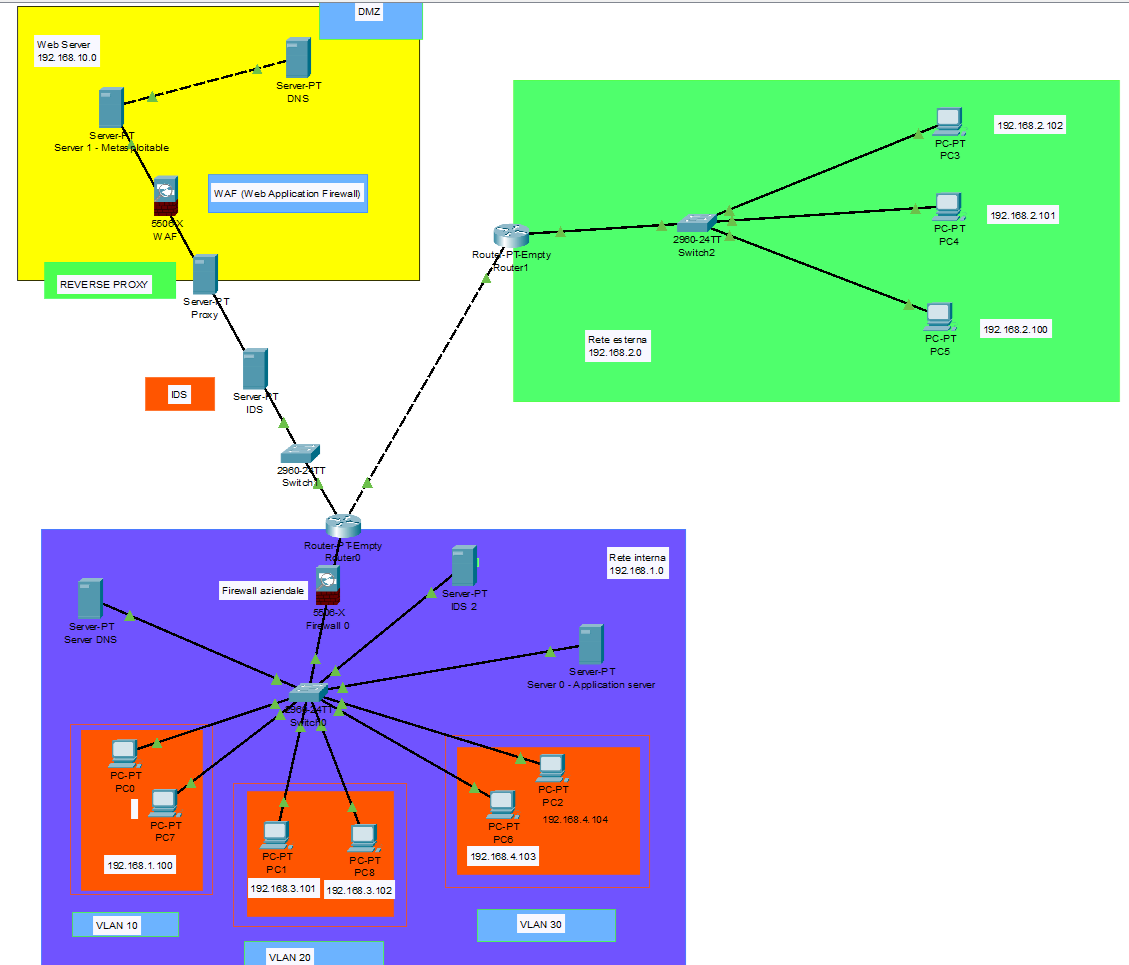
Report finale

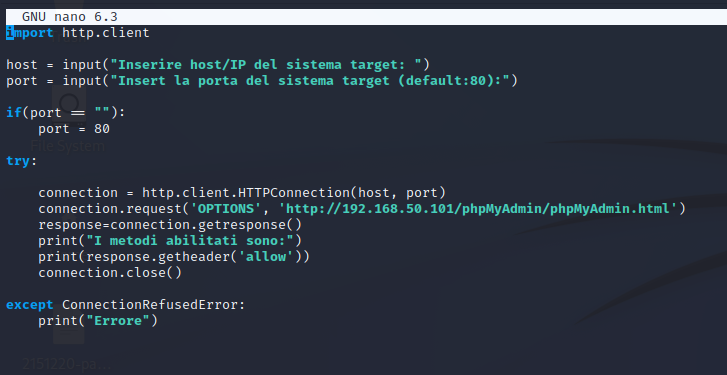
1. Design di rete



Per la sicurezza delle reti dell’azienda Theta abbiamo optato per le seguenti soluzioni:

* **IDS (Intrusion Detection System)**: è un sistema di rilevamento delle intrusioni, serve a individuare in anticipo attacchi verso un computer o una rete. Lo abbiamo inserito all’interno della rete aziendale e all’esterno tra il proxy e lo switch di collegamento alla Wide Area Network (WAN).
* **VLAN (Virtual Local Area Network)**: segmentazione virtuale della rete al fine di aumentare la sicurezza e la velocità.
* **PROXY**: Dispositivo/servizio che si interpone tra la LAN e la WAN, facendo da filtro. In questo caso, abbiamo utilizzato un reverse proxy, che viene attivato come componente di sicurezza aggiuntivo a protezione del web server, per accettare al loro posto le richieste provenienti da Internet, così da rendere i server anonimi.
* **FIREWALL**: in questo caso abbiamo utilizzato un Firewall interno alla rete e un **WAF (Web Application Firewall)**, ovvero un tipo di firewall a filtraggio del Web Server.
* **DMZ (“delimitarized zone”)**: lo scopo è quello di aggiungere un ulteriore livello di sicurezza ad una rete locale aziendale, dove un nodo appartenente ad una rete esterna può accedere soltanto ai servizi di rete messi a disposizione, senza mettere a rischio e compromettere la sicurezza dell’intera rete.

1. Enumerazione dei metodi http



Questo programma serve per elencare i metodi abilitati nel protocollo http:

* Il comando import viene utilizzato per importare il modulo “http.client”.
* Dichiariamo due variabili “host” e “port” che prendono in input corrispettivamente l’indirizzo ip e la porta, che di default sarà 80.
* Alla variabile “connection” assegniamo la realizzazione della connessione all’indirizzo e alla porta specificati.
* Il programma procede inviando una richiesta con il metodo “OPTIONS” all’url indicato.
* Otteniamo la risposta della connessione e stampiamo in output il contenuto all’interno dell’header, nello specifico ciò che è dentro al campo “allow”.
* Infine, chiudiamo la connessione con “connection.close()”.
* In caso di connessione rifiutata (“ConnectionRefusedError”), il programma stamperà la stringa “Errore”.

I metodi del protocollo http che possono essere attivi sono:

* **GET**: richiede un file al server.
* **HEAD**: richiede solo l'header, senza il corpo del messaggio.
* **POST**: invia informazioni all'URI specificato.
* **OPTIONS**: richiede l'elenco dei metodi permessi dal server.
* **TRACE**: traccia una richiesta, visualizzando come viene trattata dal server.
* **DELETE**: cancella una risorsa (file) sul server.
* **PUT**: carica un file sul server, creandolo o riscrivendolo.

1. Port scanner

Immagine che contiene testo

Descrizione generata automaticamente

La scansione delle porte prende in input un indirizzo IP e un range di porte e controlla se queste sono aperte o meno. È anche un processo per inviare pacchetti a porte specifiche su un host e analizzare le risposte per identificare le vulnerabilità.

I passaggi del programma di port scanning sono:

* Importiamo, anche in questo caso, il modulo “socket”.
* Creiamo due variabili (“target” e “portrange”) che prendono in input l’indirizzo IP e un range di porte.
* Inseriamo gli estremi “lowport” e “highport” dell’intervallo di porte, utilizzando il metodo “split” e il simbolo “-” come separatore.
* Il ciclo for successivo ci serve per tentare la connessione TCP ad ogni porta nell’intervallo specificato. Utilizziamo il metodo “socket.socket” che ci restituisce un socket che abbiamo chiamato “s”. Abbiamo scelto come parametri della socket: IPv4 e TCP.
* La funzione successiva “s.connect\_ex” tenta la connessione alla coppia IP:PORTA specificata e ci restituisce uno stato che può essere:
  + “0”: la connessione è andata a buon fine, quindi la porta risulta aperta.
  + “diverso da 0”: la connessione non è andata a buon fine, quindi la porta risulta chiusa.

1. Bruteforce

Immagine che contiene testo

Descrizione generata automaticamente

**4.1 Descrizione del codice di Bruteforce**

Partendo dall’inizio, possiamo definire “import” come un comando di Python utilizzato per richiamare all’interno del codice delle funzioni esterne. In questo caso le funzioni richiamate da Import sono «http.client», «urllib.parse».

“Username\_ file” e “password\_file” sono le due variabili che andranno a contenere al loro interno i due file “.lst” con i nomi utente e le password più utilizzate. La funzione “open” è utilizzata per aprire i file all’interno delle variabile e restituisce ad esse il contenuto dei suddetti files.

“User\_list” e “pwd\_list” sono due variabili che verranno utilizzate per leggere il contenuto dei file .lst inseriti sopra.

«Readlines ()” è un comando di python che ci permette di leggere il contenuto dei file riga per riga, in questo caso dei nomi utente e delle password che andremo a scansionare.

I due cicli for vengono inseriti all’interno di questo codice per scansionare i vari file e le varie password, creando un ciclo nidificato.

Viene richiamata la variabile user\_list e il ciclo for andrà ad associare tutti i nomi utente della lista a tutte le password. Il punto di uscita di questo ciclo avviene nel momento in cui un nome utente della lista corrisponderà ad una password della seconda lista.

Nel “print(user, “-“, pwd)” verranno stampati in output il nome utente e la password corrispondente.

Con la variabile “post\_parameters” assegniamo la coppia username-password, associata alla combinazione realizzata nel for.

Poi andiamo ad inserire l’header, il quale contenuto è sempre lo stesso.

Con la variabile “conn” andiamo a creare la connessione, indicando l’indirizzo e la porta desiderati, per poi inviare una richiesta con il metodo POST all’url esplicitato all’interno della riga di codice.

Successivamente, assegniamo alla variabile “response” il valore ottenuto dal comando “conn.getresponse()”.

Infine, attraverso la funzione “if”, se rispettata la condizione assegnata, verrà stampato in output l’username e la password corrette.

* 1. **. Esito dell’attacco Bruteforce sulla pagina DVWA**

Le password e gli username vengono combinate tra di loro all’interno dell’attacco bruteforce nei confronti della pagina di login: </dvwa/login.php> attraverso la richiesta http “POST”.

Il codice ha dato come esito le seguenti credenziali d’accesso:

Username: **admin**

Password**: password**

1. Misure di sicurezza da adottare dall’azienda Theta per prevenire attacchi informatici

* Non utilizzare le credenziali di default (“admin” e “password”).
* Non riciclare le password già utilizzati in precedenza.
* Non utilizzare password per i social network identiche a quelle usate su siti aziendali.
* Utilizzo di username più elaborati e password più complesse con combinazioni alfanumeriche e caratteri speciali (-\_@#!?) di minimo 10 caratteri, per aumentare il livello di sicurezza all’interno della pagina login.
* Cambiare mensilmente o trimestralmente le password delle reti aziendali e creare una sigla facilmente memorizzabile dai dipendenti. Si raccomanda di memorizzare la password, piuttosto che salvarla all’interno del pc, così da mantenerne l’integrità.
* Aggiungere un’ulteriore misura di sicurezza, come l’autenticazione tramite altro dispositivo mobile che riconosca che l’utente non sia un intruso, come ad esempio l’invio di notifiche push allo smartphone associato: two-factor authentication o multi-factor authentication.