脆弱性の発見・報告に関する勉強会
～初めての脆弱性調査～

独立行政法人情報処理推進機構（IPA）
セキュリティセンター

2019年9月26日(木)

目次

1. 脆弱性発見手法の学習
2. 脆弱性情報の取扱いについて
3. 脆弱性の評価方法
4. IPAの届出制度を利用した経験談

1. 脆弱性発見手法の学習

独立行政法人情報処理推進機構（IPA）
セキュリティセンター

本プログラムの概要

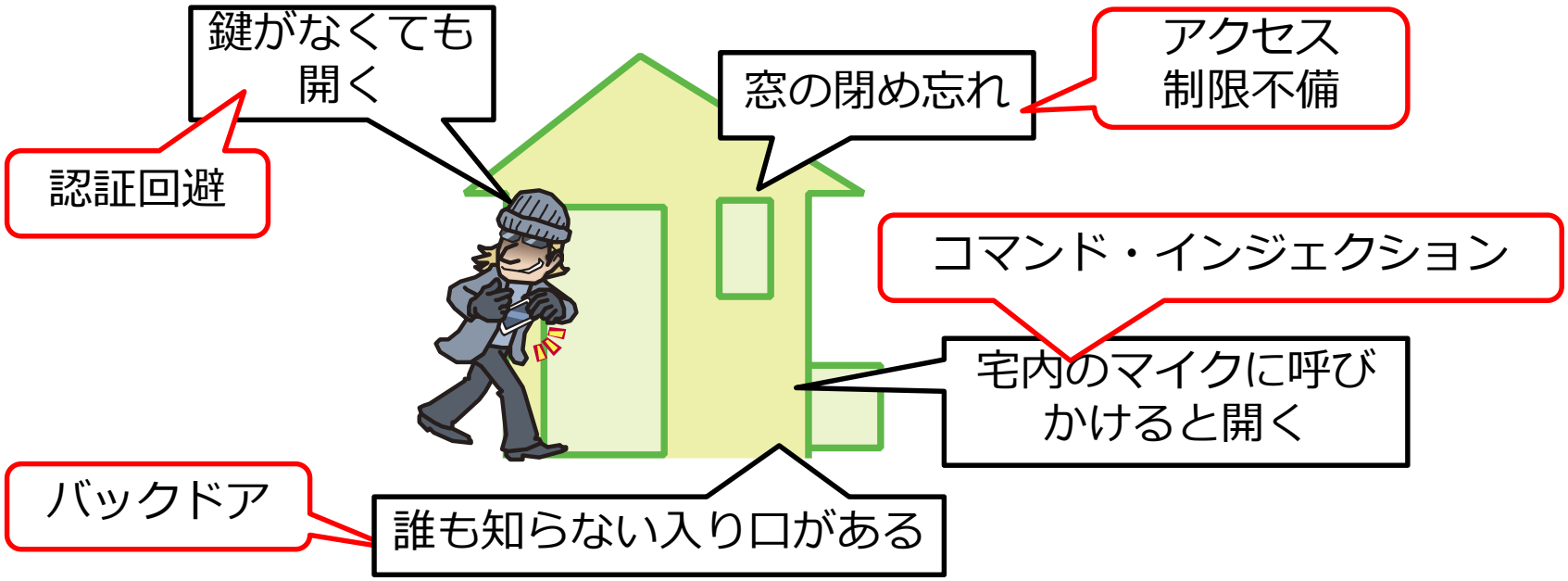
「**ソフトウェア製品**」を対象とした脆弱性の発見手法の学び方を解説し、以下についてについて理解を深める

- 脆弱性の学習方法と検証方法

本プログラムの内容はソフトウェア製品を対象としたものです。解説した内容を決してウェブサイトに対して実施しないでください。

目次

1.1 脆弱性の存在箇所



あらかじめ、どのような攻撃が可能か調査し、
攻撃方法を想定する必要がある

脆弱性を見つけるためには

脆弱性はどこにあるのか？

外部API？

動的表示箇所？

例：検索欄、URLパラメータ 等

どうやったら発見できる？

ソースコードの解読？

手あたり次第に値を入力？

例：<script>alert(“1”)</script>, ‘ 等

ご自身で脆弱性を検証を行い、検証の経験を積むことで脆弱性を発見する目を養いましょう！

どうやって脆弱性の学習や
調査手法の経験を積むの？

1.2 AppGoatとは

⇒脆弱性の概要や対策方法等の脆弱性に関する基礎的な知識を実習形式で学べるツール

AppGoatの機能

どんなことが出来るの？

攻撃者視点

- 脆弱性の内容を学習
- 仮想環境に埋め込まれた脆弱性を攻撃

開発者視点

- 脆弱性の対策方法を学習
- 仮想環境に埋め込まれた脆弱性を修正

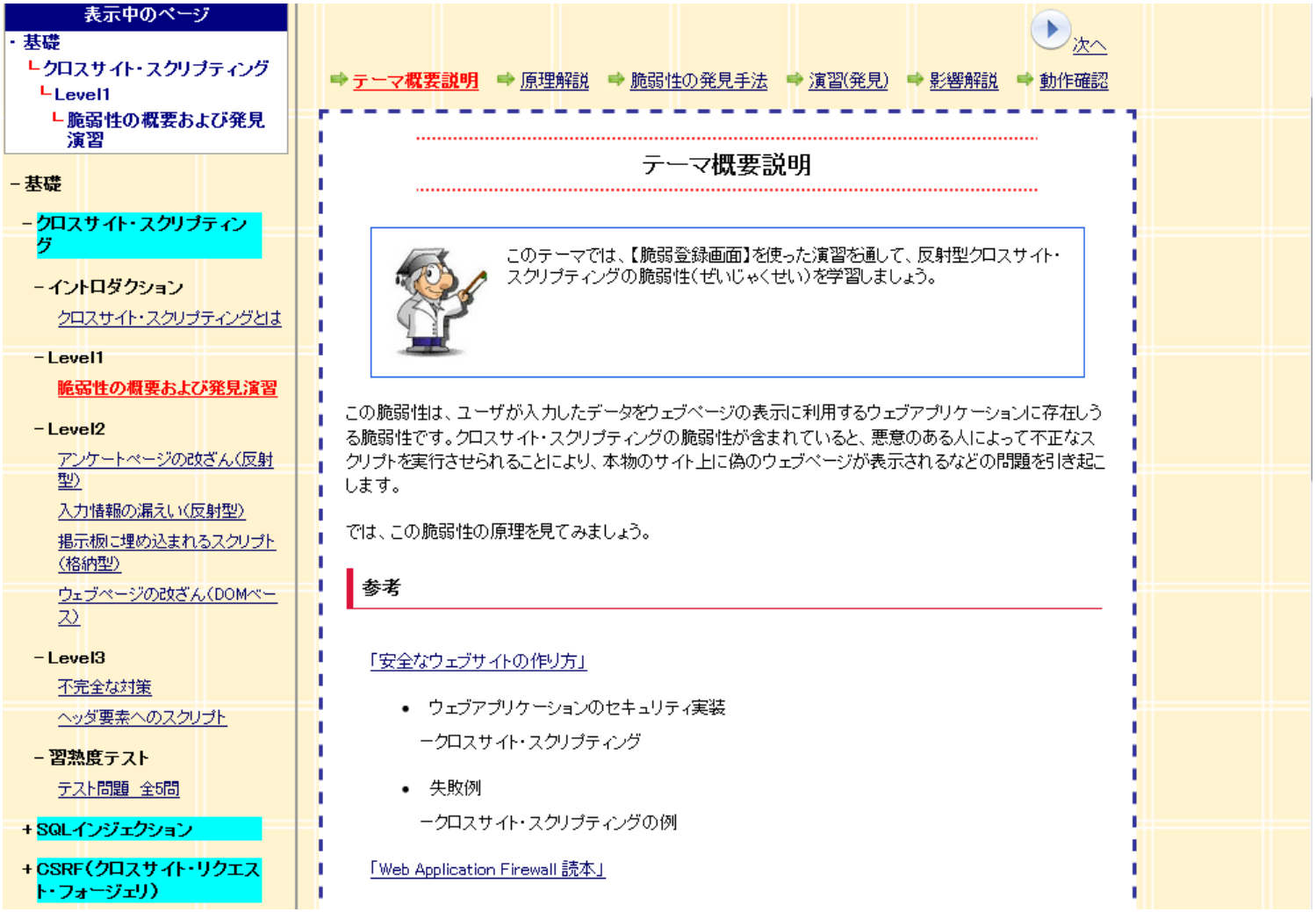
1.2 AppGoatとは
学習テーマ

- 学習できる脆弱性の紹介
  - ウェブアプリケーションに作り込まれやすい脆弱性を
    中心とした12個のテーマを用意

学習できる脆弱性	
クロスサイト・スクリプティング	認証制御や認可制御の欠落
SQLインジェクション	HTTPヘッダ・インジェクション
クロスサイトリクエストフォージェリ	バッファオーバーフロー
ディレクトリ・トラバーサル	クリックジャッキング
OSコマンド・インジェクション	メールヘッダ・インジェクション
セッション管理の不備	その他の脆弱性(システム情報漏えい等)

1.2 AppGoatとは
利用イメージ

- 脆弱性の概要を確認します。

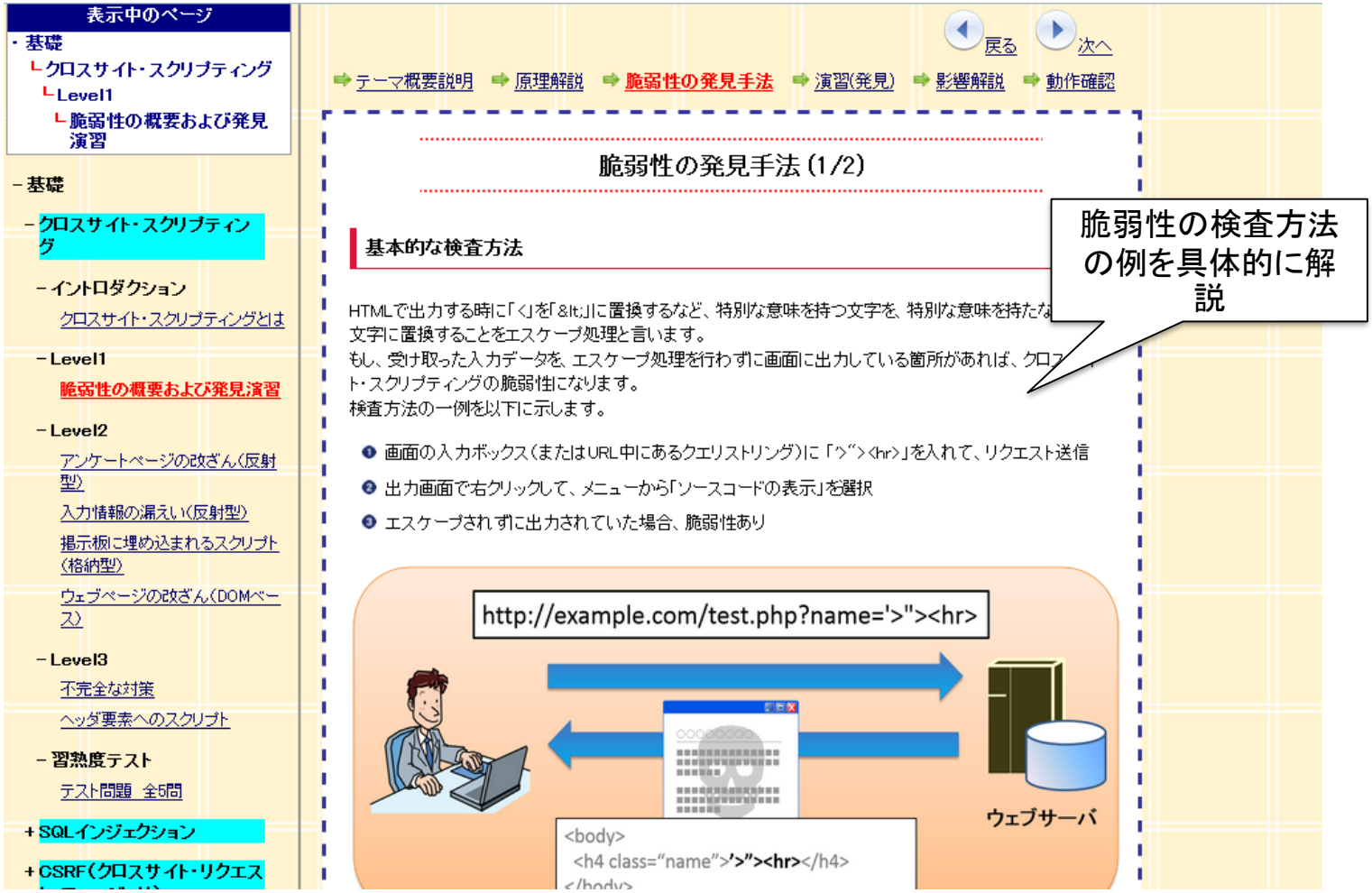


1.2 AppGoatとは
利用イメージ - 原理の解説

- 脆弱性の原理を学習します。

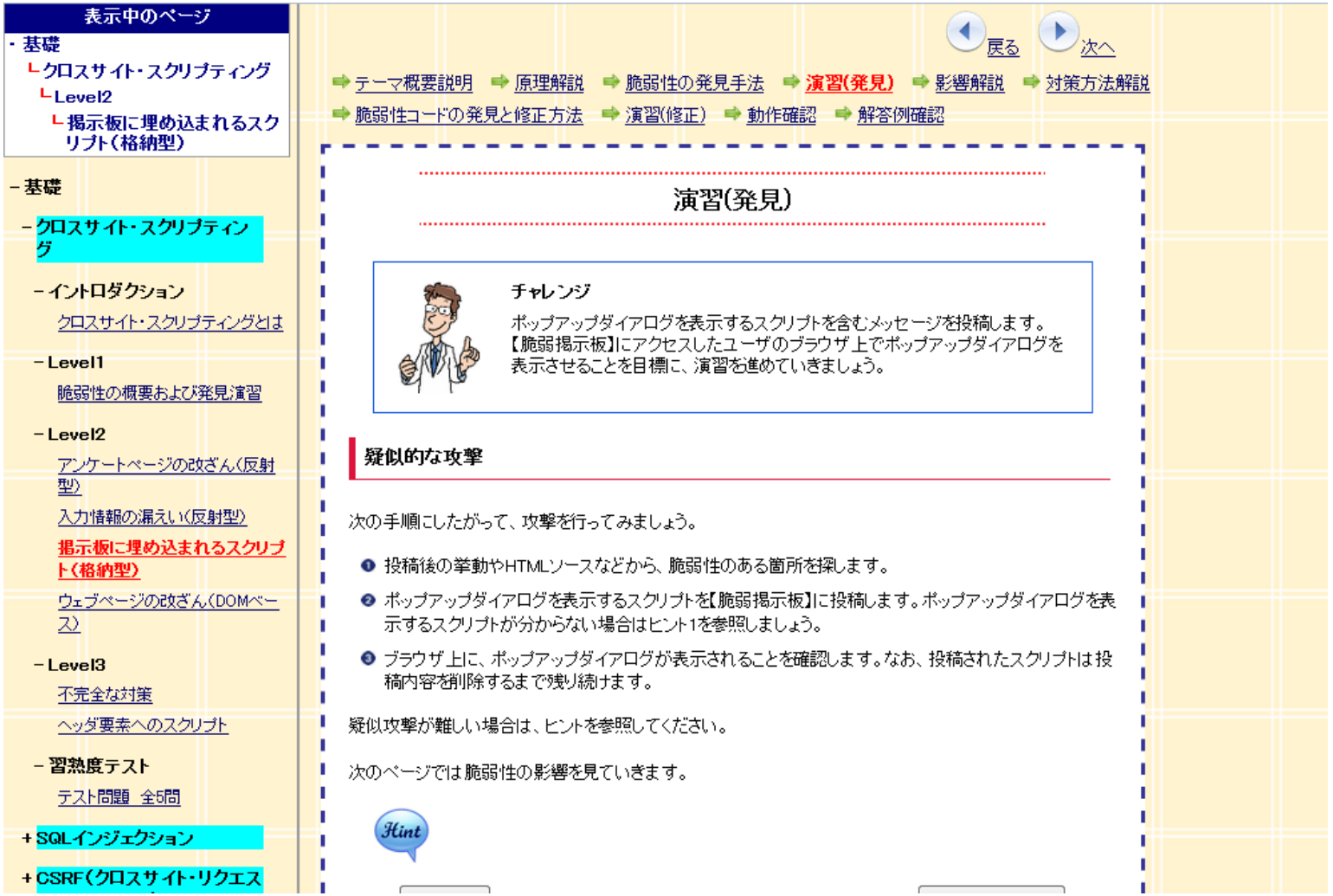
1.2 AppGoatとは
利用イメージ - 発見手法の解説

- 脆弱性の発見手法を学習します。

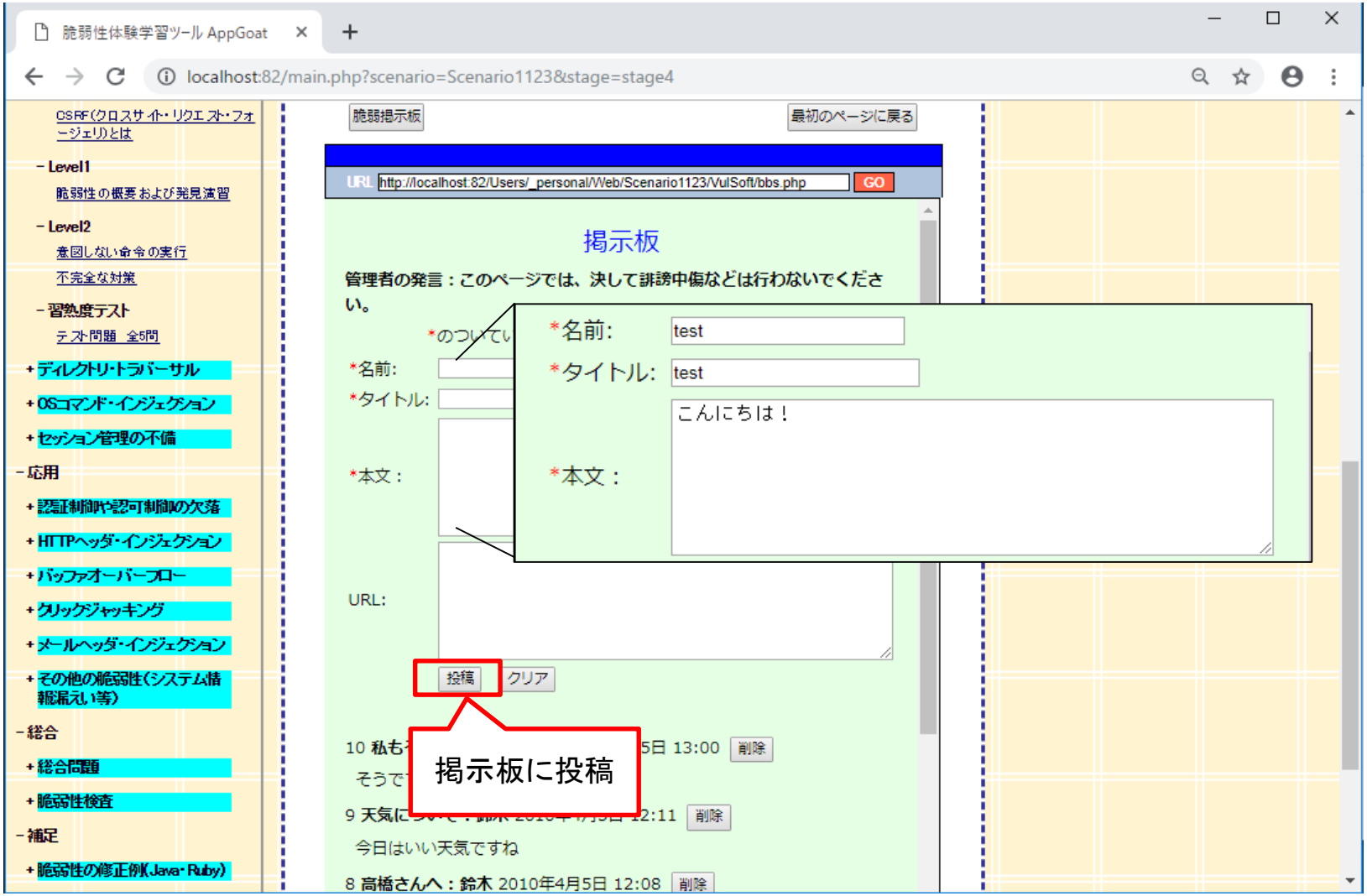


1.2 AppGoatとは
利用イメージ - 攻撃演習

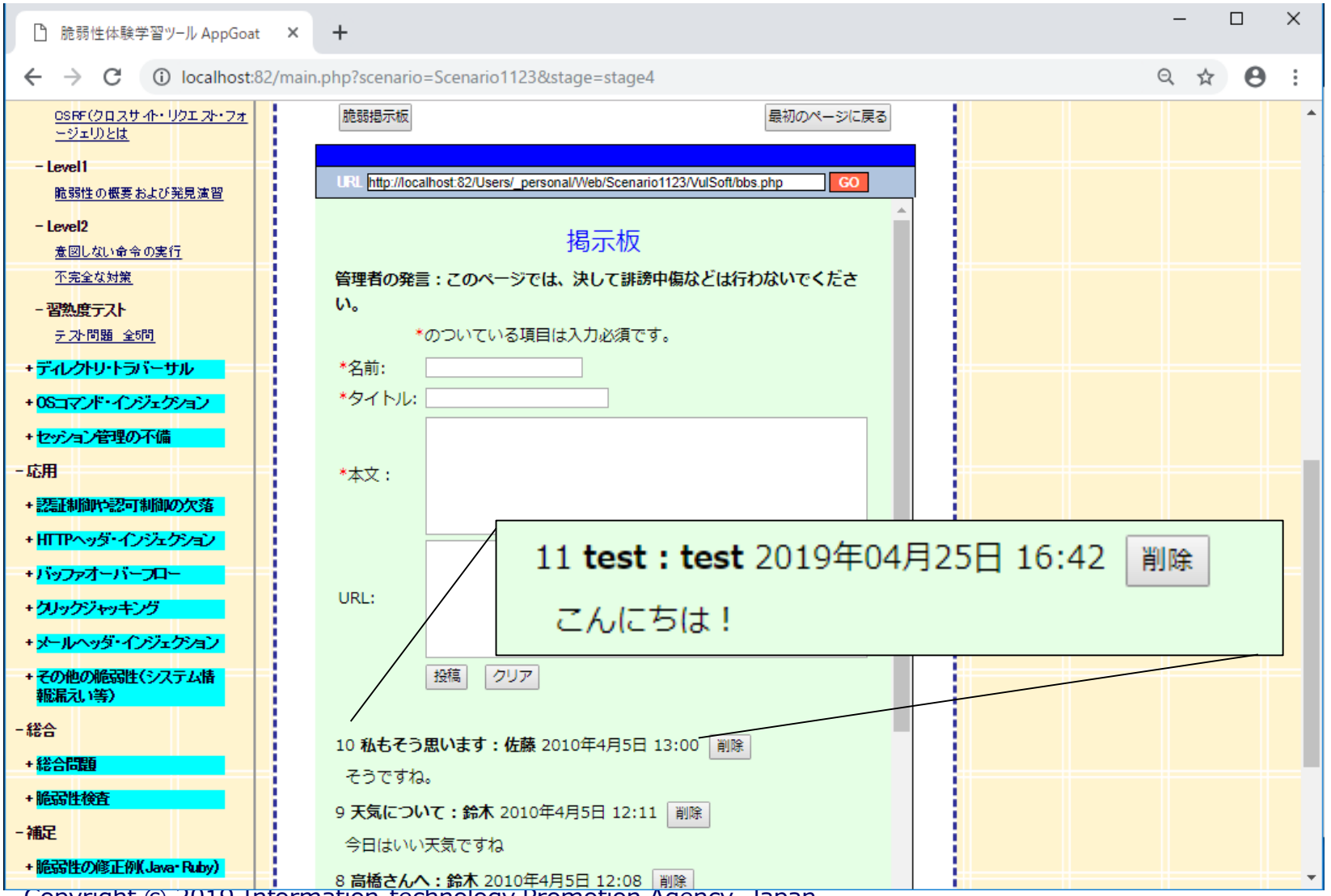
- 攻撃者の立場で脆弱性を攻撃します。



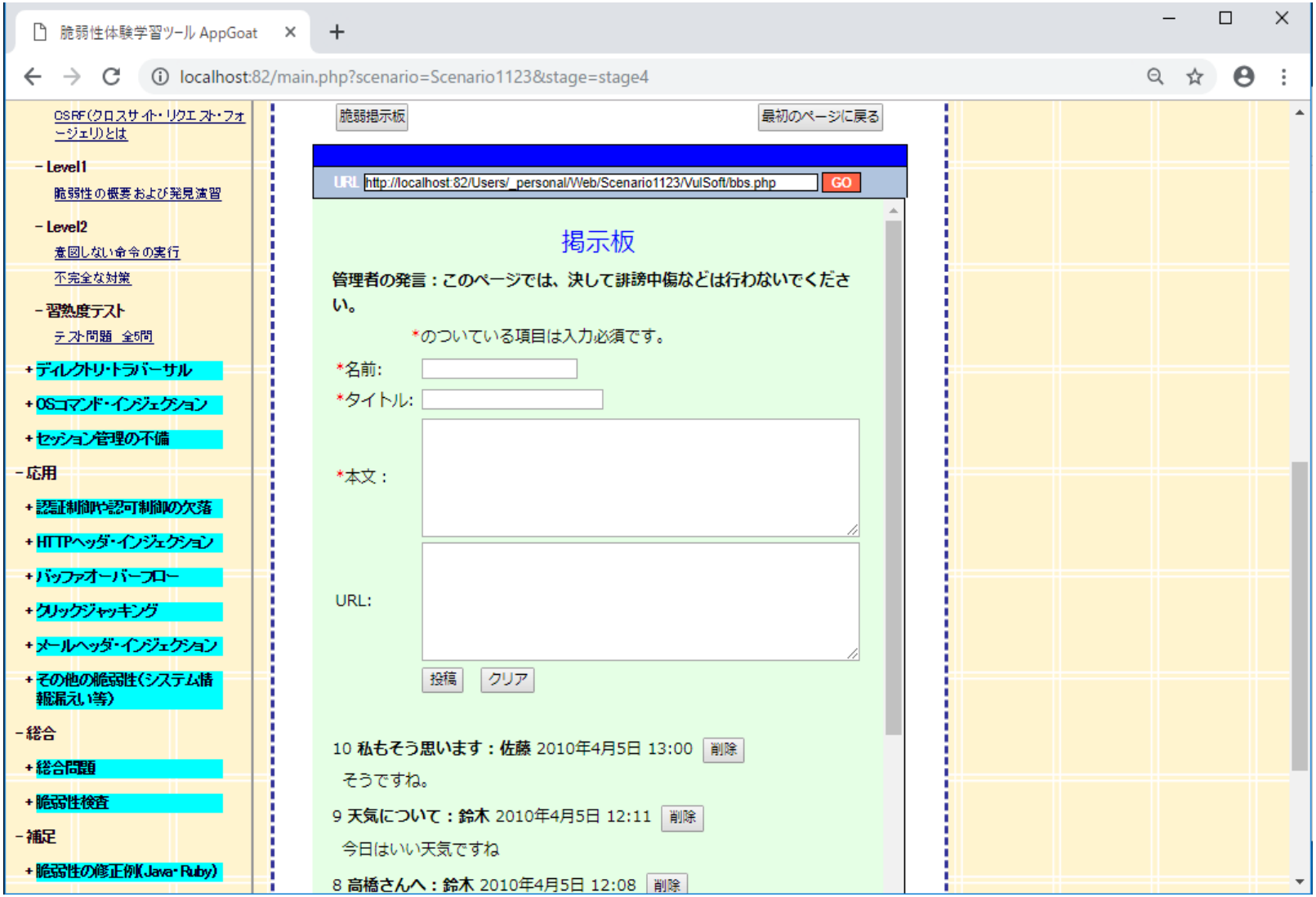
1.2 AppGoatとは
利用イメージ - 攻撃実施



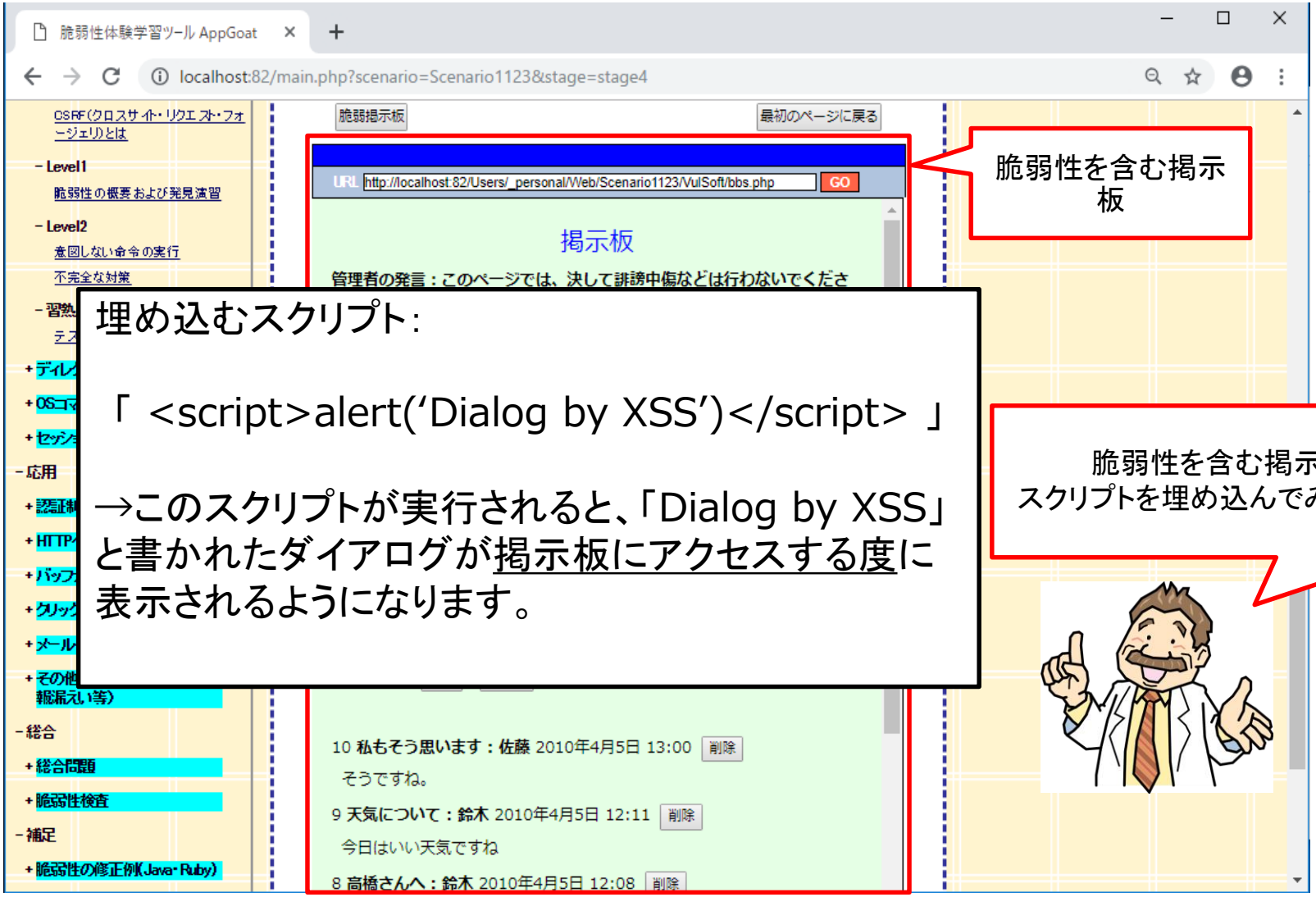
1.2 AppGoatとは
利用イメージ - 脆弱性箇所の発見1



1.2 AppGoatとは
利用イメージ - 脆弱性箇所の発見2



1.2 AppGoatとは
利用イメージ - 脆弱性箇所の発見2



脆弱性を含む掲示板

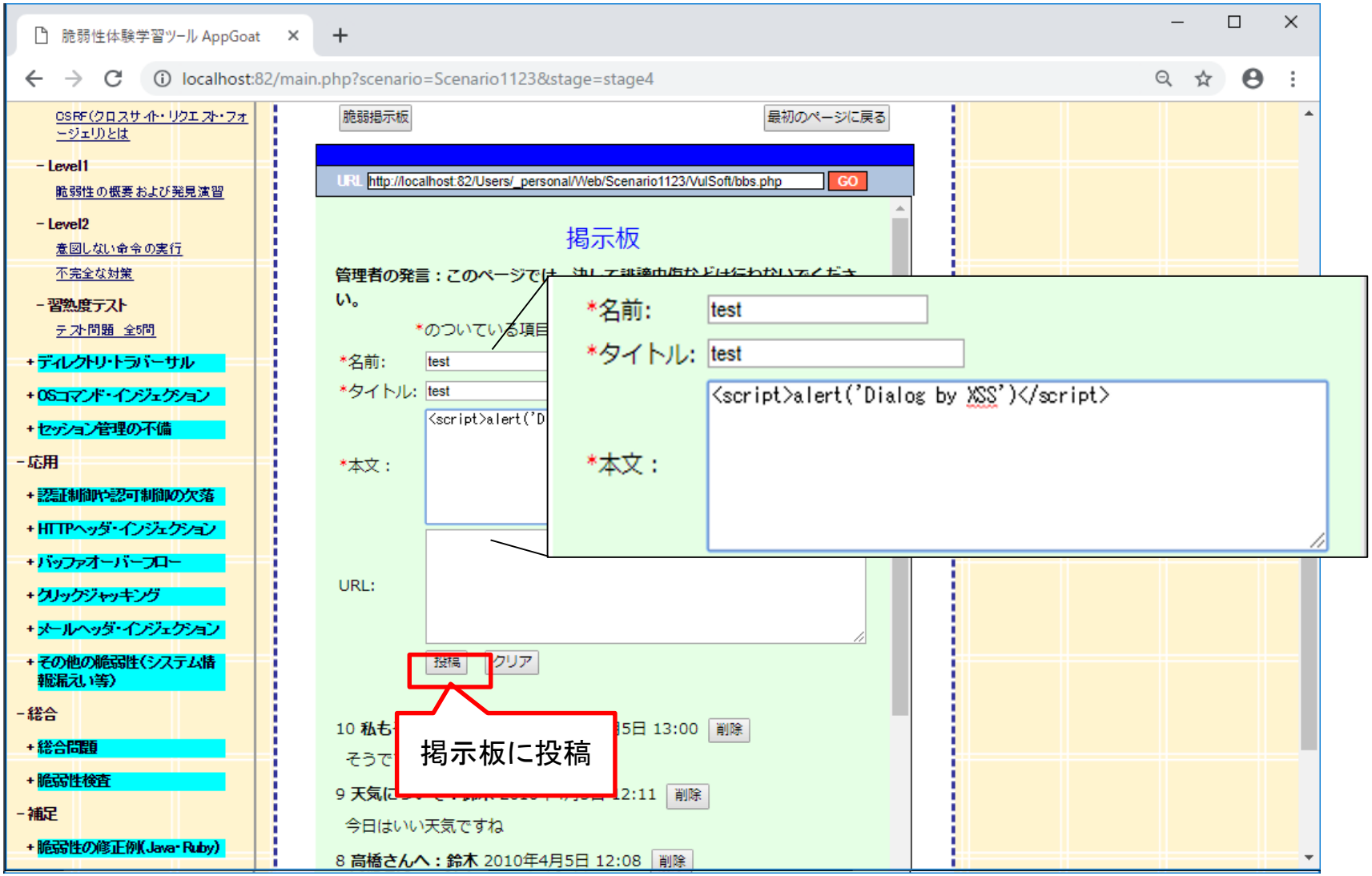
埋め込むスクリプト：

「 <script>alert(‘Dialog by XSS’)</script> 」

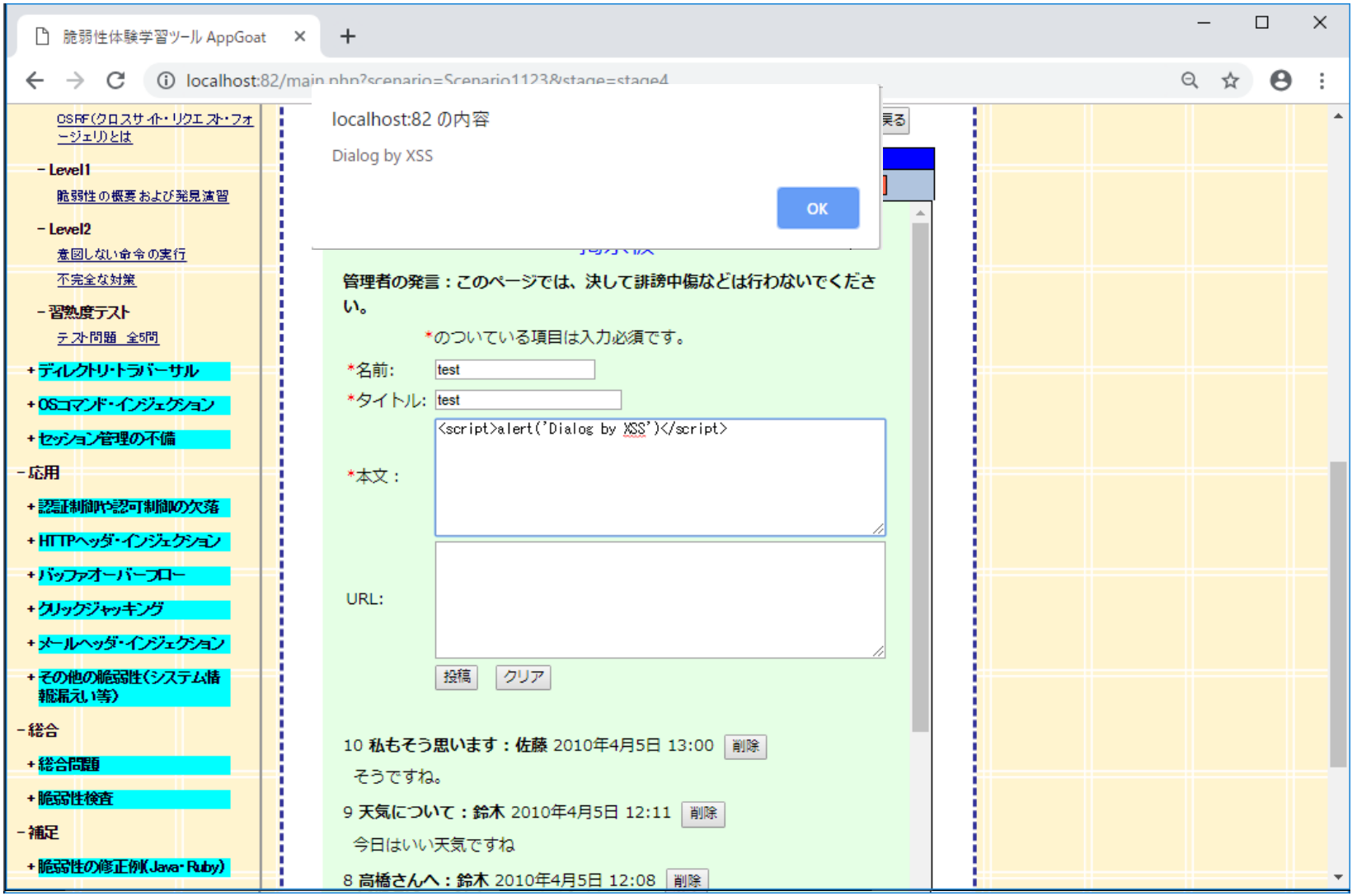
→このスクリプトが実行されると、「Dialog by XSS」と書かれたダイアログが掲示板にアクセスする度に表示されるようになります。

脆弱性を含む掲示板に
スクリプトを埋め込んでみましょう！

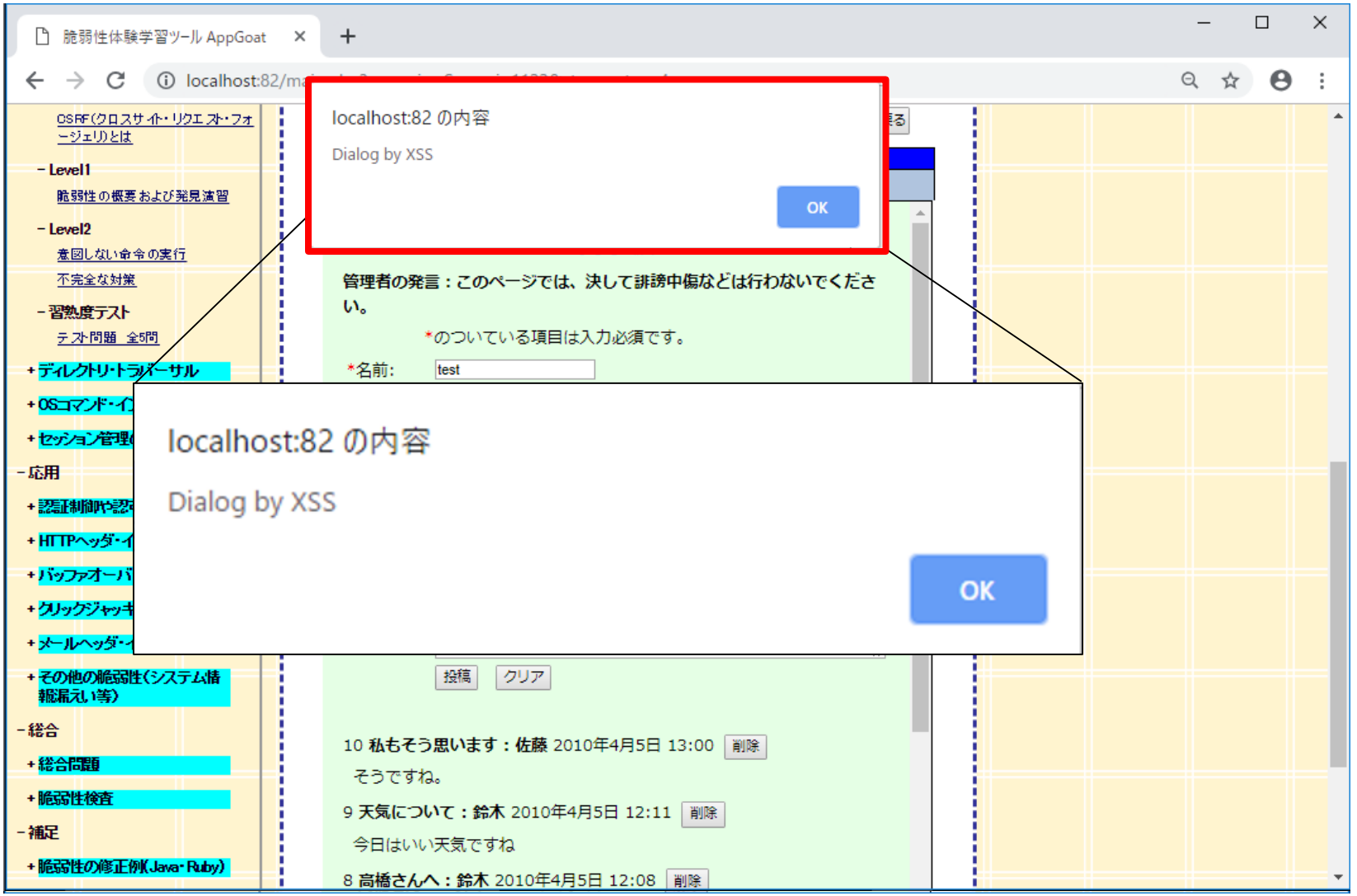
1.2 AppGoatとは
利用イメージ - 脆弱性箇所の発見3



1.2 AppGoatとは
利用イメージ - 脆弱性箇所の発見4

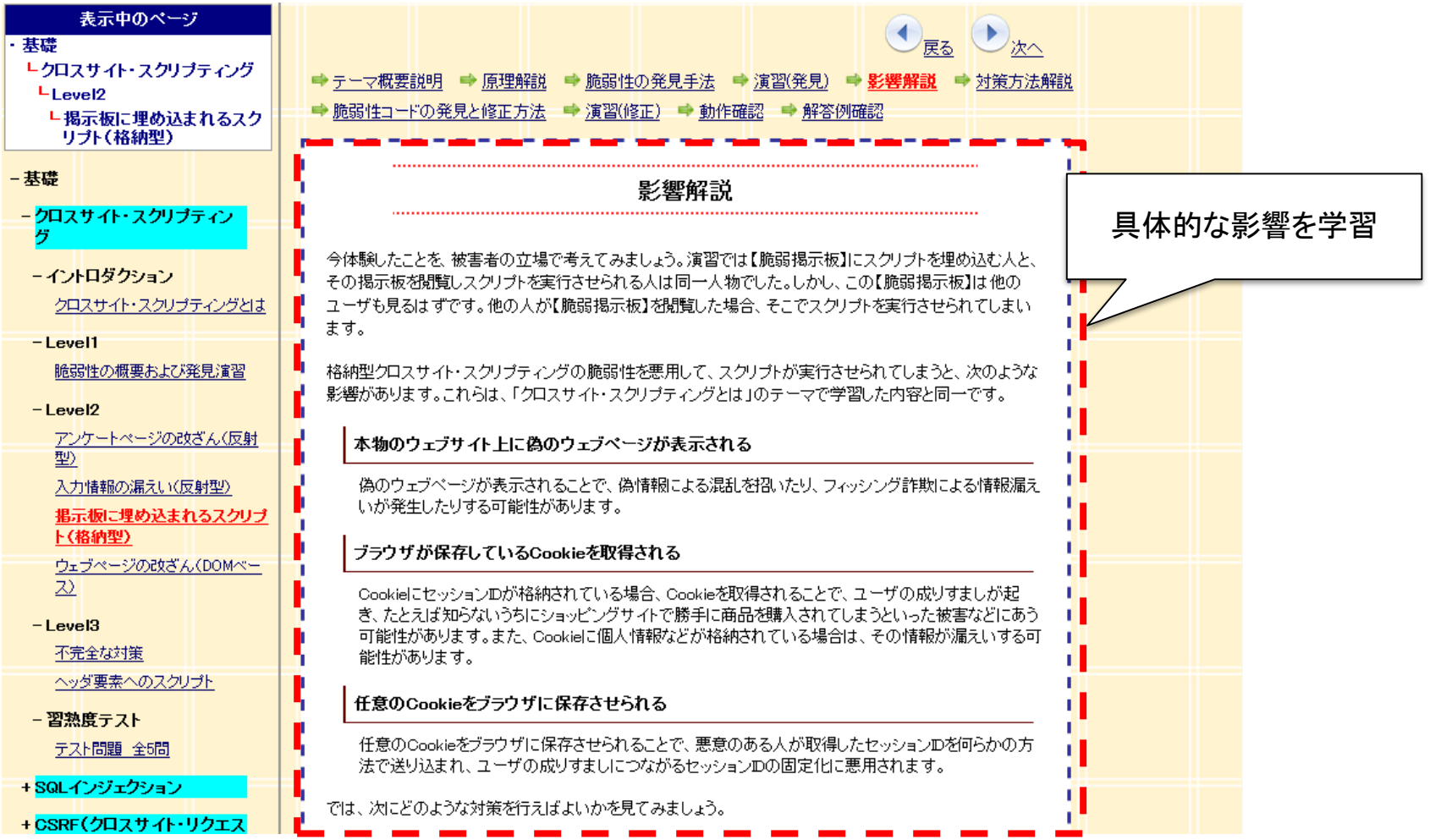


1.2 AppGoatとは
利用イメージ - 脆弱性箇所の発見4



1.2 AppGoatとは
利用イメージ

- 攻撃を受けた場合の影響を把握します。



1.2 AppGoatとは
AppGoatの演習の流れ

STEP.1 攻撃者の立場で脆弱性を体験

テーマ概要説明 → 原理説明 → 脆弱性の発見手法 → 演習(発見) → 影響確認

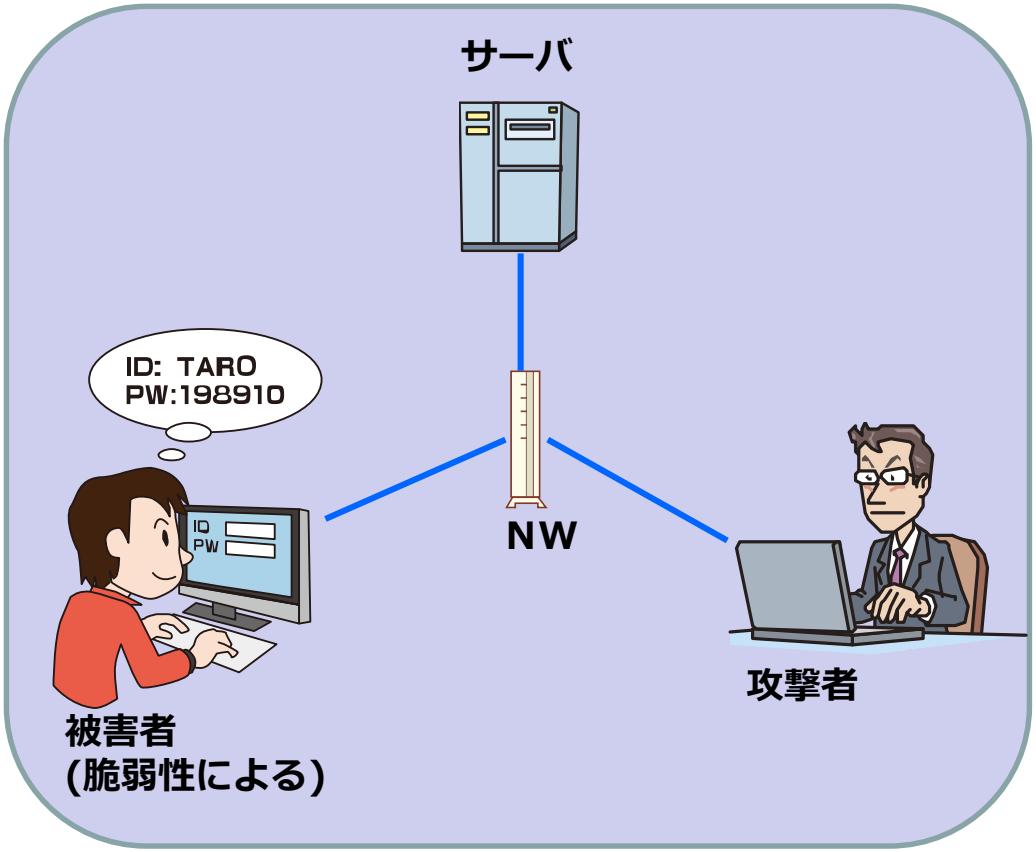
STEP.2 開発者の立場で脆弱性を対策

対策方法解説 → 脆弱性コードの発見と修正方法 → 演習(修正) → 動作確認 → 解答例確認

実際に脆弱性を検証する際
はどんな環境が必要？

1.3 検証環境の構築
検証環境（イメージ図）

- 検証対象以外に影響が生じない環境が必要
- できれば、自由に既定の状態に巻き戻したい



導入ソフトウェア例
〇攻撃者役
・Windows10
・ブラウザ
・パケット解析ツール
…etc

〇サーバ役
・Windows ServerあるいはLinux OS
・XAMPP
　　- PHP
　　- Apache
…etc

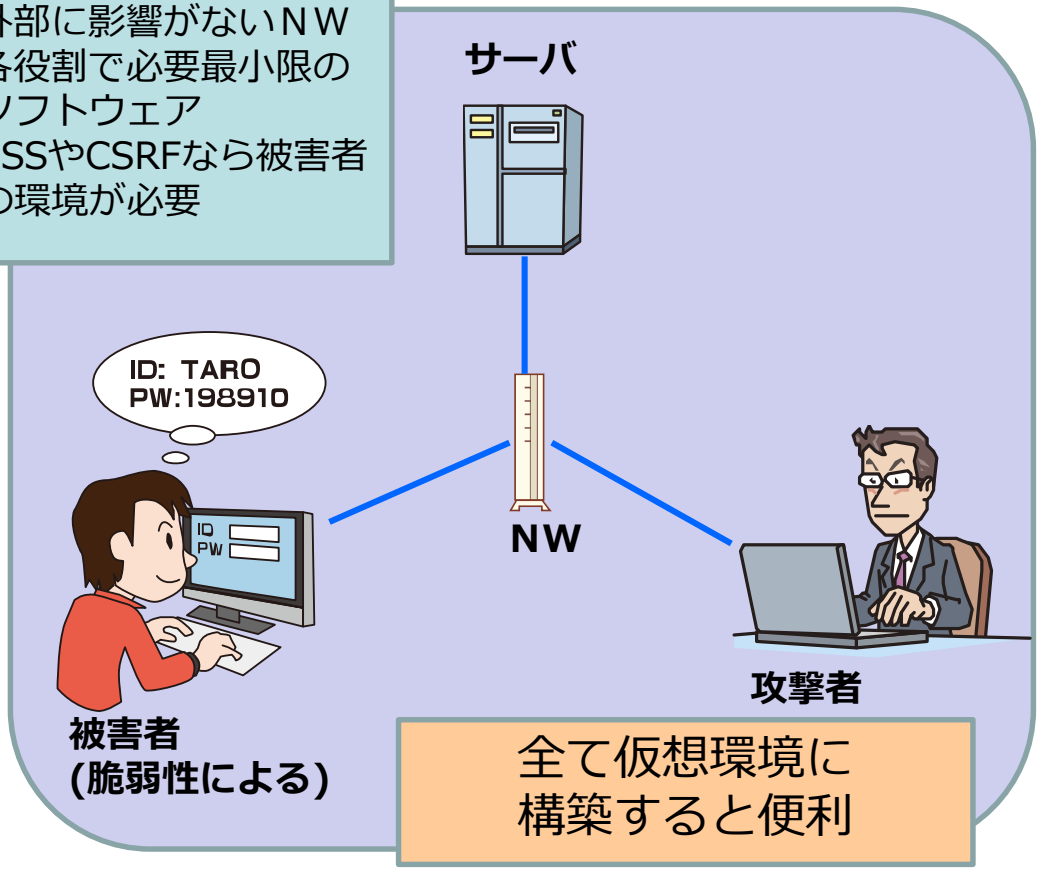
〇被害者役
・Windows10

〇その他
・VM(仮想環境)上にすべてを構築することも可能
例：vSphere、VirtualBoX

1.3 検証環境の構築
検証環境（イメージ図）

- 検証対象以外に影響が生じない環境が必要
- できれば、自由に既定の状態に巻き戻したい

・外部に影響がないＮＷ
・各役割で必要最小限のソフトウェア
・XSSやCSRFなら被害者の環境が必要



全て仮想環境に構築すると便利

導入ソフトウェア例

〇攻撃者役
・Windows10
・ブラウザ
・パケット解析ツール
…etc

〇サーバ役
・Windows Ser…あるいはLinux OS
・XAMPP
　- PHP
　- Apache
…etc

簡易にウェブサーバ環境を構築できる

〇被害者役
・Windows10

〇その他
・VM(仮想環境)上に…くを構築することも可能
例：vSphere、VirtualBoX

・クライアント端末やサーバにインストール
・構成変更が容易

1.4 まとめ

- 脆弱性の発見技術の向上には、脆弱性を**見抜く目を養う**ことが必要
- IPAではAppGoatを公開しており、活用していただくことで、実習形式で脆弱性の発見手法や脆弱性そのものについて効率的に学習することが出来る

- 稼働中のウェブサイトの脆弱性を調査すると、意図しないシステム停止の発生や、不正アクセスとして訴えられるといったリスクがある

自分が管理するシステムやウェブサイト以外に対して、絶対に調査を行わないこと。

これで皆さんも快適な
脆弱性発見ライフを！

2. 脆弱性情報の取扱いについて

独立行政法人情報処理推進機構（IPA）
セキュリティセンター

目次

- 2.1 脆弱性情報とは
- 2.2 脆弱性情報の取扱いの流れについて
  - 協調的に脆弱性情報を取り扱う
- 2.3 発見者の担う役割と注意点
  - 「発見」
    - 発見にかかわる注意点
  - 「報告」
    - 報告にかかわる注意点
- 2.4 脆弱性報告のメリット
- 2.5 まとめ

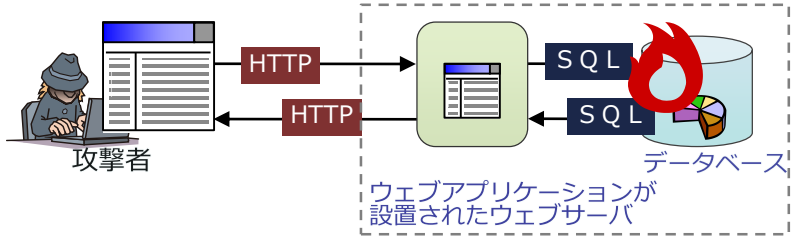
2.1 脆弱性情報とは

- 脆弱性の例

① 攻撃者が、 細工したリクエストを送信する
例：専用のソフトで通信データを直接細工する

② 動作が停止してしまう

① 攻撃者が、 細工したリクエストを送信する
例：通常の入力　･･･パスワード
細工した入力･･･SQL クエリ等



② 使用しているデータベースを不正に操作されてしまう

- アプリケーションが利用できなくなる
- データベースに保存している個人情報が漏洩する

セキュリティ上の弱点として「脆弱性」と呼ばれる

脆弱性情報の内容と性質

- 脆弱性情報とは
  - 「脆弱性情報」とは何か
    - どのソフトウェア/ウェブサイトにどのような脆弱性が存在するのかを示す情報

      例：
      - » 脆弱性のある箇所の情報
      - » 脆弱性を確認するための手順
      - » PoC（Proof of Concept）
  - 脆弱性情報は誰に、どのように利用されるか
    - 脆弱性情報は相対する2つの性質をもつ
      - 脆弱性の**対策のために必要**な情報
      - 脆弱性を**攻撃するために悪用**される情報

脆弱性情報は適切に取り扱うことが重要

脆弱性情報と対応

♦ 推奨できない脆弱性情報の取扱い

脆弱性情報を報告しない

- 製品開発者/ウェブサイト運営者にも他の機関にも脆弱性情報を報告しない
  - 製品開発者/ウェブサイト運営者によって**脆弱性が対策されない**
  - **時間が経てば攻撃者に脆弱性を発見され、悪用される可能性が高まる**

脆弱性情報を公開する

- 脆弱性情報を公開する
  - 製品開発者/ウェブサイト運営者は公表された情報を必ずしも把握できない
  - **脆弱性に対策されるまでの間、攻撃者は脆弱性を悪用できる**

利用者を危険にさらしてしまうことになります

脆弱性情報と対応

♦ 推奨される脆弱性情報の取扱い

協調的に脆弱性情報を取り扱う

- 発見者が製品開発者/ウェブサイト運営者と**協調して脆弱性開示・対策を実施**する
- 脆弱性情報の性質を考慮した取扱いを行う
  - 脆弱性対策をするための時間的猶予が与えられる
  - 脆弱性情報は当事者以外に流出しないよう管理する

発見者と製品開発者/ウェブサイト運営者の間での
当事者同士の協力が必要です

2.2 脆弱性情報の取扱いの流れについて
協調的な取扱いにおける流れ

- ソフトウェア製品の場合の協調的な脆弱性情報の取扱いの流れ

- 脆弱性の**発見**から**対策適用**まで
  - 脆弱性が解消されセキュリティ上の脅威がなくなる
- 各段階で各当事者が**役割を担う**

2.2 脆弱性情報の取扱いの流れについて
協調的な取扱いにおける流れ：発見

当事者

- 発見者

役割

- 脆弱性情報を入手すること
  - ➢脆弱性の調査して自ら発見する
  - ➢他人から脆弱性情報を受け取る

行動

- 脆弱性の特定
- 再現手法・条件の確認
- 証跡・ログファイルの作成

2.2 脆弱性情報の取扱いの流れについて
協調的な取扱いにおける流れ：報告

当事者

- 発見者（報告者）
- 報告の受領者

  製品開発者　　専門機関（IPA）

役割

- 発見者：脆弱性情報を当事者に報告すること
- 受領者：脆弱性情報を発見者から受領すること

行動

- 発見者：「検証・対策」のために必要な情報をまとめる
- 発見者：連絡先と連絡手法を調査・確認し、報告する

2.2 脆弱性情報の取扱いの流れについて
協調的な取扱いにおける流れ：検証・対策

当事者

- 製品開発者

役割

- 脆弱性情報を検証する
- 脆弱性対策を策定する

発見者の関与

- 製品開発者が「検証・対策」ができない場合に、発見者に質問がなされることがある

2.2 脆弱性情報の取扱いの流れについて
協調的な取扱いにおける流れ：対策公開

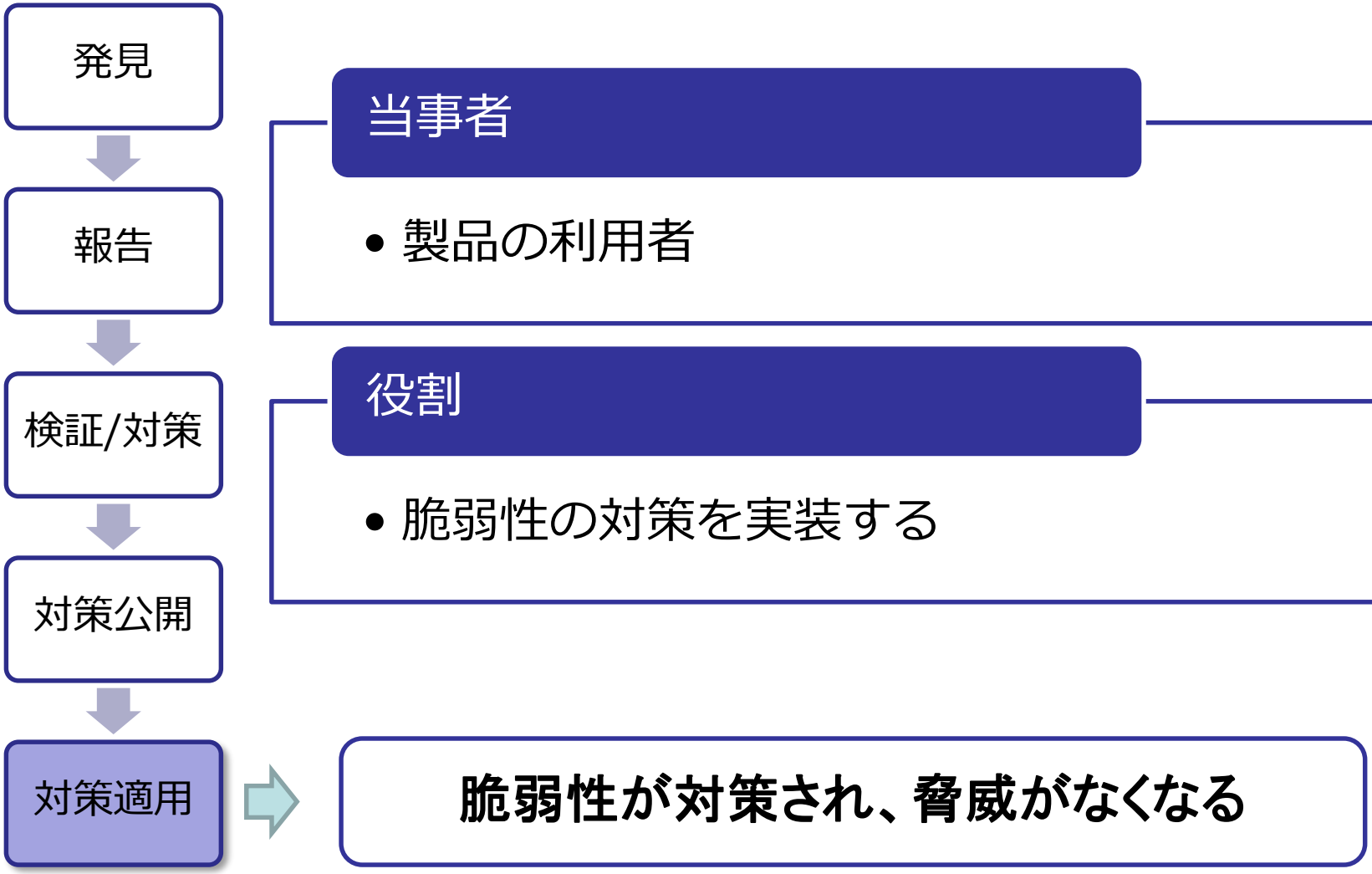
当事者

- 製品開発者
- 製品の利用者

役割

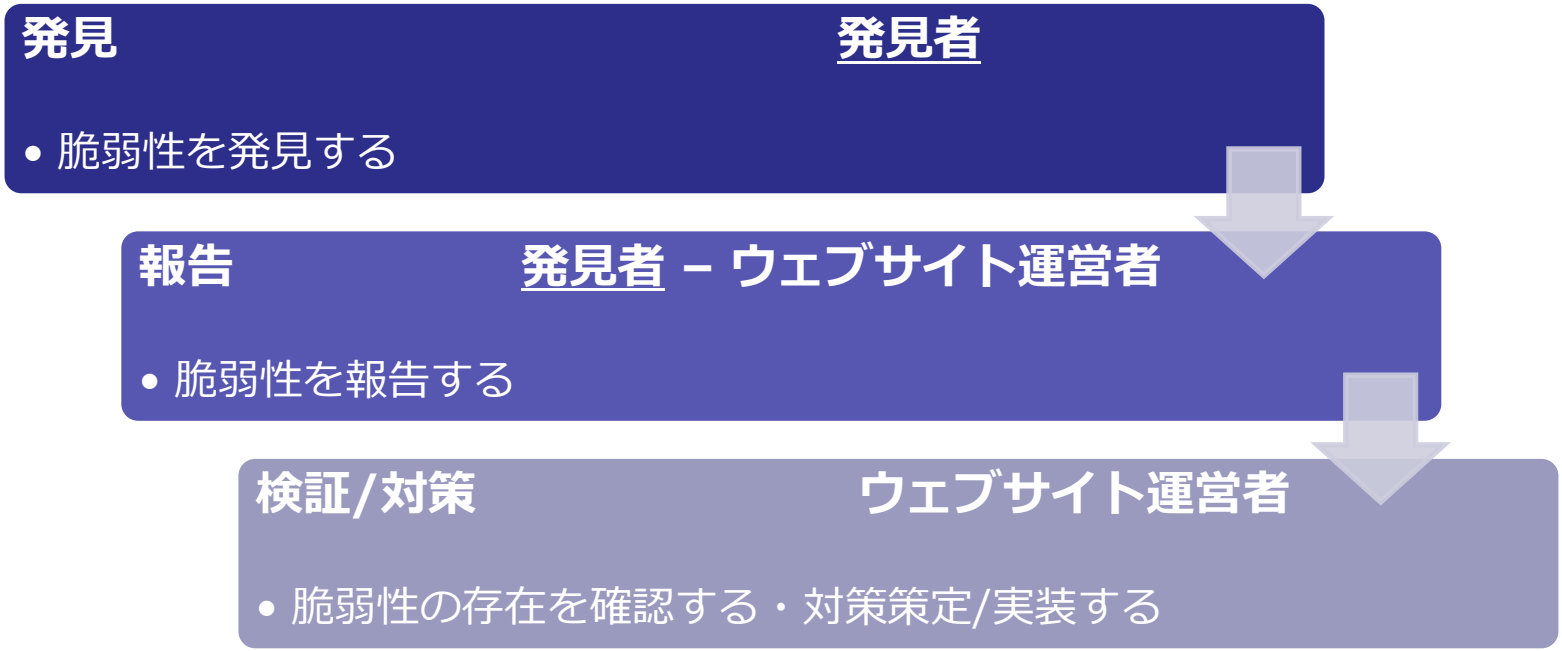
- 製品開発者：脆弱性の対策方法を利用者に伝達する
- 製品の利用者：対策情報を確認する

2.2 脆弱性情報の取扱いの流れについて
協調的な取扱いにおける流れ：対策適用



ウェブサイトの脆弱性の場合の流れ

- ウェブサイトの脆弱性の場合
  - ウェブサイト運営者が対策を実施すれば脆弱性が解消するため、「対策公開」と利用者の「対策適用」は必須ではありません。



2.3 発見者の担う役割

- 協調的な取扱いにおける発見者の役割
  - 当事者と協調しながら脆弱性情報を取り扱うこと
    - 取扱いの流れの全体において、当事者と十分なコミュニケーションをとって相互に理解を図ることが重要です。
    - とくに**「発見」**と**「報告」**の段階で重要な役割を担います。

発見

- 他人に被害を生じさせないこと

報告

- 必要な情報をまとめること
- 適切な相手に報告すること

発見時の注意点

- 他人に被害を生じさせないこと

第三者がウェブサイト運営者の許可なく脆弱性検査を行うと、ウェブサイト運営者に危険を伝える目的であっても、トラブルになる可能性があります。

- 脆弱性の調査は、**ソフトウェアの機能不全**を引き起こしたり、**ウェブサイトの停止**をもたらしたりする可能性があります。
- 善意で調査したものであっても**当事者間でトラブルとなる可能性**があります。

発見時の注意点

- 他人に被害を生じさせないこと
  - ウェブサイトの脆弱性
    - 稼働中のウェブサイトへの許可のない脆弱性調査は推奨できません
    - ウェブサイトの脆弱性を、**善意で偶然に**発見したとしても、必要以上に追加の調査行為はしないでください。
  - ソフトウェア製品の脆弱性
    - 外部との通信が発生しないような安全な環境のなかで検証してください

報告時の注意点

♦ 報告時の注意点

必要な情報をまとめること

- 「検証・対策」をするために必要な情報を記載する
- 「検証・対策」の実施者に情報が渡るようにすること

適切な相手に報告すること

- 製品開発者・ウェブサイト運営者に直接報告する
- 早期警戒パートナーシップに報告する

報告時の注意点

- 必要な情報をまとめること
  - 「検証・対策」をするために必要な情報を記載する
    - 脆弱性を特定するための情報
      - ソフトウェア製品の名称・バージョン情報
      - 脆弱性の種別
      - CVSSのスコア
    - 再現手法・条件
      - 環境情報
      - 設定の条件
    - 証跡・ログファイル
      - スクリーンショット
      - 脆弱性発見時の通信のログ

報告時の注意点

- 必要な情報をまとめること
  - 「検証・対策」の実施者に情報が渡るようにすること
    - 報告を受け取った方が、必ずしもセキュリティに詳しい技術者であるとは限りません
      - サポート窓口や広報担当に連絡する場合
    - 「分かりやすい説明」をすることで、円滑に調整が進むことが期待できます。

報告時の注意点

- 必要な情報をまとめること
  - 「検証・対策」の実施者に情報が渡るようにすること
    - 良く知られた用語に言い換える
      - 「脆弱性」→「セキュリティ上の問題」
    - 「脆弱性の種別」ではなく脅威の例を伝える
      - 「クロスサイト・スクリプティング」
        → 「悪意あるサイトに誘導されたり、情報漏えいを生じさせるような弱点」

報告時の注意点

♦ 適切な相手に報告すること

直接に製品開発者/ウェブサイト運営者に連絡する方法と第三者機関（IPA）に報告する方法があります。

製品開発者/ウェブサイト運営者

- サポート窓口
- CSIRT/PSIRT
- バグバウンティプログラム

公的機関・専門機関

- IPA（情報セキュリティ早期警戒パートナーシップ）

報告時の注意点

- 製品開発者/ウェブサイト運営者に直接報告を行う場合のよくあるトラブル
  - 製品開発者/ウェブサイト運営者の連絡先がない
  - 連絡しても信用されない
  - 自分のサイトへの攻撃を疑われる

- 製品開発者/ウェブサイト運営者も、予算や人員の問題など、脆弱性の対処のために様々な課題を抱えていることを考慮することも重要です。

- 製品開発者/ウェブサイト運営者との調整が難しい場合にはIPA（パートナーシップ）に届出を行ってください。

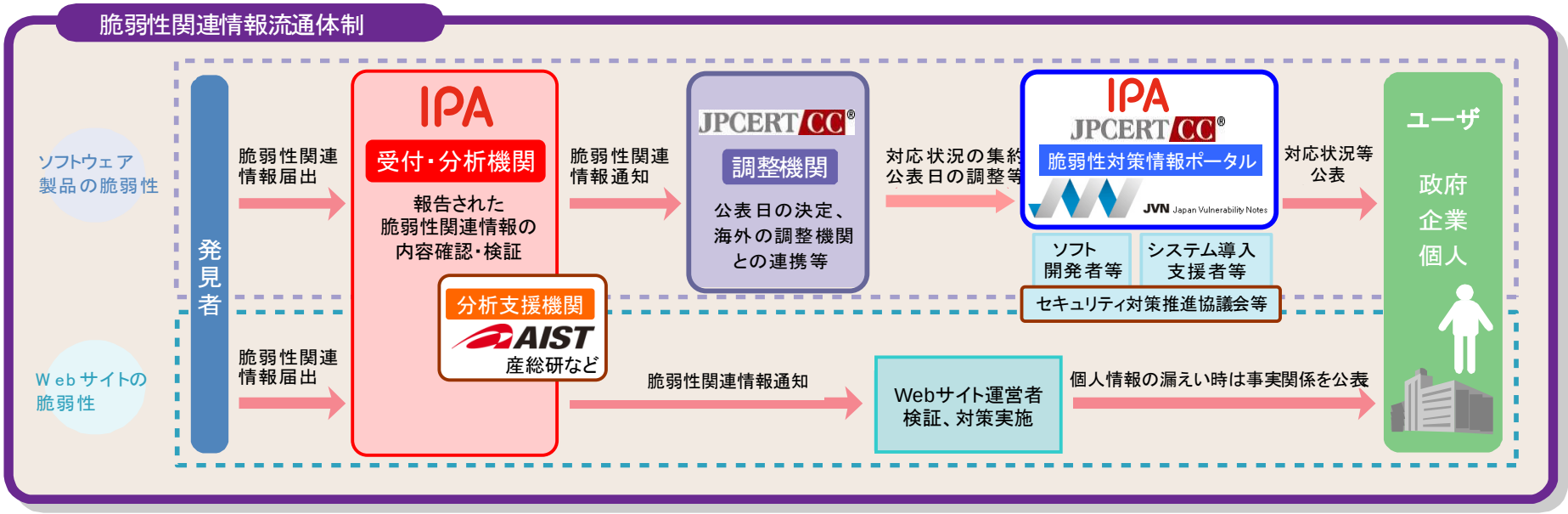
報告時の注意点

- IPA・JPCERT/CC

脆弱性の報告を IPA が受け付け、製品開発者/ウェブサイト運営者に連絡し、対策を依頼します。
製品開発者/ウェブサイト運営者と直接やり取りすることなく、脆弱性情報を報告することができます。

  - 脆弱性情報を取り扱う**「情報セキュリティ早期警戒パートナーシップ」**を運営
    - 発見者からの届出を受け付け、製品開発者/ウェブサイト運営者に通知し、対策を促す制度
    - 各当事者の**報告**/**検証**/**公開**の役割をIPA・JPCERT/CCが部分的に担う

2.3 発見者の担う役割
報告時の注意点

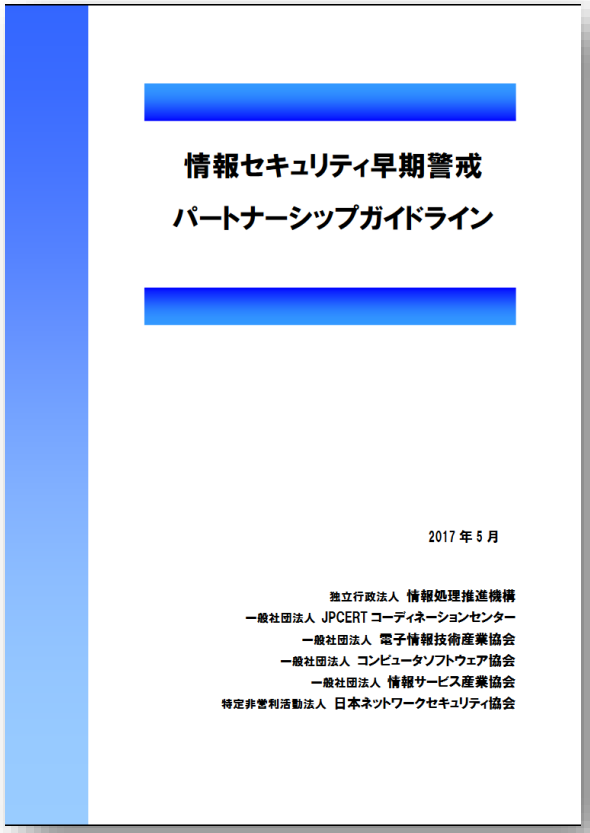


- ソフトウェア製品の脆弱性
  - 調整機関であるJPCERT/CCが製品開発者との連絡・調整を実施
  - JVNでの対策情報の公開
- ウェブサイトの場合
  - IPAがウェブサイト運営者に連絡を実施
  - 脆弱性情報は公開されない
    - 個人情報漏えいの場合は公表を推奨

報告時の注意点

♦ 制度に届出をする際の注意点

- 告示・ガイドラインに基づく運用をしているため、告示・ガイドラインの対象範囲に該当しない問題については取扱いできません。
- **「情報セキュリティ早期警戒パートナーシップガイドライン」**と**「脆弱性関連情報として取り扱えない場合の考え方の解説」**を届出する前に確認してください。



告示: https://www.meti.go.jp/policy/netsecurity/vul_notification.pdf
ガイドライン: https://www.ipa.go.jp/security/ciadr/partnership_guide.html

脆弱性関連情報として取り扱えない場合の考え方の解説:
https://www.ipa.go.jp/security/vuln/report/notice/handling_notaccept.html

2.4 脆弱性報告のメリット

謝辞の掲載

- JVN（パートナーシップへの届出）
- 製品開発者のサイト（開発者の意向に依る）

バグバウンティ（脆弱性報奨金制度）による報奨金

- 脆弱性を発見・報告した方に報奨金を支払う取り組み
- 詳細はプログラムの運営者が定める取り決めに従う

2.5 まとめ

脆弱性情報の性質について

- 対策のために必要な情報でもあり、攻撃に悪用される情報でもある
- 脆弱性情報にはその性質を考慮した取扱いが必要となる

脆弱性情報の取扱いの流れについて

- 発見者は「発見」と「報告」に密接にかかわる

脆弱性の発見と報告の際の注意事項について

- 他人に被害を与えないようにする
- 必要な情報を整理し、適切な相手に報告する

社会全体のセキュリティ向上のために
適切な脆弱性の発見･報告にご協力ください

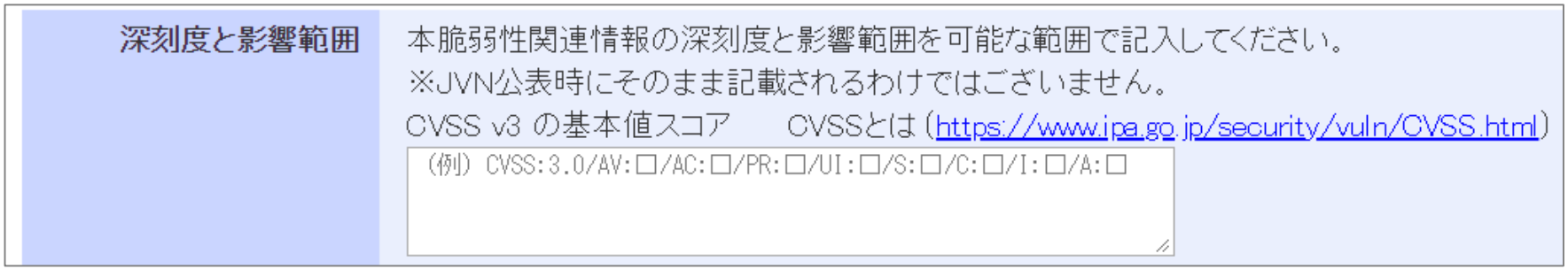
IPA Better Life with IT

3. 脆弱性の評価方法
～ 脆弱性届出に CVSS v 3 を活用しよう ～

独立行政法人情報処理推進機構（IPA）
セキュリティセンター

本プログラムの概要

- IPAでは脆弱性の届出を受け付けています。
- 届出のフォーム上で、CVSSv3 基本値の入力欄を設けています。



ソフトウェア製品脆弱性関連情報の届出
（https://isec-vul-form.ipa.go.jp/ipa-vul-main/software/vulnerability）

➡ IPA が上記の入力欄を設けた意図と、
CVSSv3 基本値の評価方法を解説します。

目次

3.1 CVSSとは

Common Vulnerability Scoring System

（共通脆弱性評価システム）

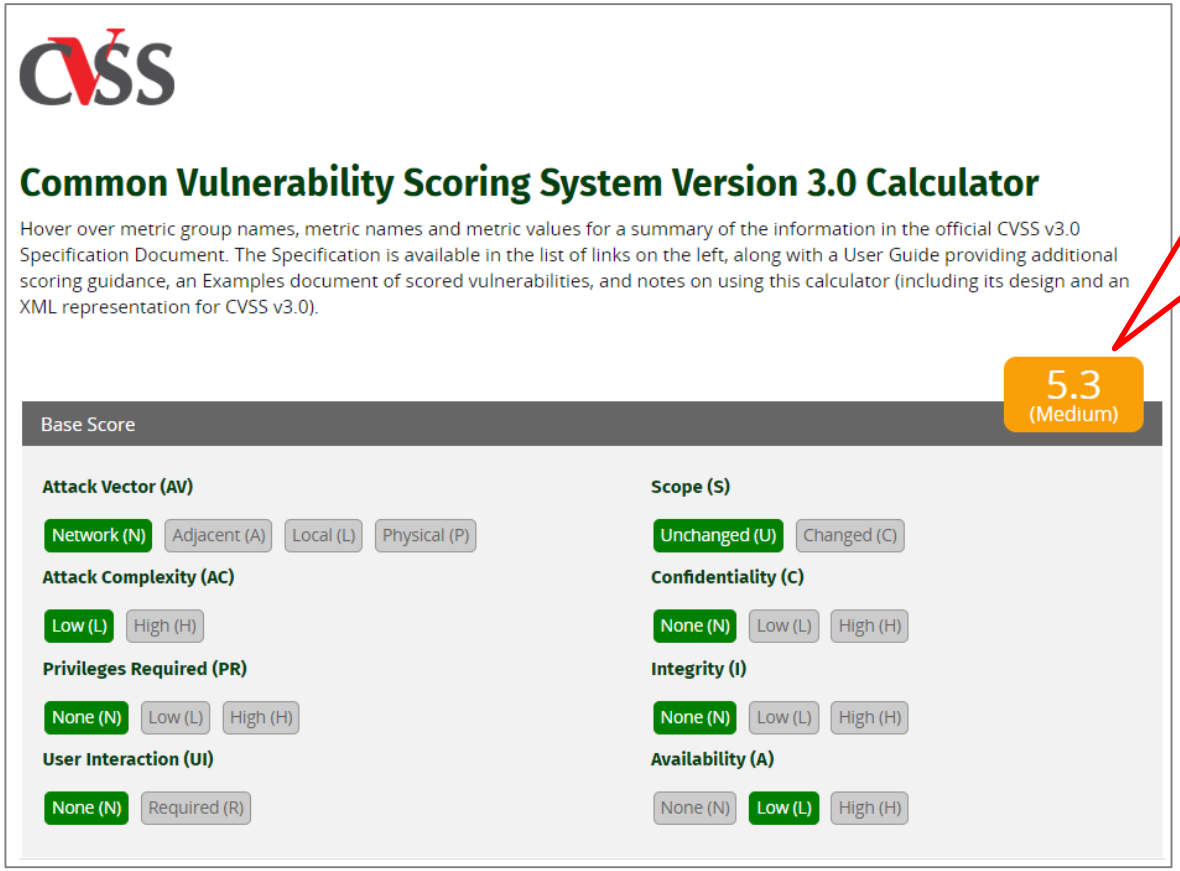
- 脆弱性に対する汎用的な評価手法
- 脆弱性の深刻度を 0.0 ～ 10.0 で評価
- CVSSv3では 3 つの基準で脆弱性の深刻度を評価

I.　基本評価基準 (Base Metrics)	【**脆弱性そのもの**の評価】 • 悪用の条件と影響
II.　現状評価基準 (Temporal Metrics)	【**脆弱性を取り巻く状況**を評価】 • 既に悪用されている？ • 修正パッチは提供されている？
III. 環境評価基準 (Environmental Metrics)	【**個別の環境における影響度**を評価】 • 自社環境での影響は？ • 影響を受けるシステムの重要度は？

3.1 CVSSとは

- CVSS計算ツールが公開されている

Common Vulnerability Scoring System Version 3.0 Calculator
https://www.first.org/cvss/calculator/3.0

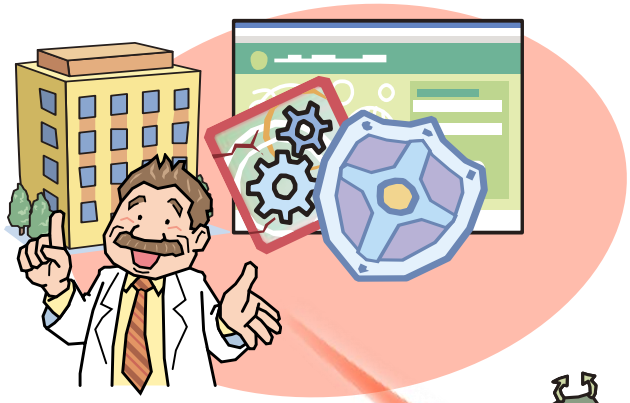


評価項目を選択していくと、CVSS値が算出される。

各評価項目については、後程解説します。

3.1 CVSSとは
～CVSSの活用イメージ～

セキュリティ機関・製品開発者



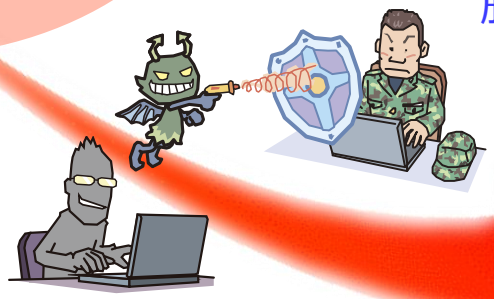
I. 基本評価基準を評価

脆弱性そのものの深刻度を評価

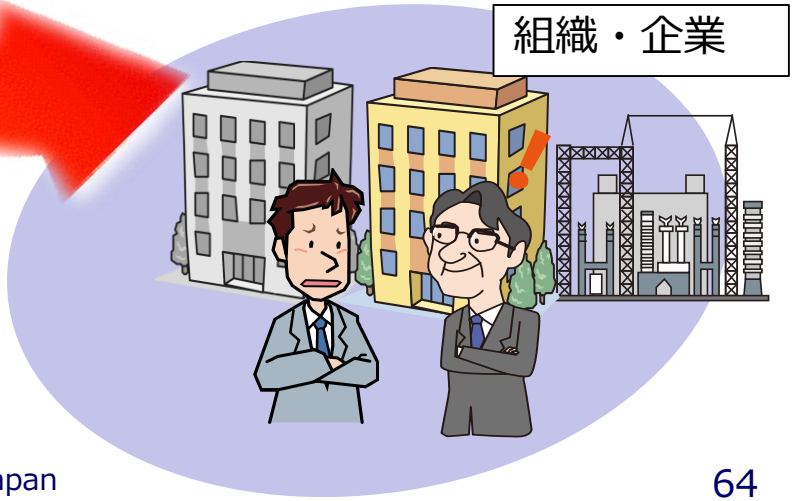
セキュリティ機関や製品開発者が
公開するのは基本値のみ！

II. 現状評価基準を評価

脆弱性の現状の深刻度
（悪用状況や修正状況）を評価



組織・企業



III. 環境評価基準を**ユーザが**評価

個別の環境における深刻度を評価

→ **脆弱性対応を検討**

3.1 CVSSとは
～CVSSの活用イメージ～

余談

基本値だけを見ると深刻度が高くない脆弱性も、
場合によっては早急な対応が必要になることも…

例）

• SQLインジェクション	基本値：7.3
• 悪用なし、修正パッチあり	現状値：6.7
• 社内向け環境で利用	環境値：4.4

• サービス運用妨害（DoS）	基本値：5.3
• 悪用あり、修正パッチなし	現状値：5.3
• 重要な外部向け環境で利用	環境値：6.1

基本値だけではなく、3つの評価基準で評価して
対応を検討することが望ましい

3.2 届出にCVSS値を利用する意図

- IPAが、脆弱性情報届出受付フォーム上で、CVSSv3 の入力欄を設けている意図
  - CVSSv3 基本値に沿った記載をいただき、脆弱性の悪用条件や影響を正確に把握したい
  - 製品開発者にも脆弱性の深刻度を正しく伝えたい
  - 不明点などの確認事項を減らし、よりスムーズな脆弱性のハンドリングを行いたい

しかしながら…

評価方法が難解な項目もあるため
以降で評価方法を説明します

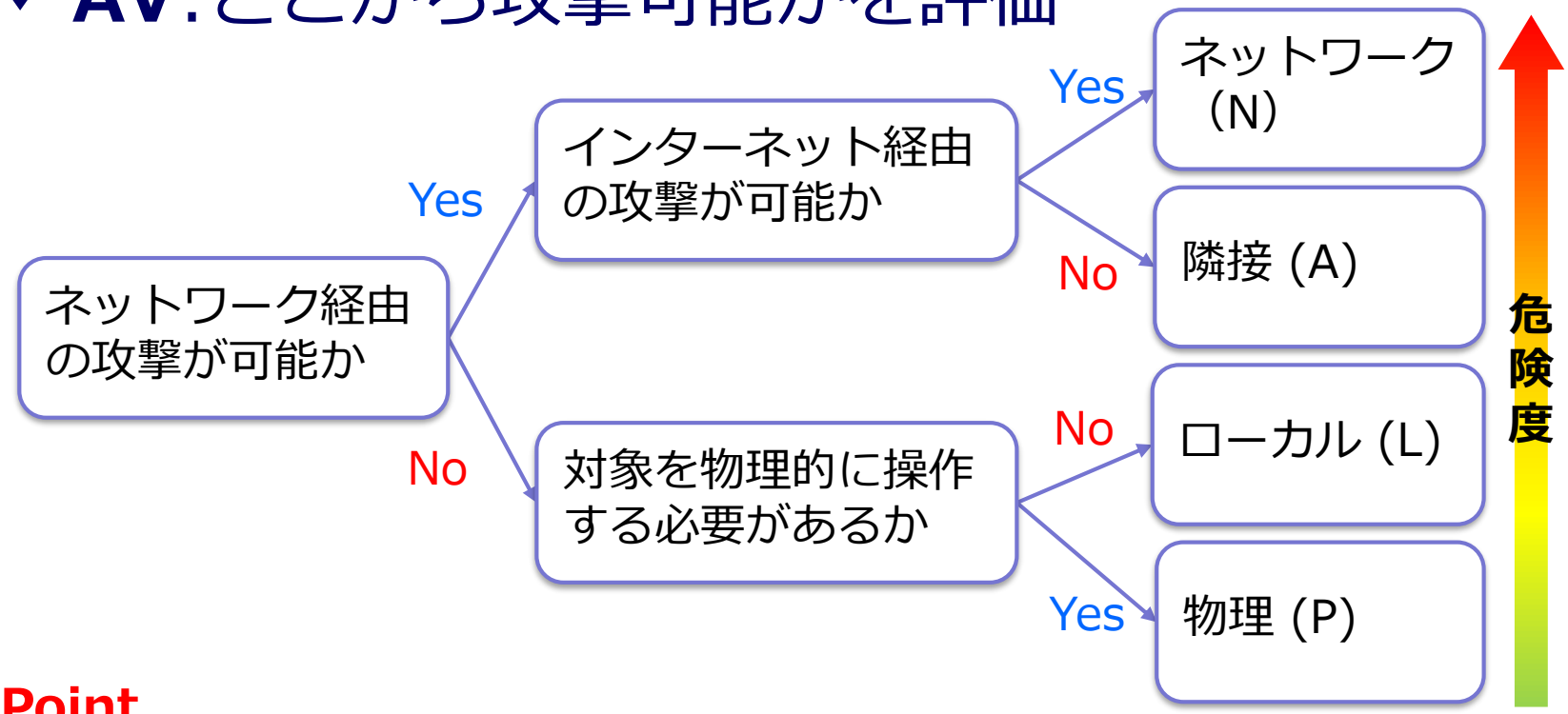
3.3 基本評価基準(Base Metrics)

危険小 ⟷ 危険大

	評価項目	選択肢・ポイント			
攻撃の難易度	どこから攻撃可能であるか 攻撃元区分　(AV : Attack Vector)	物理(P)	ローカル(L)	隣接(A)	ネットワーク(N)
	攻撃する際に必要な条件の複雑さ 攻撃条件の複雑さ(AC : Attack Complexity)	高（H）		低(L)	
	攻撃する際に必要な権限のレベル 必要な権限レベル(PR : Privileges Required)	高(H)	低(L)		不要(N)
	攻撃にユーザの関与が必要か否か ユーザ関与レベル(UI : User Interaction)	要(R)		不要(N)	
攻撃による影響	別のコンポーネントへの影響 スコープ(S : Scope)	変更なし(U)		変更あり(C)	
	機密情報の漏えい被害 機密性への影響　(C : Confidentiality Impact)	なし(N)	低(L)		高(H)
	情報の改ざん被害 完全性への影響　(I : Integrity Impact)	なし(N)	低(L)		高(H)
	業務が遅延・停止する被害 可用性への影響 (A : Availability Impact)	なし(N)	低(L)		高(H)

3.3 基本評価基準(Base Metrics)
攻撃元区分（Attack Vector）

- **AV**:どこから攻撃可能かを評価



AV
AC
PR
UI
S
C
I
A

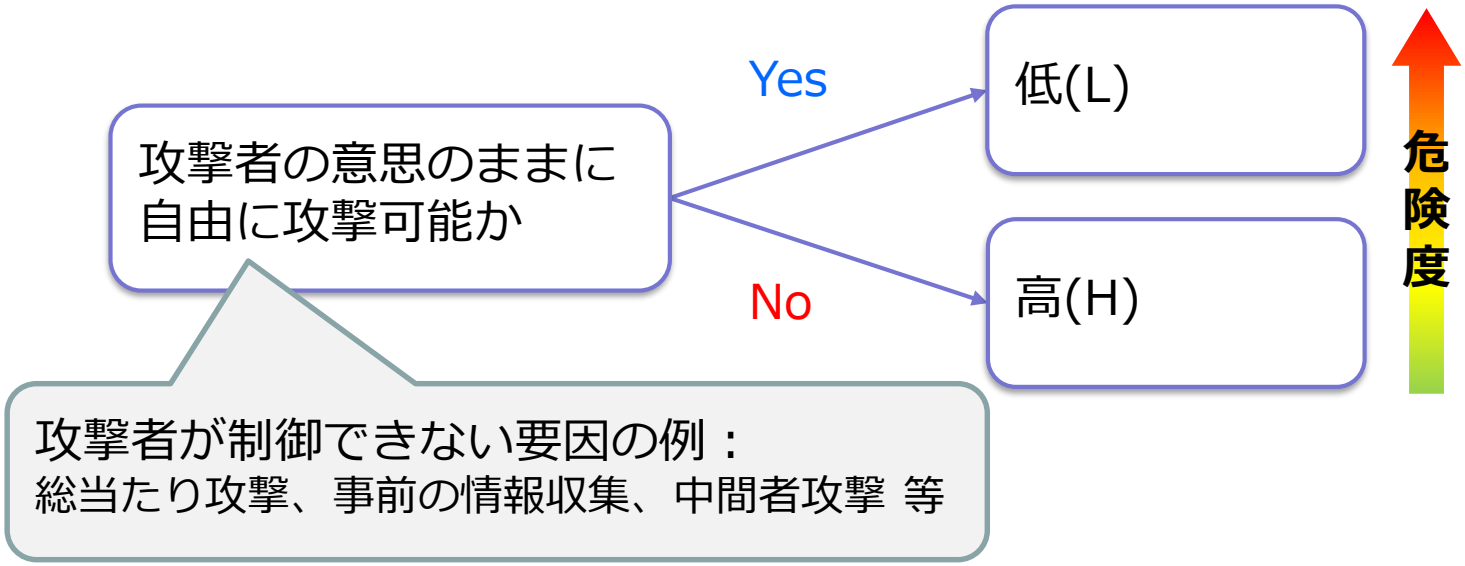
Point

✓ 悪意のあるコンテンツを送付し、ユーザに実行させる等の場合は？
→ 悪意のあるコンテンツをダウンロードまたは受信するための
ユーザ操作を必要とする場合はローカル（L）となる

3.3 基本評価基準(Base Metrics)
攻撃条件の複雑さ (Attack Complexity)

IPA

- **AC:攻撃者が制御できない要因を評価**



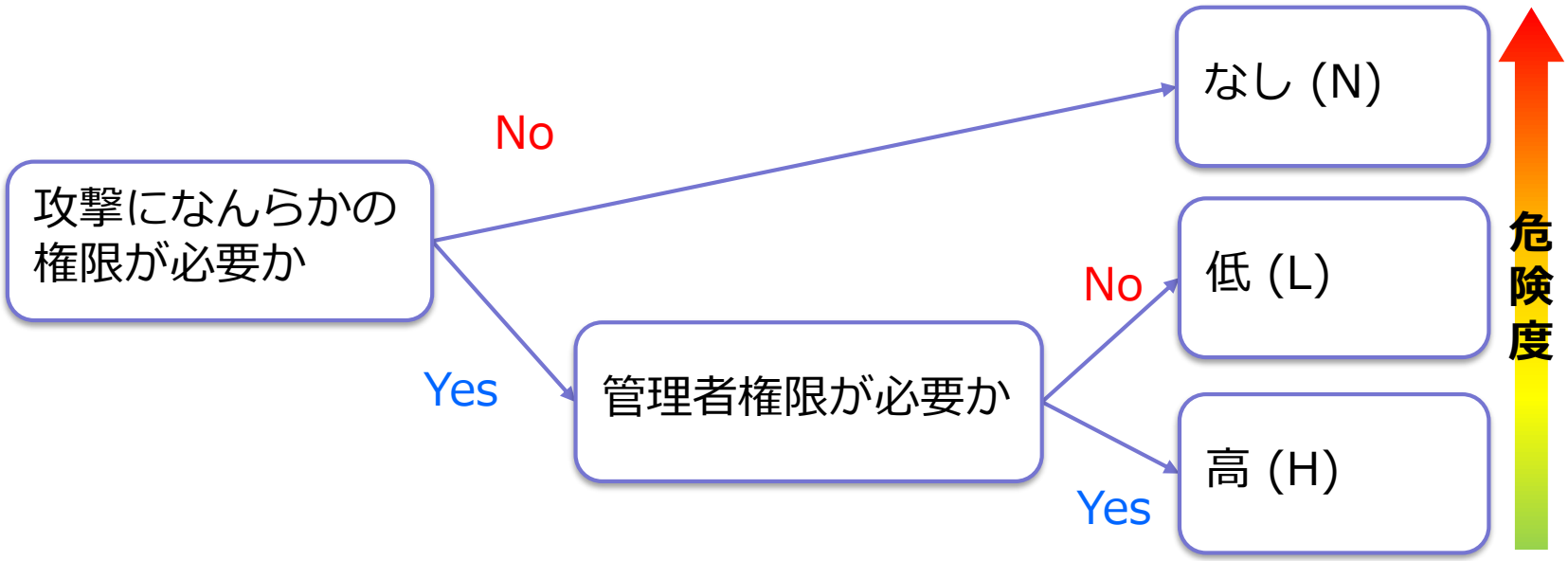
Point

- ✓ 攻撃の確率的な困難性や、特定の情報・状況の必要性といった、**攻撃者のスキルでは克服できない煩雑さ**を評価する。
- ✓ ユーザー操作の要因はここでは評価しない（後述するUIで評価）

3.3 基本評価基準(Base Metrics)
必要な権限レベル(Privileges Required)

IPA

- **PR**:攻撃に必要となる権限レベルを評価



AV
AC
PR
UI
S
C
I
A

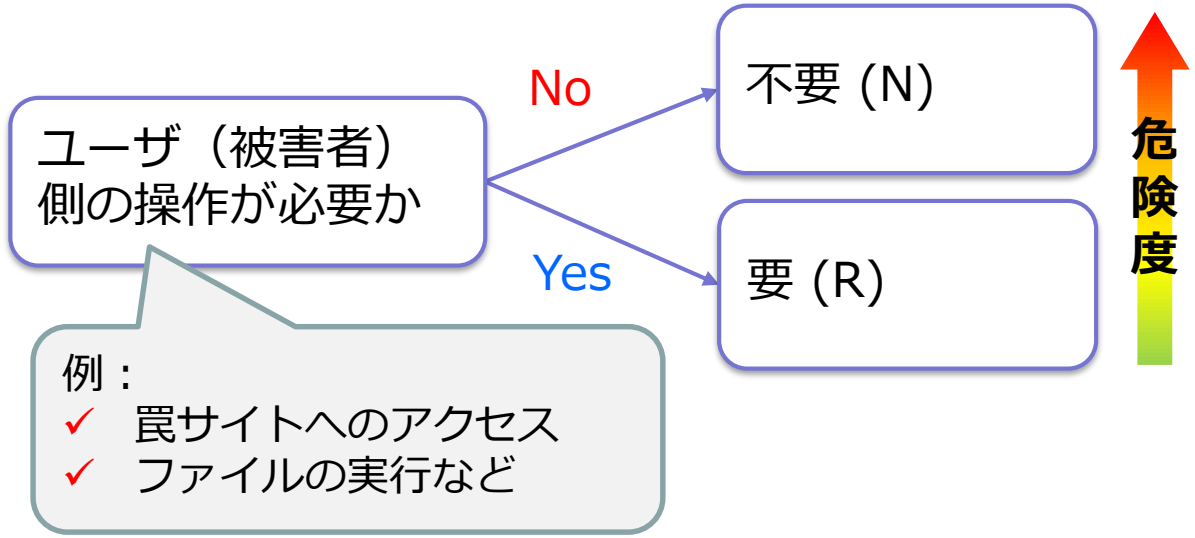
Point

✓ あくまで攻撃者に権限が必要かどうかを評価する
ログイン済みの管理者を罠サイトに誘導する攻撃シナリオ等を
誤って評価しないよう注意

3.3 基本評価基準(Base Metrics)
ユーザ関与レベル(User Interaction)

IPA

- **UI**:攻撃にユーザの操作が必要かどうか



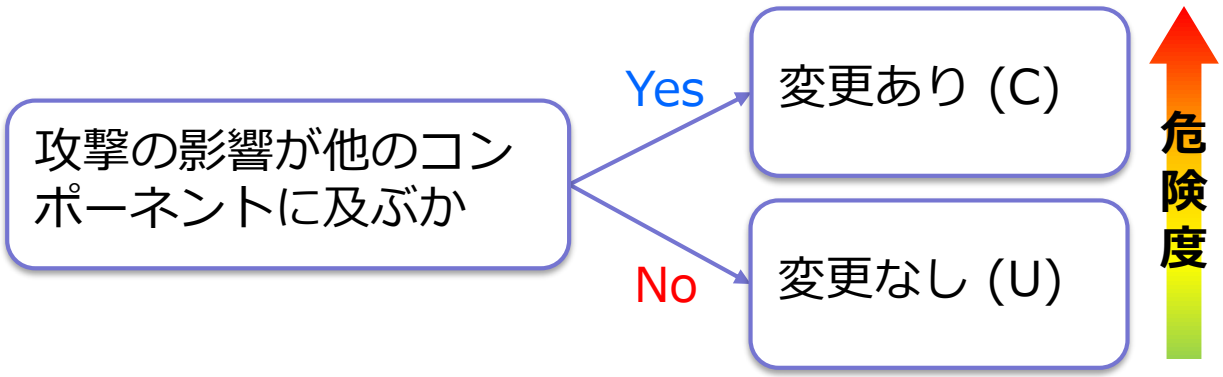
Point

- ✓ **攻撃の引き金を引くのが、被害者か攻撃者か**に着目する
- ✓ ユーザの「操作や行動」を評価する。「状態」は評価しない
  - ユーザがログインの操作をした瞬間に再現　→　要（R）
  - ユーザがログイン中ならいつでも再現　→　不要（N）

3.3 基本評価基準(Base Metrics)
スコープ(Scope)

IPA

- **S:被害の影響範囲を評価**

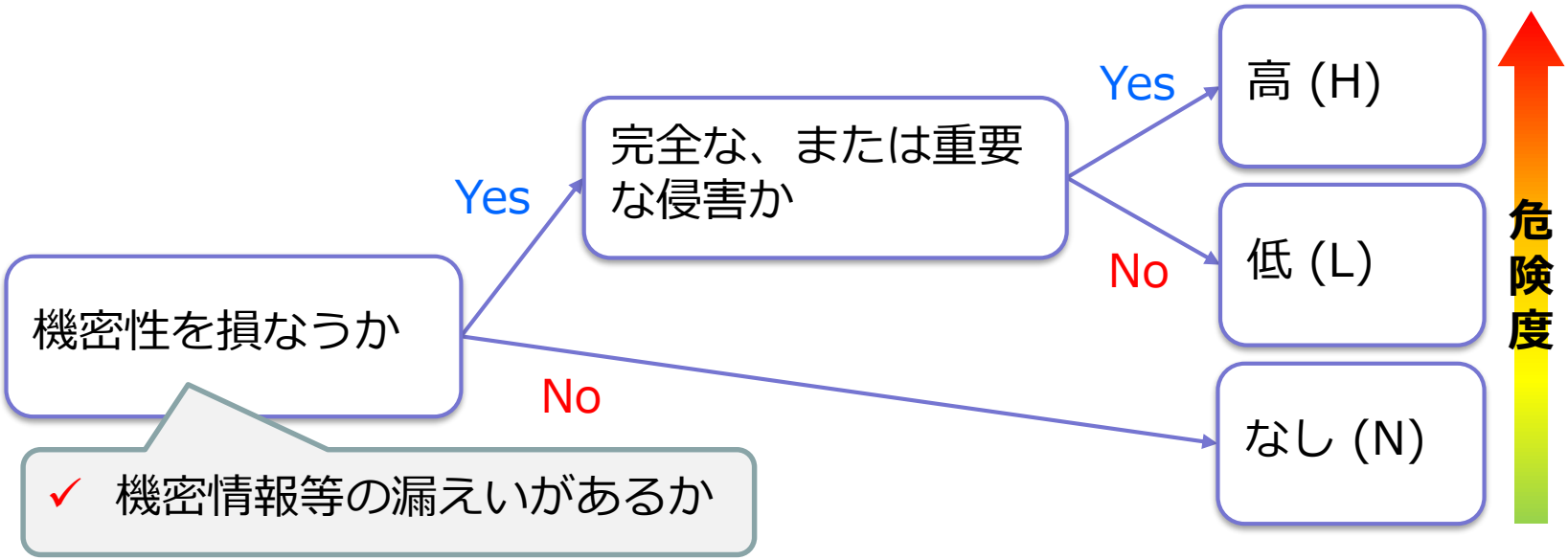


Point

- ✓ 脆弱性の影響が、脆弱なコンポーネントの権限範囲を超えて、他のコンポーネントに及ぶ場合に評価する。
  例：
  - ウェブアプリケーションの脆弱性で、ブラウザが保有する情報を取得されるような場合
  - 仮想環境においてゲスト環境からホスト環境を攻撃できるような場合

3.3 基本評価基準(Base Metrics)
機密性への影響(Confidentiality Impact)

- **C:機密性に与える影響を評価**

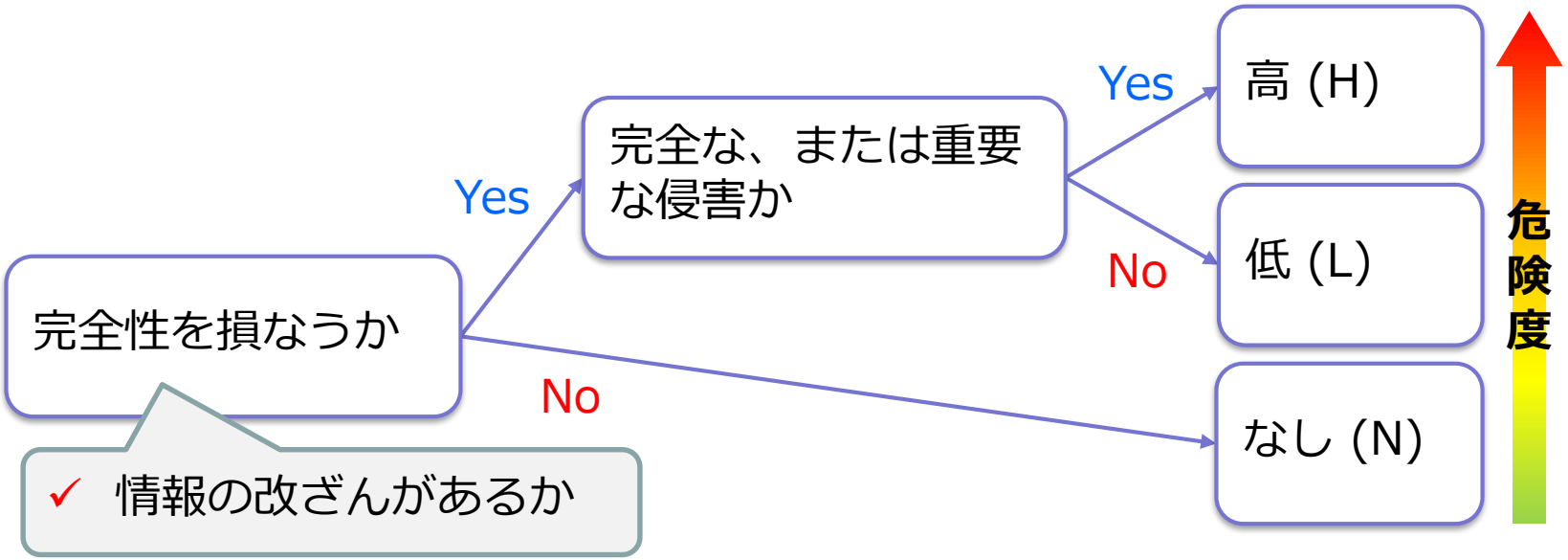


Point

- ✓ 侵害の評価例：
  - 対象製品の全ての情報にアクセス可能 → 高（H）
  - 管理者の認証情報が漏えいする → 高（H）
  - サーバのアクセスログが漏えいする → 低（L）

3.3 基本評価基準(Base Metrics)
完全性への影響 (Integrity Impact)

- **I:完全性に与える影響を評価**

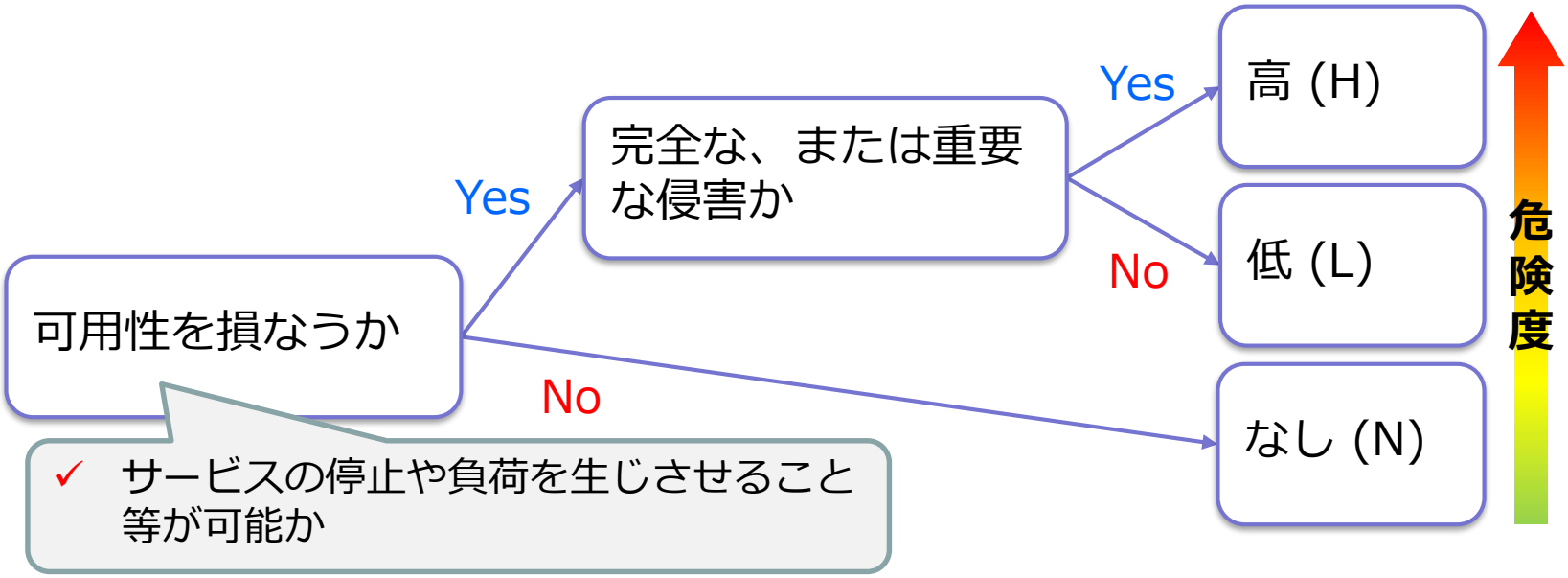


Point

✓ 機密性の評価と同様に、改ざん可能な対象情報の範囲と重要度を考慮する。

3.3 基本評価基準(Base Metrics)
可用性への影響 (Availability Impact)

- **A**:可用性に与える影響を評価



Point

- ✓ **完全に停止するかどうか**と、**影響の持続性**に着目する。
- ✓ 対象コンポーネントが完全に停止する、または攻撃が終了した後も継続的に影響が持続する場合、高（H）と評価する。

3.3 基本評価基準(Base Metrics)
基本値評価のポイント

Point（全体）

- ✓ 攻撃シナリオベースで評価する
  - 第三者に対する攻撃シナリオ（≠再現手順）を想定して、攻撃条件や影響を評価します。
  - 複数の攻撃シナリオが想定される場合は、より値が高くなる方の攻撃シナリオを評価します。
  - 想定した攻撃シナリオも届出に記入いただけると幸いです。
- ✓ 評価に迷うような場合は、値が高くなる方で評価する
  - 評価項目と詳細を届出に記載いただけると幸いです。IPAが製品開発者に確認し、再評価します。

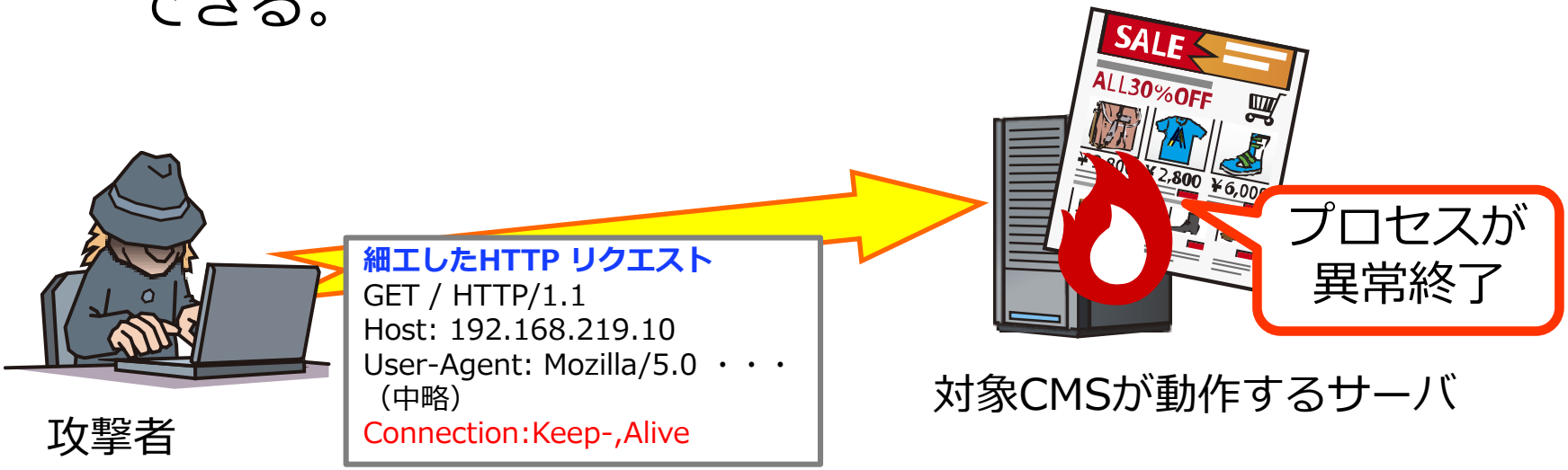
3.4 ケーススタディ
実際に脆弱性を評価してみましょう

✓ 具体的な脆弱性の概要を見ながらCVSS基本値を評価してみます。

- 目的
  - 脆弱性対策情報の CVSS の各評価を見たときに、深刻度の詳細を大まかに把握できるようになる
  - どういった条件が危険で、何をされると深刻な脆弱性となるのかを把握できるようになる
  - 自身で脆弱性の評価や優先度付けができるようになる

実際に脆弱性を評価してみましょう

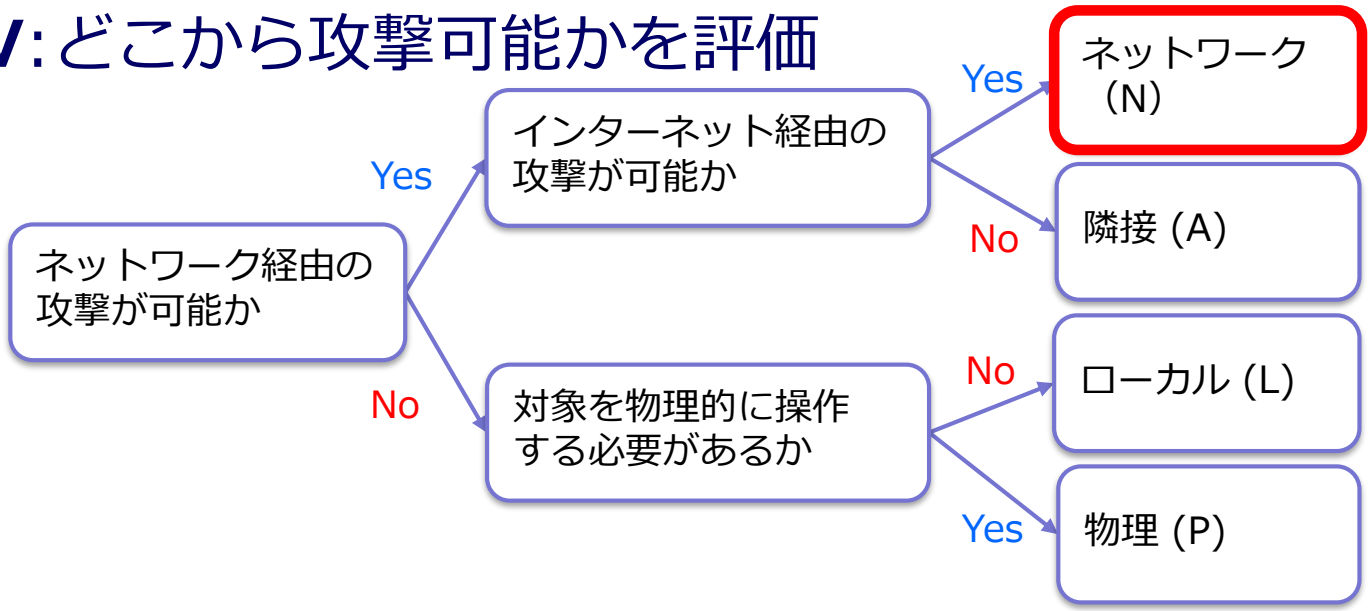
- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて送信される、 HTTPリクエストの Connection ヘッダの値に「,」を含む文字列を記載し、送信することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。



3.4 ケーススタディ
実際に脆弱性を評価してみましょう

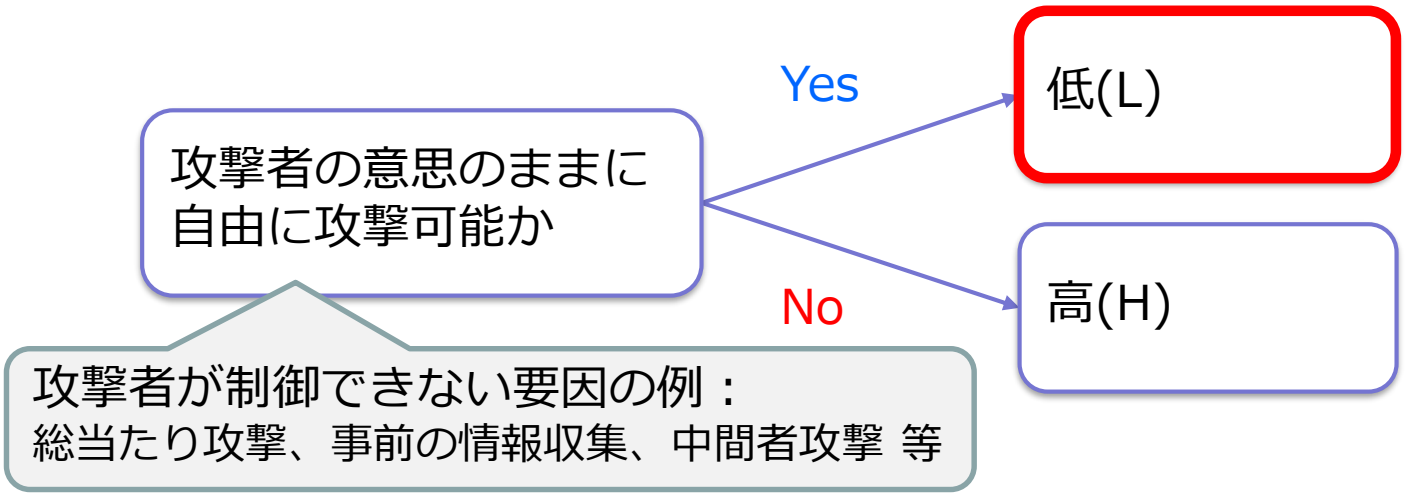
- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。

- **AV**:どこから攻撃可能かを評価



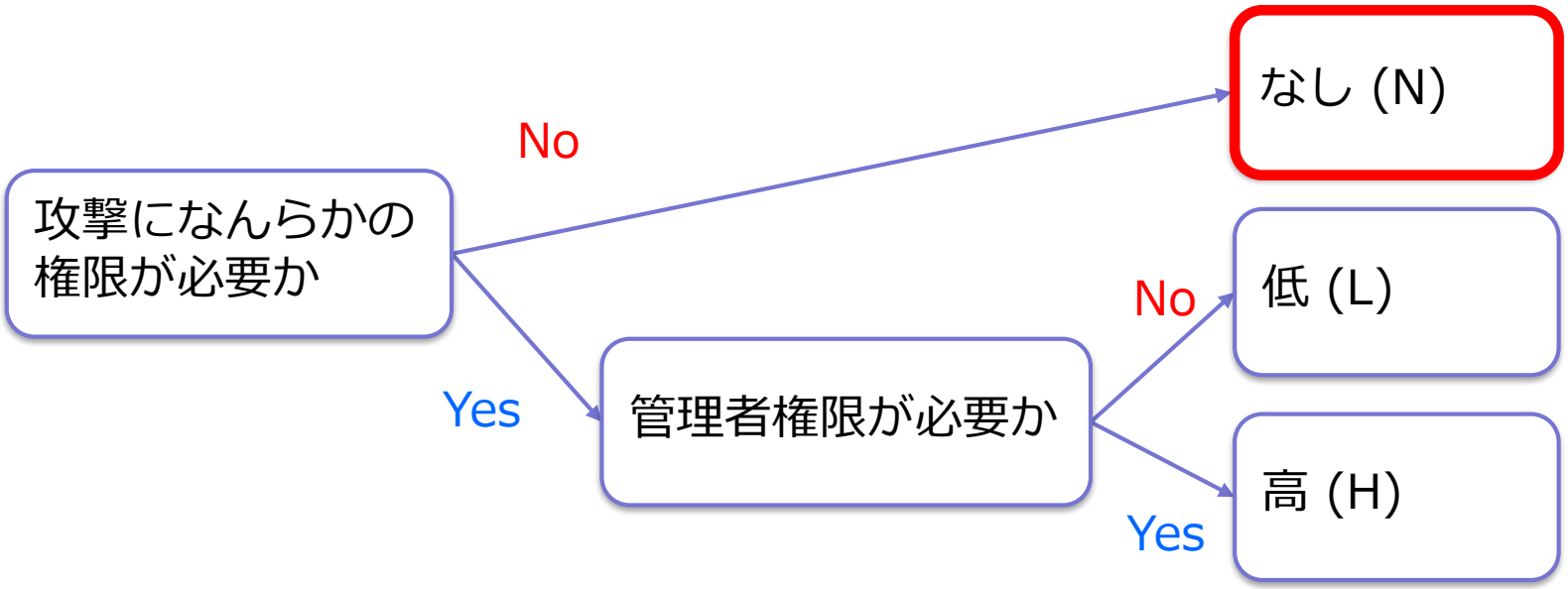
3.4 ケーススタディ
実際に脆弱性を評価してみましょう

- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。
- **AC**:攻撃者が制御できない要因を評価



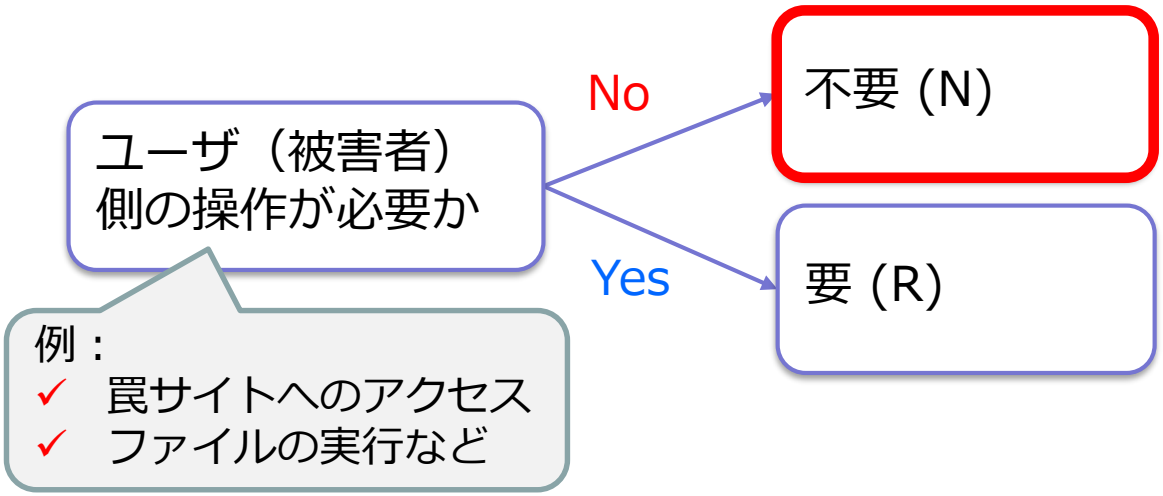
3.4 ケーススタディ
実際に脆弱性を評価してみましょう

- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。
- **PR**:攻撃に必要となる権限レベルを評価



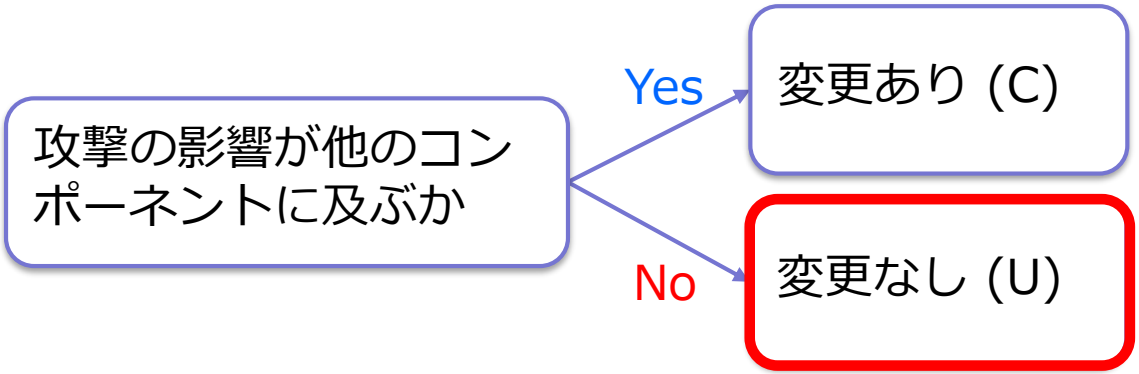
3.4 ケーススタディ
実際に脆弱性を評価してみましょう

- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。
- **UI**:攻撃にユーザの操作が必要かどうか



3.4 ケーススタディ
実際に脆弱性を評価してみましょう

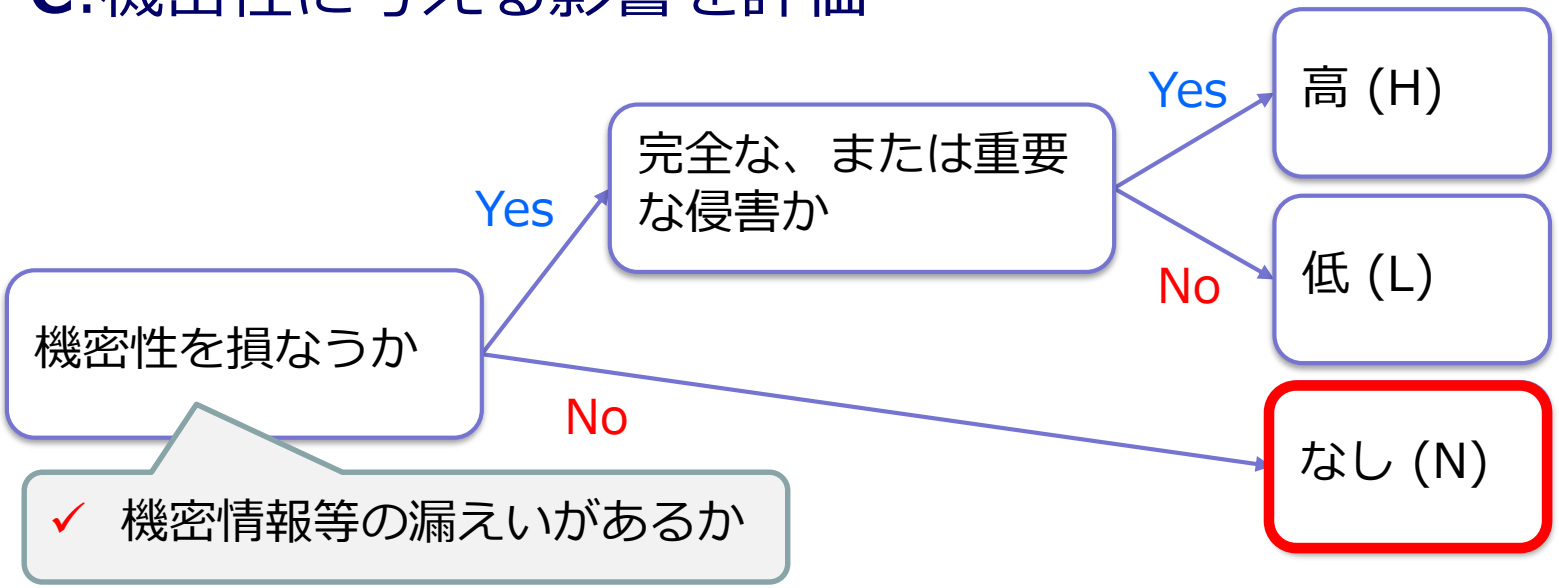
- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。
- **S**:被害の影響範囲を評価



3.4 ケーススタディ
実際に脆弱性を評価してみましょう

- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。

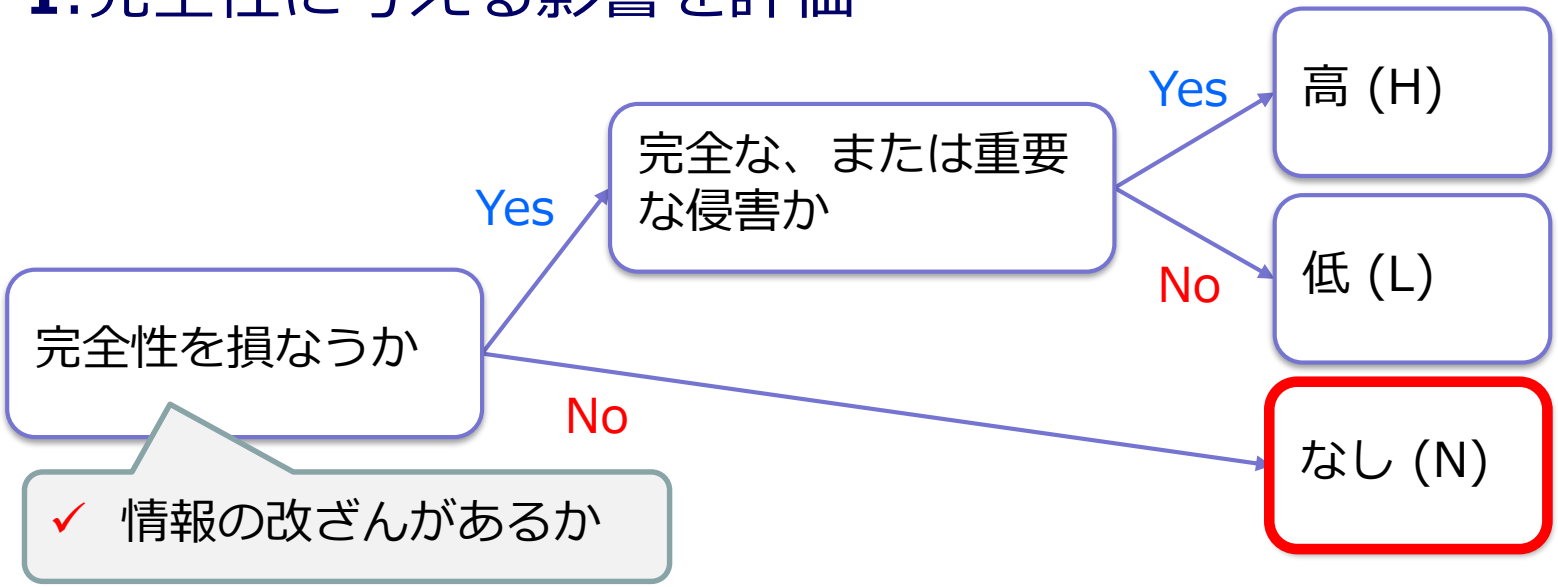
- **C**:機密性に与える影響を評価



3.4 ケーススタディ
実際に脆弱性を評価してみましょう

- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。

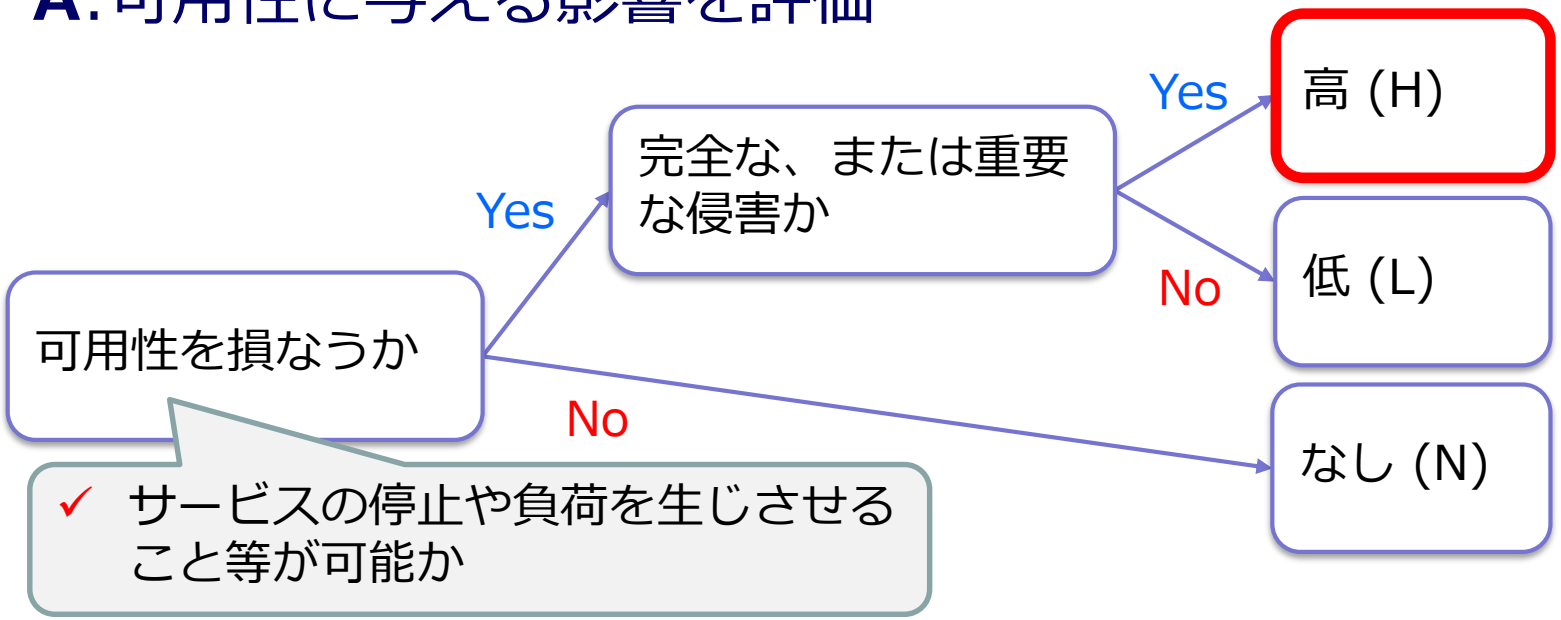
- **I**:完全性に与える影響を評価



3.4 ケーススタディ
実際に脆弱性を評価してみましょう

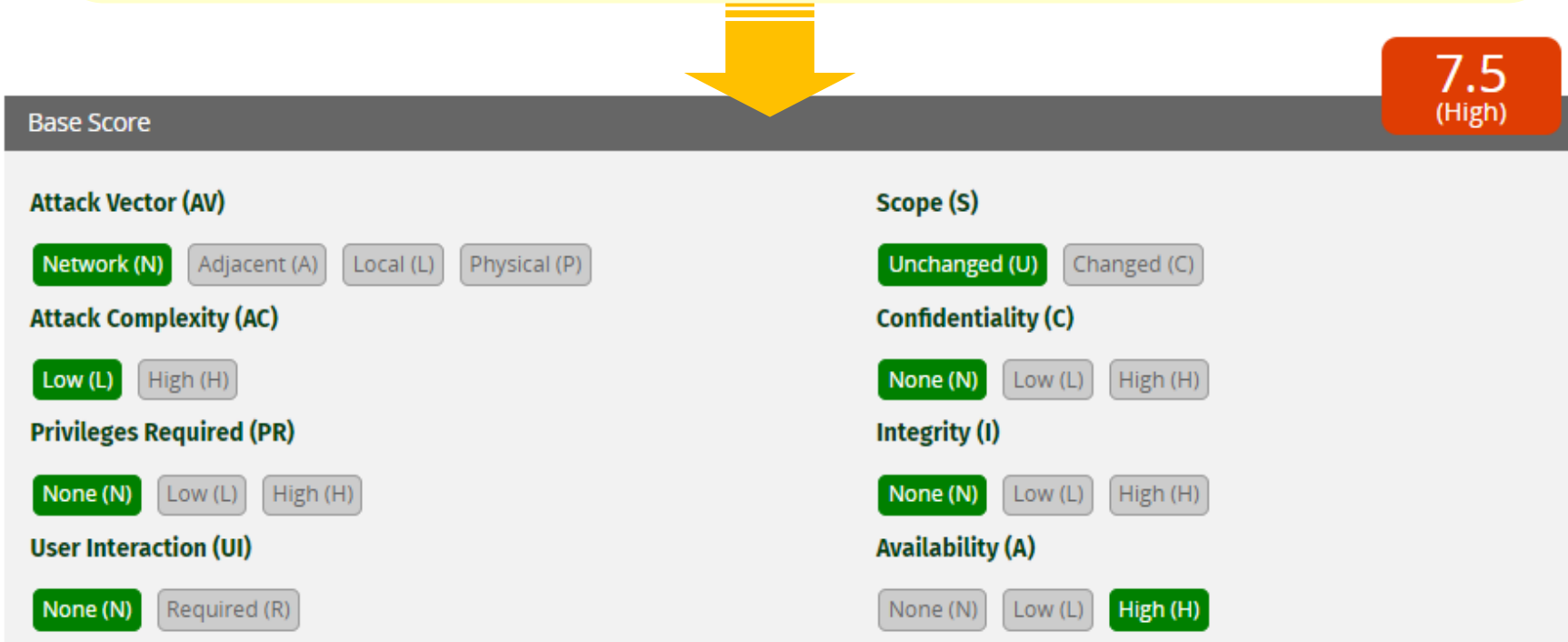
- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。

- **A**:可用性に与える影響を評価



3.4 ケーススタディ
実際に脆弱性を評価してみましょう

- ケーススタディ：サービス運用妨害（DoS）
  - 対象製品はECサイトを構築・運用できるCMS製品。
  - 商品検索フォームにおいて、Connectionヘッダの値に「,」を含む文字列を記載したHTTPリクエストを送付することで再現する。
  - 攻撃の成功により、対象製品を異常終了させることができる。



3.5 最後に

- まとめ
  - IPAが届出フォームに CVSSv3 基本値の入力欄を設けた意図を説明しました。
  - CVSSv3 の正しい活用方法を説明しました。
  - CVSSv3 基本値を評価するケーススタディを行いました。

3.5 最後に

- CVSSについてもっと知りたい方は

- 共通脆弱性評価システムCVSS v3概説
  https://www.ipa.go.jp/security/vuln/CVSSv3.html

- Common Vulnerability Scoring System v3.0: Specification Document
  https://www.first.org/cvss/specification-document

IPA Better Life with IT

4. 特別講演
IPAの届出制度を利用した経験談

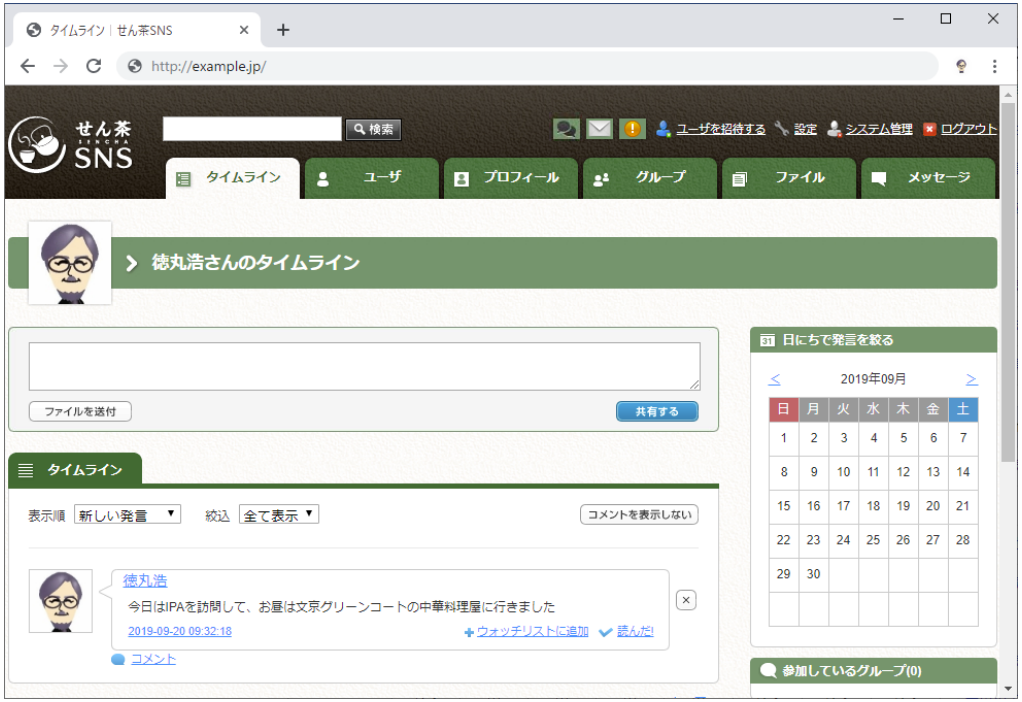
独立行政法人情報処理推進機構（IPA）
セキュリティセンター

目次

4-1. せん茶SNSの脆弱性
4-2. Android版Kindleの脆弱性
4-3. 脆弱性を発見する際に注意している点
4-4. IPAの届出制度への届出をして良かった点/大変だった点
4-5. IPA以外の組織への届出をして良かった点/大変だった点
4-6. これから脆弱性を発見しようとする方々に伝えたいこと

4-1. せん茶SNSの脆弱性
せん茶SNSとは

- 国産・オープンソースのSNS
- CakePHPにて開発されている
- 2012年2月初版リリース



脆弱性を見つけたきっかけ

- 新しい国産SNSがオープンソースでリリースされたことを知る
- 「初物」なので脆弱性があるのではないかという~~期待~~懸念
- オープンソースソフトウェアの安全性に貢献したい（建前）
- 脆弱性を発見するというワクワクを体験したい（本音）

実施したこと

- 仮想環境にせん茶SNSをダウンロード、インストール
- 普通に使ってソフトウェアに習熟する（実はこれが大変だったりする）
- 手動で通常の脆弱性診断を実施、下記の脆弱性を発見
  - セッションIDの固定化
  - CSRFが見つかる
- IPAに届出する

修正、公開までの流れ

- 2012-2-18 届出
- 2012-2-20 届出受信の連絡
- 2012-3-12 起算日および詳細情報通知のご連絡
- 2012-3-29 せん茶SNS Ver1.02として修正
- 2012-4-15 届出情報公表のご連絡

4-1. せん茶SNSの脆弱性
届出内容の紹介（抜粋）

3) 脆弱性の種類

クロスサイト・リクエストフォージェリ

4) 再現手順

利用者がSNSにログイン状態にて、後述の罠HTMLを閲覧すると、利用者の意図と無関係に、利用者のアカウントで投稿される。あるいは、投稿が削除、管理者ユーザの場合ユーザが削除される。

ただし、Refererの条件あり（次項にて説明）

5) 再現の状況

□ 常に　■ 時々　□ まれに

補足説明（バージョンによる、言語による、などを記入）

利用者がRefererを送信しない設定にしている場合。

罠サイトがHTTPSのページで、かつSNSがHTTPSでない場合。この場合はRefererが送信されないため。

4-1. せん茶SNSの脆弱性
せん茶SNS1.0.2リリースのお知らせ（引用）

投稿日: 2012年3月29日 作成者: admin

本日、せん茶SNS1.0.2をリリースいたしました。

今回のバージョンでは、セキュリティの脆弱性の対応、仕様の改善、不具合の対応含む

計19点の対応を行っています。

今回、脆弱性の対応を含むアップデートのため、バージョン1.0.2以前を使用されている

方は早めに更新下さい。

1.セキュリティの脆弱性対策

・クロスサイト・リクエストフォージェリの対策

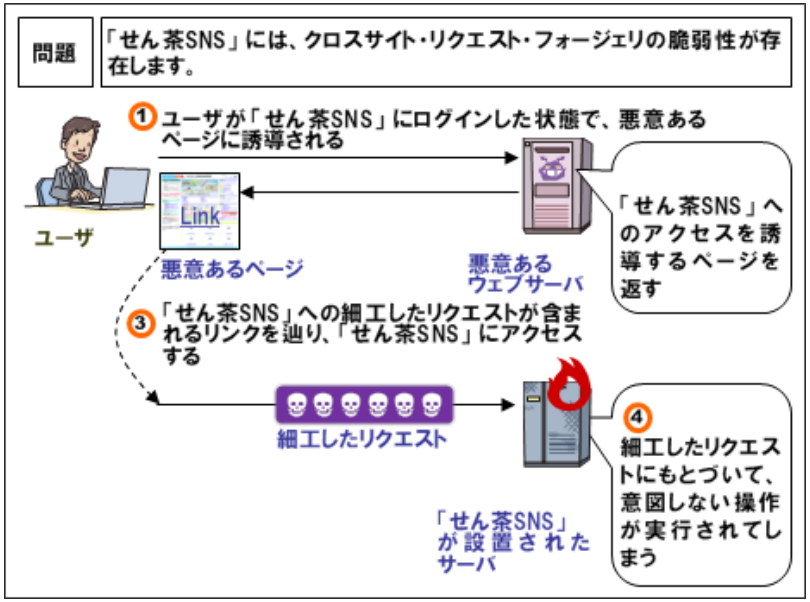
・セッションIDが固定だったが、ログイン後にセッションIDが変更されるよう対応

http://oss.icz.co.jp/news/?p=501 より引用

4-1. せん茶SNSの脆弱性
JVN iPediaでのリリース（引用）

JVNDB-2012-000029

せん茶SNS におけるクロスサイトリクエストフォージェリの脆弱性



概要

せん茶SNS には、クロスサイトリクエストフォージェリの脆弱性が存在します。

せん茶SNS は、オープンソースの SNS 構築ソフトウェアです。せん茶SNS には、クロスサイトリクエストフォージェリの脆弱性が存在します。

この脆弱性情報は、情報セキュリティ早期警戒パートナーシップに基づき下記の方が IPA に報告し、JPCERT/CC が開発者との調整を行いました。

報告者: HASHコンサルティング株式会社 徳丸 浩 氏

【中略】

共通脆弱性識別子(CVE) CVEとは?

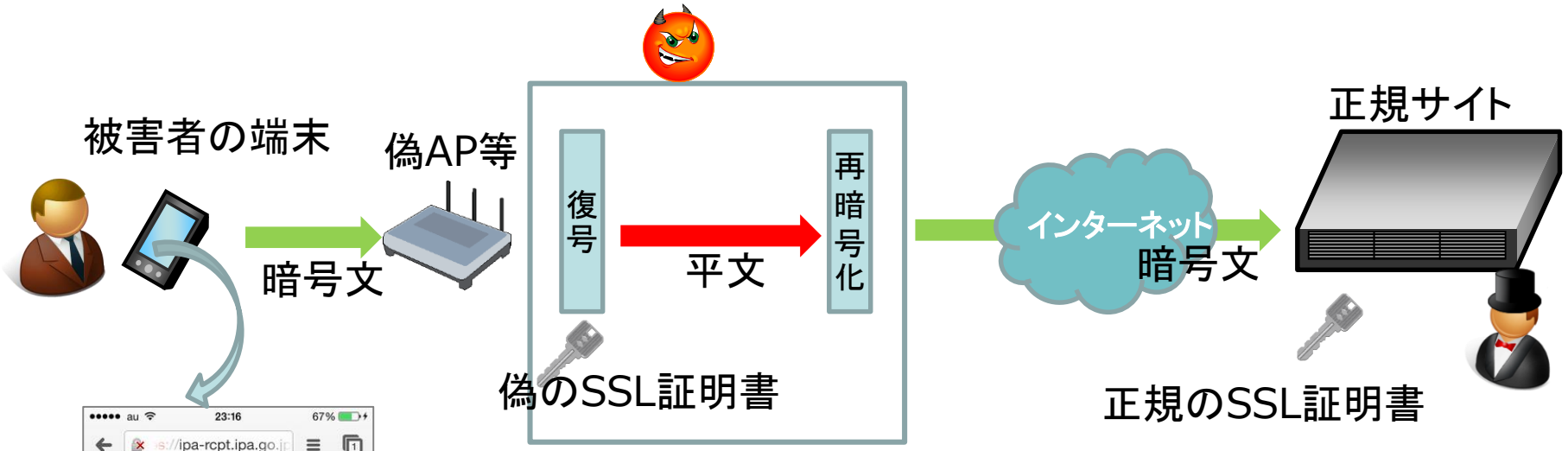
CVE-2012-1237

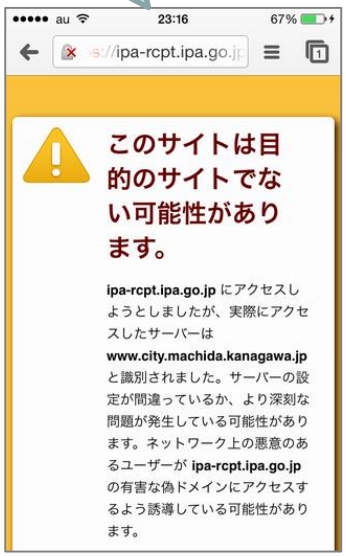
https://jvndb.jvn.jp/ja/contents/2012/JVNDB-2012-000029.html より引用

届出の感想

- 比較的スピーディに対応いただけた点が良かった
- CSRFやセッションIDの固定化等の基本的な脆弱性がまだまだあるのだな…という感想（2012年時点では）
- CVEが採番されたのは嬉しかった

4-2. Android版Kindleの脆弱性
中間者攻撃とは

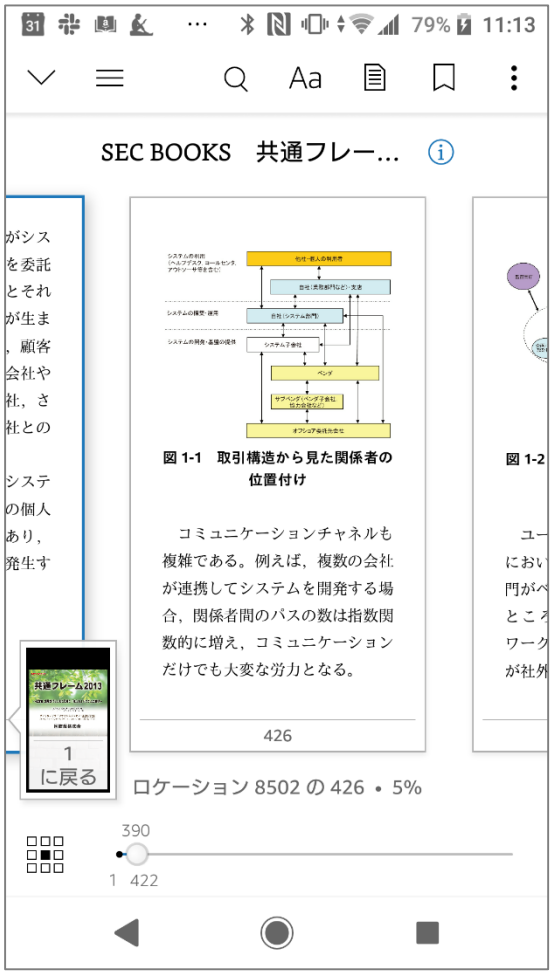




- 中間者攻撃とは上図のように通信路上の攻撃者が暗号文を受信・復号した上で盗聴・改ざんを行い、最暗号化してサーバーに送信する攻撃手法
- ブラウザからのアクセスの場合、正規の証明書でないことを確認して、左図のようなエラーを表示する
- スマホアプリの場合は、証明書の検証をアプリ側で行う必要があるが、検証が不十分なケースは、利用者が攻撃に気づけない場合がある

4-2. Android版Kindleの脆弱性
Android版Kindleとは

- Amazonが提供する電子書籍KindleのAndroid版リーダー
- 電子書籍を読むのが基本機能
- 電子書籍以外に自ら登録した文書を読むこともできる（Kindleパーソナル・ドキュメントサービス）



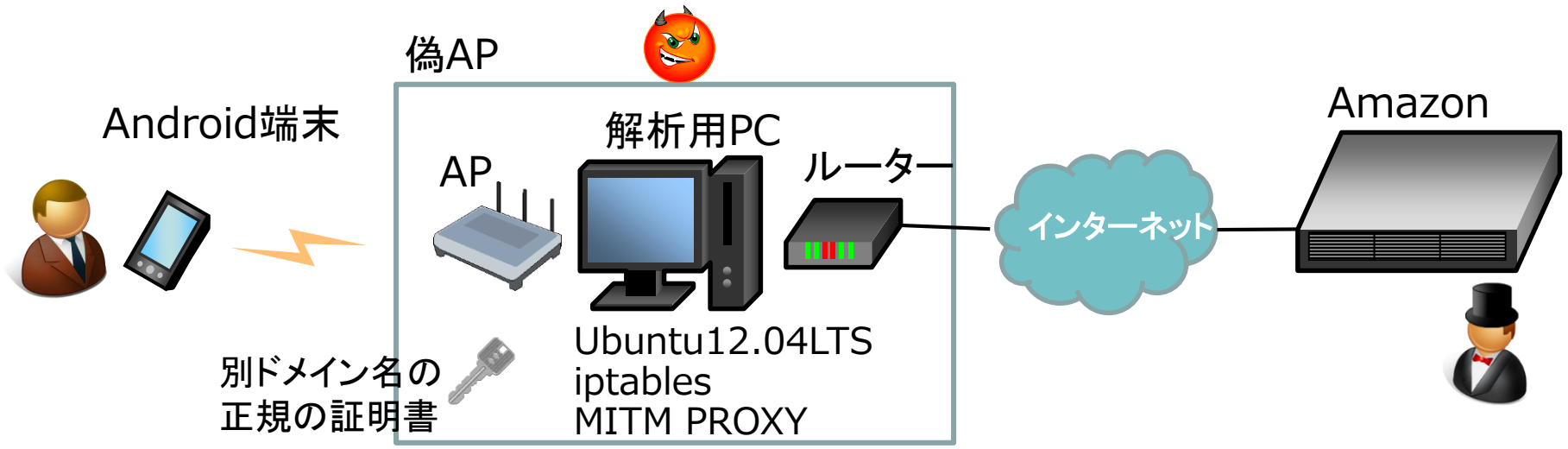
Andoid版Kindleで電子書籍を閲覧している様子

脆弱性を見つけたきっかけ

- 2013年後半から、スマホアプリのSSL証明書検証不備について確認していたが、届出しても開発者との確認がつかない状況だった
- 奥一穂氏から証明書のエラーには、自己署名証明書以外にも、コモンネームの検証不備もあり得ることを示唆いただく（2014年1月28日）
- 後述する検証環境を作り、手当たりしだいにスマホアプリを試してみた

実施したこと

- 別ドメイン名の正規証明書により中間者攻撃の仕組みを作成
- 手当たり次第にスマホアプリを試す
- Android版Kindleのパーソナルドキュメント機能に脆弱性を発見



4-2. Android版Kindleの脆弱性
修正、公開までの流れ

- 2014-2-2 届出
- 2014-2-3 届出受信の連絡
- 2014-8-1 起算日のご連絡
- 2014-8-7 詳細情報通知のご連絡
- 2014-8-12 Android 版アプリ Kindle 4.6.0 として修正公開(4.5.0にて修正されたはず)
- 2014-8-29 届出情報公表のご連絡

4-2. Android版Kindleの脆弱性
届出内容の紹介（抜粋）

3) 脆弱性の種類

SSL通信にあたり、サーバー証明書の検証をしておらず、中間者攻撃が可能となる。

4) 再現手順

Kindleアプリを導入したAndroid端末のWi-Fiからインターネット上の経路上に透過プロキシを導入する。透過プロキシには、正規のサーバー証明書を導入するが、アプリケーションサーバーのホスト名とは異なるコモンネームのものであるので、証明書検証時にエラーになるはずである。端末は予めログイン状態にセットしておく。

この状態で、Amazonのパーソナルドキュメントの機能を利用して、PDFファイルを端末にダウンロードさせる。この際の経路はSSLであるが、上記「正規のサーバー証明書ではあるがコモンネームの異なるもの」を導入している透過プロキによるMITMによりエラーなく盗聴が可能であった。その際のHTTPメッセージを mitm.txt として添付する。このメッセージのHTTPレスポンスボディは、パーソナルドキュメントの機能により登録したPDFドキュメントと目視上一致した。【以下略】

4-2. Android版Kindleの脆弱性
JVN iPediaでのリリース（引用）

JVNDB-2014-000102

Android 版アプリ Kindle における SSL サーバ証明書の検証不備の脆弱性

概要

Android 版アプリ Kindle には、SSL サーバ証明書の検証不備の脆弱性が存在します。

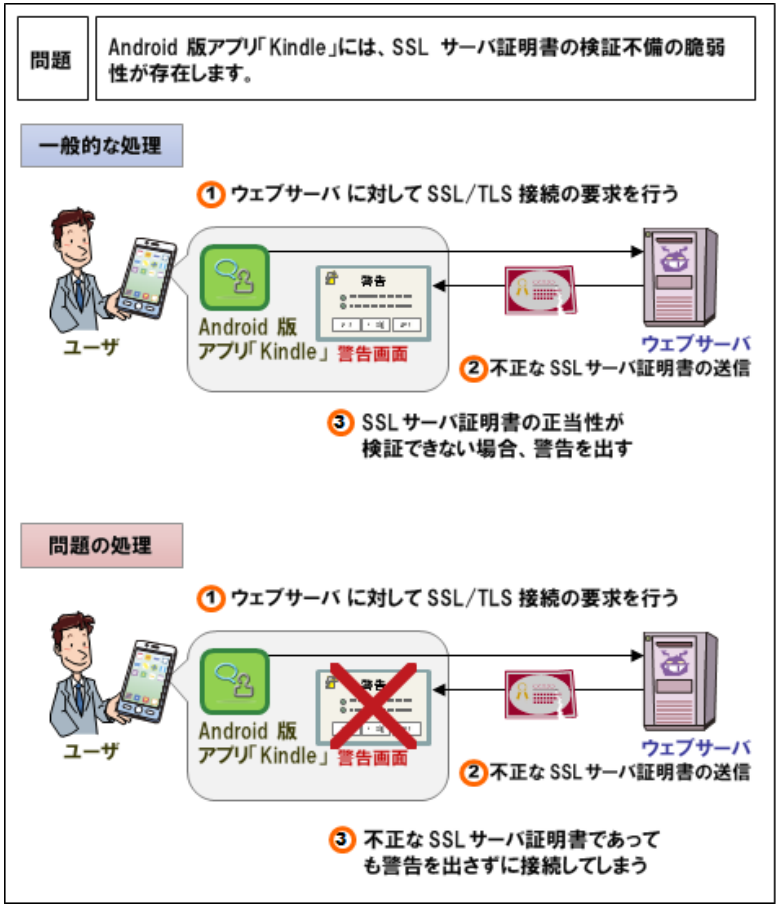
この脆弱性情報は、情報セキュリティ早期警戒パートナーシップに基づき下記の方が IPA に報告し、JPCERT/CC が開発者との調整を行いました。

報告者: HASHコンサルティング株式会社 徳丸 浩 氏

【中略】

共通脆弱性識別子(CVE) CVEとは?

CVE-2014-3908



https://jvndb.jvn.jp/ja/contents/2014/JVNDB-2014-000102.html より引用

届出の感想

- 届出から「詳細情報通知のご連絡」まで随分時間が掛かった
  - 現象の再現に時間が掛かったか
  - 緊急性の高い脆弱性の届出で混み合っていたか
- 運営元が脆弱性届出窓口を公開しているので、今後はそちらに届出したい
- 著名なソフトウェアの脆弱性ということで、世界的に注目された点が個人的には良かった

4-3. 脆弱性を発見する際に注意している点

- 法令等に違反しないこと
- 開発元/サイト運営者や利用者に迷惑や被害を与えないこと
- 秘密保持（ゼロデイ脆弱性を公開しない）
- 脆弱性発見に至った手順や証跡を記録
- 脆弱性検証に用いた環境も保持して、再現制を確保する
  （検証環境をそれぞれVMとして保存）

4-4. IPAの届出制度への届出をして良かった点/大変だった点

- 良かった点
  - 海外製品の場合でも日本語で報告できる
  - 連絡窓口を探したり、脆弱性の詳細の説明を省ける場合がある
  - CVEが採番される場合が多かった
- 大変だった点
  - 届出の後製造元等に連絡されるまでに時間が掛かる場合が多い

4-5. IPA以外の組織への届出をして良かった点/大変だった点

- 良かった点
  - スピーディな対応ができる
  - 開発元との細やかなコミュニケーションが取りやすい
- 大変だった点
  - 脆弱性報告窓口を探すのが面倒
  - 英語による報告が面倒

4-6. これから脆弱性を発見しようとする方々に伝えたいこと

- 脆弱性を発見したら適切な方法で届出をしてインターネットの安全に貢献しましょう
- 脆弱性発見は知的好奇心を刺激する楽しい行為でもあります
- 正義感ゆえの行動のはずが、届出後に思った通りにいかないと、"歪んだ正義感"になる例を見てきました
  - 例: ゼロデイ脆弱性を公表してしまう
- あくまで正しい方法で脆弱性情報を届出しましょう