

動的アプリケーションセキュリティテスト 報告書

acme/app

不合格	6 アラート	1 高	0 中	3 低	2 情報
-----	-----------	--------	--------	--------	---------

対象

検査した URL	http://127.0.0.1:62189
サーバー	PHP の組み込みウェブサーバー (APP_ENV=test)
検査	OWASP ZAP : End-to-End テストのリクエストとレスポンスの確認と攻撃 (最大 5 分)
End-to-End テスト	vendor/bin/phpunit --testsuite EndToEnd
ログイン	なし (訪問者として)
アラートのある URL	4
不合格の基準	高以上

実行

実行日時	2026-10-17 15:10:32 ~ 15:16:36 (UTC)
所要時間	6分4秒
実行環境	ローカル (macOS 27.0 arm64)
ソフトウェア	PHP 8.5.11 OWASP ZAP 2.17.0 pieceofcake2/security-dast 1.0.0+no-version-set

フルスキャンは、攻撃者と同じようにスクリプトや SQL などページとパラメーターを攻撃します。多くの脆弱性を見つけますが、クロールできなかったページのものなど、すべてではありません。

リスク別のアラート

1. 高

No.	アラート	リスク	リクエスト
1-1	クロスサイトスクリプティング (反射型)	高	1

2. 低

No.	アラート	リスク	リクエスト
2-1	SameSite属性のないCookie	低	1
2-2	X-Content-Type-Optionsヘッダーがありません	低	3
2-3	サーバーが "X-Powered-By" HTTPレスポンスヘッダーフィールド経由で情報を漏洩	低	2

3. 情報

No.	アラート	リスク	リクエスト
3-1	セッション管理レスポンスの特定	情報	1
3-2	ユーザーが制御可能なHTML要素属性 (XSSの可能性)	情報	3

1. 高

1-1 クロスサイトスクリプティング (反射型)

高

アラート ID	40012
リスク	高
確度	中
CWE	CWE-79
WASC	WASC-8
リクエスト	1

クロスサイトスクリプティング (XSS) は、攻撃者が提供したコードをユーザーのブラウザーインスタンスに反映 (echo) させる攻撃手法です。ブラウザーインスタンスには、標準的なWebブラウザークライアントのほか、WinAmp 内のブラウザ、RSSリーダー、メールクライアントなどのソフトウェア製品に埋め込まれたブラウザーオブジェクトも含まれます。コード自体は通常HTML/JavaScriptで記述されますが、VBScript、ActiveX、Java、Flash、その他ブラウザーがサポートするあらゆる技術に及ぶ可能性があります。

攻撃者がユーザーのブラウザーに自身のコードを実行させると、そのコードはホスティングしているWebサイトのセキュリティコンテキスト (またはゾーン) 内で実行されます。この権限レベルでは、コードはブラウザーがアクセス可能なあらゆる機密データの読み取り、変更、送信する能力を持ちます。クロスサイトスクリプティングの被害を受けたユーザーは、アカウントを乗っ取られたり (Cookie窃取)、ブラウザーを別の場所にリダイレクトされたり、訪問しているWebサイトから配信される詐欺的なコンテンツを表示させられたりする可能性があります。クロスサイトスクリプティング攻撃は本質的にユーザーとWebサイトとの間の信頼関係を侵害します。ファイルシステムからコンテンツを読み込むブラウザーオブジェクトインスタンスを利用するアプリケーションはローカルマシンゾーン下でコードを実行する可能性があり、システムの侵害につながる可能性があります。

クロスサイトスクリプティング攻撃には、非永続的 (non-persistent)、永続的 (persistent)、DOMベースの3つのタイプがあります。

非永続的攻撃とDOMベース攻撃では、ユーザーは悪意のあるコードが仕込まれた特別に細工されたリンクを訪問するか、脆弱なサイトにPOSTすると攻撃を開始する悪意のあるWebフォームを含むWebページを訪問する必要があります。悪意のあるフォームの使用は、脆弱なリソースがHTTP POSTリクエストのみを受け付ける場合によく行われます。そのような場合、フォームは (例えばJavaScriptを使用して) 被害者の知らないうちに自動的に送信されることがあります。悪意のあるリンクをクリックするか悪意のあるフォームを送信すると、XSSペイロードが反映されてユーザーのブラウザーによって解釈され、実行されます。Adobe Flashのような埋め込みクライアントを使用することで、ほぼ任意のリクエスト (GETおよびPOST) を送信する手法もあります。

永続的攻撃は、悪意のあるコードがWebサイトに送信され、一定期間保存される場合に発生します。攻撃者のお気に入りのターゲットの例には、メッセージボードの投稿、Webメールのメッセージ、Webチャットソフトウェアなどがよく含まれます。疑いを持たないユーザーは、追加のサイト/リンク (例: 攻撃者のサイトやメールで送信された悪意のあるリンク) と対話する必要はなく、単にコードを含むWebページを表示するだけで被害に遭います。

対策

フェーズ: アーキテクチャおよび設計

この脆弱性の発生を許さない、あるいは回避を容易にする構造を備えた、十分に検証されたライブラリやフレームワークを使用してください。

適切にエンコードされた出力を生成しやすくするライブラリやフレームワークの例には、MicrosoftのAnti-XSSライブラリ、OWASP ESAPIのEncodingモジュール、Apache Wicketなどがあります。

フェーズ: 実装、アーキテクチャおよび設計

データが使用されるコンテキストと期待されるエンコーディングを理解してください。これは、異なるコンポーネント間でデ

ータを転送する場合や、Webページやマルチパートメールメッセージのように複数のエンコーディングを同時に含む可能性のある出力を生成する場合に特に重要です。期待されるすべての通信プロトコルとデータ表現を調査し、必要なエンコーディング戦略を決定してください。

別のWebページに出力されるデータ、特に外部入力から受け取ったデータについては、すべての非英数字に対して適切なエンコーディングを使用してください。

必要なエンコーディングとエスケープの種類の詳細については、XSS対策チートシートを参照してください。

フェーズ: アーキテクチャおよび設計

クライアント側で実行されるセキュリティチェックについては、CWE-602を回避するために、これらのチェックがサーバー側で複製されていることを確認してください。攻撃者は、チェックが実行された後に値を変更したり、クライアントを変更してクライアント側のチェックを完全に削除したりすることで、クライアント側のチェックをバイパスできます。その後、これらの変更された値がサーバーに送信されます。

利用可能であれば、データとコードの分離を自動的に強制する構造化されたメカニズムを使用してください。これらのメカニズムは、開発者が出力を生成するすべての箇所でこの機能を提供することに頼るのではなく、関連するクォーティング、エンコーディング、および検証を自動的に提供できる場合があります。

フェーズ: 実装

生成されるすべてのWebページについて、ISO-8859-1やUTF-8などの文字エンコーディングを使用し、指定してください。エンコーディングが指定されていない場合、WebブラウザはWebページで実際に使用されているエンコーディングを推測して、異なるエンコーディングを選択する可能性があります。これにより、Webブラウザが特定のシーケンスを特殊なものとして扱い、クライアントが巧妙なXSS攻撃にさらされる可能性があります。エンコーディング/エスケープに関連するその他の緩和策については、CWE-116を参照してください。

ユーザーのセッションクッキーに対するXSS攻撃を緩和するために、セッションCookieを HttpOnly に設定してください。HttpOnly 機能をサポートするブラウザ (最近のバージョンのFirefoxなど) では、この属性により document.cookie を使用する悪意のあるクライアントサイドスクリプトがユーザーのセッションCookieにアクセスするのを防ぐことができます。これは完全な解決策ではありません。なぜなら、HttpOnly はすべてのブラウザでサポートされているわけではないからです。さらに重要なことに、XMLHttpRequest やその他の強力なブラウザ技術は、HttpOnly フラグが設定されている Set-Cookie ヘッダーを含むHTTPヘッダーへの読み取りアクセスを提供します。

すべての入力を悪意のあるものと想定してください。「既知の安全なものを受け入れる (accept known good)」入力検証戦略、つまり、仕様に厳密に準拠する入力の許可リスト (ホワイトリスト) を使用してください。仕様に厳密に従わない入力は拒否するか、準拠するよう変換してください。悪意のある、または不正な形式の入力を探すことだけに頼らないでください (すなわち、拒否リスト (ブラックリスト) に頼らないでください)。ただし、拒否リストは潜在的な攻撃を検出したり、どの入力が非常に不正な形式であるため完全に拒否すべきかを判断したりするのに役立ちます。

入力検証を実行する際は、長さ、入力の種類、許容可能な値の全範囲、入力の欠落または過剰、構文、関連フィールド間の一貫性、およびビジネスルールへの準拠など、すべての潜在的に関連するプロパティを考慮してください。ビジネスルールの口ジックの例として、「boat」は英数字のみを含むため構文的には有効かもしれませんが、「red」や「blue」のような色を期待している場合は有効ではありません。

アプリケーション内の明確に定義されたインターフェースで入力検証を実行することを確認してください。これにより、コンポーネントが再利用されたり、他の場所に移動されたりしても、アプリケーションを保護するのに役立ちます。

リクエスト

POST http://127.0.0.1:62189/contacts/add	
パラメーター	data[Contact][message]
攻撃	<scrIpt>alert(1);</scRipt>
根拠	<scrIpt>alert(1);</scRipt>

参考

- <https://www.zaproxy.org/docs/alerts/40012/>
- <https://owasp.org/www-community/attacks/xss/>
- <https://cwe.mitre.org/data/definitions/79.html>

2. 低

2-1 SameSite属性のないCookie

低

アラート ID	10054-1
リスク	低
確度	中
CWE	CWE-1275
WASC	WASC-13
リクエスト	1

SameSite 属性がないCookieが設定されました。このため、そのCookieは「クロスサイト」リクエストの結果として送信される可能性があります。SameSite属性は、クロスサイトリクエストフォージェリ、クロスサイトスクリプトインクルージョン、およびタイミング攻撃に対する効果的な対策です。

対策

すべてのCookieについて、SameSite属性を 'lax' または、理想的には 'strict' に設定してください。

リクエスト

GET http://127.0.0.1:62189/contacts/add	
パラメーター	CAKEPHP
根拠	Set-Cookie: CAKEPHP

参考

- <https://www.zaproxy.org/docs/alerts/10054/>
- <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-cookie-same-site>

2. 低

2-2 X-Content-Type-Optionsヘッダーがありません

低

アラート ID	10021
リスク	低
確度	中
CWE	CWE-693
WASC	WASC-15
リクエスト	3

MIME推測防止ヘッダー X-Content-Type-Options が 'nosniff' に設定されていません。この設定により、ブラウザが Content-Typeヘッダーを無視してファイル内容から推測でMIMEタイプを決定する可能性があります。これは、攻撃者がアップロードしたファイルを意図しないコンテンツタイプ (JavaScript、HTMLなど) として実行させ、XSS攻撃を実行する手法に悪用される恐れがあります。現代のブラウザでも、この設定がない場合はMIME推測が実行される可能性があります。

対策

アプリケーション/Webサーバーが Content-Type ヘッダーを適切に設定し、すべてのWebページに対して X-Content-Type-Options ヘッダーに 'nosniff' を設定していることを確認してください。
可能であれば、エンドユーザーが MIME スニффイングを一切実行しないか、Webアプリケーション/Webサーバーの指示に従って MIME スニッフイングを無効化できる、標準に準拠した最新のWebブラウザを使用していることを確認してください。

リクエスト

GET http://127.0.0.1:62189/css/fonts.css	
パラメーター	x-content-type-options
その他の情報	この問題は、エラーページ（401、403、500 など）にも依然として該当します。これらのページもインジェクション攻撃の影響を受けることが多く、その場合ブラウザが意図されたコンテンツタイプを無視して MIME スニッフイングを行うリスクが残るためです。 しきい値「高」では、このスキャンルールはクライアントエラーおよびサーバーエラーのレスポンスに対してアラートを発生させません。

GET http://127.0.0.1:62189/css/milligram.min.css	
パラメーター	x-content-type-options
その他の情報	この問題は、エラーページ（401、403、500 など）にも依然として該当します。これらのページもインジェクション攻撃の影響を受けることが多く、その場合ブラウザが意図されたコンテンツタイプを無視して MIME スニッフイングを行うリスクが残るためです。 しきい値「高」では、このスキャンルールはクライアントエラーおよびサーバーエラーのレスポンスに対してアラートを発生させません。

GET http://127.0.0.1:62189/font/raleway-latin.woff2	
パラメーター	x-content-type-options
その他の情報	この問題は、エラーページ（401、403、500 など）にも依然として該当します。これらのページもインジェクション攻撃の影響を受けることが多く、その場合ブラウザーが意図されたコンテンツタイプを無視して MIME スニффイングを行うリスクが残るためです。 しきい値「高」では、このスキャンルールはクライアントエラーおよびサーバーエラーのレスポンスに対してアラートを発生させません。

参考

- <https://www.zaproxy.org/docs/alerts/10021/>
- [https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941\(v=vs.85\)](https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85))
- <https://owasp.org/www-community/Security-Headers>

2. 低

2-3 サーバーが "X-Powered-By" HTTPレスポンスヘッダーフィールド経由で情報を漏洩

低

アラート ID	10037
リスク	低
確度	中
CWE	CWE-497
WASC	WASC-13
リクエスト	2

サーバーが1つ以上の "X-Powered-By" HTTPレスポンスヘッダーを介して情報を漏洩しています。

この情報にアクセスした攻撃者は、Webアプリケーションが依存している他のフレームワークやコンポーネントを特定し、それらのコンポーネントが抱える脆弱性を悪用できる可能性があります。

対策

Webサーバー、アプリケーションサーバー、ロードバランサーなどがX-Powered-Byヘッダーを抑制するように設定されていることを確認してください。

リクエスト

GET http://127.0.0.1:62189/contacts/add	
根拠	X-Powered-By: PHP/8.5.11

POST http://127.0.0.1:62189/contacts/add	
根拠	X-Powered-By: PHP/8.5.11

参考

- <https://www.zaproxy.org/docs/alerts/10037/>
- https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/08-Fingerprint_Web_Application_Framework
- <https://www.troyhunt.com/shhh-dont-let-your-response-headers/>

3. 情報

3-1 セッション管理レスポンスの特定

情報

アラート ID	10112
リスク	情報
確度	中
リクエスト	1

指定されたレスポンスにセッション管理トークンが含まれていることが識別されました。「追加情報」フィールドには、ヘッダーベースセッション管理方式で使用可能なヘッダートークンのセットが含まれています。このリクエストがセッション管理方式を「自動検出」に設定したコンテキスト内にある場合、このルールによって識別されたトークンを利用したセッション管理方式に変更されます。

対策

これは脆弱性ではなく情報提供のアラートであるため、修正すべき点はありません。

リクエスト

GET http://127.0.0.1:62189/ contacts/add	
パラメーター	CAKEPHP
根拠	CAKEPHP
その他の情報	cookie: CAKEPHP

参考

- <https://www.zaproxy.org/docs/alerts/10112/>
- <https://www.zaproxy.org/docs/desktop/addons/authentication-helper/session-mgmt-id/>

3. 情報

3-2 ユーザーが制御可能なHTML要素属性 (XSSの可能性)

情報

アラート ID	10031
リスク	情報
確度	低
CWE	CWE-20
WASC	WASC-20
リクエスト	3

このチェックは、クエリ文字列パラメーターやPOSTデータに含まれるユーザー提供の入力を調べて、特定のHTML属性値が制御される可能性のある場所を特定します。これは、XSS (クロスサイトスクリプティング) のホットスポット検出を提供します。悪用可能性を判断するには、セキュリティアナリストによるさらなるレビューが必要です。

対策

HTML属性に書き込む前に、すべての入力を検証し、出力をサニタイズ (無害化) してください。

リクエスト

POST http://127.0.0.1:62189/contacts/add	
パラメーター	_method
その他の情報	ユーザーが制御可能なHTML属性値が見つかりました。特殊文字を挿入して、XSSが可能かどうかを試してください。 以下のURLのページ : http://host.docker.internal:62189/contacts/add が、ユーザー入力を以下に含んでいるようです : [form] タグの [method] 属性 見つかったユーザー入力 : _method=POST ユーザーが制御する値 : post

POST http://127.0.0.1:62189/contacts/add	
パラメーター	data[_Token][fields]
その他の情報	ユーザーが制御可能なHTML属性値が見つかりました。特殊文字を挿入して、XSSが可能かどうかを試してください。 以下のURLのページ： http://host.docker.internal:62189/contacts/add が、ユーザー入力を以下に含んでいるようです： [input] タグの [value] 属性 見つかったユーザー入力： data[_Token][fields]=7b6fbe16e9adac1e320c118cd2348cb3362680ec%3A ユーザーが制御する値： 7b6fbe16e9adac1e320c118cd2348cb3362680ec%3a

POST http://127.0.0.1:62189/contacts/add	
パラメーター	data[Contact][message]
その他の情報	ユーザーが制御可能なHTML属性値が見つかりました。特殊文字を挿入して、XSSが可能かどうかを試してください。 以下のURLのページ： http://host.docker.internal:62189/contacts/add が、ユーザー入力を以下に含んでいるようです： [input] タグの [value] 属性 見つかったユーザー入力： data[Contact][message]=Hello ユーザーが制御する値： hello

参考

- <https://www.zaproxy.org/docs/alerts/10031/>
- https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html