IPA Technical Watch

サーバソフトウェアが最新版に更新されにくい現状および対策

～ 中長期的な視点からウェブサイトの組織的な管理を ～

2014 年 4 月

独立行政法人情報処理推進機構
技術本部 セキュリティセンター

目次

はじめに（Executive Summary）

ウェブサイトは、事業に欠く事のできない環境となっており、中には売上げに直結しているウェブサイトもある。しかし昨今、情報漏えいやウェブサイトの改ざん、ウイルス感染など社会的信頼を損なうような事故のニュースが後をたたない[1]。ウェブサイトは 24 時間 365 日、悪者からの無差別な攻撃に晒されており、セキュリティ上の不備があれば被害に繋がる。

セキュリティ上、注意すべき対象の一つとして、ウェブサイトで稼動しているサーバソフトウェアがある。2013 年は販売サイト[2]、保養所予約システム[3]などで Apache Struts 2 というサーバソフトウェアを狙った攻撃があり、個人情報が漏洩したなどと報じられた。また、IPA にも類似の届出[4]が寄せられた。

ウェブサイトは、開発時点においては最新のソフトウェアを用いる為、安全であるが、時間とともにソフトウェアが古くなる。それを放っておく事により上記事故が発生する事はご存知だろうか。パソコンにおいて日々、ソフトウェアの更新（例：Windows Update）が必要なのと同じように、サーバで稼動するソフトウェアも更新しなくてはならない。

今回 IPA では、どのくらい古いソフトウェアがウェブサイトで使われているか調査した。4,178 件のウェブサイトを調査したところ、PHP というサーバソフトウェアでは実に 80 パーセントが、古いままの運用が続けられている兆候を確認できた。

とは言え、サーバソフトウェアを更新するのは、必ずしも簡単ではない。場合によっては、ウェブサイトで稼動しているシステムを作り直さなくてはならない場合もあり、一朝一夕にはできない。このためウェブサイトを安全に保ち続けるには、長期的な視点に基づく組織的な管理が欠かせない。組織が人材や資本を管理しているように、ウェブサイトに対しても管理が必要だ。

本テクニカルウォッチでは、

- ウェブサイトのサーバソフトウェアのバージョンおよびパッチ管理の実態
- 長期的な安全確保のためのウェブサイト管理のあり方
- ウェブサイトの実運用例（IPA のケース）

を解説している。ウェブサイトのサーバソフトウェアが古い実態を踏まえ、長期的な安全確保のために考慮する点を挙げた。ウェブサイトを持つすべての組織で、管理の一助とし、ブランドや売上げを毀損しない円滑なウェブサイト運営、ひいては事業発展に役立ててもらいたい。

---

[1] ウェブサイト改ざん等のインシデントに対する注意喚起～ウェブサイト改ざんが急激に増えています～
https://www.ipa.go.jp/security/topics/alert20130906.html

[2] 「Struts 2 の脆弱性を突いて不正侵入」、JINS 通販サイトのカード情報漏洩
http://itpro.nikkeibp.co.jp/article/NEWS/20130501/474536/

[3] 富士重工業健保の保養所予約システムに不正アクセス - Apache Struts の脆弱性に攻撃
http://www.security-next.com/041251

[4] コンピュータウイルス・不正アクセス届出状況および相談受付状況［2013 年第 3 四半期（7 月～9 月）］
https://www.ipa.go.jp/security/txt/2013/q3outline.html

1. サーバソフトウェアのバージョン調査

情報セキュリティ早期警戒パートナーシップ[5]に基づき、IPA はウェブアプリケーションの脆弱性に関する届出を受付けている。届出を受けた際、IPA は届出内容を確認するため対象ウェブサイトを簡単に調査しており、その過程で得られた、ウェブサイトで稼動するサーバソフトウェア[6]の名称やバージョン等を記録している。この情報は通常、ウェブサイトが自動的に送信するものであり、ウェブサイトにアクセスした際には端末が自動的に受け取っている。

ウェブアプリケーションの脆弱性が指摘されるウェブサイトの場合、サーバソフトウェアのバージョンが古いままとなっていることがある。今回、記録した情報に基づいて、ウェブサイトで稼動するサーバソフトウェアがどの程度古いかの調査を実施した。

1.1. 情報セキュリティ早期警戒パートナーシップ

本テクニカルウォッチが調査の対象とするウェブサイトは、情報セキュリティ早期警戒パートナーシップの中で IPA が届出を受けたものである。

政府や IT 業界、セキュリティ機関等が我が国の情報セキュリティ確保のために協力する形で実現した情報セキュリティ早期警戒パートナーシップは、ソフトウェアの脆弱性という問題に対処する官民連携の枠組みとして機能してきた。2004 年 7 月の運用開始から 2013 年 12 月末までにソフトウェア製品及びウェブアプリケーションの脆弱性に関する届出は累計で 9,333 件に達している[7]。

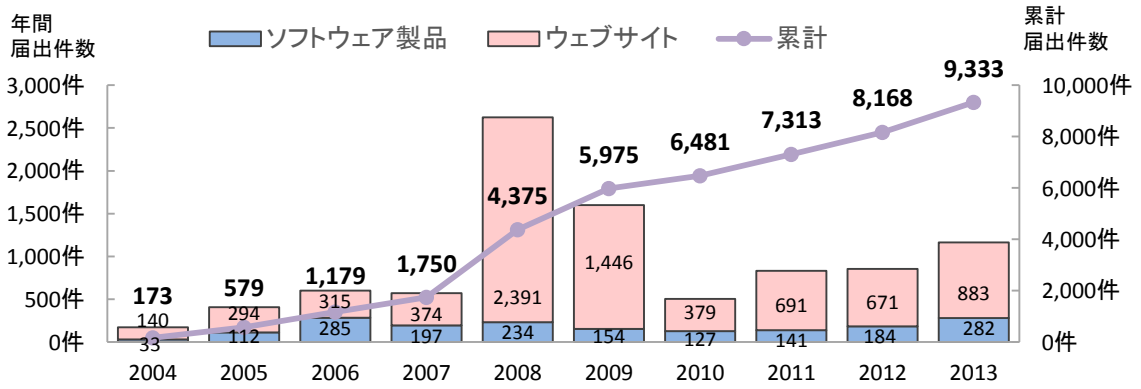


図 1：脆弱性関連情報の届出件数の年別推移

[5] IPA 独立行政法人 情報処理推進機構：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/ciadr/partnership_guide.html

[6] 本テクニカルウォッチが対象とするサーバソフトウェアは下記：
ウェブサーバ：Apache HTTP Server、IIS、nginx 等
プログラミング言語や、その処理系：Java、C#、PHP、Perl、Ruby 等
ウェブアプリケーションサーバ：Apache Tomcat、WebSphere Application Server 等
ウェブアプリケーションフレームワーク：Struts、ASP.NET、FuelPHP、Mojolicious、Ruby on Rails 等

[7] ソフトウェア等の脆弱性関連情報に関する届出状況［2013 年第 4 四半期（10 月～12 月）］
https://www.ipa.go.jp/security/vn/report/vuln2013q4.html

このうちウェブアプリケーションの脆弱性に関する届出は 7,584 件（81%）であり、脆弱性の種類別では、クロスサイト・スクリプティングが 55% と最も多い。脅威別では、クロスサイト・スクリプティングの影響として典型的な、本物サイトへの偽情報の表示がやはり多くなっている。

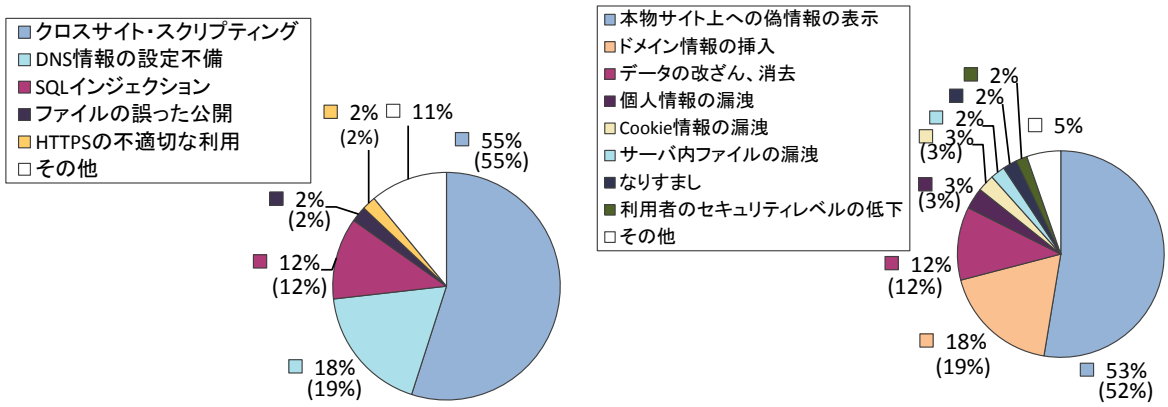


図 2：ウェブサイトの脆弱性の種類別（左）および脅威別（右）届出状況

以上のような脆弱性の問題がある一方で、サーバソフトウェアのバージョン管理も課題である。本章では、届出対象となったウェブサイトで稼動しているサーバソフトウェアについて、そのバージョン管理の状況をまとめた。

1.2. ウェブサイトが出力する情報

まず、ウェブサイトはどのような情報を出力するのかを見ていこう。ウェブサイトはウェブブラウザからのアクセスを受けた際、ウェブブラウザからのリクエストに応じてウェブページを出力する。その際にレスポンスヘッダと呼ばれる情報を付加している。以下は、あるウェブサイトが出力するレスポンスヘッダである。

```
Date: Thu, 27 Feb 2014 06:30:24 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Connection: close
```

図 3：ウェブサイトが出力するレスポンスヘッダの例

レスポンスヘッダには様々な情報が含まれており、必要に応じてウェブブラウザの内部処理等に利用される。このうち、「`Server: Apache/2.2.15 (CentOS)` 」という記載から、このウェブサイトではウェブサーバソフトウェア Apache HTTP Server のバージョン 2.2.15 が稼動していることがわかる。さらに、「`X-Powered-By: PHP/5.3.3`」という記載から、スクリプト言語処理系 PHP のバージョン 5.3.3 が動作していることがわかる。

このように、レスポンスヘッダから、サーバソフトウェアのバージョンが判ることがある。ただし、サーバソフトウェアの中にはレスポンスヘッダに情報を出さないものもあり、また、情報を出すかどうかは設定等により制御できることから、全てのサーバソフトウェアについて情報を得られるわけではない。

1.3. 調査対象サーバソフトウェア

■ 調査対象

調査対象のデータは、IPA に届け出られたウェブアプリケーションの脆弱性のうち、レスポンスヘッダを記録した 4,178 件（対象期間：2008 年 2 月～2013 年 9 月）である。

調査対象のサーバソフトウェアは、以下の 3 種類とした。

- Apache HTTP Server（Apache Software Foundation によるウェブサーバソフトウェア）
- IIS（Microsoft によるウェブサーバソフトウェア）
- PHP（The PHP Group によるスクリプト言語処理系）

選定理由は、これらのサーバソフトウェアはレスポンスヘッダの中に情報が出力されやすく、外部からの調査に適しているためである。

■ バージョン番号の構造

今回集計対象としたサーバソフトウェアはいずれも、3 種類に分かれたバージョン番号の構造を持っている。3 種類のバージョン番号はドットで区切られて記載され、先頭から 1 つ目と 2 つ目をそれぞれ「メジャーバージョン」「マイナーバージョン」と呼ぶ。3 つ目の呼称はソフトウェアによって異なり、Apache HTTP Server では「パッチバージョン[8]」、PHP では「ポイントバージョン[9]」と呼ぶ。以下は、Apache HTTP Server のバージョン番号の構造である。

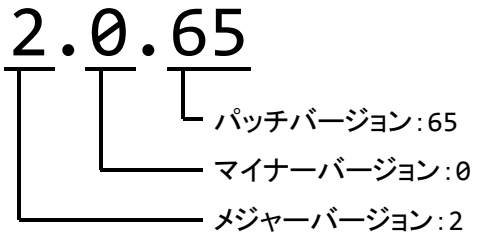


図 4：バージョン番号の構造（Apache HTTP Server の場合）

[8] APR's Version Numbering
https://apr.apache.org/versioning.html
[9] PHP のバージョン
https://php.net/manual/ja/about.phpversions.php

1.4. サポート有無に関する集計

■ 集計の方針

どのようなソフトウェアも、開発元によるサポートが終了する時期がある。調査対象から抽出した各ソフトウェアについて、調査時点で開発元によるサポートが提供されているバージョンかどうかを、「サポート範囲内」「サポート終了後 1 年未満」「サポート終了後 1 年以上」の 3 種類に区分けし、区分けした結果を集計した。下図は PHP を例に、各マイナーバージョンのサポート終了時期に基づき、各届出に対してサポート有無を区分けする方法を示したものである。同じバージョンの PHP が稼動していたとしても、調査時期により異なる区分けとなる。

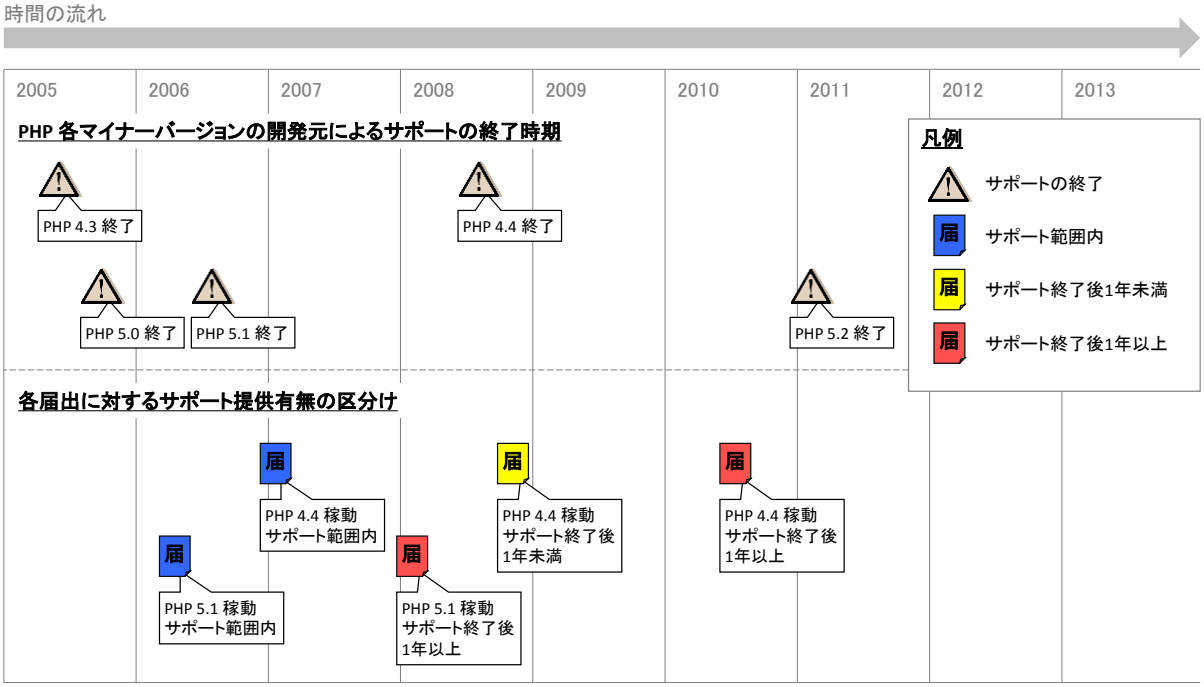


図 5：開発元のサポート終了と届出に対するサポート有無区分け（PHP）

サポート期間内かどうかの区分けは、各ソフトウェアのマイナーバージョンに基づいている。マイナーバージョンよりも細かいバージョン、たとえば Apache におけるパッチバージョンや、PHP におけるポイントバージョンは、パッチ適用有無の問題として次節で考慮する。

■ 集計結果

前項の方針でサポート有無を区分けした結果の集計は以下。

（ウェブサイト数）

	サポート範囲内	サポート終了後 1 年以内	サポート終了後 1 年以上	合計
PHP	285 (20%)	215 (15%)	915 (65%)	1,415
Apache HTTP Server	1,450 (85%)	40 (2%)	222 (13%)	1,712
IIS	496 (94%)	4 (1%)	25 (5%)	525

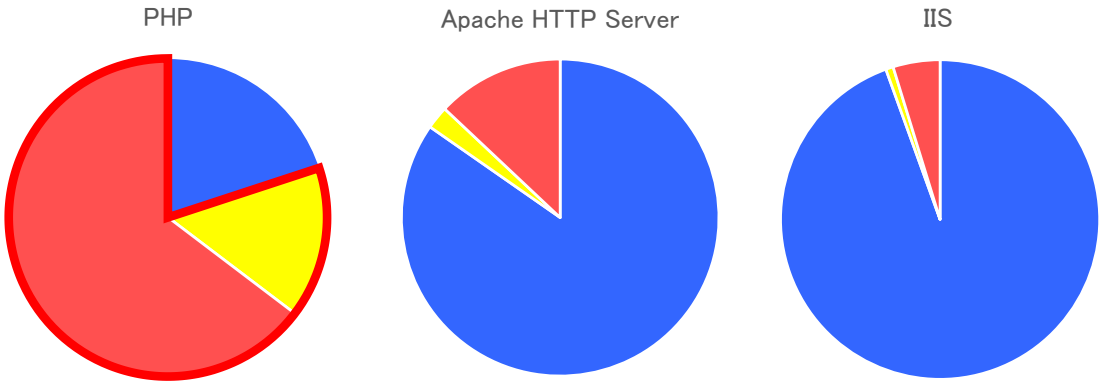


図 6：サポート有無に関する集計結果

集計結果からは、Apache および IIS についてはサポート範囲内が多い（85% 以上）一方、PHP についてはサポート範囲内が著しく少なく、実に 80% がサポート終了後であることがわかる。ただし、この結果を解釈する際には、以下の留意事項がある：

実態はより安全な可能性を示唆する要素

1. 外部からのバージョン番号取得
   サポート期間内かどうかの区分けは、外部から取得できるレスポンスヘッダと、開発元によるサポートの終了に基づいているため、本集計でサポート終了となったものの中には、実際にはサポートされている例が存在する可能性がある。たとえば、サーバ OS によっては、開発元のサポートとは別に独自のサポートを実施する一方、レスポンスヘッダに記載のバージョン番号は更新しないことがある。
2. 母集団の偏り
   集計対象は、ウェブアプリケーションの脆弱性の届出がなされたウェブサイトであるため、一般的なウェブサイトの傾向を表したものではない。

1.5. パッチ適用有無に関する集計

■ 集計の方針

集計対象から抽出した各ソフトウェアのうち、マイナーバージョンよりも細かいバージョンをレスポンスヘッダから得られる以下 2 種類のソフトウェアについて、パッチ適用有無を区分けし、区分け結果をマイナーバージョンごとに集計した。

- Apache HTTP Server（バージョン：1.3 / 2.0 / 2.2 / 2.4 系）
- PHP（バージョン：4.0 / 4.1 / 4.2 / 4.3 / 4.4 / 5.0 / 5.1 / 5.2 / 5.3 / 5.4 / 5.5 系）

下図は Apache HTTP Server の 2.2 系を例に、開発元が最新パッチを公開した時期と、各届出に対してパッチ適用有無を区分けする方法を示したものである。同じバージョンの Apache HTTP Server が稼動していたとしても、届出時期により異なる区分けとなる。また、最新パッチへの適用が僅かでも遅れた場合には、最新でないと区分けされる。

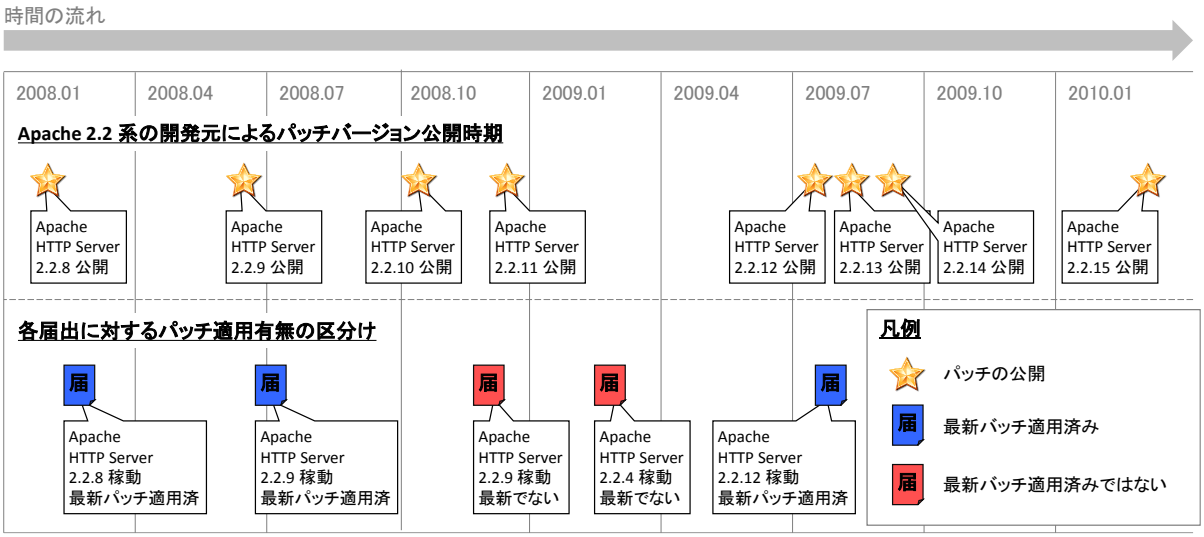
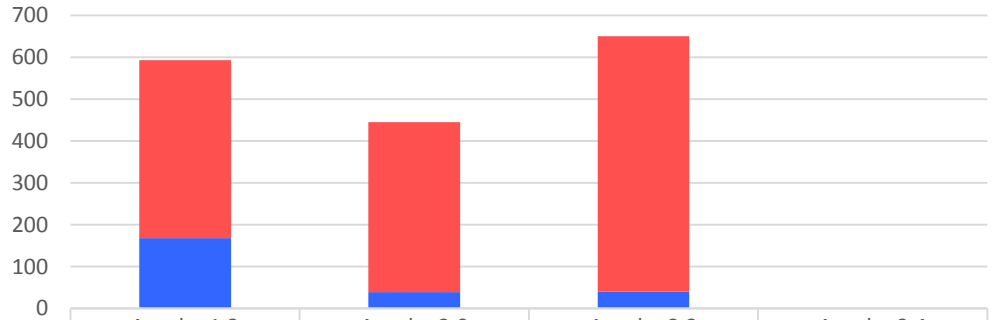


図 7：開発元パッチ公開と届出時点パッチ適用有無の区分け（Apache HTTP Server 2.2 系）

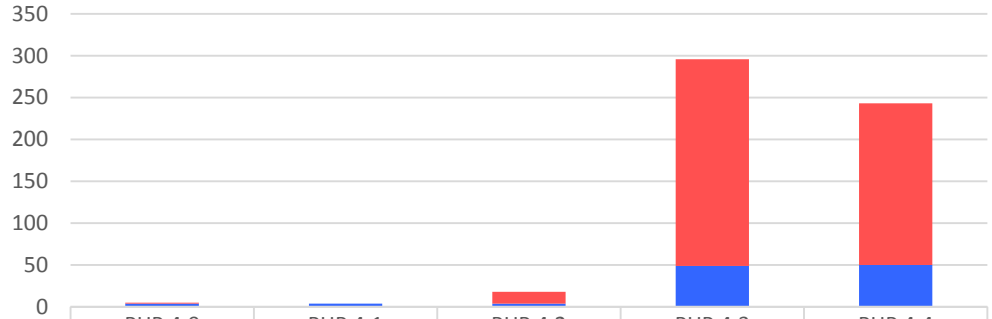
■ 集計結果

前項の方針でパッチ適用有無を区分けした結果の集計は以下。



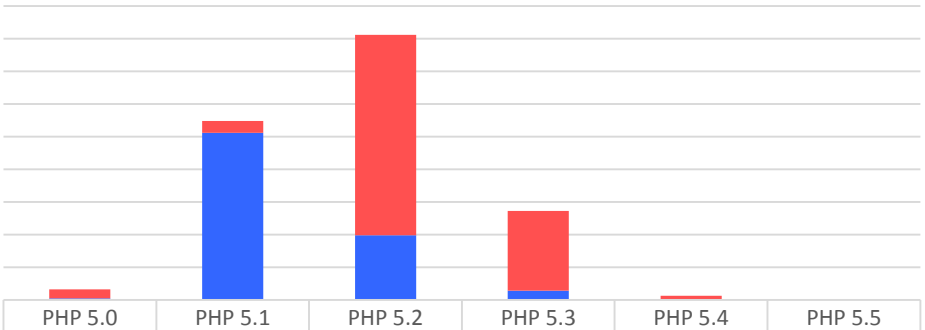
	Apache 1.3	Apache 2.0	Apache 2.2	Apache 2.4
最新でない	425	406	609	0
最新パッチ適用済	168	39	41	1

図 8：パッチ適用有無の集計結果（Apache HTTP Server）



	PHP 4.0	PHP 4.1	PHP 4.2	PHP 4.3	PHP 4.4
最新でない	1	0	14	247	193
最新パッチ適用済	4	4	4	49	50

図 9：パッチ適用有無の集計結果（PHP 4）



	PHP 5.0	PHP 5.1	PHP 5.2	PHP 5.3	PHP 5.4	PHP 5.5
最新でない	14	18	307	122	5	0
最新パッチ適用済	2	256	99	14	1	0

図 10：パッチ適用有無の集計結果（PHP 5）

集計結果からは、全体として最新でない（パッチ適用されていない）場合が多いことが判る。この結果を解釈する際には、以下の留意事項がある：

実態はより危険な可能性を示唆する要素

1. **サポートが終了した最新版**
   開発元によるサポートが提供されているかどうかとは、無関係に集計している。このため、サポートが終了したバージョンの最新版を使い続けているようなサイトは、本区分けでは最新パッチ適用済みとなる。中でも、図 10 に示す PHP 5.1 は最新パッチ適用済みが際立って多いが、今回の調査対象が 2008 年以降の届出であるのに対し、PHP 5.1 のサポート終了が 2006 年であることから、サポートが終了した最新版が多く含まれている可能性が示唆される。

実態はより安全な可能性を示唆する要素

2. **外部からのバージョン番号取得**
   ソフトウェアバージョンの取得は、外部から確認できるレスポンスヘッダに基づいているため、本集計で最新でないとなった場合でも、実際には最新パッチが適用されている例がある可能性がある。たとえば、サーバ OS によっては、開発元とは別に独自のパッチを提供する一方、レスポンスヘッダに記載のバージョン番号は更新しないことがある。
3. **セキュリティ問題ではないパッチ**
   提供されるパッチの中には、脆弱性の修正が含まれないものもある（機能アップなど）。このようなパッチは、少なくともセキュリティ上の理由からは適用する必要性がないが、集計上は区別しておらず、最新でないものとしてカウントしている。
4. **母集団の偏り**
   集計対象は、ウェブアプリケーションの脆弱性の届出がなされたウェブサイトであるため、一般的なウェブサイトの傾向を表したものではない。

1.6. 考察

サポート有無およびパッチ適用有無に関する集計結果から、以下を考察できる。

■ サポートが終了した PHP を使い続けるウェブサイトが多い

PHP を稼動させているウェブサイトのうち、実に 80 パーセントが、サポートを終了したバージョンの PHP を使っていた。このような古い PHP が稼動している理由の一つには、更新の難しさがあると考えられる。

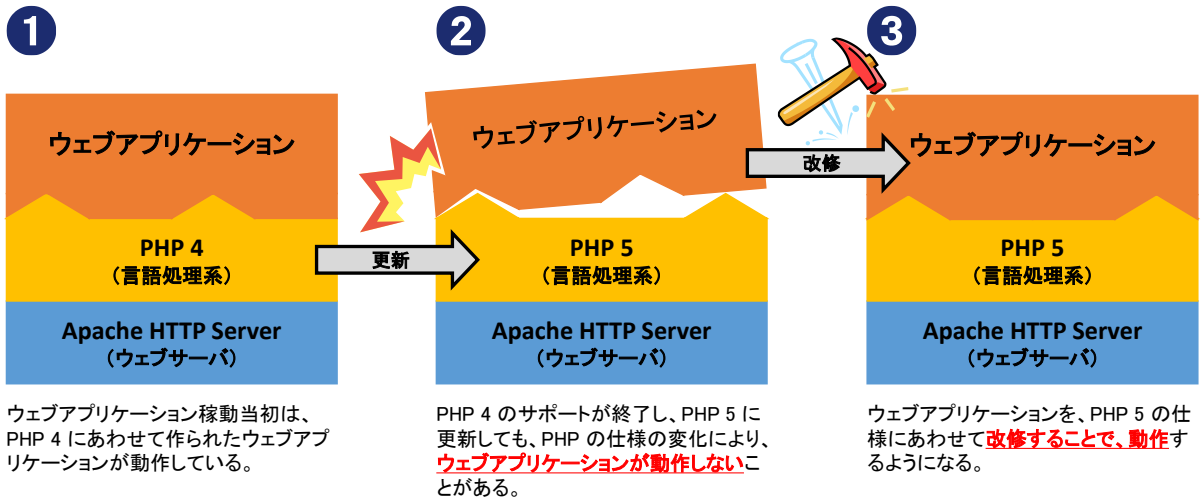


図 11：PHP 更新に伴いウェブアプリケーション改修が必要となるイメージ

PHP はウェブアプリケーションを稼動させるための言語処理系だが、バージョンアップに伴い仕様が変化することがある。このため PHP のみをバージョンアップした場合、稼動していたウェブアプリケーションが動かなくなることも少なくない（図 11 の①→②）。その場合、ウェブアプリケーションを改修する必要がある（図 11 の②→③）が、そのためにはウェブアプリケーション開発時点のノウハウが残っていなくてはならない。開発時点での資料整備等が不十分であったり、開発を担当した会社との契約が切れていたり、担当者の退職などの理由で、ノウハウが失われているケースでは、ほとんど作り直しになる場合もある。しかし現実には、作り直す余裕があるウェブサイトは少なく、危険性を認識したとしても使い続けている可能性がある。

PHP は、2012 年に深刻な被害に繋がる脆弱性[10]が公開され、攻撃に悪用されたとの情報[11]もある。PHP のサポートが終了した状態での運用は安全ではない可能性があり、サポートが存在

[10]PHP-CGI の query string の処理に脆弱性
http://jvndb.jvn.jp/jvndb/JVNDB-2012-002235
[11]PHP-CGI Vulnerability Exploited in the Wild
http://blog.sucuri.net/2012/05/php-cgi-vulnerability-exploited-in-the-wild.html

するバージョンへの更新を検討する必要がある。すぐに更新が難しい場合でも、長期的な計画を立てて対応したい。

なお、今回 PHP について思わしくない実態となったが、これは PHP だけの問題ではない。他のプログラミング言語やウェブアプリケーションサーバ、ウェブアプリケーションフレームワーク等の中にも、程度の差はあれ同様の事情から更新が難しいものは少なくなく、長期的な計画を立てることの重要性は変わらない。

■ Apache より IIS の方がサポート内の傾向がある

集計対象としたウェブサーバソフトウェア 2 種類の比較では、Apache HTTP Server と比べて IIS の方がサポート内のバージョンが使われている傾向が確認できる。その理由は明らかではないが、有償製品である IIS は無償の Apache と比べて費用がかかるため、ウェブサイト構築に潤沢な予算が投入される前提で採用されており、それが今回の結果に繋がった可能性を推測することはできる。

Apache HTTP Server は、2011 年にはサービスの停止に繋がる脆弱性[12]が公開され、それを悪用するツールが出回った。Apache HTTP Server も PHP と同様、サポートが終了した状態での運用は安全ではない可能性があり、サポートが存在するバージョンへの更新を検討する必要がある。

■ 最新パッチが適用されていない

最新パッチを適用していたウェブサイトは、Apache HTTP Server については 15%、PHP については 34% であり、ともに半分にも満たず少数に留まった。

外部から確認できる情報のみを集計した結果であり、留意事項も多いため、この数値のみから断定的に結論付けることはできないが、最新のセキュリティパッチの適用が十分にはなされていない実態が明らかになった。この原因として、パッチを適用する必要性を認識していないか、認識していても何らかの要因でパッチの適用が困難な可能性を考えることができる。

以降の 2 章では、以上の実態を受け、ウェブサイトの望ましい管理方法について解説する。

[12]Apache HTTPD サーバにサービス運用妨害（DoS）の脆弱性
http://jvndb.jvn.jp/jvndb/JVNDB-2011-002172

コラム「IPA と運営者のやり取り」事例 1

IPA では「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」に則り、7,500 件以上のウェブサイトにおける脆弱性関連情報の届出を受付、ウェブサイト運営者へ通知し対策を促してきた。この中で、対策を依頼するにあたり、実際にウェブサイト管理者がウェブサイトの運用／管理において問題となった事例を紹介する。

■ そもそも運用／管理が考慮されていない事例

IPA が届出を受付けてきたウェブサイトの運営者には、行政機関、企業、学術機関、団体などの組織および、個人がある。なかでも、非上場企業のウェブサイトに対する届出が多いが、これらの企業のウェブサイト管理者は、専任ではなく、多少 PC の操作に詳しい者が他業務と兼任し、片手間で対応しているという場合が多い。ここでは、この様な組織における事例を以下に記載する。

IPA から企業のウェブサイト管理者に「貴社ウェブサイトにおけるセキュリティ上の問題についてご連絡したい」と切り出すと、「ウェブサイト製作の委託業者(専門業者)に開発を依頼しているから、セキュリティは完璧です」と断言するウェブサイト管理者も少なくない。この様なウェブサイト管理者においては、「ウェブサイトのコンテンツの更新は行えるが、それ以上の事は分からない、対応できない」といった、ウェブサイトの運用・管理についてセキュリティが考慮されていないと考えられる。

また、委託業者（専門業者）においては、ウェブサイト運営者が利用しているサービス（委託業者が提供）に脆弱性が存在していた場合または、委託業者自身のウェブサイトに脆弱性が存在している場合や、IPA から連絡をすると、「ウェブサイトの製作は出来るが、脆弱性対策については詳しくない」という事例もある。

ウェブサイトは、開発・製作した時点では、安全性（セキュリティ）が保たれていたとしても、時間の経過と共にウェブサイトの安全性（セキュリティ）が劣化する可能性があることが認識されていないため、管理者がウェブサイトの運用／管理を適切に考慮していないという実情が覗える。

このようなウェブサイト管理者の場合は、最終的に委託業者へ対策を依頼したくても保守契約が締結されていない、または、計画外の作業費用が捻出できず対策が実施できない事態に陥ってしまうことがある。

上記以外の事例として、企業のウェブサイトであっても「重要な情報（個人情報やクレジットカード情報などの金銭に関わる情報）を扱っていないため、費用をかけてまで対策を行う必要はない」と判断し、脆弱性対策を実施しないウェブサイト管理者もいる。これは、重要な情報を扱っていなくても、脆弱性が存在することによって、第三者が被害に遭う可能性を認識していないと思われる。

2. 組織的なウェブサイト管理のあり方

サーバソフトウェアのバージョンが古いことによる悪影響を防ぐためには、長期的な視点に基づく組織的なウェブサイト管理を実践する必要がある。これは、対象ウェブサイトの役割が組織の中で理解された上で、サービスレベル目標が設定され、それに必要な経営資源が割かれ、その経営資源の中で可能な実務が回っている状態を意味する。

2.1. ウェブサイトの事業上の役割を理解する［対象：経営者を含む全関係者］

ここでいうウェブサイトの役割とは、組織にとっての事業上の役割のことだ。言い換えれば、何のために組織はそのウェブサイトを保有し運用しているか、という疑問に答えることである。役割はウェブサイトによって千差万別だが、典型的なものでは以下がある。

ウェブサイト例	役割	支障時の損失[13]
製品の紹介や、問合せの受付を主とした会社のウェブサイト	広く世間一般に対する製品や会社の紹介、顧客チャネルを確保すること。	会社に対する信用の毀損
製品の販売を主としたECサイト	製品の売上を維持、向上させること。	売上の途絶、個人情報漏洩
ブラウザゲーム	ゲーム利用者からの集金による売上げや、広告収入の確保	売上の途絶

役割を理解することで、ウェブサイトに支障が生じた際の損失についても、考慮することができる。ウェブサイトがその役割を果たせなくなることで、組織にどんな損失が生じるのかは、役割と繋がっており、直結するケースもある。たとえば、例で挙げたECサイトがウイルスに感染し、停止せざるを得なくなった場合、停止中の売上げが途絶することとなる。

組織にとっての、そのウェブサイトの役割とは何であるか、まずは組織内でよく話合う必要がある。結論がでたら、明文化して関係者で見られるようにしておくことも重要だ。

場合によっては、明確な役割が見つからないこともあるかも知れない。その際は、廃止も視野に入れて検討すべきだ。ウェブサイトを管理せずに放置すると、そのサイトを閲覧した他者に迷惑が及ぶため、補償の必要があることをリスクとして認識しよう。その上で、本当に続ける価値があるのかを判断したい。

2.2. サービスレベル目標の設定［対象：実務者］

ウェブサイトの役割を合意できたら、組織にとってどの程度重要なウェブサイトなのかも見えているはずである。これに基づき、運用におけるサービスレベル目標を定めるのが次のステップだ。目標としてどのような項目を定めるかは、ウェブサイトの役割や性質によって異なるが、典型的には可用性に関する以下の項目が設定される。

[13]表では直接的な損失のみを記載した。支障が生じた原因によっては復旧までの間に、窓口対応、セキュリティ診断、アプリケーション改修、再テスト等の費用が別途発生することがある。

項目	目標例	設定理由例
サービス時間	24 時間 365 日	通常ウェブサイトは常に稼動しているため
サービス運用期間	201x 年末まで	中期事業計画に基づく
サービス稼働率	99%（年に 3.5 日程度の停止を許容）	クリティカルなサービスではないため
応答時間	10 秒以内	利用者のストレス上の許容範囲と判断したため
障害復旧期間	1 営業日（軽度障害）	サービス稼働率より

一般に、サービスレベル目標を高めるほど、必要なコストは増大する傾向にある。このため、いたずらに高い目標は設定せず、ウェブサイトの役割とのバランスを考慮する必要がある。また、設定した目標ごとに設定理由を残しておけば、目標を関係者間で合意する際や、目標を事後に見直す際に役立つ。

サービスレベル目標について、網羅的に検討したい場合、以下の資料も参考になる。

非機能要求グレード – IPA
https://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html

2.3. 運用体制の整備［対象：実務者］

サービスレベル目標に基づき、それを実現するための経営資源を確保しよう。内製の場合には運用チームの編成と稼動、外注の場合には管理コストを含む予算確保が必要となる。ここでは、たとえば以下のようなことを検討する。

- **運用者の決定**
  一定以上の複雑さのウェブサイトを運用するには、専門知識を有する技術者が必要だ。そのような人員が組織内にいない場合には、外注を検討する必要がある。なおウェブサイトを、そのウェブサイトの開発に携わった者以外が運用する事は可能だが、一定の困難が伴う。このためウェブサイトを開発した会社による運用が可能なら、それが望ましいが、そういかないケースも多々発生する。
- **24 時間 365 日稼動の検討**
  24 時間 365 日にわたって稼動するウェブサイトで、障害復旧期間を短く設定する場合、24 時間 365 日の運用管理体制を維持する必要がある。これを自組織内で整備するためには、労働契約の見直しが必要な可能性がある。
- **長期的な稼動の検討**
  サービス運用期間が数年以上にわたる場合、稼動の基盤となるハードウェアやソフトウェアの更新を考慮する必要がある。稼動を続けるには基盤のサポートが提供され続けることが必要だが、一般にそれらのサポートは 10 年以内には終了する。サポートが終了した場合、ハードウェアは故障時の修理・交換ができなくなり、ソフトウェアは脆弱性が修正されなくなり、いずれのケースでも運用の継続が困難となる。

- **サービスレベル目標の見直し**
  運用体制の整備が難しい場合、目標としたサービスレベルが高すぎる可能性がある。また、サービスレベルに関する合意が十分に形成されていない可能性もある。その際は、現実的なサービスレベルへの見直しの上で、関係者間での合意を目指そう。

なお、外注時の契約締結に関しては、以下も参考になる。

「情報システムの信頼性向上のための取引慣行・契約に関する研究会」
～情報システム・モデル取引・契約書～
http://www.meti.go.jp/policy/it_policy/softseibi/#p02_01

2.4. 実務の設計［対象:実務者］

整備した運用体制に基づき、実務を設計して運用を開始することになる。ここで設計される実務の内容は、運用対象とするウェブサイトによって異なるが、典型的には以下の項目が検討される。

- **運用チーム編成**
  運用チームを編成する。あわせて、指揮命令系統やトラブル時のエスカレーションを設定する。
- **手順の標準化・教育**
  長期的な運用をするケースでは、人員の異動に配慮する必要がある。新人の教育、属人化の防止、作業環境や手順の標準化・文書化、などを検討する必要がある。
- **事前のトラブル対策**
  停止などのトラブルの原因は様々で、原因ごとに備える必要がある。たとえば以下。
  - データの消失に備えて、定期的なバックアップを実施する。
  - ハードディスクが故障してもサービスを維持できるよう、ミラーリングする。
  - サーバ故障時のサービス維持と負荷分散を目的として、サーバを多重化した上でロードバランシングを施す。
  - 停電時の安全なシャットダウンを目的として、無停電電源装置を導入し、設定する。
  - 急激なアクセスの増加に対応できるよう、CDN 事業者と契約する。
  - 大規模災害時のサービス継続を目的として、組織の BCP に基づき、遠隔地にミラーサイトを構築する。
- **維持管理**
  運用期間の経過とともに損耗・不足しやすい部分を、定期的に確認する。たとえば以下。
  - **システムリソース・ネットワークリソース**
    たとえばサーバのメモリやディスクの空き容量、ネットワークの占有帯域などが対象である。逼迫している場合には増設を検討する。
  - **ハードウェアの故障・損耗**
    たとえばハードディスクの S.M.A.R.T.値、温度、LED ランプの状況などを確認し記録する。異常がある場合は修理・交換を手配する。

- ソフトウェアの脆弱性

  たとえばウェブアプリケーションフレームワークやアプリケーションサーバなど、フロントに近い部分は脆弱性対応の重要性が高いことが多い。次点としてウェブサーバ、データベースエンジン、OS カーネルなど。ソフトウェアベンダや JVN iPedia などで情報を入手する。脆弱性がある場合には、計画的に更新をかける。

- 異常検知

  定期的な確認以外に、機器等に異常が起きた際に、それを検知する仕組みが必要だ。一般的には監視サービスやソフトウェアが使用される。

- 自動化

  ウェブサイトの運用においては、自動化・機械化を推進することで、大幅な省力化が可能になることがある。特に、維持管理や異常検知の部分に対しては、自動化を目的とした製品が多く存在しており[14]、検討の余地がある。

[14]社内向けクラウド構築のために活用できるソフトウェアカタログ
http://ossipedia.ipa.go.jp/doc/209/
⇒ 2-a. 物理サーバ、仮想サーバ、ネットワーク機器、アプリケーションなどに対する汎用的な管理・監視ソフトウェア

コラム「IPA と運営者のやり取り」事例 2

■ 保守サービスの契約内容が不十分な事例

ウェブサイトの運用／管理の計画を立てていても、実際には想定外の事態に陥り、脆弱性対策が円滑に進まない場合がある。

組織内で運用管理を行っているケースにおいて、「担当者が退職してしまい、新しい技術者が来るまで技術的な対応が行えない」といった、対策を実施したくても対策が実施できない状況に陥ってしまうウェブサイト管理者がいる。

ウェブサイト管理者がウェブサイト製作業者と保守契約を行っているケースにおいても、「製作業者へ対応を依頼しているが、対応してくれない」というウェブサイト管理者から報告を受ける場合がある。製作業者へ連絡をすると、「保守契約の範囲外なので、無償では対応できない（別途費用がかかる）」と両者の認識に乖離がある。この事から、ウェブサイト管理者が正しく保守契約の内容を把握していない、もしくは、保守契約の内容の認識に齟齬がある事が伺える。その結果、ウェブサイト管理者は、想定していた保守サービスが受けられないため、想定外の費用を捻出する必要が生じた。

その他の事例として、ホスティングサーバを利用しているケースにおいて、ウェブサイト管理者へ脆弱性を報告した結果、ウェブサイト管理者の調査により脆弱性の原因が利用しているホスティングサーバから提供されているサービスやツールに起因することが判明する場合がある。ホスティングサーバサービスの場合、サーバの運用／管理をサービス提供事業者に任せているため、ウェブサイト管理者が保守サービスを利用しようとした際に、「事業者の連絡先が分からない、契約書等を紛失したため、どこの業者のどのサービスを利用しているかがわからない」と回答するウェブサイト管理者もいる。

また、ウェブサイト管理者としてはサービス提供事業者に早急な対策を望んでも、サービス提供事業者から「ソフトウェア間の互換性の問題で最新版へアップデートできない」と言われる場合がある。ホスティングサーバでは、複数の利用者が同じサーバ上にあるソフトウェアを利用しているため、サービスの仕様上早急な対策が出来ない場合がある。このため、ウェブサイト管理者は、已むを得ず他サービスへの移行を検討する必要が生じた。

コラムで紹介した事例では、ウェブサイトの運用／管理に問題のある事例について紹介したが、ウェブサイト運営者の中には「ウェブサイトの運用／管理について、委託業者と脆弱性対策を含んだ適切な保守契約を締結している」や「保守費用を計画的に予算化している」などにより、脆弱性対策を円滑に進めているケースもある。また、計画的に「脆弱性診断」を行っている結果、IPA がウェブサイト運営者へ報告を行う前にウェブサイト運営者自身で脆弱性を発見し対策済みである場合もある。

3. 組織的なウェブサイト管理のケーススタディ

本節では、組織的なウェブサイト管理を目指した例として、IPA が運用しているあるウェブサイト（以下、ウェブサイト A とする）を採り上げる。

3.1. ウェブサイトの役割

ウェブサイト A の役割は、不特定多数の組織に対して情報セキュリティ対策水準の引き上げを支援することだ。その中でも、利用者の自主的な学習を支援することと、普及啓発や対策推進を担う人の活動を支援することを目指している。

3.2. サービスレベル目標

ウェブサイト A のサービスレベルにおいて重要な考え方は、学習や普及啓発をしようとする利用者のモチベーションを損なわないようにすることだ。利用者の多くは日中に働いており、その業務の一環としてウェブサイト A を利用すると想定している。

このような考え方に基づき、次のようなサービスレベル目標を設定した。

表：ウェブサイト A のサービスレベル目標

項目	目標例
サービス提供時間帯	24 時間 365 日
サービス運用期間	5 年
深夜休日のサービス提供の考え方	サービス提供を行う 障害対応は行わない
障害発生率	年に数回程度を想定
障害発生時の復旧目標時間	（平日）半日程度以内、 （深夜休日）翌営業日の半日程度以内
メンテナンス	（定期）毎月第 4 水曜日の夜間 （臨時）随時 （緊急）随時 ※ サービスの停止を伴う場合事前に告知する

その後、役割およびサービスレベル目標に加え、体制やトラブル時の対応方針などをまとめて明文化し、関係者間で合意した。

なおウェブサイト A の場合、サービスレベル目標は 2 段階に分けて定めた。まず、システム開発の段階でサーバ機器やネットワーク等の要件に関係が深い部分（例：障害発生時の復旧目標時間）を定めた。次に、運用開始の段階ではサービスのコンセプトに関係が深い部分（例：平日日中の利用が多いという想定からメンテナンス時間帯は夜間が望ましい）を定めた。

3.3. 運用体制および関係者

ウェブサイトAは、図のような体制で運用されている。

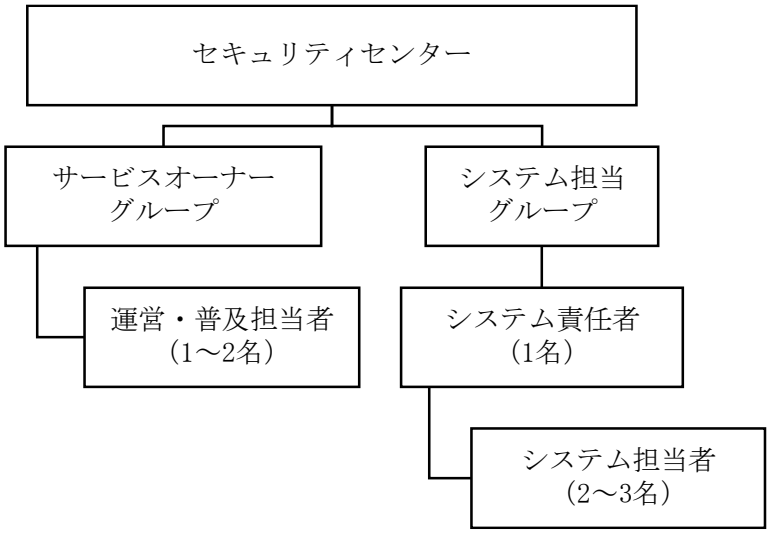


図 12：ウェブサイト A 運用体制図

サービスオーナーグループとシステム担当グループとは分かれており、それぞれに数名の関係者がいる。専任の関係者はおらず、すべて他業務との兼任である。システムに直接携わるのは 2～3 名のシステム担当者のみだが、システムをどうメンテナンスするかは組織内の関係者、とりわけサービスオーナーのグループとの調整により決定する。

3.4. 定期メンテナンスおよびミーティング

ウェブサイトAはその位置づけから､セキュリティ上の問題を起こさずに運用する必要がある。このため、セキュリティを維持するためのメンテナンスが必要であり、定期的または臨時にメンテナンスを行うことを、サービスレベル目標の一部として定めている。

このうち定期メンテナンスは、毎月第 4 水曜日の夜間に行う運用となっている。このメンテナンスでは、計画的に対応できる項目を実施することになっており、これまでに実施した項目として以下がある。

- ウェブコンテンツの更新（急ぎではないもの）
- サイバーセキュリティ注意喚起サービス「icat[15]」をページ内に追加
- サーバソフトウェアのバージョンアップ（緊急対応が必要な脆弱性対応を含まないもの）
- サーバ証明書の更新

また、定期メンテナンスは当日のみ対応するものではなく、準備や後処理がある。これらを含めた定期メンテナンスの典型的な対応は以下。

[15]サイバーセキュリティ注意喚起サービス「icat」
https://www.ipa.go.jp/security/vuln/icat.html

時期	対応
2～3 週間前	（必要に応じて）メンテナンス手順を検証用環境で実施し、手順の正しさや副作用の有無をレビューする。
1 週間前	定期メンテナンス方針に関してシステム担当グループ関係者一同で MTG を開き、その月のメンテナンス方針を検討する。検討結果はサービスオーナーグループにも伝えられ、必要に応じて利用者へ告知する。
2～3 日前	（必要に応じて）データセンタに対する入館申請や、監視センターに対する事前連絡を行う。
1 日前	（必要に応じて）システム構成のバックアップを取得する。
直前	ファイアウォールにより外部からのアクセスを遮断する。
実施時間帯	定期メンテナンスを実施。
直後	ファイアウォールにより遮断していたアクセスを元に戻す。また、監視センターに対する事後連絡を行う。
3 日後まで	定期メンテナンス結果に関する書類をまとめる。

3.5. 整備した資料およびソフトウェア環境

ウェブサイト A のシステム担当者が利用する資料として、以下を整備している。

- 開発業者によるアプリケーション設計資料、マニュアル等一式
- データセンタから提示されたマニュアル、取決め等
- 定期メンテナンスの計画、結果記録用フォーマット
- リモートログイン等標準作業手順書、クライアント環境構築手順書
- データセンタ入館手順
- 検証用環境（仮想マシン）構築手順書、および使い方
- 開発用環境（仮想マシン）利用マニュアル
- ユーザインターフェース確認手順

これらを整備した目的は、作業のチーム内標準化により担当者間での作業内容のズレ乖離を防ぐこと、また、作業が属人化するのを防ぐためである。各資料が規定する作業は、資料作者ではない者が実施し、実施可能な内容であることを確かめている。

また、資料以外に以下のソフトウェア環境を整備している。

- 検証用環境
  ウェブサイト A が稼動するソフトウェア環境を、可能な範囲で忠実に模した環境。サーバソフトウェアバージョンアップ等の際に、事前に検証するために使用する。ウェブサイト A で稼動するウェブアプリケーションのほか、ウェブサーバ、アプリケーションサーバ、データベース等の必要なソフトウェア一式を含んでいる。各ソフトウェアの設定・設置パス・バージョンはウェブサイト A のシステムに準じる。VMware の仮想マシン 2 台として実装。
- 開発用環境
  ウェブサイト A で稼動するウェブアプリケーションの開発環境を、可能な範囲で忠実に模した環境。開発時点のバージョンにあわせた統合開発環境および周辺ソフトウェアで構成。VMware の仮想マシン 1 台として実装。

3.6. サーバソフトウェアバージョンアップの典型的な手順

ウェブサイト A のサーバソフトウェアをバージョンアップする際は、前節の手順および環境を用い、次のように実施するのが典型的である。

1. バックアップ
   検証用環境に手を加えるのに先立ち、元に戻せるよう準備する。検証用環境に対するバックアップまたは、スナップショットの作成により実施する。
2. 手順作成
   バージョンアップ対象サーバソフトウェアのマニュアルやリリースノートに従い、ウェブサイト A におけるバージョンアップ手順を作成する。この手順は、実行すべきシェルコマンドを個別に記載する程度に委細なものである。また対象が Struts 2 の場合に限り、入替え対象 jar ファイルを 100 種類程度の中から、依存関係に留意して選別する過程が別途必要となる。
3. 実施可能性確認
   作成した手順を検証用環境に対して実施し、その手順でバージョンアップが可能かどうかを確認する。もし問題が生じた場合には、バージョンアップ手順を修正し、検証環境を元に戻した上で再実施する。
4. 副作用確認
   バージョンアップした検証用環境に対して、ユーザインターフェース確認（100 項目程度）を実施し、想定外の副作用がないかを確認する。もし問題が生じた場合は、原因を究明し、必要に応じてバージョンアップ手順を修正する。
5. 本番適用
   本番環境に対して、バージョンアップ手順を実施する。実施後、ユーザインターフェース確認を実施する。

3.7. 対応実施例

ウェブサイト A は内部で Apache Struts 2（以下 Struts 2）を利用しているが、Struts 2 には深刻な脆弱性が複数回発見されており、特に重点的に脆弱性対応を行っている。ここでは対応実施例として、Struts 2 の脆弱性対応を紹介する。

■ 運用開始前の Struts 2 OGNL インジェクション対応

開発工期中にユーザテストの一環として実施したウェブアプリケーション検査にて、OGNL インジェクションの脆弱性[16]が発見された。開発業者にて Struts 2 のバージョンアップを実施し、関連するテストを再実施した。運用開始前なのでサービスは稼動しておらず、停止期間は生じなかった。

[16]Apache Struts における任意の Java コードを実行される脆弱性
http://jvndb.jvn.jp/jvndb/JVNDB-2012-001033

この時の対応を経て、検証用環境およびユーザインターフェース確認手順の必要性が認識され、整備された。

■ 運用開始後 1 回目の Struts 2 脆弱性対応

セキュリティ関連サイトからの情報[17]で Struts 2 の深刻な脆弱性の存在を認識し、緊急対応に移行。既に Struts 2 開発者は修正版をリリースしていたため、5 営業日程度で Struts 2 を更新するよう計画した。しかし、以下の問題が生じたため 15 営業日程度を要した。

- CSRF 対策用トークンの名称が Struts 2 仕様変更により変化[18]し、ウェブアプリケーションの改修を要した。
- テストの過程で一部画面に表示上のバグが発見され、ウェブアプリケーションの改修が検討された。問題の画面のバグは修正できる見通しが立ったものの、他の機能への影響がないことを保証できなかったため、問題の画面と他の機能の重要度を比較した結果、他の機能へ影響がある方がより重大な問題になるとの判断から、修正は見送られた。
- Struts 2 に新たに lang3 ライブラリが追加されており、これを追加しないと動作しなかった。

この時の対応では、初めてウェブアプリケーションを一部改修する必要が生じ、それに伴い開発用環境が整備された。また、検証用環境におけるログの読み方等の、運用上の技術的ノウハウも得られた。

■ 運用開始後 2 回目の Struts 2 脆弱性対応（攻撃あり）

セキュリティベンダからの情報で、Struts 2 の深刻な脆弱性[19]と、それを狙った攻撃が国内で生じている事実[20]を認識。即座にサービスの公開を停止、緊急対応に移行し 5 営業日程度で Struts 2 を更新するよう計画し、4 営業日で更新を完了、サービスを再開した。以下、対応順を表にまとめた。

[17]Apache Struts における任意の OGNL コードを実行される脆弱性
http://jvndb.jvn.jp/jvndb/JVNDB-2013-003318
[18][Apache-SVN] Diff of /struts/struts2/trunk/core/src/main/java/org/apache/struts2/util/TokenHelper.java
https://svn.apache.org/viewvc/struts/struts2/trunk/core/src/main/java/org/apache/struts2/util/TokenHelper.java?r1=1099157&r2=1368827&pathrev=1368827&diff_format=h
[19]Apache Struts において任意のコマンドを実行される脆弱性
http://jvndb.jvn.jp/jvndb/JVNDB-2013-003469
[20]Apache Struts2 の脆弱性（S2-016）を悪用した攻撃の急増について | セキュリティ情報 | 株式会社ラック
http://www.lac.co.jp/security/alert/2013/07/18_alert_01.html

時系列	PDCA	事象
1		Struts 2 開発元から脆弱性（S2-016）に関する情報が公開される。
2		セキュリティベンダから注意喚起。Struts 2 の脆弱性（S2-016）を悪用した攻撃の急増を確認とのこと。
3	P	システム担当者にて状況を把握。関係者と状況を共有し、緊急対応を開始。
4	D	攻撃の痕跡の有無を調査し、無いと確認。
5	D	ウェブサイト A をサービス停止（ファイアウォール制御による）
6	D	5 営業日後の再開が目標として共有される。
7	D	テスト環境上で Struts の更新を完了。動作テストを開始できる状態になる。
8	C	テスト環境上での動作テストを完了、異常なし。
9	DC	本番環境上で Struts の更新を完了。さらに動作テストを完了、異常なし。ウェブサイト A をサービス再開。

表のうち PDCA 列は、それぞれの工程が PDCA サイクルのどこに当たるかを示したものだ。表には A にあたる工程がないため、今後に向けた改善が十分に行われていない可能性も示唆される。

この時の対応では、攻撃が生じている状態での緊急メンテナンスを初めて経験することになったが、関係者間での迅速な意思決定ができたことと、それまでに培われていた技術的ノウハウが生かされたことで、計画を前倒ししてメンテナンスを終了できた。しかし、サービス停止期間は 4 営業日と長かったため、今後はできるだけ短い停止期間で対応できるよう検討することが、課題として残された。

もし、ウェブサイト A を適切な体制で管理しなかった場合、Struts 2 の脆弱性は修正されないまま放置され、そこを突いた不正侵入を受け、個人情報の漏えいやウェブサイトの改ざんなどの被害が生じ、ひいては IPA の信頼性が損なわれる可能性があった。

IPA は上記問題を起こさないことを社会的責任と捉え、本章で述べたようにウェブサイト A の役割とサービスレベルを定め、相応な体制・環境・手順等を整備して、ウェブサイト A を運営している。その結果、攻撃を受ける前に脆弱性を塞ぐことができ、今日まで大きな問題を生じることなく、数千人を越える利用者に対してサービスを継続して提供できている。

おわりに

本書では、「情報セキュリティ早期警戒パートナーシップ」の届出受付業務にて得られたウェブサイトの情報を集計した結果から考察したシステムの運用状況と、組織的な管理を行うにあたっての考慮事項を IPA の実例と併せて記載した。

集計では、サポートが終了しているバージョンの PHP と最新パッチが適用されていないサーバソフトウェアが利用され続けていることが懸念される結果が示された。ウェブサイトの管理にはコンテンツの更新、利用者からの問い合わせやトラブル対応等、必ず運用がついて回るが、その運用にはサーバで稼動するソフトウェアの管理についても盛り込んで頂きたい。PHP 等のミドルウェアやサーバソフトウェアをバージョンアップした場合、現在動作しているウェブアプリケーションが動作しなくなる恐れがある。が、システムを構成するソフトウェアの脆弱性は、日々発見され、攻撃者が悪用できる既知の脆弱性は増え続けることが見込まれる。そうした状況で、ウェブサイトの脆弱性を突かれた場合、運営者のみならず、利用者にも迷惑がおよび、対応には相応のコストがかかることを組織として認識して頂きたい。

そして、組織としてシステムのセキュリティ上のリスクの発生を防ぐには、長期的な視点に基づく組織的な管理が不可欠である。組織的な管理のあり方は、組織によって千差万別だが、実践するには、対象システムの役割を理解し、適切な目標設定と経営資源の割り当てによる実務の設計が重要となる。組織内でよく検討を行い、目標に合致した最適な管理を実践頂きたい。本書内で挙げた資料と IPA が組織的な管理を目指したシステムの例が参考になれば幸いである。

本書が、組織における効果的なシステム運用を考慮するうえでの一助になることを期待している。

IPA テクニカルウォッチ

サーバソフトウェアが最新版に更新されにくい現状および対策

～ 中長期的な視点からウェブサイトの組織的な管理を ～

［発　　行］2014 年 4 月 25 日
［著作・制作］独立行政法人情報処理推進機構 セキュリティセンター

編集責任　　金野 千里

執筆者　　永安 佑希允
協力者　　木曽田 優　　森田 一行　　谷口 隼祐　　篠原 崇宏