Università degli Studi di Salerno

# Penetration Testing Report

WORST WESTERN HOTEL:1

FEDERICA PAPPALARDO | Corso di PTEH | A.A. 2023/2024

UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

# Sommario

## Executive Summary

In questo documento sono riportati i risultati del penetration testing sulla macchina "Worst Western Hotel:1".

L'obiettivo di tale attività è stato di scoprire, analizzare, sfruttare e documentare il maggior numero possibile di vulnerabilità.

L'attività di penetration testing è stata avviata il giorno 10 Maggio 2024 ed è stata conclusa il giorno 19 giugno 2024. L'attività è stata compiuta da un unico pentester, autore di questo report. Inoltre, si è utilizzato un approccio **black-box**: all'avvio dell'attività, il pentester non possiede alcuna informazione circa l'asset da testare, simulando completamente il comportamento di un utente malintenzionato.

Tutte le attività sono state condotte in modo da simulare un attore malintenzionato impegnato in un attacco contro la macchina Worst Western Hotel:1. In particolare:

1. Ricerca di vulnerabilità all'interno del sistema manualmente e con software open-source.
2. Verificare se un attaccante remoto può penetrare le difese della macchina Worst Western Hotel:1.
3. Determinare l'impatto di sicurezza su:
    (1) Riservatezza e dati presenti sulla macchina Worst Western Hotel:1
    (2) Violazione di sistemi interni ed accesso ad una shell sulla macchina Worst Western Hotel:1.

L'attività di penetration testing ha permesso di portare alla luce le vulnerabilità ed i punti deboli del sistema. In generale, le vulnerabilità individuate possono consentire ad un utente malintenzionato di ottenere un controllo parziale o completo del sistema.

Le vulnerabilità emerse sono di varia natura e vanno ad impattare sull'*affidabilità*, sull'*integrità* e sulla *confidenzialità* del sistema.

In questo documento verranno analizzate le vulnerabilità individuate durante l'attività di penetration testing e verranno presentate possibili contromisure per mitigare la criticità e diminuire il livello di rischio (che allo stato attuale del sistema è **MEDIO**), al fine di ottenere un livello di sicurezza complessivo almeno accettabile.

## Engagement Highlights

L'attività di penetration testing è stata effettuata utilizzando il sistema operativo ***Kali Linux***, installato su una macchina collegata alla stessa rete locale della macchina testata ***Worst Western Hotel:1***. Poiché si è nell'ambito di un progetto universitario, non sono state poste particolari limitazioni sui tool da utilizzare. Per questo motivo sono stati adoperati software con licenza gratuita messi a disposizione da Kali Linux.

A fronte della situazione sopra citata non è necessario creare un canale di comunicazione tra il pentester e il cliente e nemmeno ci si è posto dei limiti sull'impatto del testing vista la natura non critica del sistema target, il quale, girerà stesso sulla

macchina del pentester. Inoltre, a scopo didattico sono state installate delle **backdoor** sulla macchina target per garantire un accesso permanente.

La metodologia utilizzata per condurre il test consiste in un framework generico composto dalle seguenti fasi:


*(Tutte le fasi precedenti a quella di Reporting sono riportate del documento **Penetration Testing Narrative**).*

Quindi, non dovendo definire regole di ingaggio concordate tra le parti interessate prima dell'inizio del test, il pentester ha avuto piena libertà nella scelta delle metodologie, dei tempi, degli strumenti e delle tecniche. Inoltre, non è presente alcun accordo di non divulgazione (**NDA**).

## Vulnerability Report

Le vulnerabilità individuate durante il processo di penetration testing possono essere suddivise in tre categorie principali:

1. Vulnerabilità causate da una mancanza di *security awareness* del personale dell'organizzazione.
2. Vulnerabilità causate da errori di configurazione del sistema.
3. Vulnerabilità relative a software presenti nel sistema.

Nella prima categoria rientrano tutte quelle vulnerabilità causate da un comportamento errato e poco sicuro adottato dai dipendenti dell'organizzazione. Tali vulnerabilità possono essere sfruttate da un malintenzionato per ottenere maggiori informazioni sul sistema e sull'organizzazione e in taluni casi per accedere da remoto al sistema, ottenendone un controllo parziale e/o completo.

Nella seconda categoria rientrano tutte quelle vulnerabilità causate da errate scelte di configurazione del sistema. Tali scelte espongono il sistema al rischio di attacchi provenienti dall'esterno, che possono consentire ad un utente malintenzionato di ottenere informazioni sensibili o di accedere e controllare da remoto il sistema.

Nella terza categoria rientrano tutte quelle vulnerabilità riguardanti software e servizi messi a disposizione dal sistema. Sulla macchina server sono presenti versioni obsolete di alcuni software. Tali versioni risultano essere affette da diverse vulnerabilità, le quali possono essere sfruttate per sferrare attacchi di varia natura.

Di seguito sono elencate le principali vulnerabilità riscontrate:

- Cross-site Scripting
- SQL Injection
- JQuery 1.2 < 3.5.0 Multiple XSS – **Rischio MEDIO**
  - La versione di JQuery 1.11.0 presente attualmente nel sistema risulta avere diverse vulnerabilità che un utente malintenzionato potrebbe sfruttare per creare disservizi.

- ICMP Timestamp Request Remote Date Disclosure – **Rischio MEDIO**

- Web Application Potentially Vulnerable to Clickjacking – **Rischio MEDIO**

- Web Server Transmits Cleartext Credentials – **Rischio BASSO**
  - Il server web invia informazioni di login, come username e password, in un formato non crittografato.

- Web Server Allows Password Auto-Completion – **Rischio BASSO**

- TCP Timestamps Information Disclosure – **Rischio BASSO**

- Alcune credenziali d'accesso vengono rese note in immagini contenute nelle pagine web.

## Remediation Report

L'attività di penetration testing ha evidenziato che il livello di sicurezza complessivo della macchina analizzata è abbastanza basso. Al fine di garantire una sicurezza più elevata, si raccomanda di adottare le seguenti contromisure:

- Introdurre misure di sicurezza più rigorose al fine di garantire una maggiore protezione dei dati sensibili degli utenti e dei fruitori del sistema.

- Risolvere tutte le vulnerabilità presentate in questo documento, seguendo un ordine decrescente in base alla gravità: è consigliato risolvere quanto prima le vulnerabilità critiche e procedere successivamente alla correzione delle vulnerabilità con criticità più bassa. In generale, si raccomanda di mettere in atto i seguenti interventi:
  - Aggiornamento delle versioni dei software ritenuti a rischio.
  - Eliminare tutti i dati sensibili presenti nelle pagine web ospitate dalla macchina.
  - Riconfigurare alcuni servizi del sistema.
  - Utilizzare formati crittografati per i dati sensibili come username e password.
  - Evitare il salvataggio automatico delle credenziali d'accesso degli utenti.

- Garantire il principio dei privilegi minimi attraverso una politica migliore di assegnazione dei privilegi.

- Utilizzare controlli più robusti degli input che provengono da form o parametri passati dall'utente, soprattutto se tali parametri sono utilizzati all'interno di query. La soluzione potrebbe coinvolgere:
  - Controllo sui tipi di dati passati
  - Controllo della presenza di caratteri speciali e di escape
  - Controllo della presenza di parole chiave dei linguaggi di interrogazione
  - Utilizzo del binding dei parametri invece della concatenazione per le query.

- Pianificare periodicamente dei *Security Audits* al fine di valutare regolarmente il grado di sicurezza e la conformità agli standard del sistema.

## Findings Summary

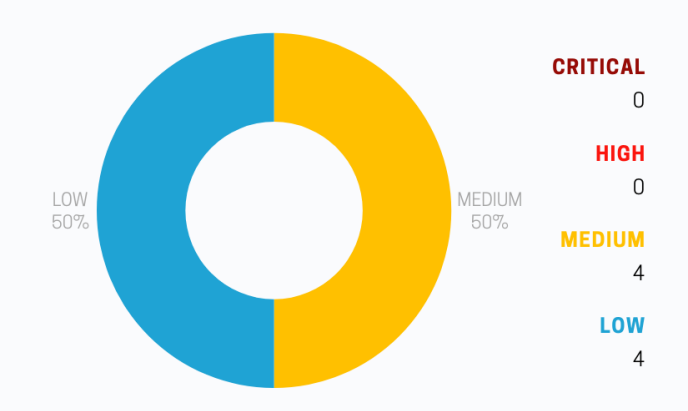
Le analisi portate avanti con gli strumenti scelti (Nessus e OpenVas) hanno evidenziato: 8 Vulnerabilità, di cui 4 a rischio medio e 4 a basso rischio.

Invece, l'analisi manuale delle vulnerabilità ha permesso di rilevare 2 vulnerabilità.

Le vulnerabilità riscontrate sono classificate in base alla gravità, suddivise su una scala con quattro livelli di gravità:

- **CRITICAL**: vulnerabilità che possono avere un impatto elevato e che possono consentire ad un utente malintenzionato di ottenere un controllo completo e/o parziale del sistema.
- **HIGH**: vulnerabilità che richiedono determinati requisiti per poter essere sfruttate e hanno un impatto relativamente alto sul sistema.
- **MEDIUM**: vulnerabilità non semplici da sfruttare e che, nella maggior parte dei casi, non hanno un impatto diretto molto significativo.
- **LOW**: vulnerabilità che hanno un impatto poco significativo e che hanno una bassa probabilità di essere sfruttate e, pertanto, non rappresentano, nell'immediato, una minaccia rilevante per il sistema.

Il grafico seguente mostra in maniera schematica la distribuzione delle vulnerabilità per categoria:




# Detailed Summary

## MEDIUM

### JQuery 1.2 < 3.5.0 Multiple XSS

| JQuery 1.2 < 3.5.0 Multiple XSS | CVE |
|---|---|
| | **2020-11022** [1] |
| **MEDIUM** | |
| **Descrizione**:<br>Secondo la versione auto-riportata nello script, la versione di JQuery ospitata sul server Web remoto è maggiore o uguale a 1.2 e precedente a 3.5.0. È quindi affetto da molteplici vulnerabilità di cross site scripting. | |
| **Impatto:**<br>Lo script dannoso può essere utilizzato per rubare informazioni sensibili agli utenti, dirottarli su siti web dannosi o assumere il controllo dei loro account. | |
| **Soluzione**:<br>Aggiornare la versione di JQuery alla 3.5.0 o successiva. | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software Nessus. | |

## ICMP Timestamp Request Remote Date Disclosure

| ICMP Timestamp Request Remote Date Disclosure | CVE |
|---|---|
| | **1999-0524** [2] |
| **MEDIUM** | |
| **Descrizione**:<br>L'host remoto risponde a una richiesta di timestamp ICMP. Ciò consente a un utente malintenzionato di conoscere la data impostata sulla macchina presa di mira, il che può aiutare un utente malintenzionato remoto non autenticato a sconfiggere i protocolli di autenticazione basati sul tempo. | |
| **Impatto:**<br>Sfrutta la risposta a una richiesta ICMP Timestamp per ottenere informazioni sulla netmask e sul timestamp del host di destinazione. | |
| **Soluzione**:<br>Filtrare le richieste di timestamp ICMP e le risposte di timestamp ICMP in uscita. | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software Nessus. | |

## Web Application Potentially Vulnerable to Clickjacking

| Web Application Potentially Vulnerable to Clickjacking | CWE |
|---|---|
| | **693** [3] |
| **MEDIUM** | |
| **Descrizione**:<br>Il server Web remoto non imposta un'intestazione di risposta X-Frame-Options o un'intestazione di risposta "frame-ancestors" di Content-Security-Policy in tutte le risposte di contenuto. Ciò potrebbe potenzialmente esporre il sito a un attacco di clickjacking o di reindirizzamento dell'interfaccia utente, in cui un utente malintenzionato può indurre un utente a fare clic su un'area della pagina vulnerabile che è diversa da ciò che l'utente si aspetta. | |
| **Impatto:**<br>Può comportare che un utente esegua transazioni fraudolente o dannose. | |
| **Soluzione**:<br>Restituire l'intestazione HTTP X-Frame-Options o Content-Security-Policy (con la direttiva 'frame-ancestors') con la risposta della pagina.<br>Ciò impedisce che il contenuto della pagina venga visualizzato da un altro sito quando si utilizzano i tag HTML frame o iframe. | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software Nessus. | |

### Cleartext Transmission of Sensitive Information via HTTP

| Cleartext Transmission of Sensitive Information via HTTP | CWE |
|---|---|
| | **319** [4] |
| **MEDIUM** | |
| **Descrizione**:<br>L'host/applicazione trasmette informazioni sensibili (nome utente, password) in testo in chiaro tramite HTTP. | |
| **Impatto:**<br>Un utente malintenzionato potrebbe sfruttare questa situazione per compromettere o intercettare la comunicazione HTTP tra il client e il server utilizzando un attacco man-in-the-middle per ottenere l'accesso a dati sensibili come nomi utente o password. | |
| **Soluzione**:<br>imporre la trasmissione di dati sensibili tramite una connessione SSL/TLS crittografata. Assicurarsi che l'host/l'applicazione reindirizzi tutti gli utenti alla connessione SSL/TLS protetta prima di consentire l'inserimento di dati sensibili. | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software OpenVas. | |

## LOW

### Web Server Transmits Cleartext Credentials

| Web Server Transmits Cleartext Credentials | CWE |
|---|---|
| | **522** [5] |
| **LOW** | |
| **Descrizione**:<br>Il server web invia informazioni di login, come username e password, in un formato non crittografato. | |
| **Impatto:**<br>Un utente malintenzionato che intercetta il traffico tra il browser Web e il server può ottenere accessi e password di utenti validi. | |
| **Soluzione**:<br>Assicurarsi che ogni form sensibile trasmetta contenuti tramite HTTPS. | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software Nessus. | |

## Web Server Allows Password Auto-Completion

| Web Server Allows Password Auto-Completion | CVE |
|---|---|
| | - |
| **LOW** | |
| **Descrizione**:<br>Il server Web remoto contiene almeno un campo form HTML con un input di tipo "password" dove "completamento automatico" non è impostato su "disattivato". | |
| **Impatto:**<br>Sebbene ciò non rappresenta un rischio per il server Web, significa che gli utenti potrebbero avere le proprie credenziali salvate nei browser, il che potrebbe portare a una perdita di riservatezza se utilizzano un dominio condiviso o se la macchina viene compromessa. | |
| **Soluzione**:<br>Aggiungere l'attributo "autocomplete=off" ai campi per impedire ai browser di memorizzare nella cache le credenziali. | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software Nessus. | |

## TCP Timestamps Information Disclosure

| TCP Timestamps Information Disclosure | CVE |
|---|---|
| | - |
| **LOW** | |
| **Descrizione**:<br>L'host remoto implementa i timestamp TCP e consente di calcolare il tempo di attività. | |
| **Impatto:**<br>- | |
| **Soluzione**:<br>Disabilitare i timestamp TCP su Linux aggiungendo la riga 'net.ipv4.tcp_timestamps = 0' a /etc/sysctl.conf ed inoltre di eseguire 'sysctl -p' per applicare le impostazioni in fase di runtime. | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software OpenVas. | |

### ICMP Timestamp Reply Information Disclosure

| ICMP Timestamp Reply Information Disclosure | CVE |
|---|---|
| | **1999-0524** [2] |
| **LOW** | |
| **Descrizione**:<br>L'host remoto risponde a una richiesta di timestamp ICMP con la data e l'ora corrente. | |
| **Impatto:**<br>- | |
| **Soluzione**:<br>Disabilitare completamente il supporto per il timestamp ICMP sull'host remoto o di protegge l'host remoto con un firewall e bloccare i pacchetti ICMP che passano attraverso il firewall in entrambe le direzioni (completamente o solo per reti non attendibili). | |
| **Metodi di detection**:<br>Vulnerabilità individuata tramite il software OpenVas. | |

## ALTRE VULNERABILITA'

### Cross-site Scripting

| Cross-site Scripting | CVE |
|---|---|
| | **2012-3499** [6] |
| **MEDIUM** | |
| **Descrizione**:<br>Molteplici vulnerabilità di cross-site scripting (XSS) nel server Apache HTTP 2.2.x prima di 2.2.24-dev e 2.4.x prima di 2.4.4 consentono agli aggressori remoti di iniettare script web arbitrari o HTML tramite vettori che coinvolgono nomi host e URI nel moduli mod_imagemap, mod_info, mod_ldap, mod_proxy_ftp e moduli mod_status. | |
| **Impatto:**<br>- | |
| **Soluzione**:<br>Aggiornare la versione di Apache installata nel sistema (> 2.4.29). | |
| **Metodi di detection**:<br>Vulnerabilità individuata manualmente. | |

### SQL Injection

| SQL Injection | CWE |
|---|---|
| | **89** [7] |
| | |
| **Descrizione**:<br>L'applicazione costruisce un comando SQL utilizzando un input esterno e non neutralizza (o lo fa in modo errato) caratteri speciali del linguaggio SQL, | |
| **Impatto:**<br>**-** | |
| **Soluzione**:<br>Implementare controlli sull'inserimento di caratteri speciali, | |
| **Metodi di detection**:<br>Vulnerabilità individuata manualmente. | |

# Appendice

### PHPIPAM 1.1.010 - MULTIPLE VULNERABILITIES

phpIPAM [8] è un'applicazione di gestione degli indirizzi IP Web open source. Il suo obiettivo è fornire una gestione degli indirizzi IP leggera, moderna e utile. È un'applicazione basata su php con database back-end MySQL, che utilizza librerie jQuery, ajax e alcune funzionalità HTML5/CSS3.

La versione utilizzata dalla macchina target è la 1.1.010 che è vulnerabile a molteplici vulnerabilità:

- Stored XSS without authentication [9]
- Reflected XSS [9]
- SQL Injection [10]
- CSRF [11]

Una soluzione per mitigare tutte le vulnerabilità presenti è quella di aggiornare phpIPAM alla versione più recente.

# Riferimenti


[1] «CVE-2020-11022,» [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2020-11022.

[2] «CVE-1999-0524,» [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-1999-0524.

[3] «CWE-693,» [Online]. Available: https://cwe.mitre.org/data/definitions/693.

[4] «CWE-319,» [Online]. Available: https://cwe.mitre.org/data/definitions/319.html.

[5] «CWE-522,» [Online]. Available: https://cwe.mitre.org/data/definitions/522.

[6] «CVE-2012-3499,» [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2012-3499.

[7] «CWE-89,» [Online]. Available: https://cwe.mitre.org/data/definitions/89.html.

[8] «phpIPAM,» [Online]. Available: https://phpipam.net/.

[9] «Cross Site Scripting (XSS),» [Online]. Available: https://owasp.org/www-community/attacks/xss/.

[10] «SQL Injection,» [Online]. Available: https://owasp.org/www-community/attacks/SQL_Injection.

[11] «Cross Site Request Forgery (CSRF),» [Online]. Available: https://owasp.org/www-community/attacks/csrf.