

【注意喚起】相次ぐ情報漏洩、その原因の一つは 「アプリ解析からのサーバ攻撃」

— AI で高度化する脆弱性探索、Android アプリを手がかりにした A2B 攻撃に注意 —

2026 年 10 月 13 日 脅威情報分析チーム LETTICE

Rev 1.1: 2026/10/13 15:00 更新

1. 概要

現在、日本国内の様々な組織において、個人情報などの漏洩事案が相次いで発生しています。これらの情報漏洩事案の原因の一つとして、攻撃者が一般に公開されている Android アプリを解析し、その解析結果を悪用してバックエンドサーバ¹を攻撃する手法が確認されました。本注意喚起では、このような攻撃手法を「A2B 攻撃 (App-to-Backend Attack)」と呼びます。

今回確認された攻撃では、攻撃者はまず、Google Play の日本地域で公開されている Android アプリを入手し、そのプログラムを解析することで、アプリが通信するバックエンドサーバの接続先、API (Application Programming Interface) の構成、通信に使用するパラメータなどの情報を収集します。そのうえで、収集した情報を基に、個々のバックエンドサーバの構成や API の仕様に応じた攻撃を組み立て、脆弱性を狙った攻撃を実行しています。

近年、AI の活用によってプログラムの解析や脆弱性の探索が効率化されており、こうした技術がサイバー攻撃に悪用されることが懸念されています。今回のように、公開アプリから得られる情報を手がかりとして、個々のシステムに合わせた攻撃が行われることには特に注意が必要です。

このような A2B 攻撃では、Android アプリ自体に脆弱性が存在しなくても、バックエンドサーバの API に認証・認可や入力値の検証などの不備があれば、個人情報の漏洩やサーバへの不正アクセスなどにつながる可能性があります。

Android アプリを公開している事業者には、自組織のアプリが攻撃者によって解析され、その結果がバックエンドサーバへの攻撃に利用されることを前提として、各種のセキュリティ対策を早急に点検することを強く推奨します。

本注意喚起は、脅威情報分析チーム LETTICE が独自に実施した調査・分析に基づき、確認された攻撃手法と事業者に推奨する対策をまとめたものです。

¹ バックエンドサーバ：Web サイトやアプリの裏側で、ログイン認証や利用者情報の管理などを行うサーバ。脆弱性が悪用されると、不正アクセスや個人情報の漏洩につながるおそれがあります。

2. A2B 攻撃の概要

今回確認された A2B 攻撃は、公開 Android アプリの解析と、その解析結果を利用したバックエンドサーバへの攻撃を組み合わせたものです。

攻撃者は、公開されている Android アプリを解析し、アプリが利用する API の接続先、エンドポイント（API の個別機能呼び出すためのアクセス先）、パラメータ、応答データの構造などを調査します。続いて、これらの情報からバックエンドサーバの攻撃対象となり得る箇所を特定し、個々の API の仕様や処理内容に合わせたリクエストを送信することで、脆弱性を探索・悪用します。

この攻撃の特徴は、不特定多数のサーバに同一の攻撃を試みるのではなく、公開アプリの解析によって得られた情報を利用し、対象となるバックエンドサーバごとに攻撃内容を変化させている点にあります。また、AI を利用したプログラム解析や脆弱性探索の効率化により、こうした攻撃がより多くのシステムに対して行われる可能性にも注意が必要です。

認証を必要としない API だけでなく、利用者アカウントによる認証後に利用できる API も攻撃対象となる場合があります。そのため、一般利用者向けに公開されている機能であっても、バックエンド側のアクセス制御に不備があれば、A2B 攻撃に悪用される可能性があります。

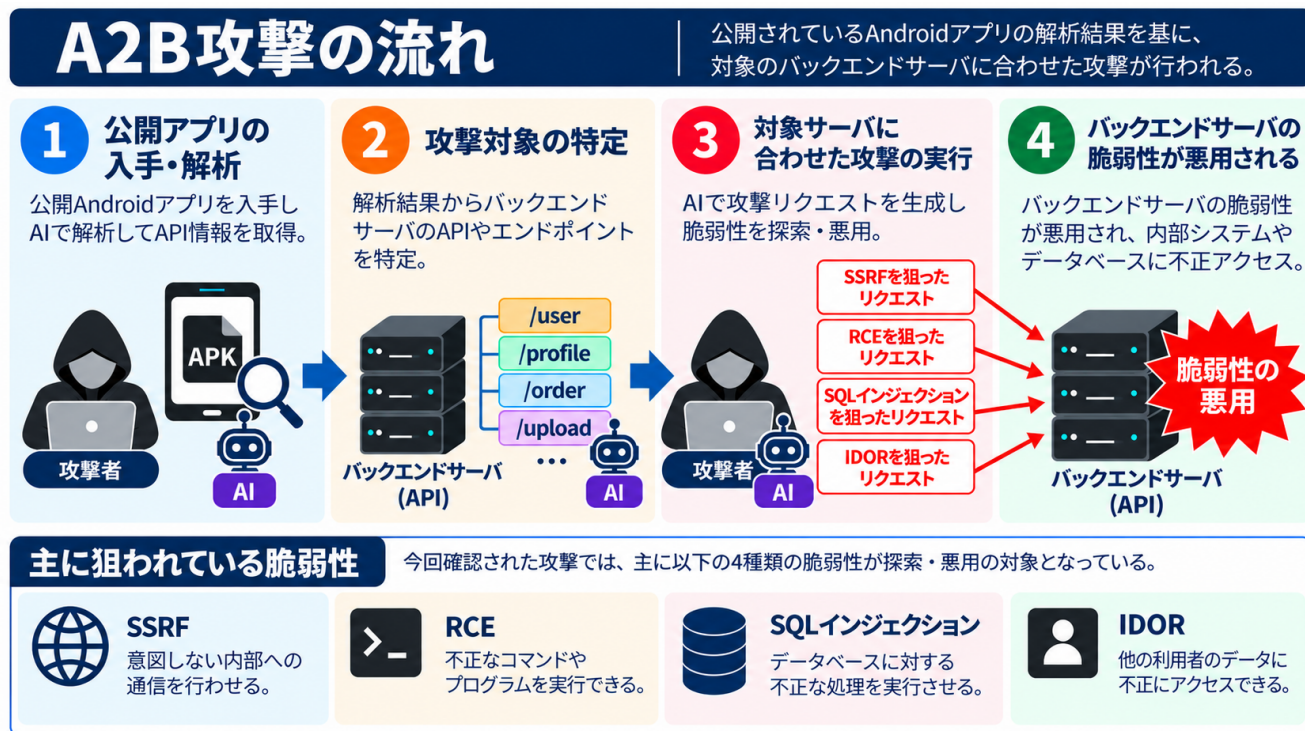


図 1 A2B 攻撃の流れと主に狙われている脆弱性

3. 攻撃対象となっている主な脆弱性

今回確認された A2B 攻撃では、主に以下の 4 種類の脆弱性が探索・悪用の対象となっています。

(1) SSRF (Server-Side Request Forgery)

SSRF は、外部からのリクエストによって、バックエンドサーバに本来意図していない通信を行わせる脆弱性です。この脆弱性が悪用されると、外部から直接アクセスできない内部システムやクラウド環境のメタデータなどにアクセスされ、機密情報が取得される可能性があります。

(2) RCE (Remote Code Execution)

RCE は、外部からのリクエストによって、バックエンドサーバ上で不正なコマンドやプログラムを実行できる脆弱性です。この脆弱性が悪用されると、サーバ内の情報の窃取、データの改ざん、不正なプログラムの実行など、深刻な被害につながる可能性があります。

(3) SQL インジェクション

SQL インジェクションは、API などへの入力を通じて、バックエンドのデータベースに対する不正な SQL 処理を実行させる脆弱性です。この脆弱性が悪用されると、データベースに保存されている個人情報などが不正に取得されたり、認証が回避されたりする可能性があります。

(4) IDOR (Insecure Direct Object Reference)

IDOR は、API などを利用者 ID やデータ ID といった識別子を指定する際に、適切なアクセス権限の確認が行われていないことで発生する脆弱性です。同様の問題を BOLA (Broken Object Level Authorization: オブジェクト単位の認可不備) とも呼びます。例えば、ある利用者が自身の情報を取得する API において、リクエスト中の識別子を変更するだけで、他の利用者の情報を取得できてしまう場合があります。この脆弱性が悪用されると、他の利用者の氏名、メールアドレス、その他の個人情報などが不正に取得される可能性があります。また、識別子が容易に推測・列挙できる場合には、大量の情報漏洩につながるおそれがあります。

4. Android アプリを公開する事業者に推奨する対策

Android アプリを公開している事業者には、以下の対策を早急を実施することを推奨します。

(1) 侵害の確認

API およびインターネット上に公開しているホストについて、特に以下の観点から不審なアクセスや侵害の有無を優先的に確認してください。

- API およびインターネット上に公開しているホストを列挙してください。対象には、アプリが利用する API、サブドメイン、旧バージョンの API、検証環境、クラウド上の公開サービスなども含めてください。
- 2026 年 8 月以降のアクセスログ、API ログ、認証ログなどを確認し、把握されている loC（IP アドレス、User-Agent 等）との一致や不審なアクセスの有無を調査してください。可能であれば、それ以前のログも確認してください。
- loC との一致に加え、認証なしでの大量の情報取得、連番 ID などを使ったアクセス、他会員情報の閲覧・変更、認証トークンの不審な発行、管理機能の利用など、通常と異なる API 操作がないか確認してください。
- 不審なアクセスが確認された場合は、認証の成否、アクセス先のデータ、情報の取得・変更、認証情報の悪用など、影響の有無と範囲を速やかに調査してください。

(2) 脆弱性の点検

API および公開ホストについて、特に以下の脆弱性や設定不備が存在しないか優先的に点検してください。

- 認可不備（IDOR/BOLA）：会員 ID などの識別子を変更することで、他者や他テナントの情報を閲覧・変更できないか確認してください。
- 認証・セッション管理の不備：認証トークンの不正発行、偽造、他者の認証情報の取得、認証回避などが可能でないか確認してください。また、固定された認証情報や暗号鍵が露出していないか確認してください。
- SQL インジェクション・NoSQL インジェクション：外部入力を介してデータベースの情報を不正に取得・変更できないか確認してください。
- SSRF：URL 取得、画像処理、文書変換などの機能を通じて、内部ネットワーク、クラウドのメタデータサービス、ローカルファイルなどへアクセスできないか確認してください。
- クラウドサービス・ストレージの設定不備：S3、GCS、Firestore などのサービスに、認証なし、または本来許可されていない権限でアクセスできないか確認してください。
- コード実行・ファイル操作に関する脆弱性：危険なデシリアライズ、任意ファイルの書き込み・上書き、不適切な管理機能の公開などがいないか確認してください。

(3) 暫定的な緩和策

攻撃による被害の発生や拡大を防止するため、業務への影響を考慮しながら、必要に応じて以下の暫定的な緩和策を実施してください。

- 国外からのアクセスや不要なアクセス元からの通信を一時的に制限してください。
- 不要な公開 API、管理機能、検証環境へのアクセスを停止または制限してください。
- API に対するアクセス頻度や連続的な識別子の指定などを監視し、必要に応じてレート制限や WAF 等によるアクセス制御を行ってください。
- 認証情報やトークンの漏洩・悪用が疑われる場合は、該当する認証情報の無効化・再発行などを実施してください。
- 脆弱性が確認された場合は速やかに修正してください。直ちに修正できない場合や、重大な脆弱性の有無を確認できず、暫定的な制限でもリスクを許容可能な水準まで低減できない場合は、該当する API や機能、必要に応じてサービス全体の一時停止を検討してください。

Android アプリを公開する事業者推奨する対策

A2B 攻撃による被害の防止・
早期発見のために、以下の対策を実施する。

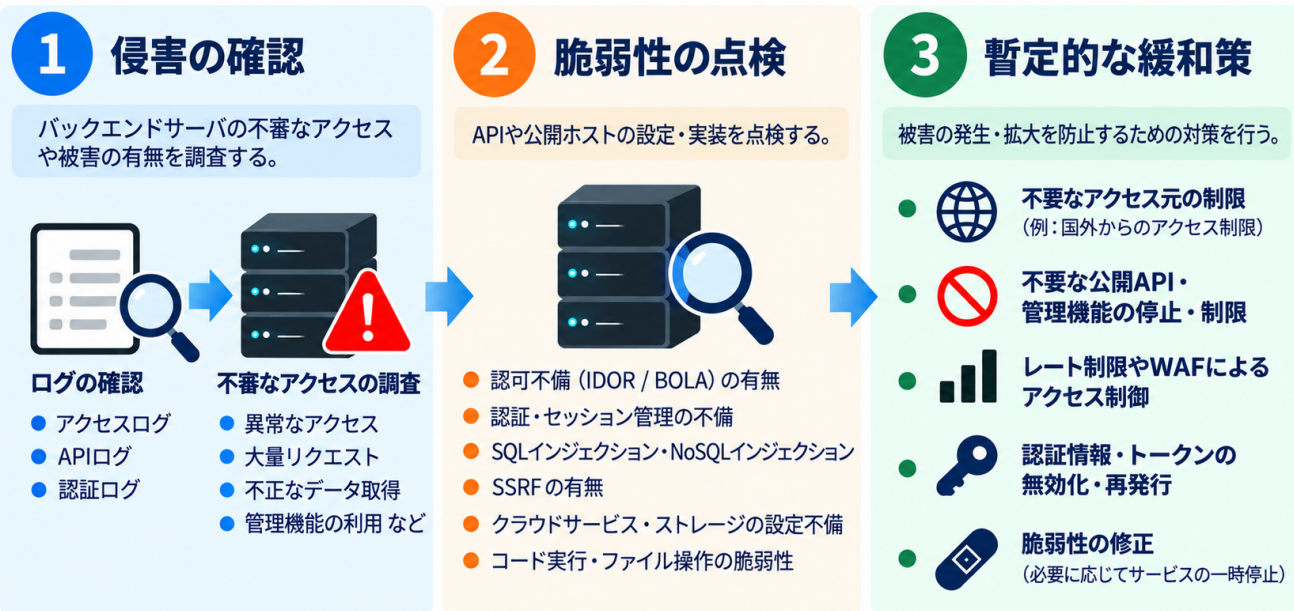


図 2 Android アプリを公開する事業者推奨する対策

なお、恒久的な対策としてシステムの設計・実装を見直す際には、攻撃者がアプリを介さずに API を直接操作できることを踏まえ、アプリ側の制御のみに依存せず、バックエンドサーバ側で適切な認証・認可や入力値の検証などを実施することが重要です。

5. まとめ

今回確認された A2B 攻撃は、一般に公開されている Android アプリを解析し、その結果を利用してバックエンドサーバの API や処理内容を把握したうえで、対象システムに合わせた攻撃を行うものです。国内で相次ぐ情報漏洩事案の原因の一つとして、こうした攻撃手法による情報の不正取得が確認されており、Android アプリを公開している事業者には早急な対応が求められます。

AI の活用によってアプリ解析や脆弱性探索が効率化される中、個々のバックエンドサーバの仕様に合わせた攻撃への備えが重要になっています。特に、SSRF、RCE、SQL インジェクション、IDOR の 4 種類の脆弱性について、バックエンド API の安全性を重点的に点検することを推奨します。

公開アプリが解析されること自体を完全に防ぐことは困難です。重要なのは、アプリの内部構造や API の仕様が攻撃者に知られた場合でも、バックエンドサーバの安全性が確保されるように設計・運用することです。

なお、同様の問題は Android アプリに限らず、iOS アプリや Web サービスのバックエンド API でも発生する可能性があるため、これらを提供する事業者においても適切なセキュリティ対策が重要です。

6. 攻撃に関連する IoC（侵害指標）

今回確認された攻撃に関連する IoC（Indicators of Compromise：侵害指標）の一部を以下に示します。これらの情報は、不審なアクセスの有無を確認する際の参考として活用してください。なお、IoC との一致だけで侵害の有無を断定することはできないため、アクセスログや通信内容などと併せて総合的に判断してください。

- 69.10.51[.]162
- 210.149.87[.]120
- 207.241.172[.]54

7. 参考文献

- [1] マクニカ「相次ぐ WEB システムからの情報漏洩事案について」
<https://security.macnica.co.jp/blog/2026/10/web-incidents2026.html> (2026 年 10 月 7 日)
- [2] 個人情報保護委員会「大規模な漏えい等事案を踏まえた対応について（注意喚起）」
https://www.ppc.go.jp/news/careful_information/261007_alert/?ref=topnews (2026 年 10 月 7 日)
- [3] JPCERT/CC「直近で相次いでいる国内組織における不正アクセスに関する注意喚起」
<https://www.jpcert.or.jp/at/2026/at260030.html> (2026 年 10 月 8 日)
- [4] 国家サイバー統括室「不正アクセスによる漏えい等の事案を踏まえた対応について」
<https://www.cyber.go.jp/pdf/press/taiou.pdf> (2026 年 10 月 9 日)
- [5] 総務省「不正アクセスによる情報漏えい等の事案を踏まえた対応について（注意喚起）」
https://www.soumu.go.jp/menu_kyotsuu/important/kinkyu02_000678.html (2026 年 10 月 9 日)

LETTICE (Leading Expert Team for Threat Intelligence Collection and Evaluation)



LETTICE は NICT のサイバーセキュリティネクサス (CYNEX) において、2024 年 2 月に発足した脅威情報の共同分析チームです。CYNEX に参画する組織の有志によって構成され、様々なサイバー脅威に関する情報を共同で分析し、得られた知見の共有に取り組んでいます。

更新履歴：

Rev 1.0：10/13 14:00 公開

Rev 1.1：10/13 15:00 更新（軽微な表現の修正）