Juice Shop
Security Assessment Findings Report

Date: Oct. 18th, 2024
Project: Juice Shop
Version 3.0

Table of Contents

Confidentiality Statement

This document is the exclusive property of Juice Shop and DEPIX Security (DEPIX). This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Juice Shop and DEPIX.

Juice Shop may share this document with auditors under non-disclosure agreements to demonstrate compliance with penetration test requirements.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment, not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. DEPIX prioritized the assessment to identify the weakest security controls an attacker would exploit. DEPIX recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

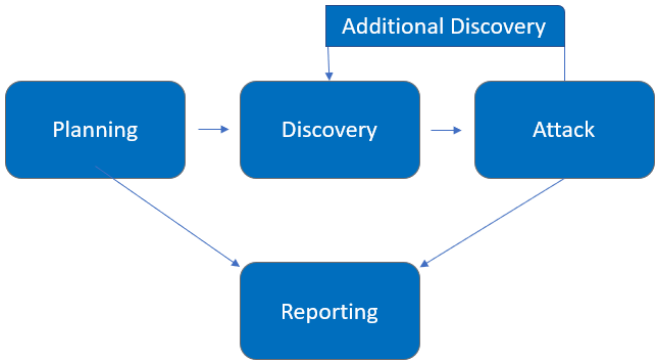
Name	Title	Contact Information
Juice shop		
Amir Wadie	Owner	Email: amirwidi@gmail.com
DEPIX Security		
Amir Wadie	Penetration Tester	Email: amirwidi@gmail.com

Assessment Overview

From Oct 1st, 2024 to Oct 18th, 2024, Juice Shop engaged DEPIX to evaluate the security posture of its store compared to current industry best practices that included an web application penetration testing.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement are obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

Web App. Penetration Test

A Web App. Penetration Testing (WAPT) with a black box approach to emulate an external attacker from outside with only a store link is known like any normal potential customer - but inside sandbox env. (Docker containers) .

Finding Severity Ratings

The following table defines the severity levels of and corresponding CVSS score ranges used throughout the document to assess vulnerability and risk impact calculated from this standard.

Severity	CVSS V3.1 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, the attacker's skill level, and the client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
container of website URL	*.localhost:1546

Scope Exclusions

Per the client's request, DEPIX did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Juice Shop.

Client Allowances

Juice Shop provided DEPIX with the following allowances:

- A Docker container was similar to the production environment. Technical Findings

Executive Summary

TCMS evaluated Demo Corp's internal security posture through penetration testing from February 22nd, 2021 to March 5th, 2021. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

Time limitations were in place for testing. Internal network penetration testing was permitted for ten
10. business days.

Testing Summary

The network assessment evaluated Demo Corp's internal network security posture. From an internal perspective, the TCMS team performed vulnerability scanning against all IPs provided by Demo Corp to evaluate the overall patching health of the network. The team also performed common Active Directory based attacks, such as Link-Local Multicast Name Resolution (LLMNR) Poisoning, SMB relaying, IPv6 man-in-the-middle relaying, and Kerberoasting. Beyond vulnerability scanning and Active Directory attacks, the TCMS evaluated other potential risks, such as open file shares, default credentials on servers/devices, and sensitive information disclosure to gain a complete picture of the network's security posture.

The TCMS team discovered that LLMNR was enabled in the network (Finding IPT-001), which permitted the interception of user hashes via LLMNR poisoning. These hashes were taken offline and cracked via dictionary attacks, which signals a weak password policy (Finding IPT-005). Utilizing the cracked passwords, the TCMS team gained access to several machines within the network, which indicates overly permissive user accounts.

With machine access, and the use of older operating systems in the network (Finding IPT-009), the team was able to leverage WDigest (Finding IPT-003) to recover cleartext credentials to accounts. The team was also able to dump local account hashes on each machine accessed. The TCMS team discovered that the local account hashes were being re-used across devices (Finding IPT-002), which lead to additional machine access through pass-the-hash attacks.

Ultimately, the TCMS team was able to leverage accounts captured through WDigest and hash dumps to move laterally throughout the network until landing on a machine that had a Domain Administrator credential in cleartext via WDigest. The testing team was able to use this credential to log into the domain controller and compromise the entire domain. For a full walkthrough of the path to Domain Admin, please see Finding IPT-025.

In addition to the compromise listed above, the TCMS team found that users could be impersonated through delegation attacks (Finding IPT-004), SMB relay attacks were possible due to SMB signing being disabled (Finding IPT-007), and IPv6 traffic was not restricted, which could lead to LDAPS relaying and domain compromise (Finding IPT-006).

The remainder of critical findings relate to patch management as devices with critical out-of-date software (Finding IPT-008), operating systems (Finding IPT-009), and Microsoft RCE vulnerabilities (Findings IPT-010, IPT-011, IPT-012, IPT-013), were found to be present within the network.

The remainder of the findings were high, moderate, low, or informational. For further information on findings, please review the Technical Findings section.

Tester Notes and Recommendations

Testing results of the Demo Corp network are indicative of an organization undergoing its first penetration test, which is the case here. Many of the findings discovered are vulnerabilities within Active Directory that come enabled by default, such as LLMNR, IPv6, and Kerberoasting.

During testing, two constants stood out: a weak password policy and weak patching. The weak password policy led to the initial compromise of accounts and is usually one of the first footholds an attacker attempts to use in a network. The presence of a weak password policy is backed up by the evidence of our testing team cracking over 2,200 user account passwords, including a majority of the Domain Administrator accounts, through basic dictionary attacks.

We recommended that Demo Corp re-evaluates their current password policy and considers a policy of 15 characters or more for their regular user accounts and 30 characters or more for their Domain Administrator accounts. We also recommend that Demo Corp explore password blacklisting and will be supplying a list of cracked user passwords for the team to evaluate. Finally, a Privilege Access Management solution should be considered.

Weak patching and dated operating systems led to the compromise of dozens of machines within the network. We believe the number of compromised machines would have been

significantly larger, however the TCMS and Demo Corp teams agreed it was not necessary to attempt to exploit any remote code execution (RCE) based vulnerabilities, such as MS17-010 (Finding IPT-012), as the domain controller had already been compromised and the teams did not want to risk any denial of service through failed attacks.

We recommend that the Demo Corp team review the patching recommendations made in the Technical Findings section of the report along with reviewing the provided Nessus scans for a full overview of items to be patched. We also recommend that Demo Corp improve their patch management policies and procedures to help prevent potential attacks within their network.

On a positive note, our testing team triggered several alerts during the engagement. The Demo Corp Security Operations team discovered our vulnerability scanning and was alerted when we attempted to use noisy attacks on a compromised machine. While not all attacks were discovered during testing, these alerts are a positive start. Additional guidance on alerting and detection has been provided for findings, when necessary, in the Technical Findings section.

Overall, the Demo Corp network performed as expected for a first-time penetration test. We recommend that the Demo Corp team thoroughly review the recommendations made in this report, patch the findings, and re-test annually to improve their overall internal security posture.

Key Strengths and Weaknesses

The following identifies the key strengths identified during the assessment:

1. Observed some scanning of common enumeration tools (Nessus)
2. Mimikatz detected on some machines
3. Service accounts were not running as domain administrators
4. Demo Corp local administrator account password was unique to each device The following identifies the key weaknesses identified during the assessment:

1. Password policy found to be insufficient
2. Critically out-of-date operating systems and weak patching exist within the network
3. Passwords were observed in cleartext due to WDigest
4. LLMNR is enabled within the network
5. SMB signing is disabled on all non-server devices in the work
6. IPv6 is improperly managed within the network
7. User accounts can be impersonated through token delegation
8. Local admin accounts had password re-use and were overly permissive
9. Default credentials were discovered on critical infrastructure, such as iDRACs
10. Unauthenticated share access was permitted

11. User accounts were found to be running as service accounts
12. Service accounts utilized weak passwords
13. Domain administrator utilized weak passwords

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Web App. Penetration Test Findings

13	5	6	0	1
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Internal Penetration Test		
IPT-001: Insufficient LLMNR Configuration	Critical	Disable multicast name resolution via GPO.
IPT-002: Security Misconfiguration – Local Admin Password Reuse	Critical	Utilize unique local admin passwords and limit local admin users via least privilege.
IPT-003: Security Misconfiguration – Wdigest	Critical	Disable WDigest via GPO.
IPT-004: Insufficient Hardening – Token Impersonation	Critical	Restrict token delegation.

IPT-005: Insufficient Password Complexity	Critical	Implement CIS Benchmark password requirements / PAM solution.
IPT-006: Security Misconfiguration – IPv6	Critical	Restrict DHCPv6 traffic and incoming router advertisements in Windows Firewall via GPO.
IPT-007: Insufficient Hardening – SMB Signing Disabled	Critical	Enable SMB signing on all Demo Corp domain computers.
IPT-008: Insufficient Patch Management – Software	Critical	Update to the latest software version.
IPT-009: Insufficient Patch Management – Operating Systems	Critical	Update Operating Systems to the latest version.
IPT-010: Insufficient Patching – MS08-067 - ECLIPSEDWING/NETAPI	Critical	Apply the appropriate Microsoft patches to remediate the issue.
IPT-011: Insufficient Patching – MS12-020 – Remote Desktop RCE	Critical	Apply the appropriate Microsoft patches to remediate the issue.
IPT-012: Insufficient Patching – MS17-010 - EternalBlue	Critical	Apply the appropriate Microsoft patches to remediate the issue.
IPT-013: Insufficient Patching – CVE- 2019-0708 - BlueKeep	Critical	Apply the appropriate Microsoft patches to remediate the issue.

Finding	Severity	Recommendation
IPT-014: Insufficient Privileged Account Management – Kerberoasting	High	Use Group Managed Service Accounts (GMSA) for privileged services.
IPT-015: Security Misconfiguration – GPP Credentials	High	Apply vendor patching. Do not use GPP cpasswords.
IPT-016: Insufficient Authentication - VNC	High	Enable authentication on the VNC Server.
IPT-017: Default Credentials on Web Services	High	Change default credentials or disable unused accounts.
IPT-018: Insufficient Hardening – Listable Directories	High	Restrict access and conduct web app assessment.
IPT-019: Unauthenticated SMB Share Access	Moderate	Disable SMB share or require authentication.
IPT-020: Insufficient Patch Management – SMBv1	Moderate	Upgrade to SMBv3 and apply latest patching.
IPT-021: IPMI Hash Disclosure	Moderate	Disable IPMI over LAN if it is not needed.
IPT-022: Insufficient SNMP Community String Complexity	Moderate	Disabled SNMP if not required.
IPT-023: Insufficient Data in Transit Encryption - Telnet	Moderate	Migrate to TLS protected protocols.

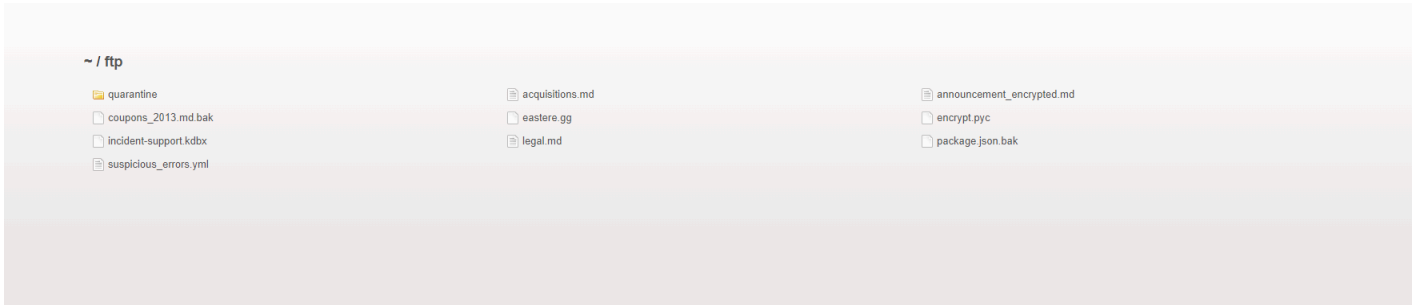
IPT-024: Insufficient Terminal Services Configuration	Moderate	Enable Network Level Authentication (NLA) on the remote RDP server.
IPT-025: Steps to Domain Admin	Informational	Review action and remediation steps.

Web App Pentesting Findings (WAPT)

Finding WAPT-001: Sensitive Directory Exposed via robots.txt (Moderate)

Description:	The `robots.txt` file is used to control which parts of a website should not be indexed by search engines. However, it is publicly accessible and can reveal sensitive directories to attackers. In this case, the `/ftp` directory was exposed, potentially containing sensitive files or data. Attackers may use this information to explore restricted areas of the web server, increasing the risk of unauthorized access or information disclosure. While `robots.txt` is intended for search engines, malicious users can exploit it to target vulnerable directories.
System:	FTP folder on the server
Tools Used:	Browser
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N score 5.3

Evidence



Steps to Reproduce:

1. 1. Navigate to the web application's robots.txt file by accessing http://localhost:1546/robots.txt

   Identify the following directive:
   User-agent: *

   Disallow: /ftp

2. Manually attempt to access the /ftp directory at http://localhost:1546/ftp
3. check for sensitive files or directories listed inside.

Remediation

Remove sensitive directories from the robots.txt file and implement proper access controls on directories containing sensitive data.

Use a more secure method to protect sensitive directories, such as authentication or IP whitelisting.

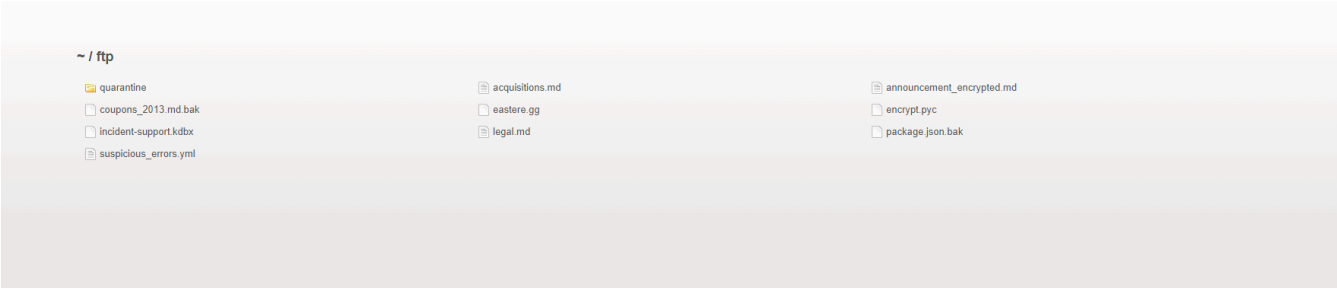
Consider serving sensitive files via a private or internal network if required.

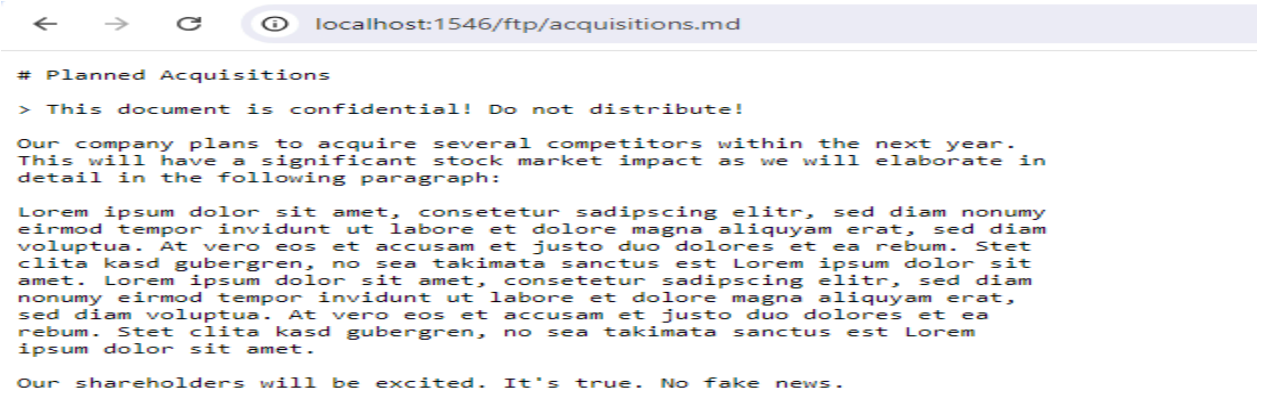
Regularly review the contents of robots.txt and ensure no critical directories are listed.

Finding WAPT-002: Sensitive information exposure (Moderate)

Description:	After opening /FTP directory we found several files there is a confidential files that directly leak company sensitive data.
System:	FTP folder on the server
Tools Used:	Browser
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N score 5.3

Evidence





Steps to Reproduce:

1. open ftp folder
2. open acquisitions.md file and you got it

Remediation

Remove confidential files and implement proper access controls on directories containing sensitive data.

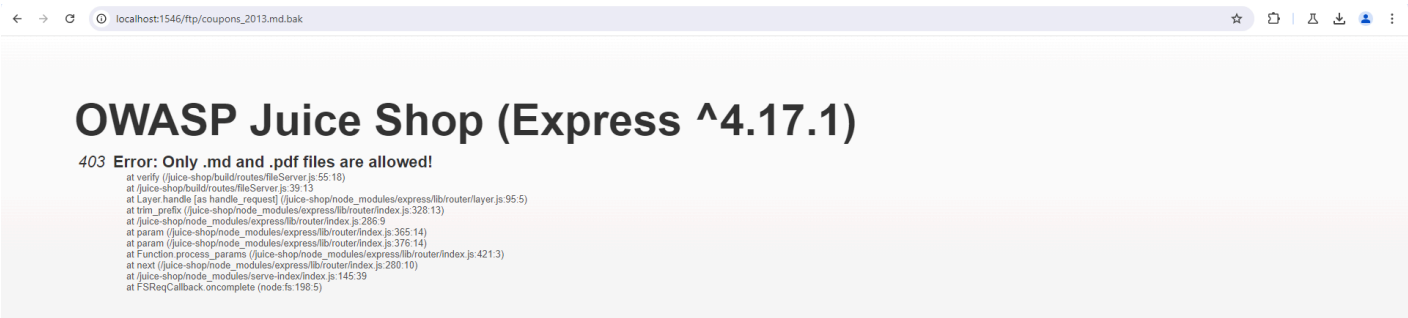
Use a more secure method to protect sensitive directories, such as authentication or IP whitelisting.

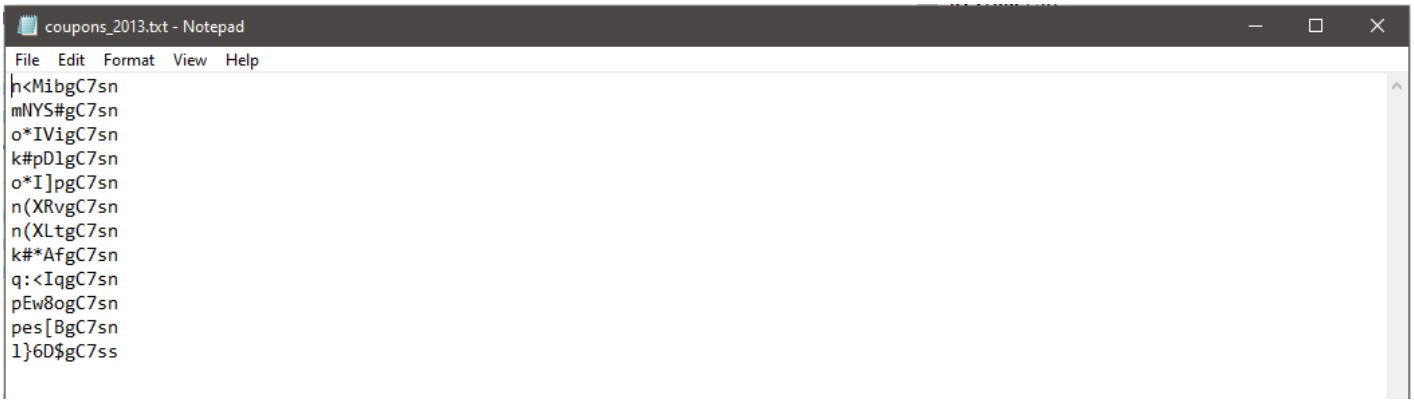
Consider serving sensitive files via a private or internal network if required.

Finding WAPT-003: Leaked discount coupons (Moderate)

Description:	after opening /FTP directory we found several files there is a confidential files that directly leak company sensitive data, one of them is “coupons_2013.md.bak” but when we try to open it we got error 403. we bypassed this by adding “%2500.md” to file name to bypass it using double URL encoding.
System:	check-out process
Tools Used:	Browser
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N score 5.3

Evidence





Steps to Reproduce:

1. open ftp folder
2. open coupons_2013.md.bak
3. add this payload to the end of url “%2500.md”
4. the file is downloaded into your pc
5. change the name of file to “coupons_2013.txt”
6. open the file

Remediation

Normalize and validate all URL-encoded characters and strings before processing them, ensuring that characters like null bytes or special encodings are properly sanitized.

Remove confidential files and implement proper access controls on directories containing sensitive data.

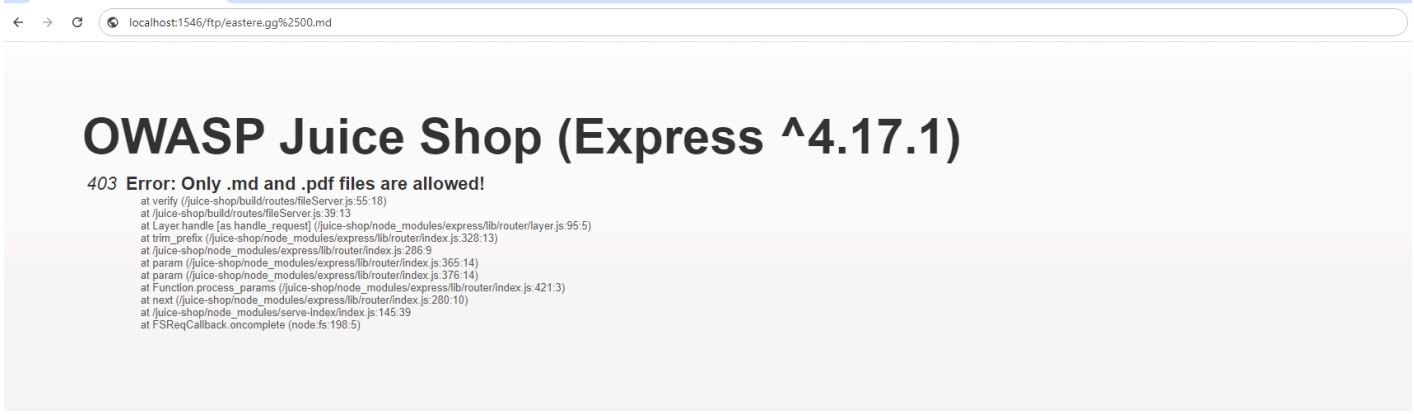
Use a more secure method to protect sensitive directories, such as authentication or IP whitelisting.

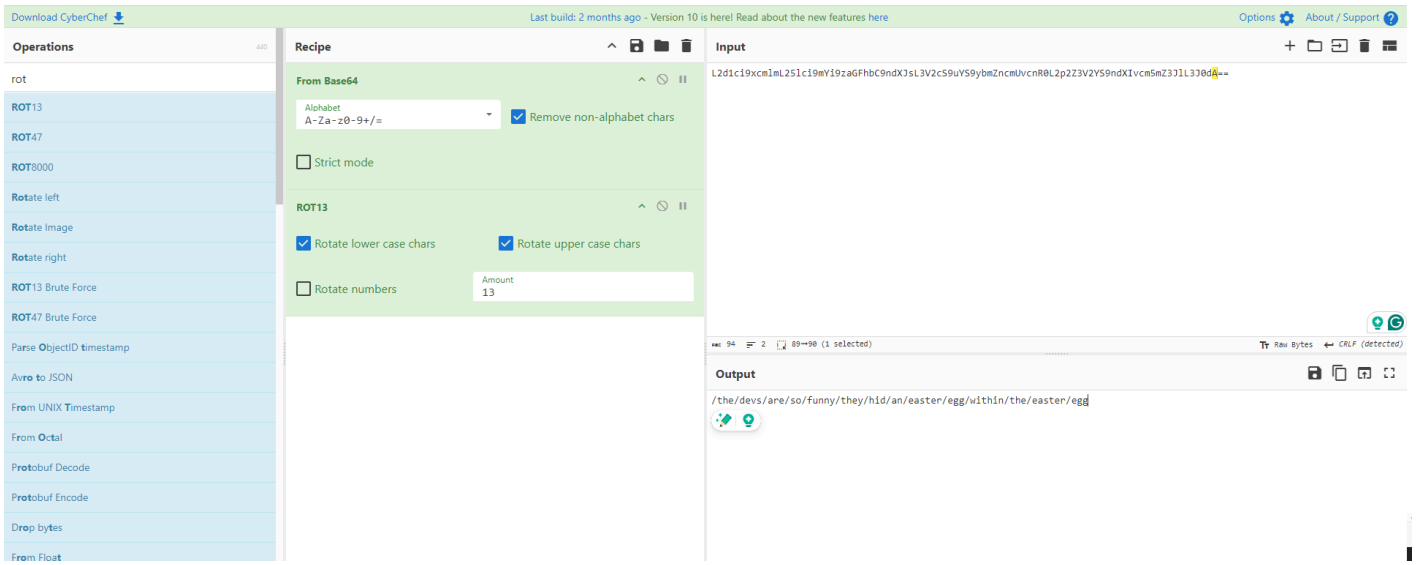
Consider serving sensitive files via a private or internal network if required.

Finding WAPT-004: cryptographic failure (info.)

Description:	after opening /FTP directory we found several files there is a confidential files that directly leak company sensitive data, one of them is “eastere.gg” but when we try to open it we got error 403. we bypassed this by adding “%2500.md” to file name to bypass it using double URL encoding.
System:	Nothing
Tools Used:	Browser
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N score 0.0

Evidence





Steps to Reproduce:

1. open ftp folder
2. open eastere.gg
3. add this payload to the end of URL “%2500.md”
4. the file is downloaded into your pc
5. change the name of the file to “eastere.txt”
6. open the file
7. we found that we have base64 encoded text
8. using CyberChef tool we decoded it
9. We seem to encrypted text after some investigation we found that it was Rot13 cipher
10. after deciphering it we got a secret directory on the site
11. but we didn’t find any sensitive info.

    you can use direct CyberChef recipe from here

Remediation

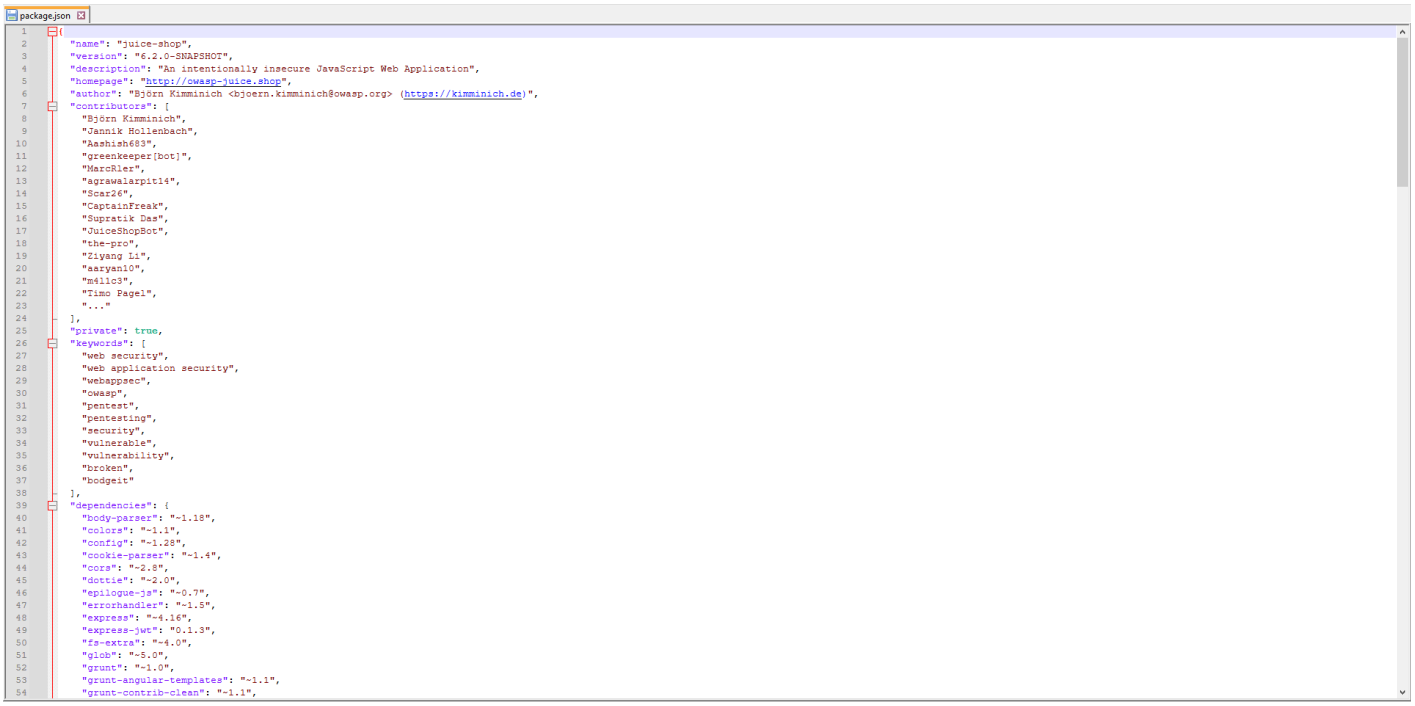
Normalize and validate all URL-encoded characters and strings before processing them, ensuring that characters like null bytes or special encodings are properly sanitized.

Consider serving sensitive files via a private or internal network if required.

Finding WAPT-005: sensitive file exposure (info.)

Description:	after opening /FTP directory we found several files there is a confidential files that directly leak company sensitive data, one of them is “package.json.bak” but when we try to open it we got error 403. we bypassed this by adding “%2500.md” to file name to bypass it using double URL encoding.
System:	Nothing
Tools Used:	Browser
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N score 0.0

Evidence



Steps to Reproduce:

1. open ftp folder
2. and open package.json.bak
3. add this payload to the end of url “%2500.md”
4. the file is downloaded into your pc
5. change the name of the file to “package.json”
6. open the file
7. we found file contain package versions which can lead to supply chain attacks.
8. also found a list of developers' names that may lead to social engineering attacks.

Remediation

Normalize and validate all URL-encoded characters and strings before processing them, ensuring that characters like null bytes or special encodings are properly sanitized.

Consider serving sensitive files via a private or internal network if required.

Finding WAPT-006: leaked pages in source code (info.)

Description:	leaked sensitive directories from JS code
System:	Nothing
Tools Used:	Browser - python
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N score 0.0

Steps to Reproduce:

1. open site source code
2. open main.js
3. take the source code into Txt file called “input.txt”
4. run this custom code to extract the hidden directory.
5. found path called “/score-board”

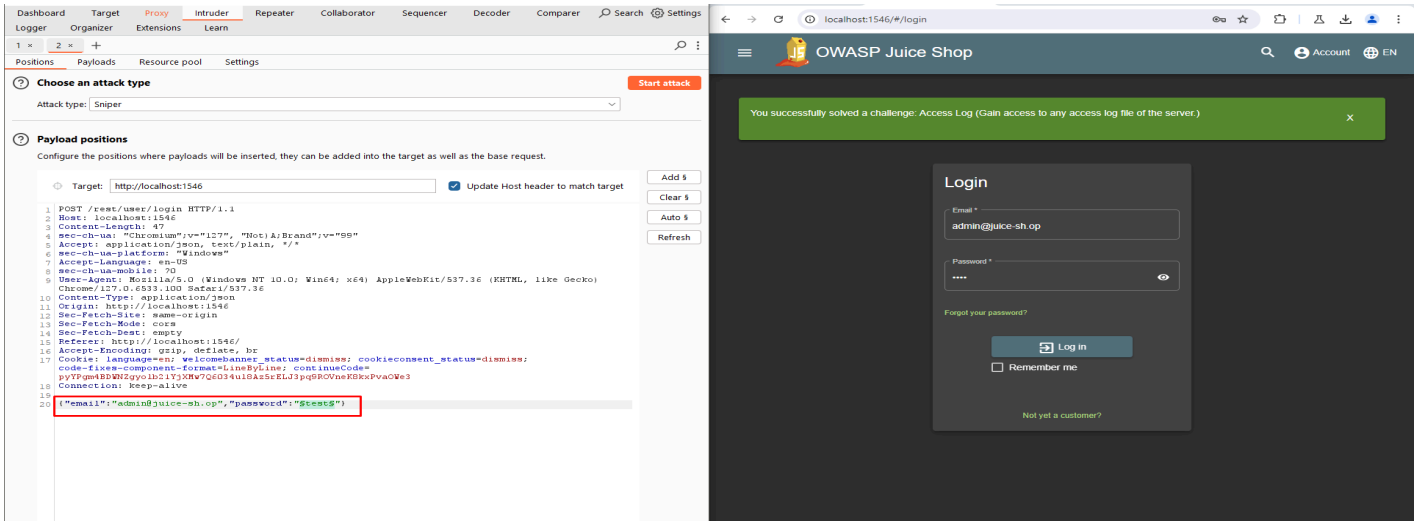
Remediation

make sure you don’t make important dir. without proper authentication

Finding WAPT-007: weak admin password (Critical)

Description:	after traversal in site, we got an admin email in one of the product comments by trying to brute force the email password.
System:	Nothing
Tools Used:	Browser - burp suite
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:H score 9.4

Evidence



Steps to Reproduce:

1. send a login request to the burp intruder
2. with sniper configuration and rockyou wordlist.
3. we got the password “admin123”

Remediation

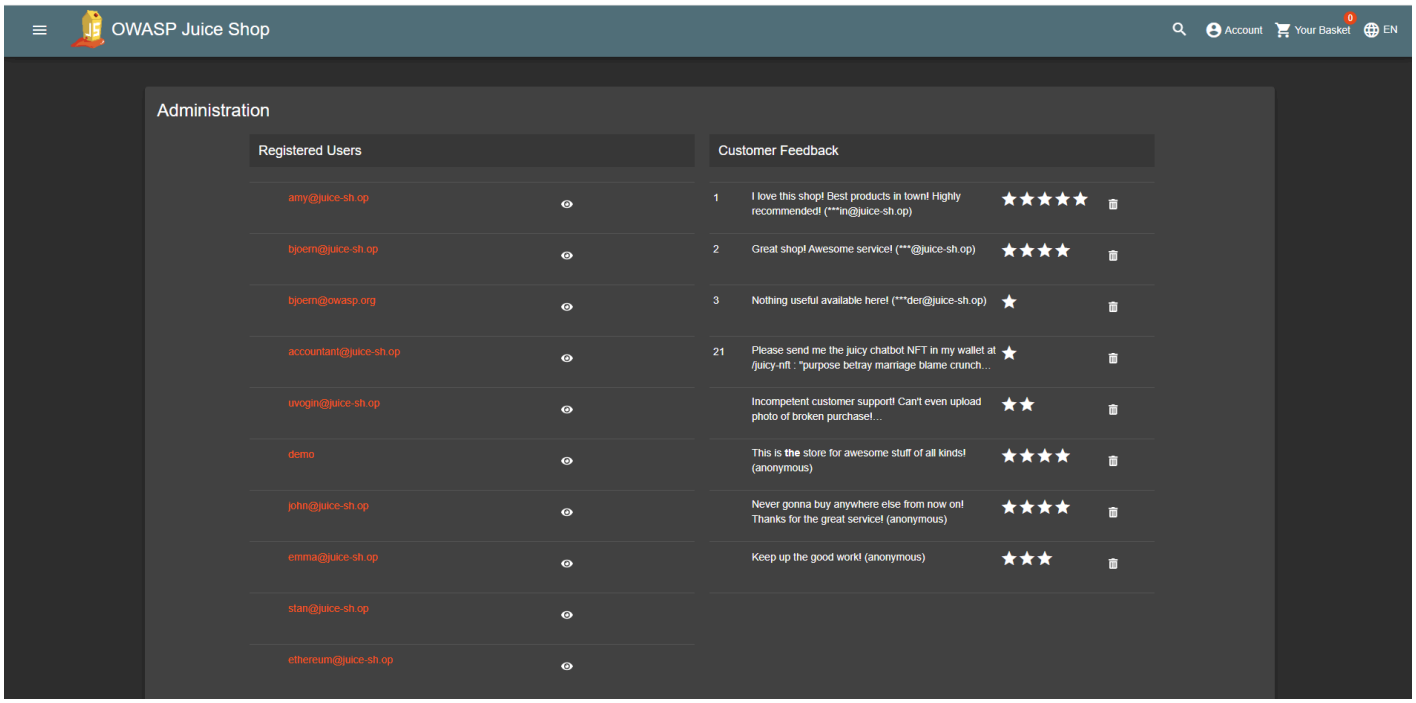
make a strong password policy inside the organization to enforce users to make strong password hard to crack.

Implement account lockout or CAPTCHA after multiple failed login attempts and rate limit to prevent brute-force attacks.

Finding WAPT-008: access admin panel (critical.)

Description:	from the path extraction process at **WAPT-006,** we found the admin panel directory
System:	admin panel
Tools Used:	Browser
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:H score 9.1

Evidence



Steps to Reproduce:

1. login into admin account using same vulnerability WAPT-007
2. go to admin panel

Remediation

make a strong password policy inside the organization to enforce users to make strong password hard to crack.

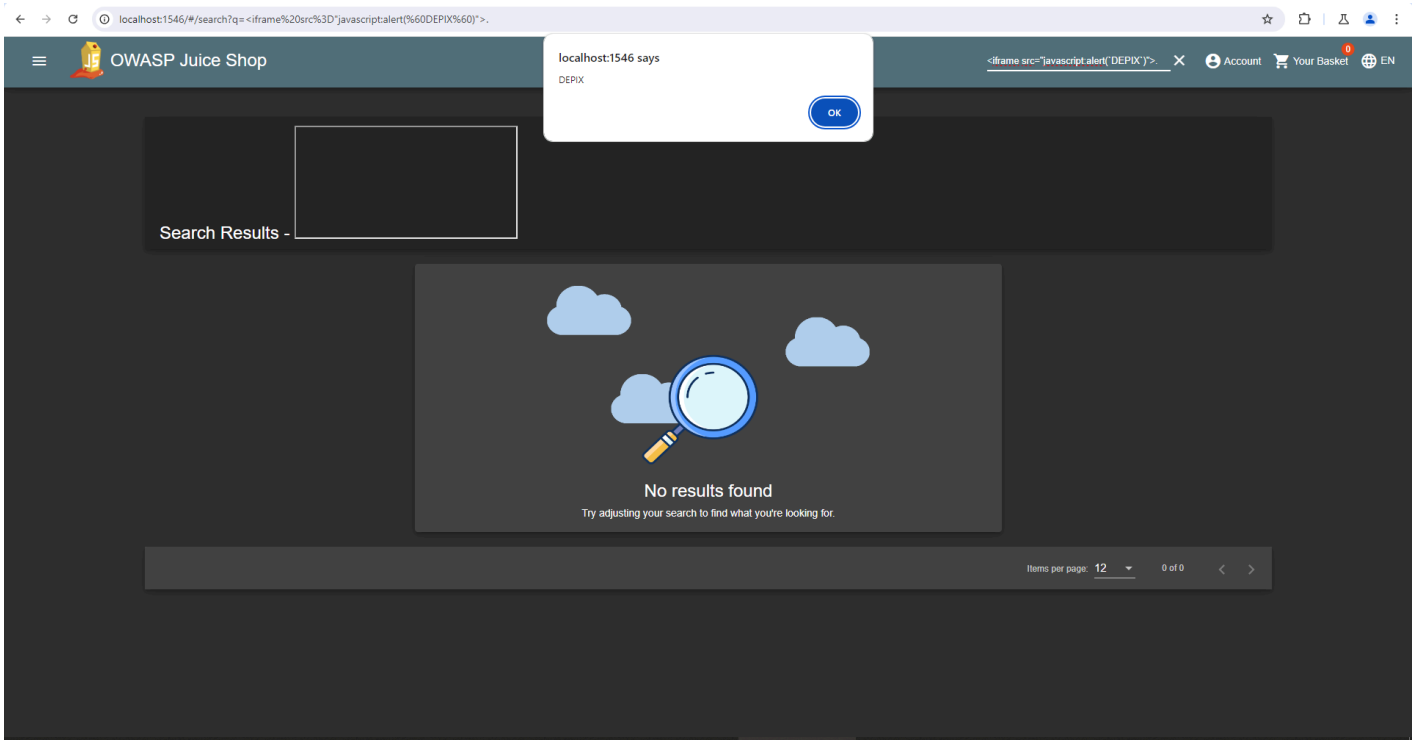
MFA for admin access. This adds an extra layer of security by requiring users to provide a second form of verification, like a one-time code from an app or SMS.

Restrict access to the admin panel to specific IP addresses using IP whitelisting, VPNs, or firewalls. This ensures that only trusted devices can reach the admin page.

Finding WAPT-009: DOM XSS (Moderate.)

Description:	in search bar craft a payload to trigger XSS
System:	search bar
Tools Used:	Browser
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N score 4.3

Evidence



Steps to Reproduce:

1. crafting a payload “<iframe src="javascript:alert(`DEPIX`)">” and put it into search bar

Remediation

Validate input on the server-side: Ensure that all user inputs are validated for expected types, formats, and lengths. Reject any input that does not conform to the defined criteria.

Sanitize input: Remove or escape any potentially malicious characters from user input before processing or storing it. Common characters to sanitize include <, >, &, ', and ".

HTML encoding: Encode user input when displaying it on a webpage to prevent it from being interpreted as HTML or JavaScript. For example, replace < with <, > with >, and & with &.

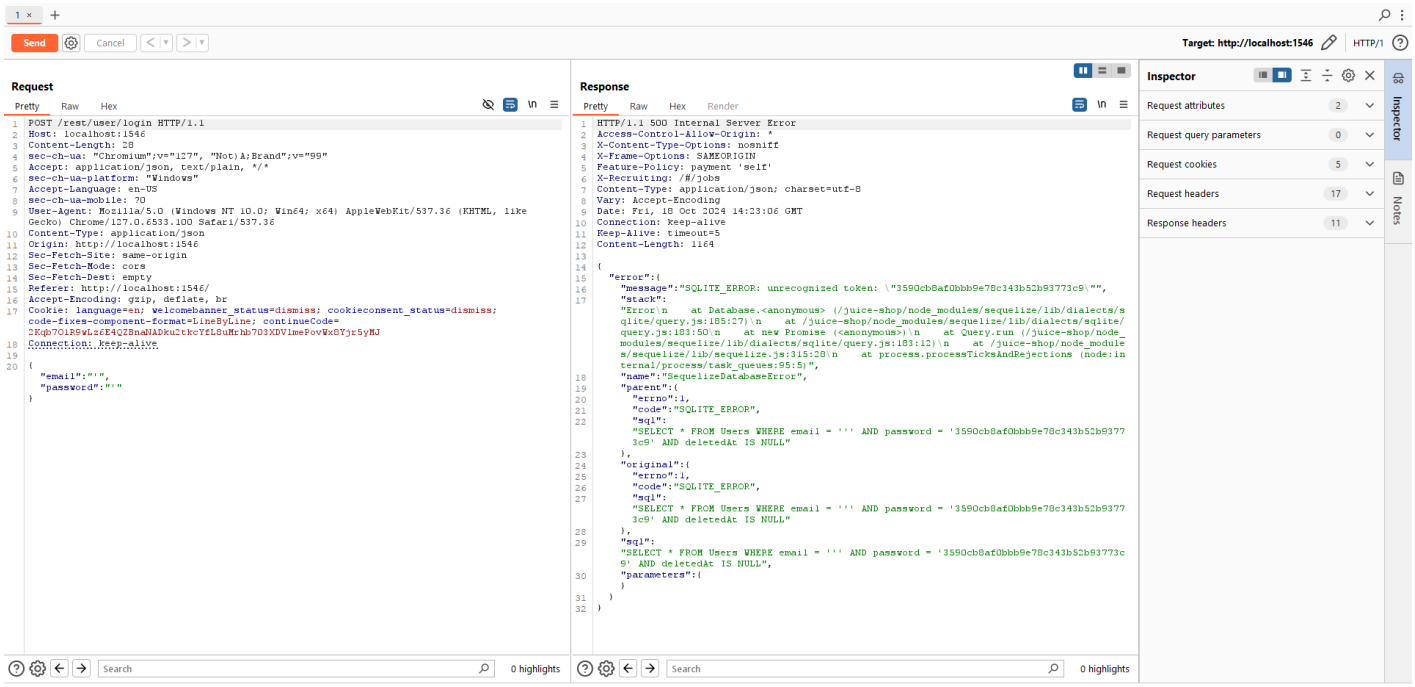
JavaScript encoding: When embedding user input into JavaScript, ensure it is properly encoded to prevent it from being executed as code. For example, escape quotes and other special characters that could break out of the script context.

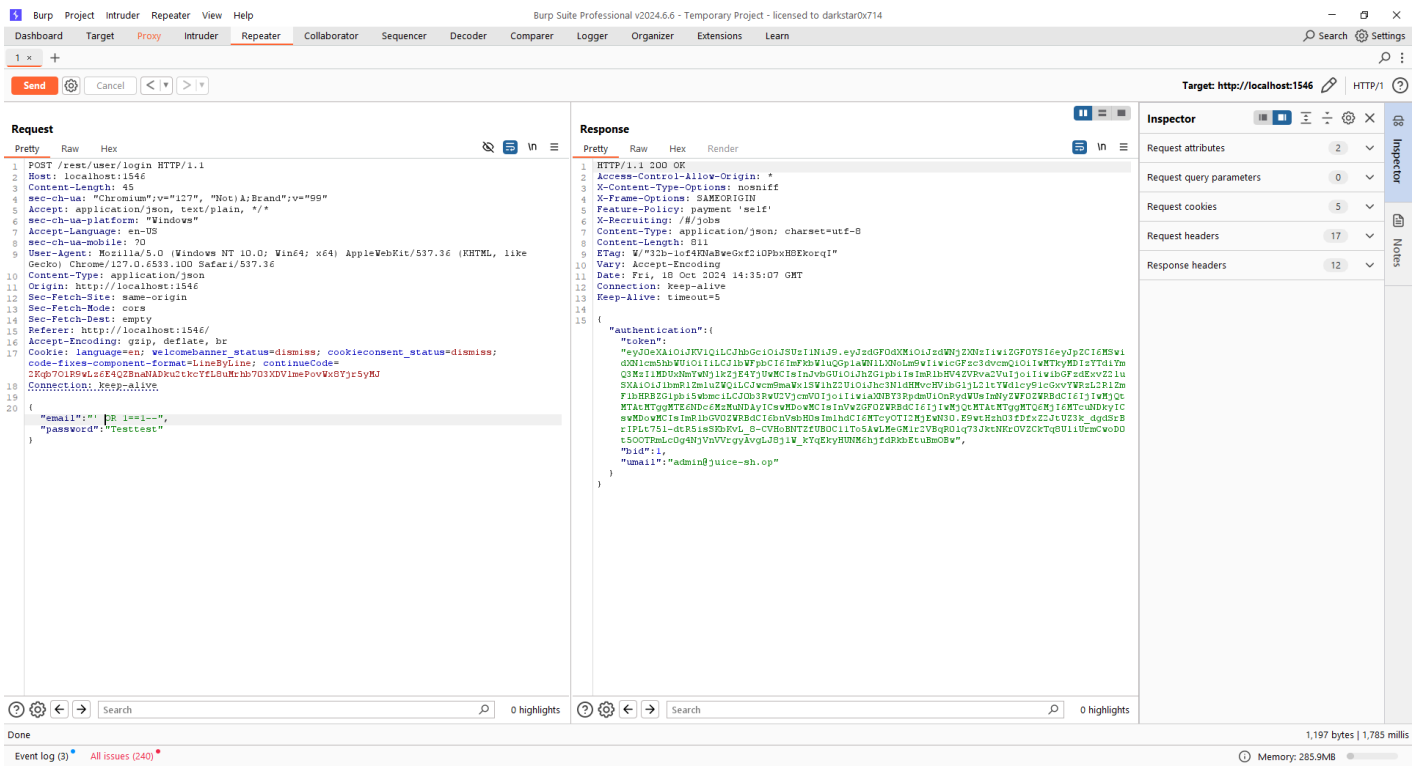
URL encoding: Encode user input used in URLs to prevent the injection of malicious scripts. Use functions like encodeURIComponent() in JavaScript or equivalent server-side methods.

Finding WAPT-010: SQL injection (Critical.)

Description:	in login page
System:	login page
Tools Used:	Browser - burp suite
CVSS v3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:H score 9.1

Evidence





Steps to Reproduce:

1. go to the login page and intercept it using burp suite but this shows the payload

Remediation

Validate user inputs: Before interacting with the database, ensure that user inputs are properly validated for expected formats (e.g., email addresses, numeric values). Use strong server-side validation to prevent malicious input from reaching the database.

Deploy a WAF: A Web Application Firewall can detect and block SQL injection attempts by analyzing HTTP requests and responses. This can act as an additional layer of security to complement application-level protections.

Minimize the use of dynamic SQL: If dynamic SQL is absolutely necessary, ensure that all input is properly sanitized or parameterized.

Always use prepared statements with parameterized queries: This is the most effective way to prevent SQL injection. Instead of directly embedding user input in SQL queries, bind user input as parameters. This ensures that user input is treated as data, not part of the SQL command.


DEPIX