グッド・プラクティス・ガイド
パッチ管理

2006 年 10 月 24 日 発行

邦訳：一般社団法人 JPCERT コーディネーションセンター

要約

本書は、重要国家インフラストラクチャに関わる組織向けにパッチ管理に関するガイドを示すものである。セキュリティ上の脆弱性を最小限に抑えるためにすべてのシステムにパッチを正しく適用するための 4 段階のプロセスについて説明するとともに、パッチ適用計画の有効性を測定する際に利用できる指標について説明する。付録では、一般的なオペレーティングシステム用のパッチ管理ツールについて詳しく説明する。

Disclaimer

Reference to any specific commercial product, process or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favouring by NISCC.The views and opinions of authors expressed within this document shall not be used for advertising or product endorsement purposes.

To the fullest extent permitted by law, NISCC accepts no liability for any loss or damage (whether direct, indirect or consequential and including, but not limited to, loss of profits or anticipated profits, loss of data, business or goodwill) incurred by any person and howsoever caused arising from or connected with any error or omission in this document or from any person acting, omitting to act or refraining from acting upon, or otherwise using, the information contained in this document or its references. You should make your own judgement as regards use of this document and seek independent professional advice on your particular circumstances.

National Infrastructure
Security Co-Ordination Centre
PO Box 832
London
SW1P 1BG

Tel: 0870 487 0748
Fax: 0870 487 0749
Email:enquiries@niscc.gov.uk
Web: www.niscc.gov.uk

本翻訳文書は、一般社団法人 JPCERT コーディネーションセンターが、原書の著作権を保有する英国 CPNI：Centre for Protection of National Infrastructure の許諾を得て翻訳したものです。

日本語版の内容について、原書に沿ってできるだけ忠実に翻訳するよう努めていますが、完全性、正確性を保証するものではありませんので、必要に応じて CPNI のホームページより原書 〝 Good Practice Guide Patch Management 〝 をご参照ください。

また、翻訳監修主体は本文書に記載されている情報により生じる損失または損害に対し、いかなる人物あるいは団体にも責任を負うものではありません。

なお、当文書に関わる最新情報は以下の CPNI のホームページをご参照ください。http://www.cpni.gov.uk/

【付記】

2007 年 2 月に国家情報インフラ安全調整局（NISCC：National Infrastructure Security Co-ordination Center）と国家セキュリティ顧問局（NSAC：National Security Advice Center）とが合併し、国家インフラストラクチャ保護局(CPNI：Center for the Protection of National Infrastructure)が発足しました。

日本語版では、原則として原書発刊当時の組織で表記し、必要に応じて訳注を記載しています。

目次

1　要点

- 本書はパッチ管理の手引きである。パッチ管理とは、暫定的なソフトウエアリリースであるパッチを管理し運用環境への展開を管制するプロセスであると定義される。

- 自組織のオペレーティングシステムやアプリケーションソフトウエアの信頼度を認識できていない場合、その組織にはセキュリティ上の脆弱性が多数存在する可能性があり、悪用された場合は事業に重大な影響が及ぶ可能性がある。このような悪用のリスクを緩和するには、IT システムを正しく構成し、最新のソフトウエアを使用し、推奨されるソフトウエアの更新をインストールする必要がある。

- 本ガイドでは、次の 4 段階のパッチ管理プロセスを提案している。

    - **査定と棚卸**　運用環境がどのようなソフトウエアコンポーネントで構成されているか、どのようなセキュリティ上の脅威および脆弱性が存在するか、組織が新しいソフトウエア更新プログラムに対応するように準備できているかを正確に記録することが、この段階の目的である。

    - **パッチの識別**　リリースされたパッチおよびソフトウエア更新プログラムを把握し、それが組織に適合するかを判断し、更新が通常の変更なのか緊急の変更なのかを判断することが、この段階の目的である。

    - **評価、計画、およびテスト**　当該パッチを運用環境に展開するかを決定し、いつどのように展開するかを計画し、実際の運用環境に類似した環境でソフトウエア更新プログラムをテストして、ビジネスの基幹システムおよびアプリケーションが侵害されないことを確認することが，この段階の目的である。

    - **展開**　システム利用者への影響を最小限に抑えながら、承認されたソフトウエア更新プログラムを運用環境に正しく展開することが、この段階の目的である。

- 本書では、組織のパッチ適用計画の有効性を評価するための指標も提案する。この指標は、以下の目的を含んでいる。

    - システムが攻撃を受ける可能性の高さを測定する。

    - 軽減対処時間を測定する。

    - パッチおよび脆弱性の管理コストを測定する。

2　パッチ管理の概要

パッチ管理は、運用環境への暫定的なソフトウエアリリースの展開と保守を管理するプロセスである。このプロセスは、事業上の実効性と効率を維持し、セキュリティの脆弱性を緩和し、組織の実運用環境の安定性を維持するのに役立つ。

自組織のオペレーティングシステムやアプリケーションソフトウエアの信頼度を認識できていないと、その組織にはセキュリティ上の脆弱性が多数存在する可能性があり、悪用された場合に事業に重大な影響が及ぶ可能性がある。このような悪用のリスクを緩和するには、IT システムを適切に設定し、最新のソフトウエアを使用し、推奨されるソフトウエアの更新をインストールする必要がある。

パッチ管理が十分でない、または存在しないことによる事業への影響全体を検討するため、次の項目について考えてみよう。

- **ダウンタイム**　組織におけるコンピュータのダウンタイムのコストはどの程度か。事業や国にとって重要なシステムが中断された場合はどうなるか。エンドユーザの生産性が損なわれた場合、重要システムのトランザクションが失われた場合、およびインシデントによって事業の損失が発生した場合の機会コストも考慮する必要がある。ハッキングによる攻撃により、ほとんどの場合ダウンタイムが生じる。攻撃そのものがダウンタイムの原因となる場合もあれば、回復処理がダウンタイムの原因になることもある。過去に、コンピュータに対する攻撃によって、数日間ダウンタイムが続いた事例も報告されている。

- **問題の修復に要する時間**　組織のさまざまな問題を修復するためのコストはどの程度か。コンピュータのソフトウエア環境を再構築するためのコストはどの程度か。多くのセキュリティ攻撃では、攻撃の際に作り込まれたすべてのバックドア（将来悪用される可能性がある）を確実に取り除くために完全な再インストールが必要となる。

- **疑わしいデータの完全性**　攻撃によってデータの完全性が損なわれた場合、最新の既知のバックアップからデータを復旧するコスト、または顧客やパートナーとともにデータの正しさを確認するためのコストはどの程度になるか。

- **信頼の喪失**　顧客の信頼を失ったことによるコストはどの程度か。単一または複数の顧客を失った場合、そのコストはどの程度になるか。

- **否定的な評判**　否定的な評判による組織への影響はどの程度か。ビジネスの相手として信用できない企業と判断された場合や、顧客の個人情報が漏れた場合、事業への影響はどの程度か。

- **法的防御**　攻撃を受けたあとに他者から訴訟を起こされた場合にかかるコストはどの程度になるか。他者に重要なサービスを提供している組織では、パッチ管理プロセス（またはその欠如）が訴訟の対象になる可能性がある。

- **知的財産の盗難**　組織の知的財産が盗まれるか破壊された場合のコストはどの程度になるか。

パッチ管理が十分でないことによる事業への影響に加え、国家的に重要と判断されている情報システムへの影響も考慮する必要がある。国家的に重要なシステムの障害や侵害は、組織の事業に影響を与えるだけでなく、国全体の社会的または経済的安定に悪影響を与える。国家的に重要なシステムに対する変更は正しく管理する必要があり、それらの変更を実施する前に正確な影響を知る必要がある。

十分に練り上げられたパッチ管理計画に基づいて、ネットワーク環境におけるソフトウエアの整合性を評価し維持することが、コンピュータへの物理的なアクセスの制限の種類にかかわらず、情報セキュリティを成功させるための重要な第一歩と言える。

3　推奨されるパッチ管理プロセス[1]

パッチ管理プロセスは、暫定的なソフトウエアリリースであるパッチの運用環境への展開方法と展開タイミングを、ホスト組織が管理できるものでなくてはならない。本書では、次のような 4 段階のプロセスを推奨している。

- **査定と調査**　運用環境がどのようなソフトウエアコンポーネントで構成されているか、どのようなセキュリティ上の脅威および脆弱性が存在するか、組織が新しいソフトウエア更新プログラムに対応するように準備できているかを正確に記録することが、この段階の目的である。

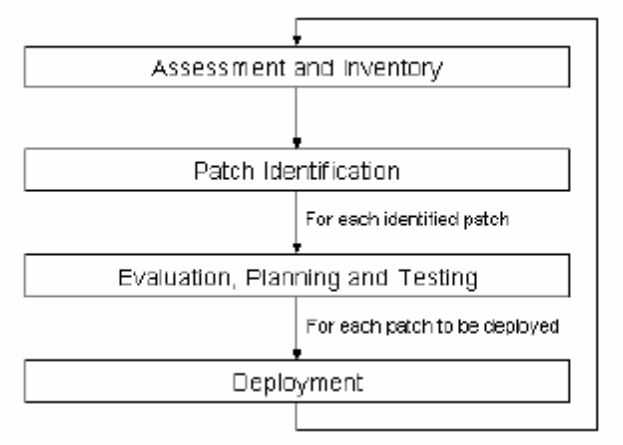
- **パッチの識別**　リリースされたパッチおよびソフトウエア更新プログラムを選別し、それが組織に適合するかを判断し、更新が通常の変更なのか緊急の変更なのかを判断することが、この段階の目的である。

- **評価、計画、およびテスト**　当該パッチを運用環境に展開するかを決定し、いつどのように展開するを計画し、実際の運用環境に類似した環境でソフトウエア更新プログラムをテストして、ビジネスの基幹システムおよびアプリケーションが侵害されないことを確認することが、この段階の目的である。

- **展開**　システム利用者への影響を最小限に抑えながら、承認されたソフトウエア更新プログラムを運用環境に正しく展開することが、この段階の目的である。

パッチ管理プロセスとその 4 つのフェーズを図 1 に示す。

[1] このプロセスの大部分は、Microsoft のパッチ管理ガイド[2]で説明されているプロセスに基づいている。詳細については同ガイドを参照のこと。



Assessment and Inventory	査定および調査
Patch Identification	パッチの識別
For each identified patch	識別された各パッチに対して
Evaluation, Planning and Testing	評価、計画、およびテスト
For each patch to be deployed	展開される各パッチに対して
Deployment	展開

図1 - 4段階から成るパッチ管理プロセス

重要なのは、パッチ管理のプロセスは大掛かりなプロセスである必要はないという点である。組織は独自の事業の要件に合わせたプロセスを採用するべきである。

以降の各セクションでは、上記の 4 つのフェーズについてそれぞれ詳しく説明する。

3.1 査定および調査フェーズ

査定および調査フェーズは、組織内にどのようなコンピューティング資産があり、それらをどのように保護できるかと、パッチ管理をサポートするソフトウエア配布アーキテクチャを組織がどのように開発できるかを判断するための継続的なプロセスである。

このフェーズは次のような手順で構成される。

- 既存の IT システムを調べる。
- ベースラインのシステム設定を適用する。
- 必要なセキュリティポリシーと技術標準を把握する。
- セキュリティプロセスを設定または現状のプロセスを見直す。
- 新しいソフトウエアの更新に関する最適な情報源を決定する。
- 既存のソフトウエア配布インフラストラクチャの査定を行う。
- 運用上の有効性を査定する。

3.1.1 既存のITシステムを棚卸する

パッチ管理を効果的に実施するには、運用環境に展開されているハードウェアとソフトウエアの最新情報を正確に把握する必要がある。この情報がなければ、環境内のどのコンピュータにソフトウエア更新プログラムが必要であるかを判断できない。また、クライアントコンピュータをネットワークに接続する方法を判断することも重要である。低速または信頼できないリンクでの接続や、ダイヤルアップ機能などのリモートアクセスを使用している場合、ソフトウエア更新プログラムを配布およびインストールする方法は、高速で信頼できるネットワークに接続されているコンピュータで使用する方法とは異なる。

効果的なパッチ管理を実行するためには、最低でも、運用環境に配置されているコンピュータから以下の情報を取得する必要がある。

- **ハードウェアの種類とバージョン**　コンピュータがポータブルコンピュータ（モバイルクライアント）であるか、デスクトップコンピュータであるか、またはサーバであるかを把握していると、特定のソフトウエア更新プログラムをどのようにインストールする必要があるかを管理者が判断するのに役立つ。たとえば、サーバにパッチを適用するときには、停止時間（変更およびコンピュータの再起動が許可される特定の時間帯）を順守するか、またはソフトウエア更新プログラムを展開する前にサーバのバックアップを取る必要が生じる場合がある。

- **オペレーティングシステムの種類とバージョン**　管理者は、運用環境に展開されているすべてのオペレーティングシステムの種類とバージョンを把握している必要がある。

- **アプリケーションとミドルウェア**　管理者は、オペレーティングシステムおよびサービスパックのバージョンと同様に、クライアントに展開されたソフトウエアアプリケーションとそのバージョンを把握している必要がある。

- **コンピュータの役割**　ソフトウエア更新プログラムの展開後に実行されるコンピュータの再起動の影響を評価するには、個々のコンピュータの機能を調べることが不可欠である。たとえば、コンピュータがビジネスの基幹アプリケーションを実行しているサーバである場合は、事業への影響が最小限に抑えられる時間帯にソフトウエアを更新するようにスケジュールするのが賢明である。また、たとえばサーバの再起動中もユーザがアプリケーションを継続して使用できるように、事業継続性を確保する必要が生じる場合もある。

- **ネットワークアーキテクチャおよび接続**　ネットワークインフラストラクチャのレイアウト、その機能、セキュリティレベル、リンクの速度、およびリンクの可用性について把握しておくことは、パッチを効果的に適用するために重要である。ソフトウエア更新プログラムはそれぞれサイズが異なるため、ネットワークインフラストラクチャの制限事項を知っていると、ソフトウエア更新プログラムの配布の遅れをできるだけなくすことができる。また、ソフトウエア更新プログラムが特定のクライアントコンピュータに展開される方法を決めることもできる。

- **インストール済みの更新プログラムとインストールされていない更新プログラム**　どのソフトウエア更新プログラムがコンピュータにインストールされていて、どのソフトウエア更新プログラムがインストールされていないかを識別することは不可欠である。

- **旧来のコンピュータシステムの凍結状態**　あるコンピュータシステム用のソフトウエアまたはハードウェアの更新プログラムが提供されなくなった場合には、そのことを記録する必要がある。そのシステムをパッチ管理体制の枠から外して、たとえば、より堅牢な構成や多層化された防御（多層防御）を施すなど、そのシステム専用のセキュリティ管理対策が必要になるからである。

棚卸を実行すると、組織の運用環境についての正確な記録をより容易に把握および取得し、それに基づいてパッチ管理の基準となるベースラインを設定することができる。

棚卸の実行後、管理者は結果が完全であるかどうかをチェックし、管理されているすべてのコンピュータが最新の情報を報告していることを確認する必要がある。次の作業を実行する必要がある。

- 棚卸結果を前の棚卸と比較し、検出されたコンピュータの増減を記録する必要がある。以前の棚卸と比較して数に大きな変化がある場合は、完全性に問題があることを示している可能性がある（その差異が、システムに新しく導入されたコンピュータの数から除去されたコンピュータの数を引いた数ということで説明がつく場合を除く）。

- アクティブに使用されているシステムは、DNS（Domain Name System：ドメインネームシステム）などの名前解決サービスを登録して使用する。このサービスのようなアクティブなインフラストラクチャに対する相互参照表を作るスクリプトにより、パッチ管理のインベントリがどの程度完全かを包括的に把握できる場合がある。より正確な結果を得るには、名前解決サーバ上で古い記録の除去またはクリーンアップが有効になっていることを確認する必要がある。

- AD（Microsoft Active Directory）を使用している IT システムの場合は、棚卸結果を AD ディレクトリサービスドメイン内のコンピュータオブジェクト（コンピュータアカウント）と比較することもできる。古いアカウントがクリーンアップされている限り、これは、名前解決サービスとの比較と同様に役立つ。

棚卸の分析によって、記録と実態とに矛盾があることが検出された場合は、さらに調査して、その原因を判別し、棚卸から漏れたシステムが次の棚卸の実行に含まれるようにする必要がある。

3.1.2　ベースラインとなるシステム設定を適用する

パッチ管理において、*ベースライン*とは、製品やシステムのある基準点における設定を文書化したものである。ベースラインは、同じクラスおよび同じカテゴリのシステムが合致しなければならない標準を定める。システムの構築と展開の基準となる信頼できるポイントとして、ベースラインを使用すれば、効果的な IT 運

用が可能である。通常、ベースラインによって定義された設定は、厳しくテストされ、ベンダによって認定される。

アプリケーションやソフトウエアのベースラインは、システムを目的の状態に再構築するために必要な情報を提供する。ソフトウエアアプリケーション、ハードウェアベンダ、またはコンピュータの種類ごとにベースラインを設定する必要がある場合もある。

ベースラインの設定には、IT 環境内のコンピュータとサービスの正確で最新のインベントリが必要である。環境のベースラインを設定する際には、次の点に留意する必要がある。

- ベースラインを満たしていないインフラストラクチャは、ベースラインを満たしていないコンピュータをすべて準拠レベルまで引き上げるために、問題管理によって解決をはかる必要がある。これらのコンピュータは、パッチの配布、スケジュール、または許可に問題があった可能性がある。また、例外的な取扱による特別な配慮が必要なコンピュータである場合もある。

- ベースラインを超えるインフラストラクチャが、必ずしも有利なわけではない。各クラスのベースライン（つまり、あるカテゴリまたは構成に属するすべてのシステムに対して定義されているベースライン）を超えるコンピュータをチェックし、無許可の変更が行われていないかどうかを判断する必要がある。場合によっては、システムを信頼済みのレベルに戻す必要があるかもしれず、変更の停止によってシステムを制御することが適切な場合もある。承認されているベースラインを超えるシステムには、相互接続性がテストされておらず、公式に承認されていないアプリケーションのバージョンまたはソフトウエア更新プログラムが含まれる場合がある。

- システムによっては、クラスのベースラインの対象外とされる特別な状況になっている場合がある。たとえば、モデムを使って外部委託先に接続する、更新が凍結された給与処理アプリケーションを実行している古いワークステーションでは、設定されたベースラインをかなり下回るオペレーティングシステムレベルが必要な場合がある。古いアプリケーションが実行できなくなる可能性があるため、このシステムを最新のベースラインにまでアップグレードすることは不適切な可能性がある。

- もっとも効果的なパッチは、適用する必要がないパッチである。つまり、ベースラインを設定するときには、ビジネス要件で必要とされていないサービスをすべて無効にすることを考慮すべきである。それらのサービスに脆弱性があったとしても、そのサービスが稼働していなければ悪用されることはない。

3.1.3 セキュリティポリシーと技術標準を識別する

パッチ管理は IT リスク管理の一部であるため、本書の内容を実践するにあたっては、IT システムのリスク管理と認定に関する英国政府のガイド[3]か同等のガイドも使用することを推奨する。

パッチ管理プロセスでは、どのポリシーフレームワークに準拠している場合も、組織で使用しているセキュリティポリシーおよび技術標準が矛盾することのないようにしなければならない。識別されたポリシーや技術標準には、次のような情報が含まれる場合がある。

- サポート対象のインストール場所とインストール方法を記述する、インストール標準。
- 名前や TCP/IP（Transmission Control Protocol/Internet Protocol）情報がどのように割り当てられるかと、コンピュータが所属するべきドメインを示す、ネットワークおよびドメイン標準。
- 開くポートを必要なサービスに基づいて削減するオプションなどの、オペレーティングシステムのセキュリティオプションとポリシー設定。
- 暗号化ファイルシステムの使用を記述する任意の標準。
- 各セキュリティリリースが提供される、サービスパックまたはソフトウエア更新プログラムの最低限の準拠基準。
- ウィルス対策プログラムの準拠基準。
- マクロファイル保護やセキュリティゾーンなどの、アプリケーションセキュリティ構成設定。
- アカウントの名前変更や無効化、デコイアカウントのセットアップなどのアカウント管理標準。
- パスワード管理標準。

効果的なセキュリティポリシーは、コンピュータの最低限のセキュリティ標準を認識し、潜在的なセキュリティ上の脆弱性の露出を最小限に抑えるために役立つものである必要がある。効果的にするには、運用環境において IT システムやソフトウエアが変更されたときに必ず組織のセキュリティポリシーを見直す必要がある。セキュリティポリシー違反は、環境から取り除く必要のある脆弱性を示唆するものである。脆弱性とは、さまざまなソフトウエア更新プログラムが欠けていたり、設定が誤っていたり、ユーザが強固なパスワードを使用していないなどの状態にコンピュータが陥っていることを示している。

セキュリティポリシーは、ポリシー準拠違反の種類ごとにガイドラインを提供する必要がある。そのガイドラインによって、特定の脆弱性の発生について深刻度を判断できる。

セキュリティポリシーを効果的に実施するには、強力な管理構造が必要である。効果的な管理を実現するには、次のことが必要である。

- セキュリティポリシー、実施期限、およびアプローチは、組織全体にまたがる経営層の支持を必要とする。
- 管理されていないコンピュータの所有権や管理情報を調べる手法やツールを、特定のサービスデスク技術者が利用できるようにする必要がある。

- サービスデスク技術者は、セキュリティポリシーに違反する各脆弱性を緩和することについてトレーニングを受ける必要がある。
- コンピュータシステムが企業のセキュリティポリシーおよび標準に常に準拠し続けるためには、可能であれば、自動ツールおよび手法を使用するべきである。

準拠していないシステムへの対応をどの程度積極的に行うかは、悪用のリスクや復旧のコストなどのセキュリティの脆弱性の性質に基づいて判断するべきである。要求された期限内に脆弱性を解決できなかった場合は、そのコンピュータをネットワークから物理的に切断する（または、ネットワークからデバイスを自動的に削除するようにネットワークハードウェアを構成する）ことが必要になる可能性がある。詳細については、『NISCC First Responders Guide: Policy and Principles』9 章[7]を参照のこと。

3.1.4 継続的なセキュリティプロセスの構築または見直しを行う

セキュリティの問題を継続的にスキャンしレポートすることは、潜在的なセキュリティの脆弱性を確実に識別して対処するためにきわめて重要である。少なくとも、組織は以下のことを実施する必要がある。

- 定期的にすべてのコンピュータをスキャンし、ウィルスの感染がないかチェックする。
- コンピュータシステムやインフラストラクチャデバイス内の脆弱性を識別するために、脆弱性スキャンツールを展開する。利用可能なスキャンツールの種類については、NISCC Technical Note 08/04、『Introduction to Vulnerability Assessment Tools』9 章[8]を参照のこと。
- ネットワーク監視ツール、イベントログ、およびその他の監視ツールから返された情報と、すべての侵入検知システムの出力を確認して、運用環境内のコンピュータシステムに対して攻撃が行われているかどうかを判断する。
- 危殆化したシステムを既知の適切に機能していた状態にすばやく復元するために、所定のハードウェアプラットフォームに適用できるオペレーティングシステムのバックアップイメージを作成して維持管理する。
- すべてのユーザおよび管理者が、運用環境でコンピュータシステムに攻撃を受けた場合に取る必要のある手順を認識していることを確認する。
- 攻撃を受けた場合に、最初に保護する必要のあるすべての主要な情報および資産の優先順位付き一覧を維持管理する。
- ルータログ、ファイアウォールログ、および構成を検証し、これらのデバイスに対する組織の標準に一貫性があることを確認する。
- ドメインコントローラのセキュリティポリシーを検討する。

システムに対する潜在的なセキュリティの脆弱性のスキャンはできるだけ自動化し、定期的に（ほとんどのシステムでは毎日）実行する必要がある。スキャンの頻度は、これらの領域のそれぞれに対して利用可能な自動化のレベル、組織の IT

セキュリティスタッフの稼動状況とスキル、およびセキュリティで保護された環境を目指す取組みの度合いによって異なる。

3.1.5 ソフトウエアの更新に関する情報の最適なソースを判断する

ソフトウエアの更新に関する情報源としては、次のようなものがある。

- CERT（Computer Emergency Response Team：コンピュータ緊急対応チーム）
- 独立の Web サイト
- ベンダの Web サイト
- 電子メールによる通知

確立した運用ベースラインを維持および更新し、効率的なパッチ管理プロセスを実装するためには、適切な通知を購読することが不可欠である。

各国のCERTは、そのサービス提供を受けているコミュニティが脆弱性とソフトウエアの更新に関する情報を見つけられるようにしている。Unirasは、イギリス政府およびイギリスの国家的重要インフラストラクチャを対象とするCERTである[2]。

NISCCのWebサイト（http://www.niscc.gov.uk/niscc/index-en.html）[3]を始めとする独立のWebサイトは、新しく発見された脆弱性とそれに対応するパッチに関する警告をできる限り迅速に公開している。国家的に重要なインフラストラクチャ組織は、NISCC Unirasの警告と状況説明の購読を申し込むことができる。それには、正確な購読用の電子メールアドレスを指定して、uniras@niscc.gov.uk[4]宛てに電子メールを送信する。

また、主要なソフトウエアベンダやサプライヤはすべて独自のセキュリティ Web サイトを設けており、そこで脆弱性とパッチの詳細を公開している。ベンダによっては、購読者がセキュリティ上の更新を直接受け取れる電子メールによるアドバイスサービスを提供している。

セキュリティ上の更新（緊急の更新を除く）は定期的にリリースされる傾向がある点に注意する必要がある。たとえば Microsoft は、毎月の第 2 火曜日の午前 10 時から 11 時の間に（グリニッジ標準時から 8 時間遅れ）、新しい更新の詳細を会報の形で公表している（訳注; 日本では時差の関係上、毎月第 2 火曜日の翌日 (第 2 水曜または第 3 水曜) に公開）。

3.1.6 既存のソフトウエア配布インフラストラクチャの査定を行う

ソフトウエアの配布インフラストラクチャの査定は、効果的なパッチ管理プロセスのもう 1 つの重要な部分である。この査定は次のような質問に対応して行われる必要がある。

---

2 （訳注）5.2 Uniras を参照。
3 （訳注）代わりに CPNI の URL を示す。http://www.cpni.gov.uk/
4 （訳注）現在は、このアドレスは無効である。

- ソフトウエア配布用のインフラストラクチャは設定されているか。
- それはソフトウエアの更新の配布にも使用できるか。
- 環境内のすべてのコンピュータがそのサービスを受けられるか。
- それはビジネス上重要なコンピュータのパッチ適用を処理するように設計されているか。

3.1.7 運用の効果を査定する

効果的なパッチ管理を実現するためにもっとも重要なのは、おそらく効果的な IT 運用プロセスである。運用の効果を査定するときには、次の質問を確認する必要がある。

- パッチ管理を行なうスキルを持つ人が十分いるか。
- パッチ管理が必要であることを人々は認識しているか。
- 責任者はセキュリティ設定、一般的なコンピュータの脆弱性、ソフトウエア配信技法、リモート管理、およびパッチ管理プロセスについて理解しているか。
- 標準の運用プロセスがあるか。それとも、日々の運用が主に暗黙の了解であいまいな状態になっているか。
- 非公式なものでも、変更管理やリリース管理に関するプロセスは存在するか。

組織は、その IT 環境の現在の管理モデルを再検討し、そのモデルがパッチ管理をどの程度適切にサポートしているかを判断する必要もある。次の問題を考慮する必要がある。

- スタッフ数と比較して、インフラストラクチャはどのくらいの規模か。管理者対システムの割合はどのくらいか。
- 現在のサポートモデルは、集中、分散、または共有のどれか。
- サポートスタッフの運用時間はどのくらいか。保守と管理に割り当てる時間は十分あるか。
- チームメンバの役割が明確に定義されてチームメンバに伝えられているか。
- 正式なサービス管理プロセスが確立されて準拠されているか。
- 個人がその役割を効果的に実行するのに十分なツールはあるか。

最後に、効果的なパッチ管理の実施に必要な適切なスキルセットを決定するには、次の質問が役に立つことがある。

- 担当者は、インフラストラクチャの規模および複雑さに対処できる十分な経験があるか。
- スタッフは正しくトレーニングを受け、関連するテクノロジの知識を持っているか。
- 担当者はまとまりのあるチームとして共同作業を行っているか。

- 担当者は、重要な運用規律と実施方法を理解するために必要な経験があるか、その訓練を受けているか。
- 担当者はスクリプティングのスキルがあるか。
- 担当者は、ユーザおよびシステムの権限とコンテキストを理解しているか。
- 担当者はソフトウエア更新プログラムの構造および依存関係を理解しているか。

3.1.8 要約

パッチ管理プロセスの査定および調査フェーズで留意する必要がある重要な点は、次のとおりである。

- 組織にどのような IT 資産があり、そのうち何がビジネス上重要であるかを理解する必要がある。
- 運用環境に何が展開されているか、何が管理対象として分類されるか（管理ツールによって）、および何が管理対象として分類されないかを理解する必要がある。
- ソフトウエア配布ツールが構成、保守されており、通常および緊急時のパッチ管理をサポートできることを確認する必要がある。
- スタッフが役割と責任を割り当てられ、緊急時に対応する方法（ソフトウエア更新の対処方法やその影響を軽減する方法）を知っていることを確認する必要がある。

3.2 パッチの識別フェーズ

パッチ管理プロセスのこのフェーズの目標は、次のとおりである。

- 信頼できる方法で新しいソフトウエア更新プログラムを探索する。
- ソフトウエア更新プログラムが運用環境に関連があるかどうかを判断する。
- ソフトウエア更新プログラムのファイルを取得して確認する。
- ソフトウエア更新プログラムが通常の変更と緊急の変更のどちらであるかを判断し、展開するための手順などに関する変更の要求書（RFC：Request For Change）を提出する。

3.2.1 新しいソフトウエア更新プログラムを探索する

ソフトウエア更新プログラムの識別は、信頼できる安全な方法で更新プログラムを探索することから始まる。探索には、次の 2 つの主要な構成要素がある。

- 新しいソフトウエア更新プログラムの通知がどのように受信されるか。
- 通知が本物であることの保証はどのように得られるか。

通知は、スキャンおよびレポート機能を提供する信頼できるソースの購読を通じて、またはその他の信頼できる通知メカニズムによって提供されるべきである。

もっとも一般的に使用される通知メカニズムは、脆弱性スキャンツールからの電子メール通知とレポートである。

電子メール通知は慎重に扱うことが重要である。

- ベンダは通常、セキュリティ上の理由で電子メール通知にソフトウエアファイルを添付しない。代わりに、電子メールでベンダのセキュリティ Web サイトをユーザに知らせることが多い。ソフトウエアベンダからと称する電子メール通知に添付された実行可能ファイルは、実行もインストールもしないのが安全である。

- 電子メール通知内のリンクは直接クリックしてはならない。元の URL がベンダの Web サイトのアドレスを示しているのに悪意のある Web サイトにリンクしていた場合は、その URL をブラウザウィンドウにタイプ入力するか、テキストのみのエディタにコピーしてからそのテキストをブラウザウィンドウに貼り付けること。

- 多くのベンダは、セキュリティ上の更新に関連した電子メール通知を顧客に送信するときに、それらの通知にデジタル署名をする。このような場合は、そのデジタル署名が有効であることを、通知内の指示に従って必ず確認すること。

3.2.2 ソフトウエア更新プログラムの適合性を判断する

ソフトウエア更新プログラムは、多くの理由でさまざまなソースから、ますます頻繁にリリースされるようになっている（そのすべてがセキュリティに関連しているわけではない）。このセクションで概説する通知の審査プロセスによって、無関係なソフトウエア更新プログラムの大半は除外できる。

配信されたソフトウエア更新プログラムは、関連のあるものであるかを 1 つずつ確認する必要がある。通知に複数のソフトウエア更新プログラムに関する情報が含まれている場合は、各ソフトウエア更新プログラムについて、組織に関連のあるものであるかをそれぞれ確認する必要がある。

関連があるかを確認するには、まず、ソフトウエア更新プログラムが運用環境内のオペレーティングシステムまたはアプリケーション用に設計されたものであるかを判断する。そうである場合は、その更新プログラムが適用されるアプリケーションまたはシステムに、ソフトウエア更新プログラムで対処するべき脆弱性があるかを判断する。

環境内のシステムなどに適用されるセキュリティ更新プログラムは、すべてが当該環境に適合するとは限らない。既存のセキュリティ更新プログラムを認識して正しく理解することは重要であるが、展開するのは組織の環境に関連があるセキュリティ更新プログラムだけにしなければならない。それにより、環境を最新で安全な状態に保つのに必要なコストと労力を最小限に抑えることができる。

ソフトウエア更新プログラムの情報が無関係なものと分類された場合、その情報を問題管理担当者に渡し、必要に応じてそのソフトウエア更新プログラムのコピーを配布用に保存することによって、そのソフトウエア更新プログラムが存在す

ることを記録しておくことも重要である。将来、このソフトウエア更新プログラムが組織の環境に関連のあるものになり、必要になった場合は、最初の発行元からこの情報を取得できる。

ベンダは通常、脆弱性が悪用された場合に攻撃者がどのようなアクセスまたは権限レベルを得るか（たとえば、その脆弱性によってプログラムのリモート実行が可能になるか）に基づいて、セキュリティ上の更新をランク付けする。これらのランクによって、必要な措置の緊急性を判断できる。

特定の技術や製品に特化したソフトウエア更新プログラムの関連性を判断することは、たいていの環境で簡単な作業ではない。たとえば、Microsoft の IIS（Internet Information Services）などの、クライアントにもサーバにもインストールできるソフトウエアの場合は、自組織内のコンピュータの特定のサブセットに対して、どのソフトウエア更新プログラムが関連するかどうかを判断するのが困難だ。こうした問題に対処するためにも、環境のインベントリをできる限り正確に維持することが重要である。

通知内の各ソフトウエア更新プログラムは、詳細かつ綿密に検討する必要がある。この検討では、ソフトウエア更新プログラムとともに送付されたものも含めて関連するすべてのドキュメントと、ベンダのセキュリティ Web サイトなどにある入手可能なサポート情報を参照する必要がある。

適用可能なソフトウエア更新プログラムを特定する電子メールメッセージを受信したら、調査担当者を決める必要がある。指名されたチームメンバがこのソフトウエア更新プログラムを担当することになる。

ソフトウエア更新プログラムは、特定のシナリオまたは構成に固有の場合がある。検討の担当者は、運用環境に展開されているシナリオや構成が、通知や Web に掲載されている文書の内容と一致するかどうかを確認する必要がある。

また、ソフトウエア更新プログラムの依存関係について、次の点を明らかにする必要がある。

- 更新プログラムに関連する依存関係はあるか。たとえば、更新プログラムを有効にするために、特定の機能を有効または無効にする必要があるか。
- ソフトウエア更新プログラムを適用するには、特定のサービスパックがインストールされている必要があるか。ソフトウエア更新プログラムは、サービスパックまたは別のソフトウエア更新プログラムによって置き換えられるか。また、新しいバージョンが提供されるまで待つのは妥当か。

上記の依存関係を識別することは、ソフトウエア更新プログラムのリリースと展開の計画に直接影響するため非常に重要である。そのソフトウエア更新プログラムがどのサービスパックに含まれるかと、アクティブなサービスパックに応じて異なるバージョンのソフトウエア更新プログラムが必要かどうかを文書化する必要がある。ユーザがあるサービスパックから別のサービスパックにアップグレードした結果として整合性の問題が発生する場合があるため、この点を把握することが重要である。

3.2.3 ソフトウエア更新プログラムのファイルを取得して確認する

ソフトウエア更新プログラムを識別して、その関連性を確認したら、ソフトウエア更新プログラムのファイルを取得して、それらが安全で、正しくインストールされることを確認する必要がある。確認プロセスでは、セキュリティ更新プログラムを真性とするか、セキュリティ上問題のあった更新プログラムにマークをする。後者の場合には、不正な通知が届いたときに、より詳しい調査のために、購読プロセスの担当者やセキュリティチームに、その通知に関する情報を、送信しておく必要がある。たとえば、通常信頼している発行元から配信されていたのに、エラーであることを示す通知がだされた場合、この発行元からの通知の品質についてセキュリティ上の懸念が生じる可能性がある。その発行元を調査し、問題がある場合は解決する必要がある。

ソフトウエア更新プログラムの確認では、少なくとも次の手順を実行する必要がある。

- ソフトウエア更新プログラムを識別する。
- 更新プログラムのデジタル署名を確認する（もしあれば）。
- 付属のドキュメントをすべて確認する。
- ソフトウエア更新プログラムのファイルを確認する。
- ソフトウエア更新プログラムのサイズを識別する。
- ソフトウエア更新プログラムの依存関係を識別する。
- 更新プログラムの適用前または適用後に必要なアクションがあればそれを識別する。
- ソフトウエア更新プログラムのインストール手順書が存在することを確認する。
- ソフトウエア更新プログラムのアンインストール手順書が存在することを確認する。
- ソフトウエア更新プログラムを隔離されたテスト用ネットワークに実装することにより、それが安全であることを確認する（以下を参照）。

ウィルス感染や悪意のあるコードによるITインフラストラクチャへの影響を防ぐため、ソフトウエア更新プログラムに関連するファイルはすべて、隔離された（検疫用の）環境で調べる必要がある。この検疫は、すべてのソフトウエアおよびドキュメントに対して行う必要がある。検疫用の環境では厳密な管理を行う必要があり、これを保証するために、組織内の専門家のグループが検疫プロセスを実行する必要がある。

関連ドキュメントは、複数の人が読んで確認するのが理想的である。それにより、1 人だけで更新プログラムを評価すると重要な点を見落とす可能性があるというリスクを緩和できる。ドキュメントを読むときには、次の疑問に対する答を探すこと。

- 更新プログラムを適用するとほかの問題が起こって運用システムが侵害される可能性はあるか。
- 更新プログラムを展開する前になんらかのアクションを実行する必要があるか。
- 更新プログラムを展開したあとでなんらかのアクションを実行する必要があるか。
- 環境に更新プログラムを適用中に使用できる回避策や軽減手順はあるか。
- ソフトウエア更新プログラムのインストール手順書はあるか。
- ソフトウエア更新プログラムのアンインストール手順書はあるか。
- ソフトウエア更新プログラムのファイルのサイズはどのくらいか。ファイルサイズは、リリースのプロセスおよび計画の全体（たとえば、自宅や移動中に作業をするユーザへの対応方法など）に影響する。

まれに、ソフトウエアの更新が、レジストリまたは構成ファイルの変更や、アプリケーションの設定の調整だけで済む場合があるが、ほとんどのソフトウエア更新ではファイルのダウンロードが必要となる。

ソフトウエア更新プログラムのインストール手順書を確認するためのガイドラインを次に示す。

- ソフトウエアの更新に再起動が必要かどうか確認する。再起動が必要な場合は、基幹業務サーバまたはコアインフラストラクチャサーバの計画フェーズと展開フェーズで、特別な配慮が必要になる。
- アンインストールフォルダも含めて、ソフトウエアの更新に必要なディスク容量を評価する。
- 更新プログラムのインストール時に使用できる構成オプションが提供されるかを確認する。
- 補足文書を読み、ソフトウエア更新プログラムのインストールに関するその他の情報がないか確認する。

テストを行っても、ソフトウエア更新プログラムのインストール後に、アンインストールを必要とする問題が発生する場合がある。したがって、アンインストール手順が正しく実行されるかをテストすることが重要である。アンインストール後に、サーバが問題なく動作することを確認するとともに、システム監視ツールのログを調べる必要がある。

3.2.4 ソフトウエア更新プログラムの性質を判断してRFCを提出する

ソフトウエア更新プログラムについて、識別、自組織に適合するかの判定、ソフトウエア更新ファイルの取得、安全性、および正しくインストールされたかの確認が終わったら、RFC（Request For Change：変更要求書）を提出してソフトウエア更新の評価および計画フェーズを開始する。

提出する変更要求書では、次の点を明確にする必要がある。

- 変更点は何か。
- どのような脆弱性に対応する変更なのか。
- 変更によってどのサービスが影響を受けるか。
- そのサービスに対してすでにソフトウエア更新プログラムが展開されているか。
- そのソフトウエア更新プログラムでは、インストールを完了させるために再起動が必要か。
- ソフトウエア更新プログラムをアンインストールできるか。
- さらにソフトウエア更新プログラムのテストとパッチ展開までにどのような対応策（もしあれば）を実施できるか。
- この変更に対して推奨されるテスト戦略は何か。
- その RFC についてどのような優先度を提案するか。
- この変更の影響（カテゴリ）は何か。

ソフトウエア更新プログラムによって、非常に重要なセキュリティの問題やシステムの不安定性に対処する場合は、RFC の優先度を「緊急」とマークする必要がある。緊急 RFC を作成するのは、ソフトウエア更新プログラムの展開やセキュリティ対応策（ネットワークポートを閉じるなど）の実装を緊急に実行する必要がある場合だけにするべきである。

3.2.5 要約

パッチの識別フェーズで留意する必要がある重要な点は、次のとおりである。

- 新しいソフトウエア更新プログラムの通知を確実に受けられるようにする必要がある。
- ソフトウエア更新プログラムの通知が認証済みの配布元から送信されていることを確認する必要がある。
- ソフトウエア更新プログラムが運用環境内のシステムに関連していることを確認する必要がある。
- ソフトウエア更新プログラムのファイルを取得し、ウィルスやほかのマルウェアに感染していないことを確認する必要がある。
- ソフトウエア更新プログラムのインストールが正しく実行されることを確認する必要がある。
- ソフトウエア更新プログラムが緊急のものかどうかを判断し、運用環境に展開するために RFC を提出する必要がある。

3.3 評価、計画、およびテストフェーズ

この第 3 段階では、ソフトウエア更新プログラムを評価し、その結果展開が承認された場合は運用環境への展開を計画してテストする。このフェーズの終わりま

でに、更新プログラムを展開するための変更要求が確認され、承認または拒否されることになる。承認された場合は、緊急かどうかの分類をする。

このフェーズで行う必要がある主な作業は次のとおりである。

- 適切な対応策を決定する。
- ソフトウエア更新プログラムのリリースを計画する。
- リリースを作成する。
- リリースをテストする。

3.3.1 適切な対応策を決定する

ソフトウエア更新の RFC には運用環境に必要な変更が記述されており、ほかの人々がそれに基づいて作業を実施できる。評価、計画、およびテストフェーズの最初の手順では、RFC を確認して、ソフトウエアの脆弱性または脅威に対してもっとも適切な対応策を決定する。これには、要求の優先度およびカテゴリの決定と、ソフトウエア更新プログラムを展開するための承認の取得が含まれる。

3.3.1.1 RFCの優先度を決定する

優先度とカテゴリは、変更を開始した人によって最初に割り当てられて RFC に含められるが、変更要求を承認する前にそれらの割り当てを自組織に適合するように見直して、合意または変更する必要がある。ソフトウエアの更新が変更プロセスを通過する速さは優先度によって決まるため、優先度は特に重要である。次の考慮事項は、ソフトウエア更新プログラムの優先度を判断するときに役立つ。

- 重要な事業資産（または国家的資産）は何か。それらの資産は、ソフトウエア更新プログラムがインストールされるまで、セキュリティ違反やシステムの不安定性などのリスクにさらされる可能性があるか。高価値資産に更新プログラムを適用する場合としない場合の影響に基づいて、変更要求の優先度を決定する必要がある。
- ソフトウエア更新プログラムは、過去に攻撃者の対象となった、ビジネスまたは国家にとって重要なサービスを実行しているシステムに適用されるか。これは、変更要求の優先度を上げるのに十分な理由になり得る。
- 特定のセキュリティ脆弱性の脅威を軽減する対応策がすでに展開されているか。展開済みの場合は変更要求の優先度が下がる可能性があるが、それでも脆弱性をなくすソフトウエア更新プログラムを展開することが適切な場合がある。
- 運用環境にとって問題となっている脆弱性の脅威とはどのようなものか。セキュリティ情報および関連するソフトウエア更新プログラムの多くは、環境内の数台のコンピュータにのみ適用される場合がある。脆弱性の脅威のレベルが低い場合は、この理由で要求の優先度が下がる可能性がある。
- 脆弱性が悪用された場合、攻撃者はどのようなアクセスまたは権限を得るか。たとえば、その脆弱性によって、プログラムのリモート実行、ローカル権限の

エスカレーション、またはサービス拒否が可能になるか。ビジネス上または国家的な重要性の要件という観点から、技術的影響を考慮する必要がある。

優先度の尺度は、当該の組織の問題である。ただし、次の表 1では 4 段階の優先度を、それぞれの優先度で RFC を実装する場合の推奨期限とともに提案している。

表 1　更新の優先度と推奨展開期限

分類	優先度	推奨期限
緊急	緊急	24時間以内
非緊急	高	1週間以内
	中	可用性に応じて、この脆弱性の解決を含む、新しいサービスパックまたは更新のロールアップを、1～2か月以内に展開する。
	低	可用性に応じて、この脆弱性の解決を含む、新しいサービスパックまたは更新のロールアップを、6か月以内に展開する。

高価値の資産または露出度の高い資産が脆弱性の影響を受ける場合や、影響を受ける資産が以前から攻撃者のターゲットになっている場合には、対応の優先度を本来の計算値より上げることもあり得る。逆に、侵害のリスクを最小限にする対応策など、多くの軽減要因が展開されている場合や、ビジネスへの影響が小さい資産しか脆弱性の影響を受けない場合には、対応の優先度を本来の計算値より下げることもあり得る。

セキュリティ更新プログラムの対象となる脆弱性がすでに悪用されているかまたは悪用されそうになっている場合、または運用環境で生じているシステムの不安定性が更新プログラムによって修正される場合は、状況に応じて要求を「緊急」と分類する必要がある。この分類により、その変更の優先度が、運用環境内で実施されるほかのどの変更よりも高くなる。

3.3.1.2　RFC を分類する

RFC の優先度はその緊急性を決定するが、RFC のカテゴリは当該 RFC を実装するために必要な作業量と、展開作業中の組織の IT システムへの潜在的影響を決定する。RFC のカテゴリは、変更確認の担当者が運用環境内のシステムとサービスへの影響を把握するのに役立つため、RFC のカテゴリを正しく決定することは重要である。変更要求のカテゴリを確立するには、次の点を判断する必要がある。

- ソフトウエア更新プログラムをどのコンピュータにインストールする必要があるかと、それらのコンピュータの役割（ビジネスに対する重要度）。たとえば、ビジネスに不可欠なコンピュータの再起動が必要なソフトウエア更新プログラムは、再起動を必要としないものより影響が大きい。

- ソフトウエア更新プログラムを展開するために追加の変更が必要かどうか。たとえば、ソフトウエア更新プログラムが現在のサービスパックだけに適用される場合、そのサービスパックがインストールされていない運用システムがある

と、それらのシステムを特定のセキュリティ脆弱性から保護できない可能性がある。この場合は、サービスパックとソフトウエア更新プログラムの両方を展開する必要があるため影響が大きくなり、それに応じて変更要求のカテゴリも上がる。

- いったんインストールしたソフトウエア更新プログラムをアンインストールできるかどうか。アンインストールできない場合は、問題なくアンインストールできるソフトウエア更新プログラムと比べて、運用環境に対するリスクが大きくなる。特定のセキュリティ脆弱性から保護したり、特定のシステムの不安定性に対処したりするために、アンインストールできないソフトウエア更新プログラムを展開することが必要になる場合もあるが、要求のカテゴリにこの点を反映させる必要がある。

- ネットワークインフラストラクチャが受ける可能性がある影響。同時に多数のコンピュータを対象に大きなソフトウエア更新プログラムを展開すると、ネットワークのパフォーマンスが低下し、環境全体の適切な運用に悪影響を及ぼす可能性がある。ソフトウエア更新プログラムのすべてのドキュメントに目を通し、ソフトウエア更新プログラムのサイズとその適用先のコンピュータの数を把握しておく必要がある。この情報は、リリースのスケジュールを適切に設定するのに役立つ。

- インストール中に特定のサービスを停止、一時停止、または終了する必要があるか。これにより、組織の重要サービスが影響を受けたり、インストール中にエンドユーザがコンピュータを操作できなくなったりする場合もある。

3.3.1.3 ソフトウエア更新プログラムを展開するための承認を取得する

変更要求の優先度とカテゴリが決まったあと、ソフトウエア更新プログラムを運用環境に展開する前に、変更要求を確認して承認する必要がある。変更要求を承認するには、次の手順を実行する必要がある。

- 意思決定プロセスに関与する者を決定する。
- 変更要求を確認し、ソフトウエア更新プログラムを展開した場合のリスクと結果を評価して、最適なアクションの方向性を選択する。
- 影響を受けるすべてのシステムにソフトウエア更新プログラムを展開する担当者を特定する。

ソフトウエア更新プログラムを運用環境に展開するには、その確認と承認にだれが関与するかを決定することが重要である。多くの組織では、影響を受けるビジネスの全領域の代表者で構成された CAB（Change Advisory Board：変更諮問機関）を設けている。CAB のメンバには、更新プログラムの展開に使用されるテクノロジとサービスの経験者を含める必要がある。ビジネス、ネットワーク、セキュリティ、サービスデスク、テクニカルサポートのチーム代表も、このグループに含める必要がある。

たとえば緊急のソフトウエア更新など、迅速な決定が必要な場合には、緊急の変更を承認する権限を持っていて迅速な決定を行える要員で構成された CAB のサブセットに承認が委任されることがある。

CAB が更新を考慮するときには、運用環境に対するソフトウエア更新プログラムのリスクと影響を評価して、展開するかどうかを決定する必要がある。この決定を行うには、次の点を考慮する必要がある。

- 運用環境でほかに何が起きているか。
- ソフトウエア更新プログラムを適用する場合としない場合の影響は何か。
- ソフトウエア更新プログラムを展開する場合としない場合の予想コストはどのくらいか。
- ソフトウエア更新プログラムの展開時に、セキュリティ上の脆弱性またはシステムの不安定性の脅威を軽減するために実行できる手順は何か。
- コンピュータのダウンタイムの影響は何か。ソフトウエア更新プログラムの展開を延期することのリスクと、環境へソフトウエア更新プログラムを展開するときにコンピュータのダウンタイムが生じることのリスクを比較検討しなければならない場合がある。
- ソフトウエア更新プログラムを展開するための最良でもっとも効果的なメカニズムは何か。
- ソフトウエア更新プログラムに伴う既知の問題や副作用は存在するか。また、システムを再起動する必要はあるか。
- ソフトウエア更新プログラムを展開したり、展開中に発生する問題に対処したりするためのリソースは十分に確保されているか。
- ソフトウエア更新プログラムを展開する前に解決する必要がある依存関係や前提条件に対してどう対処するか。

ソフトウエアの脆弱性に対する最良の対応策は、ソフトウエア更新プログラムを展開して問題を解決することであるが、運用環境内のシステムにソフトウエア更新プログラムをロールアウトしているときに、ネットワークポートのクローズやシステムへの外部アクセスのシャットダウンなど、短期の対応策を展開することが望ましい場合がある。このような対応策を適用することには、次のような利点がある。

- ほとんどのソフトウエア更新プログラムでは、修正を有効にさせるため、パッチを適用後にそのコンピュータを再起動する必要がある。コンピュータの再起動が保守契約などで制限されているため、ソフトウエア更新プログラムをすぐに展開できない場合には、推奨された対応策を講じることにより、ソフトウエア更新プログラムを展開するまでの間の保護手段を講じる。あるいは、セキュリティ更新プログラムを展開だけしておいて、すぐには自動再起動しないようにする場合もある。この場合は、セキュリティ更新プログラムを通常の時間帯にインストールして、あとで保守に適した時間帯にコンピュータを再起動することができる。

- 対応策は、ソフトウエア更新プログラム自体よりリスクが小さく、より短時間で適用できて、必要なテストもより少ないことが多い。たとえば、ネットワークポートを無効化したり、特定のセキュリティ脆弱性が露出するようなサービスやシステムをシャットダウンしたあとでソフトウエア更新プログラムを適用したりする方がはるかに簡単な場合がある。

多くの場合、コンピュータを堅牢にする対応策を実装すると、多くの一般的なセキュリティ脆弱性からコンピュータを保護できる。効果的にコンピュータを保護する代表的な対応策として、特定のネットワークポートをブロックする方法と、使用されていないサービスを無効にする方法の 2 つがある。コンピュータを堅牢にする対応策の詳細については、[6]を参照のこと。

対応策を展開してセキュリティ脆弱性にさらされるリスクを軽減する場合でも、セキュリティ更新プログラムの展開スケジュールを確立する必要がある。たとえば、システムにまだパッチを適用していない状態で、ワームやウィルスに感染したコンピュータをネットワークに導入すると、保護されていない全システムに感染が一気に広がる可能性がある。対応策を展開しても、ソフトウエアを更新する必要がなくなるわけではなく、単に変更要求の優先度が下がるだけである。

3.3.1.4　ソフトウエア更新プログラムの展開の責任者を決める

ソフトウエア更新プログラムを展開することと、必要に応じてなんらかの対応策を講じることについて合意に達したら、これらの変更を確実に行う責任者を特定する必要がある。この責任者は、以下の手順を実行する必要がある。

- 必要な変更を行うための計画を作成する。
- 必要なリソースを決定して取得する。
- 変更を適用するために必要なスクリプト、ツール、およびドキュメントの開発を手配する。
- 適切なテストが確実に実施されるようにする。
- 変更が運用環境に確実に展開されるようにする。
- 展開の成功または失敗を評価する。

上記の活動を監督する責任者がいないと、ソフトウエア更新プログラムが展開されないというリスクが生じる。

3.3.2　リリースを計画する

リリースの計画は、ソフトウエア更新プログラムを運用環境にどのようにリリースするかを策定するプロセスである。

さまざまな問題や制約の存在によって、ソフトウエア更新プログラムを運用環境に完全に展開するために必要な手順が決まる場合がある。たとえば、ソフトウエア更新プログラムの展開を担当する場合は、以下の点を考慮する必要がある。

- ソフトウエア更新プログラムが自動でインストールされるまで、ユーザに与えられる時間はどのくらい必要か。許される時間は、ユーザの役割と担当範囲、ソフトウエア更新プログラムで対処するシステムの不安定性やセキュリティ脆弱性の特徴など、さまざまな要素によって決まる。

- ソフトウエア更新プログラムによっては、インストール先のコンピュータに対する管理権限が必要になる場合がある。ほとんどのエンドユーザはローカル管理者権限を与えられていないため、ソフトウエア更新プログラムのインストールに使用されるツールが上位の権利および権限を取得して、ソフトウエア更新プログラムをクライアントコンピュータにインストールできるようにする必要がある。

- ソフトウエア更新プログラムのインストール時に一定のディスク容量が必要な場合、またはソフトウエア更新プログラムがインストール前にローカルにキャッシュされる場合は、各クライアントコンピュータの空きディスク容量をチェックする必要がある。

- ソフトウエア更新プログラムのサイズが大きい場合は（たとえば数メガバイト）、リモート IT クライアント（在宅勤務者が使用するものなど）でダウンロードするのに時間がかかることがある。ソフトウエア更新プログラムが緊急と分類されていない場合は、それらのクライアントが物理的にネットワークに接続されるまで、インストールを延期した方がよい場合がある。

- ビジネスに不可欠なコンピュータは、変更とコンピュータの再起動に使用できる時間（停止期間）が限られていることがある。ソフトウエア更新プログラムの展開と、展開後に必要になるシステム再起動を、この停止期間内にスケジュールする必要がある。

- オペレーティングシステムのセキュリティ設定を考慮しなければならない場合がある。たとえば、Windows ベースのクライアントコンピュータがグループポリシーの設定を使ってロックダウンされている場合、ソフトウエア更新プログラムを正しくインストールできないことがある。

- 更新プログラムの適用先の製品が Microsoft の Windows インストーラなどの独立したインストールソフトウエアを使って展開されたものである場合は、更新時に元のインストールファイルへのアクセスが必要になることがある。製品が当初 CD ドライブなどの物理メディアからインストールされた場合、インストールソフトウエアは現在ドライブに挿入されている CD 内を探して元のファイルを見つけようと試みる。

3.3.2.1 リリース計画を作成する

この時点で、運用環境内の各コンピュータにソフトウエア更新プログラムを展開する順序を計画し決定する必要がある。リリース計画を作成するときに考慮する必要が生じる可能性のある問題を以下に示す。

- 運用環境内のすべてのサーバにソフトウエア更新プログラムを適用する場合、最初にどのサーバにパッチを適用するべきか。最初に管理インフラストラクチ

ャにパッチを適用すると、管理者がそれらのサービスを使用して展開の進捗状況を監視できるようになる。

- 運用環境の一部に対してほかより先にパッチを適用するビジネス上の理由はあるか。ソフトウエア更新プログラムを、セキュリティ脆弱性または潜在的なシステムの不安定性のリスクがあるコンピュータに対してまず適用し、それらのコンピュータへのパッチ適用が終わってから、ほかのコンピュータへのロールアウトを続行する強固な理由が存在する場合がある。

- サイト間の使用可能ネットワーク帯域幅がロールアウトの順序にどのような影響を与えるか。サイトによっては、ネットワーク帯域幅の制約により、ソフトウエア更新プログラムをほかのサイトのように迅速にロールアウトできない場合がある。ネットワーク接続が良好なサイトには、そうでないサイトよりも速くソフトウエア更新プログラムを展開できる。

最後に、ソフトウエア更新プログラム、その重大度、影響、および展開する手順に関する情報を、いつどのようにして一般ユーザ、企業、およびサービスデスクに伝えるかを決定する必要がある。

変更要求が緊急の場合は、以下の点を考慮する必要がある。

- 管理アーキテクチャサーバにもパッチを適用する必要がある場合は、ローカル管理者が手動でこれらのコンピュータにパッチを適用することによって、運用環境内のほかのコンピュータにソフトウエア更新プログラムがロールアウトされている間にこれらのサーバが再起動されないように計画するのが適切な場合がある。

- すでに 1 つのサイトまたはコンピュータグループが、ソフトウエア更新プログラムの対象になるセキュリティ侵害やシステムの不安定性の悪影響を受けている場合は、最初にそれらのコンピュータにソフトウエア更新プログラムを適用する必要がある。

3.3.3 リリースを作成する

リリース計画を策定し終わったら、プロセスの次の段階として、ソフトウエア更新プログラムを運用環境に展開するときに管理者が使用するスクリプト、ツール、および手順を開発する。

更新プログラムがすでに実行可能なファイル形式にパッケージされている場合は、展開用に再パッケージするための追加作業は必要ない。パッケージされていない場合は、ソフトウエア更新プログラムを配布してインストールするためのプログラムを作成する必要が生じることがある（このプロセスを自動化するためのツールまたはウィザードがオペレーティングシステムに付属している場合がある）。

3.3.4 受入れテスト

この時点までのテストの目的は、開発環境内でのソフトウエア更新プログラムおよびリリースパッケージの正常な動作を確認することであった。*受入れテスト*段

階において、開発者とビジネス代表者は、運用環境を忠実にミラー化した環境で更新プログラムが機能することと、ソフトウエア更新プログラムが展開されたあとでビジネスに不可欠なシステムが正常に稼働し続けることを確認できる。管理者はビジネスの代表者と共同で、ソフトウエア更新プログラムがビジネスに不可欠と見なされる場合に実行する簡単なテストセットと、ソフトウエア更新プログラムの優先度が低い場合に使用できる詳細なテストセットを作成する必要がある。

ただし、ソフトウエア更新プログラムの重要度がどの程度であっても、次のことがわかる最低レベルのテストは必ず実行する必要がある。

- インストールが完了すると、設計どおりにコンピュータが再起動される。
- 低速または信頼度の低いネットワーク接続を使用するコンピュータが対象になる場合に、これらのリンクを通じてソフトウエア更新プログラムをダウンロードでき、ダウンロード完了後に正しくインストールできる。
- ソフトウエア更新プログラムとともに、ソフトウエア更新プログラムを正しく削除できるアンインストール手段が提供されている。
- ビジネスに不可欠なシステムとサービスが、ソフトウエア更新プログラムがインストールされたあとも稼働し続ける。

ソフトウエア更新プログラムを運用環境に展開する前に、テストで使用されたトラブルシューティング手順、操作、およびツールについて情報を収集し、サービスデスクサポートスタッフおよび運用チームがその情報を利用できるようにすることが重要である。テストの結果として次のものが作成されることが望ましい。

- 標準的なトラブルシューティングの手順と、関連する回避策を記述した文書。
- 連絡先とエスカレーションパスの一覧。
- 運用スタッフが運用環境でのリリースを効果的に監視できるようにする、スクリプト、規則、および情報（カウンタ、イベント、しきい値など）。

テストをどれだけ実行しても、ソフトウエア更新プログラムを運用環境にロールアウトすると、実験環境では予期も再現もできない結果が生じることが多い。起こり得る障害の影響が多くのクライアントコンピュータに及ばないようにするために、管理者は、組織全体への展開を実行する前に、ソフトウエア更新プログラムを限られた範囲の代表的コンピュータグループにロールアウトし、ビジネスに不可欠なシステムとアプリケーションに影響がでないかを確認すべきである。

3.3.5 要約

評価、計画、およびテストフェーズで留意する必要がある重要な点は、次のとおりである。

- ソフトウエア更新プログラムを展開することがビジネスにとって最良の対策かどうかを、正式のプロセスを使用して判断する。
- ソフトウエア更新プログラムを確実に展開するための責任者を決定する。

- ソフトウエア更新プログラムの承認後に、組織への展開方法を計画する必要がある。
- 展開前に更新プログラムを実験環境でテストし、必要に応じて運用環境でパイロットテストを実施して、重要なビジネスアプリケーションが損なわれないことを確認する。

3.4 展開フェーズ

展開フェーズでは、ソフトウエア更新プログラムを運用環境に展開するのに必要なタスクと作業に重点が置かれる。ソフトウエア更新プログラムの展開は、次の作業で構成される。

- 展開の準備
- 対象のコンピュータに対するソフトウエア更新プログラムの展開
- 実装後のレビュー

3.4.1 展開の準備

新しいリリースごとに、運用環境の準備を行う必要がある。準備のうちでもっとも重要なのは、さし迫った更新プログラムのリリースについて、エンドユーザと管理者に通知することである。更新プログラムとそのインストール方法について知らせる、明確で識別しやすい電子メールメッセージをユーザと管理者に送信するのが理想的である。ユーザや管理者が実行するべきアクションを忘れないように、このメールには追跡用のフラグを付ける必要がある。

営業時間外にデスクトップに更新プログラムを展開する場合は、ユーザが指定された日の夜間にコンピュータの電源を入れたままにする必要があることを、（それが標準的な行為でないなら）電子メールメッセージで知らせる必要がある。

3.4.2 対象のコンピュータにソフトウエア更新プログラムを展開する

ソフトウエア更新プログラムを運用環境に展開するために使用するプロセスは、リリースの種類と性質、および選択したリリースメカニズムによって決まる。また、ソフトウエア更新プログラムが緊急かどうかによっても大きく左右される。緊急の変更は迅速に実施する必要があるため、変更の展開方法に違いがある。その違いは、このセクションの全体を通して強調されている。

段階的な展開によってソフトウエア更新プログラムをリリースすることが理想的である。それにより、ソフトウエア更新プログラムの初期配布によって生じる可能性のある障害や悪影響を最小限に抑えることができる。

コンピュータは、次のような理由で更新プログラムのインストールに失敗する場合がある。

- コンピュータがオフラインになっている。
- コンピュータが再構築中、または再イメージ化中である。

- コンピュータに十分な空きディスク容量がない。

通常の展開中に例外が発生した場合は、展開を停止し、根本的な原因を特定し、再展開を行うのに十分な時間がある。しかし、緊急展開の実行中は、優先順位付けと根本原因の評価に、ごく短い時間しか割り当てられない。どちらのケースでも、ロールアウトを停止し、失敗した更新プログラムをアンインストールし、その更新プログラムを再展開するための計画を用意しておくことが重要である。

3.4.3 実装後のレビュー

実装後のレビューは、通常はリリースを展開してから 1 週間から 4 週間のうちに実行し、パッチ管理プロセスに改善するべき点があるか調べる必要がある。一般的な確認事項は次のとおりである。

- 脆弱性通知によって特定された脆弱性を脆弱性スキャンレポートとセキュリティポリシー標準に追加し、攻撃が再発する機会が生じないようする。
- 展開後に、ビルドイメージが更新されて最新のソフトウエア更新プログラムを含むようになったことを確認する。
- 計画と実際の結果を比較検討する。
- リリースに関連するリスクを検討する。
- インシデント全体を通じての組織のパフォーマンスをレビューする。この機会を利用して対応計画を改善し、得られた教訓を含める。
- サービスの時間帯の変更を検討する。
- ダウンタイムのコストと復旧のコストの両方について、インシデントの損害とコストの合計を査定する。
- 環境に合わせて別のベースラインを作成するか、既存のベースラインを更新する。

3.4.4 要約

展開フェーズでは、次の主な作業を完了する必要がある。

- ソフトウエア更新プログラムを運用環境にロールアウトする順序を確立する。
- 運用環境を調査して、ソフトウエア更新プログラムを処理できることを確認する。
- ソフトウエア更新プログラムファイルを、SMS 配布ポイントまたは SUS サーバ（Windows 環境の場合）に配置する。
- ソフトウエア更新プログラムを運用環境に展開する。
- 環境を再スキャンして結果を査定し、ソフトウエア更新プログラムのインストールに失敗したコンピュータにパッチを適用する。
- 展開が完了したら、パッチ管理プロセスのレビューを実行する。

ソフトウエア更新プログラムの展開は、プロセスの終わりではない。査定および調査フェーズがまだ続いており、展開が終わったらこのフェーズに戻る必要がある（その結果に従ってITインベントリを更新するためにも）。

4　パッチ管理の指標[5]

パッチおよび脆弱性の指標は、攻撃を受ける可能性の高さ、軽減対処時間、およびコストの3つに大きく分類される。このセクションでは、分類ごとの指標の例を示す。

4.1　システムが攻撃を受ける可能性の高さの測定

組織が攻撃を受ける可能性の高さは、複数の測定値から概算できる。各組織は、必要なパッチの数、脆弱性の数、および実行されているネットワークサービスの数をシステム単位で測定できる。これらの測定値は、システム内のコンピュータごとに個別に取得し、それらの結果を累計してシステム全体の結果とするべきである。

生データと比率データ（たとえば、コンピュータあたりの脆弱性の数など）の両方が重要である。脆弱性の数、未適用のパッチの数、および公開されているネットワークサービスの数が多いほどシステムが侵入される可能性が高まるため、これらの生データを測定することが、システムが直面するリスクの全体を明らかにするのに役立つ。多数のコンピュータで構成される大規模なシステムは、同様に構成された小規模なシステムに比べてセキュリティが低い。これは、大規模なシステムが小規模なシステムに比べて常にセキュリティが甘いことを意味するわけではない。誤った解釈を避けるために、複数のシステムのセキュリティプログラムの効果を比較するときには、比率データを使用するべきである。比率データ（コンピュータあたりの未適用パッチ数など）を使用すると、システム間の比較を効果的に行うことができる。加工されていない生の結果と割合はどちらも有用であり、それぞれ使用目的が異なるため、システムごとに必要に応じて両方を測定し、公開するべきである。

最初の測定では、攻撃者がシステムのコンピュータの脆弱性に直接アクセスするのを防ぐために必要となる、システムのセキュリティ境界のアーキテクチャ（ファイアウォールなど）については考慮するべきではない。これは、たとえシステムが強力なセキュリティ境界によって保護されていても、システム内のすべてのコンピュータのセキュリティを確保することが基本であるからである。これによって、内部の者による攻撃を防いだり、外部から侵入に成功した攻撃者がシステム内のすべてのコンピュータに影響を及ぼすのを防いだりすることができる。

ほとんどのシステムでは（さまざまな理由で）セキュリティが完全には確保されないという認識に立ち、システムのセキュリティ境界のアーキテクチャを考慮しながら、測定値を再計算する必要がある。これにより、システムが外部の攻撃者から実際に攻撃を受ける可能性について、有意な測定値を得ることができる。た

[5] このセクションの大部分は、パッチ管理に関するNISTのホワイトペーパー[1]からの引用である。詳細については同ホワイトペーパーを参照のこと。

とえば、この 2 番目の測定では、システムが主要なファイアウォールを経由して悪用される可能性がないと判断される場合は、コンピュータの脆弱性の数、ネットワークサービスの数、または必要なパッチの数をカウントする必要はない。

システムが攻撃を受ける可能性の高さに関する最初の測定では、システムのセキュリティ境界のアーキテクチャを考慮するべきではないが、個々のコンピュータのセキュリティアーキテクチャについては考慮することが望ましい場合もある。たとえば、ネットワーク接続によって悪用されるおそれがある脆弱性の数は、コンピュータのパーソナルファイアウォールによってそのような悪用行為が防止される場合はカウントしないこともある。ただし、コンピュータのセキュリティアーキテクチャの変更によって脆弱性が悪用される可能性があるため、このような措置は慎重に行うべきである。

4.1.1 パッチの数

システムごとに必要なパッチの数の測定は、エンタープライズ向けパッチ管理ツールを導入している組織ではごくあたりまえの作業である。この種のデータはパッチ管理ツールが自動的に提供するためである。必要なパッチの数は、システムが攻撃を受ける可能性の高さを見積もる上で重要ではあるが、特定のセキュリティパッチによって修正される脆弱性は 1 つの場合もあれば複数の場合もあり、脆弱性の深刻度もさまざまである可能性があるため、指標としての有効性は限られている。また、対応するパッチが存在しない脆弱性が公開されることも多い。このような脆弱性は組織のリスクを増大させるが、必要なパッチの数を測定するだけではそれらの脆弱性を把握できない。この測定値の品質は、パッチを公表したベンダが「重要」と評価したパッチの数を考慮に入れ、重要なパッチと重要でないパッチの数を比較することで改善できる。

4.1.2 脆弱性の数

システムごとに存在する脆弱性の数の測定は、組織が攻撃を受ける可能性の高さを測る尺度としては有効であるが、完璧にはほど遠い。脆弱性スキャンツールを採用している組織では、ツールによって必要な統計情報が出力されるため、ほとんどの場合この指標が採用される。パッチ数の測定と同様に、各組織は脆弱性の深刻度評価を考慮に入れ、測定では深刻度（または深刻度の範囲）ごとに脆弱性の数が出力されるようにするべきである。通常、脆弱性の評価システムは、脆弱性データベース、脆弱性スキャナ、およびパッチベンダ自体によって提供されるが、現在のところ標準化された評価システムは存在しない。これらの評価システムは、一般的で典型的な組織に対する脆弱性の影響を見積もっているだけである。脆弱性の本当の影響は、組織固有のセキュリティインフラストラクチャおよびアーキテクチャに照らして個々の脆弱性を調べることによってのみ明らかにできる。また、システムに対する脆弱性の影響は、ネットワーク上のシステムの位置によっても異なる（通常、インターネットからアクセスできるシステムでは脆弱性の深刻度が増す）。

4.1.3 ネットワークサービスの数

攻撃を受ける可能性の高さを示す指標の最後の例は、システムごとに実行されているネットワークサービスの数の測定である[6]。この指標の背景にあるのは、個々のネットワークサービスは潜在的な脆弱性の集合を表しており、システムで実行されるネットワークサービスの数の増加に伴いセキュリティリスクが増大する、という考え方である。大規模なシステムでは、この値を測定することによって、（現在および将来において）システムがネットワーク攻撃を受ける可能性の高さがわかる。また、複数のシステムの間でネットワークサービスの数を比較することで、ネットワークサービスを効率よく減らして稼働しているシステムを特定することもできる。アクティブなネットワークサービスの数が多いからといって、必ずしもシステム管理者の管理が誤っているわけではない。

しかし、そのような結果を注意深く調べ、不要なネットワークサービスがすべてオフになっていることを確認するべきである。

4.2 軽減対処時間

組織がどれだけ迅速に新しい脆弱性を識別して分類し、それに対処し、組織に与える潜在的影響を軽減できるかを測定することも重要である。脆弱性が公表されてから悪用手段が公開されるまでの平均時間はここ数年で劇的に短くなったため、対処時間はますます重要になっている。取得できる主な対処時間の測定値としては、脆弱性およびパッチの識別に要する対処時間、パッチの適用に要する対処時間、および緊急の構成変更に要する対処時間の 3 つがある。

4.2.1 脆弱性およびパッチの識別に要する対処時間

この指標は、責任者（システム管理者、IT 運用管理者など）が新しい脆弱性またはパッチの情報を知るまでに要する時間を測定する。この時間は、脆弱性またはパッチが公表された時点から始まる。この測定値は、さまざまなパッチと脆弱性をサンプリングして取得するべきであり、責任者が情報収集に使用する各種リソースをすべてこの測定値に含めるべきである。

4.2.2 パッチの適用に要する対処時間

この指標は、システム内の関連するすべての IT 機器にパッチを適用するのに要した時間を測定する。この時間は、責任者がパッチの存在を認識した時点から始まる。この測定は、パッチが正常にインストールされたことを比較的簡単に検証できるパッチに対して実施するべきである。この測定には、次の作業に要した個別の時間と集計時間を含めるべきである。

- パッチの分析

[6] ネットワークサービスやネットワークポートの重要性はそれぞれ異なるため、各組織はネットワークサービスやネットワークポートの数を数えるときに、重み付けすることを検討するべきである。たとえば、1つのネットワークポートが複数のサービスによって使用されている場合もある。また、あるサービスが、ほかのサービスよりも攻撃される可能性がはるかに高い場合や、より重要な機能を実行している場合もある。

- パッチのテスト
- 構成管理プロセス
- パッチ展開作業

検証は、エンタープライズ向けパッチ管理ツールの使用または脆弱性のスキャン（ホストベースとネットワークベースの両方）によって行うことができる。重要なセキュリティパッチと重要でないセキュリティパッチでは、通常、組織が使用するプロセスが異なり、タイミングも異なる場合があるため、必要に応じて両種類のパッチに対してこの測定を行うことが有用である。

4.2.3 緊急の構成変更に要する対処時間

この指標は、軽減しなければならない脆弱性がありながらパッチが提供されていない場合に適用される。このような場合、組織は脆弱性の悪用から組織を守るために、機能の削減につながる可能性がある緊急の構成変更を行わなければならない。このような変更は、多くの場合ファイアウォール、電子メールサーバ、Web サーバ、または DMZ 内のサーバで行われる。変更には、特定の電子メール添付ファイル、電子メール件名、ネットワークポート、およびサーバアプリケーションの無効化やフィルタリングが含まれる場合がある。この指標では、責任者が脆弱性の情報を得た時点から、許容できる回避策が適用され検証された時点までの時間を測定しなければならない。多くの脆弱性には緊急の構成変更は必要ないため、この指標はシステムの脆弱性の一部だけを対象とする。

通常、これらの作業は緊急時に行われるため、妥当な数の測定サンプルを得ることは難しい。とはいえ作業の重要性を考えれば、これらの緊急プロセスをテストするべきであり、テストケースを通じて対処時間の指標を得ることができる。時間を測定できる緊急プロセスの例を次に示す。

- ファイアウォールまたはルータの構成変更
- ネットワークの切断
- 侵入防止機器の作動または再設定
- 電子メールのフィルタリング規則の追加
- コンピュータの隔離
- スタッフへの緊急通知

システムによっては、テストの対象となる緊急プロセスが大きく異なるため、この指標の結果も大きく異なる可能性がある。各組織は、一貫したテスト結果を得るために、可能な限り標準のシステム緊急時対応プロセスを作成するべきである。各組織は、緊急の構成変更が発生したあとは必ず指標を取得し、運用報告の一部として評価することにより、緊急変更プロセスにおける今後の改善措置や改善領域を明らかにする必要がある。

4.3 コスト

パッチおよび脆弱性管理は、その作業を多数の要員やグループで分担することが多いため、コストの測定が難しい。ほとんどの組織では、パッチや脆弱性にかかわる職務を複数のグループで分担し、フルタイムとパートタイムのさまざまな職員に割り当てる。コストについて取得するべき主な測定値は 3 つある。システム管理サポートのコスト、エンタープライズ向けパッチおよび脆弱性管理ツールのコスト、パッチおよび脆弱性管理プログラムの不具合により発生したインシデントに対応するためのコストである。

4.3.1 システム管理のコスト

この測定値は、通常は正確に取得することが難しいが、重要な測定値である。主な問題は、システム管理者が従来からセキュリティに費やした時間を計算するように求められていない点にある。セキュリティパッチおよび脆弱性管理に費やした時間についてはなおさらである。各組織が IT セキュリティの実際のコストを測定するための全体的な取り組みを改善するにしたがって、システム管理者によるパッチおよび脆弱性管理の時間的コストの測定も容易になる。

4.3.2 エンタープライズ向けパッチおよび脆弱性管理ツールのコスト

この測定値には、パッチ適用ツール、脆弱性スキャンツール、脆弱性 Web ポータル、脆弱性データベース、および（パッチの検証に使用する）ログ分析ツールを含める。侵入検知ツール、侵入防止ツール、および（侵入検知に使用する）ログ分析ツールは含めない。組織はまず、各ソフトウエアパッケージの購入価格と年間保守コストを計算する必要がある。次に、すべてのソフトウエアの購入価格（各ソフトウエアの購入価格の合計）と年間保守コスト（各ソフトウエアの年間保守コストの合計）を含む年間コストを計算しなければならない。この指標を作成するには、各ソフトウエアパッケージの購入価格を推定耐用年数で割ったものに、年間保守コストを加算する。ソフトウエアが定期的にアップグレードされる場合は、購入価格の代わりにアップグレード価格を使用する。

一般に、概算年間コストは、各製品の年間保守費の合計に、各製品の償却費の合計を足した金額になる。償却費は、製品の購入価格（またはアップグレード価格）をその製品の推定稼働年数（耐用年数）で割った金額として定義される。

たとえば、ある組織に以下のソフトウエアがあるとする。

製品	購入価格	アップグレード価格	耐用年数	年間保守費
エンタープライズ向けパッチ管理ソフトウエア	£30,000	£15,000	4年	£3,000
脆弱性スキャナ	£20,000	£10,000	3年	£2,000

この組織では、脆弱性スキャナは 3 年後にアップグレードする予定だが、エンタープライズ向けパッチ管理ソフトウエアは 4 年後に新しい製品に切り替える予定であるとする。この場合、年間コストの見積もりは、(£3,000 + £2,000) + (£30,000/4) + (£10,000/3) = £15,833 となる。

4.3.3 プログラムの不具合のコスト

この測定値では、パッチおよび脆弱性軽減プログラムがもっと効果的であれば防止できたはずのすべてのインシデントと、パッチ適用プロセス自体が原因で起きたすべての問題（パッチ適用による意図しないアプリケーションの破損など）による、ビジネスへの影響の総コストを計算する。コストの数値には、有形の損失（従業員の作業時間の損失や破損したデータなど）とともに無形の損失（組織の評判の低下など）も含める。この数値は年間単位で計算する。この測定結果は、パッチおよび脆弱性管理プログラムの費用対効果を評価するときに使用する。プログラムの不具合により発生するコストが非常に大きい場合、組織はパッチおよび脆弱性管理プログラムに投入するリソースを増やすことによって、費用を節約できる可能性がある。不具合に伴うコストが非常に小さい場合、組織はパッチおよび脆弱性管理に対するサポートのレベルを現状維持するか、費用対効果を最適化するためにレベルを下げることも考えられる。

4.3.4 パフォーマンス目標と費用対効果

システムの所有者とシステムのセキュリティ責任者には、各指標の現実的な履行目標を伝える必要がある。これらの目標を達成できたら、さらに意欲的な目標を設定することもできる。パッチおよび脆弱性のセキュリティレベルを上げる場合は、システムのセキュリティ責任者やシステム管理者の負担が重くなりすぎないように慎重に行うことが重要である。

プログラムの費用対効果は、プログラムの実行に関連するコスト指標と、プログラムの不具合によるコストとを比較することによって計算できる。また、プログラムの実行に関連するコスト指標と、プログラムのパフォーマンスを示す指標（応答時間や攻撃を受ける可能性の高さに関する指標）とを比較することによっても計算できる。

4.4 要約

すべての組織は、組織のパッチおよび脆弱性管理プログラムの有効性を常に測定し、必要に応じて是正措置を適用するべきである。これは、パッチおよび脆弱性指標プログラムを策定することによって実現できる。導入する指標は、組織の成熟度にも依存するため、パッチおよび脆弱性管理の手法には、成熟度に適したものを選ぶべきである。各組織は、各システムについて測定する指標の種類と各指標の詳細を文書化するべきである。システムの所有者とシステムのセキュリティ責任者には、現実的な履行目標を伝える必要がある。

5 その他の情報源

5.1 WARP

WARP（Warning, Advice and Reporting Point）は、イギリスの国家的重要インフラストラクチャを電子的攻撃から守るための NISCC の情報共有戦略の一部である。WARP は、警報および警告のより効率的な伝達をシミュレートし、認識およ

び教育を改善し、インシデントの報告を奨励することによって、情報セキュリティを改善する活動において、成果を上げている。

WARP の詳細については、WARP のホームページ（http://www.warp.gov.uk/home.htm）を参照のこと。

パッチ管理は、WARP が取り組む対象として一般的である。既存の WARP は次のような活動を行ってきた。

- WARP 掲示板と電子メールリストを使用して、コミュニティ内でパッチ管理の経験を共有してきた。
- 共同でソフトウエア更新通知と勧告情報のフィルタリングに取り組んできた。

5.2 Uniras

イギリス政府の CERT（Computer Emergency Response Team：コンピュータ緊急対応チーム）である Uniras は、電子攻撃インシデントへの対応におけるサポートを政府および CNI 組織に提供する。このサポートは、電話による質問への回答から現場での支援までさまざまである。このサポートは NISCC によって監督される。

チームのメンバは、IT セキュリティインシデント管理の分野における専門家であり、各自の日常的な活動のほかに、NISCC コミュニティのためのコースでインシデント管理の講義を定期的に行っている。また、インシデント対応チームの設立を望むほかの組織にも支援を提供している。

（訳注）Uniras の機能は、英政府機関である CESG (通信機器セキュリティグループ：Communications-Electronics Security Group）の下に設置された GovCERTUK が受け継いでいる。

参考：http://www.govcertuk.gov.uk/

6 付録A：Windowsのパッチ管理ツール [7]

Microsoft がセキュリティの脆弱性を認識すると、その問題が MSRC（Microsoft Security Response Centre）および適切な製品群によって評価され、確認される。MRSC はその問題を解決するセキュリティパッチを作成し、脆弱性の報告者と共同で、セキュリティパッチの詳細を含むセキュリティ情報の形での公開情報のリリースを手配する。

Microsoft はその後、Microsoft ダウンロードセンターや以下を始めとするほかのサービスを通じて、ソフトウエア更新プログラムを配信する。

- Microsoft Windows Update
- Microsoft Office Update

[7] このセクションは、Microsoft の Web サイトに掲載されている情報に基づいている（[1]を参照）。（訳注）すべて 2006 年 10 月現在の情報である。

- SUS（Microsoft Software Update Services）
- WSUS（Microsoft Server Update Services - SUS とは別物）
- SUS Feature Pack を適用した SMS（Microsoft Systems Management Server）
- SMS（Microsoft Systems Management Server）2003

MSRC は、ソフトウエア更新プログラムがリリースされるときに、関連するセキュリティ情報を発信する。

通常、セキュリティパッチは、サポートされている製品の現在のサービスパックだけでなく 1 つ前のサービスパックでも利用できるようになっている。しかし、これが必ずしも当てはまらない場合があるので、使用している製品のプロダクトサポートライフサイクルポリシーを確認する必要がある。

6.1.1 ソフトウエアの更新に関する用語

表 2に、ソフトウエアの更新に関する Microsoft の用語を示す。Microsoft が「パッチ」という用語を使用するのは、「セキュリティパッチ」という用語の一部として、またはパッチ管理のプロセスを説明するときだけである。

表 2 ソフトウエアの更新に関する Microsoft の用語

用語	定義
セキュリティ更新プログラム（Security patch）	特定の製品に対して広範にリリースされる、セキュリティの脆弱性に対処する更新プログラム。セキュリティ更新プログラムは、通常、深刻度が設定されている。この深刻度は実際には、セキュリティ更新プログラムが対処する脆弱性のMSRC深刻度評価を示している。
重要な更新プログラム（Critical update）	特定の問題に対して広範にリリースされる、セキュリティ関連以外の重大なバグを解決する修正。
更新プログラム（Update）	特定の問題に対して広範にリリースされる、セキュリティ関連以外の重大ではないバグを解決する修正。
修正プログラム（Hotfix）	製品の問題を解決するために使用される1つまたは複数のファイルにより構成される単一のパッケージ。修正プログラムは特定のユーザの状況に対処するものであり、Microsoftとのサポート関係を通じてのみ利用できる。Microsoftの書面による法的な同意がなければ、そのユーザの組織外に修正プログラムを配布することはできない。以前はQFE（Quick Fix Engineering update）、パッチ、更新という用語が修正プログラムの同義語として使用されていた。
更新プログラムのロールアップ（Update rollup）	セキュリティ更新プログラム、重要な更新プログラム、更新プログラム、および修正プログラムの集合。累積的にリリースされていくか、IIS（Microsoft Internet Information Services）やMicrosoft Internet Explorerなどの単一の製品コンポーネントを対象にリリースされる。複数のソフトウエア更新プログラムをより簡単に展開できる。
サービスパック	製品のリリース後のすべての修正プログラム、セキュリティ更新プログラム、重要な更新プログラム、および更新プログラムと、ほかのソフトウエア更新プログラムで提供できなかった多くの問題の解決を含む、累積的なセット。サービスパックは、ユーザから要望のあった限定された数の設計変更や機能を含む場合もある。サービスパックは、Microsoftにより、ほかのどのソフトウエア更新プログラムよりも広範に配布され、テストされる。
統合サービスパック	製品とサービスパックを1つのパッケージに組み合わせたもの。
Feature Pack	製品に機能を追加する新しい機能のリリース。通常は、製品の次期リリースに組み込まれる。

6.1.2 ツールとテクノロジ

ここでは、あらゆる規模の組織でソフトウエア更新プログラムのインストールを管理および制御するために使用できる自動化されたツールについて検討する。企

業における Windows ベースシステムのパッチ管理に使用できる Microsoft の主なテクノロジは、次の 3 つである。

- SUS（Software Update Services）
- WSUS（Windows Server Update Services：SUS の最近の拡張機能）
- SMS（Systems Management Server）

6.1.3 SUS（Software Update Services）

SUS は、Microsoft Windows Update Web サイト（http://windowsupdate.microsoft.com/en/default.asp）に重要な更新プログラム、セキュリティ更新プログラム、およびサービスパックが掲載されたときに、それらをすべてダウンロードするためのサービスをインストールできる無料のツールである。これらの更新プログラムを承認すると、SUSは、Microsoft Windows Server 2003 またはWindows 2000 を実行しているすべての構成済みサーバに対して、また、Windows XP ProfessionalまたはWindows 2000 Professionalを実行しているデスクトップに対して、これらの更新プログラムを自動的に使用可能にする。SUSは、オペレーティングシステムやオペレーティングシステムに含まれるコンポーネントに適用される重要な更新プログラムとセキュリティ更新プログラム（サービスパックを含む）だけをサポートしている。アプリケーション用などの、それ以外のソフトウエア更新プログラムは、すべて異なるメカニズムを使って処理する必要がある。

SUS には次の機能が含まれている。

- ソフトウエア更新プログラムは、各 SUS サーバ上で独自に承認できるため、企業全体への段階的な展開や、隔離した環境でのテストが可能である。
- SUS を使用してソフトウエアの更新を配布できる（共有インターネット接続の帯域幅を節約できる）。また、Windows Update サイトからソフトウエア更新プログラムをダウンロードするように、SUS クライアントを構成できる。
- SUS は、インターネットにアクセスできないコンピュータに、Windows Update のソフトウエア更新プログラムを提供できる。
- SUS サーバのアーキテクチャは単純な親子関係から成り、各 SUS サーバが 15,000 クライアントまでサポートできるため、SUS は非常に大規模な環境に対応できる。
- CD を使用して、インターネットに接続された SUS サーバからインターネットにアクセスできない SUS サーバアーキテクチャへ、ソフトウエア更新プログラムをコピーできる。

SUS サーバは、SUS クライアントと通信するために Microsoft Windows Server 2003 オペレーティングシステムまたは Windows 2000 Server、Internet Information Services、およびポート 80 を必要とする。すべての SUS サーバは、ソフトウエア更新パッケージや承認を、親 SUS サーバから手動または自動で同期するように構成できる。これにより、環境の保守方法の柔軟性を実現できる。

SUS クライアントは自動更新クライアント（Windows Update でも使用されている）を使用する。クライアントは特定のサーバに接続するように構成され、ソフトウエア更新プログラムの自動インストールやエンドユーザへの確認を構成できる。

SUS を通じて提供された更新プログラムのインストールに失敗したコンピュータがあるかどうかを調べるために、定期的に MBSA（Microsoft Baseline Security Analyser）を実行する必要がある。MBSA は、欠けているセキュリティ更新プログラムがないかスキャンし、一般的なセキュリティ上のベストプラクティス（強固なパスワードなど）へのコンピュータの準拠の程度を報告し、そのコンピュータをセキュリティの脆弱性にさらす可能性のある構成オプションを識別する。また、SUS サーバ上ですでに承認されているがまだインストールされていない更新プログラムを報告するように、MBSA を構成することもできる。

6.1.4 WSUS（Windows Server Update Services）

WSUS は、Microsoft が最近提供するようになった製品であり、SUS より優れた機能を提供するとされている。次のような機能がある。

- 更新のメニューがより豊富である。
- 製品および種類ごとに更新プログラムを Microsoft Update から自動的にダウンロードできる。
- 世界中の顧客のための追加の言語サポート。
- BITS（Background Intelligent Transfer Service）2.0 を通じて帯域幅効率を最大化する。
- 更新プログラムの適用先として特定のコンピュータおよびコンピュータグループを特定できる。
- 更新プログラムが各コンピュータに適合することをインストール前に確認できる。
- より柔軟な展開オプション、レポート機能、およびデータベースオプション。
- データの移行機能とインポート/エクスポート機能。
- API（Application Programming Interface）を通じた拡張性。

Microsoft は、最終的に SUS を WSUS で置き換えるよう計画している。ただし、本書を執筆した時点では、Microsoft に SUS のサポートをやめる計画はない。

6.1.5 SMS（Systems Management Server）

SMS（Microsoft Systems Management Server）は、多数のクライアントへのソフトウエア更新プログラムの配布を展開し、管理するように設計されている。SMS は次のような機能を提供する。

- 展開されているコンピュータの台数を調べ、その場所および役割を識別するためのインベントリ機能。

- 展開されているコンピュータにインストールされているソフトウエアアプリケーションおよびソフトウエア更新プログラムと、インストールする必要があるソフトウエアアプリケーションおよびソフトウエア更新プログラムを識別するためのインベントリ機能。
- 通常の勤務時間外やビジネス運用への影響がもっとも少ない時間帯にソフトウエア更新プログラムを展開できるようにする、スケジューリング機能。
- 管理者がインストールの進捗状況を監視できるステータスレポート機能。

SMS 2003 のインベントリスキャンプログラムは、検出ロジックの自動化されたソースを使用して、各クライアントコンピュータに適用可能な更新プログラムとインストール済みの更新プログラムのインベントリを作成するために使用される。結果のデータは Systems Management Server インベントリに含められ、Web ベースのレポート機能を使用してステータスの包括的なビューが提供される。通常、インベントリデータは Microsoft がセキュリティ情報としてリリースする項目に制限される。

SMSの詳細は、http://www.microsoft.com/smserverに掲載されている。

7 付録B：その他のオペレーティングシステム用のパッチ管理ツール

7.1 Linux

Linux システムのパッチ管理を担当するシステム管理者は、パッチ管理に使用されるアプリケーションとユーティリティが Linux ディストリビューションごとに異なる点に注意する必要がある。

Red Hat Enterprise Linux、Fedora、SUSE、および Debian ディストリビューション用に、それぞれ異なるパッチ管理ツールが存在する。たとえば、up2date（Red Hat）、*apt*（Debian）、*yum*（Red Hat）、YaST online update（SUSE）、Zenworks Linux Management（Novell）などがある。したがって、管理者は、担当するディストリビューション用のパッチ管理ツールについてよく知っておく必要がある。

Michael Jang 著の『Linux Patch Management: Keeping Linux Systems Up To Date』[5]は、Linux パッチ管理のテクニックとツールについてよく知りたいと望むすべての Linux 管理者にとって、よい出発点となる。

7.1.1.1 Solaris

Solaris Patch Managerは、Sunが提供する [8]Solaris用の統合パッチ管理ツールである。Solaris Patch Managerには 2 つのバージョンがあるが、どちらも同じパッチ分析エンジンに基づいている。

8 （訳注）現在は Oracle 社から提供されている。
http://www.oracle.com/us/products/servers-storage/solaris/index.html

- **Solaris Patch Manager Base Version 1.0 for the Solaris 2.6, 7 and 8 Operating Environments**は、SunSolve Online Webサイトにおいて無料で提供されている。Base Version 1.0 にはコマンドラインインターフェイス（CLI）が付属しており、次のタスクを実行する。
    - システムに必要なパッチの判別
    - パッチの自動ダウンロード
    - 単純なパッチの自動インストール
    - パッチの依存関係の解決
    - インストールの順序の指定
    - パッチの削除

- **Solaris Patch Manager 1.0** は、Solaris 9 オペレーティングシステムにバンドルされている。Solaris Patch Manager 1.0 は、上記の Base Version 1.0 の機能に加えて、グラフィカルユーザインターフェイス（GUI）と以下の機能を備えている。
    - Solaris 9 Operating Environment を実行しているほかのシステムに対してリモートパッチ管理を実行する
    - Solaris 9 Operating Environment を実行している均一のシステムに、パッチまたはパッチのリストを自動的にインストールする
    - 最初のうち、Solaris Patch Manager が提供するのは、Solaris Operating Environment、ネットワークストレージ製品、Sun Cluster ソフトウエア、Sun Enterprise 10000 および 15000 サーバ、ならびに Sun Fire サーバ用のパッチだけである。やがて、その他の Sun 製品のパッチもサポートされるようになる。

8 付録C：ネットワークインフラストラクチャへのパッチ適用

ファイアウォールやルータなどのネットワークインフラストラクチャコンポーネントがますます複雑な装置になるにつれて、ネットワーク上の実際のコンピュータだけでなくそれらの装置にパッチを適用する必要性が増大している。ネットワークインフラストラクチャへのパッチ適用には、サーバ環境と比べて、集中化された脆弱性報告と自動化されたパッチ管理ツールが比較的未熟であるという課題がある。

インフラストラクチャコンポーネント用のパッチは、通常はベンダがオペレーティングシステム用のパッチと同じ方法で提供する。ネットワークインフラストラクチャ市場で支配的立場にある 2 つの企業、Cisco と Juniper は、それぞれ以下の各サイトでソフトウエアアップグレードを提供している。

- Cisco のソフトウエアダウンロードセンター：

  http://www.cisco.com/kobayashi/swcenter/

- Juniper のサポートセンター：

  http://www.juniper.net/support

どちらのサイトへのアクセスも、両社の登録済みの顧客だけに制限されている。

さらに、ネットワークインフラストラクチャのベンダは、パッチ管理に役立つ補助ツールを提供している場合がある。たとえば、Cisco の IOS Upgrade Planner は、ユーザが個々のインフラストラクチャコンポーネントに対する正しいソフトウエア更新プログラムを選択するのを助ける。

従来、インフラストラクチャコンポーネントを更新するには、ネットワーク管理者が SNMP（Simple Network Management Protocol）、SSH（Secure Shell）、または Telnet を使用して、ネットワークを通じて更新プログラムを手動で適切な装置へ送信する必要があった。しかし、インフラストラクチャベンダは、サーバ（特に Windows サーバ）環境で慣れ親しんだより自動的なパッチ管理へと移行しつつある。手動更新を実行するために SNMP、SSH、または Telnet を使用するシステム管理者は、Uniras やベンダの勧告に記載されているこれらのプロトコルの既知の脆弱性を知っておく必要がある。

9 参考資料

[1] “Creating a Patch and Vulnerability Management Program”, NIST Special Publication 800-40 (version 2.0), November 2005. See http://csrc.nist.gov/publications/nistpubs/

[2] Microsoft’s Security Guidance for Patch Management. http://technet.microsoft.com/ja-jp/library/dd433788.aspx

[3] HMG Infosec Standard No. 2, “Risk Management and Accreditation of Information Systems”, Issue 2.0, July 2005. Also available as a NISCC document at http://www.cpni.gov.uk/docs/re-20050804-00653.pdf

[4] “Essentials of Patch Management Policy and Practice”, Jason Chang, Jan 2004. See http://www.patchmanagement.org/pmessentials.asp

[5] Michael Jang, “Linux Patch Management:Keeping Linux Systems Up To Date”, January 2006. ISBN 0132366754.

[6] The Microsoft Threats and Countermeasures Guide. http://www.microsoft.com/downloads/details.aspx?familyid=1b6acf93-147a-4481-9346-f93a4081eea8&displaylang=en

[7] NISCC First Responders Guide:Policy and Principles, v1.2, October 2005. http://www.cpni.gov.uk/Docs/re-20051004-00868.pdf

[8] NISCC Technical Note 08/04, “Introduction to Vulnerability Assessment Tools”, October 2004. http://www.cpni.gov.uk/Docs/re-20041006-00750.pdf