



*Christopher Caruso
Paolo Aglieco
Dario Calderone
Maurizio Pietrangeli
Valentina Arana
Giulio Sorgente
Andrea Molla
Michele Pepe*

Progetto Theta **CIPHER SQUAD**

OGGETTO RICHIESTA

Il cliente Theta ingaggia la scrivente società ChipherSquad S.r.l. per attività volte a migliorare il livello di sicurezza della propria infrastruttura

Il perimetro della richiesta è circoscritto alle seguenti attività:

Proporre un modello di rete che permetta di garantire i livelli di sicurezza previsti dalle normative vigenti (e.g. GDPR, NIS2) e dagli standard comunemente adottati (e.g. ISO27001, NIST, COBIT). Il modello di rete deve includere un web server esposto e un application server accessibile solo in rete interna

1

2 Analisi dei servizi attivi sulla macchina tramite port scanning, con evidenza delle relative porte aperte e/o chiuse

Enumerazione dei metodi HTTP abilitati sul web server e sull'application server in base al context-path

3

4 Analisi robustezza della login ad un eventuale attacco bruteforce

Per non generare impatti sull' ambiente di produzione è stato ricreato un laboratorio di test in house con due appliance sui quali sono stati caricati i backup delle macchine del cliente.

Lo scopo del progetto è la messa in sicurezza di un application server e di un web server posizionati in base ai rispettivi requisiti di sicurezza.

I dettagli relativi a modelli e prezzi sono riportati nel preventivo della proposta commerciale.

Nel progetto proposto non sono previste alcune tecnologie, configurazioni e servizi accessori che escono dallo scope della richiesta.

Tuttavia il progetto è compatibile con la loro implementazione futura.

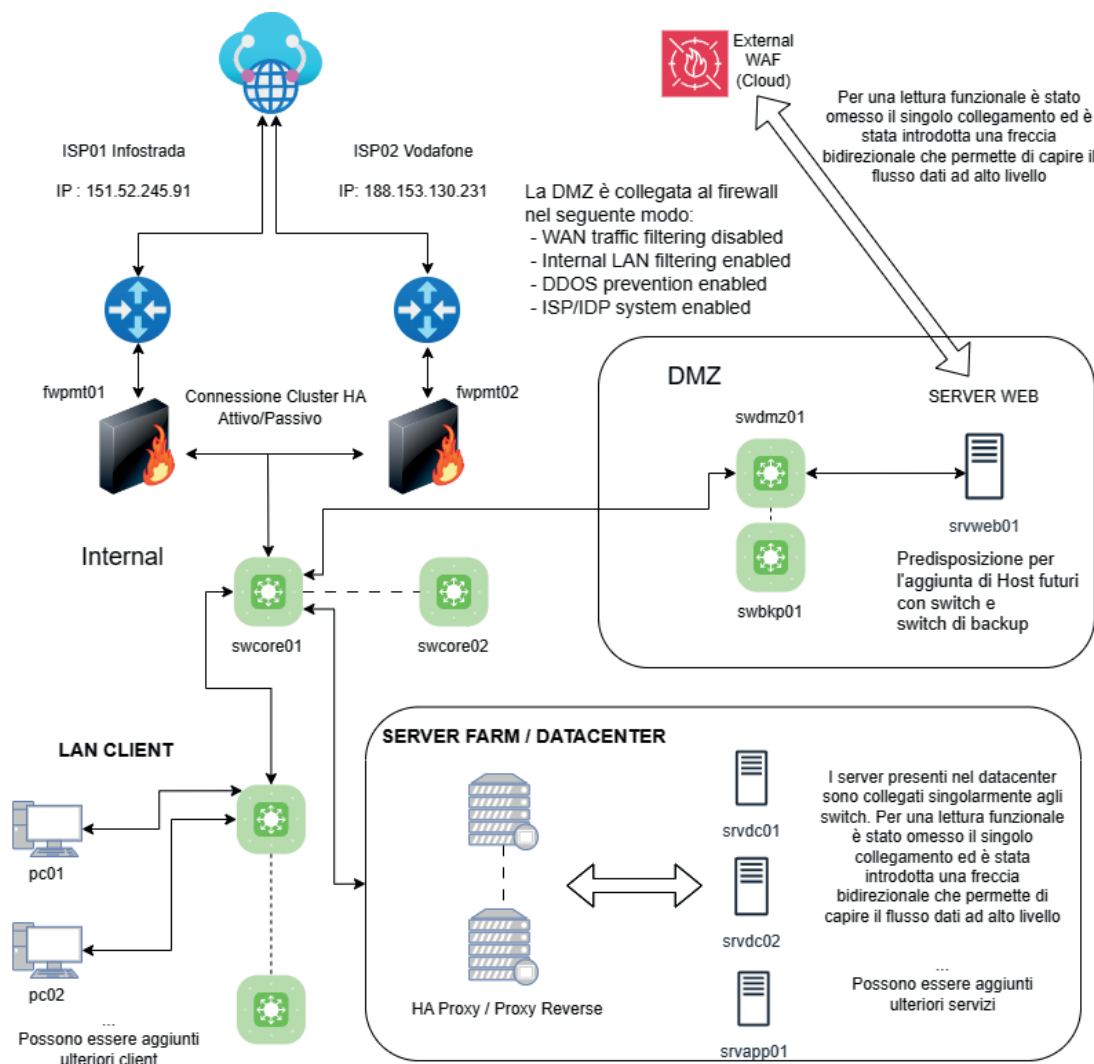
Tecnologie previste in linea con la richiesta:

- Strumenti per la rete (eth cable, rack, network ports)
- Apparati di network security e routing (firewall, switch, proxy)
- Apparati di gestione degli accessi e dei servizi core (domain controllers, WAF in cloud),

Più servizi possono essere posizionati in un sistema cloud, la scelta è a discrezione del cliente e per questo non vengono introdotti nel dettaglio, salvo esplicita futura richiesta del cliente.

Verranno tuttavia forniti suggerimenti al riguardo.

MODELLO DI RETE



Tecnologie extra non previste ma applicabili in futuro:

- Apparati per la rete wifi (Wireless Lan Controllers, Lightweight Access Points)
- Sistema DR degeolocalizzato
- Sistema di posta interno (a discrezione del cliente se integrarlo o usare un servizio di posta esterno)
- Client Desktop (inseriti nello schema a scopo illustrativo)

Gli apparati di rete in essere, rispetto allo schema riportato, supportano la configurazione delle VLAN per la segmentazione rispetto ad una rete Guest. Il sistema DR può essere posizionato anche su un cloud provider aprendo lo scenario di una configurazione Hybrid

Al cliente viene proposta una doppia linea ISP su diversi provider per garantire la connettività in caso di fermo di una delle due linee.

A valle della configurazione delle due linee gestite dal relativo ISP, sono posti due Firewall perimetrali in un cluster Attivo/Passivo che permette di ridurre l'inoperatività dell'azienda in caso di guasto di uno dei due apparati.

La rete è suddivisa in:

- INTERNAL con un' area dedicata alle PDL client e un area dedicata al datacenter
- DMZ dedicata alle macchine esposte verso la WAN

Tutte le reti sono collegate ad uno switch core da 24 porte con connessione fino a 10Gbit con relativo switch di backup

DMZ

E' presente uno switch, con relativo switch di backup, a 24 Porte che supporta la connessione fino a 10Gbit che predispone tale rete per ulteriori macchine espse.

In questa rete è stato inserito il web server che espone i servizi in rete WAN.

Il firewall perimetrale è configurato per gestire il traffico in uscita dalla DMZ solo verso la rete interna per prevenire intrusioni.

Inoltre la configurazione del firewall prevede l'attivazione del modulo IDS/IPS e DDOS prevention.

Viene suggerita l'implementazione di servizi cloud WAF per la protezione del web server (e.g. cloudflare).

INTERNAL

E' presente uno switch, con relativo switch di backup, a 24 Porte che supporta la connessione fino a 10Gbit che predispone tale rete per ulteriori host.

La parte LAN CLIENT è introdotta nello schema per completezza ma le attività a carico della scrivente escludono la fornitura di apparati client.

L'area DATACENTER è perimetrata da due appliance, in un cluster High Availability, che fungono da Proxy per verificare eventuali richieste anomale in ingresso.

Questo tipo di configurazione permette di proteggere la farm dei server da richieste anomale di eventuali client infetti da virus o resi in qualsiasi modo vulnerabili.

Nel DATACENTER è stato posizionato l' application server e sono presenti due domain controller che offrono servizio DNS interno (evitando un possibile Man in the middle a livello DNS), DHCP e autenticazione interna.



PORT SCANNING

Come da richiesta, è stata effettuata una scansione delle porte aperte, in ascolto, sul web server esposto.

Per questo tipo di operazione è stata utilizzata una utility python sviluppata in house dal nostro Red Team.

Sono state rilevate aperte tutte le porte del range UDP (0-65535).

In merito al protocollo TCP sono state rilevate le seguenti porte:

Porta 21 - FTP
Porta 22 - SSH
Porta 23 - TELNET
Porta 25 - SMTP
Porta 53 - DNS
Porta 80 - HTTP
Porta 111 - ident
Porta 139 - NETBIOS -Session Service
Porta 445 - Microsoft-DS
Porta 512 - Act P202S VoIP WiFi phone
Porta 513 - rlogin
Porta 514 - SysLog
Porta 1099 - rmiregistry
Porta 1524 - inglesrock
Porta 2049 - Network File System
Porta 2121 - FTP Proxy Server -
Porta 3306 - MySQL
Porta 3632 - distcc
Porta 5432 - PostgreSQL
Porta 5900 - VNC - Virtual Network Computing / remote Desktop protocol
Porta 6000 - X11 (X Windows Server)
Porta 6667 - IRC Internet Relay Chat
Porta 6697 - IRC SSL (Secure Internet Relay Chat)
Porta 8009 - Netware HTTP Server, Apache JServ Protocol - AJP13
Porta 8180 - Sconosciuto
Porta 8787 - msgsrvr - Scientia-ssdb message Server
Porta 32797 - Sconosciuto
Porta 37531 - Sconosciuto
Porta 52559 - Sconosciuto
Porta 60081 - Sconosciuto

Rimandiamo alle immagini allegate per un dettaglio sull'output ottenuto.

Per le attività mitigative e relativi suggerimenti si rimanda alla relativa sezione del documento.

Porte open TCP

```
(kali@kali)-[~/Desktop/BuildWeek]
$ python portscanning.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci la porta minima (range: 0-65534):
0
Inserisci la porta massima (range: 0-65535) maggiore della precedente:
65535
Scegli il formato output:
1- Porte OPEN
2- Porte CLOSED
3- Tutte le porte
1
Scanning host 192.168.50.101 from port 0 to port 65535

Porta 21 TCP OPEN
Porta 22 TCP OPEN
Porta 23 TCP OPEN
Porta 25 TCP OPEN
Porta 53 TCP OPEN
Porta 80 TCP OPEN
Porta 111 TCP OPEN
Porta 139 TCP OPEN
Porta 445 TCP OPEN
Porta 512 TCP OPEN
Porta 513 TCP OPEN
Porta 514 TCP OPEN
```

```
Porta 514 TCP OPEN
Porta 1099 TCP OPEN
Porta 1524 TCP OPEN
Porta 2049 TCP OPEN
Porta 2121 TCP OPEN
Porta 3306 TCP OPEN
Porta 3632 TCP OPEN
Porta 5432 TCP OPEN
Porta 5900 TCP OPEN
Porta 6000 TCP OPEN
Porta 6667 TCP OPEN
Porta 6697 TCP OPEN
Porta 8009 TCP OPEN
Porta 8180 TCP OPEN
Porta 8787 TCP OPEN
Porta 40285 TCP OPEN
Porta 45437 TCP OPEN
Porta 48874 TCP OPEN
Porta 55726 TCP OPEN
Porta 0 UDP OPEN
Porta 1 UDP OPEN
Porta 2 UDP OPEN
Porta 3 UDP OPEN
Porta 4 UDP OPEN
Porta 5 UDP OPEN
Porta 6 UDP OPEN
Porta 7 UDP OPEN
Porta 8 UDP OPEN
```

Porte open UDP

```
Porta 65509 UDP OPEN
Porta 65510 UDP OPEN
Porta 65511 UDP OPEN
Porta 65512 UDP OPEN
Porta 65513 UDP OPEN
Porta 65514 UDP OPEN
Porta 65515 UDP OPEN
Porta 65516 UDP OPEN
Porta 65517 UDP OPEN
Porta 65518 UDP OPEN
Porta 65519 UDP OPEN
Porta 65520 UDP OPEN
Porta 65521 UDP OPEN
Porta 65522 UDP OPEN
Porta 65523 UDP OPEN
Porta 65524 UDP OPEN
Porta 65525 UDP OPEN
Porta 65526 UDP OPEN
Porta 65527 UDP OPEN
Porta 65528 UDP OPEN
Porta 65529 UDP OPEN
Porta 65530 UDP OPEN
Porta 65531 UDP OPEN
Porta 65532 UDP OPEN
Porta 65533 UDP OPEN
Porta 65534 UDP OPEN
Porta 65535 UDP OPEN
```

Per le attività mitigative e relativi suggerimenti si rimanda alla relativa sezione del documento.

Per questa specifica attività è stato realizzato una utility in python, dal nostro Red Team, che permette l'enumerazione dei metodi HTTP abilitati. L'utility è stata eseguita su tutti i context/path disponibili sul server web e sull' application server.

L'utility richiede in input all'utente l'indirizzo IP target, la porta su cui effettuare il controllo e infine il path/context da prendere in esame con i quali costruisce l'url del target.

Nelle immagini allegate sono messi in evidenza i metodi HTTP abilitati per i seguenti context:

/dvwa/	- Metodi abilitati: GET, POST, HEAD, TRACE
/phpMyAdmin/	- Metodi abilitati: GET, POST, HEAD, PUT, DELETE, TRACE
/twiki/	- Metodi abilitati: GET, HEAD, POST, OPTIONS, TRACE
/mutillidae/	- Metodi abilitati: GET, POST, HEAD, PUT, DELETE, TRACE
/dav/	- Metodi abilitati: OPTIONS, GET, HEAD, POST, DELETE, TRACE, PROPFIND, PROPPATCH, COPY, MOVE, LOCK, UNLOCK
/	- Metodi abilitati: GET, POST, HEAD, PUT, DELETE, TRACE

HTTP Method - /dvwa/,/phpMyAdmin/,/twiki/

```
(kali㉿kali)-[/mnt]
$ python3 checkHttpMethod.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci la porta web target (80/443):
80
Inserisci il path per comporre la Request-URI:
/twiki/
I metodi abilitati sono: GET,HEAD,POST,OPTIONS,TRACE

(kali㉿kali)-[/mnt]
$ python3 checkHttpMethod.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci la porta web target (80/443):
80
Inserisci il path per comporre la Request-URI:
/phpMyAdmin/
Metodi abilitati: GET, POST, HEAD, PUT, DELETE, TRACE

(kali㉿kali)-[/mnt]
$ python3 checkHttpMethod.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci la porta web target (80/443):
80
Inserisci il path per comporre la Request-URI:
/dvwa/
Metodi abilitati: GET, POST, HEAD, TRACE
```

HTTP Method - /mutillidae/,/dav/, /

```
(kali㉿kali)-[/mnt]
$ python3 checkHttpMethod.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci la porta web target (80/443):
80
Inserisci il path per comporre la Request-URI:
/mutillidae/
Metodi abilitati: GET, POST, HEAD, PUT, DELETE, TRACE

(kali㉿kali)-[/mnt]
$ python3 checkHttpMethod.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci la porta web target (80/443):
80
Inserisci il path per comporre la Request-URI:
/dav/
I metodi abilitati sono: OPTIONS,GET,HEAD,POST,DELETE,TRACE,PROPFIND,PROPPATCH,COPY,MOVE,LOCK,UNLOCK

(kali㉿kali)-[/mnt]
$ python3 checkHttpMethod.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci la porta web target (80/443):
80
Inserisci il path per comporre la Request-URI:
/
Metodi abilitati: GET, POST, HEAD, PUT, DELETE, TRACE
```

4 BRUTE FORCE

Per l'attacco bruteforce il nostro Red Team ha prodotto una utility python che effettua l'attacco su un indirizzo url, costruito con ip, context/path forniti. L'utility scorre una lista di utenti e password fornita tramite files e per ogni coppia di credenziali effettua un tentativo di login.

La login avviene su più livelli e vengono sfruttati alcuni attributi tra i quali il PHPSESSID e il forcing del livello di security tramite cookie. Come si evince dalle immagini allegate è stata eseguita l'utility sul context /dvwa/ e /dvwa/vulnerabilities/brute/ rilevando le seguenti credenziali di accesso: username = admin, password = password.

Attacco avvenuto con successo livello LOW

```
(kali@kali)~[~/Desktop/BuildWeek]
$ python bruteforce_OK.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci percorso file usernames:
userlist.txt
Inserisci percorso file passwords:
passwordlist.txt
Inserisci il livello di sicurezza per il cookie di DVWA:
1 - low
2 - medium
3 - high
3
Attacco bruteforce login.php!

admin - admin
Login /login.php KO!

Credenziali non valide!
Attacco bruteforce login.php!

admin - guest
Login /login.php KO!

Credenziali non valide!
Attacco bruteforce login.php!

admin - password
Login /login.php OK!

Session ID: 347f4699ae6c8506e42afd28c96841c0
Attacco bruteforce context /vulnerabilities/brute/!

admin - admin
Login /dvwa/vulnerabilities/brute/ KO!

Attacco bruteforce context /vulnerabilities/brute/!

admin - guest
Login /dvwa/vulnerabilities/brute/ KO!

Attacco bruteforce context /vulnerabilities/brute/!

admin - password
Login /dvwa/vulnerabilities/brute/ OK!
```

```

192.168.50.101
Inserisci percorso file usernames:
userlist.txt
Inserisci percorso file passwords:
passwordlist.txt
Inserisci il livello di sicurezza per il cookie di DVWA:
1 - low
2 - medium
3 - high
2
Attacco bruteforce login.php!

admin - admin
Login /login.php KO!

Credenziali non valide!
Attacco bruteforce login.php!

admin - guest
Login /login.php KO!

Credenziali non valide!
Attacco bruteforce login.php!

admin - password
Login /login.php OK!

Session ID: 5bc5de1b781849fd071d3d293aa0f6f0
Attacco bruteforce context /vulnerabilities/brute/!

admin - admin
Login /dvwa/vulnerabilities/brute/ KO!

Attacco bruteforce context /vulnerabilities/brute/!

admin - guest
Login /dvwa/vulnerabilities/brute/ KO!

Attacco bruteforce context /vulnerabilities/brute/!

admin - password
Login /dvwa/vulnerabilities/brute/ OK!

```

Attacco avvenuto con successo livello MEDIUM

Attacco avvenuto con successo livello HARD

```

(kali@kali)~[~/Desktop/BuildWeek]
$ python bruteforce_OK.py
Inserisci l'indirizzo IP del target:
192.168.50.101
Inserisci percorso file usernames:
userlist.txt
Inserisci percorso file passwords:
passwordlist.txt
Inserisci il livello di sicurezza per il cookie di DVWA:
1 - low
2 - medium
3 - high
1
Attacco bruteforce login.php!

admin - admin
Login /login.php KO!

Credenziali non valide!
Attacco bruteforce login.php!

admin - guest
Login /login.php KO!

Credenziali non valide!
Attacco bruteforce login.php!

admin - password
Login /login.php OK!

Session ID: 50742f7d405e3211093dea334a67a54d
Attacco bruteforce context /vulnerabilities/brute/!

admin - admin
Login /dvwa/vulnerabilities/brute/ KO!

Attacco bruteforce context /vulnerabilities/brute/!

admin - guest
Login /dvwa/vulnerabilities/brute/ KO!

Attacco bruteforce context /vulnerabilities/brute/!

admin - password
Login /dvwa/vulnerabilities/brute/ OK!

```

CRITICAL ALERT

Durante la costruzione dell'attacco sono emerse ulteriori criticità :

- SQL Injection con livello security low per DVWA
- Scrivendo il path /phpMyAdmin/setup/ viene effettuato un bypass della login
- Da console sulla macchina web è stato possibile effettuare la login al client mysql senza inserire la password di root
- Effettuando una query da client mysql sono emersi tutti users privi di password (login effettuata su phpMyAdmin senza password per i seguenti users: guest, de bian-sys-maint)
- L'header restituito nelle interrogazioni web contiene la versione del web server apache, questo consente il tentativo di attacchi mirati sulle vulnerabilità di quella versione

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The 'Vulnerability: Brute Force' section is active. The 'Login' form is visible with fields for 'Username' (containing 'admin' -- AND passwords: '-') and 'Password'. Below the form, the 'More info' section lists several links. A browser window is overlaid on the page, displaying the source code of the login form. The source code is as follows:

```
<?php
if( isset( $_GET['Login'] ) ) {
    $user = $_GET['username'];
    $pass = $_GET['password'];
    $pass = md5($pass);

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

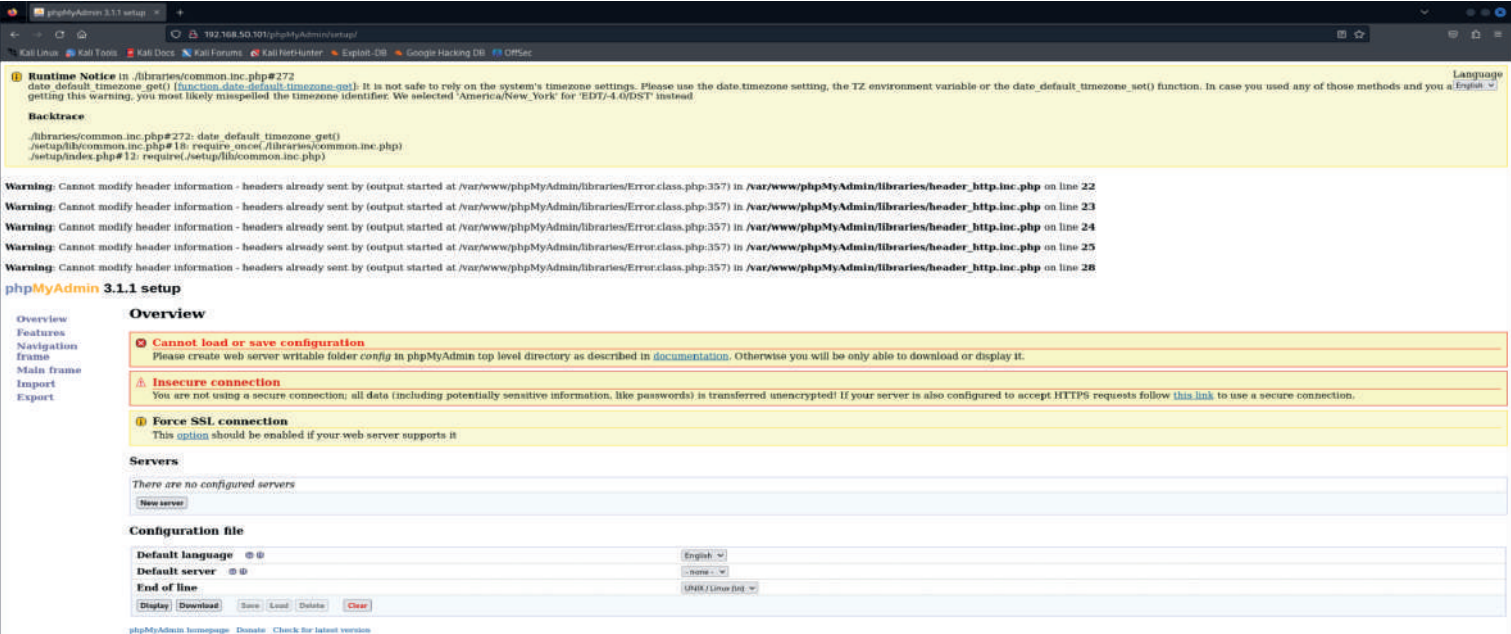
    if( $result && mysql_num_rows( $result ) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // 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>";
    }
}
```

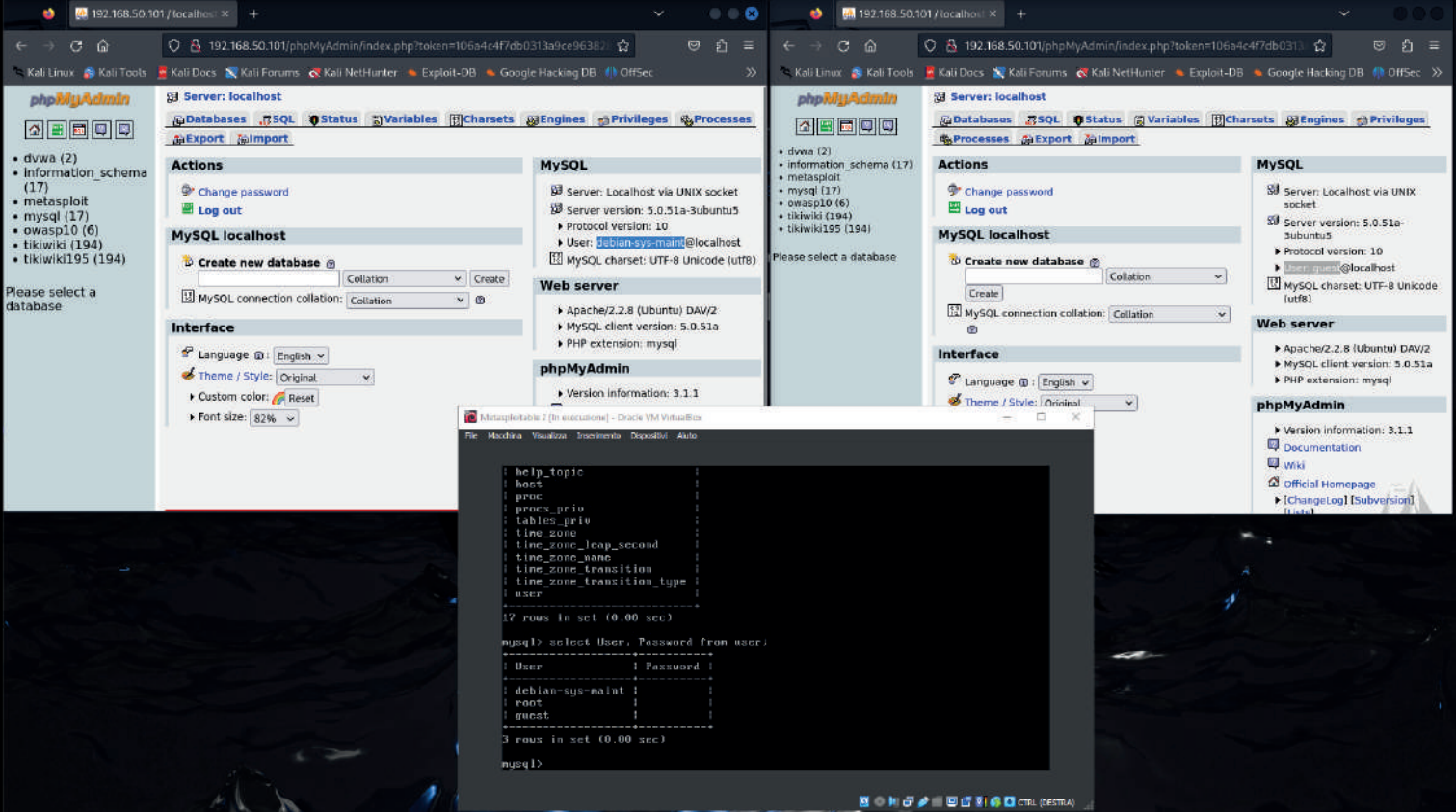
SQL Injection DVWA -Level Low

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The 'Vulnerability: Brute Force' section is active. The 'Login' form is visible with fields for 'Username' and 'Password'. Below the form, the 'More info' section lists several links. The 'View Source' and 'View Help' buttons are visible at the bottom right. The 'Usernames' and 'Security' sections are also visible on the left sidebar.

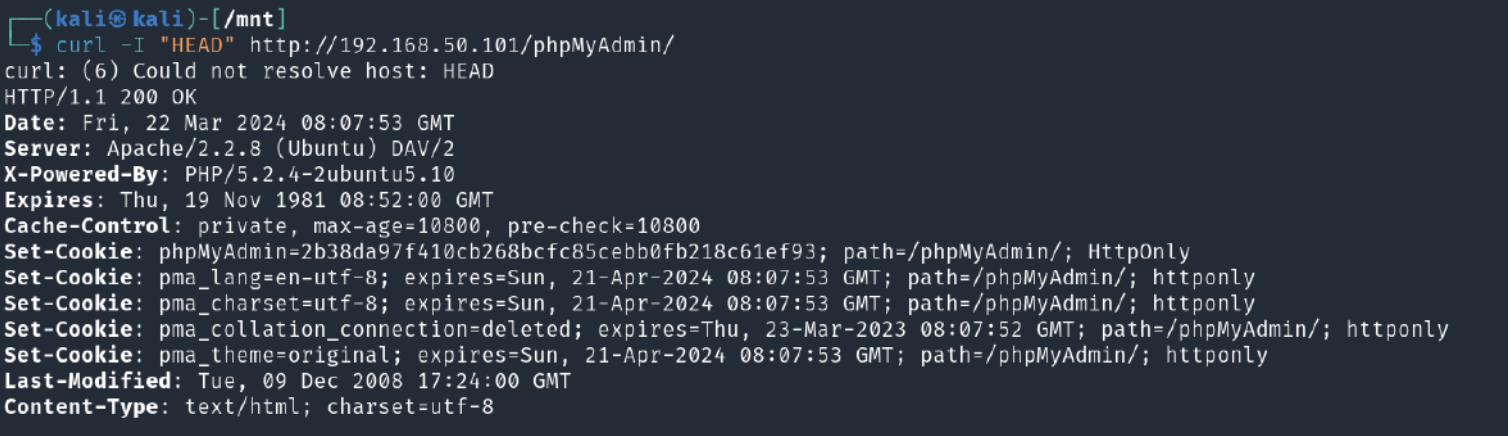
Bypass Login phpMyAdmin



No Password Set for Users mysql



Apache version and php version in headers



ATTIVITA' MITIGATIVE E SUGGERIMENTI

In questa sezione vengono proposte le attività mitigative inerenti alle criticità riscontrate.

Vengono inoltre forniti ulteriori suggerimenti inerenti alle best practice comuni.

Port scanning e servizi attivi

- 1) Chiudere sul web server le porte che non espongono un servizio core relativo al suo funzionamento (e.g. tutte le porte UDP sono aperte ma non espongono servizi)
- 2) Possibilmente effettuare un NAT delle porte rispetto ai servizi di default e preferibilmente abilitare solo canali cifrati (FTPS, SMTPS, etc.) per evitare sniffing del traffico (e.g. risulta aperta la porta 25 SMTP)

Robustezza delle password e sistemi di autenticazione

- 1) Scegliere password con un livello di complessità superiore (e.g. min. 8 caratteri, min. 1 cifra, maiuscola, minuscola e carattere speciale)
- 2) Implementare sistemi di autenticazione MFA (garantiscono la protezione da attacchi bruteforce, da impersonificazione etc.)
- 3) Impostare una scadenza nelle password per tutte le utenze (eventualmente implementare sistemi di Single Sign On che permettono la corretta gestione lato utente)
- 4) Utilizzare credenziali amministrative con username diversi da quelli di default (e.g. non usare utenze come admin, administrator etc. soprattutto su servizi esposti)

Bug nel codice

- 1) Gestire in modo più opportuno le verifiche sulla sessione e sui token per inibire la loro manipolazione tramite script
- 2) Verificare che i reindirizzamenti tra i context/path e l'uso dei cookie di sessione non permettano di "bypassare" la login
- 2) Implementare patch note su sistemi di attacco comuni come SQL Injection

HTTP Methods e Headers

- 1) Abilitare i metodi HTTP necessari al singolo context, prestare attenzione ai metodi PUT, DELETE che permettono di operare sui contenuti
- 2) L'header restituito dalle pagine web dovrebbe mascherare la versione del web server che espone i servizi, altrimenti un eventuale attaccante potrebbe avere evidenza immediata delle vulnerabilità da sfruttare per la versione specifica

Robustezza Infrastruttura

- 1) Implementare una segmentazione a livello di VLAN nel caso in cui si necessiti di implementare una rete Guest per gli ospiti
- 2) Implementare l'uso di un web application firewall - WAF (per servizi web esposti)
- 3) Implementare sistemi di backup con opportuna crittografia
- 4) Implementare la ridondanza sui dispositivi critici per ridurre i tempi di inattività
- 5) Implementare un piano DR (disaster Recovery) in caso di calamità naturale e/o simili (i sistemi cloud aprono a scenari Hybrid che aiutano a ridurre i costi in tal senso)