BUILD WEEK 1

BYTE REBELS

Preparato da:

CANNAVACCIUOLO DAVIDE
DI MAIO PAOLO
FORLENZA SIMONE
RUSSO FEDERICO - LEADER
TIZZI FEDERICO
VAN ZWAM ARJEN

TRACCIA

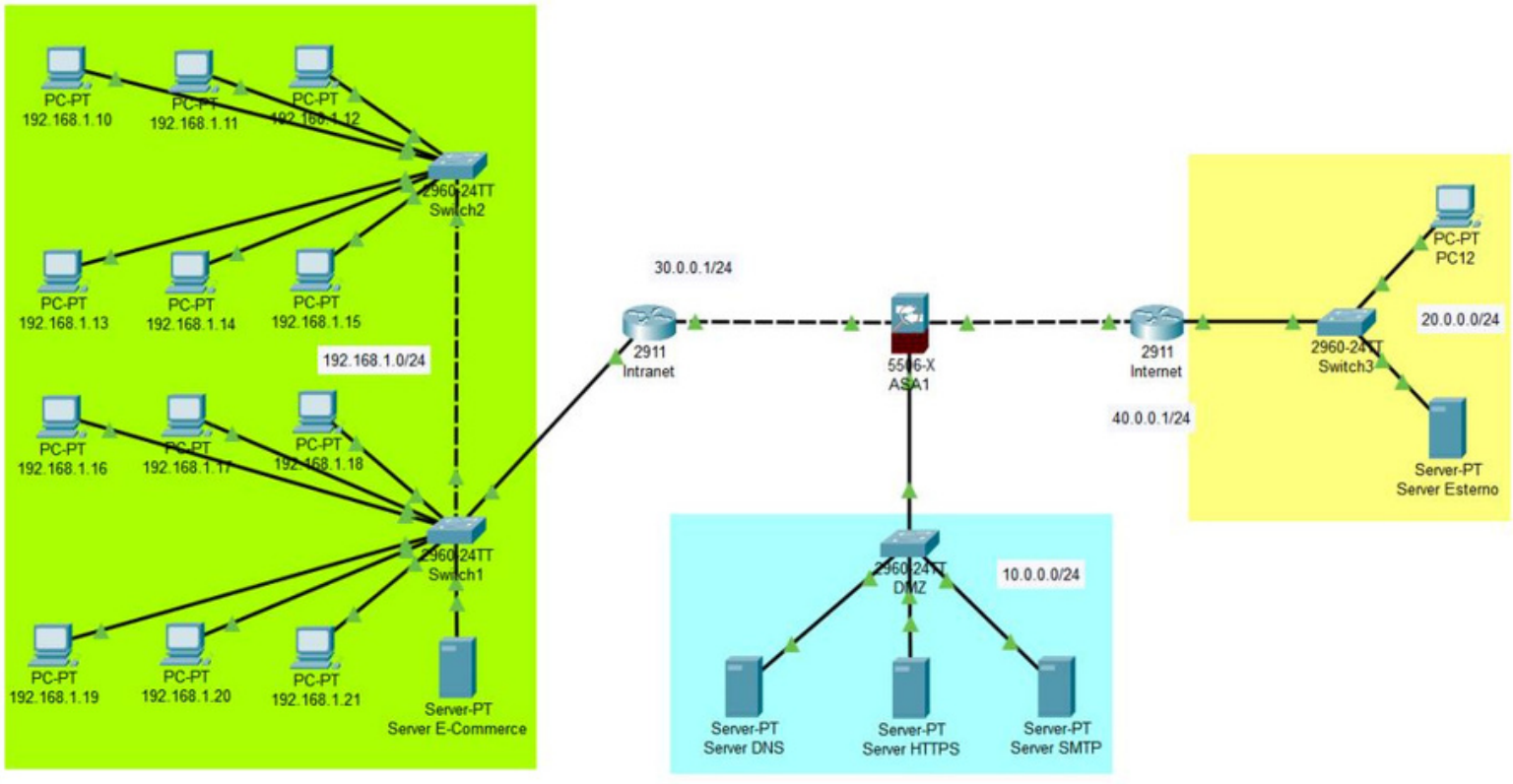
02

Siamo stati ingaggiati dalla compagnia **Theta** per eseguire delle valutazioni di sicurezza su alcune delle infrastrutture critiche dei loro data center.
Il perimetro delle attività si concentra principalmente su:
-Un Web server che espone diversi servizi su internet (e quindi accessibili al pubblico)
-Un Application server che espone sulla rete interna un applicativo di e-commerce accessibile dai soli impiegati della compagnia Theta (quindi non accessibile da resti esterne, ovvero internet) In base alle informazioni sopra, il capo della sicurezza informatica di Theta, chiamato anche CISO (chief information security officer), ci richiede:
1. Di proporre un modello (design) di rete per mettere in sicurezza le due componenti critiche, includendo nell'analisi i dispositivi di sicurezza che potrebbero servire per aumentare la protezione della rete.
2. Di effettuare dei test puntuali sulle due componenti critiche per valutarne lo stato di sicurezza. Nella fattispecie, il CISO ci chiede di effettuare i controlli riportati nella slide successiva.
Sul Web Server:
● Scan dei servizi attivi sulla macchina.
● Eventuale enumerazione dei metodi HTTP abilitati sul servizio HTTP in ascolto sulla porta 80. Sull'application server:
● Enumerazione dei metodi HTTP abilitati.
● Valutazione della robustezza della pagina di login agli attacchi di tipo Brute Force. Il CISO ci ha esplicitamente richiesto di non effettuare nessun test invasivo in ambiente di produzione, e quindi gli abbiamo proposto di riprodurre le due componenti nei nostri laboratori di test, così da poter effettuare i test in sicurezza, separando gli ambienti di test dagli ambienti di lavoro.

03

STRUTTURA THETA

La struttura informatica aziendale Theta è stata ricostruita per semplificarne la comprensione.
Sono presenti zone di **Intranet** (dove non è accessibile dall'esterno, quindi da internet, ma solo dalla rete aziendale)
E' presente un **server** e-commerce interno (anch'esso accessibile solamente da intranet.
In fine, una **DMZ** con un **web server** accessibile da tutti.



Per migliorare drasticamente e subito la sicurezza della rete si ipotizza l'inserimento di un **Firewall perimetrale** posizionandolo tra la rete interna e la rete esterna (ad esempio Internet) e agisce come una **barriera di difesa** che *controlla* e *filtra* il traffico di rete in entrata e in uscita. In sintesi servirà per:
controllare il traffico, filtrare i pacchetti, proxy, VPN e per creare regole di sicurezza personalizzate

WEB SERVER

SERVER CHE ESPONE DIVERSI SERVIZI SU INTERNET

04

SCAN DEI SERVIZI ATTIVI SULLA MACCHINA

Nel Web Server di **Theta** è stata simulata la macchina **Metasploitable2**
Abbiamo dunque simulato un port scanner dei servizi sul server **Theta** con output la lista delle porte aperte e chiuse.

ENUMERAZIONE DEI METODI HTTP ABILITATI

Abbiamo simulato una serie di richieste **HTTP** al server al fine di determinare quali **verbi** HTTP sono supportati per un'analisi più approfondita.

```
Inserire l'opzione desiderata: 4
Inserire l'URL da vreificare: http://192.168.50.101/dvwa/vulnerabilities/brute/
Verbi HTTP supportati per http://192.168.50.101/dvwa/vulnerabilities/brute/: [OPTIONS], [GET], [POST], [PUT], [DELETE]
Inserire l'opzione desiderata:
```

```
[1] English
[2] Italiano
Enter the correct choice (Inserire la scelta corretta): 2

Lingua impostata su Italiano.

ITA
[1] Port scanner
[2] Phpmyadmin
[3] DVWA
[4] Verbi HTTP
[5] Visualizzazione schema packet tracer
[6] Visualizzazione documentazione
[7] Apri il preventivo
[8] Apri report phpmyadmin
[9] Apri report dvwa
[10] Ricarica il menu
[11] Seleziona lingua
[12] Uscita

Inserire l'opzione desiderata: 1
Inserisci l'indirizzo IP: 192.168.50.101
Inserisci l'intervallo di porte (formato es: 1-65535): 1-20
Sto scannerizzando l'host 192.168.50.101 dalle porte 1 alla 20:
Porta 1 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 2 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 3 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 4 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 5 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 6 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 7 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 8 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 9 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 10 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 11 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 12 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 13 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 14 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 15 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 16 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 17 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 18 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 19 - UDP [CHIUSA] - TCP [CHIUSA]
Porta 20 - UDP [CHIUSA] - TCP [CHIUSA]
```

APPLICATION SERVER

SERVER E-COMMERCE SOLO SU INTRANET

05

[redacted]

ENUMERAZIONE DEI METODI HTTP ABILITATI

Abbiamo simulato una serie di richieste **HTTP** al server al fine di determinare quali **verbi** HTTP sono supportati per un'analisi più approfondita.

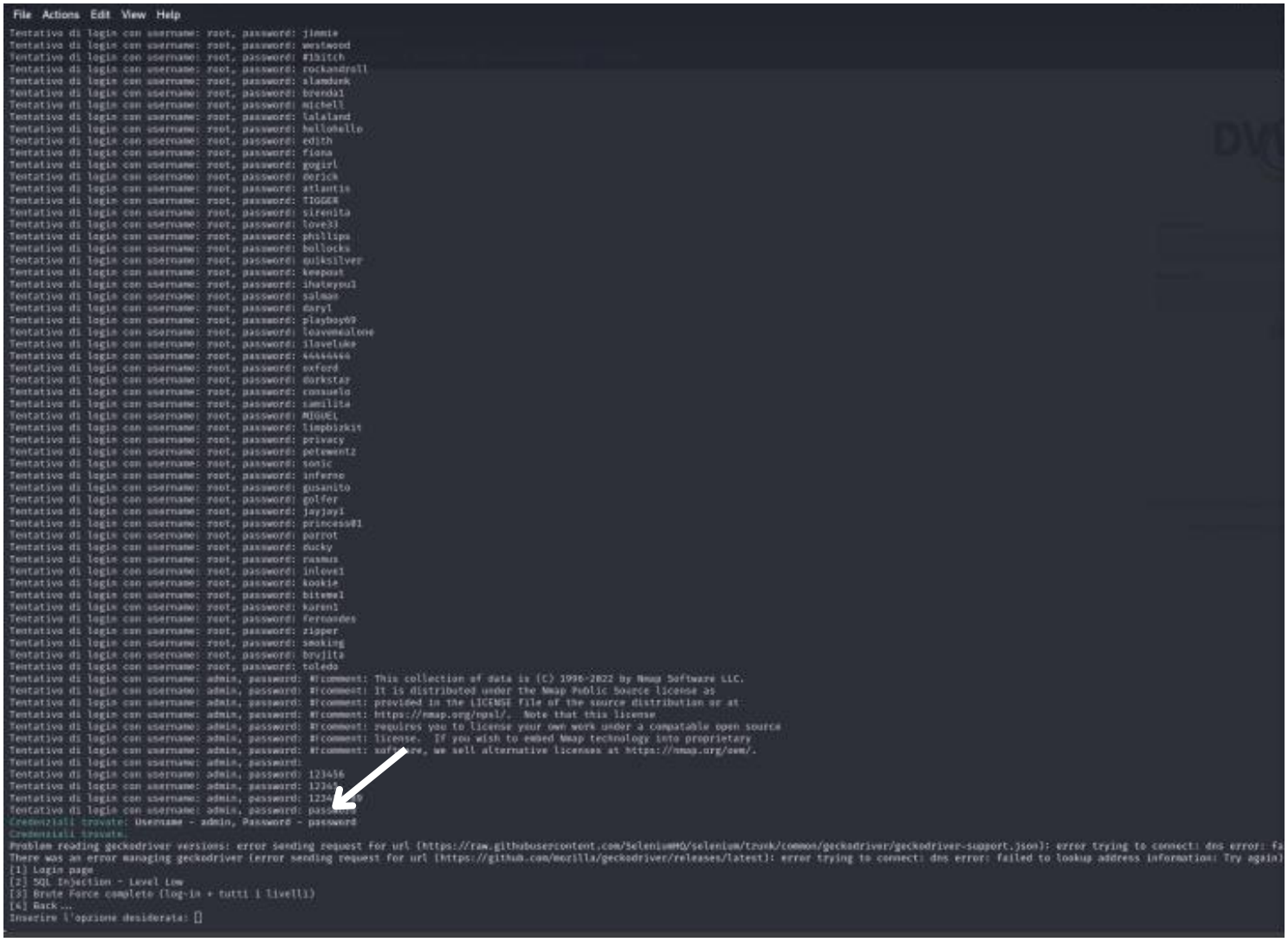
[redacted]

VALUTAZIONE DELLA ROBUSTEZZA DELLA PAGINA DI LOGIN AGLI ATTACCHI DI TIPO BRUTE FORCE

Abbiamo simulato svariati attacchi **Brute Force** per testare l'effettiva sicurezza dei form **login**.

[redacted]

Abbiamo rilevato molteplici fragilità IN TUTTI i form login dell'azienda, successivamente spiegheremo il perché e come risolvere.



APPLICATION SERVER

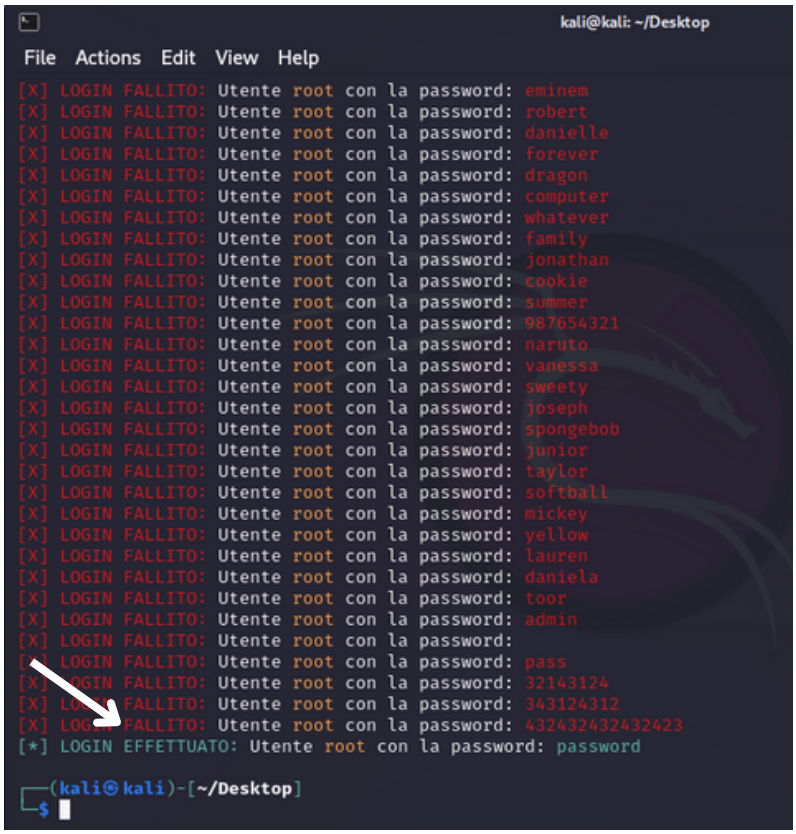
SERVER E-COMMERCE SOLO SU INTRANET

06

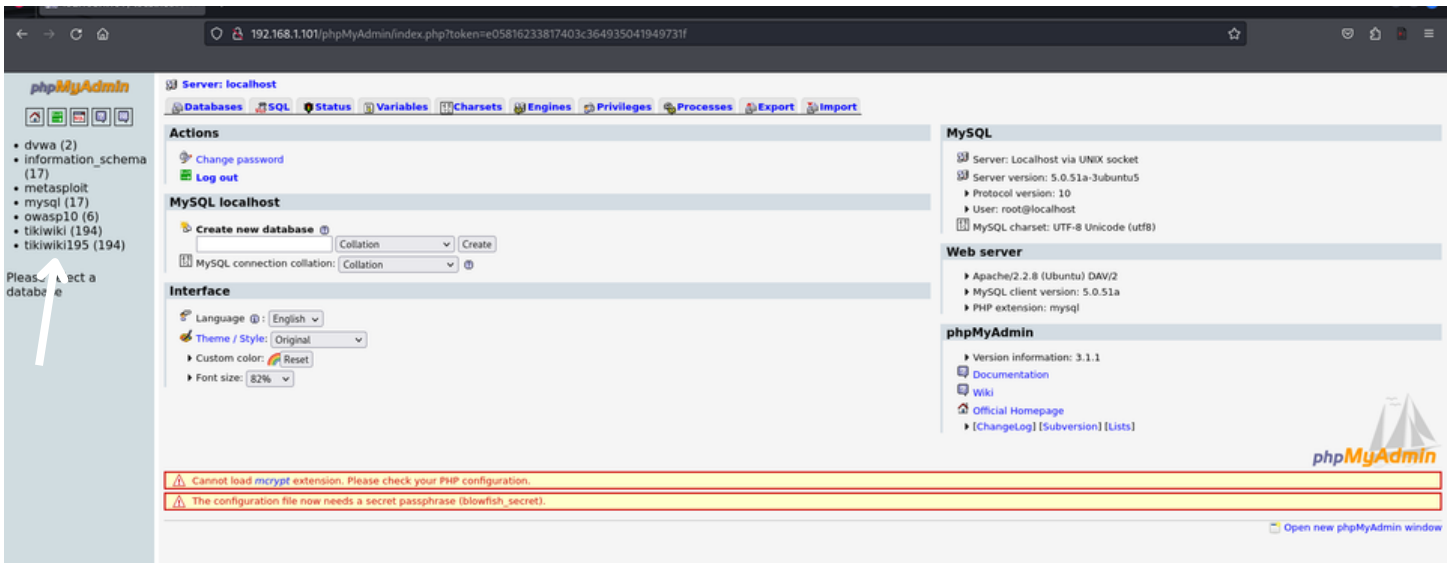
ESITO: MOLTO VULNERABILE

Con i test effettuati in totale **sicurezza** nelle nostre macchine virtuali possiamo affermare che le pagine di login sono totalmente vulnerabili a QUALSIASI **attacco**, lasciamo qui di seguito qualche screen:

Attacco **Brute Force** all'interno dell'index.php nel form **login**



Login effettuato con le relative **credenziali**



APPLICATION SERVER

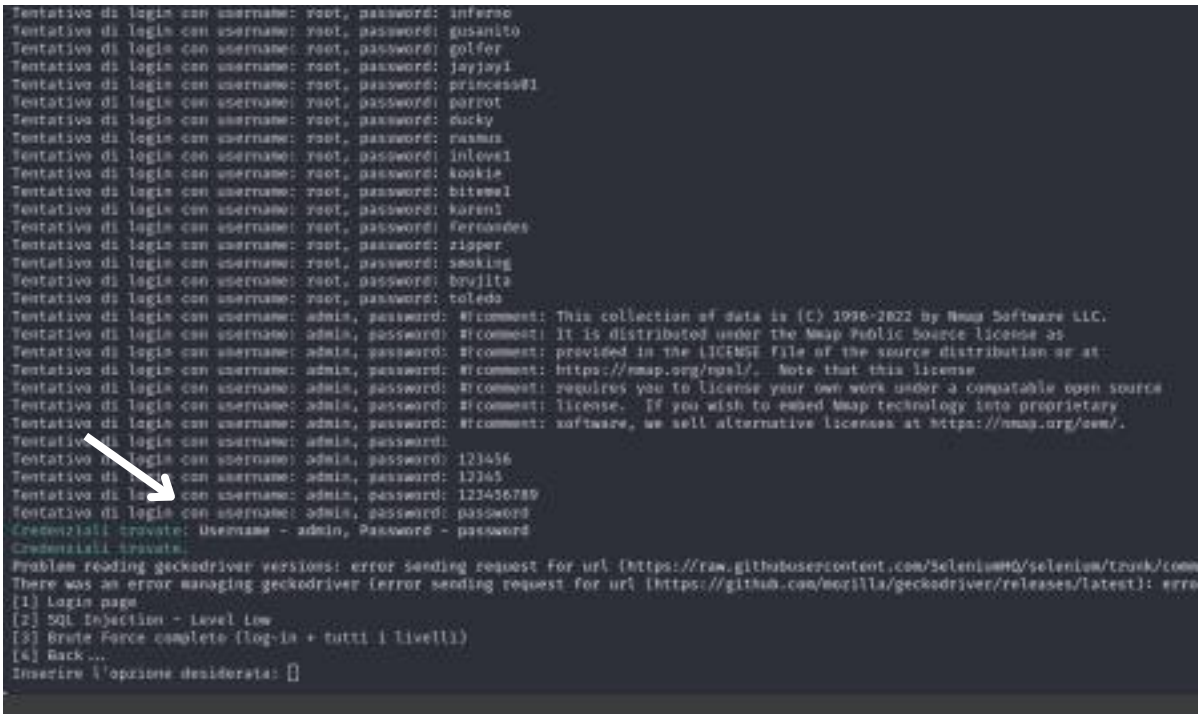
SERVER E-COMMERCE SOLO SU INTRANET

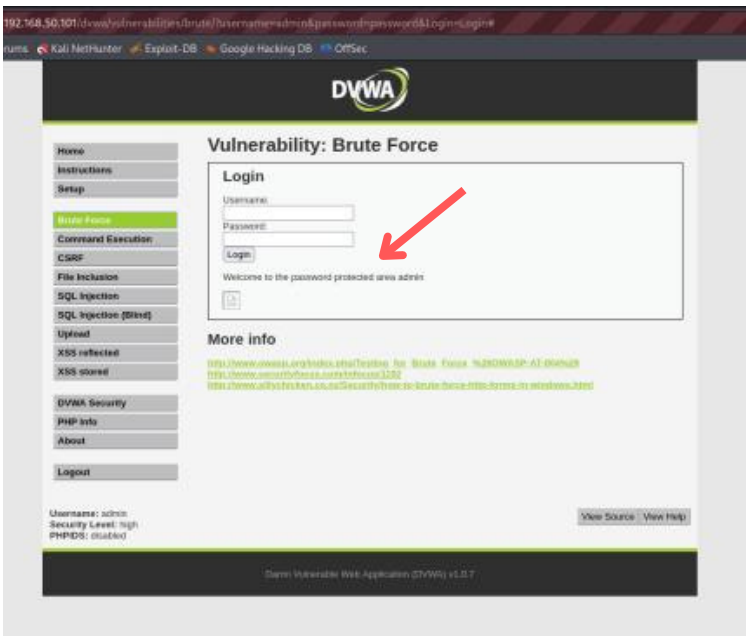
07

ESITO: MOLTO VULNERABILE

Il server che ospita l'e-commerce ha evidenziato falle molto evidenti anche nella pagina **DVWA** login, dove all'interno si può anche eseguire un **sql injection** oltre che utilizzare il Brute Force

Attacco **Brute Force** all'interno della pagina login di **DVWA**





Inoltre, abbiamo anche implementato un login **AUTOMATICO** tramite lo script in python che una volta trovare le credenziali corrette esegua il **login**!

COME MIGLIORARE

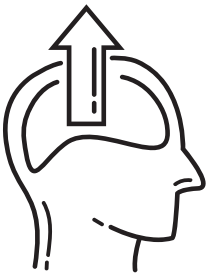
08

I servizi da noi testati in ambienti totalmente sicuri hanno evidenziato molteplici falle.
Di seguito abbiamo stilato una lista sul come migliorare la sicurezza:

Nel livello di sicurezza **BASSO** del DVWA (accessibile con sql injection)
1. **Utilizzo di Parametrizzazione delle Query o ORM**: Utilizzare istruzioni SQL parametrizzate o un Object-Relational Mapping (ORM) per evitare gli attacchi di SQL injection.
2. **Utilizzo di Algoritmi di Hash Sicuri**: Sostituire l'hashing MD5 con algoritmi di hash più sicuri come bcrypt o Argon2 per proteggere le password dagli attacchi di hash rainbow table.
3. **Meccanismo blocco**: implementare un meccanismo di blocco dell'account dopo un certo numero di tentativi falliti.

Nel livello di sicurezza **MEDIO** del DVWA
1. **Utilizzo di Funzioni di Sanitizzazione Moderne**: Utilizzare funzioni di sanitizzazione moderne come `mysqli_real_escape_string` o parametrizzazione delle query per prevenire gli attacchi di SQL injection.
2. **Utilizzo di Algoritmi di Hash Sicuri**: Sostituire l'hashing MD5 con algoritmi di hash più sicuri come bcrypt o Argon2 per proteggere le password dagli attacchi di hash rainbow table.

Nel livello di sicurezza **ALTO** del DVWA
1. **Utilizzo di Funzioni di Sanitizzazione Moderne**: Utilizzare funzioni di sanitizzazione moderne come `mysqli_real_escape_string` o parametrizzazione delle query per prevenire gli attacchi di SQL injection.
2. **Utilizzo di Algoritmi di Hash Sicuri**: Sostituire l'hashing MD5 con algoritmi di hash più sicuri come bcrypt o Argon2 per proteggere le password dagli attacchi di hash rainbow table.
3. **Implementazione di Controlli Anti-Brute Force**: Implementare meccanismi di rilevamento e mitigazione degli attacchi di forza bruta, come limiti di tentativi di accesso, captcha o ritardi crescenti.



COME MIGLIORARE

09

I servizi da noi testati in ambienti totalmente sicuri hanno evidenziato molteplici falle.
Di seguito abbiamo stilato una lista sul come migliorare la sicurezza:

Pagina index.php del phpMyAdmin
Le *soluzioni* possono essere molteplici:

1) **Monitoraggio degli accessi:** Implementazione di controlli sulle richieste dallo stesso IP e di conseguenza un blocco temporaneo/ban di quel determinato IP
2) **Limitazioni di tentativi:** Se una pagina di login non applica alcuna limitazione ai tentativi di accesso, un attaccante può eseguire un numero illimitato di tentativi senza restrizioni. Questo consente loro di eseguire attacchi di brute force senza impedimenti.
3) **Password deboli:** Se gli utenti utilizzano password deboli o facilmente indovinabili, diventa più facile per un attaccante individuare le combinazioni corrette tramite un attacco di brute force.
4) **Meccanismi di rilevamento degli attacchi**: Le pagine di login devono avere un rilevamento degli attacchi di brute force, non avendoli potrebbero consentire ad un attaccante di eseguire l'attacco senza essere rilevato o bloccato dal sistema.
5) **Utilizzare misure di autenticazione a più fattori (MFA)**: L'utilizzo di MFA, come l'invio di un codice di verifica via SMS o l'utilizzo di un'applicazione di autenticazione, può rendere più difficile per un attaccante ottenere l'accesso anche se riesce a indovinare la password.

Ricordiamo che il preventivo che abbiamo già inviato ha durata 30 giorni