

第三者の目線で  
サイバー攻撃の脅威を未然に防ぐ

# 脆弱性診断サービス

Sun\* Cyber Security Test

# Who we are?



当サイバーセキュリティチームは、Sun\*のR&D部門を母体に2019年5月に発足しました。最新の攻撃手法に精通したセキュリティ・スペシャリストが、Webアプリケーション脆弱性診断やペネトレーションテストを通じて、企業のシステムに潜むリスクを可視化し、被害の未然防止を支援します。

私たちは、攻撃者の視点で弱点を見つけ出すだけでなく、改善につながる実践的な提案まで行うことで、企業の重要なデジタル資産を守ります。

加えて、トレーニングやセミナーを通じてセキュリティ意識の向上にも取り組み、信頼できるパートナーとして継続的な支援を提供しています。



# 複雑化するセキュリティ課題に対し、まず必要なのは“弱点の把握”

## 【外部環境】広範囲化する脅威

IPA「情報セキュリティ10大脅威 2026[組織編]」より



ランサム攻撃



サプライチェーン攻撃



脆弱性悪用・標的型攻撃



AI利用リスク

## 【内部の起点】被害の起点となる弱点



脆弱性の放置



認証不備／設定不備



メールや Web 経由の侵入



委託先の管理不足



対策の第一歩: 自社のシステムやWebアプリケーションに  
存在するリスクを可視化し、優先順位をもって改善する

## Sun\*のソリューション: 脆弱性診断によるリスクの可視化と優先順位付け



### 攻撃者視点の点検

Webアプリケーションやシステムに対し、  
攻撃者と同じ視点から技術的なリスクを  
徹底点検。



### 潜在的リスクの可視化

単なる検出にとどまらず、放置されている  
潜在的な脆弱性(弱点)を明確に可視化し、  
実効性ある対策の基盤を構築。



### 影響度の整理と優先順位付け

リスクの重要性や想定される影響を整理。  
迅速かつ効果的な対応につながる「優先順位付  
け」の情報を提供。

# セキュリティ診断サービス

脆弱性診断とペネトレーションテストの結果をもとに、リスク評価と改善支援を通じて、セキュリティリスクの低減を支援します

## 診断対象



Webアプリケーション診断



スマートフォンアプリ診断



プラットフォーム  
／ネットワーク診断



クラウド設定監査



## 脆弱性診断

脆弱性の特定、評価、再現確認

攻撃者に悪用される可能性のある脆弱性を洗い出し、対策が必要な弱点を明確にします



## 脆弱性評価

CVSS等に基づくリスク評価

検出した脆弱性を深刻度や影響度の観点から評価し、対応優先度を整理します



## ペネトレーションテスト

攻撃シナリオに基づく侵入・影響検証

検出された脆弱性に対して、攻撃シナリオに基づく悪用可能性の検証 (PoC) を実施し、実際の影響範囲を確認します



## 開発チームへの改善支援

具体的な改善策の提示・実装支援

診断結果を踏まえ、開発チームに対して具体的な対策案を提示し、改善の実行を支援します

# 診断メニューの比較



## 共通アプローチ

ツール診断と手動診断を組み合わせ、Webアプリケーションからクラウド設定までを横断的に評価。  
Sun\*は、診断で見つけるだけでなく、開発・インフラの知見を活かして、改修方針の提示から改善まで一気通貫で支援します。

|                 | Webアプリケーション<br>脆弱性診断            | モバイルアプリケーション<br>脆弱性診断          | プラットフォーム<br>／ネットワーク脆弱性診断   | クラウド<br>設定監査                | フルペネトレーション<br>テスト                                     |
|-----------------|---------------------------------|--------------------------------|----------------------------|-----------------------------|-------------------------------------------------------|
| 主な目的            | Webアプリ層の脆弱性特定と<br>悪用可能性確認       | モバイルアプリの脆弱性特<br>定と悪用可能性確認      | サーバー・NW機器・公開サー<br>ビスの脆弱性特定 | クラウド環境の設定不備・権限<br>不備の確認     | 攻撃者視点で侵入から影響拡大<br>までを検証                               |
| 対象領域            | Webアプリケーション層                    | アプリ本体、通信、保存<br>データ、認証周り        | OS、ミドルウェア、NW機器、公<br>開サービス  | IAM、ネットワーク設定、公開<br>設定、監査設定等 | Web、アプリ、OS、AD、NW、クラ<br>ウドを横断可能                        |
| 侵入・悪用可能<br>性の確認 | あり<br>一部、ペネトレーションテスト<br>初期段階に相当 | あり<br>一部、対象領域内での悪用<br>可能性確認を含む | 一部あり<br>限定的な侵入口確認を含む       | 原則なし<br>監査・評価が中心            | あり<br>侵入、権限奪取、横展開、重要資<br>産到達まで含む<br>※従業員に対する標的型メールを除く |
| 権限奪取<br>横展開     | 通常は対象外                          | 通常は対象外                         | 通常は対象外                     | 対象外                         | 対象                                                    |
| 向いている<br>ケース    | Webの弱点把握                        | モバイルアプリの弱点把握                   | インフラの弱点把握                  | クラウド設定の健全性確認                | 実被害リスク・侵害範囲の把握                                        |

# チームの専門性を支える保有資格・認定

Sun\*のサイバーセキュリティチームは、各領域に対応する実践的かつ高度なセキュリティ資格・認定を保有しています。

## 侵入診断 ペネトレーションテスト



Offensive Security Certified Professional



EC-Council Certificated Licensed  
Penetration Tester Master



EC-Council Certified Security Analyst

## Web アプリケーション診断



Offensive Security Web Expert



EC-Council Certified Security Analyst



Burp Suite Certified Practitioner

## 高度攻撃検証 フォレンジック



Offensive Security Certified Expert



Certified Hacking Forensic Investigator

## クラウド・プラットフォーム／ネット ワーク



Certified Network  
Security Specialist



AWS  
Advanced Tier

Microsoft  
Solution Partners

Microsoft  
Solution Partner

# 私たちが準拠する基準とフレームワーク

世界的に採用されているセキュリティ基準やフレームワークに準拠し、診断サービスを提供しています。

## 診断・評価の基準



### OWASP

Webアプリケーションセキュリティの代表的なガイドラインとベストプラクティス



### OSSTMM

ペネトレーションテストやセキュリティ評価に用いる代表的な方法論

## 準拠・対応を支援する基準



### PCI-DSS

クレジットカード情報を保護するための業界標準セキュリティ基準



### GDPR

EU域内の個人データ保護に関する代表的な規制

# 診断品質を担保する主要ツール

診断の効率化と再現性の確保のため、業界標準の診断ツールを活用しています。

## Webアプリケーション診断

Webアプリの脆弱性確認に用いる代表的な診断ツール



Webアプリ診断の統合テストツール



OWASP  
Zed Attack Proxy

Web脆弱性確認用の OSSツール

## ネットワーク・脆弱性調査

ホストやポート、既知脆弱性を調査するツール



ホスト・ポート・サービス調査ツール



既知脆弱性を検出するスキャナー

## モバイル・バイナリ解析

アプリやバイナリの挙動を解析する動的解析ツール

**FRIDA**

モバイル・バイナリの動的解析ツール

## 設定監査・ハードニング確認

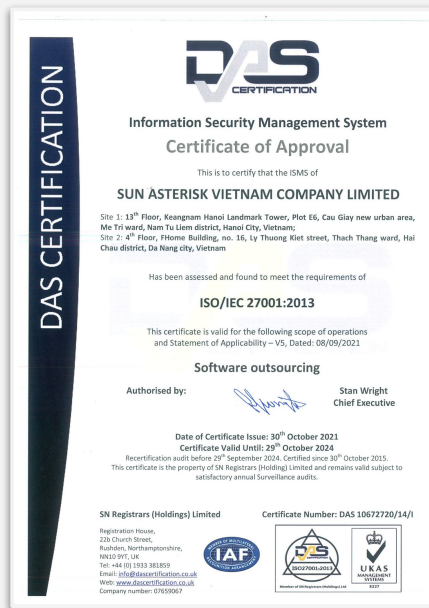
CIS Benchmarksに基づき設定不備を確認する監査ツール



CIS準拠の設定監査ツール

# ISO 9001認証／ISO 27001認証

品質管理および情報セキュリティ管理体制の継続的な改善に取り組み、ISO 9001およびISO 27001の認証を取得しています。



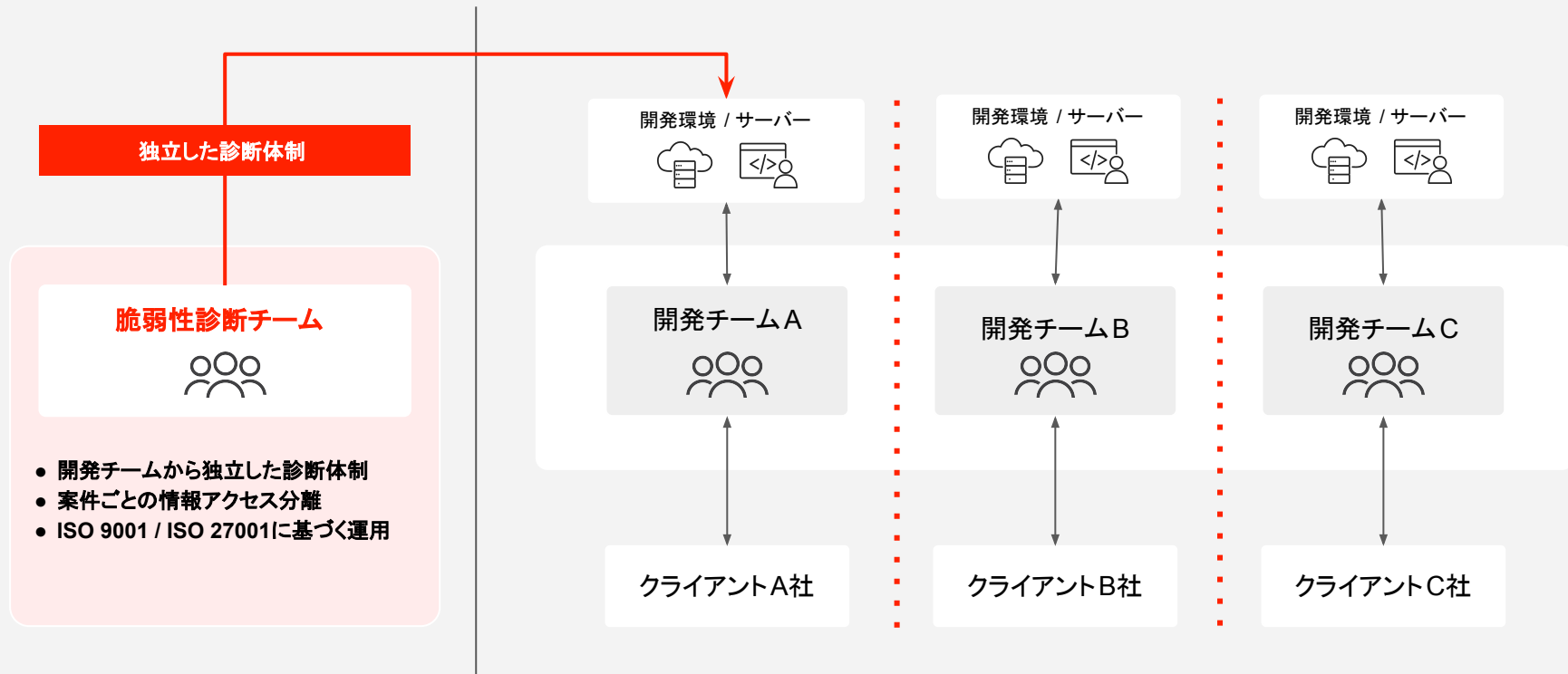
## 品質と情報セキュリティを支える管理体制

Sun\* Vietnamでは、品質管理と情報セキュリティ管理体制の両面で、ISO 9001およびISO 27001の認証を取得しています。

開発品質の向上と、情報の機密性・完全性・可用性を支える管理体制の整備・運用を推進しています。

# 診断チームの独立性

脆弱性診断はR&D部門内で運営されていますが、各クライアントの開発チームとは分離・独立した体制で実施しています。  
また、診断メンバーはISO 9001およびISO 27001に基づく情報管理ルールのもと、担当案件以外の情報にアクセスできない運用となっています。



# 脆弱性診断サービス / プラン・価格一覧



## 共通アプローチ

ツール診断と手動診断を組み合わせ、Webアプリからクラウドまで幅広く評価。

Sun\*は診断だけで終わらず、開発・インフラの知見を活かして、改善提案から対応支援まで伴走します。

|          | Webアプリ診断<br>EC、会員サイト、業務システムの<br>公開前・定期診断に | スマホアプリ診断<br>iOS / Androidアプリのリリース前や<br>大規模改修時に | プラットフォーム &<br>ネットワーク診断<br>サーバ、ミドルウェア、ネットワークの<br>露出リスク確認に | クラウド設定診断<br>AWS / Azure / GCP環境の設定不備や<br>権限過多の確認に |
|----------|-------------------------------------------|------------------------------------------------|----------------------------------------------------------|---------------------------------------------------|
| 参考費用     | 1,500,000 円~                              | 1,500,000 円~                                   | 345,000 円~                                               | 234,000 円~                                        |
| 目安範囲     | 対象40ページ分                                  | 対象40画面相当                                       | 対象5IP分                                                   | 1クラウド環境                                           |
| 主な対象     | Webサービス、業務システム                            | iOS / Androidアプリ                               | サーバ、NW機器、公開基盤                                            | AWS / Azure / GCP                                 |
| 向いているケース | リリース前、定期診断                                | 公開前、アップデート時                                    | 外部公開資産の確認                                                | 設定不備・権限見直し                                        |
| 全プラン共通   | 計画書、速報レポート、最終報告書、再診断、改善指示書                |                                                |                                                          |                                                   |

## 個別設計メニュー

### ペネトレーションテスト: 個別見積

侵入シナリオを設計し、重要資産への到達可能性や実被害につながる経路を重点検証します。

# 主な診断実績

スタートアップから大手・上場企業まで、幅広い業種・サービス領域で継続的な脆弱性診断をご支援しています。



## 金融・Fintech

Fintechサービス / 大手銀行 / 大手証券会社



## Webサービス・SaaS

SaaS系サービス / Webサービス / BtoBサービス



## 大規模サービス

ECサイト 70万ユーザー / 小売流通 130万ユーザー / BPOサービス



## 新規事業・新規サービス

新規モバイルサービス / 人材系新規サービス / 新規サービス



## 基幹・社会インフラ系

社会インフラ系基幹システム / 不動産賃貸サービス / 電気通信新規サービス



## 診断項目

# 主な確認項目

Web/API、スマートフォン、プラットフォーム／ネットワークの各領域について、主要な脆弱性や設定不備、不正利用リスクを診断します。

| 診断カテゴリ            | 主な診断項目                                                       |
|-------------------|--------------------------------------------------------------|
| Web/APIアプリ診断      | 認証・認可不備、セッション管理、XSS、SQLインジェクション等の各種インジェクション、情報漏えい、ビジネスロジック不備 |
| スマートフォンアプリ診断      | アプリ設定不備、認証・セッション管理、端末内データ保護、通信内容の保護、WebViewの脆弱性、アプリ機能の不正利用   |
| プラットフォーム／ネットワーク診断 | 不正ログイン、権限管理不備、不要サービス・設定不備、既知脆弱性、ライブラリ／ミドルウェアの脆弱性、クラウド設定監査    |

# OWASP Testing Guide と ASVS に基づく診断

Webアプリケーション診断では、実践的な診断ガイドとセキュリティ要件標準の両方を活用しています。



# 脆弱性評価基準

診断で検出した脆弱性は、CVSS v3.1 を基準に、Critical / High / Medium / Low の深刻度で評価しています。

|                                                                                                   | 深刻度        | 修正対応 |
|---------------------------------------------------------------------------------------------------|------------|------|
|  <b>Critical</b> | 9.0 - 10.0 | 必須   |
|  <b>High</b>     | 7.0 - 8.9  | 必須   |
|  <b>Medium</b>   | 4.0 - 6.9  | 必須   |
|  <b>Low</b>      | 0.1 - 3.9  | 推奨   |
|  <b>None</b>     | 0.0        | 不用   |

参考 FIRST: [CVSSの定義](#) IPA: [CVSS v3 解説](#)

## CVSSとは？

攻撃のしやすさ、必要権限、利用者操作の有無、機密性・完全性・可用性への影響などをもとに、脆弱性の深刻度を評価する共通指標です。

深刻度の客観評価

対応優先度の整理

改善計画の判断材料

共通脆弱性評価システム  
(Common Vulnerability Scoring System) Version 3.1 Calculator  
脆弱なコンポーネントとユーザエージェントに基づいています。

| 攻撃経路<br>ATTACK VECTOR (AV) | 攻撃条件の複雑さ<br>ATTACK COMPLEXITY (AC) | 攻撃に必要な特権レベル<br>PRIVILEGES REQUIRED (PR) | 利用者の関与<br>USER INTERACTION (UI) |
|----------------------------|------------------------------------|-----------------------------------------|---------------------------------|
| ネットワーク (N)                 | 低 (L)                              | 不要 (N)                                  | 不要 (N)                          |
| 間接ネットワーク (A)               | 高 (H)                              | 低 (L)                                   | 必要 (R)                          |
| ローカル (L)                   |                                    | 高 (H)                                   |                                 |
| 物理 (P)                     |                                    |                                         |                                 |
| 影響に及ぼす範囲<br>SCOPE (S)      | 機密性への影響<br>CONFIDENTIALITY (C)     | 完全性への影響<br>INTEGRITY (I)                | 可用性への影響<br>AVAILABILITY (A)     |
| 変更あり (C)                   | 高 (H)                              | 高 (H)                                   | 高 (H)                           |
| 変更なし (N)                   | 低 (L)                              | 低 (L)                                   | 低 (L)                           |
| なし (N)                     | なし (N)                             | なし (N)                                  | なし (N)                          |

算出された結果

|      |                                             |
|------|---------------------------------------------|
| 8.2  | CVSS3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:L/A:L |
| High |                                             |

攻撃手法として最も多いとされているクロスサイトスクリプティング(XSS)など、クライアント側から行われる攻撃に対する診断を行います。

|                      |                                                                                                                       |
|----------------------|-----------------------------------------------------------------------------------------------------------------------|
| クロス・サイト・スクリプティング     | サイトをまたがって不正な要求を送り、ユーザが意図していないスクリプトを実行させられてしまいます。その結果、例えば偽ページを表示することが可能になり、フィッシング詐欺などに悪用されてしまいます。                      |
| クロス・サイト・リクエスト・フォージェリ | サイトをまたがって不正な要求を送り、ユーザが意図していない操作を実行させられてしまいます。例えば、ユーザが意図しないままオンラインショップで買い物をさせられたりしてしまいます。                              |
| クロス・サイト・トレーシング       | ウェブのヘッダ情報を不正に読み出されてしまいます。これにより、他の脆弱性を利用して管理者や他のユーザに成りすまされてしまいます。                                                      |
| コンテンツの詐称             | 偽のコンテンツをあたかも正式なものであるかのように装いウェブサイトに表示し、ユーザを欺きます。これにより、パスワードを抜き取られたり、フィッシング詐欺サイトへ誘導されたりする危険性があります。                      |
| セキュアでない cookie の使用   | セッション・クッキーが類推可能な簡単なものであったり、セキュアでない通信経路で送られた過程で盗まれたりすると、セッション自体が盗まれる可能性があります。これにより、個人情報が盗まれたり、コンピュータに侵入されたりする危険性があります。 |
| hiddenフィールドの不正操作     | Hiddenフィールドを不正に改ざんされてしまう可能性があります。例えば商品の価格などをHiddenで受渡ししている場合に、価格を任意に変更されてしまうなどの可能性があります。                              |

複数のサイトを経由し不正な要求を送り、意図しない操作の実行が可能な状態だと、第三者が管理者になりすましサイトを乗っ取られる危険性があります。

|                 |                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------|
| SQLインジェクション     | DBサーバへのアクセスを不正に実行されてしまいます。これにより、ウェブサイトからの情報漏えい、改ざん等の危険性があります。                                          |
| SSIインジェクション     | SSIコマンドを不正に実行されてしまいます。これにより、ウェブサイトからの情報漏えい、改ざん等の危険性があります。                                              |
| メールヘッダインジェクション  | 問い合わせフォームなどのメールを送信する画面で、メールの内容を改ざんし、迷惑メールの送信などに悪用される可能性があります。                                          |
| HTTPヘッダインジェクション | 動的にHTTPヘッダを生成する機能の不備を突いてヘッダ行を挿入することで不正な動作が発生し、セッションハイジャック、XSS等につながる可能性がある攻撃。                           |
| XMLインジェクション     | XMLデータにスクリプト等を混入して攻撃されてしまいます。これにより、ウェブサイトからの情報漏えい、改ざん等の危険性があります。                                       |
| LDAPインジェクション    | LDAPコマンドを不正に使用されてしまいます。これにより、ウェブサイトからの情報漏えい、改ざん等の危険性があります。                                             |
| コマンドインジェクション    | サーバ内のOSのコマンドを不正に実行されてしまいます。これにより、ウェブサイトからの情報漏えい、改ざん等の危険性があります。                                         |
| バッファ・オーバーフロー    | アプリケーションの予期しないデータを送り、アプリケーションを異常終了させられてしまいます。これにより、ウェブサーバのサービスを停止させられたり、ウェブサーバを乗っ取られる危険性があります。         |
| ディクショナリアタック     | 入力された文字列を書式加工する際にプログラムをクラッシュさせたり、不正なコードを実行させられてしまいます。これにより、ウェブサーバのサービスを停止させられたり、ウェブサーバを乗っ取られる危険性があります。 |
| パラメータ改ざん        | パラメータを不正に改ざんされてしまいます。その結果、管理者や他のユーザに成りすまされてしまいます。                                                      |

外部から見れない情報が閲覧できる状態かどうか診断します。サーバから情報が漏えいすると攻撃の糸口になりサイトそのものを乗っ取られる可能性があります。

|                    |                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------|
| ディレクトリ・トラバーサル      | ウェブサーバ内のファイルを閲覧されることにより、ウェブサーバ攻撃の足がかりとされてしまいます。                                                |
| 情報漏洩               | ウェブサーバから意図していない内部情報が外部に漏えいしてしまいます。                                                             |
| 不適切なコメント記載         | HTMLや可読可能なスクリプトファイルへの不適切なコメントが記載されていると、外部から侵入のきっかけを与えています。                                     |
| パスフレーズジャック         | ウェブブラウザのアドレスバーやファイル名を指定するパラメータなどの箇所から任意のパスを受け付けてしまうため、機密情報などが保管されているパスを指定されることにより情報漏えいにつながります。 |
| 推測可能なリソースの位置       | フォルダ名やファイル名が推測可能な簡単な名称になっているなど、内部のリソースの配置が推測可能な場合、重要な情報や機能が外部に漏えいする危険性があります。                   |
| ウェブサーバ・アプリケーションの特定 | ウェブサーバ、ウェブアプリケーションの種類やバージョン情報から脆弱性が探り出され、攻撃の足がかりとされてしまいます。                                     |
| ディレクトリ・リスティング      | サーバ内のディレクトリ情報が外部から閲覧できる状態になっています。機密情報が含まれている場合は直接問題となり、内部のディレクトリの推測等に使用されるなどの間接的な脆弱性にも繋がります。   |
| リクエストヘッダによる情報漏洩    | リクエストヘッダにシステムの内部情報が外部から閲覧できる状態になっています。                                                         |

認証を使用しているウェブサイトで適切に認証がされているか診断を行います。成りすましや情報漏洩の危険性があります。

|             |                                                                                       |
|-------------|---------------------------------------------------------------------------------------|
| 総当たり攻撃      | IDやパスワードが簡単に推測可能であり、管理者や他のユーザに成りすまされてしまいます。                                           |
| 認証関連が不適切    | サーバーのセッション管理等により正常なログイン処理を介せずにログイン後の画面にアクセスされてしまうため、成りすましや情報漏えいの危険性があります。(セッションジャック等) |
| パスワード復元が不適切 | ユーザがパスワードを忘れた際の回復方法に問題があり、パスワードの情報が外部に漏えいします。                                         |

認証後のセッション管理が適切か診断します。推測しやすい場合、アカウントの乗っ取りが可能となり様々な攻撃の糸口として利用される危険な箇所です。

|             |                                                                              |
|-------------|------------------------------------------------------------------------------|
| セッションの推測    | セッション情報が推測しやすい値の場合、攻撃者は正しい値を推測し、管理者やユーザに成りすますことができます。                        |
| 承認が不適切      | 承認が不適切だと、アクセス権限の高いコンテンツや機能へのアクセスを認めてしまいます。これにより、攻撃者が他のユーザや管理者に成りすます危険性があります。 |
| セッション期限が不適切 | セッション期限が不適切である場合、ユーザのセッション情報を盗用しやすくなり、攻撃者が管理者やユーザに成りすますことができます。              |
| セッションの固定    | 攻撃者が任意のセッション情報を使って管理者やユーザに成りすますことができます。                                      |
| セッションの盗難    | SSL等を使用して暗号化をしていない場合、攻撃者はセッション情報を容易に取得することができ、管理者やユーザに成りすますことができます。          |

## ロジックを狙った攻撃

サーバやアプリケーションの機能を狙った攻撃が可能か診断します。有名なDos攻撃等が該当し、サービスの継続が難しくなるだけではなくサイトの乗っ取り、情報漏洩などに繋がる危険性があります。

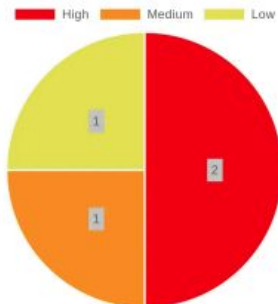
|            |                                                                                       |
|------------|---------------------------------------------------------------------------------------|
| 機能の悪用      | ウェブサーバ、ウェブアプリケーションの持つ機能を不正に実行されてしまいます。その結果、SPAM(メールを大量配信すること)の中継地点に使われるなど、悪用されてしまいます。 |
| サービス拒否     | ウェブサーバのサービスを停止、もしくは低下させられてしまいます。(Dos攻撃)                                               |
| 自動化の停止が不適切 | ロボットなどによるウェブサーバへの連続攻撃を受け、正しいIDやパスワードを探られたり、ウェブサーバに負荷をかけられたりしてしまいます。                   |
| 不適切なプロセス検証 | 正常な画面遷移を無視して特定の画面にアクセスされてしまうため、成りすましや情報漏えいの危険性があります。                                  |



**診断報告書  
サンプル**

# Webアプリケーション診断結果報告書例

|                             | Critical | High | Medium | Low |
|-----------------------------|----------|------|--------|-----|
| http://testphp.vulnweb.com/ | 0        | 2    | 1      | 1   |



| 脆弱性                      | 深刻度    | 脆弱性の数 |
|--------------------------|--------|-------|
| ##### ファイルインクルージョン       | High   | 1     |
| ##### 格納型クロスサイトスクリプティング  | High   | 1     |
| ##### 反射型クロスサイト・スクリプティング | Medium | 5     |
| ##### Cookieのセキュリティ不備    | Low    | 1     |

## 診断結果詳細

### ##### ファイルインクルージョン

深刻度: High

#### CVSSスコア

<https://research.sun-asterisk.com/cvss.html#CVSS3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N>

#### 解説

ファイルインクルードとは、プログラムの中で別ファイルを参照するコードがあった場合に、実際に参照すべきファイルとは別のファイルやデータを読み込ませて、本来意図しない不正なデータ処理を行わせる攻撃です。このウェブサイトはfileの/パラメータを通じて、参照のイメージパス値を取得しています。ただし、不適切な入力処理により、ハッカーがデータを悪用して取得する可能性があります。

#### 該当部分・URL

GET <http://testphp.vulnweb.com/showimage.php?file=x>

#### 影響

影響は、ウェブサーバーのユーザーの実行権限と読み取り権限によって異なる場合があります。想定される脆弱性は以下が挙げられます。

- 機密データの取得
- 機密データの漏れによって、システムの乗っ取り

#### 再現手順

ステップ1:

<http://testphp.vulnweb.com/listproducts.php?cat=1> にアクセスします。

## Request

```
1 GET /showimage.php?file=/pictures/1.jpg HTTP/1.1
2 Host: testphp.vulnweb.com
3 DNT: 1
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
  Edg/131.0.0.0
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
  webp,image/apng,*/*;q=0.8,application/signed-exchange;vmb3;q=0.7
7 Referer: http://testphp.vulnweb.com/listproducts.php?cat=1
8 Accept-Encoding: gzip, deflate, br
9 Accept-Language: vi,en-US;q=0.9,en;q=0.8,nl;q=0.7
10 Connection: keep-alive
11
12
```

## ステップ2

fileのパラメータの値をURLから変更して、別のファイルを読み込むことができます。

例: /proc/version のファイルを読み込んで、システムのバージョンを確認できます。

```
Request
1 GET /showimage.php?file=/proc/version HTTP/1.1
2 Host: testphp.vulnweb.com
3 DNT: 1
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
  Edg/131.0.0.0
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
  webp,image/apng,*/*;q=0.8,application/signed-exchange;vmb3;q=0.7
7 Referer: http://testphp.vulnweb.com/listproducts.php?cat=1
8 Accept-Encoding: gzip, deflate, br
9 Accept-Language: vi,en-US;q=0.9,en;q=0.8,nl;q=0.7
10 Connection: keep-alive
11
12
```

## 対策方法

- 外部ファイルの参照を無効にしてください。
- ファイル・フォルダのアクセス権や権限を見直してください。
- 推測されにくいファイル名やフォルダ構造をおすすめです。
- パスを直接追加することを許可しないでください。動的パスの連結が絶対に必要な場合は、「{a-Z0-9}」などの必要な文字のみを受け入れてください。

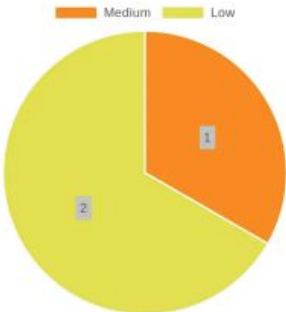


診断レポートのサンプル PDFはこちら

# モバイルアプリケーション診断結果報告書例

## 診断結果の概要

|                                                                         | Critical | High | Medium | Low |
|-------------------------------------------------------------------------|----------|------|--------|-----|
| Androidアプリケーション<br>v1.0.0                                               | 0        | 0    | 1      | 1   |
| <a href="https://wsun-sun-asterisk.vn">https://wsun-sun-asterisk.vn</a> | 0        | 0    | 0      | 1   |



| 脆弱性                               | 深刻度    | 脆弱性の数 |
|-----------------------------------|--------|-------|
| ダッシュボードにおける安全でない直接オブジェクト参照 (IDOR) | Medium | 1     |
| 弱いアカウントロックアウトメカニズム                | Low    | 1     |
| 難読化されないソースコード                     | Low    | 1     |

## 診断結果詳細

ダッシュボードにおける安全でない直接オブジェクト参照 (IDOR)

深刻度: Medium

### CVSSスコア

<https://research.sun-asterisk.com/va/cvss.html#CVSS.3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A/N>

### 解説

アプリのダッシュボードの画面で、権限がない攻撃者は `user_id` のパラメータを変更すると、システムに登録している他のユーザーの個人情報 (メールアドレスや生年月日など) をリクする可能性があります。

### 該当部分・URL

<https://wsun-sun-asterisk.vn/api/dashboard/users/4010>

### 影響

権限がない攻撃者はユーザー列挙攻撃して、システムに登録している全部のユーザーの個人情報が漏れる可能性があります。

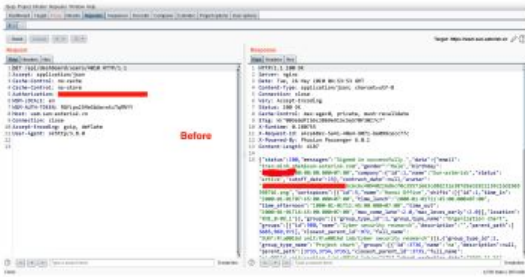
### 再現手順

ステップ1:

例のアカウントにログインして、Burp Suiteを使って、`/api/dashboard/users/{user_id}`へのリクエストをキャプチャします。

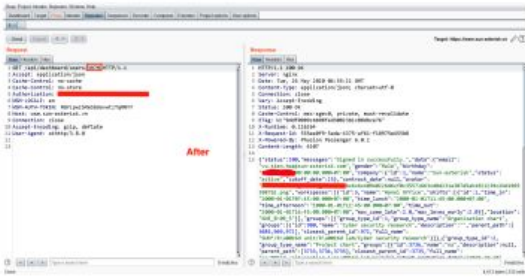


例のアカウント (tran.minh.nhat@sun-asterisk.com)のIDは4010です。



### ステップ2

`user_id`のパラメータを任意のID (例えば、3979) に変更し、リクエストを送信します。サーバーからのレスポンスはID:3979と該当するユーザー (vu.tien.hoa@sun-asterisk.com)の個人情報が表示され、他人の個人情報漏れました。IDを順々に変えれば、システムに登録している全部のユーザーの個人情報を取得できます。



### 対策方法

ダッシュボードを表示するとき、実行するユーザーの権限とパラメータをチェックしてください。

# ネットワーク診断結果報告書例

## 診断結果詳細

### ホストおよびポートによるサービス検出

提供されたIPアドレスに対してネットワーク偵察を実施し、オープンポートを特定しました。各IPアドレスは、NmapやSpartaなどの標準スキャンツールを使用して、すべてのTCPおよびUDPポートがテストされました。以下のポートが特定され、利用可能な脆弱性を持つポートは強調表示されています。

| ホスト       | プロトコル | ポート  | サービス                    | バージョン                                 |
|-----------|-------|------|-------------------------|---------------------------------------|
| 10.0.2.7  | tcp   | 22   | SSH                     | OpenSSH 5.3 (プロトコル 2.0)               |
|           | tcp   | 4848 | Oracle                  | Oracle Glassfish server 4.0           |
|           | tcp   | 443  | Nginx<br>Laravel<br>PHP | nginx 1.12.2                          |
|           | tcp   | 443  | Nginx<br>Laravel<br>PHP | Laravel<br>nginx 1.12.2<br>PHP 7.1.13 |
|           | tcp   | 443  | Nginx<br>Laravel<br>PHP | Laravel<br>nginx 1.12.2<br>PHP 7.1.13 |
| 10.0.2.8  | tcp   | 22   | SSH                     | OpenSSH 5.3 (プロトコル 2.0)               |
|           | tcp   | 80   | Nginx                   | nginx 1.12.2                          |
|           | tcp   | 443  | Apache                  | Apache Solr 6.5.1                     |
|           | tcp   | 8000 | Nginx                   | nginx 1.12.2                          |
| 10.0.2.12 | tcp   | 22   | SSH                     | OpenSSH 5.3 (protocol 2.0)            |
|           | tcp   | 80   | HTTP                    | PRTG Network Monitor                  |
|           | tcp   | 443  | Nginx                   | nginx 1.12.2                          |
|           | tcp   | 8000 | Apache                  | apache 10.1.0                         |
|           | tcp   | 9100 | Nginx                   | nginx 1.12.2/jetdirect?               |

### [#1861] Sun/Oracle GlassFish Server 認証コード実行

深刻度: Critical

#### CVSSスコア

<https://research.sun-asterisk.com/ja/cvss.html#CVSS3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H>

#### 解説

Oracle Sun GlassFish Enterprise Server 2.1, 2.1.1, および 3.0.1, Sun Java System Application Server 9.1 における未検定の脆弱性により、リモート攻撃者は管理に関連する未知のベクターを介して機密性、整合性、および可用性に影響を与えることができます。この脆弱性を悪用するために使用できる Metasploit モジュールが 2 つ存在します。

#### 該当URL

<https://10.0.2.7:4848>

#### 影響

認証されていないユーザーが昇格された権限でサーバーを制御できるようになります。

#### 再現手順

ステップ1:

auxiliary/scanner/http/glassfish\_login Metasploit モジュールを使用して、Glassfish インスタンスを保護する認証コントロールをバイパスするか、ログイン資格情報をブルートフォースしました。認証バイパスの試みは失敗しましたが、インスタンスの管理者資格情報をブルートフォースすることは成功しました。



### [#1863] 既知の脆弱性のあるApache Solr使用

深刻度: Critical

#### CVSSスコア

<https://research.sun-asterisk.com/ja/cvss.html#CVSS3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H>

#### 解説

システムは安全でないソフトウェア (Apache Solr 6.5.1) を使用しており、ハッカーが攻撃してシステムを制御することを可能にし、データの整合性に影響を与えます。

#### 該当URL

<http://10.0.2.8:8983>

#### 影響

ハッカーがシステムを攻撃して制御できるようになり、データの整合性が損なわれます。

#### 再現手順

ステップ1:

Metasploit Framework を使用してApache Solrの脆弱性を悪用します (悪用コードはサーバー環境に合わせて修正されています)

```
msf5 exploit(multi/http/solr_velocity_req) > run
[*] Handler failed to bind to 15.197.142.173:8082...
[*] Started reverse TCP handler on 0.0.0.0:8082
[*] Found Apache Solr 6.5.1
[*] OS version is Windows Server 2016 amd64 10.0
[*] Found core(s): seo-keywords
[*] Found Velocity Response Writer in use by core 'seo-keywords'
[*] params.resource.loader.enabled for core 'seo-keywords' is set to true.
[*] Targeting core 'seo-keywords'
[*] params.resource.loader.enabled is true for core 'seo-keywords'
```

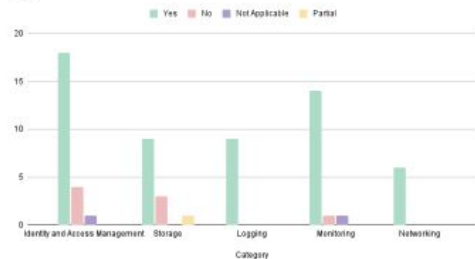
ステップ2:

システムの制御を成功裏に取得し、特権を昇格させました。

# クラウドセキュリティ診断結果報告書例

| No | 区分           | 診断項目数 | 適合数 | 不適合数 | 不適用 |
|----|--------------|-------|-----|------|-----|
| 1  | ID およびアクセス管理 | 23    | 18  | 5    | 0   |
| 0  | ストレージ        | 13    | 9   | 4    | 0   |
| 3  | ログ設定         | 9     | 9   | 0    | 0   |
| 4  | モニタリング       | 16    | 14  | 2    | 0   |
| 5  | ネットワーク       | 6     | 6   | 0    | 0   |
| 合計 |              | 67    | 56  | 11   | 0   |
| 割合 |              | 100%  | 83% | 17%  | 0%  |

Chart



| No | 区分                      | 不適合項目                                     | レベル |
|----|-------------------------|-------------------------------------------|-----|
| 1  | ID およびアクセス管理            | 「ルート」ユーザーのアクセスキーが存在する                     | 1   |
| 2  | MFAが「ルート」ユーザーで有効になっていない |                                           | 1   |
| 3  | ログ設定                    | CloudTrailはすべてのリージョンで有効になっていない            | 1   |
| 4  | ネットワーキング                | EC2インスタンスはRDPポート3389でパブリックインバウンドアクセスを許可する | 1   |

## 診断結果詳細

### [IAM] 「ルート」ユーザーのアクセスキーが存在する

レベル: 1

対策優先度: High

#### 解説

ルートアカウントはアクセスキーの使用を避けるべきです。ルートアカウントはアカウント全体に対して完全な権限を持っているため、それに対してアクセスキーを作成することは、侵害される可能性を高めるだけです。代わりに、事前定義されたロールを持つIAMユーザーを作成してください。

#### 影響

ルートアカウントはすべてのサービスおよびリソースに無制限のアクセス権を有しているため、資格情報が漏洩した場合、システム全体が攻撃にさらされる恐れがあります。

#### 設定変更による影響

AWS環境に対する影響はありません。

#### 再現手順

ステップ1:

AWSシステムはルートアカウントにAPIキーを使用することを確認できます。



## 対策方法

AWS コンソールでの設定方法

1. 'root' として AWS Management Console にサインインし、IAM コンソールを開きます。
2. 右上の <root\_account> をクリックし、ドロップダウンリストから My Security Credentials を選択します。
3. ポップアップ画面で Continue to Security Credentials をクリックします。
4. Access Keys (Access Key ID と Secret Access Key) をクリックします。
5. ステータス列の下で (アクティブなキーがある場合)
6. Delete をクリックします (注意: 削除されたキーは復元できません)。

### [IAM] MFAが「ルート」ユーザーで有効になっていない

・・・省略・・・

### [ログ設定] CloudTrailはすべてのリージョンで有効になっていない

レベル: 1

対策優先度: High

#### 解説

AWS CloudTrail は AWS、 マネジメントコンソールのアクションと API コールを記録します。呼び出したユーザーとアカウント AWS、 呼び出し元のソース IP アドレス、 呼び出しが発生した日時を特定できます。CloudTrail MULTI\_REGION\_CLOUD\_TRAIL\_ENABLED が有効になっている場合、 すべての AWS リージョンから S3 バケットにログファイルを配信します。さらに、 が新しいリージョン AWS を起動すると、CloudTrail は新しいリージョンに同じ経路を作成します。その結果、アクションを実行しなくても、新しいリージョンの API アクティビティを含むログファイルを受け取ることができるようになります。

#### 影響

CloudTrailがすべてのリージョンで有効化されていない場合、特定のリージョンで発生した操作やイベントが記録されず、セキュリティ監査やトラブルシューティングにおいて重要な情報が欠落する可能性があります。

#### 設定変更による影響

すべてのリージョンでCloudTrailを有効にすると、月々のコストが増加します。



**検知した脆弱性の例**

# 個人情報漏えいのリスクを発見

## 潜在的にあったリスク

学生の管理システムにおいて、表示されている学生情報は、「名前」と「学生番号」のみだが、プロキシツールを使いAPIにリクエストをしたところ、Eメール、生年月日、日本語能力といった**個人情報を閲覧できる状態**になっていた。

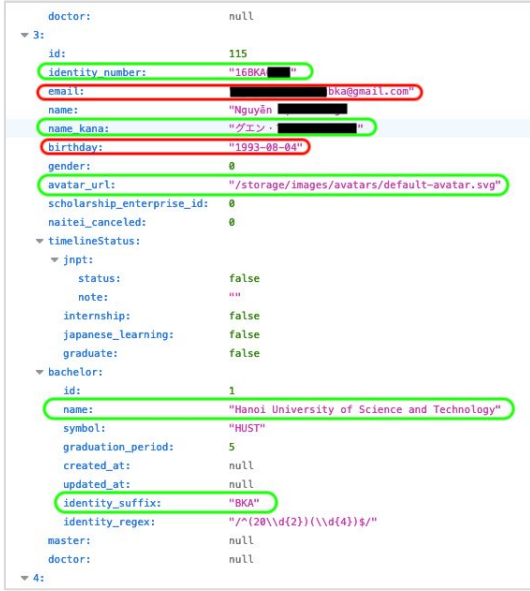
## アプリケーション

フロントエンド:Vue.js

バックエンド:Rails



A user profile card for a student named Nguyen BKA. It features a circular placeholder for a profile picture, the name 'グエン・[redacted] Nguyễn [redacted]', and the affiliation 'Hanoi University of Science and Tec...'. Below the name are four circular icons with minus signs, and a row of language skills: '日本語 IT科目 論理 英語'. A note indicates 'CPA: 未登録'. At the bottom, the student ID '22BKA [redacted]' and '男性 / 20歳' are displayed.



```

{
  "doctor": null,
  "3": {
    "id": 115,
    "identity_number": "16BKA[redacted]",
    "email": "[redacted]bka@gmail.com",
    "name": "Nguyễn [redacted]",
    "name_kana": "グエン・[redacted]",
    "birthday": "1993-08-04",
    "gender": 0,
    "avatar_url": "/storage/images/avatars/default-avatar.svg",
    "scholarship_enterprise_id": 0,
    "naitei_canceled": 0,
    "timelineStatus": {
      "jnpt": {
        "status": false,
        "note": "",
        "internship": false,
        "japanese_learning": false,
        "graduate": false
      }
    },
    "bachelor": {
      "id": 1,
      "name": "Hanoi University of Science and Technology",
      "symbol": "HUST",
      "graduation_period": 5,
      "created_at": null,
      "updated_at": null,
      "identity_suffix": "BKA",
      "identity_regex": "/^(28\\d{2})(\\d{4})$/",
      "master": null,
      "doctor": null
    }
  },
  "4": {

```

# 管理者を勝手に登録できてしまうリスクを発見

## 潜在的にあったリスク

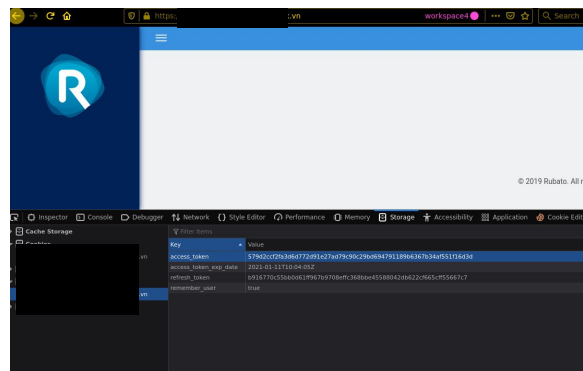
ブラウザからはスクリプトを実行することはできないが、APIを直接叩くことで、スクリプトの実行が可能な状態に。

サーバーの管理者グループに追加するリクエストを使うことで、**自分のIDを入れて、管理者として追加できてしまう状態** になっていた。

## アプリケーション

フロントエンド:Vue.js

バックエンド:Rails



```
1 POST /api/v1/group_users HTTP/1.1
2 Host: [REDACTED].vn
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Authorization: Bearer 53524529785d5087329a63d30a748ff21b8061446755cb06b3e0d41caf322f7c
9 Content-Length: 27
10 Origin: https://[REDACTED].vn
11 Connection: close
12 Referer: https://[REDACTED].vn/groups/11
13
14
15 {
  "group_id": 4,
  "user_id": 23
}
```

# クロスサイトスクリプト( Cross-site Scripting)

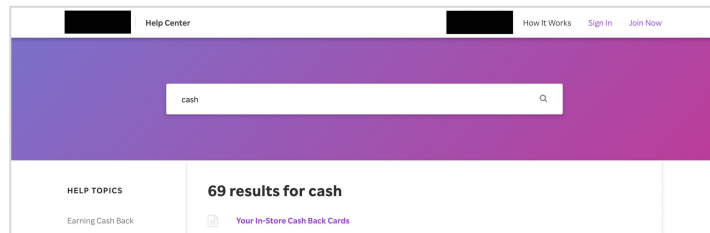
## 脆弱性の詳細

ブラウザからはスクリプトを実行することはできないが、HTMLソースコードを分析したところ、特集の文字を含むキーワードを入れることで、**任意の Javascriptを実行**できてしまう脆弱性を発見。

## アプリケーション

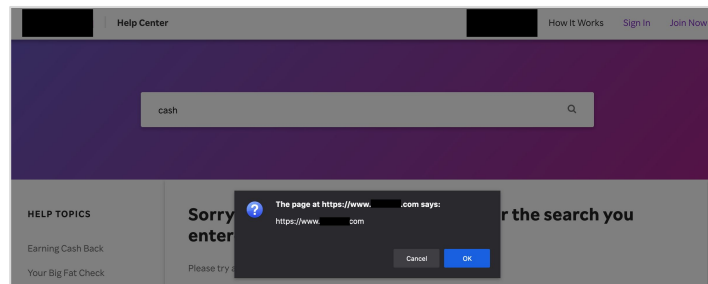
言語:Java

機能:検索



```
<input id="query" class="text ht-60 lh-60 textDefault" name="query" value="cash" placeholder="Search Help"
```

[https://www.shoppingsite.com/help/suggest?pageNum=1&query=cash&onfocus='confirm\(origin\)'&autofocus='m3l5l'](https://www.shoppingsite.com/help/suggest?pageNum=1&query=cash&onfocus='confirm(origin)'&autofocus='m3l5l')



# サーバの機密情報の漏洩リスクを発見

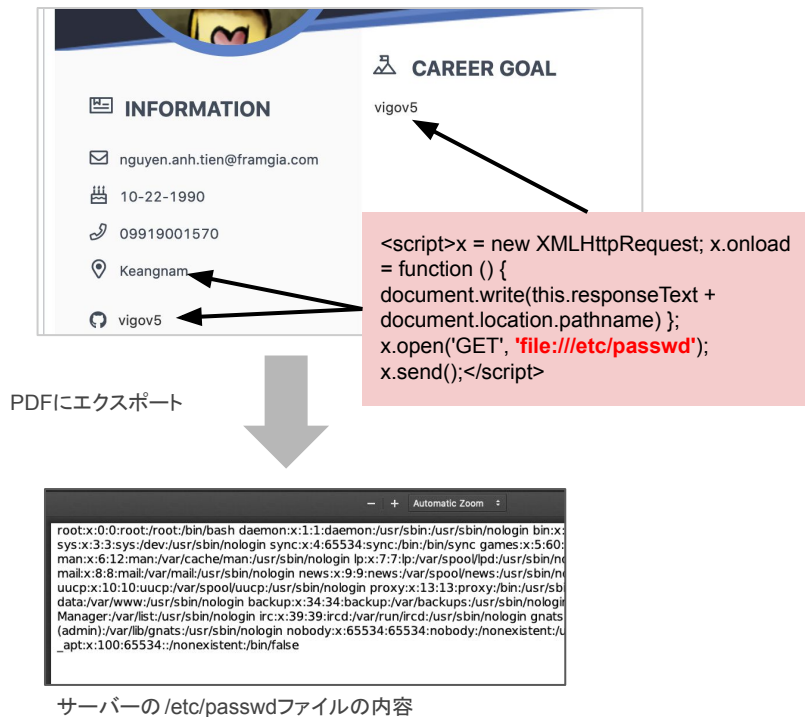
## 潜在的にあったリスク

特殊の Javascriptコードを入れることで、  
サーバーのパスワードを含む機密ファイルを  
PDFに書き出せる状態。

## アプリケーション

フレームワーク:Laravel

機能:CV作成、PDFにエクスポート



# ブルートフォース攻撃(総当たり攻撃)のリスクを発見

## 潜在的にあったリスク

サーバ側でレート制限がないことを利用し、特殊なツールを使い、確認コードを総当たり攻撃し、他ユーザーのアカウントへの不正アクセスできてしまう状態。

## アプリケーション

機能:SMSの確認コード

The screenshot shows a web application security tool interface with a table of attack results. The table has columns: Request, Payload, Status, Error, Timeout, Length, and Comment. The first row (Request 906) shows a successful attack (Status 200) with a Length of 1002. A blue arrow points to this row with the text '当たり' (Hit). A red arrow points to the 'Length' column with the text 'はずれ' (Miss). The 'Payload' column contains a long alphanumeric string. The 'Comment' column contains the text '確認コード (OTP)'. Below the table, the 'Response' tab is selected, showing the raw response data. The response includes headers like 'Accept-Encoding', 'Access-Control-Allow-Origin', and 'Content-Length'. The body of the response is a JSON object with 'code' and 'data' fields. The 'code' field is highlighted with a blue box and contains the value '200'.

| Request | Payload | Status | Error | Timeout | Length | Comment |
|---------|---------|--------|-------|---------|--------|---------|
| 906     | 260905  | 200    |       |         | 1002   |         |
| 0       |         | 200    |       |         | 280    |         |
| 1       | 260000  | 200    |       |         | 280    |         |
| 2       | 260001  | 200    |       |         | 280    |         |
| 4       | 260003  | 200    |       |         | 280    |         |
| 5       | 260004  | 200    |       |         | 280    |         |
| 7       | 260006  | 200    |       |         | 280    |         |
| 9       | 260008  | 200    |       |         | 280    |         |
| 10      | 260009  | 200    |       |         | 280    |         |
| 12      | 260011  | 200    |       |         | 280    |         |
| 14      | 260013  | 200    |       |         | 280    |         |
| 16      | 260015  | 200    |       |         | 280    |         |
| 19      | 260018  | 200    |       |         | 280    |         |

確認コード (OTP)

Response

```
6 Vary: Accept-Encoding
7 Access-Control-Allow-Origin: http://[redacted].vn
8 Content-Length: 776
9
10 {
  "code": 200,
  "data": "N9JEaSzwnTqqkZPLdyLZj0SaiQ6TR2Dh5I/1Gr6FL+6DYqkg1eskDfyWwHftZGVbZxsryQiMF9a46BnVl8eLqtKEFN"
}
```



## サイバーセキュリティチー ムの活動

# サイバーセキュリティチームの活動

常に脆弱性を探し、重要な課題を発見した際は、ベンダーに報告し、セキュリティコミュニティに貢献する活動を行っています。

## 活動領域

- バグバウンティプログラム
- オープンソースソフトウェア
- CVE
- サードパーティベンダー

これまでの脆弱性を報告した企業例

- 国内EC事業最大手
- 国内大手公共放送局
- 国内大手インターネットサービス企業
- 国内有力経済メディア
- 海外トップクラスの大学
- 世界的なCMSプラットフォーム
- 世界的なネットワーク機器大手



### Stored XSS in <https://www.binance.com/> (Non-Privileged to Anyone) 2020-09-30

binance.comでStoredXSSに関連するセキュリティの脆弱性を発見しました。それが bugcrowd でP2 (優先度2) の重大度として記録されました。



### [Wordpress Plugin] Advanced Database Cleaner < 3.0.2 - Authenticated SQL injection 2020-09-05

WordpressのAdvancedDatabase Cleaner <3.0.2に脆弱性が見つかりました。プラグインはユーザー入力を不適切にフィルタリングするにより、特権の高いユーザーがSQLインジェクション攻撃を実行できる。



### CVE-2020-24948 2020-08-26

チームはワードプレスプラグインAutooptimize Wordpress Plugin 2.7.6に脆弱性を発見した。この脆弱性はCVE-2020-24948として割り当てられました。



### SymphonyCMS 3.0.0 - Persistent Cross-Site Scripting 2020-08-27

チームはSymphonyCMS 3.0.0に脆弱性を発見した。永続的なクロスサイトスクリプティング (Stored XSS) の脆弱性により、リモートの攻撃者は任意のスクリプトまたはHTMLを挿入できます。

# サイバーセキュリティチームの活動



## ハーバード大学への脆弱性の報告



サードパーティベンダーが、チームのメンバーによって発見されたセキュリティの脆弱性に関するアドバイザリを公開  
<https://ajdg.solutions/blog/adrotate-5-8-4-security-update/>

| Source                                                                                                                                                                                                            |                         |         |        |             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|---------|--------|-------------|
| Cisco would like to thank Nguyen Anh Tien of Sun Asterisk Cyber Security Research Team for reporting this vulnerability.                                                                                          |                         |         |        |             |
| URL                                                                                                                                                                                                               |                         |         |        |             |
| <a href="https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-webex-xss-Lz6HbGct">https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-webex-xss-Lz6HbGct</a> |                         |         |        |             |
| Revision History                                                                                                                                                                                                  |                         |         |        |             |
| Version                                                                                                                                                                                                           | Description             | Section | Status | Date        |
| 1.0                                                                                                                                                                                                               | Initial public release. | —       | Final  | 2021-FEB-17 |

## Cisco Webex Meetingsに脆弱性を発見

CVE-2021-1351

2021/02/17にアップデートされた内容

<https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-webex-xss-Lz6HbGct.html>

# サイバーセキュリティチームの活動



504+

CVEを公開



27

ExploitDBで公開されたエクスプロイト



6

Metasploit Frameworkモジュール

**CVE (Common Vulnerabilities and Exposures)**は、情報セキュリティにおける脆弱性やインシデントについて、それぞれ固有の名前や番号を付与し、リスト化した事典です。

**ExploitDB (Exploit Database)** : ペネトレーションテスターと脆弱性研究者が使用する、攻撃コードの有無を調べるためのサイトです。

**Metasploit**は、脆弱性を見つけて悪用するのに役立つ侵入テストフレームワークです。

**Metasploit Framework**は、セキュリティの専門家が利用できる最も便利なテストツールの1つです。

# 外部公開された脆弱性発見実績

November 11th, 2021

## Orchestra C1 CMS Composite Deserialization of Untrusted Data Remote Code Execution Vulnerability

ZDI-21-1304  
ZDI-CAN-14740

大手製品に対して、リモートコード実行（RCE）につながる深刻な脆弱性を発見・報告

CVSS SCORE 8.8, AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

CREDIT Le Ngoc Anh - Sun\* Cyber Security Research Team

掲載しているのは、当チームが実際に発見し、CVE や ZDI を通じて外部公開・認定された脆弱性の一例です。これは、既知の脆弱性診断にとどまらず、製品やシステムの内部挙動を解析し、新たな脆弱性を見つけ出す技術力を示しています。

- **CVE-2024-5008 (WUG-15922) CVSS: 8.8**

- In WhatsApp Gold versions released before 2023.1.3, an authenticated user with certain permissions can upload an arbitrary file and obtain RCE using `Apm.UI.Areas.APM.Controllers.Api.Applications.AppProfileImportController`

認証後に悪用可能なアップロード機能の不備を突き、RCEにつながる脆弱性を発見・報告

- **CVE-2024-4884 (WUG-16139) CVSS: 9.8**

- In WhatsApp Gold versions released before 2023.1.3, an unauthenticated Remote Code Execution vulnerability exists in Progress WhatsApp. The `Apm.UI.Areas.APM.Controllers.CommunityController` allows execution of commands with `iisappnpool\l\mconsole` privileges

未認証で悪用可能なコマンド実行脆弱性を発見し、外部機関を通じて公開・報告



スケジュール

# 脆弱性診断の主なフロー

1

## テスト環境の準備 & 事前打ち合わせ

見積り必要情報のご提出、秘密保持契約、テストの範囲と対象の決定、環境/テストデータの準備、連絡体制、診断日程・方法確認  
**リリースの 2ヶ月前までにやっておくことをおすすめします**

2

## ドキュメントの共有とシステムの分析

可能であればシステムに関するドキュメントを共有いただき、それをもとにQAのテストケースを事前分析  
ドキュメントの準備は余裕を持ったスケジュールでご検討ください

3

## 脆弱性診断

自動化ツールを使用して基本的なスキャンを実行、ペネトレーションテスト（侵入テスト）を実施

4

## 手動によるセキュリティ診断

業界標準の脆弱性項目をカバーする診断項目「OWASPテストینگチェックリスト（第4版）」による診断を行う

5

## 速報の提出

重大な脆弱性を発見した場合は、即座に報告するとともに修正方法を提示

6

## 報告書の提出

発見された脆弱性の一覧を提出し、脆弱性が及ぼす影響の評価、解決方法、緊急度や必要性を評価、脆弱性の再現と対策方法を提示  
**期間の目安 3～6までで2週間程度**

7

## レポートのレビューと脆弱性修正

危険度の「中」以上のバグ修正を提案（シルバー以上のプランの場合は、Sun\*の開発チームへバグ修正方針を指導）  
**リリース 1ヶ月前に実施することが理想**

8

## 再テスト

脆弱性の対応を行った箇所を再テスト  
軽度な脆弱性はリリース後に対応するケースもあり

# 参考スケジュール

弊社で開発をしているプロジェクトに関しては、仕様の変更などでスケジュールが伸びた場合でも、リリーススケジュールを重視した柔軟な体制で診断を行うことが可能です。

|                    |      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|--------------------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                    | 役割   | 水 | 木 | 金 | 土 | 日 | 月 | 火 | 水 | 木 | 金  | 土  | 日  | 月  | 火  | 水  | 木  | 金  | 土  | 日  | 月  | 火  | 水  | 木  | 金  | 土  | 日  | 月  | 火  | 水  | 木  |
| テスト環境の準備 & 事前打ち合わせ | 両社   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ドキュメントの共有とシステムの分析  | 両社   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 自動による脆弱性診断         | Sun* |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 手動によるセキュリティ診断      | Sun* |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 速報の提出              | Sun* |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 報告書の提出             | Sun* |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| レポートのレビュー／脆弱性修正    | お客様  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 再テスト               | Sun* |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 再診断報告書の提出          | Sun* |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |



## QAで脆弱性対策はできないの？

A QAと脆弱性診断は、目的、ツール、ワークフローが異なるため、セキュリティ専門の対策が必要になります。

|        | QA                      | 脆弱性診断                               |
|--------|-------------------------|-------------------------------------|
| 目的     | 仕様書の通りに、アプリが動いているかの確認   | 情報漏えい等のセキュリティリスクがないかの確認             |
| ツール    | テストケース                  | セキュリティーチェックリスト、プロキシツール              |
| ワークフロー | シナリオ、画面のボタンUIによって、動作を確認 | 特殊なツールを使い、直接APIを叩いて、データを編集して、脆弱性を確認 |

# Amazon Inspectorとの違いは？

**A** Amazon Inspectorは、サーバー内の自動スキャンツールのため、検知できる脆弱性の範囲が限定されます。

| Amazon Inspector                                                                                                                                                                                                                                                                   | Sun*の脆弱性診断                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>WEB診断</b><br/>対象外</p> <p><b>プラットフォーム診断</b><br/>サーバー内のツールでサーバーにあるソフトウェアバージョンをチェックして脆弱性を洗い出します。サーバーの設定はCIS Operating System Security Configuration Benchmarks-1.0 に沿ってOSをチェックします。</p> <p>様々な脆弱性を発見しますが、リスクが少ないものもあります。(例：サーバーにawkがインストールされ、脆弱性があるが、実際にウェブアプリケーションが使われていない)</p> | <p><b>WEB診断</b><br/>ウェブアプリケーションに集中して診断することで、ウェブアプリケーション特有な脆弱性を発見します。<b>ビジネスロジック・認証の不備・アクセス制御の不備など、ツールでは発見できない脆弱性を経験豊富な診断者が診断します。</b></p> <p><b>プラットフォーム診断</b><br/>外部(インターネット)から、公開されたサーバーのポート・サービスに脆弱性があるかどうかを包括的に診断します。</p> |

Q

見積もりを APIリクエスト数ではなくスクリーン数で算出している理由は？

A

スクリーン数を採用している理由は、リクエスト数を正確に算出できない開発中や、見積もり段階のプロジェクトでも算出できるため です。

スクリーン(画面)の定義

リンク、ボタンまたは他の要素に対するユーザーアクションが行われ、その結果、現在の画面のコンテンツが根本的に変更され、新しく表示された場合は一つの画面とカウントされます。

Q

Sun\*のセキュリティテストの品質に問題はないのか？なぜ問題ないと言えるのか？

A

Sun\*の脆弱性診断は、R&Dという弊社内で技術レベルがトップクラスの人材がこの部門にあたっています。例えば、日本でも資格保持者が少ないOSCP、OSCE、OSWEなど世界最高峰の難度のセキュリティ資格を取得しています。品質テストを専門に行う会社は、フォーマットに従ってテスト項目を設定しています。一方、Sun\*はエンジニアでもあるエシカルハッカーが、業界標準の脆弱性項目を漏れなくカバーする最新の診断項目「OWASPテストチェックリスト」による診断を行いますので、大手セキュリティコンサル会社の診断項目と差がありません。



## OWASP Top 10

|                                                  |                                                                                                                                                                                   |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div>A1:2021</div> <div><b>アクセス制御の不備</b></div>   | アクセス制御が適切に実装されていない場合、ユーザーは本来許可されていない機能やデータへアクセスできてしまいます。この問題は、URL改ざん、強制ブラウジング、IDの推測などにより発生し、他ユーザーの情報閲覧、データ改ざん、権限昇格などの重大な影響を引き起こします。アクセス制御はサーバー側で厳格に実装し、すべてのリクエストに対して検証を行うことが重要です。 |
| <div>A2:2021</div> <div><b>暗号の失敗</b></div>       | 機微な情報の保護に必要な暗号化が適切に実装されていない場合、データ漏洩や改ざんのリスクが高まります。特に、通信の暗号化不足、弱い暗号アルゴリズム、不適切な鍵管理などが原因となります。保存時および通信時の暗号化を適切に実施し、最新の標準に従った安全な暗号方式を採用することが求められます。                                   |
| <div>A3:2021</div> <div><b>インジェクション</b></div>    | インジェクションの脆弱性は、信頼されていない入力コマンドやクエリの一部として処理される場合に発生します。SQL、NoSQL、OSコマンド、LDAP、XSSなどが含まれ、攻撃者はアプリケーションの意図しない処理を実行させたり、不正なデータ取得を行うことが可能となります。入力値の検証やエスケープ処理、パラメータ化クエリの使用が重要です。           |
| <div>A4:2021</div> <div><b>設計上の不備</b></div>      | 設計段階でセキュリティが十分に考慮されていない場合、実装が正しくても重大な脆弱性が残る可能性があります。脅威モデリングの不足やセキュリティ要件の未定義が原因となり、攻撃に対して脆弱な構造が生まれます。セキュア設計の原則を取り入れ、設計段階からリスクを評価・対策することが重要です。                                      |
| <div>A5:2021</div> <div><b>セキュリティ設定の不備</b></div> | 不適切なセキュリティ設定は、最も一般的に見られる問題の一つです。安全でないデフォルト設定、不完全な設定、不要な機能の有効化、過剰なエラーメッセージ、クラウド設定の誤りなどにより発生します。すべてのシステムやコンポーネントを安全に構成し、不要な機能の無効化や定期的な設定見直しを行うことが必要です。                              |

## OWASP Top 10

|                                                         |                                                                                                                                                          |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A6:2021</p> <p><b>脆弱で古いコンポーネント</b></p>               | <p>アプリケーションで使用するライブラリやフレームワークに既知の脆弱性が存在する場合、攻撃者に悪用される可能性があります。これらのコンポーネントはアプリケーションと同等の権限で動作するため、影響範囲が広がります。コンポーネントのバージョン管理と脆弱性情報の把握、定期的なアップデートが重要です。</p> |
| <p>A7:2021</p> <p><b>識別と認証の失敗</b></p>                   | <p>認証およびセッション管理の不備により、攻撃者が他ユーザーのアカウントを不正に利用する可能性があります。弱いパスワードポリシー、多要素認証の未導入、セッション管理の不備などが主な原因です。安全な認証方式の導入と適切なセッション管理の実装が求められます。</p>                     |
| <p>A8:2021</p> <p><b>ソフトウェアとデータの完全性の失敗</b></p>          | <p>ソフトウェアやデータの完全性が検証されていない場合、攻撃者により改ざんされたコードやデータが実行される可能性があります。安全でないデシリアライゼーションや信頼されていない更新プロセスなどが原因となります。コード署名や整合性検証、信頼できる配布プロセスの確立が重要です。</p>            |
| <p>A9:2021</p> <p><b>セキュリティログと監視の失敗</b></p>             | <p>ログの取得や監視が不十分な場合、攻撃の検知および対応が遅れ、被害が拡大する可能性があります。ログの未記録、不十分なアラート設定、監視体制の不備などが原因です。適切なログ収集と監視、インシデント対応プロセスの整備が必要です。</p>                                   |
| <p>A10:2021</p> <p><b>サーバーサイドリクエストフォージェリ (SSRF)</b></p> | <p>SSRFは、サーバーが外部リソースへリクエストを送信する機能を悪用されることで発生します。攻撃者は内部ネットワークやクラウドサービスへアクセスし、機密情報の取得やシステムの悪用を行う可能性があります。入力値の検証やアクセス制御、不要な外部通信の制限が重要です。</p>                |

| ASVSの章名                                  | ASVSのセクション名                     | ASVSの要件の説明                                                                                                                                                                                                                                                                                                                                        | OWASP Testing Guide                                |
|------------------------------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| Architecture, Design and Threat Modeling | Authentication Architecture     | Verify the use of unique or special low-privilege operating system accounts for all application components, services, and servers. ([C3](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                                                                                                        | OTG-AUTHN-002                                      |
| Architecture, Design and Threat Modeling | Authentication Architecture     | Verify that communications between application components, including APIs, middleware and data layers, are authenticated. Components should have the least necessary privileges needed. ([C3](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                                                   | OTG-AUTHN-001                                      |
| Architecture, Design and Threat Modeling | Authentication Architecture     | Verify that the application uses a single vetted authentication mechanism that is known to be secure, can be extended to include strong authentication, and has sufficient logging and monitoring to detect account abuse or breaches.                                                                                                            | OTG-AUTHN-006                                      |
| Architecture, Design and Threat Modeling | Access Control Architecture     | Verify the application uses a single and well-vetted access control mechanism for accessing protected data and resources. All requests must pass through this single mechanism to avoid copy and paste or insecure alternative paths. ([C7](https://owasp.org/www-project-proactive-controls/#div-numbering))                                     | OTG-INFO-010                                       |
| Architecture, Design and Threat Modeling | Access Control Architecture     | Verify that attribute or feature-based access control is used whereby the code checks the user's authorization for a feature/data item rather than just their role. Permissions should still be allocated using roles. ([C7](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                    | OTG-AUTHZ-003                                      |
| Architecture, Design and Threat Modeling | Input and Output Architecture   | Verify that serialization is not used when communicating with untrusted clients. If this is not possible, ensure that adequate integrity controls (and possibly encryption if sensitive data is sent) are enforced to prevent deserialization attacks including object injection.                                                                 | OTG-INPVAL-012                                     |
| Architecture, Design and Threat Modeling | Input and Output Architecture   | Verify that input validation is enforced on a trusted service layer. ([C5](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                                                                                                                                                                      | OTG-INPVAL-001<br>OTG-INPVAL-002                   |
| Architecture, Design and Threat Modeling | Input and Output Architecture   | Verify that output encoding occurs close to or by the interpreter for which it is intended. ([C4](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                                                                                                                                               | OTG-INPVAL-001<br>OTG-INPVAL-002                   |
| Architecture, Design and Threat Modeling | Communications Architecture     | Verify that application components verify the authenticity of each side in a communication link to prevent person-in-the-middle attacks. For example, application components should validate TLS certificates and chains.                                                                                                                         | OTG-CRYPST-001<br>OTG-CRYPST-002<br>OTG-CRYPST-003 |
| Architecture, Design and Threat Modeling | Secure File Upload Architecture | Verify that user-uploaded files - if required to be displayed or downloaded from the application - are served by either octet stream downloads, or from an unrelated domain, such as a cloud file storage bucket. Implement a suitable Content Security Policy (CSP) to reduce the risk from XSS vectors or other attacks from the uploaded file. | OTG-BUSLOGIC-008<br>OTG-BUSLOGIC-009               |
| Authentication                           | Password Security               | Verify that user set passwords are at least 12 characters in length (after multiple spaces are combined). ([C6](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                                                                                                                                 | OTG-AUTHN-007                                      |
| Authentication                           | Password Security               | Verify that passwords of at least 64 characters are permitted, and that passwords of more than 128 characters are denied. ([C6](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                                                                                                                 | OTG-AUTHN-007                                      |
| Authentication                           | Password Security               | Verify that password truncation is not performed. However, consecutive multiple spaces may be replaced by a single space. ([C6](https://owasp.org/www-project-proactive-controls/#div-numbering))                                                                                                                                                 | OTG-AUTHN-007                                      |



Make awesome things *that secured.*