継続的セキュリティ監視に向けた脆弱性調査プロセスの自動化

掛井 将平 [1)]，守屋 賢知 [2)]，伊藤 宏隆 [2)]，打矢 隆弘 [2)]，齋藤 彰一 [1)]，松尾 啓志 [2)]

1) 名古屋工業大学 サイバーセキュリティセンター
2) 名古屋工業大学 情報基盤センター

kakei.shohei@nitech.ac.jp

Automating Vulnerability Assessment Processes for Continuous Security Monitoring

Shohei Kakei[1)], Masanori Moriya[2)], Hirotaka Itoh[2)], Takahiro Uchiya[2)], Shoichi Saito[1)], Hiroshi Matsuo[2)]

1) Cybersecurity Center, Nagoya Institute of Technology
2) Information Technology Center, Nagoya Institute of Technology

概要

名古屋工業大学では申請に基づいた計算機の学外公開を許可している．これらの学外公開ホストは個別に管理されており，統一的なセキュリティ管理が課題となっている．本学では現在手動で脆弱性調査を実施しているが，手動による作業負荷が調査頻度を制限している．本稿では，脆弱性スキャナの Web API と資産管理データベースを活用した脆弱性調査自動化システムの構築と脆弱性調査プロセスの自動化について報告する．本システムは定期的なスキャンの実行，結果の自動収集，および資産管理データベースと連携した結果提示機能を有する．従来の手動調査と比較して，作業時間の削減と調査実施の頻度向上を実現しており，本システムを基盤とした継続的なセキュリティ監視体制の構築が期待される．

1 はじめに

大学は個人情報，研究データ，知的財産など多様かつ価値の高い機密情報を保有しており，これらの情報はサイバー攻撃者の標的となっている [1]．学生や教職員含む関係者の個人情報は身元詐称やフィッシング攻撃に悪用される可能性があり，研究データは知的財産の窃取や産業スパイ活動の標的となるリスクを持つ．また，大学が持つ国際的な研究ネットワークや外部組織との連携関係は，攻撃者が他の組織への侵入の足がかりとして利用する動機を与えている [2]．

サイバー攻撃に関する事例は依然として報告されている．英国の研究者らによる大学セクターにおける調査 [3] では，DDoS 攻撃や SQL インジェクション攻撃は減少傾向にあるが依然として対策が必要であること，ランサムウェアとフィッシングが主要な脅威であることが報告されている．日本でも，ランサムウェア攻撃により大学の情報システムの停止が教育・研究活動に深刻な影響を与えた事例や，研究データが漏洩した事例などが報告されている．

サイバー攻撃の侵入経路はいくつかあるが，学外からアクセス可能なホスト（学外公開ホスト）の脆弱性を悪用した侵入が一つの起点となっている．そのため，学外公開ホストに対する定期的な脆弱性調査は，組織の情報資産を保護するための必須の対策となっている．脆弱性の早期発見と適切な対応により，攻撃者の侵入を未然に防ぎ，サイバー攻撃に関するリスクを削減することが望ましい．しかし，セキュリティ部門における業務負荷は増加しており，重大アラートの見落としや対応の遅れが懸念されている [4]．

大学の情報システムは，研究室単位での独立性を特徴としており，各研究室が独自のシステム構成やソフトウェア環境を管理している．この多様性は研究活動の柔軟性を支える一方で，セキュリティ管理の観点では複雑性を生み出している．学外公開ホストについても，Web サーバ，メールサーバ，研究用データベース，リモートアクセスシステムなど，多種多様なサービスが各研究室の判断により運用されており，組織内の管理者による一元的な管理は容易でない．また，大学組織では情報システムの管理者が学部，研究科，研究所レベルで分散しており，それぞれが異なる技術的背景と管理方針を持っている．情報システム部門が全学的

なセキュリティポリシーを策定したとしても，実際の運用は各分散した管理者の理解と自主的な対応に委ねられており，結果として強制力を伴わない要請による対応となることが多い．この構造的な背景により，統一的なセキュリティ対策の実施や迅速な脆弱性対応の難易度を高めている．

名古屋工業大学では年二回の頻度で脆弱性スキャンシステム（脆弱性スキャナ）を用いた学外公開ホストの脆弱性調査を実施している．この調査では，まず，実務担当者が対象となる学外公開ホストのリストを脆弱性スキャナに設定し，実行する．設定された対象ホストに対する脆弱性スキャンが完了した後，実務担当者はスキャン結果から得られた脆弱性の深刻度に基づいて各管理者に個別に結果と修正依頼を通知する．現在の調査プロセスでは，スキャン自体は自動で実行されるものの，スキャン対象リストの作成・設定，スキャン結果の整理，各管理者への個別通知といった一連の作業が手動で行われている．これらの作業負荷により，年二回の調査頻度が限界となっている．

本稿では，継続的なセキュリティ監視に向けて，本学で使用している脆弱性スキャナ Nessus Professional (Pro) [5] の Web API と本学が運用する資産管理データベースを活用した脆弱性調査プロセスの自動化について報告する．スキャンの実行から結果の整理，管理者への通知までの一連のプロセスを自動化することで，実務担当者に依存しない脆弱性調査を実現する．この自動化により，年二回しか実施できなかった脆弱性調査の頻度を高めて，脆弱性の把握と対処を継続的に実施可能なセキュリティ監視の基盤とする．

2 背景

2.1 脆弱性調査プロセスの課題と自動化

脆弱性調査プロセスでは，スキャン対象の設定，スキャン実行の管理，結果の分析・整理，管理者への通知といった一連の作業が必要となる．これらの作業を手動で実施する場合，作業負荷が調査頻度を制約する要因となる．加えて，手動による調査は，人為的な見落としや判断の一貫性の欠如が避けられず，結果の信頼性や再現性の低下を引き起こす．実務担当者の技術レベルや経験が異なれば，脆弱性調査業務の品質にばらつきが生じる．また，手動作業では実務担当者の業務状況に依存して作業の実施が左右され，定期的な調査スケジュールの維持に支障をきたす場合がある．

Seara ら [6] は，組織におけるサイバーセキュリティの確保には，人手による煩雑な作業に依存する従来の脆弱性管理では限界があるとし，専門知識を持たない利用者でも容易に扱え，かつ低コストで導入可能な自動化手法の必要性を強調している．彼らは，オープンソースソフトウェアを基盤としたアプローチによって，ネットワーク上の資産把握から脆弱性の検出，情報整理に至るまでの一連のプロセスを効率化し，セキュリティ監視を継続的かつ組織横断的に実現するための枠組みを提案している．

脆弱性スキャナの利用そのものにも人的要因に起因する課題が存在する．Aksu ら [7] は，オープンソースの脆弱性スキャナである OpenVAS を対象に，専門家による評価と実務者によるユーザテストを組み合わせた分析を行い，ユーザインタフェースや操作手順の不備が誤用や誤解を招き，結果として誤った安心感を与える危険性を指摘している．特に，デフォルト設定の不透明さやオプションの説明不足，結果表示方法の不適切さが，検出精度や利用者の判断に直接影響することが示された．この知見は，脆弱性調査プロセスを自動化する上で，単にスキャンを定期的に実行するだけでなく，スキャナの可用性や結果の提示方法を改善し，利用者が正しく理解・解釈できる環境を整備することの重要性を示している．

2.2 大学組織における脆弱性管理の課題

大学では研究室や学部レベルでの統治が基本となっており，各自で IT システムの導入・運用を行っている．その結果，大学内には，多岐にわたる情報資産や多様なシステムが存在している．一方で，各構成主体の独立性が尊重される文化にあり，組織全体として画一的な情報セキュリティ対策を強制することが難しく，この点が攻撃者にとって優位に働き得ることが指摘されている [8]．

各研究室でシステム管理を担当する者は，高度な専門知識を持つ者から基本的な IT 操作に不安を持つ者まで，技術的なバックグラウンドは多様である．この技術レベルの多様性は，脆弱性情報の理解と適切な対応の実施において障害となりうる．専門的な脆弱性情報をそのまま提供しても，技術レベルが異なる管理者全てが適切な理解のもとで対応できるとは限らない．そのため，管理者の技術レベルに応じた段階的な情報提示や，専門知識がなくても理解できる形での脆弱性情報の提供が望ましい．このような大学組織の特性は個々の大学によっても異なることが考えられる．そのため，理論的な検討だけでなく，脆弱性調査を継続的に実行できる仕組みを構築し，実際の運用を通じて組織の特性や管理者のニーズ，実際の脆弱性修正状況を

把握しながら，段階的に改善を重ねていくアプローチが有効と考えられる．

3 関連システム

本章では，本学が脆弱性調査に利用しているシステムについて説明する．

3.1 脆弱性スキャンシステム

本学では，脆弱性調査に Nessus Professional（Pro）[5] を使用している．Nessus Pro は，Tenable 社が開発・提供する商用の脆弱性評価ツールであり，ネットワーク上のホストに対してポートスキャンを実行し，発見されたサービスに対して脆弱性の有無を検査する機能を持つ．

調査可能な脆弱性はプラグイン [9] と呼ばれる単位で管理されている．新しい脆弱性が発見されると，Tenable 社の研究スタッフがその脆弱性を解析し，プラグインとしてリリース[*1]する．Nessus Pro は 88,000 件以上の CVE（Common Vulnerabilities and Exposures）に対応しており，各種オペレーティングシステム（Windows，macOS，Linux），ネットワークデバイス，Web アプリケーション，データベースシステムなど幅広い IT 資産を調査対象としてカバーしている．脆弱性スキャナは，指定された調査対象の IP アドレスに対して，選択されたプラグインを適用したスキャンを実行する．しかし，プラグインの数は膨大であるため，Basic Network Scan や Web Application Test，Malware Scan など，複数のプラグインをまとめたテンプレートで指定することも可能である．また，スケジュール設定により，定期的なスキャンも可能である．

スキャンにより発見された脆弱性は深刻度別に分類され，Nessus Pro のダッシュボードで確認できる．ダッシュボードでは，調査対象ホストごとに，各脆弱性の CVSS（Common Vulnerability Scoring System）スコア，詳細な説明，推奨される対応策などがまとめられている．脆弱性の深刻度評価は，CVSS に基づく基準に従って実施される．CVSS スコアは脆弱性の深刻度を 0.0 から 10.0 の数値で表現するもので，CVSS スコアを考慮して Critical，High，Medium，Low の 4 段階で分類されている．これにより，発見された脆弱性の対応優先度を客観的に判断できる．

Nessus Pro では，スキャンの設定やスキャン結果の取得などの操作を Web API を通じて実行できる．Web API は RESTful 形式で設計されており，HTTP リクエストを通じて JSON 形式でデータの送受信が行われる．Web API によるスキャンの作成・実行・削除などの操作は Nessus Manager が必要であるが，スケジュール機能と組み合わせることで，脆弱性調査の定期実行が可能である．

[*1] 2025 年 9 月の時点で 194,000 個を超えるプラグインがリリースされている．

3.2 資産管理データベース（MAINS DB）

名古屋工業大学情報基盤センターでは，学内ネットワーク（LAN）としてキャンパス情報ネットワーク（MAINS）を構築し，その運用，保守を行っている．MAINS には研究室や個人の PC，スマートフォンなどが接続され，教育や研究，事務などに活用されている．MAINS に接続して，MAINS 内のリソースやインターネットを利用するには計算機の認証が必要であり，有線接続の場合は MAC アドレス認証，無線接続の場合は IEEE 802.1X 認証が適用される．MAINS DB は MAC アドレス認証で MAINS に接続する計算機を管理する資産管理データベースであり，MAC アドレスを MAINS DB に登録することで，明示的な認証なしに MAINS に接続できる．学内に常設される計算機や毎日持参する計算機など，日々の活動で利用される計算機が登録されている．

MAINS DB に MAC アドレスを登録する際に，当該計算機の利用者情報を登録する必要がある．計算機が登録されると，固定 IP アドレスが割り当てられ，図 1 に示すように，MAINS DB 上で MAC アドレス，IP アドレス，利用者情報などを確認できるようになる．脆弱性調査では，調査対象の IP アドレスのリストと脆弱性発見時の連絡先として利用者の情報を取得するために MAINS DB を使用している．

本学では，先行研究において，MAINS DB に OS 更新状況を表示する機能を実装している [10]．この実装では，Microsoft Defender for Endpoint（MDE）から取得した計算機の OS 情報と MAINS DB に登録されている計算機情報を MAC アドレスと IP アドレスで対応付け，各計算機の OS 更新の要否を MAINS DB 上に表示する．計算機の OS バージョン情報を取得し，当該バージョンのリリース日から 2 か月以内であれば「更新不要（Pass）」，2 か月を超えている場合は「要更新（Error）」としてアイコンで表示される．この枠組みにより，計算機の利用者と管理者は MAINS DB の Web インタフェースを通じて，自身が管理する計算機の OS 更新状況を確認することが可能となっている．今回構築する脆弱性調査システムでは，この枠組みに沿って，脆弱性調査の結果を計算機の利用者に

図 1　MAINS DB の Web インタフェースのサンプル

提示する．

4　現状の脆弱性調査プロセスとその問題

4.1　現状の脆弱性調査プロセス

名古屋工業大学では，学外公開ホストに対する脆弱性調査を年二回（4 月と 10 月）実施している．現行の調査プロセスは，複数の手動作業を含む段階的な手順により構成されている．

調査の第一段階では，MAINS DB から学外公開ホストのリストを抽出し，スキャン対象を特定する．抽出された情報には，IP アドレス，利用者情報などの基本的な情報が含まれている．2025 年 4 月の調査では，300 台のホストが調査対象であった．

第二段階では，Nessus Pro の Web インタフェースから脆弱性スキャンを実行する．ここでは，第一段階で抽出した対象ホストの IP アドレスの入力，スキャンテンプレートの選択を行う．スキャンテンプレートとしては，Basic Network Scan と Web Application Test の標準テンプレートを利用している．

第三段階では，スキャン結果を Nessus Pro のダッシュボードで確認する．ダッシュボードでは，スキャン結果の概要として，各ホストで発見された脆弱性の数を深刻度ごとに確認できる．本学では，深刻度が Critical および High の脆弱性を修正の対象としており，それらの脆弱性を有するホストをリストアップする．

第四段階では，各管理者への個別通知として，脆弱性の概要，詳細，修正方法などをメールで送信する．脆弱性に関する各種情報は Nessus Pro のダッシュボードから取得可能であり，これらの情報をメール送信に適した形に整形して，各ホストの管理者にメールで修正の依頼を行う．

最終段階では，管理者からの脆弱性修正の完了報告を受け，対象ホストに対する再スキャンを実施する．脆弱性が解消されていることを確認できれば対応完了とする．再スキャンの結果，脆弱性の修正が不十分であった場合は，追加の修正作業が必要である旨を管理者に再度通知し，修正が完了するまでこれを繰り返す．この確認のプロセスにより，脆弱性対応の実効性を保証している．

4.2　手動プロセスにおける作業負荷と制約

現行の手動による脆弱性調査プロセスでは，複数の段階において人的作業負荷が発生しており，これが調査実施の制約要因となっている．スキャン対象の設定では，MAINS DB から学外公開ホストの一覧は取得可能であるが，Nessus Pro への手動設定時に人為的ミスが発生する可能性がある．具体的には，IP アドレスの入力間違い，スキャン対象の設定漏れ，スキャンテンプレートの誤選択などが想定される．設定ミスは脆弱性の見落としに繋がるため，セキュリティリスクを高める要因となる．さらに，一回のスキャンで全ての学外公開ホストをスキャンできるとは限らない．あらかじめ脆弱性調査の実施時期はアナウンスしているが，スキャン対象のホストがシャットダウンされていることもある．その場合は管理者へ連絡し，スキャン未完了のホストに対して再スキャンを行う．この追加の作業では，スキャン未完了のホストに対して脆弱性スキャナの設定からスキャン結果の通知までの同様の作業を繰り返すことになる．

修正依頼作業では，スキャン結果から各管理者向け

の脆弱性情報として，修正方法，概要，詳細，出力メッセージを抽出する作業が煩雑である．これらの情報は Nessus Pro のダッシュボードの決められた箇所から取得できるものの，各管理者の管理対象ホストごとに該当する脆弱性情報を抽出し，メール送信に適した形に整理する作業には時間を要する．Critical および High の脆弱性のみを通知対象とする運用により，個別の脆弱性の内容にまで踏み込んだ整理は必要ないが，依然として作業負荷が高い．

また，脆弱性の修正作業に時間を要する場合，修正確認時に別の新たな脆弱性が検出されることがある．この場合，新しい脆弱性情報の通知作業が必要となり，管理者との個別対応が発生する．

5 構築した脆弱性調査システム

5.1 設計方針とシステム概要

本報告では，4 章で述べた手動プロセスにおける作業負荷と制約を解決するため，脆弱性調査プロセスを自動化するシステムを構築した．従来の調査プロセスでは，スキャン対象の設定，結果の整理，管理者への個別通知といった一連の作業が人手で行われており，調査頻度を年二回に制約する要因となっていた．本システムでは，Nessus Pro の Web API と本学の資産管理データベースを組み合わせてこの問題を解決する．

Nessus Pro では，Web API によるスキャンの作成や設定，実行などの操作が制限されているため，完全な Web API ベースでのスキャンの管理ができない．具体的には，スキャン対象の設定は手動で行う必要があり，MAINS DB の学外公開ホストのリストとスキャナに設定したスキャン対象を自動で同期できない．一方で，設定されたスキャン対象の一覧の Web API での取得やスキャンのスケジュール実行は可能である．そこで，スキャンの設定は従来通り手動で行い，定期実行については Nessus Pro のスケジュール機能を利用することとした．MAINS DB に登録された学外公開ホストが更新されると，スキャナに設定したスキャン対象との齟齬が生じる．そこで，Nessus Pro の Web API を使って定期的にスキャン対象の一覧を取得して，MAINS DB の学外公開ホストとの一致をチェックしている．

図 2 に構築した脆弱性調査システムの構成を示す．本システムは MAINS DB，Nessus Pro，脆弱性データ処理部，脆弱性レポート生成部の五つの要素から構成される．MAINS DB は学外公開ホストの情報を管理し，Nessus Pro はスキャンのスケジュール実行とスキャン結果の生成を担う．脆弱性データ処理部は，Nessus Pro からスキャン結果を取得し，ホストごとに脆弱性データをまとめる．また，ホストごとの脆弱性スキャンの状態（脆弱性の有無やスキャン日時など）を MAINS DB に記録する．加えて，発見された脆弱性の情報をホストごとに脆弱性履歴管理部に保存する．脆弱性履歴管理部はリレーショナルデータベース（RDB）で実装されたシステムであり，スキャン実施日ごとの脆弱性データを履歴として管理する．MAINS DB は Web インタフェースを通して，各ホストの脆弱性スキャンの状態を学外公開ホストの管理者に提供する．ホスト管理者が脆弱性の詳細を要求すると，脆弱性レポート生成部は脆弱性レポートのテンプレートに脆弱性データを埋め込んで，脆弱性レポートを生成し，MAINS DB の Web インタフェース上に表示する．

5.2 脆弱性データの処理

脆弱性データ処理部は Ruby で実装されたプログラムで，図 3 に示す Nessus Pro の三つの API エンドポイントを組み合わせて，ホストごとに脆弱性データをまとめる．脆弱性データ処理部は，まず，API エンドポイント`/scans/<SCAN_ID>`を用いて，スキャン結果の概要を JSON 形式で取得する．API レスポンスには，スキャン対象ホストごとにその IP アドレスや識別子 `HOST_ID` などが含まれる．次に，ホストごとのスキャン結果の概要を API エンドポイント`/scans/<SCAN_ID>/hosts/<HOST_ID>`から取得する．API レスポンスには，発見された脆弱性を示すプラグインの識別子 `PLUGIN_ID` や脆弱性の深刻度などが含まれている．最後に，各脆弱性に対して脆弱性の概要や修正方法などの詳細情報を API エンドポイント`/scans/<SCAN_ID>/hosts/<HOST_ID>/plugins/<PLUGIN_ID>`から取得する．以上の処理を経て，ホストごとに発生した脆弱性の情報を脆弱性データとして整理する．

5.3 脆弱性レポートの表示

従来の個別メール通知に代わり，各管理者に脆弱性情報を提供するシステムを Python で構築した．脆弱性レポート生成部は，脆弱性データ処理部から受け取った脆弱性データをもとに，ホストごとに脆弱性レポートを html ファイル（図 4）として作成する．本レポートは MAINS DB の Web インタフェース上で確認できる．3.2 節で述べたように，MAINS DB には OS 更新状況（図 1）を表示する機能が実装されている．この既存の枠組みを活用して「脆弱性スキャンの状態」を表示する機能を追加しており，表 1 に示す

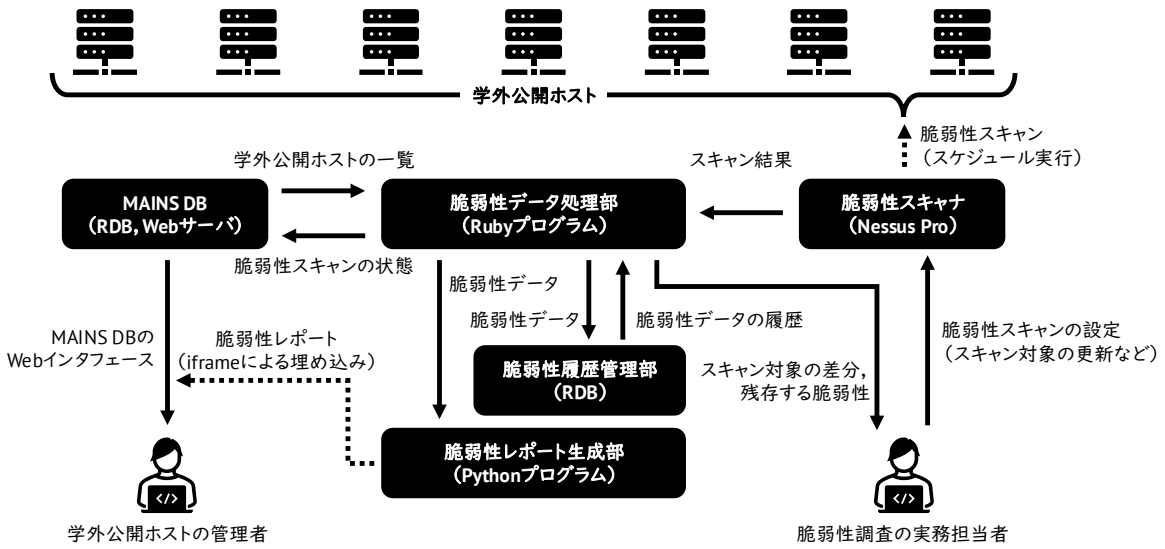

図 2　構築した脆弱性調査システムのアーキテクチャ

Endpoint: /scans/<SCAN_ID>

```
{
  “hosts”: <IPアドレスとホストIDの配列>
}
```

Endpoint: /scans/<SCAN_ID>/hosts/<HOST_ID>

```
{
  “vulnerabilities”: [
    {
      “plugin_name”: <脆弱性の名称>,
      “plugin_id”: <プラグインID>,
      “severity”: <深刻度>,
      ...
    }, {...}, ...
}
```

Endpoint: /scans/<SCAN_ID>/hosts/<HOST_ID>/plugins/<PLUGIN_ID>

```
{
  “pluginattributes”: [
    {
      “synopsis”: <脆弱性の概要>,
      “description”: <脆弱性の詳細>,
      “solution”: <脆弱性の修正方法>,
      “output”: <スキャナの出力>,
      ...
    }, {...}, ...
}
```

図 3　脆弱性データ処理部でアクセスする API エンドポイントとレスポンス

表 1　脆弱性スキャンの状態の説明

Icon	Status	Message
✔	Pass	修正が必要な脆弱性なし
⚠	Vulnerable	修正が必要な脆弱性あり
🔔	Error	脆弱性スキャン失敗
⛔	NA	調査対象外

状態が定義されている．修正が必要な脆弱性がある状態（Status が Vulnerable）でアイコンをクリックすると，iframe で埋め込まれた脆弱性レポートを MAINS DB の Web インタフェース上で閲覧できる．

5.4　スキャン対象の整合性確認

Nessus Pro の Web API を用いて，現在のスキャン設定情報を自動で取得し，MAINS DB の学外公開ホストの一覧と比較する機能を実装した．API エンドポイント/scans/<SCAN_ID>を通じて，SCAN_ID で指定したスキャンの設定を JSON 形式で取得できる．取得した JSON データから，スキャン対象として設定された IP アドレスのリストを抽出する．MAINS DB からは，SQL クエリにより学外公開ホストとして設定された IP アドレスのリストを取得する．両者に不一致があれば，脆弱性調査の実務担当者に通知し，担当者がスキャン対象を最新の状態に更新する．

本機能は定期的に実行されるように設計されており，学外公開ホストの変更に対するスキャン設定の更新を可能にしている．脆弱性スキャンが行われる前に整合性チェックを行い，なるべく最新の状態でスキャンできるようなスケジュールを組んでいる．

6　期待される関係者の行動変化

5 章で構築したシステムの導入により，脆弱性調査の実務担当者と学外公開ホスト管理者の業務プロセスに変化が生じることが期待される．本章では，システム利用時に想定される行動パターンの変化と，それによる従来の問題の解決効果について議論する．

6.1　センター実務担当者の行動変化

従来，実務担当者は年二回の脆弱性調査において，調査スケジュールの計画・調整，学内への調査実施の事前周知，手動でのスキャン対象設定，結果の整理，個別メール作成・送信といった一連の作業を集中的に

図 4　脆弱性レポートのサンプル

実施していた．構築したシステムでは，実務担当者は基本的に作業は不要となり，システムからのアラート通知に応じて対応を行う運用パターンに変化する．

5.4 節で実装したスキャン対象の整合性確認機能により，MAINS DB と Nessus Pro のスキャン設定に不一致が生じた際にアラートとして通知を受け取る．実務担当者はこのアラートに基づいて Nessus Pro のスキャン設定を最新の状態に更新する作業を行う．また，脆弱性の履歴管理機能により，一定期間残存している脆弱性に対してシステムがアラートを生成して，実務担当者とホスト管理者に通知される．このアラートにより実務担当者は当該ホストを要チェック対象として注意を向けておき，必要に応じてホスト管理者の脆弱性の修正作業をフォローすることとなる．

この運用パターンの変化により，4 章で述べた段階的な手動プロセス（スキャンの実施，結果の整理，個別通知）が不要となる．従来は年二回の調査期間中に集中していた作業負荷が自動化により解消され，より効率的な業務運営が期待できる．さらに，システムによる自動化は実務担当者の技術レベルや経験に依存する作業品質のばらつきを解消し，属人性を排除できる．システムは普段自動実行され，注意が必要な問題についてのみ実務担当者に通知される．その結果，実務担当者の状況に依存しない継続的なセキュリティ監視体制の実現が期待できる．

6.2　学外公開ホスト管理者の行動変化

学外公開ホストの管理者は MAINS DB にアクセスすることで，自身の管理ホストの脆弱性情報を確認できる．MAINS DB のセキュリティ情報で脆弱性スキャンの状態を視覚的なアイコン（表 1）で提示することで，管理者は一目で自身の管理ホストに修正が必要な脆弱性が存在するかどうかを把握できる．この視覚的指標により，脆弱性の有無を容易に認識できる．警告マークが表示された場合，アイコンをクリックして図 4 の脆弱性レポートで詳細情報を確認する．6.1 節で述べた一定期間残存する脆弱性に対するアラート

機能により，重要な脆弱性については自動的にメール通知を受け取る．これにより，日常的な確認作業に加えて，緊急性の高い脆弱性については確実に管理者の注意を喚起する仕組みが構築される．

この利用パターンの変化により，積極的な修正を望む管理者らは，通知を待つ受動的な対応から能動的な対応への行動変化が期待できる．従来は約六ヶ月間の調査間隔があったため，その間に発生した新たな脆弱性については次回調査まで把握できなかったが，継続的なスキャン実行により常に最新の情報にアクセス可能となる．管理者は自身の判断とタイミングで脆弱性状況を確認でき，センターからの通知を待たずに対応が可能となる．なお，一定期間残存する脆弱性に対しては，修正依頼を自動通知する仕組みで対応状況を管理することで，脆弱性調査業務の実効力を確保する．

一方で，従来の年二回の集中的な対応から継続的な監視への移行により，ホスト管理者の負担が増加する懸念がある．従来は調査期間に集中して対応すればよかったが，構築したシステムでは年間を通じて継続的な対応が求められることで作業負荷が高まる可能性がある．この問題に対して，対応優先度の明確化や修正方法の分かりやすい提示など，作業負担を低減する工夫が必要と考えられる．

7 おわりに

本稿では，継続的なセキュリティ監視体制の構築に向けて，名古屋工業大学における脆弱性調査プロセスの自動化の取り組みを報告した．本学では，脆弱性スキャナ Nessus Professional（Pro）を用いて，学外公開ホストの脆弱性調査を行なっているが，実務担当者による手動調査のため，年二回の頻度での調査が限界であった．そこで，本学が運用する資産管理データベースと Nessus Pro の Web API の機能を活用して，脆弱性調査を自動で実施するシステムを開発した．自動化により，作業時間の削減に加え，従来の年二回の調査では把握できなかった脆弱性の検知状況や脆弱性の修正状況を継続的に記録できるようになった．今後は，その記録を用いた脆弱性の発生傾向や修正状況の推移の分析を行う予定である．

参考文献

[1] Trend Micro，“被害事例とリサーチから見る教育機関を狙うサイバー攻撃の動向”，May 2023，https://www.trendmicro.com/ja_jp/jp-security/23/e/securitytrend-20230502-01.html（アクセス日：2025-09-23）．

[2] M. T. Intelligence，“Cyber Signals Issue 8 — Education under siege: How cybercriminals target our schools,” Oct. 2024, https://www.microsoft.com/en-us/security/blog/2024/10/10/cyber-signals-issue-8-education-under-siege-how-cybercriminals-target-our-schools/（アクセス日：2025-09-23）．

[3] H. S. Lallie, A. Thompson, E. Titis, and P. Stephens, “Analysing Cyber Attacks and Cyber Security Vulnerabilities in the University Sector,” *Computers*, vol. 14, no. 2, 2025. [Online]. Available: https://www.mdpi.com/2073-431X/14/2/49

[4] 独立行政法人情報処理推進機構, “セキュリティ業務の自動化推進”，情報処理推進機構, July 2024, https://www.ipa.go.jp/jinzai/ics/core_human_resource/final_project/2024/automation-of-security-operations.html（アクセス日：2025-09-23）．

[5] Tenable，“Nessus Professional による詳細な脆弱性評価”，https://jp.tenable.com/products/nessus/nessus-professional（アクセス日：2025-09-24）．

[6] J. P. Seara and C. Serrão, “Automation of system security vulnerabilities detection using open-source software,” *Electronics*, vol. 13, no. 5, p. 873, 2024.

[7] M. U. Aksu, E. Altuncu, and K. Bicakci, “A first look at the usability of openvas vulnerability scanner,” in *Workshop on usable security (USEC)*, 2019.

[8] 内閣サイバーセキュリティセンター，“サイバーセキュリティ 2024（2023 年度年次報告・2024 年度年次計画）”，サイバーセキュリティ戦略本部, July 2024. [Online]. Available: https://www.nisc.go.jp/pdf/policy/kihon-s/cs2024.pdf

[9] Tenable, Inc., “Nessus Plugins Database,” Tenable, Inc., https://jp.tenable.com/plugins（アクセス日：2025-09-20）．

[10] 掛井 将平，守屋 賢知，齋藤 彰一，松尾 啓志，“Microsoft Defender for Endpoint を用いた OS 更新状況可視化システムの構築”，大学 ICT 推進協議会年次大会論文集，vol. 2023，pp. 531–538，Dec. 2023.