COSE ANCORA DA FARE/SCRIVERE:

* Promemoria: controllare che i nomi dei programmi corrispondano a quelli degli screen e a quelli consegnati in file .py a parte
* Va scritta la sezione in cui descriviamo il design di rete e motiviamo le scelte;
* Vogliamo abbellire il design di rete in Canva o altri?
* Occorre dettagliare meglio le caratteristiche del primo programma di Brute Force utilizzato su phpMyAdmin e aggiungere gli screen per completare la sezione.
* Manca tutta la sezione sul Brute Force di DVWA e gli screen per ogni livello.
* Le contromisure per ora sono SOLO generiche; occorrerà scrivere quelle specifiche per la Theta e specificare con quale cadenza implementare alcune delle misure di sicurezza (to do list la rotazione delle password)
* Vogliamo inserire un fattore di rischio numerico?
* Se vogliamo si può inserire un indice con i titoli cliccabili delle varie sezioni

# Introduzione

L’azienda Theta ci ha ingaggiato per eseguire valutazioni di sicurezza su alcune infrastrutture critiche del loro data center.

Il perimetro delle attività si concentra principalmente su:

Un Web server che espone diversi servizi su internet (e quindi accessibili al pubblico)

Un Application server che espone sulla rete interna un applicativo di e-commerce accessibile dai soli impiegati della compagnia Theta (quindi non accessibile da resti esterne, ovvero internet)

# Metodologia

L’azienda ha chiesto di riprodurre Web Server e Application Server in ambiente di test così da non effettuare test invasivi sull’ambiente di produzione. Per questo, ci siamo avvalsi di Metasploitable per simulare il Web Server di Theta, mentre abbiamo utilizzato la macchina di Kali nel ruolo di attaccante.

Come da ulteriore richiesta, non abbiamo utilizzato tool già disponibili in Kali, ma abbiamo programmato i tool da zero in linguaggio Python.

Il CISO ha stabilito i seguenti task da portare a termine:

1. Design di rete per la messa in sicurezza delle componenti critiche oggetto di analisi;
2. Programma in Python per l’enumerazione dei metodi HTTP abilitati su un determinato target
3. Programma in Python per la valutazione dei servizi attivi (port scanning);
4. Report degli attacchi Brute Force sulla pagina phpmyadmin con evidenza della coppia username-password utilizzata per ottenere accesso all’area riservata;
5. Report degli attacchi Brute Force sulla DVWA per ogni livello di Sicurezza (low, medium, high);
6. Report totale che include i risultati trovati e le contromisure da adottare per ridurre eventuali rischi;

Nelle sezioni di seguito vedremo lo svolgimento di ciascun task.

# Design di rete per la messa in sicurezza delle componenti critiche oggetto di analisi

# A diagram of a network Description automatically generated

Il CISO della compagnia Theta ci ha richiesto di proporre un modello (design) di rete per mettere in sicurezza le due componenti critiche dell’azienda (Web Server e Application Server), includendo nell’analisi i dispositivi di sicurezza che potrebbero servire per aumentare la protezione della rete.

Il nostro design di rete prevede tre zone principali: una DMZ (zona demilitarizzata), una rete interna e una rete esterna. Occorrerà inoltre predisporre due aree fisiche all’interno dell’azienda: la Sala Server, che include i server della DMZ e quelli della rete interna, e la Administrator room, che include il rooter e il PC dell’Amministratore, anch’essi collegati alla linea interna.

DMZ (Zona Demilitarizzata):

La DMZ serve a ospitare servizi accessibili dall'esterno, come server web o servizi di posta elettronica, in modo che siano isolati dalla rete interna per motivi di sicurezza.

Nel nostro caso la DMZ contiene il Server Metasploitable, il DNS e il Web Server.

Rete Interna:

La rete interna è la parte della rete aziendale che contiene risorse e servizi critici per l'organizzazione. Nel nostro caso ospita un Server Metasploitable 2 di backup, un altro Server di backup e l’Application Server. Ospita inoltre il PC dell’Amministratore con relativo router e 6 PC per i dipendenti.

Rete Esterna:

La rete esterna rappresenta l'area al di fuori della rete aziendale.

Dispositivi di sicurezza:

Consigliamo un

# Programma in Python per la valutazione dei servizi attivi (port scanning);

Il **port scanning** è una tecnica utilizzata in sicurezza informatica per identificare i servizi o le porte aperte su un sistema informatico. Consiste nel sondare un range di porte su una macchina per determinare quali di esse sono in ascolto e quali servizi sono attivi. Questo può essere utile sia per garantire che solo i servizi necessari siano attivi, sia per rilevare quelle vulnerabilità che potrebbero essere sfruttate per lanciare attacchi mirati.

Per eseguire il port scanning su Metasploitable, abbiamo programmato **Port\_scanner.py**. Il programma riceve in input l’IP della macchina di cui vogliamo scansionare le porte e restituisce le porte aperte e i servizi attivi su ciascuna di esse:

A screenshot of a computer program

Description automatically generated

Abbiamo scansionato le porte dalla 1 alla 1024 e risultano aperte le seguenti porte con i seguenti servizi:

A screenshot of a computer

Description automatically generated

**Si raccomanda di chiudere le porte non necessarie.**

# Programma in Python per l’enumerazione dei metodi HTTP abilitati su un determinato target

Conoscere i **metodi HTTP** attivi su una porta consente agli amministratori di sistema di identificare i servizi web in esecuzione su un server. Questo è utile per la gestione e la manutenzione della rete. Se un server supporta metodi obsoleti o non sicuri, è necessario prendere provvedimenti per mitigare i rischi associati. Se vengono utilizzati metodi noti per essere vulnerabili, gli amministratori possono adottare misure preventive o correttive.

Vi è inoltre un aspetto di efficientamento non direttamente legato alla sicurezza informatica ma comunque importante a livello di sistema: conoscere i metodi HTTP attivi può aiutare nell'ottimizzazione e nella gestione del traffico di rete. Ad esempio, potrebbe essere necessario indirizzare o filtrare specifici metodi per migliorare le prestazioni.

Per ottenere informazioni sui metodi abilitati sulla nostra porta target abbiamo elaborato un secondo programma “NOME\_PROGRAMMA.PY”:

—------- INSERIRE SCREEN CODICE—---------

L’obiettivo di questo script Python è effettuare una richiesta HTTP di tipo OPTIONS a un server di destinazione specifico e visualizzare i metodi HTTP consentiti dal server.

Il programma inizia richiedendo all’utente di inserire l’indirizzo IP del server di destinazione (nel nostro caso quello di Metasploitable) e, se necessario, la porta. Se questa non viene specificata viene utilizzato il valore di default 80 (che è la porta genericamente associata ai servizi HTTP).

Successivamente, il programma tenta di stabilire una connessione al server utilizzando la classe ‘HTTPConnection’ del modulo ‘http.client’. Una volta stabilita la connessione, viene inviata una richiesta HTTP di tipo OPTIONS al server, specificando il percorso della richiesta (nel nostro caso http://192.168.XX.YYY/phpMyAdmin/phpMyAdmin.html).

Dopo aver ricevuto la risposta dal server, il programma estrae i metodi consentiti dalla risposta, analizzando il campo dell’intestazione ‘Allow’. Questi metodi vengono quindi stampati a schermo uno per uno.

Infine, il programma chiude la connessione al server.

In caso di connessione rifiutata (ad esempio, se il server non è in ascolto sulla porta specificata), gestisce un’eccezione ‘ConnectionRefusedError’ e stampa un messaggio d’errore.

—-----EVENTUALE SCREEN DI CONNESSIONE NON RIUSCITA—----

Lanciando il programma, abbiamo rilevato che sulla porta 80 di Metasploitable sono attivi i seguenti metodi: GET, HEAD, POST, OPTIONS, TRACE:

A screenshot of a computer screen

Description automatically generated

Elenchiamo brevemente alcune vulnerabilità associate a ciascun metodo:

GET:

I parametri della richiesta GET sono inclusi nell'URL, quindi possono essere memorizzati nei log del server web o nei log intermediari, esponendo potenzialmente informazioni sensibili come token di sessione.

HEAD:

E’ simile a GET, ma restituisce solo le intestazioni della risposta senza il corpo. Valgono le medesime considerazioni sulla raccolta di informazioni sensibili.

POST:

Injection di parametri: I dati del corpo della richiesta POST possono essere soggetti ad attacchi di injection (ad esempio, SQL injection o XSS) se non vengono trattati correttamente dal lato server.

Cross-Site Request Forgery (CSRF): Le richieste POST possono essere soggette a attacchi CSRF se non vengono adottate misure di sicurezza adeguate, come l'uso di token CSRF.

OPTIONS:

La richiesta OPTIONS può restituire informazioni sulla configurazione del server, rivelando dettagli potenzialmente sensibili.

TRACE:

Il metodo TRACE può essere sfruttato per eseguire attacchi di Cross-Site Tracing, in cui un malintenzionato può ottenere informazioni sulle intestazioni della richiesta inviata da un client. TRACE è spesso disabilitato per motivi di sicurezza.

Come già detto, l'utilizzo dei metodi HTTP dovrebbe essere limitato solo a quelli necessari per l'applicazione.

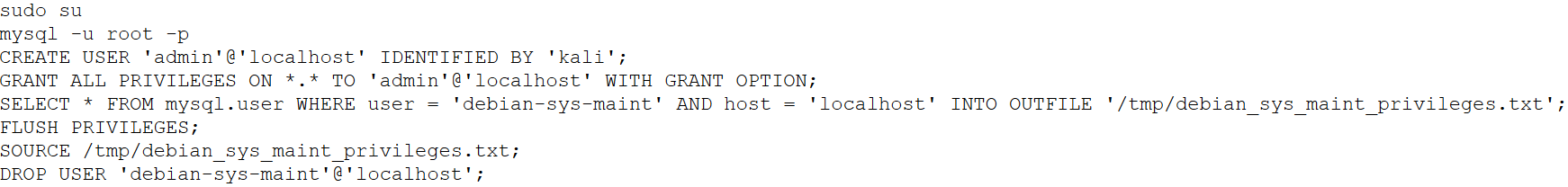
# Report degli attacchi Brute Force sulla pagina phpmyadmin con evidenza della coppia username-password utilizzata per ottenere accesso all’area riservata;

L’attacco **brute force** è una tecnica utilizzata in ambito di cybersecurity per ottenere accesso non autorizzato a un sistema o a un account. In questo tipo di attacco, l'attaccante cerca di indovinare la password di un account provando una serie di possibili combinazioni, spesso sfruttando un "dizionario" contenente parole comuni, frasi, o combinazioni di caratteri. L'esecuzione di attacchi controllati aiuta a valutare quanto un sistema sia esposto a rischi di sicurezza e quale potrebbe essere l'impatto di un attacco reale.

Nel caso in oggetto, ci è stato richiesto di effettuare un attacco Brute Force dizionario per effettuare il login su PhpMyAdmin.

Come prima cosa, ci spostiamo su Metasploitable e andiamo a settare la macchina creando un utente “admin” cui associamo la password “kali”.

Lo facciamo utilizzando i seguenti comandi:



Che implementiamo naturalmente su Metasploitable:

A screenshot of a computer screen

Description automatically generated

Ora ci spostiamo su Kali. Per eseguire il Brute Force scriviamo un terzo programma in Python.

Questo programma si avvale delle liste di username e password più comuni già presenti su Kali e tramite due cicli for, l’uno nidificato nell’altro, prova tutte le combinazioni finché il codice di risposta del sito non risulta essere 200. A quel punto stampa “Successo !!” per comunicarci qual è la coppia corretta di credenziali per entrare e interrompe l’esecuzione.

A screenshot of a computer

Description automatically generated

Poiché conosciamo già username e password corretti avendoli creati noi, per ottimizzare il processo modifichiamo la lista username spostando “admin” in cima e inseriamo invece nella lista password “kali”, anche qui in cima alla lista.

Pochi secondi dopo il lancio del Brute Force, il programma ci comunica di aver trovato la coppia corretta, come da screen di seguito:

A screenshot of a computer

Description automatically generated

# Report totale che include i risultati trovati e le contromisure da adottare per ridurre eventuali rischi;

## Contromisure da adottare per ridurre eventuali rischi

Politiche di sicurezza: Definire e attuare politiche di sicurezza chiare e ben definite per proteggere le risorse aziendali. Queste politiche dovrebbero coprire l'accesso autorizzato, la gestione delle password, l'uso sicuro dei dispositivi, e altro ancora.

Formazione degli Utenti: Fornire formazione sulla sicurezza informatica ai dipendenti per aumentare la consapevolezza sui rischi e insegnare pratiche di sicurezza come la creazione di password robuste e la gestione delle e-mail.

Gestione delle Password: Implementare politiche robuste per la gestione delle password, inclusa la richiesta di password complesse, la rotazione periodica delle password e l'autenticazione a due fattori (2FA) quando possibile.

Aggiornamenti e Patch: Mantenere tutti i sistemi operativi, applicazioni e software di sicurezza aggiornati con le ultime patch e aggiornamenti di sicurezza per mitigare vulnerabilità note.

Protezione Antivirus e Antimalware: Installare e mantenere aggiornati programmi antivirus e antimalware su tutti i dispositivi per rilevare e prevenire attacchi malware.

Firewall: Configurare e mantenere firewall per proteggere la rete aziendale da accessi non autorizzati e attacchi provenienti dall'esterno.

Controllo degli Accessi: Implementare un sistema di controllo degli accessi che limiti l'accesso alle risorse aziendali solo a coloro che ne hanno bisogno per svolgere le proprie mansioni.

Backup Regolari: Effettuare regolarmente il backup dei dati aziendali critici e assicurarsi che i backup siano archiviati in modo sicuro. In caso di attacco ransomware o perdita di dati, sarà possibile ripristinare le informazioni.

Monitoraggio della Sicurezza: Implementare strumenti di monitoraggio della sicurezza per rilevare e rispondere tempestivamente agli incidenti di sicurezza. Questo può includere il monitoraggio degli accessi, la rilevazione delle minacce e l'analisi dei log.

Politiche di Utilizzo dei Dispositivi Mobili: Se i dipendenti utilizzano dispositivi mobili aziendali o personali, implementare politiche chiare sulla sicurezza dei dispositivi mobili, inclusa la crittografia dei dati e il controllo remoto.

Consapevolezza della Privacy: Rispettare le normative sulla privacy dei dati, come il GDPR, e garantire che le pratiche aziendali siano conformi a tali normative.

Test di Sicurezza e Audit: Condurre regolarmente test di sicurezza e audit per identificare e correggere potenziali vulnerabilità prima che possano essere sfruttate da attaccanti.

Queste sono solo alcune delle pratiche di sicurezza di base che le aziende possono adottare. La sicurezza informatica è un processo continuo che richiede una vigilanza costante e l'adattamento alle nuove minacce. Le raccomandazioni specifiche possono variare in base alle esigenze e al contesto specifico di ciascuna azienda.