Fortify Audit Workbench

OWASP Top 10 2021

DB-Card-main

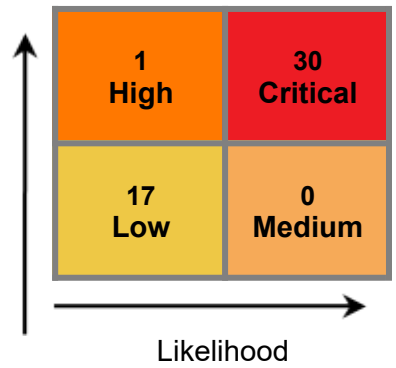
Table of Contents

Executive Summary

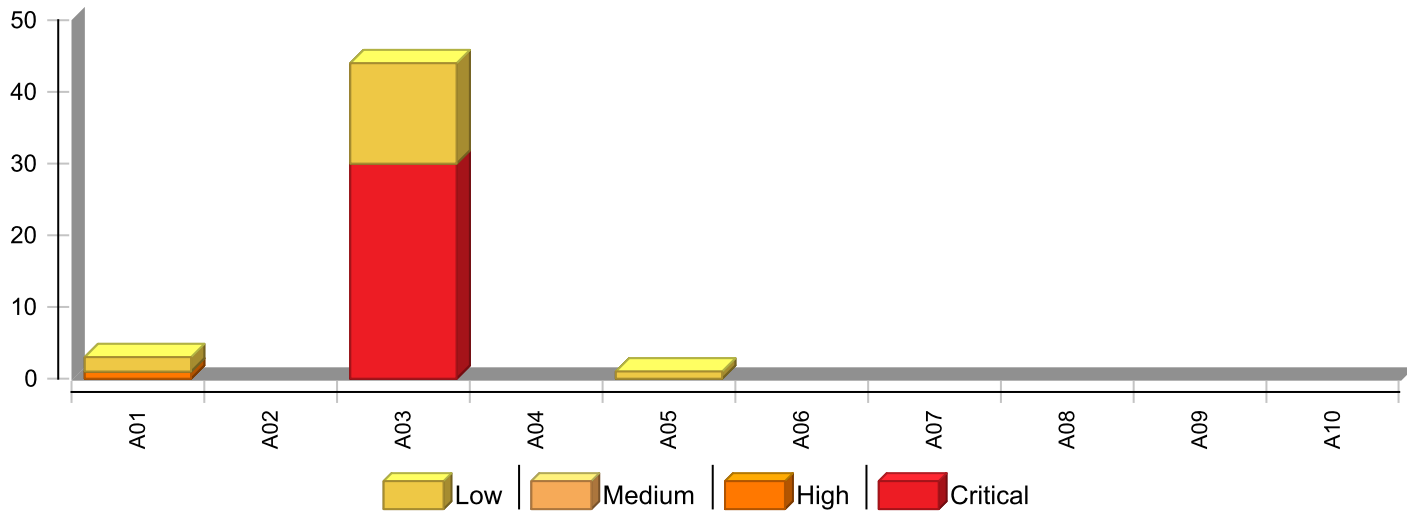
OWASP Top 10 2021 提供了一份強大的 Web 應用程式安全性意識文件，將重點放在讓社群瞭解最常見和最重要的 Web 應用程式安全性弱點的後果。OWASP Top 10 呈現出對於最關鍵 Web 應用程式安全性漏洞的廣泛共識，並從資料收集和調查結果中取得共識。專案成員包括來自世界各地的各種安全專家，在他們彼此分享專業知識之下，產生了這份清單。

Project Name:	DB-Card-main
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present
Remediation Effort (Hrs):	2.0

Issues by Priority



Issues by OWASP Top 10 2021 Categories



* The detailed sections following the Executive Summary contain specifics.

Project Description

This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

SCA

Date of Last Analysis:	Sep 5, 2025 11:14 AM	**Engine Version:**	24.4.1.0005
Host Name:	IISIFTFYSRV02	**Certification:**	VALID
Number of Files:	31	**Lines of Code:**	13,947

Rulepack Name	Rulepack Version
Fortify 安全編碼規則、社群、Cloud	2025.2.1.0001
Fortify 安全編碼規則、社群、Universal	2025.2.1.0001
Fortify 安全編碼規則、核心、Cloud	2025.2.1.0001
Fortify 安全編碼規則、核心、JavaScript	2025.2.1.0001
Fortify 安全編碼規則、核心、Universal	2025.2.1.0001
Fortify 安全編碼規則、延伸、配置	2025.2.1.0001
Fortify 安全編碼規則、延伸、內容	2025.2.1.0001
Fortify 安全編碼規則、延伸、JavaScript	2025.2.1.0001

Issue Breakdown

The following table summarizes the number of issues identified across the different OWASP Top 10 2021 categories and broken down by Fortify Priority Order.

	Fortify Priority				Total Issues	Effort (hrs)
	Critical	High	Medium	Low		
A01 Broken Access Control	0	1	0	2	3	0.5
A02 Cryptographic Failures	0	0	0	0	0	0.0
A03 Injection	30	0	0	14	44	1.7
A04 Insecure Design	0	0	0	0	0	0.0
A05 Security Misconfiguration	0	0	0	1	1	0.1
A06 Vulnerable and Outdated Components	0	0	0	0	0	0.0
A07 Identification and Authentication Failures	0	0	0	0	0	0.0
A08 Software and Data Integrity Failures	0	0	0	0	0	0.0
A09 Security Logging and Monitoring Failures	0	0	0	0	0	0.0
A10 Server-Side Request Forgery	0	0	0	0	0	0.0

NOTE:
1. Reported issues in the above table may violate more than one OWASP Top 10 2021 category. As such, the same issue may appear in more than one row. The total number of unique vulnerabilities are reported in the Executive Summary table.
2. For the same reason, the Project-level remediation effort total shown in the Executive Summary removes the effect of any duplication and may be smaller than the sum of the remediation effort per individual category.
3. Similarly, the remediation effort per external category is not intended to equal the sum of the remediation effort from the issue details section since individual files may contain issues in multiple Fortify priorities or audit folders.

Issue Details

Below is an enumeration of all issues found in the project. The issues are organized by OWASP Top 10 2021, Fortify Priority Order, and vulnerability category. The issues are then further broken down by the package, namespace, or location in which they occur. Issues reported at the same line number with the same category originate from different taint sources.

A01 Broken Access Control

OWASP Top 10 應用程式安全性風險，A01:2021 (OWASP Top 10 Application Security Risks, A01:2021) 稱：「存取控制會強制執行原則，讓使用者不能在指定權限之外採取行動。否則通常會導致未經授權的資訊洩露、修改或破壞所有資料，或在使用者限制之外執行業務功能。」

Open Redirect ***Remediation Effort(Hrs): 0.2***		**High**
Package: assets		
Location	**Analysis Info**	**Analyzer**
assets/bilingual-common.js:535	**Sink:** Assignment to link.href **Enclosing Method:** createSocialElement() **Source:** Read value **from** processTest() **In** test-social-links.html:153	SCA

Cross-Site Request Forgery ***Remediation Effort(Hrs): 0.4***		**Low**
Package: <none>		
Location	**Analysis Info**	**Analyzer**
nfc-generator-bilingual.html:273	**Sink:** **Enclosing Method:** () **Source:**	SCA
nfc-generator.html:191	**Sink:** **Enclosing Method:** () **Source:**	SCA

A02 Cryptographic Failures

OWASP Top 10 應用程式安全性風險，A02:2021 (OWASP Top 10 Application Security Risks, A02:2021) 稱：「首先是判斷傳輸中資料和靜止資料的保護需求。例如，密碼、信信用卡號碼、健康記錄、個人資訊和商業機密需要額外保護，主要是這類資料屬於歐盟的通用資料保護規則 (GDPR) 等隱私權法、或法規、亦或 PCI 資料安全標準 (PCI DSS) 等金融資料保護之範疇。」

No Issues

A03 Injection

OWASP Top 10 應用程式安全性風險，A03:2021 (OWASP Top 10 Application Security Risks, A03:2021) 稱：「以下情況，應用程式容易受到攻擊： - 應用程式不會驗證、過濾或清理使用者提供的資料。 - 直接在解譯器中使用沒有內容感知逸出的動態查詢或非參數化呼叫。 - 在物件關係對應 (ORM) 搜尋參數中，使用惡意資料來擷取額外的敏感記錄。 - 直接使用或串連惡意資料。SQL 或指令包含動態查詢、指令或已儲存程序中的結構和惡意資料。一些更常見的插入項目是 SQL、NoSQL、OS 指令、物件關係對應 (ORM)、LDAP 和運算式語言 (EL) 或物件圖形導覽程式庫 (OGNL) 插入。」

Cross-Site Scripting: DOM ***Remediation Effort(Hrs): 1.1***	**Critical**
Package: <none>	

Location	Analysis Info	Analyzer
index-en.html:691	**Sink:** Assignment to avatar.src **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-en.html:898	SCA
index-en.html:695	**Sink:** Assignment to emailLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-en.html:898	SCA
index-en.html:702	**Sink:** Assignment to phoneLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-en.html:898	SCA
index-en.html:713	**Sink:** Assignment to mobileLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-en.html:898	SCA
index-en.html:746	**Sink:** Assignment to tempDiv.innerHTML **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-en.html:898	SCA
index-personal-en.html:627	**Sink:** Assignment to avatar.src **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal-en.html:787	SCA
index-personal-en.html:631	**Sink:** Assignment to emailLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal-en.html:787	SCA
index-personal-en.html:637	**Sink:** Assignment to phoneLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal-en.html:787	SCA
index-personal-en.html:647	**Sink:** Assignment to mobileLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal-en.html:787	SCA
index-personal-en.html:690	**Sink:** Assignment to tempDiv.innerHTML **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal-en.html:787	SCA

A03 Injection

OWASP Top 10 應用程式安全性風險，A03:2021 (OWASP Top 10 Application Security Risks, A03:2021) 稱：「以下情況，應用程式容易受到攻擊： - 應用程式不會驗證、過濾或清理使用者提供的資料。 - 直接在解譯器中使用沒有內容感知逸出的動態查詢或非參數化呼叫。 - 在物件關係對應 (ORM) 搜尋參數中，使用惡意資料來擷取額外的敏感記錄。 - 直接使用或串連惡意資料。SQL 或指令包含動態查詢、指令或已儲存程序中的結構和惡意資料。一些更常見的插入項目是 SQL、NoSQL、OS 指令、物件關係對應 (ORM)、LDAP 和運算式語言 (EL) 或物件圖形導覽程式庫 (OGNL) 插入。」

Cross-Site Scripting: DOM ***Remediation Effort(Hrs): 1.1***	**Critical**
Package: <none>	

Location	Analysis Info	Analyzer
index-personal.html:627	**Sink:** Assignment to avatar.src **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal.html:787	SCA
index-personal.html:631	**Sink:** Assignment to emailLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal.html:787	SCA
index-personal.html:637	**Sink:** Assignment to phoneLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal.html:787	SCA
index-personal.html:647	**Sink:** Assignment to mobileLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal.html:787	SCA
index-personal.html:690	**Sink:** Assignment to tempDiv.innerHTML **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index-personal.html:787	SCA
index.html:661	**Sink:** Assignment to avatar.src **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index.html:892	SCA
index.html:665	**Sink:** Assignment to emailLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index.html:892	SCA
index.html:672	**Sink:** Assignment to phoneLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index.html:892	SCA
index.html:683	**Sink:** Assignment to mobileLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index.html:892	SCA
index.html:722	**Sink:** Assignment to tempDiv.innerHTML **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index.html:892	SCA

A03 Injection

OWASP Top 10 應用程式安全性風險，A03:2021 (OWASP Top 10 Application Security Risks, A03:2021) 稱：「以下情況，應用程式容易受到攻擊： - 應用程式不會驗證、過濾或清理使用者提供的資料。 - 直接在解譯器中使用沒有內容感知逸出的動態查詢或非參數化呼叫。 - 在物件關係對應 (ORM) 搜尋參數中，使用惡意資料來擷取額外的敏感記錄。 - 直接使用或串連惡意資料。SQL 或指令包含動態查詢、指令或已儲存程序中的結構和惡意資料。一些更常見的插入項目是 SQL、NoSQL、OS 指令、物件關係對應 (ORM)、LDAP 和運算式語言 (EL) 或物件圖形導覽程式庫 (OGNL) 插入。」

Cross-Site Scripting: DOM ***Remediation Effort(Hrs): 1.1***	**Critical**
Package: <none>	

Location	Analysis Info	Analyzer
index1-en.html:691	**Sink:** Assignment to avatar.src **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1-en.html:898	SCA
index1-en.html:695	**Sink:** Assignment to emailLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1-en.html:898	SCA
index1-en.html:702	**Sink:** Assignment to phoneLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1-en.html:898	SCA
index1-en.html:713	**Sink:** Assignment to mobileLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1-en.html:898	SCA
index1-en.html:746	**Sink:** Assignment to tempDiv.innerHTML **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1-en.html:898	SCA
index1.html:661	**Sink:** Assignment to avatar.src **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1.html:892	SCA
index1.html:665	**Sink:** Assignment to emailLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1.html:892	SCA
index1.html:672	**Sink:** Assignment to phoneLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1.html:892	SCA
index1.html:683	**Sink:** Assignment to mobileLink.href **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1.html:892	SCA
index1.html:722	**Sink:** Assignment to tempDiv.innerHTML **Enclosing Method:** renderCard() **Source:** Read window.location.search **from** getCardDataFromNFC() **In** index1.html:892	SCA

A03 Injection

OWASP Top 10 應用程式安全性風險，A03:2021 (OWASP Top 10 Application Security Risks, A03:2021) 稱：「以下情況，應用程式容易受到攻擊： - 應用程式不會驗證、過濾或清理使用者提供的資料。 - 直接在解譯器中使用沒有內容感知逸出的動態查詢或非參數化呼叫。 - 在物件關係對應 (ORM) 搜尋參數中，使用惡意資料來擷取額外的敏感記錄。 - 直接使用或串連惡意資料。SQL 或指令包含動態查詢、指令或已儲存程序中的結構和惡意資料。一些更常見的插入項目是 SQL、NoSQL、OS 指令、物件關係對應 (ORM)、LDAP 和運算式語言 (EL) 或物件圖形導覽程式庫 (OGNL) 插入。」

Cross-Site Scripting: Poor Validation ***Remediation Effort(Hrs): 0.8***	**Low**
Package: <none>	

Location	Analysis Info	Analyzer
index-en.html:863	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.pathname **from** genera teVCardContent() **In** index-en.html:780	SCA
index-en.html:863	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.search **from** getCardD ataFromNFC() **In** index-en.html:898	SCA
index-personal-en.html:752	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.pathname **from** genera teVCardContent() **In** index-personal-en.html:721	SCA
index-personal-en.html:752	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.search **from** getCardD ataFromNFC() **In** index-personal-en.html:787	SCA
index-personal.html:752	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.pathname **from** genera teVCardContent() **In** index-personal.html:721	SCA
index-personal.html:752	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.search **from** getCardD ataFromNFC() **In** index-personal.html:787	SCA
index.html:855	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.pathname **from** genera teVCardContent() **In** index.html:760	SCA
index.html:855	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.search **from** getCardD ataFromNFC() **In** index.html:892	SCA
index1-en.html:863	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.pathname **from** genera teVCardContent() **In** index1-en.html:780	SCA
index1-en.html:863	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.search **from** getCardD ataFromNFC() **In** index1-en.html:898	SCA

A03 Injection

OWASP Top 10 應用程式安全性風險，A03:2021 (OWASP Top 10 Application Security Risks, A03:2021) 稱：「以下情況，應用程式容易受到攻擊： - 應用程式不會驗證、過濾或清理使用者提供的資料。 - 直接在解譯器中使用沒有內容感知逸出的動態查詢或非參數化呼叫。 - 在物件關係對應 (ORM) 搜尋參數中，使用惡意資料來擷取額外的敏感記錄。 - 直接使用或串連惡意資料。SQL 或指令包含動態查詢、指令或已儲存程序中的結構和惡意資料。一些更常見的插入項目是 SQL、NoSQL、OS 指令、物件關係對應 (ORM)、LDAP 和運算式語言 (EL) 或物件圖形導覽程式庫 (OGNL) 插入。」

Cross-Site Scripting: Poor Validation ***Remediation Effort(Hrs): 0.8***		**Low**
Package: <none>		
Location	**Analysis Info**	**Analyzer**
index1.html:855	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.pathname **from** genera teVCardContent() **In** index1.html:760	SCA
index1.html:855	**Sink:** Assignment to contactLink.href **Enclosing Method:** setupVCardLink() **Source:** Read window.location.search **from** getCardD ataFromNFC() **In** index1.html:892	SCA

Cross-Site Scripting: Self ***Remediation Effort(Hrs): 0.4***		**Low**
Package: <none>		
Location	**Analysis Info**	**Analyzer**
nfc-generator-bilingual.html :507	**Sink:** Assignment to previewContent.innerHTML **Enclosing Method:** updateSocialPreview() **Source:** Read value **from** updateSocialPreview() **In** nfc-generator-bilingual.html:498	SCA
Package: assets		
Location	**Analysis Info**	**Analyzer**
assets/bilingual-common.js:5 35	**Sink:** Assignment to link.href **Enclosing Method:** createSocialElement() **Source:** Read value **from** processTest() **In** test-so cial-links.html:153	SCA

A04 Insecure Design

OWASP Top 10 應用程式安全性風險，A04:2021 (OWASP Top 10 Application Security Risks, A04:2021) 稱：「不安全的設計是一大類別，代表各種不同的弱點，並以『欠缺控制設計或控制設計無效』表示。不安全的設計並不是所有其他 Top 10 十大風險類別的來源。不安全的設計與不安全的實作是有所不同的。」

No Issues

A05 Security Misconfiguration

OWASP Top 10 應用程式安全性風險，A05:2021 (OWASP Top 10 Application Security Risks, A05:2021) 稱：「符合以下情況的應用程式可能存在漏洞： - 在應用程式堆疊的任何部分中，缺少適當的安全強化或對雲端服務的權限設定不正確。 - 啟用或安裝了不必要的功能 (例如，不必要的連接埠、服務、頁面、帳戶或權限)。 - 預設帳戶及其密碼仍處於啟用狀態且未更改。 - 錯誤處理向使用者顯示堆疊軌跡或其他資訊過多的錯誤訊息。 - 對於已升級的系統，最新的安全功能被停用或未安全設定。 - 應用程式伺服器、應用程式架構 (例如，Struts、Spring、ASP.NET)、程式庫、資料庫等項目中的安全設定未設定為安全值。 - 伺服器不會傳送安全性標頭或指令，或者它們未設定為安全值。 - 軟體已過時或存在漏洞。」

System Information Leak: Internal ***Remediation Effort(Hrs): 0.1***		**Low**
Package: <none>		
Location	**Analysis Info**	**Analyzer**
nfc-generator.html:676	**Sink:** ~JS_Generic.error() **Enclosing Method:** lambda() **Source:** lambda(0) **from** lambda() **In** nfc-generator.html:675	SCA

A06 Vulnerable and Outdated Components

OWASP Top 10 應用程式安全性風險，A06:2021 (OWASP Top 10 Application Security Risks, A06:2021) 稱：「您的系統可能存在漏洞： - 您不知道自己所用的所有元件的版本 (用戶端和伺服器端)。這包括您直接使用的元件以及巢狀相依項。 - 軟體存在漏洞、不受支援或已過時。這包括作業系統、Web/應用程式伺服器、資料庫管理系統 (DBMS)、應用程式、API 和所有元件、執行階段環境和程式庫。 - 未定期掃描漏洞並訂閱與所使用元件相關的安全公告。 - 未根據風險及時修正或升級基礎平台、架構及相依項。這通常發生在變更控制下實施每月或每季修補工作的環境，進而造成組織在數天或數月時間內，不必要地暴露於固定漏洞的風險。 - 軟體開發人員不測試更新、升級或修補後的程式庫相容性。 - 未保護元件的組態。」

No Issues

A07 Identification and Authentication Failures

OWASP Top 10 應用程式安全性風險，A07:2021 (OWASP Top 10 Application Security Risks, A07:2021) 稱：「確認使用者的身分、驗證和工作階段管理，對於防止與驗證相關的攻擊至關重要。如果應用程式符合以下情況，則可能存在驗證漏洞： - 允許憑證填充等自動攻擊，其中攻擊者擁有有效使用者名稱和密碼的清單。 - 允許暴力破解或其他自動攻擊。 - 允許使用預設密碼、低強度密碼或眾所周知的密碼，例如「Password1」或「admin/admin」。 - 使用無法確保安全的低強度或無效憑證復原和忘記密碼流程，例如「知識庫答案」。 - 使用純文字、加密或弱雜湊密碼資料存放區。 - 缺少多重驗證或多重驗證無效。 - 在 URL 中暴露工作階段識別碼。 - 成功登入後重複使用工作階段識別碼。 - 未正確讓工作階段識別碼失效。使用者工作階段或驗證權杖 (主要是單一登入 (SSO) 權杖) 在登出或閒置一段時間後未正確使其失效。」

No Issues

A08 Software and Data Integrity Failures

OWASP Top 10 應用程式安全性風險，A08:2021 (OWASP Top 10 Application Security Risks, A08:2021) 稱：「軟體和資料完整性故障與不能防止完整性違規的程式碼和基礎架構有關。這種情況的一個例子就是，應用程式所依賴的外掛程式、程式庫或模組來自不受信任的來源、存放庫和內容傳遞網路 (CDN)。不安全的 CI/CD 管線可能會導致未經授權的存取、惡意程式碼或危及系統安全。最後，現在的許多應用程式都包括自動更新功能，而更新在沒有充分驗證完整性的情況下被下載並套用於以前信任的應用程式。攻擊者可能會上傳自己的更新以散發在所有安裝上執行。另一個例子是物件或資料被編碼或序列化到一個結構中，而攻擊者可以看到和修改這個結構，進而容易遭受不安全的還原序列化。」

No Issues

A09 Security Logging and Monitoring Failures

OWASP Top 10 應用程式安全性風險，A09:2021 (OWASP Top 10 Application Security Risks, A09:2021) 稱：「協助偵測、提報和回應活躍的違規情況。如果沒有記錄和監控，就無法偵測到違規情況。記錄、偵測、監控和主動回應不足的情況，在任何時候都會發生：- 未記錄可稽核的事件，例如登入、登入失敗和高價值交易。- 警告和錯誤不會產生、不充分或不清楚的記錄訊息。- 未監控應用程式和 API 的記錄是否存在可疑活動。- 記錄僅儲存在本機。- 適當的警示閾值和回應提報流程沒有實施或無效。- 動態應用程式安全測試 (DAST) 工具所執行的滲透測試和掃描不會觸發警示。- 應用程式無法即時或近乎即時地偵測、提報或警示有關活躍的攻擊。 如果使用者或攻擊者可以看到記錄記錄和警示事件，您就容易遭受資訊洩漏問題。」

No Issues

A10 Server-Side Request Forgery

OWASP Top 10 應用程式安全性風險，A10:2021 (OWASP Top 10 Application Security Risks, A10:2021) 稱：「只要 Web 應用程式擷取遠端資源而不驗證使用者提供的 URL，就會出現 SSRF 缺陷。即使受到防火牆、VPN 或其他類型的網路存取控制清單 (ACL) 的保護，這也會允許攻擊者強制應用程式將精心設計的要求傳送至意外目的地。」

No Issues

Description of Key Terminology

Likelihood and Impact

Likelihood
Likelihood is the probability that a vulnerability will be accurately identified and successfully exploited.

Impact
Impact is the potential damage an attacker could do to assets by successfully exploiting a vulnerability. This damage can be in the form of, but not limited to, financial loss, compliance violation, loss of brand reputation, and negative publicity.

Fortify Priority Order

Critical
Critical-priority issues have high impact and high likelihood. Critical-priority issues are easy to detect and exploit and result in large asset damage. These issues represent the highest security risk to the application. As such, they should be remediated immediately.

SQL Injection is an example of a critical issue.

High
High-priority issues have high impact and low likelihood. High-priority issues are often difficult to detect and exploit, but can result in large asset damage. These issues represent a high security risk to the application. High-priority issues should be remediated in the next scheduled patch release.

Password Management: Hardcoded Password is an example of a high issue.

Medium
Medium-priority issues have low impact and high likelihood. Medium-priority issues are easy to detect and exploit, but typically result in small asset damage. These issues represent a moderate security risk to the application. Medium-priority issues should be remediated in the next scheduled product update.

Path Manipulation is an example of a medium issue.

Low
Low-priority issues have low impact and low likelihood. Low-priority issues can be difficult to detect and exploit and typically result in small asset damage. These issues represent a minor security risk to the application. Low-priority issues should be remediated as time allows.

Dead Code is an example of a low issue.

Remediation Effort

The report provides remediation effort estimates. You can use these estimates to perform a relative comparison of projects and as a starting point for estimates specific to your organization. Remediation effort estimates are provided in the following report sections:

- Executive Summary
- Issue Breakdown
- Issue Details

To determine remediation effort for a collection of issues, Software Security Center weights each issue based on its category (“remediation constant”) and adds an overhead calculation based on the number of distinct

files which contain the set of issues. The formula used at each report level is the same:

- Remediation Effort (in mins) = SUM(remediation constant for each issue in the set) + 6 * Number of distinct files in that set of issues.

At the lowest level of detail, issues are grouped based on Fortify category and Fortify priority OR Fortify category and folder name, depending on report options. So, for example, the Issue Details section of the report might show the remediation effort for “SQL Injection, Critical” or “SQL Injection, MyFolder”.

At the Issue Breakdown level, remediation effort is shown at the level of each external (non-Fortify) category (such as “AC-3 Access Enforcement” in the case of NIST, or “A1 Unvalidated Input” in the case of OWASP Top10). Remediation effort is calculated for the set of all issues that fall into that external category (irrespective of Fortify priority or folder name). As an example, if there are two SQL injection vulnerabilities, one critical and one medium, within the same file, the file overhead is only included once.

At the Executive Summary level, all issues of that project which are mapped to the specified external category list (such as NIST or CWE) are used in the remediation effort calculation.

Fortify recommends that you treat the different levels of remediation effort as information relevant at that level only. You cannot add up remediation effort at a lower level and expect it to match the remediation effort at a higher level.

About Fortify Solutions

Fortify is the leader in end-to-end application security solutions with the flexibility of testing on-premise and on-demand to cover the entire software development lifecycle. Learn more at www.microfocus.com/solutions/application-security.