Obiettivo del mio vulnerability scanning è la Metasploitable con indirizzo ip 192.168.50.101.

Host Information

Netbios Name:	METASPLOITABLE
IP:	192.168.50.101
MAC Address:	08:00:27:FE:80:23
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)

Il test è stato effettuato sul seguente range di porte: 21-3389.

Dallo screen sottostante si evincono le vulnerabilità individuate con i rispettivi gradi di priorità.

CRITICAL

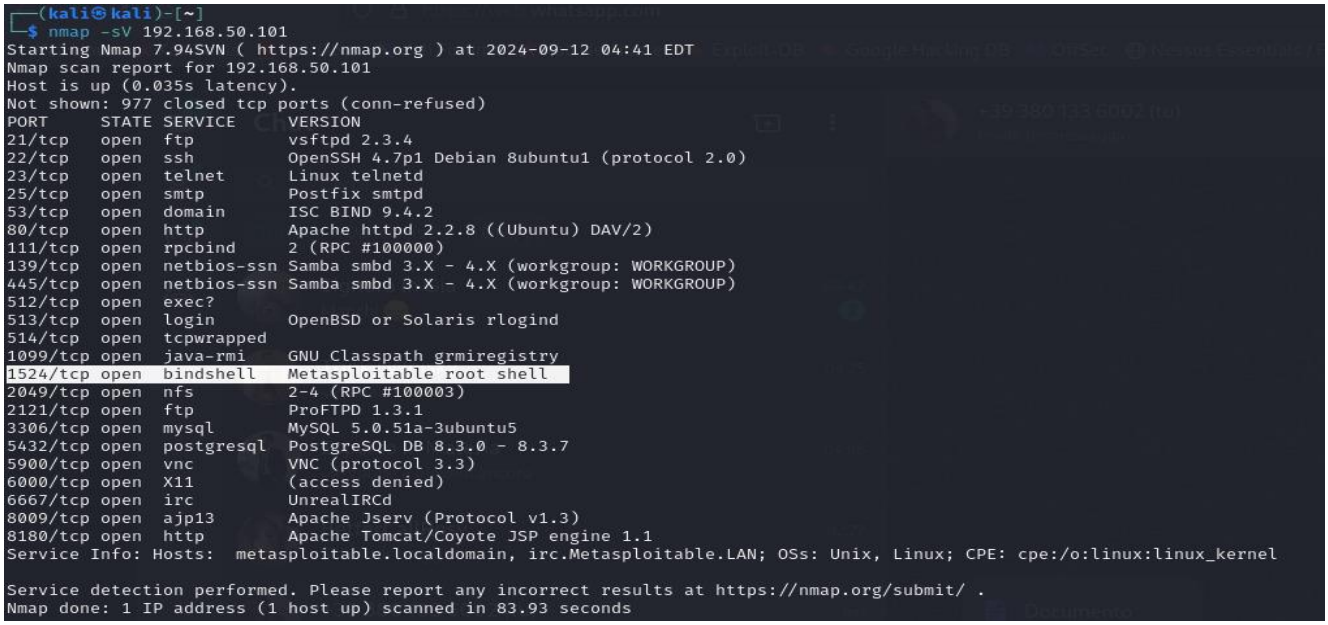
1. **Apache Tomcat A JP Connector Request Injection (Ghostcat)**

È stata riscontrata una vulnerabilità nella lettura/inclusione di file nel connettore AJP. Un aggressore remoto non autenticato potrebbe sfruttare questa vulnerabilità per leggere i file dell'applicazione web da un server vulnerabile. Nei casi in cui il server vulnerabile consente il caricamento di file, un utente malintenzionato potrebbe caricare codice JavaServer Pages (JSP) dannoso all'interno di una serie di tipi di file e ottenere l'esecuzione di codice remoto (RCE).

I server Apache Tomcat rilasciati negli ultimi tredici anni sono vulnerabili a un bug noto come “Ghostcat” (CVE-2020-1938) che consente agli hacker di prendere il controllo dei sistemi non patchati.
Scoperto dalla società cinese di cybersicurezza Chaitin Tech, Ghostcat è una falla nel protocollo Tomcat AJP. Stando a quanto si legge nel report pubblicato dall’azienda cinese, il bug riguarda **Tomcat AJP Connector**, che i ricercatori definiscono “il canale che consente a Tomcat di comunicare con l’esterno”. La vulnerabilità individuata, che interessa tutte le versioni di Tomcat rilasciate negli ultimi 13 anni, consentirebbe l’accesso in lettura e scrittura, oltre alla possibilità di impiantare backdoor sui sistemi nel caso sia possibile eseguire upload da remoto. Unica condizione perché l’exploit funzioni è che AJP Connector sia abilitato e che il pirata informatico sia in grado di collegarsi alla porta relativa al servizio stesso. La cattiva notizia è che AJP Connector **è abilitato come impostazione predefinita** in tutte le installazioni di Apache Tomcat; il produttore ha sviluppato e pubblicato le versioni aggiornate (9.0.31, 8.5.51 e 7.0.100) che correggono la vulnerabilità.

2. Bind Shell Backdoor Detection

Una backdoor “Bind Shell” è un software dannoso che si infiltra in un sistema informatico e crea una shell che ascolta le connessioni in entrata su una porta specifica. Ciò consente a un aggressore di accedere al sistema e di controllarlo, eseguendo comandi arbitrari da remoto. Questa vulnerabilità è particolarmente pericolosa perché non richiede alcuna forma di autenticazione.



3. Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Il problema riguarda il generatore random usato da openSSL per generare le chiavi crittografiche, che risulta predicibile, in quanto il seme usato per la generazione di numeri casuali in realtà è solo il PID del processo, che su linux è al massimo 32768.
Di conseguenza, per ogni tipologia di chiavi e per ogni architettura, il software è in grado di generare solo 32768 chiavi diverse. Chiunque può rigenerarle tutte facilmente e quindi le chiavi private sono da considerare compromesse.
Tutte le applicazioni che utilizzano le chiavi crittografiche di OpenSSL possono essere interessate dal problema: SSH, OpenVPN, i certificati X.509 e le chiavi usate nelle sessioni SSL/TLS (https, imaps, pops, etc.).
Il problema non è limitato ai sistemi Debian, ma interessa tutti i sistemi su cui siano state importate chiavi generate da sistemi Debian vulnerabili.
Ad esempio un utente può avere sul suo portatile una versione di Debian o Ubuntu con OpenSSL vulnerabile e con essa può generare una coppia di chiavi SSH da usare per l'autenticazione mediante chiave pubblica: gli basta copiare la chiave pubblica appena generata sugli host a cui ha accesso.
A questo punto questi host (non necessariamente con sistema Debian) diventano vulnerabili ad un attacco brute-force. Particolarmente grave è il caso in cui l'autenticazione con chiave pubblica compromessa sia abilitata per l'utente root.

4. **VNC Server 'password' Password**

Il server VNC in esecuzione sull'host remoto è protetto da una password debole. Nessus è stato in grado di effettuare il login utilizzando l'autenticazione VNC e una password di tipo “password”. Un attaccante remoto non autenticato potrebbe sfruttare questa situazione per prendere il controllo del sistema.

HIGH

1. **Samba Badlock Vulnerability**

La vulnerabilità “Badlock” era una falla di sicurezza scoperta nel software Samba, un'implementazione open-source del protocollo di rete SMB/CIFS, comunemente utilizzato per i servizi di file e stampa nei sistemi operativi Unix-like. La vulnerabilità è stata divulgata pubblicamente il 12 aprile 2016.

Lo sfruttamento di Badlock potrebbe consentire a un utente malintenzionato di eseguire codice arbitrario, eseguire attacchi man-in-the-middle e rubare informazioni sensibili.
Lo sfruttamento comporta tipicamente l'invio di richieste appositamente create a un server Samba vulnerabile. Manipolando queste richieste, gli aggressori potevano innescare problemi di buffer overflow o altri problemi di corruzione della memoria, portando all'esecuzione di codice arbitrario o alla divulgazione di informazioni sensibili.
Gli aggressori effettuano la scansione delle reti per identificare i sistemi che eseguono versioni vulnerabili di Samba. Una volta identificato un sistema vulnerabile, gli aggressori creerebbero richieste di rete appositamente progettate per sfruttare la vulnerabilità. Queste richieste contengono tipicamente dati malformati o attivano percorsi di codice specifici all'interno del software Samba che sono vulnerabili allo sfruttamento. Quando riceve le richieste dannose, il server Samba vulnerabile tenta di elaborarle. In caso di successo, il processo di sfruttamento innescherebbe la corruzione della memoria o altre debolezze della sicurezza all'interno del software Samba. La corruzione della memoria o la debolezza della sicurezza innescata dall'exploit potrebbe consentire all'aggressore di eseguire codice arbitrario sul sistema di destinazione. Questo codice potrebbe eseguire diverse azioni dannose, come prendere il controllo del server, rubare dati sensibili o lanciare ulteriori attacchi all'interno della rete.

2. **rlogin Service Detection**

Il servizio viene eseguito sulla porta 513 e consente agli utenti di accedere all'host da remoto.
Il servizio rlogin è in esecuzione sull'host remoto. Questo servizio è vulnerabile poiché i dati vengono passati tra il client rlogin e il server in chiaro. Un attaccante man-in-the-middle può sfruttare questa situazione per sniffare login e password. Inoltre, può consentire accessi scarsamente autenticati senza password.

MEDIUM

1. **DNS Server Cache Snooping Remote Information Disclosure**

Lo snooping della cache DNS si verifica quando qualcuno esegue una query su un server DNS per scoprire (snoop) se il server DNS ha un record DNS specifico memorizzato nella cache e dedurre quindi se il proprietario del server DNS (o i suoi utenti) ha visitato di recente un sito specifico. Ciò può rivelare informazioni sul proprietario del server DNS, ad esempio il fornitore, la banca, il provider di servizi e così via. Soprattutto se questo è confermato (snooped) più volte in un periodo. Questo metodo può anche essere usato per raccogliere informazioni statistiche, ad esempio a che ora il proprietario del server DNS accede in genere alla sua banca di rete e così via. Il valore TTL rimanente del record DNS memorizzato nella cache può fornire dati molto accurati.

LOW

1. **SSL/TLS Diffie-Hellman Modulus <= 1024 Bits (Logjam)**

L'host remoto consente connessioni SSL/TLS con uno o più moduli Diffie-Hellman inferiori o uguali a 1024 bit. Attraverso la crittoanalisi, una terza parte potrebbe essere in grado di trovare il segreto condiviso in un breve lasso di tempo (a seconda della dimensione del modulo e delle risorse dell'attaccante). Ciò può consentire a un aggressore di recuperare il testo in chiaro o di violare potenzialmente l'integrità delle connessioni.