

**Università degli Studi di Salerno**

Penetration Testing Report

Five86:1

Paolo Labanca | Corso di PTEH | A.A. 2022/2023

Sommario

[1 Executive Summary 2](#_Toc139472601)

[2 Engagement Highlights 2](#_Toc139472602)

[2.1 Accordo di non divulgazione 2](#_Toc139472603)

[2.2 Consegna stimata 3](#_Toc139472604)

[2.3 Tecniche e strumenti consentiti 3](#_Toc139472605)

[2.4 Ambito d’applicazione 3](#_Toc139472606)

[2.5 Processo di analisi 3](#_Toc139472607)

[3 Vulnerability Report 4](#_Toc139472608)

[4 Remediation Report 4](#_Toc139472609)

[5 Findings Summary 5](#_Toc139472610)

[6 Detailed Summary 6](#_Toc139472611)

[6.1 Critical 6](#_Toc139472612)

[6.2 High 8](#_Toc139472613)

[6.3 Low 10](#_Toc139472614)

[6.4 informative 13](#_Toc139472615)

[References 13](#_Toc139472616)

# 1 Executive Summary

Le attività del progetto per i corsi di Penetration Testing ed Ethical Hacking prevedono l'esecuzione di penetration test su macchine virtuali **Five86:1**. Lo scopo di questo progetto è analizzare la sicurezza della macchina target e proporre contromisure contro le vulnerabilità scoperte. Poiché non sono disponibili informazioni rilevanti sulla macchina target e sulla struttura della rete, viene utilizzato un approccio di tipo *black-box*. I test sono stati eseguiti sulla stessa rete locale delle macchine da analizzare e hanno simulato, il comportamento di un attaccante con accesso a tale rete. Le vulnerabilità rilevate potrebbero consentire ad un utente malizioso di ottenere pieno controllo della macchina causando anche gravi danni al sistema e agli utenti che usufruiscono dei servizi erogati dalla macchina target. Ciò potrebbe compromettere la disponibilità, l’integrità e la confidenzialità del sistema. Possiamo affermare, che il *livello di sicurezza* della macchina target è al quanto **basso**, invece per quanto riguarda il *rischio di compromissione,* risulta essere **alto***.*

Quindi, è necessario apportare delle modifiche che ci permettano di diminuire il livello di compromissione della macchina target, tramite degli aggiornamenti che siano mirati alla correzione delle vulnerabilità, in modo da riportare il sistema su livelli accettabili.

Le varie vulnerabilità saranno descritte successivamente nel dettaglio con le relative contromisure.

# 2 Engagement Highlights

Le regole di ingaggio, in questo caso, non sono state contestualizzate facendo questo parte di un’attività progettuale di tipo accademica e quindi non soggetta ad accordi di non divulgazione (**NDA**). In particolare, non sono stati presentati limiti riguardo agli strumenti e alle tecniche consentite a patto di non sconfinare la rete **NAT** creata appositamente per l’analisi di questa macchina. Di seguito riportate le sezioni delle regole d’ingaggio comuni. Usualmente tutto ciò che non viene definito non è consentito, in questo caso non vengono apposti limiti e l’analista ha dichiarato le tecniche e gli strumenti nell’apposito documento nella sezione **Strumenti utilizzati.**

## 2.1 Accordo di non divulgazione

Non sono stati siglati accordi di non divulgazione avente parte il docente del corso o l’ente dell’università degli studi di Salerno, pertanto, l’analisi è libera da accordi che ne vietino la divulgazione.

## 2.2 Consegna stimata

È stato stimato un tempo di completamento di 30 giorni lavorativi per poter effettuare l’analisi e la stesura dei documenti riguardanti gli strumenti e le metodologie utilizzate e il *Penetration Testing Report.*

## 2.3 Tecniche e strumenti consentiti

L’analisi non è caratterizzata da vincoli riguardo gli strumenti e le tecniche da utilizzare quindi è stata data libera scelta allo studente. Sono state definite le responsabilità legali qualora l’analisi possa compromettere macchine o servizi esterni alla macchina target o alla rete **NAT**, definita per l’attività di analisi.

## 2.4 Ambito d’applicazione

L’ambito d’applicazione è legato alla sola analisi della macchina target; pertanto, è vietato il recupero di informazioni tramite le forme di **Intelligence**: Human Intelligence e Signal Intelligence. Il processo di analisi non deve includere persone terze come potrebbe essere il creatore della macchina.

## 2.5 Processo di analisi

Durante il processo di Analisi non verranno segnalate le vulnerabilità gravi riscontrate. Queste verranno rese note al termine dell’analisi, essendo uno strumento di esercitazione didattico, effettuato su una macchina vulnerabile by design, questa non ha servizi pubblicamente accessibili, pertanto, non necessita di una segnalazione repentina delle vulnerabilità più gravi.

# 3 Vulnerability Report

Da un’analisi della macchina sono emerse alcune vulnerabilità che la espongono ad attacchi da parte di utenti maliziosi:

* La versione di openAdmin ci viene indicata sulla home page, ci viene detto anche di scaricare l’ultima versione. Facendo ciò si fa capire all’attaccante che si ha davanti una versione obsoleta e quindi dove ci potrebbero essere delle vulnerabilità.
* Alcuni file di configurazione come .htaccess e .htpasswd sono leggibili da tutti, tali file non dovrebbero essere accessibili a tutti.
* All’interno del file .htpasswd oltre all’hash ci sono delle informazioni aggiuntive che aiutano l’attaccante a creare una wordlist con lunghezza e tipologia di password indicati nei commenti del file, per accedere come un determinato utente.
* Alcuni utenti possono effettuare dei comandi “cp” come root, non si dovrebbero dare privilegi così elevati a un utente normale.
* All’interno di alcuni utenti ci sono dei file relativi alle credenziali che consentono l’accesso tramite OpneSSH in chiaro.
* Informazioni in chiaro riguardante la password dell’utente *moss* all’interno dell’e-mail,
* l’utente a cui si è fatto riferimento in precedenza fa parte della lista di **SUDOERS**, cioè, in un sistema UNIX, vuol dire che ha i permessi per diventare amministratore.
* È presente un file eseguibili con il bit SETUID attivo, tale file una volta eseguito consente di ottenere una shell di root che garantisce pieno controllo della macchina.

# 4 Remediation Report

Date le problematiche di sicurezza riscontrate durante l’attività di penetration testing dovrebbero essere messe in atto le seguenti strategie al fine di migliorare la sicurezza del sistema:

* Aggiornare la versione di OpenAdmin e non effettuare la stampa in chiaro della versione.
* Modificare i permessi dei file.
* Non utilizzare tecniche di occultamento delle informazioni all’interno di altri file, quindi; eliminare i commenti relativi ad alcune informazioni aggiuntive e molto utili all’attaccante e impostare tecniche di offuscamento dell’hash.
* Se necessario utilizzare tecniche di steganografia, utilizzare password per la lettura del contenuto molto robuste, in modo che un attacco a dizionario non abbia successo. Quindi utilizzare password lunghe pi`u di 16 caratteri, alfanumeriche, con caratteri speciali e non utilizzare password con nomi comuni o di breve lunghezza
* Modificare i permessi “sudo” su alcuni utenti relativi ai comandi che possono effettuare.
* Programmare un Penetration testing, come quello effettuato, fissando un numero di volte per il quale effettuarlo durante l’anno.

Si consiglia di risolvere tutte le vulnerabilità presentate in questo documento seguendo un ordine decrescente in base alla gravità: è consigliato dunque risolvere tempestivamente le vulnerabilità critiche e procedere successivamente alla correzione delle vulnerabilità con criticità più bassa.

# 5 Findings Summary

La tabella seguente mostra il numero di vulnerabilità individuate per categoria:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Severity | Info | Low | High | Critical |
| #Vulnerabilità | 41 | 6 | 5 | 4 |

Il grafico seguente mostra la distribuzione delle vulnerabilità per categoria:

Figura 1: Grafico a torta delle vulnerabilità

# 6 Detailed Summary

Di seguito verranno riportate e descritte dettagliatamente le vulnerabilità individuate, partendo da quelle più critiche fino a quelle meno critiche. Inoltre, per ognuna di essa verranno fornite alcune raccomandazioni su come mitigarle.

## 6.1 Critical

**Credenziali presenti all’interno di file in chiaro**

**Descizione**

All’interno di un file navigabile dal web server sono presenti le credenziali dell’utente di sistema

**Impatto**

Sfruttando questa password è possibile accedere all’utente che esegue il web server.

**Souzione**

Rimuovere la password in chiaro dal file, rimuovere il file oppure mitigare le debolezze di *full path disclosure, directory listing, information leakage*. [1]

**Password Deboli**

**Descrizione**

L’utente *douglas* stage utilizza una password facilmente calcolabile con un attacco del dizionario

**Impatto**

Questa password permette di avere accesso remoto alla macchina tramite connessione SSH

**Soluzione**

Aumentare la complessità delle password utilizzate, quindi utilizzare password lunghe, alfanumeriche, con caratteri speciali e assicurarsi che non facciano parte di dizionari comuni di password. [2]

**Cross Site Scripting**

**Descrizone**

Cross-site Scripting (XSS) è una tecnica di attacco che prevede l'eco del codice fornito da un utente malintenzionato nell'istanza del browser di un utente.

**Impatto**

Quando un utente malintenzionato ottiene il browser di un utente per eseguire il suo codice, esso verrà eseguito all'interno del contesto di sicurezza del sito web di hosting. Con questo livello di privilegio, il codice ha la capacità di leggere, modificare e trasmettere qualsiasi dato sensibile accessibile dal browser. Le conseguenze, riguardano: account dirottato (furto di cookie), browser reindirizzato a un'altra posizione o eventualmente mostrare contenuto fraudolento.

**Soluzione**

Per ogni pagina Web generata, utilizzare e specificare una codifica dei caratteri come ISO-8859-1 o UTF-8. Utilizzare una strategia di convalida dell'input, ovvero utilizzare un elenco consentito di input accettabili strettamente conformi alle specifiche.

Per mitigare gli attacchi XSS contro il cookie di sessione dell'utente, impostare il cookie di sessione su HttpOnly (non supportata da tutti i browser) . Ancora più importante, XMLHTTPRequest e altre potenti tecnologie browser che forniscono l’accesso in lettura alle intestazioni HTTP, inclusa l'intestazione Set-Cookie in cui è impostato il flag HttpOnly. [3] [4]

SQL Injection

**Descrizione**

SQL injectionè una tecnica di *code injection*, usata per [attaccare](https://it.wikipedia.org/wiki/Attacco_informatico) applicazioni che gestiscono dati attraverso [database relazionali](https://it.wikipedia.org/wiki/Database_relazionali) sfruttando il linguaggio [SQL](https://it.wikipedia.org/wiki/Sql).

**Impatto**

Il mancato controllo dell'input dell'utente permette di inserire artificiosamente delle [stringhe](https://it.wikipedia.org/wiki/Stringa_(informatica)) di [codice](https://it.wikipedia.org/wiki/Codice_sorgente) [SQL](https://it.wikipedia.org/wiki/SQL) che saranno eseguite dall'applicazione [server](https://it.wikipedia.org/wiki/Server). Grazie a questo meccanismo è possibile far eseguire comandi SQL, anche molto complessi, dall'alterazione dei dati (es. creazione di nuovi utenti) al [download](https://it.wikipedia.org/wiki/Download) completo dei contenuti nel database.

**Soluzione**

Concedere l'accesso minimo al database necessario per l'applicazione. Se l'applicazione utilizza JDBC, utilizzare PreparedStatement o CallableStatement, con i parametri passati da '?'. Se l'applicazione utilizza ASP, utilizzare ADO Command Objects con un controllo avanzato del tipo e query con parametri. Non creare query SQL dinamiche utilizzando una semplice concatenazione di stringhe. Applicare un 'elenco consentito' di caratteri consentiti o un 'elenco negato' di caratteri non consentiti nell'input dell'utente. [5]

## 6.2 High

**Limitazione non corretta di un percorso a un file o una directory limitata. Path Traversal**

**Descrizione**

Il web server espone file e directory a utenti non esplicitamente autorizzati ad accedervi ma non prevede protezioni.

**Impatto**

Sfruttando questa debolezza si può accedere a file e directory del web server e, in questo caso, trovare alcune credenziali di sistema.

**Soluzione**

Convalidare gli input, quando si convalidano nomi di file o percorsi, utilizzare liste consentite e, se possibile, non consentire mai la sequenza ”..” per evitare **Relative Path Traversal** [6]ed escludere ”/” per evitare **Absolute Path Traversal** [7]

**File con SETUID**

**Descrizione**

Il file upyourgame ha il SETUID attivo.

**Impatto**

Un utente con accesso al sistema può eseguire questo file con SETUID per ottenere una shell di root e avere pieno controllo della macchina.

**Soluzione**

Rimuovere il bit SETUID dal file.

**Assenza di token anti-CSRF**

**Descrizione**

Nessun token Anti-CSRF è stato trovato in un modulo di invio HTML. Una contraffazione di richieste cross-site è un attacco che consiste nel costringere una vittima a inviare una richiesta HTTP a una destinazione target a sua insaputa o senza intenzione, al fine di eseguire un'azione come vittima.

**Impatto**

CSRF viene utilizzato principalmente per eseguire un'azione contro un sito bersaglio utilizzando i privilegi della vittima. Il rischio di divulgazione di informazioni aumenta notevolmente quando il sito di destinazione è vulnerabile a XSS, poiché XSS può essere utilizzato come piattaforma per CSRF.

**Soluzione**

Utilizzare una libreria o un framework controllato che non consenta il verificarsi di questa debolezza o fornisca costrutti che rendano questa debolezza più facile da evitare. Ad esempio, utilizzare pacchetti anti-CSRF come OWASP CSRFGuard. Controlla l'intestazione HTTP Referer per vedere se la richiesta ha avuto origine da una pagina prevista. [8] [9]

**Intestazione CSP (Content Security Policy) non impostata**

**Descrizione**

Content Security Policy (CSP) è un ulteriore livello di sicurezza che aiuta a rilevare e mitigare determinati tipi di attacchi, tra cui Cross Site Scripting (XSS) e attacchi di data injection.

**Impatto**

Questi attacchi vengono utilizzati per qualsiasi cosa, dal furto di dati alla deturpazione del sito o alla distribuzione di malware.

**Soluzioni**

Assicurati che: il server Web, il server delle applicazioni, il bilanciamento del carico etc., siano configurati per impostare l'intestazione Content-Security-Policy. [10] [11]

Intestazione anti-clickjacking mancante

**Descrizione**

L' intestazione della risposta HTTP X-Frame-Optionspuò essere utilizzata per indicare se a un browser deve essere consentito o meno il rendering di una pagina.

La risposta non include la Content-Security-Policy con la direttiva ‘frame-ancestors’ o X-Frame-Options per la protezione dagli attacchi 'ClickJacking'.

**Impatto**

Un utente malintenzionato può anche scegliere di reindirizzare i clic per scaricare malware o ottenere l'accesso a sistemi vitali come punto di partenza per una minaccia persistente avanzata (APT).

**Soluzione**

I browser Web moderni supportano le intestazioni HTTP Content-Security-Policy e X-Frame-Options. Assicurati che uno di essi sia impostato su tutte le pagine web restituite dal tuo sito/app. Consigliamo di utilizzare SAMEORIGIN o DENY etc. [12]

## 6.3 Low

TCP timestamps

**Descrizione**

L’host remoto implementa i timestamp TCP e quindi consente di calcolare il tempo di attività.

**Impatto**

Un effetto collaterale di questa funzionalità `e che il tempo di attività dell’host remoto può talvolta essere calcolato.

**Soluzione**

Disabilitare i TCP timestamps all’interno del sistema: aggiungere la riga ”net.ipv4.tcp timestamps=0” in /etc/sysctl.conf. Lanciare successivamente il comando sysctl -p.

**ICMP Timestamp Reply Information Disclosure**

**Descrizione**

Il Timestamp Reply è un messaggio ICMP che risponde a un messaggio di Timestamp. È composto dal timestamp di origine inviato dal mittente del Timestamp, nonché da un timestamp di ricezione e da un timestamp di trasmissione.

**Impatto**

Queste informazioni potrebbero teoricamente essere utilizzate per sfruttare i generatori di numeri casuali deboli basati sul tempo in altri servizi

**Soluzione**

Disattivare completamente il supporto per il timestamp ICMP sull’host remoto. Proteggere l’host remoto con un firewall e bloccare i pacchetti ICMP che passano attraverso il firewall in entrambe le direzioni (completamente o solo per le reti non attendibili).

**Cookie Nessun flag HttpOnly**

**Descrizione**

È stato impostato un cookie senza il flag HttpOnly, il che significa che è possibile accedere al cookie tramite JavaScript.

**Impatto**

Se uno script dannoso può essere eseguito su questa pagina, il cookie sarà accessibile e può essere trasmesso a un altro sito. Se si tratta di un cookie di sessione, potrebbe essere possibile il dirottamento della sessione.

**Soluzione**

Assicurarsi che il flag HttpOnly sia impostato per tutti i cookie. [13]

**Cookie senza attributo SameSite**

**Descrizione**

L'attributo SameSite è una contromisura efficace contro la contraffazione di richieste tra siti

**Impatto**

È stato impostato un cookie senza l'attributo SameSite, il che significa che il cookie può essere inviato come risultato di una richiesta "cross-site".

**Soluzione**

Assicurarsi che l'attributo SameSite sia impostato su "lax" o idealmente su "strict" per tutti i cookie. [14]

**Perdita di informazioni del Server**

**Descrizione**

Il server web/applicativo perde informazioni sulla versione, tramite l'intestazione della risposta HTTP "Server".

**Impatto**

L'accesso a tali informazioni può facilitare agli aggressori l'identificazione di altre vulnerabilità a cui è soggetto il tuo server web/applicativo.

**Soluzione**

Assicurati che: il server Web, il server delle applicazioni, il sistema di bilanciamento del carico etc., siano configurati per sopprimere l'intestazione "Server" o fornire dettagli generici. [15] [16]

**Intestazione X-Content-Type-Options mancante**

**Descrizione**

L'intestazione X-Content-Type-Options di Anti-MIME-Sniffing non è stata impostata su "nosniff".

**Impatto**

Ciò, consente, alle versioni precedenti di Internet Explorer e Chrome di eseguire lo sniffing MIME sul corpo della risposta, causando potenzialmente l'interpretazione e la visualizzazione del corpo della risposta.

**Soluzione**

Assicurati che l'applicazione/server web imposti l'intestazione Content-Type in modo appropriato e che imposti l'intestazione X-Content-Type-Options su 'nosniff' per tutte le pagine web. Se possibile, assicurarsi che l'utente finale utilizzi un browser Web moderno e conforme agli standard. [17] [18]

## 6.4 informative

Le vulnerabilità informative sono quelle individuate in maggioranza dagli strumenti di scansione automatica delle vulnerabilità; tuttavia, non sono state riportate dato che non hanno rilevanza nei confronti di potenziali attaccanti.

# References

|  |  |
| --- | --- |
| [1] | https://cwe.mitre.org/data/definitions/256.. |
| [2] | https://cwe.mitre.org/data/definitions/1391.html.. |
| [3] | http://projects.webappsec.org/Cross-Site-Scripting . |
| [4] | http://cwe.mitre.org/data/definitions/79.html . |
| [5] | https://cheatsheetseries.owasp.org/cheatsheets/SQL\_Injection\_Prevention\_Cheat\_Sheet.html . |
| [6] | https://cwe.mitre.org/data/definitions/23.html.. |
| [7] | https://cwe.mitre.org/data/definitions/36.html.. |
| [8] | http://projects.webappsec.org/Cross-Site-Request-Forgery . |
| [9] | http://cwe.mitre.org/data/definitions/352.html. |
| [10] | https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing\_Content\_Security\_Policy . |
| [11] | https://cheatsheetseries.owasp.org/cheatsheets/Content\_Security\_Policy\_Cheat\_Sheet.html . |
| [12] | https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options. |
| [13] | https://owasp.org/www-community/HttpOnly. |
| [14] | https://tools.ietf.org/html/draft-ietf-httpbis-cookie-stesso-sito. |
| [15] | http://httpd.apache.org/docs/current/mod/core.html#servertokens . |
| [16] | http://msdn.microsoft.com/en-us/library/ff648552.aspx#ht\_urlscan\_007 . |
| [17] | http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx . |
| [18] | https://owasp.org/www-community/Security\_Headers. |
| [19] | http://cwe.mitre.org/data/definitions/352.html. |