

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	17 アラート	0 高	4 中	10 低	3 情報
-----	------------	--------	--------	---------	---------

対象

検査した URL	http://127.0.0.1:56557
サーバー	PHP の組み込みウェブサーバー (APP_ENV=test)
検査	OWASP ZAP のベースラインスキャン：クロール (1 分) とレスポンスの確認
ログイン	なし (訪問者として)
アラートのある URL	7
不合格の基準	中以上

実行

実行日時	2026-10-17 14:53:11 ~ 14:54:26 (UTC)
所要時間	1分15秒
ソフトウェア	PHP 8.5.11 OWASP ZAP 2.17.0 pieceofcake2/security-dast 1.0.0+no-version-set

ベースラインスキャンは、サイトをクロールしてリクエストとレスポンスを確かめます。攻撃は送らないため、すべての脆弱性を見つけるものではありません。

リスク別のアラート

1. 中

No.	アラート	リスク	リクエスト
1-1	Anti-CSRFトークンが使用されていない	中	2
1-2	Content Security Policy (CSP) ヘッダーが設定されていない	中	5
1-3	クリックジャッキング対策ヘッダーの欠如	中	3
1-4	サブリソース整合性属性の欠落	中	5

2. 低

No.	アラート	リスク	リクエスト
2-1	CookieにHttpOnlyフラグがない	低	2
2-2	Cross-Origin-Embedder-Policy ヘッダーの欠落または無効	低	5
2-3	Cross-Origin-Opener-Policy ヘッダーの欠落または無効	低	5
2-4	Cross-Origin-Resource-Policy ヘッダーの欠落または無効	低	5
2-5	Permissions-Policy ヘッダー未設定	低	5
2-6	SameSite属性のないCookie	低	2
2-7	X-Content-Type-Optionsヘッダーがありません	低	5
2-8	アプリケーションエラーの露見	低	1
2-9	クロスドメインJavaScriptソースファイルのインクルージョン	低	5
2-10	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	5

3. 情報

No.	アラート	リスク	リクエスト
3-1	保存不可能なコンテンツ	情報	1
3-2	保存可能でキャッシュ可能なコンテンツ	情報	5
3-3	認証リクエストが特定されました	情報	1

1. 中

1-1 Anti-CSRFトークンが使用されていない

中

アラート ID	10202
リスク	中
確度	低
CWE	CWE-352
WASC	WASC-9
リクエスト	2

HTML 送信フォームでAnti-CSRFトークンが使用されていませんでした。

CSRF攻撃は被害者の知識や同意なしに標的Webサイトに対してHTTPリクエストを送信させ、被害者の権限で不正な操作を実行させる攻撃手法です。根本原因は、アプリケーションが予測可能なURL/フォームアクションを使用していることにあります。この攻撃の性質は、CSRFがWebサイトがユーザーに対して持つ信頼を悪用することです。対照的に、クロスサイトスクリプティング (XSS) は、ユーザーがWebサイトに対して持つ信頼を悪用します。XSSと同様に、CSRF攻撃は必ずしもクロスサイトである必要はありませんが、クロスサイトになることもあります。クロスサイトリクエストフォージェリはCSRF、XSRF、ワンクリック攻撃、セッションライディング、混乱した代理人 (confused deputy)、シーサーフ (sea surf) としても知られています。

- CSRF攻撃は以下のような状況で効果的です：
- * 被害者が標的サイトでアクティブなセッションを持っている場合
 - * 被害者が標的サイトでHTTP認証により認証されている場合
 - * 被害者が標的サイトと同じローカルネットワーク上にいる場合

CSRFは主に被害者の権限を使用して標的サイトに対してアクションを実行するために使用されてきましたが、レスポンスへのアクセスを獲得することで情報を開示する最近の技術が発見されています。標的サイトがXSSに脆弱である場合、情報開示のリスクは劇的に増加します。なぜなら、XSSはCSRFのプラットフォームとして使用でき、同一オリジンポリシーの範囲内で攻撃を動作させることができるからです。

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。

たとえば、OWASP CSRFGuard などのAnti-CSRFパッケージを使用します。

フェーズ：実装

XSSの脆弱性が存在する場合、多くのCSRF防御策が無効化される可能性があるため、アプリケーションにクロスサイトスクリプティングの脆弱性がないことを確認してください。

フェーズ: アーキテクチャと設計

各フォームに対して一意のNonceを生成し、Nonceをフォームに配置し、フォームを受け取った際にNonceを検証する。Nonceは予測不可能なものにする (CWE-330)

これはXSSを利用してバイパスされる可能性があることに注意してください。

特に危険な操作を特定してください。ユーザーが危険な操作を実行する際には別の確認リクエストを送信し、ユーザーがその操作を意図したことを確認してください。

これはXSSを利用してバイパスされる可能性があることに注意してください。

ESAPIのセッション管理コントロールを使用してください。
このコントロールにはCSRF対策コンポーネントが含まれています。

状態を変更するリクエストにはGETメソッドを使用しないでください。

フェーズ: 実装

HTTPの Referer ヘッダーをチェックし、リクエストが予期されたページから送信されたものであるかを確認してください。
ただし、この方法はユーザーやプロキシがプライバシー上の理由でRefererの送信を無効にしている場合に、正当な機能を妨げる可能性があります。

リクエスト

GET http://127.0.0.1:56557/login	
根拠	<form method="post" action="/login">
その他の情報	HTML フォーム [Form 1: "email" "password"] に既知のAnti-CSRFトークン [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken, data[_Token][key]] が見つかりませんでした。

POST http://127.0.0.1:56557/login	
根拠	<form method="post" action="/login">
その他の情報	HTML フォーム [Form 1: "email" "password"] に既知のAnti-CSRFトークン [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken, data[_Token][key]] が見つかりませんでした。

参考

- <https://www.zaproxy.org/docs/alerts/10202/>
- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
- <https://cwe.mitre.org/data/definitions/352.html>

1. 中

1-2 Content Security Policy (CSP) ヘッダーが設定されていない

中

アラート ID	10038-1
リスク	中
確度	高
CWE	CWE-693
WASC	WASC-15
リクエスト	5

コンテンツ・セキュリティ・ポリシー (CSP) は、クロスサイト・スクリプティング (XSS) やデータインジェクション攻撃などの特定の攻撃を検出・軽減するのに役立つ追加のセキュリティ層です。これらの攻撃は、データの窃取からサイトの改ざん、マルウェアの配布に至るまで幅広く悪用されます。CSPは標準のHTTPヘッダーを提供し、Webサイト所有者がブラウザに対してそのページでの読み込みを許可するコンテンツの承認済みソース（JavaScript、CSS、HTMLフレーム、フォント、画像、Javaアプレット、ActiveX、音声・動画ファイルなどの埋め込みオブジェクト）を宣言できるようにします。

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、Content-Security-Policyヘッダーを設定するように構成されていることを確認してください。

リクエスト

GET	http://127.0.0.1:56557/
GET	http://127.0.0.1:56557/error
GET	http://127.0.0.1:56557/login
GET	http://127.0.0.1:56557/robots.txt
GET	http://127.0.0.1:56557/sitemap.xml

参考

- <https://www.zaproxy.org/docs/alerts/10038/>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CSP>
- https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
- <https://www.w3.org/TR/CSP/>
- <https://w3c.github.io/webappsec-csp/>
- <https://web.dev/articles/csp>
- <https://caniuse.com/#feat=contentsecuritypolicy>
- <https://content-security-policy.com/>

1. 中

1-3 クリックジャッキング対策ヘッダーの欠如

中

アラート ID	10020-1
リスク	中
確度	中
CWE	CWE-1021
WASC	WASC-15
リクエスト	3

このレスポンスは「クリックジャッキング」攻撃から保護されていません。Content-Security-Policy の 'frame-ancestors' ディレクティブ、または X-Frame-Options のいずれかを含める必要があります。

対策

モダンWebブラウザは Content-Security-Policy および X-Frame-Options HTTP ヘッダーをサポートしています。サイト/アプリから返されるすべてのWebページに、これらのいずれかが設定されていることを確認してください。ページが自身のサーバー上のページからのみフレーム表示されることを想定している場合（例: FRAMESETの一部である場合）は SAMEORIGIN を使用し、一切のフレーム表示を許可しない場合は DENY を使用します。または、Content Security Policy の "frame-ancestors" ディレクティブの実装を検討してください。

リクエスト

GET http://127.0.0.1:56557/	
パラメーター	x-frame-options

GET http://127.0.0.1:56557/login	
パラメーター	x-frame-options

GET http://127.0.0.1:56557/robots.txt	
パラメーター	x-frame-options

参考

- <https://www.zaproxy.org/docs/alerts/10020/>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/X-Frame-Options>

1. 中

1-4 サブリソース整合性属性の欠落

中

アラート ID	90003
リスク	中
確度	高
CWE	CWE-345
WASC	WASC-15
リクエスト	5

外部サーバーから提供されるスクリプトまたはリンクタグに、integrity 属性がありません。この属性は、攻撃者がこのサーバーへのアクセスを獲得した場合に、悪意のあるコンテンツを注入するのを防ぎます。

対策

タグに有効な integrity 属性を追加してください。

リクエスト

GET	http://127.0.0.1:56557/
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET	http://127.0.0.1:56557/
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET	http://127.0.0.1:56557/posts?page=1
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET	http://127.0.0.1:56557/robots.txt
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET	http://127.0.0.1:56557/sitemap.xml
根拠	<script src="http://cdn.example.com/lib.js"></script>

参考

- <https://www.zaproxy.org/docs/alerts/90003/>
- https://developer.mozilla.org/en-US/docs/Web/Security/Defenses/Subresource_Integrity

2. 低

2-1 CookieにHttpOnlyフラグがない

低

アラート ID	10010
リスク	低
確度	中
CWE	CWE-1004
WASC	WASC-13
リクエスト	2

CookieがHttpOnlyフラグなしで設定されています。この設定により、CookieはJavaScriptから読み取り可能となります。XSS (クロスサイトスクリプティング) 攻撃などで悪意のあるスクリプトがページ上で実行された場合、攻撃者はCookieの値を取得して外部サーバーに送信できます。特にセッションCookieの場合、セッションハイジャック攻撃により、攻撃者がユーザーのセッションを乗っ取る可能性があります。

対策

すべてのCookieにHttpOnlyフラグが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://127.0.0.1:56557/login	
パラメーター	session
根拠	Set-Cookie: session

POST http://127.0.0.1:56557/login	
パラメーター	session
根拠	Set-Cookie: session

参考

- <https://www.zaproxy.org/docs/alerts/10010/>
- <https://owasp.org/www-community/HttpOnly>

2. 低

2-2 Cross-Origin-Embedder-Policy ヘッダーの欠落または無効

低

アラート ID	90004-2
リスク	低
確度	中
CWE	CWE-693
WASC	WASC-14
リクエスト	5

Cross-Origin-Embedder-Policy ヘッダーは、ドキュメントが明示的に許可されていないクロスオリジンリソース (CORP または CORS を使用) を読み込むのを防ぐレスポンスヘッダーです。

対策

Webサーバー/アプリケーションで Cross-Origin-Embedder-Policy ヘッダーを 'require-corp' に設定してください。この設定により、クロスオリジンリソースの埋め込みにCORP (Cross-Origin-Resource-Policy) ヘッダーによる明示的許可が必要となります。このヘッダーをサポートする標準準拠のモダンブラウザでの利用を推奨します (https://caniuse.com/mdn-http_headers_cross-origin-embedder-policy)。

リクエスト

GET	http://127.0.0.1:56557/
パラメーター	Cross-Origin-Embedder-Policy

GET	http://127.0.0.1:56557/
パラメーター	Cross-Origin-Embedder-Policy

GET	http://127.0.0.1:56557/login
パラメーター	Cross-Origin-Embedder-Policy

GET	http://127.0.0.1:56557/robots.txt
パラメーター	Cross-Origin-Embedder-Policy

GET	http://127.0.0.1:56557/sitemap.xml
パラメーター	Cross-Origin-Embedder-Policy

参考

- <https://www.zaproxy.org/docs/alerts/90004/>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Cross-Origin-Embedder-Policy>

2. 低

2-3 Cross-Origin-Opener-Policy ヘッダーの欠落または無効

低

アラート ID	90004-3
リスク	低
確度	中
CWE	CWE-693
WASC	WASC-14
リクエスト	5

Cross-Origin-Opener-Policy ヘッダーは、サイトが、他の含まれたドキュメントと同一のブラウジングコンテキストを共有するかどうかを制御するためのレスポンスヘッダーです。信頼できないドキュメントと同一のブラウジングコンテキストを共有すると、データ漏えいにつながる可能性があります。

対策

Webサーバー/アプリケーションで Cross-Origin-Opener-Policy ヘッダーを 'same-origin' に設定してください。この設定により、異なるオリジンからのウィンドウ参照が遮断され、サイドチャネル攻撃からの保護が向上します。セキュリティレベルの低い 'same-origin-allow-popups' は使用を避け、このヘッダーをサポートする標準準拠のモダンブラウザでの利用を推奨します (https://caniuse.com/mdn-http_headers_cross-origin-opener-policy)。

リクエスト

GET	http://127.0.0.1:56557/
パラメーター	Cross-Origin-Opener-Policy

GET	http://127.0.0.1:56557/
パラメーター	Cross-Origin-Opener-Policy

GET	http://127.0.0.1:56557/login
パラメーター	Cross-Origin-Opener-Policy

GET	http://127.0.0.1:56557/robots.txt
パラメーター	Cross-Origin-Opener-Policy

GET	http://127.0.0.1:56557/sitemap.xml
パラメーター	Cross-Origin-Opener-Policy

参考

- <https://www.zaproxy.org/docs/alerts/90004/>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Cross-Origin-Opener-Policy>

2. 低

2-4 Cross-Origin-Resource-Policy ヘッダーの欠落または無効

低

アラート ID	90004-1
リスク	低
確度	中
CWE	CWE-693
WASC	WASC-14
リクエスト	5

Cross-Origin-Resource-Policy ヘッダーは、Spectre のようなサイドチャネル攻撃に対抗するために設計されたオプトインのヘッダーです。リソースは、異なるオリジン間で共有可能であることを明示的に設定する必要があります。

対策

Webサーバー/アプリケーションで Cross-Origin-Resource-Policy ヘッダーを適切に設定してください。すべてのWebページには最も安全な 'same-origin' を設定し、CDNや共有リソースなど他オリジンからの読み込みが必要な場合のみ 'cross-origin' を使用してください。セキュリティレベルの低い 'same-site' は使用を避け、このヘッダーをサポートする標準準拠のモダンブラウザでの利用を推奨します (https://caniuse.com/mdn-http_headers_cross-origin-resource-policy)。

リクエスト

GET	http://127.0.0.1:56557/
パラメーター	Cross-Origin-Resource-Policy

GET	http://127.0.0.1:56557/
パラメーター	Cross-Origin-Resource-Policy

GET	http://127.0.0.1:56557/login
パラメーター	Cross-Origin-Resource-Policy

GET	http://127.0.0.1:56557/robots.txt
パラメーター	Cross-Origin-Resource-Policy

GET	http://127.0.0.1:56557/sitemap.xml
パラメーター	Cross-Origin-Resource-Policy

参考

- <https://www.zaproxy.org/docs/alerts/90004/>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Cross-Origin-Embedder-Policy>

2. 低

2-5 Permissions-Policy ヘッダー未設定

低

アラート ID	10063-1
リスク	低
確度	中
CWE	CWE-693
WASC	WASC-15
リクエスト	5

Permissions Policy Headerは、WebページがアクセスできるブラウザAPI機能を制限することで、ユーザーのプライバシーとセキュリティを保護する仕組みです。このポリシーにより、Webサイト管理者は、カメラ、マイクロフォン、位置情報、フルスクリーン表示などの機能への無許可アクセスを防ぐことができます。標準HTTPヘッダーを通じて、ページおよび埋め込みコンテンツが使用可能な機能を細かく制御できます。

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、Permissions-Policy ヘッダーを設定するように構成されていることを確認してください。

リクエスト

GET	http://127.0.0.1:56557/
GET	http://127.0.0.1:56557/
GET	http://127.0.0.1:56557/error
GET	http://127.0.0.1:56557/login
GET	http://127.0.0.1:56557/sitemap.xml

参考

- <https://www.zaproxy.org/docs/alerts/10063/>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Permissions-Policy>
- <https://developer.chrome.com/blog/feature-policy/>
- <https://scotthelme.co.uk/a-new-security-header-feature-policy/>
- <https://w3c.github.io/webappsec-feature-policy/>
- <https://www.smashingmagazine.com/2018/12/feature-policy/>

2. 低

2-6 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	2

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://127.0.0.1:56557/login	
パラメーター	session
根拠	Set-Cookie: session

POST http://127.0.0.1:56557/login	
パラメーター	session
根拠	Set-Cookie: session

参考

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

2. 低

2-7 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	5

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://127.0.0.1:56557/	
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://127.0.0.1:56557/	
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://127.0.0.1:56557/login	
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。

GET http://127.0.0.1:56557/robots.txt	
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://127.0.0.1:56557/sitemap.xml	
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。

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

2-8 アプリケーションエラーの露見

低

アラート ID	90022
リスク	低
確度	中
CWE	CWE-550
WASC	WASC-13
リクエスト	1

このページには、未処理例外を発生させたファイルの場所などの機密情報を開示する可能性があるエラー/警告メッセージが含まれています。この情報は攻撃者によってWebアプリケーションへのさらなる攻撃に悪用される可能性があります。エラーメッセージがドキュメントページ内に記載されている場合、このアラートは誤検知である可能性があります。

対策

このページのソースコードを確認してください。カスタムエラーページを実装してください。エラーの詳細情報はサーバー側でログに記録してユーザーには開示しないようにしつつ、クライアント（ブラウザ）には一意のエラー参照ID / 識別子を提供する仕組みの実装を検討してください。

リクエスト

GET http://127.0.0.1:56557/error	
根拠	HTTP/1.1 500 Internal Server Error

参考

- <https://www.zaproxy.org/docs/alerts/90022/>

2. 低

2-9 クロスドメインJavaScriptソースファイルのインクルージョン

低

アラート ID	10017
リスク	低
確度	中
CWE	CWE-829
WASC	WASC-15
リクエスト	5

このページには、1つ以上のサードパーティドメインからのスクリプトファイルが含まれています。

対策

JavaScriptソースファイルは、信頼できるソースからのみ読み込まれ、かつ、そのソースがアプリケーションのエンドユーザーによって制御されないことを確認してください。

リクエスト

GET http://127.0.0.1:56557/	
パラメーター	http://cdn.example.com/lib.js
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET http://127.0.0.1:56557/	
パラメーター	http://cdn.example.com/lib.js
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET http://127.0.0.1:56557/posts?page=1	
パラメーター	http://cdn.example.com/lib.js
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET http://127.0.0.1:56557/robots.txt	
パラメーター	http://cdn.example.com/lib.js
根拠	<script src="http://cdn.example.com/lib.js"></script>
GET http://127.0.0.1:56557/sitemap.xml	
パラメーター	http://cdn.example.com/lib.js
根拠	<script src="http://cdn.example.com/lib.js"></script>

参考

- <https://www.zaproxy.org/docs/alerts/10017/>

2. 低

2-10 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	5

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://127.0.0.1:56557/
根拠	X-Powered-By: PHP/8.4

GET	http://127.0.0.1:56557/
根拠	X-Powered-By: PHP/8.4

GET	http://127.0.0.1:56557/error
根拠	X-Powered-By: PHP/8.5.11

GET	http://127.0.0.1:56557/login
根拠	X-Powered-By: PHP/8.5.11

GET	http://127.0.0.1:56557/sitemap.xml
根拠	X-Powered-By: PHP/8.4

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 保存不可能なコンテンツ

情報

アラート ID	10049-1
リスク	情報
確度	中
CWE	CWE-524
WASC	WASC-13
リクエスト	1

レスポンスの内容は、プロキシサーバーなどのキャッシュコンポーネントによって保存されません。レスポンスに機密情報、個人情報、ユーザー固有の情報が含まれていない場合は、保存およびキャッシュすることでパフォーマンスの向上が期待できます。

対策

コンテンツがキャッシュ可能と判定されるための条件：

基本条件 (すべて必須)：

- * キャッシュ可能なHTTPメソッド (GET、HEAD、POST)
- * 認識可能なステータスコード (1XX～5XXの各クラス)
- * no-storeディレクティブの不存在 (リクエスト・レスポンス共に)

共有キャッシュ (プロキシキャッシュ等) での追加条件：

- * private ディレクティブの不存在
- * Authorization ヘッダーの不存在 (ただし、must-revalidate、public、s-maxage の指定がある場合は例外)

有効期限条件 (いずれか1つ以上)：

- * Expires ヘッダーの存在
- * max-age ディレクティブの存在
- * s-maxage ディレクティブの存在 (共有キャッシュ用)
- * キャッシュを許可する Cache Control Extension の存在
- * デフォルトキャッシュ可能ステータスコード (200、203、204、206、300、301、404、405、410、414、501)

リクエスト

GET http://127.0.0.1:56557/error	
根拠	500

参考

- <https://www.zaproxy.org/docs/alerts/10049/>
- <https://datatracker.ietf.org/doc/html/rfc7234>
- <https://datatracker.ietf.org/doc/html/rfc7231>
- <https://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html>

3. 情報

3-2 保存可能でキャッシュ可能なコンテンツ

情報

アラート ID	10049-3
リスク	情報
確度	中
CWE	CWE-524
WASC	WASC-13
リクエスト	5

レスポンスが共有キャッシュサーバー (プロキシキャッシュ等) に保存可能な状態です。機密情報、個人データ、またはセッション固有の情報が含まれている場合、他のユーザーからの同様のリクエストに対してキャッシュから配信され、情報漏洩やセッション乗っ取りが発生する可能性があります。この問題は主に、企業や教育機関などの共有プロキシキャッシュサーバーが設置されたネットワーク環境で発生します。

対策

レスポンスに機密情報、個人情報、またはユーザー固有の情報が含まれていないか検証してください。機密情報が含まれている場合は、以下のHTTPレスポンスヘッダーを設定して、他のユーザーによるキャッシュからのアクセスを防止してください：Cache-Control: no-cache, no-store, must-revalidate, private

Pragma: no-cache

Expires: 0

この設定は、HTTP/1.0およびHTTP/1.1準拠のキャッシュサーバーに対して、レスポンスをキャッシュに保存せず、同様のリクエストに対してもキャッシュからの配信を行わないよう指示します。

リクエスト

GET http://127.0.0.1:56557/	
その他の情報	レスポンスにキャッシュの有効期間を明示する指示がない場合、1年という寛容な期間のヒューリスティックが仮定されます。これは RFC 7234 により許可されています。

GET http://127.0.0.1:56557/	
その他の情報	レスポンスにキャッシュの有効期間を明示する指示がない場合、1年という寛容な期間のヒューリスティックが仮定されます。これは RFC 7234 により許可されています。

GET http://127.0.0.1:56557/login	
その他の情報	レスポンスにキャッシュの有効期間を明示する指示がない場合、1年という寛容な期間のヒューリスティックが仮定されます。これは RFC 7234 により許可されています。

GET http://127.0.0.1:56557/robots.txt	
その他の情報	レスポンスにキャッシュの有効期間を明示する指示がない場合、1年という寛容な期間のヒューリスティックが仮定されます。これは RFC 7234 により許可されています。

GET http://127.0.0.1:56557/sitemap.xml	
その他の情報	レスポンスにキャッシュの有効期間を明示する指示がない場合、1年という寛容な期間のヒューリスティックが仮定されます。これは RFC 7234 により許可されています。

参考

- <https://www.zaproxy.org/docs/alerts/10049/>
- <https://datatracker.ietf.org/doc/html/rfc7234>
- <https://datatracker.ietf.org/doc/html/rfc7231>
- <https://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html>

3. 情報

3-3 認証リクエストが特定されました

情報

アラート ID	10111
リスク	情報
確度	高
リクエスト	1

指定されたリクエストが認証リクエストとして識別されました。「追加情報」フィールドには、関連するフィールドを特定する key=value 形式の行セットが含まれています。このリクエストが認証方法を「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リクエストに一致するように認証設定が変更されます。

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

POST http://127.0.0.1:56557/login	
パラメーター	email
根拠	password
その他の情報	userParam=email userValue=zaproxy@example.com passwordParam=password referer=http://host.docker.internal:56557/login

参考

- <https://www.zaproxy.org/docs/alerts/10111/>
- <https://www.zaproxy.org/docs/desktop/addons/authentication-helper/auth-reg-id/>