Scrut Automation

Vulnerability Assessment & Penetration Testing

Web Application Final Report

Presented to - **Digiicampus**

July 29, 2025

Report By – Riversys Technologies Private Limited (Scrut Automation)

Engagement Overview

Digiicampus (Herein referred as **Digiicampus**) has engaged with Scrut Automation to conduct a penetration test of their Web Application. This report contains all the results of the report as well as all the action items that were included in the penetration test. The purpose of this report is to present the current security level of the external perimeters including gaps, vulnerabilities, and misconfigurations. The findings presented in this report should be fixed to improve the security level of the network systems.

Service Description

Web application Vulnerability Assessment and Penetration Testing (VAPT) is the process of simulating real-world attacks by using the same techniques as malicious hackers. For a security assessment that goes beyond a simple vulnerability scanner, you need experts in the industry. Scrut Automation conducts its penetration test by approaching the scope with both a manual and automatic approach.

Web Application Penetration Test

Our application-level penetration testing consists of both unauthenticated and authenticated testing using both automated and manual methods with particular emphasis placed on identifying vulnerabilities associated with the OWASP Top 10 Most Critical Application Vulnerabilities. It is important to note that a penetration test is not just an automated vulnerability scan, and a large portion of web application penetration testing is a manual process with a skilled engineer attempting to identify, exploit, and evaluate the associated risk of security issues.

Project Objectives

Scrut Automation consultants conduct all testing manually combined with custom and commercial tools that perform unique attack approaches on the network to make sure we cover the whole system in the test. Our expert knowledge and experience are the value we provide in our services.

Document Revision

DETAILS	
Title	BI - Web Application Final Report
Version	1.1
Date	July 29, 2025
Submitted to	Mr. Saksham Gupta
Submitted By	Mrs. Anusha Pyla

Test Performed Details

Tester Name	Reviewer Name	Test Date	Version
Mrs. Anusha Pyla	Mr. Kush Kaushik	June 02, 2025	1.0
Mrs. Anusha Pyla	Mr. Kush Kaushik	July 28, 2025	1.1

Table of Content

1. Executive Summary

1.1. Summary

This report presents the results of penetration testing and retesting conducted by Scrut Automation. The team carried out the penetration testing using automated tools and manual checks on **BI**. The assessment was conducted at staging Environment. The assessment started on June 02, 2025

The purpose of this assessment was to (i) Test the application to identify technical vulnerabilities and discover whether a malicious user may leverage these flaws to compromise the security of **BI**. (ii) Provide recommendations for risk mitigation that may arise on successful exploitation of these vulnerabilities.

The subsequent sections of this document provide statistics of the vulnerabilities identified, severity and proof of findings of **BI**. The detailed technical findings section constitutes identified vulnerabilities with recommendations to mitigate security risks associated.

1.2. Approach

- Exploring various application functionalities to enumerate threat & vulnerability in alignment with Open Web Application Security Project (OWASP) Top 10 vulnerabilities.
- Performing information gathering/ fingerprinting to identify software used/ its version, web server details, ports, and services open, etc.
- Performing vulnerability scanning to identify common vulnerabilities in the application layer and by using Burp and various testing tools in the Kali Linux distribution in conjunction with a range of manual analysis. It should be noted that customized payloads and attack vectors were configured in Burp Suite to further enhance the identification of weakness in the application.
- Analysing the automated scan results for any vulnerabilities and ease of exploitability and providing proof of concept where safe exploits are possible.
- Post-Exploitation process will be performed once we get access to the device using identified vulnerabilities/exploits.
- Reporting identified vulnerabilities and recommended solutions to mitigate them; for ease of mitigation activities for application support personnel/ developers' further details of CWEs were added.

1.3. Disclaimer

This report and any supplements are Confidential and may be protected by one or more legal privileges. It is intended solely for the use of the relevant IT personnel in the organization. This report is prepared based on the IT environment that prevailed in the period of assessment. This report is not a guarantee or certification that all vulnerabilities have been discovered and reported in the findings. Subsequent reviews may report on previously unidentified findings or on new vulnerabilities. The sample screen shots should not be treated as the final vulnerabilities. Gaps which we have identified can also get replicated in any part of the infrastructure. Organization should ensure that the Vulnerability Management Program should be adapted continuously rather than fixing just the issues identified within the areas of **BI**.

1.4. Limitations

The Scrut team did not have any limitations for this engagement.

1.5. OWASP TOP 10

- Broken Access Control
- Cryptographic Failures
- Injections
- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components
- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery

1.6. Scope

The scope included the following IP Addresses / Systems for vulnerability scanning and penetration testing.

Web Application Details:

S No.	Application Name	Application URLs
1	BI	https://bi.digiicampus.com/

1.7. Standards

OWASP: The OWASP web application security testing methodology and explains how to test for evidence of vulnerabilities within the application due to deficiencies with identified security controls.

The test is divided into 2 phases:

Phase one Passive mode:

In the passive mode, the tester tries to understand the application's logic and walkthrough with the application.

Phase two Active mode:

Information Gathering

Configuration and Deployment Management Testing

Identity Management Testing

Authentication Testing

Authorization Testing

Session Management Testing

Input Validation Testing

Error Handling

Cryptography

Business Logic Testing

Client-Side Testing

1.8. Vulnerability Scoring

The Risk level is divided in four categories:

Severity	DESCRIPTION
Critical	Critical vulnerabilities provide attackers with remote root or administrator capabilities. Malicious users have the ability to compromise the entire host. Easy to detect and exploit and result in large asset damage.
High	Exploitation of the vulnerability discovered on the system can directly lead an attacker to information allowing them to gain privileged access (e.g., administrator or root) to the system. These issues are often difficult to detect and exploit but can result in large asset damage.
Medium	The vulnerability discovered on the system can directly lead to an attacker gaining non-privileged access (e.g., as a standard user) to the system or the vulnerability provides access that can be leveraged within one step to gain administrator-level access. These issues are easy to detect and exploit but typically result in small asset damage.
Low	The vulnerability discovered on the system provides low-level, but sufficient data to the attacker that may be used to launch a more informed attack against the target environment. In addition, the vulnerability may indirectly lead to an attacker gaining some form of access to the system. These issues can be difficult to detect and exploit and typically result in small asset damage.

1.9. Key Findings

Vulnerability	OWASP TOP 10	Severity	Re-Test Status
Password guessing via Brute force attack	Broken Access Control	High	FIXED
Exploiting Metadata by File Upload vulnerability	Software and Data Integrity Failures	High	FIXED
Broken Access Control via Parameter Tampering	Broken Access Control	High	FIXED
TLS Version 1.0 & 1.1 Protocol Detection	Cryptographic Failures	Medium	FIXED
Application vulnerable to Sweet32 Attack	Cryptographic Failures	Medium	FIXED
Local Adversary can hijack authenticated sessions due to High session In- activity period	Identification and Authentication Failures	Medium	BUSINESS USE CASE
WEAK Cipher (3DES)/IDEA & Obsolete CBC ciphers supported	Cryptographic Failures	Medium	FIXED
Content Security Policy header does not present	Security Misconfiguration	Low	FIXED
Application is potentially vulnerable to LUCKY13 attack	Cryptographic Failures	Low	FIXED
Arbitrary HTTP methods enabled	Security Misconfiguration	Low	FIXED
Application is potentially vulnerable to BREACH attack	Cryptographic Failures	Low	CLOSED
Vulnerable version of Java Script Library Found	Vulnerable and Outdated Components	Low	OPEN
HTTP Strict Transport Security not configured in Application	Identification and Authentication Failures	Low	FIXED

Definitions:

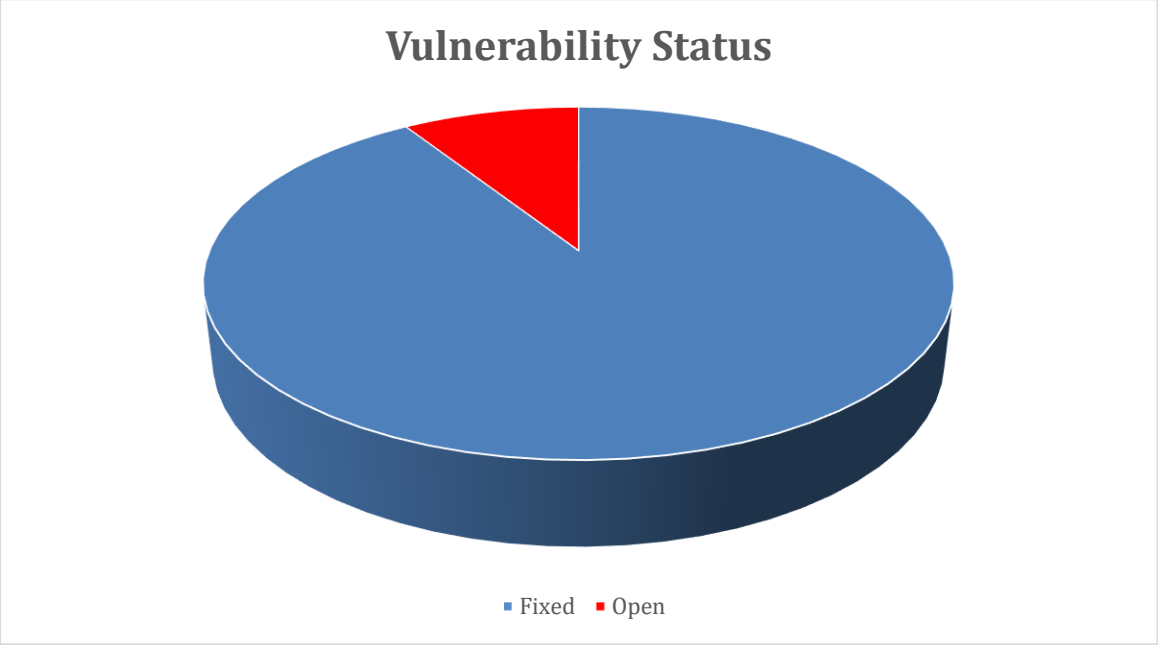
Open: The vulnerability has been identified and documented, but no action has been taken to address it yet. It requires attention and mitigation to reduce the associated risk.

Fixed: The vulnerability has been successfully addressed, and the necessary steps have been taken to eliminate or remediate it. Appropriate measures, such as applying patches or updates, have been implemented.

Closed: After thorough assessment, it has been determined that the reported vulnerability is not applicable, is not valid, or does not pose a genuine risk. No further action is required, and the vulnerability is considered closed.

Business Use Case: After thorough assessment, it has been determined that the reported vulnerability is not applicable, as this is required by the client application to work, is flow, or business logic required by the application

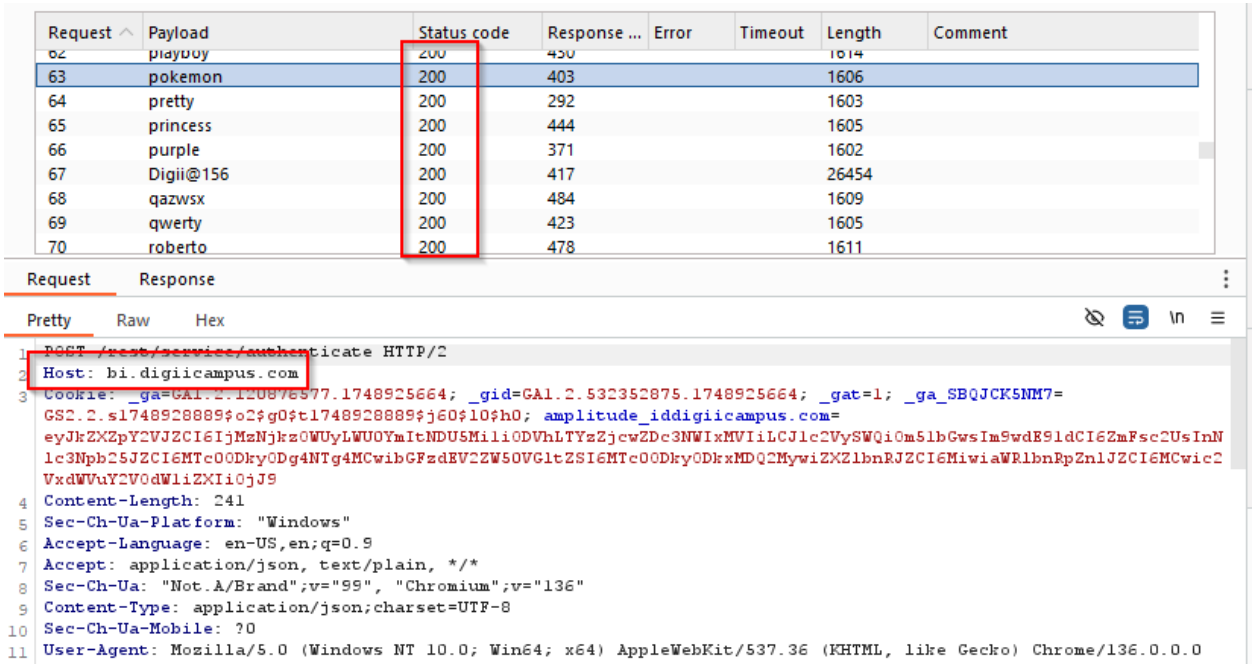
1.10. Vulnerability Graph

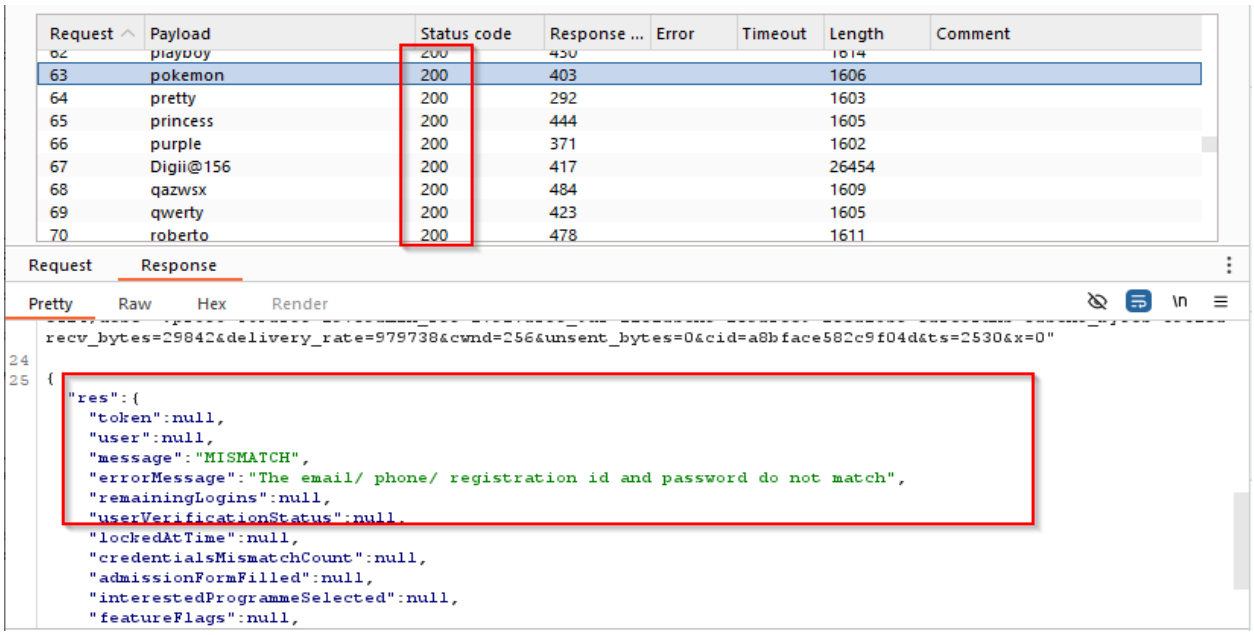


2. Findings

1. Password guessing via Brute force attack	
Severity:	**HIGH**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com/
Description:	Password guessing via brute force attack is a common method used by attackers to gain unauthorized access to user accounts or systems by systematically attempting various combinations of passwords until the correct one is found.
Impact:	Attack can easily do automated password guessing attacks can be tried against user accounts. There are tools that try to brute force user accounts by trying many passwords for a given username.
Recommendations:	It is recommended to implement a captcha mechanism or locked out mechanism in application Login Page.
References:	https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks
Step To Reproduce	• Navigate to Login Page as you can see there is no captcha on login page • Intercept the request of login User that requests to perform the brute force attack

Proof of Concept:



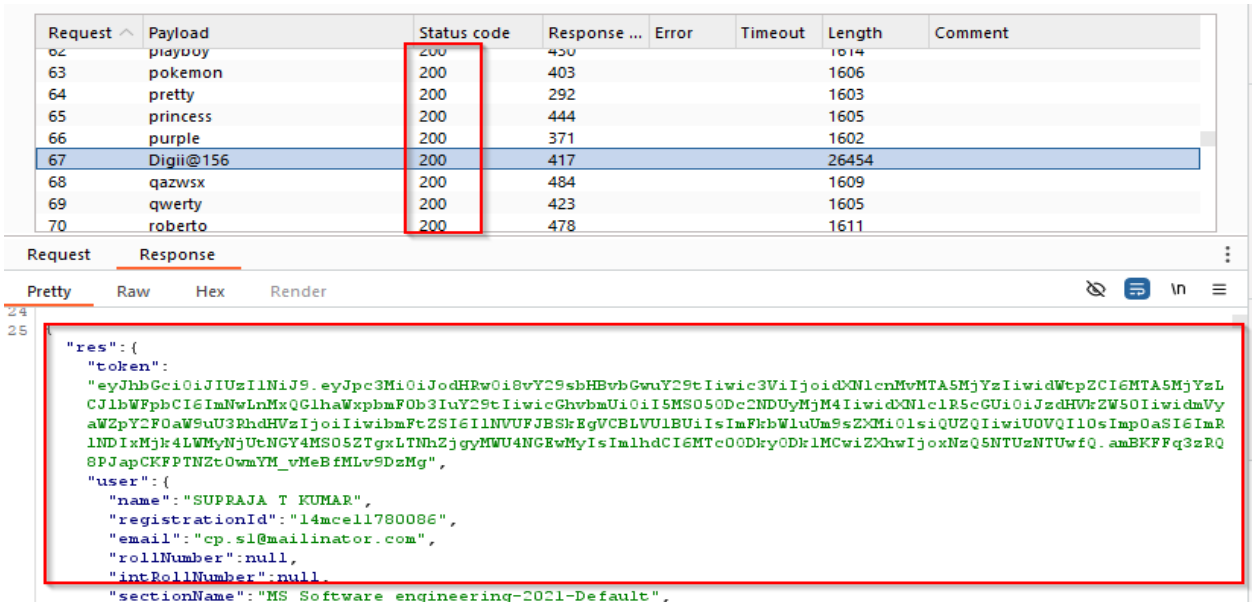


Request	Payload	Status code	Response ...	Error	Timeout	Length	Comment
62	playboy	200	450			1614	
63	pokemon	200	403			1606	
64	pretty	200	292			1603	
65	princess	200	444			1605	
66	purple	200	371			1602	
67	Digii@156	200	417			26454	
68	qazwsx	200	484			1609	
69	qwerty	200	423			1605	
70	roberto	200	478			1611	

Request Response

Pretty Raw Hex Render


```
recv_bytes=29842&delivery_rate=979738&cwnd=256&unsent_bytes=0&cid=a8bface582c9f04d&ts=2530&x=0"

{
  "res":{
    "token":null,
    "user":null,
    "message":"MISMATCH",
    "errorMessage":"The email/ phone/ registration id and password do not match",
    "remainingLogins":null,
    "userVerificationStatus":null,
    "lockedAtTime":null,
    "credentialsMismatchCount":null,
    "admissionFormFilled":null,
    "interestedProgrammeSelected":null,
    "featureFlags":null,
```

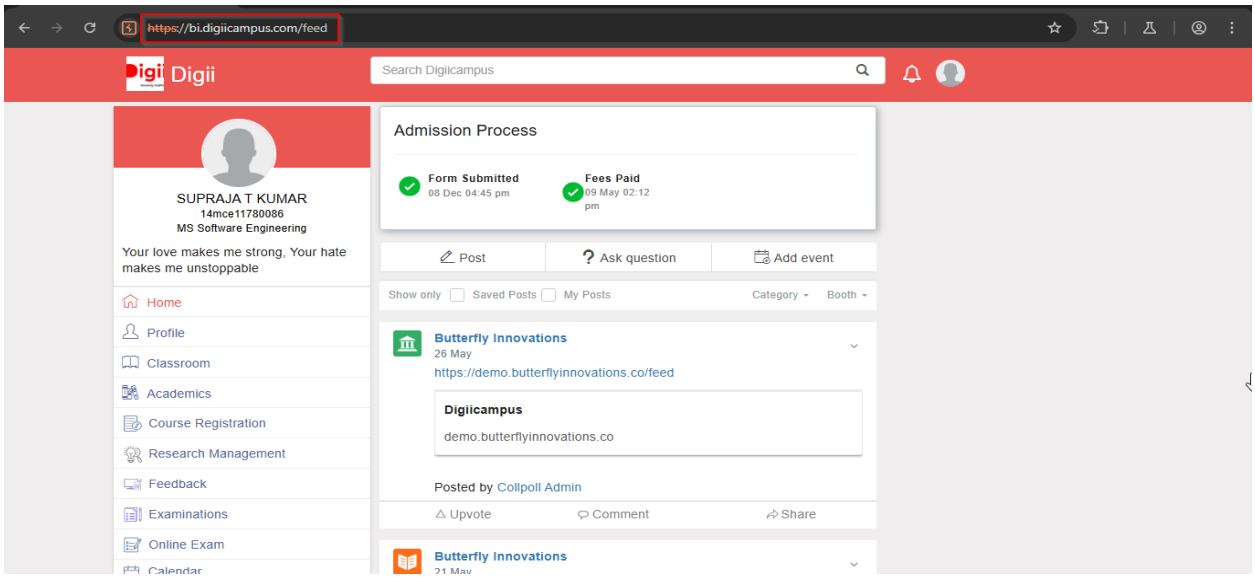



Request	Payload	Status code	Response ...	Error	Timeout	Length	Comment
62	playboy	200	450			1614	
63	pokemon	200	403			1606	
64	pretty	200	292			1603	
65	princess	200	444			1605	
66	purple	200	371			1602	
67	Digii@156	200	417			26454	
68	qazwsx	200	484			1609	
69	qwerty	200	423			1605	
70	roberto	200	478			1611	

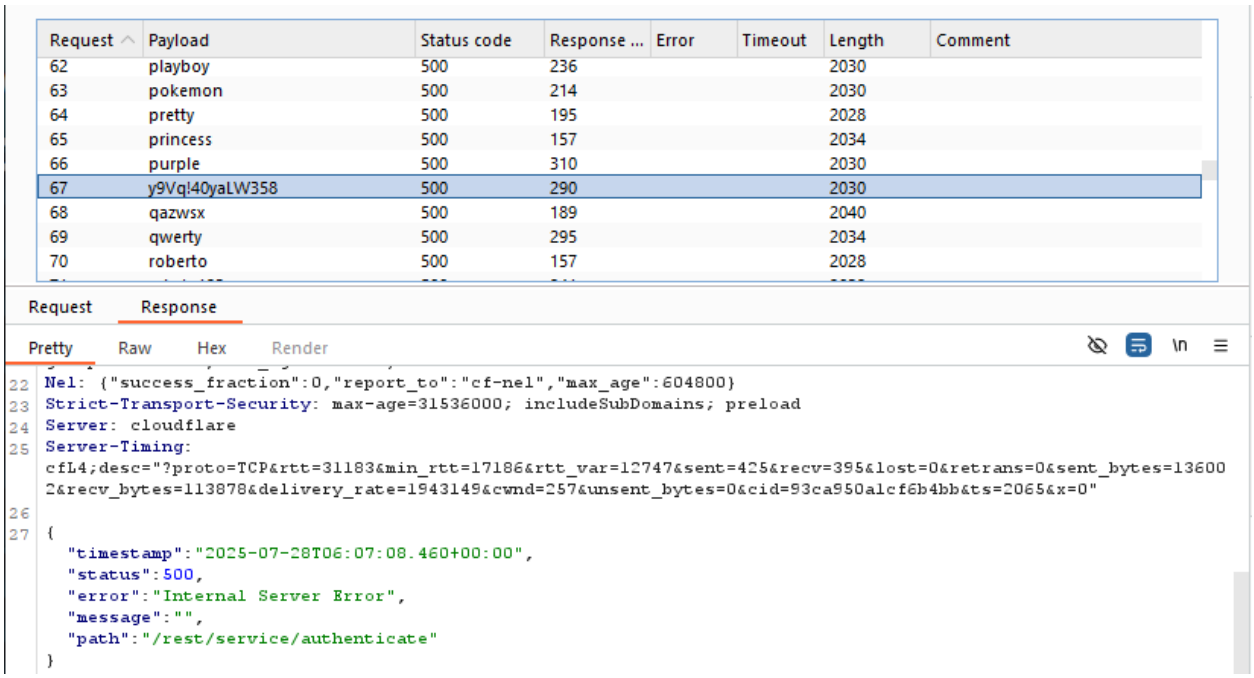
Request Response

Pretty Raw Hex Render


```
{
  "res":{
    "token":
    "eyJhbGci0iJIUzI1NiJ9.eyJpc3Mi0iJodHRw0i8vY29sbHBvbGwuY29tIiwic3ViIjoidXNlcnMvMTA5MjYzIiwidWtpZCI6MTA5MjYzL
    CJlbWFpbCI6ImNwLnMxQGlhaWxpbmF0b3IuY29tIiwicGhvbmUi0iI5MS050Dc2NDUyMjM4IiwidXNlclR5cGUi0iJzdHVkZW50IiwidmVy
    aW2pY2F0aW9uU3RhdHVzIjoiIiwibmFtZSI6IlNVUFJBSkEgVCBLVUlBUiIsImFkbWluUm9sZXMi0lsiQUZQIiwiUOVQIl0sImp0aSI6ImR
    lNDIxMjk4LWMyNjUtNGY4MS05ZTgxLTNhZjgyMWU4NGEwMyIsImlhdCI6MTc00Dky0Dk1MCwiZXhwIjoxNzQ5NTUzNTUwfQ.amBKFFq3zRQ
    8PJapCKFPTNZt0wmYM_vMeBfMLv9DzMg",
    "user":{
      "name":"SUPRAJA T KUMAR",
      "registrationId":"14mce11780086",
      "email":"cp.sl@mailinator.com",
      "rollNumber":null,
      "intRollNumber":null,
      "sectionName":"MS Software engineering-2021-Default",
```

Retest Proof of Concept:



Request	Payload	Status code	Response ...	Error	Timeout	Length	Comment
62	playboy	500	236			2030	
63	pokemon	500	214			2030	
64	pretty	500	195			2028	
65	princess	500	157			2034	
66	purple	500	310			2030	
67	y9Vq!40yaLW358	500	290			2030	
68	qazwsx	500	189			2040	
69	qwerty	500	295			2034	
70	roberto	500	157			2028	

Request Response

Pretty Raw Hex Render

```
Nel: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
Server: cloudflare
Server-Timing:
cfL4;desc="?proto=TCP&rtt=31183&min_rtt=17186&rtt_var=12747&sent=425&recv=395&lost=0&retrans=0&sent_bytes=13600
2&recv_bytes=113878&delivery_rate=1943149&cwnd=257&unsent_bytes=0&cid=93ca950a1cf6b4bb&ts=2065&x=0"

{
  "timestamp":"2025-07-28T06:07:08.460+00:00",
  "status":500,
  "error":"Internal Server Error",
  "message":"",
  "path":"/rest/service/authenticate"
}
```

2. Exploiting Metadata by File Upload vulnerability	
Severity:	**HIGH**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com/studentManagement/details/
Description:	A file upload vulnerability occurs when a website or application allows users to upload files but fails to properly validate and sanitize the file metadata, allowing malicious users to upload files with malicious content or execute arbitrary code. The application performs check (jpeg/png) based on the meta data. It is possible to upload a php script by exploiting the metadata in the application.
Impact:	An attacker could embed a script in the metadata that would download additional malware onto the server, steal sensitive data, or perform other harmful actions.
Recommendations:	It is recommended to ensure that all file uploads are properly validated and sanitized, including the metadata. This can be done by using strict file type checking, limiting file size, and filtering out any known malicious content.
References:	https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload CWE-434: Unrestricted Upload of File with Dangerous Type
Step To Reproduce	• Create a php file with malicious command and embed the png metadata in the file • Magic byte of .png: 0x89 0x50 0x4E 0x47 0x0D 0x0A 0x1A 0x0A • Use cmd: echo -n 0x89 0x50 0x4E 0x47 0x0D 0x0A 0x1A 0x0A > metafile.php.pdf • Navigate to a documents in application and upload the file in upload file. • Intercept the request and change the Metafile.php.pdf to Metafile.php and save it. • You can observe that the application has not sanitize the metadata in the saved metafile.

Proof of Concept:

Request

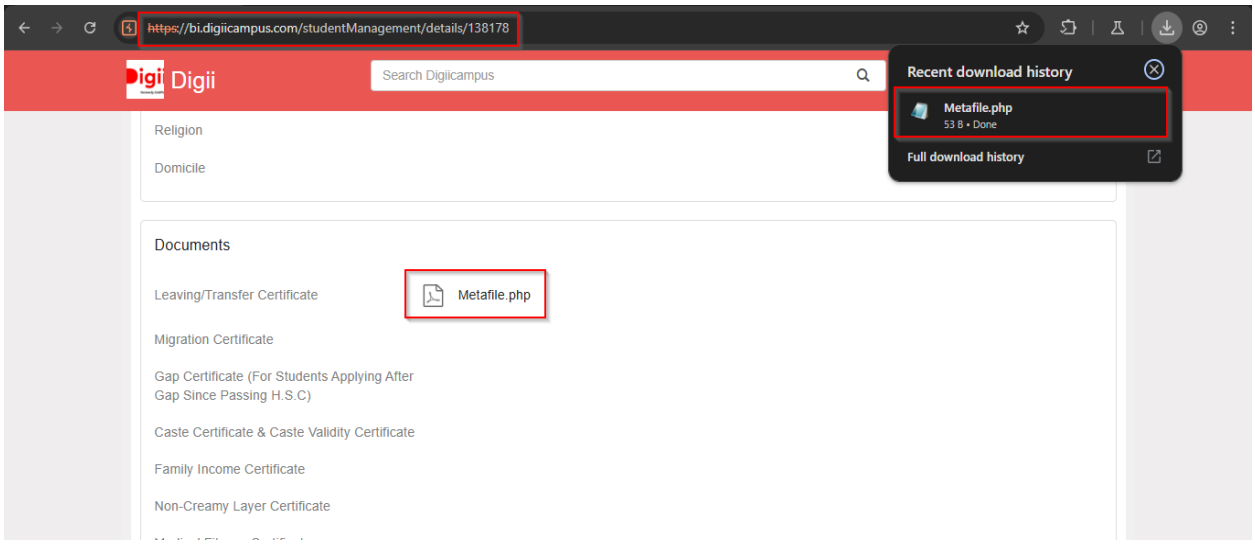
Pretty Raw Hex

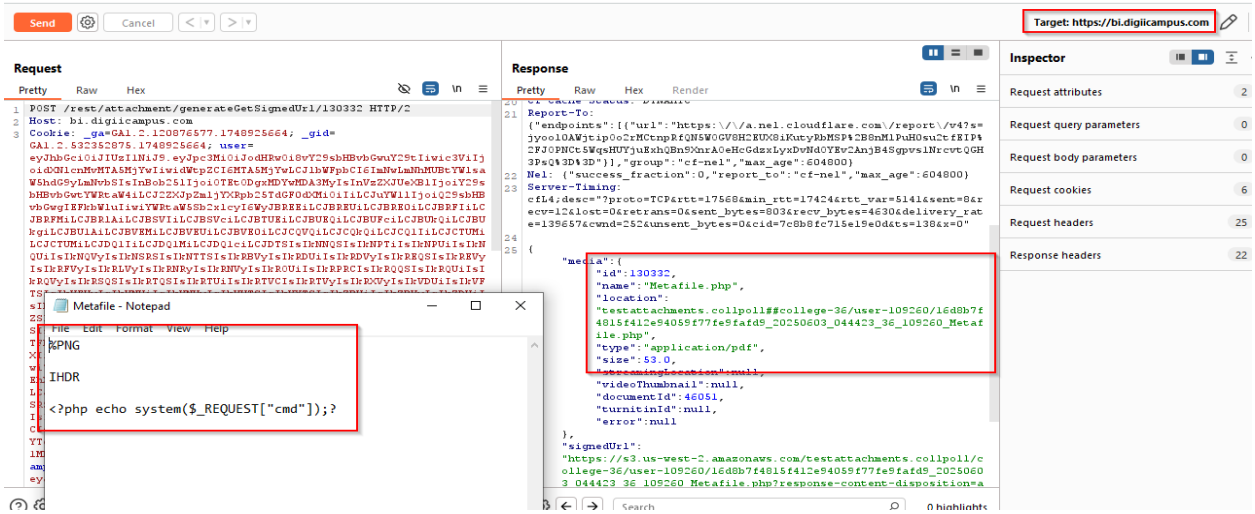
```
Auth-Token:
eyJhbGciOiJIUzI1NiJ9.eyJpc3Mi0iJodHRw0i8vY29sbHBvbGwuY29tIiwic3ViIjoidXNlcnMvMTA5MjYwIiwidWtpZCI6MTA5MjYwLCJlbWFpbCI6ImNwLmNhMUBtYWlsaW5hdG9
yLmNvbSIsInBob25lIjoi0TEt0DgxMDYwMDA3MyIsInVzZXJUeXBlIjoiY29sbHBvbGwtYWRtaW4iLCJ2ZXJpZmljYXRpb25TdGF0dXMi0iIiLCJuYWlIjoiQ29sbHBvbGwgIEFkbWl
uIiwiYWRtaW5Sb2xlcyI6WyJBREEiLCJBREUiLCJBREE0iLCJBRFIiLCJBRFMiLCJBRlAiLCJBSVIiLCJBSVciLCJBTUEiLCJBUEQiLCJBUFciLCJBURQiLCJBUkgiLCJBUlAiLCJBVEM
iLCJBVEUiLCJBVE0iLCJCQVQiLCJCQkQiLCJCQ1IiLCJCTUMiLCJCTUMiLCJDQlIiLCJDQlMiLCJDQlciLCJDTSIsIkNNQSIsIkNPTiIsIkNPUiIsIkNQUiIsIkNQVyIsIkNSRSIsIkN
TTSIsIkRBVyIsIkRDUiIsIkRDVyIsIkREQSIsIkREVyIsIkRFVyIsIkRLVyIsIkRNRyIsIkRNVyIsIkROUiIsIkRPRCIsIkRQQSIsIkRQUiIsIkRQVyIsIkRSQSIsIkRTQSIsIkRTUiI
sIkRTVCIsIkRTVyIsIkRXVyIsIkVDUiIsIkVFTSIsIkVFUyIsIkVNUiIsIkVNUyIsIkVUTSIsIkVVTSIsIkZBUiIsIkZBVyIsIkZDUiIsIkZFRSIsIkZJQSIsIkZJTCIsIkZMVSIsIkZ
NUiIsIkZNUyIsIkZ0QyIsIkZ0Qy1sIkZ0RSIsIkZSViIsIkdPRyIsIkdQTSIsIkdQUyIsIkhBUyIsIkhTUiIsIkhTVyIsIkhUQSIsIkhUTCIsIklDTCIsIklEQyIsIkl0TSIsIkpNQyI
sIktPSCIsIkxCVSIsIkxEIiwiTFBBIiwiTFBTIiwiTFRTIiwiTUlXIiwiTUlFIiwiTUlTIiwiTVBBIiwiTVBXIiwiTVJXIiwiTVNSIiwiTVNUIiwiTVNXIiwiTVdXIiwiTkFQIiwiTkR
NIiwiTlBEIiwiTlRDIiwiTlREIiwiTOJBIiwiTOJEIiwiTOJTIiwiUEFZIiwiUEZSIiwiUEZXIiwiUEhSIiwiUEhXIiwiUExBIiwiUE0iLCJQTUUiLCJQTVYiLCJQUEEiLCJQUFMiLCJ
QVEOiLCJSTUMiLCJSUCIsIlJQUiIsIlNBUiIsIlNBVyIsIlNDSCIsIlNGUiIsIlNMTSIsIlNMVyIsIlNSRSIsIlNSViIsIlNUUiIsIlNUUyIsIlNVUCIsIlRBUiIsIlRBVyI
sIlRDUiIsIlRJQyIsIlRQTSIsIlVNIiwiVUlTIiwiVVAiLCJWQUMiLCJWQiIsIlZCRCIsIlZDSCIsIlZGRCIsIlZJUyIsIlZQTCIsIlpBQyJdLCJqdGki0iI0OTllMTk5ZClmMjk2LTQ
wYTct0DRlNCliN2ViYTRjNjMzYTgiLCJpYXQi0jE3NDg5MjU3NTEsImV4cCI6MTc00TU1MDM1MX0.YOcH9DCGREFhcKF23ipVo7CEl4acq45e7MhXYqpboMr
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=UTF-8
Origin: https://bi.digiicampus.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://bi.digiicampus.com/studentManagement/admin/edit/138178
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

{
  "contentLength":53,
  "fileName":"Metafile.php.pdf",
  "contentType":"application/pdf",
  "feature":"CLASSQUIZ",
  "isSynchronous":false,
  "shouldStream":false
}
```

Request

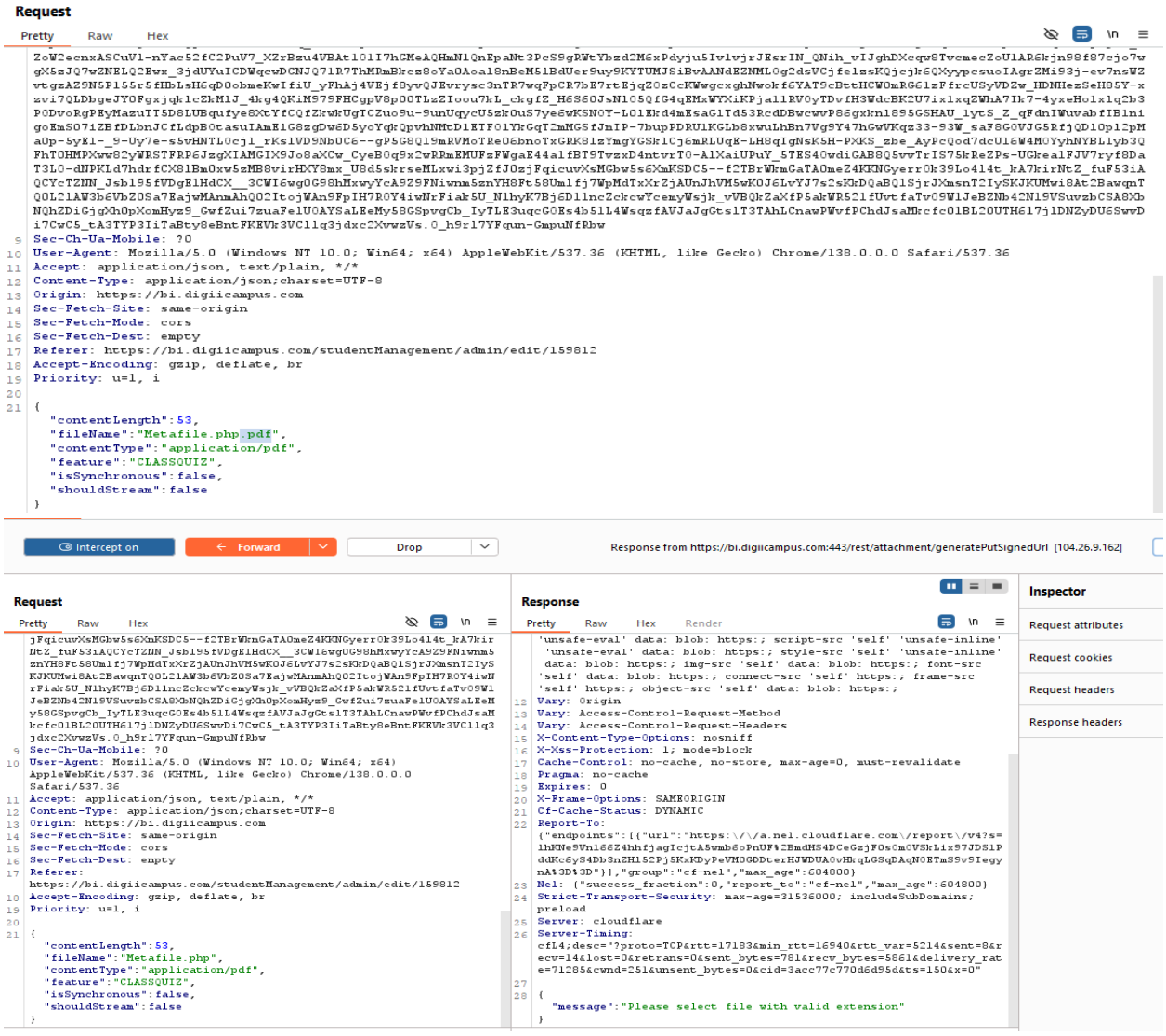
Pretty Raw Hex

```
Auth-Token:
eyJhbGciOiJIUzI1NiJ9.eyJpc3Mi0iJodHRw0i8vY29sbHBvbGwuY29tIiwic3ViIjoidXNlcnMvMTA5MjYwIiwidWtpZCI6MTA5MjYwLCJlbWFpbCI6ImNwLmNhMUBtYWlsaW5hdG9
yLmNvbSIsInBob25lIjoi0TEt0DgxMDYwMDA3MyIsInVzZXJUeXBlIjoiY29sbHBvbGwtYWRtaW4iLCJ2ZXJpZmljYXRpb25TdGF0dXMi0iIiLCJuYWlIjoiQ29sbHBvbGwgIEFkbWl
uIiwiYWRtaW5Sb2xlcyI6WyJBREEiLCJBREUiLCJBREE0iLCJBRFIiLCJBRFMiLCJBRlAiLCJBSVIiLCJBSVciLCJBTUEiLCJBUEQiLCJBUFciLCJBURQiLCJBURgiLCJBUlAiLCJBVEM
iLCJBVEUiLCJBVE0iLCJCQVQiLCJCQkQiLCJCQ1IiLCJCTUMiLCJCTUMiLCJDQlIiLCJDQlMiLCJDQlciLCJDTSIsIkNNQSIsIkNPTiIsIkNPUiIsIkNQUiIsIkNQVyIsIkNSRSIsIkN
TTSIsIkRBVyIsIkRDUiIsIkRDVyIsIkREQSIsIkREVyIsIkRFVyIsIkRLVyIsIkRNRyIsIkRNVyIsIkROUiIsIkRPRCIsIkRQQSIsIkRQUiIsIkRQVyIsIkRSQSIsIkRTQSIsIkRTUiI
sIkRTVCIsIkRTVyIsIkRXVyIsIkVDUiIsIkVFTSIsIkVFUyIsIkVNUiIsIkVNUyIsIkVUTSIsIkVVTSIsIkZBUiIsIkZBVyIsIkZDUiIsIkZFRSIsIkZJQSIsIkZJTCIsIkZMVSIsIkZ
NUiIsIkZNUyIsIkZ0QyIsIkZ0Qy1sIkZ0RSIsIkZSViIsIkdPRyIsIkdQTSIsIkdQUyIsIkhBUyIsIkhTUiIsIkhTVyIsIkhUQSIsIkhUTCIsIklDTCIsIklEQyIsIkl0TSIsIkpNQyI
sIktPSCIsIkxCVSIsIkxEIiwiTFBBIiwiTFBTIiwiTFRTIiwiTUlXIiwiTUlFIiwiTUlTIiwiTVBBIiwiTVBXIiwiTVJXIiwiTVNSIiwiTVNUIiwiTVNXIiwiTVdXIiwiTkFQIiwiTkR
NIiwiTlBEIiwiTlRDIiwiTlREIiwiTOJBIiwiTOJEIiwiTOJTIiwiUEFZIiwiUEZSIiwiUEZXIiwiUEhSIiwiUEhXIiwiUExBIiwiUE0iLCJQTUUiLCJQTVYiLCJQUEEiLCJQUFMiLCJ
QVEOiLCJSTUMiLCJSUCIsIlJQUiIsIlNBUiIsIlNBVyIsIlNDSCIsIlNGUiIsIlNMTSIsIlNMVyIsIlNSRSIsIlNSViIsIlNUUiIsIlNUUyIsIlNVUCIsIlRBUiIsIlRBVyI
sIlRDUiIsIlRJQyIsIlRQTSIsIlVNIiwiVUlTIiwiVVAiLCJWQUMiLCJWQiIsIlZCRCIsIlZDSCIsIlZGRCIsIlZJUyIsIlZQTCIsIlpBQyJdLCJqdGki0iI0OTllMTk5ZClmMjk2LTQ
wYTct0DRlNCliN2ViYTRjNjMzYTgiLCJpYXQi0jE3NDg5MjU3NTEsImV4cCI6MTc00TU1MDM1MX0.YOcH9DCGREFhcKF23ipVo7CEl4acq45e7MhXYqpboMr
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=UTF-8
Origin: https://bi.digiicampus.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://bi.digiicampus.com/studentManagement/admin/edit/138178
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

{
  "contentLength":53,
  "fileName":"Metafile.php",
  "contentType":"application/pdf",
  "feature":"CLASSQUIZ",
  "isSynchronous":false,
  "shouldStream":false
}
```



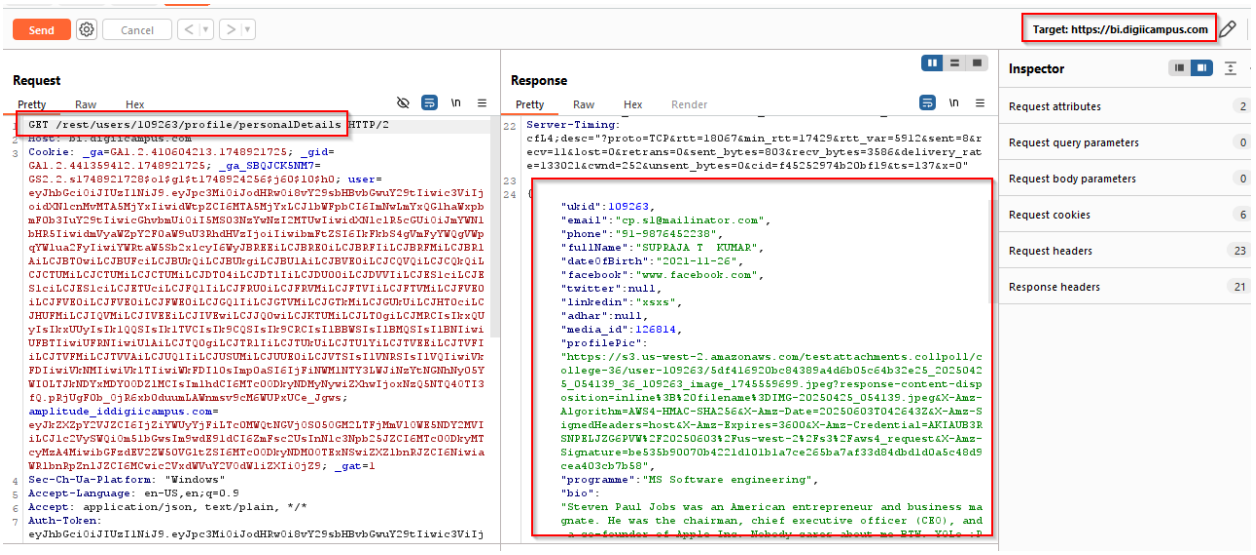


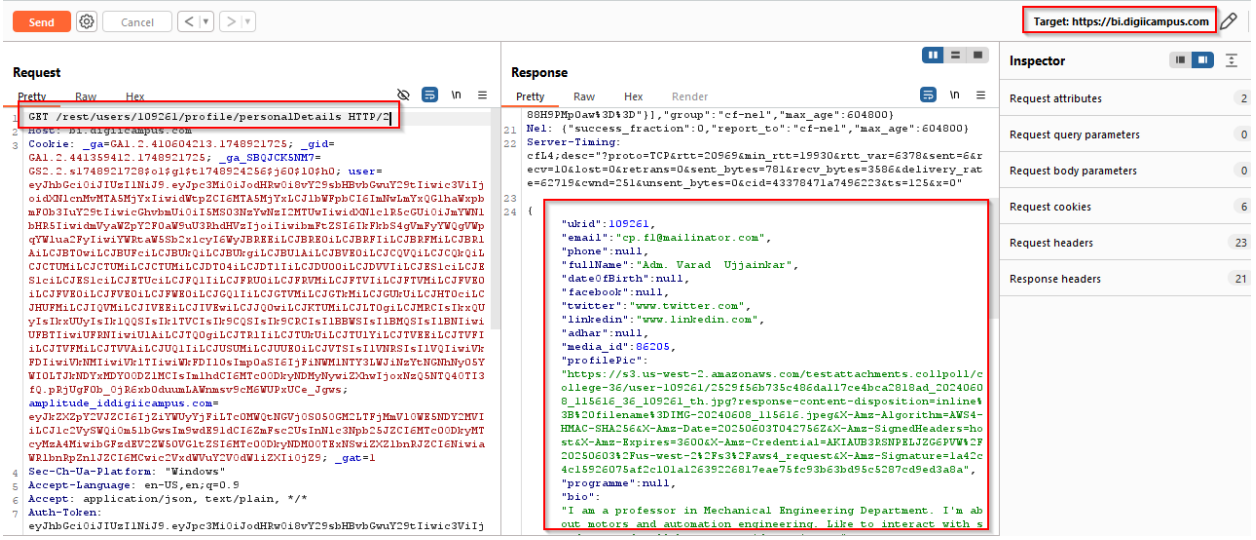
Retest Proof of Concept:



3. Broken Access Control via Parameter Tampering	
Severity:	**HIGH**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com/users/profile/personaldetails
Description:	Broken access control occurs when an application fails to adequately enforce user permissions and access restrictions. This can allow users to perform actions or access data that should be outside their scope of authority. One common form of this issue is insecure direct object reference (IDOR), where user-controllable parameters can be manipulated to access or modify resources belonging to other users due to missing or improper authorization checks on the server side.
Impact:	➢ Unauthorized disclosure of user data such as full names, user IDs, and assigned roles. ➢ Enables horizontal privilege escalation, where a low-privileged user can enumerate and extract information belonging to other users. ➢ Increases the risk of targeted attacks, social engineering, and internal reconnaissance. ➢ Violates the principle of least privilege and could lead to compliance issues under data protection regulations (e.g., GDPR, CCPA).
Recommendations:	✓ Implement strict object-level access controls on the server to verify that the authenticated user is authorized to access the requested user_id. ✓ Avoid making access control decisions based solely on client-supplied identifiers. ✓ Use role-based access checks or ownership validation on all endpoints returning user-specific data. ✓ Consider replacing predictable identifiers (e.g., numeric user IDs) with securely scoped identifiers (e.g., UUIDs or opaque tokens) to reduce the risk of enumeration.
References:	https://owasp.org/www-community/attacks/Parameter_tampering CWE-472: External Control of Assumed-Immutable Web Parameter
Step To Reproduce	• Log in to the application and intercept the user's request • You can observe the user details in the response. • Change the user id and send the request. • You can observe the other user's details can be accessed successfully. • Send the request to intercept and add a range to the user id and observe that you can access all the other user's details successfully.

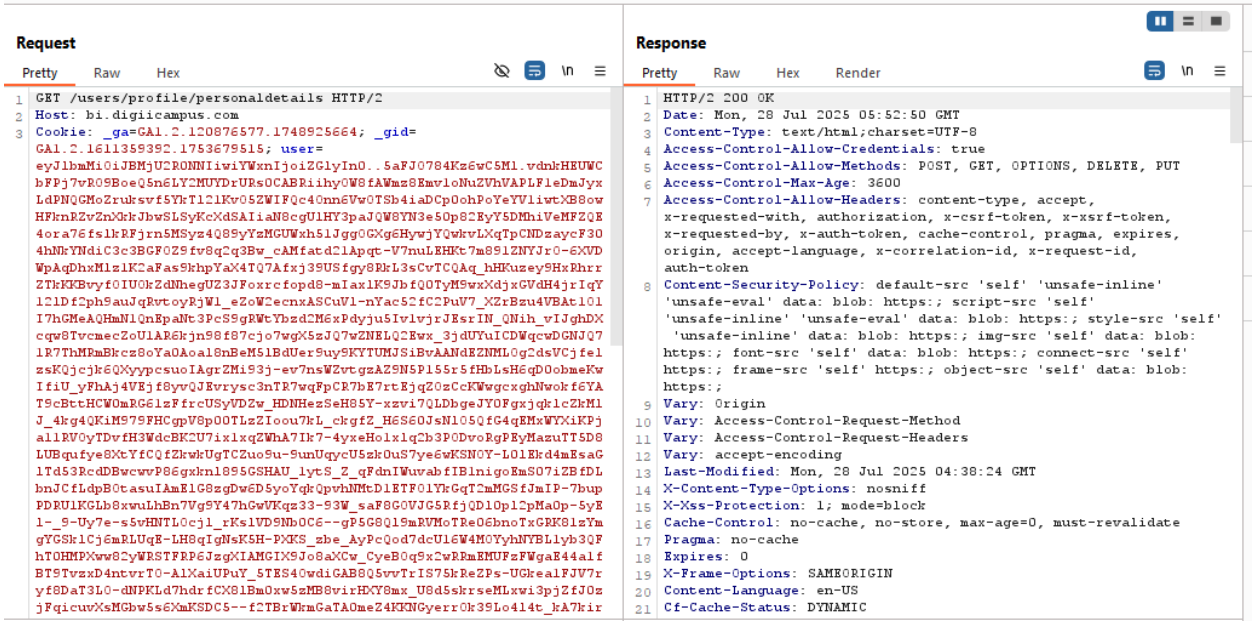
Proof of Concept:





Request	Payload	Status code	Response ...	Error	Timeout	Length	Comment
33	109282	200	71			1647	
34	109283	200	104			2127	
35	109284	200	133			2131	
36	109285	200	86			1651	
37	109286	200	110			1652	
38	109287	200	115			2138	
39	109288	200	103			1616	

Retest Proof of Concept:



Request

Pretty Raw Hex

```
1 GET /users/profile/personaldetails HTTP/2
2 Host: bi.digiicampus.com
3 Cookie: _ga=GA1.2.120876577.1748925664; _gid=
GA1.2.1611359392.1753679515; user=
eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiZGlyIn0..5aFJ0784Kz6wC5M1.vdnkHEUWC
bFPj7vR09BoeQ5n6LY2MUYDrURs0CABRiihy0W8fAWmz8EmvloNuZVhVAPLFleDmJyx
LdPNQGMoZrukswf5YkTl2lKv05ZWIFQc40nn6Vw0TSb4iaDCp0ohPoYeYVliwtXB8ow
HFknRZvZnXGkJbwSLSyKcXdSAIiaN8cgUlHY3paJQW8YN3e50p82EyY5DMhiVeMFZQE
4ora76fslkRFjrn5MSyz4Q89yYzMGUWxh5lJgg0GXg6HywjYQwkvLXqTpCNDzaycF30
4hNkYNdiC3c3BGF0Z9fv8q2q3Bw_cAMfatd2lApqt-V7nuLEHKt7m89lZNYJr0-6XVD
WpAqDhxMlzlK2aFas9khpYaX4TQ7Afxj39USfgy8RkL3sCvTCQAq_hHKuzey9HxRhrr
ZTkKKBvyf0IU0kZdNhegUZ3JFoxrcfopd8-mIaxlK9JbfQOTyMSwxXdjxGVdH4jrIqY
l2lDf2ph9auJqRvtoyRjWl_eZoWZecnxASCuVl-nYac52fC2PuV7_XZrBzu4VBAtl0l
I7hGMeAQHmNlQnEpaNt3PcS9gRWtYbzdZM6xPdyju5IvlvjrJEsrIN_QNih_vIJghDX
cqw8TvcmecZoUlAR6kjn98f87cjo7wgX5zJQ7wZNELQ2Ewx_3jdUYuICDWqcwDGNJQ7
lR7ThMRmBkcz8oYaOaal8nBeM5lBdUer9uy9KYTUMJSiBvAANdEZNML0g2dsVCjfel
zsKQjcjk6QXyypcsuoIAgrZMi93j-ev7nsWZvtgzAZ9N5P155r5fHbLsH6qD0obmeKw
IfiU_yFhAj4VEjf8yvQJEvrysc3nTR7wqFpCR7bE7rtEjqZ0zCcKWwgcxghNwokf6YA
T9cBttHCW0mRG6lzFfrcUSyVDZw_HDNHezSeH85Y-xzvi7QLDbgeJYOFgxjqklcZkMl
J_4kg4QKiM979FHCgpV8p00TLzZIoou7kL_ckgfZ_H6S6OJsNl05QfG4qEMxWYXiKPj
allRV0yTDvfH3WdcBK2U7ixlxqZWhA7Ik7-4yxeHolxlqCb3P0DvoRgPEyMazuTT5D8
LUBqufye8XtYfCQfZkwkUgTCZuo9u-9unUqycU5zk0uS7ye6wKSN0Y-L0lEkd4mEsaG
lTd53RcdDBwcwvP86gxknl895GSHAU_lytS_Z_qFdnIWuvabfIBlnigoEmSO7iZBfDL
bnJCfLdpB0tasuIAmE1G8zgDw6D5yoYqkQpvhNMtDlETF0lYkGqT2mMGSfJmIP-7bup
PDRUlKGLb8xwuLhBn7Vg9Y47hGwVKqz33-93W_saF8G0VJG5RfjQDlOpl2pMaOp-5yE
l-_9-Uy7e-s5vHNTL0cjl_rKslVD9Nb0C6--gP5G8Q19mRVMoTReO6bnoTxGPK8lzYm
gYGSklCj6mRLUqE-LH8qIgNsK5H-PXKS_zbe_AyPcQod7dcUl6W4M0YyhNYBLlyb3QF
hT0HMPXww82yWRSTFRP6JzgXIAMGIX9Jo8aXCw_CyeB0q9x2wRRmEMUFzFWgaK44alf
BT9TvzxD4ntvrT0-AlXaiUPuY_5TES40wdiGAB8Q5vvTrIS75kReZPs-UGkealFJV7r
yf8DaT3L0-dNPKLd7hdrfCX8lBm0xw5zMB8virHXY8mx_U8d5skrseMLxwi3pjZfJOz
jFqicuvXsMGbw5s6XmKSDC5--f2TBrWkmGaTA0meZ4KKNGyerr0k39Lo414t_kA7kir
```

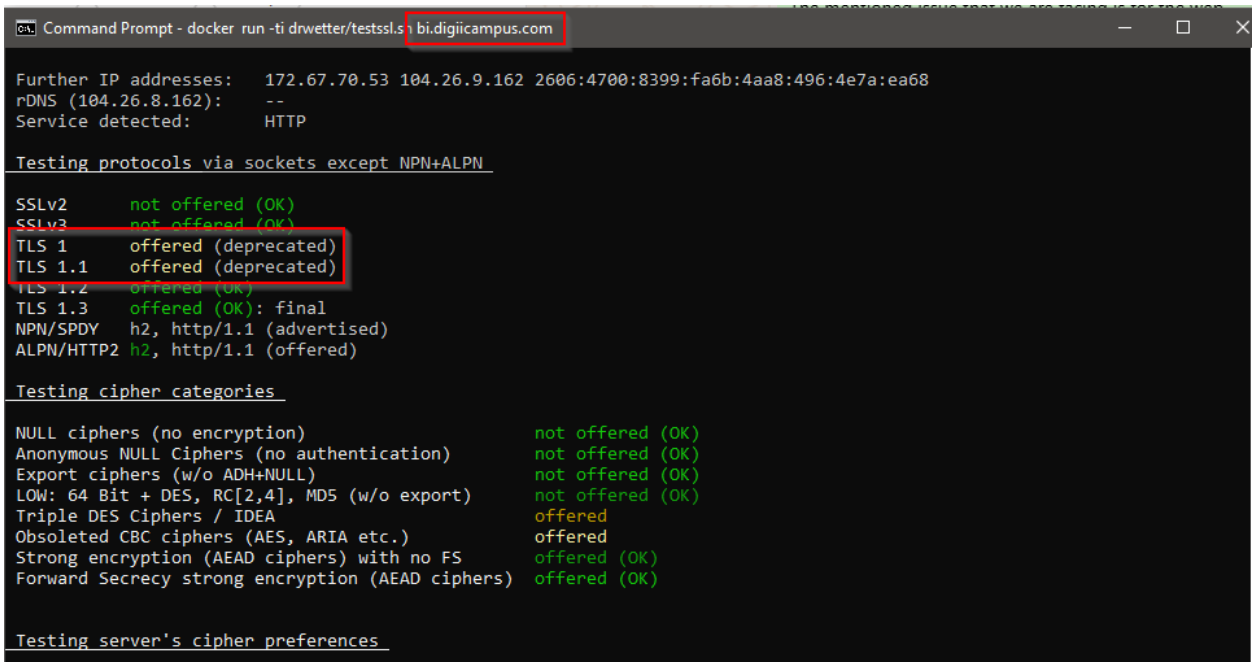
Response

Pretty Raw Hex Render

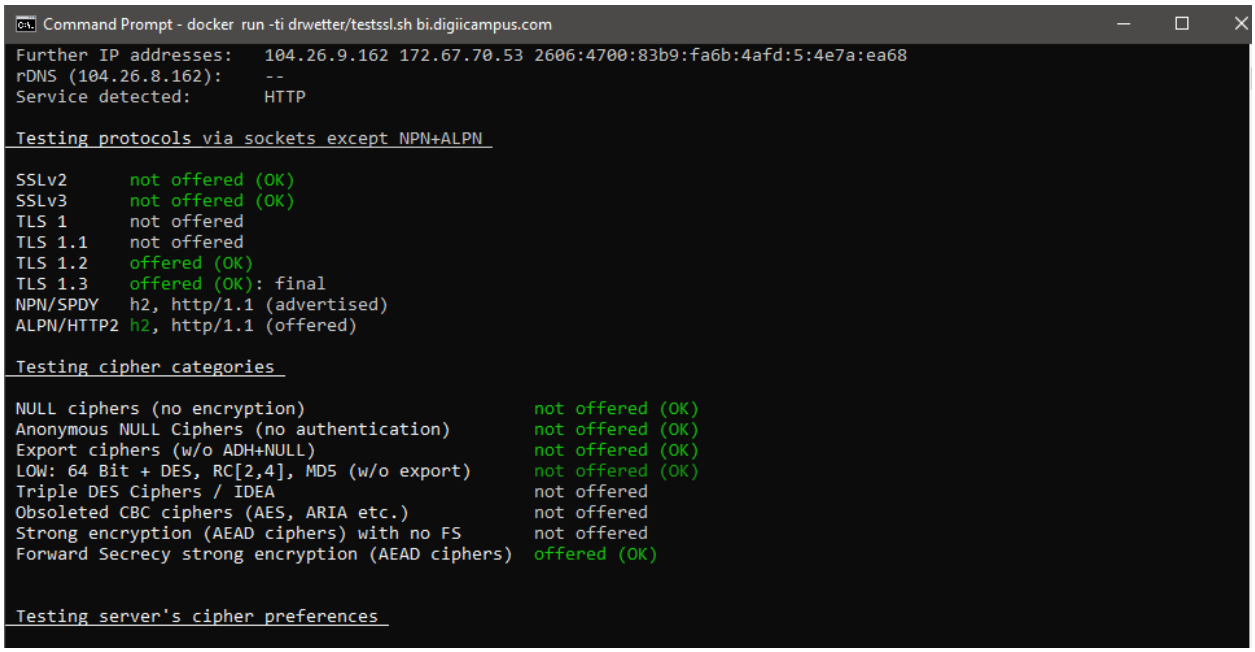
```
1 HTTP/2 200 OK
2 Date: Mon, 28 Jul 2025 05:52:50 GMT
3 Content-Type: text/html;charset=UTF-8
4 Access-Control-Allow-Credentials: true
5 Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
6 Access-Control-Max-Age: 3600
7 Access-Control-Allow-Headers: content-type, accept,
x-requested-with, authorization, x-csrf-token, x-xsrf-token,
x-requested-by, x-auth-token, cache-control, pragma, expires,
origin, accept-language, x-correlation-id, x-request-id,
auth-token
8 Content-Security-Policy: default-src 'self' 'unsafe-inline'
'unsafe-eval' data: blob: https:; script-src 'self'
'unsafe-inline' 'unsafe-eval' data: blob: https:; style-src 'self'
 'unsafe-inline' data: blob: https:; img-src 'self' data: blob:
https:; font-src 'self' data: blob: https:; connect-src 'self'
https:; frame-src 'self' https:; object-src 'self' data: blob:
https:;
9 Vary: Origin
10 Vary: Access-Control-Request-Method
11 Vary: Access-Control-Request-Headers
12 Vary: accept-encoding
13 Last-Modified: Mon, 28 Jul 2025 04:38:24 GMT
14 X-Content-Type-Options: nosniff
15 X-Xss-Protection: 1; mode=block
16 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
17 Pragma: no-cache
18 Expires: 0
19 X-Frame-Options: SAMEORIGIN
20 Content-Language: en-US
21 Cf-Cache-Status: DYNAMIC
```

4. TLS Version 1.0 & 1.1 Protocol Detection	
Severity:	**MEDIUM**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com
Description:	The remote service accepts connections encrypted using TLS 1.0. TLS 1.1 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible. As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors. PCI DSS v3.2 requires that TLS 1.0 & 1.1 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits
Recommendations:	Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0 & TLS 1.1
References:	https://developer.mozilla.org/en-US/docs/Web/Security/Transport_Layer_Security
Step To Reproduce	Using test SSL scanner to observe this vulnerability

Proof of Concept:

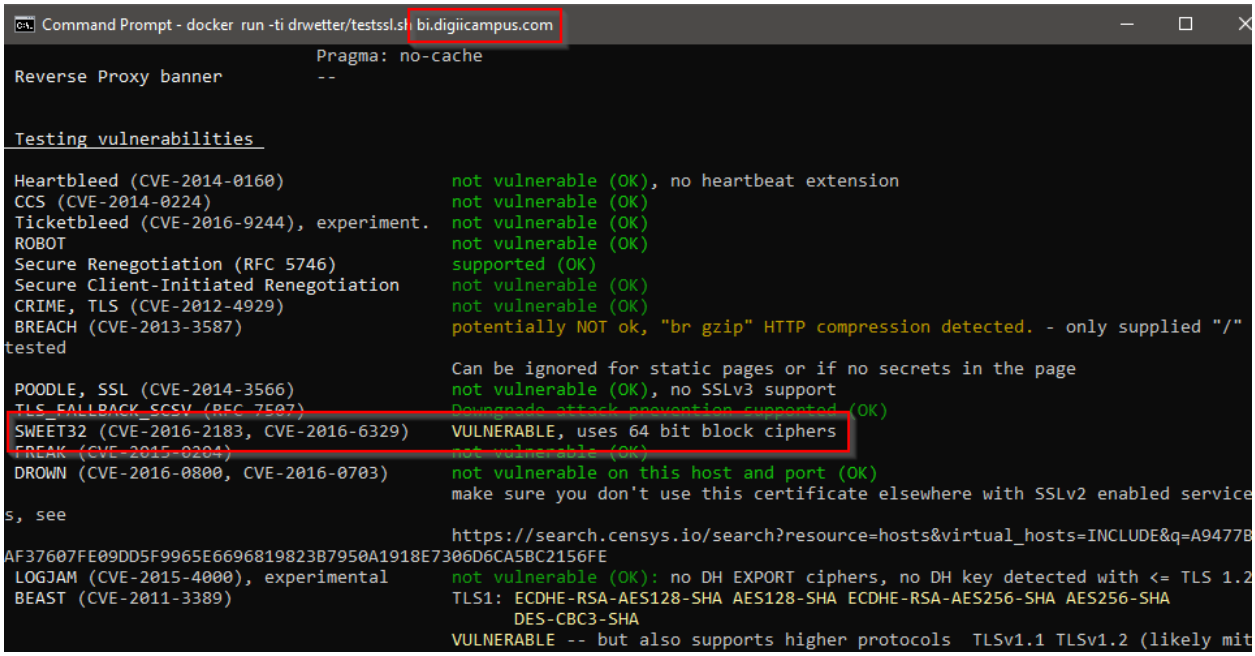


Retest Proof of Concept:

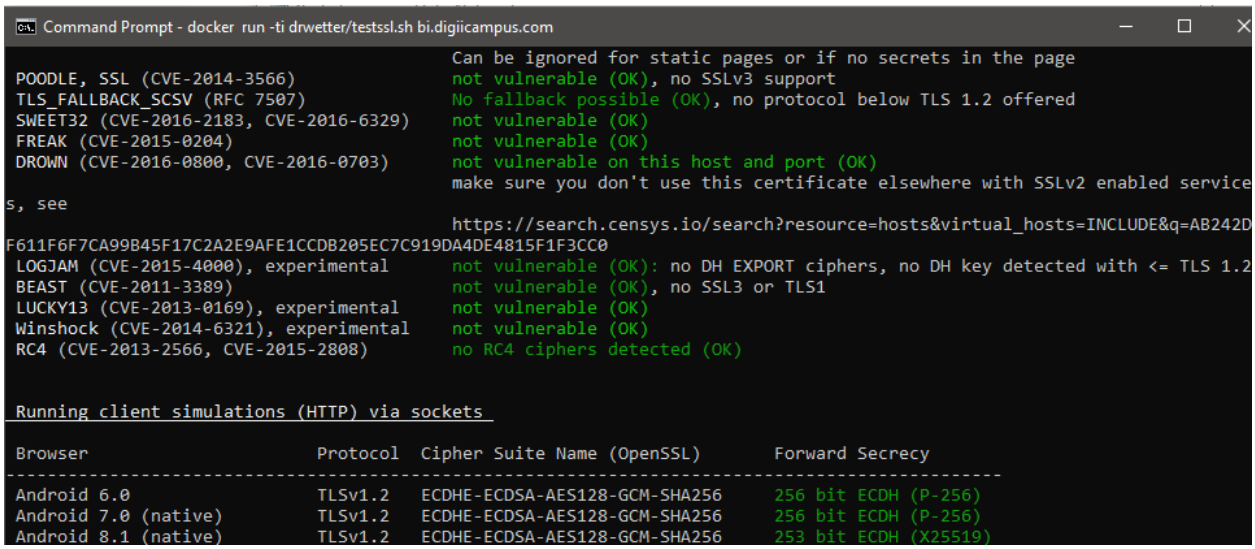


5. Application vulnerable to Sweet32 Attack	
Severity:	**MEDIUM**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com
Description:	The Sweet32 attack is an SSL/TLS vulnerability that allows attackers to compromise HTTPS connections using 64-bit block ciphers.
Recommendations:	Reconfigure the affected SSL/TLS server to disable support for obsolete 64-bit block ciphers.
References:	https://sweet32.info/ CWE 310: Cryptographic Issues
Step To Reproduce	Using test SSL scanner to observe this vulnerability

Proof of Concept:



Retest Proof of Concept:

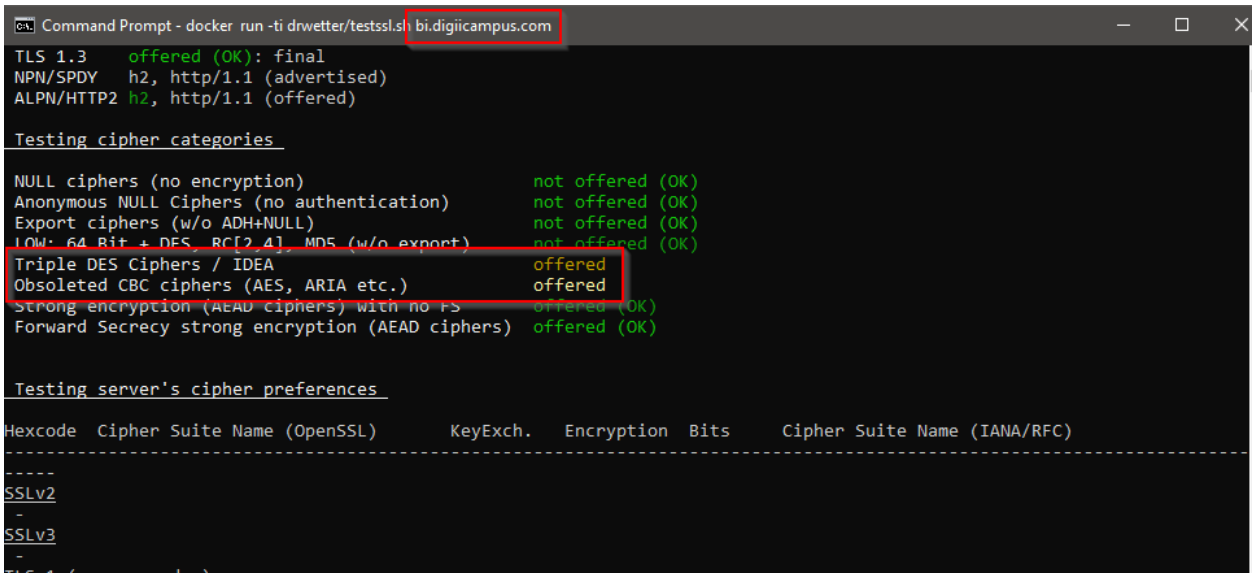


6. Local Adversary can hijack authenticated sessions due to High session In- activity period	
Severity:	**MEDIUM**
Retesting Status:	**BUSINESS USE CASE**
URL:	https://bi.digiicampus.com/
Description:	It was found that the session expiry limit is not set or is too long for the application. Even if the user session is inactive for a prolonged period after login, the application does not terminate the session automatically.
Impact:	A local adversary can hijack authenticated sessions due to a high session-inactivity period which can lead to gain access to the victim's account.
Recommendations:	The value you are setting in the timeout attribute is the one of the correct ways to set the session timeout value. The timeout attribute specifies the number of minutes a session can be idle before it is abandoned. The default value for this attribute is 20. The session management implementation defines the exchange mechanism that will be used between the user and the web application to share and continuously exchange the session ID. There are multiple mechanisms available in HTTP to maintain session state within web applications, such as cookies (standard HTTP header), URL parameters (URL rewriting – RFC2396), URL arguments on GET requests, body arguments on POST requests, such as hidden form fields (HTML forms), or proprietary HTTP headers.
References:	https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/06-Session_Management_Testing/07-Testing_Session_Timeout
Step To Reproduce	Step 1: Log in to the application and do not submit any request for a prolonged period of time. Step 2: It was observed that the application does not terminate the user session even after a prolonged period of inactivity (more than 30 minutes).

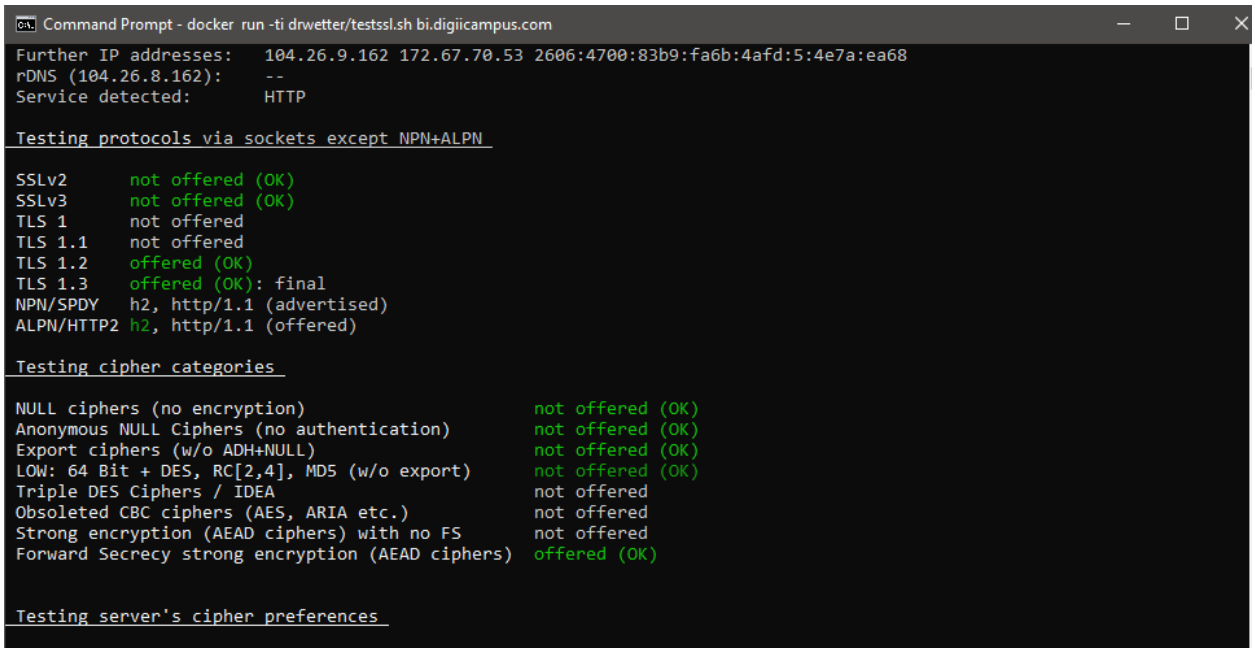
Proof of Concept: N/A

7. WEAK Cipher (3DES)/IDEA & Obsolete CBC ciphers supported	
Severity:	**MEDIUM**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com
Description:	➢ This server offers weak ciphers (3DES)/ IDEA. ➢ The remote host supports obsolete CBC ciphers. This may allow an attacker to recover the plain text message from the cipher text.
Recommendations:	➢ Disable weak ciphers (3DES, SEED, IDEA, RC2, RC4, MD5) on the server. ➢ Reconfigure the affected application to avoid use of CBC Ciphers. ➢ Standard cryptography packages such as SHA-256, RSA. GCM should be used. ➢ It's important to regularly review and ensure the security protocols and ciphers in use are up to date to mitigate potential vulnerabilities.
References:	https://www.geeksforgeeks.org/triple-des-3des/ https://dev.to/markyu/encrypting-with-block-ciphers-a-guide-to-aes-cbc-and-more-31gn https://medium.com/@coD3M/lucky-13-attack-explained-dd9a9fd42fa6
Step To Reproduce	Using test SSL scanner to observe this vulnerability

Proof of Concept:

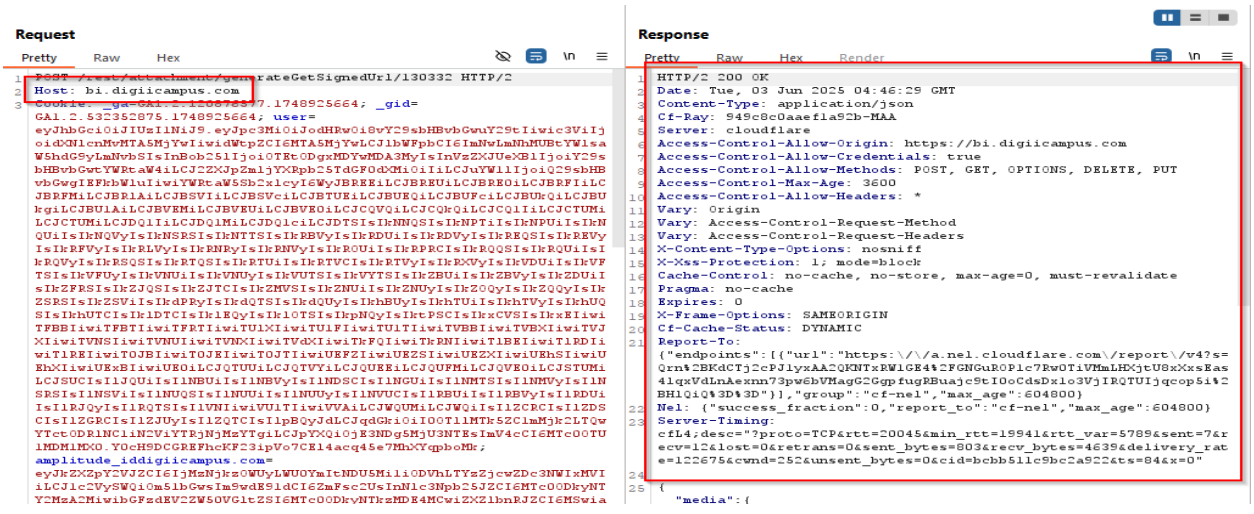


Retest Proof of Concept:

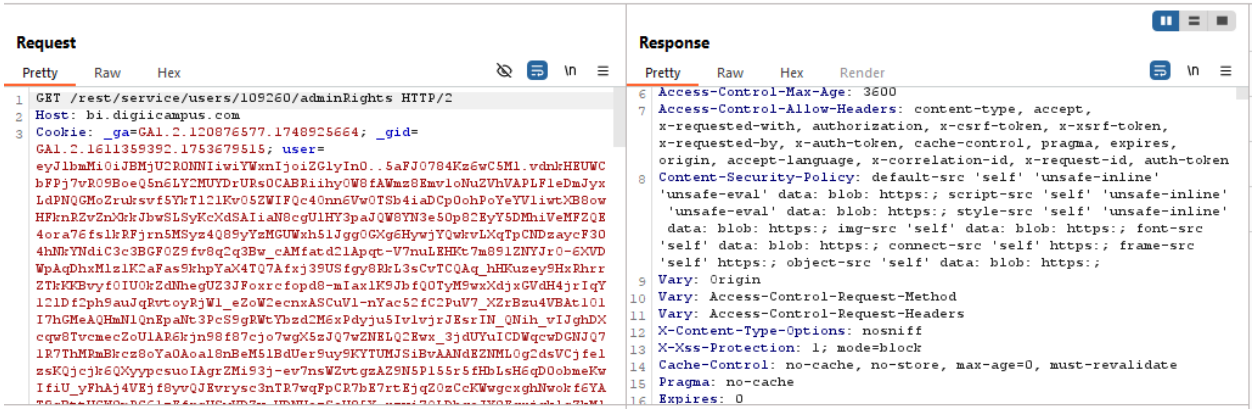


8. Content Security Policy header does not present	
Severity:	**LOW**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com
Description:	The vulnerability of a missing Content Security Policy (CSP) header refers to a security flaw where a web application fails to include the necessary CSP header in its HTTP responses. A Content Security Policy is a browser security feature that helps prevent various types of attacks, such as cross-site scripting (XSS) and data injection, by specifying which sources of content are considered legitimate. The absence of a CSP header leaves the application vulnerable to these attacks.
Impact:	Using this vulnerability, an attacker can: - • use this vulnerability to perform cross-site scripting. • perform clickjacking on the end users. • perform code injection attacks.
Recommendations:	Addressing the vulnerability of a missing CSP header requires immediate action. Developers should ensure that the application includes a proper CSP header with appropriate policies to restrict the sources of content and scripts that can be executed. Implementing a strong default-src policy and specifying other relevant directives can enhance security.
References:	https://content-security-policy.com/ https://web.dev/articles/strict-csp
Step To Reproduce	• In Burp Suite, capture any request and response • In response, we can see there is no Content Security Policy header present

Proof of Concept:



Retest Proof of Concept:



Request

Pretty Raw Hex

```
1 GET /rest/service/users/109260/adminRights HTTP/2
2 Host: bi.digiicampus.com
3 Cookie: _ga=GA1.2.120876577.1748925664; _gid=
GA1.2.1611359392.1753679515; user=
eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiZGlyIn0..5aFJ0784Kz6wC5M1.vdnkHEUWC
bFPj7vR09BoeQ5n6LY2MUYDrURs0CABRiihy0W8fAWmz8EmvloNuZVhVAPLFleDmJyx
LdPNQGMoZrukswf5YkTl2lKw05ZWIFQc40nn6Vw0TSb4iaDCp0ohPoYeYVliwtXB8ow
HFknRZwZnXrkJbwSLSyKcXdSAIiaN8cgUlHY3paJQW8YN3e50p82EyY5DMhiVeMFZQE
4ora76fslkRFjrn5MSyz4Q89yYzMGUWxh5lJgg0GXg6HywjYQwkvLXqTpCNDzaycF30
4hNkYNdiC3c3BGF0Z9fv8q2q3Bw_cAMfatd2lApqt-V7nuLEHKt7m891ZNYJr0-6XVD
WpAqDhxMlzlK2aFas9khpYaX4TQ7Afxj39USfgy8RkL3sCvTCQAq_hHKuzey9HxRhrr
ZTkKKBvyf0IU0kZdNhegUZ3JFoxrcfopd8-mIaxlKSJbfQOTyM9wxXdjxGVdH4jrIqY
121Df2ph9auJqRvtoyRjWl_eZoW2ecnxASCuVl-nYac52fC2PuV7_XZrBzu4VBAt101
I7hGMeAQHmNlQnEpaNt3PcS9gRWtYbzd2M6xPdyju5IvlvjrJEsrIN_QNih_vIJghDX
cqw8TvcmecZoUlAR6kjn98f87cjo7wgX5zJQ7wZNELQ2Ewx_3jdUYuICDWqcwDGNJQ7
lR7ThMRmBkcz8oYaOaal8nBeM5lBdUer9uy9KYTUMJSiBvAANdEZNML0g2dsVCjfel
zsKQjcjk6QXyypcsuoIAgrZMi93j-ev7nsWZvtgzAZ9N5P155r5fHbLsH6qD0obmeKw
IfiU_yFhAj4VEjf8yvQJEvrysc3nTR7wqFpCR7bE7rtEjqZ0zCcKWwgcxghNwokf6YA
```

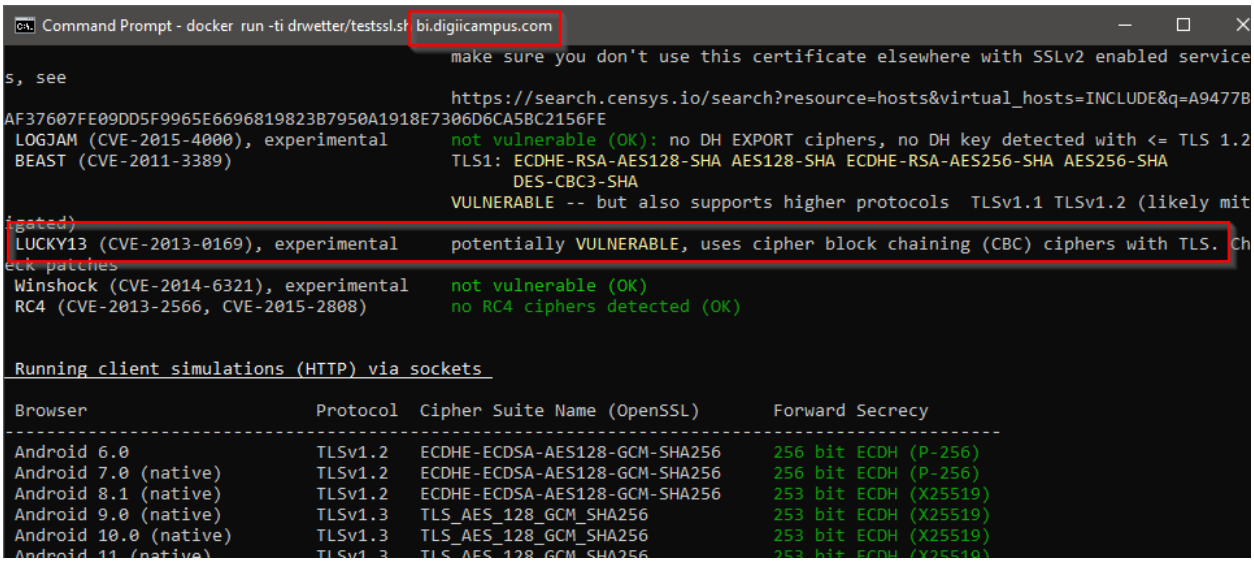
Response

Pretty Raw Hex Render

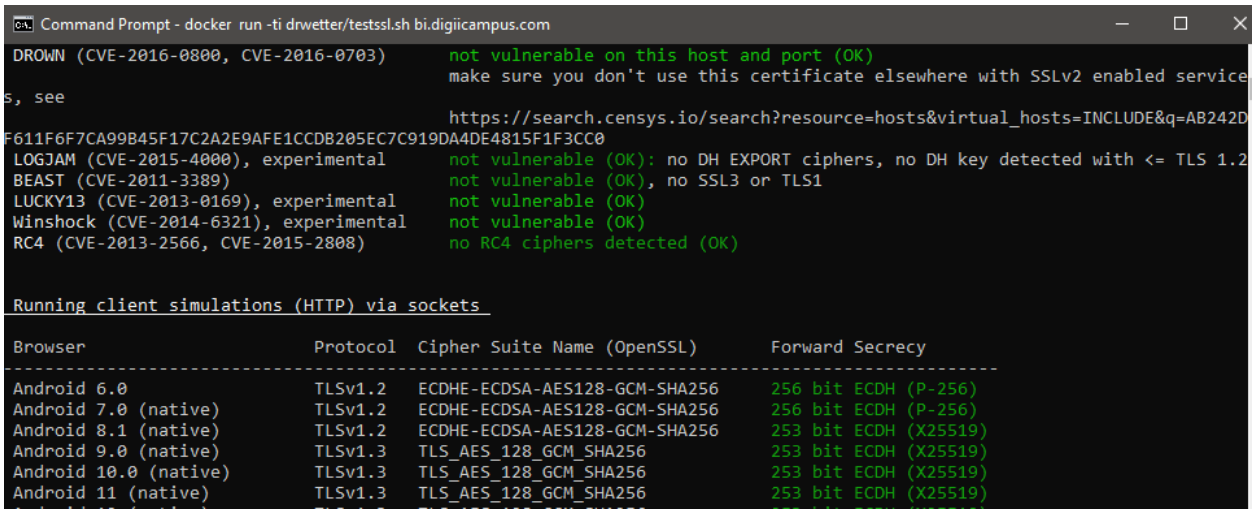
```
6 Access-Control-Max-Age: 3600
7 Access-Control-Allow-Headers: content-type, accept,
x-requested-with, authorization, x-csrf-token, x-xsrf-token,
x-requested-by, x-auth-token, cache-control, pragma, expires,
origin, accept-language, x-correlation-id, x-request-id, auth-token
8 Content-Security-Policy: default-src 'self' 'unsafe-inline'
'unsafe-eval' data: blob: https:; script-src 'self' 'unsafe-inline'
'unsafe-eval' data: blob: https:; style-src 'self' 'unsafe-inline'
data: blob: https:; img-src 'self' data: blob: https:; font-src
'self' data: blob: https:; connect-src 'self' https:; frame-src
'self' https:; object-src 'self' data: blob: https:;
9 Vary: Origin
10 Vary: Access-Control-Request-Method
11 Vary: Access-Control-Request-Headers
12 X-Content-Type-Options: nosniff
13 X-Xss-Protection: 1; mode=block
14 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
15 Pragma: no-cache
16 Expires: 0
```

9. Application is potentially vulnerable to LUCKY13 attack	
Severity:	**LOW**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com
Description:	This server offers ciphers with CBC mode of operation. Some implementations do not properly consider timing side-channel attack on a MAC check requirement. Please check if the version of the software is vulnerable to LUCKY13 or not. (CVE-2013-0169)
Impact:	The server has a vulnerability that may cause Lucky Thirteen attack against implementations of the Transport Layer Security protocol. This may result in the loss of sensitive information.
Recommendations:	Check the server version and update it to the latest version
References:	https://medium.com/@coD3M/lucky-13-attack-explained-dd9a9fd42fa6 https://cyberpedia.reasonlabs.com/EN/lucky13%20attack.html
Step To Reproduce	Using test SSL scanner to observe this vulnerability

Proof of Concept:

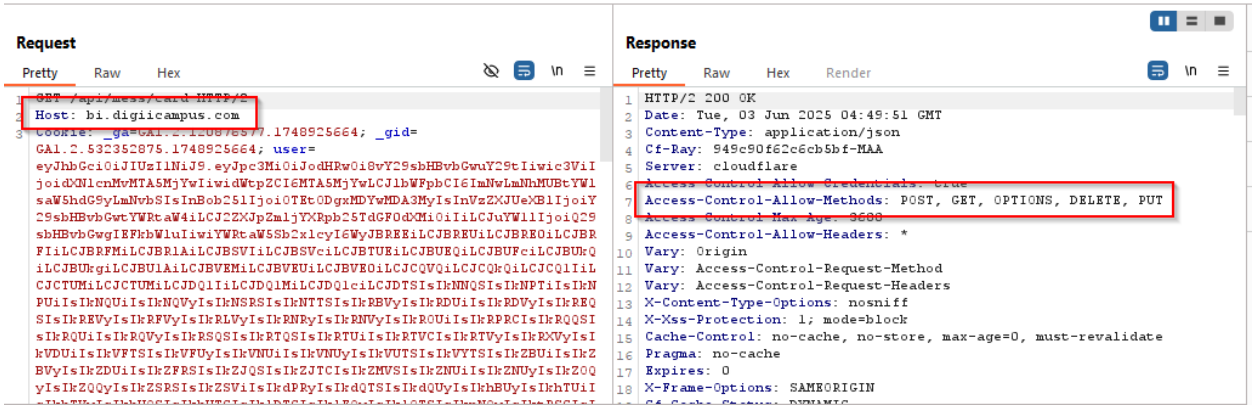


Retest Proof of Concept:

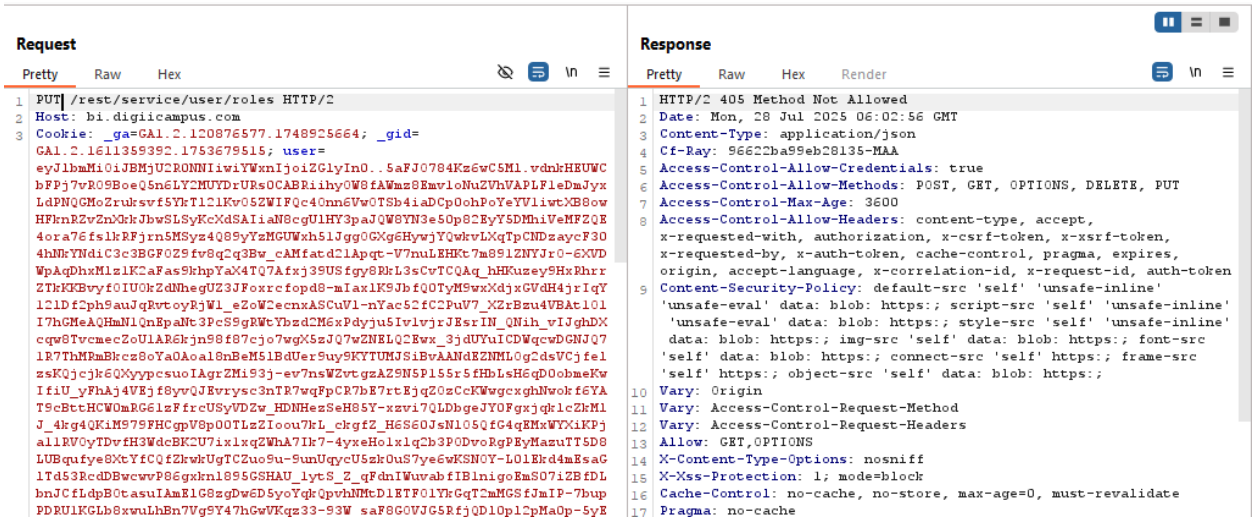


10. Arbitrary HTTP methods enabled	
Severity:	**LOW**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com
Description:	Insecure HTTP methods like PUT and DELETE are enabled on this web server. In the presence of other cross-domain vulnerabilities in web browsers, sensitive header information could be read from any domain that supports these methods. It represents a request for information about the communication options available on the request/response chain identified by the Request-URL.
Impact:	Due to this, Attacker can easily enumerate what methods are used by the application
Recommendations:	It is recommended to disable these insecure HTTP- PUT and DELETE methods on the web server
References:	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/06-Test_HTTP_Methods CWE-937: Using Components with Known Vulnerabilities
Step To Reproduce	• Navigate to application. • Intercept any request using a proxy tool. • You can see the methods being used by the application

Proof of Concept:



Retest Proof of Concept:



Request

Pretty Raw Hex

```
PUT /rest/service/user/roles HTTP/2
Host: bi.digiicampus.com
Cookie: _ga=GA1.2.120876577.1748925664; _gid=
GA1.2.1611359392.1753679515; user=
eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiZGlyIn0..5aFJ0784Kz6wC5Ml.vdnkHEUWC
bFPj7vR09BoeQ5n6LY2MUYDrURs0CABRiihy0W8fAWmz8EmvloNuZVhVAPLFleDmJyx
LdPNQGMoZruksvf5YkTl2lKv05ZWIFQc40nn6Vw0TSb4iaDCpOohPoYeYVliwtXB8ow
HFknRZvZnXkkJbwSLSyKcXdSAIiaN8cgU1HY3paJQW8YN3e50p82EyY5DMhiVeMFZQE
4ora76fslkRFjrn5MSyz4Q89yYzMGUWxh5lJgg0GXg6HywjYQwkvLXqTpCNDzaycF30
4hNkYNdiC3c3BGF0Z9fv8q2q3Bw_cAMfatd2lApqt-V7nuLEHKt7m891ZNYJr0-6XVD
WpAqDhxMlzlK2aFas9khpYaX4TQ7Afxj39USfgy8RkL3sCvTCQAq_hHKuzey9HxRhrr
ZTkKKBvyf0IU0kZdNhegUZ3JFoxrcfopd8-mIaxlK9JbfQOTyM9wxXdjxGVdH4jrIqY
l2lDf2ph9auJqRvtoyRjWl_eZoW2ecnxASCuVl-nYac52fC2PuV7_XZrBzu4VBAtl0l
I7hGMeAQHmNlQnEpaNt3PcS9gRWtYbzd2M6xPdyju5IvlvjrJEsrIN_QNih_vIJghDX
cqw8TvcmecZoUlAR6kjn98f87cjo7wgX5zJQ7wZNELQ2Ewx_3jdUYuICDWqcwDGNJQ7
lR7ThMRmBkcz8oYa0Aoal8nBeM5lBdUer9uy9KYTUMJSiBvAANdEZNML0g2dsVCjfel
zsKQjcjk6QXyypcsuoIAgrZMi93j-ev7nsWZvtgzAZ9N5P155r5fHbLsH6qD0obmeKw
IfiU_yFhAj4VEjf8yvQJEvrysc3nTR7wqFpCR7bE7rtEjqZ0zCcKWwgcxghNwokf6YA
T9cBttHCWOmRG6lzFfrcUSyVDZw_HDNHezSeH85Y-xzvi7QLDbgeJYOFgxjqklcZkMl
J_4kg4QKiM979FHCgpV8p00TLzZIoou7kL_ckgfZ_H6S60JsNl05QfG4qKMxWYXiKPj
allRV0yTDvfH3WdcBK2U7ixlxqZWhA7Ik7-4yxeHolxlq2b3P0DvoRgPEyMazuTT5D8
LUBqufye8XtYfCQfZkwkUgTCZuo9u-9unUqycU5zk0uS7ye6wKSN0Y-L0lEkd4mEsaG
lTd53RcdDBwcwvP86gxknl895GSHAU_lytS_Z_qFdnIWuvabfIBlnigoEmSO7iZBfDL
bnJCfLdpB0tasuIAmE1G8zgDw6D5yoYqkQpvhNMtDlETF0lYkGqT2mMGSfJmIP-7bup
PDRUlKGLb8xwuLhBn7Vg9Y47hGwVKqz33-93W saF8G0VJG5RfjQDl0pl2pMaOp-5yE
```

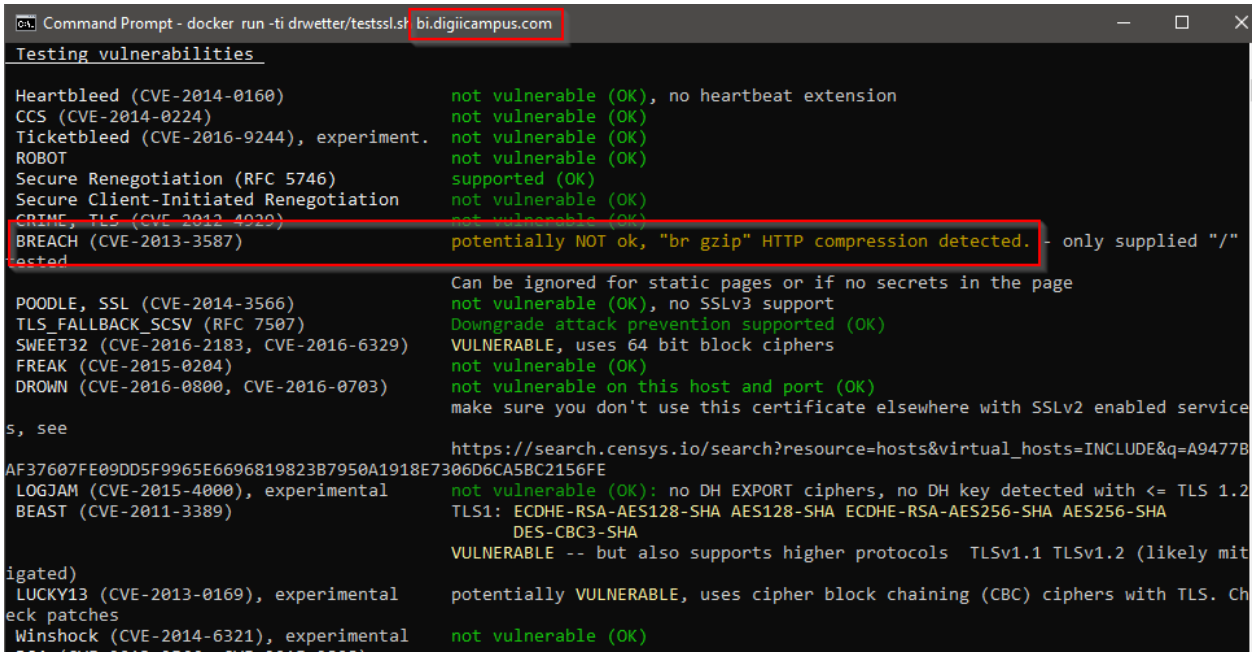
Response

Pretty Raw Hex Render

```
HTTP/2 405 Method Not Allowed
Date: Mon, 28 Jul 2025 06:02:56 GMT
Content-Type: application/json
Cf-Ray: 96622ba99eb28135-MAA
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
Access-Control-Max-Age: 3600
Access-Control-Allow-Headers: content-type, accept,
x-requested-with, authorization, x-csrf-token, x-xsrf-token,
x-requested-by, x-auth-token, cache-control, pragma, expires,
origin, accept-language, x-correlation-id, x-request-id, auth-token
Content-Security-Policy: default-src 'self' 'unsafe-inline'
'unsafe-eval' data: blob: https:; script-src 'self' 'unsafe-inline'
 'unsafe-eval' data: blob: https:; style-src 'self' 'unsafe-inline'
 data: blob: https:; img-src 'self' data: blob: https:; font-src
'self' data: blob: https:; connect-src 'self' https:; frame-src
'self' https:; object-src 'self' data: blob: https:;
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Allow: GET,OPTIONS
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
```

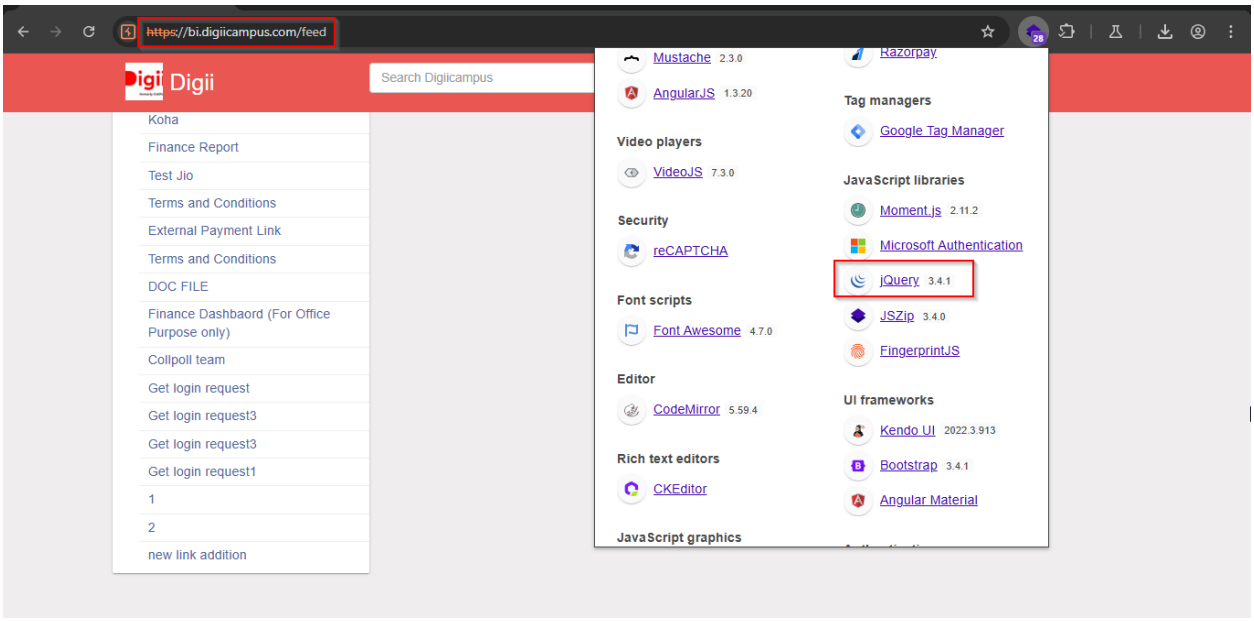
11. Application is potentially vulnerable to BREACH attack	
Severity:	**LOW**
Retesting Status:	**CLOSED**
URL:	bi.digiicampus.com
Description:	The BREACH (Browser Reconnaissance and Ex filtration via Adaptive Compression of Hypertext) attack is a cryptographic attack that targets web applications that use HTTP compression, such as GZIP, to improve page load times. The attack can allow an attacker to steal sensitive information, such as authentication tokens, from a user's session by exploiting the way compression works.
Impact:	The BREACH attack steals information about how data is encrypted from HTTPS-enabled Web applications by essentially combining two existing types of attacks: using cross-site request forgery (CSRF) to change data in transport, and injecting data into the HTTPS headers using a man-in-the-middle attack.
Recommendations:	It is recommended to implement random data padding to make the compression more unpredictable and updating TLS protocol in the application to a version that is not vulnerable to the attack.
References:	https://www.thesmartscanner.com/vulnerability-list/breach-attack
Step To Reproduce	Using test SSL scanner to observe this vulnerability

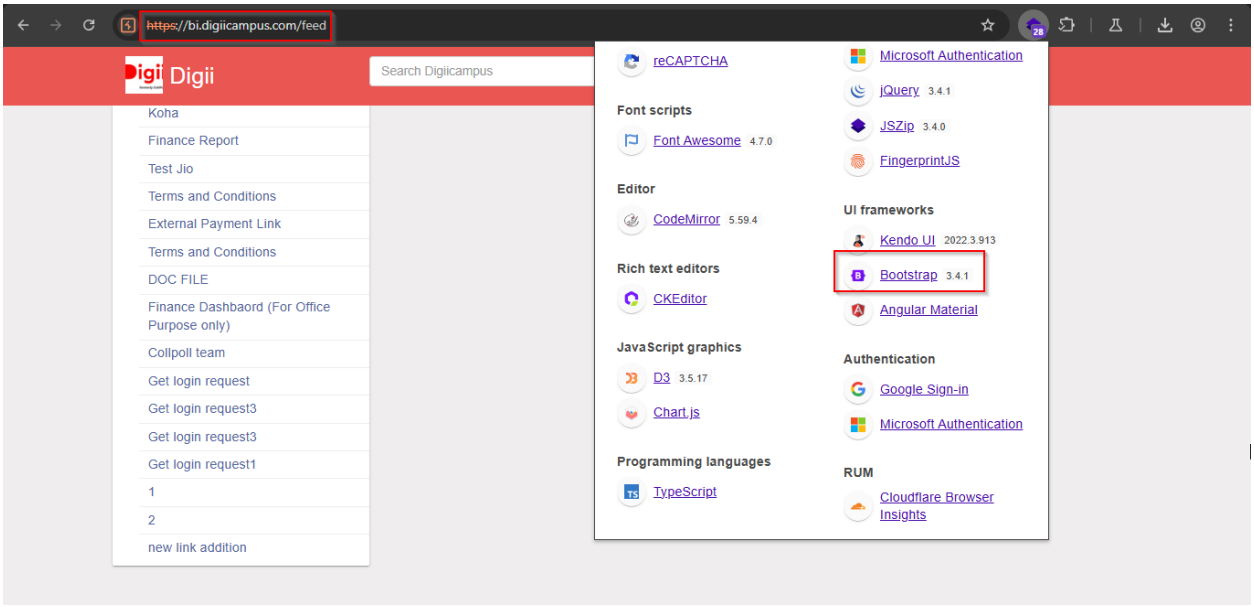
Proof of Concept:



12. Vulnerable version of Java Script Library Found	
Severity:	**LOW**
Retesting Status:	**OPEN**
URL:	bi.digiicampus.com/feed
Description:	The use of third-party JavaScript libraries can introduce a range of DOM-based vulnerabilities, including some that can be used to hijack user accounts like DOM-XSS. Common JavaScript libraries typically enjoy the benefit of being heavily audited. This may mean that bugs are quickly identified and patched upstream, resulting in a steady stream of security updates that need to be applied. Although it may be tempting to ignore updates, using a library with missing security patches can make your website exceptionally easy to exploit. Therefore, it's important to ensure that any available security updates are applied promptly. Some library vulnerabilities expose every application that imports the library, but others only affect applications that use certain library features. Accurately identifying which library vulnerabilities apply to your website can be difficult, so we recommend applying all available security updates regardless.
Impact:	JavaScript library's security vulnerabilities can be exploited to perform cross-site scripting, cross-site request forgery, and buffer overflow.
Recommendations:	Develop a patch-management strategy to ensure that security updates are promptly applied to all third-party libraries in your application. Also, consider reducing your attack surface by removing any libraries that are no longer in use.
References:	https://portswigger.net/kb/issues/00500080_vulnerable-javascript-dependency CWE-1104: Use of Unmaintained Third Party Components A9: Using Components with Known Vulnerabilities https://security.snyk.io/package/npm/jquery/3.4.1 https://security.snyk.io/package/npm/bootstrap/3.4.1
Step To Reproduce	• Navigate to application. • You can see the vulnerable version of java script is being used by the application

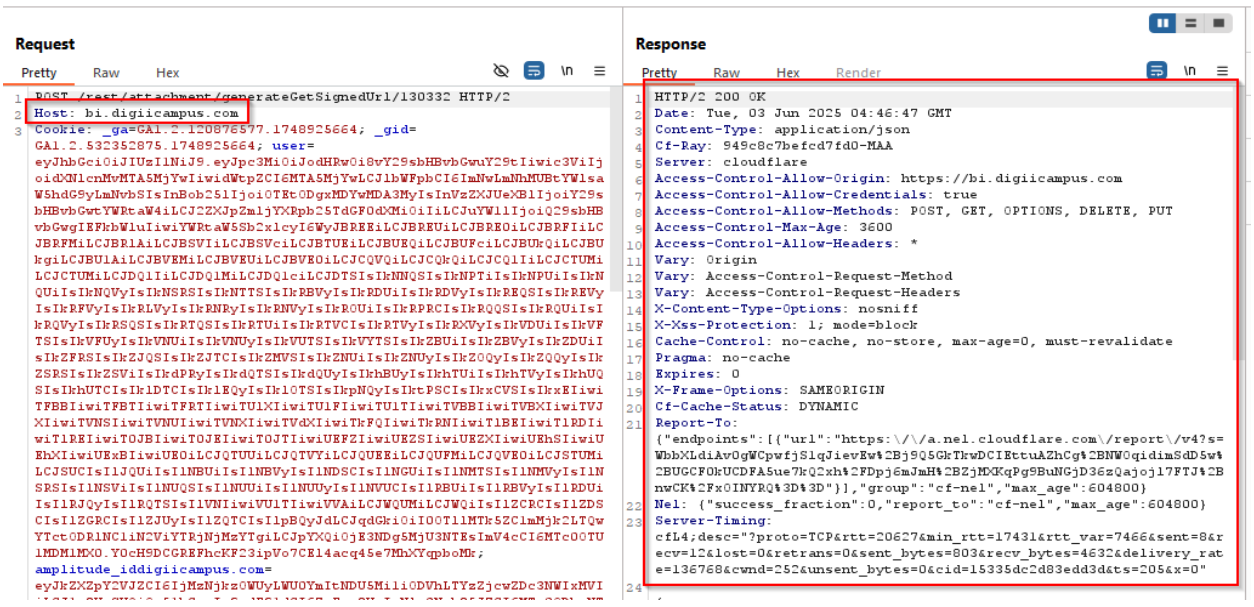
Proof of Concept:



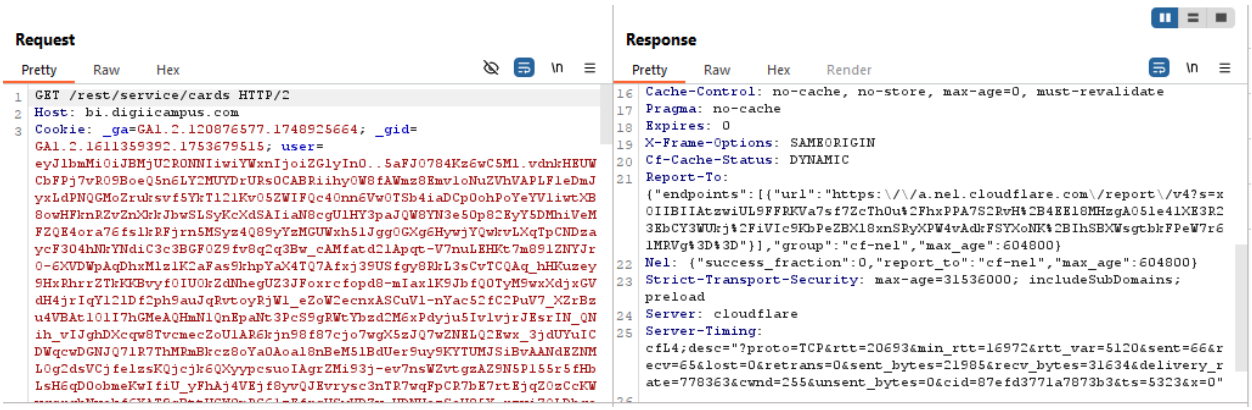


13. HTTP Strict Transport Security not configured in Application	
Severity:	**LOW**
Retesting Status:	**FIXED**
URL:	bi.digiicampus.com
Description:	The HTTP Strict Transport Security (HSTS) feature allows a web application to inform the browser, through the use of a special response header, that it should never establish a connection to the specified domain servers using HTTP. Instead, it should automatically establish all connection requests to access the site through HTTPS. This will prevent the information from being passed over an unencrypted channel.
Impact:	An adversary can access unencrypted traffic, thus by gaining access to the complete user account.
Recommendations:	• Set the HSTS directives in the response header. • This needs to be implemented throughout the application.
References:	https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html CWE-523: Unprotected Transport of Credentials CAPEC-94: Man in the Middle Attack CAPEC-157: Sniffing Attacks
Step To Reproduce	• Go to the application and intercept any request using proxy tool. • You can observe that the HSTS header is absent.

Proof of Concept:



Retest Proof of Concept:



Request

Pretty Raw Hex

```
GET /rest/service/cards HTTP/2
Host: bi.digiicampus.com
Cookie: _ga=GA1.2.120876577.1748925664; _gid=
GA1.2.1611359392.1753679515; user=
eyJlbmMi0iJBMjU2R0NNIiwiYWxnIjoiZGlyIn0..5aFJ0784Kz6wC5Ml.vdnkHEUW
CbFPj7vR09BoeQ5n6LY2MUYDrURs0CABRiihy0W8fAWmz8EmvloNuZVhVAPLFleDmJ
yxLdPNQGMoZruksvf5YkTl2lKv05ZWIFQc40nn6Vw0TSb4iaDCp0ohPoYeYVliwtXB
8owHFknRZvZnXrkJbwSLSyKcXdSAIiaN8cgUlHY3paJQW8YN3e50p8ZEyY5DMhiVeM
FZQE4ora76fslkRFjrn5MSyz4Q89yYzMGUWxh5lJgg0GXg6HywjYQwkvLXqTpCNDza
ycF304hNkYNdiC3c3BGF0Z9fv8q2q3Bw_cAMfatd21Apqt-V7nuLEHKt7m891ZNYJr
0-6XVDWpAqDhxMlzlK2aFas9khpYaX4TQ7Afxj3SUSfgy8RrL3sCvTCQAq_hHKuzey
9HxRhrrZTkKKBvyf0IU0kZdNhegUZ3JFoxrcfopd8-mIaxlK9JbfQ0TyMSwxXdjxGV
dH4jrIqY12lDf2ph9auJqRvtoyRjWl_eZoW2ecnxASCuVl-nYac52fC2PuV7_XZrBz
u4VBAt101I7hGMeAQHmN1QnEpaNt3PcS9gRWtYbzd2M6xPdyju5IvlvjrJEsrIN_QN
ih_vIJghDXcqw8TvcmecZoU1AR6kjn98f87cjo7wgX5zJQ7wZNELQ2Ewx_3jdUYuIC
DWqcwDGNJQ71R7ThMRmBkcz8oYaOaal8nBeM51BdUer9uy9KYTUMJSiBvAANdEZNM
L0g2dsVCjfelzsKQjcjk6QXyypcsuoIAgrZMi93j-ev7nsWZvtgzAZ9N5P155r5fHb
LsH6qDOobmeKwIfiU_yFhAj4VEjf8yvQJEvrysc3nTR7wqFpCR7bE7rtEjqZ0zCcKW
```

Response

Pretty Raw Hex Render

```
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: SAMEORIGIN
Cf-Cache-Status: DYNAMIC
Report-To:
{"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v4?s=x
0IIBIIAtzwiUL9FFRKVa7sf7ZcTh0u%2FhxPPA7S2RvH%2B4EE18MHzgA05le4lXE3R2
3EbCY3WUrj%2FiVIc9KbPeZBX18xnSRyXPW4vAdkFSYXoNK%2BIhSBXWsgtbkFPeW7r6
lMRVg%3D%3D"}],"group":"cf-nel","max_age":604800}
Nel: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Strict-Transport-Security: max-age=31536000; includeSubDomains;
preload
Server: cloudflare
Server-Timing:
cfL4;desc="?proto=TCP&rtt=20693&min_rtt=16972&rtt_var=5120&sent=66&r
ecv=65&lost=0&retrans=0&sent_bytes=21985&recv_bytes=31634&delivery_r
ate=778363&cwnd=255&unsent_bytes=0&cid=87efd3771a7873b3&ts=5323&x=0"
```

3. Conclusions

The above Application shows vulnerabilities, which earlier include High, Medium and Low vulnerabilities.

Hence, the testing and re-testing has been done for the web application Digiicampus.

4. Tools Used

- Burp Suite Licensed
- Nessus Licensed
- OWASP ZAP
- Kali Linux – Open Source trusted
- Nmap