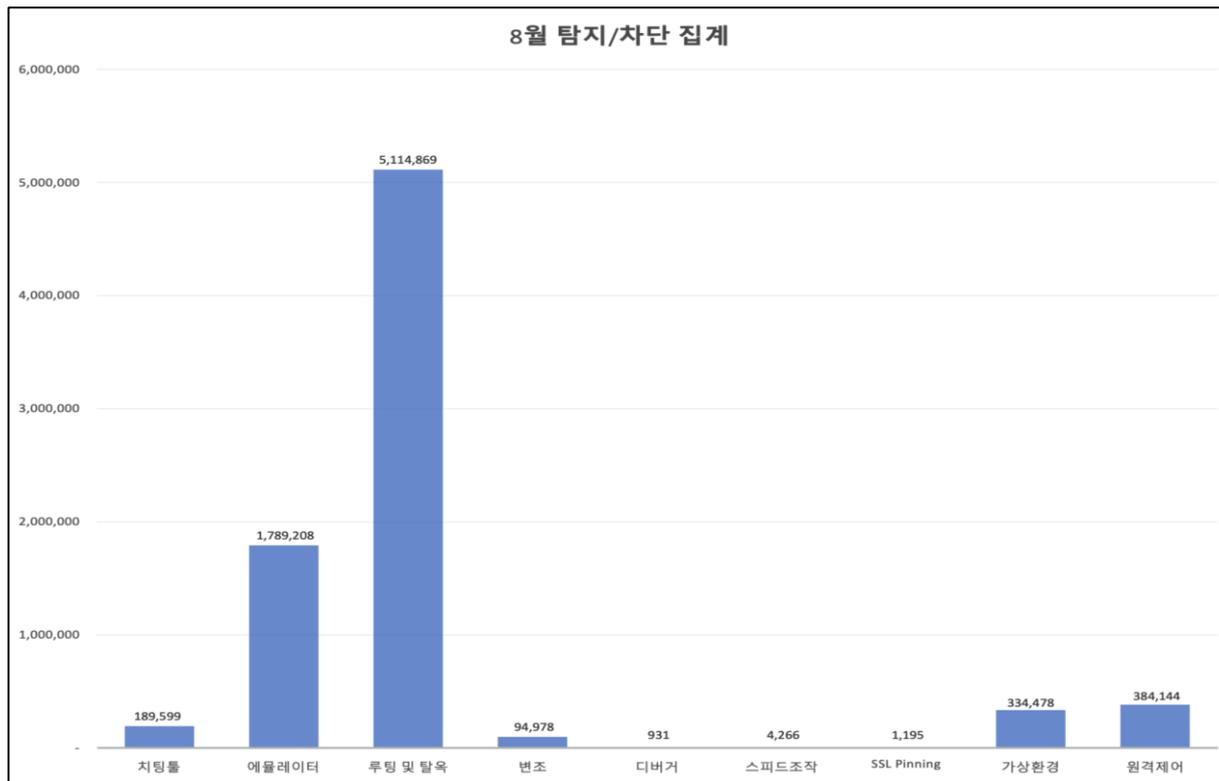
조회
등록

**상세 내용**

블랙리스트	차단 기준	상태	차단 사유	시작 일자	만료 일자	관리
dl_appguard_batch@nhn.com	유저 ID	해제	TEST	2022-09-27 10:14:10	2022-09-27 10:14:16	<span style="background-color: #666; color: white; padding: 2px 5px; border-radius: 3px;">재등록</span>
r3d5h4rk	유저 ID	해제	test	2021-07-27 08:30:20	2021-07-27 08:36:26	<span style="background-color: #666; color: white; padding: 2px 5px; border-radius: 3px;">재등록</span>
fweF	유저 ID	해제	aewf	2021-07-27 08:30:18	2021-07-30 00:00:00	<span style="background-color: #666; color: white; padding: 2px 5px; border-radius: 3px;">재등록</span>

<< < 1 > >>

# 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://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](https://www.facebook.com/SuperSUofficial?ref=br_rs)
- [http://www.hungryapp.co.kr/bbs/bbs\\_view.php?durl=YmNvZGU9ZWxnYXJkbnBpZD0xMDIzMDM0JnVzZXI9](http://www.hungryapp.co.kr/bbs/bbs_view.php?durl=YmNvZGU9ZWxnYXJkbnBpZD0xMDIzMDM0JnVzZXI9)
- <https://ko.101-help.com/ef3cd1e2e0-android-phoneseseo-geulgogleul-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/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://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/ll2CppDumper>
- [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/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/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/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://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/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/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/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/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/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/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](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/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/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/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](https://www.flaticon.com/free-icon/search_7079548?related_id=7079548&origin=search)

Cloud

**유연하게, 안전하게  
비즈니스에 힘이 되다.**

