

A2B攻撃

公開アプリの解析から
バックエンド侵害まで

App-to-Backend Attack

仕組み・観測事実・診断と対策

2026年10月17日時点の公開情報に基づく解説

A2B攻撃で問題になること

公開アプリが調査の起点になる

攻撃者はアプリの配布物や通信処理からAPIの使い方を調べ、対象に合わせたリクエストを組み立てる。APIは、アプリがサーバー上の機能やデータを利用するためのインターフェースである。

攻撃が成立するかはバックエンド側の制御で決まる

接続先が分かるだけで情報を取得できるとは限らない。サーバーが、利用者・対象データ・操作の権限や入力内容を適切に検証しているかが重要になる。

アプリとAPIを一体として点検する

アプリ解析で見つかる機能を検査対象に含め、実際のデータアクセスまで確認する。A2Bという呼称は攻撃過程を指しており、単一のCVEや共通の修正パッチを示すものではない。

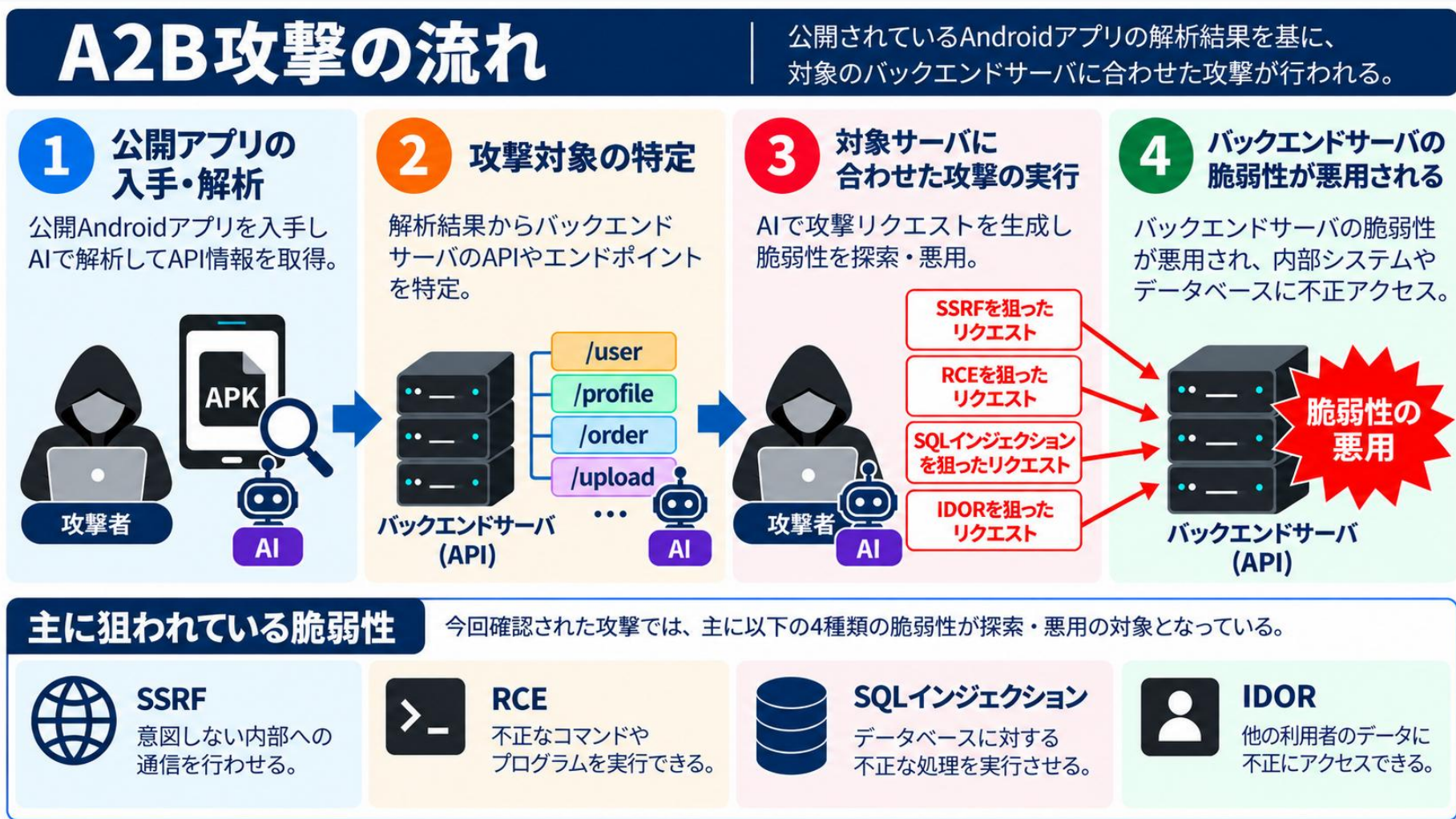
2026年10月の一次情報

各資料は公開日と観測範囲が異なる。次の3件を、この資料の事実確認の基礎とした。

公表日	公表主体	公表内容
10月8日 9日更新	JPCERT/CC	国内の連続不正アクセスについて複数の攻撃類型を公表。ケースBで、アプリ解析やAPI経由の不正操作を説明した。
10月13日	NICT／LETTICE	公開Androidアプリの解析結果を使う攻撃をA2Bと呼び、注意喚起した。詳細PDFは同日15時更新のRev 1.1。
10月15日	JPCERT/CC Eyes	攻撃元サーバーを調査した結果を公表。APK解析の痕跡と、一部の攻撃におけるAI利用を確認した。

これらの公表から、国内のすべての漏えい事案をA2Bや同一の攻撃者に帰属させることはできない。

NICTが示すA2B攻撃の全体像



原図は攻撃過程を説明する概念図である。AI利用の確認範囲は、後掲の調査結果と分けて扱う。

アプリ解析で分かる情報

APKはAndroidアプリの配布ファイルである。解析では、通信先だけでなく、リクエストの形式や機能の対応関係を調べられる。

調べる情報	分かること	診断への使い方
接続先とAPIパス	どの環境・どの機能へ通信するか。	API台帳と照合し、未把握の公開機能を確認する。
入力値とデータ型	会員ID、検索条件、URLなどの入力形式。	入力の検証と対象の認可を確認する。
トークンと署名	認証情報の付け方やリクエストの生成方法。	固定の秘密情報や、クライアントへの信頼を確認する。
応答と画面処理	返却項目や非表示の値の扱い。	不要な情報の返却や、画面側だけの制限を確認する。

これは診断観点を整理した表であり、上記の情報が得られたこと自体を脆弱性とは判定しない。

認証済みでも、すべての操作は許可されない

認証は、誰として操作しているかを確認する

有効なトークンを持つ一般会員であっても、他会員の注文や管理者向け機能にアクセスする権限まで得るわけではない。

認可は、その人が対象に対して何をできるかを確認する

サーバーは各リクエストで、利用者の権限、データの所有者、所属テナント、操作内容などを照合する必要がある。ログインの成否だけでは判断できない。

アプリの画面制限は、サーバーの認可を代替しない

ボタンを表示しない、IDの入力欄を設けないといった制限があっても、APIに直接届くリクエストを想定する。想定外の呼び出し順序や機能にも、サーバー側の制御が必要になる。

具体例：他会員の注文を取得できるBOLA

架空の会員APIで考える。会員Aは注文1001の所有者であり、注文1002は会員Bのものである。どちらも検証用データとする。

Aのトークンで行う要求	期待する結果	認可に不備がある場合
GET /orders/1001	A自身の注文を返す。	この要求だけでは不備を判定できない。
GET /orders/1002	Bの注文情報を返さず、アクセスを拒否する。	Bの氏名・住所・注文内容などを返してしまう。
IDを別の有効値へ変更	対象が変わっても毎回権限を確認する。	複数の他人の注文を取得できる可能性がある。

BOLAはオブジェクト単位の認可不備を指す。IDをUUIDなどに変えても、権限確認の欠落そのものは修正されない。

情報取得につながる別の不備

NICTが挙げる主な対象はIDOR／BOLA、SSRF、RCE、SQLインジェクションである。以下は各不備の一般的な成立条件と影響を整理したもの。

類型	成立する条件の例	想定される影響
SSRF サーバー側の不正な通信	URL取得機能などで宛先を十分に制限せず、サーバーが攻撃者指定先へ接続する。	サーバーから到達できる内部サービスなどへの不正アクセス。
RCE リモートコード実行	OSコマンドへの入力混入など、外部入力があるコードやコマンドの実行に影響する。	サーバー権限の範囲での情報取得、改ざん、プログラム実行。
SQL インジェクション	入力値をSQL文字列に混ぜるなど、データと命令の区別が崩れる。	DBの権限やクエリに応じた情報取得、変更、認証回避。

4類型は選択肢の例であり、すべてが順番に連鎖するという意味ではない。RCEには複数の原因がある。

アプリに埋め込んだ秘密情報の扱い

全利用者に配る秘密値は、抽出される可能性を考える

アプリ全体で共通のAPIキーや署名用秘密鍵を配布すると、抽出後にアプリ外で利用されるリスクがある。その値の保有だけで高い権限を与える設計は避ける。

APIキーと利用者の権限を区別する

キーがアプリ識別や利用量管理のための公開可能な値なのか、保有者に権限を与える認証情報なのかを確認する。文字列の名前だけでは、漏えいの影響を判断できない。

サーバーで検証できる権限と失効手段を用意する

利用者・用途・有効期限に応じて権限を制限し、漏えい時に失効できるようにする。難読化や独自署名を加えても、オブジェクト単位の認可は別途必要である。

攻撃元サーバーで確認された痕跡

APK解析に関するファイルが存在した

JPCERT/CCは、調査したサーバーで複数のAPKや、jadx・Blutterに関連するファイルを確認した。jadxはAndroidアプリの解析、BlutterはFlutterアプリの解析に使われる。

APIの脆弱性を調べる処理が存在した

APKから特定したAPIに対し、SSRFなどを調査した痕跡を確認している。公開例には、HTTP 302のリダイレクトを返す検証用とみられるスクリプトが含まれる。

証拠から分かる範囲を限定する

解析ツールや検証ファイルの存在は、すべての対象への侵入成功を意味しない。同記事の「調査その3」は今回の攻撃と直接関係しない別サーバーであり、同じ攻撃の続きとして扱わない。

AIについて確認されたことと、分析上の推論

確認された事実

JPCERT/CCは、攻撃元サーバーの調査を通じて、攻撃者がAIを使い、既知の脆弱性のスキャンやAPI経由の攻撃を行っていたと報告した。

技術的な含意としての推論

アプリごとに異なるコードやデータ形式を読み取り、検証用リクエストを作る作業をAIが支援すれば、対象別の調査に必要な工数が下がる可能性がある。これは攻撃の規模に関する推論である。

公表資料から確定できないこと

全A2B攻撃でのAI利用、特定企業の漏えいとAIの因果関係、攻撃全体の自律性、使用モデル、効率化の倍率は確定できない。通信が高速・大量であることだけでもAI利用は判定できない。

A2Bの成立にAIは必須ではない。防御では、攻撃者のツールによらずAPI側の制御を検証する。

調査対象は、現行アプリの通常画面より広い

診断対象の母集団を確定するには、アプリ解析の結果を、API仕様書・ゲートウェイ設定・実アクセスログと照合する必要がある。

確認対象	見落としやすい点
APIの版と公開ホスト	現行版だけでなく旧API、別サブドメイン、直接到達できる接続先も確認する。
検証・運用用の環境	検証環境や管理機能が外部公開され、本番データや本番相当の権限を持っているか確認する。
利用者とテナント	未認証、一般会員、他会員、別テナント、管理者で許可する操作を定義する。
データの保存先と連携先	外部ストレージや連携サービスについて、誰が何にアクセスできるかと管理責任を確認する。

アプリから見えないAPIも存在し得る。アプリ解析だけで、バックエンドの全機能を網羅したとは判断しない。

A2Bを想定した診断の進め方

以下は一次資料とOWASPの観点をもとにした診断設計案である。許可された対象と検証用アカウントで、機能と権限の対応を確認する。

工程	実施内容	残す成果物
1. 対象把握	配布アプリと通信からAPIを抽出し、既存の台帳・仕様書と照合する。	版、接続先、機能、認証方式を対応づけた一覧。
2. 権限検証	本人・他人・別テナント・管理機能について、許可と拒否の両方を確かめる。	期待結果と実際の応答、検証用データの所有関係。
3. 処理の検証	外部URL取得、検索、アップロードなど、機能に応じた入力の扱いを確認する。	不備が成立する入力条件と、到達できる影響範囲。
4. 再検証	修正後の拒否を確認し、正規機能への影響と再発防止テストを確認する。	修正内容、再検証結果、未確認範囲。

検証成功率や検出数だけでは網羅性を示せない。調査したAPI・権限・操作と、未調査範囲を併記する。

認可テストは、利用者と対象の組合せで考える

架空の注文サービスの例。利用者AとBを同一テナント、Cを別テナントとし、通常会員は自分の注文だけを閲覧できる仕様を想定する。

操作主体	操作対象	仕様上の期待	確かめる点
未認証	Aの注文	拒否	認証なしで注文本文を返さないか。
会員A	Aの注文	許可	正しいデータだけを返すか。
会員A	Bの注文	拒否	同じ権限の別会員へ越境しないか。
会員A	Cの注文	拒否	テナントの区分を越えていないか。
会員A	管理者向け処理	拒否	非表示の管理機能も拒否するか。

閲覧だけでなく更新・削除・一括取得も確認する。共有や委任がある実システムでは、その業務仕様に合わせて期待値を定める。

各対策が効く範囲

対策は目的ごとに組み合わせる必要がある。アプリ解析を難しくする対策だけでは、サーバー上の権限確認を完成させられない。

対策	主な役割	残る課題
難読化・改ざん対策	クライアント解析や改変に必要な労力を増やす。	解析後に届く要求の正当性は、サーバー側で確認する。
WAF・レート制限	既知の攻撃パターンや過剰な利用を抑制する。	正常形式の要求による他人のデータ取得は、別途認可で防ぐ。
認証・認可	利用者を確認し、対象と操作ごとに権限を検証する。	認可が正しくても、SQLiやSSRFなどの処理上の不備は残り得る。
入力・出力と実行環境の制御	入力の意味、接続先、返却データ、実行権限を制限する。	機能の変更時や旧APIでも同じ制御が働くか、継続して確認する。

本表は対策の役割を整理した技術的評価であり、特定のWAF製品や難読化製品の性能比較ではない。

恒久対策はAPIの処理と権限に実装する

認証と認可をサーバー側で完結させる

トークンの正当性・用途・有効期限を検証する。利用者が送る所有者IDや権限値をそのまま信用せず、実際のデータとの関係を照合する。権限が確認できなければ拒否する。

入力を処理する機能ごとに対策する

SQLはパラメータ化し、外部URL取得は宛先を制限する。リダイレクトは無効化するか遷移先まで確認する。OSコマンド呼出しは可能な限り安全なAPIに置き換え、不要な実行権限を与えない。

情報を返す範囲と、保管する範囲を小さくする

必要な項目だけを返し、管理機能と一般会員の機能を分離する。旧APIや不要な公開環境を廃止し、サービスアカウント・DB・ストレージの権限を用途に応じて制限する。

検知には、誰が何を取得したかが必要になる

HTTPの成功・失敗だけでは、正当なアクセスかどうかは判断できない。アプリケーションの操作ログと、認証・API・DBの記録を関連づける。

残す情報	調べる観点
主体と対象の関係	利用者ID、テナント、対象オブジェクト、実行操作、認可結果を照合する。
取得・変更した量	要求数に加え、ユニークな対象数、返却件数、応答サイズや更新件数を見る。
操作の連続性	対象IDの連続変更、短時間の広範な参照、管理機能の利用、トークン発行の変化を見る。
相関と調査可能性	時刻、リクエストID、接続元、APIの版を残し、関連システム間で追跡できるようにする。

上記は監視設計案。ログへパスワード・生のトークン・不要な個人情報を記録せず、保存期間と閲覧権限を決める。

公開IoCの照合と使い方

NICTの詳細資料にある3つのIPは、JPCERT/CCのケースBにも掲載されている。IPは誤接続を避けるため無害化表記にした。

IPアドレス	NICT／LETTICE	JPCERT/CC ケースB
69.10.51[.]162	10月13日資料に掲載	10月8日公開分に掲載
210.149.87[.]120	10月13日資料に掲載	10月8日公開分に掲載
207.241.172[.]54	10月13日資料に掲載	10月9日の追記分に掲載

一致は調査の手掛かりになるが、侵害や同一攻撃者の確定には使えない。NICTはJPCERT/CC資料を参照しており、重複を独立した証拠として数えない。

侵害が疑われる場合の初動

証拠の保全と、継続中の被害の抑制を並行する

API・認証・操作ログや設定を保全し、不正取得が続く機能は停止または制限する。原因を調べる前にログを消したり、必要な記録を失う変更を加えたりしない。

アクセスされたことと、情報が出たことを分けて確認する

要求先、認証・認可の結果、返却内容、取得件数、更新操作を調べる。影響対象となり得る件数と、実際に取得を確認した件数は区別して記録する。

不備の修正と認証情報の対処を対応づける

漏えい・悪用が疑われる認証情報を失効し、影響する機能を修正する。修正後は同条件で再検証し、旧版APIや別環境から同じ不備を利用できないことも確認する。

NICTは2026年8月以降を中心に、可能ならそれ以前も含めたログ確認を推奨している。

A2B対策として優先したい確認

アプリから到達できるAPIを把握しているか

公開配布物から分かる機能と、組織が管理するAPI台帳を突き合わせる。差分にある旧版・管理機能・検証環境を確認する。

一般会員の要求で他人のデータに届かないか

本人・他人・別テナント・管理者の区分で権限を検証する。アプリを使わずに同じAPIを呼び出す場合も、期待どおり拒否できる状態にする。

不正操作を防ぎ、起きた場合に追跡できるか

認可と入力処理を修正し、秘密情報や実行権限を制限する。誰が、どの対象に、何を行ったかを調査できる記録を残す。

A2Bを想定した評価の中心は、公開クライアントの情報とバックエンドの実際の制御を照合することである。

参考文献 1／3：観測事実と注意喚起

[1] NICT／LETTICE 詳細注意喚起

<https://cynex.nict.go.jp/pdf/library/lettice/261013-01.pdf>

2026-10-13、Rev 1.1。A2Bの定義、観測された攻撃過程、4類型、IoC。

[2] NICT CYNEX 公表ページ

<https://cynex.nict.go.jp/library/261013-01.html>

2026-10-13。注意喚起の要旨と事業者への対応要請。

[3] JPCERT/CC 注意喚起 JPCERT-AT-2026-0030

<https://www.jpCERT.or.jp/at/2026/at260030.html>

2026-10-08公開、10-09更新。ケースBのAPI不正操作、攻撃元IPと観測上の留保。

[4] JPCERT/CC Eyes 「連続不正アクセス事案の攻撃手法に関する調査」

https://blogs.jpCERT.or.jp/ja/2026/10/web_api_attacks.html

2026-10-15。攻撃元サーバーの調査、APK解析ツール、AI利用の確認。

参考文献 2／3：解析・認可・API管理

[5] OWASP API1:2023 Broken Object Level Authorization

<https://api-security.owasp.org/editions/2023/en/0xa1-broken-object-level-authorization/>

オブジェクト単位の認可不備と対策。

[6] OWASP Authorization Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html

最小権限、既定拒否、各リクエストでの権限確認とテスト。

[7] OWASP MASTG-TECH-0017: Decompiling Java Code

<https://mas.owasp.org/MASTG/techniques/android/MASTG-TECH-0017/>

Android配布物のデコンパイルと解析可能性。

[8] OWASP API9:2023 Improper Inventory Management

<https://api-security.owasp.org/editions/2023/en/0xa9-improper-inventory-management/>

APIホスト、旧版、環境、データ連携の棚卸し。

[9] OWASP REST Security Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html

APIキーの限界、トークン検証、入力検証、監査。

参照日：2026年10月17日。本文の角括弧番号は本一覧に対応する。

参考文献 3／3：入力処理とログ

[10] OWASP Server Side Request Forgery Prevention Cheat Sheet

[https://cheatsheetseries.owasp.org/cheatsheets/Server Side Request Forgery Prevention Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html)

宛先制限、リダイレクト、ネットワーク側の制御。

[11] OWASP SQL Injection Prevention Cheat Sheet

[https://cheatsheetseries.owasp.org/cheatsheets/SQL Injection Prevention Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html)

パラメータ化クエリとデータベース権限。

[12] MITRE CWE-78: OS Command Injection

<https://cwe.mitre.org/data/definitions/78.html>

入力によるコマンド実行と防止策。

[13] OWASP Logging Cheat Sheet

[https://cheatsheetseries.owasp.org/cheatsheets/Logging Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html)

認証・認可・操作の記録と秘密情報の除外。

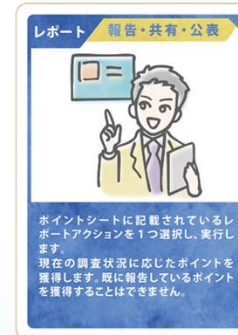
CoRepo VR

インシデント対応を、VRで体験する。

証拠や情報を集めよう



情報を整理しよう



10/24. 11/28. 12/26



CoRepoはJNSAの所有するコンテンツです。
VRChat版はJNSAの許諾を得て再構成したものです。

主催・イベント運営：セキュリティ集会
後援：NPO 日本ネットワークセキュリティ協会
協力：VRC技術・学術系イベントHUB

