Università degli Studi di Salerno

Penetration Testing Report

NOTER MACHINE

Santonastaso Manlio | Corso di PTEH | A.A. 2021/2022

UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Sommario

Executive Summary

É stato condotto un'attività di penetration testing della macchina *Noter* messa a disposizione dal sito HackTheBox in modo da simulare le possibili azioni di un attaccante. Lo scopo del test è quello di determinare se un utente malintenzionato riesce ad accedere alla macchina sfruttando le debolezze presenti sulla macchina, inoltre, stimare l'impatto di un'eventuale intrusione.

L'attività è stata condotta avendo a disposizioni alcune informazioni come l'indirizzo *ip* della macchina e il sistema operativo; pertanto, si tratta di un approccio *grey box testing*.

Come risultato dell'attività di penetration testing sono state rilevate alcune vulnerabilità che possono permettere a un utente malintenzionato di accedere alla macchina *Noter* e la possibilità di mettere in crisi il servizio *FTP*; quindi, questo significa che il livello di sicurezza della macchina è basso e bisogna adottare con urgenze le contromisure descritte in questo documento, alla fine di aumentare il livello di sicurezza della macchina.

Engagement Highlights

Essendo un'attività a scopo didattico e non avendo una diretta contrattazione con il cliente, si è effettuato il processo di penetration testing utilizzando i tool visti a lezioni allo scopo di ricavare informazioni e condurre al meglio il processo di penetration testing.

Le fasi che sono state fatte sono:

- Target Discovery
- Enumeration Target & Port Scanning
- Vulnerability Mapping
- Exploitation
- Post-Exploitation (Privilege Escalation)
- Post-Exploitation (Mantaining Access)

Ognuno di queste fasi si è utilizzato diversi strumenti, in particolare nelle fasi di Target Discovery ed Enumeration Target & Port Scanning si è adoperato di Nmap, poi nella fase di Vulnerability Mapping si sono utilizzati tool automatici come Nessus e OpenVas, sempre nella stessa fase, si sono ricorsi a diversi strumenti come OWASP ZAP, Nikto2, Joomla Scan e dirb per analizzare le vulnerabilità nell'applicazione Web. Nella fase finale di Exploitation, soprattutto nella fase Mantaining Access, si è utilizzato lo strumento Metasploit, insieme all'ausilio di msfvenom, in maniera tale da preservare l'accesso alla macchina target (backdoor).

Vulnerability Report

Durante l'analisi della macchina *Noter* sono state trovate alcune vulnerabilità che espongono la macchina a violare i requisiti di sicurezza, ovvero confidenzialità, integrità e disponibilità, cosiddetta *triade CIA*, quindi permettendo ad un attaccante di compromettere la sicurezza dell'intero sistema.

Le vulnerabilità trovate sono diverse:

- Il servizio *VSFTPD 3.0.3* sulla porta 22 è vulnerabile ad un attacco *DoS* che potrebbe consentire a chiunque di mandare in crisi il servizio anche con una solo macchina, senza dover usufruire di grandi quantità di banda, come richiesto per un attacco di tipo *DoS*.
- La connessione al server avviene attraverso il protocollo *HTTP*, quindi le credenziali d'accesso di alcune pagine sono trasmesse in chiaro; pertanto, possono essere intercettate da un utente malintenzionato.
- *Cross-Site Scripting* (*Reflected*) permette ad un attaccante remoto di iniettare del codice malevolo nei contenuti di un sito *Web* esterno. Quando una vittima visualizza una pagina infetta, il codice malevolo viene eseguito nel browser della vittima.
- Cookie contraffatto (*forged cookie*) permette ad un attaccante, sfruttando una chiave segreta debole e messaggi di errori, riesce a riprodurre un cookie di un altro utente.
- Il package *md-to-pdf* prima della versione 5.0.0 è vulnerabile a *Remote Code Execution* (RCE) a causa dell'utilizzo della libreria *gray-matter*; pertanto, permette ad utente malintenzionato l'esecuzione di codice arbitrario da remoto in maniera tale da prendere il controllo della macchina.
- Il database *MariaDb* espone una vulnerabilità di *local privilege escalation* questo permette ad un attaccante che ha accesso al database come utente root, grazie alla presenza di credenziali d'accesso all'interno del codice sorgente, di effettuare *Vertical Privilege Escalation* passando da utente

normale a un utente amministratore(*root*) permettendo all'attaccante di avere il totale controllo della macchina.

Remediation Report

Durante l'attività di penetration testing sono state riscontrate diverse vulnerabilità che permettono ad un attaccante di ottenere pieno controllo sulla macchina *Noter*. Per porre rimedio alle vulnerabilità trovate dovrebbero essere messo in atto le seguenti strategie:

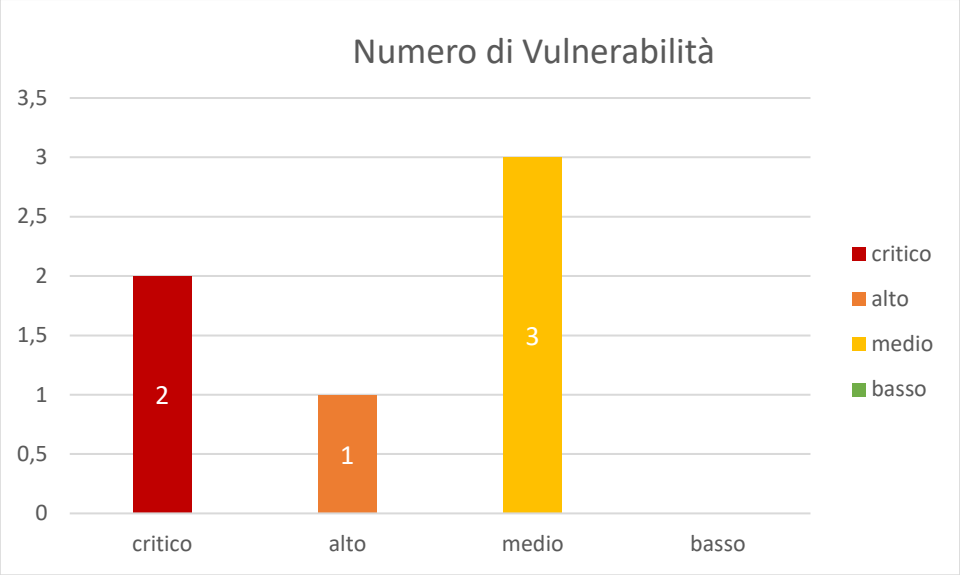
- Utilizzare un *firewall* o *Access Control List (*ACL) per controllare il traffico che raggiunge il servizio di *FTP*.
- Utilizzare la connessione HTTPS. Questo è importante soprattutto quando è necessario inviare informazioni sensibili come username e password, in maniera tale non permettere ad un attaccante di intercettare (*sniffing*) il traffico che transita sulla rete.
- Bisogna utilizzare tre ulteriori meccanismi di protezione per irrobustire le proprie sessioni *HttpOnly*, *Secure e SameSite,* tre attributi assegnabili ai cookie per richiedere al browser garanzie di sicurezza aggiuntive sul loro utilizzo. Inoltre, cambiare la chiave segreta in modo che non è possibile effettuare attacchi basati sul dizionario.
- Aggiornare md-to-pdf alla versione 5.0.0 o successiva
- I dati di accesso, nel nostro al caso al database, non devono mai essere inseriti all'interno dei sorgenti. Nei casi in cui non sia possibile, tali dati, non devono mai apparire in forma non cifrata (*plaintext*).

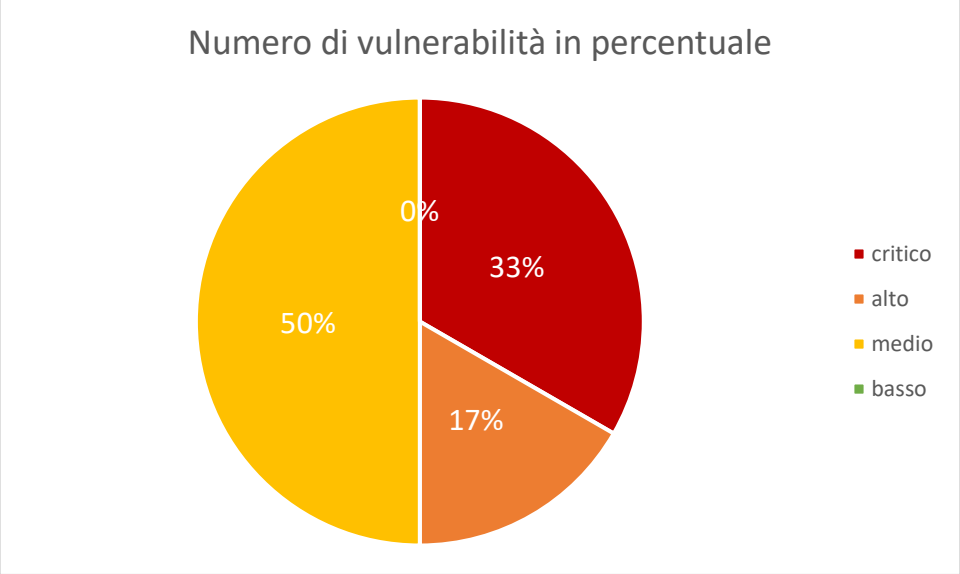
Findings Summary

Durante l'attività di penetration testing sono state riscontrate vulnerabilità e a ciascuna di esse è associato un livello di rischio:

- Critico: le vulnerabilità di questo tipo devono essere mitigate immediatamente in quanto rappresentano un serio pericolo per la sicurezza del sistema. Il loro sfruttamento non richiede tecniche avanzate, esperienza personale e particolare conoscenza dell'obiettivo.
- Alto: le vulnerabilità di questo tipo devono essere risolte al più presto in quanto mettono il sistema in pericolo, ma sono in genere più difficile da sfruttare per un attaccante che deve avere un'adeguata esperienza e strumenti adeguati.
- Medio: le vulnerabilità di tipo medio dovrebbe essere affrontate tempestivamente, ma sfruttarle è più difficile in quanto richiede social engineering o particolari circostanze.
- Basso: le vulnerabilità di questo tipo possono essere affrontate in un secondo momento, in quanto offrono ad un attaccante pochissime informazioni, quindi, non rappresentano una minaccia reale.

Nella figura sottostante verranno mostrati il numero di vulnerabilità trovate all'interno della macchina Noter.


Numero di Vulnerabilità
3,5
3
2,5
2
1,5
1
0,5
0
2
1
3
critico
alto
medio
basso
critico
alto
medio
basso


Numero di vulnerabilità in percentuale
0%
33%
50%
17%
critico
alto
medio
basso

Detailed Summary

Vulnerabilità Md-to-Pdf
Host: Noter
Rischio Alto
Descrizione Il package md-to-pdf prima della 5.0.0 è vulnerabile all'esecuzione di codice remoto (RCE) a causa dell'utilizzo della libreria *gray-matter* per analizzare il contenuto, senza disabilitare *JS engine*.
Spiegazione Analizzando il codice sorgente della *Web App* si è notato alla riga 308 utilizza la funzione md-to-pdf, versione antecedente della 5.0.0, eseguita utilizzando un'interfaccia a linea di comando (*bash*).
Rischi Un utente malintenzionato potrebbe sfruttare questa vulnerabilità effettuando la funzionalità di "export remote note" del sito iniettando codice arbitrario in maniera tale da instaurare una connessione con la macchina controllandola da remoto. [1]
Raccomandazione Aggiorna md-to-pdf alla versione 5.0.0 o successiva.

Vulnerabilità Forgery Cookie
Host Noter
Rischio Alto
Descrizione *Forged Cookie* è un cookie http pericoloso progettato per impersonare utenti validi senza conoscere la loro password indipendentemente dalla complessità della password e in alcuni casi può aggirare l'autenticazione a più fattori.
Spiegazione Per creare un cookie contraffatto si è decodificato localmente i dati di sessione (*cookie*), poi si è ottenuta la chiave segreta utilizzando un attacco di brute-force, dopodichè è possibile creare un cookie personalizzato firmando il cookie con la chiave segreta.
Rischi Un utente malintenzionato potrebbe sfruttare questa vulnerabilità autenticandosi come un altro utente compiendo delle operazioni illegittime.
Raccomandazione Utilizzare all'interno delle connessioni i flag *HTTPOnly, SECURE* e SameSite per rendere i cookie più sicuri permettendo di inviare i cookie solo HTTPS, in maniera tale l'attaccante che intercetta il canale di comunicazione dal browser al server non sarà in grado di leggere il cookie. Cambiare la chiave segreta in una più sicura, in modo tale da non permettere ad un attaccante di sfruttare attacchi basati sul dizionario (*CWE: Weak Password Requirements*). Esporre errori generici in fase di autenticazione (*login*), in maniera tale non permette ad un attaccante di scoprire quali utenti sono registrati alla *Web App (CWE: Exposure of Sensitive Information to an Unauthorized Actor).*

Vulnerabilità VSFTPD 3.0.3 - Remote Denial of Service
Host: Noter
Rischio Alto
Descrizione La versione VSTPD 3.0.3 è vulnerabile ad un attacco DoS in cui si fanno esaurire consapevolmente le risorse di un sistema che fornisce servizio al client.
Spiegazione Il servizio VSTPD consente di effettuare solo una certa quantità di connessioni al server, quindi, effettuando ripetutamente nuove connessioni al server, si può impedire ad altri utenti legittimi di effettuare una connessione al server, se le connessioni/ip non sono limitate (se è limitato, bisogna eseguire lo script da diversi proxy usando proxychains).
Rischi Un utente malintenzionato utilizzando questo script potrebbe impedire ad altri utenti legittimi di usufruire del servizio e causare danni a livello hardware e software.
Raccomandazione Utilizzare strumenti di monitoraggio della rete come Firewall, IDS (*Intrusion Detection System*) o IPS (Intrusion Prevention System), in maniera tale da controllare il traffico in entrata.

Vulnerabilità Connessione HTTP verso il Server
Host: Noter
Rischio Medio
Descrizione Il server trasmette informazioni sensibili come username e password in chiaro utilizzando il protocollo HTTP.
Spiegazione Le funzionalità di “registrazione” e “login” della *Web App,* si trovano nelle seguenti pagine http://noter/register e http://noter/login, richiedono all’utente di inserire informazioni sensibili che possono essere intercettate da un’attaccante.
Rischi Un utente malintenzionato utilizzando appositi strumenti potrebbe ottenere username e password di un utente e che potrà quindi identificarsi al sistema come quell’utente effettuando operazioni illegittime.
Raccomandazione Trasmettere informazioni utilizzando il protocollo HTTPS, in maniera tale la comunicazione avviene all’interno di una connessione criptata, come *Secure Socket Layer* (SSL), garantendo autenticazione e integrità dei dati trasmessi. Le informazioni scambiate tra il client e il server vengono protette da eventuali intercettazioni e compromissioni.

Vulnerabilità Cross-site-Scripting (*XSS Reflected*) Vulnerability
Host: Noter
Rischio Medio
Descrizione Nell'interfaccia Web è presente una vulnerabilità di XSS di tipo *reflected* che consente a un utente malintenzionato di eseguire codice dannoso nell'intento che gli utenti accedono link malevoli.
Spiegazione All'interno della pagina http://noter/login è possibile iniettare script dannoso nei campi "username" e "password". In seguito, è possibile vedere le linee di codice vulnerabili: *<input type="text" name="username" class="form-control" value="">* *.....* *<input type="password" name="password" class="form-control" value="">*
Rischi Un utente malintenzionato potrebbe creare degli script in grado di rubare (*forgery*) i cookie degli utenti in modo tale da impersonificarsi al posto dell'utente legittimo per compiere operazioni maliziose.
Raccomandazione Effettuare sanificazione dell'input e bloccare le richieste anomale.

Vulnerabilità User-Defined Function (UDF) Dynamic Library
Host: Noter
Rischio Medio
Descrizione User-Defined Function (UDF) [2] permette di effettuare *privilege escaltion* consentendo autorizzazioni e privilegi più alti a quelli assegnatigli.
Spiegazione Analizzando il codice all'interno è possibile trovare "in chiaro" i dati di accesso del database come *root*; quindi, questo permette di creare una libreria dinamica per effettuare *Vertical Privilege Escalation* passando da utente normale a utente *root*.
Rischi Un utente malintenzionato crea una libreria condivisa UDF e potrà passare da utente normale a utente *root* per cui avrà il totale controllo della macchina, inoltre, la possibilità di abbassare i privilegi di altri amministratori al livello di semplice utente.
Raccomandazione Non memorizzare i dati accesso all'interno del codice sorgente (*CWE-256: Plaintext Storage of a Password*). Non ospitare server Web e DBMS (*DataBase Management System*) sulla stessa macchina, ma fare in modo che il Web Server e il DBMS siano in due macchine diverse.

References

[1] https://github.com/simonhaenisch/md-to-pdf/issues/99

[2] https://www.codeguru.com/database/mysql-udfs/