

**Università degli Studi di Salerno**

Penetration Testing Report

HACKABLE III

Federica Ungherese | Corso di PTEH | A.A. 2021/2022

Sommario

[Executive Summary 3](#_Toc111739687)

[Engagement Highlights 3](#_Toc111739688)

[Vulnerability Report 0](#_Toc111739689)

[Remediation Report 1](#_Toc111739690)

[Findings Summary 1](#_Toc111739691)

[Detailed Summary 7](#_Toc111739692)

[apache 2.4.46 7](#_Toc111739693)

[directory browsing 13](#_Toc111739694)

[cgi generic local file inclusion 13](#_Toc111739695)

[Web Application Potentially Vulnerable to Clickjacking *e* Content Security Policy (CSP) Header Not Set 13](#_Toc111739696)

[Application Error Disclosure 14](#_Toc111739697)

[Assenza di Token Anti-CSRF 14](#_Toc111739698)

[Missing Anti-clickjacking Header 14](#_Toc111739699)

[Web Server Transmits Cleartext Credentials 15](#_Toc111739700)

[Cross-Domain JavaScript Source File Inclusion 15](#_Toc111739701)

[INFORMAZIONI SENSIBILI in COMMENTI DI CODICE 15](#_Toc111739702)

[Content-Type Header Missing 16](#_Toc111739703)

[References 17](#_Toc111739704)

# Executive Summary

L’attività di penetration testing esaminata nel corso del presente documento è stata svolta alla macchina virtuale “**Hackable III”**, asset vulnerable-by-design, disponibile presso il sito VulnHub [1]. Tale testing è stato condotto ai fini della presentazione dell’attività progettuale per il corso di Penetration Testing & Ethical Hacking svoltosi nell’anno accademico 2021/2022.

Lo **scopo** dell’attività di penetration testing è stato quello di determinare l’esposizione dell’asset ad eventuali attacchi, derivandone il livello di sicurezza e le relative debolezze, simulando un piano d’attacco che potrebbe essere portato a compimento da un utente malevolo che abbia lo scopo di compromettere confidenzialità, l’integrità e la disponibilità della macchina.

Per simulare un attacco svolto da un utente privo di informazioni provenienti dall’interno, è stato condotto un **testing** di tipo black box: al tester non sono state fornite informazioni in merito all’asset.

La macchina attaccante dispone di sistema operativo ParrotOS Security Edition 5.0.1, e l’**ambiente** operativo, nel quale sono inserite il target e l’attaccante, è una rete NAT denominata “Corso” e presente all’interno dell’ambiente di virtualizzazione di Oracle VM Virtualbox.

Le azioni intraprese nel corso del processo di testing ai fini di svelare debolezze e falle nella sicurezza hanno fatto emergere delle vulnerabilità, alcune delle quali con gravi livelli di criticità e rischio, che, se sfruttate da un attaccante remoto, potrebbero condurre a gravi conseguenze per la sicurezza dell’asset, che porterebbero l’attaccante ad acquisire pieno controllo sulla macchina.

Nelle sezioni successive tali vulnerabilità vengono dettagliate e corredate dalle opportune mitigazioni ai fini di mettere in sicurezza l’asset.

# Engagement Highlights

Il testing è stato eseguito per fini accademici, ragion per cui non si è reso necessario stabilire regole d’ingaggio tra le parti, né un accordo di non divulgazione, ed è stato possibile analizzare l’asset nella sua interezza e senza restrizioni sugli strumenti da utilizzare.

# Vulnerability Report

Il penetration testing ha portato alla luce sia dall’analisi manuale che dall’uso di strumenti automatici numerose vulnerabilità.

Un’analisi manuale del sito ospitato sulla porta 80 da Hackable III ha fatto emergere delle vulnerabilità gravi dovute alla cattiva configurazione del web server, e i tool automatici utilizzati, Nessus, OWASP ZAP e nikto2 hanno confermato che la criticità più impattante sulla sicurezza dell’asset, che lo mette a serio rischio di compromissione da parte di un avversario remoto, risulta essere la versione di web server in uso.

Oltre questa, sono state individuate anche vulnerabilità di alto, medio e basso livello, insieme a vario contenuto informativo.

In particolare, le vulnerabilità individuate sono le seguenti:

* **Versione web server obsoleta**: consente a un attaccante di sfruttare vulnerabilità note al fine di prendere il controllo del sistema;
* **Cattiva configurazione del web server**: rende possibile a un attaccante e a qualsiasi utente visiti il servizio web offerto dall’asset di fare *directory browsing* e visualizzarne i file, ottenendo informazioni che è possibile sfruttare;
* **Informazioni sensibili in commenti del codice**: il codice HTML delle pagine web di Hackable III racchiude dei commenti che segnalano informazioni compromettenti;
* **Password a basso livello di sicurezza**: le password utilizzate dagli utenti del sistema non rispettano i requisiti minimo di lunghezza dei caratteri, uso di maiuscole e minuscole, numeri e caratteri speciali;
* **Cattiva configurazione dei permessi**: sul sistema a un utente non autorizzato è possibile reperire un file di backup sul quale non dovrebbe invece avere permessi di lettura;

# Remediation Report

Le vulnerabilità menzionate sono mitigabili tramite i seguenti accorgimenti:

* **Aggiornare la versione del web server**;
* **Correggere la configurazione del web server** affinché non renda possibile il *directory browsing*;
* **Attuare corrette pratiche di programmazione** e non inserire informazioni compromettenti, quali credenziali utente o segnalazioni di bug, all’interno del codice;
* **Istruire gli utenti** del sistema a corrette pratiche di sicurezza, come utilizzare password complesse;
* **Correggere la configurazione dei permessi** di lettura dei documenti relativi agli utenti con privilegi inferiori, sui file di backup del sistema contenenti informazioni sulle credenziali di utenti terzi.

# Findings Summary

Le vulnerabilità vengono di seguito esposte in base al servizio di riferimento, e sono state raggruppate in base al livello di gravità:

* **CRITICA**: vulnerabilità che rappresentano un grave pericolo per la sicurezza del sistema e che possono condurre a danni rilevanti al sistema;
* **ALTA**: vulnerabilità di grande impatto che possono portare a importanti perdite di confidenzialità o integrità del sistema;
* **MEDIA**: vulnerabilità che richiedono particolari condizioni per essere sfruttate, ma che potrebbero compromettere il sistema;
* **BASSA**: vulnerabilità non prioritarie e con un minimo impatto sul sistema;
* **INFO**: contenuto informativo relativo al sistema;

Per prima è stata svolta una ricerca manuale inerente al server web individuato in fase di testing: **Apache 2.4.46** e sono state individuate le seguenti vulnerabilità.

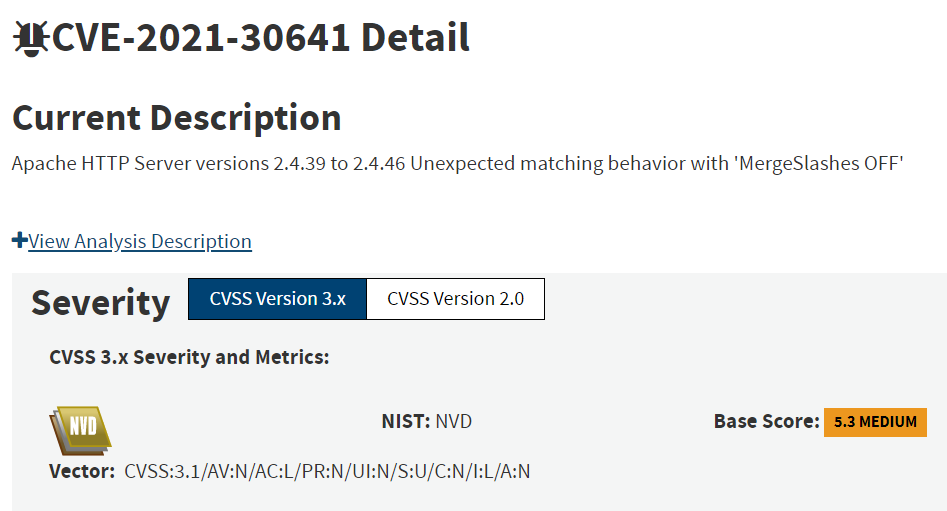


Figura . Vulnerabilità Apache: CVE-2021-30641

Immagine che contiene testo

Descrizione generata automaticamente

Figura . Vulnerabilità Apache: CVE-2021-26691

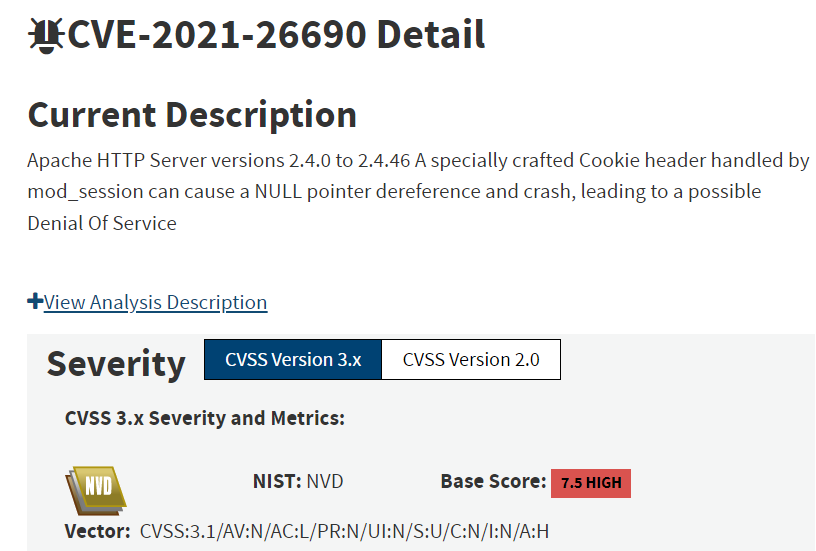


Figura . Vulnerabilità Apache: CVE-2021-26690

Immagine che contiene testo

Descrizione generata automaticamente

Figura . Vulnerabilità Apache: CVE-2020-35452

Successivamente lo strumento automatico **Nessus** ha prodotto ulteriori vulnerabilità di vari livelli di gravità, fra le quali numerose di livello critico, tutte legate alla versione del web server in uso – fra le quali sono presenti anche quelle individuate con la ricerca manuale.

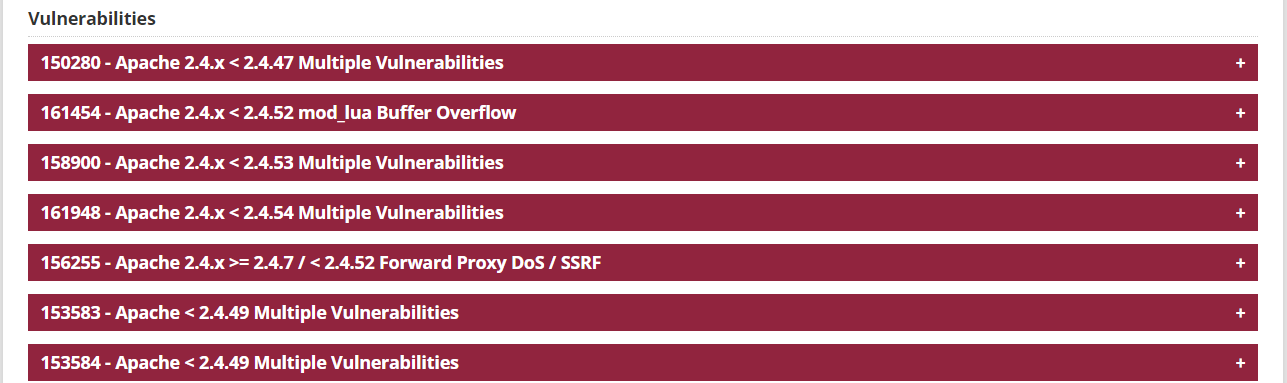


Figura . Nessus: Vulnerabilità Critiche.

Immagine che contiene testo

Descrizione generata automaticamente

Figura . Nessus: Vulnerabilità Alte.



Figura . Nessus: Vulnerabilità Medie.

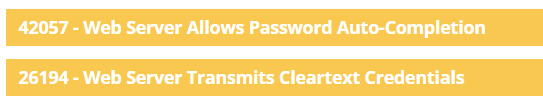


Figura . Nessus: Vulnerabilità Basse.

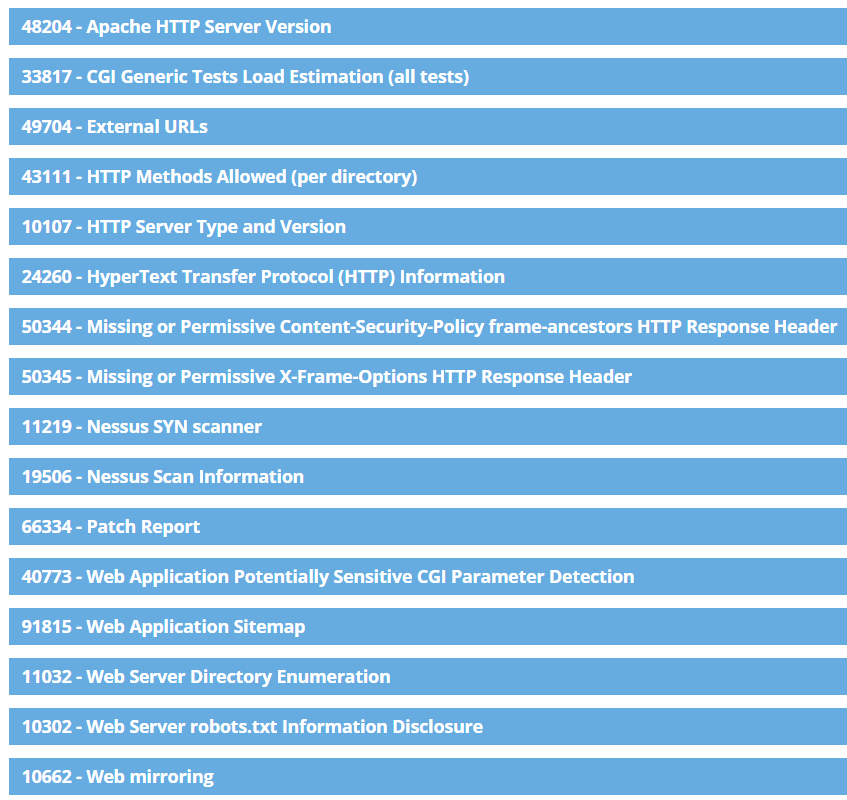


Figura . Nessus: Info.

Le vulnerabilità e le informazioni individuate da Nessus sono complessivamente 30.

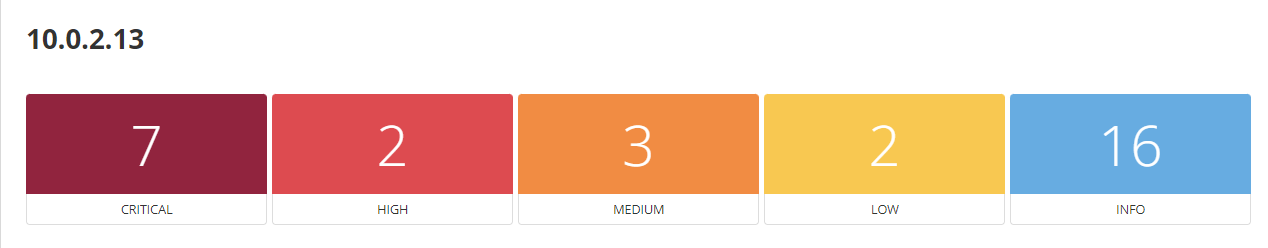


Figura . Nessus: totale vulnerabilità.

Come secondo strumento di vulnerability assessment è stato utilizzato **OWASP ZAP**, che ha prodotto i seguenti risultati, alcuni dei quali comuni a quelli prodotti da Nessus.



Figura . OWASP ZAP: Vulnerabilità.

Immagine che contiene tavolo

Descrizione generata automaticamente

Figura . OWASP ZAP: Vulnerabilità totali.

Una panoramica delle vulnerabilità individuate dagli strumenti automatici è illustrata nel grafico.

# Detailed Summary

In questa sezione verranno dettagliate le vulnerabilità individuate e il grado di rischio che comportano, con la relativa mitigazione e il riferimento. Laddove non viene specificato, il CVSS di riferimento è il 3.0.

## apache 2.4.46

|  |  |
| --- | --- |
| Codice | CVE-2021-26691 |
| Rischio | 9.8 - Critico |
| Descrizione | Un Session header creato ad hoc inviato da un server potrebbe causare un heap overflow |
| Soluzione | Aggiornare ad Apache 2.4.47 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2021-26691 |

|  |  |
| --- | --- |
| Codice | CVE-2022-23943 |
| Rischio | 9.8 - Critico |
| Descrizione | Una vulnerabilità di scrittura fuori dai limiti in mod\_sed di Apache HTTP Server consente a un utente malintenzionato di sovrascrivere la memoria heap con dati eventualmente forniti dall'attaccante. |
| Soluzione | Aggiornare ad Apache 2.4.53 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2022-23943 |

|  |  |
| --- | --- |
| Codice | CVE-2022-31813 |
| Rischio | 9.8 - Critico |
| Descrizione | Apache HTTP Server 2.4.53 e precedenti potrebbero non inviare le intestazioni X-Forwarded-\* al server di origine in base al meccanismo hop-by-hop dell'intestazione di connessione lato client. Ciò può essere usato per aggirare l'autenticazione basata su IP del server/applicazione di origine. |
| Soluzione | Aggiornare ad Apache 2.4.54 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2022-31813 |

|  |  |
| --- | --- |
| Codice | CVE-2022-28615 |
| Rischio | 9.1 - Critico |
| Descrizione | Versioni di Apache HTTP server 2.4.53 e precedenti possono andare in crash o esibire informazioni in maniera incorretta a seguito di una lettura al di fuori dei limiti tramite la funzione ap\_strcmp\_match() quando usata con un input buffer molto grande. Sebbene nessun codice distribuito con il server possa essere forzato a effettuare una chiamata di questo tipo, i moduli di terze parti o gli script lua che utilizzano ap\_strcmp\_match() potrebbero essere ipoteticamente interessati. |
| Soluzione | Aggiornare ad Apache 2.4.54 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2022-28615 |

|  |  |
| --- | --- |
| Codice | CVE-2021-44224 |
| Rischio | 8.2 - Alto |
| Descrizione | Un URI modificato inviato a httpd configurato come forward proxy (ProxyRequests on) può causare un arresto anomalo (dereferenziazione di un puntatore NULL) o, per le configurazioni che mescolano dichiarazioni di forward e reverse proxy, può consentire di indirizzare le richieste a un endpoint Unix Domain Socket dichiarato (Server Side Request Forgery). |
| Soluzione | Aggiornare ad Apache 2.4.52 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2021-44224 |

|  |  |
| --- | --- |
| Codice | CVE-2022-26377 |
| Rischio | 7.5 - Alto |
| Descrizione | La vulnerabilità Inconsistent Interpretation of HTTP Requests ("HTTP Request Smuggling") in mod\_proxy\_ajp di Apache HTTP Server consente a un utente malintenzionato di inviare di nascosto le richieste al server AJP a cui le inoltra. |
| Soluzione | Aggiornare ad Apache 2.4.54 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2022-26377 |

|  |  |
| --- | --- |
| Codice | CVE-2021-26690 |
| Rischio | 7.5 - Alto |
| Descrizione | Un cookie header creato appositamente e gestito da mod\_session può causare un riferimento a un NULL pointer e un crash, che può portare a un possibile Denial of Service. |
| Soluzione | Aggiornare ad Apache 2.4.47 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2021-26690 |

|  |  |
| --- | --- |
| Codice | CVE-2020-13950 |
| Rischio | 7.5 - Alto |
| Descrizione | È possibile ottenere un crash tramite riferimento a un NULL pointer tramite mod\_proxy\_http creando appositamente richieste che fanno uso sia di header Content-Length che di Transfer-Encoding. È possibile causare un Denial of Service. |
| Soluzione | Aggiornare ad Apache 2.4.47 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2020-13950 |

|  |  |
| --- | --- |
| Codice | CVE-2021-33193 |
| Rischio | 7.5 - Alto |
| Descrizione | Un apposito metodo inviato tramite HTTP/2 raggirerà la convalida e sarà inoltrato da mod\_proxy, il che può portare alla suddivisione delle richieste o a cache poisoning. |
| Soluzione | Aggiornare ad Apache 2.4.49 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2021-33193 |

|  |  |
| --- | --- |
| Codice | CVE-2022-29404 |
| Rischio | 7.5 - Alto |
| Descrizione | Una richiesta dannosa a uno script lua che chiama r:parsebody(0) può causare un Denial of Service a causa dell'assenza di un limite predefinito sulle dimensioni possibili dell'input. |
| Soluzione | Aggiornare ad Apache 2.4.54 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2022-29404 |

|  |  |
| --- | --- |
| Codice | CVE-2022-30556 |
| Rischio | 7.5 - Alto |
| Descrizione | Versioni di Apache HTTP server 2.4.53 e inferiori possono restituire lunghezze alle applicazioni che invocano r:wsread() che puntano dopo la fine dello storage allocato per il buffer. |
| Soluzione | Aggiornare ad Apache 2.4.54 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2022-30556 |

|  |  |
| --- | --- |
| Codice | CVE-2021-34798 |
| Rischio | 7.5 - Alto |
| Descrizione | Richieste malformate possono causare un riferimento a un puntatore nullo. |
| Soluzione | Aggiornare ad Apache 2.4.49 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2021-34798 |

|  |  |
| --- | --- |
| Codice | CVE-2021-36160 |
| Rischio | 7.5 - Alto |
| Descrizione | Un percorso uri della richiesta accuratamente modificato può far sì che mod\_proxy\_uwsgi legga oltre la memoria allocata e causi un crash (DoS). |
| Soluzione | Aggiornare ad Apache 2.4.49 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2021-36160 |

|  |  |
| --- | --- |
| Codice | CVE-2020-35452 |
| Rischio | 7.3 - Alto |
| Descrizione | Un Digest nonce creato appositamente può causare uno stack overflow in mod\_auth\_digest. |
| Soluzione | Aggiornare ad Apache 2.4.47 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2020-35452 |

|  |  |
| --- | --- |
| Codice | CVE-2021-30641 |
| Rischio | 5.3 - Medio |
|  |  |
| Descrizione | Versioni di Apache HTTP Server 2.4.39 – 2.4.46 hanno un comportamento di matching inatteso con “MergeSlashes OFF” |
| Soluzione | Aggiornare ad Apache 2.4.47 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2021-30690 |

|  |  |
| --- | --- |
| Codice | CVE-2022-28614 |
| Rischio | 5.3 - Medio |
| Descrizione | La funzione ap\_rwrite() in versioni precedenti di Apache HTTP server 2.4.53 possono leggere porzioni di memoria non intenzionalmente se un attaccante fa sì che il server rifletta un input molto grande usando ap\_rwrite() o ap\_rputs(), come con la funzione mod\_luas r:puts(). |
| Soluzione | Moduli compilati e distribuiti separatamente da Apache HTTP server che fanno uso della funzione “ap\_rputs” e inviano una stringa di grandi dimensioni (INT\_MAX o superiore) devono essere compilati con gli header correnti per risolvere il problema.  Aggiornare ad Apache 2.4.54 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2022-28614 |

|  |  |
| --- | --- |
| Codice | CVE-2019-17567 |
| Rischio | 5.3 - Medio |
| Descrizione | mod\_proxy\_wstunnel configurato su un URL che non è necessariamente aggiornato dal server di origine eseguivano il tunneling dell’intera connessione, consentendo così il passaggio di richieste successive sulla stessa connessione senza alcuna convalida HTTP, autenticazione o autorizzazione eventualmente configurata. |
| Soluzione | Aggiornare ad Apache 2.4.47 o superiore. |
| Riferimenti | https://nvd.nist.gov/vuln/detail/CVE-2019-17567 |

Lo stesso Nessus, nella sezione **Patch Report** suggerisce la contromisura più adatta a risolvere le problematiche relative al web server, ossia aggiornare alla versione 2.4.54 o superiore di Apache.

Immagine che contiene testo

Descrizione generata automaticamente

Figura . Nessus: Patch Report.

## directory browsing

|  |  |
| --- | --- |
| Codice | CWE-548: Exposure of Information Through Directory Listing |
| Rischio | 5.3 – Medio |
| Descrizione | Un directory listing fornisce a un attaccante l’indice completo delle risorse all’interno della directory. I rischi e le conseguenze dipendono dalla natura dei file elencati e accessibili. |
| Soluzione | Limitare l’accesso a directory o file importanti, disattivare features come Automatic Directory Listings, che potrebbero esporre file privati e fornire informazioni utilizzabili dall’attaccante. |
| Riferimenti | https://cwe.mitre.org/data/definitions/548.html |

## cgi generic local file inclusion

|  |  |
| --- | --- |
| Codice | CWE-73: External Control of File Name or Path |
| Rischio | 5.0 – Medio (CVSS 2.0) |
| Descrizione | Il software consente all’input dell’utente di controllare o influenzare i path o i nomi dei file che sono usati in operazioni del filesystem. |
| Soluzione | Validare l’input dell’utente |
| Riferimenti | https://cwe.mitre.org/data/definitions/73 |

## Web Application Potentially Vulnerable to Clickjacking *e* Content Security Policy (CSP) Header Not Set

|  |  |
| --- | --- |
| Codice | CWE-693: Protection Mechanism Failure |
| Rischio | 4.3 – Medio (CVSS 2.0) |
| Descrizione | Il prodotto non fa uso di un meccanismo di protezione (o lo usa in maniera non corretta) che dà una difesa sufficiente contro attacchi mirati. |
| Soluzione | Restituire l’header HTTP X-Frame-Options o Content-Security-Policy con la risposta dalla pagina, così da prevenire che il contenuto della pagina sia elaborato da un altro sito quando si usano i tag HTML frame o iframe. |
| Riferimenti | https://cwe.mitre.org/data/definitions/693 |

## Application Error Disclosure

|  |  |
| --- | --- |
| Codice | CWE-200: Exposure of Sensitive Information to an Unauthorized Actor |
| Rischio | Medio |
| Descrizione | Informazioni sensibili sono esposte ad attori che non sono esplicitamente autorizzati ad averne accesso. |
| Soluzione | Creare compartimenti nel sistema per avere delle aree sicure per designare dei confini di sicurezza e non consentire a dati sensibili di fuoriuscirne. |
| Riferimenti | https://cwe.mitre.org/data/definitions/200.html |

## Assenza di Token Anti-CSRF

|  |  |
| --- | --- |
| Codice | CWE-352: Cross-Site Request Forgery (CSRF) |
| Rischio | Medio |
| Descrizione | L'applicazione Web non verifica, o non può verificare in modo sufficiente, se una richiesta ben formata, valida e coerente è stata fornita intenzionalmente dall'utente che ha inviato la richiesta. |
| Soluzione | Usare librerie e framework che evitino tale debolezza, non usare il metodo GET per richieste che azionano un cambio di stato, controllare l’header HTTP Refer per verificare se la richiesta è stata originata dalla pagina prevista,… |
| Riferimenti | https://cwe.mitre.org/data/definitions/352.html |

## Missing Anti-clickjacking Header

|  |  |
| --- | --- |
| Codice | CWE-1021: Improper Restriction of Rendered UI Layers or Frames |
| Rischio | Medio |
| Descrizione | L'applicazione Web non limita o limita in modo errato gli oggetti frame o i livelli dell'interfaccia utente che appartengono a un'altra applicazione o a un altro dominio, il che può generare confusione nell'utente circa l'interfaccia con cui sta interagendo. |
| Soluzione | Usare X-Frame-Options per restringere l’uso nell’applicazione di form di overlay, frame o iframe, usare uno script frame-breaker, usare la tecnica della difesa in profondità per prevenire un uso improprio dei frame nell’applicazione web. |
| Riferimenti | https://cwe.mitre.org/data/definitions/1021.html |

## Web Server Transmits Cleartext Credentials

|  |  |
| --- | --- |
| Codice | CWE-522: Insufficiently Protected Credentials |
| Rischio | 2.6 – Basso (CVSS 2.0) |
| Descrizione | Vengono trasmesse o memorizzate credenziali di autenticazione con un metodo non sicuro e soggetto ad intercettazioni o retrieval non autorizzati. |
| Soluzione | Uso di appropriati meccanismi di sicurezza per proteggere le credenziali, cifrare le credenziali ecc. |
| Riferimenti | https://cwe.mitre.org/data/definitions/522 |

|  |  |
| --- | --- |
| Codice | CWE-523: Unprotected Transport of Credentials |
| Rischio | 2.6 – Basso (CVSS 2.0) |
| Descrizione | La pagina di login non usa misure appropriate per proteggere lo username e la password mentre transitano tra il client e il server. |
| Soluzione | Usare SSL per la pagina di login e ogni altra pagina che trasmette credenziali utente o informazioni sensibili. |
| Riferimenti | https://cwe.mitre.org/data/definitions/523 |

## Cross-Domain JavaScript Source File Inclusion

|  |  |
| --- | --- |
| Codice | CWE-829: Inclusion of Functionality from Untrusted Control Sphere |
| Rischio | Basso |
| Descrizione | Il software importa, richiede o include funzionalità eseguibili (come una libreria) da una fonte che non rientra nella sfera di controllo prevista. |
| Soluzione | Utilizzare una libreria o un framework collaudato che non permetta questa debolezza o che fornisca costrutti che la rendano più facilmente evitabile, eseguire il codice in una sandbox, eseguire il codice con i privilegi minimi necessari, validare l’input, usare un firewall,… |
| Riferimenti | https://cwe.mitre.org/data/definitions/829.html |

## INFORMAZIONI SENSIBILI in COMMENTI DI CODICE

|  |  |
| --- | --- |
| Codice | CWE-546: Suspicious Comment |
| Rischio | Basso |
| Descrizione | Il codice contiene commenti che suggeriscono la presenza di bug, funzionalità incomplete o punti deboli. |
| Soluzione | Rimuovere i commenti che suggeriscono la presenza di bug, funzionalità incomplete o punti deboli, prima di distribuire l'applicazione. |
| Riferimenti | https://cwe.mitre.org/data/definitions/546.html |

|  |  |
| --- | --- |
| Codice | CWE-615: Inclusion of Sensitive Information in Source Code Comments |
| Rischio | Basso |
| Descrizione | Alcuni programmatori tendono a lasciare nei commenti di codice dati importanti, come nomi di file relativi all'applicazione web, vecchi link o link che non erano destinati a essere consultati dagli utenti, ecc. |
| Soluzione | Rimuovere i commenti che contengono informazioni sensibili sulla progettazione/implementazione dell'applicazione. |
| Riferimenti | https://cwe.mitre.org/data/definitions/615.html |

## Content-Type Header Missing

|  |  |
| --- | --- |
| Codice | CWE-345: Insufficient Verification of Data Authenticity |
| Rischio | Informativo |
| Descrizione | Il software non verifica sufficientemente l'origine o l'autenticità dei dati, in modo da accettare dati non validi. |
| Riferimenti | https://cwe.mitre.org/data/definitions/345.html |

# References

|  |  |
| --- | --- |
| [1] | E. Sousa, «Hackable: III - VulnHub,» 2 Giugno 2021. [Online]. Available: https://www.vulnhub.com/entry/hackable-iii,720/. |
| [2] | «CWE - CWE-546: Suspicious Comment,» 19 07 2006. [Online]. Available: https://cwe.mitre.org/data/definitions/546.html. |
| [3] | «CWE - CWE-615: Inclusion of Sensitive Information in Source Code Comments,» 07 05 2007. [Online]. Available: https://cwe.mitre.org/data/definitions/615.html. |
| [4] | «Directory indexing attacks - IBM documentation,» 08 03 2021. [Online]. Available: https://www.ibm.com/docs/en/snips/4.6.0?topic=categories-directory-indexing-attacks. |
| [5] | «CWE - CWE-548: Exposure of Information Through Directory Listing,» 19 07 2006. [Online]. Available: https://cwe.mitre.org/data/definitions/548.html. |