

2026年10月21日（水） 21時55分

PHP の脆弱性を直す 修正版の公開まで、あと1時間

首相の呼びかけから23時の公開まで。
その夜の社会の判断を読み解く



高等学校 情報Ⅰ（1）情報社会の問題解決／（4）情報通信ネットワークとデータの活用

問い（ワークシート①）

首相が、一つのソフトウェアの更新を X で呼びかけた

「皆様に、今夜に対応して頂きたい
大至急のお願いです」

なぜ、
そこまで
急いだのか？

高市早苗首相の X の投稿（10月21日 21時55分ごろ）

→ ワークシート①に、予想を一言で

その夜

銀行もネットショップもゲームも、 予防のために自ら止めた

一時停止

銀行

一時停止

ネット
ショップ

一時停止

ゲーム

一時停止

保育の連絡

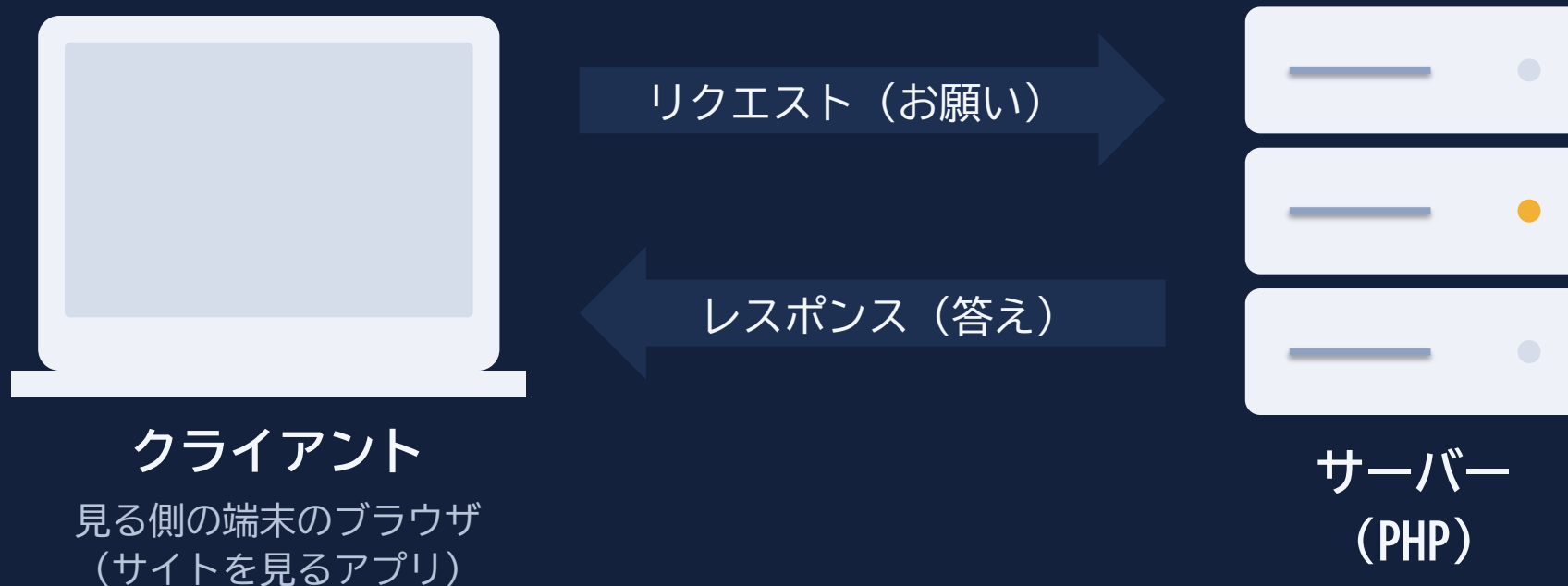
一時停止

省庁のサイト

共通点：PHP を使っていた（または、その可能性を確かめる必要があった）

PHP の位置

PHP はサーバー側で動く言語。
ウェブサイトの約7割が PHP を使う



約7割

ウェブサイトの
69.2% が PHP

※サーバー側の言語が
分かるサイトの中で
数えた割合 (W3Techs)

直すのは、サーバーを運営する側

2 件の脆弱性 (2 件を合わせて「PHP2shell」と呼ばれる)

PHP に 2 件の脆弱性が見つかった。 認証なしで、外からコードを実行されうる

CVE-2026-109618

メモリ不足 (OOM) の途中で、
中途半端なデータが残る

CVE-2026-109619

前のリクエストの、解放された
メモリを指す状態が残る

CVSS v4.0 2 件とも

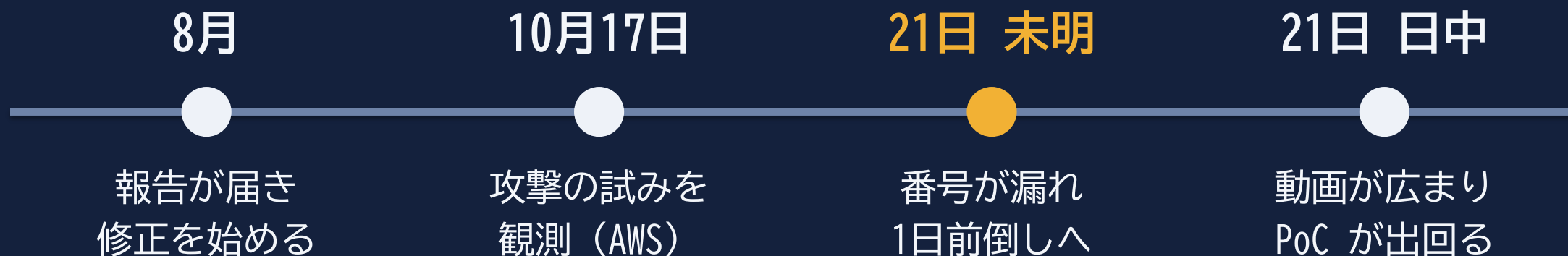
9.2 / 10

完全な回避策はなく、
更新だけが安全な選択

近くの人と30秒：「認証なし」「リモート」「コード実行」。3つがそろくと、なぜ最悪？

時系列① 公開まで

伏せて直すはずが、 公開の前に情報が漏れた



見方が割れる： AWS 「公開の約12時間前から複数の PoC」
PHP 側 「公開時点で、機能する攻撃コードは把握せず」

AWS (エー・ダブリュー・エス) = Amazon のサーバー貸し出しサービス。今回の報告はその調査チーム

PoC (ピー・オー・シー) = 悪用できることを確かめるための、お試しのプログラム

時系列② 公開の夜と、その後

23時に公開。26日、米 CISA が 「悪用が確認された脆弱性」の一覧に



IPA（10月27日）：国内での攻撃被害の事例は「把握しておりません」。ただし更新は今も必要

米 CISA（シーサ）＝アメリカ政府の、サイバーセキュリティを担当する役所

IPA（アイ・ピー・エー）＝情報処理推進機構。日本で情報セキュリティの注意を呼びかける国の機関

NCO＝国家サイバー統括室。日本の国のサイバーセキュリティの司令塔

一般論：なぜ「修正の公開は、弱点の公開でもある」のか

直す前と後のコードを比べると、 弱かった場所が分かってしまう

修正前の
コード



修正後の
コード



差分（さぶん）

直す前と後で、
変わった部分

数時間

今回、公開後に PoC が
次々と出るまで（AWS）

1時間足らず

AI が差分から攻撃コードを
作るまで（Firefox などでの
研究。PHP ではない）

1

PHP のコードは公開されている。修正版が出ると、
直す前と後を誰でも比べられる

2

比べて変わった部分＝差分は「どこを直したか」。
つまり「どこが弱かったか」を示す

3

攻撃する側も差分を読み、まだ更新していない
サーバーをねらう → 公開したら、すぐ更新

今回急いだ理由は両方：公開前の漏れ（→ 1日前倒し）と、公開後に差分から来る攻撃

守りの考え方のまとめ：NIST CSF 2.0（ニスト・シー・エス・エフ）

守りを 6 つの機能に分ける。順番ではなく 同時に回し、真ん中に「統治」

アメリカの標準化機関 NIST がまとめた、サイバーセキュリティ対策の考え方

統治 誰が決め、誰に伝えるか（ほかの 5 つを貫く中心）

識別

どこで
動いている？

防御

更新する

検知

攻撃の跡を
探す

対応

止める・
知らせる

復旧

直して再開

問い：10月21日の夜、ふだんから準備ができていなかったら困ったのは、6 つのうちどれ？

（例：どこで PHP が動いているかの一覧＝識別、夜に止めると決める人＝統治。一つでなくてよい）

話し合い：ワークシート②（個人1分 → 班5分 → 発表3班）

21時。あなたはネットショップの責任者。 お店のサイトを、今夜止める？ 止めない？

止めると

- ・ 利用者が使えない
- ・ 売上・信用への影響
- ・ 問い合わせが増える

止めないと

- ・ 認証なしで乗っ取られるおそれ
- ・ 会員の情報が盗まれるおそれ
- ・ ほかに攻撃の踏み台に

判断に必要な情報は？ どこで動いている？ 外から届く？ 誰が決める？

誰が、何をするか

直すのはサーバーの運営者。 利用者にも、利用者の仕事がある

サーバーの運営者

- ・どこで PHP が動くかを調べる
- ・修正版に更新する
- ・難しければ一時停止・利用者に告知

利用者（私たち）

- ・止まったことを受け止める
- ・公式の発表で確かめる
- ・漏れた情報やうわさを広めない

自分にできること（「今回」と「ふだん」を分ける）

情報社会の一員として、今日からやる4つ

1

ふだん

更新を
後回しにしない

自動更新をオンに。
再起動も早めに。

2

ふだん

パスワードを
使い回さない

今回の攻撃とは別。
管理の仕組みと
多要素認証を。

3

今回も

情報源を
確かめる

公式の発表を見る。
漏れた情報を
広めない。

4

今回も

運営するなら
担当を決める

部活のサイトや
個人の開発も。

ふり返し（ワークシート③④）

この夜を、自分の言葉で説明する

③ 自分がとる行動を一つ（いつ・何を） ← 授業の中で

④ 10月21日に何が起きたか（200字） ← 宿題でよい

使う言葉：「脆弱性」「修正の公開」「一時停止」

最後に：①の予想を、一言で言い直す

守りは、それぞれの仕事の組み合わせ。自分の分を、今日から。



この作品は クリエイティブ・コモンズ 表示 4.0 国際 ライセンスで提供されています。

出典：Data is CITY (teleport.jp/dataiscity/)

<https://creativecommons.org/licenses/by/4.0/deed.ja>

付録 A (時間があれば・質問が出たら)

侵入された後、組織の中に広がる速さ

29分

侵入から、別のマシンへ
広がるまでの平均 (2025年)

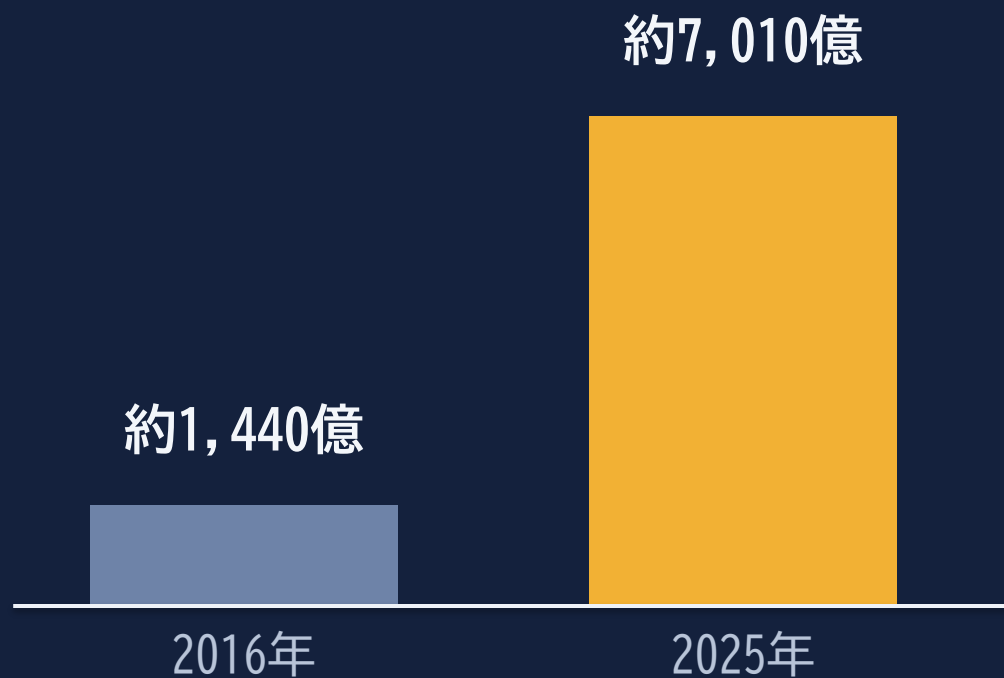
27秒

観測された最速

対応は
「7日・30日」でなく
「時間」で測る
(AWS の提言)

侵入された後に中で広がる速さ。修正を適用する速さとは別の量 (PHP の攻撃の速さではない)

使われていない番地に届く通信は、
2016年から9年で約4.9倍



約4.9倍

年間の総観測パケット数
2025年の約55% は
調査目的のスキャン
(すべてが攻撃ではない)