

FOREST SECURITY SAMPLE REPORT

セキュリティ診断 技術報告書 (サンプル)

サンプル商事株式会社 様 在庫連携アプリ「SampleSync」

発行 株式会社Forest Book (Forest Book Inc.)

株式会社Forest Book

2026年11月28日

本資料の複製・社内配布は自由。再配布時は出典を明記のこと。

本書について

本書は架空の会社「サンプル商事株式会社」と架空のアプリ「SampleSync」を対象とした、報告書の記載例です。登場する会社名、アプリ名、アカウントID、ファイルパス、指摘内容はすべて架空であり、実在の顧客・案件・システムとは関係ありません。営業資料として報告書の形式を示す目的で作成しています。

指摘内容は、ShopifyアプリとAWS サーバーレス構成で起こりうる典型的な問題を、Forest Book の自社検証環境で再現できるものに限って構成しています。

表紙

項目	内容
報告書名	SampleSync セキュリティ診断 技術報告書
宛先	サンプル商事株式会社 様
作成者	株式会社Forest Book（Forest Book Inc.）
商品	A 標準ベースライン診断（Shopifyアプリ+AWS）
検査期間	2026-11-10 ～ 2026-11-21
送信元IPアドレス	203.0.113.45（文書用の例示アドレス）
版数	1.0（2026-11-28）
外部レビュー	実施（2026-11-26、外部レビュー担当1名）
取扱い	秘密。サンプル商事株式会社様の社内および委託先の関係者に限る

本報告書の位置づけ

- 本報告書は、検査期間中に、合意した検査範囲と手法で確認した結果を記載したものです。検査時点・検査範囲の結果であり、対象システムに脆弱性がないことを保証するものではありません
- 「実施しなかった検査」に記載した項目は確認していません。網羅的なペネトレーションテストではありません
- 重大度は Forest Book の3軸基準（攻撃の成立しやすさ、影響を受ける資産、業務影響）で判定しています
- 修正の実施と最終承認はサンプル商事株式会社様の責任で行われます。Forest Book は推奨修正を提示し、合意した範囲で再確認を行います
- Forest Book は本報告書の対象資産の開発に関与していません

検査範囲

対象資産

区分	対象	備考
Shopifyアプリ	SampleSync（公開アプリ、クライアントID末尾 7f3a）	管理画面（埋め込み）、Webhook受信（orders/create、app/uninstalled、コンプライアンス3種）、App Proxy は未使用
リポジトリ	sample-shoji/samplesync-app（コミット 4e1c9b2）、sample-shoji/samplesync-infra（コミット a07d3f1）	2026-11-10 時点
AWSアカウント	末尾 2210、ap-northeast-1	読み取り専用ロール ForestBookReadOnly（外部ID付き、2026-12-05 失効）
APIエンドポイント	14件（付録A）	顧客提出の一覧に基づく
動的確認の環境	https://staging.samplesync.example	本番は対象外

実施した検査

チェックリスト	項目数	実施	該当なし	未実施
Shopifyアプリ固有	28	23	4	1
AWSサーバーレス	21	18	2	1
Web/API（ASVS 5.0.0 参照）	32	26	5	1
CI/CD	15	14	0	1

実施した手法：静的解析（Semgrep 1.x、gitleaks 8.x、Shopify CLI app security check）、設定点検（AWS CLI、Prowler 5.x）、ステージング環境での手動動的確認（Burp Suite Community、OWASP ZAP ベースライン、1秒あたり5リクエスト上限）。

実施しなかった検査とその理由

項目	理由
本番環境への動的検査	同意書で対象外。ステージングと本番は同一コミットをデプロイしていることを CI ログで確認
能動スキャン（ZAP full scan、Burp Active Scan）	顧客の要望により受動スキャンと手動確認のみ
App Proxy 関連項目（SH-16～18）	アプリが App Proxy を使用していないため該当なし
SH-24 PCD Level 2 統制の根拠確認	顧客側の方針文書の提出が検査期間内に間に合わなかった。再診断時に確認予定

項目	理由
AW-MON-01 予算アラート	顧客ロールに <code>budgets:ViewBudget</code> が含まれておらず確認できなかった
WA-16 GraphQL	自前の GraphQL API を公開していないため該当なし
CI-14 本番デプロイの承認ゲート	Environment 設定の読み取り権限がなく、顧客管理者のスクリーンショット提出待ち
Shopify本体、決済代行、CDN	第三者資産のため対象外

使用したツールとバージョン

ツール	バージョン	用途
Semgrep	1.x（作業ログに記録）	静的解析
gitleaks	8.x	シークレット検出
Shopify CLI	3.x	app security check
Prowler	5.x	AWS設定点検
OWASP ZAP	stable（Docker）	受動スキャン
Burp Suite Community	2026.x	手動確認

総合所見

重大度別の件数

重大度	件数
Critical	1
High	2
Medium	2
Low	0
Info	0
合計	5

全体の傾向

SampleSync は Shopify CLI の React Router テンプレートを基に構築されており、管理画面のルートは `authenticate.admin` で一貫して保護されています。HTTPS、HSTS、CSP の `frame-ancestors` も設定されており、フロントエンドの基本的な防御は整っています。

一方で、テンプレートの外側に独自実装した部分に問題が集中しています。Webhook 受信は Express ベースの別サービス（Lambda）で実装されており、そこで署名検証とテナント分離の不備が見つかりました。セッショントークンの検証は署名と有効期限のみで、発行先ストアの照合がないため、別のストアからのアクセスで他ストアのデータに到達できる状態でした（FB-SMPL-001）。この1件は単独で Critical です。

AWS 側では、Lambda 実行ロールの権限がワイルドカードで付与され、Shopify のアクセストークンが環境変数に平文で置かれています。これらは単独では外部から成立しませんが、FB-SMPL-001 のようなアプリ層の問題と組み合わせると、1ストアの侵害が全ストアのデータに波及する経路になります。総合所見として、アプリ層のテナント分離の修正を最優先とし、並行して AWS 側の権限と秘密情報の扱いを是正することを推奨します。

良い点として、CI/CD は GitHub Actions から OIDC で AWS にデプロイしており、長期アクセスキーは使われていません。依存パッケージの Dependabot も有効で、未対応の High アラートはありませんでした。

今すぐ着手すべき指摘

指摘ID	件名	重大度	推奨する着手時期
FB-SMPL-001	セッショントークンの発行先ストアを照合せず、リクエストの shop パラメータでデータを絞り込んでいる	Critical	即時（1週間以内）
FB-SMPL-002	注文作成Webhookの署名検証がJSON再シリアライズ後のボディで行われ、比較がタイミング安全でない	High	2週間以内
FB-SMPL-003	同期ジョブ一覧APIがジョブIDのみで取得し、ストア識別子で絞り込んでいない	High	2週間以内

指摘事項

指摘ID：FB-SMPL-001

件名：セッショントークンの発行先ストア（dest）を照合せず、リクエストの shop クエリパラメータでデータを絞り込んでいる

重大度：Critical（攻撃の成立しやすさ：高／影響資産：全ストアの在庫・注文連携データ／業務影響：他テナントへの波及）

対象：GET /api/sync/status、GET /api/sync/history、POST /api/sync/run
（samplesync-app/app/lib/auth.server.ts、app/routes/api.sync/*.ts。ステージング環境）

証拠：auth.server.ts 31～48行目の verifySessionToken は jwt.verify(token, secret, { algorithms: ['HS256'] }) で署名と exp を検証した後、payload.aud と payload.dest を参照していない。各ルートは const shop = url.searchParams.get('shop') でストアを決定し、prisma.syncJob.findMany({ where: { shop } }) に渡している（api.sync.status.ts 18行目）。ステージングで、開発ストアB用に発行された正規のセッショントークンを Authorization ヘッダーに付け、?shop=store-a.myshopify.com を指定したところ、ストアAの同期履歴（商品ID、在庫数、最終同期時刻）が 200 で返った（HTTPログ：付録F-01、個人情報には含まれない）

影響：SampleSync をインストールした任意のストアのスタッフが、他の全ストアの同期履歴・在庫データを閲覧でき、POST /api/sync/run で他ストアの同期を実行できる。在庫データは保護顧客データには当たらないが、history に含まれる注文IDと配送先都道府県は注文データに紐づく情報であり、ストア間のデータ分離の不備として Shopify PCD Level 1「最小限のデータ処理」の前提を崩す

再現条件：SampleSync をインストールした2つのストアのスタッフ権限。前提条件は他にない。報告前に 2026-11-25 に再現手順を最初から実行し直し、再現を確認した

推奨修正：verifySessionToken で aud がアプリのクライアントIDと一致すること、iss と dest のドメインが整合することを検証し、dest から取り出したストアドメインを戻り値とする。全ルートで searchParams.get('shop') の使用をやめ、検証済みの dest 由来のストアを where 条件に使う。React Router テンプレートの authenticate.admin(request) が返す session.shop に統一するのが最も確実。Shopify 公式の検証手順 (exp、nbf、aud、iss/dest) は <https://shopify.dev/docs/apps/build/authentication-authorization/session-tokens/set-up-session-tokens> を参照

関連要件：Shopify Set up session tokens (aud はクライアントIDと一致、iss と dest の整合) / Shopify App security checks request-derived-shop-scope、missing-tenant-isolation (High) / ASVS 5.0.0 9.2.3、8.4.1、8.2.2 / OWASP API Security Top 10 2023 API1

修正担当（顧客記入）：

期限（合意）：

再確認結果：未実施

検査範囲の注記：対象3ルートで確認した。同じ verifySessionToken を使う他の管理画面ルート（付録Aの6件）は、authenticate.admin 経由で session.shop を使っており、本指摘の影響を受けないことをコードで確認した。動的確認は3ルートのみ

指摘ID：FB-SMPL-002

件名：注文作成Webhookの署名検証がJSON再シリアライズ後のボディで行われ、比較がタイミング安全でない

重大度：High（攻撃の成立しやすさ：中／影響資産：注文・顧客データ／業務影響：不正な注文データの登録）

対象：POST /webhooks/orders/create (samplesync-infra/functions/webhook/src/verify.ts。ステージング環境)

証拠：verify.ts 42行目で const digest = createHmac('sha256', secret).update(JSON.stringify(req.body)).digest('base64') としており、express.json() でパース済みのオブジェクトを再シリアライズしている。比較は 44行目 return digest === req.get('X-Shopify-Hmac-SHA256')。ステージングで、同一内容の JSON の空白位置を変えた2通りのリクエストを送ったところ、一方は 200、他方は 401 となり、検証結果が送信形式に依存することを確認した（付録F-02）。CloudWatch Logs には直近30日で 401 が 312件記録されており、うち 287 件は Shopify の送信元 IP 帯からのものだった

影響：正規の通知の一部が拒否されており、Shopify は応答がエラーの場合 4時間で 8回再試行し、8回連続失敗で Admin API 経由の購読を自動削除する。運用者が検証を無効化する誘因が生じており、無効化された場合は任意の注文データを登録できる状態になる。=== による比較は、現時点で実用的な攻撃に直結しないが、タイミング差を利用した署名推測の余地を残す

再現条件：ステージングの Webhook URL と任意の注文 JSON。署名は Shopify から送られた正規のものを使用。報告前に 2026-11-25 に再現を確認した

推奨修正: Webhook ルートにのみ `express.raw({ type: 'application/json' })` を適用し、生ボディ (Buffer) で HMAC-SHA256 を計算する。比較は

`crypto.timingSafeEqual(Buffer.from(digest, 'base64'), Buffer.from(header, 'base64'))` とし、長さ不一致時は `false` を返す。X-Shopify-Webhook-Id を DynamoDB に条件付き書き込み (`attribute_not_exists`) して重複配信を無視する。Shopify 公式の手順は <https://shopify.dev/docs/apps/build/webhooks/subscribe/https> を参照

関連要件: Shopify Webhooks HTTPS delivery (署名はクライアントシークレットと生のリクエストボディから生成。 `express.json()` の前に生ボディを取得) / Shopify Webhooks overview (X-Shopify-Webhook-Id による重複排除) / ASVS 5.0.0 9.1.1、11.4.1 / OWASP API Security Top 10 2023 API2

修正担当 (顧客記入):

期限 (合意):

再確認結果: 未実施

検査範囲の注記: 本指摘は対象エンドポイント1件について確認した。 `app/uninstalled` とコンプライアンス Webhook 3種は同じ `verify.ts` を使用しているが、個別の動作確認は行っていない

指摘ID: FB-SMPL-003

件名: 同期ジョブ詳細APIがジョブIDのみでレコードを取得し、ストア識別子で絞り込んでいない

重大度: High (攻撃の成立しやすさ: 高 / 影響資産: 他ストアの同期ジョブ詳細 (注文ID、SKU、数量、エラーメッセージ) / 業務影響: 特定ストアのデータ漏えい)

対象: GET `/api/sync/jobs/:id` (`samplesync-app/app/routes/api.sync.jobs.$id.ts` 14行目。ステージング環境)

証拠: `const job = await prisma.syncJob.findUnique({ where: { id: params.id } })`。`authenticate.admin(request)` は通過しているが、`session.shop` を `where` 条件に使っていない。ジョブIDは連番 (整数) で、ステージングで開発ストアBのセッションから `/api/sync/jobs/1041` (ストアAのジョブ) を取得できた (付録F-03)

影響: SampleSync をインストールした任意のストアのスタッフが、ジョブIDを順に指定して他ストアの全同期ジョブの詳細を取得できる。エラーメッセージには Shopify Admin API の応答本文が含まれる場合があり、注文の付随情報が漏れる

再現条件: 2つのストアのスタッフ権限と、連番のジョブID。報告前に 2026-11-25 に再現を確認した

推奨修正: `prisma.syncJob.findFirst({ where: { id: params.id, shop: session.shop } })` とし、該当なしは 404 を返す。ジョブIDを UUID v4 等の推測困難な値に変更することも推奨するが、ID の変更だけでは認可の代替にならない

関連要件: Shopify App security checks missing-tenant-isolation (High) 「テナント所有の読み取りはすべて `where` 句に `shop` を含める」 / ASVS 5.0.0 8.2.2、8.4.1 / OWASP API Security Top 10 2023 API1

修正担当 (顧客記入):

期限 (合意):

再確認結果: 未実施

検査範囲の注記: 付録Aの全14ルートについて、ID を受け取るクエリの `where` 条件を静的に確認した。本指摘以外の ID 受け取りルート (3件) は `session.shop` を条件に含んでいた。動的確認は本ルートのみ

指摘ID：FB-SMPL-004

件名：Lambda 実行ロールに `dynamodb:*` と `Resource: "*"` が付与されている

重大度：Medium（攻撃の成立しやすさ：低／影響資産：DynamoDB 全テーブル／業務影響：他の脆弱性と連鎖した場合に全ストアのデータへ到達）

調整理由：単独では外部から成立しないため、「成立しにくいものを Critical にしない」調整を適用した。FB-SMPL-001 と連鎖した場合の経路として総合所見に記載

対象：IAM ロール `samplesync-api-role`（Lambda 関数 `samplesync-api`、`samplesync-webhook` が共用）。本番・ステージング共通

証拠：`aws iam get-role-policy --role-name samplesync-api-role --policy-name inline-ddb` の出力に `{"Effect": "Allow", "Action": "dynamodb:*", "Resource": "*"}`（付録F-04）。IaC では `samplesync-infra/lib/api-stack.ts` 88行目 `actions: ['dynamodb:*'], resources: ['*']`。関数のコードが実際に呼ぶ API は `GetItem`、`PutItem`、`Query`、`UpdateItem` の4種で、対象テーブルは `SyncJobs` と `Sessions` の2つ。Prowler

`iam_inline_policy_no_administrative_privileges` は PASS（サービス限定のため）だが、Security Hub IAM.21 相当のワイルドカードアクションに該当

影響：関数内でコード実行や SSRF が成立した場合、アカウント内の全 DynamoDB テーブル（他システムの `CustomerMaster` テーブルを含む）の読み書き・削除が可能になる。2関数がロールを共用しているため、Webhook 受信関数の侵害が管理 API 側のデータにも及ぶ

再現条件：IAM の読み取り権限で確認可能。動的な再現は行っていない

推奨修正：関数ごとにロールを分け、アクションを `dynamodb:GetItem`、`PutItem`、`Query`、`UpdateItem` に、リソースを対象2テーブルの ARN（と必要なインデックス）に限定する。CDK では `table.grantReadWriteData(fn)` を使う

関連要件：AWS IAM best practices（最小権限）／Security Hub IAM.21「IAM customer managed policies that you create should not allow wildcard actions for services」（インラインポリシーのため直接の対象外だが同趣旨）／ASVS 5.0.0 13.2.2

修正担当（顧客記入）：

期限（合意）：

再確認結果：未実施

検査範囲の注記：アカウント内の Lambda 実行ロール 4件すべてを確認した。他の2ロールはテーブル ARN を限定していた

指摘ID：FB-SMPL-005

件名：Lambda 環境変数に Shopify オフラインアクセストークンと外部在庫 API のキーが平文で設定されている

重大度：Medium（攻撃の成立しやすさ：低／影響資産：全ストアの Admin API アクセス権、外部在庫 API／業務影響：内部者・CI 侵害時の即時のトークン流出）

調整理由：環境変数は AWS マネージドキーで保存時暗号化されているが、

`lambda:GetFunctionConfiguration` 権限で平文が取得できる。外部から直接は成立しないため Medium とした。トークンが全ストア分を1つの環境変数に JSON で保持している点は、単一ストア侵害時の影響を広げるため High への引き上げを外部レビューで検討したが、取得経路が内部に限られることから Medium を維持した

対象：Lambda 関数 samplesync-webhook の環境変数 SHOP_TOKENS、INVENTORY_API_KEY。本番・ステージング

証拠：aws lambda get-function-configuration --function-name samplesync-webhook --query 'Environment.Variables' | jq 'keys' の出力に SHOP_TOKENS、INVENTORY_API_KEY（値は記録していない。付録F-05）。laC samplesync-infra/lib/webhook-stack.ts 41行目で environment: { SHOP_TOKENS: JSON.stringify(tokens) }。tokens はCDKのデプロイ時に .env.production から読み込まれ、同ファイルは .gitignore 済みだがCIのアーティファクトとして90日保持されている（CI-05で確認）。Prowler awslambda_function_no_secrets_in_variables がFAIL

影響：GetFunctionConfiguration を実行できるIAM主体（開発者3名、CIのデプロイロール、FB-SMPL-004のロールは該当しない）が、全ストアのAdmin APIトークンを即時に取得できる。CIのアーティファクトからも取得できる。Secrets Manager 利用時に残るアクセスログ（CloudTrailの GetSecretValue）が残らないため、流出の検知もできない

再現条件：Lambdaの読み取り権限。動的な再現は行っていない

推奨修正：ストアごとのトークンはPrismaのSessionテーブル（保存時暗号化済み）に一元化し、環境変数から削除する。外部APIキーはSecrets Managerに移し、関数から GetSecretValue で取得する（実行ロールに当該シークレットのみの secretsmanager:GetSecretValue を付与）。CIアーティファクトの .env.production は即時削除し、保持期間を見直す。既存のトークンとキーはローテーションする

関連要件：AWS Lambda environment variable encryption（データベース認証情報にはSecrets Managerの使用を推奨）／Prowler awslambda_function_no_secrets_in_variables／Shopify PCD Level 1「保存時の暗号化」の運用面／Shopify App security checks committed-secret の趣旨／ASVS 5.0.0 13.3.1、13.3.2

修正担当（顧客記入）：

期限（合意）：

再確認結果：未実施

検査範囲の注記：アカウント内のLambda関数6件の環境変数キー名をすべて確認した。他の関数にはシークレットに該当するキー名はなかった。値の内容は確認していない

付録

付録A：対象APIエンドポイント一覧と認証手段

ルート	メソッド	認証手段	確認した行／設定	備考
/app	GET	セッショントークン (authenticate.admin)	app/routes/app.tsx 12	
/app/settings	GET, POST	同上	app/routes/app.settings.tsx 9	
/app/jobs	GET	同上	app/routes/app.jobs.tsx 11	

ルート	メソッド	認証手段	確認した行／設定	備考
/app/jobs/:id	GET	同上	app/routes/app.jobs.id.tsx10 session.shopで絞り込みあり /app/mappings GET, POST 同上 app/routes/app.mappings.tsx9 /app/mappings/:id DELETE 同上 app/routes/app.mappings.id.tsx 8	session.shop で絞り込みあり
/api/sync/status	GET	独自検証 (verifySessionToken)	app/lib/auth.server.ts 31	FB-SMPL-001
/api/sync/history	GET	同上	同上	FB-SMPL-001
/api/sync/run	POST	同上	同上	FB-SMPL-001
/api/sync/jobs/:id	GET	セッショントークン (authenticate.admin)	app/routes/api.sync.jobs.\$id.ts 9	FB-SMPL-003
/webhooks/orders/create	POST	Webhook HMAC (独自)	functions/webhook/src/verify.ts 42	FB-SMPL-002
/webhooks/app/uninstalled	POST	同上	同上	個別動作確認なし
/webhooks/compliance	POST	同上	同上	3トピック共通。削除処理の実装を確認済み
/healthz	GET	なし	functions/api/src/index.ts 7	固定文字列応答のみ

付録B：チェックリスト実施記録

(本サンプルでは省略。実際の報告書では4つのチェックリストの判定欄をすべて転記する)

付録C：誤検知として除外した候補の件数

ツール	候補数	指摘に採用	誤検知・情報
Shopify CLI app security check	7	3	4 (テストコード内の検出、テンプレート由来の既知パターン)
Semgrep	23	1	22 (テストコード、到達不能なコードパス、ルールの過検出)
gitleaks	2	0	2 (サンプル値、失効済みテストキー。顧客に確認済み)
Prowler	41 FAIL	2	39 (タグ付け、範囲外サービス、設計判断として受容済みの項目)

ツール	候補数	指摘に採用	誤検知・情報
ZAP ベースライン	5	0	5（情報レベルのヘッダー指摘。CSP 等は設定済みで誤検出）

付録D：用語集

（技術報告書テンプレートの付録Dと同じ）

付録E：参照した一次情報

文書	URL	確認日
Shopify Set up session tokens	https://shopify.dev/docs/apps/build/authentication-authorization/session-tokens/set-up-session-tokens	2026-11-20
Shopify Subscribe to webhooks (HTTPS)	https://shopify.dev/docs/apps/build/webhooks/subscribe/https	2026-11-20
Shopify Webhooks overview	https://shopify.dev/docs/apps/build/webhooks	2026-11-20
Shopify App security checks	https://shopify.dev/docs/apps/build/security/app-security-checks	2026-11-20
Shopify Protected customer data	https://shopify.dev/docs/apps/launch/protected-customer-data	2026-11-20
AWS Lambda environment variable encryption	https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars-encryption.html	2026-11-20
AWS IAM best practices	https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html	2026-11-20
AWS Security Hub controls reference	https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-controls-reference.html	2026-11-20
OWASP ASVS 5.0.0	https://github.com/OWASP/ASVS （タグ v5.0.0_release）	2026-11-20
OWASP API Security Top 10 2023	https://owasp.org/API-Security/editions/2023/en/0x11-t10/	2026-11-20

付録F：HTTPログ・CLI出力（抜粋）

（本サンプルでは省略。実際の報告書では、個人情報・秘密情報をマスクしたリクエスト／レスポンスとCLI 出力を添付する）

免責・取扱い

- 本報告書は、記載の検査期間・検査範囲・手法で確認した結果に基づきます。記載のない項目は確認していません。対象システムに脆弱性が存在しないことを保証するものではありません
- 本報告書の内容は、検査時点の状態に基づきます。その後の変更により結果は変わりえます

- 本報告書は法令・ガイドライン・認証への適合を保証するものではなく、Shopify App Store の審査結果を予測・保証するものでもありません
- 脆弱性の修正、インシデント対応、関係機関への届出はサンプル商事株式会社様の責任において行われます
- 本報告書はサンプル商事株式会社様に対してのみ提供します。第三者への開示は、契約の定めに従ってください
- 証拠データは契約で定めた期間保管し、期間満了後に破棄します

株式会社Forest Book

〒220-0004 神奈川県横浜市西区北幸二丁目10番48号 むつみビル3階

info@forest-book.jp