標準から学ぶ **ICS** セキュリティ **#4**

セキュリティ更新（パッチ）管理

2023 年 7 月 13 日

一般社団法人 JPCERT コーディネーションセンター
制御システムセキュリティ対策グループ

1. はじめに

IEC 62443-2-3 では、アセットオーナーと ICS 製品提供事業者のそれぞれにおけるパッチ（セキュリティ更新）の管理が論じられています。初版は技術報告書（TR）として発行されましたが、現在検討中の第 2 版では国際標準への格上げが予定されています。国際標準になると、「当該標準に準拠している」と主張するためには必ず実施しなければならない要件（英語では助動詞「shall」を伴った記述）が標準の中に書き込まれます。また、本文の改訂と同時に、文書名も「IACS 環境におけるパッチ管理」（Patch Management in the IACS Environment）から「IACS 環境におけるセキュリティ更新（パッチ）管理」（Security Update （Patch） Management in the IACS Environment）に変更する方向で検討が進んでいます。

本分冊が登場する以前の 2010 年頃までは、ICS の利用現場では、問題なく動いているシステムに手を加えるのは邪道であり、パッチの適用で不具合を招くリスクの方が高いと考えられていました。ICS の提供者側も、汎用 OS 上で動作する SCADA 等のアプリケーション・プログラムに関して、「OS にパッチを適用した場合には動作を保証できない」などと言って、利用者に OS のパッチを適用することを思いとどまらせるのみで、パッチが出るたびに迅速に適用後の動作を検証することも稀でした。そのような風潮の中で、パッチの適用を含めたシステムの変更は、変更の副作用による不具合の発生を恐れて、よほどの必要性がない限りは検討されることがありませんでした。ところが 2010 年代に入ると、ICS 用製品が多数の脆弱性を内在させており、Stuxnet の事例から、ICS がマルウェアにより実際に攻撃されることが広く知られるところとなり、ICS 環境においても脆弱性管理の重要性が注目されるようになりました。このような時代背景の 2015 年に IEC 62443-2-3 の初版が登場しました。

実務的なセキュリティ対策において脆弱性への対処は非常に重要です。それにもかかわらず、IT 分野でも脆弱性管理を真正面から論じられた標準が今日に至るまでありませんでした。そのような状況の中で IEC 62443-2-3 の「ICS のためのパッチ管理」は最初かつ唯一のものと言えるかと思います。

なお、パッチは「補修用のつぎはぎ」が原義で、ソフトウェアの一部を上書きすることにより応急的な修正を施す手法です。機能追加や性能改善を目的としたパッチもあり、必ずしも脆弱性の修正を目的とはしていません。また、ソフトウェア等の脆弱性を修正する方法は、改版いわゆるバージョンアップによることもあり、パッチに限りません。IEC 62443-2-3 は、セキュリティ・パッチを論じると書かれていますが、稼働中のソフトウェア等を多様な目的で更新する活動に広く適用できそうです。

2. パッチの状態のライフサイクル

IEC 62443-2-3 では、個々のパッチについて、作成されてから実運用システムに適用されて稼働するまでのライフサイクルを下図に示したような状態遷移によりモデル化しています。こうした状態遷移は 2015 年版でも示唆されてはいましたが、改定に伴って包括的な状態遷移図として整理されました。下図は改定案に採用されてい

る図をベースに邦訳し注を加えたものです。IEC 62443-2-3 では、アセットオーナーと ICS 製品提供事業者のそれぞれにおけるパッチ管理が論じられています。パッチの状態遷移も、ICS 製品提供事業者における対応活動に伴う状態遷移（図の灰色で示した部分）と、アセットオーナーにおける対応活動に伴う状態遷移（図の水色で示した部分）とから構成されています。

ICS 製品提供事業者においては、ICS 用ソフトウェアの稼働環境を構成する、他社製の OS などのパッチが提供されるケース（例えば HMI ソフトウェアが稼働している Windows OS のパッチ）と、自社が提供している ICS 用ソフトウェアあるいはファームウェア自身のパッチが開発されるケースとがあり得ます。前者の場合には図の中の A が初期状態となり、後者の場合には B が初期状態になると考えられます。

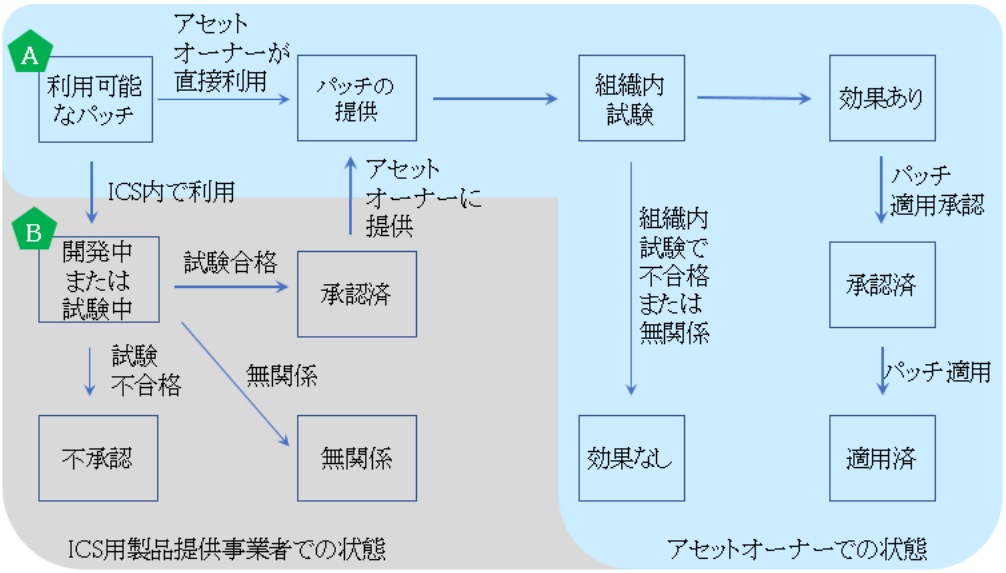

図 パッチの状態のライフサイクル・モデル
（改定案の図 1 をベースに JPCERT/CC で邦訳し注を加えて作成）

ICS 製品提供事業者は、他の製品提供事業者などからパッチが提供されると、まずはパッチの内容を吟味またはパッチを試験環境に適用してみて、パッチの効果を調べます。パッチの適用により不具合が生じる場合には「不承認」、パッチを適用しても ICS 環境では効果がない場合には「無関係」、パッチの適用により不具合が改善され他の不具合が生じないことが確認されれば「承認済」とします。

アセットオーナーにおいては、パッチが提供されると試験環境を利用するなどして組織内試験により、パッチの効果の有無を調べます。パッチの適用により不具合が解消されるなどの効果があり、副作用による不具合のないことが確認されれば「承認済」とします。パッチの適用により不具合が生じる、あるいは、パッチで解消されるべき不具合がそのまま残っている場合には「効果なし」とします。

3. アセットオーナーにおけるパッチ管理

2015 年版でも、アセットオーナーにおけるパッチ管理プロセスの概念を提示していましたが、詳細には踏み込まず、概念的な記述にとどまっていました。検討中の改定案においては、パッチ管理に取り組むための準備段階として、1） 基本方針や体制を骨子とするパッチ管理プログラムの制定や、2） パッチの入手から適用に至る作業手順の策定、3） パッチを適用する対象システムの構成情報の棚卸、が推奨要件とされています。その上で、パッチが提供されるたびに、4） パッチに関する記録、5） パッチの特性の見極め、6） 真正性と完全性の確認、7） パッチ適用試験、8） 実運用システムへのパッチ適用スケジュールの作成、9） 自組織内の必要な部署へのパッチの配給、を実施することを要件としています。

なお、目標としてのセキュリティ水準が 2 以上に設定された ICS の中でパッチ対象製品が使われている場合には、より厳格な実施要件が設定されています。ちなみに「セキュリティ水準」については本シリーズ「標準から学ぶ ICS セキュリティ」の第 3 回で紹介していますので必要に応じてご参照ください。

4. ICS 製品提供事業者におけるパッチ管理

ICS 製品の提供事業者には、顧客サービスの一環として、自社が提供する ICS 製品で見つかった脆弱性を修正するためのパッチを提供することと、自社が提供する ICS 製品の稼働環境を構成している製品（例えば OS）にパッチが提供された場合に、それを適用しても不具合が生じないことを確認し確認結果を提供することを ICS 利用者は期待しています。こうした期待に応えて ICS 製品の提供事業者はパッチに関するポリシーを策定してお

くべきです。さらに、顧客との認識の齟齬を避けるために、ポリシーの一部は顧客にも開示しておきます。さらに、パッチの提供状況または提供予定や、製品ごとに提供されているパッチの一覧、サポート期間が終了しパッチが提供されなくなった製品などの情報をタイムリーに顧客に開示することも求める方向で改定案の検討が進んでいます。

脆弱性が見つかった場合にパッチなどの対策方法を開発することや、提供前に実施すべき試験によるパッチの品質保証、また、第三者によるパッチの改ざんなどを防ぐセキュアな方法によるパッチの配付は製品提供事業者の最も重要な責務と言えましょう。

また、OS をはじめとする第三者が提供するソフトウェアの個々のパッチに関して、ICS 製品の側で不具合を起こすことがないことを確認し、その結果を顧客に伝達することも IEC 62443-2-3 では期待しています。

なお、目標としてセキュリティ水準が 3 以上に設定された ICS の中の利用が想定される製品に対しては、より厳格な要件が設定されることになりそうです。

5. パッチと脆弱性管理

以上で IEC 62443-2-3 が定義したパッチ管理の概略を紹介しました。セキュリティ・パッチが作られる理由は、製品の出荷後に見つかる脆弱性があり、そうした脆弱性を悪用したサイバー攻撃を防ぐために、製品の改版よりも小さなコストで脆弱性を除去しておきたいという需要があるからです。この課題を IEC 62443-2-3 では「パッチ管理」として掲げましたが、「脆弱性管理」として論じられる方が一般的かと思います。

オフィス用情報システムでは、アプリケーションの多様性が大きいこともあり、提供されたパッチを速やかに適用するなど、報告された脆弱性に対して広範囲に対処が行われます。しかしながら、ICS では動作が事前に定まった範囲に長期間にわたり限られることが多いために、報告された脆弱性を個々に調べて悪用できるための条件や悪用による影響を吟味すれば、対策なしで放置しても大きな問題になる可能性がなかったり、当該製品や周囲の製品の設定を若干変更することにより脆弱性を悪用されるリスクを回避することができて、パッチの適用を延期することができる場合もあります。逆に、そのまま ICS を稼働させると、プラントの安全性や製造品の品質に大きな不安を招くようなタイプの脆弱性もあり得ます。こうした事例からもアセットオーナーにおける ICS の脆弱性管理の重要性を認識しておきたいものです。

さらに、サプライチェーンに関連して注目される 2 つの脆弱性問題を紹介しておきます。

一つは、パッチや改版パッケージがサプライチェーンを経由して利用者に届くまでの途上で改ざんされ、マルウェアとしてのコードを組み込まれることがあるという課題です。これは「サプライチェーン攻撃」とも呼ばれています。パッチや改版パッケージを作成する環境が汚染されていて、作成時にマルウェアが組み込まれる高度な攻撃もあれば、まったくの第三者が「緊急パッチ」などと偽ってマルウェアを送り付ける単純な詐欺的攻撃もあります。

もう一つは、他の製品を組み込んで、別の製品やサービスが作られている場合に、組み込まれた製品の脆弱性がサプライチェーンをたどって、下流の製品やサービスに継承されて出現する問題です。ICS の世界においても、例えば CODESYS 社製のモジュールを組み込んで作られた PLC が、工作機械に組み込まれてプラントに設置されるような事例がしばしば見られます。こうした状況で、CODESYS 社製のモジュールで脆弱性が見つかり、パッチあるいは改版パッケージが提供されたとしましょう。脆弱性は PLC や工作機械に継承されている可能性があり、そのことを PLC や工作機械の利用者が認知でき、さらには必要なパッチなどの提供を確実に受け取れることが理想ですが、それが非常に危ういのが実情と言わざるを得ません。危うさはサプライチェーンが長くなれば一層高まります。このような継承された脆弱性に攻撃者側は比較的気付きやすいのですが、利用者や下流の製品の提供者は知らずに使い続ける、あるいは出荷し続けることがしばしばです。その結果、脆弱性が長期間存在することになり、そうした脆弱性は「ゼロデイ脆弱性」との対比で「N ディ脆弱性」とも呼ばれます。こうした状況を解消するために、製品内に組み込まれているソフトウェア・コンポーネントを示す情報 SBOM（ソフトウェア成分表：Software Bill of Materials）を開示することを義務付ける、あるいは継承された脆弱性を効率的に認知するための SBOM 利用技術の開発などの対策も進み始めています。

また日本国内では、ICS を含む各種製品の脆弱性に関する情報の取り扱いについて、脆弱性の発見者から製品の開発者までを含めた体制が 2004 年に整備され運用されています。この体制については JPCERT/CC も調整機関としてその一端を担っています。詳細については、平成 29 年経済産業省告示 19 号「ソフトウエア製品等の脆

弱性関連情報に関する取扱規程」[1]や「情報セキュリティ早期警戒パートナーシップガイドライン」[2]、脆弱性情報ポータル・サイト「JVN （Japan Vulnerability Notes）」[3]などを参照ください。また、関連する国際標準としては ISO/IEC 29147 「脆弱性情報開示」[4]や ISO/IEC 30111 「脆弱性取扱手順」[5]も発行されています。

6. まとめ

今回は IEC 62443-2-3 で論じられているパッチ管理の概略と、それに関連して、他の国際標準や脆弱性に関する最近の課題を紹介しました。脆弱性の問題は、ソフトウェア（ファームウェアを含む）を使う限り存在し続ける課題であり、製品やシステムの提供者と利用者が連携しつつ、それぞれの守備範囲で必要な対策にあたっていくことが求められています。

参考文献

[1] 平成 29 年経済産業省告示 19 号「ソフトウエア製品等の脆弱性関連情報に関する取扱規程」
http://www.meti.go.jp/policy/netsecurity/vul_notification.pdf

[2] 情報セキュリティ早期警戒パートナーシップガイドライン
https://www.ipa.go.jp/security/todokede/vuln/ug65p90000019gda-att/000098799.pdf

[3] 脆弱性対策情報ポータルサイト JVN （Japan Vulnerability Notes） https://jvn.jp/

[4] ISO/IEC 29147:2018 Information technology — Security techniques — Vulnerability disclosure

[5] ISO/IEC 30111:2019 Information technology — Security techniques — Vulnerability handling processes