2025

脆弱性診断内製化ガイド

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

中核人材育成プログラム 8 期生　脆弱性診断内製化ガイド作成プロジェクト

改訂履歴

改訂年月日	改訂箇所	改訂内容
2025 年 7 月 31 日	-	初版公開

用語集（A-Z 順 -> あいうえお順）

用語	意味
API (**A**pplication **P**rogramming **I**nterface)	ソフトウェア同士が機能やデータをやり取りするための接続仕様のこと。アプリケーションが他のシステムの機能を呼び出す際の「窓口」となるもので、例えば天気情報の取得や決済機能の連携などに使われる。API を活用することで、開発効率の向上やサービス間の連携が容易になる。
CSRF (クロスサイト・リクエスト・フォージェリ)	ログイン中のユーザの権限を悪用して、意図しない操作を実行させる攻撃手法。攻撃者は別の Web サイトなどを通じて、不正なリクエストをユーザのブラウザから送信させることで、知らぬ間に設定変更や情報送信などを行わせる。
CTF (**C**apture **T**he **F**lag)	情報セキュリティの分野で、隠されている Flag（答え）を専門知識や技術を駆使して見つけ出し、時間内に獲得した合計点数を競うハッキングコンテストを指す。クイズ形式の問題を解くほか、ネットワーク内で疑似的な攻防戦も行ったりすることもある。
CVE (**C**ommon **V**ulnerabilities and **E**xposures)	ソフトウェアやシステムに存在する脆弱性を一意に識別するための共通番号（ID）を付与する仕組み。CVE 番号により、世界中で公開される脆弱性に関する情報を標準化して管理・共有でき、脆弱性の影響範囲や対策状況を効率よく把握することが可能になる。
DevOps	「Development（開発）」と「Operations（運用）」を組み合わせた言葉で、ソフトウェア開発と運用が密に連携し、継続的かつ効率的にシステムを提供・改善する考え方を指す。自動化やチーム間の協働を重視し、素早いリリースと品質の向上を両立させる手法として広く採用されている。
DevSecOps	「Development（開発）」「Security（セキュリティ）」「Operations（運用）」を統合した考え方で、開発・運用のプロセスにセキュリティを最初から組み込むアプローチを指す。継続的な開発・リリースの中でセキュリティチェックを自動化・定常化し、リスクを早期に発見・対処することで、スピードと安全性の両立を図る。
IDS (**I**ntrusion **D**etection **S**ystem)	ネットワークやシステムに対する不正アクセスや攻撃の兆候を検知するためのシステム。パケットの内容やログを監視し、既知の攻撃パターンや異常な振る舞いを検出する。リアルタイムでの監視が可能で、管理者へ通知することで早期対応を促すが、単独では防御機能を持たないため、他のセキュリティ対策と併用するのが一般的である。
OT (**O**perational **T**echnology)	製造業やインフラなどで使われる機械や装置を制御・監視するための技術を指す。生産ラインの制御装置や電力・水道の制御システムなどが該当し、リアルタイム性と安全性が重視される。近年は IT との連携が進み、セキュリティ対策の重要性が増している。
SAST (**S**tatic **A**pplication **S**ecurity **T**esting)	ソースコードやバイナリコードを実行せずに静的に解析し、脆弱性を検出する手法。開発段階から適用可能で、SQL インジェクションやバッファオーバーフローなどのセキュリティ欠陥を早期に発見できる。修正コストの低減や品質向上に役立つ。

SIEM (**S**ecurity **I**nformation and **E**vent **M**anagement)	ネットワーク機器やサーバ、アプリケーションなどのログを一元的に収集・分析し、異常検知やインシデント対応を支援するシステム。脅威の迅速な可視化や過去のログからの原因追跡が可能で、セキュリティ運用の効率化と高度化に貢献する。
SPA (**S**ingle **P**age **A**pplication)	Web ページ全体を再読み込みせずに、必要なデータだけを動的に取得・表示するアプリケーションの構築手法。画面遷移がスムーズで、ユーザ体験に優れる反面、初回読み込みに時間がかかる場合がある。JavaScript による動的処理が中心で、代表的な実装には React や Vue.js などがある。
SQL インジェクション (**S**tructured **Q**uery **L**anguage インジェクション)	Web アプリケーションの入力欄などに悪意ある SQL 文を埋め込むことで、データベースを不正に操作する攻撃手法。適切な入力チェックやパラメータ化されていないクエリが原因で発生し、情報漏えいやデータ改ざん、管理者権限の奪取といった深刻な被害につながるおそれがある。
VPN (**V**irtual **P**rivate **N**etwork)	インターネットなどの公衆回線を通じて、安全な通信経路（トンネル）を確立し、遠隔地から社内ネットワークなどに安全にアクセスできる技術。通信内容は暗号化されるため、第三者による盗聴などのリスクを軽減できる。リモートワークや拠点間通信のセキュリティ確保に広く活用されている。
XSS (クロスサイト・スクリプティング)	Web アプリケーションの入力値検証の不備を突いて、悪意のあるスクリプトをユーザのブラウザ上で実行させる攻撃。これにより、Cookie の不正取得や不正リダイレクト、フィッシングなどが発生する可能性がある。ユーザが信頼する Web サイト上で攻撃が行われるため、被害に気づきにくい特徴がある。適切な入力値の検証やエスケープ処理が対策として有効である。
脅威インテリジェンス	サイバー攻撃に関する情報（攻撃手法、攻撃者の動機、標的、関連するマルウェアなど）を収集・分析し、組織の防御力を高めるために活用する知見を指す。過去の攻撃事例やリアルタイムの脅威動向をもとに、将来の攻撃予測や対策方針の策定に役立てられる。セキュリティ運用の判断材料として重要な役割を担う。
セキュアコーディング	ソフトウェア開発において脆弱性を作り込まないように、安全性を考慮した実装を行う手法。入力値の検証やエラーハンドリング、認証・認可の適切な処理などが含まれる。脆弱性を未然に防ぐことで、攻撃リスクの低減と保守性の向上につながる。
バグバウンティプログラム	企業や組織が自社のシステムやサービスに存在する脆弱性を発見してもらうために、外部のセキュリティ研究者やホワイトハッカーに報奨金を支払う制度。公募型の脆弱性診断とも言え、実際の攻撃者視点からの検証が可能になる。発見された脆弱性は修正に活かされ、セキュリティの向上に寄与する。
ビジネスロジック	アプリケーションにおいて、業務上のルールや手順、データの処理方法などを定めた中心的な処理部分のこと。例えば、EC サイトにおける商品の価格計算や在庫管理、金融システムの利息計算などが該当する。このロジックの設計や実装に不備があると、割引の不正適用や想定外の取引の成立といった脆弱性につながることがある。
ファイアウォール	ネットワークの出入口に設置し、不正なアクセスや不要な通信を制御するセキュリティ機器またはソフトウェア。ルールに基づき通信を監視・制限することで、マルウェア侵入や情報漏えいのリスクを低減する。ファイアウォールにはホスト型とネットワーク型があるが、本ガイドでは主にネットワーク型を対象とする。

目次

1 はじめに 1

1.1 背景 1

1.2 本ガイドの目的 3

1.3 想定読者・適用範囲 3

1.4 本ガイドの構成と読み方 3

1.5 利用規約 4

2 脆弱性診断について 6

2.1 脆弱性とは 6

2.2 脆弱性診断とは 6

2.3 脆弱性診断の対象範囲と位置付け 6

2.4 脆弱性診断の種類 7

2.5 脆弱性診断の手法 9

2.6 診断アプローチと選定のポイント 12

2.7 セキュリティ全体における脆弱性診断の位置づけ 12

2.8 本章のまとめ 13

3 外部発注と内製の違い 15

3.1 本章の目的 15

3.2 外部発注と内製を考えるうえでの基本的な視点 15

3.3 内製化の特徴と考慮点 15

3.4 外部発注の特徴と考慮点 17

3.5 外部発注と内製のコスト構造の違い 18

3.6 外部発注と内製のメリット・デメリット 20

3.7 ハイブリッド運用という選択肢 20

3.8 本章のまとめ 21

4 内製化に必要な組織体制と人材 23

4.1 本章の目的 23

4.2 経営層の関与と推進体制 23

4.3 内製チームの役割とブランディング 23

4.4 チーム構成と役割 24

4.5 組織横断的なコミュニケーションと関係組織連携 27

4.6 本章のまとめ 28
5 内製化の進め方と継続的改善プロセス 29
5.1 本章の目的 29
5.2 事前に決めておくべきこと 29
5.3 ステップ 1：スモールスタートによる段階的な運用開始 34
5.4 ステップ 2：手動診断の段階的な導入 34
5.5 ステップ 3：全社・グループ展開 35
5.6 品質向上と継続的改善 35
5.7 本章のまとめ 37
6 関係組織との連携とセキュリティ意識の醸成 38
6.1 本章の目的 38
6.2 仮想企業における組織モデルの概要 38
6.3 開発時のリリース前診断 39
6.4 運用中の定期診断 40
6.5 組織間セキュリティ意識の醸成 41
6.6 本章のまとめ 41
7 人材確保・育成 43
7.1 本章の目的 43
7.2 人材確保の基本的な考え方 43
7.3 新卒・中途採用のポイント 44
7.4 人材育成 45
7.5 モチベーション維持とキャリアパス 46
7.6 本章のまとめ 47
8 ツール選定におけるポイント 48
8.1 本章の目的 48
8.2 有償ツールと無償ツール 48
8.3 AI・自動化技術の動向 48
8.4 本章のまとめ 49
9 謝辞 50

付録 A：技術検証結果について.....51
A.1 概要.....51
A.2 VulnHub 環境１に対するプラットフォーム診断.....52
A.3 VulnHub 環境２に対するプラットフォーム診断.....54
A.4 BadTodo に対する Web アプリケーション診断.....56
A.5 検証で得られた知見.....57

1 はじめに

1.1 背景

企業システムに対するサイバー攻撃は年々多様化・高度化している。さらに、CVE 付与数の年別推移（図 1-1[1]）が示すように、世の中のソフトウェアに新たに確認される脆弱性の件数も増加傾向にある。このような状況下で、企業システムに潜む脆弱性を迅速に把握し、適切な対策を施す必要性が年々高まっている。脆弱性診断は、運用中やリリース前のシステムに存在するセキュリティ上の欠陥や設定ミスを検出し、被害を未然に防ぐための重要な手段の一つである。

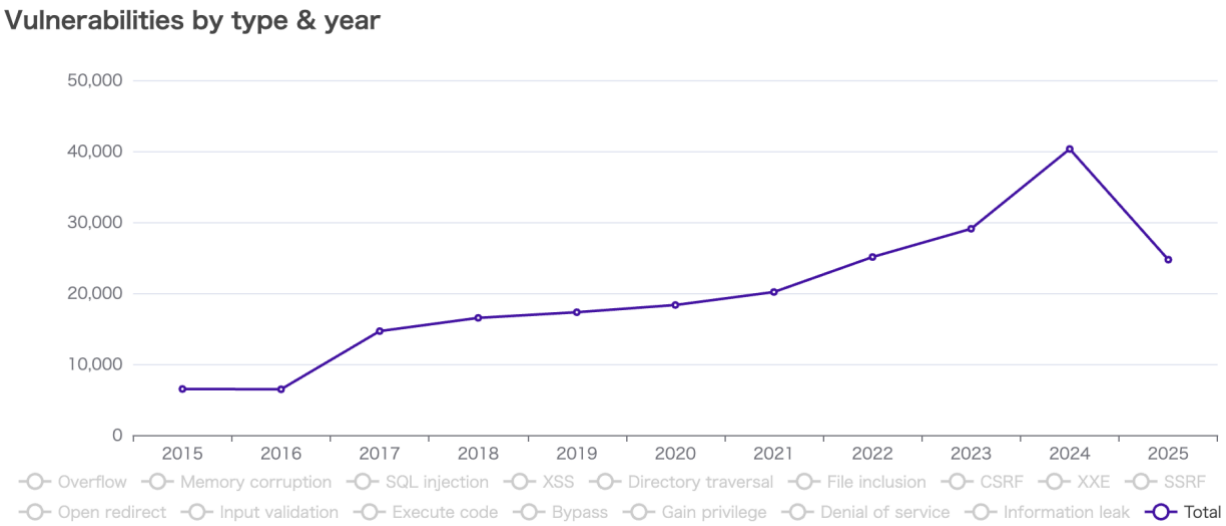

図 1-1 CVE 付与数の推移（2025 年 7 月現在）（CVE details）

一方、近年企業では DX（デジタルトランスフォーメーション）の推進とともに、IT システムを内製化する動きが強まっている。例えば、独立行政法人情報処理推進機構（IPA）が公表した「DX 動向 2024」での調査結果（図 1-2[2]）によれば、内製によるシステム開発の割合は 2022 年度から 2023 年度にかけてすべての領域で増加している。この結果も含め、IT システム全般で外部委託から内製へのシフトが加速していることがうかがえる。

[1] CVE details https://www.cvedetails.com/

[2] 「DX 動向 2024」/ 独立行政法人情報処理推進機構（IPA）/ P.23 より
https://www.ipa.go.jp/digital/chousa/dx-trend/eid2eo0000002cs5-att/dx-trend-2024.pdf

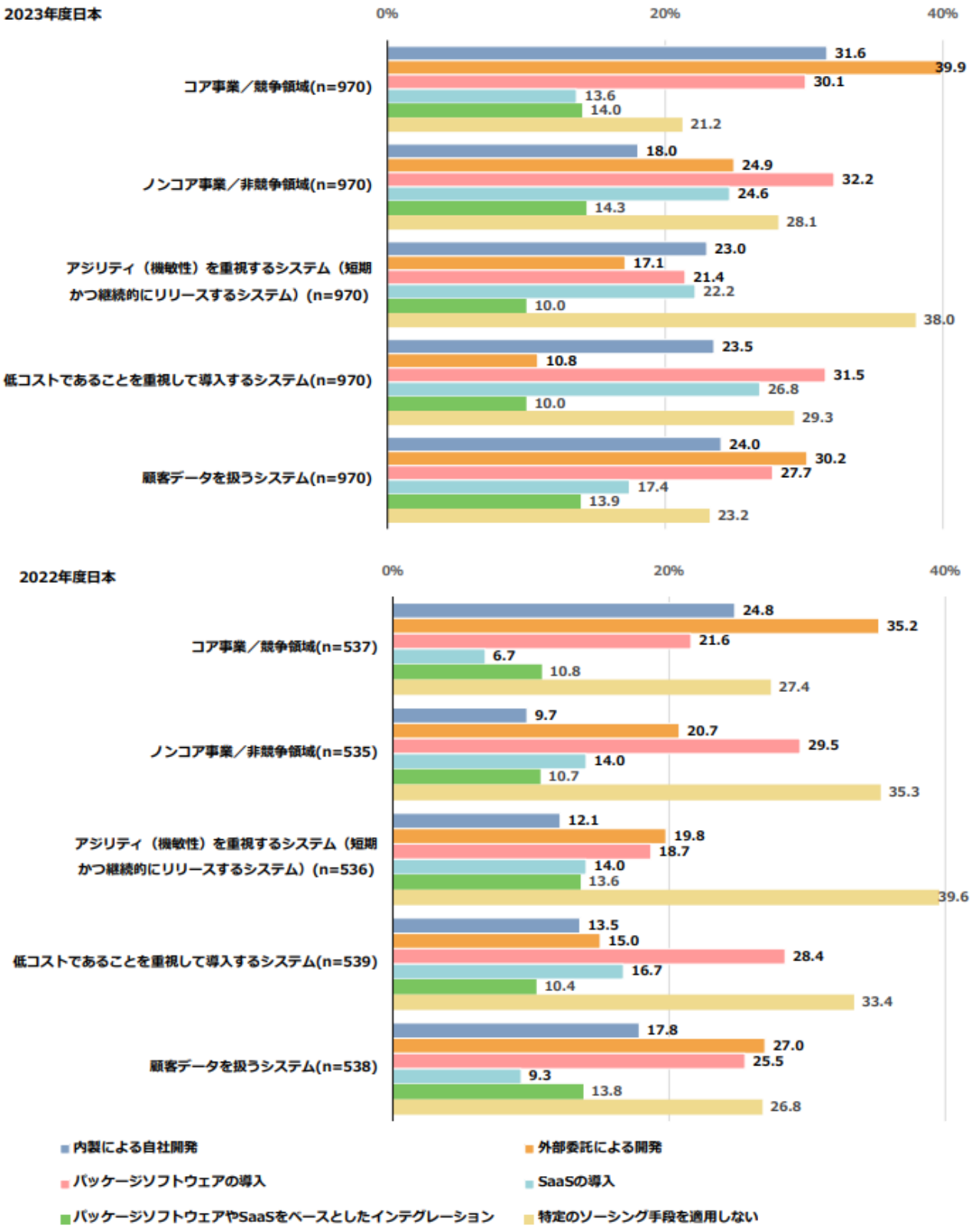

図1-2 ソーシング手段の状況（経年変化）（IPA）

こうした内製化の流れの中で、脆弱性診断に関しても外部発注から内製への切り替えを検討するケースが見受けられる。多くの企業が外部委託時の課題として指摘するのは、「診断を外部発注しているがコスト負担が大きい」、「システムごとに診断タイミングを柔軟に調整しづらい」、「社内情報を外部ベンダーに共有することへの抵抗がある」といった点であ

る。一方で、脆弱性診断を内製化するには専門的な技術力の確保や組織的な体制の整備が不可欠であり、人材の確保と育成、診断手法の標準化、品質管理など、乗り越えるべき課題は多岐にわたる。

1.2 本ガイドの目的

本ガイドでは、脆弱性診断の内製化を進めるうえでの基本的な考え方や導入ステップ、関連組織との連携方法を具体的に整理することで、内製化を検討する企業を支援することを目的としている。読者が自社の開発・運用体制やリソースに合わせて、どのように診断体制を構築・運用すればよいかを具体的にイメージできるように支援する構成としている。

脆弱性診断の内製化は、あくまでもセキュリティ向上という目的を達成するための手段であり、内製化をゴールにして重要なリスクや課題を見失ってはならない。自社が抱える問題を正しく把握したうえで、ビジネス環境に即した診断体制を構築できるよう、本ガイドがその一助となれば幸いである。

1.3 想定読者・適用範囲

本ガイドは、ユーザ企業が自社で保有・運用するシステムに対し、脆弱性診断を内製化するケースを想定したものであり、主な読者として以下の担当者・管理者を対象とする。

- 脆弱性診断の方針を検討するセキュリティ担当者・管理者
- 脆弱性診断の内製化の立ち上げを担うセキュリティ担当者・管理者
- 脆弱性診断を内製化済みであるものの組織運営に課題を抱える企業の担当者

方針検討および決定を担う層が把握すべきポイントを中心にまとめているため、全体を通じて内製化の概要をつかめる構成としている。

本ガイドの適用範囲は以下とする。

- プラットフォーム診断
- Web アプリケーション診断

これらの診断は一般的な企業環境で共通して実施される脆弱性診断である。オンプレミス環境だけでなく、クラウド上で動くシステムや、OT システムの診断にも応用できるよう基本的な考え方を含んだ構成としている。

なお、脆弱性診断の技術的詳細や実務的な診断手順については、本ガイドの対象外とする。これらの内容については、専門書籍やツールベンダーの公式ドキュメント、専門的なトレーニングプログラムなどを参照いただきたい。

1.4 本ガイドの構成と読み方

本ガイドは以下の流れで章を構成し、脆弱性診断の概要から内製化の進め方、そして関係組織との連携までを幅広く説明する。

- **「2 脆弱性診断について」** では、脆弱性診断の基本概念や種類、診断手法、およびその位置付けについて説明する。

- **「3 外部発注と内製の違い」** では、外部発注と内製それぞれの特徴や考慮点を整理し、どちらの方法を選ぶのが適切か判断するための材料を提供する。
- **「4 内製化に必要な組織体制と人材」** では、経営層の関与や推進体制、脆弱性診断チームの役割・構成、そして各関係組織との連携方法などについて説明する。
- **「5 脆弱性診断内製化の進め方と継続的改善プロセス」** では、段階的な導入ステップや事前に決めておくべき方針、継続的な品質向上・改善の仕組みについて説明する。
- **「6 関係組織との連携とセキュリティ意識の醸成」** では、システム開発時・運用時の診断プロセスにおける組織連携の在り方や、セキュリティ意識を高めるための具体的な施策について説明する。
- **「7 人材確保・育成」** では、脆弱性診断に必要な人材の採用戦略やスキル育成、モチベーション維持とキャリア設計など、人材面のアプローチについて説明する。
- **「8 ツール選定におけるポイント」** では、脆弱性診断ツールの概要や選定基準、活用上の留意点を説明する。

本章で記した背景や目的を踏まえたうえで、第 2 章以降を順次読み進め、適切な脆弱性診断体制を検討するきっかけとしていただきたい。

1.5 利用規約

1.5.1 著作権及びその他すべての知的所有権

「脆弱性診断内製化ガイド（以下、「本ガイド」）」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 ～脆弱性診断内製化ガイド作成プロジェクト～（以下、「本プロジェクト」）」に帰属する。

1.5.2 免責事項

- 本ガイドは単に情報として提供され、内容は予告なしに変更される場合がある。
- 本ガイドに誤りがないことの保証や、商品性または特定目的への適合性を含め、いかなる明示的・黙示的な保証や条件も行われない。
- 本プロジェクトは、本ガイドの品質的・法律的いずれについても一切の保証をしない。また、本ガイドの使用に起因して生じる一切の直接的、間接的、付随的または結果的損害や利益の損失などについて、法的原因の如何にかかわらず、いかなる責任も負わない。
- 本ガイドには外部のサイトへのリンクが含まれているが、リンク先の内容や安全性について本プロジェクトおよび本ガイド著者ならびに監修者は一切関知しない。
- 本ガイドの有効期限は、発行日から 2 年間とする。

1.5.3 注意事項

- 本ガイドは本プロジェクトの見解に基づいて作成されたものであり、独立行政法人情報処理推進機構（IPA）および本プロジェクトメンバー（以下、メンバー）の所属企業の見解を反映するものではない。
- メンバーの一部は、内製化された企業で脆弱性診断業務に携わっているが、内製化を経験したメンバーはいない。そのため、専門家や内製企業へのヒアリングおよび各種文献の調査を通じて得られた知見を取り入れるとともに、プロジェクト内で全く知識のない段階から、診断手順の実践までを模擬的に進める中で得た経験も本ガイドに取り入れている。
- また、メンバーが自社業務において組織運営・システム開発・脆弱性診断などに携わった経験に加え、中核人材育成プログラムの講義や演習を通じて得た知見をもとに内容を補完している。

1.5.4 利用条件・範囲

本ドキュメントは企業または団体が自社の業務において活用することを目的として提供している。ただし、以下の行為は禁止とする。

- 本ドキュメントの全てまたは一部を書籍や雑誌、Web メディア等の出版物に転載、再配布、販売すること。
- 販売や宣伝を目的とした教材、セミナー資料、コンテンツ商品への転用。
- 本ドキュメントの内容を第三者に販売または再配布する行為（有償・無償問わず）。

以上の各項目をご了承いただいたうえで、本ガイドの活用をお願いする。

2 脆弱性診断について

2.1 脆弱性とは

脆弱性とは、もともと「外部からの影響に対して脆い状態」を意味し、心理面や経済面などサイバーセキュリティ以外でも使われる汎用的な言葉である。

本ガイドで取り扱う脆弱性は、コンピュータの OS やソフトウェアにおいて、プログラムの不具合や設計上のミスが原因となって発生するサイバーセキュリティ上の欠陥を指す。脆弱性が存在すると、第三者がその脆弱性を悪用して、情報漏えいや不正アクセス、システムの乗っ取り、サービス停止（DoS 攻撃）などの被害を引き起こされる可能性がある。また、脆弱性は完全に対策を施すことが困難であり、次々と新たな脆弱性が発見されているのが現状である。

2.2 脆弱性診断とは

脆弱性診断とは、システムやアプリケーションに潜む脆弱性を洗い出すために、さまざまな手法を用いて診断を行い、その結果を報告書として提示するプロセスである。診断対象のシステムに対し、データベース化およびナレッジ化された脆弱性の検査リストに照らし合わせ、網羅的にチェックを行い、リスクを明確化することで、開発や運用の段階で適切な修正措置を講じられるよう支援する。 脆弱性診断は、ネットワークや OS などの基盤（プラットフォーム）を対象とした診断や Web アプリケーションの動的挙動を対象とした診断など、対象範囲によって着目点や検証手法が異なる。また、脆弱性診断は、システムの新規構築時や、機能の追加・更改に実施する「リリース前診断」とシステムの運用開始後に定期的に実施する「定期診断」の 2 つに分けられる。「リリース前診断」は、開発したシステムがセキュリティ仕様に適合しているかを確認するのに対し、「定期診断」は、稼働中の構成ソフトウェアに関する新たな脆弱性や最新の脅威動向の変化を踏まえ、各システムの脆弱性対策が適切に実施されていることの点検や監査を目的としたものである。

2.3 脆弱性診断の対象範囲と位置付け

脆弱性診断は、システムやアプリケーションに潜む脆弱性のうち、「発生が想定される脆弱性」を洗い出すことを目的としている。図 2-1[3] に示すとおり、「システムにおける脆弱性全体（U）」の中でも、開発・運用の関係者が過去事例や一般的な攻撃手法から発生を予測できるものを 「発生の想定される脆弱性（A）」と呼ぶ。さらに （A）のうち、設計要件やセキュアコーディング規約などで対策が求められている部分は「仕様や要件としてカバーしている脆弱性（B）」に区分される。

[3] 「政府情報システムにおける脆弱性診断導入ガイドライン（2024）」 / デジタル庁 / P.5 より / https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/e2a06143-ed29-4f1d-9c31-0f06fca67afc/b08708cd/20240131_resources_standard_guidelines_guidelines_05.pdf

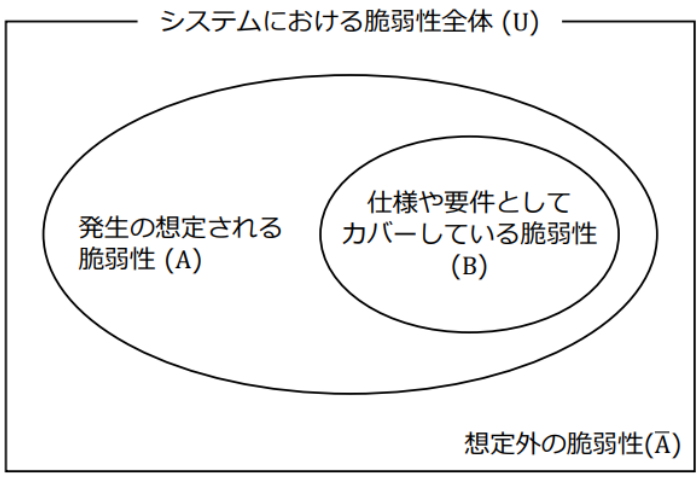

図 2-1 システムにおける脆弱性の分類 (デジタル庁)

脆弱性診断がフォーカスするのは、（A）の領域全体である。これは、（B）ではカバーされていない実装上の不備や考慮漏れを発見することに加え、仕様・要件で対策が求められている（B）の領域が、意図通りに実装されているかを第三者の視点で検証する目的がある。さらに、診断は単にその場の脆弱性を指摘するだけでなく、発見された問題を（B）の設計要件やセキュアコーディング規約にフィードバックし、開発プロセスそのものを改善していく循環的な活動と位置づけられる。これにより、将来的な脆弱性の発生を未然に防ぎ、場当たり的な対応からの脱却を目指す。

一方、未知の攻撃手法や新技術に起因する「想定外の脆弱性 （Ā：A 以外の領域）」までを網羅的に検出することは難しいため、バグバウンティプログラムや脅威インテリジェンスの活用など、別途補完策を組み合わせることが望ましい。

2.4 脆弱性診断の種類

脆弱性診断には多様な分類方法があるが、代表的な分類方法として、以下のような区分が用いられる。ここでは、プラットフォーム診断と Web アプリケーション診断を中心に、その概要と目的を整理する。

2.4.1 プラットフォーム診断

- **概要**
  主に OS やミドルウェア、ネットワーク機器など、システムを構成する基盤部分を対象に行う診断である。公開サーバやネットワーク内部のサーバ、ルータ、ファイアウォール（以下、FW）などをチェックし、既知の脆弱性や設定不備を検出する。
- **目的**
  パッチ適用状況や不要なポートの公開、有効化されていないセキュリティ機能の存在などを洗い出し、脆弱性やバージョン情報の公開、詳細なエラーメッセージ出力などを特定する。近年では VPN や FW などの境界防御機器の脆弱性 が大規模に悪用さ

れる事例も増えており[4]、プラットフォーム診断によって、攻撃の侵入経路を事前に発見し対策につなげることができる。

- **主な指摘事項の例**
  - OS やミドルウェアのパッチ未適用
  - 不要なポートの公開や古い暗号化プロトコルの使用
  - デフォルト認証情報の放置
  - 不適切なアクセス制御設定
- **診断における考慮点**

  図 2-2 が示すように、外部ネットワークから診断対象に対して脆弱性診断を実施する場合、FW 等のネットワーク機器を介して通信するため、これら機器の制御ポリシーや設定内容も含めて診断結果に反映される。したがって診断計画を立てる際には、ネットワーク機器の存在とその設定内容を事前に把握しておくとともに、診断の目的に応じて診断通信が対象に到達するよう一時的に FW の設定を変更するなど、ネットワーク構成の考慮を含めた診断設計が求められる。

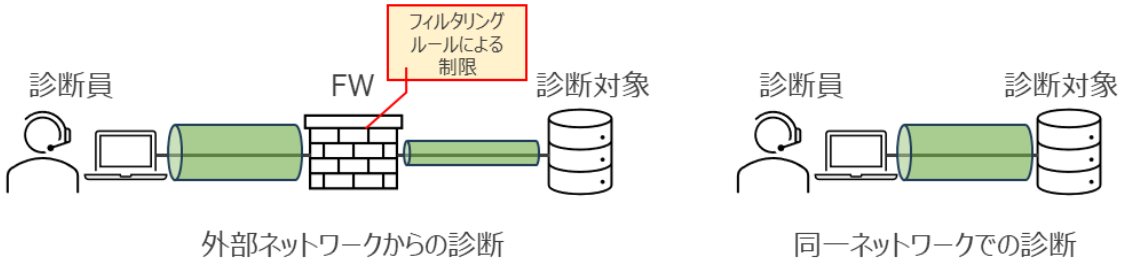

図 2-2 プラットフォーム診断における考慮点

2.4.2 Web アプリケーション診断

- **概要**

  Web サーバ上で稼働するアプリケーションを対象に、主に HTTP/HTTPS でアクセスし、入力パラメータの扱いや認証・セッション管理の設計などを検証することで、脆弱性の有無を確認する診断方法である。動的にページ生成を行う仕組みやフロントエンドの JavaScript 処理、API 連携など、多岐にわたる要素を含むため、診断員のナレッジや経験に基づいた手動検証が重要となる領域である。
- **目的**

  SQL インジェクションやクロスサイト・スクリプティング（XSS）、認証回避、クロスサイト・リクエスト・フォージェリ（CSRF）など、Web 特有の脆弱性を洗い出すことが目的である。これにより、ユーザ情報の漏えい・改ざんやシステムの不正利用等を防止する。
- **主な指摘事項の例**

---

[4] インターネット境界に設置された装置に対するサイバー攻撃について～ネットワーク貫通型攻撃に注意しましょう～ / 独立行政法人情報処理推進機構（IPA） / https://www.ipa.go.jp/security/security-alert/2023/alert20230801.html

- 入力値検証の不備、インジェクション系脆弱性
- 認証、セッション管理の不備（セッショントークンの漏えい・再利用可能性など）
- アクセス制御の誤実装や、ユーザ権限の不適切な管理
- ビジネスロジックの穴（例えば EC サイトでは、割引計算やオプション追加などの悪用）
- OWASP Top 10[5]や IPA「安全なウェブサイトの作り方[6]」などに記載の脆弱性

- **診断における考慮点**
  Web アプリケーション診断を実施する際、システム負荷増加やデータ変更が起こり得るため、本番環境ではなく検証環境を用意し、安全に診断を行える体制を構築することが望ましい。また、動的生成ページや JavaScript を多用する SPA（Single Page Application）などでは、ツールの検出漏れが発生しやすいため、手動診断との併用が不可欠である。

2.4.3 その他の診断

プラットフォーム診断や Web アプリケーション診断以外にも、モバイルアプリケーション診断やソースコードの静的解析（SAST）、クラウド環境固有の設定診断など、様々な診断の種類が存在する。しかし、多くの企業システムではプラットフォーム診断と Web アプリケーション診断から着手することが一般的であり、本ガイドもこれらを主眼に説明している。業務フローや危険度評価など基本的な考え方や手順は他の診断でも共通する部分が多いため、運用の実情に合わせて応用を検討するとよい。

2.5 脆弱性診断の手法

脆弱性診断では、自動化された診断ツールを活用する手法と、手動診断を組み合わせるのが一般的である。それぞれに長所と短所があるため、対象システムの特性や重要度に応じてバランスを考慮する必要がある。

2.5.1 ツール診断

- **概要**
  診断ツールを用いて、既知の脆弱性や標準的な誤設定を網羅的にチェックする。シグネチャやルールベースで検知を行うため、定常的な診断を自動で実施しやすい点が特徴である。
- **ツールの種類と特徴**
  脆弱性診断で用いられるツールも、前述（2.4 脆弱性診断の種類）と同様に、プラットフォーム診断向けと Web アプリケーション診断向けのツールが存在する。

---

[5] OWASP Top 10 / https://owasp.org/Top10/ja/

[6] 安全なウェブサイトの作り方 / 独立行政法人情報処理推進機構（IPA）/ https://www.ipa.go.jp/security/vuln/websecurity/about.html

- **プラットフォーム診断ツール**
  プラットフォーム診断ツールは、サーバやネットワーク機器に対しポートスキャンやバナー情報の取得などを行い、既知の脆弱性や設定ミスを自動でチェックする機能を備えている。Nessus や OpenVAS などが代表的な選択肢として挙げられる。これらのツールは、データベースに蓄積された CVE 情報等を参照してシステムのバージョンや設定を照合し、脆弱性を含む疑いのある資産やその脆弱性の詳細を一覧化するものである。プラットフォーム診断ツールのメリットは、手動で確認しづらい多数のホストに対して広範囲にスキャンを行い、パッチ未適用や古い暗号化プロトコルの使用などを効率よく発見できる点である。一方で、誤検知も発生しやすく、実際には問題ない箇所を高い危険度と報告してしまうケースもあるため、診断対象の環境下で成立するかを考慮したうえで、最終的な精査は診断員が行う必要がある。

- **Web アプリケーション診断ツール**
  Web アプリケーション診断ツールは、HTTP リクエストやレスポンスを自動で解析し、SQL インジェクションやクロスサイト・スクリプティング（XSS）などの典型的な脆弱性を検知するものである。Burp Suite や ZAP といったツールが広く使われており、それぞれに一長一短の機能やサポート体制が存在する。多くのツールはまずクローリング機能を用いてアプリケーション内のリンクをたどり、ページ遷移を自動で探索する。次に、クローリングで取得した各ページに対して入力パターンを自動生成し、そのリクエストを送信して脆弱性の有無をスキャンすることで、一定の網羅性を確保した診断を行う。ただし、ビジネスロジックに深く依存する脆弱性（状態遷移の不整合など）や複雑な認証・権限管理の問題については自動検出が難しく、そういった脆弱性の発見には、ビジネスロジックの理解にもとづく手動診断が不可欠となる。

- メリット
  - スキャン範囲が広く、レポート出力までを短時間で行える
  - 多数のポートや設定ミス，セキュアコーディングの不備などを機械的に検知可能
  - スキャン結果の履歴管理がしやすく、定期的な比較を行う際に有効

- **課題**
  - 誤検知や未検出が発生する場合があるため、最終的な結果の妥当性評価を人が行う必要がある
  - ビジネスロジックに起因する脆弱性などは発見しづらい

2.5.2 手動診断

- **概要**
  診断員が実際にアプリケーションやシステムにアクセスし、さまざまな入力パターンを試行することで、ツールでは検知が難しい脆弱性を掘り起こす手法である。特に Web アプリケーションでは、開発者が想定していない利用パターンを探るケースが多い[7]。

- メリット

[7] Business logic vulnerabilities / PortSwigger / https://portswigger.net/web-security/logic-flaws

- アクセス制御不備やビジネスロジック上の問題、システム設計固有のセキュリティ欠陥を発見しやすい
- ユーザの操作フローに沿った脆弱性検証や複数の脆弱性を組み合わせることでより深刻な影響をもたらす問題の発見が可能
- 対象環境に合わせた現実的な危険度評価が可能

- **課題**
  - 診断員のスキルに左右されやすい
  - 診断に工数がかかるため、システムの全体を網羅するには負荷が高い
  - 自社の診断員が診断をする場合、攻撃者視点を十分に保つための訓練や定期的な知識のアップデートが必要

2.5.3 ツールと手動の組み合わせ

診断の実務では効率的かつ精度の高い診断を実現するため、まずはツール診断で広範かつ効率的なスキャンを行い、誤検知が疑われる部分やツールで検出することが難しい部分を中心に手動診断でカバーするパターンが一般的である[8]。特に Web アプリケーション診断では、ツールで表面的なインジェクションやパラメータ異常を検知した後、ビジネスロジックを診断員が読み解くことで深刻な攻撃シナリオを見つけ出す、といった流れが定着している。自社内での内製化を進める際も、導入しやすいツールの活用を起点にしながら、診断員のスキル向上と手動診断のノウハウ蓄積を並行して進めるのが望ましい。

2.5.4 コスト面での考慮事項

脆弱性診断を内製化する際には、コスト面の評価も重要な検討項目となる。一般的に、ツール診断は初期導入費用やライセンス費用はかかるものの、一度環境を構築すれば定期的かつ広範囲な診断を低コストで実施できる。

手動診断は担当者の人件費や教育費用、診断にかかる工数が多いため、費用面での負担が大きくなる傾向がある。しかし、ツール診断のみではビジネスロジックや固有の設定に起因する複雑な脆弱性を見逃す可能性があるため、求める脆弱性診断の品質次第では、手動診断のコスト負担が生じる点も留意が必要である。

こうしたツールや診断ノウハウは、IT 領域だけでなく産業制御システム（OT）など他の領域でも共有・活用できるため、類似したツール導入や人材育成にかかる投資の重複を避けることも期待できる。

これらのコスト要素やメリットを理解した上で、自社の規模や業務特性、予算状況などを総合的に勘案し、最適な脆弱性診断の組み合わせを選択していくことが重要となる。

[8] Web セキュリティ担当者のための脆弱性診断スタートガイド / 上野 宣 著 / P.113 より

2.6 診断アプローチと選定のポイント

脆弱性診断では、種類と手法以外にも診断アプローチの考え方もある。診断員が利用できる情報の範囲に応じて、「ブラックボックス診断」「ホワイトボックス診断」「グレーボックス診断」の 3 つに分類され、それぞれの特徴は以下の通りである。

- **ブラックボックス診断**
  攻撃者と同じ状況下で診断を行うアプローチであり、システム内部の情報を一切取得せずに診断を行う。攻撃の実現性やリスクを客観的に判断しやすいが、実際の攻撃者に比べ診断員は時間やリソースに制約があるため、他のアプローチに比べ脆弱性の見落としが発生しやすいともいわれている[9]。
- **ホワイトボックス診断**
  ソースコード、詳細仕様書、実装仕様書など詳細な内部情報を診断員に提供したうえで診断を行う。より精度が高く網羅的な診断が可能だが、診断で確認する情報が多いことから、実施に時間を要する。
- **グレーボックス診断**
  ホワイトボックス診断とブラックボックス診断の中間に位置するアプローチで、一部の内部情報（例えば、ネットワーク構成、仕様書、設定情報など）をもとに診断を行う。これにより通常のブラックボックス診断では発見が困難な脆弱性を特定することができる。ある程度の情報を診断員に提供することで、効率的な診断が可能となる。

一般的に「脆弱性診断」といった場合は、攻撃者と同じ状況下でのブラックボックス診断を指すことが多い。

表 2-1 に各診断アプローチの特徴をまとめた比較表を示す。

表 2-1 診断アプローチの特徴

診断アプローチ	特徴や利点	必要なスキル	診断にかかる時間	共有情報の例
ブラックボックス診断	攻撃者の視点に最も近く、攻撃の実現可能性を評価しやすい	攻撃者視点での推測力や発想力	中	URL、IPアドレス、認証アカウント情報
グレーボックス診断	共有情報に基づき的を絞った効率的かつ精度の高い診断が可能	限られた内部情報を活用し、効率的に攻撃箇所を特定するスキル	中	ネットワーク構成、仕様書、設定情報など内部情報の一部
ホワイトボックス診断	ソースコードから内部ロジックを把握し、複雑な脆弱性も発見できる	ソースコード解析能力	中～大	ソースコード、詳細仕様書、実装仕様書など

2.7 セキュリティ全体における脆弱性診断の位置づけ

脆弱性診断は、サイバー脅威からのシステム保護のための一要素に過ぎず、それだけであらゆるセキュリティリスクを排除できるわけではない。FW による境界防御や侵入検知システム（IDS）などの防御装置の設置、ログ監視や SIEM ツールの導入、従業員教育やインシデ

[9] 「What are black box, grey box, and white box penetration testing? [Updated 2020]」 / Infosec https://www.infosecinstitute.com/resources/penetration-testing/what-are-black-box-grey-box-and-white-box-penetration-testing/

ント対応訓練等、企業全体のセキュリティ水準を高める取り組みの中で、脆弱性診断は「システムの欠陥を見つけ出す」ための仕組みとして機能する。

また、脆弱性診断で指摘された脆弱性に対し、システム開発やシステム運用を担う組織が素早く修正を行わなければ、本来の効果は得られない。とりわけ近年は、DevOps を採用する企業が増加しており、高速リリースを前提とした開発体制が広がりつつある。これに伴い、セキュリティを開発プロセスに組み込む DevSecOps の考え方も注目されている。こうした環境下では、脆弱性診断を外部発注だけで運用する場合に生じるタイムラグやコスト負担が無視できなくなり、内製化の検討が一層重要になっている。

2.8 本章のまとめ

本章では、脆弱性診断の基本的な概念や種類、そして実際の診断手法やプロセスについて説明した。脆弱性とは、設計上のミスやプログラムの不具合により発生するシステム上の欠陥を指し、悪用されると情報漏えい、システム乗っ取りなど深刻な被害を引き起こす可能性がある。また、脆弱性診断とは、プラットフォーム診断（OS やミドルウェアなどの基盤部分）や Web アプリケーション診断を中心に、想定し得る脆弱性を網羅的に洗い出し、リスクを明確化して対策を促すプロセスである。

診断方法としては、診断ツールによるスキャンと手動による検証を組み合わせる形が一般的である。ツール診断は広範囲かつ効率的に脆弱性を検知できるが、誤検知やビジネスロジック固有の問題を見逃すリスクがある。一方、手動診断は、ビジネスロジックや複雑な仕様を踏まえた深い検証が可能な反面、工数や専門人材の確保といったコスト面の課題も生じる。また、診断アプローチとしては、攻撃者と同じ状況下でのブラックボックス診断に加え、ソースコードや設定情報などを一部あるいは詳細に提供されて実施するグレーボックス診断、ホワイトボックス診断があり、どの手法を選択するかはシステムの特徴や診断目的によって異なる。

さらに、脆弱性診断は企業のセキュリティ対策全体の一部であり、防御装置の導入やログ監視、従業員教育といった施策と合わせて進めることで総合的な効果を発揮する。診断で検出された脆弱性をいかに早く修正し、継続的に再検証を行うかが重要であり、特に DevOps をベースとした開発サイクルの高速化が進む中で、セキュリティを組み込んだ DevSecOps への取り組みが注目されている。

コラム

～脆弱性診断とペネトレーションテストはどう違うの？～

脆弱性診断とペネトレーションテストは、どちらも「システムのセキュリティを確認する」ための手法ですが、目的とアプローチが異なります。例えるなら、城を守るために行う「城の防御設備点検」と「攻城戦リハーサル」の関係です。

- **脆弱性診断**
  - **目的**：既知の脆弱性や設定ミスを網羅的に把握し、修正すべき項目を整理する。
  - **進め方**：自動スキャンで全体を広く調査しつつ、認証やビジネスロジックなどツールが苦手な領域は手動で深掘り。
  - **成果物**：各弱点の「影響度・原因・修繕策」を一覧化した報告書。
  - **タイミング**：リリース前やリリース後に定期的に実施。
  - **例**：石垣・門・見張り台・堀など「城の防御設備」をくまなく点検し、ひび割れや鍵の不具合を洗い出す。

- **ペネトレーションテスト**
  - **目的**：攻撃シナリオを組み立て、侵入・権限昇格・情報奪取などの目的が成立するかを実証する。脆弱性診断とは異なり、網羅的に脆弱性を発見するわけではない。
  - **進め方**：複数経路を組み合わせ、目的達成の可否に焦点を当てて検証。
  - **成果物**：成功した侵入経路、奪取した機密情報の証拠、防御側の対応の課題などを示すレポート。
  - **タイミング**：大規模リリース前や年次など、要所でのスポット実施。
  - **例**：点検で残った隙や運用の穴を使い、実際に攻め込んで「本丸に到達できるか」を試す攻城戦リハーサル。

脆弱性診断は、リリース前やリリース後、システム構成を変更した際に実施し、網羅的に脆弱性の有無を確認します。こうした診断を継続的に実施し、社内で脆弱性への対応プロセスやルールがある程度確立された状態になったうえで、重要なデータを扱うシステムや外部要件で実証的な検証が求められる場合にペネトレーションテストを実施し、残存リスクが実際に悪用可能かを評価するといった形が一般的です。

3 外部発注と内製の違い

3.1 本章の目的

本章の目的は、脆弱性診断の実施方法を選択するうえで、外部発注と内製それぞれのメリットおよびデメリットを整理し、企業が自社に適した実施方法を判断できるよう支援することである。そのために、両者の特徴を比較検討するための判断材料を提示するとともに、ハイブリッド運用という選択肢についても説明する。

3.2 外部発注と内製を考えるうえでの基本的な視点

脆弱性診断の実施方法は、企業の環境や目的に応じて慎重に選択する必要がある。外部発注と内製にはそれぞれ異なる長所と短所が存在し、企業はこれらを正しく理解したうえで、自社の規模、業務形態、予算、診断頻度、システム特性などを考慮し適切な選択を行うことが求められる。

具体的には、外部発注を選択した場合、専門企業のノウハウや技術を迅速に活用できる反面、診断対象の増加による費用の増大や、診断が開発スケジュールに噛み合いにくいことなど欠点もある。一方、内製化を進めれば、診断スキルやノウハウを社内に蓄積でき、開発プロセスと密接に連携した迅速な対応が可能となるが、専門的人材の確保・育成や継続的な知識のアップデートなど、組織として一定の負担を覚悟する必要がある。

また、本プロジェクトでは脆弱性診断の内製化状況、課題を認識するために、中核人材育成プログラム参加企業を対象にアンケート調査を行った。本章では外部発注・内製を具体的に比較したメリット・デメリットをアンケート結果も踏まえ整理するとともに、実際に内製化を進める際の注意点やハイブリッド運用という第三の選択肢についても言及する。

3.3 内製化の特徴と考慮点

脆弱性診断を内製化する場合、自社内に専任または兼任の脆弱性診断チームを設置し、継続的かつ計画的に診断を実施することになる。この方法では、システム仕様や業務特性について社内の関係組織から情報提供を受けたうえで脆弱性診断を行えるため、脆弱性が見つかった場合でも提供情報に基づいた的確な報告が可能である。また、診断を通じた関係組織との連携や結果共有により、セキュリティ知識の伝播が促進され、診断に関わる組織のセキュリティ意識向上にもつながる。さらに、自社内で診断能力を保有することで、頻繁なリリースサイクルや急な仕様変更にも柔軟に対応でき、問題発見後のフォローアップも外部発注特有の契約調整や対外的な利害関係の影響を受けず、迅速かつ円滑に実施できるという利点がある。

図 3-1 は、「脆弱性診断を内製化することによるメリット」のアンケート結果である。外部発注コストの削減や機密情報を社外に出さずにすむ点、自社のリスク基準に則った診断ができる点などが大きな割合を占めており、企業全体のセキュリティスキル向上や、重大な脆弱性発見時の柔軟な対応なども理由として挙げられている。

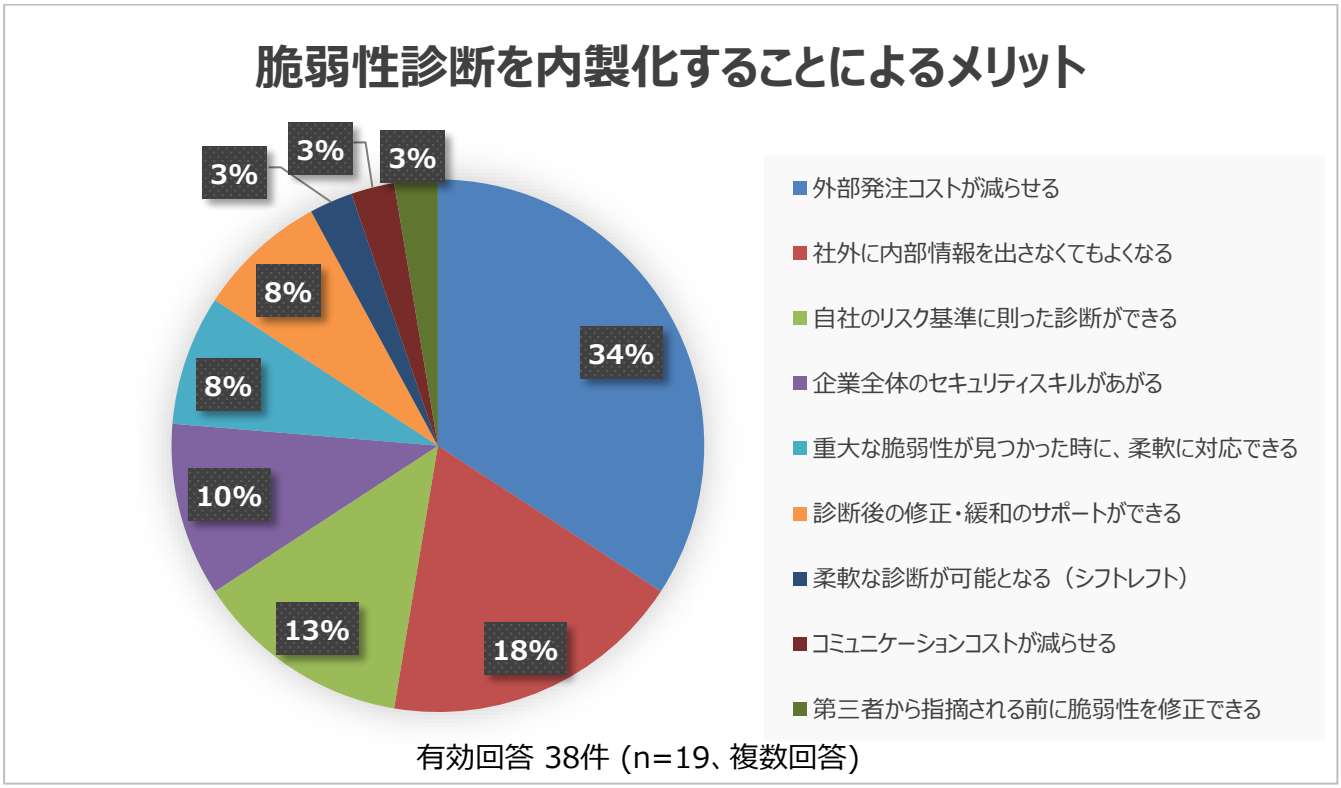

図 3-1 脆弱性診断を内製化することによるメリット

一方、内製化にはいくつか考慮すべき課題がある。診断を担当できる人材の確保や継続的な育成・維持にかかる負担は大きく、専門的なトレーニングや資格取得支援といった教育コストも無視できない。特に診断員は、実際の攻撃を再現する高度な技能が求められるため、最新の脅威動向を追うだけでなく、演習環境での継続的な学習と検証が必要となる。また、診断ツールの導入や運用管理、環境構築などにも技術的負荷が発生する。

図 3-2 は、「脆弱性診断内製企業の課題」のアンケート結果である。リソース（人材・コスト）の確保が困難であることが最も高い割合を示し、次点にスキル維持やノウハウの蓄積・共有手段の確立が課題となっていることがわかる。

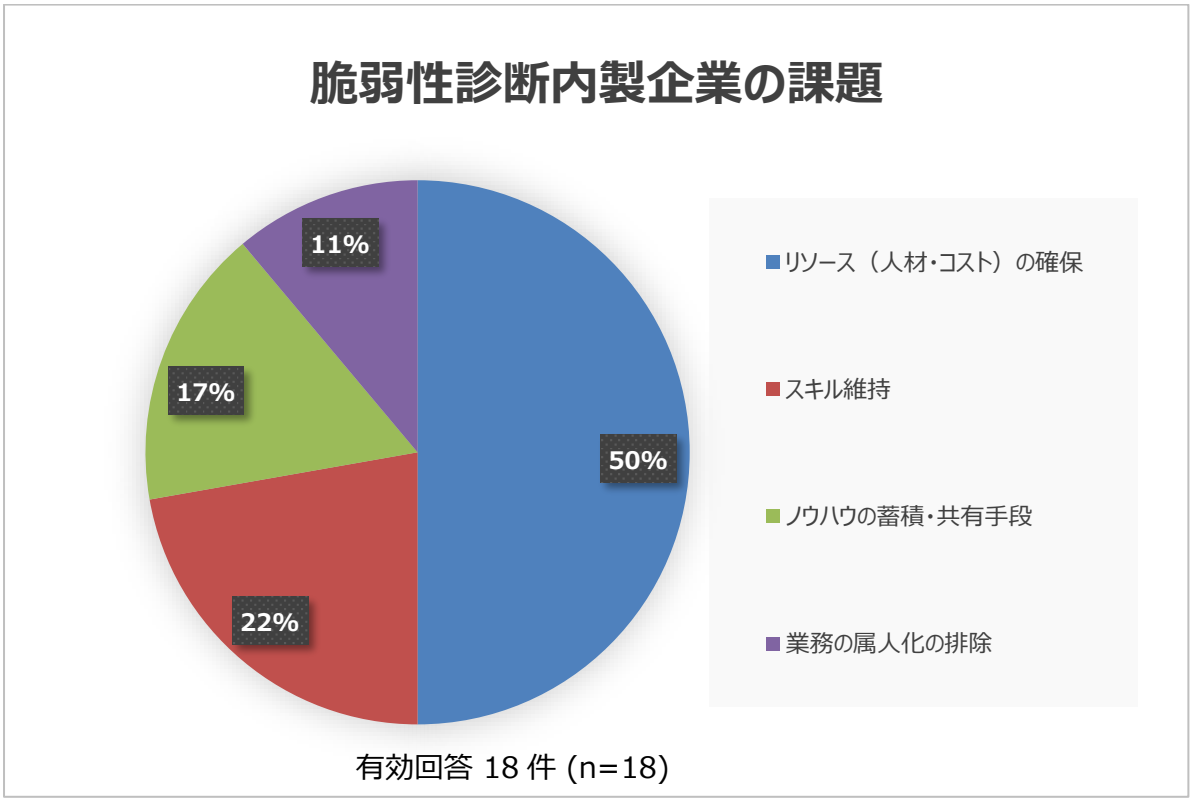

図 3-2 脆弱性診断内製企業の課題

3.4 外部発注の特徴と考慮点

脆弱性診断を外部に委託する場合、専門的な知識と豊富な実績を持つベンダーに脆弱性診断を任せることになる。外部発注では、専門企業が保有する診断手法やツールを活用できるため、自社で人材育成や環境構築を行う必要がなく、発注と調整のみで診断が可能となる。また、第三者視点で客観的な評価を得られるため、経営層や顧客に対する報告の説得力が増し、社内診断と異なる視点で脆弱性を発見できる可能性も高まる。

さらに、診断の頻度が少ない企業においては、必要なときだけ発注することで固定コストの負担を抑えられる点もメリットである。突発的な案件が発生した際にも、外部発注先に空きがあれば追加のリソースを確保できるため、ピーク時の診断需要に柔軟に対応することが可能である。

一方で、外部発注に伴う課題も考慮が必要である。外部発注は診断範囲や頻度が増加するにつれ費用が高額化しやすく、特に迅速な対応や緊急の追加診断が必要になる場合には予想外の費用が発生することもある。また、ベンダーごとの技術レベルや診断品質に差があり、適切な企業選定に一定の労力とノウハウが求められる。さらに、診断結果に対するフォローアップは契約で定められた範囲内となり、柔軟な追加対応や詳細なサポートが制限される場合がある。

加えて、診断を全面的に外部発注に依存すると社内にノウハウが蓄積されにくく、スキル向上が停滞する可能性もある。また、情報セキュリティの観点からは、機密性の高い情報を外部に共有する必要があるため、情報漏えいなどのベンダー管理上のリスクが生じることにも注意が必要である。

図 3-3 は「脆弱性診断を外部発注するデメリット」のアンケート結果を示している。費用面の高さや、診断企業の選定の困難さ、スキル向上が図れないことやコミュニケーションコストの増大などが大きな割合を占めており、企業全体としてのセキュリティスキル向上が停滞する懸念も見て取れる。

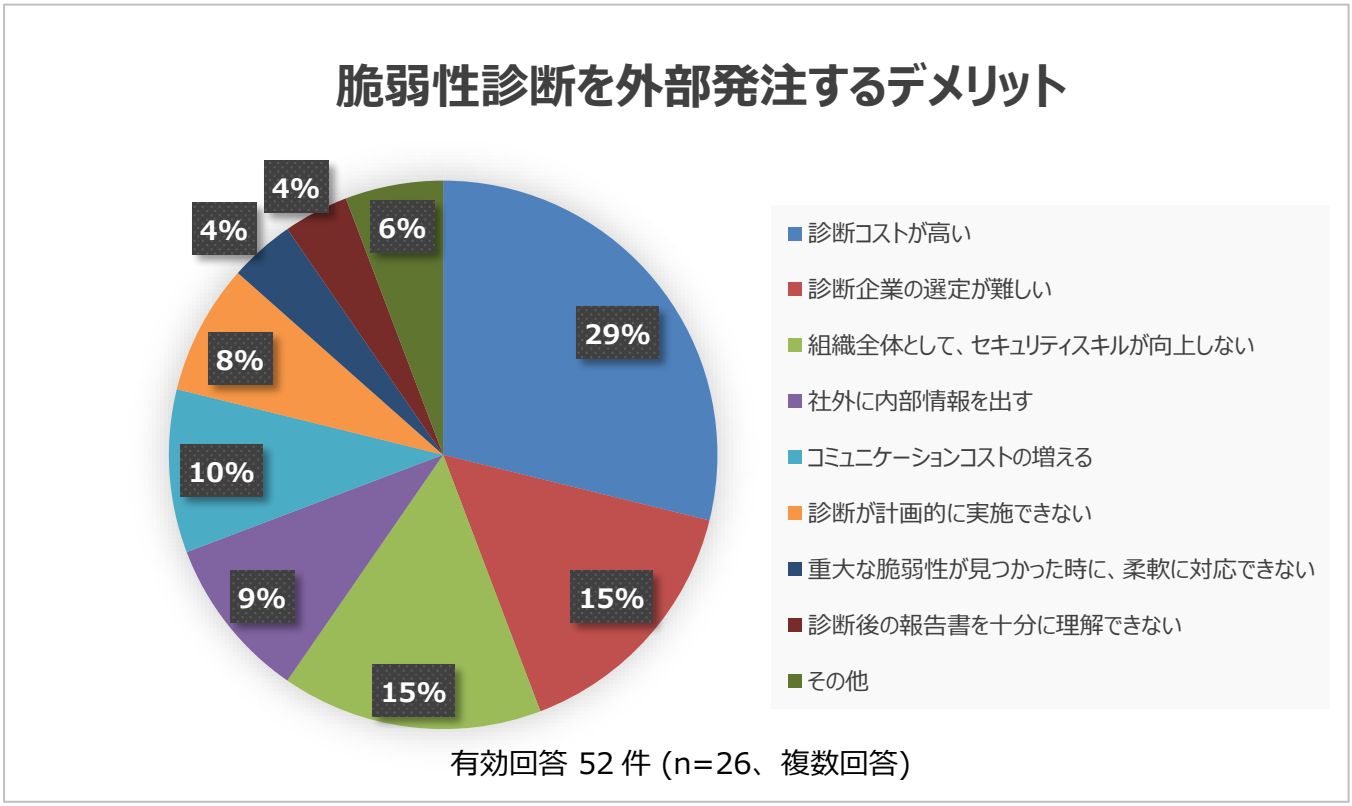

図 3-3 脆弱性診断を外部発注するデメリット

3.5 外部発注と内製のコスト構造の違い

脆弱性診断を外部発注するか内製するかを検討する際、コストの構造と推移の違いも大きな判断材料となる。

外部発注の場合は、契約形態や発注範囲の拡大に伴って診断ベンダーへ支払う費用が増加し、社内でのベンダー選定やスケジュール調整も含めたコストが継続的に発生する。一方で、内製化を進めると診断環境の構築や専門人材の育成といった初期投資が必要になるが、診断件数やシステム数が多い企業ほど長期的には診断コストの単価が下がっていく傾向がある。

図 3-4[10]に示すとおり、外部発注では委託費用と調整にかかるコストが診断対象や回数の増加に応じて大きくなりやすく、内製診断ではツールライセンスや診断員の運営コストが発生するものの、一定の運用体制を確立すれば相対的に運用負担を抑えられる。

[10] 株式会社ユービーセキュア ブログ https://www.ubsecure.jp/blog/20210630

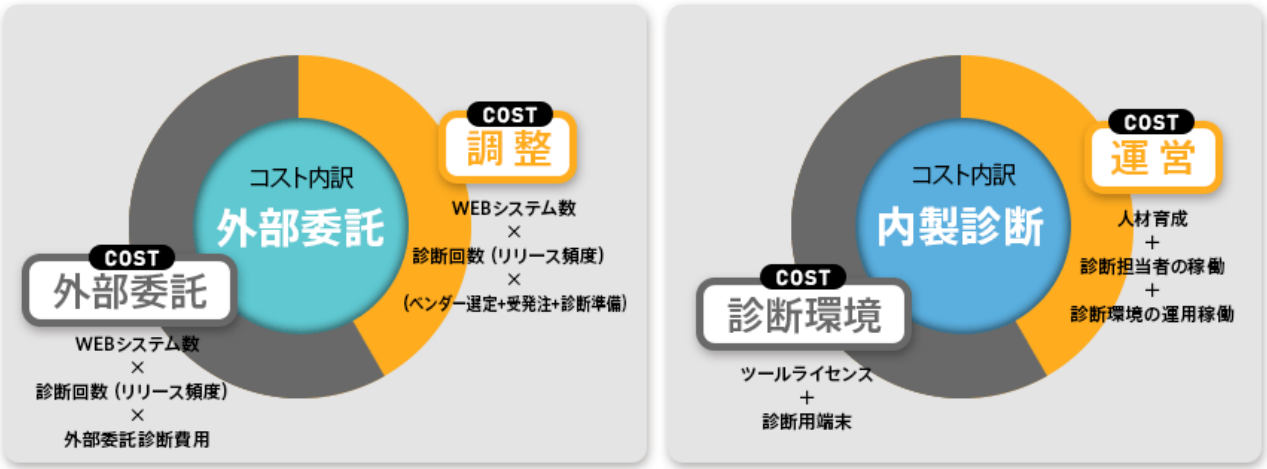

図 3-4 外部委託/内部診断 コスト構造 (株式会社ユービーセキュア)

一定数以上の診断案件が定常的に発生する企業では、図 3-5[11]にあるように、外部発注では費用が横ばいまたは緩やかに上昇するのに対し、内製は導入初期こそ高額の投資を伴うが、脆弱性診断チームが成熟するにつれ運用コストが低減していく。診断案件が増えるほど外部委託では発注コストが膨らみやすいのに対し、内製では人材とノウハウが蓄積されるほど効率が上がるという違いがあり、自社のリリース頻度やセキュリティリスク、長期的な投資計画などを踏まえて適切な選択をすることが重要になる。

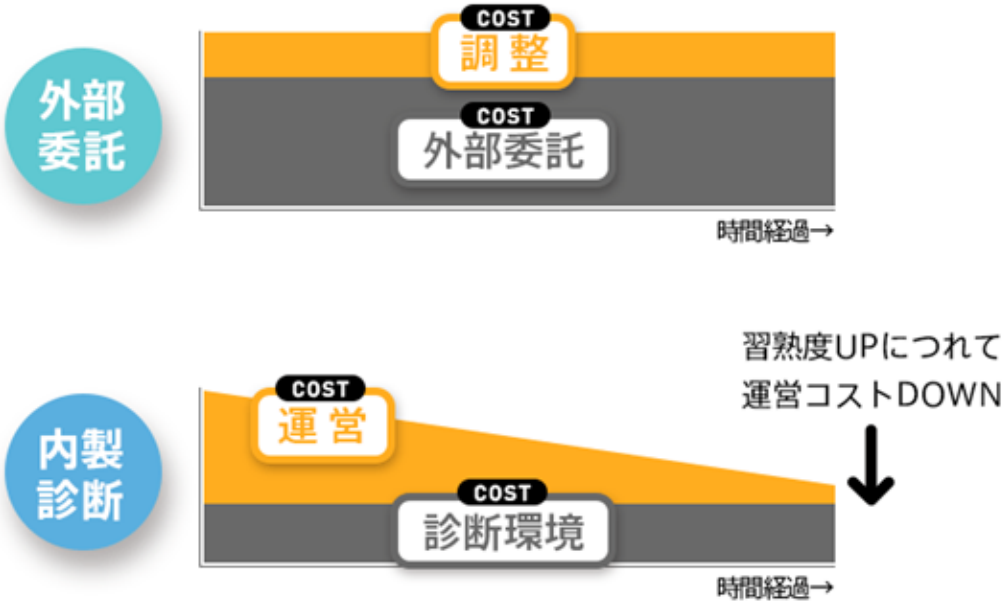

図 3-5 コストと時間の関係 (株式会社ユービーセキュア)

[11] 株式会社ユービーセキュア ブログ https://www.ubsecure.jp/blog/20210630

3.6 外部発注と内製のメリット・デメリット

表 3-1 に外部発注と内製の比較をマトリックス表で示す。企業の事情や規模によって重視するポイントは異なるため、以下の観点を総合的に考慮して適切な手段を選択することが望ましい。

表 3-1 外部発注と内製のメリット・デメリット

項目	外部発注のメリット	外部発注のデメリット	内製のメリット	内製のデメリット
スキル・ノウハウ	• 専業ベンダーの豊富な経験と手法を利用できる	• 社内にノウハウが蓄積しづらい • ベンダーごとに品質差があり選定が困難	• 診断を通じた知見が社内資産になる • 開発現場の仕様を理解したアドバイスが可能 • 脆弱性の発生傾向を基に、開発者向けトレーニングを実施できる	• 専門人材の確保/育成/維持に大きな投資が必要
コスト	• 社内で診断チームやツールを維持するための固定費が不要 • 費用総額を事前見積りしやすい	• 診断件数/範囲に応じて、費用が増加 • 緊急や短期間での診断を求める場合は費用が高騰	• 高頻度/大量診断では長期的にコスト圧縮	• 初期にツール導入/環境構築費が発生 • 診断件数が少ない場合 は外部発注より割高になることも
柔軟性・機動力	• 費用次第で必要な時期に診断体制を柔軟に拡充可能	• 発注～検収の手間が契約単位で発生 • ベンダーの空き状況に左右され開発サイクルに噛み合いにくい • 診断結果に対するフォローアップは契約範囲内まで	• 契約手続きが不要 • 開発中に診断員と開発者の修正に関する調整を密に行いやすい • 修正後の再チェックも迅速 • 診断結果に対する柔軟なフォローアップが可能	• 人的リソースを柔軟に拡充しにくい • 兼務が多いとピーク時の優先度調整が困難
情報管理・機密性	–	• 診断対象の情報を社外共有する必要がある	• 診断対象の情報を社外に出さずに済む • グレーボックス/ホワイトボックス診断が実施しやすい	–
客観性・網羅性	• 第三者視点のため、経営層/顧客に対して客観的かつ説得力のある説明ができる	• ベンダーごとにアプローチ/スコープにばらつき	• 開発部門との対話で修正方針まで踏み込める	• 外部発注と比べ客観性で劣る

3.7 ハイブリッド運用という選択肢

脆弱性診断を内製化していく中で、より客観的で高品質な診断を継続的に実施するために、定期的に外部の専門企業を活用するハイブリッド運用という選択肢もある。この運用方法では、内製チームが主体となって計画的かつ網羅的に診断を実施し、脆弱性の迅速な発見と柔軟な対応を行いつつも、一定の周期や重要なリリースタイミングで外部発注による診断を取り入れる。

外部発注による診断を取り入れる主な目的は、内製チームの診断手法や判断基準を第三者視点と比較し、診断品質の妥当性を担保することである。これにより、内製チームは自社の診断基準が業界標準や最新の脅威動向に即しているかを継続的に確認できる。また、内製化の導入初期段階では診断体制やスキルが未成熟な場合が多い。外部専門家による手動診断の結果と比較・検証を行うことで、自社の診断精度やスキルの向上と、誤検知や検知漏れの傾向を把握することができる。

このように、内製診断の強みを最大限に活かしながら外部発注による診断のメリットを適切に組み合わせることで、企業全体の脆弱性診断体制をより強固にすることができる。

3.8 本章のまとめ

本章では、脆弱性診断を実施するにあたり、外部発注と内製のそれぞれが持つ特徴やメリット・デメリットについて整理した。外部発注を選択すると、専門企業のスキルや手法を活用できる反面、診断件数や範囲が拡大するにつれて費用が増加するという課題がある。また、外部に情報を提供する必要が生じることから、情報管理上のリスクも考慮が必要である。

一方、内製化は、診断ノウハウを社内に蓄積し、自社のリスク基準に合わせた柔軟な運用が可能になるという利点をもつ。さらに、柔軟なフォローアップや機密情報を社内に留められることも魅力である。ただし、専門人材の確保や継続的なスキル維持、属人化の防止といった組織的な課題に対する取り組みが求められる。

これらの特徴を踏まえ、内製化を進める企業にとっては、日常的かつ網羅的な診断を自社で実施しつつ、定期的に外部の専門企業による診断を組み合わせるハイブリッド運用という選択肢も考えられる。このアプローチにより、診断品質の妥当性を確保しながら、自社診断能力の継続的な改善や向上を図ることが可能となる。

企業が適切な診断体制を構築するには、自社のビジネス環境やリソース、セキュリティリスクを総合的に評価し、そのうえで適した診断実施方法を選択することが重要である。本章で提示した各要素をもとに、自社の特性に適した診断方法を検討・選択することが望ましい。

コラム

～自社に適した診断手法を扱う外部発注先を見極めよう～

脆弱性診断を外部発注する際、ツールによる自動スキャンのみを提供し、低価格で脆弱性診断が行えるサービスもあります。費用面でハードルが下がる分、導入しやすいというメリットはあるものの、本ガイドでも述べたように、ビジネスロジックや複雑な設定ミスなどツールが苦手とする領域は手動診断でこそ正確に洗い出せる場合が多い点に留意が必要です。また、自動スキャンは誤検知を含むこともあるため、専門家の目で検証しながら不要な指摘を除外し、本質的なリスクを明らかにする作業が欠かせません。

外部発注先を選ぶ際は、ツールによる診断と手動診断の組み合わせや、発見した脆弱性の再現手順・修正方針の提示、フォローアップなど、**どこまでの範囲をサポートしてもらえるのかをしっかりと確認することが大切です。**もし、信頼性を客観的に判断したい場合には、経済産業省策定の「情報セキュリティサービス基準」に適合している事業者かどうかをチェックするのも有効な手段です。独立行政法人情報処理推進機構（IPA）のウェブサイトには、基準適合済みの事業者一覧が公表されていますので、こうした基準や実績を踏まえたうえで、**自社のニーズに応じた深い診断を提供してくれる外部発注先を見極めましょう。**結果として、誤検知による手戻りや重大なリスクの見落としを回避し、より効果的な脆弱性対策へとつながります。

IPA 情報セキュリティサービス基準適合サービスリスト

https://www.ipa.go.jp/security/service_list.html

4 内製化に必要な組織体制と人材

4.1 本章の目的

脆弱性診断の内製化は、セキュリティ部門の努力のみで実現できるものではなく、経営層やシステム開発、システム運用、アセットオーナーを含む IT 部門、セキュリティ部門内でもセキュリティ統括[12]など幅広い関係組織の理解と協力が不可欠である。本章では、内製化を実現するための組織体制や脆弱性診断チームに必要な人材の考え方について説明する。

4.2 経営層の関与と推進体制

経営層は、組織におけるリスク管理の最終責任者として、脆弱性診断の結果を単なる技術的な問題としてではなく、事業に影響を及ぼす経営リスクとして正しく認識し、対応方針を判断する責務がある。

また、技術部門は初動対応や可能な範囲でのリスクコントロールを主体的に実施し、その実施状況や効果を経営層に適切に報告して、的確な判断を支援する責務がある。

このような各々の責務を果たすためには、IT 部門やセキュリティ部門等の技術部門が診断結果の深刻度や影響範囲を整理し、リスクの内容と対応の選択肢（例：回避・低減・受容など）をわかりやすく説明することが不可欠であり、組織としての対応方針や優先順位の決定に経営層が関与する体制を整備する必要がある。

4.3 内製チームの役割とブランディング

脆弱性診断内製化を成功させるうえでは、内製チームの組織内における適切なブランドイメージの確立が重要となる。ここでは、チームが指摘役に留まらずシステム開発・運用を支援するサポーターとしてのブランディングをするための、具体的な活動について説明する。

4.3.1 セキュリティ意識の醸成とサポート役へのシフト

脆弱性診断チームは、脆弱性を指摘するだけでなく、それを如何に“建設的な形”で開発・運用担当者に伝え、改善を促すかが鍵となる。企業内でセキュリティを根付かせるためには、脆弱性診断チームが「お目付役」だけではなく「サポーター」として認識されることが望ましい。脆弱性報告に付随して具体的な修正例やヒントを示す、または開発部門向けのハンズオンや勉強会を開催するなど、実践的な支援に力を入れると効果が高い。

4.3.2 ブランドイメージを高める工夫

セキュリティに関係するチームは他の部署から見ると、関わりづらいイメージを持たれやすいことがある。しかし、そのままではシステム開発やシステム運用との円滑な連携が難しく、脆弱性診断の成果が十分に活かされないおそれがある。そこで脆弱性診断チームでは、

---

[12] サイバーセキュリティ経営ガイドライン Ver2.0 付録 F サイバーセキュリティ体制構築・人材確保の手引き / 経済産業省商務情報政策局サイバーセキュリティ課、独立行政法人情報処理推進機構（IPA）/ P.5 より / http://www.meti.go.jp/policy/netsecurity/tebikihontai2.pdf

以下のような工夫によってチームブランドを刷新し、親しみやすいイメージを築く方法が考えられる。

- **定期レポートや勉強会の実施**
  定期的なセキュリティレポート作成や勉強会にて、診断で見つかった脆弱性とその対処法を共有する。成功事例や修正がうまくいったケースも積極的に取り上げ、脆弱性の修正が重要なプロセスであることを理解してもらう。
- **相談・意見交換がしやすい文化醸成**
  チャットツールやポータルサイトなどを活用して、いつでも脆弱性診断チームに相談できる環境を用意する。安心して相談・意見交換できる場を整備することで、他部門からの抵抗感を和らげる。
- **セキュリティ研修やワークショップのゲーミフィケーション**
  チーム対抗の簡易 CTF（Capture The Flag）や演習形式のイベントを実施し、楽しみながらセキュリティ知識を身につける機会を提供する。ゲーム要素が加わることでシステム開発やシステム運用のメンバーも積極的に参加しやすくなり、脆弱性診断チームの存在感が自然に高まる。

4.4 チーム構成と役割

脆弱性診断を内製化するうえでは、診断を実施するチームの編成や人材の確保・育成方針が重要なポイントとなる。本節では、チームをどのように構成し、どのような役割やスキルセットが求められるかについて説明する。

4.4.1 マネジメント担当

- **役割**
  脆弱性診断を組織として継続的に運用し、リスクコントロールの仕組みとして機能させるためには、企業のセキュリティ戦略との整合性を踏まえたうえで、セキュリティ統括や他部門との調整をはじめ、診断体制の設計・予算管理・ガイドライン整備などを統括するマネジメント担当が不可欠である。この役割は、診断実務担当が専門的な検証に専念できるよう、体制整備や支援を主導するとともに、組織内での意識醸成や教育施策の企画・運営も推進する。診断を一過性のイベントで終わらせず、組織的リスクマネジメントの基盤として根付かせるために、「制度面」と「運用面」の両輪を整備する存在となる。
- **心掛け**
  マネジメント担当は、脆弱性診断の基本的な技術構造や診断手法への理解をある程度持ちつつも、システム開発・システム運用・アセットオーナーなど多様な関係組織の意見を丁寧に取りまとめていく姿勢が欠かせない。特定の部署への偏りが生じないよう中立的な立場を保ちつつ、各部門の業務状況やリリーススケジュールを把握し、診断計画を柔軟に調整することが大切である。さらに、診断実務担当が抱える課題や悩みに寄り添い、組織内での学習機会や情報共有の場を設けることで、長期的なセキュリティレベル向上に寄与する。
  また、業務要件などにより修正実施が困難な脆弱性については、システム側の都合も考慮して脆弱性修正ではなく代替手段でのセキュリティ対策も提言できるよう意識・心遣いも重要である。

- **適正人材**

  マネジメント担当には、セキュリティ分野の基礎知識や IT システム全般への理解はもちろん、プロジェクト管理や予算調整、ドキュメント作成、社内調整など多面的なスキルが求められる。論理的思考力とコミュニケーション力に加えて、他者からの信頼を得られるリーダーシップを持った「中核人材[13]」が適任である。また、診断体制の運営だけでなく、攻撃手法や業界環境の変化を踏まえ、自組織のリスクに応じた診断方針を策定・見直す視点も求められる。診断手法や内製チームの活用を戦略的に位置づけ、企業文化や事業戦略を踏まえながら施策を継続的に改善できる柔軟性も重要である

- **人的リソース**

  小規模な組織では、マネジメント担当を兼任にすることで回る場合もあるが、診断対象が多い企業や複数プロジェクトが同時進行する状況では、1 名以上の専任配置が望ましい。組織が拡大し、脆弱性診断の件数や対象が増加していく段階では、マネジメント担当を補佐する人材を追加で確保し、標準化や教育施策などを並行して推進できる体制づくりが必要となる。

- **想定業務**
  - 脆弱性診断の全体戦略・年間計画の策定
  - セキュリティ統括へのリスク報告、および予算・リソースの確保
  - 診断対象の優先度付けとスケジュール管理
  - チェックリストやテンプレートの整備、報告書品質の標準化
  - システム開発・システム運用・アセットオーナーとの協議・合意形成
  - 脆弱性診断のナレッジ管理と情報共有の仕組みづくり
  - 教育プログラムやトレーニングの企画・運営
- **必要スキル**
  - プロジェクトマネジメントスキル
  - セキュリティおよび IT 全般の基礎知識
  - 交渉・折衝・プレゼンテーション能力
  - 成果物レビューと品質管理の知見
  - KPI・KGI などの指標設計や進捗レポート作成能力
- **連携先**
  - セキュリティ統括
  - システム開発・システム運用・アセットオーナー

[13] 中核人材育成プログラム 事業内容 / 独立行政法人情報処理推進機構（IPA）/ https://www.ipa.go.jp/jinzai/ics/core_human_resource/about.html

4.4.2 診断実務担当

- **役割**
  診断実務担当は、脆弱性診断の“実働部隊”としてシステムやアプリケーションの診断を実施し、潜在的な脆弱性を発見・評価する中心的な役割を担う。攻撃者視点で脆弱性を洗い出し、指摘事項を関係組織に正しく伝えることで、具体的な対策を促進し、組織全体のセキュリティレベルを向上させる原動力となる。診断の結果を単に指摘として終わらせるのではなく、システム開発・システム運用と協調しながら修正や改善に結び付けることが求められる。

- **心掛け**
  診断実務担当は、専門的な技術に精通し、脆弱性を的確に見抜く探究心や日々進化する攻撃・防御・検出手法へのキャッチアップが必要となる。また、発見した問題点や修正方針を分かりやすく提示することで、納得感を与えることがポイントとなる。加えて、指摘において相手を責めるのではなく、「共により安全なシステムを作る」という協働姿勢を意識し、改善策を見出す過程まで積極的に伴走する姿勢が、企業のセキュリティレベル向上に重要となる。

- **適正人材**
  診断実務担当には、ネットワークや OS、ミドルウェア、Web アプリケーション、クラウド環境ひいてはコンピュータサイエンスなど幅広い技術分野に対する知識と経験が求められる。脆弱性診断ツールの使いこなしや脆弱性診断のスキルだけでなく、自発的に新しいセキュリティ動向を追いかける好奇心と探究心、システム開発・システム運用担当者との協力体制を築くためのコミュニケーション能力やプレゼンテーション能力が重要となる。

- **人的リソース**
  企業規模や診断対象数によっては、数名の兼任体制でも運用は可能であるが、より専門的で高品質な診断を実施するためには、チームとして専任の診断実務担当を複数人配置し、それぞれの得意分野を生かしながら補完し合う形を整備することが望ましい。属人化を防ぎ、診断ノウハウを組織として蓄積していくためには、定期的な情報共有や共同作業の機会を設け、チーム内でスキルアップを図る仕組みづくりが不可欠である。

- **想定業務**
  - 診断対象の把握・アクセス確認
  - ツールや手動検証を用いた脆弱性診断の実施
  - 検出した脆弱性の評価と報告書作成
  - 改善提案の提示とシステム開発・システム運用部門との協議
  - 過去事例や新しい脆弱性情報の収集とナレッジ共有

- **必要スキル**
  - ネットワーク・OS・ミドルウェア・Web アプリケーションなど幅広い技術知識
  - ツールベースの診断と手動検証の両面に関するノウハウ
  - 問題を的確に説明・提案するレポーティング能力

- 攻撃者視点でリスクを想定する批判的思考と探究心
- 改善プロセスをシステム開発・システム運用と協働して進めるコミュニケーション能力
- 報告会におけるプレゼンテーション能力

より詳細なスキルセットについては、OWASP Japan が公開しているスキルマップ[14]が参考になるため、必要に応じて参照するとよい。

- **連携先**
  - システム開発・システム運用・アセットオーナー

4.5 組織横断的なコミュニケーションと関係組織連携

内製化した脆弱性診断を効果的なものとするためには、診断チームと関係組織間の密なコミュニケーションが必要となる。この連携の必要性と実効的なコミュニケーション体制について説明する。

4.5.1 他組織との連携が必要な理由

診断の計画や実施許可の事前調整、診断結果の報告、修正時のフォローアップといった各プロセスは関係組織との連携なしには進めることができないため、関連組織とコミュニケーションを密接に行う必要がある。計画段階ではシステム開発やシステム運用を担う組織と診断対象や実施時期を調整し、アセットオーナーへシステムへの影響を説明したうえで実施承認を得る手続きが発生する。診断実施後は結果を速やかに関係組織へ報告し、指摘事項への対応や修正に関するフォローアップを行う。こうした連携が不足すると、診断が形だけのものとなることや、指摘事項が放置されるといった懸念が発生する。すべての関係組織が情報を継続的に共有し、各自の役割と責任を認識して連携することが、内製診断によるセキュリティレベル維持・向上の土台となる。

4.5.2 実効的なコミュニケーション体制の構築

- **定例ミーティングやワーキンググループ**
  脆弱性診断チームとシステム開発・システム運用が定期的に情報交換を行う場を設定する。例えば、「6.5 組織間セキュリティ意識の醸成」にあるような取り組みが実施可能である。これにより、相互でコミュニケーションを図りやすい環境を構築するとともに、定期的に脆弱性診断に関する要望や課題を共有することで、調整や修正対応の円滑化を図ることが出来る。
- **共有ツールの活用**
  チャットツール等の活用により、脆弱性診断チームとシステム開発やシステム運用が容易にコミュニケーションを図りやすい環境を構築する。また、課題管理システムなどを用い、脆弱性に関する報告・対応状況を一元管理する。脆弱性修正に係る関係組織がリアルタイムに進捗を把握できるようにすることで、修正漏れや認識の食い違いを防げる。

---

[14] 脆弱性診断士スキルマッププロジェクト - 脆弱性診断士スキルマップ＆シラバス
https://github.com/OWASP/www-chapter-japan/tree/master/skillmap_project

- **脆弱性診断スケジュールの共有**
  診断実施に先立ち、具体的な診断内容や予定日時、対象システムなどを事前に関係各所へ共有する。定期的な診断だけでなく、リリース直前のスポット診断や緊急の診断についてもスケジュールを可視化し、脆弱性診断に関わる組織があらかじめ対応準備を行えるように配慮することで、業務への影響を最小限に抑えることが可能になる。

4.6 本章のまとめ

本章では、脆弱性診断を内製化する際の組織体制や人材に求められる役割・スキルについて整理した。内製化を成功に導くためには、経営層が診断活動をガバナンスの一環として捉え、潜在リスクの可視化や迅速な意思決定を可能とする推進体制を整える必要がある。また、脆弱性診断チームは単なる指摘役ではなく、システム開発・システム運用を支援しながらセキュリティ意識を醸成し、企業内でポジティブなブランドを確立することが重要となる。

具体的なチーム編成としては、診断活動の運用基盤を支えるマネジメント担当と、実務を担う診断担当を適切に配置し、各々の役割と責任を明確化する必要がある。さらに、各関係組織との横断的なコミュニケーション体制を構築することで、診断業務の円滑な実施と効果的な脆弱性対策を推進できる。本章で示した要素を踏まえ、自社に適した人材戦略と推進体制を整備することが求められる。

コラム

～現場で本当にあった“あるある”失敗談：警告メールが大騒ぎに発展～

ある日、脆弱性診断を予定どおりに実施したものの、アセットオーナーへ診断日を伝え忘れてしまいました。すると、監視システムが診断で発生するアクセスを攻撃と検知し、警告メールがアセットオーナーに大量に送信されてしまいました。アセットオーナーは「何事だ！」と血相を変えてシステム運用に飛び込んできました。

きちんと計画を立て、実施承認を得たうえでの診断だったため大事には至りませんでしたが、「予定の共有を怠った」「診断により想定される影響（リスク）を伝えられていなかった」ことで少々お叱りを受ける羽目に…。

この一件から学べるのは、**“どんなに正当な診断でも、事前の情報共有や説明がなければ、自分にも相手にも不要な負担をかけてしまう”ということ。関係組織への連絡を怠らないようにするのが重要です。**

5 内製化の進め方と継続的改善プロセス

5.1 本章の目的

脆弱性診断の内製化においては、単にツール導入や担当者を割り当てるだけでなく、診断プロセスの全体像を明確にし、段階的に運用範囲を拡大していくことが重要である。本章では、まず脆弱性診断の内製化を行ううえで事前に決めておくべきことと、実際に内製化を進める際の導入ステップの一例として、「スモールスタート」「手動診断の導入」「全社・グループ展開」の流れで説明する。これにより、初期段階での導入方法や、徐々に社内スケールを広げる具体的なアプローチを把握できる。

5.2 事前に決めておくべきこと

脆弱性診断の内製化を進める上では、診断プロセスに関わる重要事項を事前に明確化しておくことが不可欠となる。ここでは、危険度基準や報告書フォーマット、体制と役割分担、業務フロー、診断アプローチの選定など、あらかじめ組織として合意・定義すべき主要な項目について説明する。

5.2.1 危険度基準と対応基準

脆弱性の危険度評価は、修正の優先度やリリース可否を判断するうえでも非常に重要である。組織として以下のような要素を検討し、基準を文書化しておくとよい。

- **危険度の段階設定**
  脆弱性が発見された際、その危険性や影響度を統一的に評価するために、共通の基準を定めておく必要がある。具体的には、攻撃が成立する現実的な可能性、影響を受けるシステムの範囲や深刻度を診断員が総合的に判断し、段階的に危険度を分類する。
- **対応期限やリリースへの影響**
  危険度ごとの対応期限やリリース判断基準を脆弱性修正に関わる組織と検討のうえ事前に明確化することで、迅速で一貫した対応が可能となる。特に重大な危険性が認められた場合はリリースを延期してでも修正を行う必要があるなど、危険度とリリース判断を関連付けておくと効果的である。

例えば、表 5-1 に示すような危険度基準を設ける。これはあくまで一例であり、自社システムの特性や業務特性を踏まえて独自の基準を設定することが望ましい。

表 5-1 危険度基準の例

危険度	説明	例	対応の目安
Critical	深刻な影響を与える可能性があり、容易に攻撃が可能である	インターネット上から認証なしでリモートコード実行が可能	即時修正・対応、システムの一時停止検討
High	深刻な影響を与える可能性があり、攻撃が成立する可能性がある	特定のユーザでリモートコード実行が可能	速やかに修正（1週間以内）
Medium	攻撃された場合の影響が限定的、または間接的な攻撃である	反射型クロスサイト・スクリプティング（XSS）	次回リリースまでに修正
Low	影響は限定的だが、改善が望ましい	情報漏えいの可能性のあるエラーメッセージ	定期的な監査・アップデート
Info	直接的なリスクはないが、注意が必要	セキュリティ関連レスポンスヘッダの未設定	推奨対応、次回改善策に組み込む

こうした対応基準をあらかじめ明確にしておけば、脆弱性診断チームが脆弱性を発見した際に混乱なく対処できる。

5.2.2 診断実施前のヒアリングシート

脆弱性診断の実施には、診断対象システムの仕様や構成、診断実施にあたっての確認事項を事前に把握することが不可欠である。そのための手段として、診断依頼元であるシステム開発やシステム運用に記入を依頼する「ヒアリングシート」を事前に用意し、運用することが有効である。一般的には、以下の項目を含むヒアリングシートを用意するケースが多い。

- 診断対象の基本情報
  （対象システム、対象 IP アドレス、対象 URL・診断用アカウント、保有するデータ資産分類（個人情報、クレジットカード情報など）等）
- 技術仕様に関する情報
  （システム仕様や構成図、フレームワーク、外部連携サービス 等）
- 診断実施にあたっての確認事項
  （診断アクセスによるメール等の外部通知の有無 等）

5.2.3 報告書フォーマットと提出方法

脆弱性診断の結果をどのようにまとめ、関係者へ共有するかは、あらかじめフォーマットと提出方法を決定しておく必要がある。一般的には、以下の項目を含む報告書を用意するケースが多い。

- 診断概要（対象システム、実施日時、使用したツール・手法・総合評価）
- ポートスキャン結果
- 発見された脆弱性の一覧と危険度評価
- 脆弱性の詳細と再現手順、悪用された場合の影響、修正ガイド

フォーマットが統一されていれば、システム開発やシステム運用が複数のシステムに対して同じ形式で報告書を受け取り、優先度に応じた修正計画を立てやすくなる。

報告書の内容をもとにシステム開発やシステム運用は管理するシステムのリスク評価を行い、適切に修正対応を行うことが望ましい。

5.2.4 診断体制と役割分担

診断を内製化するにあたっては、どの組織が何を担当するのかを事前に決めておかなければならない。例えば以下のような役割分担が考えられる。

- **マネジメント担当**
  ツールライセンスの管理、診断スケジュール調整を含む各関係組織との協議などの脆弱性診断チームのマネジメント全般を担う。
- **診断実務担当**
  実際のツール操作や手動診断、報告書作成などの技術的作業を担う。
- **アセットオーナー**
  保有するシステムやデータ資産に対する影響を把握し、診断の承認を与える役割を担う。
- **システム開発・システム運用**
  脆弱性診断の結果を踏まえたシステムのリスクアセスメントによる対応方針の検討や修正実装を担う。定期メンテナンスやリリース計画などシステム稼働に対する調整を鑑みて、脆弱性解消を進める。
  リスクアセスメントについては、ISOG-J から「脆弱性トリアージガイドライン作成の手引き[15]」が公開されているため参考にするとよい。
- **経営層**
  セキュリティ投資やリスク受容の判断、全社方針の承認。

また、脆弱性診断には複数部門が関与するため、誰がどの工程で関与し、何を担うのかを事前に整理しておくことも重要である。例えば、診断対象の確認や日程調整や診断の実施・報告、脆弱性診断チームへの対応依頼など、役割や承認ルートを明確にしておくことで、対応漏れや連携の遅れを防ぐことができる。こうした分担の文書化は、内製化の定着にもつながる。

5.2.5 診断業務フロー

社内で脆弱性診断を実施する際のフローを、あらかじめ定義しておくことが望ましい。下記のように、脆弱性診断におけるリスクをどの段階で説明し、承認を得るかを文書化するとスムーズに運用できる。

1. **対象・日程調整**
   診断対象システムの確定とスケジュールの調整を行う。

---

[15]脆弱性トリアージガイドライン作成の手引き / ISOG-J WG1 / https://wg1.isog-j.org/TriageGuidelines/

2. **診断前情報の取得**
   診断を円滑かつ正確に実施するため、診断対象に関する基本情報を依頼元からヒアリングシートをもとに事前に取得する。

3. **事前準備とリスク周知**
   システム開発・システム運用、アセットオーナーへ診断の概要と想定リスクを説明し診断ツールによるシステムへの負荷やネットワークへの影響なども含めて合意を取る。

4. **診断実施**
   診断ツールや手動診断により脆弱性を検出する。

5. **評価と報告**
   発見した脆弱性を危険度基準に照らし合わせて評価し、報告書を作成し報告する。

6. **修正とフォローアップ**
   システム開発・システム運用が修正作業を行い、必要に応じて再診断と修正作業に関する技術的サポートを実施する。

診断実施にあたっては、業務影響や技術的制約により、特定のシステムや処理について診断対象から除外、または一部制限を設けるケースがある。このような除外・制限事項については、脆弱性診断チームとアセットオーナーの間で除外の理由や未診断領域の残存リスクなどあらかじめ調整・合意し、文書で明確に管理しておく必要がある。

また、本番環境への影響を避けるために検証環境を用いて診断を行う場合は、検証環境が本番と同じ画面遷移可能な構成・データを持っているかを確認する必要がある。検証環境での診断は有効な代替手段となるが、構成差異による見落としのリスクがあるため、診断報告書には「検証環境での診断であること」とその前提条件を明記しておくことが望ましい。

5.2.6 診断アプローチの選定（ブラックボックス、ホワイトボックス、グレーボックス）

2.6 で説明したように、診断アプローチは一般的に「ブラックボックス診断」「ホワイトボックス診断」「グレーボックス診断」の 3 つに分類される。これらは、診断対象ごとに個別に選定するものではなく、自組織におけるセキュリティ戦略や診断の目的に基づいて、標準的な診断方針としてあらかじめ定義し、方針として文書化しておくことが望ましい。

内製においては、外部発注と比べシステム開発やシステム運用との連携が容易で、システムの設計書やソースコード、詳細な設定情報などの情報共有がしやすいため、ホワイトボックス診断やグレーボックス診断を実施しやすい環境にある。特にこれらの診断では、内部情報を活用することで、効率的かつ高精度で脆弱性を特定できるメリットがある。例えば、複雑なビジネスロジックや内部処理に起因する脆弱性をより迅速かつ網羅的に検出することが可能である。しかし、診断員にコード解析のスキルが必要となり、工数も多くかかるため、担当者の技術レベルや診断に割けるリソースを事前に考慮する必要がある。

また、診断アプローチによってはシステム開発やシステム運用からの情報共有が必要となるため、診断依頼元へのヒアリング（5.2.2 参照）に加え、情報共有の範囲や手順についてもあらかじめ調整し、明確にしておく必要がある。

コラム

～資産管理と脆弱性診断――“何を持っているか”を把握する大切さ～

脆弱性診断を実施するにあたって、まず必要になるのが「そもそも何を診断するか」という“対象範囲の把握”です。インターネット上に公開されているサーバやサービスをどれだけ持っているか、どのポートを開放しているか、本番環境とステージング環境がどのように混在しているか――こういった**資産情報**がきちんと整理されていなければ、脆弱性診断を受けたくても「どこから調べればいいのか正確に分からない」という状態になりかねません。最悪の場合、数あるシステムのうち一部しか認識できず、潜在的に危険なサービスを把握しきれないまま放置してしまう、といったリスクもあります。

そのため、**脆弱性診断を受ける前提として、インターネットへの公開状況やサーバ台数、稼働中のアプリケーションなど、自社が持っている資産全体の棚卸し（資産管理）をしておくこと**がとても大切です。いざ診断対象を決めようとするときに「そもそも私たちってどんなサービスを公開してるんだっけ……？」と首をひねってしまうようでは、正確な診断計画は立てられません。

さらに、脆弱性診断をうけてみると、思わぬところで「実はこんな古いライブラリが稼働中だった！」「このフレームワーク、もうサポート切れてたんだ……」といった事実が次々と判明することがあります。こうした情報は診断の結果レポートを確認するだけでなく、**社内の一元的な資産管理データベース**にも蓄積しておくのがおすすめです。

なぜなら、**もし後日、そのライブラリやフレームワークに新たな脆弱性が発見された場合**、すぐに「どのシステムが影響を受けるか」を洗い出し、迅速にパッチ適用やバージョンアップを行えるからです。逆に、資産情報がバラバラだったり管理されていなかったりすると、「あのサービスって何を使ってたっけ……？」と毎回人力で調べ回る羽目になります。

つまり、脆弱性診断は単なる“不具合探し”にとどまらず、**資産管理の観点でも非常に役立つ**側面を持っています。診断結果を細かく見返してみると、どの環境でどのバージョンのミドルウェアを使っているか、依存関係にあるライブラリは何かなどが見えてくるはず。これらの情報を社内で一元的にまとめておくことで、後々の脆弱性対応やソフトウェア更新計画の立案が楽になります。

どんなシステムやデータがあるかを知らなければ、適切に守ることなどできません。脆弱性診断はその“持ち物”を知るよい機会でもあるので、**まずはしっかりと資産管理を行い、インターネットへの公開状況などを正確に把握してから診断に臨む**ことが大切です。

5.3 ステップ 1：スモールスタートによる段階的な運用開始

脆弱性診断の内製化を進めるにあたり、まずは小規模で影響度の低いシステムから段階的に運用を始めることが望ましい。特に初期段階では、運用ミスや診断漏れなどのリスクを考慮し、社内向けの限定的な影響範囲の内部システムやアプリケーションを対象として運用経験を積むことが適切である。この段階では自動診断ツールによるツール診断を中心に、診断プロセス（ツール選定、実行手順、脆弱性評価基準など）を確立しつつ、手動診断のスキル習得を目指す。

また、内製と外部発注を併用して診断品質を確保しつつノウハウを段階的に蓄積することが有効である。例えば、報告書に記載された脆弱性の具体的な再現手順や、内製診断結果で発見できなかった部分の考察は、内製チームにとって貴重な学習機会となる。さらに、報告会の場で診断員へ直接質問することで、報告書からは読み取れない思考プロセスや着眼点といった、より実践的なノウハウを得ることも可能な場合がある。このようにして得られた知見や最新の視点を内製チームの育成やプロセス改善に組み込みながら、徐々に内製化の範囲を拡大していけば、社内の診断体制を着実に強化できる。

こうしたスモールスタートを通じて診断プロセス全体の課題や改善点を洗い出し、脆弱性診断チームとシステム開発・システム運用との担当者間のコミュニケーション体制を整えることで、本格的な内製化運用への基盤を固めることが可能になる。

5.4 ステップ 2：手動診断の段階的な導入

5.4.1 ツール診断の限界を補う手動診断の拡大

ツール診断が定着してきたら、より高度な脆弱性を発見できる手動診断の比重を徐々に増やす。このステップにおける「定着」とは、脆弱性診断チームがツールの操作・設定や運用フローを習熟し、診断結果として出力される脆弱性リストを十分に理解できるようになり、誤検知や検知漏れの傾向まで把握できている状態を指す。特に Web アプリケーションでは、ビジネスロジックに依存する脆弱性や複雑な認証・セッション管理の問題をツールだけで網羅するのは難しいため、診断員による手動検証が不可欠である。手動診断を加えることで、発見可能な脆弱性の幅が一段と広がるだけでなく、ツール診断の結果を補完し、より高い診断精度を実現できるようになる。

5.4.2 チェックリストと標準手順の作成

手動診断は属人的な作業になりやすいが、社内で共通のチェックリストや標準手順を作成しておくことで、一定の品質を維持しながら複数のメンバーが対応可能になる。チェックリストには、代表的な攻撃手法（SQL インジェクション、クロスサイト・スクリプティング（XSS）、アクセス制御不備など）をはじめ、脆弱性診断を通して得られた傾向からよく見られる脆弱性事例を反映しておくとよい。手順の標準化をすることで、新任メンバーが手動診断に参加するハードルを下げ、組織としての継続性を高めることができる。

5.5 ステップ 3：全社・グループ展開

5.5.1 開発および運用プロセスへの内製による脆弱性診断の公式組み込み

スモールスタートや一部システムへの手動診断導入を経て、社内ノウハウが蓄積してきた段階で、内製化による脆弱性診断を全社的な活動として定着させる必要がある。新規開発プロジェクトや既存システムへの新機能追加においては工程に診断フェーズを正式に組み込み、リリース前に必ず脆弱性評価と修正を行う仕組みを整備する。同時に、既存システムに対しても年次や四半期ごとといった定期的な診断スケジュールを決定・文書化し、継続的に安全性を維持・確認することが重要となる。

ウォーターフォール型開発であれば実装フェーズ以降の各フェーズやリリース前に脆弱性診断を実施する。アジャイル型の場合は、主要マイルストーンごとにツール診断を繰り返し行い、リリース前にツールと手動による診断を行うことで、継続的にリスクを低減させる運用が望ましい。また、運用するシステムには定期的な脆弱性診断を実施する。企業ごとの開発および運用サイクルに応じて、診断タイミングを適切に設けることは、全社的なセキュリティ品質の向上とリスクの低減を実現する上で欠かせない要素である。

5.5.2 リスク基準と対応基準の社内共有

全社展開を行う際には、あらかじめ定めた危険度基準と脆弱性対応基準を広く周知し、どのランクの脆弱性についてどの程度の期間内に修正すべきかを明確にする必要がある。特に危険度の高い脆弱性については、「サービスリリース前に必ず修正完了」や「定期診断後 X 日以内に対応」など、具体的に文書化し、システム開発・システム運用に浸透させることが重要である。このような基準を明確化しなければ、プロジェクトごとに対応の優先順位がばらつき、重大な脆弱性が放置されるリスクが高まる。

5.6 品質向上と継続的改善

前節までは、脆弱性診断内製化の導入ステップを説明してきた。本節からは、その体制を実際に運用し、業務品質の向上と継続的な改善のための具体的な取り組みについて説明する。

5.6.1 属人化を防ぐための運用

脆弱性診断の品質維持と組織的な能力向上のためには、特定の個人に依存する属人化を可能な限り防止した運用体制の構築が重要となる。ここでは、ナレッジ共有や標準化、レビュー、複数人診断、情報共有会などを通じた、チーム全体の能力を底上げするための具体的な運用方法について説明する。

5.6.1.1 過去事例の蓄積と再利用

脆弱性診断を継続するなかで見つかった問題や修正手順は、組織の貴重なノウハウである。診断の品質を高めるためには、こうした情報を個別の報告書だけにとどめるのではなく、体系的に整理して蓄積し、組織で再利用できるナレッジマネジメントの仕組みが重要である。

過去に検出された脆弱性のパターンやそれに対する修正方法、効果的な診断手法などを一元的にまとめ、診断員が迅速かつ容易に参照できる環境を整備することにより、同じ問

題に再度遭遇した際の対応速度や品質を大きく向上させることができる。ナレッジの更新・参照が簡単で日常業務に自然に組み込めるような仕組み作りを目指すことが望ましい。

5.6.1.2 チェックリスト更新と標準化

脆弱性診断の品質を一定以上に保ち、診断員間での属人化を防ぐためには、診断時に用いるチェックリストを定期的に見直し、最新の脅威動向や自社環境の変化に応じて更新することが必要となる。脆弱性診断チームが診断の都度、新たな手法で発見した脆弱性や診断手順をフィードバックとしてチェックリストに反映させ、内容を常に最新化しておくことが重要である。

また、診断プロセスの中で効果的であった手法や修正対応策については、組織内で標準化して展開することで、診断全体の効率化と精度向上が図れる。標準化により診断員のスキルや経験の差を最小限に抑え、組織として安定した診断品質を確保することができる。

5.6.1.3 報告書のレビュー

報告書作成や報告を特定の担当者だけで進めることが常態化すると、報告内容の視点や表現が属人化しやすく、診断結果の正確な伝達や比較が難しくなるリスクがある。これを回避するには、作成された報告書をチーム内の別のメンバーが査読し、内容を多角的にチェックするプロセスを定着させることが有効である。また、レビュー担当を持ち回り制にすることで、報告書の品質を平準化し、組織内にレビュー文化を根付かせる効果も期待できる。

5.6.1.4 複数人診断

診断作業を特定の個人に依存させないために、診断を複数の診断員で行うことが推奨される。ペア診断や交代制などを採用することで、診断手法や判断基準が共有され、属人化リスクが軽減される。また、複数の視点で診断を行うことで脆弱性の見落としを防ぐ効果も期待できる。

5.6.1.5 定期的な情報共有会

診断後に結果を報告して終わりにするのではなく、チーム内や関連組織を交えた情報共有の場を設けることが、ナレッジを広めるうえで大いに役立つ。そこでは、頻出している脆弱性や修正に手間取った事例、誤検知が多発したケースなどを発表し合い、要点をまとめる。こうした共有会を定期的に開催することで、脆弱性診断チーム全体で修正までのプロセスにおける新たな視点を得られ、次回以降の診断に向けチェックリストや報告書フォーマットへ反映することができる。

5.6.2 指摘事項の修正状況の追跡と残存リスクの管理

脆弱性診断の結果、指摘された脆弱性がどの程度修正されているかを継続的に把握することで、修正の遅延や対応漏れを早期に発見・解消できる。具体的には、検出した脆弱性ごとにステータスを管理し、あらかじめ定めた修正期限に基づいて進捗を確認する仕組みを整備するとよい。こうした可視化によって、重大度の高い問題を優先的に対処しやすくなり、企業内でリスクを共有しながら迅速な対応を進められる。

また、脆弱性診断チームは、指摘事項に対して不備なく修正が行われているかを確認し、修正状況をセキュリティ統括へ定期的に報告することが重要である。もし修正が遅延している場合や対応策が十分でない場合は、経営層からの修正指示を仰ぐなどの体制を整えることで、企業としてリスクを適切にコントロールできる。こうした継続的なモニタリングと報告の仕組みが定着すれば、常に現状のリスクを可視化し最小化できるため、結果として企業全体のセキュリティ水準を高めることにつながる。

5.6.3 危険度基準の見直し

攻撃手法や脆弱性に関する情報は日々変化しており、脅威環境も継続的に進化している。こうした変化を踏まえ、組織で定めている脆弱性の危険度基準が、現状に即したものとなっているかを定期的に確認・更新することが重要である。

必要に応じて危険度基準を見直すことで、診断結果に基づく対応の優先順位付けや対応判断の精度を維持し、診断活動全体の有効性を高めることができる。

5.7 本章のまとめ

脆弱性診断の内製化を実現させるためには、導入当初から診断プロセス全体の文書化と、段階的な運用範囲の拡大が求められる。本章では、内製化にあたり事前に決めるべき事項として、危険度評価の基準や報告書フォーマット、役割分担、診断フロー、さらには診断アプローチの選定といった項目を提示した。これらを事前に整備しておくことで、脆弱性診断チームがスムーズに業務を遂行でき、他部門との連携が円滑になる。

実際の導入プロセスとしては、まずリスクが限定的なシステムを対象にしたスモールスタートを推奨した。その後、自動診断ツールの活用を定着させつつ、徐々に手動診断を導入して脆弱性診断の範囲や精度を高めていく。これらの診断ノウハウが蓄積された段階で、診断プロセスを開発・運用の公式な工程として全社的に展開し、定期診断も含めて企業全体のセキュリティに関する安全性を確保する必要があることを説明した。

さらに、脆弱性診断の品質向上と診断体制の継続的な改善を行うためには、属人化の防止や修正状況の追跡、危険度基準の見直しが重要となる。特に診断業務の属人化を防ぐため、過去の診断事例を体系的に蓄積・再利用するナレッジマネジメント、チェックリストの更新と標準化、報告書のレビュー制度、複数人での診断、そして定期的な情報共有の場を設けることが効果的である。

これら一連のプロセスと体制を確立・維持することで、脆弱性診断チームのパフォーマンス向上と企業全体のセキュリティ強化が実現できる。本章の内容を踏まえ、各企業の状況に合わせて具体的なアクションを検討し、実践に移していくことが求められる。

6 関係組織との連携とセキュリティ意識の醸成

6.1 本章の目的

脆弱性診断を内製化し、システム開発やシステム運用の各段階で効率よく脆弱性診断を実施するには、関係する複数の組織と連携しながらセキュリティに対する共通認識を育てていくことが必要である。本章では、仮想企業におけるセキュリティ組織体制の一例を示しつつ、開発時のリリース前診断と運用中の定期診断をどう位置づけ、どのように各部門が連携すべきかを具体的に説明するとともに、関係組織同士でセキュリティ意識の醸成を行うための効果的な手段を提示する。

6.2 仮想企業における組織モデルの概要

図 6-1 はあくまで一つの例として、セキュリティ部門、IT 部門など、複数の関係組織がどのように脆弱性診断に関わり、どのように役割を分担するかを示したものである。

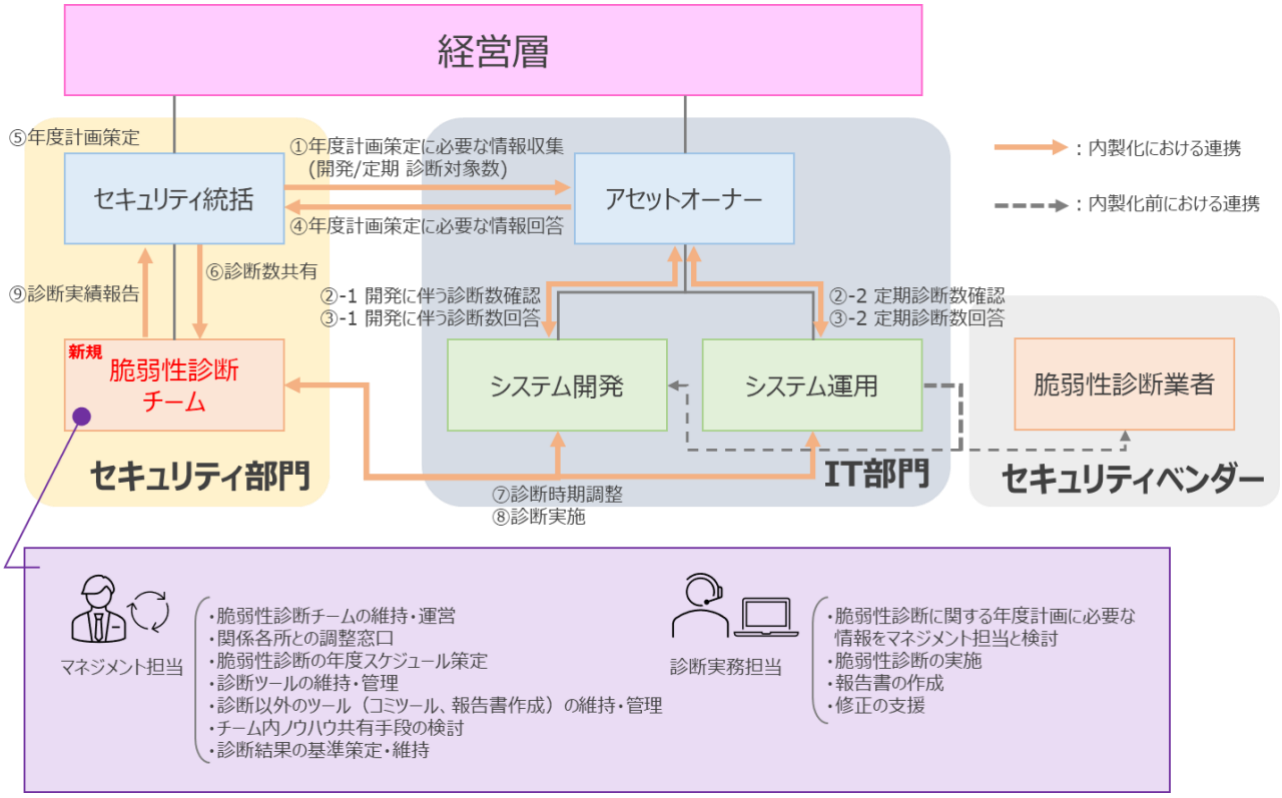

図 6-1 組織体制例

図に示す仮想企業では、「セキュリティ部門」のセキュリティ統括の配下に新たに脆弱性診断チームを設置し、開発・運用を担う「IT 部門」（システム開発、システム運用、アセットオーナー）と連携しながら、全社のセキュリティ対策を推進している。

6.2.1 セキュリティ部門と脆弱性診断チームの位置づけ

- **セキュリティ統括**
  全社的なセキュリティポリシーや年度計画を策定し、経営層や IT 部門と調整を行

う役割を担う。脆弱性診断においては、診断対象の優先度や年間のスケジュールを決定する際の情報を収集し、脆弱性診断チームと緊密に連携していく。

- **脆弱性診断チーム**
  脆弱性診断を実施し、報告書作成を行う。ここではマネジメント担当と診断実務担当に役割を分け、脆弱性診断チームの維持運営やツール管理、脆弱性修正状況の管理、実際の診断作業などを分担している。

6.2.2 システム開発、システム運用、アセットオーナーとの関係

- **システム開発**
  新規開発や機能追加など、開発時のリリース前診断を受ける立場である。脆弱性診断チームからの結果を踏まえ、リスクアセスメントを行い修正や対策を実装する。
- **システム運用**
  稼働中のシステムに対して定期的な脆弱性診断を受け、必要なパッチ適用や設定変更を行う。システムの安定稼働と脆弱性診断を両立させるためにも、診断時期や内容をあらかじめ調整し、業務影響を最小限に抑える工夫が求められる。
- **アセットオーナー**
  自身が担当するシステムやデータ資産に対する脆弱性診断の必要性やリスクを事前に把握し、診断の承認を与える役割を担う。

脆弱性診断チームは、それぞれの関係組織との十分なコミュニケーションと合意形成が必要となる。

6.3 開発時のリリース前診断

システムの新規開発や大規模改修のリリース前には脆弱性診断を行い、深刻なリスクが残ったまま本番稼働しないようにする体制を整える必要がある。以下に、その流れと組織間の連携ポイントを示す。

6.3.1 診断計画・要件定義

- システム開発と脆弱性診断チームが、プロジェクト計画の初期段階で診断の大まかな実施時期・対象範囲を共有する。
- 脆弱性診断チームは、事前にアセットオーナーに診断実施の概要を伝え、アセットオーナーはシステム運用への影響が想定される場合には調整を行う。

6.3.2 スケジュール調整

- 脆弱性診断チームは、年度計画および開発進捗を踏まえ、システム開発と調整を行いリリース前診断の具体的な実施日時を確定する。
- 仕様変更やプロジェクト延期などがあれば、システム開発から脆弱性診断チームへ早めに連絡し、診断時期や診断範囲を再調整する。

6.3.3 実施準備

- システム開発は、リリース前診断に必要なテスト環境やアクセス権限、技術情報を脆弱性診断チームへ提供する。

- アセットオーナーが関与する場合は、システム仕様や運用条件の確認などの調整を行う。

6.3.4 診断結果の共有と対応策

- 診断後、脆弱性診断チームが結果を報告書としてまとめ、アセットオーナーおよびシステム開発へ提出する。
- システム開発は内容を確認し、リスクアセスメントにより危険度が高いと評価された指摘事項について、アセットオーナーと調整のうえ修正する。
- システム開発は修正対応後、再診断を脆弱性診断チームに依頼する。

6.4 運用中の定期診断

運用中のシステムについては、定期的な脆弱性診断を実施することで、新たに発見された脆弱性や設定ミス、運用フェーズで生じた変更などを把握し、継続的にリスクを管理していく。本仮想企業では、以下のプロセスが想定される。

6.4.1 診断計画・要件定義

- 脆弱性診断チームが、運用中システムの診断枠を年間計画として設定する。
- アセットオーナーと連携し、診断サイクルを考慮して対象システムを決定する。
- アセットオーナーは診断内容を把握し、システム運用への影響を鑑みた承認を与える。

6.4.2 スケジュール調整

- アセットオーナー、システム運用、脆弱性診断チームの三者で、システムへの負荷やサービスへの影響を考慮して、定期診断の実施日時をすり合わせる。

6.4.3 実施準備

- システム運用は、定期診断に必要な環境や対象システムの情報、アクセス権限を脆弱性診断チームへ提供する。
- システム運用は、禁止事項を脆弱性診断チームへ共有し、診断実施によるサービス提供への影響を最小化するための協力体制を整える。
- アセットオーナーには、診断に伴うシステム稼働への影響や作業概要を改めて伝える。

6.4.4 診断結果の共有と対応策

- 定期診断結果を脆弱性診断チームが報告書にまとめ、アセットオーナーまたはシステム運用へ提出する。
- アセットオーナーとシステム運用は内容を確認し、危険度が高いと評価された指摘事項を優先して修正する。
- システム運用は修正対応後、修正箇所の再診断を脆弱性診断チームに依頼する。

- 脆弱性診断チームはセキュリティ統括に、年度内に実施した脆弱性診断について、年度計画通り行ったか、診断結果やその傾向等をとりまとめ報告する。

6.5 組織間セキュリティ意識の醸成

開発時・運用時の診断を定常的に実施していくうえでは、組織間の連携を促進し、セキュリティに対して当事者意識をもつ文化を育てることが重要である。以下のような施策が効果的と考えられる。

- **定期的な情報共有・報告会**
  診断結果や修正状況、他社で発生したインシデント事例などを題材に、IT 部門を含む関係組織が参加する情報共有会を定期的に開催する。具体的な事例を共有し合うことで脆弱性対応の重要性を部門横断で理解し、セキュリティ意識を高める。
- **年間計画の見直しと継続的改善**
  毎年の計画策定時に、前年度の診断実績を振り返り、診断計画や手順の見直しなどの改善点を抽出する。もし、特定のシステムに脆弱性が集中する傾向がみられる場合は、コーディング規約の見直し等をシステム開発へ提言するなど、相互に協調することが望ましい。
- **セキュア開発トレーニングの実施**
  システム開発担当者のセキュリティへの理解を深め、脆弱性を自ら防ぐ意識を高めるためには、脆弱性診断チームが主体となり、診断結果に基づくセキュア開発トレーニングを定期的かつ継続的に実施することも効果的である。

6.6 本章のまとめ

本章では、企業内で脆弱性診断を内製化し、開発から運用まで一貫して効率的かつ効果的に実施するための組織体制と連携の在り方、そしてセキュリティ意識の醸成について説明した。

まず、仮想企業の事例を通じて、セキュリティ部門（脆弱性診断チームを含む）と IT 部門（システム開発・システム運用・アセットオーナー）がどのように役割を分担し、連携するかを示した。リリース前診断や定期診断を体系的に整理し定着させることで、システムが本番稼働する前のリスク低減と運用段階における継続的なリスク管理を両立しやすくなる。

さらに、各種診断プロセス（リリース前診断・運用中の定期診断）を通じて各組織が協働し、セキュリティに対して当事者として捉える文化を醸成するための具体的な施策として、情報共有・報告会や年間計画の見直し、セキュア開発トレーニングなどを挙げた。こうした取り組みを継続することで、単に脆弱性診断を回すだけでなく、開発・運用プロセス全体がセキュアな方向へと進化していくことが期待できる。

本章で示した例はあくまでも一つのモデルであり、自社の事業規模や技術スタック、組織構造などにあわせてカスタマイズしていく必要がある。しかしいずれの場合でも、複数組織が同じ目標を共有し、対話と連携を強化することが、脆弱性診断の内製化を成功に導く鍵となる。

コラム

～診断結果に基づくセキュア開発トレーニングとは～

企業やチームによって作り込みやすい脆弱性の種類には傾向があります。例えばSQLインジェクションが多発するケースや、特定のフレームワークを使った際に権限管理が疎かになりがちなケースなどです。

まずは、事例紹介を通してそうしたパターンを共有し、次にセキュア開発トレーニングとして扱うことで、**参加者は「自分たちのコードで起きがちな問題」をリアルに理解し、修正・再発防止のノウハウを習得しやすくなります。**

さらに、リリース前レビュー時のチェック項目に「前回診断で多かった脆弱性」「セキュア開発トレーニングで学んだ対策の実装有無」を含めれば、トレーニング内容が実際の開発プロセスに結びつきやすくなります。これにより、診断担当者が指摘する以前に開発者が積極的に脆弱性を探し、**未然に防ぐ「能動的なセキュリティ文化」が形成されやすくなります。**

7 人材確保･育成

7.1 本章の目的

脆弱性診断を内製化し、継続的に高いセキュリティ品質を保つためには、幅広い知識や高度なスキルを持った人材を確保し、組織として育成していく仕組みが不可欠である。攻撃手法や技術基盤が進化するなか、脆弱性診断チームが最新情報をキャッチアップし、的確にリスクを評価していくには、安定したチーム体制と人材のモチベーション維持が要となる。本章では、脆弱性診断チームの人材確保や採用時の評価ポイント、研修やキャリア形成の仕組みなど、人材面でのアプローチを多角的に説明する。

7.2 人材確保の基本的な考え方

脆弱性診断を社内で担う人材を確保する際には、以下のような基本的な視点を押さえておくと効果的である。

- **多様なスキルセットの組み合わせ**
  攻撃手法や診断ツールに関する深い知識だけでなく、開発工程やインフラ設計、クラウドサービスの活用方法など、周辺領域の知見も重要である。脆弱性を発見・報告するだけでなく、システム開発・システム運用との協業を円滑に進め、修正提案にまで踏み込めるエンジニアが揃うと、内製化のメリットが最大限に活かされる。

- **長期的な視点でのチームビルディング**
  セキュリティの知見が日々アップデートされるなかで、組織として継続的に学習する文化を育むことが大切である。システム開発やシステム運用、アセットオーナーなどシステムの維持に関与する組織との横の連携を意識し、関係組織間でノウハウを共有できる仕組みづくりが必要である。

- **採用経路の多様化と人材候補者の確保**
  セキュリティ人材は市場でも需要が高く[16]、競合他社との取り合いになりやすいと考える。技術コミュニティやセキュリティ専門の勉強会などを通じて、広く人材と関わる場を確保するアプローチが欠かせない。SNS や企業ブログでの発信を強化することで、潜在的な候補者との接点を増やす企業も増えている。

  また、社外からの採用に加えて、社内のシステム開発部門、システム運用部門、セキュリティ統括部門などからの異動によって人材を確保することも有効である。これらの部門には、既に自社システムの構成や運用上の制約を熟知した人材が多く、セキュリティ視点の育成を通じて即戦力に育てやすい土壌がある。

- **企業文化・風土への適応力**
  セキュリティ分野は技術力も重視される一方、社内調整やドキュメント作成、システム開発・システム運用との対話など、コミュニケーション能力も不可欠な場面が多い。高い技術力があっても、企業文化に溶け込めずにパフォーマンスが発揮できない

[16] サイバーセキュリティ 2024 / 内閣サイバーセキュリティセンター(NISC) / P.31 より / https://www.nisc.go.jp/pdf/policy/kihon-s/cs2024.pdf

ケースも考えられるため、採用面接や実技試験だけでなく、企業のミッション・バリューに共感できるかといった定性的な評価も重視することが望ましい。

- **脆弱性診断業務に対する素質**
  脆弱性診断士にとって高度な技術力はもちろん必要となるが、その技術をどのように扱うかという倫理的な素質も持っている必要がある。診断とは許された範囲で攻撃者の視点による脆弱性の洗い出しを行う行為である以上、「できること」と「してよいこと」の違いを適切に判断できる倫理観が不可欠である。このような性質から、飛行機の操縦士に対して実施されるように、倫理観や責任感を見極めるためのスクリーニングを行うことも有効な手段のひとつである。

このように外部からの採用と内部からの異動の両面をバランスよく活用しつつ、多面的な評価と組織づくりの観点を持つことで、脆弱性診断チームとして必要な人材を確保し、長期的に活躍する土台を築くことができる。

7.3 新卒・中途採用のポイント

脆弱性診断チームの人材確保には、新卒・中途それぞれの特性に応じた採用基準が重要となる。ここでは、各採用で重視すべき基礎能力や実務経験といった、具体的なポイントを説明する。

7.3.1 新卒採用：基礎的エンジニアリングスキルの重視

新卒採用では、一般的に企業の人事部門で採用基準が定められているが、脆弱性診断業務への従事を考慮した採用を行う場合は、人事部門へ必要な能力に関して調整する必要がある。脆弱性診断業務に携わるうえでは、ネットワークや Web の基本構造、通信の仕組みなど、エンジニアとしての基礎的な技術知識を備えていることが望ましい。また、学生時代に CTF（Capture The Flag）やセキュリティ関連の活動に取り組んだ経験がある場合、セキュリティ分野への関心が一定程度確認できるため、採用時の参考指標として活用するのもよい。採用後は段階的な研修や OJT を通じて実案件に慣れさせ、チーム内で日々の診断プロセスや報告手順などが学べると効果的である。

7.3.2 中途採用：経験年数やコミュニケーション能力の重要性

一方、中途採用を行う場合には、即戦力として実務経験を有する人材を確保できるメリットがある。実際に業務として、脆弱性診断の作業や調整の実績があると、チーム内での業務がスムーズに進むことが期待できる。

また、脆弱性診断チームの人材要件を設定する際は、デジタル庁が公表する「政府情報システムにおける脆弱性診断導入ガイドライン[17]」や、経済産業省が策定する「情報セキュリ

[17] 「政府情報システムにおける脆弱性診断導入ガイドライン（2024）」/ デジタル庁 / https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/e2a06143-ed29-4f1d-9c31-0f06fca67afc/b08708cd/20240131_resources_standard_guidelines_guidelines_05.pdf

ティサービス基準[18]」などが参考となる。以下は政府公開文書に記載された要件をもとにした一例である。

- 脆弱性診断経験が 2 年以上ある者
- 外部とのコミュニケーションや調整業務が 2 年以上ある者
- OSCP、OSWA、GWAPT、GPEN、GMOB などの専門資格の保有者
- CTF での上位入賞実績がある者
- 脆弱性届出の実績がある者

さらに、中途採用時に応募者の適性や技術力を見極めるための課題として、「脆弱性を仕込んだ演習環境の診断」を課す方法も効果的である。具体的には、簡易的な Web アプリケーションに意図的に複数の脆弱性を埋め込み、応募者に発見・報告してもらう課題を与える。この方法により、ツール操作や手動検証の基礎力に加え、脆弱性報告の書き方やコミュニケーション能力も評価できる。

これらはあくまで一つの目安として示されているものであり、こうした客観的な基準を活用すれば、採用時の評価指標を社内でも設計しやすくなる。ただし、資格や実績だけが実力を保証するわけではないため、面接や実技課題を組み合わせて総合的なスキルを見極めることが重要である。

7.4 人材育成

人材確保も重要な観点ではあるが、確保した人材が組織としてセキュリティの専門性を継続的に高める「人材育成」の仕組みづくりも重要である。診断員は、最新技術や高度化する攻撃手法のキャッチアップを継続して取り組むことが求められる。

7.4.1 座学と実践演習による育成

脆弱性診断のベースとなるネットワークや Web アプリケーションの知識を、体系的に理解するための教育から始める。TCP/IP、HTTP、セッション管理、データベース連携など、脆弱性診断に必要な基礎知識を座学で学ぶことで、業務実施の土台を築く。

座学と並行して、研修環境内にサーバや Web アプリケーションをいちから構築し、受講者自身が脆弱性診断を行う演習を取り入れる。このような演習の中で診断から報告書作成・修正・再診断までという脆弱性対応のライフサイクルを疑似体験できる。実務に近い形で学べるため、座学だけでは身に着けにくい「実践感覚」を得ることができる。

7.4.2 実案件と並行した OJT

採用された診断員が入社後に成長する手段として有効なのは、実案件に参加しながら学ぶ OJT である。初期段階では OJT 担当がメインで診断を行いつつ、新人がツールの操作や報告書作成の補助を担当し、プロセス全体を俯瞰的に学ぶ。段階を追って、簡易なスキャンや小

18 「情報セキュリティサービス基準」 / 経済産業省
https://www.meti.go.jp/policy/netsecurity/shinsatouroku/touroku.html

規模な手動診断を任せることで、成功体験を積ませるとともにスキルの伸びを測定することができる。

7.4.3 勉強会・コミュニティ参加

社外で開催される国内外のセキュリティカンファレンスや勉強会、コミュニティ活動、外部競技会へ参加し、最新の脆弱性情報や攻撃手法を吸収する仕組みを整えることも欠かせない。最新動向をキャッチし、新たに得た知見を社内に持ち帰って共有し合うことも重要である。また、こうした場を通じて他社の専門家やエンジニアと人的ネットワークを築くことで、新たな視点の獲得や情報交換がスムーズになるなど、長期的な協力関係が生まれるメリットも大きい。

7.5 モチベーション維持とキャリアパス

高度なセキュリティスキルを持つ人材は、市場での需要が高く、十分な評価や報酬を得られないと離職リスクが高まる可能性がある。脆弱性診断チームのメンバーが長期にわたって意欲を持ち続け、組織に貢献できるようなモチベーション維持の仕組みを構築しておくことが重要である。

なお、脆弱性診断士としてのキャリア構築については、ISOG-J（日本セキュリティオペレーション事業者協議会）が公開している「脆弱性診断士のキャリアデザインガイド[19]」が参考になる。業務内容やキャリアパス、学習リソース、実態や経験談などが網羅されているため、キャリア形成を検討する際には参照するとよい。

7.5.1 評価制度

企業として脆弱性診断の成果をどのように可視化し、評価するかを明確にする必要がある。発見した脆弱性の件数や危険度など定量的に判断可能な指標に加え、システム開発との連携を円滑に進めたか、修正を促す際に技術的背景を踏まえた適切なアドバイスを行ったかなど、質的な面も含めた多面的な評価を導入することが、公平性と納得感を高めるうえで有効である。

「適切なアドバイス」とは、単に脆弱性の存在を指摘するだけでなく、対象システムの設計や業務要件を理解したうえで、システム開発側の実装方針や制約に配慮しつつ、現実的な対処案を提示できているかを評価基準とする。さらに、同じような脆弱性を再発させないための予防策まで提案できているかも重要な観点である。

加えて、マネジメント担当者に対しては、チーム運営やスケジュール管理、メンバーの成長支援の観点からの評価も設けるべきである。単に診断業務を納期通りに進めたかだけでなく、ナレッジ共有の推進、若手メンバーの OJT 体制構築、システム開発との関係性強化といった、組織的な成熟度を高める取り組みも評価軸として取り込むことで、現場のモチベーション向上と持続可能な内製体制の確立が期待できる。

---

[19]脆弱性診断士のキャリアデザインガイド / ISOG-J / https://wg1.isog-j.org/CareerDesignGuide/

7.5.2 スキル向上に対する費用補助

報奨金の支給や受験料の補助、資格維持にかかる年会費や更新費用の支援、学習書籍・外部講習費用への補助といった制度設計を通じて、企業として人材の成長を継続的に支援する姿勢を示すことができる。こうした取り組みは、診断員のモチベーション維持やスキルの向上に寄与すると考えられる。

経済産業省が公表する「情報セキュリティサービス基準[20]」では、一定の専門資格や実務経験が求められており、内製化を進める企業にとっても、外部発注と同等の品質レベルを担保するうえで、資格取得は一つの参考指標となる。脆弱性診断に従事する人材の専門性を可視化し、継続的な能力開発を促す仕組みとして、制度の整備・運用を検討することが望ましい。

7.5.3 診断員としての専門性を高める長期的なキャリアパスの提示

脆弱性診断チームのメンバーが長期間モチベーションを維持し、高い専門性を持った診断員として継続的に活躍するためには、専門職としてのキャリア設計と、希望や適性に応じた多様な経験の機会をうまく調和させることが重要である。企業としては、診断業務を専門領域として位置づけつつも、脆弱性診断に関連するシステム開発やシステム運用などの業務も経験できるような柔軟なキャリアパスを提示することで、スキルの幅を広げたい診断員の成長を後押しすることができる。

また、関連会社や取引先など外部の企業に対して脆弱性診断をサービス提供する機会を持つことも、診断スキルを対外的に活かす場として刺激となり、診断員のモチベーション向上につながる取り組みといえる。

こうした柔軟で明確なキャリアパスの提示は、結果的に人材の定着にもつながる。

7.6 本章のまとめ

脆弱性診断を内製化し、継続的に高品質な診断を実施していくためには、組織として人材確保と育成に関する体系的な取り組みが不可欠である。特に、診断員のスキルレベルやコミュニケーション力、企業文化への適合性を踏まえた採用基準の設定や、実務を想定した体系的な研修、OJT、外部コミュニティ参加など、多角的な育成施策が効果的となる。

また、診断業務の成果を適切に評価する仕組みや、スキル向上に対する費用補助といった制度面の充実、柔軟なキャリアパスの提示を行うことにより、診断員のモチベーションを高く維持することが重要である。

このように、人材確保・育成を包括的に進めることで、脆弱性診断チームの持続的な成長を支え、企業全体のセキュリティ水準を向上させることが可能となる。

---

[20] 情報セキュリティサービス基準 / 経済産業省
https://www.meti.go.jp/policy/netsecurity/shinsatouroku/touroku.html

8 ツール選定におけるポイント

8.1 本章の目的

脆弱性診断を内製化するうえで、ツールの選定とその活用方法は成功を左右する重要な要素となる。本章では、有償・無償ツールの特徴を整理し、組織の環境や運用体制に適した選択の考え方を説明する。また、AI や自動化技術など最新動向にも触れ、診断精度と効率をバランスよく保つために、ツール選定におけるポイントを示す。

8.2 有償ツールと無償ツール

ツール選定の際には、有償ツールを導入するか、無償ツールを活用するかという判断が生じる。どちらにも利点と考慮すべき課題が存在するため、組織の予算、人材のスキルレベル、診断対象の規模、求めるサポートの有無などを総合的に検討した上で決定することが望ましい。

8.2.1 有償ツールの特徴と考慮点

有償ツールの最大の利点は、安定した製品品質と充実したサポート体制にある。トラブル発生時や製品のアップデート時にはベンダーによる迅速かつ的確な支援が期待でき、日本語でのサポートやドキュメントが整備されていることも多い。また、ベンダーによる体系的な研修プログラムによるトレーニングや定期的なセミナー開催など、人材育成面でもメリットがある。

一方、有償ツールは数万から数百万円程度のライセンス費用や保守費用が必要になることが多く、予算面での負担が大きくなる場合がある。また、ベンダーが提供するプラグイン機能により、機能拡張が可能なケースはあるものの、ソースコードレベルでの自由な改変はできないため、自社独自のニーズに完全に応えられないケースも想定される。

8.2.2 無償ツールの特徴と考慮点

無償ツールは、初期導入コストが抑えられ、特にスモールスタートを検討している企業にとって導入障壁が低い。無償ツールは、オープンソースとしてソースコードや多様なプラグインが公開されている。利用者自身でのカスタマイズや機能拡張も可能であるため、柔軟性や拡張性を発揮しやすいという特徴がある。

一方、無償ツールは原則として公式なサポートがなく、トラブル時には英語のフォーラムなどを活用し、自力で問題を解決する必要がある。また、ドキュメントの整備状況やツールの安定性にばらつきがある。さらに、カスタマイズや機能拡張を行う場合、運用にはある程度高い技術力を有するエンジニアの確保が必須である。

8.3 AI・自動化技術の動向

近年、AI や機械学習を活用した自動診断技術が注目を集めている。AI 技術を応用した診断ツールには、診断対象システムを自動でクローリングしてテストシナリオを生成する機能や、ソースコードや仕様を理解したうえで自律的に診断を実施する機能が搭載された製品も登場している。これにより、従来型のツールに比べて診断の網羅性、効率性が向上しているため、診断業務の効率化や人的リソース不足の解消につながることも期待されている。

一方で、AI 技術を診断に導入する上では、現段階で考慮すべき点が存在する。例えば、従来型の自動ツールでは検出が困難なアクセス制御の不備やビジネスロジック上の脆弱性などが挙げられる。これらに対しては、AI による検知能力の向上が期待されているものの、現状の AI 技術が期待通りの成果をもたらすとは限らない。また、ツール診断の一般的な課題である誤検知や未検知が含まれる可能性も依然として考慮する必要がある。そのため、AI を活用したツールの出力をそのまま判断材料とするのではなく、診断員が専門的な知見に基づいて最終的なレビューを行い、結果の正確性や危険度を判断することが重要である。このレビューを経ずに結果をシステム開発へ共有した場合、意図しない混乱や余分な対応作業が生じる可能性もある。AI による自動化がもたらす効率性と、診断員による詳細な分析や手動診断をバランス良く組み合わせることが、現時点では効果的なアプローチであると考えられる。

今後、AI・自動化技術のさらなる発展に伴い、脆弱性診断における自動化の精度と効率は一層向上すると期待される。これらの技術進化を注視し、その特性を理解した上で必要に応じてツールの導入を検討していくことが、高品質な脆弱性診断の実現につながると考えられる。

8.4 本章のまとめ

脆弱性診断の内製化にあたり、適切なツールの選定および運用方法の確立は重要な成功要素である。本章では、有償・無償ツールの利点と考慮点を比較した。また、近年注目される AI を活用した診断ツールの可能性と限界にも触れ、効率化と診断精度のバランスを保つためには、AI 診断ツールと手動診断を併用することが現実的であると示した。

9 謝辞

本ガイドの作成にあたり、ヒアリングへのご協力ならびにレビューに貴重なお時間とご意見を賜りました皆様に、心より御礼申し上げます。皆様からいただいた専門的な知見と励ましが、本ガイドの内容をより充実したものへと導いてくださいました。

ご協力いただいた皆様

デジタル庁	西村　宗晃　様
株式会社トライコーダ	上野　宣　　様
株式会社神戸デジタル・ラボ	松田　康司　様
株式会社 STNet	セキュリティ診断チームの皆様

また、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として本プロジェクトのメンターを実施いただきました門林雄基先生、満永拓邦先生および奈良先端科学技術大学院大学の方々、東洋大学の方々につきましても、ご指導・ご助言とともに、各検証機材のご支援を賜りました。改めて御礼申し上げます。

そして、本ガイドの作成や本プロジェクトをともに実施した、下記メンバーの皆様にも感謝を伝えたいと思います。

【プロジェクトメンバー】

【リーダー】

川添　恭平

【サブリーダー】

森山　響　　佐藤　湧太

【メンバー】

結城　直也	吉田　晋久
川崎　政吾	守屋　友貴
渡邊　晃大	桂　隆一
永井　巽	和田　歩

付録 A：技術検証結果について

A.1 概要

本付録は、手動診断と各種診断ツールにおける検出事項の結果を比較し、考察をまとめたものである。

この検証の主な目的は、ツール診断と手動診断の結果の差異から、診断ツールの特性や診断結果の傾向などを理解することである。

本検証では、対象システムとして、意図的に脆弱性を含ませて構築された学習用の環境である VulnHub および BadTodo を用いた。診断ツールとしては、プラットフォーム診断ツールとして 3 製品、Web アプリケーション診断ツールとして 5 製品を準備し、これらを用いて検証を行った。

以降の章では下記の技術検証結果および検証を通して得られた知見をまとめている。

- A.2　VulnHub 環境 1 に対するプラットフォーム診断
- A.3　VulnHub 環境 2 に対するプラットフォーム診断
- A.4　BadTodo に対する Web アプリケーション診断
- A.5　検証で得られた知見

なお、本検証で採用した危険度基準は本ガイドの「表 5-1 危険度基準の例」に基づく。

手動診断は脆弱性診断の実務経験を持つメンバーが担当したが、ツール診断に関してはこれまでに利用経験のないツールも含まれているため、ツールの設定によっては本検証とは異なる結果が検出される可能性がある。

A.2 VulnHub 環境 1 に対するプラットフォーム診断

実施環境

- 対象： VulnHub の「Basic Pentesting: 1」
  - WordPress を含む Web/FTP/SSH サービスが稼働するサーバ

手動診断とツール診断の比較結果

VulnHub 環境 1 に対するプラットフォーム診断で得られた主な脆弱性について、手動診断の指摘事項と各ツールの検出結果を以下の表に示す。

凡例 「○」：検出 、「×」：検出なし

No.	危険度	プロトコル	手動診断による指摘事項	製品A	製品B	製品C
1	Critical	HTTP	WordPressの管理者に推測可能な認証情報が使用されており、任意コマンドが実行可能	×	×	×
2	Critical	FTP	脆弱性のあるFTPサービスが使用されており任意コマンドを実行できる	○	○	○
3	Medium	HTTP	脆弱性が存在するWordPressサービスが使用されている	×	×	×
4	Medium	SSH	脆弱性が存在するSSHサービスが使用されている	○	×	×
5	Medium	—	すでにサポートが終了しているOSが使用されている可能性がある	○	○	×
6	Low	SSH	SSHサービスでパスワード認証が有効	○	×	×
7	Low	FTP	FTPサービスでパスワード認証が有効	×	○	○
8	Low	HTTP	平文(HTTP)で重要情報を通信している	×	×	×
9	Low	HTTP	WordPressの管理用ログインが公開されている	×	×	×
10	Info	HTTP	セキュリティ関連レスポンスヘッダの不足	×	×	×
11	Info	HTTP	ディレクトリリスティングが有効	×	×	×
12	Info	FTP	バージョン情報が表示されている (FTP)	○	○	○
13	Info	SSH	バージョン情報が表示されている (SSH)	○	○	×
14	Info	HTTP	バージョン情報が表示されている (Apache)	○	○	×
15	Info	HTTP	デフォルトページが表示されている (Apache)	×	○	×

考察

- **ツールによる検出が困難だった脆弱性**
  WordPress に関連する検出事項（No.1, 3, 9）および同じサブディレクトリに存在したディレクトリリスティング（No.11）については、検証に用いた 3 製品のツールでは検出できなかった。これは、本検証環境の WordPress がルートディレクトリではなく特定のサブディレクトリ配下で稼働しており、多くの診断ツールがこのサブディレクトリ上に存在する対象を自動的には認識できないことが原因と考えられる。また、平文（HTTP）での重要情報通信（No.8）や、セキュリティ関連レスポンスヘッダの不足（No.10）といった Web サーバの設定やアプリケーションの応答に関わる一般的なセキュリティ項目も検出されなかった。これはプラットフォーム診断ツールの診断範囲に Web アプリケーションのレスポンスヘッダ情報や通信内容の機密性評価が含まれていないことが原因だと考えられる。

- **ツールで比較的検出された脆弱性**

  脆弱性のある FTP サービスの利用（No.2）や OS サポート終了の可能性（No.5）、FTP および SSH、Apache のバージョン情報表示（No.12, 13, 14）は、多くのツールで検出された。これらは、既知の脆弱性情報データベースやバナー情報取得といった、多くのプラットフォーム診断ツールが標準的に備えるシグネチャベースの検出ロジックで捉えやすいためと考えられる。

- **ツール間で検出結果に差異が見られた脆弱性**

  脆弱性のある SSH サービスの利用（No.4）、SSH および FTP サービスにおけるパスワード認証の有効性（No.6, 7）については、ツールによって検出可否が分かれた。これは、各ツールが参照する脆弱性データベースの違い、スキャンポリシーのデフォルト設定（例えば、認証試行の有無やその深度）、あるいは認証方式への対応状況の違いなどが影響した可能性がある。

A.3 VulnHub 環境２に対するプラットフォーム診断

実施環境

- 対象：VulnHub の「Basic Pentesting: 2」
  - Tomcat ベースの Web/AJP/SSH/SMB サービスが稼働するサーバ

手動診断とツール診断の比較結果

VulnHub 環境２に対するプラットフォーム診断で得られた主な脆弱性について、手動診断の指摘事項と各ツールの検出結果を以下の表に示す。

凡例 「○」：検出 、「×」：検出なし

No.	危険度	プロトコル	手動診断による指摘事項	製品A	製品B	製品C
1	Critical	SSH	推測可能な認証情報が使用されている	×	×	×
2	High	AJP	脆弱性のあるAJPサービスが公開されており非公開の設定ファイルを取得可能 (CVE-2020-1938)	○	○	○
3	Medium	SSH	脆弱性のあるSSHサービスが使用されている	○	×	×
4	Medium	―	すでにサポートが終了しているOSが使用されている可能性がある	○	○	×
5	Medium	SMB	Sambaのバージョンが古く脆弱性が存在する可能性がある	○	○	×
6	Medium	SMB	SMBv1が有効	○	○	×
7	Medium	HTTP (8080)	Tomcatのバージョンが古く脆弱性が存在する	○	○	○
8	Low	SSH	SSHサービスに対してパスワード認証が有効	○	×	×
9	Low	HTTP (80)	公開する必要がないファイルが公開されている	×	×	×
10	Low	SMB	任意のユーザから読み取り可能なフォルダが存在する	○	○	×
11	Low	HTTP (8080)	管理用ログイン画面が公開されている	×	×	○
12	Low	HTTP (8080)	不要なエラーメッセージ出力	×	×	×
13	Low	HTTP (8080)	平文 (HTTP) で重要情報を通信している	×	×	○
14	Info	HTTP (80)	平文 (HTTP) が有効	×	×	×
15	Info	HTTP (80)	セキュリティ関連レスポンスヘッダの不足	×	×	×
16	Info	HTTP (80)	ディレクトリリスティングが有効	×	×	×
17	Info	HTTP (80)	バージョン情報が表示されている (Apache)	○	○	×
18	Info	HTTP (8080)	セキュリティ関連レスポンスヘッダの不足	×	×	×
19	Info	HTTP (8080)	バージョン情報が表示されている (Tomcat)	○	○	○
20	Info	HTTP (8080)	デフォルトページの表示	○	○	○

考察

- **ツールによる検出が困難だった脆弱性**

  SSH における推測可能な認証情報（No.1）は、辞書攻撃等の能動的な試行を伴うため、診断ツールのデフォルト設定ではシステム負荷や検知時間を考慮して限定的な探

索に留まることが多く、検出が難しい典型例である。HTTP(80/tcp)ポートで公開されている不要ファイル（No.9）や Tomcat の不要なエラーメッセージ出力（No.12）は、不要なファイルや情報を判断するためにアプリケーションのコンテキストに対する理解が求められ、ツールによる自動判定が困難であったと考えられる。その他の HTTP 設定不備（No.14, 15, 16, 18）も、A.2 と同様にプラットフォーム診断ツールの主眼とする範囲外であった、あるいは動的なページ構造の把握が不十分だった可能性がある。これらは運用上の設定ミスや情報漏洩に繋がり得るものの、ツール単独での検出は難しい。

- ツールで比較的検出された脆弱性

  AJP サービスの脆弱性（No.2）や、Tomcat の古いバージョンに起因する脆弱性（No.7）、関連するバージョン情報やデフォルトページの表示（No.19, 20）は、全ツールで検出された。これらの情報は診断ツールの持つ脆弱性シグネチャや既知のパターンと照合しやすく、バージョン情報から機械的にリスクを判断できる典型的なケースであると分析できる。

- ツール間で検出結果に差異が見られた脆弱性

  脆弱な SSH サービス(No.3)、OS サポート終了(No.4)、Samba 関連(No.5, 6, 10)、Tomcat 管理画面(No.11)など、多くの検出事項で各ツールの検出結果に差異が見られた。これは、各ツールが内包するチェック項目やプラグインの種類・更新頻度の違い、特定サービスへの対応深度、スキャン設定のデフォルト値などが複合的に影響した結果と考えられる。例えば、Tomcat 管理画面の公開（No.11）は、一般的なパスを探索するツールと、より詳細なクロールや設定ファイルの解析を試みるツールとで差が出やすい。

A.4 BadTodo に対する Web アプリケーション診断

実施環境

- 対象：「BadTodo」
  - 代表的な脆弱性が網羅的に実装された Web アプリケーション環境

手動診断とツール診断の比較結果

「安全なウェブサイトの作り方[21]」の 11 項目に沿って、手動診断で発見された脆弱性が各ツールで検出されるかを検証した。

凡例 「○」：検出 、「×」：検出なし、「△」：部分的に検出

No.	大項目	製品D	製品E	製品F	製品G	製品H
1	SQLインジェクション	△	△	△	△	△
2	OSコマンド・インジェクション	×	×	△	×	△
3	ディレクトリ・トラバーサル	○	△	×	○	×
4	セッション管理の不備	×	○	×	○	×
5	クロスサイト・スクリプティング（XSS）	△	△	△	△	△
6	クロスサイト・リクエスト・フォージェリ（CSRF）	×	×	×	×	×
7	HTTPヘッダ・インジェクション	×	×	×	○	×
8	メールヘッダ・インジェクション	×	×	×	×	×
9	クリックジャッキング	○	×	○	○	○
10	バッファオーバーフロー（※対象外）	-	-	-	-	-
11	アクセス制御や認可制御の欠落	×	×	×	×	×

(※) 以下理由により「バッファオーバーフロー（No.10）」は、検証対象外としている
① 検証環境 BadTodo 自体に当該脆弱性が意図的に作り込まれていない
② 「安全なウェブサイトの作り方」における同項目の趣旨にもあるとおり、主に外部ソフトウェア部品の脆弱性確認と解釈されるため

[21]安全なウェブサイトの作り方 / 独立行政法人情報処理推進機構（IPA）/ https://www.ipa.go.jp/security/vuln/websecurity/about.html

考察

- **多くのツールで部分的検出または検出困難だった脆弱性**
  SQL インジェクション（No.1）やクロスサイト・スクリプティング（XSS）（No.5）といった代表的なインジェクション系の脆弱性は、多くのツールで部分的な検出（△）に留まった。これは、ツールが持つ攻撃パターンの網羅性の限界、DOM ベースのクロスサイト・スクリプティングのような静的解析では検出しにくいなど、多様な要因により全てのパターンを捉えきれないためと考えられる。OS コマンド・インジェクション（No.2）も同様の理由で限定的な検出となった。また、全てのツールが検出しなかったメールヘッダ・インジェクション（No.8）は、その性質上、脆弱性を含む場合の特徴がウェブページの HTTP レスポンスに直接現れにくく、自動診断ツールでは検知が困難となるケースであったと考えられる。

- **各ツールで検出可否に大きな差が出た脆弱性**
  ディレクトリ・トラバーサル（No.3）、セッション管理の不備（No.4）、HTTP ヘッダ・インジェクション（No.7）、クリックジャッキング（No.9）については、ツールによって検出できるものとできないものの差が明確に現れた。これらの脆弱性は、各ツールにおける検出ロジックの実装アプローチの違いや、リクエスト・レスポンス分析の詳細度、特定の攻撃パターンやキーワードに対するシグネチャの網羅範囲といった、各ツールの仕様の違いが結果に反映されたものと考えられる。

- **ロジック依存の脆弱性の検出限界**
  クロスサイト・リクエスト・フォージェリ（CSRF）（No.6）やアクセス制御の不備（No.11）は、ツールによる検出が困難であった。これらの脆弱性は、診断ツールが個々のアプリケーションの仕様を理解しなければその脆弱性を検出すること自体が難しく、現在の自動スキャン技術では対応が難しい。

A.5 検証で得られた知見

検証で実施したプラットフォーム診断および Web アプリケーション診断を通じて得られた知見を、以下に整理する。

- **手動診断による補完の有効性**
  プラットフォーム診断、Web アプリケーション診断のいずれにおいても、推測可能な認証情報の使用、特定のアプリケーション構成やビジネスロジックに起因する問題、運用上の設定ミスなどの脆弱性はツールでの網羅的な発見が困難であった。これらのギャップを埋めるためには、診断員による結果の精査、システムの構成・運用状況、アプリケーションの仕様を理解した上での手動診断による補完が有効である。

- **診断ツールの特性理解と適切な活用**
  診断ツールは、既知の脆弱性や典型的なパターンを効率的に検出するうえで有効な手段である。しかし、その検出能力や範囲、精度は、各ツールの仕様や参照する脆弱性データベース、スキャン設定（検出信頼度の閾値、スキャン深度など）に大きく左右されることが確認された。特に Web アプリケーション診断においては、SQL インジェクションやクロスサイト・スクリプティング（XSS）といった代表的な脆弱性であってもツールによる検出は部分的であり、複雑なアプリケーションでは網羅性に限界が見られた。ツールの特性（得意な領域と不得意な領域）を正確に把握し、診断対

象や目的に応じたツールの選択および適切な設定を行うことが、その効果を最大限に引き出すうえで求められる。

- **ツール選定における考慮点**

  効果的な脆弱性診断を求めるうえで、ツール選定は重要な要素であり、ツールによる脆弱性の検出能力だけでなく、以下のような観点からの評価を行うことが望ましいと考える。

  - 操作性やカスタマイズの柔軟性
  - 自動探索機能の網羅性
  - レポート品質と実用性
  - ツールタイプに応じた利用時の特有の制約
    （例：クラウド型ツールの診断対象公開要否、接続元 IP 管理の必要性）
  - 自動スキャン実行時の意図しないシステムへの影響
    （例：データ更新リスク、サービス負荷）
  - 公式ドキュメントの充実度
  - サポート体制

  また昨今では AI による自動化技術も導入されてきており、その動向に注視していくことも必要となる。

- **出力結果の精査と専門的判断の必要性**

  ツールから出力される情報は、時に膨大な量に及ぶことがあった。例えば、本検証の「A.2 VulnHub 環境 1 に対するプラットフォーム診断」において、製品 A では約 86 件、製品 B では約 128 件、製品 C では約 6 件の検出事項が出力された。これらの情報には誤検知も含まれており、同じ脆弱性でもツールによって危険度評価が異なることもあった。出力された情報をもとに診断員が内容を精査し、報告書に反映するプロセスが必要である。