CORSO DI PTEH

A.A. 2023/2024

Penetration Testing Report

Caso di studio: "shenron: 3"

DARIO MAZZA

Mat. 0522501553

d.mazza6@studenti.unisa.it

03 Ottobre 2024

Dipartimento di Informatica

Università degli Studi di Salerno

Indice

Elenco delle figure

CAPITOLO 1

Executive Summary

Il presente progetto di penetration testing, realizzato nell'ambito del corso di Penetration Testing and Ethical Hacking, aveva come scopo la valutazione della sicurezza della macchina virtuale denominata Shenron 3, messa a disposizione dalla piattaforma Vulnhub. L'obiettivo era identificare le vulnerabilità presenti nel sistema target e suggerire possibili soluzioni per la loro mitigazione. Questo tipo di attività è stato svolto utilizzando un approccio black-box, il che implica che non si disponeva di alcuna informazione preliminare riguardante la macchina, simulando una condizione di attacco da parte di un avversario esterno senza conoscenza interna.

Il penetration test è stato eseguito mediante una macchina Kali Linux, connessa alla stessa rete del target, con lo scopo di emulare le azioni di un possibile attaccante. Durante il test sono state individuate diverse vulnerabilità, che potrebbero consentire a un attaccante di ottenere il pieno controllo della macchina. Tali criticità rendono la macchina estremamente vulnerabile, minacciando la confidenzialità, l'integrità e la disponibilità delle informazioni gestite dal sistema.

Alla luce delle analisi condotte, si ritiene che il livello di sicurezza della macchina Shenron 3 sia attualmente inadeguato per contrastare attacchi informatici mirati, lasciando il sistema esposto a un rischio elevato di compromissione. Le sezioni successive del report forniscono una descrizione dettagliata delle vulnerabilità riscontrate,

delle metodologie utilizzate per la loro individuazione e delle raccomandazioni per implementare contromisure volte a ridurre il rischio.

CAPITOLO 2

Engagement Highlights

In questa parte del documento, viene descritto il contesto immaginario che ha permesso di eseguire il penetration test nel rispetto delle leggi e degli standard etici. Sebbene l'analisi sia stata effettuata in ambito accademico per scopi formativi, è stato simulato un accordo con una società fittizia, interessata a valutare la sicurezza della propria rete IT. A tale scopo, è stato stilato e firmato un contratto di riservatezza (NDA) per tutelare le informazioni trattate durante il test. L'attività di penetration test si è concentrata completamente sulla macchina Shenron 3. Le operazioni si sono svolte durante le ore lavorative standard, seguendo quanto previsto dagli accordi. In conformità con le linee guida stabilite, sono state impiegate tecniche di scansione attiva, mentre l'uso di strategie come l'ingegneria sociale è stato escluso per mantenere l'integrità didattica e rispettare i vincoli etici imposti. Per l'analisi sono stati utilizzati strumenti avanzati disponibili nella suite Kali Linux, adottando soluzioni tecniche moderne per identificare e valutare le vulnerabilità presenti. Questo approccio ha consentito di creare un ambiente di test realistico, permettendo un'esperienza pratica di valutazione della sicurezza in un contesto sicuro e controllato.

CAPITOLO 3

Vulnerability Report

L'analisi del sistema ha rilevato numerose vulnerabilità e debolezze che lo rendono vulnerabile a potenziali attacchi da parte di utenti malintenzionati, tra cui:

- **Assenza di Token Anti-CSRF**: L'assenza di token CSRF sui form rende il sito suscettibile ad attacchi di Cross-Site Request Forgery (CSRF), consentendo ad un attaccante di manipolare le azioni degli utenti autenticati.
- **Assenza dell'Header CSP:** Senza una Content Security Policy (CSP), attori malintenzionati possono inserire contenuti dannosi nelle pagine, facilitando attacchi di tipo Cross-Site Scripting (XSS).
- **Header Anti-clickjacking Mancante:** L'assenza di questa misura di protezione espone l'applicazione a possibili attacchi di clickjacking, aumentando il rischio che gli utenti interagiscano con elementi ingannevoli.
- **Inclusione di File JavaScript da Domini Esterni:** La possibilità di includere file JavaScript da domini esterni potrebbe permettere ad un attaccante di eseguire codice malevolo sul client, aumentando il rischio di attacchi XSS.

- **Divulgazione della Versione del Server:** La divulgazione della versione del server tramite l'header HTTP può fornire informazioni utili ad un attaccante per identificare vulnerabilità specifiche di quella versione.
- **X-Content-Type-Options Header Mancante:** L'assenza di questa intestazione può indurre i browser a interpretare erroneamente il MIME type dei file scaricati, aumentando il rischio di sicurezza.
- **Uso di Componenti Obsoleti:** Il server utilizza Apache/2.4.41, che potrebbe essere vulnerabile ad attacchi noti. Inoltre, la versione di WordPress non aggiornata presenta vulnerabilità note (XSS, RCE).
- **User Agent Fuzzer:** La risposta del server varia in base alle stringhe dell'User-Agent utilizzate, suggerendo che il server potrebbe essere suscettibile a exploit che utilizzano User-Agent specifici per eludere la sicurezza.
- **Navigazione delle Directory:** Identificate directory importanti come /wp-admin/, /wp-content/, /wp-includes/ e /server-status/, alcune delle quali possono rappresentare potenziali target per attacchi mirati.
- **Vulnerabilità di Brute Force al Pannello di Login Admin di WordPress:** La pagina di login dell'admin di WordPress è vulnerabile ad attacchi di tipo brute force, consentendo a un attaccante di tentare ripetutamente combinazioni di credenziali fino a trovare quella corretta.
- **Vulnerabilità all'Abuso della Variabile di Ambiente PATH:** La configurazione del sistema è vulnerabile all'abuso della variabile di ambiente PATH, consentendo ad un attaccante di manipolare il PATH per eseguire programmi o script non autorizzati.

CAPITOLO 4

Remediation Report

Considerando le problematiche di sicurezza rilevate durante il penetration testing, si raccomanda di adottare le seguenti misure per migliorare la sicurezza del sistema:

- **Assenza di Token Anti-CSRF:** Implementare token di validazione nelle richieste POST per verificare che ogni richiesta provenga dall'utente autenticato, bloccando così gli attacchi CSRF. Questo aiuterà a proteggere gli utenti da richieste indesiderate effettuate a loro insaputa.
- **Assenza dell'Header CSP:** Implementare una Content Security Policy (CSP) per specificare quali risorse possono essere caricate o eseguite nella pagina, prevenendo attacchi XSS e altri tipi di iniezione. Una CSP ben definita limiterà l'inclusione di script e contenuti da fonti non fidate.
- **Header Anti-clickjacking Mancante:** Aggiungere l'header X-Frame-Options a tutte le risposte HTTP contenenti contenuti sensibili per prevenire attacchi di clickjacking. Questo impedisce che il sito venga caricato all'interno di frame esterni, evitando interazioni ingannevoli.

- **Inclusione di File JavaScript da Domini Esterni:** Limitare l'inclusione di JavaScript a domini fidati e verificati per ridurre il rischio di attacchi XSS. Rivedere e aggiornare i riferimenti esterni per garantire che provengano da fonti sicure.

- **Divulgazione della Versione del Server:** Configurare il server web per non divulgare informazioni sulla versione del server nei banner o nelle risposte HTTP. Questo può essere fatto modificando le impostazioni di configurazione del server per minimizzare le informazioni visibili agli attaccanti.

- **X-Content-Type-Options Header Mancante:** Aggiungere l'header X-Content-Type-Options: nosniff alle risposte HTTP per impedire ai browser di interpretare erroneamente i tipi MIME dei file scaricati, riducendo il rischio di attacchi MIME sniffing.

- **Uso di Componenti Obsoleti:** Aggiornare il server Apache e WordPress alla versione più recente e supportata. Se non è possibile, valutare alternative più sicure o implementare contromisure aggiuntive per mitigare i rischi associati ai componenti vulnerabili.

- **User Agent Fuzzer:** Implementare controlli sul server per garantire che le risposte non varino in base alle stringhe dell'User-Agent utilizzate. Questo renderà più difficile per gli attaccanti sfruttare specifici exploit attraverso la manipolazione dell'User-Agent.

- **Navigazione delle Directory:** Disabilitare la navigazione delle directory sul server web configurando opportunamente il file .htaccess o le impostazioni del server. Questo impedirà agli utenti non autorizzati di visualizzare i file interni e le configurazioni che potrebbero essere sfruttate.

- **Vulnerabilità di Brute Force al Pannello di Login Admin di WordPress:** Implementare meccanismi di limitazione dei tentativi di login e utilizzare un CAPTCHA per ridurre il rischio di attacchi di tipo brute force. Inoltre, utilizzare credenziali robuste e politiche di autenticazione a due fattori per proteggere l'accesso all'area amministrativa.

- **Vulnerabilità all'Abuso della Variabile di Ambiente PATH:** Assicurarsi che la variabile di ambiente PATH sia correttamente configurata e che contenga solo percorsi sicuri. Limitare l'accesso agli script e ai programmi non autorizzati e verificare che le directory contenute nel PATH siano protette da modifiche non autorizzate. Assicurarsi inoltre, che qualsiasi script personalizzato utilizzi i percorsi assoluti dei comandi di sistema e non i percorsi relativi.

Si raccomanda di risolvere tempestivamente tutte le vulnerabilità identificate in questo documento, dando precedenza a quelle più critiche. Adottando un approccio graduale che affronta prima le vulnerabilità di gravità elevata e poi quelle meno critiche, è possibile garantire interventi mirati ed efficaci per migliorare la sicurezza complessiva del sistema e ridurre il rischio di compromissione.

CAPITOLO 5

Findings Summary

Nella tabella e nei grafici sottostanti sono riportate il numero di vulnerabilità rilevate classificate per Severity.

Severity	Info	Low	Medium	High	Critical
Vulnerabilities	1	4	4	2	0

Tabella 5.1: Vulnerabilità divise per categoria

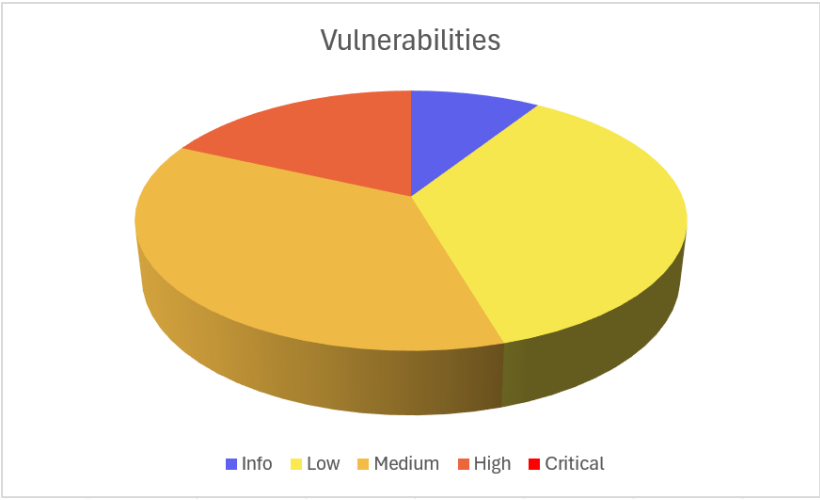

Figura 5.1: Grafico a torta

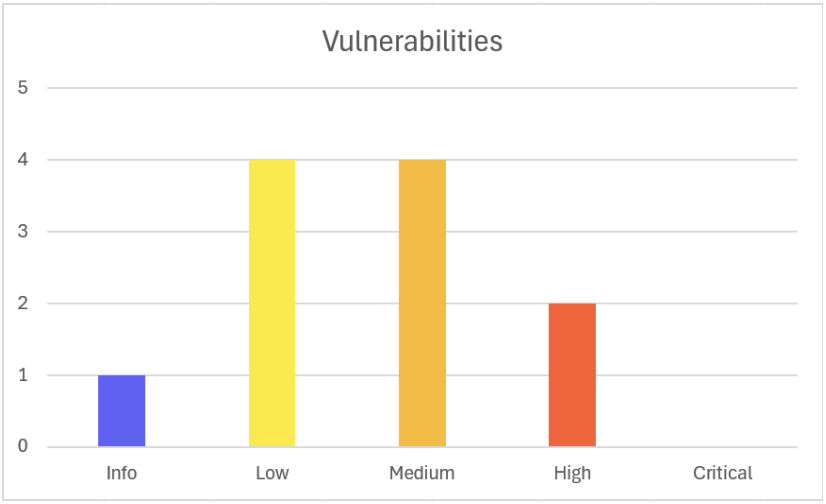

Figura 5.2: Istogramma

CAPITOLO 6

Detailed Summary

Questa sezione fornisce un riepilogo dettagliato delle vulnerabilità rilevate durante il penetration testing. Ogni vulnerabilità è classificata in base alla sua severità e include una descrizione, l'impatto potenziale e le soluzioni raccomandate per mitigare o risolvere il problema.

Vulnerability	Description	Impact	Solution
Uso di Componenti Obsoleti	Utilizzo di versioni obsolete di Apache vulnerabili	Potenziale esecuzione di codice remoto (RCE) e attacchi XSS	Aggiornare tutte le componenti alla versione più recente e supportata. Inoltre, verificare regolarmente la disponibilità di patch di sicurezza e applicarle tempestivamente. Valutare la sostituzione di componenti obsolete con alternative più sicure e moderne
Abuso della Variabile di Ambiente PATH	Manipolazione della variabile PATH per eseguire script malevoli con privilegi elevati	Potenziale ottenimento di privilegi root e compromissione del sistema	Proteggere e limitare l'accesso agli script di sistema, assicurarsi che solo utenti autorizzati possano modificarli e verificare la configurazione della variabile PATH. Rimuovere directory potenzialmente pericolose dalla variabile PATH e usare percorsi assoluti per eseguire comandi critici negli script

Tabella 6.1: Vulnerabilità con Severità Alta

Vulnerability	Description	Impact	Solution
Brute Force Login Admin WordPress	Mancanza di meccanismi di limitazione per il login al pannello admin di WordPress	Accesso non autorizzato mediante tentativi ripetuti	Implementare CAPTCHA per prevenire attacchi automatizzati, limitare il numero di tentativi di login per indirizzo IP e configurare l'autenticazione a due fattori (2FA) per migliorare la sicurezza dell'accesso. Inoltre, monitorare i log di accesso per rilevare tentativi di brute force e bloccare gli IP sospetti
Assenza di Token Anti-CSRF	Mancanza di token di convalida nelle richieste POST	Manipolazione delle azioni degli utenti autenticati mediante attacchi CSRF	Implementare token di validazione (CSRF token) per verificare la provenienza delle richieste. Assicurarsi che ogni richiesta sensibile includa un token univoco e che il server verifichi la validità del token. Inoltre, utilizzare framework o librerie che supportano automaticamente la protezione CSRF
Assenza di Content Security Policy (CSP)	Mancanza di una policy per specificare quali risorse possono essere caricate	Aumentato rischio di attacchi XSS	Implementare una Content Security Policy per limitare le risorse che possono essere caricate, specificando quali script, stili e altri contenuti sono consentiti. Configurare CSP per prevenire l'esecuzione di codice non autorizzato e ridurre l'esposizione agli attacchi XSS. Effettuare test approfonditi per assicurarsi che la policy non interferisca con le funzionalità legittime del sito
Header Anti-clickjacking Mancante	Assenza dell'header 'X-Frame-Options'	Possibile clickjacking che induce l'utente a compiere azioni non desiderate	Aggiungere l'header 'X-Frame-Options' con il valore 'DENY' o 'SAMEORIGIN' per impedire il caricamento del sito all'interno di frame esterni.

Tabella 6.2: Vulnerabilità con Severità Media

Vulnerability	Description	Impact	Solution
Navigazione delle Directory	Directory del server accessibili pubblicamente	Potenziale esposizione di file sensibili o informazioni interne	Disabilitare la navigazione delle directory nel server web configurando il file '.htaccess'. Assicurarsi che solo i file necessari siano accessibili pubblicamente e utilizzare permessi appropriati per limitare l'accesso ai file sensibili
X-Content-Type-Options Header Mancante	Assenza dell'header 'X-Content-Type-Options: nosniff'	Rischio che il browser interpreti erroneamente i tipi MIME dei file scaricati	Aggiungere l'header 'X-Content-Type-Options: nosniff' alle risposte HTTP per evitare che il browser interpreti i file in modo errato. Questa misura riduce il rischio di attacchi basati sul MIME type, come l'esecuzione di script non autorizzati. Configurare il server per includere automaticamente questo header in tutte le risposte
Divulgazione della Versione del Server	La versione del server viene divulgata nelle risposte HTTP	Attaccanti possono utilizzare queste informazioni per sfruttare vulnerabilità note	Configurare il server per non divulgare la versione nelle risposte HTTP. Rimuovere o oscurare le informazioni di versione dai banner del server e dalle intestazioni delle risposte per ridurre il rischio di exploit basati sulle versioni note vulnerabili
Inclusione di File JavaScript da Domini Esterni	Possibilità di includere file JavaScript da domini esterni	Aumentato rischio di attacchi XSS	Limitare l'inclusione di JavaScript a domini fidati e verificati utilizzando una Content Security Policy (CSP) per definire i domini autorizzati. Evitare di includere script da fonti non verificate e considerare l'utilizzo di Subresource Integrity (SRI) per garantire l'integrità degli script inclusi

Tabella 6.3: Vulnerabilità con Severità Bassa

Vulnerability	Description	Impact	Solution
User Agent Fuzzer	Il server risponde in modo diverso a seconda dell'User-Agent	Potenziale elusione delle protezioni basate sull'User-Agent	Uniformare le risposte del server indipendentemente dall'User-Agent, assicurandosi che il comportamento del server non vari sulla base delle stringhe User-Agent inviate. Questo riduce il rischio che attaccanti possano eludere le protezioni del server utilizzando User-Agent specifici

Tabella 6.4: Vulnerabilità Informative

Ogni vulnerabilità identificata è stata valutata per determinare il suo impatto potenziale, e sono state fornite soluzioni dettagliate per garantire una mitigazione efficace. Si consiglia di dare priorità alla risoluzione delle vulnerabilità con severità alta per ridurre al minimo il rischio di compromissione.