



Valutazione sicurezza reparto IT

All'attenzione di: CISO di THETA

16 febbraio 2024

GitHub: https://github.com/francescoficetti/CS0124_BW1/tree/main

INDICE

1. INTRODUZIONE
2. DESIGN DI RETE
3. PROGRAMMA PER L'ENUMERAZIONE DEI METODI HTTP
4. PROGRAMMA PER IL PORT SCANNING
5. ATTACCO BRUTE FORCE
6. CONTROMISURE DA ADOTTARE E CONSIGLI
7. LINK UTILI

1. INTRODUZIONE

È con piacere che presentiamo il seguente report, dedicato alla valutazione e alla raccomandazione delle strategie di sicurezza informatica per la vostra organizzazione. Esso si propone di fornire un'analisi esaustiva dello stato attuale della sicurezza dell'organizzazione, identificando le aree di vulnerabilità e suggerendo soluzioni e best practices per mitigare i rischi e rafforzare la protezione dei dati sensibili. Come richiesto dal cliente, abbiamo virtualizzato il contesto di rete, usando Metasploitable, per condurre test non invasivi.

2. DESIGN DI RETE

Nell'attuale panorama delle reti informatiche, la progettazione di un'architettura di rete efficace e sicura riveste un ruolo fondamentale per garantire la disponibilità e l'integrità dei servizi offerti. Il presente report si concentra sull'implementazione di una rete a due zone, composta da un Web server e un Application server, con l'obiettivo di fornire una panoramica dettagliata della configurazione di rete adottata e delle misure di sicurezza implementate.

Abbiamo sviluppato un'architettura di rete costituita da due elementi fondamentali:

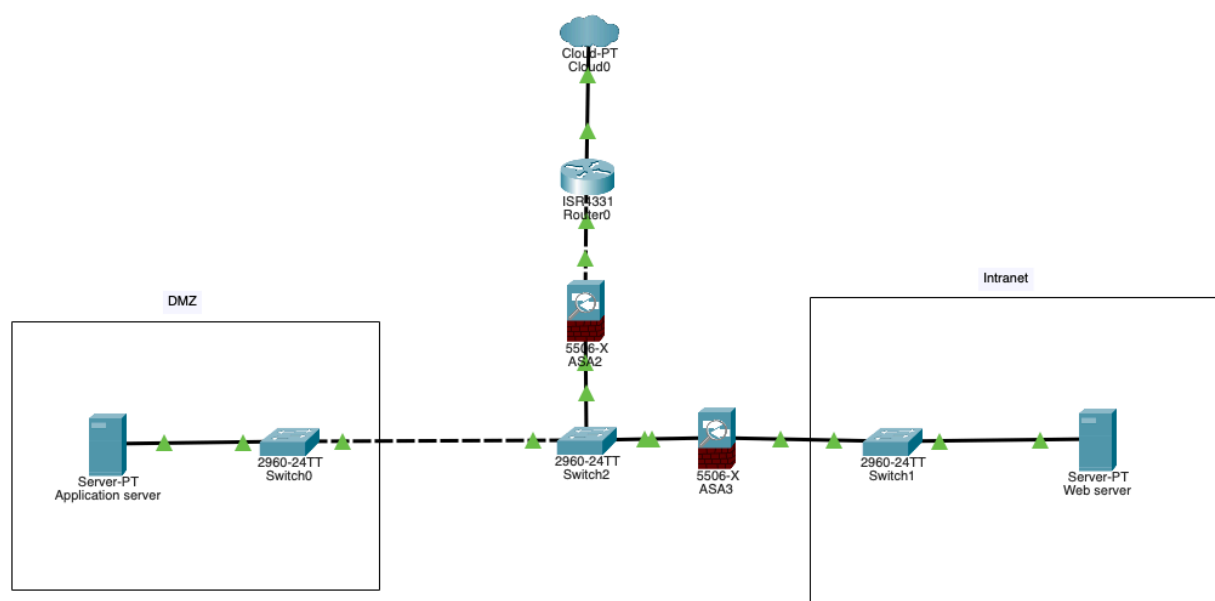
- Un server Web, il quale fornisce diversi servizi su Internet.
- Un server di applicazioni, dedicato a un'applicazione di e-commerce accessibile solo agli impiegati tramite la rete interna.

La rete è stata suddivisa in due zone distinte, classificate in base al flusso di traffico:

- Zona DMZ (Demilitarized Zone): In questa area è ubicato il server di applicazioni. La DMZ è progettata per ospitare servizi accessibili esclusivamente dall'esterno, garantendo l'isolamento della rete privata da potenziali minacce.
- Zona Intranet (LAN): Questa zona ospita i servizi accessibili internamente all'organizzazione.

Gli utenti si trovano...	Accesso alla DMZ	Accesso alla LAN	Accesso a Internet
... su Internet (WAN)	permesso	Non consentito	-
... su intranet(LAN)	permesso	-	permesso
... nella DMZ	-	Non consentito	Permesso

La tabella ci mostra tutte le connessioni possibili nella rete



Design di rete

3. PROGRAMMA PER L'ENUMERAZIONE DEI METODI HTTP

Il programma per l'enumerazione dei metodi è un'applicazione Python concepita per identificare i metodi HTTP abilitati su un sistema target specifico. Utilizzando il modulo `http.client`, il software invia una serie di richieste HTTP per verificare la disponibilità di vari metodi. L'utente fornisce l'indirizzo IP del sistema target, specificando anche la porta e il path di interesse. In caso di omissione della porta, il programma assegna automaticamente la porta predefinita 80.

Il programma stabilisce una connessione HTTP con il sistema target e invia richieste HTTP per ciascun metodo nella lista specificata. Nel caso in cui riceva una risposta con il codice di stato 200 (risposta standard per le richieste andate a buon fine), segnala che il metodo corrispondente è abilitato sul sistema target. Successivamente, la connessione viene chiusa.

Il programma gestisce l'eccezione "ConnectionRefusedError" per notificare all'utente eventuali problemi di connessione al sistema target.

```
(francesco@francesco) - [~/Desktop/Workspace]
• $ /bin/python3.12 /home/francesco/Desktop/Workspace/VerbScanner.py
Inserire l'IP del sistema target: 192.168.50.101
Inserire la porta del sistema target (default '80'): 80
Inserire il path di cui si vuole avere una lista dei metodi abilitati (default '/'): /dwa/login.php
Il metodo OPTIONS è abilitato.
Il metodo PUT è abilitato.
Il metodo HEAD è abilitato.
Il metodo DELETE è abilitato.
Il metodo TRACE è abilitato.
Il metodo GET è abilitato.
Il metodo POST è abilitato.

(francesco@francesco) - [~/Desktop/Workspace]
• $ /bin/python3.12 /home/francesco/Desktop/Workspace/VerbScanner.py
Inserire l'IP del sistema target: 192.168.50.101
Inserire la porta del sistema target (default '80'):
Inserire il path di cui si vuole avere una lista dei metodi abilitati (default '/'): /phpMyAdmin/index.php
E' stata impostata la porta di default.

Il metodo OPTIONS è abilitato.
Il metodo PUT è abilitato.
Il metodo HEAD è abilitato.
Il metodo DELETE è abilitato.
Il metodo TRACE è abilitato.
Il metodo GET è abilitato.
Il metodo POST è abilitato.
```

In figura l'esecuzione del programma appena descritto

4. PROGRAMMA PER IL PORT SCANNING

Questo software esegue una scansione su un range di porte di un determinato indirizzo IP specifico.

Lo scopo del port scanner è identificare potenziali vulnerabilità che possono essere sfruttate attraverso diversi tipi di attacchi, i quali richiedono l'apertura di una specifica porta con un relativo protocollo di comunicazione.

Prima di avviare la scansione, è necessario stabilire una connessione utilizzando un'interfaccia di rete (socket). Questo ci consente di tentare una connessione e, in caso di successo, stabilire una comunicazione per ottenere informazioni sullo stato, il protocollo e il tipo di porta.

```
(francesco@francesco)-[~/Desktop/Workspace]
$ /bin/python3.12 /home/francesco/Desktop/Workspace/PortScanner.py
Inserisci l'indirizzo IP della macchina target: 192.168.50.101
Inserisci la porta di partenza: 0
Inserisci l'ultima porta: 30

Il prospetto delle porte UDP è: {0: 'Chiusa', 1: 'Chiusa', 2: 'Chiusa', 3: 'Chiusa', 4: 'Chiusa', 5: 'Chiusa', 6: 'Chiusa', 7: ('Aperta', 'echo'), 8: 'Chiusa', 9: ('Aperta', 'discard'), 10: 'Chiusa', 11: 'Chiusa', 12: 'Chiusa', 13: ('Aperta', 'daytime'), 14: 'Chiusa', 15: 'Chiusa', 16: 'Chiusa', 17: 'Chiusa', 18: 'Chiusa', 19: ('Aperta', 'chargen'), 20: 'Chiusa', 21: ('Aperta', 'fsp'), 22: 'Chiusa', 23: 'Chiusa', 24: 'Chiusa', 25: 'Chiusa', 26: 'Chiusa', 27: 'Chiusa', 28: 'Chiusa', 29: 'Chiusa', 30: 'Chiusa'}

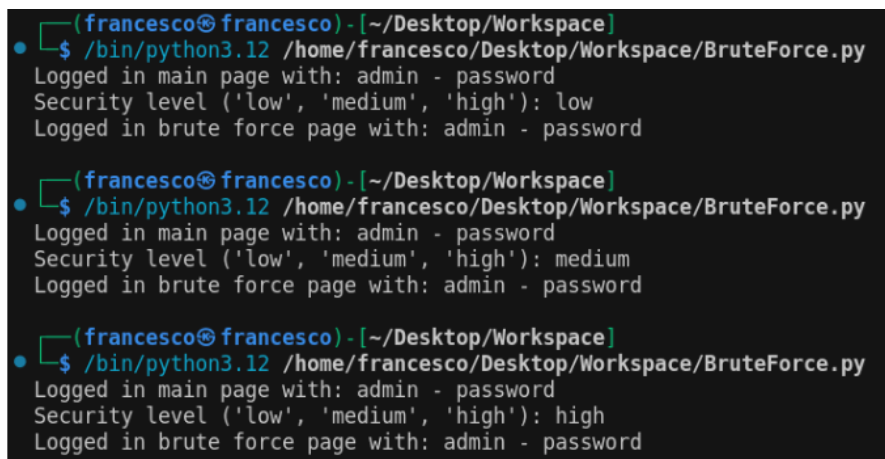
Il prospetto delle porte TCP è: {0: 'Chiusa', 1: 'Chiusa', 2: 'Chiusa', 3: 'Chiusa', 4: 'Chiusa', 5: 'Chiusa', 6: 'Chiusa', 7: 'Chiusa', 8: 'Chiusa', 9: 'Chiusa', 10: 'Chiusa', 11: 'Chiusa', 12: 'Chiusa', 13: 'Chiusa', 14: 'Chiusa', 15: 'Chiusa', 16: 'Chiusa', 17: 'Chiusa', 18: 'Chiusa', 19: 'Chiusa', 20: 'Chiusa', 21: ('Aperta', 'ftp'), 22: 'Chiusa', 23: 'Chiusa', 24: 'Chiusa', 25: 'Chiusa', 26: 'Chiusa', 27: 'Chiusa', 28: 'Chiusa', 29: 'Chiusa', 30: 'Chiusa'}
```

In figura l'esecuzione del programma appena descritto

5. ATTACCO BRUTE FORCE

Abbiamo sviluppato un programma in linguaggio Python per eseguire un attacco di tipo Brute Force contro la piattaforma DVWA, il sistema web vulnerabile del nostro cliente. L'obiettivo è individuare credenziali d'accesso valide. Inizialmente, abbiamo mirato alla pagina di login principale: Il programma esegue iterazioni attraverso una lista predefinita di nomi utente e password al fine di ottenere accesso al sistema bersaglio mediante richieste HTTP POST.

Dopo aver superato questa fase, abbiamo eseguito un secondo attacco alla pagina di login del Brute Force. Una volta ottenuto l'accesso alla pagina di login iniziale, il programma memorizza il valore del cookie "PHPSESSID" che contiene un codice rilasciato dal server al client al momento del login riuscito. Tale codice è necessario per effettuare successivi tentativi di accesso fingendo di essere il client stesso. Poiché la pagina non esegue un redirect su un'altra pagina in caso di login corretto, per verificare l'autenticità delle credenziali utilizzate, è necessario analizzare il codice HTML restituito.



```
(francesco@francesco) - [~/Desktop/Workspace]
• $ /bin/python3.12 /home/francesco/Desktop/Workspace/BruteForce.py
  Logged in main page with: admin - password
  Security level ('low', 'medium', 'high'): low
  Logged in brute force page with: admin - password

(francesco@francesco) - [~/Desktop/Workspace]
• $ /bin/python3.12 /home/francesco/Desktop/Workspace/BruteForce.py
  Logged in main page with: admin - password
  Security level ('low', 'medium', 'high'): medium
  Logged in brute force page with: admin - password

(francesco@francesco) - [~/Desktop/Workspace]
• $ /bin/python3.12 /home/francesco/Desktop/Workspace/BruteForce.py
  Logged in main page with: admin - password
  Security level ('low', 'medium', 'high'): high
  Logged in brute force page with: admin - password
```

Se il contenuto della pagina caricata include la stringa "Welcome to the password protected area admin", allora le credenziali sono state correttamente utilizzate.

In fase di esecuzione dei test, abbiamo osservato che il menù che permette di cambiare la difficoltà nel TAB di DVWA security non funziona, per ovviare a questo problema è stato sufficiente aggiungere il cookie "security", con il relativo livello di difficoltà, alla richiesta inviata al server. Inoltre i controlli ulteriori che sono presenti ai livelli Medium e High sono per la maggior parte contromisure inerenti agli attacchi di tipo SQL injection. L'unica contromisura efficace per gli attacchi di tipo BruteForce è l'introduzione, a livello High, della funzione sleep().

6. CONTROMISURE DA ADOTTARE E CONSIGLI

Dopo un'attenta analisi e valutazione di tutti i punti critici dell'azienda, possiamo consigliare queste contromisure da adottare, per implementare la sicurezza e l'efficienza dell'azienda.

Per quanto riguarda il design della rete, consigliamo l'uso di due firewall: In generale, l'uso di due firewall può aggiungere un livello aggiuntivo di protezione alla rete, migliorando la sua resilienza e riducendo la probabilità di successo di attacchi informatici. Questi due strumenti permettono di isolare la rete e ci danno la possibilità di implementare una difesa a strati. Ultimo, ma non per importanza, è la corretta impostazione delle policy firewall, per filtrare il traffico in base a determinate regole di sicurezza di base.

Dai vari risultati del Port Scanner, abbiamo notato che molte porte risultavano aperte, consigliamo quindi di tenere aperte solo le porte necessarie a far funzionare i servizi necessari.

Visti i risultati dell'attacco di BruteForce, consigliamo ai dipendenti di utilizzare password robuste e complesse, che includano una combinazione di lettere maiuscole e minuscole, numeri e caratteri speciali. Inoltre, è consigliabile cambiare le password ogni 30 o 60 giorni. Vi è anche da implementare l'autenticazione a due fattori 2FA e per ultimo limitare l'accesso privilegiato solo a coloro che ne hanno veramente bisogno, monitorando attentamente l'accesso e le attività per rilevare eventuali comportamenti sospetti.

Per ultimo, una lista di consigli generali utili per la sicurezza della vostra azienda:

- Aggiungere un Honey Pot per rilevare, monitorare e bloccare attività dannose e non autorizzate
- Promuovere la consapevolezza della sicurezza tra i dipendenti attraverso programmi di formazione e simulazioni di phishing.
- Assicurarsi che tutti i sistemi siano aggiornati con le patch di sicurezza più recenti per mitigare le vulnerabilità note.
- Implementare un rigoroso controllo degli accessi per limitare l'accesso solo agli utenti autorizzati e applicare il principio del privilegio minimo.
- Utilizzare la crittografia per proteggere i dati sensibili durante la trasmissione e l'archiviazione.
- Implementare procedure di backup regolari e testare regolarmente i processi di ripristino per garantire la disponibilità e l'integrità dei dati.

7. LINK UTILI

- Design di rete

https://github.com/francescoficetti/CS0124_BW1/blob/main/Progetto/DesignRete.png

- Programma per l'enumerazione dei metodi HTTP

https://github.com/francescoficetti/CS0124_BW1/blob/main/Progetto/VerbScanner.py

- Programma per il Port scanning

https://github.com/francescoficetti/CS0124_BW1/blob/main/Progetto/PortScanner.py

- Programma di attacco BruteForce sulla DVWA

https://github.com/francescoficetti/CS0124_BW1/blob/main/Progetto/BruteForce.py