GHOSTCAT

Ghostcat sfrutta configurazioni non sicure dell'AJP Connector di Apache Tomcat per consentire accessi non autorizzati ai file e potenzialmente eseguire codice malevolo.

Cos'è l'AJP?

L'AJP Connector è un componente utilizzato nei server web e nei server di applicazioni per gestire le comunicazioni tramite il protocollo AJP (Apache JServ Protocol) Il suo ruolo principale è quello di facilitare il trasferimento efficiente delle richieste HTTP dal web server all'application server e delle risposte HTTP dall'application server al web server. Questo avviene grazie al fatto trasmette i dati in codice binario così da rendere più snella la comunicazione.

Descrizione di un attacco Ghostcat

- L'attaccante inizia con una scansione di rete per identificare server che eseguono Apache Tomcat. Tramite strumenti come Nmap può trovare la porta aperte dell'AJP Connector (di default la porta è la 8009)
- Dopo aver individuato un server con la porta 8009 aperta, l'attaccante verifica se l'AJP Connector è vulnerabile tramite strumenti come ajpShooter o script personalizzati che possono essere utilizzati per inviare richieste AJP.
- Se il server è vulnerabile, l'attaccante invia richieste AJP per leggere file sensibili sul server.

Mitigazioni per Ghostcat

- Aggiornamento: la vulnerabilità può essere risolta aggiornando Tomcat server alle versioni 7.0.100, 8.5.51, 9.0.31 o superiori.
- Modificare le configurazioni del AJP connector: nella cartella usr/share/tomcat5.5/conf/ apriamo con il comando nano il file server.xml e troviamo la voce AJP AJP 1.3 connector. Da qui potremmo:
    a. Disabilitare il Connettore AJP commentando la riga <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />;
    b. Limitare l'Accesso al Connettore AJP: si può limitare l'accesso solo a indirizzi IP specifici modificando la stringa <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" address="127.0.0.1" />;
    c. Secret key: nelle versioni più aggiornate di Tomcat è possibile inserire una secret key

NFS Exported Share Information Disclosure

NFS è un protocollo di rete standard che consente a sistemi operativi UNIX-like di montare condivisioni di file da un server NFS remoto. L'exploit in questione sfrutta il fatto che le condivisioni NFS esportate da un server possono rivelare informazioni sensibili, come percorsi di file, configurazioni di rete, e altre informazioni di sistema.

Descrizione di un attacco su NFS

- Gli attaccanti interrogano un server NFS per ottenere una lista dei file delle condivisioni di file esportate. Questo può essere fatto utilizzando strumenti come showmount (che è un commando per visualizzare la condivisione dei file) in ambiente UNIX o attraverso richieste RPC (Remote Procedure Call) manipolate.
- Ottenendo informazioni su condivisioni esportate, gli attaccanti possono mappare la struttura di file directory del server NFS, identificare file sensibili, scoprire configurazioni di rete e trovare punti deboli nel sistema.
- Gli attaccanti possono utilizzare queste informazioni per pianificare ulteriori attacchi, come tentativi di accesso non autorizzato a file sensibili, eseguire attacchi di tipo directory traversal o preparare attacchi mirati basati sulla conoscenza della topologia di rete e delle configurazioni di sistema.

Mitigazioni

- Configurazione di /etc/exports: si può modificare il file exports per modificare i file condivisi. La configurazione di base della metasploitable è:
  / *(rw,sync,no_root_squash,no_subtree_check)
  Questo permette a chiunque di accedere con i permessi di scrittura e lettura.
  È possibile modificare la stringa per cambiare i permessi, impostare i path delle cartelle da condividere e gli indirizzi IP a cui è concesso farlo.
- Configurare hosts.deny: per ulteriore sicurezza è possibile modificare il file hosts.deny per impedire l'accesso ad uno specifico IP
- Aggiungere una regola nel firewall iptables.

Bind Shell Backdoor Detection

Una backdoor con shell bind (bind shell backdoor) è una grave vulnerabilità di sicurezza che permette a un attaccante di ottenere accesso remoto non autorizzato al sistema. Una bind shell apre una porta sulla macchina della vittima e attende connessioni in ingresso, consentendo agli attaccanti di eseguire comandi sul sistema compromesso.

Mitigazione

Per eliminare la vulnerabilità possiamo inserire una regola di firewall per bloccare l'accesso alla porta vulnerabile (in questo caso 1524).
Andiamo a creare una regola con questa riga di comando:
sudo iptables -A INPUT -p TCP —-dport 1524 -j DROP
Così abbiamo inserito una regola che scarta un qualunque tentativo di connessione sulla porta 1524 con protocollo TCP

VNC server password debole

Un VNC (Virtual Network Computing) server consente di controllare un computer remoto utilizzando un protocollo di condivisione dello schermo. Una password debole permette ad un attaccante di penetrare facilmente tramite un brute force e prendere il controllo del sistema.

Mitigazione

Uso di una password complessa: tramite il comando vncpasswd possiamo modificare la password di accesso a VNC server. Per rendere effettive le modifiche bisogna prima diventare utente root tramite il comando sudo su, poi inserire il commando vcnpasswd. Questo perché nel momento in cui qualcuno si collega alla porta di VNC (5900) viene usato direttamente l'utente root e non msfadmin.
La password inserita rispetta le seguenti regole:

- Complessità: Utilizza una combinazione di lettere maiuscole e minuscole, numeri e caratteri speciali.
- Unicità: Non utilizzare parole comuni, frasi o informazioni personali facilmente reperibili.
- Varietà: Non riutilizzare le stesse password per diversi account o servizi.
- Evita pattern prevedibili: Evita sequenze di caratteri semplici come "123456", "password", "qwerty" o simili.

Nota: La lunghezza massima accettata per la password di VNC e di 8 caratteri