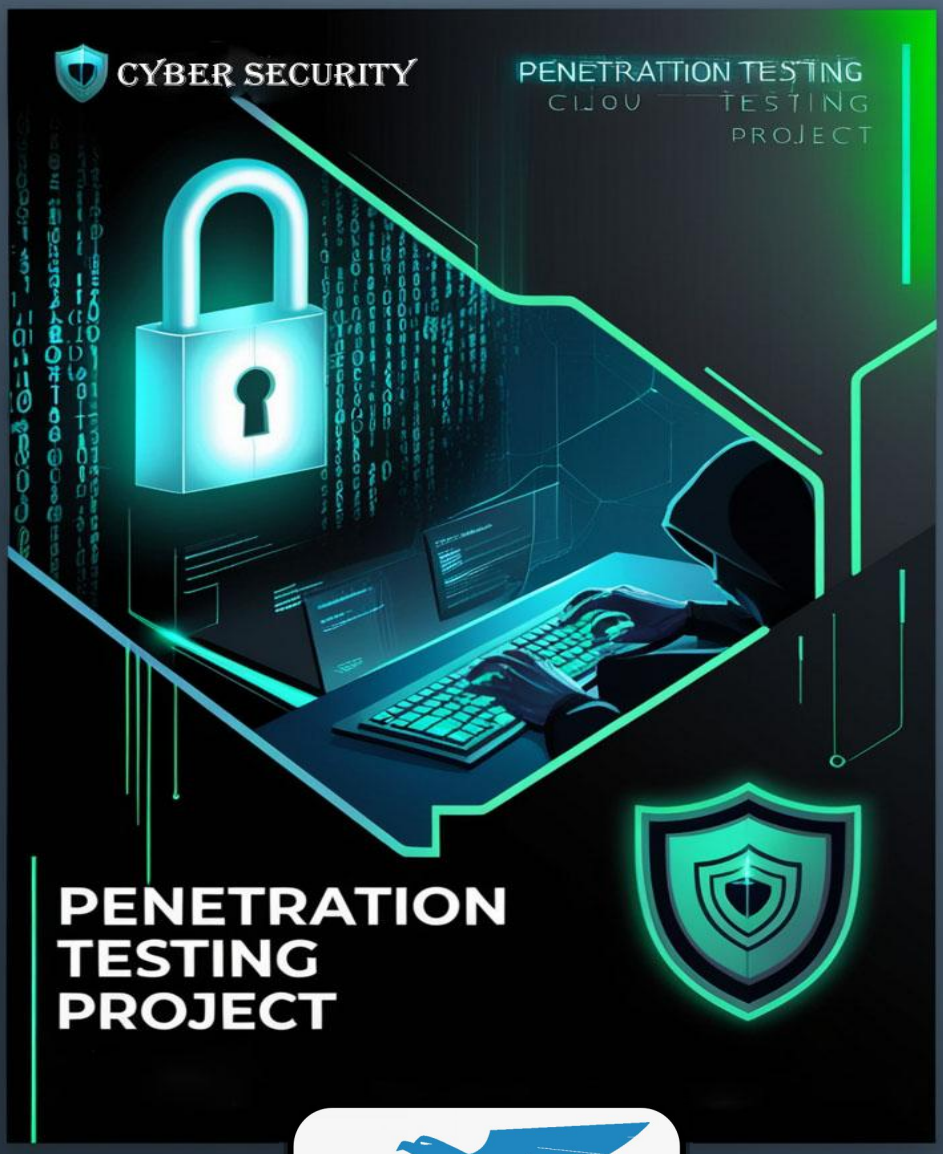
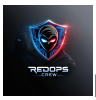


Program: BS-CYS-5

Course: Penetration Testing

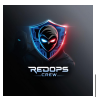
Instructor: Sir Muhammad Azam



RedOps Crew

Comprehensive Penetration Testing on OWASP Juice Shop

Table of Contents



RedOps Crew

Executive Summary:

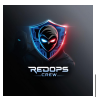
The OWASP Juice Shop project involves accessing a vulnerable hosted web application hosted at https://juice-shop.herokuapp.com/#/. The Project is aimed to gather information about the target, assessing the information and finding vulnerabilities in the target by practicing penetration testing techniques.

Introduction to Penetration Testing:

Penetration testing, often referred to as "pen testing," is a methodical and proactive security assessment process used to evaluate the robustness of an application, system, or network against potential Cyber threats. It involves simulating real-world attacks to identify vulnerabilities that could be exploited by malicious actors. Penetration testing goes beyond traditional vulnerability assessments by actively attempting to exploit identified weaknesses, thereby providing a comprehensive understanding of security risks and their potential impacts.

The primary goal of penetration testing is to improve the security posture of the target system by uncovering gaps in defenses, assessing their severity, and recommending practical solutions. It is an essential practice in modern Cybersecurity, ensuring compliance with security standards and protecting sensitive data from unauthorized access.

For this project, penetration testing will be conducted on the **OWASP Juice Shop**, a deliberately vulnerable web application widely used for security testing and training. This will involve performing all phases of penetration testing, from reconnaissance to post-exploitation, to produce a detailed, professional report documenting findings and mitigation strategies.



RedOps Crew

Scope of Work:

The project will focus on the following key phases of penetration testing:

Reconnaissance:

Reconnaissance is the first step of cyber kill chain---Reconnaissance is the first step of cyber kill chain. It is the practice of discovering and collecting information about a system. Gathering information about our target is crucial in attacking the target .It is the practice of discovering and collecting information about a system. Gathering information about our target is crucial in attacking the target system, network, or organization. The goal is to gather intelligence that will help the attackers understand the targets infrastructure, vulnerabilities and overall security posture, enabling them to plan an effective attack.

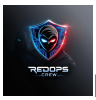
Target:

Our Target is going to be vulnerable web application named Owasp Juice Shop. We will be performing reconnaissance on our target website to figure out information about it for better plans of attack.

Tools:

The tools we will be using for reconnaissance are as follows:

- Whois
- Spiderfoot
- Nslookup
- subbrute



RedOps Crew

Information Gathering:

Information gathered using above tools are as follows:

Registrar information:

Registrar:

MarkMonitor Inc.

Registrar URL:

www.markmonitor.com

Register Abuse contact:

Email:abusecomplaints@markmonitor.com

Phone: +12086851750

Domain Details:

Domain Name:

Herokuapp.com

Registry Domain ID:

1616440759_DOMAIN_COM-VRSN

Creation date:

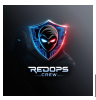
2010-09-19T05:55:31Z

Updated Date:

2023-08-18T09:07:43Z

Expiration Date

2025-09-19T05:55:31Z



RedOps Crew

Registrant Information:

Registrant organization:

Salesforce.com, Inc.

Address:

Salesforce Tower, 415 Mission street, 3rd Floor, San Francisco, CA 94105, Us

Contact Email:

registrar-updates@salesforce.com

Phone:

+18006676389

Name servers:

Primary name servers:

- dns1.p03.nsone.net
- dns2.p03.nsone.net
- dns3.p03.nsone.net
- dns4.p03.nsone.net

Heroku Name servers:

- ns01.herokudns.net
- ns02.herokudns.net
- ns03.herokudns.net
- ns04.herokudns.net

RedOps Crew

DNSSEC:

Status:

Signed Delegation (ensures authenticity and integrity of DNS data).

DNSSEC DS DATA:

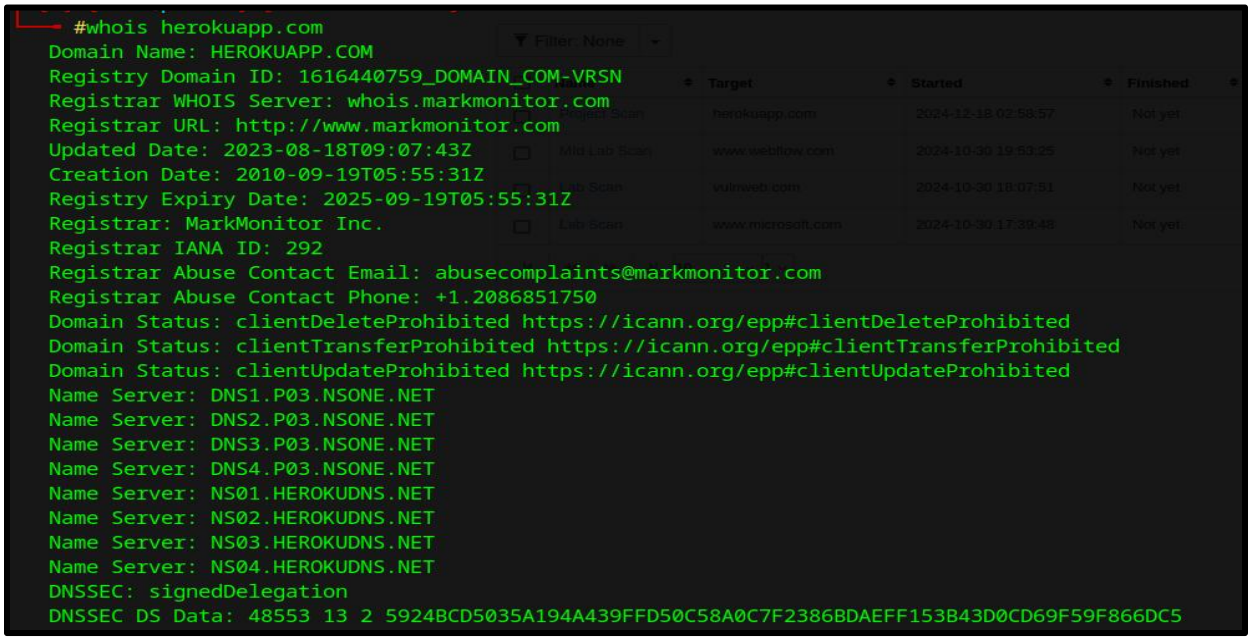
Key Tag: 48553

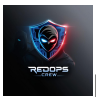
Algorithm: 13

Digest Type: 2

Digest: 5924BCD5035A194A439FFD50C58A0C7F2386BDAEFF153B43D0CD69F59F866DC5

Screenshots:





RedOps Crew

```
Registry Admin ID:
Admin Name: Domain Adminstrator
Admin Organization: Salesforce.com, Inc.
Admin Street: Salesforce Tower, 415 Mission Street, 3rd Floor
Admin City: San Francisco
Admin State/Province: CA
Admin Postal Code: 94105
Admin Country: US
Admin Phone: +1.8006676389
Admin Phone Ext:
Admin Fax: +1.4159017040
Admin Fax Ext:
Admin Email: registrar-updates@salesforce.com
Registry Tech ID:
Tech Name: Domain Adminstrator
Tech Organization: Salesforce.com, Inc.
Tech Street: Salesforce Tower, 415 Mission Street, 3rd Floor
Tech City: San Francisco
Tech State/Province: CA
Tech Postal Code: 94105
Tech Country: US
Tech Phone: +1.8006676389
Tech Phone Ext:
Tech Fax: +1.4159017040
Tech Fax Ext:
Tech Email: registrar-updates@salesforce.com
Name Server: ns02.herokudns.net
Name Server: dns1.p03.nsone.net
Name Server: ns04.herokudns.net
Name Server: ns03.herokudns.net
Name Server: ns01.herokudns.net
Name Server: dns4.p03.nsone.net
Name Server: dns2.p03.nsone.net
Name Server: dns3.p03.nsone.net
DNSSEC: signedDelegation
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
>>> Last update of WHOIS database: 2024-12-17T16:39:46+0000 <<<
```

RedOps Crew

Key Findings from Spiderfoot Scan:

1. Open TCP Ports:

- We have found 7 Open TCP ports that can expose services running on the the server, potentially leading to exploitation if those services have known vulnerabilities and are misconfigured.
- We can use tools like Nmap to identify specific services and there versions running on these ports .Focusing on common vulnerabilities associated with these services we can exploit them.

---

2. DNS Records (SPF, TXT, NS, MX):

- **SPF Records:**

1. SPF stands for Sender Policy Framework. SPF specifies which mail servers are authorized to send email on behalf of your domain.
2. Weak or misconfigured SPF records can make the domain vulnerable to email spoofing or phishing attacks

- **TXT Records:**

1. TXT record stands for text. It stores arbitrary text data used for a variety of purposes, including SPF, DKIM, and other configurations
2. TXT Records can expose sensitive configuration details.

- **Name Servers:**

1. These are the DNS servers responsible for resolving your domain name to IP addresses.
2. It identifies the DNS infrastructure, which could be a target for DNS poisoning or hijacking.

- **MX Records:**

1. MX records stands for mail exchange server. This specifies the mail servers responsible for receiving email for the domain.
2. Indicates the email infrastructure, which could be targeted for phishing campaigns or email service attacks

---

3. SSL Certificate Details

- Host Mismatch (1 Found):
    - Why Critical: Indicates potential SSL misconfiguration, which attackers can exploit to perform Man-in-the-Middle (MitM) attacks.
- Issued By/Issued To (7 Found):
    - Why Critical: Reveals details about the certificate and its chain of trust. Expired or improperly issued certificates weaken the security posture.
- Action: Validate the SSL certificate using tools like SSL Labs to check for weaknesses.

---

4. HTTP Headers (48 Found) & Non-Standard Headers (93 Found)

- Why Critical: HTTP headers can reveal sensitive information about the server, framework, or configuration, including:
    - Outdated software versions.
    - Missing security headers (e.g., X-Frame-Options, Content-Security-Policy).
- Action: Review the headers for sensitive details or lack of standard security practices.

---

5. Co-Hosted Sites (102 Found)

- Why Critical: Sharing a host with other vulnerable websites can expose the target to cross-contamination attacks or shared hosting vulnerabilities.
- Action: Investigate co-hosted domains for misconfigurations or signs of compromise.

---

6. Affiliate and Related Data

- Affiliate - Email Address (10 Found):
  - Why Critical: Could be used in social engineering or phishing attacks.
- Affiliate - Domain Name (7 Found):
  - Why Critical: Related domains might share vulnerabilities or infrastructure.
- Affiliate - IP Address (61 Found):
  - Why Critical: Provides a broader view of the server's environment, potentially exposing shared weaknesses.
- Action: Investigate affiliates for signs of compromise or misconfiguration.

---

7. Interesting Files (27 Found)

- Why Critical: Files such as .git, .env, or backup files can contain sensitive data, including credentials or application secrets.
- Action: Access these files (if legally permissible) to determine if they expose sensitive information.

---

8. Raw Data from APIs/RIRs (18 Found)

- Why Critical: Could reveal sensitive IP allocation or routing information that aids in network-based attacks.
- Action: Correlate with open ports and hosting details to identify misconfigurations.

---

9. Linked URLs (External/Internal)

- External URLs (9 Found):
    - Why Critical: Could redirect users to malicious third-party websites or expose integrations with vulnerable services.
- Internal URLs (84 Found):
    - Why Critical: May reveal sensitive internal paths or endpoints not intended for public access.
- Action: Test the internal URLs for API access or endpoints exposing sensitive data.

---

10. Cloud Storage Buckets (3 Found)

- Why Critical: Misconfigured buckets can expose sensitive data or allow unauthorized access.
- Action: Check for public access permissions or exposed data within these buckets.

RedOps Crew

Screenshots:

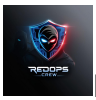
Open TCP Ports

Browse / Open TCP Port

Data Element	Source Data Element	Source Module	Identified
46.137.15.86:443	46.137.15.86	sfp_sslcert	2024-12-21 18:21:48
54.220.192.176:443	54.220.192.176	sfp_sslcert	2024-12-21 18:22:43
54.73.53.134:443	54.73.53.134	sfp_sslcert	2024-12-21 18:22:29
ec2-46-137-15-86.eu-west-1.compute.amazonaws.com:443	ec2-46-137-15-86.eu-west-1.compute.amazonaws.com	sfp_sslcert	2024-12-21 18:35:57
ec2-54-220-192-176.eu-west-1.compute.amazonaws.com:443	ec2-54-220-192-176.eu-west-1.compute.amazonaws.com	sfp_sslcert	2024-12-21 18:36:13
ec2-54-73-53-134.eu-west-1.compute.amazonaws.com:443	ec2-54-73-53-134.eu-west-1.compute.amazonaws.com	sfp_sslcert	2024-12-21 18:36:07
juice-shop.herokuapp.com:443	juice-shop.herokuapp.com	sfp_sslcert	2024-12-21 17:59:11

DNS SPF Records

Data Element	Source Data Element	Source Module	Identified
spf2.0/pra include:amazon.com ~all	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
v=spf1 include:amazon.com ~all	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45



RedOps Crew

DNS TXT Records

☐	atlassian-domain-verification=ZT4AapXgobCpXIWoNcd7gtMjZyOUdr4EDFMnFUWrqqqgdaQVbDvoGpRaIwj/tgPH	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	google-site-verification=EEVHeL7fVZb5ix5bR0draHWtJ5MfS0538OwXAfY8HCY	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	pf2vv39dfkf9tszsg5lggfs6tp6bkjn4	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	spf2.0/pra include:amazon.com ~all	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	v=spf1 include:amazon.com ~all	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45

Name Server

Browse / Name Server (DNS NS Records)

☐	Data Element	Source Data Element	Source Module	Identified
☐	dns1.p03.nsone.net	herokuapp.com	sfp_dnsraw	2024-12-21 17:59:46
☐	dns2.p03.nsone.net	herokuapp.com	sfp_dnsraw	2024-12-21 17:59:46
☐	dns3.p03.nsone.net	herokuapp.com	sfp_dnsraw	2024-12-21 17:59:46
☐	dns4.p03.nsone.net	herokuapp.com	sfp_dnsraw	2024-12-21 17:59:46
☐	ns-1321.awsdns-37.org	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	ns-1670.awsdns-16.co.uk	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	ns-27.awsdns-03.com	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	ns-967.awsdns-56.net	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
☐	ns01.herokudns.net	herokuapp.com	sfp_dnsraw	2024-12-21 17:59:46

RedOps Crew

DNS MX Records

Browse / Email Gateway (DNS MX Records)

Data Element	Source Data Element	Source Module	Identified
amazon-smtp.amazon.com	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45

SSL Certificate Details

SSL Certificate - Issued by	1	7	2024-12-21 18:36:13
SSL Certificate - Issued to	1	7	2024-12-21 18:36:13
SSL Certificate - Raw Data	1	7	2024-12-21 18:36:13
SSL Certificate Host Mismatch	1	3	2024-12-21 18:36:13

HTTP Headers & Non-standard Headers

Browse / HTTP Headers

Data Element	Source Data Element	Source Module	Identified
{"content-length": "563", "cache-control": "no-cache, no-store", "content-type": "text/html; charset=utf-8", "date": "2024-12-21 08:35:41.048876649 +0000 UTC", "server": "heroku-router"}	ec2-46-137-15-86.eu-west-1.compute.amazonaws.com	sfp_spider	2024-12-21 18:35:52
{"content-length": "563", "cache-control": "no-cache, no-store", "content-type": "text/html; charset=utf-8", "date": "2024-12-21 08:35:51.72727719 +0000 UTC", "server": "heroku-router"}	ec2-54-73-53-134.eu-west-1.compute.amazonaws.com	sfp_spider	2024-12-21 18:36:02
{"content-length": "563", "cache-control": "no-cache, no-store", "content-type": "text/html; charset=utf-8", "date": "2024-12-21 08:35:55.407170799 +0000 UTC", "server": "heroku-router"}	ec2-54-220-192-176.eu-west-1.compute.amazonaws.com	sfp_spider	2024-12-21 18:36:06
{"server": "Cowboy", "report-to": "{\"group\":\"heroku-nel\",\"max_age\":3600,\"endpoints\":[{\"url\":\"https://nel.heroku.com/reports?ts=1734767936&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=OulwXL0GkLIN0zpqP2affhYE7CGv3KUIHX450OclQZ4%3D\"}]}", "reporting-endpoints": "heroku-nel=https://nel.heroku.com/reports?ts=1734767936&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=OulwXL0GkLIN0zpqP2affhYE7CGv3KUIHX450OclQZ4%3D", "nel": "{\"report_to\":\"heroku-nel\",\"max_age\":3600,\"success_fraction\":0.005,\"failure_fraction\":0.05,\"response_headers\":[\"Via\"]}", "connection": "keep-alive", "access-control-allow-origin": "*", "x-content-type-options": "nosniff", "x-frame-options": "SAMEORIG	juice-shop.herokuapp.com	sfp_spider	2024-12-21 17:59:08

RedOps Crew

Browse / Non-Standard HTTP Header

Data Element	Source Data Element	Source Module	Identified
feature-policy: payment 'self'	{"server": "Cowboy", "report-to": "{\"group\":\"heroku-nel\",\"max_age\":3600,\"endpoints\":[{\"url\":\"https://nel.heroku.com/reports?ts=1734767936&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=OulwXL0GkLIN0zpqP2affhYE7CGv3KUIHX4500clQZ4%3D\"}]}", "reporting-endpoints": "heroku-nel=https://nel.heroku.com/reports?ts=1734767936&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=OulwXL0GkLIN0zpqP2affhYE7CGv3KUIHX4500clQZ4%3D", "nel": "{\"report_to\":\"heroku-nel\",\"max_age\":3600,\"success_fraction\":0.005,\"failure_fraction\":0.05,\"response_headers\":[\"Via\"]}", "connection": "keep-alive", "access-control-allow-origin": "*", "x-content-type-options": "nosniff", "x-frame-options": "SAMEORIGIN", "feature-policy": "payment 'self'", "x-recruiting": "/#/jobs", "accept-ranges": "bytes", "cache-control": "public, max-age=0", "last-modified": "Fri, 20 Dec 2024 23:09:26 GMT", "etag": "W/\"ea4-193e6553f79\"", "content-type": "text/html; charset=UTF-8", "vary": "Accept-Encoding", "content-encoding": "gzip", "date": "Sat, 21 Dec	sfp_stringheaders	2024-12-21 17:59:42
feature-policy: payment 'self'	{"server": "Cowboy", "report-to": "{\"group\":\"heroku-nel\",\"max_age\":3600,\"endpoints\":[{\"url\":\"https://nel.heroku.com/reports?ts=1734767937&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=kC64Kz97eoF33RZ6eSyZvmIJweASUgXHtN%2BYyt9O%2Fas%3D\"}]}", "reporting-endpoints": "heroku-nel=https://nel.heroku.com/reports?ts=1734767937&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=kC64Kz97eoF33RZ6eSyZvmIJweASUgXHtN%2BYyt9O%2Fas%3D", "nel": "{\"report_to\":\"heroku-nel\",\"max_age\":3600,\"success_fraction\":0.005,\"failure_fraction\":0.05,\"response_headers\":[\"Via\"]}", "connection": "keep-alive", "access-control-allow-origin": "*", "x-content-type-options": "nosniff", "x-frame-options": "SAMEORIGIN", "feature-policy": "payment 'self'", "x-recruiting": "/#/jobs", "accept-ranges": "bytes", "cache-control": "public, max-age=0", "last-modified": "Mon, 28 Oct 2024 15:59:55 GMT", "etag": "W/\"ce1-192d3daf478\"", "content-type": "application/javascript; charset=UTF-8", "vary": "Accept-Encoding", "content-encoding": "gzip",	sfp_stringheaders	2024-12-21 18:00:22

Co-Hosted Sites

Browse / Co-Hosted Site

Data Element	Source Data Element	Source Module	Identified
a12-map.herokuapp.com	46.137.15.86	sfp_hackertarget	2024-12-21 18:00:36
a316-gr.herokuapp.com	46.137.15.86	sfp_hackertarget	2024-12-21 18:00:36
aaf-site-2016.herokuapp.com	46.137.15.86	sfp_hackertarget	2024-12-21 18:00:37
aakonlabs.herokuapp.com	46.137.15.86	sfp_hackertarget	2024-12-21 18:00:37
abcdent-pro.herokuapp.com	46.137.15.86	sfp_hackertarget	2024-12-21 18:00:37
aboss-external-staging.herokuapp.com	46.137.15.86	sfp_hackertarget	2024-12-21 18:00:38
aboss-staging.herokuapp.com	46.137.15.86	sfp_hackertarget	2024-12-21 18:00:40

RedOps Crew

Affiliate and Related Data

Browse / Affiliate - Domain Name

Data Element	Source Data Element	Source Module	Identified
amazon.com	amazon.com	sfp_dnsresolve	2024-12-21 18:39:35
amazon.com	amazon.com	sfp_dnsresolve	2024-12-21 18:39:35
amazon.com	amazon-smtp.amazon.com	sfp_dnsresolve	2024-12-21 18:39:36

Browse / Affiliate - Email Address

Data Element	Source Data Element	Source Module	Identified
abuse@amazonaws.com	{'prefix': '46.137.0.0/16', 'ip': '46.137.0.0', 'cidr': 16, 'asns': [{'asn': 16509, 'name': 'AMAZON-02', 'description': 'Amazon.com, Inc.', 'country_code': 'US', 'prefix_upstreams': [{'asn': 1299, 'name': 'TWELVE99', 'description': 'Arelion, f/k/a Telia Carrier', 'country_code': 'SE'}, {'asn': 2914, 'name': 'NTT-LTD-2914', 'description': 'NTT America, Inc.', 'country_code': 'US'}, {'asn': 174, 'name': 'COGENT-174', 'description': 'Cogent Communications', 'country_code': 'US'}, {'asn': 6453, 'name': 'AS6453', 'description': 'TATA COMMUNICATIONS (AMERICA) INC', 'country_code': 'US'}]}], 'name': 'IE-AMAZON-20101116', 'description_short': 'Amazon Data Services Ireland Ltd', 'description_full': ['Amazon Data Services Ireland Ltd'], 'email_contacts': ['abuse@amazonaws.com', 'ripe-interest@amazon.com', 'noc@amazon.com'], 'abuse_contacts': ['abuse@amazonaws.com'], 'owner_address': ['1 Burlington Plaza', 'Burlington Road', 'D04RH96', 'Dublin', 'IRELAND'], 'country_codes': {'whois_country_code': 'IE', 'rir_allocation_c	sfp_email	2024-12-21 18:28:33

Browse / Affiliate - IP Address

Data Element	Source Data Element	Source Module	Identified
198.51.44.3	dns1.p03.nsone.net	sfp_dnsresolve	2024-12-21 18:27:33
198.51.44.67	dns3.p03.nsone.net	sfp_dnsresolve	2024-12-21 18:27:26

Browse / Affiliate - IPv6 Address

Data Element	Source Data Element	Source Module	Identified
2600:9000:5300:1b00::1	ns-27.awsdns-03.com	sfp_dnsresolve	2024-12-21 18:39:38
2600:9000:5303:c700::1	ns-967.awsdns-56.net	sfp_dnsresolve	2024-12-21 18:39:36

Browse / Affiliate - Internet Name

Data Element	Source Data Element	Source Module	Identified
amazon-smtp.amazon.com	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45
amazon.com	amazonaws.com	sfp_dnsraw	2024-12-21 18:22:45

RedOps Crew

Interesting Files

Browse / Interesting File

Data Element	Source Data Element	Source Module	Identified
https://juice-shop.herokuapp.com/ftp/order_0256-24a908c3362002fa.pdf	https://juice-shop.herokuapp.com/ftp/order_0256-24a908c3362002fa.pdf	sfp_intfiles	2024-12-21 18:34:03
https://juice-shop.herokuapp.com/ftp/order_11b3-19b981c63869ef92.pdf	https://juice-shop.herokuapp.com/ftp/order_11b3-19b981c63869ef92.pdf	sfp_intfiles	2024-12-21 18:34:05

Raw Data from APIs/RIRs

Browse / Raw Data from RIRs/APIs

Data Element	Source Data Element	Source Module	Identified
[{'submitter': {}, 'task': {'visibility': 'public', 'method': 'api', 'domain': 'doebaya.be', 'apexDomain': 'doebaya.be', 'time': '2024-12-21T07:57:55.159Z', 'uuid': '69a68fd6-7b8e-46f7-95f1-eda041cb18f8', 'url': 'https://doebaya.be/', 'tags': ['falconsandbox']}, 'stats': {'uniqIPs': 13, 'uniqCountries': 4, 'dataLength': 64591639, 'encodedDataLength': 1122794, 'requests': 98}, 'page': {'country': 'CA', 'server': 'cloudflare', 'redirected': 'off-domain', 'ip': '23.227.38.69', 'mimeType': 'text/html', 'title': 'Doebaya | FOR ALL YOUR ABAYAS', 'url': 'https://doebaya.com/', 'tlsValidDays': 90, 'tlsAgeDays': 0, 'tlsValidFrom': '2024-12-20T15:48:36.000Z', 'domain': 'doebaya.com', 'apexDomain': 'doebaya.com', 'asnname': 'CLOUDFLARENET, US', 'asn': 'AS13335', 'tlsIssuer': 'WE1', 'status': '200'}, '_id': '69a68fd6-7b8e-46f7-95f1-eda041cb18f8', '_score': None, 'sort': [1734767875159, '69a68fd6-7b8e-46f7-95f1-eda041cb18f8'], 'result': 'https://urlscan.io/api/v1/result/69a68fd6-7b8e-46f7-95f1-eda041cb18f8/', 'screenshot'	ec2-46-137-15-86.eu-west-1.compute.amazonaws.com	sfp_urlscan	2024-12-21 18:22:28

Linked URLs (External/Internal)

Browse / Linked URL - External

Data Element	Source Data Element	Source Module	Identified
http://cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css	http://juice-shop.herokuapp.com/	sfp_spider	2024-12-21 17:59:08
http://cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js	http://juice-shop.herokuapp.com/	sfp_spider	2024-12-21 17:59:08

Browse / Linked URL - Internal

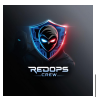
Data Element	Source Data Element	Source Module	Identified
http://ec2-46-137-15-86.eu-west-1.compute.amazonaws.com	ec2-46-137-15-86.eu-west-1.compute.amazonaws.com	sfp_spider	2024-12-21 18:35:51
http://ec2-54-220-192-176.eu-west-1.compute.amazonaws.com	ec2-54-220-192-176.eu-west-1.compute.amazonaws.com	sfp_spider	2024-12-21 18:36:05

RedOps Crew

Cloud Storage Buckets

Browse / Cloud Storage Bucket

Data Element	Source Data Element	Source Module	Identified
ec2-46-137-15-86.eu-west-1.compute.amazonaws.com	http://ec2-46-137-15-86.eu-west-1.compute.amazonaws.com/	sfp_s3bucket	2024-12-21 18:35:52
ec2-54-220-192-176.eu-west-1.compute.amazonaws.com	http://ec2-54-220-192-176.eu-west-1.compute.amazonaws.com/	sfp_s3bucket	2024-12-21 18:36:12
ec2-54-73-53-134.eu-west-1.compute.amazonaws.com	http://ec2-54-73-53-134.eu-west-1.compute.amazonaws.com/	sfp_s3bucket	2024-12-21 18:36:04



RedOps Crew

Key Findings from Ns lookup:

Ns lookup was performed to gather DNS information for the domain we are targeting. We have gathered following information using nslookup.

IP Addresses:

The following ip addresses are associated with our target website:

- **46.137.15.86**
- **54.220.192.176**
- **54.73.53.134**

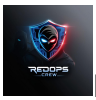
We can use these ip addresses to look for open ports, services being used and gain further insights to figure out the vulnerabilities which could be exploited.

DNS Server Information:

- DNS Server: 192.168.227.2
- DNS Server Address: 192.168.227.2#53

We can further use this information for network mapping, identifying potential hosting locations and further reconnaissance.

Screenshots:



RedOps Crew

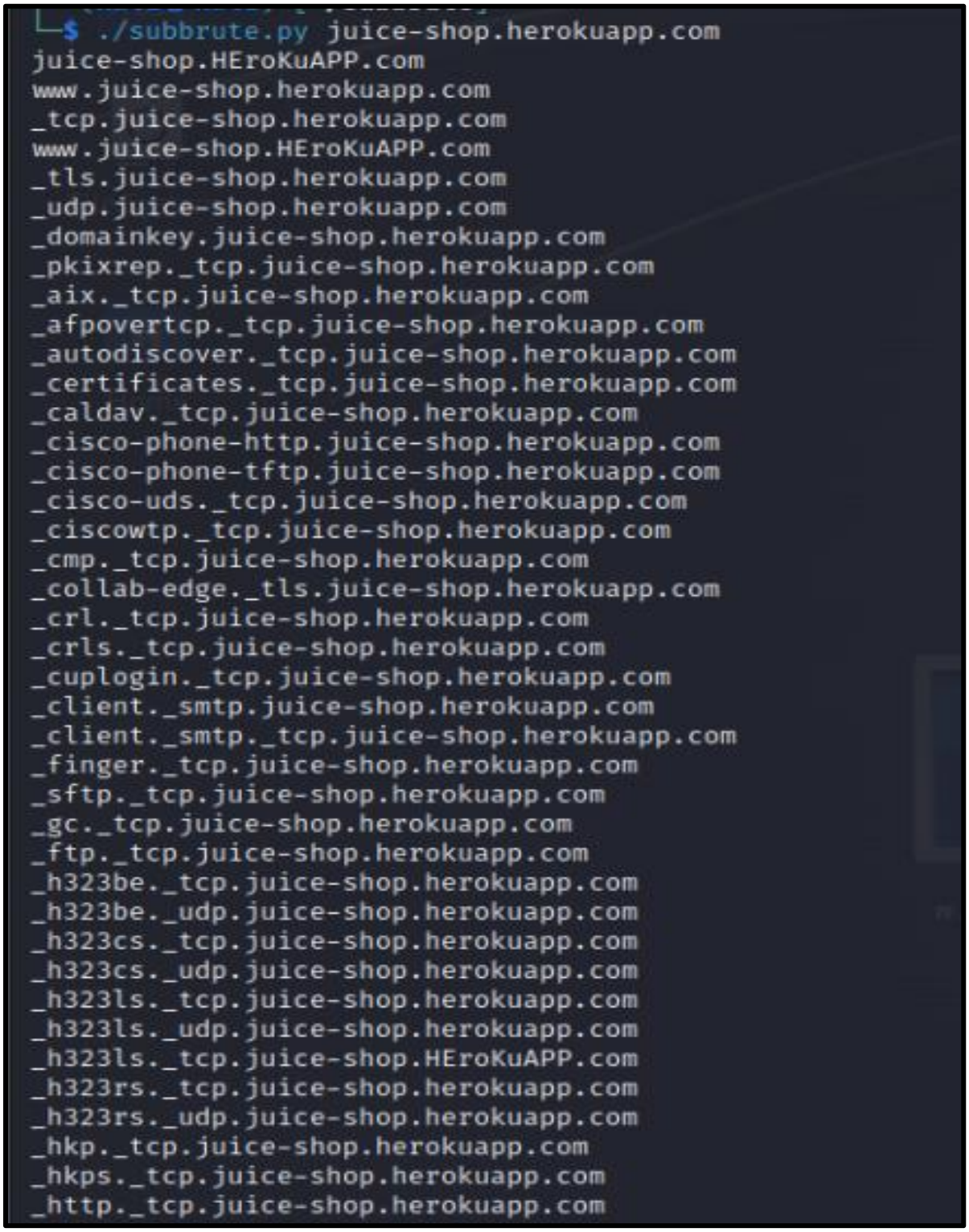
Key Findings from Subbrute:

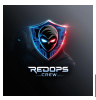
Subbrute is used to find sub domains of the target domain which could help us get further insights of the sub domains which may contain some sensitive information.The sub domains extracted by subbrute are as follows I have listed some of them.

- juice-shop.HEroKuAPP.com
- www.juice-shop.herokuapp.com
- tcp.juice-shop.herokuapp.com
- www.juice-shop.HEroKuAPP.com

These domains can be used to get further insights of the target as these sub domains may contain sensitive files which are not secured or may be misconfigured. Misconfigurations or exposed files on these subdomains may offer further attack vectors for an attacker to exploit.

Screenshots:





RedOps Crew

Scanning And Enumeration:

In scanning and enumeration phase we use the information Gathered in our reconnaissance phase to gain further insights. The primary goal of this phase is to actively scan and probe the target system to identify open ports, running services, and potential vulnerabilities that an attacker could exploit. By conducting detailed scanning, we can further uncover opportunities for exploitation that were not visible during the initial phase of reconnaissance.

Tools:

The tools we will be using in this phase are as follows:

- **Dirb**
- **Nmap**
- **Owasp Zap**
- **BurpSuit**
- **Nikto**

Dirb:

Dirb is a popular command-line-based web content scanner designed for penetration testing and security assessments. It is used to discover hidden files, directories, and content on a web server by brute-forcing its way through with a wordlist.

Report:

Report on Directory Scanning Results for OWASP Juice Shop:

Scan Details:

1. **Target URL:** http://juice-shop.herokuapp.com
2. **Wordlist used:**/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

RedOps Crew

Discovered Directories and Files:

- **./ftp:**

We have founded an ftp directory with status code 503 means unavailable , but the directory has listed files within it, which could indicate exposed sensitive data or misconfigured FTP services.

- **/promotion:**

Another directory listed by dirb is promotion with status code 200 means ok we can view it .This contains promotional data which could be analyzed for sensitive information .

- **./robots.txt:**

This directory Indicates that the /ftp directory is excluded from indexing by search engines. No other directories are restricted.

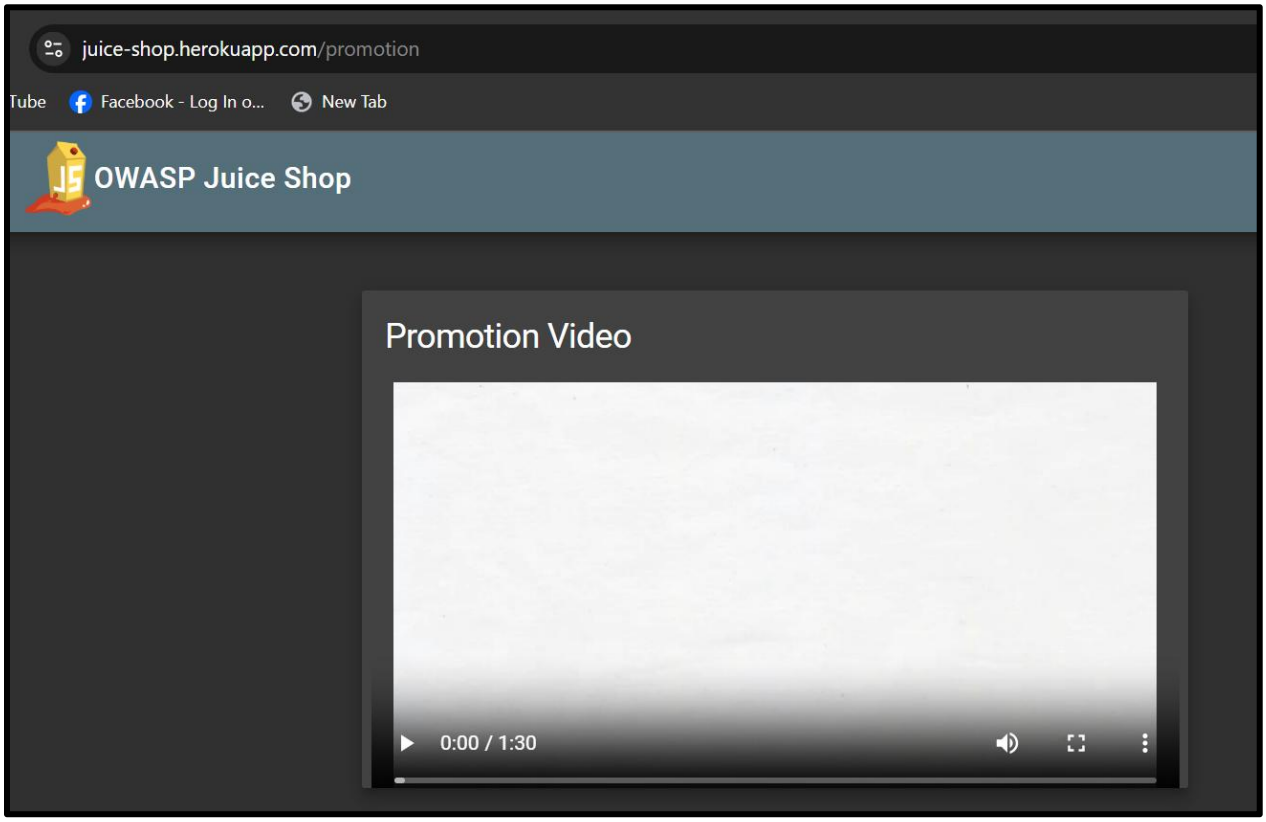
- **./snippets:**

This directory with code 200 means we can access it. It provides a list of challenges implemented in the application.These challenges indicates areas where vulnerabilites exists.

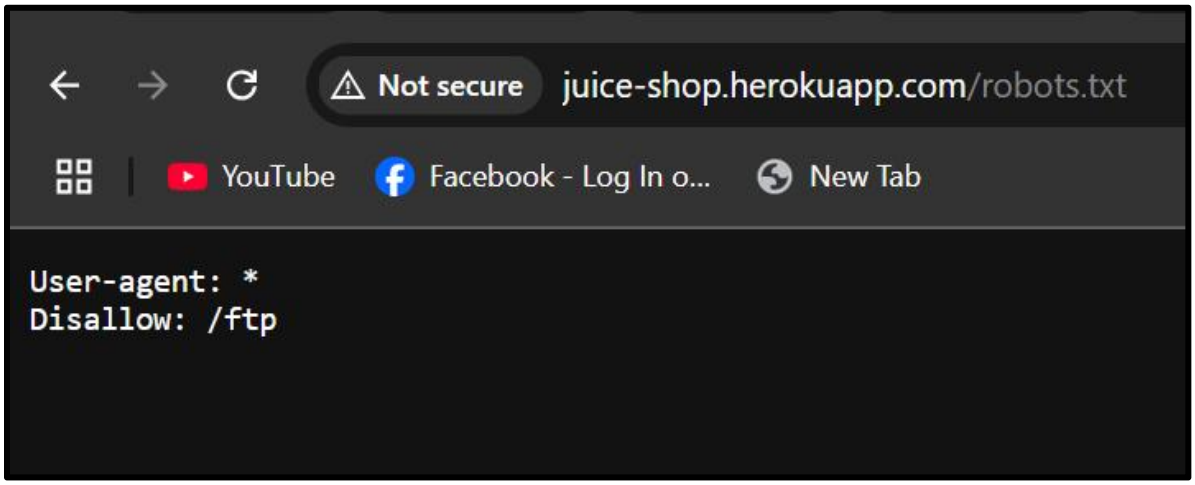
Screenshot:

```
-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Dec 22 17:16:15 2024
URL_BASE: http://juice-shop.herokuapp.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://juice-shop.herokuapp.com/ ----
+ http://juice-shop.herokuapp.com/assets (CODE:301|SIZE:156)
+ http://juice-shop.herokuapp.com/ftp (CODE:503|SIZE:506)
+ http://juice-shop.herokuapp.com/profile (CODE:500|SIZE:1147)
+ http://juice-shop.herokuapp.com/promotion (CODE:200|SIZE:6586)
+ http://juice-shop.herokuapp.com/redirect (CODE:500|SIZE:2965)
+ http://juice-shop.herokuapp.com/robots.txt (CODE:200|SIZE:28)
+ http://juice-shop.herokuapp.com/snippets (CODE:200|SIZE:792)
+ http://juice-shop.herokuapp.com/video (CODE:200|SIZE:10075518)
+ http://juice-shop.herokuapp.com/Video (CODE:200|SIZE:10075518)

-----------------
END_TIME: Sun Dec 22 17:37:38 2024
DOWNLOADED: 4612 - FOUND: 9
```

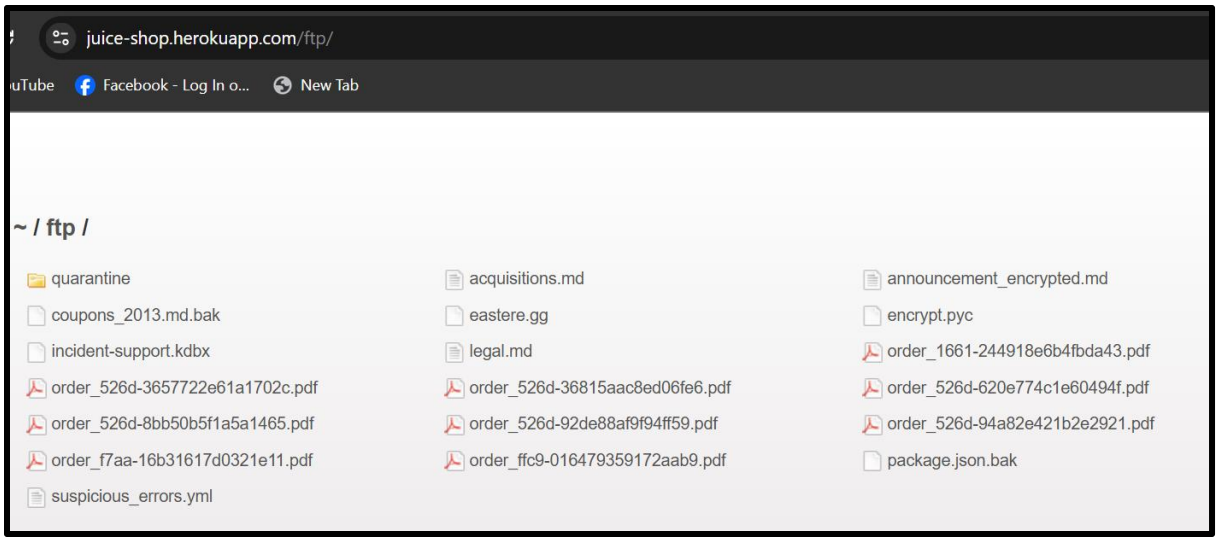
./Promotion directory Screenshot:



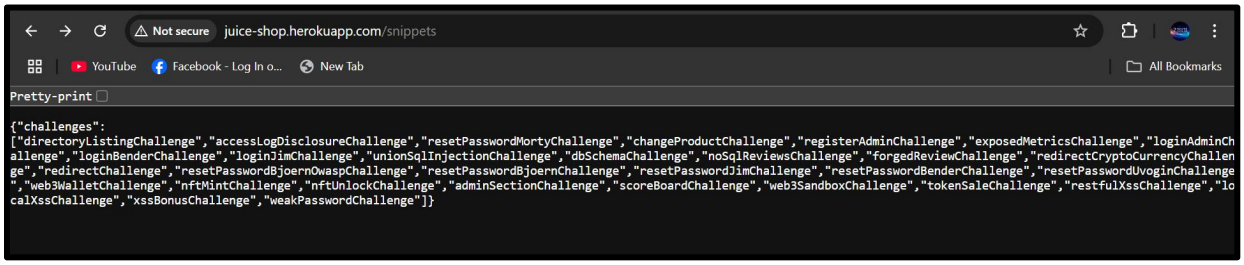
./robots.txt screenshot:

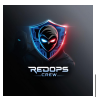


./ftp screenshot:



./snippets screenshot:





RedOps Crew

Nmap:

Nmap (Network Mapper) is a powerful and widely used open-source tool for network discovery and security auditing. It helps identify hosts, services, operating systems, and vulnerabilities on a network, making it an essential tool for penetration testers and system administrators. Known for its versatility, Nmap supports various scanning techniques, from simple ping sweeps to advanced port scanning and scripting.

Scan Details:

Targer ip addresses: 46.137.15.86, 54.220.192.176, 54.73.53.134

Scan Results:

Key Findings for the first ip address:

Service: tcppwrapped

HTTP Title:Heroku | Application error:

This indicates that the target is likely hosted on Heroku, and the application is misconfigured or encountering issues.

OS Detection: Netgear SC101 NAS or Brother MFC-7820N Printer, These guesses might be incorrect as our services are behind the security layer.

Traceroute:

Shows a distance of 2 hops.

SSL Certificate:

Common Name: *.herokuapp.com.

Validity: From March 2024 to March 2025

We have found similar results for the other two ip addresses as well.

Screenshots:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-28 21:31 PKT
Nmap scan report for ec2-46-137-15-86.eu-west-1.compute.amazonaws.com (46.137.15.86)
Host is up (0.037s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE   VERSION
80/tcp  open  http      heroku-router
```

```
|_http-server-header: heroku-router
443/tcp open  ssl/https heroku-router
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=*.herokuapp.com
| Subject Alternative Name: DNS:*.herokuapp.com
| Not valid before: 2024-03-02T00:00:00
|_Not valid after:  2025-03-31T23:59:59
```

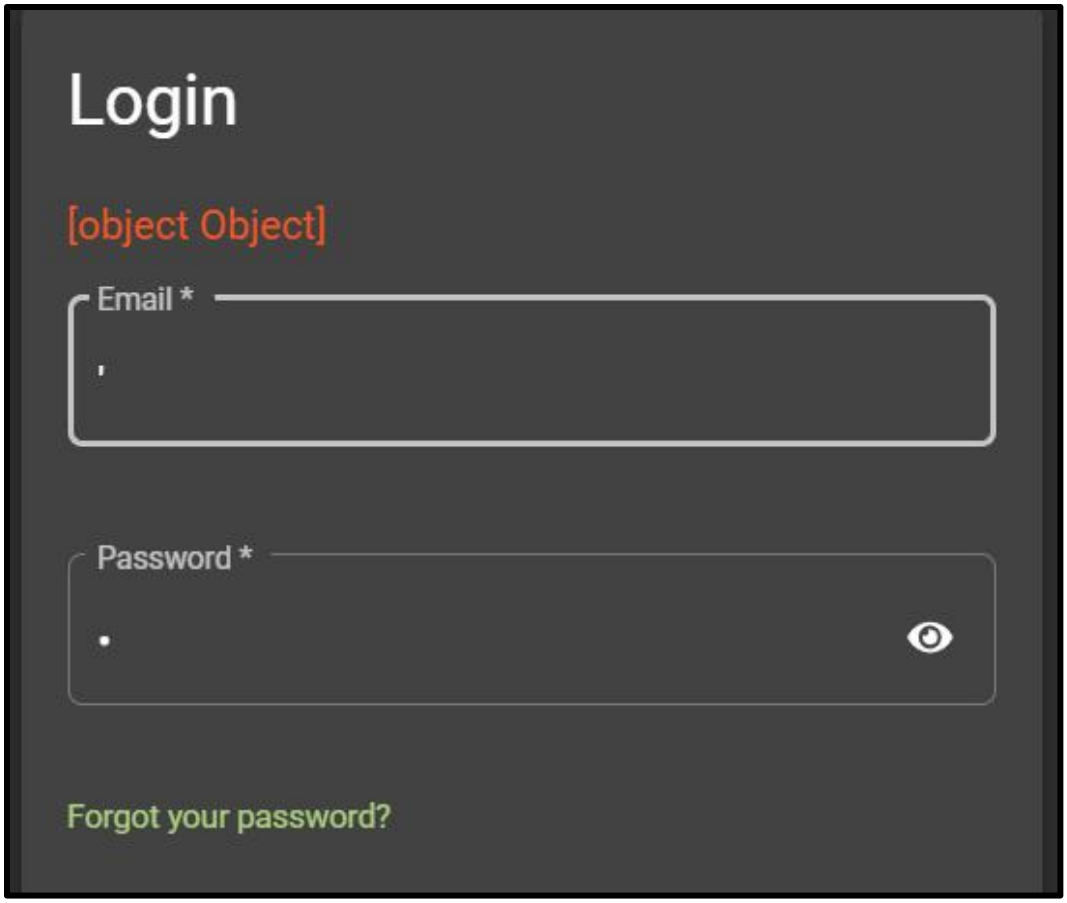
RedOps Crew

Manually Testing Owasp Juice Shop:

SQL Injection:

On the login page of OWASP Juice Shop, I entered a single quote (') in both the email and password input fields to check for SQL Injection vulnerabilities. The resulting **[object Object]** error indicates that the application is improperly handling or escaping user input. This behavior suggests that the application might be susceptible to SQL Injection attacks and warrants further testing to confirm the vulnerability.

We can also use burp suit to find out if the application is vulnerable to Sql injection or not.



RedOps Crew

Broken Access Control or Insecure Direct Object References:

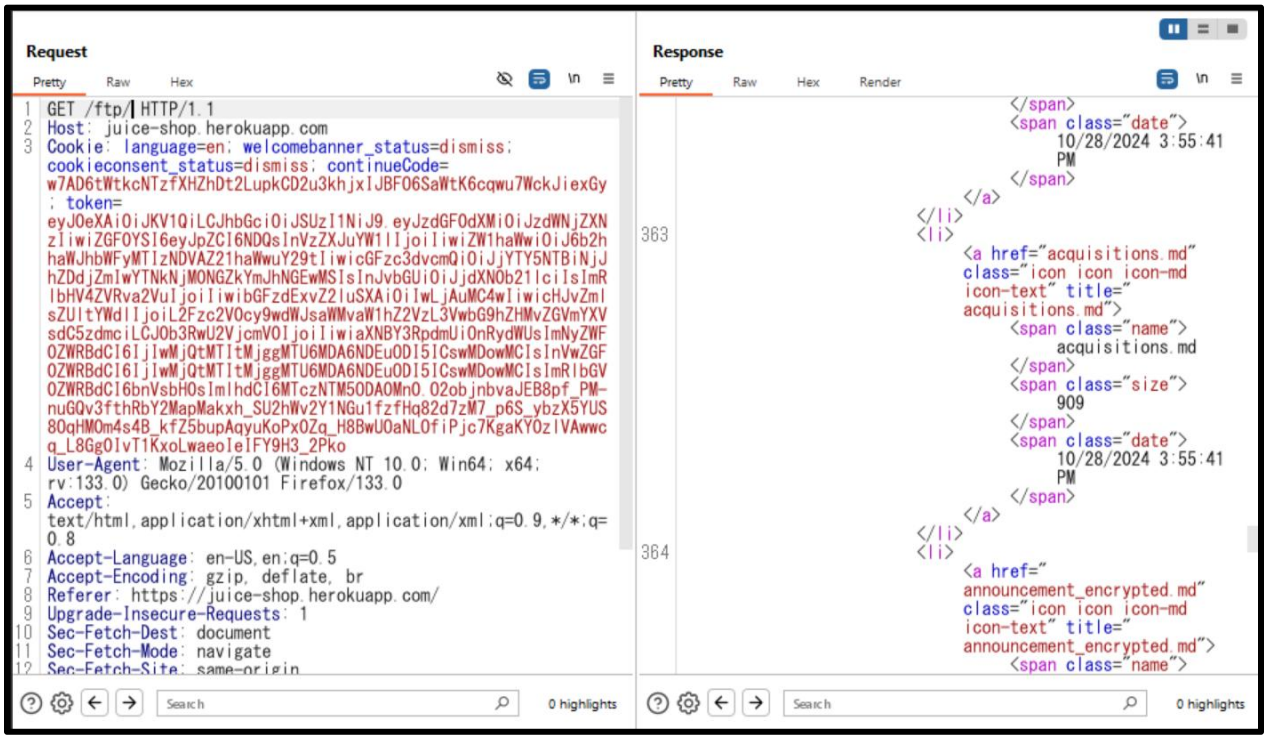
Broken Access Control is a security vulnerability that occurs when users can access data, resources, or functionalities beyond their intended permissions. It results from improper enforcement of access control policies, allowing attackers to bypass restrictions and gain unauthorized access.

While testing for Broken Access Control, we utilized the Burp Suite tool to identify vulnerabilities. By capturing and analyzing requests, we noticed that the web application's API assigns a unique ID to each user. Manipulating this unique ID in Burp Suite and forwarding the modified request granted access to another user's bucket, exposing improper enforcement of access control.

Sensitive Data Exposure:

When navigating to the About Us page of the OWASP Juice Shop, we encounter a line that says, “Check out our boring terms of use if you are interested in such lame stuff.” Clicking this line initiates the download of a file. By capturing this request in Burp Suite, we observe a GET request for the file named /ftp/legal.md.

We send this request to the Repeater tool in Burp Suite and modify the file name to /ftp/. After forwarding the modified request, the response reveals a large amount of data. Scrolling through the response, we find a directory listing of legal files, exposing sensitive information due to improper access control on the server.



Broken Authentication:

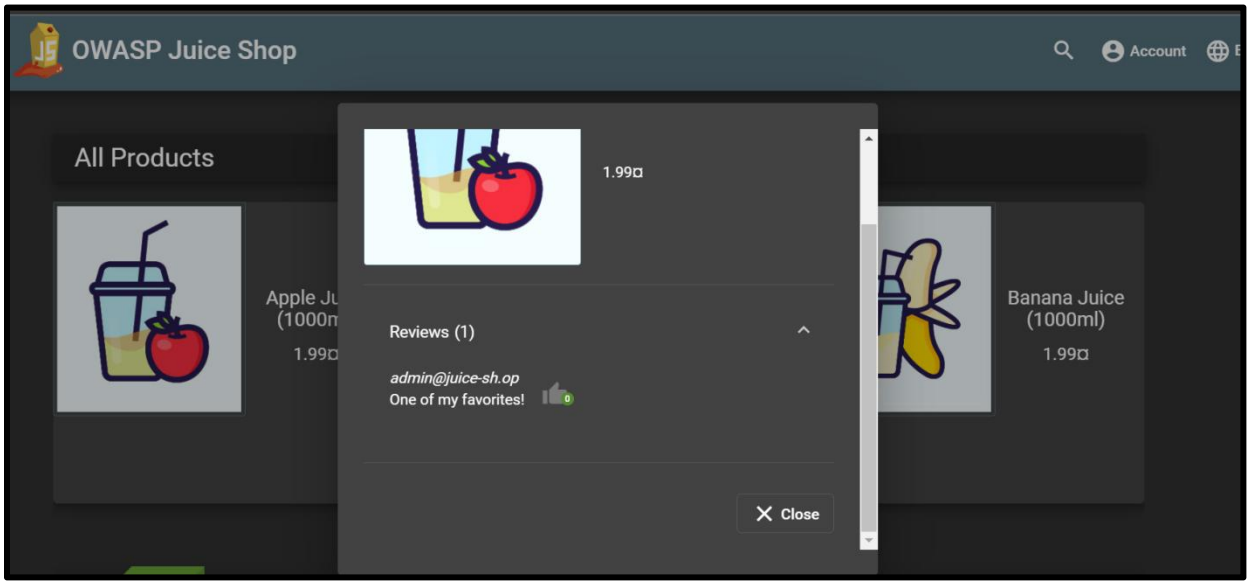
Broken Authentication is a security vulnerability that occurs when an application's authentication mechanisms are improperly implemented, allowing attackers to bypass authentication and impersonate legitimate users.

Common causes include:

1. Weak Password Policies: Allowing easy-to-guess passwords.
2. Session Management Issues: Exposing session IDs in URLs or not expiring them after logout.
3. Lack of Multi-Factor Authentication (MFA): Missing additional security layers

Upon navigating to the "All Products" page of the OWASP Juice Shop, the list of available products is displayed. By inspecting the review section associated with each product, it is possible to identify sensitive information such as the administrator's email address. This email address could potentially be leveraged to perform a brute-force attack on the admin login page using tools like Burp Suite, highlighting vulnerabilities in both data exposure and authentication mechanisms.

This information reveals a broken authentication vulnerability because it exposes weaknesses in the application's ability to protect sensitive user accounts, particularly the administrator's



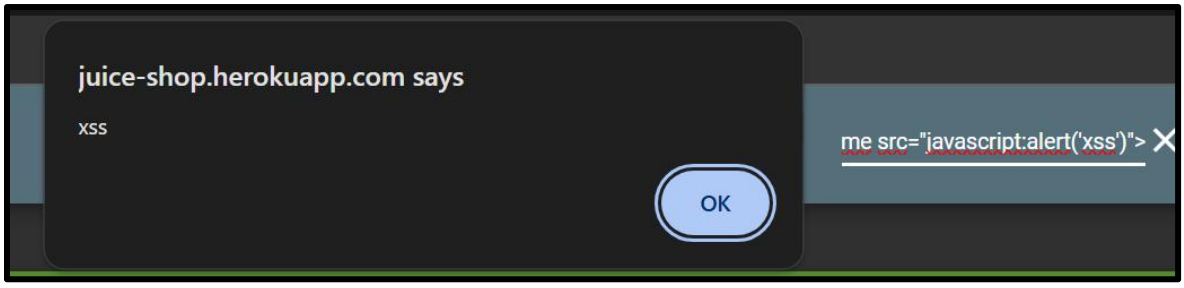
Dom XSS:

DOM XSS (DOM-based Cross-Site Scripting) happens when malicious scripts are injected into a website and executed directly in the user's browser, without involving the server. This occurs because the website's JavaScript code improperly handles untrusted data (like URL parameters) and updates the web page's content or structure (DOM) insecurely

To identify a DOM XSS vulnerability in the OWASP Juice Shop, we can attempt to inject a JavaScript payload into an input field.

```
<iframe src="javascript:alert('xss')">
```

Upon submitting the form, the browser will execute the injected JavaScript, triggering an alert pop-up with the message "xss." This indicates that the input is being improperly handled and is directly injected into the DOM without proper sanitization. This kind of vulnerability can be further exploited to execute malicious scripts or steal sensitive data, making it a critical issue that needs to be addressed.

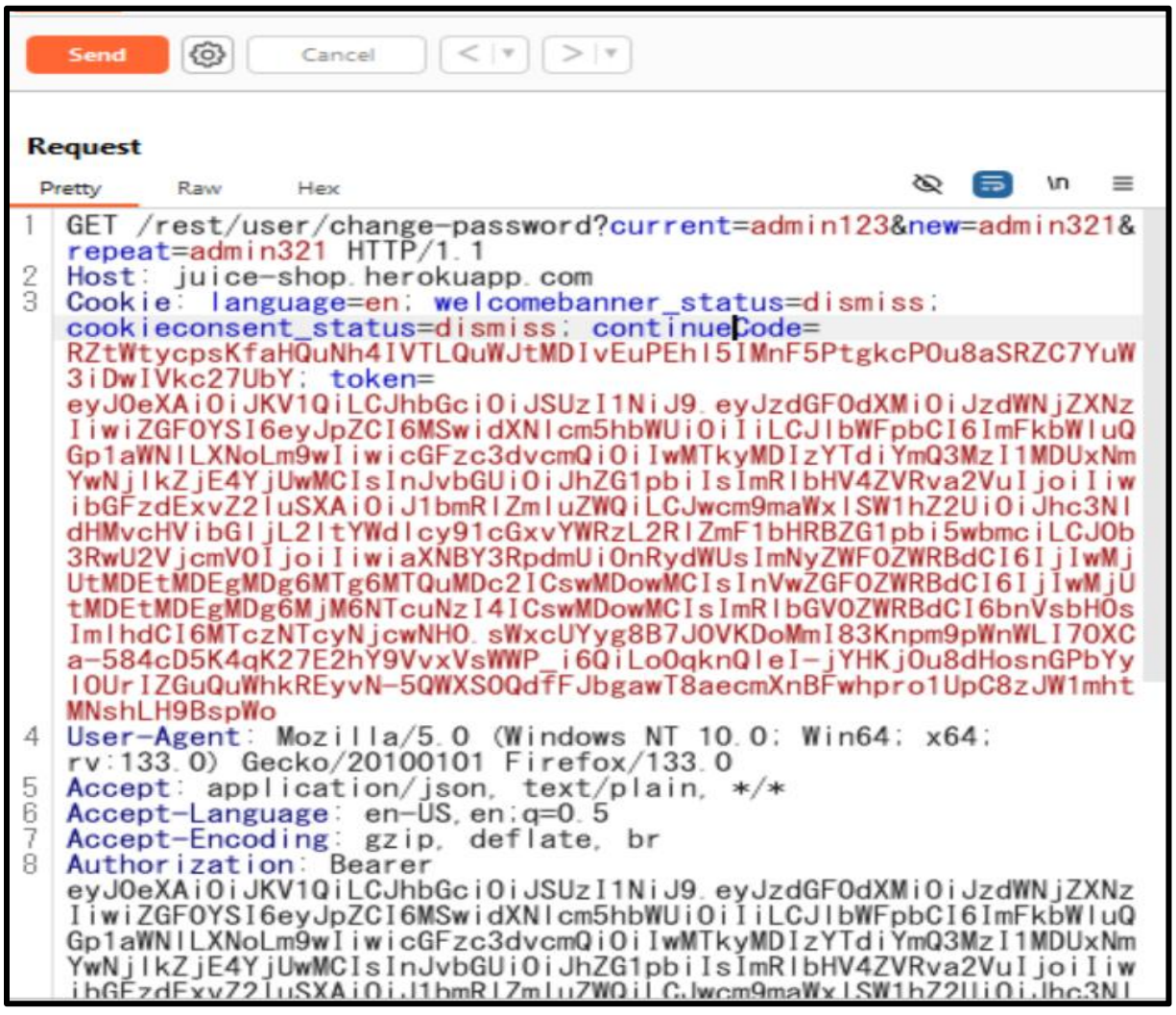


RedOps Crew

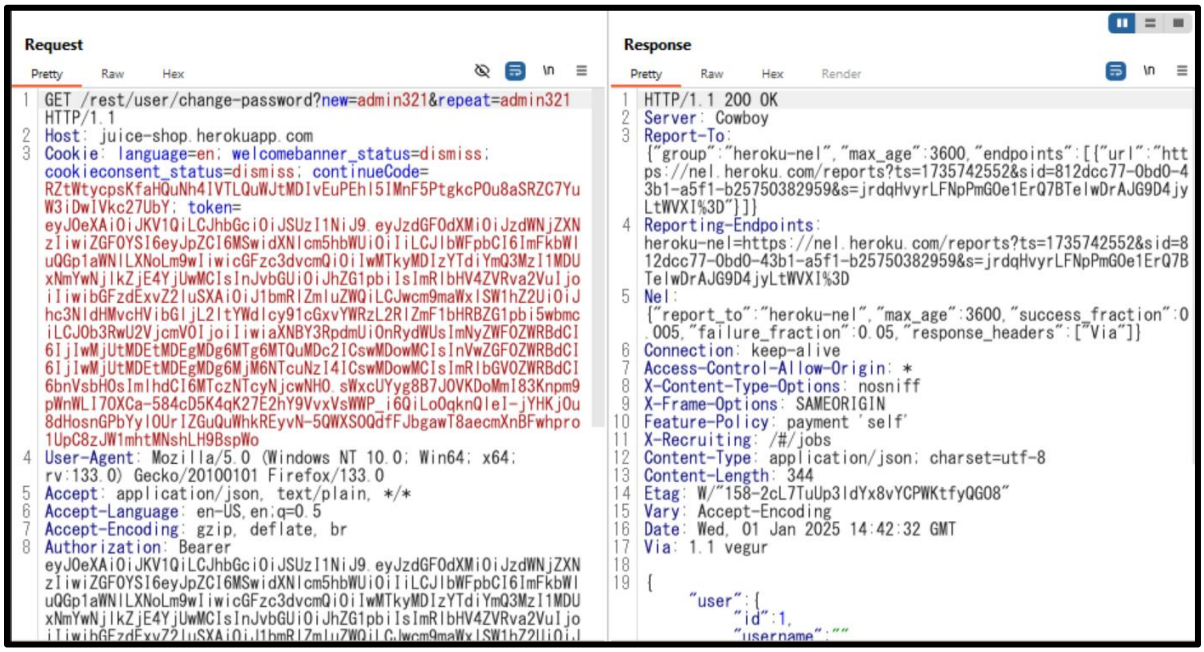
Cross Site Request Forgery:

Cross-Site Request Forgery (CSRF) is an attack where a hacker tricks a logged-in user into unknowingly performing unauthorized actions on a website by exploiting the trust the website has in the user's browser, such as automatically sending cookies with requests.

We will change the password of our account in the OWASP Juice Shop application and open Burp Suite's HTTP Proxy tab to analyze the request. The request will then be forwarded to the Repeater tab for further inspection, as demonstrated below:



Next, we will modify the GET request by removing the value of the current password parameter. Upon sending the altered request, we will observe that it is still successfully processed as shown below:



The server accepts any request originating from a logged-in user without verifying the current password, thereby exposing a vulnerability to Cross-Site Request Forgery (CSRF) attacks.

RedOps Crew

Owasp Zap:

The OWASP Zed Attack Proxy (ZAP) is a powerful and open-source web application security scanner developed by the Open Web Application Security Project (OWASP). It is widely used by security professionals and developers to identify vulnerabilities in web applications during development and testing.

ZAP acts as a man-in-the-middle proxy, intercepting and inspecting HTTP/HTTPS traffic between the client and the server. Its intuitive interface and extensive automation capabilities make it a popular choice for both beginners and experienced penetration testers.

Key Findings from Owasp Zap:

Content Security Policy (CSP) Header Not Set:

The server is not enforcing a Content Security Policy (CSP). This increases the risk of Cross-Site Scripting (XSS) attacks and data injection vulnerabilities.

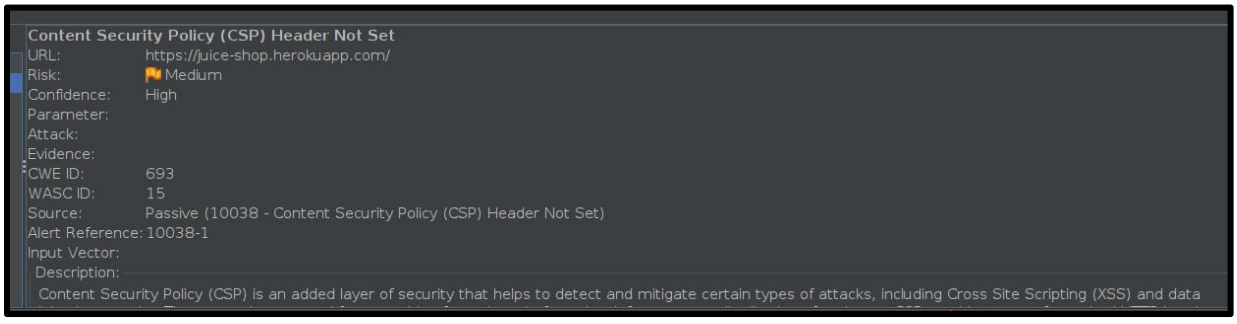
Strict-Transport-Security Header Not Set:

HTTP Strict Transport Security (HSTS) is not enforced, making the application vulnerable to SSL stripping attacks.

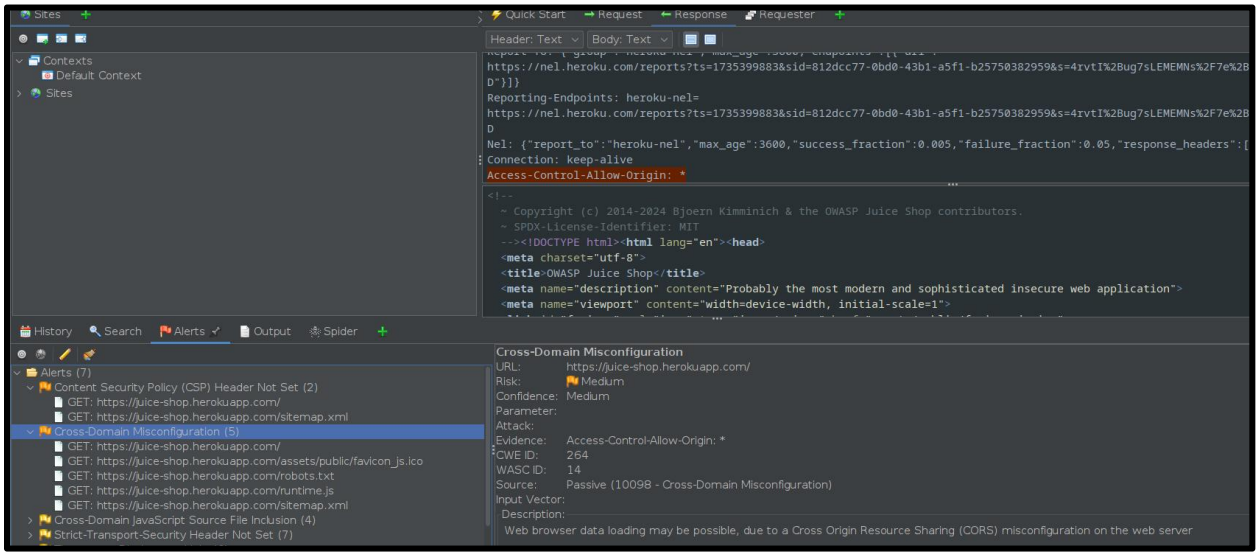
Cross-Domain Misconfiguration:

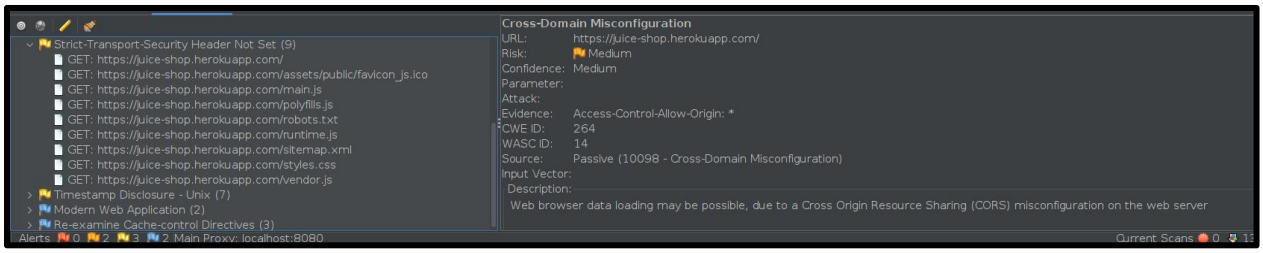
Cross-origin requests may be improperly handled, which can lead to data exposure or unauthorized access.

Screenshots:



RedOps Crew


Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
Contexts
Default Context
Sites
Access-Control-Allow-Origin: *
History
Search
Alerts
Output
Spider
Alerts (7)
Content Security Policy (CSP) Header Not Set (2)
GET: https://juice-shop.herokuapp.com/
GET: https://juice-shop.herokuapp.com/sitemap.xml
Cross-Domain Misconfiguration (5)
GET: https://juice-shop.herokuapp.com/
GET: https://juice-shop.herokuapp.com/assets/public/favicon_js.ico
GET: https://juice-shop.herokuapp.com/robots.txt
GET: https://juice-shop.herokuapp.com/runtime.js
GET: https://juice-shop.herokuapp.com/sitemap.xml
Cross-Domain JavaScript Source File Inclusion (4)
Strict-Transport-Security Header Not Set (7)
Cross-Domain Misconfiguration
URL: https://juice-shop.herokuapp.com/
Risk: Medium
Confidence: Medium
Parameter:
Attack:
Evidence: Access-Control-Allow-Origin: *
CWE ID: 264
WASC ID: 14
Source: Passive (10098 - Cross-Domain Misconfiguration)
Input Vector:
Description:
Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server


Strict-Transport-Security Header Not Set (9)
GET: https://juice-shop.herokuapp.com/
GET: https://juice-shop.herokuapp.com/assets/public/favicon_js.ico
GET: https://juice-shop.herokuapp.com/main.js
GET: https://juice-shop.herokuapp.com/polyfills.js
GET: https://juice-shop.herokuapp.com/robots.txt
GET: https://juice-shop.herokuapp.com/runtime.js
GET: https://juice-shop.herokuapp.com/sitemap.xml
GET: https://juice-shop.herokuapp.com/styles.css
GET: https://juice-shop.herokuapp.com/vendor.js
Timestamp Disclosure - Unix (7)
Modern Web Application (2)
Re-examine Cache-control Directives (3)
Cross-Domain Misconfiguration
URL: https://juice-shop.herokuapp.com/
Risk: Medium
Confidence: Medium
Parameter:
Attack:
Evidence: Access-Control-Allow-Origin: *
CWE ID: 264
WASC ID: 14
Source: Passive (10098 - Cross-Domain Misconfiguration)
Input Vector:
Description:
Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server
Alerts
Main Proxy: localhost:8080
Current Scans

Nikto:

Nikto is an open-source web server scanner designed to identify potential vulnerabilities and security issues in web applications. It performs comprehensive tests against web servers, scanning for dangerous files, outdated software, misconfigurations, and other common issues.

Key Findings from Nikto Scan:

Following are key findings from the nikto scan written below:

Strict-Transport-Security-Header Missing:

- This site does not enforce secure connections via HTTPs due to absence of the Strict-Transport-Security header.
- Vulnerable to man in the middle attack, potentially downgrading HTTPS connections to HTTP.
- User may unknowingly transmit sensitive information over an insecure connection.

Uncommon Header Found:

- **x-recuriting:** header contains /#/job, which may reveal internal job postings or development-related opportunities
- **reporting-endpoints**:header contains metadata such as heroku-nel, which could aid reconnaissance by revealing internal configurations
- Attackers could analyze uncommon headers for metadata or leverage them in targeted attacks

Missing X-Content-Type-Options Header:

- The **X-Content-Type-Options header** is not set, allowing browsers to interpret files as a different MIME type.
- This could allow MIME sniffing attacks, enabling attackers to exploit unintended browser behavior.

Wildcard SSL Certificate:

- The server uses a wildcard SSL certificate (*.herokuapp.com) issued by Amazon.
- While not inherently vulnerable, a compromised wildcard certificate affects all subdomains

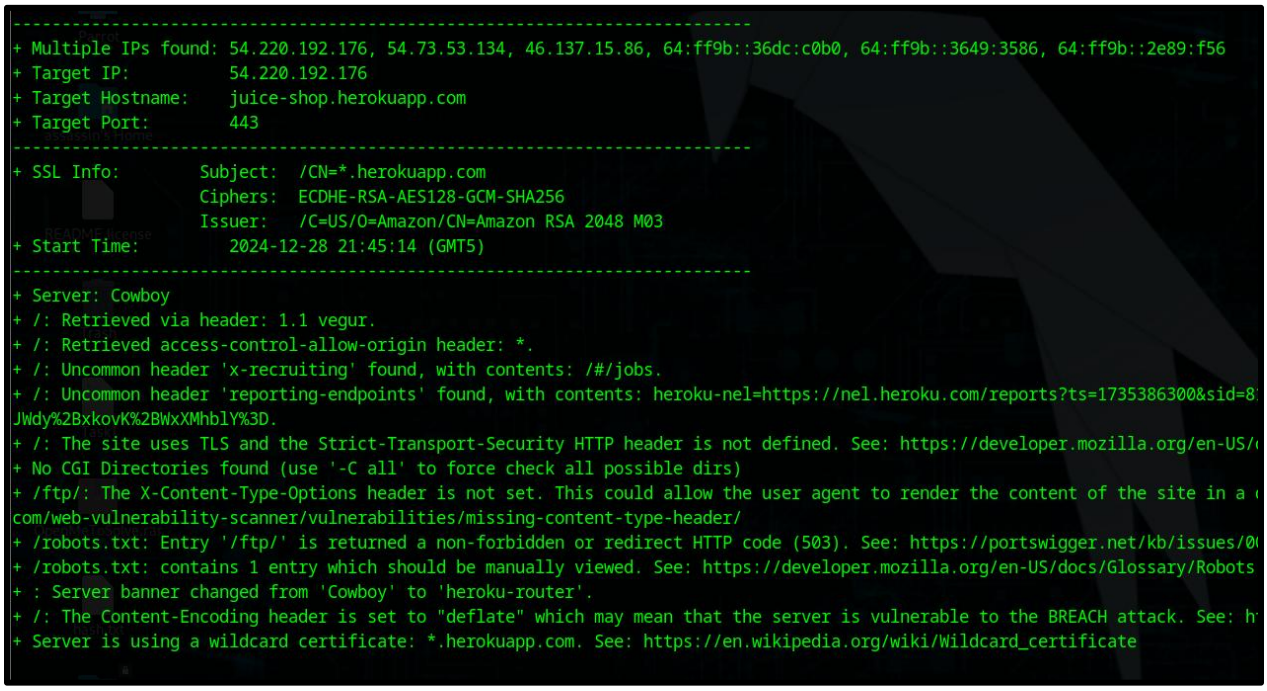
BREACH Attack Potential:

- The Content-Encoding header is set to deflate, indicating potential vulnerability to the BREACH attack
- BREACH attacks exploit HTTP compression to recover sensitive information like CSRF tokens or session cookies.
- A BREACH attack is a way hackers can steal sensitive data (like session cookies or security tokens) by exploiting how websites compress data before sending it to your browser. The trick lies in figuring out what sensitive information is hidden in the compressed response by testing how the response size changes.

Server Banner Inconsistencies:

- The server banner changes from Cowboy to heroku-router during the scan.
- Inconsistent banners leak information about the underlying infrastructure, aiding reconnaissance and fingerprinting.

Screenshots:



Exploitation:

Exploitation refers to the process of taking advantage of vulnerabilities or misconfigurations within a system to gain unauthorized access, disrupt operations, or extract sensitive information. Attackers often use tools and techniques to identify weaknesses, such as missing security headers, misconfigured files, or exposed metadata. By understanding these vulnerabilities, defenders can proactively mitigate risks and strengthen the security posture of their systems.

We will use the vulnerabilities identified during the scanning and enumeration phases to exploit the application, gaining unauthorized access, extracting sensitive data, escalating privileges, or performing other actions that demonstrate the impact of these security weaknesses.

SQL Injection Attack:

We will leverage the SQL injection vulnerability to craft malicious SQL queries, allowing us to bypass authentication mechanisms and log in to the web application with administrative privileges. This will enable unauthorized access to sensitive features and data.

We will use simple SQL payloads in the email and password fields to exploit the vulnerability. These payloads will manipulate the SQL query logic to bypass authentication and gain access to the web application.

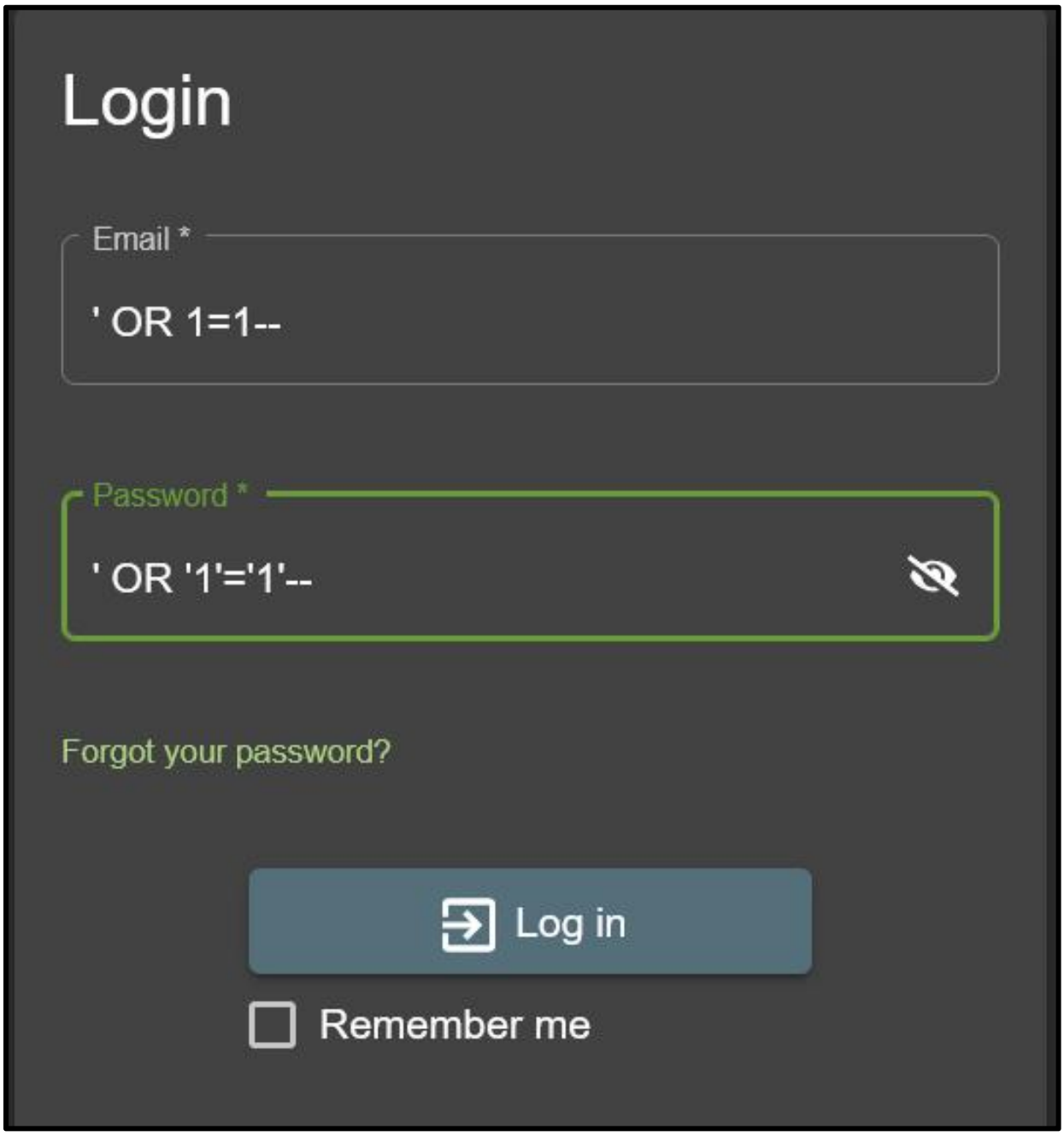
For email :

```
' OR 1=1--
```

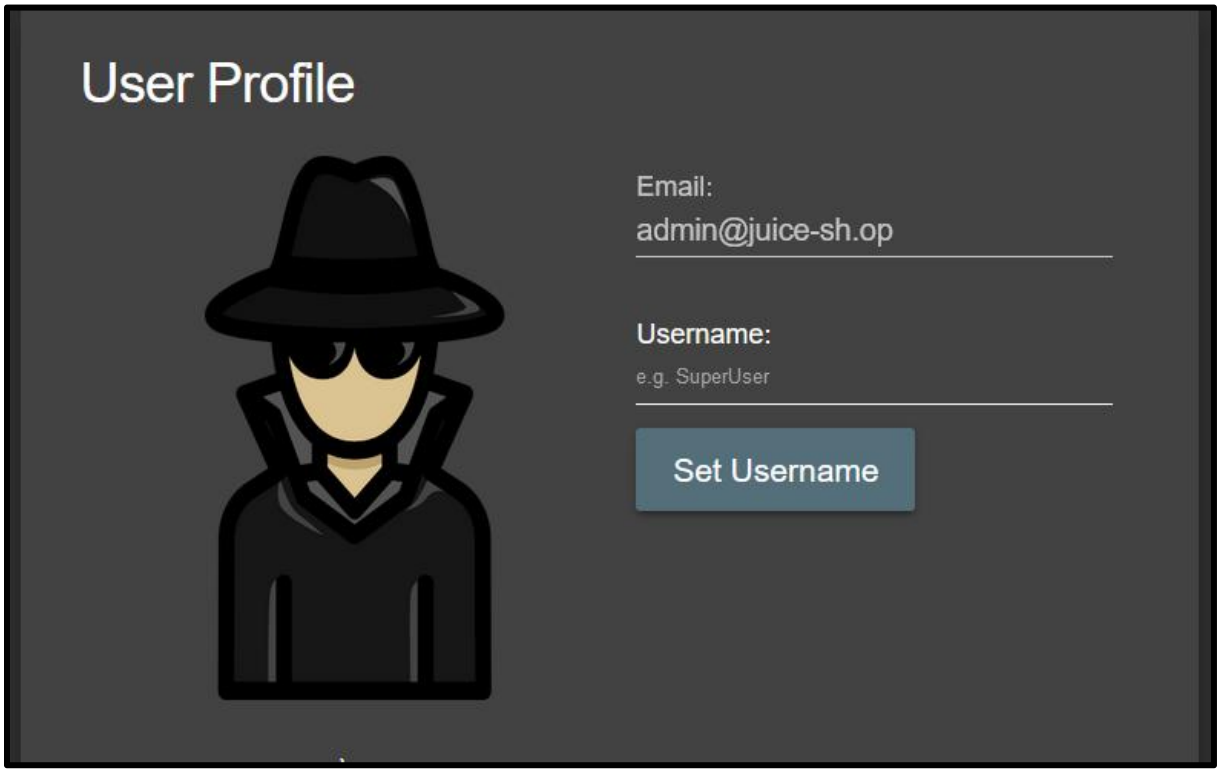
For password:

```
' OR '1'='1'--
```

Screenshot:


Login
Email *
' OR 1=1--
Password *
' OR '1'='1'--
Forgot your password?
Log in
Remember me

As you can see we have logged in to the application in the below screenshot.



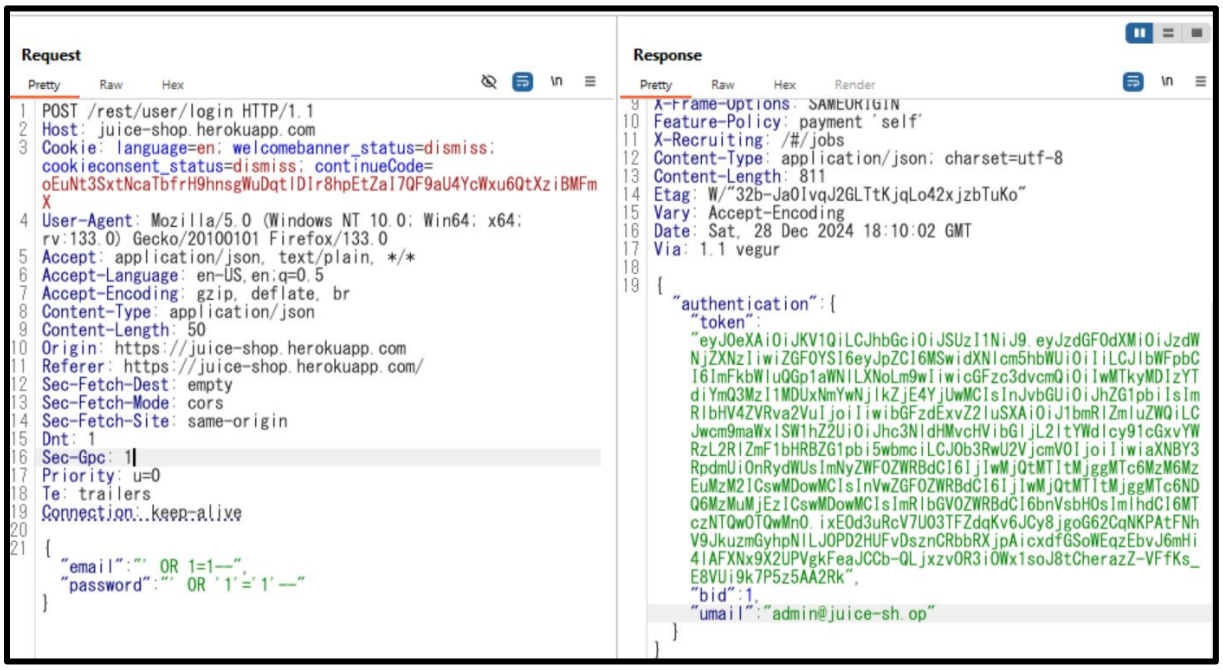
RedOps Crew

Broken Access Control Attack :

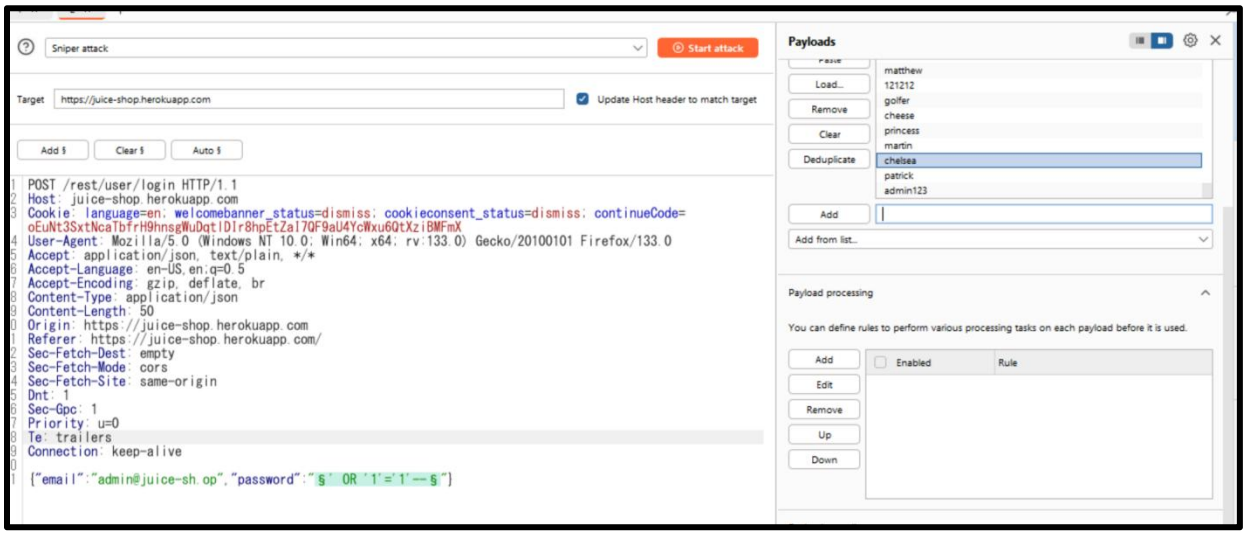
We will enter SQL queries in the email and password fields to exploit the SQL injection vulnerability. After submitting the form, we will capture the request using Burp Suite. We will then send the request to the Repeater tab, forward it, and inspect the server's response. By analyzing the response, we can potentially retrieve the administrator's email. Additionally, since no TLS certificate is being used, the communication between the client and server is unencrypted, allowing us to view sensitive information, such as the email and password, in plain text format. This makes it easier to exploit the vulnerability and gain unauthorized access to the system.

As you can see in the below screenshot server has sent the admins email.

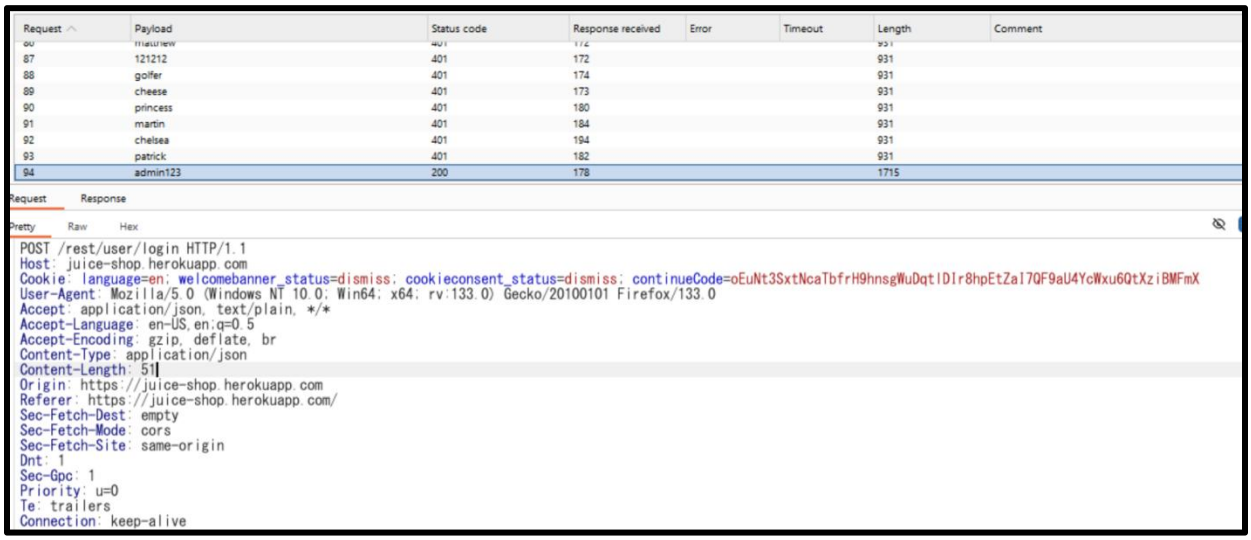
Screenshot:



Now we will replace our email with the sql query and will send this request to intruder to further brute force the admins password to gain unauthorized access to admins account.As you can see below we have added a payload to passwords fields and we have utilized a password list to brute force it.



As we can see below, we have successfully found the password for the admin's account. Our request returned a status code of 200, indicating that the password is correct and the authentication was successful.



RedOps Crew

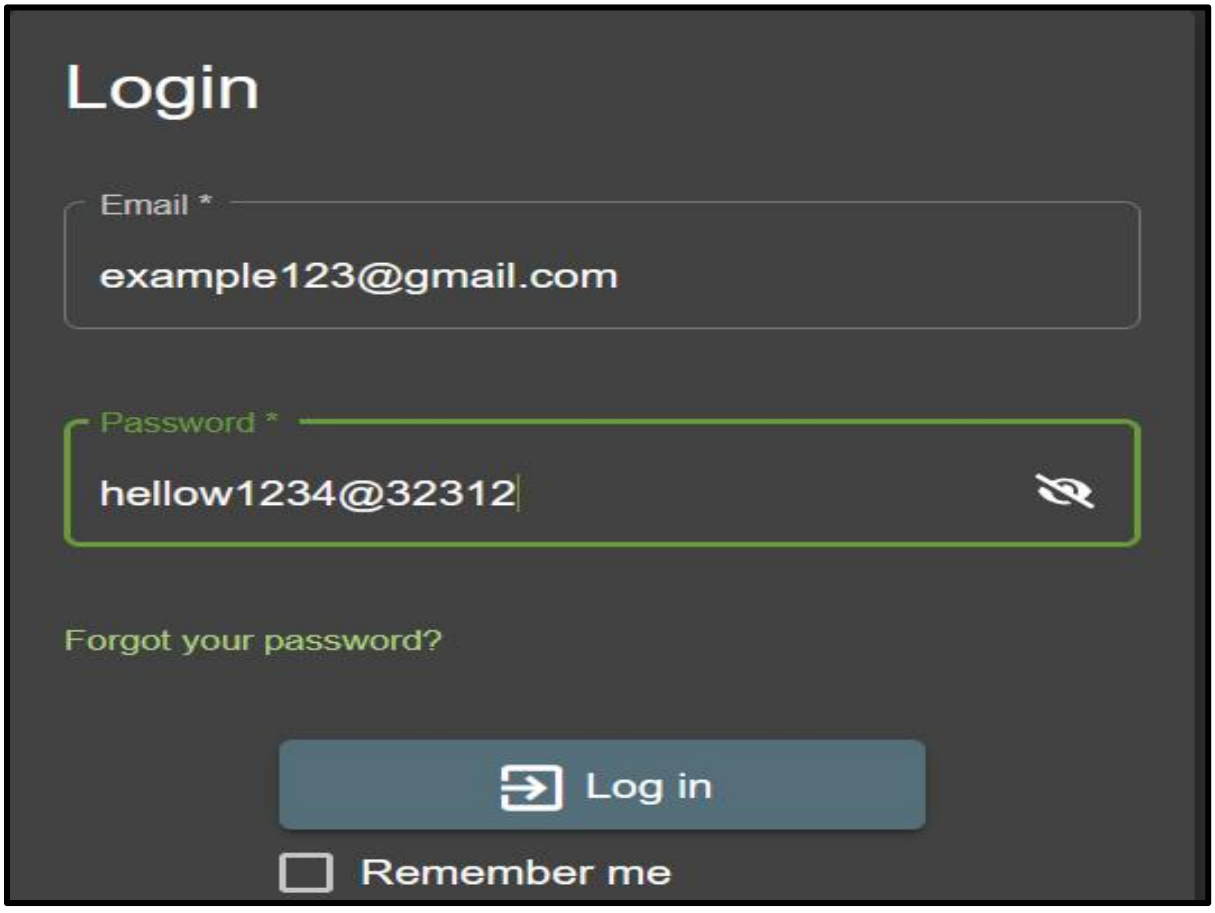
Man in the Middle:

A Man-in-the-Middle (MitM) attack occurs when an attacker secretly intercepts and potentially alters communication between two parties (like a user and a website) without their knowledge. This allows the attacker to steal sensitive data, such as login credentials, or inject malicious content.

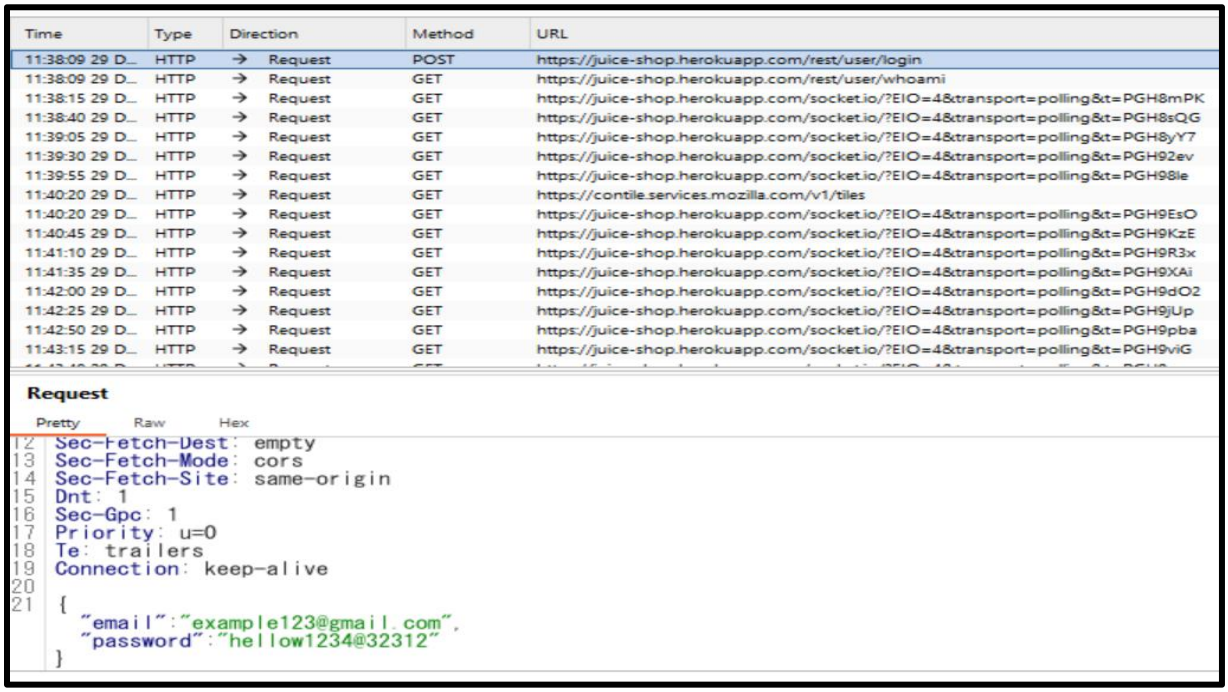
In this attack, we leverage a vulnerability identified by Nikto, where the absence of the Strict-Transport-Security (HSTS) header leaves the server susceptible to insecure communication. The missing HSTS header indicates that the server does not enforce secure HTTPS connections. This oversight allows attackers to conduct Man-in-the-Middle (MITM) attacks and intercept sensitive information, such as login credentials, transmitted over unencrypted HTTP connections.

We have entered credentials in the login from as you can see in the below screenshot.

Screenshot:



We will capture this login request using Burp Suite, allowing us to observe the user's login credentials being transmitted in plaintext without encryption. This vulnerability could result in credential theft, unauthorized access, and misuse of the user's account. As shown in below screenshot.



Time	Type	Direction	Method	URL
11:38:09 29 D...	HTTP	→ Request	POST	https://juice-shop.herokuapp.com/rest/user/login
11:38:09 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/rest/user/whoami
11:38:15 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH8mPK
11:38:40 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH8sQG
11:39:05 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH8yY7
11:39:30 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH92ev
11:39:55 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH98le
11:40:20 29 D...	HTTP	→ Request	GET	https://contile.services.mozilla.com/v1/tiles
11:40:20 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9EsO
11:40:45 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9KzE
11:41:10 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9R3x
11:41:35 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9XAi
11:42:00 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9dO2
11:42:25 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9jUp
11:42:50 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9pba
11:43:15 29 D...	HTTP	→ Request	GET	https://juice-shop.herokuapp.com/socket.io/?EIO=4&transport=polling&t=PGH9viG

Request

Pretty Raw Hex

```
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Dnt: 1
16 Sec-Gpc: 1
17 Priority: u=0
18 Te: trailers
19 Connection: keep-alive
20
21 {
     "email":"example123@gmail.com",
     "password":"hellow1234@32312"
   }
```

RedOps Crew

IDOR:

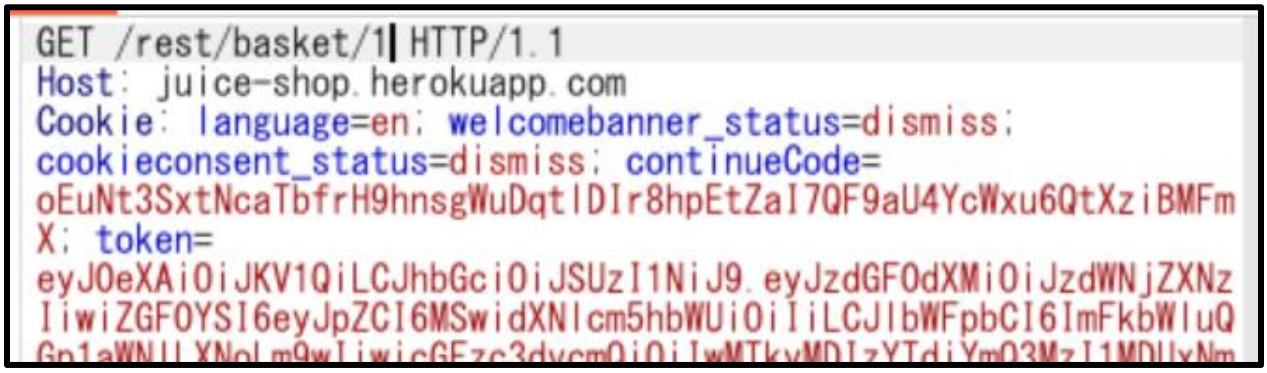
During the security assessment of the application, we identified a vulnerability where a unique identifier (UID) for each user is exposed. This UID is used to associate specific functionalities, such as a shopping basket, with a user. Exploiting this vulnerability can potentially lead to unauthorized access to another user's basket by manipulating the UID.

As you can see our basket only includes these two products.

RedOps Crew

During the assessment, we identified a vulnerability where the application assigns a session ID (e.g., id=1) to each user. This session ID is exposed in the application's requests and is used to identify and manage the user's session.

By modifying this session ID, we successfully demonstrated unauthorized access to another user's account



As you can see, there is an id which is 1 given to a user for a session, we can change this to access someones else account.

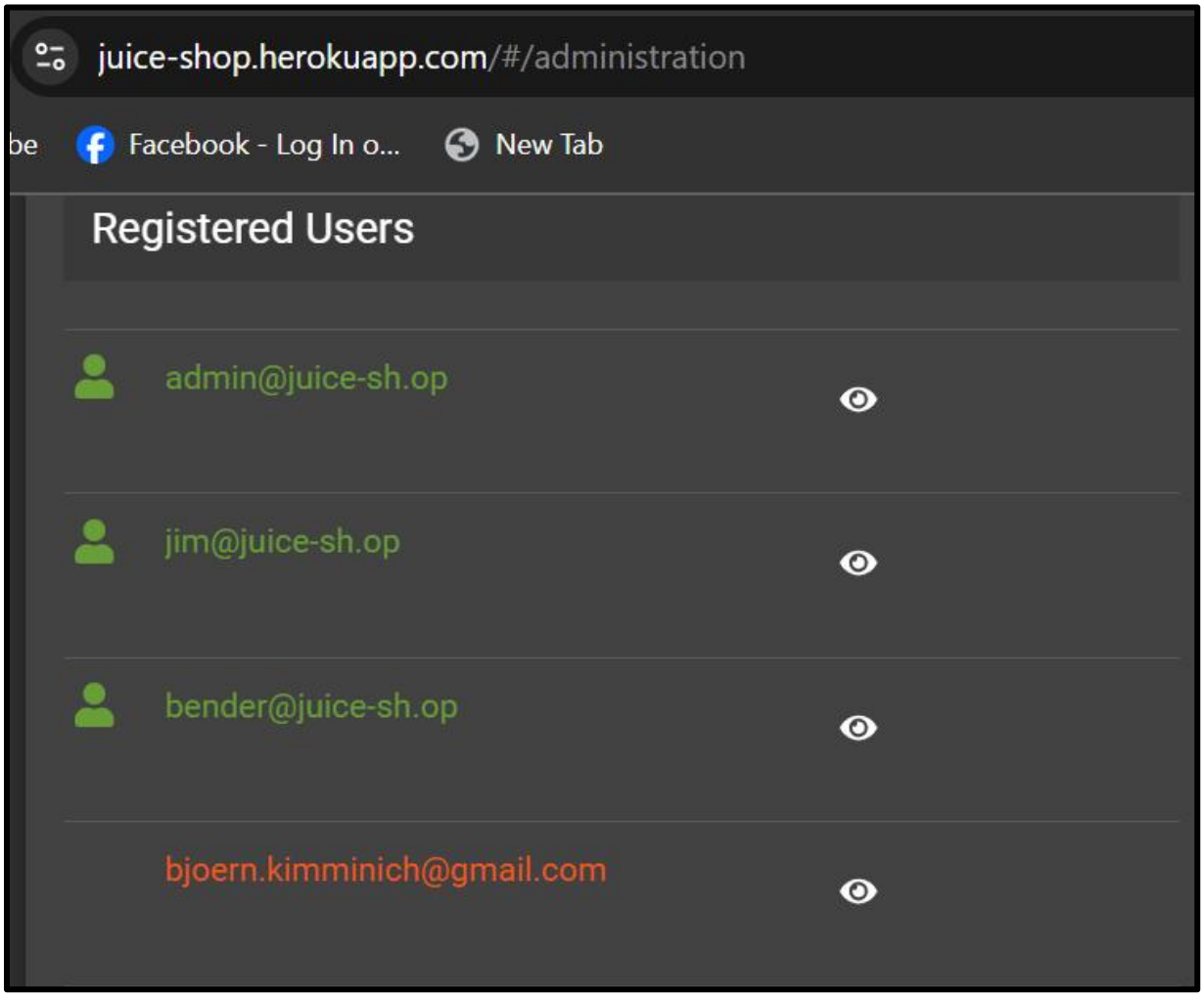
As demonstrated, by exploiting the identified vulnerability, we successfully gained unauthorized access to another user's account. This was confirmed when the products from their basket were displayed in the response to our manipulated request.

Exploiting Forced Browsing to Access Restricted Administrative Pages:

After successfully logging in as an administrator by brute-forcing the credentials, we can leverage forced browsing to navigate directly to the restricted administration page. By altering the URL from https://juice-shop.herokuapp.com/#/search to https://juice-shop.herokuapp.com/#/administration, we can bypass navigation restrictions and potentially gain access to sensitive administrative functionalities and information.

As you can see by doing above we have navigated to the administrations page, which has revealed the emails for all the registered users which could help us further send phishing emails to further get access to unauthorized data.

Screenshot:



Attacking using XSS vulnerability:

The discovery of a Cross-Site Scripting (XSS) vulnerability in the web application presents a significant opportunity to assess its overall security posture. This vulnerability may be exploited to evaluate additional attack vectors and examine potential impacts on application functionality, data integrity, and user confidentiality.

To demonstrate the risk associated with this XSS flaw, we can inject the following script into a search box or any input field that fails to properly sanitize user input:

<iframe src="javascript:alert('xss')">

This script triggers an alert box, confirming the presence of an XSS vulnerability. To escalate the attack, we can utilize the following script to capture session cookies from the victim's browser:

```
<iframe src="javascript:alert(document.cookie)">
```

RedOps Crew

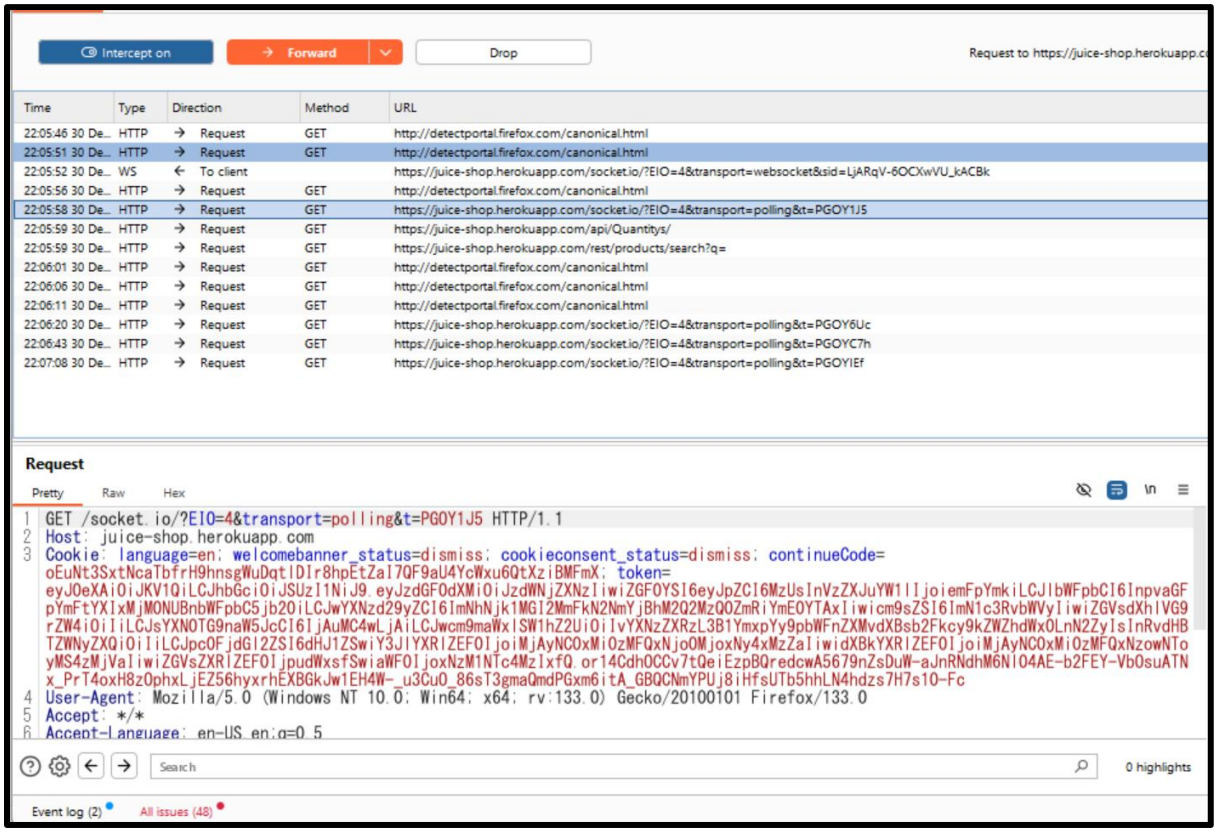
This script retrieves the victim's cookies, which can be captured using a proxy tool like Burp Suite.

Once the cookie request is intercepted, we can forward it to Burp Suite's Repeater, where the captured session cookie is modified to the admin's session cookie.

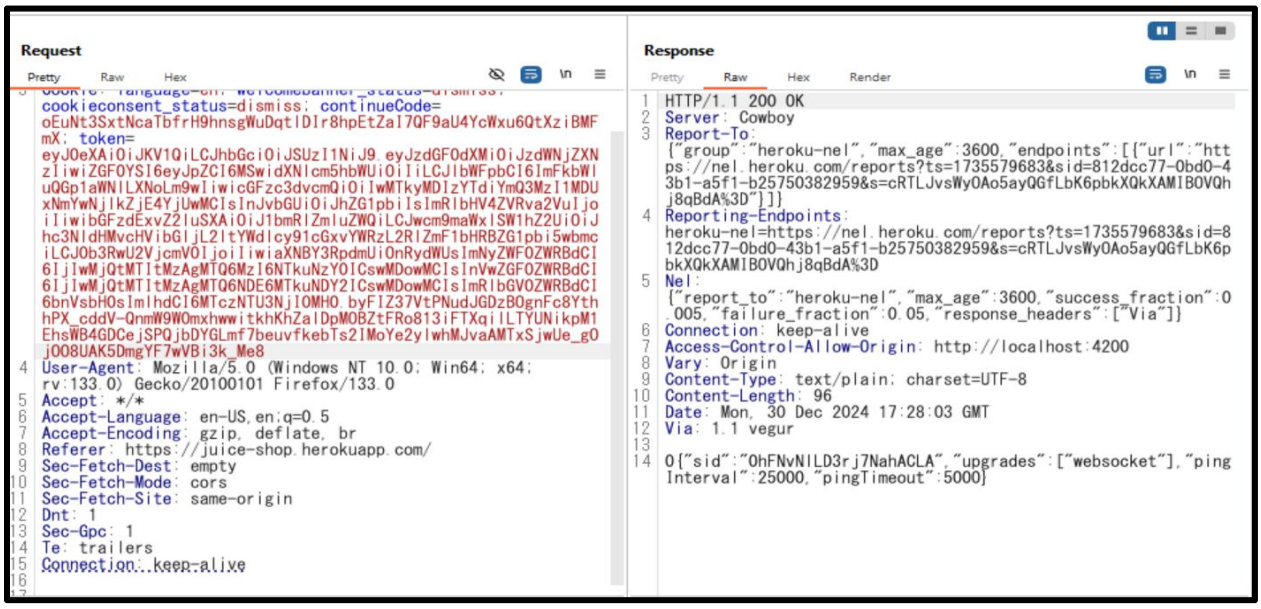
By sending the modified request to the server, the attacker may hijack the admin's session and gain unauthorized access to the administrator's account, compromising the security and confidentiality of the web application.

Screenshot:

As you can see, we have added above script in the search box and have captured the request in the burpsuit as you can see below:



We will now proceed to replace the captured session cookies with those of the admin user in order to gain unauthorized access to the admin account by sending the modified request.



While the server did return a 200 OK response, indicating that the request was processed successfully, the session hijacking did not result in the anticipated outcome, likely due to session binding to the original IP/user-agent, JWT validation, or additional security measures designed to prevent unauthorized access.

CSRF Attack:

In this demonstration, we explore how a specific vulnerability can be leveraged to manipulate user behavior and demonstrate the potential consequences of such exploitation. For instance, by exploiting a known XSS (Cross-Site Scripting) vulnerability, it is possible to craft and inject a malicious script into a web application's input field, such as a search bar. This script could theoretically execute actions like altering account credentials (e.g., changing a user's password) without the user's awareness.

```
<iframe src="javascript:var xmlhttp = new XMLHttpRequest();

        xmlhttp.open('GET', '/rest/user/change-password?new=zaibi22&repeat=zaibi22', true);
        xmlhttp.setRequestHeader('Authorization', `Bearer ${localStorage.getItem('token')}`);
xmlhttp.send();">

</iframe>
```

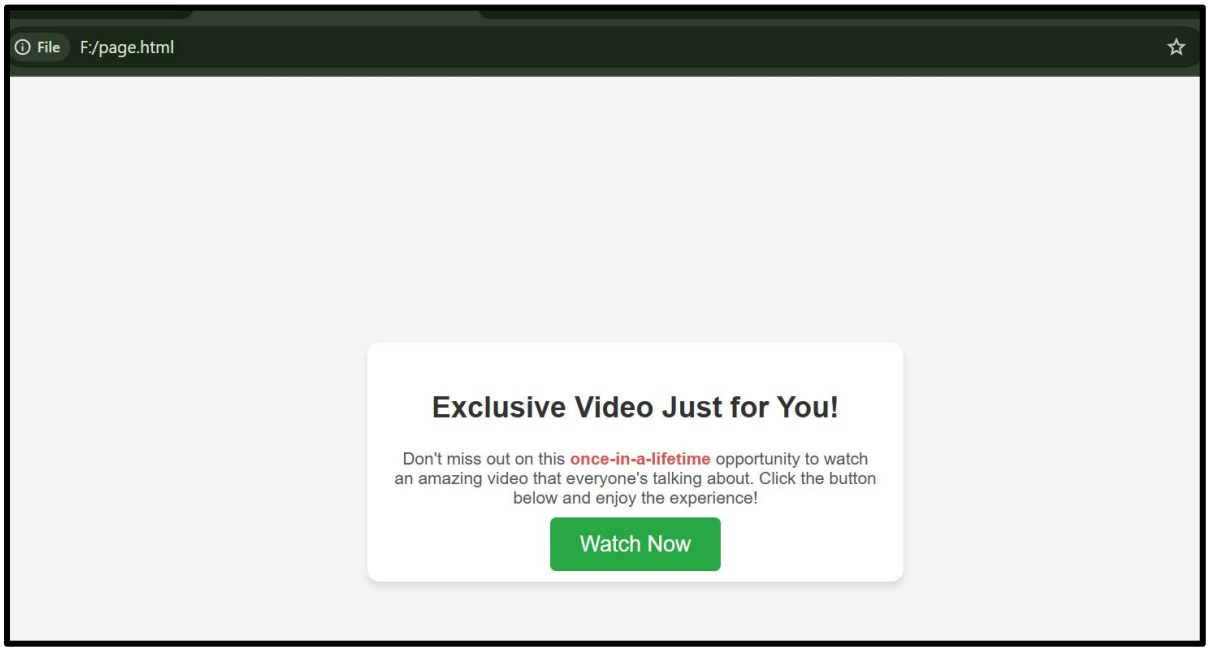
After executing the script we will get a page no results found as the script is being search but it has already changed our password, but the URL of this page will be key to perform CSRF, copy the URL and shorten it with tinyurl.com to make the URL less suspicious.

Now we will change it using tinyurl and we will create a simple HTML webpage to trick the user into clicking the link.

How the CSFR webpage looks like, seems harmless.

Below is the Screenshot of the Page:

Screenshots:



The "Watch Video" button is crafted to appear engaging and trustworthy, enticing users to interact with it. When clicked, instead of performing its intended function (e.g., opening a video), the button executes a hidden script or redirects the user to a page that triggers an unintended action, such as a password change.

Risk Analysis:

We will utilize a Risk Matrix to conduct a comprehensive risk analysis for the identified vulnerabilities. This approach will help assess the potential impact and likelihood of each vulnerability, allowing us to prioritize and address them effectively.

Definitions for the Risk Matrix:

Likelihood:

- Low(1):Rare or difficult to exploit
- Medium(2):Medium chance of exploitation
- High(3):Easily exploitable

Impact:

- 1.Low(1):Minor Impact on system or data.
- Medium(2):Significant but non-critical data.
- High(3):Severe damage or breach of critical systems.

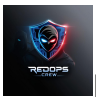
Risk Score:

- Risk = Likelihood X Impact

Risk Levels:

- Low (1–2), Medium (3–4), High (6), Critical (9).

Vulnerability	Likelihood (L)	Impact (I)	Risk Score (L X I)	Risk Level
Broken Authentication	High (3)	High (3)	9	Crictical
Cross-Site Scripting (XSS)	Medium (3)	Medium(2)	4	Medium
Sql Injection	High (3)	High(3)	9	Critical
Broken Access Control	High(3)	High(3)	9	Crictical
IDOR	Medium(2)	High(3)	6	High
CSRF	Medium(2)	Medium(2)	4	Medium



RedOps Crew

Mitigations: Risk Analysis Table

Preliminary step : Employee Training:

Before implementing technical mitigations it is necessary to conduct a comprehensive training sessions for all employee guiding them on following:

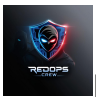
1. Educating them on nature and impact of vulnerabilites.
2. Poviding the best practices for secure coding,system configurations, and access control.
3. Encourage the team to actively spot and address risks in their daily work.

Broken Authentication:

- Enforcing Multi Factor Authentication
- Implement Secure Session cookies.
- Limit login attempts to prevent brute force
- Conduct Regular audits of authentication mechanisms
- Educating Employee on secure session management and strong password policies.
- Add account lockout functionality
- Implement secure password recovery
- Avoid exposing emails and usernames.

Cross-Site Scripting (XSS):

- Train Employee on importance of input validation.
- Sanitize All user inputs (Cleaning up user inputs by removing any potential harmful content).
- Ensure that special characters in user input should be treated as plain text and not as code.
- Deploy a content security policy (prevents malicious scripts from running on your website.
- Regularly Test Applications for XSS vulnerabilites.



RedOps Crew

SQL injection (SQL):

- Train Developers to use parameterized queries and prepared statements.
- Validate and Sanitize all user input data.
- Avoid Dynamic Sql queries contructed with user inputs.
- Use a database firewall to block Malicious Sql Queires.
- Conduct Regular Penetration Testing .

Broken Access Control:

- Conduct workshops on designing and enforcing role-based access controls (RBAC).
- Implement server-side authorization checks for every request.
- Regularly audit access control rules to ensure proper implementation.
- Test for access control bypass techniques during application updates.
- Use logging and monitoring to detect unauthorized access attempts.

Insecure Direct Object Reference (IDOR)

- Train developers to use indirect references (e.g., hashed or tokenized IDs).
- Validate user permissions for every object access request.
- Avoid exposing sensitive identifiers (e.g., database keys) in URLs.
- Regularly review code for potential IDOR vulnerabilities.

Cross-Site Request Forgery (CSRF)

- Educate teams on the use of CSRF tokens in forms and APIs.
- Enforce SameSite attributes on cookies to restrict cross-origin requests.
- Validate Referer or Origin headers for sensitive actions.
- Implement re-authentication for high-risk actions (e.g., password changes).
- Test web applications regularly for CSRF vulnerabilities.