IDC IT SECURITY FORUM 2026 JAPAN ・ 2026.11.11

自律化したのは攻撃だけではない

AI エージェントの統制と能動的検証

Snyk 株式会社 Solutions Engineering
浦底 博幸 Staff Solutions Engineer

snyk

AGENDA

本日の内容

01	いま何が起きているか	CONTEXT
02	社内の AI エージェントが何をしているか、説明できるか	Q1
03	攻撃者より先に、ビジネスロジックの穴を見つけられるか	Q2
04	それを経営に事実として報告できる体制があるか	Q3

PART 1 ・ いま何が起きているか

AIの自律化は攻撃側と社内で同時に進んでいる



攻撃側

攻撃側のAIは
アプリケーションの設計意図を読んで弱点を探す

社内

開発者の端末で動くAIエージェントが
ソースコードや認証情報、社内システムに直接触れている

snyk

PART 1 ・ 国内で起きていること

国内でも直されていない脆弱性が入口になっている

2026 年 9 月から 10 月

相次ぐ情報漏えい

原因として「使っていたソフトウェアの脆弱性」や
「修正前の既知の脆弱性」を突かれたと
公表された事案が続いた

2026 年 10 月 21 日 修正版公開

PHP2Shell

PHP 本体のリモートコード実行の脆弱性 2 件
修正版の公開前から攻撃の試みが観測された
米 CISA が悪用確認済みの脆弱性に登録
AI の支援で見つかった脆弱性でもある

言えること

見つかるのも突かれるのも速くなった
直されないまま残った脆弱性が、そのまま入口になる

snyk

出典：各社の公表資料・報道（2026 年 9 月から 10 月）、PHP Project 、CISA KEV 、Snyk Blog（2026 年 10 月 26 日）

開発者の端末では AI エージェントがすでに社内システムにつながっている

43%

2 つ以上の AI コーディング環境を使っている開発者

50.8%

MCP サーバーを導入している開発者

1/12

MCP サーバーを導入している開発者のうち、High/Critical の指摘があった割合

出典 : Snyk「 Inside the Agentic Development Supply Chain 」（2026 年 6 月 ）
約 1 万の開発環境の匿名化テレメトリに基づく

PART 1 ・ 実際に起きたこと

信頼して使っていた MCP サーバーが ある日から情報を流し始めた

1,643

削除されるまでのダウンロード数

世界で初めて確認された
悪意ある MCP サーバーとされる事例
導入時に一度審査するだけでは防げない

01 **本物そっくりの名前で公開**

公式の MCP サーバーのコードを複製し、同じ名前で npm に公開した

02 **15 回、問題のない更新を重ねる**

普通に動く版を出し続け、利用者の信頼を得た

03 **16 回目の更新で、1 行だけ追加**

送信するすべてのメールを、BCC で外部のアドレスへ転送

04 **AI エージェントが使うたびに流出**

パスワードの再設定、請求書、社内の連絡などが対象に

snyk

出典：Koi Security 、Snyk Blog ほか（2025 年 9 月）

これまでの統制は、人手で回す前提になっている

これまでの統制
年 1 回の診断・ペネトレーションテスト
人手によるコードレビュー
文書で定めた AI 利用のガイドライン

→

AI エージェント時代に必要な統制
変更のたびに続く、自動の攻撃検証
コードが生成された瞬間のチェック
実行時に効き、記録が残るポリシー

PART 2 ・ 社内の AI エージェントを統制する

統制は 3 つの層で効かせる：使うもの、すること、生み出すもの

LAYER 01

使うもの

Agent Supply Chain Security

端末の MCP サーバー・スキルを検出

各コンポーネントのリスクを継続評価

危険な構成をポリシーで指摘

LAYER 02

すること

Agent Behavior Governance

リスクに応じて MCP 呼び出しを遮断

今後： 破壊的な操作の防止

今後： 機微データ・シークレットの保護

今後： プロンプトインジェクション対策

LAYER 03

生み出すもの

Trusted Output Assurance

AI が生成したコードをその場で検査

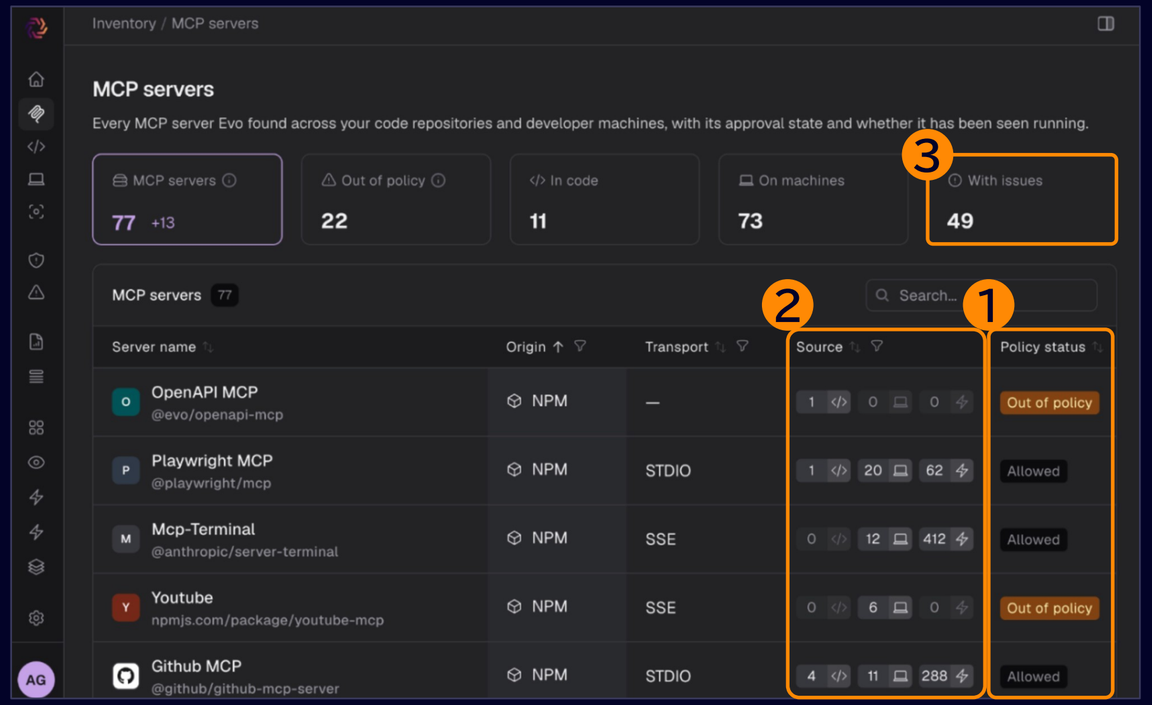
安全な書き方の指示をチームへ配布

フックでエージェントに組み込む

snyk

※2026 年 11 月時点、「今後」は公表済みの予定で変更される可能性があります

許可リストとの差分を一覧にし
外れた呼び出しはその場で止める



MCP servers の一覧（Evo）



開発者側の表示（Claude Code）

画面の見どころ

1. 許可リストに載っているか
2. リポジトリ・端末で見つかった数と実行回数
3. 指摘が見つかったサーバーの数
4. 止めたポリシーとサーバーを表示

PART 2 ・ すること：監査証跡

何が使われ、何を止めたかを記録で示せる

WHICH SERVER

どの MCP サーバーが

コードと端末から
使われているサーバーを一覧に

WHERE

どこで使われているか

コード・端末・実行時の
どこで見つかったか

POLICY

許可されているか

許可リストの内外とリスクの高さ

IMPACT

どの端末に影響するか

遮断した場合に
影響を受ける端末の数

RESPONSE

どう扱ったか

記録か遮断か
止めた呼び出しとサーバー

EXPORT

記録をどう残すか

Evo MCP を通じて
Jira など既存の仕組みへ

snyk

PART 3 ・ 攻撃者より先に見つける



診断は年 1 回
リリースは毎日

平均的なペネトレーションテストの期間は 15 日
残りの約 350 日、変更は攻撃者の視点で検証されないまま本番で動いている

snyk

出典 : Snyk プレスリリース（2026 年 5 月 27 日）

PART 3 ・ 2 種類の脆弱性

パターンで見つかる穴と
設計を知らないと見えない穴

TYPE A

パターンで見つかる穴

XSS や SQL インジェクション
既知の脆弱性を持つライブラリなど
SAST・SCA・DAST が漏れなく検出できる

TYPE B

設計を知らないと見えない穴

「利用者 A が利用者 B の請求書を
閲覧できてはならない」のような認可の判断
これまでは人手のペネトレーションテストに頼ってきた

いま起きていること

攻撃側が AI で設計意図を読み解き、この種の穴を狙い始めている
人手の診断だけでは、頻度も範囲も足りない

PART 3 ・ 能動的な検証

AI が設計意図を読み
攻撃者と同じ手順で検証する

Web アプリケーションとそれが使う API に対して、AI エージェントが偵察から悪用までを自律的に試す
SAST・SCA・DAST がすでに見つけた穴を踏まえ、スキャナーでは届かない欠陥に絞って試す

01

認可の不備

権限の低い利用者が
本来アクセスできない機能や
データに到達できないかを試す

02

他テナントのデータ参照

マルチテナントのサービスで
別の顧客のデータを参照・操作
できないかを試す

03

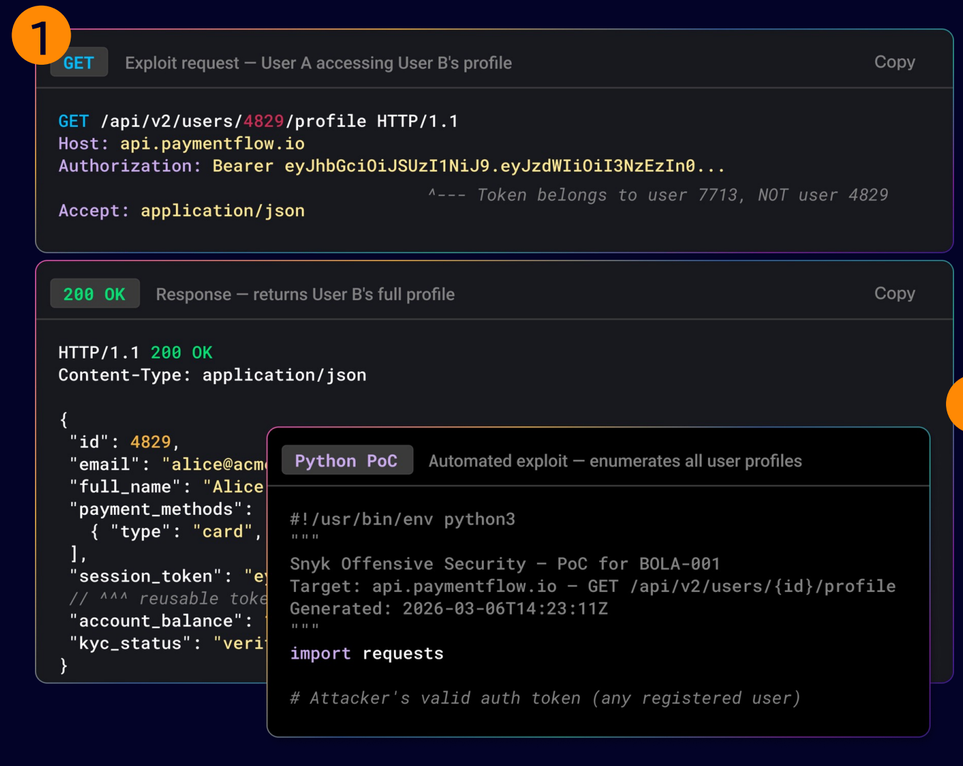
脆弱性の連鎖

軽微な指摘や古いライブラリの
脆弱性を組み合わせ
被害につながる経路かを確かめる

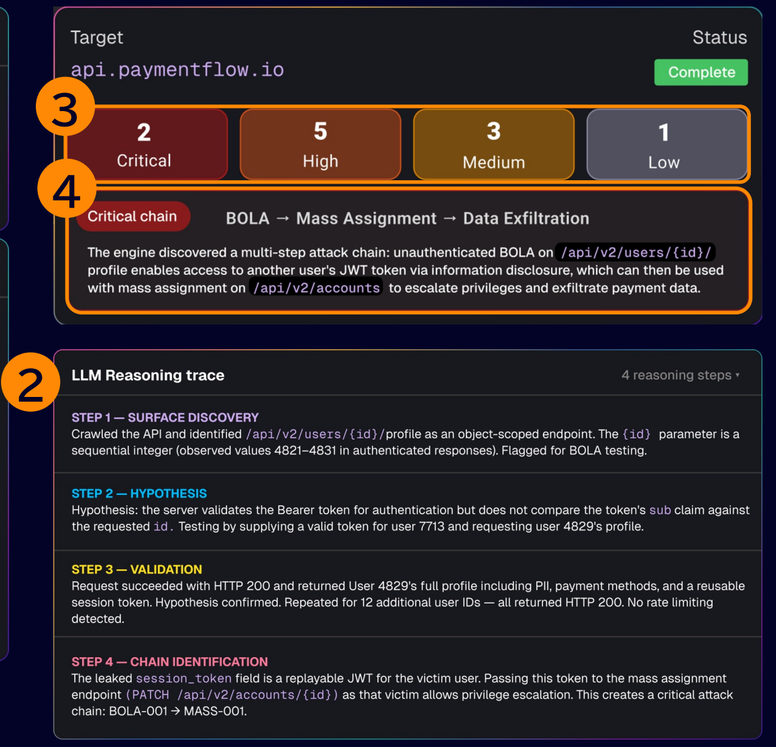
snyk

PART 3 ・ 実際の画面

指摘ごとに、実際に悪用できた証拠が付く「本当に悪用できるか」で優先順位を決められる



例： 利用者 A の認証で、利用者 B のプロフィールが返ってきた



対象ごとの結果と、AI が攻撃に至った推論

画面の見どころ

1. 悪用の実証結果
2. 攻撃に至った手順
3. 深刻度ごとの件数
4. 攻撃の連鎖

snyk

出典：Snyk 製品ページ（Evo Continuous Offensive Security）



PART 3 ・ LLM を組み込んだアプリ

LLM を検出したら
自動でレッドチーミングへ

偵察の段階でアプリに LLM が組み込まれていると分かれば
次の攻撃を多段階で試す

01 プロンプトインジェクション

02 ツールやエージェントの悪用

03 データの持ち出し

snyk

“

セキュリティチームは、単にアラートをさばくのではなく
本当のリスクに優先順位を付けられる手段を求めています
Continuous Offensive Security によって
悪用できる脆弱性と、それらがどう連鎖するかがはっきり見えるようになります

EMBURSE　Colleen Carroll 氏・Senior Director, Information Security Officer

“

常に変わり続けるリスクに、スケジュールを組むことでは追いつけません
必要なのは、理論上あり得るものではなく、本当に悪用できるものに集中できる攻撃テストです

YALO　Gabriel Brolo 氏・Staff Security Engineer

出典：Snyk プレスリリース（2026 年 8 月 4 日、日本語は抄訳）

PART 4 ・ 経営への報告

経営に報告できるのは「証跡」と「証明」

証跡 統制は、何が使われ何を止めたかを記録で示す

証明 検証は、実際に悪用できたことを実証結果で示す

取引先の評価、監査、インシデント時の説明に、推測ではなく事実で答えられる



snyk

PART 4 ・ 導入の順序

小さく始めて、段階的に効かせる

PHASE 01

記録だけで見える化

端末とリポジトリの
AI 利用を棚卸しし
記録だけで実態をつかむ
開発は止まらない

PHASE 02

許可リストを決める

既存の許可リストを
取り込み、リスト外の
利用を洗い出す

PHASE 03

遮断に切り替える

影響を確かめ
高リスクのものから遮断
振る舞いの制御も
段階的に広げる

PHASE 04

重要アプリから継続検証

顧客データを扱う
アプリから検証し
原因の依存関係や
コードまで遡って直す

PART 4 ・ 体制

誰が何を持つか：統制は一つの部門では完結しない

POLICY

セキュリティ部門

ポリシーの設計と
違反・指摘への対応

DEPLOY

IT 部門

端末への配布と
ID と権限の管理

BUILD

開発部門

許可リストへの追加申請と
問題の修正

OWNER

事業部門

守るべきビジネスロジックと
重要アプリの優先度

ASSURANCE

監査・リスク管理

証跡と検証結果で
統制の有効性を確認

DECISION

経営

リスク許容度と
投資の優先順位の判断

統制も
AI の速さで

Q1　社内の AI エージェントが何をしているか、説明できるか

Q2　攻撃者より先に、ビジネスロジックの穴を見つけられるか

Q3　それを経営に事実として報告できる体制があるか

答えに迷う問いがあれば、講演後に展示ブースでお話ししましょう

INFO　11 月 30 日 (月)14:00 から、Evo by Snyk のウェビナーを開催予定

snyk

アンケートはこちら

QR コードから簡単なアンケート (約 1 分) に
ご回答いただいた方に、本日の資料と
IDC ホワイトペーパー (日本語) をお送りします