오픈소스로 여는 뉴노멀

2020 공개SW 페스티벌

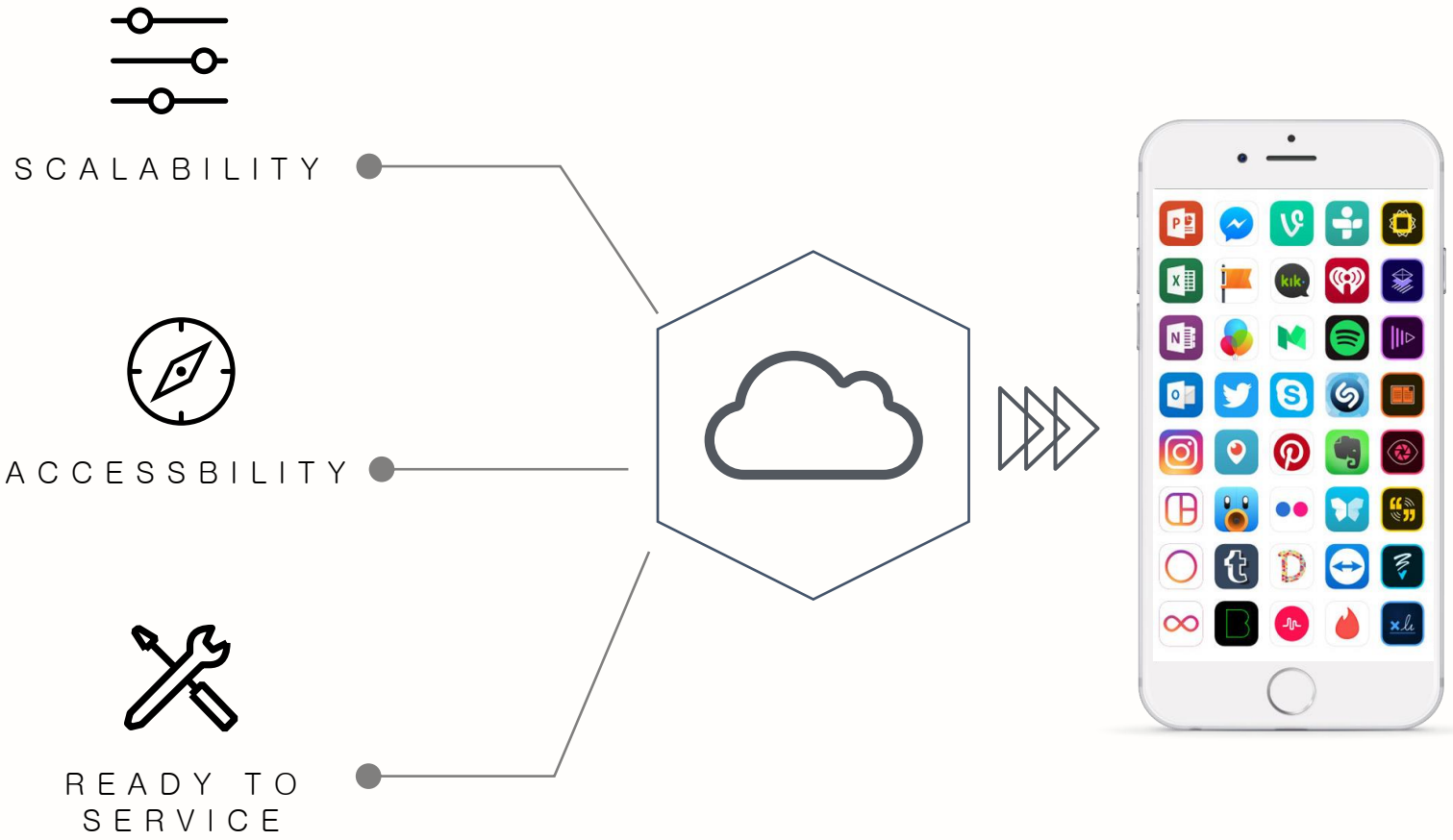
과학기술정보통신부
Ministry of Science and ICT

nipa 정보통신산업진흥원
National IT Industry Promotion Agency

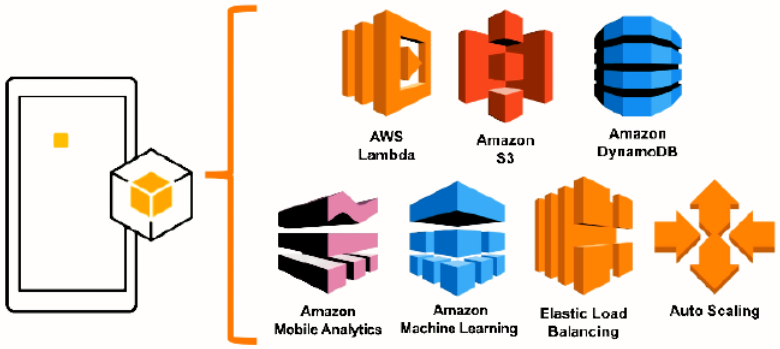
모바일 앱의 클라우드 보안 취약점과 대응방안

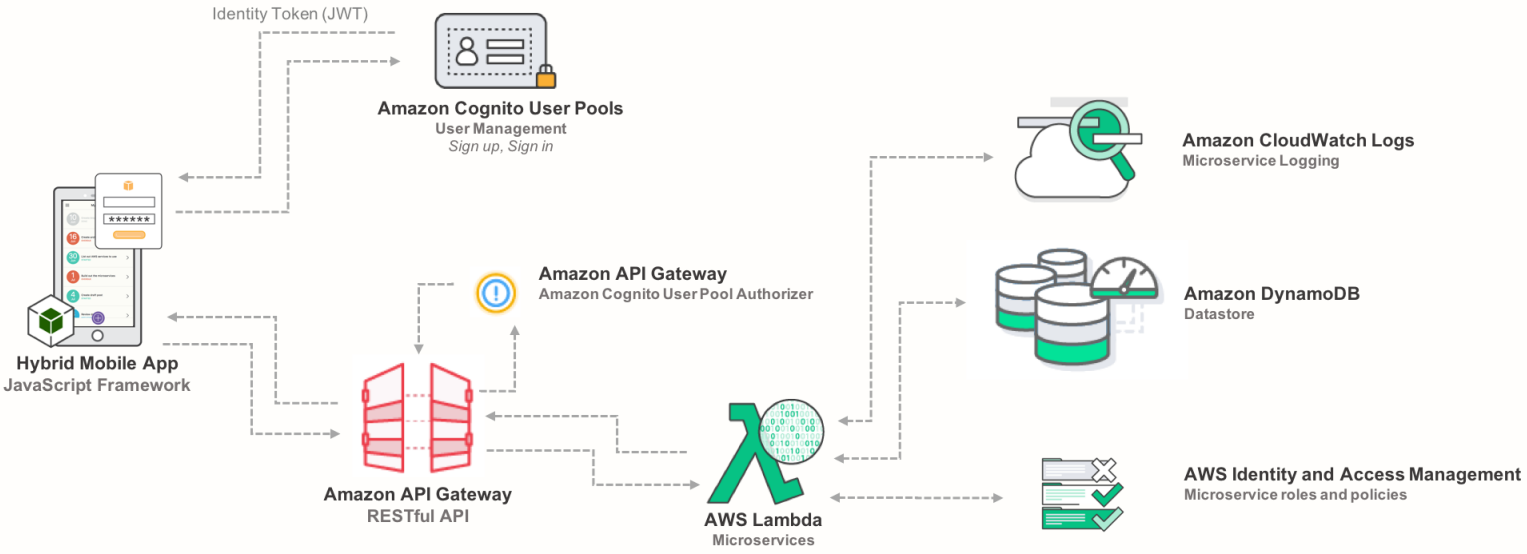
네이버
박현준

모바일 앱 개발 환경의 변화



클라우드 모바일 백엔드 활용 예시





클라우드 모바일 백엔드 활용 예시

과거 클라우드 데이터 유출 사례

UBER

Uber
(2016)

AWS Account Hacked
Personal Information
of 57 Million users

Time Warner Cable

TimeWarner
Cable
(2017)

4 Million Customer
Records Exposed

FedEx

FedEx
(2018)

119,000 US Citizens'
Personal Information
Leaked

Is it OK
now?

여전히 계속되는 클라우드 데이터 유출 사고


Ransomware

AWS Cloud Storage Leak Affects 10,000 Users

Learn how the data of U.K. train commuters using the free Wi-Fi in Network Rail-managed stations was accidentally leaked. Also, over 200 million records containing property-related information on U.S. residents were exposed.

By: Jon Clay
March 06, 2020


Unsecured AWS S3 Bucket Found Leaking Data of Over 30K Cannabis Dispensary Customers

January 27, 2020


Exposed AWS buckets again implicated in multiple data leaks

A series of data leaks in the past week have once again implicated poorly secured Amazon S3 buckets, which are supposed to be private by default

클라우드 보안의 3가지 실패 원인

Key Management Failure

루트키 사용, 키 값의 하드코딩 실수

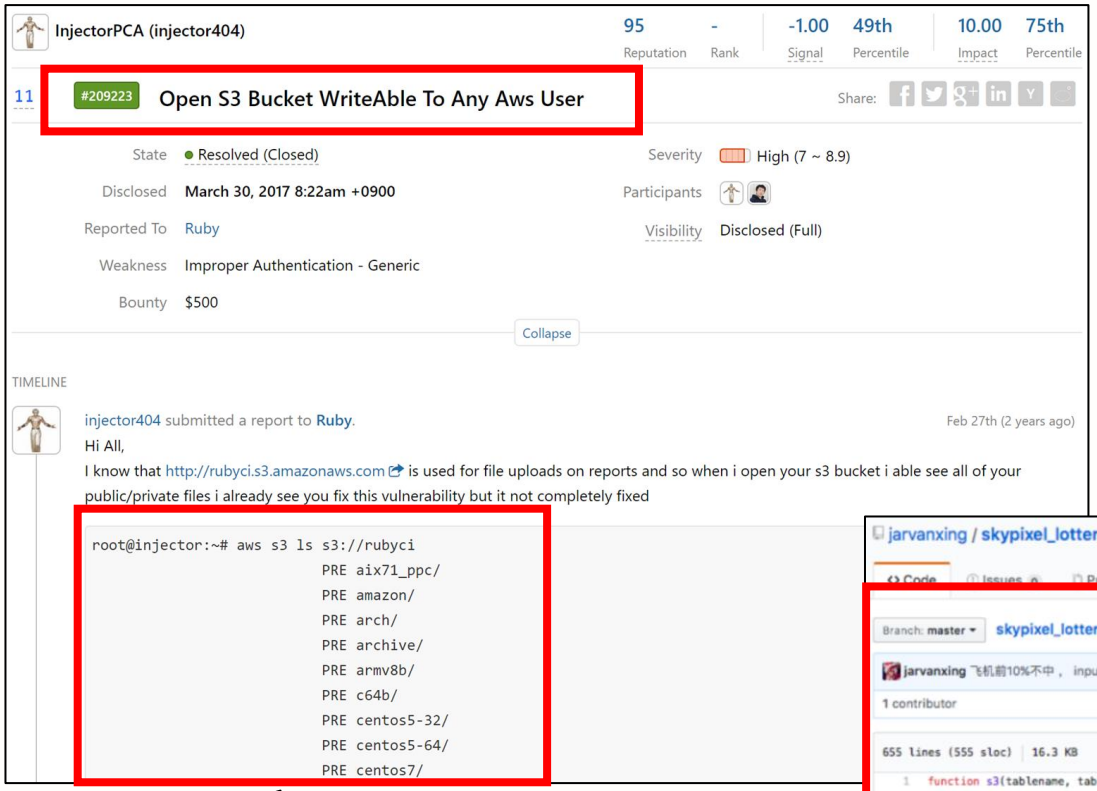
Cloud Permission Failure

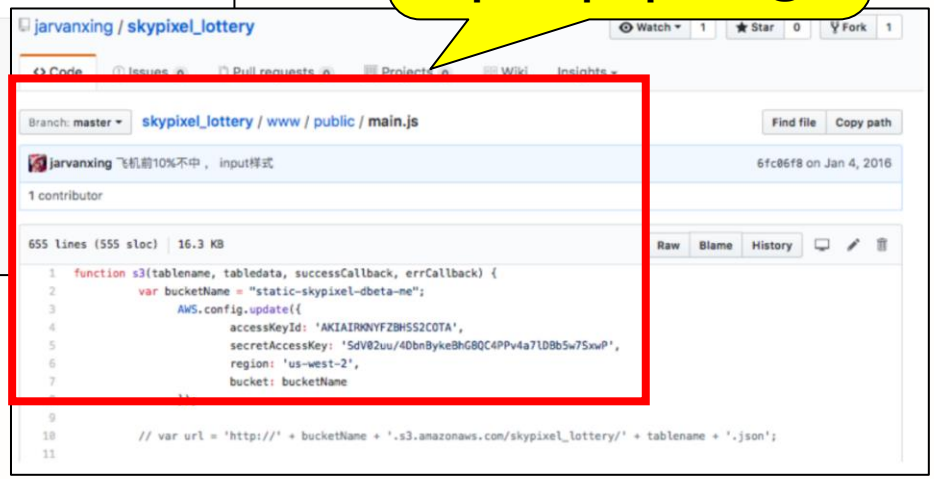
앱 유저간의 권한 분리 실패

Monitoring Failure

비용 절감을 위한 로깅/모니터링 미적용

클라우드 취약점 사례

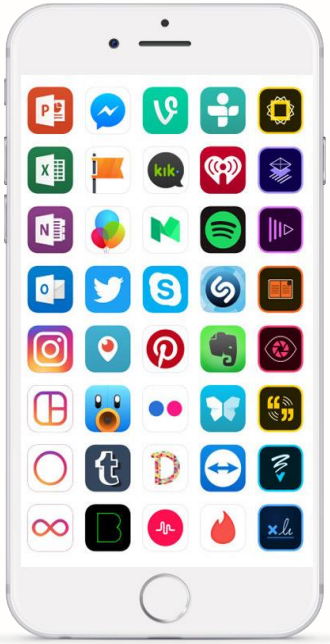


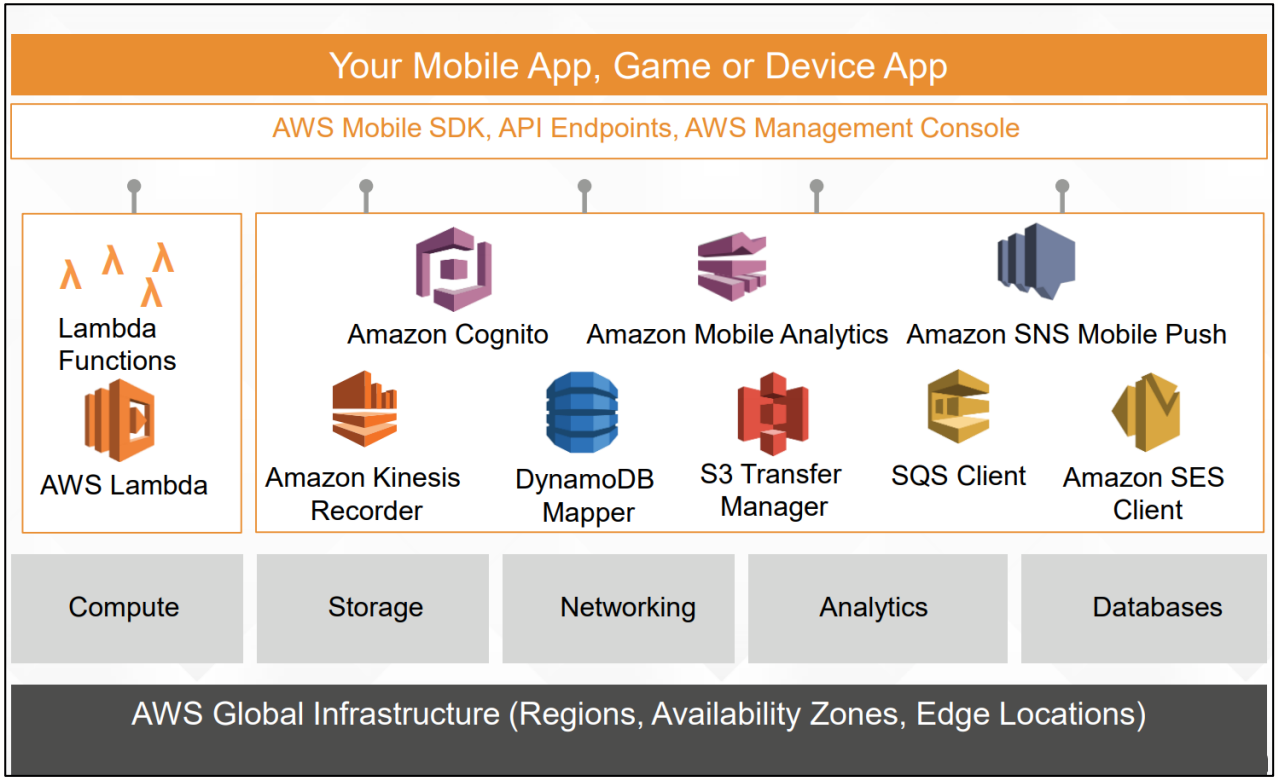


사진공유 앱의 클라우드 저장소 루트키 하드코딩

클라우드 저장소의 잘못된 권한 부여(읽기/쓰기 가능)

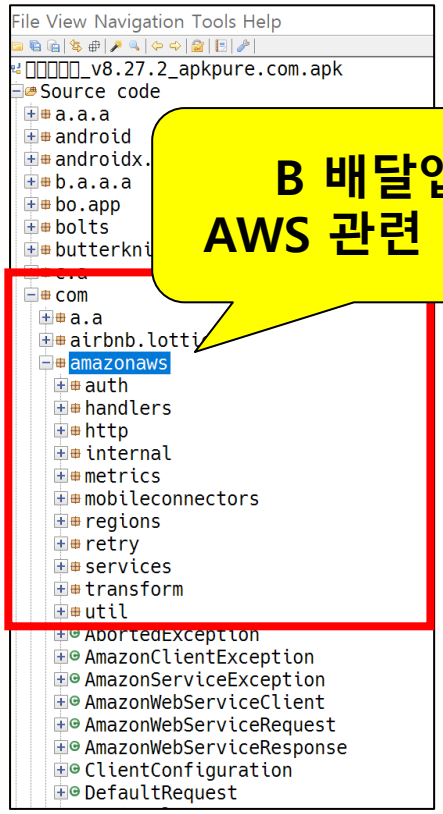
모바일 앱과 클라우드 Back-End 아키텍쳐 구성





- 모바일 앱에 내장된 클라우드 SDK를 통해 서비스 호출 (REST API)
- 클라우드 사용자 인증을 위한 Credential (Access/Secret Key) 필요

▶ 클라우드 SDK를 사용하는 모바일 앱의 취약점 분석



- 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**

클라우드 SDK를 사용하는 모바일 앱의 취약점 분석

```
private /* synthetic */ void a(String str, S3UploadInfo s3UploadInfo, final ac acVar) throws Exception {
    AwsTemporaryToken awsTemporaryToken = s3UploadInfo.getAwsTemporaryToken();
    String bucketName = s3UploadInfo.getBucketName();
    String a = a(s3UploadInfo);
    AmazonS3 amazonS3Client = new AmazonS3Client(new BasicSessionCredentials(awsTemporaryToken.getAccessKeyId(), awsTemporaryToken.getSecretAccessKey(
    String str2 = a;
    str2 = MimeTypeMap.getSingleton().getMimeTypeFromExtension(g.a(this.b));
    if (str2 == null) {
        str2 = HttpPostBodyUtil.DEFAULT_BINARY_CONTENT_TYPE;
    }
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.f(str2);
    final FileInfo fileInfo = new FileInfo(bucketName, a, str);
    TransferUtility a2 = TransferUtility.a().a(amazonS3Client).a(this.c).a();
    acVar.a(d.a(new -$$Lambda$a$z64C0DJUWsO_u07N-uyeN4dj20w(a2.a(bucketName, a, this.b, objectMetadata, null, new TransferListener() {
        public void a(int i, long j, long j2) {
        }

        public void a(int i, TransferState transferState) {
            if (!acVar.a()) {
                if (transferState == TransferState.COMPLETED) {
                    acVar.a(fileInfo);
                } else if (transferState == TransferState.FAILED) {
                    acVar.a(new RuntimeException("aws upload failed"));
                } else if (transferState == TransferState.CANCELED) {
                    acVar.a(new CancellationException());
                }
            }
        }

        public void a(int i, Exception exception) {
            acVar.b(exception);
        }
    }), a2)));
}
```

Hooking point

- 앱에서 클라우드 자원에 접근하는 순간의 파라미터 값을 가로채기
- https://github.com/june5079/soFrida

클라우드 SDK를 사용하는 모바일 앱의 취약점 분석

[*] Return com.amazonaws.auth.BasicSessionCredentials.a [] -> [java.lang.String] : [java.lang.String]
ASIASEK3E6MUUSBHRJUG

[*] Return com.amazonaws.auth.BasicSessionCredentials.b [] -> [java.lang.String] : [java.lang.String]
07Zacgn9ouT1/Gounumri+w4m4jVUUYEsoPt+tTN

[*] Return com.amazonaws.auth.BasicSessionCredentials.c [] -> [java.lang.String] : [java.lang.String]
FQoGZXIvYXdzEN//////////wEaDH7yvPp0Tp4C4c5e8CKVAiIFydlroLWfiLFpOJil7opq16w8WxZ0UyGlddor/bng+6rPnHxUr+2jwiZ5nmI1IK8jJ4ojRchQfoCDT/QA9jja3hl72xFWlcof6kNRwIvXv9eSPsK3lVyjNECszFgdgfU2u/M4wUx/ht2hR2bM09kfO0bezInJnScF+Ib49ES3R8NxYx8A1P9003pT5lvVe2R9/ZgncycUYj+wYw1nCW5hxVomJfcAdh607SCcuXxr3PjMtT44i+aGUK625l2hXcjzHJCJlK7DYkp68sIt0B3FN+34JFwZydTQnCgfSY9Y2QTJbkUm6dZWldWSxmpww3YNjcddH4xN0GR0JSz7OlZYxN9rImIyEtCm6sAdGnU7wSuUO7goto335QU=

All Access Keys Extracted

- 클라우드 접근에 필요한 Credential정보 획득
- 획득한 키를 CLI Tool로 셋팅한 후 클라우드에 직접 접근 시도

클라우드 SDK를 사용하는 모바일 앱의 취약점 사례

```
june@JuneMacbook:~/Tools/soFrida$ aws s3 ls s3://*****/********/contract/
20**-**-** 15:39:00     366013 contract.pdf
20**-**-**  15:39:00     189136 contract.pdf
20**-**-**  15:39:01    1267294 contract.pdf
20**-**-**  15:39:01    2262738 contract.pdf
...
...
...
20**-**-**  15:39:02    1022818 _contract.pdf
20**-**-**  15:39:02   23702853 _contract.pdf
20**-**-**  15:39:06     850544 _contract.pdf
20**-**-**  15:39:06     850266 _contract.pdf
20**-**-**  15:39:07    2363402 _contract.pdf
.....
```

A 앱
내부 계약서 파일

데이팅 앱 B
사용자간 전송 파일

```
hyeonjun-ui-iMac:~ JunePark$ aws configure
AWS Access Key ID [**********TZEA]: AK[redacted]
AWS Secret Access Key [**********AcuO]: 7FRo[redacted]92LPnRADYhj
Default region name [us-east-1]: us-west-1
Default output format [None]:
hyeonjun-ui-iMac:~ JunePark$
hyeonjun-ui-iMac:~ JunePark$
hyeonjun-ui-iMac:~ JunePark$ aws s3 ls s3://[redacted]
                           PRE /
                           PRE MGRegisterTicket/
                           PRE _room_images/
                           PRE _tradeshow_floor_maps/
                           PRE app_d99ae545e6b869bca728ac0530fdac9f/
                           PRE channel_1/
                           PRE channel_110/
                           PRE channel_113/
                           PRE channel_114/
                           PRE channel_118/
```

클라우드 SDK를 사용하는 모바일 앱의 취약점 실태 (2019년)

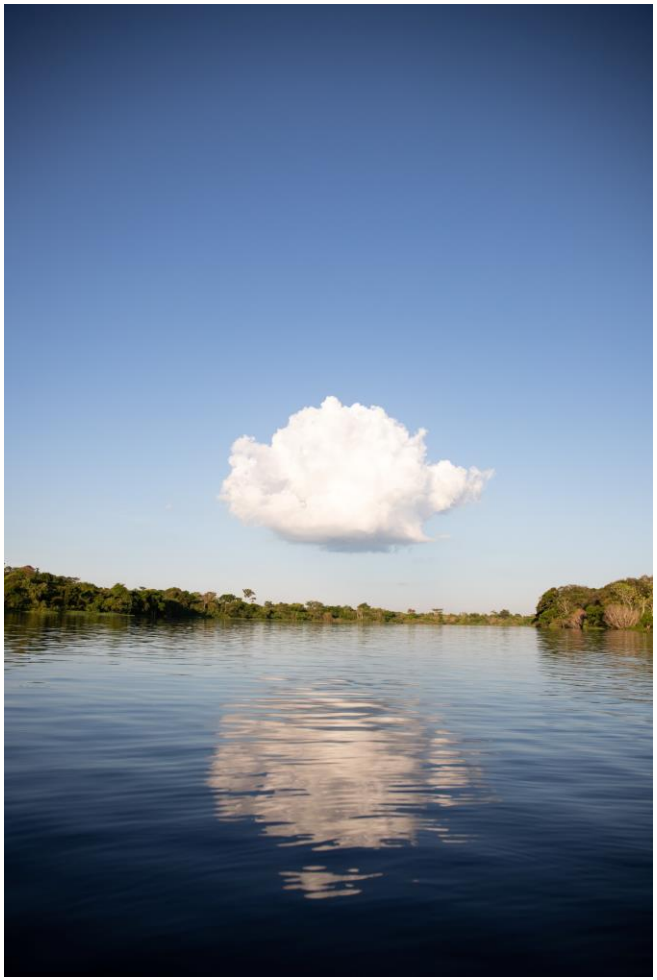
App Store Download Count	Number of potentially Vulnerable Apps
+100,000,000	12
+50,000,000	12
+10,000,000	98
+5,000,000	88
+1,000,000	318
<1,000,000	2172
Total	2700

date of collection : April 4, 2019 09:55 KS
Number of Analysis Target App : 4,000,000

개발자의 실수가 서비스에 치명적 영향을 줄 수 있습니다.

- 253 개 앱에서 서비스에 영향을 줄 수 있는 취약점 발견
- Critical : 53 Apps (30 of 1M+ Downloads)
- Medium : 13 Apps (12 of 1M+ Downloads)
- https://sofrida.github.io (상세 분석 결과 참조)

클라우드를 안전하게 사용하기 위한 방법



- 클라우드 키에 대한 적절한 권한 통제

- 루트키 사용 금지
- 키 하드코딩 금지

- 클라우드 보안에 대한 비용 투자

- 로깅 + 모니터링

- 내부 토론 + 외부 협업을 통한 리스크 관리

- 버그바운티 프로그램 참여