NIST Special Publication 800-40
Revision 3

エンタープライズ向け
パッチ管理技術ガイド

Murugiah Souppaya
Karen Scarfone

http://dx.doi.org/10.6028/NIST.SP.800-40r3

コンピュータセキュリティ

ZJTP2020212180

NIST Special Publication 800-40
Revision 3

エンタープライズ向け
パッチ管理技術ガイド

Murugiah Souppaya
米国国立標準技術研究所情報技術
ラボラトリ コンピュータ
セキュリティ部門

Karen Scarfone
Scarfone Cybersecurity
Clifton, VA

http://dx.doi.org/10.6028/NIST.SP.800-40r3

2013 年 7 月

米国商務省 長官
Penny Pritzker

米国国立標準技術研究所
標準技術担当商務長官
Patrick D. Gallagher

作成機関

この文書は、Federal Information Security Management Act（連邦情報セキュリティマネジメント法、以下 FISMA と称す）、公法 107-347 に基づくその法的責任を推進するために、NIST により作成された。NISTは、連邦情報システムの最低限の要求事項を含んだ情報セキュリティ標準およびガイドラインを作成する責務があるが、このような標準およびガイドラインは、国家安全保障に関わるシステムに対し政策を実施する権限を有する、連邦政府当局者の明示的な承認なしに、そのようなシステムには適用されない。このガイドラインは、行政管理予算局（OMB: Office of Management and Budget） Circular A-130、第 8b（3）項、『政府機関の情報システムの保護（*Securing Agency Information Systems*）』の要求事項に一致しており、これは A-130 の付録 IV「重要部門の分析」で分析されているとおりである。補足情報は、A-130、付録 III「連邦政府の自動化された情報リソースのセキュリティ（*Security of Federal Automated Information Resources*)」に記載されている。

本文書における一切は、商務長官が法的権威に基づき連邦政府に対して義務および拘束力を与えた標準およびガイドラインを否定するものではない。また、これらのガイドラインは、商務長官、行政管理予算局長、またはほかのすべての連邦政府当局者の既存の権威に変更を加えたり、これらに取って代わったりするものと解釈してはならない。本文書は非政府組織が自由意志で使用することもでき、米国内での著作権の制約はない。ただし、NIST に帰属するものとする。

米国国立標準技術研究所、Special Publication 800-40 Revision 3
米国国立標準技術研究所Spec.Publ.800-40 Rev. 3、26 ページ（2013 年 7 月）
http://dx.doi.org/10.6028/NIST.SP.800-40r3
CODEN: NSPUE2

この文書に関するコメントの送付先:

米国国立標準技術研究所
宛先: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

コンピュータシステムの技術に関する報告書

米国国立標準技術研究所（NIST: National Institute of Standards and Technology、以下、NIST と称す）の情報技術ラボラトリ（ITL: Information Technology Laboratory）は、国家の測定および標準に関する基盤において技術的リーダーシップを提供することにより、米国の経済と公共福祉に貢献している。情報技術ラボラトリは、テストの実施、テスト技法の開発、参照データの作成、実装によるコンセプト実証、技術的分析を通じて、情報技術の開発と生産的利用の発展に努めている。情報技術ラボラトリの責務は、連邦政府の情報システムにおいて、費用対効果の高いセキュリティを施し、国家安全保障に関わらない情報のプライバシを確保するための技術的、物理的、管理的および運用のための標準とガイドラインを策定することにある。NIST Special Publication 800 シリーズでは、情報システムセキュリティにおける情報技術ラボラトリの調査、ガイドライン、普及活動並びに産業界、政府機関および教育機関との共同活動について報告する。

概要

パッチ管理とは、製品およびシステム向けのパッチを特定し、取得し、インストールし、検証するプロセスのことである。パッチは、ソフトウェアおよびファームウェアのセキュリティと機能の問題を修正する。パッチ管理は、いくつかの課題によって複雑になる。組織がこれらの課題を解決できなければ、システムに効果的かつ効率的にパッチを適用できず、容易に回避できる侵害が発生する可能性がある。この文書の目的は、組織がエンタープライズ向けパッチ管理技術の基本事項を理解できるよう支援することにある。ここでは、パッチ管理の重要性を説明し、パッチ管理の実行に伴う課題を確認する。また、エンタープライズ向けパッチ管理技術の概要についても説明するとともに、この技術の効果を測定し、パッチの相対的な重要度を比較するためのメトリクスについても簡単に論じる。

キーワード

情報セキュリティ、パッチ管理、修正措置、ソフトウェアパッチ、脆弱性管理

謝辞

本書執筆者である Murugiah Souppaya（NIST）および Karen Scarfone（Scarfone Cybersecurity）は、本書の草稿を査読し、技術的内容に貢献してくれた同僚達、特に Peter Mell（NIST）に感謝の意を表したい。

謝辞 2

本書執筆者である Peter Mell（NIST）、Tiffany Bergeron（MITRE Corporation）、および David Henning（Hughes Network Systems, LLC）は、本文書の作成を支援してくれた Rob Pate（US-CERT: United States Computer Emergency Readiness Team）に感謝の意を表したい。さらに、本文書の初版の共同執筆者であり、今回の版にも重要なアドバイスを提供してくれた Miles Tracy（米国連邦準備制度理事会）と、付録に掲載したパッチ適用のリソースを収集してくれた Tanyette Miller（Booz Allen Hamilton）にも感謝したい。また、洞察に満ちた論評を提供してくれた Timothy Grance（NIST）、Manuel Costa と Todd Wittbold（MITRE Corporation）、Matthew Baum（the Corporation for National and Community Service）、および Karen Kent（Booz Allen Hamilton）と、特に有益な意見や提案を提供してくれた保健社会福祉省、国務省、環境保護局、連邦準備制度理事会、および PatchAdvisor の代表者にも、感謝の意を表したい。

商標について

目次

付録

要旨

パッチ管理とは、製品およびシステム向けのパッチを特定し、取得し、インストールし、検証するプロセスのことである。パッチは、ソフトウェアおよびファームウェアのセキュリティと機能の問題を修正する。パッチは、ソフトウェアの欠陥による脆弱性を軽減するため、ほとんどの場合、セキュリティの観点からは有益である。パッチを適用してこのような脆弱性を排除することで、悪用の機会が大幅に減る。パッチには、ソフトウェアの欠陥の単なる修復以外にも目的があり、セキュリティ機能など、ソフトウェアやファームウェアに新しい機能を追加することもできる。

パッチ管理は、いくつかの課題によって複雑になる。これらの課題を解決できない組織は、システムに効果的かつ効率的にパッチを適用できず、容易に回避できる侵害が発生する可能性がある。パッチの処理に費やす時間を最小限に抑えることができる組織は、この時間をほかのセキュリティの問題の解決にあてることができる。すでに多くの組織がパッチ管理を大規模に運用しており、これをセキュリティの一環ではなく、主要な IT 機能にしている。
しかし、どの組織にとっても、パッチ管理をセキュリティの観点から慎重に検討することが依然として重要である。なぜなら、強固なセキュリティを達成し、維持するためにはパッチ管理が非常に重要であるからである。

この文書の目的は、組織がエンタープライズ向けパッチ管理技術の基本事項を理解できるよう支援することにある。ここでは、パッチ管理の重要性を説明し、パッチ管理の実行に伴う課題を確認する。また、エンタープライズ向けパッチ管理技術の概要について説明するとともに、この技術の効果を測定し、パッチの相対的な重要度を比較するためのメトリクスについても簡単に論じる。

組織は、エンタープライズ向けパッチ管理技術の効果と効率を向上させるために、以下の推奨事項を実施する必要がある。

各組織は、段階的な方法でエンタープライズ向けパッチ管理ツールを導入すること。

このアプローチにより、パッチ適用の全面的な導入に先立ち、小さなグループを使ってプロセスの問題やユーザとのコミュニケーションに関する問題を解決できる。ほとんどの組織では、パッチ管理ツールを導入する際に、まずは標準化されたデスクトップシステム、および同様の構成を持つサーバ群により構成される単一プラットフォームのサーバファームを対象とする。これが完了すると、次に組織は、マルチプラットフォーム環境、標準でないデスクトップシステム、レガシーコンピュータ、および例外的な構成のコンピュータの統合という、より難しい問題に取り組む必要がある。自動化されたパッチ適用ツールが適用できないオペレーティングシステムやアプリケーションのほか、例外的な構成のコンピュータに対しては、手作業による方法が必要となる場合がある。

各組織は、エンタープライズレベルのアプリケーションの導入時に使用すべき標準的なセキュリティ手法によって、エンタープライズ向けパッチ管理ツールに関連するリスクを削減すること。

組織内にエンタープライズ向けパッチ管理ツールを導入することにより、新たなセキュリティリスクが生まれる可能性がある。しかし、システムにパッチを効果的に適用していない組織は、それよりもはるかに大きなリスクに直面する。このようなツールは、特にセキュリティリ

スクや脅威からシステムを保護するためのセキュリティ対策機能がツールに組み込まれている場合、セキュリティの向上がセキュリティの低下を大きく上回るのが通常である。これらのツールに関連するリスクには、パッチの改変、資格情報の誤用、ツールの脆弱性の悪用、脆弱性を特定するための他者によるツールの通信状況の監視などがある。このようなリスク対策の例として、パッチ適用ソリューションのコンポーネントの厳重な保護と最新の状態の維持、ネットワーク通信の暗号化、インストール前のパッチの完全性の検証、導入前のパッチのテストなどを挙げることができる。

各組織は、セキュリティのニーズと、使用性および可用性に対するニーズとのバランスをとること。

たとえば、パッチのインストールによってほかのアプリケーションが「中断」されることがあるが、これは導入前のパッチのテストによって適切に対応することができる。また、アプリケーションの強制的な再起動、オペレーティングシステムの再起動、およびほかのホスト状態の変化によって混乱が生じ、データやサービスが失われる可能性もある。この場合も、組織はパッチ適用のニーズと運用のサポートに関するニーズとのバランスをとる必要がある。最後の例として、モバイル機器にとって特に重要な、低帯域幅または従量制課金接続を経由した更新の取得を挙げることができる。そのような接続を介したサイズの大きいパッチのダウンロードは、技術的にも経済的にも不可能である。組織には、モバイルホストや、低帯域幅または従量制課金ネットワークで使用されているほかのホストでエンタープライズ向けパッチ適用ソリューションを確実に使用できるよう準備が必要である。

1. 序論

1.1 目的および有効範囲

この文書の目的は、組織がエンタープライズ向けパッチ管理技術の基本事項を理解できるよう支援することにある。この文書では、組織にパッチ管理の能力が十分にあり、自動化のレベルを上げることに注目しているものと想定している。パッチ管理プログラムの確立に関する基本的なガイダンスが必要な場合や、現在のエンタープライズ向けパッチ管理技術では対応できない従来からのニーズがある場合、この文書に加えて、以前の補足バージョン、NIST SP 800-40 Version 2、『パッチおよび脆弱性管理プログラムの策定』も参照のこと。[1]

1.2 対象読者

この文書は、セキュリティパッチの取得、テスト、優先順位付け、実装、検証を担当するセキュリティ管理者、エンジニア、管理者を対象としている。また、監査員や、システムのセキュリティを評価する必要があるほかの担当者にとってもこの文書は有効である。

1.3 構成

この文書は、次のセクションと付録で構成されている。

- セクション 2 では、パッチ管理の重要性について説明する。
- セクション 3 では、パッチ管理の実行に伴う課題を確認する。
- セクション 4 では、エンタープライズ向けパッチ管理技術の概要を提供する。
- セクション 5 では、パッチ管理技術の効果を測定し、パッチの相対的な重要性を比較するために使用できるメトリクスについて簡単に論じる。
- 付録 A では、セキュリティ設定共通化手順（SCAP）と、エンタープライズ向けパッチ管理におけるその役割に関するチュートリアルを提供する。
- 付録 B に、この文書に記載されている主な推奨事項をまとめる。
- 付録 C に、この文書で使用した主な頭字語やほかの略語の定義を示す。

[1] http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf

2. パッチ管理の重要性

パッチ管理とは、製品およびシステム向けのパッチを特定し、取得し、インストールし、検証するプロセスのことである。パッチは、ソフトウェアおよびファームウェアのセキュリティと機能の問題を修正する。パッチは、ソフトウェアの欠陥による脆弱性を軽減するため、ほとんどの場合、セキュリティの観点からは有益である。パッチを適用してこのような脆弱性を排除することで、悪用の機会が大幅に減る。また、通常パッチは、ソフトウェアの欠陥による脆弱性を軽減するための最も効果的な方法であり、多くの場合、完全に効果的な唯一のソリューションである。場合によっては、ソフトウェアやセキュリティ管理策の再構成を含む一時的な回避策など、パッチの代わりになるものがあるが、これらの回避策は、機能に悪影響を与えることが往々にしてある。

パッチには、ソフトウェアの欠陥の単なる修復以外にも目的がある。セキュリティ機能など、ソフトウェアやファームウェアに新しい機能を追加することもできる。アップグレードでも新しい機能を追加できるが、この場合はソフトウェアやファームウェアのバージョンが新しくなり、単なるパッチの適用よりも大幅な変更が加わる。
アップグレードによっても、旧バージョンのソフトウェアおよびファームウェアのセキュリティや機能の問題を修復できる。また、ベンダーは自社製品の旧バージョンのサポートを停止することがよくある。つまり、新たな脆弱性に対処するためのパッチのリリースも停止されることになるため、サポートされない旧バージョンはセキュリティが徐々に低下していく。
そのような製品を、パッチが適用され、新たに検出された脆弱性へのパッチ適用が継続的にサポートされるサポート対象バージョンにするには、アップグレードが必要となる。

セクション 3 で説明するように、パッチ管理は、いくつかの課題によって複雑になる。これらの課題を解決できない組織は、システムに効果的かつ効率的にパッチを適用できず、容易に回避できる侵害が発生する可能性がある。パッチの処理に費やす時間を最小限に抑えることができる組織は、この時間をほかのセキュリティの問題の解決にあてることができる。すでに多くの組織がパッチ管理を大規模に運用しており、これをセキュリティの一環ではなく、主要な IT 機能にしている。しかし、どの組織にとっても、パッチ管理をセキュリティの観点から慎重に検討することが依然として重要である。なぜなら、強固なセキュリティを達成し、維持するためにはパッチ管理が非常に重要であるからである。

パッチ管理は、多様なセキュリティ遵守フレームワーク、指令、指針によっても求められる。たとえば、NIST Special Publication (SP) 800-53[2]は、SI-2、欠陥修正措置のセキュリティ管理策を求めている。ここには、セキュリティに関連するソフトウェアおよびファームウェアのパッチのインストール、インストール前のパッチのテスト、組織の構成管理プロセスへのパッチの組み込みが含まれる。別の例として PCI DSS（Payment Card Industry Data Security Standard）[3]では、最新のパッチのインストールを求め、最も重要なパッチのインストールに関する最大の期限を設定している。

[2] http://csrc.nist.gov/publications/PubsSPs.html#800-53-rev4
[3] https://www.pcisecuritystandards.org/security_standards/

3. パッチ管理の課題

このセクションでは、パッチ管理の実行に伴う課題を簡単に確認する。これらは、セクション 4 で検討するパッチ管理技術で解決しようとしている課題である。

3.1 タイミング、優先順位付け、テスト

タイミング、優先順位付け、テストは、エンタープライズ向けパッチ管理の相互に関連する問題である。組織が、すべての新しいパッチを直ちに導入して、関連するソフトウェアの欠陥に対してシステムが脆弱になる時間を最小限に抑えるのが理想的である。ただし、組織のリソースは限られているため、実際にはこれは不可能である。そのため、どのパッチをどのパッチの前にインストールすべきか、優先順位を付ける必要がある。この問題をさらに複雑にしているのが、最初にテストなしでパッチをインストールすることに付随する大きなリスクである。これによって運用の大規模な中断が発生し、パッチを適用しないときのセキュリティの影響よりも大きな損害につながる可能性がある。残念なことに、パッチのテストでは組織の限られたリソースがさらに消費される。このため、パッチの優先順位付けはいっそう重要になる。パッチ管理では、タイミング、優先順位付け、テストの間に競合が発生することが頻繁にある。

製品ベンダーは、パッチの品質を向上し、自社製品向けのパッチをバンドル化して、この競合に対応している。ベンダーは、数十個のパッチを 3 か月かけて一度にリリースし、数日ごとにテストとパッチの適用を求めるのではなく、四半期ごとに 1 つのバンドルとしてパッチをリリースすることができる。この結果、組織はテストの実施とパッチの適用を 1 回で済ませることができる。これは、すべてのパッチを個別にテストし、それぞれ適用するよりも断然効率的である。また、パッチに優先順位を付ける必要性も減る。つまり、組織はバンドルに含まれる各パッチの優先順位を個別に指定するのではなく、バンドルの優先順位を指定するだけである。パッチをバンドル化しているベンダーは、パッチが適用されていない脆弱性が積極的に悪用されているのでない限り、バンドルを毎月または四半期ごとにリリースする傾向がある。悪用されている場合は、通常、次のバンドルまで待たずに、適切なパッチを即座に発行する。

パッチのバンドル化には欠点もある。脆弱性が検出されてから、そのためのパッチを一般に入手できるようになるまでの時間が長引くことである。パッチのリリースを意図的に遅らせると、パッチのリリース前に同じ脆弱性を見つけた攻撃者にその脆弱性を悪用できる時間的猶予を与えてしまうこともある。しかし、これを軽減するための要因が 2 つある。1 つは、悪用されることがわかっている場合は、ベンダーがパッチを直ちにリリースするはずであること、もう 1 つは、バンドル化されていれば、個別にリリースされている場合よりもパッチをすばやくインストールできる可能性があることである。したがって、環境によっては、運用上、脆弱性が悪用される可能性がある期間をパッチのバンドル化によって効果的に短縮することができる。

タイミングについては、考慮すべき点がさらに数多くある。パッチのリリースによって、該当する脆弱性を悪用するために必要な情報を攻撃者に与えてしまう可能性がある（パッチからの脆弱性のリバースエンジニアリングなど）。これは、侵害を回避するためには新たにリリースされたパッチを直ちに適用する必要があるといことである。ただし、脆弱性が悪用されていない場合は、組織はパッチを適用しない場合のセキュリティリスクと、最初に徹底したテストをしないままパッチを適用する場合の運用上のリスクとを慎重に比較する必要がある。スナップショット機能を有効にした仮想ホストなどの一部の運用環境では、使用性や機能に関する問題が発生する場合、組織がパッチの適用を十分に準備しているのであれば、テストなしにパッチを適用することが好ましい場合もある。

タイミングに関するもう 1 つの基本的な問題は、パッチを有効にするために強制的な変更が必要になる場合がある点である。これにより、パッチを適用したアプリケーションやサービスの再起動、オペレーティングシステムの再起動[4]、またはホストの状態にそのほかの変化を加えるなどの処理が必要になる。最終的な問題は、いつパッチをインストールしたかではなく、いつパッチを実際に有効にするかである。場合によっては、少なくともパッチを完全に導入し、運用するまでは、別の方法で脆弱性を軽減する方が適切な場合がある。脆弱なアプリケーションの機能を一時的にブロックするために脆弱なソフトウェアの構成設定を変更する方法がその一例である。脆弱なホストのセキュリティ、機能、動作の意味が軽減オプションごとに異なるため、いくつかのオプションから 1 つのオプションを選択することは単純ではない。また、構成設定を変更する場合は、古い設定値を保存し、適切な時期に復元する必要がある。構成設定の変更に関わるもう 1 つの問題は、多くの場合、アプリケーションの再起動など、ホスト状態の変化が必要となる点である。構成の変更は、ホストの運用にとって、パッチのインストールと同様の混乱を生じる場合がある。

どのパッチをいつ適用するかという優先順位付けはタイミングと密接に関連しているが、ほかにも検討が必要な点がある。これは、脆弱なシステムの相対的な重要性（サーバとクライアントなど）と各脆弱性の相対的な重大度（共通脆弱性評価システム [CVSS] などの脆弱性重大度メトリクス）によって異なる。もう 1 つの検討事項は、各パッチ間の依存関係である。1 つのパッチをインストールするために別のパッチを先にインストールする必要がある場合や、パッチを順次有効にするために、アプリケーションやホストの再起動が何回も必要になる場合などである。

つまり、組織がエンタープライズ向けパッチ管理プロセスを計画し、実行する場合は、タイミング、優先順位付け、およびテストに関連する問題を慎重に検討する必要がある。

3.2 パッチ管理の構成

通常、パッチの適用には複数のメカニズムがあり、これも、エンタープライズ向けパッチ管理のもう 1 つの大きな課題である。その例を以下に挙げる。

- 自動更新ができるソフトウェアがある。
- 一元的な OS 管理ツールがパッチの適用を開始できる。
- サードパーティ製のパッチ管理アプリケーションがパッチの適用を開始できる。
- ネットワークアクセス管理、ヘルスチェック技術など類似の技術がパッチの適用を開始できる。
- ユーザがソフトウェアの自動更新を手作業で指示できる。
- ユーザが、パッチまたは新しいバージョンのソフトウェアを手作業でインストールできる。

このようにパッチの適用には複数の方法があるため、競合が発生することがある。複数の方法のそれぞれが同じソフトウェアにパッチを適用しようとすることがあるが、パッチの問題やテ

[4] これは、FDE（full disk encryption）ソフトウェアの使用など、起動前にホストが認証を求める場合に問題になることがある。起動前に認証が必要な FDE ソフトウェアやほかの技術を使用する組織は、これらの技術がパッチのインストールに与える影響を慎重に検討する必要がある。

ストの遅延などが原因で、組織が特定のパッチの適用を望まない場合に特に問題となる。また、それぞれのツールまたは管理者が、別のツールや管理者が特定のパッチをすでに処理したものと見なすことがあるため、複数の方法があると、パッチの適用が遅れたり、適用をし損ねたりすることもある。組織は、パッチを適用できるすべての方法を特定し、パッチの適用方法の間で競合が生じた場合は、それを解決しなければならない。

パッチ管理の構成に関連する問題として、ユーザがパッチ管理プロセスを無視したり、回避したりすることがある。設定の変更（直接的な更新、パッチ管理ソフトウェアの無効化など）、旧バージョンのソフトウェアのインストール、パッチのアンインストールなど、ユーザがそのホストのソフトウェアを変更できる場合は、パッチ管理プロセスの完全性が損なわれることがある。このような問題に対応するために、組織は、ユーザがエンタープライズ向けパッチ管理技術を無効にしたり、そのほかの方法でこの技術に悪影響を与えたりしないようにする必要がある。また、組織は、エンタープライズ向けパッチ管理技術を絶えず監視し、発生した問題を特定しなければならない。

3.3 代替ホストアーキテクチャ

エンタープライズ向けパッチ管理は、すべてのホストが完全に管理されており、一般的なアプリケーションやオペレーティングシステムを通常のプラットフォームで実行している場合は比較的簡単である。代替ホストアーキテクチャを採用している場合は、パッチ管理が非常に難しくなることがある。このようなアーキテクチャの例として以下のものがある。

- **管理されていないホスト。**セクション 3.2 で検討したように、ホストが一元管理されていない場合（ユーザが独自のホストを管理しているなど）は、パッチの適用を制御することが非常に難しくなることがある。

- **オフィスにないホスト（在宅勤務用ラップトップなど）。**ほかのネットワーク上のホストは、エンタープライズのネットワークセキュリティ管理策（ファイアウォール、ネットワーク侵入検出システム、脆弱性スキャナなど）によって保護されていない。

- **非標準 IT コンポーネント（アプライアンス機器など）。**そのようなホスト上では、多くの場合、個々のアプリケーションに個別にパッチを適用することはできない。組織は、更新されたソフトウェアをコンポーネントベンダーがリリースするまで待つ必要がある。この待ち時間は、主要なアプリケーションベンダーの場合よりも大幅に長くなることがあり、その結果、脆弱な時間帯が大幅に延びる。

- **モバイル機器。**スマートフォン、タブレットや、ほかのモバイル機器（ラップトップを除く）は、通常はモバイルオペレーティングシステムを実行しており、これらの機器へのパッチの適用は基本的に難しい。多くの場合、モバイル機器をデスクトップまたはラップトップに接続し、そのデスクトップまたはラップトップを介して更新を取得し、ダウンロードする必要がある。一部のモバイル機器は更新を直接ダウンロードできるが、帯域幅を考えるとこの方法には問題がある（サイズの大きい更新のダウンロードに時間がかかる、ダウンロードでデータ料金が課金されるなど）。モバイル機器を最新の状態に維持するためのもう 1 つのオプションとして、エンタープライズ向けモバイル機器管理ソフトウェアがある。エンタープライズ向けモバイル機器管理ソフトウェアは、組織によって管理されていない個人所有のモバイル機器も含めたモバイル機器の管理に使用されている。このソフトウェアは、アプリケーションのインストール、更新、削除ができ、スマートフォンのオペレーティングシステムやモバイル機器管理ソフトウェア

が最新のものでない場合に、エンタープライズのアクセスを制限できる。詳細については、SP 800-124 Revision 1、『エンタープライズにおけるモバイル機器の管理とセキュリティ保護のガイドライン（*Guidelines for Managing and Securing Mobile Devices in the Enterprise*）』のセクション 3 を参照のこと。

- **オペレーティングシステムの仮想化。**完全な仮想化のために使用するすべての OS イメージとスナップショットについてパッチを維持する必要がある。オフラインイメージにパッチを適用し、休止状態の仮想マシンを隔離する機能などのパッチ適用機能は、仮想化環境に組み込まれていることが多い。詳細については、NIST SP 800-125、『完全な仮想化技術のためのセキュリティガイド（*Guide to Security for Full Virtualization Technologies*）』を参照のこと。特に、セクション 3.3 では仮想マシンのイメージとスナップショットの管理について説明している。

- **ファームウェア。**一般的に、システム BIOS の更新などのファームウェアの更新には特殊な権限が必要で、ほかのタイプの更新とは手順が異なる。BIOS の更新の詳細については、NIST SP 800-147、『BIOSの保護ガイドライン（*BIOS Protection Guidelines*）』を参照のこと。

エンタープライズ向けパッチ管理ポリシーおよびソリューションを設計する場合、組織はエンタープライズで使用しているすべての代替ホストアーキテクチャを慎重に検討する必要がある。

3.4 そのほかの課題

このセクションでは、これまでにこのセクションで説明していないほかの課題について簡単に論じる。この文書で触れないほかの課題については、NIST SP 800- 40 Version 2、『パッチおよび脆弱性管理プログラムの策定』も参照のこと。[5]

3.4.1 ソフトウェアインベントリ管理

エンタープライズ向けパッチ管理は、パッチを適用できるソフトウェア（アプリケーションおよびオペレーティングシステム）の最新の完全なインベントリが各ホストにインストールされているかどうかに左右される。このインベントリには、その時点で各ホストにインストールされているソフトウェアだけではなく、インストールされている各ソフトウェアのバージョンも含まれていなければならない。この情報なしでは、正しいパッチの特定、取得、インストールはできない。このインベントリ情報は、インストールされているソフトウェアを最新のバージョンにするために旧バージョンを特定する場合にも必要になる。旧バージョンを更新することの主な利点は、パッチを適用し、そのパッチをテストする必要があるソフトウェアバージョンの数が減ることである。

3.4.2 リソースの過負荷

エンタープライズ向けパッチ管理によってリソースの過負荷が生じることがある。数多くのホストがサイズの大きい同じパッチ（またはパッチのバンドル）のダウンロードを同時に開始する場合などがその例である。この結果、ネットワーク帯域幅が過度に消費されたり、組織のパッチサーバからパッチを取得している場合は、そのサーバのリソースが大量に使用されたりすることがある。組織は、予想されるリクエストの量に対応するようにソリューションのサイズを設定し、エンタープライズ向けパッチ管理システムが、一度に多くのホストにパッチを送

[5] http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf

信しないようにパッチの供給を調整するなどして、そのエンタープライズ向けパッチ管理がリソースの過負荷の状況を確実に回避できるようにしなければならない。

3.4.3 インストールの副作用

パッチのインストールによって副作用が発生することがある。一般的な例として、インストールの結果、既存のセキュリティの構成設定が想定外に変更されたり、新しい設定が追加されたりすることがある。このため、パッチの適用による元の脆弱性の修復プロセスにおいて、新しいセキュリティの問題が発生することがある。組織は、パッチのインストールによって生じるセキュリティの構成設定の変更など、副作用を検出できなければならない。

3.4.4 パッチの実装の検証

セクション 3.1 で説明したように、インストールしたパッチは、関連するソフトウェアを再起動するか、ほかの状態の変化が生じるまでは有効にならない。ホストを確認し、特定のパッチが有効になったかどうかを確認することはきわめて難しい場合がある。これは、パッチが有効になったことを示す状況がない場合にさらに複雑になる（再起動が必要/不要など）。1 つのオプションとして、脆弱性の悪用を試みる方法がある。しかし、この方法は通常、悪用がすでに存在している場合にのみ実行可能である。また、高度に管理されている状況下においても悪用の試みは大きなリスクを伴う。組織は、パッチ管理システムからは独立した脆弱性スキャナなど、インストールを確認するためのほかの方法を使用する必要がある。

3.4.5 アプリケーションのホワイトリストへの登録

アプリケーションホワイトリスト登録技術がパッチ管理技術と競合することがある。これは、アプリケーションホワイトリスト登録技術が、パッチの適用により変更される可能性がある実行可能ファイルやほかのアプリケーションコンポーネントの既知の特性に基づいて機能するためである。ベンダーがホワイトリスト情報を提供している場合、そのベンダーがパッチを取得し、そのファイルの特性を記録し、対応する情報を顧客に送る必要がある。組織が独自のホワイトリスト情報を作成している場合、その組織は各パッチを取得し、そのファイルの特性を記録し、新しい情報でそのホワイトリストを更新しなければならない。いずれの方法でも、パッチをすばやく、特に自動的に適用している組織にとって問題となる遅延が発生する可能性や、パッチを適用したソフトウェアが不明なソフトウェアと見なされ、実行を禁止される可能性がある。

このような更新の問題を回避するために、ほとんどのアプリケーションのホワイトリスト登録技術はメンテナンスオプションを提供している。たとえば、管理者が特定のサービス（パッチ管理ソフトウェアなど）を信頼できる更新として選択できるものが多い。つまり、ホストに追加された、またはホスト上で変更されたすべてのファイルは、ホワイトリストに自動的に追加される。同様のオプションとして、ホワイトリストを更新できる信頼できる発行元（ソフトウェアベンダーなど）、ユーザ（システム管理者など）、ソース（信頼できるネットワークパスなど）、またそのほかの信頼できる組織を指定することができる。アプリケーションのホワイトリスト登録技術を使用する組織は、この技術が更新の問題を回避するように構成されていることを確認する必要がある。

4. エンタープライズ向けパッチ管理技術

このセクションでは、エンタープライズ向けパッチ管理技術の主要な概念を確認する。その構成について説明し、提供されるセキュリティおよび管理機能に注目し、使用方法に関する推奨事項を挙げる。

4.1 コンポーネントおよびアーキテクチャ

エンタープライズ向けパッチ管理技術のアーキテクチャについては、ほかのエンタープライズセキュリティソリューションと類似しており、管理とレポート生成を行う 1 つまたは複数の一元管理されたサーバと 1 つまたは複数のコンソールで構成されている。[6]個々のエンタープライズ向けパッチ管理技術のアーキテクチャの違いは、欠落しているパッチの特定に使用する手法にある。この手法とは、エージェント型、エージェントレススキャン、パッシブネットワーク監視の 3 つである。多くの製品はこれらの手法の 1 つだけをサポートしているが、複数をサポートするものもある。すべての手法について以下に詳細に説明する。組織は、エンタープライズ向けパッチ管理技術を選択するときに、各手法の利点と欠点を慎重に検討する必要がある。

4.1.1 エージェント型

エージェント型パッチ管理技術では、パッチ適用プロセスを管理し、エージェントとの調整を行う 1 つまたは複数のサーバを使用して、パッチを適用する個々のホストでエージェントを実行する必要がある[7]。各エージェントは、ホストにインストールされている脆弱なソフトウェアの確認、パッチ管理サーバとのやり取り、そのホストに使用できる新しいパッチの確認、これらのパッチのインストール、またそのパッチを有効にするために必要な状態変化（アプリケーションの再起動や OS の再起動など）を実行する責任を負う。各エージェントは管理者権限で実行されるため、これらの操作を実行できる。パッチ管理サーバは、パッチの入手元や必要な状態変化など、脆弱なソフトウェアと使用可能なパッチに関する情報をエージェントに提供する責任を負う。

エージェントレススキャンやネットワークのパッシブ監視と比較すると、エージェント型パッチ管理技術は、在宅勤務用のラップトップやスマートフォンなど、常にローカルネットワーク上にあるわけではないホストにとって非常に望ましい。

エージェント型パッチ管理技術には、いくつかの制限がある。多くのアプライアンス機器など、管理者によるオペレーティングシステムへの直接のアクセスが許可されないホストでは、一般的にエージェントを実行できない。また、組織のプラットフォームには、技術的な理由または運用上の理由によりエージェントを使用できないものがある（制御システム、医療機器、そのほかの専用のシステムなど）。

4.1.2 エージェントレススキャン

エージェントレススキャンパッチ管理技術には、パッチを適用する各ホストのネットワークスキャンを実行し、各ホストに必要なパッチを確認する 1 つまたは複数のサーバがある。一般的に、エージェントレススキャンでは、より正確なスキャン結果を返すことができるように、また、パッチをインストールし、ホストで状態を変更できるように（アプリケーション

[6] エンタープライズ向けパッチ管理技術は管理サービスとしても提供することができる。
[7] エージェント型パッチ管理技術は一部のオペレーティングシステムに組み込まれている。

の再起動、OS の再起動など）、サーバが各ホストについて管理権限を有している必要がある。

エージェントレススキャンの主な利点は、各ホストへのエージェントのインストールと実行が不要なことである。

エージェントレススキャンの主な制限の 1 つに、在宅勤務用のラップトップやモバイル機器など、ローカルネットワークにないホストは対象としない点がある。また、ネットワークセキュリティ管理策（ホストベースのファイアウォールなど）やネットワーク技術（ネットワークアドレス変換など）が、想定外にスキャンをブロックしたり、スキャンの結果に悪影響を与えることがある。エージェントレススキャンでも、帯域幅を過度に消費することにより運用に悪影響を与えることがある。最後に、エージェントレススキャンは組織の一部のプラットフォームをサポートできない場合がある。

4.1.3 ネットワークのパッシブ監視

パッチ管理用のネットワークのパッシブ監視技術では、ローカルネットワークトラフィックを監視し、パッチの適用が必要なアプリケーション（場合によってはオペレーティングシステム）を特定する。

これらの技術によって、ほかのパッチ管理ソリューション（エージェント型、エージェントレススキャン）では維持されないホストを効果的に特定できる。監視するホストに関する権限が不要のため、組織が管理していないホスト（管理されていないシステム、ビジターシステム、契約者システムなど）のパッチステータスの監視に使用できる。

ネットワークのパッシブ監視の主な欠点は、ネットワークトラフィックに基づいてバージョンを特定できるソフトウェア以外には使用できない点である（暗号化されていないものと仮定）。また、当然のことながら、ローカルネットワーク上のホストだけに使用できる。

4.1.4 手法の比較

表 4-1 にこの 3 つの手法の主な特長をまとめる。

表 4-1: 手法の比較

特長	エージェント型	エージェントレススキャン	ネットワークのパッシブ監視
ホストに管理権限が必要か?	はい	はい	いいえ
管理されていないホストをサポートするか?	いいえ	いいえ	はい
リモートホストをサポートするか?	はい	いいえ	いいえ
アプライアンス機器をサポートするか?	いいえ	いいえ	はい
スキャンに帯域幅が必要か?	最小限	中程度から過度	なし
検出されるアプリケーションの範囲は?	包括的	包括的	暗号化されていないネットワークトラフィックを生成するものに限定

4.2 セキュリティ機能

このセクションでは、パッチ管理技術によって提供される一般的なセキュリティ機能を、インベントリ管理、パッチ管理、そのほかの 3 つに分けて説明する。多くの製品では、これらの機能は、セキュリティ設定共通化手順（SCAP）を使用することで提供される。SCAP は、セキュリティに関連する情報を標準化された方法で整理し、表し、測定するために策定されたものである。SCAP とパッチ管理におけるその役割の詳細については、付録 A を参照のこと。

4.2.1 インベントリ管理機能

通常、パッチ管理技術には、各ホストにインストールされたソフトウェアとソフトウェアのバージョンを特定する機能、またはインストールされているソフトウェアの脆弱なバージョンだけを特定する機能がある。さらに一部の製品には、新しいバージョンのソフトウェアをインストールする機能、ソフトウェア機能をインストールまたはアンインストールする機能、またソフトウェアをアンインストールする機能がある。

4.2.2 パッチ管理機能

パッチ管理技術が幅広いパッチ管理機能を提供していることは明らかである。一般的な機能として、必要なパッチの特定、配布用のパッチのバンドル化および順序付け、管理者が導入する（または導入しない）パッチを選択する許可、また、パッチのインストールとインストールの検証などが挙げられる。多くのパッチ管理技術では、パッチの一元的な保存（組織内）や、外部ソースからの必要に応じたダウンロードが可能である。

4.2.3 そのほかの機能

パッチ管理機能を備えた多くのホストベース製品も、アンチウイルスソフトウェア、構成管理、脆弱性スキャンなど、ほかのセキュリティ機能を幅広く提供している。これらの機能について詳しく説明することは、この文書の範囲外である。

4.3 管理機能

パッチ管理技術を選択したら、その管理者はソリューションのアーキテクチャを設計し、テストを実行し、ソリューションを導入してセキュリティを確保し、その運用とセキュリティを維持する必要がある。このセクションでは、パッチ管理技術の管理に特に関わる問題（実装、運用、メンテナンス）を重点的に説明し、これらの操作を効果的かつ効率的に実行するための推奨事項を提供する。

4.3.1 技術のセキュリティ

組織内にエンタープライズ向けパッチ管理ツールを導入することにより、新たなセキュリティリスクが生まれる可能性がある。しかし、システムにパッチを効果的に適用していない組織は、それよりもはるかに大きなリスクに直面する。このようなツールは、特にセキュリティリスクや脅威からシステムを保護するためのセキュリティ対策機能がツールに組み込まれている場合、セキュリティの向上がセキュリティの低下を大きく上回るのが通常である。これらのツールを使った場合のリスクの例を次に示す。

- パッチが改変される（想定外に、または意図的に）
- 資格情報が悪用される
- ソリューションコンポーネント（エージェントを含む）の脆弱性が悪用される

- 他者がツールの通信を監視し、脆弱性を特定する（特にホストが外部ネットワーク上にある場合）

各組織は、エンタープライズレベルのアプリケーションの導入時に使用すべき標準的なセキュリティ手法によって、これらのリスクを削減する必要がある。このような対策の例として以下のものがある。

- パッチ適用ソリューションコンポーネントの厳重な保護（パッチの適用を含む）
- ネットワーク通信の暗号化
- インストール前のパッチの完全性の検証（チェックサムの使用など）
- 導入前のパッチのテスト（破損を特定するため）

4.3.2 段階的な導入

各組織は、段階的な方法でエンタープライズ向けパッチ管理ツールを導入すること。これにより、パッチ適用の全面的な導入に先立ち、小さなグループを使ってプロセスの問題やユーザとのコミュニケーションに関する問題を解決できる。ほとんどの組織では、パッチ管理ツールを導入する際に、まずは標準化されたデスクトップシステム、および同様の構成を持つサーバ群により構成される単一プラットフォームのサーバファームを対象とする。これが完了すると、次に組織は、マルチプラットフォーム環境、標準でないデスクトップシステム、レガシーコンピュータ、および例外的な構成のコンピュータの統合という、より難しい問題に取り組む必要がある。自動化されたパッチ適用ツールが適用できないオペレーティングシステムやアプリケーションのほか、例外的な構成のコンピュータに対しては、手作業による方法が必要となる場合がある。そのような例としては、組み込みシステム、産業用制御システム、医療機器、実験システムなどがある。このようなコンピュータについては、手作業のパッチ適用プロセスに関する明文化された実践済みの手順が必要である。

4.3.3 使用性および可用性

各組織は、セキュリティのニーズと、使用性および可用性に対するニーズとのバランスをとること。たとえば、パッチのインストールによってほかのアプリケーションが「中断」されることがあるが、これは導入前のパッチのテストによって適切に対応することができる。ほかの例として、アプリケーションの再起動、OS の再起動、ほかのホスト状態の変化によって混乱が生じ、データやサービスが失われる可能性がある。この場合も、各組織はパッチ適用のニーズと運用のサポートに関するニーズとのバランスをとること。最後の例として、モバイル機器にとって特に重要な、低帯域幅または従量制課金接続を経由した更新の取得を挙げることができる。そのような接続を介したサイズの大きいパッチのダウンロードは、技術的にも経済的にも不可能である。組織には、モバイルホストや、低帯域幅または従量制課金ネットワークで使用されているほかのホストでエンタープライズ向けパッチ適用ソリューションを確実に使用できるよう準備が必要である。

5. メトリクス

NIST SP 800-55 Revision 1、『情報セキュリティパフォーマンス測定ガイド（*Performance Measurement Guide for Information Security*）』のセクション 3.3 で説明されているように、測定指標には 3 つのタイプがある。

- 「実施測定指標は、セキュリティプログラム、特定のセキュリティ管理策、および対応するポリシーと手順の実施状況を示すのに用いられる…
- 効率/有効性測定指標は、プログラムレベルのプロセスと、システムレベルのセキュリティ管理策が、正しく導入され、意図したとおりに運用され、望ましい結果を満たしているか監視するために使用できる…
- 影響度測定指標は、情報セキュリティが組織のミッションにもたらす影響を、明確にするために用いられる…」

これらのタイプの測定に関しては、「成熟していないセキュリティプログラムを使って効果的な測定を行うためには、事前にプログラムの目標と目的を策定する必要がある。より成熟したプログラムは、実施測定指標を使用してパフォーマンスを測定し、最も成熟したプログラムは、効率/有効性測定指標およびビジネス影響度測定指標を使用して、情報セキュリティプロセスと手順の効果を特定する。」このため、組織は、そのエンタープライズ向けパッチ管理技術およびプロセスに適した測定指標を実装して使用する必要がある。

考えられる実施測定指標の例には以下のようなものがある。

- エンタープライズ向けパッチ管理技術の対象となる組織のデスクトップおよびラップトップの割合
- エンタープライズ向けパッチ管理技術によって自動的にインベントリが管理されるアプリケーションを持つ組織のサーバの割合

考えられる効率/有効性測定指標の例には以下のようなものがある。

- ホストで欠落している更新をチェックする頻度
- ホストアプリケーションの資産インベントリを更新する頻度
- ホストの X% にパッチを適用するときの最小/平均/最大時間
- パッチのリリース後 X 日以内にパッチが適用される組織のデスクトップおよびラップトップの割合Y 日以内の場合。Z 日以内の場合（ここで、X、Y、Z には 10、20、30 のように異なる値が入る）
- 特定の時間内に完全にパッチが適用されるホストの割合（平均）大きな影響を受けるホストの割合。中程度の影響を受けるもの。影響が小さいもの
- 完全に自動的に適用されるパッチの割合、一部が自動的に適用されるパッチの割合、また手作業で適用されるパッチの割合

考えられる影響度測定指標の例には以下のようなものがある。

- パッチ管理プロセスによって組織が達成したコスト削減
- 政府機関の情報システムの予算の中でパッチ管理に費やした割合

付録 A - セキュリティ設定共通化手順（SCAP）のチュートリアル

この付録では、エンタープライズ向けパッチ管理技術に関連するセキュリティ設定共通化手順（SCAP）の概要を説明する。この付録は、この文書作成時の最新版である NIST SP 800-117 Revision 1、『セキュリティ共通化手順（SCAP）バージョン 1.2 を採用および使用するためのガイド（*Guide to Adopting and Using the Security Content Automation Protocol (SCAP) Version 1.2*）』に基づくものである。SCAP の詳細については、NIST SP 800-117 の最新版を参照のこと。

NIST Special Publication (SP) 800-126 によると、SCAP（エスキャップと発音）は、「ソフトウェアの欠陥およびセキュリティ設定情報をマシンと人間の両方に伝えるための形式と命名法を標準化する一連の仕様」である。SCAP は、セキュリティに関連する情報に加えて、ソフトウェアの欠陥やセキュリティ構成問題の識別子など、関連する参照データを標準化された方法で整理し、表し、測定するために策定されたものである。SCAP は、パッチのインストールの自動検証、システムセキュリティの構成設定のチェック、システムでの侵害の兆候の確認など、エンタープライズシステムのセキュリティの維持に使用できる。

表 A-1 に、SCAP バージョン 1.2 プロトコルのコンポーネントの仕様を示す。コンポーネントは、次のようにタイプ別にグループ化される。

- **言語。**SCAP 言語は、セキュリティポリシー、技術チェックメカニズム、および評価結果を表現するための標準的な語彙や慣習を提供する。
- **レポート生成形式。**SCAP のレポート生成形式は、収集した情報を標準化された形式で表現するために必要な構造を提供する。
- **列挙。**SCAP の個々の列挙で、標準的な命名法（命名形式）と、その命名法を使用して表現した公式の辞書または項目一覧を定義する。
- **測定および評価システム。**SCAP では、これはセキュリティの弱点に関する特性を評価し（ソフトウェアの脆弱性やセキュリティ構成の問題など）、これらの特性に基づいて、その相対的な重大度を反映する評価を生成することを指している。
- **完全性の保護。**SCAP の完全性の保護の仕様は、SCAP の内容と結果の完全性を保護するために役立つ。

表 A-1.SCAP バージョン 1.2 のコンポーネントの仕様

SCAP コンポーネント	説明
言語	
XCCDF（Extensible Configuration Checklist Description Format: セキュリティ設定チェックリスト記述形式）1.2	セキュリティチェックリスト/ベンチマークを作成し、その評価の結果のレポートを生成するための言語
OVAL（Open Vulnerability and Assessment Language: セキュリティ検査言語）5.10	システム構成情報を表し、マシンの状態を評価し、評価の結果レポートを生成するための言語
OCIL（Open Checklist Interactive Language: 対話型チェックリスト記述言語）2.0	人や、ほかのデータ捕捉によって作成された既存のデータストアから情報を収集する評価内容を表す言語
レポート生成形式	
ARF（Asset Reporting Format）1.2	資産に関する情報の交換と、資産とレポートの関係を表すための形式

Asset Identification	既知の識別子や資産に関する既知の情報に基づいて資産を一意に識別するための形式

SCAP コンポーネント	説明
列挙	
CPE（Common Platform Enumeration: 共通プラットフォーム一覧）2.3	ハードウェア、オペレーティングシステム、アプリケーションの命名法および辞書、また、CPE 名の複雑な論理的グループ化を構成するための適用言語
CCE（Common Configuration Enumeration: 共通セキュリティ設定一覧）5	ソフトウェアセキュリティ設定の命名法および辞書
CVE（Common Vulnerabilities and Exposures: 共通脆弱性識別子）	セキュリティに関連するソフトウェアの欠陥の命名法および辞書
測定および評価システム	
CVSS（Common Vulnerability Scoring System: 共通脆弱性評価システム）2.0	ソフトウェアの欠陥による脆弱性の相対的な重大度を測定するためのシステム
CCSS（Common Configuration Scoring System）1.0	システムセキュリティ構成問題の相対的な重大度を測定するためのシステム
完全性の保護	
TMSAD（Trust Model for Security Automation Data）1.0	ほかのセキュリティ自動化仕様に適用される一般的なトラストモデルでデジタル署名を使用するための仕様

SCAP の各コンポーネントには一意の機能があり、独立して使用することができるが、これらのコンポーネントを組み合わせて使用すると、さらに大きいメリットが得られる。たとえば、CCE、CPE、および CVE 識別子を OVAL 定義と組み合わせて技術チェックの規則と関係を表す XCCDF ドキュメントと、OCIL の質問を使用して管理および運用チェックを表す XCCDF ドキュメントが *SCAP で表された*チェックリストの構築ブロックを構成する。[8] つまり、SCAP で表されたチェックリストでは、標準化された言語（XCCDF）を使用して、実行する必要があるチェック（OVAL、OCIL）、対象となるプラットフォーム（CPE）、また、対応する必要があるセキュリティ設定（CCE）とソフトウェアの欠陥による脆弱性（CVE）を表している。

オペレーティングシステムのセキュリティを確保するためのチェックリストなど、SCAP で表された包括的なチェックリストも、より専門的なチェックリストも、いずれも重要である。専門的なチェックリストを使用すると、システムの特性をチェックし、潜在的なセキュリティの問題を特定することができる。この一般的な例として、SCAP チェックリストを使用したパッチのインストールの確認と欠落しているパッチの特定がある。パッチのチェック用の SCAP 形式のデータは、ソフトウェアベンダーが自社製品のために公開する。組織はこのデータをダウンロードし、SCAP 対応ツールで使用することができる。[9]

[8] SCAP で表されたチェックリストは、NIST SP 800-70 Revision 1 の表 4-1 で詳しく定義されている。
[9] パッチ情報は .http://oval.mitre.org/repository/ の MITRE OVAL Repository からダウンロードできる。

付録 B - 推奨事項の要約

この付録に、この文書で紹介した主な推奨事項をまとめる。

セクション 3

セクション 3.1: 脆弱性がまだ悪用されていない場合、組織はパッチを適用しない場合のセキュリティリスクと、最初に徹底したテストをしないでパッチを適用する場合の運用上のリスクとを慎重に比較する必要がある。

セクション 3.1: 組織がエンタープライズ向けパッチ管理プロセスを計画し、実行する場合は、タイミング、優先順位付け、テストに関連する問題を慎重に検討する必要がある。

セクション 3.2: 組織は、パッチを適用できるすべての方法を特定し、パッチの適用方法の間で競合が生じた場合は、それを解決しなければならない。

セクション 3.2: 組織は、ユーザがエンタープライズ向けパッチ管理技術を無効にしたり、そのほかの方法でこの技術に悪影響を与えないようにする必要がある。また、組織はエンタープライズ向けパッチ管理技術を絶えず監視し、発生した問題を特定しなければならない。

セクション 3.3: エンタープライズ向けパッチ管理ポリシーおよびソリューションを設計する場合、組織はエンタープライズで使用しているすべての代替ホストアーキテクチャを慎重に検討する必要がある。

セクション 3.4.1: 各ホストにインストールされたパッチを適用可能なソフトウェア（アプリケーションおよびオペレーティングシステム）のインベントリには、その時点で各ホストにインストールされているソフトウェアだけではなく、インストールされている各ソフトウェアのバージョンも含まれていなければならない。

セクション 3.4.2: 組織は、そのエンタープライズ向けパッチ管理がリソースの過負荷の状況を回避できることを確認しなければならない。

セクション 3.4.3: 組織は、パッチのインストールによって生じるセキュリティの構成設定の変更など、副作用を検出できなければならない。

セクション 3.4.4: 組織は、パッチ管理システムからは独立した脆弱性スキャナなど、インストールを確認するためのほかの方法を使用する必要がある。

セクション 3.4.5: アプリケーションのホワイトリスト登録技術を使用する組織は、この技術が更新の問題を回避するように構成されていることを確認する必要がある。

セクション 4

セクション 4.1: 組織は、エンタープライズ向けパッチ管理技術を選択するときに、欠落したパッチを特定する各手法（エージェント型、エージェントレススキャン、パッシブネットワーク監視）の利点と欠点を慎重に検討する必要がある。

セクション 4.3: パッチ管理技術の管理者は、ソリューションのアーキテクチャを設計し、テストを実行し、ソリューションを導入してセキュリティを確保し、その運用とセキュリティを維持する必要がある。

セクション 4.3.1: 組織は、エンタープライズレベルのアプリケーションの導入時に使用すべき標準的なセキュリティ手法によって、エンタープライズ向けパッチ管理ツールを使用するリスクを削減する必要がある。

セクション 4.3.2: 組織は、段階的な方法でエンタープライズ向けパッチ管理ツールを導入する必要がある。

セクション 4.3.3: 組織は、セキュリティのニーズと、使用性および可用性に対するニーズとのバランスをとる必要がある。

セクション 5

セクション 5: 組織は、そのエンタープライズ向けパッチ管理技術およびプロセスに適した測定指標を実装して使用する必要がある。

付録 C - 頭字語および略語

このガイドで使用している主な頭字語および略語を以下に定義する。

ARF	Asset Reporting Format
CCE	Common Configuration Enumeration（共通セキュリティ設定一覧）
CCSS	Common Configuration Scoring System CPE Common Platform Enumeration（共通プラットフォーム一覧）
CVE	Common Vulnerabilities and Exposures（共通脆弱性識別子）
CVSS	Common Vulnerability Scoring System（共通脆弱性評価システム）
FISMA	Federal Information Security Management Act（連邦情報セキュリティマネジメント法）
IT	Information Technology（情報技術）
ITL	Information Technology Laboratory（情報技術ラボラトリ）
NIST	National Institute of Standards and Technology（米国国立標準技術研究所）
OCIL	Open Checklist Interactive Language（対話型チェックリスト記述言語）
OMB	Office of Management and Budget（行政管理予算局）
OVAL	Open Vulnerability and Assessment Language（セキュリティ検査言語）
SCAP	Security Content Automation Protocol（セキュリティ設定共通化手順）
SP	Special Publication
TMSAD	Trust Model for Security Automation Data
XCCDF	Extensible Configuration Checklist Description Format（セキュリティ設定チェックリスト記述形式）