Sri Lanka Institute of Information Technology

Cyber Security Assignment (2025)

Software and Data Integrity Failures

Bug Bounty Report- 05

IT23363366

TABLE OF CONTENT

1. Title 3

Software and Data Integrity Failures (Introduction Topic) 3

Understanding Integrity: 3

2. Scope & Objective 3

3. Enumeration and Reconnaissance 4

Tools Used: 4

Steps Taken 4

4. Vulnerability Description 6

1) Software Integrity Failures: 6

2) Data Integrity Failures 6

5. Affected Component 6

6. Impact Assessment 6

1) Software Integrity Failures: 6

2) Data Integrity Failures 6

7. Proof of Concept (PoC) 7

1) Software Integrity Failures: 7

2) Bypassing JWT Signature Validation: 7

8. Proposed Mitigation 11

9. Conclusion 11

10. References 11

1. Title

Report Title: Software and Data Integrity Failures
Reported By: Raahim Mahmooth
Tested On: https://www.payhere.lk/
Platform: https://bugzero.io

Software and Data Integrity Failures (Introduction Topic)

Understanding Integrity:

Before moving into the bug bounty findings, it is important to understand the concept of "integrity." Integrity ensures that data remains unaltered without proper authorization. The mechanism primarily used to protect integrity is **hashing**.

Hashing is the process of generating a fixed-size string or number (hash) from input data using an algorithm. This hash acts as a digital fingerprint of the data.

Example:
When installing software, we can verify its authenticity by calculating its hash value and comparing it to the publisher’s hash value.

```
md5sum WinSCP-5.21.5-Setup.exe
20c5329d7fde522338f037a7fe8a84eb  WinSCP-5.21.5-Setup.exe
```

If the hashes match, the software is original and unmodified.

2. Scope & Objective

The objective of this assessment was to identify any **Software Integrity Failures** or **Data Integrity Failures** that could compromise the security of the target application. The goal is to ensure that:

- Software libraries used by the application are trusted and validated.
- User authentication tokens (such as **JWTs**) are properly validated to **maintain data integrity**.

3. Enumeration and Reconnaissance

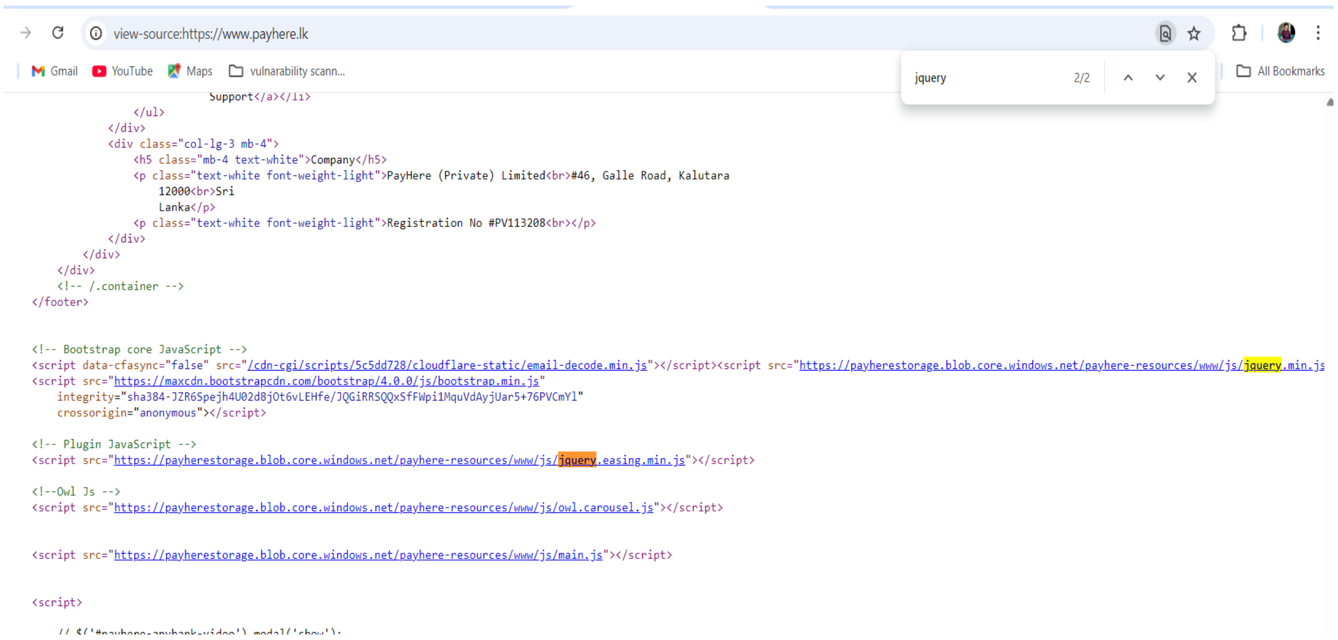
Tools Used:

- Manual Check (for source code review)
- Burp Suite (for intercepting and analyzing network traffic)

Steps Taken

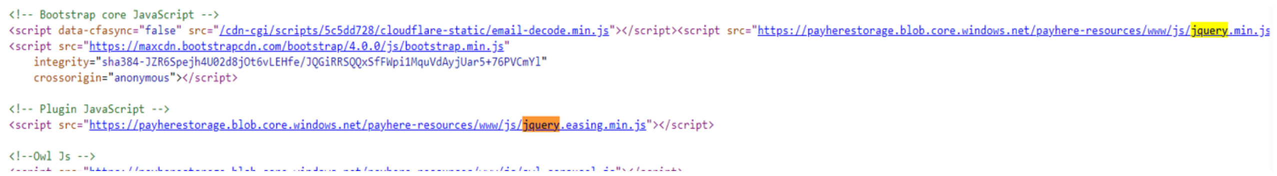
1) Checking Software Integrity:

- Manually analyzed the web application's source code.



```
                    Support</a></li>
                </ul>
            </div>
            <div class="col-lg-3 mb-4">
                <h5 class="mb-4 text-white">Company</h5>
                <p class="text-white font-weight-light">PayHere (Private) Limited<br>#46, Galle Road, Kalutara
                    12000<br>Sri
                    Lanka</p>
                <p class="text-white font-weight-light">Registration No #PV113208<br></p>
            </div>
        </div>
    </div>
    <!-- /.container -->
</footer>


<!-- Bootstrap core JavaScript -->
<script data-cfasync="false" src="/cdn-cgi/scripts/5c5dd728/cloudflare-static/email-decode.min.js"></script><script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/jquery.min.js
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
    integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
    crossorigin="anonymous"></script>

<!-- Plugin JavaScript -->
<script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/jquery.easing.min.js"></script>

<!--Owl Js -->
<script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/owl.carousel.js"></script>


<script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/main.js"></script>


<script>

    // $('#payhere-anybank-video').modal('show');
```

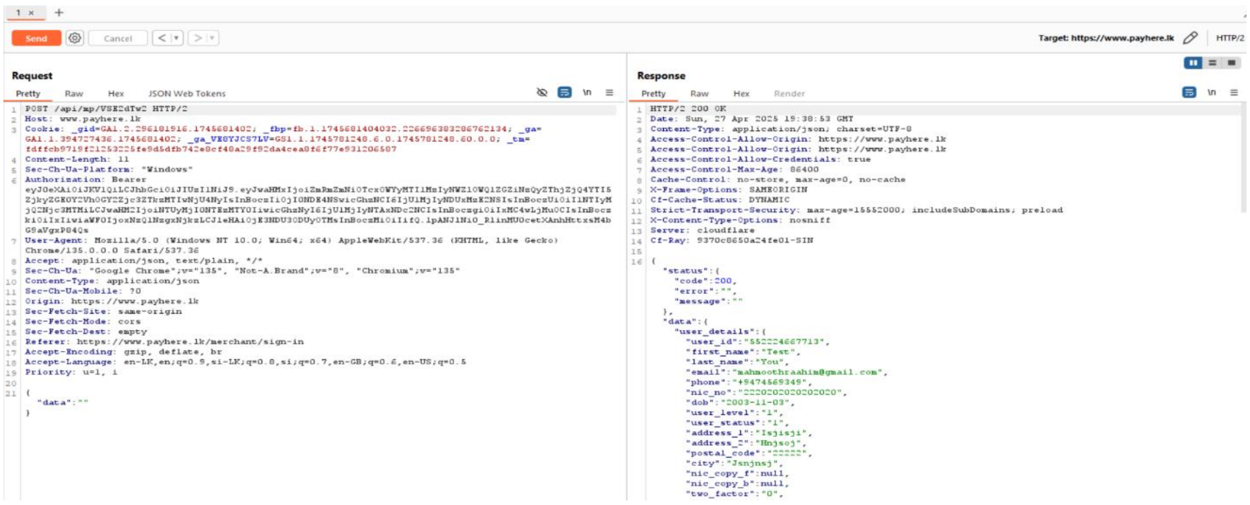
- Identified that the website was using a jQuery library.



```
<!-- Bootstrap core JavaScript -->
<script data-cfasync="false" src="/cdn-cgi/scripts/5c5dd728/cloudflare-static/email-decode.min.js"></script><script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/jquery.min.js
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
    integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
    crossorigin="anonymous"></script>

<!-- Plugin JavaScript -->
<script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/jquery.easing.min.js"></script>

<!--Owl Js -->
```

2) Checking Data Integrity:

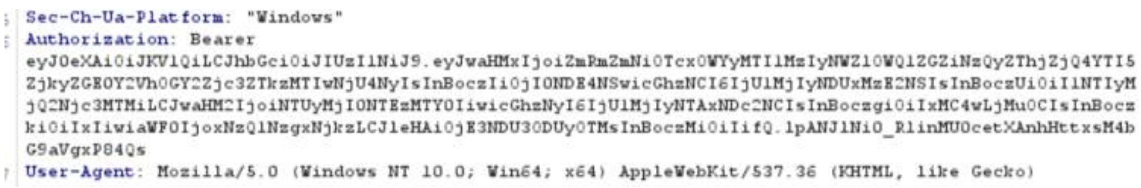
- Captured the login request using Burp Suite and Analyzed the authentication flow.



```
POST /api/mp/VSE2dTw2 HTTP/2
Host: www.payhere.lk
Cookie: _gid=GA1.2.296181916.1745681402; _fbp=fb.1.1745681404032.226696383286762134; _ga=
GA1.1.394727436.1745681402; _ga_VE8YJCS7LV=GS1.1.1745781248.6.0.1745781248.60.0.0; _tm=
fdffcb9719f21253225fe9d5dfb742e8cf48a29f92da4cea8f6f77e931206587
Content-Length: 11
Sec-Ch-Ua-Platform: "Windows"
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwaHMxIjoiZmRmZmNiOTcxOWYyMTI1MzIyNWZlOWQ1ZGZiNzQyZThjZjQ4YTI5
ZjkyZGE0Y2VhOGY2Zjc3ZTkzMTIwNjU4NyIsInBoczIiOjI0NDE4NSwicGhzNCI6IjU1MjIyNDUxMzE2NSIsInBoczUiOiI1NTIyM
jQ2Njc3MTMiLCJwaHM2IjoiNTUyMjI0NTEzMTY0IiwicGhzNyI6IjU1MjIyNTAxNDc2NCIsInBoczgiOiIxMC4wLjMuOCIsInBocz
kiOiIxIiwiaWF0IjoxNzQ1NzgxNjkzLCJleHAiOjE3NDU3ODUyOTMsInBoczMiOiIifQ.1pANJ1NiO_RlinMU0cetXAnhHttxsM4b
G9aVgxP84Qs
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/135.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Google Chrome";v="135", "Not-A.Brand";v="8", "Chromium";v="135"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
Origin: https://www.payhere.lk
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.payhere.lk/merchant/sign-in
Accept-Encoding: gzip, deflate, br
Accept-Language: en-LK,en;q=0.9,si-LK;q=0.8,si;q=0.7,en-GB;q=0.6,en-US;q=0.5
Priority: u=1, i

{
  "data":""
}
```

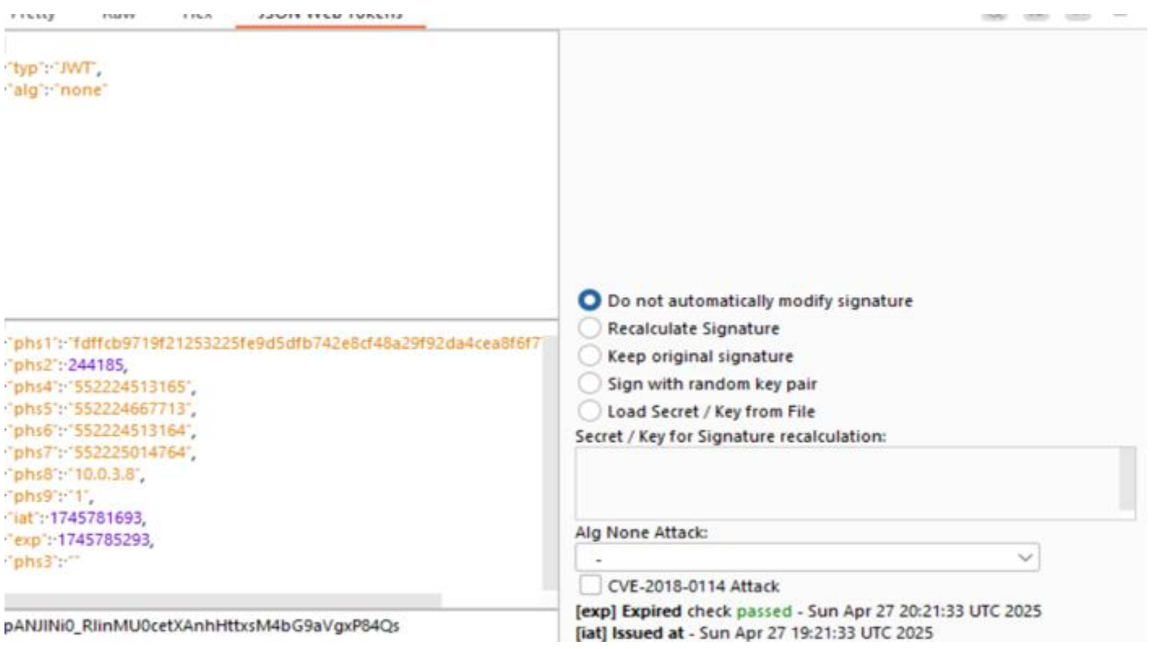
```
HTTP/2 200 OK
Date: Sun, 27 Apr 2025 19:38:53 GMT
Content-Type: application/json; charset=UTF-8
Access-Control-Allow-Origin: https://www.payhere.lk
Access-Control-Allow-Origin: https://www.payhere.lk
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 86400
Cache-Control: no-store, max-age=0, no-cache
X-Frame-Options: SAMEORIGIN
Cf-Cache-Status: DYNAMIC
Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
X-Content-Type-Options: nosniff
Server: cloudflare
Cf-Ray: 9370c8650a24fe01-SIN

{
  "status":{
    "code":200,
    "error":"",
    "message":""
  },
  "data":{
    "user_details":{
      "user_id":"552224667713",
      "first_name":"Test",
      "last_name":"You",
      "email":"mahmoothraahim@gmail.com",
      "phone":"+9474569349",
      "nic_no":"222020202020202020",
      "dob":"2003-11-03",
      "user_level":"1",
      "user_status":"1",
      "address_1":"Isjisji",
      "address_2":"Hnjsoj",
      "postal_code":"22222",
      "city":"Jsnjnsj",
      "nic_copy_f":null,
      "nic_copy_b":null,
      "two_factor":"0",
```

- Found that the application was using **JWT (JSON Web Token)** for session management.



```
Sec-Ch-Ua-Platform: "Windows"
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwaHMxIjoiZmRmZmNiOTcxOWYyMTI1MzIyNWZlOWQ1ZGZiNzQyZThjZjQ4YTI5
ZjkyZGE0Y2VhOGY2Zjc3ZTkzMTIwNjU4NyIsInBoczIiOjI0NDE4NSwicGhzNCI6IjU1MjIyNDUxMzE2NSIsInBoczUiOiI1NTIyM
jQ2Njc3MTMiLCJwaHM2IjoiNTUyMjI0NTEzMTY0IiwicGhzNyI6IjU1MjIyNTAxNDc2NCIsInBoczgiOiIxMC4wLjMuOCIsInBocz
kiOiIxIiwiaWF0IjoxNzQ1NzgxNjkzLCJleHAiOjE3NDU3ODUyOTMsInBoczMiOiIifQ.1pANJ1NiO_RlinMU0cetXAnhHttxsM4b
G9aVgxP84Qs
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
```

- **JWT (JSON Web Token)**



```
"typ":"JWT",
"alg":"none"
```

```
"phs1":"fdffcb9719f21253225fe9d5dfb742e8cf48a29f92da4cea8f6f7
"phs2":244185,
"phs4":"552224513165",
"phs5":"552224667713",
"phs6":"552224513164",
"phs7":"552225014764",
"phs8":"10.0.3.8",
"phs9":"1",
"iat":1745781693,
"exp":1745785293,
"phs3":""
```

pANJINiO_RlinMU0cetXAnhHttxsM4bG9aVgxP84Qs

[✓ Do not automatically modify signature]
Recalculate Signature
Keep original signature
Sign with random key pair
Load Secret / Key from File

Secret / Key for Signature recalculation:

Alg None Attack: -

CVE-2018-0114 Attack

[exp] Expired check passed - Sun Apr 27 20:21:33 UTC 2025
[iat] Issued at - Sun Apr 27 19:21:33 UTC 2025

4. Vulnerability Description

1) Software Integrity Failures:

The web application included external libraries (e.g., jQuery) via direct URL references. Without integrity verification (such as **Sub resource Integrity** (SRI)), there is a risk that if the source is compromised, malicious scripts could be injected into the website without detection. Attackers could exploit this to execute arbitrary JavaScript on users' browsers.

2) Data Integrity Failures:

JWT tokens must be properly validated. If token validation is weak or missing (such as accepting unsigned tokens or using predictable secrets), attackers can forge tokens and escalate their privileges — such as gaining unauthorized access to administrative functionalities.

5. Affected Component

- **Source Code:** *view-source:https://www.payhere.lk/*
- **Login Request:** *https://www.payhere.lk/merchant/sign-in*
  Captured during user authentication flow.

6. Impact Assessment

1) Software Integrity Failures:

If an attacker **modifies the referenced jQuery library** at its hosted location, every user visiting the legitimate website would unknowingly load and execute malicious JavaScript code. This can **lead** to attacks like **credential theft, session hijacking, or even full client-side compromise.**

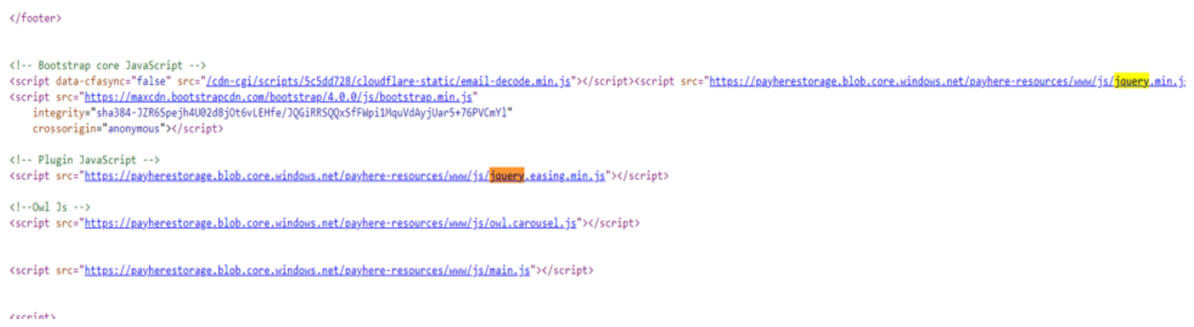
2) Data Integrity Failures:

If the server accepts **unsigned or weakly signed JWT tokens**, attackers could craft their own tokens, impersonate legitimate users (including admins), and gain unauthorized access to protected resources, potentially leading to full account takeover and privilege escalation.

7. Proof of Concept (PoC)

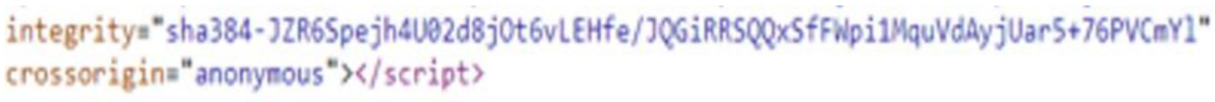
1) Software Integrity Failures:

- The source code revealed the usage of a jQuery library.



```
</footer>


<!-- Bootstrap core JavaScript -->
<script data-cfasync="false" src="/cdn-cgi/scripts/5c5dd728/cloudflare-static/email-decode.min.js"></script><script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/jquery.min.j
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
    integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
    crossorigin="anonymous"></script>

<!-- Plugin JavaScript -->
<script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/jquery.easing.min.js"></script>

<!--Owl Js -->
<script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/owl.carousel.js"></script>


<script src="https://payherestorage.blob.core.windows.net/payhere-resources/www/js/main.js"></script>


<script>
```

- However, integrity checks (such as Subresource Integrity attributes) were present alongside the jQuery URL.



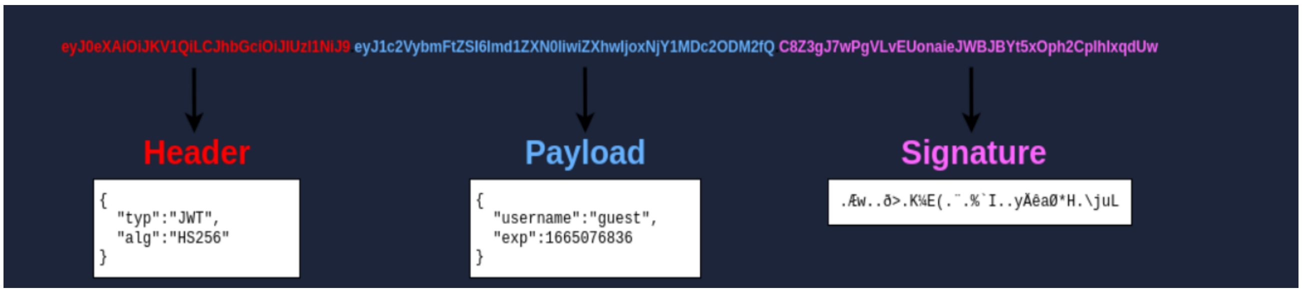
```
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
```

- **Conclusion:** This was a downloaded jQuery fetched from there storage however sub resource Integrity checks were in place, reducing the risk of software integrity failure.

2) Bypassing JWT Signature Validation:

Overview:
A JWT token generally looks like three base64-encoded parts separated by dots:

`Header.Payload.Signature`



```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Imd1ZXN0IiwiZXhwIjoxNjY1MDc2ODM2fQ.C8Z3gJ7wPgVLvEUonaieJWBJBYt5xOph2CplhlxqdUw
```

```
{
  "typ":"JWT",
  "alg":"HS256"
}
```

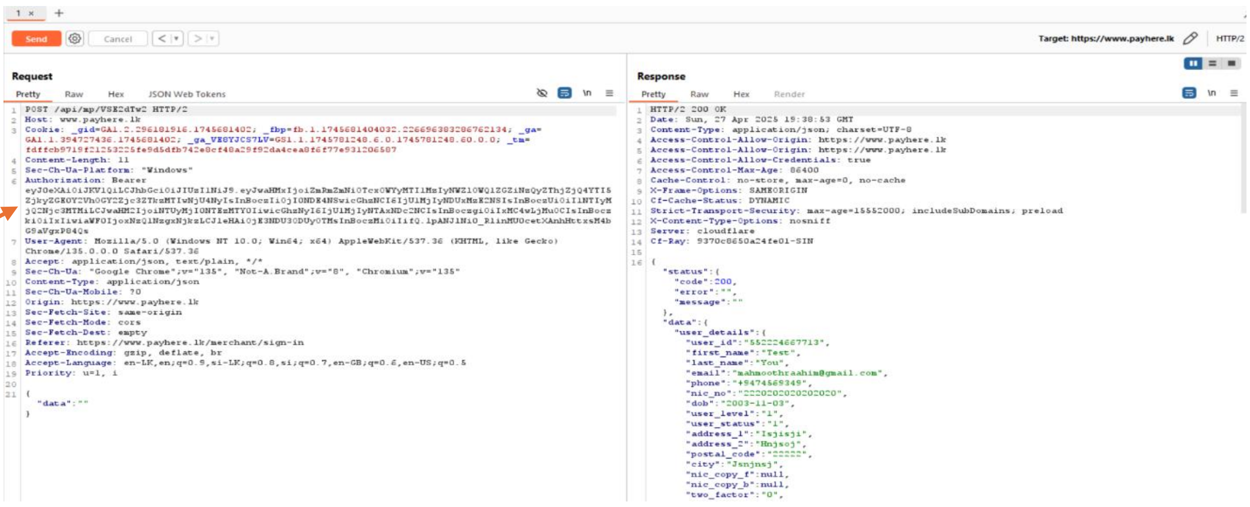
```
{
  "username":"guest",
  "exp":1665076836
}
```

```
.Æw..ð>.K¼E(.¨.%`I..yĂêaØ*H.\juL
```

It allows you to store key-value pairs in the payload and ensures integrity using the signature.

Steps Taken:

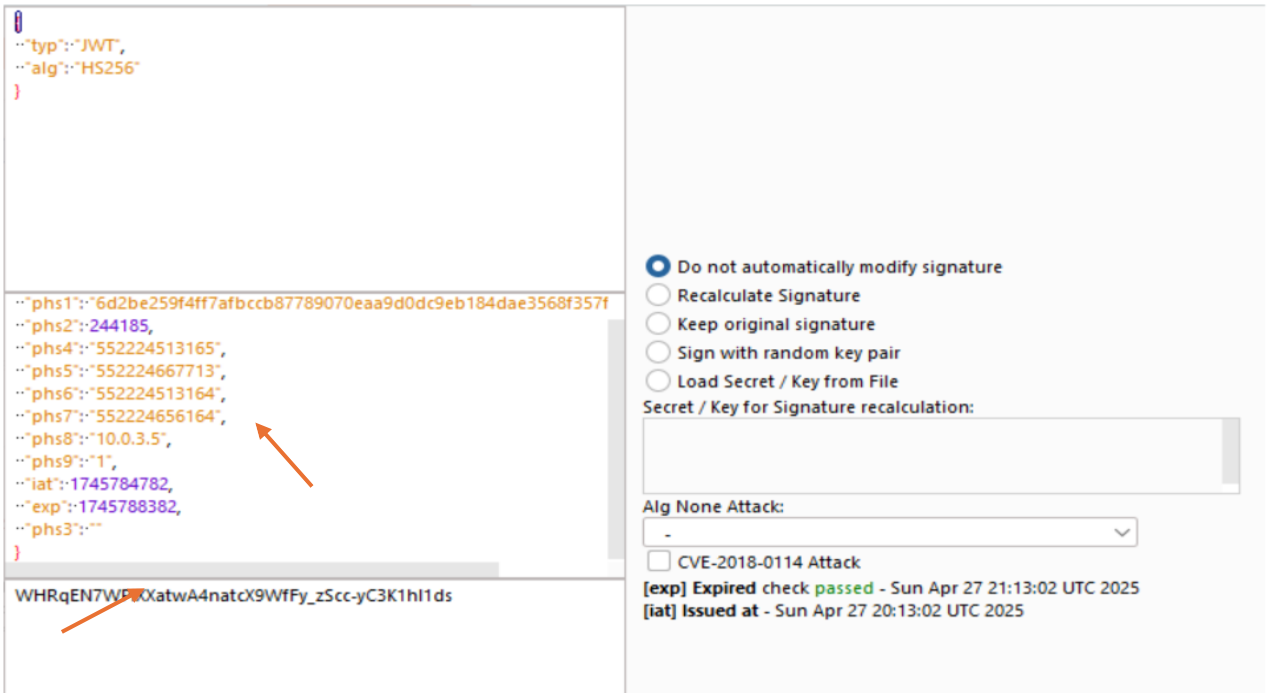
2.1 Captured the login request and identified JWT usage.



```
POST /api/mp/VSE2dTw2 HTTP/2
Host: www.payhere.lk
Cookie: _gid=GA1.2.296181916.1745681402; _fbp=fb.1.1745681404032.226696383286762134; _ga=
GA1.1.394727436.1745681402; _ga_VE8YJCS7LV=GS1.1.1745781248.6.0.1745781248.60.0.0; _tm=
fdffcb9719f21253225fe9d5dfb742e8cf48a29f92da4cea8f6f77e931206587
Content-Length: 11
Sec-Ch-Ua-Platform: "Windows"
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwaHMxIjoiZmRmZmNiOTcxOWYyMTI1MzIyNWZlOWQ1ZGZiNzQyZThjZjQ4YTI5
ZjkyZGE0Y2VhOGY2Zjc3ZTkzMTIwNjU4NyIsInBoczIiOjI0NDE4NSwicGhzNCI6IjU1MjIyNDUxMzE2NSIsInBoczUiOiI1NTIyM
jQ2Njc3MTMiLCJwaHM2IjoiNTUyMjI0NTEzMTY0IiwicGhzNyI6IjU1MjIyNTAxNDc2NCIsInBoczgiOiIxMC4wLjMuOCIsInBocz
kiOiIxIiwiaWF0IjoxNzQ1NzgxNjkzLCJleHAiOjE3NDU3ODUyOTMsInBoczMiOiIifQ.1pANJ1NiO_RlinMU0cetXAnhHttxsM4b
G9aVgxP84Qs
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/135.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Google Chrome";v="135", "Not-A.Brand";v="8", "Chromium";v="135"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
Origin: https://www.payhere.lk
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.payhere.lk/merchant/sign-in
Accept-Encoding: gzip, deflate, br
Accept-Language: en-LK,en;q=0.9,si-LK;q=0.8,si;q=0.7,en-GB;q=0.6,en-US;q=0.5
Priority: u=1, i

{
  "data":""
}
```

```
HTTP/2 200 OK
Date: Sun, 27 Apr 2025 19:38:53 GMT
Content-Type: application/json; charset=UTF-8
Access-Control-Allow-Origin: https://www.payhere.lk
Access-Control-Allow-Origin: https://www.payhere.lk
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 86400
Cache-Control: no-store, max-age=0, no-cache
X-Frame-Options: SAMEORIGIN
Cf-Cache-Status: DYNAMIC
Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
X-Content-Type-Options: nosniff
Server: cloudflare
Cf-Ray: 9370c8650a24fe01-SIN

{
  "status":{
    "code":200,
    "error":"",
    "message":""
  },
  "data":{
    "user_details":{
      "user_id":"552224667713",
      "first_name":"Test",
      "last_name":"You",
      "email":"mahmoothraahim@gmail.com",
      "phone":"+94745689349",
      "nic_no":"222020202020202020",
      "dob":"2003-11-03",
      "user_level":"1",
      "user_status":"1",
      "address_1":"Isjisji",
      "address_2":"Hnjsoj",
      "postal_code":"22222",
      "city":"Jsnjnsj",
      "nic_copy_f":null,
      "nic_copy_b":null,
      "two_factor":"0",
```

2.2 Installed a JWT extension to inspect the token contents.



```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

```
  "phs1": "6d2be259f4ff7afbccb87789070eaa9d0dc9eb184dae3568f357f
  "phs2": 244185,
  "phs4": "552224513165",
  "phs5": "552224667713",
  "phs6": "552224513164",
  "phs7": "552224656164",
  "phs8": "10.0.3.5",
  "phs9": "1",
  "iat": 1745784782,
  "exp": 1745788382,
  "phs3": ""
}
```

```
WHRqEN7WP[illegible]XatwA4natcX9WfFy_zScc-yC3K1hl1ds
```

[✓ Do not automatically modify signature]
Recalculate Signature
Keep original signature
Sign with random key pair
Load Secret / Key from File

Secret / Key for Signature recalculation:

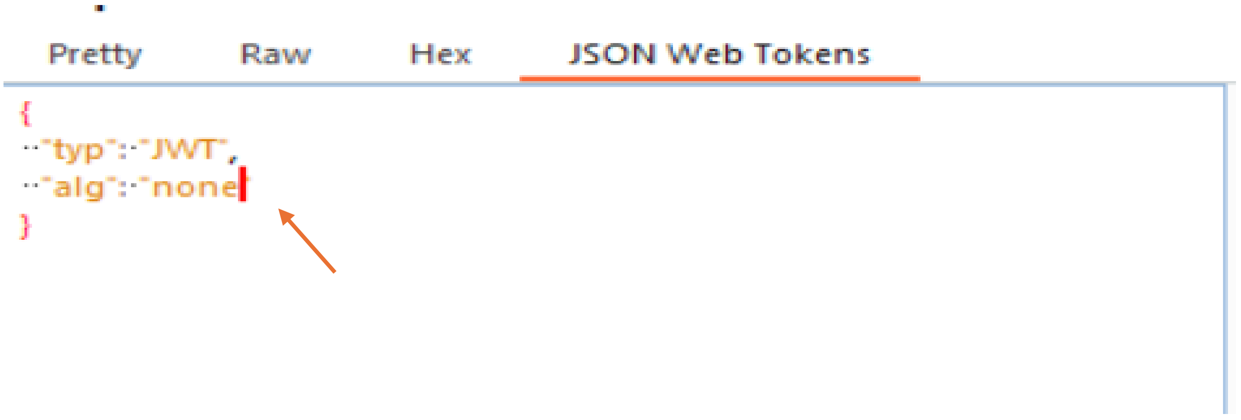
Alg None Attack: -

CVE-2018-0114 Attack

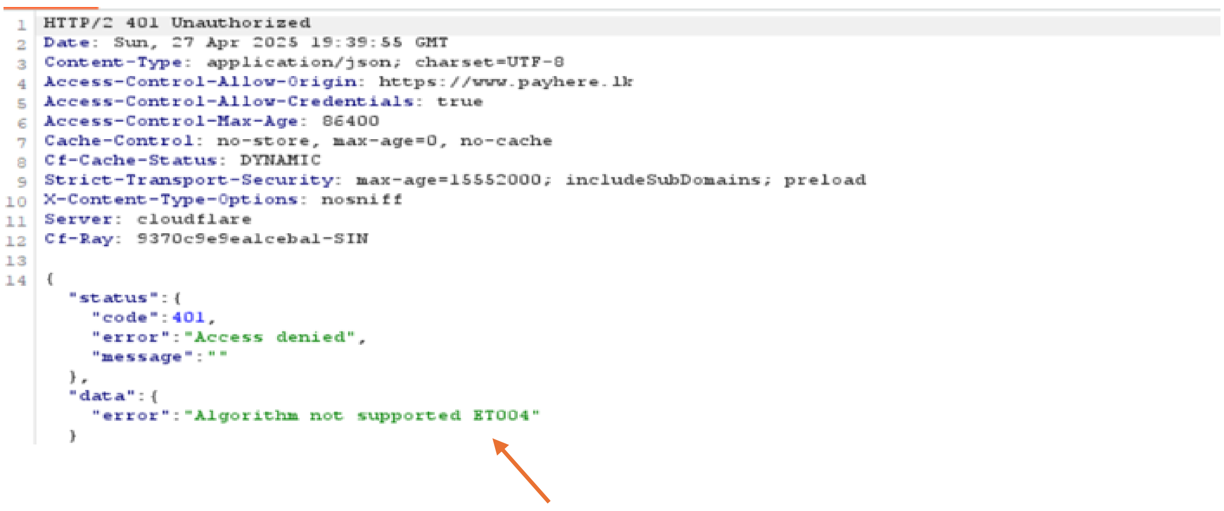
[exp] Expired check passed - Sun Apr 27 21:13:02 UTC 2025
[iat] Issued at - Sun Apr 27 20:13:02 UTC 2025

2.3 Modified the header:

```
{
  "typ": "JWT",
  "alg": "none"
}
```

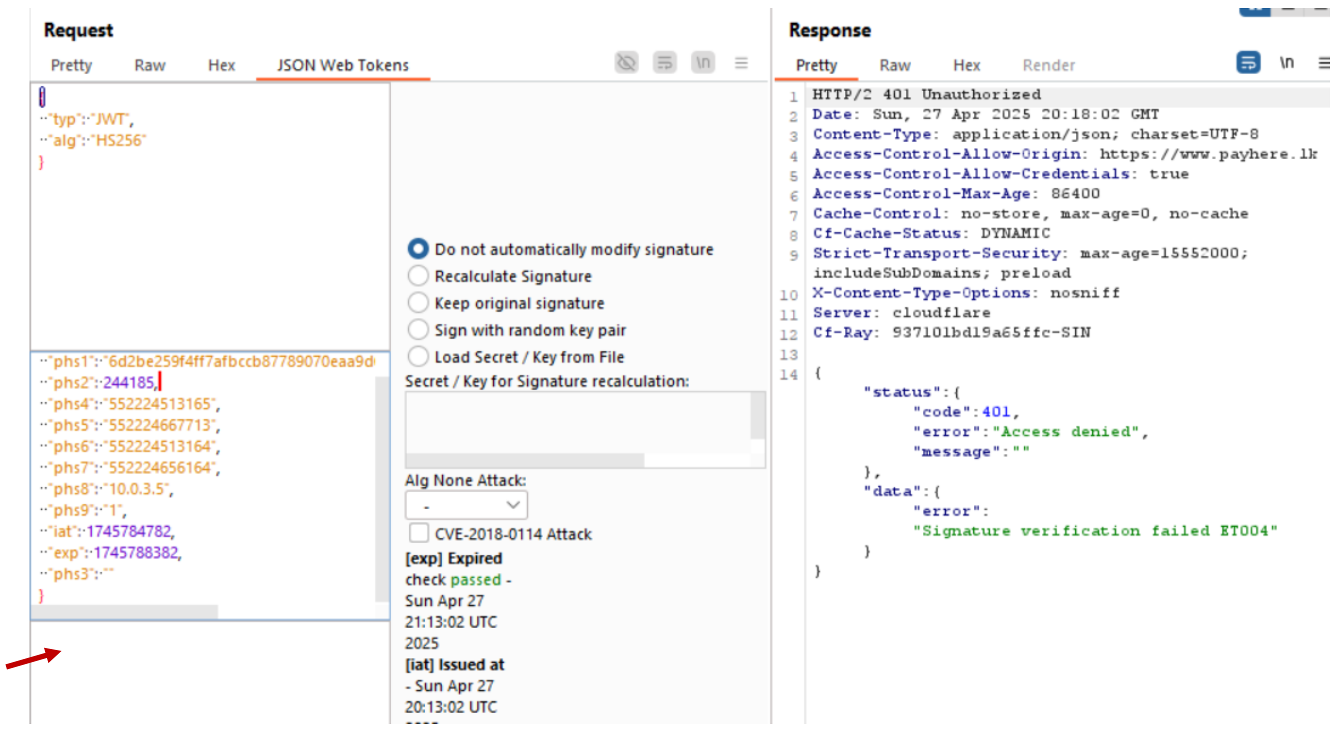


```
{
··"typ":·"JWT",
··"alg":·"none
}
```



```
HTTP/2 401 Unauthorized
Date: Sun, 27 Apr 2025 19:39:55 GMT
Content-Type: application/json; charset=UTF-8
Access-Control-Allow-Origin: https://www.payhere.lk
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 86400
Cache-Control: no-store, max-age=0, no-cache
Cf-Cache-Status: DYNAMIC
Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
X-Content-Type-Options: nosniff
Server: cloudflare
Cf-Ray: 9370c9e9ea1ceba1-SIN

{
  "status":{
    "code":401,
    "error":"Access denied",
    "message":""
  },
  "data":{
    "error":"Algorithm not supported ET004"
  }
```

- And attempted to resend the token without a signature — **request failed**, indicating the server properly enforced signature validation.



```
{
··"typ":·"JWT",
··"alg":·"HS256"
}
```

```
··"phs1":·"6d2be259f4ff7afbccb87789070eaa9d
··"phs2":·244185,
··"phs4":·"552224513165",
··"phs5":·"552224667713",
··"phs6":·"552224513164",
··"phs7":·"552224656164",
··"phs8":·"10.0.3.5",
··"phs9":·"1",
··"iat":·1745784782,
··"exp":·1745788382,
··"phs3":·""
}
```

[✓ Do not automatically modify signature]
Recalculate Signature
Keep original signature
Sign with random key pair
Load Secret / Key from File

Secret / Key for Signature recalculation:

Alg None Attack: -

☐ CVE-2018-0114 Attack

[exp] Expired check passed - Sun Apr 27 21:13:02 UTC 2025

[iat] Issued at - Sun Apr 27 20:13:02 UTC

```
HTTP/2 401 Unauthorized
Date: Sun, 27 Apr 2025 20:18:02 GMT
Content-Type: application/json; charset=UTF-8
Access-Control-Allow-Origin: https://www.payhere.lk
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 86400
Cache-Control: no-store, max-age=0, no-cache
Cf-Cache-Status: DYNAMIC
Strict-Transport-Security: max-age=15552000;
includeSubDomains; preload
X-Content-Type-Options: nosniff
Server: cloudflare
Cf-Ray: 937101bd19a65ffc-SIN

{
	"status":{
		"code":401,
		"error":"Access denied",
		"message":""
	},
	"data":{
		"error":
		"Signature verification failed ET004"
	}
}
```

It proving that token validation mechanisms were correctly implemented. Indicating the server not validate the **unsigned or weakly signed JWT tokens.**

8. Proposed Mitigation

- Always use **Subresource Integrity (SRI)** when embedding third-party libraries such as jQuery to ensure the library has not been tampered with.
- Avoid loading external scripts from untrusted sources.
- When using **JWT**:
    - Enforce strong signature validation on every token received.
    - Use strong, complex secrets and regularly rotate them.
    - Avoid accepting tokens with `"alg":"none"` or missing signature parts.
    - Implement token expiration and revocation mechanisms.

These practices will ensure both software and data integrity are maintained effectively.

9. Conclusion

Software and data integrity failures can severely compromise a web application's trustworthiness and security. By implementing best practices such as Sub resource Integrity for external libraries and strong signature validation for JWT tokens, the risk of integrity-based attacks can be significantly minimized. It is crucial to regularly audit and update the security mechanisms to adapt to evolving threats.

10. References

- OWASP Top 10 - *A08:2021 - Software and Data Integrity Failures*
- Subresource Integrity - *https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity*
- JSON Web Tokens (JWT) Introduction
- CVE-2017-11427 - Insecure JWT "None" Algorithm Attack