



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	5
1.1.	Summary	5
1.2.	Approach	5
1.3.	Disclaimer	6
1.4.	Limitations.....	6
1.5.	OWASP TOP 10	6
1.6.	Scope.....	6
1.7.	Standards	7
1.8.	Vulnerability Scoring.....	8
1.9.	Key Findings	9
1.10.	Vulnerability Graph.....	10
2.	Findings	11
1.	Password guessing via Brute force attack	11
2.	Exploiting Metadata by File Upload vulnerability	14
3.	Broken Access Control via Parameter Tampering	17
4.	TLS Version 1.0 & 1.1 Protocol Detection	20
5.	Application vulnerable to Sweet32 Attack.....	22
6.	Local Adversary can hijack authenticated sessions due to High session In- activity period	24
7.	WEAK Cipher (3DES)/IDEA & Obsolete CBC ciphers supported.....	25
8.	Content Security Policy header does not present	27
9.	Application is potentially vulnerable to LUCKY13 attack	29
10.	Arbitrary HTTP methods enabled	31
11.	Application is potentially vulnerable to BREACH attack.....	33
12.	Vulnerable version of Java Script Library Found	34
13.	HTTP Strict Transport Security not configured in Application	36
3.	Conclusions	38
4.	Tools Used.....	38



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.

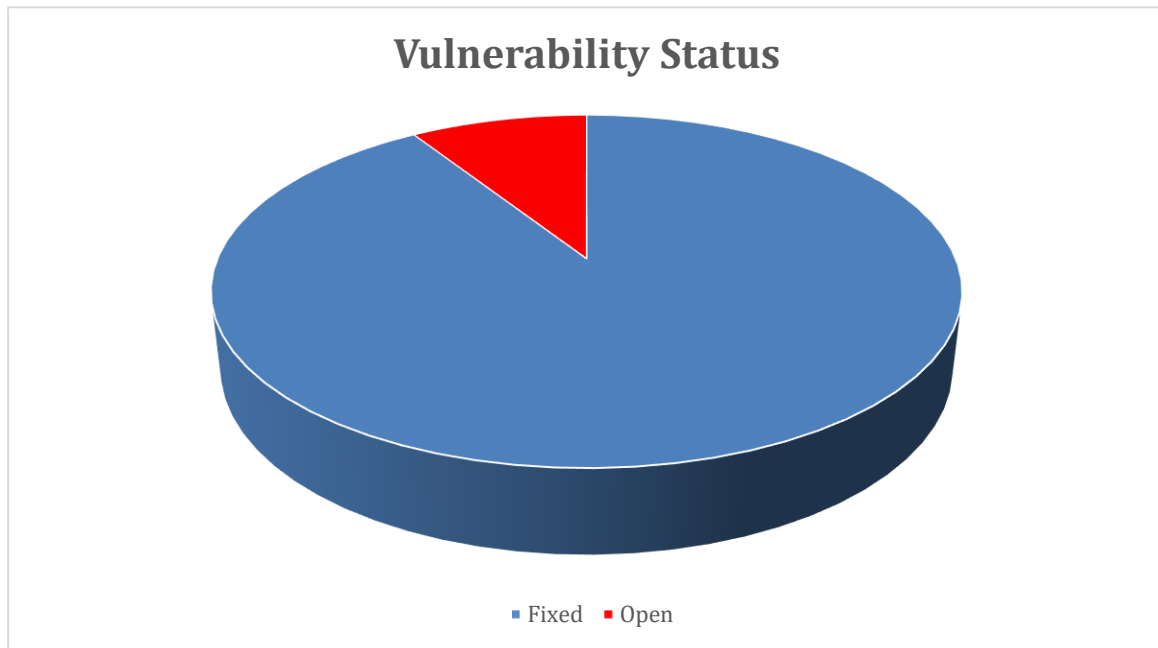
Fixed: The vulnerability has been successfully addressed, and the necessary steps have been taken to eliminate or remediate it. Appropriate measures, such as applying patches or updates, have been implemented.

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	430			1014	
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	

RequestResponse

PrettyRawHex

```
1 POST /rest/service/authenticate HTTP/2
2 Host: bi.digiicampus.com
3 Cookie: _ga=GA1.2.120876577.1748925664; _gid=GA1.2.532352875.1748925664; _gat=1; _ga_SBQJCK5NM7=GS2.2.s1748928889$o2$g0$t1748928889$j60$10$h0; amplitude_iddigiicampus.com=eyJkZXZpY2VJZCI6IjMzNjkzOWUyLWU0YmItNDUSMiliODVhLTZzZjcwZDc3NWlzMVViLlJlY2VzSWQiOm5lbGwsIm9wdE9ldCI6ZmFsc2UsInNlc3Npb25JZCI6MTc0ODkyODg4NTg4MwibGFzdEV2ZW50VGltZSI6MTc0ODkyODkxMDQ2MywiZXZlbnRlZCI6MiwiaWRlbnRlZnJlZCI6MCwic2VxdWUyY2V0dWliZXIiOjJ9
4 Content-Length: 241
5 Sec-Ch-Ua-Platform: "Windows"
6 Accept-Language: en-US,en;q=0.9
7 Accept: application/json, text/plain, */*
8 Sec-Ch-Ua: "Not.A/Brand";v="99", "Chromium";v="136"
9 Content-Type: application/json;charset=UTF-8
10 Sec-Ch-Ua-Mobile: ?0
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0
```

The screenshot shows the DigiCampus website interface. On the left is a sidebar with a user profile for SUPRAJAT KUMAR (14mce11780086, MS Software Engineering) and a list of navigation links: Home, Profile, Classroom, Academics, Course Registration, Research Management, Feedback, Examinations, and Online Exam. The main content area displays the 'Admission Process' status, which includes 'Form Submitted' (08 Dec 04:45 pm) and 'Fees Paid' (09 May 02:12 pm). Below this, there are buttons for 'Post', 'Ask question', and 'Add event'. A post by 'Butterfly Innovations' is visible, dated 26 May, with a link to their feed. The post content includes the DigiCampus logo and the URL 'demo.butterflyinnovations.co'. The post is attributed to 'Colpoll Admin' and has options to 'Upvote', 'Comment', and 'Share'.



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!40ya!W358	500	290			2030	
68	qazwsx	500	189			2040	
69	qwerty	500	295			2034	
70	roberto	500	157			2028	

RequestResponse

PrettyRawHexRender

22

Nel: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}

23

Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

24

Server: cloudflare

25

Server-Timing:

26

cfl4;desc="?proto=TCP&rtt=31183&min_rtt=17186&rtt_var=12747&sent=425&recv=395&lost=0&retrans=0&sent_bytes=136002&recv_bytes=113878&delivery_rate=1943149&wnd=257&unsent_bytes=0&cid=93ca950alc6b4bb&ts=2065&x=0"

27

{

"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:

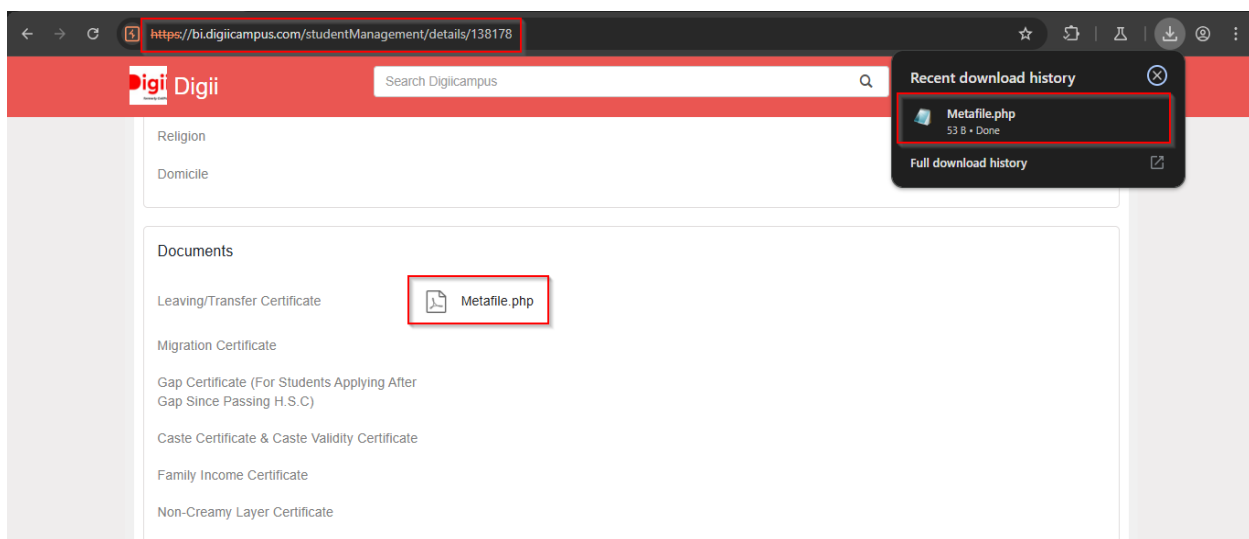
Request

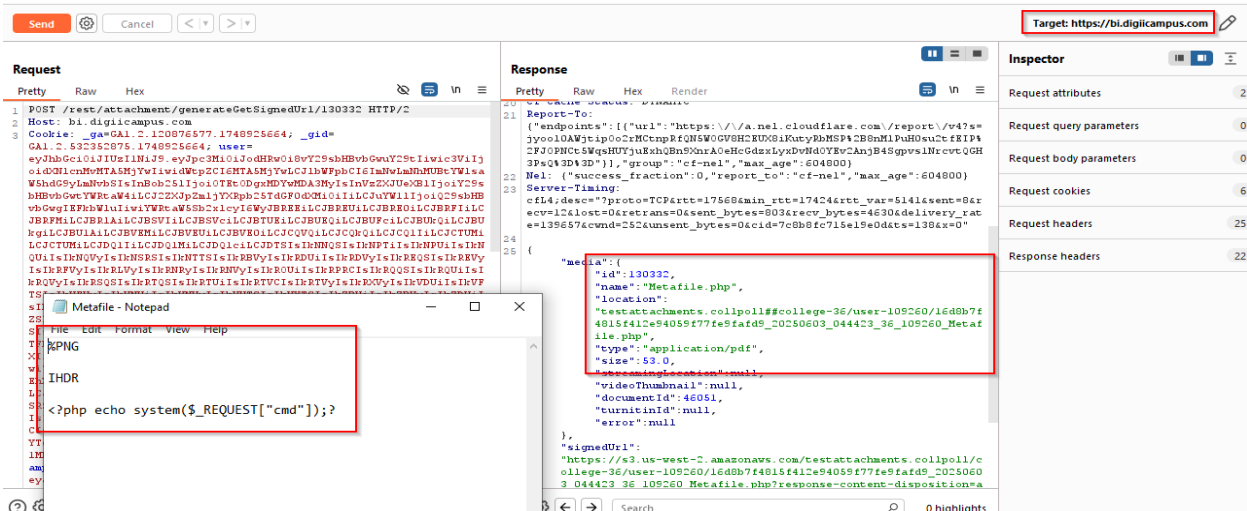
[illegible]

Request

```

Pretty      Raw      Hex
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
```





```
Request
Pretty Raw Hex
ZoW2eencASCuVl~nTAc5fIC2PuV7_XZrBnu4VBAt10Li7HmGAQHn1OnEpaNt3PeSSgPwTybzdCM6xPdYjzU5IvIvJrJEsrIN_QNhI_vIJghDXCqv8TvmeacZoUIARekjs98i87Gcj7o7y
q5SeJg7QwNEHLQ2Eww_3j4UUYiICDWqcWdQNJ071R7ThPRBkcs8oYa0Aoa18nBeM1SdUerSuY9KYTUJMJS1BvAAANdENMLQgdcdVCjfelasHCQjcjh6QXyppcxsuoIagr2NA923~ev7nsW2
vtgcAZ29SPl155r5fhBlhs6qdOobmeKwf1ia_Uy_yhJA4VEJ_f0yqVQEvryysc3ntF7wqFPcr7b7rt8jt8j20zcKQWwgqcgdnholmf6YATV9cButCHKOmRG6LefrxcUsyUDzw_HDNHesWeSH8Y~-x
zwi7QDlbgJePY7FGqjgq1ck2kMIJ_4kg4QKMS75FHCGvpV8p00TLz2Ioou7KL_ckgzE_HS6G0Jn10J5GFCqEMxWTYfKIPallRVQyTDvIH3WdcBKU27Jxlxxq2WhA7Ik7-4yxeholxlg2b3
PDovDoBgEPyMaauITSD5LDQuifye8XeYfcQfZkwUgTCZuo9a-SunUqycU5zkUrS7yeSwKN0Y-L0LEkd4mEsagLTd53ReddBvcuvP86gxkn189SGSHAU_lytS_2_qfDnIWuvahfIBlni
goEms07JZBfIdLnJClfdpBoatsu1AmELG8zgDw6DSyoYqGPvhNMtDIETFOlyYrGqTCzmGSfJaIP-7buypDRULIKGLb8xwulhBn7V9gSY47hcGVFKge33-93W_saf8G0VJG5rfjQD1Op12pM
a0p~SyEl-1_S-Uyte~seSVHNLT0clj_rKslVD9NB0CG--gpG80J9marVMotReOGbnToGRKRL1Cj6mRLUqE-LHS8iglnK5H-PXKS_zbe_AyPQod7cdUleWM4MOTyhNYBLiyb3Q
FhtOHMPQwXz2WSTFFRP6JsgK1ANGIXMJ9JoSaKaCw_CyeBoQ9x2CwRmEMUFsFWgaE44alEBT5TVgzdDntcvTo-AlXaiUPuY_5TES40wdiGABQ85vvTrIS75ReZPs-UGGealfJV7ryf8Da
3T0~dnPKfId7hdrfciGB18maKsH8BviricPT8xm_U8dsksreSMaVil3p32Ji08jFqicucvK8hwGbw5sEmKSDCS--fCTBzWmGaTA0meZ4HKHGeyerrOk39Lo414ke_kA7IrInN2_cuf531A
QCYZNM_Jkl55dYFvHm3_WCVfwe0OSRhWxyYA2SZPFVWnn5znYHF8F63y7hpMatxxczJAUUnJhVMSW0J6LvU7szc3XmasnT1L5JKUKUwi9Ak_ZBswgnT
QOL1AW36UVZ05A?EajwAnamAhoOCteojWAnsPpITHROTatwFikakSU_NlhkyK7BJdEdlincZokcveYecawWsj~yVB6JW46f5kakWR52litofetfaTRO9WIy6ZBNh4L19VSuwzbCBAGwD
Nqh2DAIGqChdpQomHyss9_Cufzi2UuaEfLUOAYtsAlReMy5G8SPvgvCb_IyTL83uqqGE4sb514WsqgFAVJAjGtcslT3TAhLCnawPWvFPChdJsamKcfco1BL2OUTH617j1DNHzDYUESwD
i7CwC5_eA3TYTP31itab7Se8EntFEKWhe3VCUl1q3jdxcc2XrwvVs_o_h5RI7TFqun-GapuuERhw
Sec-Ch-Uua-Mobile: 70
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
10 Accept: application/json, text/plain, */*
11 Content-Type: application/json;charset=UTF-8
12 Origin: https://bi.digii-campus.com/
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://bi.digii-campus.com/studentManagement/admin/edit/159812
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=1, i
19
20 {
21   "contentLength": "53",
    "fileName": "Metafile.php.pdf",
    "contentType": "application/pdf",
    "feature": "CLASSQUIZ",
    "isSynchronous": false,
    "shouldStream": false
  }
}
```

16



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.

[illegible]

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	

Confidential



Retest Proof of Concept:

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre>1 GET /users/profile/personaldetails HTTP/2 2 Host: bi.digicampus.com 3 Cookie: ga=GAL.2.120876577.1748925664; _gid= GAL.2.1611359392.1753679515; user= eyJlbmMiOiJBMjU2R0NNIiwiaWwiOiJ2GlyIn0..5aFJ0784Kz6wC5M1.vdmkHEUWC bFPj7vR09B0eQ5n6LY2HUYDRUR5OCABRIihyQW8fAWmz8Bmw1oNaZVhVAPLFleDmJyx LdPNQGMoZrksvfv5YkT12LKv0S2WIFQc40nn6Vw0TSb4iaDp0ohPoYeYVliwtXB8ow HfkmRZvZnGkHbW5LSyKcKdSAIiaN8cgULHY3paJQW8YN3e5Op82EYySDMhiVeMFZQE 4ora76fslkRFjrn5MSyz4Q89yYzMGUWxh51Jgg0GXg6HywYQwkvLXqTpCNDzaycF30 4hNkYNDiC3c3BGF0Z9fv8q2q3Bw_cAmfatd21Apqt-V7nuLEHkt7m8912NYJr0-6XVD WpAqDhxM1s1KC2aFas9khpYa4TQ7Afxj39USfgy8RkL3sCvTCQAq_hHKuzy9HxRhrr ZThKGBvyf0IU0kZdNhegUZ3JF0xrcfopd8-mIaxlK9Jb fQ0TyM9wxXdxjxGVdH4jrIqY 121Df2ph9auJqRvt0yRjW1_e2oW2ecnxASCuV1-nYac52f0C2PuV7_XZrBzu4VBat101 I7hGMeAQhANlQnEpaNt3PcS8gRwTbzdCM6xPdyju5Iv1vjrJEsrIN_QNih_v1JghDX cq98TvcmeCzoU1AR6krjn98f87cjo7wgX5zJQ7wZNElQ2Kwx_3jdUYuICDWqcvdGNJQ7 1R7ThMRmBkcz8oYa0aol8nBEM51BdUer9uy9KYTUMJ3iBvAANDZENML0g2dsVCjfel zsKQjckK6QXyypcsuoIagrZM193j-ev7nsWZvtgzAZ9N5P155r5fHbLsH6gD0obmeKw Ifiu_yPhA4VEj f8yvQJErvysc3nTr7wqFpCR7bE7rtEjq2QzCcKwgcxghNwokf6YA T9cBctHCW0mB661zfircUSyVDZw_HDNHesSeH85Y-xzvi7QLDbgeJY0Fgxjqlc2KML J_4kg4QKIm979PHGpV8p00TLzZi0ou7kL_ckgfZ_H6S60Jsn105QfG4qEMWYXiKPj a1lRV0yTdVrH3WdcBKCU7ix1kgZWhA7Ik7-4yxeHolx1q2b3p0DvoRgPEyManuTT5D8 LUBqufye8XtYfCQf2kwhUgTCZuoSu-9unUqycU5zk0uS7ye6wKSN0Y-L01Ek4mEsaG 1Td53RcdBwcvwP86gxmi1895GSHAU_lytS_Z_qFdnIWuvabfIBInigoEms07iZBfDL bnJCfLdp80tasu1AmE1G8zgDw6D5yoYqkQpvhNMtD1RTF01YkGqT2mMGSfJmIP-7bup PDRUIKGLb8xwulhBn7Vg947hGwVKqz33-93W_saF8G0VJG5RfjQD10p12pMaOp-5yE 1-_9-Uy7e-s5vHNTL0cj1_rKs1VD9Nb0C6--gP5G8Q19mRVH0TrE06bnoTxGRK81zYm gYGSk1Cj6mRLUqE-LH8gIgNsKSH-PXKS_zbe_AyPcQod7dcU16W4M0TyhNTBLlyb3QF htOHMPXww82yWSTFRFP6JzgXIAMGIXSJo8aXCw_CyeB0q9x2wRRmEMUFzFWgaE44af BT5Tvzxd4ntvrT0-A1XaiUPuY_STES40wdiGAB8Q5vvTrIS75kReZ2Ps-UGrealFJV7r yf8DaT3L0-dNPKLd7hdcFKX81Bm0xw5zMB0virHCY8mx_U8d5skrseMLxwi3pjZfJ0z jfQicwXsMGbw5s6XmKSDC5--f2TBxWkmGaTA0me24KGNcyerr0k39Lo414t_jkA7kri</pre>				<pre>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</pre>			



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:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com

Further IP addresses: 172.67.70.53 104.26.9.162 2606:4700:8399:fa6b:4aa8:496:4e7a:ea68
rDNS (104.26.8.162): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA             offered
Obsolete CBC ciphers (AES, ARIA etc.)  offered
Strong encryption (AEAD ciphers) with no FS offered (OK)
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

Testing server's cipher preferences
```



Retest Proof of Concept:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com
Further IP addresses: 104.26.9.162 172.67.70.53 2606:4700:83b9:fa6b:4afd:5:4e7a:ea68
rDNS (104.26.8.162): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      not offered
TLS 1.1    not offered
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA             not offered
Obsolete CBC ciphers (AES, ARIA etc.)  not offered
Strong encryption (AEAD ciphers) with no FS not offered
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

Testing server's cipher preferences
```



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:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com
Pragma: no-cache
Reverse Proxy banner
--

Testing vulnerabilities

Heartbleed (CVE-2014-0160)      not vulnerable (OK), no heartbeat extension
CCS (CVE-2014-0224)            not vulnerable (OK)
Ticketbleed (CVE-2016-9244), experiment. not vulnerable (OK)
ROBOT                          not vulnerable (OK)
Secure Renegotiation (RFC 5746) supported (OK)
Secure Client-Initiated Renegotiation not vulnerable (OK)
CRIME, TLS (CVE-2012-4929)     not vulnerable (OK)
BREACH (CVE-2013-3587)        potentially NOT ok, "br gzip" HTTP compression detected. - only supplied "/"
tested

POODLE, SSL (CVE-2014-3566)    Can be ignored for static pages or if no secrets in the page
                                not vulnerable (OK), no SSLv3 support
TLS_FALLBACK_SCSV (RFC 7507)  Downgrade attack prevention supported (OK)
SWEET32 (CVE-2016-2183, CVE-2016-6329) VULNERABLE, uses 64 bit block ciphers
CREAK (CVE-2015-0204)         not vulnerable (OK)
DROWN (CVE-2016-0800, CVE-2016-0703) not vulnerable on this host and port (OK)
                                make sure you don't use this certificate elsewhere with SSLv2 enabled service
s, see
                                https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=A9477B
AF37607FE09DD5F9965E6696819823B7950A1918E7306D6CA5BC2156FE
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3389)         TLS1: ECDHE-RSA-AES128-SHA AES128-SHA ECDHE-RSA-AES256-SHA AES256-SHA
                                DES-CBC3-SHA
                                VULNERABLE -- but also supports higher protocols TLSv1.1 TLSv1.2 (likely mit
```



Retest Proof of Concept:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digijcampus.com

POODLE, SSL (CVE-2014-3566)      Can be ignored for static pages or if no secrets in the page
                                not vulnerable (OK), no SSLv3 support
TLS_FALLBACK_SCSV (RFC 7507)   No fallback possible (OK), no protocol below TLS 1.2 offered
SWEET32 (CVE-2016-2183, CVE-2016-6329) not vulnerable (OK)
FREAK (CVE-2015-0204)          not vulnerable (OK)
DROWN (CVE-2016-0800, CVE-2016-0703) not vulnerable on this host and port (OK)
                                make sure you don't use this certificate elsewhere with SSLv2 enabled service
s, see                          https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=AB242D
F611F6F7CA99B45F17C2A2E9AFE1CCDB205EC7C919DA4DE4815F1F3CC0
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3389)          not vulnerable (OK), no SSL3 or TLS1
LUCKY13 (CVE-2013-0169), experimental not vulnerable (OK)
Winshock (CVE-2014-6321), experimental not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)

Running client simulations (HTTP) via sockets

Browser      Protocol  Cipher Suite Name (OpenSSL)  Forward Secrecy
-----
Android 6.0   TLSv1.2   ECDHE-ECDSA-AES128-GCM-SHA256  256 bit ECDH (P-256)
Android 7.0 (native) TLSv1.2   ECDHE-ECDSA-AES128-GCM-SHA256  256 bit ECDH (P-256)
Android 8.1 (native) TLSv1.2   ECDHE-ECDSA-AES128-GCM-SHA256  253 bit ECDH (X25519)
```



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:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com
TLS 1.3 offered (OK): final
NPN/SPDY h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories
NULL ciphers (no encryption) not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL) not offered (OK)
LOW: 64 Bit + DES, RC2, 41, MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA offered
Obsoleted CBC ciphers (AES, ARIA etc.) offered
Strong encryption (AEAD ciphers) with no FS offered (OK)
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

Testing server's cipher preferences
Hexcode Cipher Suite Name (OpenSSL) KeyExch. Encryption Bits Cipher Suite Name (IANA/RFC)
-----
SSLv2
-
SSLv3
-
TLSv1
-
TLSv1.1
-
TLSv1.2
-
TLSv1.3
-

```



Retest Proof of Concept:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com
Further IP addresses: 104.26.9.162 172.67.70.53 2606:4700:83b9:fa6b:4afd:5:4e7a:ea68
rDNS (104.26.8.162): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      not offered
TLS 1.1    not offered
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)          not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA              not offered
Obsoleted CBC ciphers (AES, ARIA etc.) not offered
Strong encryption (AEAD ciphers) with no FS not offered
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

Testing server's cipher preferences
```



Proof of Concept:



Retest Proof of Concept:

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	GET /rest/service/users/109260/adminRights HTTP/2			6	Access-Control-Max-Age: 3600		
2	Host: bi.digicampus.com			7	Access-Control-Allow-Headers: content-type, accept,		
3	Cookie: _ga=GAL.2.120876577.1748925664; _gid=GAL.2.1611359392.1753679515; user=			8	x-requested-with, authorization, x-csrf-token, x-xsrf-token,		
4	eyJ3bWl0iGJEMjUwRmN1aWVWmIjoizGlyIn0.5aFQ7084KzcvC5M1.vdnkHEUWC			9	x-requested-by, x-auth-token, cache-control, pragma, expires,		
5	bFp37r09S0bQ5nELI3CMUVDU95OCABRIhlyW9f4aWm8Emvln0uZ7VAPLpEdmJyx			10	origin, accept-language, x-correlation-id, x-request-id, auth-token		
6	LdAPQCMoZruksv5fKt121Kv052WPfQc40nmv5v0TSb44dCP00ph0PvEYt1wvXB5o6			Content-Security-Policy: default-src 'self' 'unsafe-inline'			
7	HPfkRzVzn0kDk7bW5LSyKxk4A81An0cGUH73pajQW6YhS50p8E8y2YDNDH4VeMF2Q6			'unsafe-eval' data: blob: https; script-src 'self' 'unsafe-inline'			
8	4ora76is13fK3jrh5MSYz4059YzMGUWkhq51JggQcG6Y9h7YQwkvLkqTPCNDzaycF30			'unsafe-eval' data: blob: https; style-src 'self' 'unsafe-inline'			
9	WpNkYnd4ic3z3BG0cF25fv8qz3Q3Bv_cAMfAt2d1Apqz-W7mLHHK7m091L2NYrXo-RXV			data: blob: https; img-src 'self' data: blob: https; font-src			
10	4u4qgqMz13LKc4a5hkpYw4QT7Q7a3735USfgy8KdL3svCvTC0Aq_hLkHkusey5Ds0hdd			'self' data: blob: https; connect-src 'self' https; frame-src			
11	ZThKKBvyt0iU0kZdH6eU233F0rcfopd8-m1ax1K93bv40DTY9vSxdk3cGV4H4j1rQy			'self' https; object-src 'self' data: blob: https;			
12	121d12ph5su4qBrt0yR3JL_e2wVccnxA5CwU1-nrAc5fC2PuV7_XZrBnu4VBA1o1			5	Vary: Origin		
13	I7hGHeaQHMt10nEpAmE8gYwBsdCMeXpdyjsi1v1jvr3rJEIH_0n4h_v1jghDX			6	Vary: Access-Control-Request-Method		
14	k7r7ThMPmBkzcsoYa0Aoa18nB6bcj7c07wg5x3Q7vZNELQgCd5VCjfel			7	Vary: Access-Control-Request-Headers		
15	zskKQcjK6QXyppesuo1agrZm193;-ev7m5WZvrgyGAZSN5P155r55HbLH6gQ00bneKv			8	X-Content-Type-Options: nosniff		
16	I4iUyJyFAj4Vj7f8yVqJ8vrryc3R7rqfPCR7bE7rEgQ0zcKcWgxcghWookf6EYA			9	X-Xss-Protection: 1; mode=block		
				10	Cache-Control: no-cache, no-store, max-age=0, must-revalidate		
				11	Pragma: no-cache		
				12	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-dd9agfd42fa6 https://cyberpedia.reasonlabs.com/EN/lucky13%20attack.html
Step To Reproduce	Using test SSL scanner to observe this vulnerability

Proof of Concept:

```
Ca: Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com
make sure you don't use this certificate elsewhere with SSLv2 enabled service
s, see
https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=A9477B
AF37607FE09DD5F9965E6696819823B7950A1918E7306D6CA5BC2156FE
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3389) TLS1: ECDHE-RSA-AES128-SHA AES128-SHA ECDHE-RSA-AES256-SHA AES256-SHA
DES-CBC3-SHA
VULNERABLE -- but also supports higher protocols TLSv1.1 TLSv1.2 (likely mit
igated)
LUCKY13 (CVE-2013-0169), experimental potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Ch
eck patches
Winshock (CVE-2014-6321), experimental not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)

Running client simulations (HTTP) via sockets

Browser Protocol Cipher Suite Name (OpenSSL) Forward Secrecy
-----
Android 6.0 TLSv1.2 ECDHE-ECDSA-AES128-GCM-SHA256 256 bit ECDH (P-256)
Android 7.0 (native) TLSv1.2 ECDHE-ECDSA-AES128-GCM-SHA256 256 bit ECDH (P-256)
Android 8.1 (native) TLSv1.2 ECDHE-ECDSA-AES128-GCM-SHA256 253 bit ECDH (X25519)
Android 9.0 (native) TLSv1.3 TLS_AES_128_GCM_SHA256 253 bit ECDH (X25519)
Android 10.0 (native) TLSv1.3 TLS_AES_128_GCM_SHA256 253 bit ECDH (X25519)
Android 11 (native) TLSv1.3 TLS_AES_128_GCM_SHA256 253 bit ECDH (X25519)
```



Retest Proof of Concept:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com
DROWN (CVE-2016-0800, CVE-2016-0703) not vulnerable on this host and port (OK)
make sure you don't use this certificate elsewhere with SSLv2 enabled service
s, see https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=AB242D
F611F6F7CA99B45F17C2A2E9AFE1CCDB205EC7C919DA4DE4815F1F3CC0
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3389) not vulnerable (OK), no SSL3 or TLS1
LUCKY13 (CVE-2013-0169), experimental not vulnerable (OK)
Winshock (CVE-2014-6321), experimental not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)

Running client simulations (HTTP) via sockets

```

Browser	Protocol	Cipher Suite Name (OpenSSL)	Forward Secrecy
Android 6.0	TLSv1.2	ECDHE-ECDSA-AES128-GCM-SHA256	256 bit ECDH (P-256)
Android 7.0 (native)	TLSv1.2	ECDHE-ECDSA-AES128-GCM-SHA256	256 bit ECDH (P-256)
Android 8.1 (native)	TLSv1.2	ECDHE-ECDSA-AES128-GCM-SHA256	253 bit ECDH (X25519)
Android 9.0 (native)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH (X25519)
Android 10.0 (native)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH (X25519)
Android 11 (native)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH (X25519)

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:

[illegible]



Retest Proof of Concept:

Request		Response	
Pretty	Raw Hex	Pretty	Raw Hex Render
<pre>1 PUT /rest/service/user/roles HTTP/2 2 Host: bi.digiicampus.com 3 Cookie: _ga=GAL.2.120876577.1748925664; _gid= GAL.2.1611359392.1753679515; user= eyJlbmMiOiJBMjU2R0NNIiwiaWwiOiJ2GlyIn0..5aFJ0784Kz6wCSM1.vdnkHEUWC bFPj7vR09B0eQSn6LY2MUTdURsOCABR1ihy0W8fAWms8Emv1oNuZVhVAPLFleDmJyx LdPNQGH0Zrurksvf5YhT121Kw05ZW1FQc40m6Vw0TSb4iaDcP0ohPoYeYV1iwtXB8ow HFknR2vZmGkHJbSLSyKcXkSAIiaN8cgU1HY3paJQW8YN3e50p82ByY5DMhiVeMFZQE 4ora76fs1kRFjrn5MSyz4Q89yYzMGUWxh51JggOGG6HyvYQwkvLXqTpCNDzaycF30 4hNkYNDiC3c3BGF0Z9fv8q2q3Bw_cAMfatd21Apqt-V7nuLEHKt7m891ZNYJr0-6XVD WpAqDhxM1z1KCaFas9khpYaX4TQ7Afxj39USfgy8RkL3sCvTCQAq_hHKuzey5HxPhrr ZThKGbvyf0IU0kZdNhegU23JFoxrcfopd8-mIaxlK9JbfQ0TyM9wxXdjxGVdH4jrIqY 121d f2ph9auJqRvt0yRjW1_eZoW2ecnXASCuV1-nYac52fC2PuV7_XZrBzu4VBAt101 I7hGMeAQHmN1QnEpaNt3PcS9gRWtYbzd2M6xPdyju5Iv1vjxJEsrIN_QNih_vIJghDX cqvs8TvcmeCZoU1AR6k9n8f87cjo7wgXszJQ7wZNElQ2Ewx_3jdUYuICDWqcwDGNJQ7 1R7ThMmBkcz8oYa0aol8nBeM51BdUer9uy9KYTUMJSiBvAANDREZNMLOg2dsVCjfel zskQjck6QYypcsuoIAGRZMi83j-ev7nsWZrtgzAZSN5P155r5fHbLsH6qd0ohmeKw IfiU_yPhaj4VEjfs9vQJFvrysc3nTR7wqPpCR7bE7rtEjgqZ0zCcKWgexghNwokf6YA TSeBtHCW0mRG61zFrcUSyVDZw_HDNHesSeH85Y-xzvi7QLDbgeJY0Fgxjgk1cZkM1 J_4kg4QKiM979FHCgpV8p00TLzZIoou7kL_ckgfZ_H6S60JsN105qfG4qEHxWYxiKPj al1RV0yTdvh3WdcBK2U71x1xqZWhA7Ik7-4yxeHolx1q2b3P0DvoRgPEyMazuITSD8 LUBqufye8XtYfCQfZkwhUgTCZuo9u-9unUqycU5zkOuS7ye6wKSN0Y-LO1Ek4mEsaG 1Td53RcdBwcvvP86gxkn1895GSHAU_lytS_2_qFdnIWuvabfIBInigoEmS07iZBfDL bnJCfLdpB0casuIAmE1G8zgdw6D5yoYqkQpvhNMtD1ETP01YkGqT2mMGSfJmIP-7bup PDRU1KGLb8xvLhBn7Vg9Y47hGvKqz33-93W saF8G0VJG5RfjQD10p12pMaOp-SyE</pre>		<pre>1 HTTP/2 405 Method Not Allowed 2 Date: Mon, 28 Jul 2025 06:02:56 GMT 3 Content-Type: application/json 4 Cf-Ray: 96622ba99eb28135-MAA 5 Access-Control-Allow-Credentials: true 6 Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT 7 Access-Control-Max-Age: 3600 8 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 9 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;; 10 Vary: Origin 11 Vary: Access-Control-Request-Method 12 Vary: Access-Control-Request-Headers 13 Allow: GET, OPTIONS 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</pre>	



11. Application is potentially vulnerable to BREACH attack	
Severity:	LOW
Retesting Status:	CLOSED
URL:	bi.digiicampus.com
Description:	The BREACH (Browser Reconnaissance and Exfiltration 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:

```
Command Prompt - docker run -ti drwetter/testssl.sh bi.digiicampus.com

Testing vulnerabilities

Heartbleed (CVE-2014-0160)      not vulnerable (OK), no heartbeat extension
CCS (CVE-2014-0224)            not vulnerable (OK)
Ticketbleed (CVE-2016-9244), experiment. not vulnerable (OK)
ROBOT                          not vulnerable (OK)
Secure Renegotiation (RFC 5746) supported (OK)
Secure Client-Initiated Renegotiation not vulnerable (OK)
CRIME, TLS (CVE-2012-4020)     not vulnerable (OK)
BREACH (CVE-2013-3587)        potentially NOT ok, "br gzip" HTTP compression detected. - only supplied "/"
tested

POODLE, SSL (CVE-2014-3566)    Can be ignored for static pages or if no secrets in the page
TLS_FALLBACK_SCSV (RFC 7507)  not vulnerable (OK), no SSLv3 support
SWEET32 (CVE-2016-2183, CVE-2016-6329) Downgrade attack prevention supported (OK)
FREAK (CVE-2015-0204)         VULNERABLE, uses 64 bit block ciphers
DROWN (CVE-2016-0800, CVE-2016-0703) not vulnerable (OK)
                                not vulnerable on this host and port (OK)
                                make sure you don't use this certificate elsewhere with SSLv2 enabled service
s, see
                                https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=A9477B
AF37607FE09DD5F9965E6696819823B7950A1918E7306D6CA5BC2156FE
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3389)         TLS1: ECDHE-RSA-AES128-SHA AES128-SHA ECDHE-RSA-AES256-SHA AES256-SHA
                                DES-CBC3-SHA
                                VULNERABLE -- but also supports higher protocols TLSv1.1 TLSv1.2 (likely mit
igated)
LUCKY13 (CVE-2013-0169), experimental potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Ch
eck patches
Winshock (CVE-2014-6321), experimental not vulnerable (OK)
```



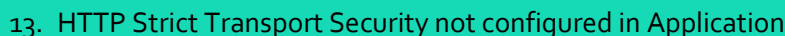
12. Vulnerable version of Java Script Library Found	
Severity:	LOW
Retesting Status:	OPEN
URL:	bi.digiicampus.com/feed
Description:	The use of third-party JavaScript libraries can introduce a range of DOM-based vulnerabilities, including some that can be used to hijack user accounts like DOM-XSS. Common 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 Ag: 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:

The image displays two screenshots of a web browser showing the DigiCampus website. The browser's address bar shows the URL <https://bi.digicampus.com/feed>. The website has a red header with the DigiCampus logo and a search bar. A sidebar menu on the left lists various links, including Koha, Finance Report, Test Jio, Terms and Conditions, External Payment Link, DOC FILE, Finance Dashbaord (For Office Purpose only), Collpoll team, Get login request, Get login request3, Get login request3, Get login request1, 1, 2, and new link addition. The main content area displays a list of various web technologies and frameworks, categorized into sections like Video players, Security, Font scripts, Editor, Rich text editors, JavaScript graphics, Tag managers, JavaScript libraries, UI frameworks, Authentication, and RUM. The following table lists the technologies and frameworks shown in the screenshots:

Category	Technology/Framework	Version
Video players	VideoJS	7.3.0
Security	reCAPTCHA	
Font scripts	Font Awesome	4.7.0
Editor	CodeMirror	5.59.4
Rich text editors	CKEditor	
JavaScript graphics	D3	3.5.17
	Chart.js	
	TypeScript	
Tag managers	Google Tag Manager	
JavaScript libraries	Moment.js	2.11.2
	Microsoft Authentication	
	jQuery	3.4.1
	JSZip	3.4.0
	FingerprintJS	
UI frameworks	Kendo UI	2022.3.913
	Bootstrap	3.4.1
	Angular Material	
Authentication	Google Sign-in	
	Microsoft Authentication	
RUM	Cloudflare Browser Insights	



Proof of Concept:

Confidential



Retest Proof of Concept:

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre>1 GET /rest/service/cards HTTP/2 2 Host: bi.digiicampus.com 3 Cookie: _ga=GA1.2.120876577.1748925664; _gid=GA1.2.1611359392.1753679515; user=eyJlbmMiOiJBMjU2R0NNIiwiaWYwXnIjoIoiZGlyIn0..5aFJ0784Kz6wCSM1.vdmkHEUWChFPj7vR09B0eQ5n6LY2MUYD:URsOCABRiihy0W8fAWmz8Emv1oNuZVhVAPLFleDmJyxLdPNQGMoZruksvf5YhT121Kv05ZWIFQc40nn6Vw0TSb4iaDcP0ohPoYeYVliwtXB8owHFKnRZvZn0KkJbwSLSyKcXsALiaN8cgULHY3paJQW8YN3e5Op82EYyYSDMhiVeMFZQ84ora76fslkRFjrn5MSyz4Q8SyYzMGUWxh51Jgg0GKXgHywjYQwkvLXqTpCNDzaaycF304hNkYndiC3c3BGf029fv8q2q3Bw_cAMfatd2LApgt-V7nuLEHKt7m8912NYJr0-6XVDWpAqDhxMizIK2aFas9khpYax4TQ7Afxj39USfgy8BkL3sCvTCQAq_hhKasey9HxRhrr2ThKGBvyf0IU0k2dNhegUZ3JFoxrcfopd8-m1ax1K9Jb fQ0TyH5wxKdjxGVdH4jrIqY121D f2ph9auJqRvtOyRjW1_e2oW2ecnxASCuV1-nYac52fc2PuV7_XZrBzu4VBAt10117hGHeAQHmN1QnEpaNt3PcS9gEWtYbzd2M6xPdyju5Iv1vjRJEsrIN_QNih_v1JghDXcqW8TvcmeCzoU1AR6kijn98f87cjo7wgX5zJQ7wZNELQC2Ewx_3jdUYuICDWqcwDNJQ71R7ThMpmBkcz8oYa0Aoa18nBeM51BdUerSuy9KYTUMJSiBvAANdE2NMLOg2dsVCjfelzsKQjcgk6QXyppcsuoIAgrZM193j-ev7nsWZvtgzAZSN5Pl55r5fHbLsH6qD0obmeKwI fiU_yFhAj4VEj f8yvJJEvrysc3nTR7wqFpCR7bE7rtEj q20zCcKW</pre>				<pre>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 Cf-Cache-Status: DYNAMIC 21 Report-To: 22 {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v4?s=x0IIBIIAtzwiUL9FFKVa7sf72cTh0u%2FhxPPA7S2RvH%2B4EE18MHgA051e41XE3R23EbCY3WUk%2FiVlc9KbPe2BX18xnSryXPW4vAdkFSTXoNK%2Bh5BXWsgtbkFPeW7r61MRVg%3D%3D"}],"group":"cf-nel","max_age":604800} 23 Nel: {"success_fraction":0,"report_to":"cf-nel","max_age":604800} 24 Strict-Transport-Security: max-age=31536000; includeSubDomains; preload 25 Server: cloudflare 26 Server-Timing: 27 cfl4;desc="?proto=TCP&rtt=20693&min_rtt=16972&rtt_var=5120&sent=66&recv=65&lost=0&retrans=0&sent_bytes=21985&recv_bytes=31634&delivery_rate=778363&cwnd=255&unsent_bytes=0&cid=87efd3771a7873b3&ts=5323&x=0"</pre>			



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