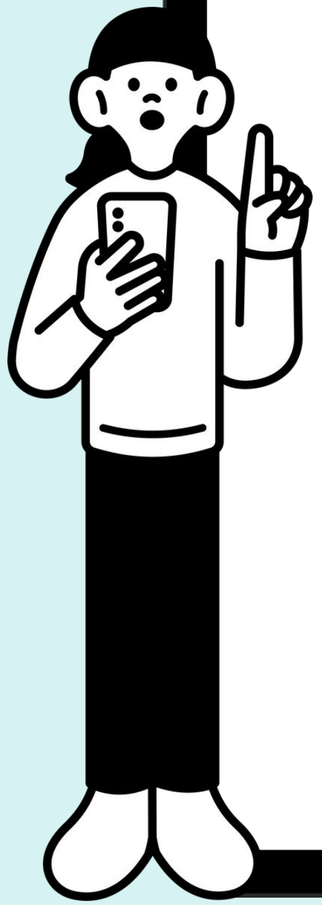
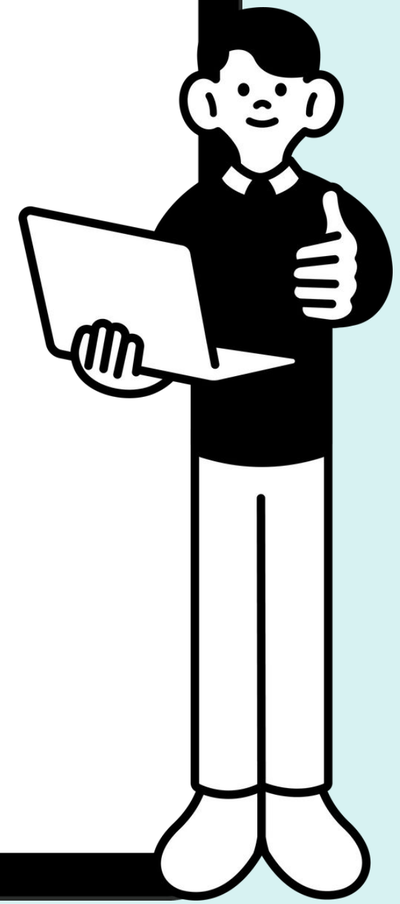
ICSCoE8期 卒業プロジェクト製作

ローコード・ノーコードツール セキュリティビギナーズガイド



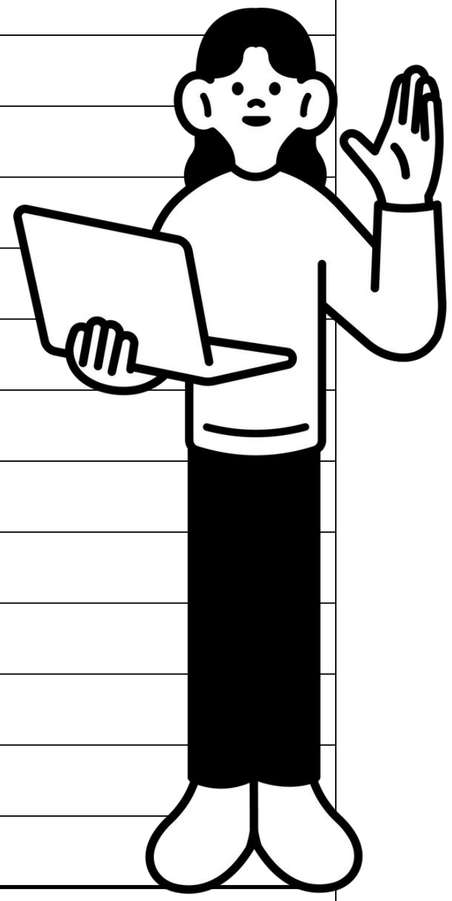


DX Security Committee

Created With Chat-GPT

DXセキュリティ対策委員会

目次



本ガイドについて

概要・目的

本ガイドは、ローコードツールおよびノーコードツールを利用される皆さまが、安全に開発および運用を進めるために必要なセキュリティの基本知識と、実践的な対策を学べるように作成されています。
専門的な知識をお持ちでない方でも理解できるように、最低限知っておくべきリスクやベストプラクティスについて、わかりやすく解説※しています。

想定対象者

- ローコード・ノーコードツールを用いて業務アプリケーションの開発・運用に携わっている方
- DX推進や業務改善プロジェクトに従事されている方
- ローコード・ノーコード開発のセキュリティに関心をお持ちの方

活用例

- チームメンバーとのセキュリティ認識共有のための研修資料や説明補助として活用
- アプリケーション設計・開発時のチェックリストとして活用
- IT部門やセキュリティ部門との連携時に、対話のベースラインとして活用

※本ガイドは、特定のツール製品に関する脆弱性等を取り上げる様なものではなく、ツール利用に際してのセキュリティ注意事項を解説している資料となる

Session 1

セキュリティについて

基本知識

01 セキュリティってなに？

セキュリティとは、情報やシステムを不正侵入などの外部脅威や内部のミスによる情報漏えいから守り、**「安心して使い続けられる状態」**を保つことを指します。
たとえば、顧客の個人情報や会社の機密データが、 **勝手に見られたり書き換えられたり、あるいは消えてしまったりしないようにする** ことも、セキュリティの一部です。

02 なぜセキュリティが重要か

現代の社会やビジネスでは、情報こそが価値そのものであり、同時に　**最も狙われやすい資産** でもあります。
もし、あなたの職場で使っているデータやシステムが、外部から不正にアクセスされたり、操作ミスで誤って消えてしまったりしたら、 **業務の継続が難しくなる** かもしれません。

※本ガイドでは、情報セキュリティをセキュリティとして定める

セキュリティの 3原則とは

機密性

（Confidentiality）

情報に関して、アクセスを認められた者だけがこれにアクセスできる状態 を確保することをいいます。

機密性が守られなかった場合、漏えいした情報が不正利用され、組織の信頼が損なわれるおそれがあります。

完全性

（Integrity）

情報が破壊、改ざん又は消去されていない状態を確保することをいいます。

完全性が保たれない場合、改ざんされた情報に基づいた判断や処理が行われ、業務の正確性や公平性が失われる 可能性があります。

可用性

（Availability）

情報へのアクセスを認められた者が、必要時に、アクセスできる状態 を確保することをいいます。

可用性が確保されていない場合、必要なときにシステムや情報が利用できず、業務の停滞やサービス停止 といった深刻な影響を及ぼします。

※参考：政府機関の情報セキュリティ対策のための統一基準（第4版）解説書

Session 2

ローコード・ノーコードツールとは

定義

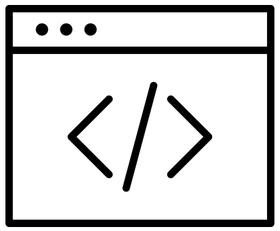
ローコード・ノーコード（ Low-Code / No-Code）ツールとは、プログラミングの専門知識がなくても、業務アプリケーションを構築できる開発ツールを指します。

ローコード（ Low-Code）

最小限のコード記述 で構築可能

ローコードとは、最小限のプログラミング知識 があれば、視覚的な操作を中心にアプリケーションや Webサイトなどの開発をすることができる手法・ツールです。

コーディングの余地がある ため、カスタマイズすることができ、一定の拡張性を保持している点が特徴です。



ノーコード（ No-Code）

コード記述不要 で構築可能

ノーコードとは、一切コードを書かず に開発をすることができる手法・ツールです。

多くはパーツ（テンプレート・ UIコンポーネントなど）を、配置するだけで作成することができ、ITスキルのない業務担当者 でも比較的容易に利用できる点が特徴です。

Session 3

セキュリティリスク・対策

ローコード・ノーコードセキュリティ8か条

01. ログイン情報の管理

ログインに関するID・パスワード情報に注意！

02. アプリへのアクセス

作成したアプリへのアクセス権に注意！

03. データベースへのアクセス

アプリ上で管理するデータへのアクセス権に注意！

04. 追加プラグイン

怪しいプラグインの追加に注意！

05. 外部連携（API連携）

API等による外部アプリケーションとの接続に注意！

06. ブラックボックス化（資産管理の失敗）

設計や内部処理の「ブラックボックス化」に注意！

07. ログの取得

問題が発生した場合に備えたログの管理に注意！

08. ツール独自の仕様理解

ツール特有の落とし穴に注意！

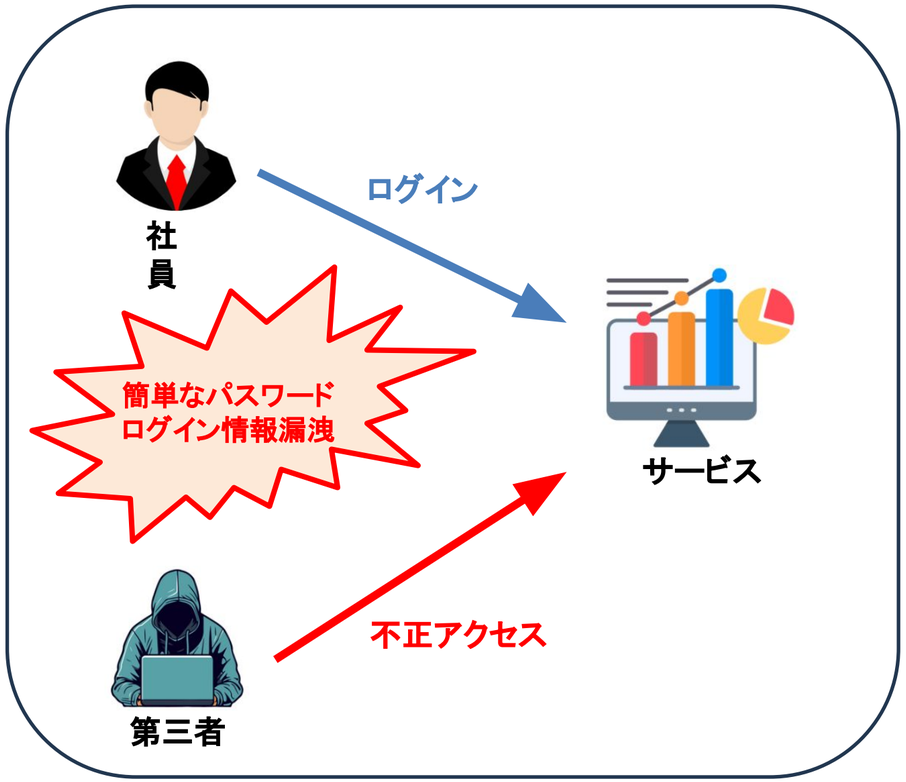
開発者　管理者　利用者

01 ログイン情報の管理

ローコード・ノーコードツールのサービス管理画面等に不正アクセスされないために、
ログイン情報の管理 に気を付ける必要があります。

セキュリティリスク

共有アカウント のログイン情報が漏えいしたり、 **推測されやすい簡単なパスワード** を使用していたりすると、第三者によって不正にログインされ、**重要なデータが盗まれる危険性** があります。特に、管理者アカウントのログイン情報については、厳重な管理が求められます。



対策例

共有アカウント等の運用 を廃止し、適切な権限設定含めた管理 / 運用ルールを定めましょう。
また、ログイン ID・パスワードだけでなく、**メール認証** なども使用する**多要素認証を全ユーザーで必須** にするなどの対策があります。

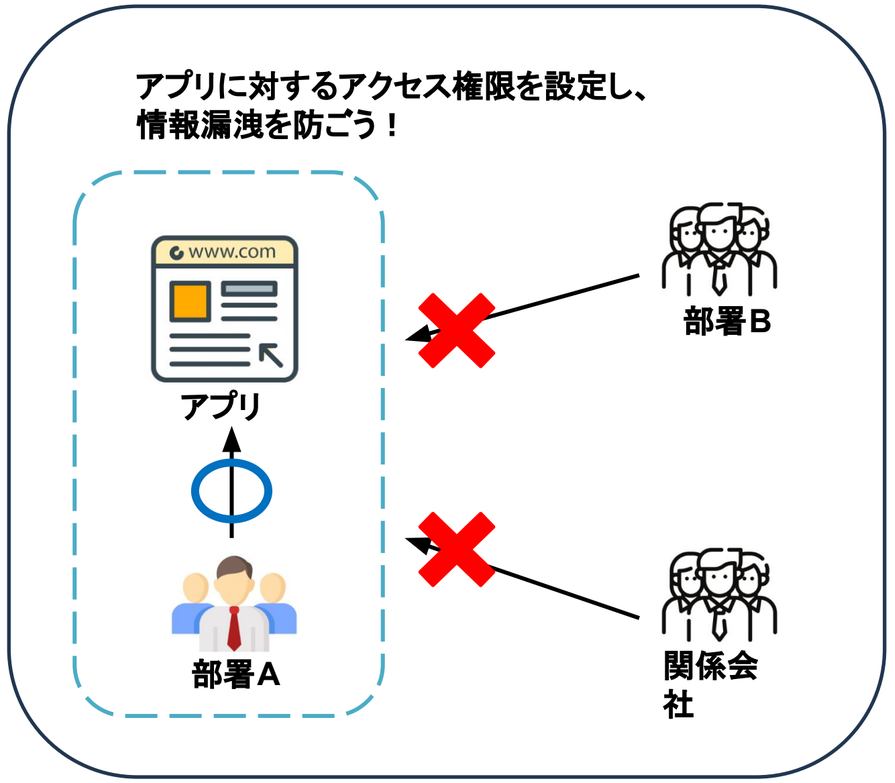
開発者　管理者

アプリへのアクセス

「誰がどのアプリにアクセスしていいか」をきちんと設定していないと、
部外者に知られてはいけない情報を知られてしまうかもしれません。

セキュリティリスク

アプリが「誰でもアクセス可」の状態で公開されていると、部外者に個人情報や機密情報を不正に取得・悪用される危険があります。また、業務上アクセス不要なアプリを開放したままにしておくと、誤操作や内部不正を契機として情報が流出するおそれも否定できません。



対策例

アプリを作成したら、アプリへのアクセス権を必ず設定しましょう。また、アプリ内で外部の人に漏れてはいけない重要な情報を扱っているかの確認や、年一回程度アクセス権の棚卸をするようにしましょう。
適切なアクセス権限設定含め、アプリでの個人情報取り扱いについては、管理・運用ルールを定義することが重要と言えます。

03

開発者　管理者

データベースへのアクセス

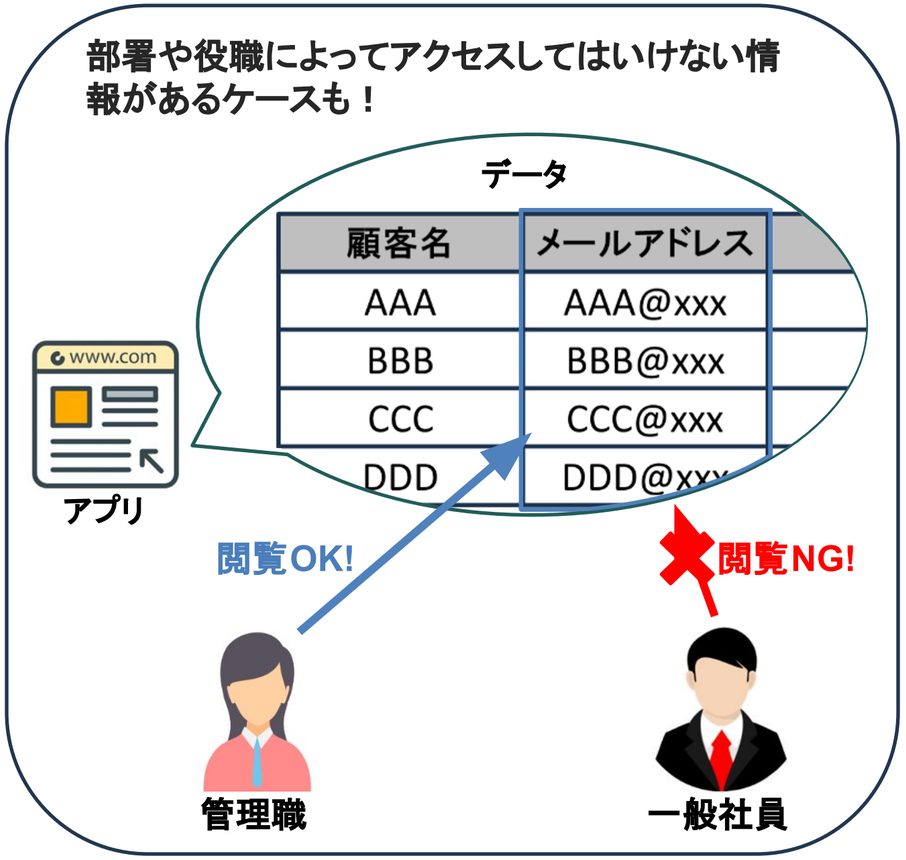
データベースへのアクセスは、**アプリの中のデータを閲覧・操作する権限**で、設定ミスは重大な機密情報の一括流出に直結します。

セキュリティリスク

CSV 出力や一括ダウンロードを誰でも実行できる設定のままにすると、顧客情報や内部メモが丸ごと抜き取られ、メール添付や共有クラウド経由で社外へ流出しかねません。

対策例

レコード閲覧・出力ができる権限をもつユーザーを限定し、データのCSV出力は承認制にするなどの対策が考えられます。また定期的な棚卸しで余分な権限を削除し、ダウンロード操作をログ取得・保管するなどしましょう。



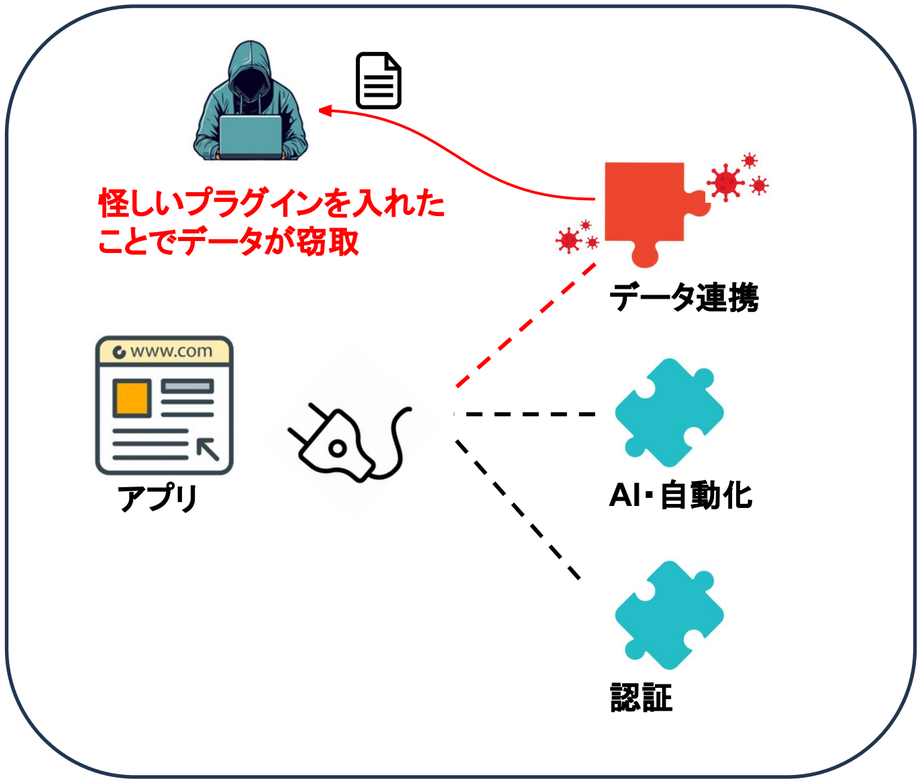
開発者　管理者

追加プラグイン

ローコード／ノーコード開発では、機能を拡張するために「プラグイン」と呼ばれる追加部品を使うことがあります。ドラッグ＆ドロップで簡単に導入でき、便利さが増す一方で、外部の開発者が提供することも多く、**安全性には注意**が必要です。

セキュリティリスク

安全性の低いプラグインを使うと、攻撃者に**データが窃取されたり、システムが不正に操作**される危険があります。
また、必要以上の権限を持つプラグインや、更新されずに脆弱性が残っているものもあります。



対策例

信頼できるプラグイン（公式ストア提供、更新頻度が高い、レビュー評価が安定しているもの等）を選ぶのが安心です。だけを使い、**必要最小限の権限**で動かすようにします。
更新が止まっていないか**定期的に確認**し、古いものは削除や差し替えを検討します。
導入は管理者の許可制にし、**利用統制**することも重要です。

開発者　管理者

外部連携（API連携）

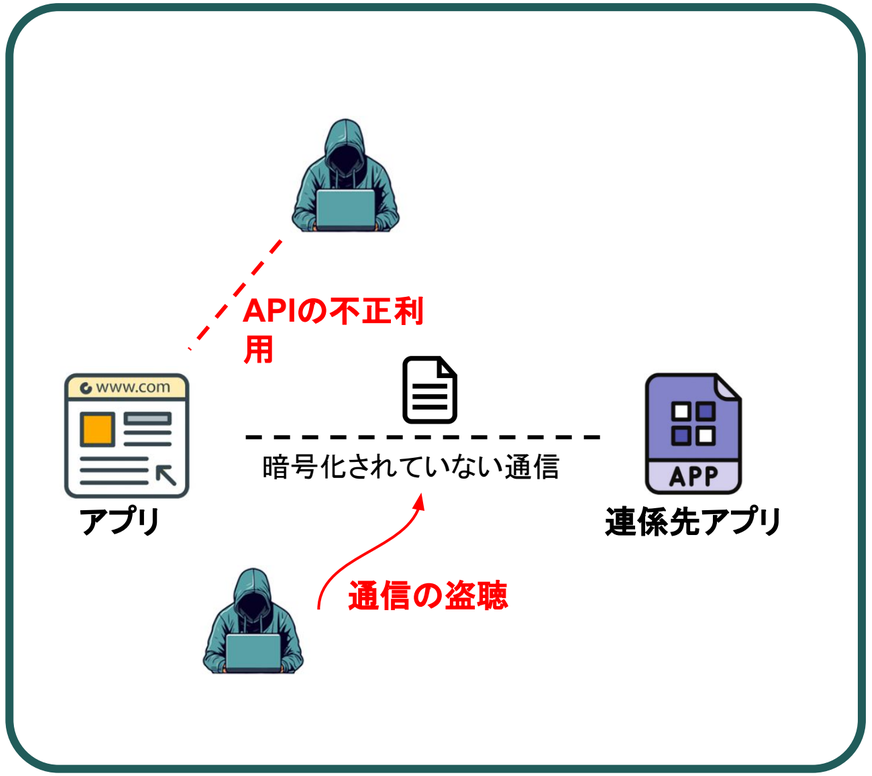
ローコード／ノーコードでは、他のサービスやシステムとデータをやり取りするために「　API」を使うことがよくあります。API連携を使うと、外部との接続が自動化されて便利になりますが、 セキュリティ設定を誤ると大きなリスクになります。

セキュリティリスク

認証が弱かったり、通信が暗号化されていない場合、 第三者にデータを盗まれる 可能性があります。
APIキーが漏れると、悪意のある相手に APIを不正利用 される危険もあります。

対策例

API連携には必ず認証を設定し、通信は暗号化（ HTTPS）で行います。
APIキーは外部に見えないように保管し、 定期的に更新 します。
外部に渡す情報は必要なものだけに絞り、入力データは必ず　事前に確認・フィルタ することが大切です。
また、APIを利用できる人や範囲 をきちんと決め（認可）、利用回数の制限（レート制限）をかけることも重要です。



※API：Application Programming Interface

開発者　管理者

ブラックボックス化（資産管理の失敗）

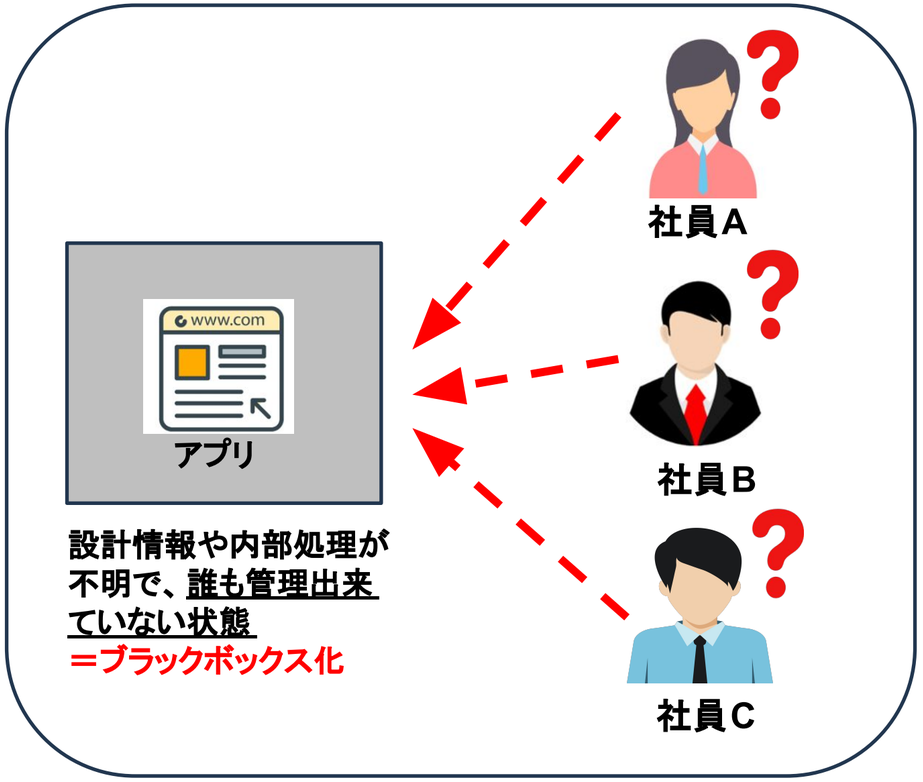
ローコード／ノーコードツールは開発を効率化し、スピードアップが図れる反面、 **設計や内部処理がブラックボックス化しやすい** という特有のリスクを伴います。

セキュリティリスク

作成者しか設計情報が分からないアプリが乱立することで、 **内部処理や脆弱性の把握が困難** になり、ユーザー側のセキュリティ管理も甘くなりがちです。
また、**作ったアプリが放置されたり管理されない** と、外部から不正アクセスされる危険があります。誰が何を作ったか分からないと、**情報漏えいや法令違反に繋がる** こともあります。

対策例

機能や処理内容をドキュメント化 し、見えにくい部分を可視化することが重要です。関係者間でナレッジを共有し、定期的な教育を通じて **設定ミスや属人化を防ぎましょう** 。
また、会社のルールや IT部門と連携して、**アプリへのアクセス権や保存先をしっかり管理** することも大切です。



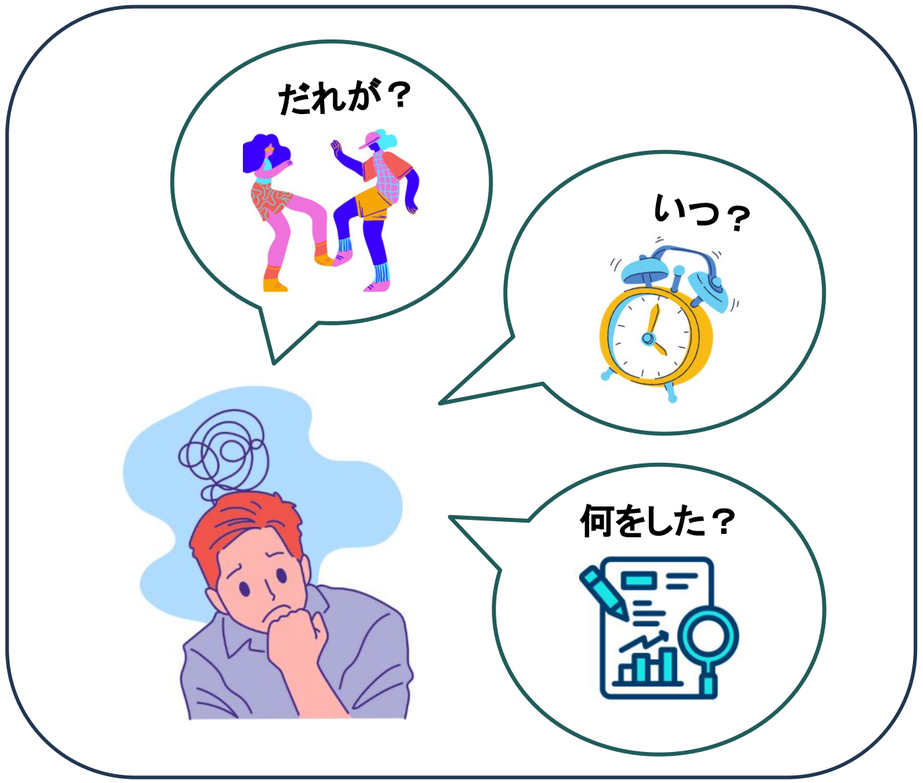
管理者

ログの取得

アプリ上での操作を「誰が」「いつ」「何をしたか」等を見えるようにする のがログ監査です。もしアプリ上でミスやトラブルが起きた時、「記録（ログ）」が残っていないと、原因がわからず困ってしまうケースがあります。

セキュリティリスク

悪意のある操作のあと、ログを消されて証拠が残らない 恐れがあります。記録がすぐに消えたり、バラバラに保存されて見つけにくいことも問題です。その結果、 問題が起きても原因を特定できないリスク があります。



対策例

ログインやデータの変更がきちんと記録されるよう に、ツールの設定を確認しましょう。 ログは管理者だけが見られるようにし、誰にも編集・削除できないようにすることが大切です。これにより、不正操作やトラブルの証拠をしっかり残せます。

08

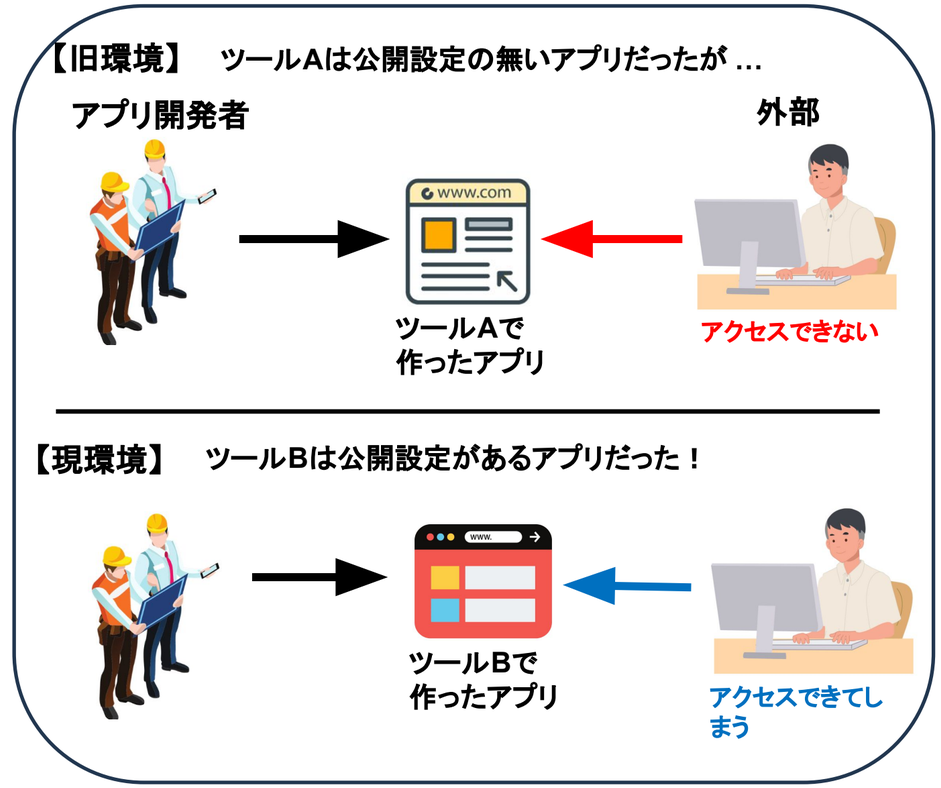
開発者　管理者

ツール独自の仕様理解

ローコード・ノーコードツールごとにある、 **初期値** や **製品（制作ベンダ）特有の設定** を知らない、
または無視したまま使うと、思わぬトラブルにつながることがあります。

セキュリティリスク

ツールごとのルールや仕様を知らずに使うと、 **意図せず情報が外部に公開** されてしまうことがあります。
たとえば、内部のみで公開する想定だったアプリが、実は **誰でもアクセスできる様な設定（初期値）** になっているケースがあります。



対策例

公式のマニュアル等で **ツールの仕様（初期値等）をしっかりと把握** した上で、開発を行うことが重要です。
また、初期導入時だけでなく、 **アプリの構成や更新情報を定期的にチェック** するような運用も重要と言えるでしょう。

開発者　管理者

参考. その他のリスクと対策

関連パート	リスク例	対策例
01.ログイン情報の管理	JavaScript/CSS等をカスタマイズ出来るツールを利用する際に、ログイン情報（認証）やアクセス権（認可）に関わる情報を導入したまま、放置される恐れ。	・認証/認可情報の保存場所検討 ・動的に生成しない等のセキュアコーディングの徹底 ・適用前にコード内容を事前確認
05.外部連携（API連携）	無計画にシークレット情報をコードに記載（ハードコーディング）してしまうことで、以下のようなリスクが考えられる ・コードや画面に直接キーを埋め込むと、共有や公開時に漏洩する危険 ・漏れたキーは第三者に不正利用され、データ改ざんや課金被害が発生 ・修正や失効が遅れると被害が長期化	・環境変数やシークレットマネージャーで管理し、コードから分離 ・プラットフォームの接続管理機能等を利用してキーを秘匿 ・期限付き/最小権限のキーを発行し、定期的にローテーション
05.外部連携（API連携）	外部連携（API）利用時には、他には以下のようなリスクが考えられる。 ・古いライブラリや外部システムに潜む脆弱性を悪用され、API自体が乗っ取られる可能性 ・OAuth 2.0認可コードやAPIトークンが使い回されたり削除されず放置されることで、不正アクセスや横展開を許容 ・定期ローテーションを行わないと、漏えいしたトークンが長時間有効となり、侵入後の被害拡大や永続的な不正利用 ・ユーザー権限が変わったり退職した後もトークンが残存していると、本来アクセスできないデータや機能に触れられてしまう恐れ ・署名検証がない、または不備があると、攻撃者が偽のWebhook通知を送って任意の処理を実行させる恐れ	・依存ライブラリや連携システムとの関係管理 ・OAuth 2.0認可コードフローやAPIトークンの適切な管理（使い回し防止、放置防止） ・重要機能を担うトークンの定期ローテーションとライフサイクル管理プロセス整備 ・不要トークンの速やかな無効化（プラットフォーム管理者の責務として） ・Webhook受信時の署名検証手順と検証失敗時の対応フロー整備
06.ブラックボックス化（資産管理の失敗）	使用されなくなったアプリケーションの廃棄管理について、以下のようなリスクが考えられる。 ・全社のアプリ情報やバージョン情報が攻撃者に取得される恐れ ・脆弱性対応やアクセス制御が行われないまま残存 ・実際に稼働中のアプリが誤分類され停止されることで、業務プロセスが突発的に中断される可能性 ・データベースやバックアップ、ログなどに機密情報が残存し、廃棄後も外部からアクセスされる危険性	・全アプリケーション一覧の自動生成と管理画面表示（APIを活用した定期的な棚卸し） ・乱立するアプリケーションの把握という、これらプラットフォーム特有の課題への対応 ・長期間未更新アプリケーション（例：180日以上）の自動アーカイブまたは停止ワークフロー整備 ・アプリケーション廃棄時のデータ処理 ・廃棄プロセスの監査/証跡管理
ー	その他推奨される対策を、参考までに記載する。	・新機能追加時の既存アプリケーションへの適用検討プロセス ・デフォルト設定が必ずしも安全でないことの明示と、業界標準ベンチマーク（CIS、FedRAMP等）に基づく初期設定テンプレートの提供 ・各プラットフォーム固有のTLS・暗号化設定の最適化 ・シークレット取り扱いやローテーション時の動作確認プロセス ・ API連携のエラー処理を中心とした丁寧な検証 ・権限違反（許可されないフィールドの表示・操作）の防止確認 ・プラットフォーム別セキュリティ機能の定期的な調査と社内対応見直し ・操作ログの可視化・自動分析から、異常操作を即時検知し、通知連携を実施 ・定期的なログレビューとインシデント対応演習で運用体制を強化

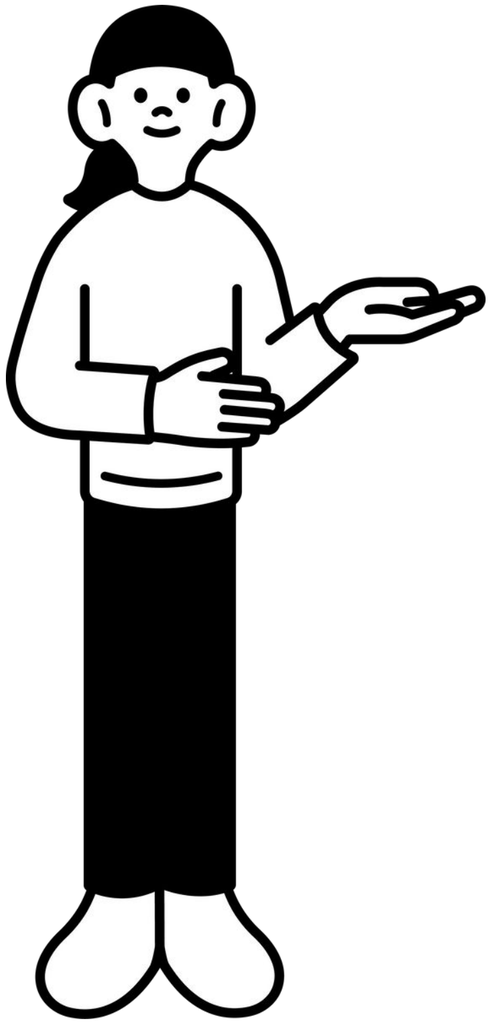
Session 4

チェックリスト

Session 4　チェックリスト

- □ ログイン情報（ID・パスワード）の管理方法を決めましたか？
- □ ログイン時に二要素認証が求められるようになっていますか？
- □ アプリやデータへのアクセス権はちゃんと設定しましたか？
- □ 連携するプラグインの製造元は信頼できるものを選択しましたか？
- □ API連携時の通信は暗号化されていますか？
- □ 作成するアプリの機能や仕組みを共有できるようにしていますか？
- □ 適切なログ取得設定をしましたか？
- □ ローコード・ノーコードアプリごとの仕様を理解していますか？

おわりに



ローコード・ノーコードツールは、専門的な知識がなくても業務アプリをすばやく作れる便利なツールです。
ただ一方で、設定ミスや運用管理の不足が原因で、情報漏えいや不正アクセスなどのリスクが発生する可能性もあります。

このガイドでは、以下のようなポイントを整理してご紹介しました。

- セキュリティの基本「 CIA」（機密性・完全性・可用性）の考え方
- 設定ミス、アクセス制御、外部連携、ログ管理などのよくあるリスクと対策
- 特定の社員だけに依存しない運用
- ツールごとの設定理解の重要性

大切なのは「ツールを安全に使い続ける仕組み 」を作ることです。
誰でも簡単にアプリを作ることが出来る 反面、誰でも簡単にセキュリティのリスクを生み出してしまうことになりやすいという点を忘れないで開発・利用することが重要となります。

その他 参考サイト

「OWASP Low-Code/No-Code Top 10（※英文サイト）」

https://owasp.org/www-project-top-10-low-code-no-code-security-risks

「e-GOV 法令検索：個人情報保護に関する法律」

https://laws.e-gov.go.jp/law/415AC0000000057/

「Kaspersky daily 危険なブラウザ拡張機能（※英文サイト）」

https://www.kaspersky.com/blog/dangerous-browser-extensions-2023/50059/

「Canva（資料フォーマット等）」

https://www.canva.com/

免責事項

・「ローコード・ノーコードツールセキュリティビギナーズガイド」（以下、「本資料」）」に関する著作権及びその他すべての知的所有権は、「情報処理推進機構産業サイバーセキュリティセンター中核人材育成プログラム　8期生～DXセキュリティ対策委員会～（以下、「本プロジェクト」）」に帰属します。

・本資料は、ローコード・ノーコードツールを安全に利用するための一般的な注意事項や対策例を紹介するものであり、特定の製品やサービスの利用を推奨または批判するものではありません。

・本資料に記載の情報は 2025年8月発行時点のものであり、今後のツール仕様変更やセキュリティ動向等により内容が適用できない場合があります。本資料の利用により発生したいかなる損害（直接的・間接的を問わず）についても、作成者および所属団体は一切の責任を負いません。

・本資料に含まれる情報は執筆時点での知見に基づくものであり、完全性・正確性・安全性を保証するものではありません。実際の運用にあたっては、各組織のポリシーや契約条件、法律・規制等を必ずご確認ください。

・本資料中のスクリーンショット、ロゴ、サービス名称等は各権利者に帰属します。本資料の一部または全部を無断で複製、転載、配布することを禁じます。

・本資料の内容は予告なく更新・改訂されることがあります。最新版は　IPA（情報処理推進機構）公式サイト等にてご確認ください。

改訂履歴

2025年 8月 29日 : 初版発行

独立行政法人情報処理推進機構
産業サイバーセキュリティセンター（ICSCoE）第8期生
DXセキュリティ対策委員会