Sri Lanka Institute of Information Technology

Web Security - IE2062

Bug Bounty Report 9

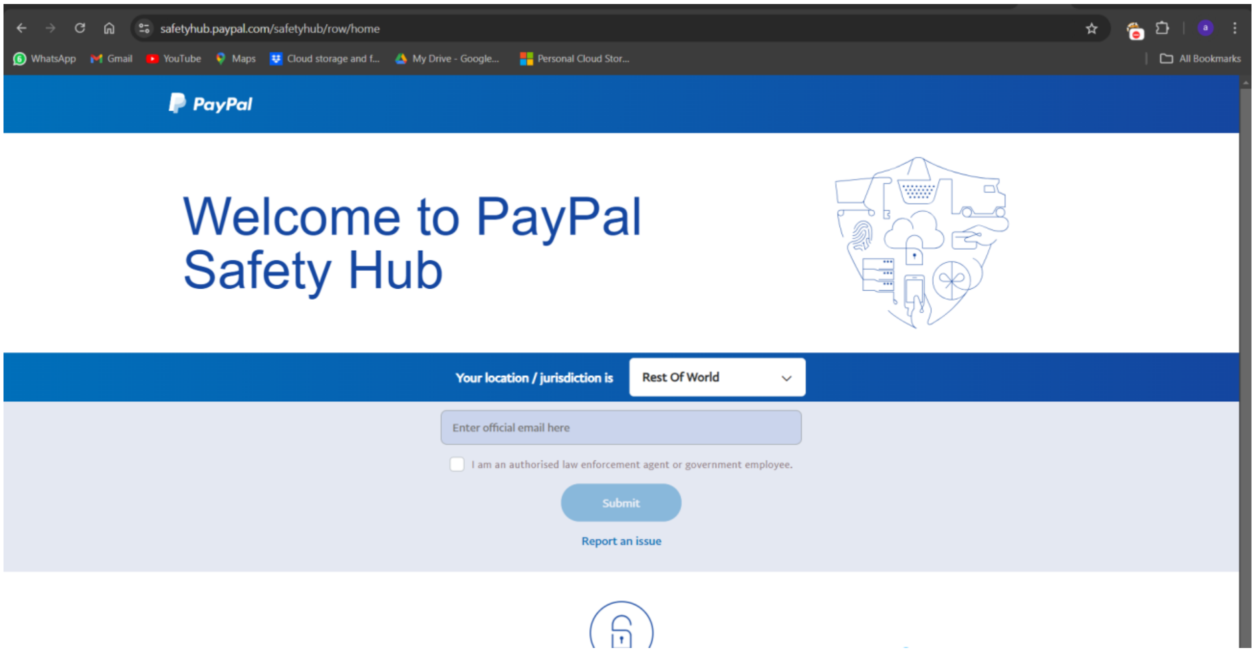
PERERA A.P.J

IT22280992

Group Y2S2.CS

Table of Contents

1. TARGET: http://safetyhub.paypal.com



safetyhub.paypal.com/safetyhub/row/home

Welcome to PayPal Safety Hub

Your location / jurisdiction is: Rest Of World

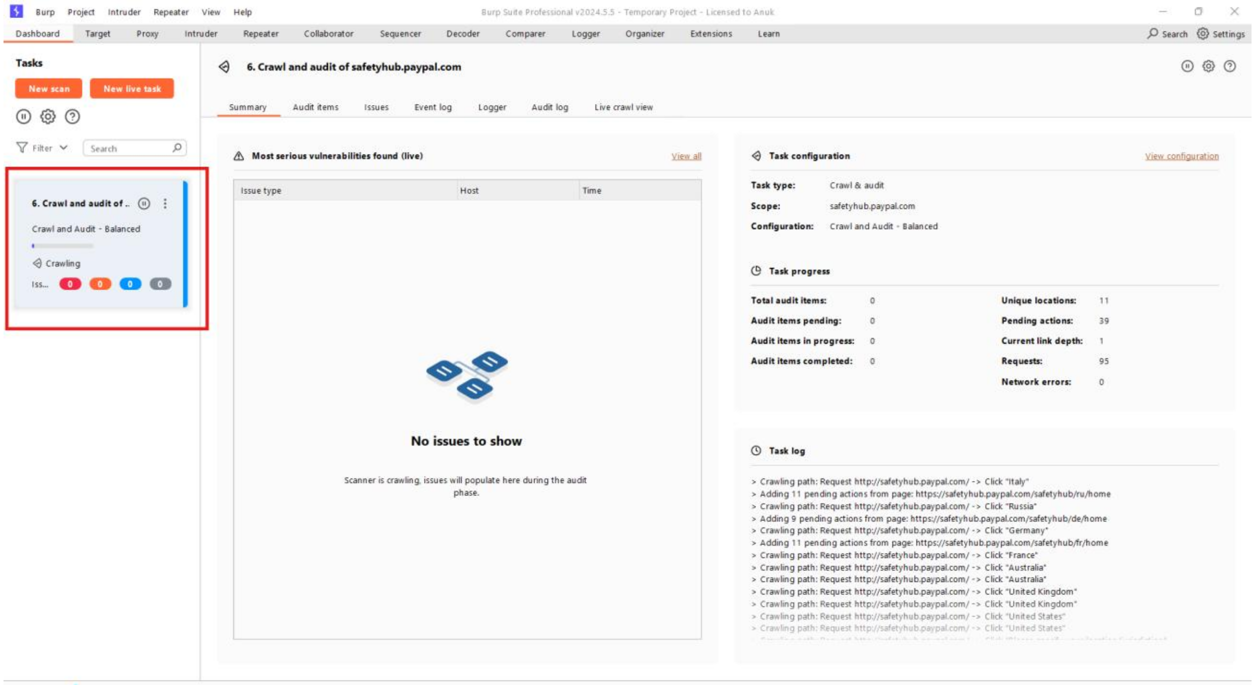
Enter official email here

□ I am an authorised law enforcement agent or government employee.

2. Vulnerability

2.1 Vulnerability title

- HTML 5 Storage Manupalation (DOM-based).



6. Crawl and audit of safetyhub.paypal.com

Most serious vulnerabilities found (live)

Issue type	Host	Time

No issues to show

Scanner is crawling, issues will populate here during the audit phase.

Task configuration

Task type: Crawl & audit
Scope: safetyhub.paypal.com
Configuration: Crawl and Audit - Balanced

Task progress

Total audit items:	0	Unique locations:	11
Audit items pending:	0	Pending actions:	39
Audit items in progress:	0	Current link depth:	1
Audit items completed:	0	Requests:	95
		Network errors:	0

Task log

```
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Italy"
> Adding 11 pending actions from page: https://safetyhub.paypal.com/safetyhub/ru/home
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Russia"
> Adding 9 pending actions from page: https://safetyhub.paypal.com/safetyhub/de/home
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Germany"
> Adding 11 pending actions from page: https://safetyhub.paypal.com/safetyhub/fr/home
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "France"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Australia"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Australia"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United Kingdom"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United Kingdom"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United States"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United States"
```

2.2 Vulnerability description

- The HTML5 Storage Manipulation Vulnerability, which is DOM-based, encapsulates those security lapses which result from incorrect utilization of local storage or session storage on the web client. Such a vulnerability can be noticed in applications that do not carry out any validation or security measures on data considered possessed by the browser due to its actual physical storage. The attackers here can cause a script to be saved in the storage and when this data is fetched and drawn by the concerned application, it results in an XSS attack. They may also exploit the technique by changing the data that has already been stored in the application to change the way the application behaves, access the application without being cleared to do so, or escalate the access level of use. Besides, putting encrypted information in these storage formats only increases the threat because attackers will have easier access to internal secrets. In conclusion, HTML5 Storage Manipulation vulnerabilities demonstrate the importance of security protocols and practices focused on the protection of client-side storage against possible abuses.

2.3 Affected components

The affected components in an HTML5 Storage Manipulation (DOM-based) vulnerability can include various aspects of a web application. Here are the primary components at risk:

1. Client-Side Storage Mechanisms

- localStorage and sessionStorage: These are the primary storage APIs used to store data in the user's browser. Improper handling of data in these storage locations can lead to vulnerabilities, such as data injection or unauthorized access.

2. JavaScript Code

- Data Retrieval and Manipulation: JavaScript functions that read from localStorage or sessionStorage and insert that data into the DOM without proper validation or sanitization are critical points of failure. This includes any code that directly manipulates the DOM based on untrusted storage data.

3. DOM Elements

- HTML Elements: Any HTML elements that display or use data retrieved from client-side storage are at risk. If this data is not properly sanitized, it can lead to XSS attacks when malicious scripts are executed in the browser.

4. Application Logic

- User Authentication and Session Management: If user session information or authentication tokens are stored in localStorage or sessionStorage, attackers can manipulate this data to gain unauthorized access or escalate privileges.

5. Input Validation Mechanisms

- Form Handling and Input Validation: Components responsible for validating user input may also be affected if they do not account for data sourced from storage. Insufficient input validation can lead to the execution of malicious code or manipulation of application state.

6. Error Handling and Logging

- Error Messages: If error handling mechanisms reveal sensitive information about stored data or the application’s internal state, it can provide attackers with insights into how to exploit vulnerabilities.

7. Security Controls

- Cross-Site Scripting (XSS) Protections: Web applications that rely on client-side storage without implementing strong XSS protections can expose themselves to manipulation and exploitation through stored XSS vulnerabilities.

2.4 Impact assessment

The impact assessment of HTML5 Storage Manipulation (DOM-based) vulnerabilities involves evaluating the potential risks and consequences that can arise from exploiting these weaknesses. Here are the key impacts to consider:

1. Security Breaches

- Cross-Site Scripting (XSS): Successful exploitation can lead to XSS attacks, allowing attackers to execute arbitrary JavaScript in the context of the victim's browser. This can result in session hijacking, data theft, or redirecting users to malicious sites.

2. Data Leakage

- Exposure of Sensitive Information: If sensitive data (such as authentication tokens, personal information, or user credentials) is stored in client-side storage without proper encryption, attackers can easily access this information, leading to data breaches.

3. Unauthorized Access and Privilege Escalation

- Manipulation of Application State: Attackers can modify values in storage to alter the application's behavior, potentially gaining unauthorized access to restricted areas or elevating user privileges. This could lead to significant security incidents and loss of data integrity.

4. Loss of User Trust

- Reputation Damage: Data breaches or security incidents stemming from these vulnerabilities can erode user trust in the application or organization. Users may be hesitant to use the application if they believe their data is not secure.

5. Legal and Regulatory Consequences

- Compliance Violations: Exposing personal data or failing to protect sensitive information can lead to violations of data protection regulations, such as GDPR or CCPA, resulting in fines, legal action, and additional compliance requirements.

6. Operational Disruption

- Downtime and Recovery Efforts: If a vulnerability is exploited and leads to a breach, the organization may need to undergo extensive recovery efforts, including incident response, system audits, and potential downtime, impacting overall business operations.

7. Increased Attack Surface

- Exploiting Client-Side Logic: Attackers can leverage client-side storage manipulation to find other vulnerabilities within the application, leading to further exploitation and more severe security issues.

2.5 Steps to reproduce

- Steps to Reproduce HTML5 Storage Manipulation Vulnerability.

1. Set Up the Testing Environment

- Use a web browser with developer tools (e.g., Chrome, Firefox) that allow you to inspect and manipulate localStorage and sessionStorage.

2. Identify the Target Application

- Choose a web application that utilizes HTML5 storage for managing user data, preferences, or sessions. Ensure you have the necessary permissions to test the application.

3. Inspect Existing Data in Storage

- Open the developer tools (F12) and navigate to the Application tab (in Chrome) or the Storage tab (in Firefox).
- Look for localStorage or sessionStorage entries associated with the target application. Note the key-value pairs stored there.

4. Analyze Application Behavior

- Determine how the application uses data from localStorage or sessionStorage. Look for areas where the stored data is rendered in the DOM or affects application functionality, such as:
  - User preferences (e.g., themes, layout settings)
  - User authentication tokens or session IDs
  - Configuration settings for user roles or permissions

5. Modify Storage Data

- Select a key in localStorage or sessionStorage that the application uses and modify its value. You can:
  - Change a legitimate value (e.g., a user role or preference).
  - Inject a script or payload into a field that will be rendered in the DOM (e.g., <script>alert('XSS')</script>).

6. Trigger Application Logic

- Reload the page or navigate to the part of the application that reads the modified storage value. Observe how the application responds to the altered data.
- For example, if you injected a script into the storage, check if it executes when the application reads and renders the value.

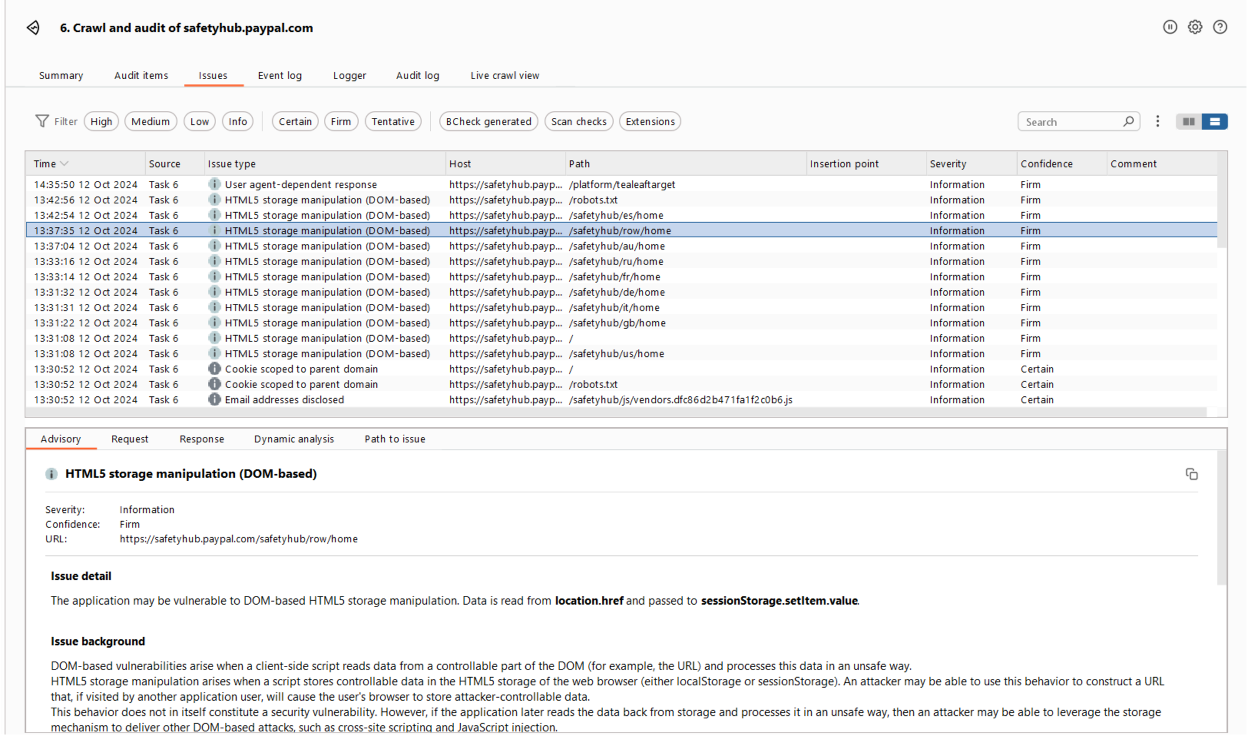
7. Check for Security Issues

- Analyze the outcome of your modifications:
  - XSS Vulnerabilities: If your injected script executed, the application is vulnerable to XSS attacks.
  - Privilege Escalation: If you were able to gain elevated privileges or access restricted areas, document the steps taken.
  - Data Exposure: Check if sensitive information is accessible through the manipulated storage.

8. Document Findings

- Record the steps taken, including the original and modified storage values, the impact of your modifications, and any security issues identified. This documentation will be helpful for reporting the vulnerability to the relevant stakeholders.

2.6 Proof of concept



6. Crawl and audit of safetyhub.paypal.com

Time	Source	Issue type	Host	Path	Insertion point	Severity	Confidence	Comment
14:35:50 12 Oct 2024	Task 6	User agent-dependent response	https://safetyhub.payp...	/platform/tealeaftarget		Information	Firm	
13:42:56 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/robots.txt		Information	Firm	
13:42:54 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/es/home		Information	Firm	
13:37:35 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/row/home		Information	Firm	
13:37:04 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/au/home		Information	Firm	
13:33:16 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/ru/home		Information	Firm	
13:33:14 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/fr/home		Information	Firm	
13:31:32 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/de/home		Information	Firm	
13:31:31 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/it/home		Information	Firm	
13:31:22 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/gb/home		Information	Firm	
13:31:08 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/		Information	Firm	
13:31:08 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/us/home		Information	Firm	
13:30:52 12 Oct 2024	Task 6	Cookie scoped to parent domain	https://safetyhub.payp...	/		Information	Certain	
13:30:52 12 Oct 2024	Task 6	Cookie scoped to parent domain	https://safetyhub.payp...	/robots.txt		Information	Certain	
13:30:52 12 Oct 2024	Task 6	Email addresses disclosed	https://safetyhub.payp...	/safetyhub/js/vendors.dfc86d2b471fa1f2c0b6.js		Information	Certain	

HTML5 storage manipulation (DOM-based)

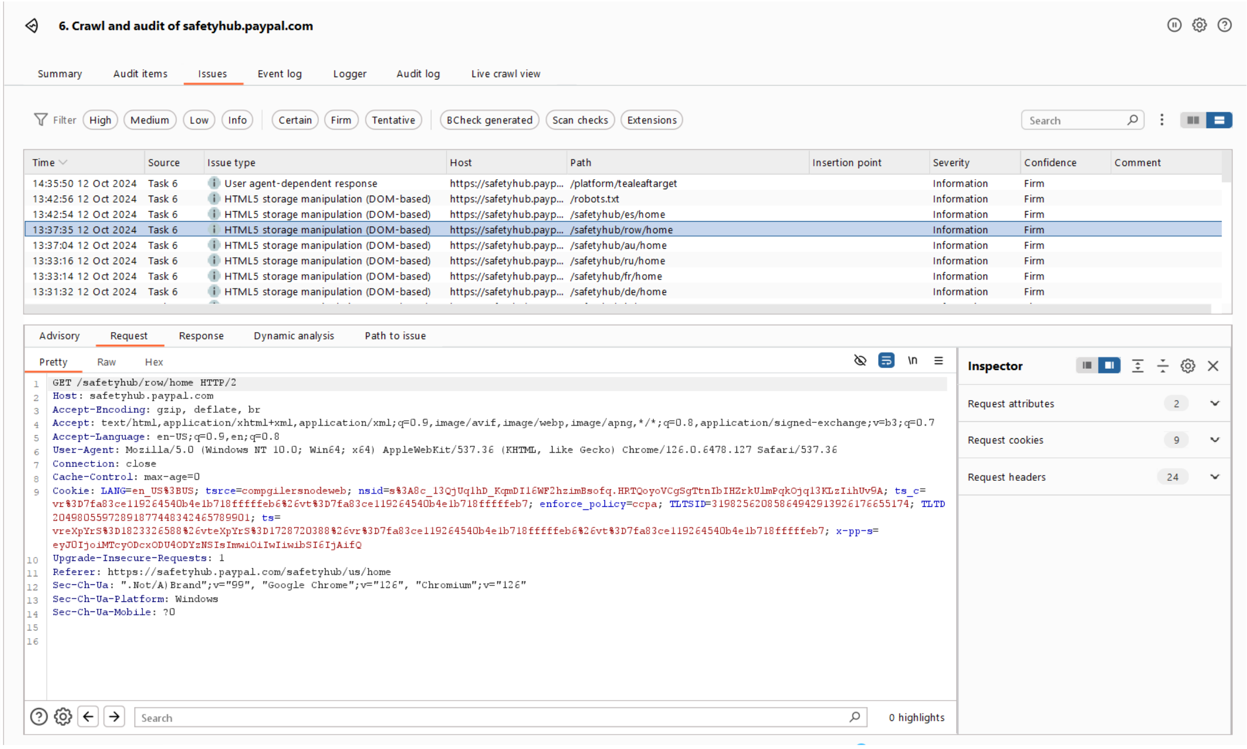
Severity: Information
Confidence: Firm
URL: https://safetyhub.paypal.com/safetyhub/row/home

Issue detail

The application may be vulnerable to DOM-based HTML5 storage manipulation. Data is read from **location.href** and passed to **sessionStorage.setItem.value**.

Issue background

DOM-based vulnerabilities arise when a client-side script reads data from a controllable part of the DOM (for example, the URL) and processes this data in an unsafe way.
HTML5 storage manipulation arises when a script stores controllable data in the HTML5 storage of the web browser (either localStorage or sessionStorage). An attacker may be able to use this behavior to construct a URL that, if visited by another application user, will cause the user's browser to store attacker-controllable data.
This behavior does not in itself constitute a security vulnerability. However, if the application later reads the data back from storage and processes it in an unsafe way, then an attacker may be able to leverage the storage mechanism to deliver other DOM-based attacks, such as cross-site scripting and JavaScript injection.



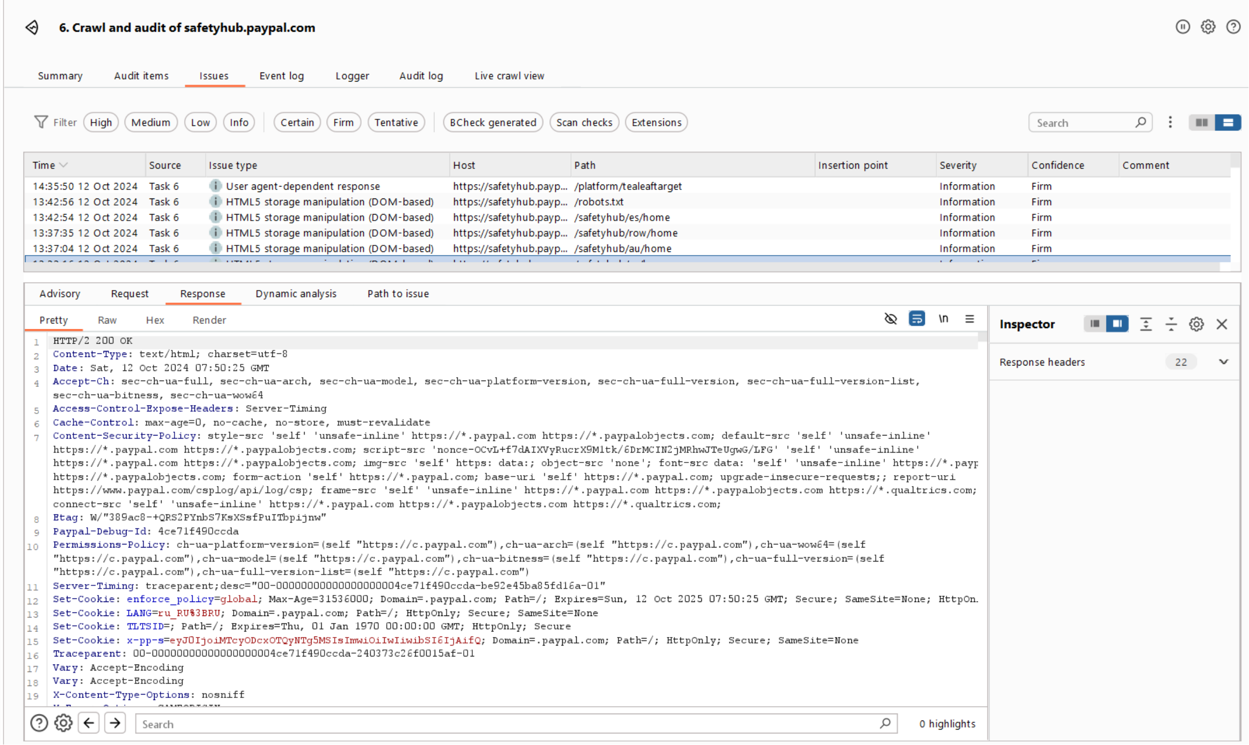
6. Crawl and audit of safetyhub.paypal.com

Time	Source	Issue type	Host	Path	Insertion point	Severity	Confidence	Comment
14:35:50 12 Oct 2024	Task 6	User agent-dependent response	https://safetyhub.payp...	/platform/tealeaftarget		Information	Firm	
13:42:56 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/robots.txt		Information	Firm	
13:42:54 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/es/home		Information	Firm	
13:37:35 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/row/home		Information	Firm	
13:37:04 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/au/home		Information	Firm	
13:33:16 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/ru/home		Information	Firm	
13:33:14 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/fr/home		Information	Firm	
13:31:32 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/de/home		Information	Firm	

```
GET /safetyhub/row/home HTTP/2
Host: safetyhub.paypal.com
Accept-Encoding: gzip, deflate, br
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: LANG=en_US%3BUS; tsrce=compgilersnodeweb; nsid=s%3A8c_13QjUq1hD_KqmDI16WF2hzimBsofq.HRTQoyoVCgSgTtnIbIHZrkUlmPqkOjq13KLzIihUv9A; ts_c=
vr%3D7fa83ce119264540b4e1b718fffffeb6%26vt%3D7fa83ce119264540b4e1b718fffffeb7; enforce_policy=ccpa; TLTSID=31982562085864942913926176655174; TLTD
20498055972891877448342465789901; ts=
vreXpYrS%3D1823326588%26vteXpYrS%3D1728720388%26vr%3D7fa83ce119264540b4e1b718fffffeb6%26vt%3D7fa83ce119264540b4e1b718fffffeb7; x-pp-s=
eyJ0IjoiMTcyODcxODU4ODYzNSIsImwiOiIwIiwibSI6IjAifQ
Upgrade-Insecure-Requests: 1
Referer: https://safetyhub.paypal.com/safetyhub/us/home
Sec-Ch-Ua: ".Not/A)Brand";v="99", "Google Chrome";v="126", "Chromium";v="126"
Sec-Ch-Ua-Platform: Windows
Sec-Ch-Ua-Mobile: ?0


```

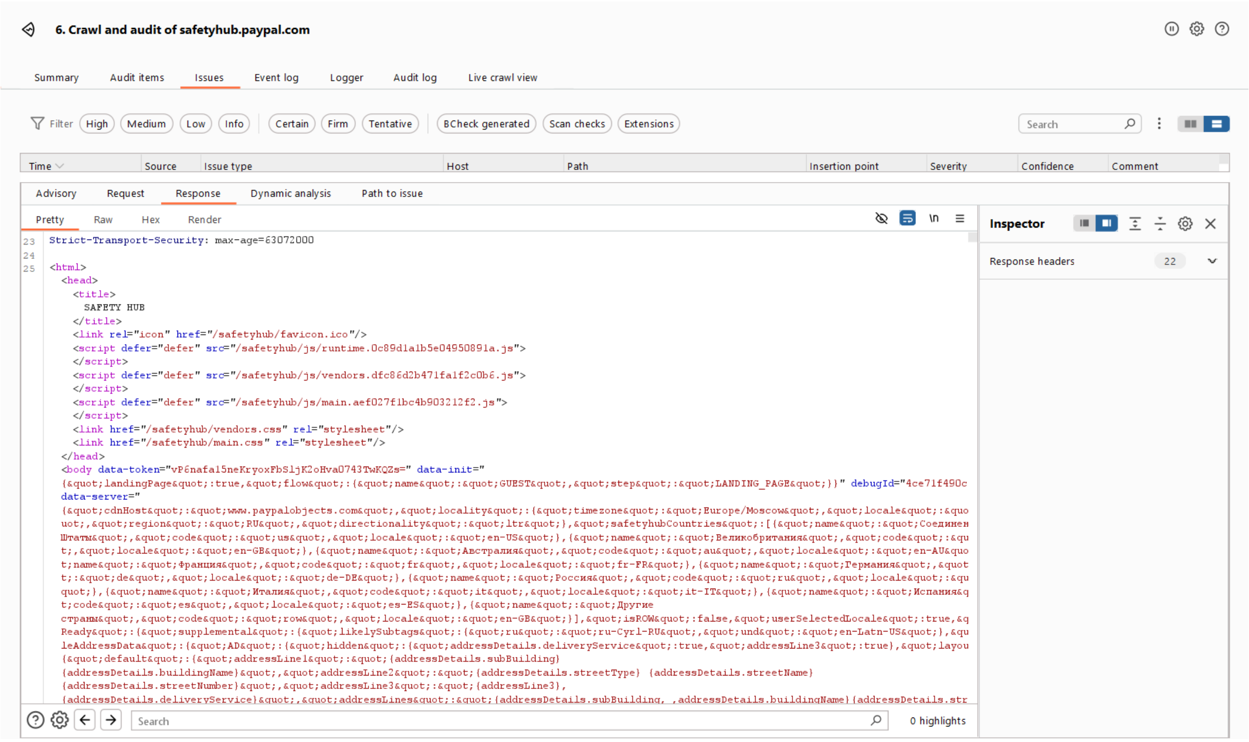
Inspector	
Request attributes	2
Request cookies	9
Request headers	24



6. Crawl and audit of safetyhub.paypal.com

Time	Source	Issue type	Host	Path	Insertion point	Severity	Confidence	Comment
14:35:50 12 Oct 2024	Task 6	User agent-dependent response	https://safetyhub.payp...	/platform/tealeaftarget		Information	Firm	
13:42:56 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/robots.txt		Information	Firm	
13:42:54 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/es/home		Information	Firm	
13:37:35 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/row/home		Information	Firm	
13:37:04 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/au/home		Information	Firm	

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Date: Sat, 12 Oct 2024 07:50:25 GMT
Accept-Ch: sec-ch-ua-full, sec-ch-ua-arch, sec-ch-ua-model, sec-ch-ua-platform-version, sec-ch-ua-full-version, sec-ch-ua-full-version-list,
sec-ch-ua-bitness, sec-ch-ua-wow64
Access-Control-Expose-Headers: Server-Timing
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Content-Security-Policy: style-src 'self' 'unsafe-inline' https://*.paypal.com https://*.paypalobjects.com; default-src 'self' 'unsafe-inline'
https://*.paypal.com https://*.paypalobjects.com; script-src 'nonce-OCvL+f7dAIXVyRucrX9M1tk/6DrMCIN2jMRhwJTeUgwG/LFG' 'self' 'unsafe-inline'
https://*.paypal.com https://*.paypalobjects.com; img-src 'self' https: data:; object-src 'none'; font-src data: 'self' 'unsafe-inline' https://*.payp
https://*.paypalobjects.com; form-action 'self' https://*.paypal.com; base-uri 'self' https://*.paypal.com; upgrade-insecure-requests;; report-uri
https://www.paypal.com/csplog/api/log/csp; frame-src 'self' 'unsafe-inline' https://*.paypal.com https://*.paypalobjects.com https://*.qualtrics.com;
connect-src 'self' 'unsafe-inline' https://*.paypal.com https://*.paypalobjects.com https://*.qualtrics.com;
Etag: W/"389ac8-+QRS2PYnbS7KsXSsfPuITbpijnw"
Paypal-Debug-Id: 4ce71f490ccda
Permissions-Policy: ch-ua-platform-version=(self "https://c.paypal.com"),ch-ua-arch=(self "https://c.paypal.com"),ch-ua-wow64=(self
"https://c.paypal.com"),ch-ua-model=(self "https://c.paypal.com"),ch-ua-bitness=(self "https://c.paypal.com"),ch-ua-full-version=(self
"https://c.paypal.com"),ch-ua-full-version-list=(self "https://c.paypal.com")
Server-Timing: traceparent;desc="00-00000000000000000004ce71f490ccda-be92e45ba85fd16a-01"
Set-Cookie: enforce_policy=global; Max-Age=31536000; Domain=.paypal.com; Path=/; Expires=Sun, 12 Oct 2025 07:50:25 GMT; Secure; SameSite=None; HttpOn.
Set-Cookie: LANG=ru_RU%3BRU; Domain=.paypal.com; Path=/; HttpOnly; Secure; SameSite=None
Set-Cookie: TLTSID=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT; HttpOnly; Secure
Set-Cookie: x-pp-s=eyJ0IjoiMTcyODcxOTQyNTg5MSIsImwiOiIwIiwibSI6IjAifQ; Domain=.paypal.com; Path=/; HttpOnly; Secure; SameSite=None
Traceparent: 00-00000000000000000004ce71f490ccda-240373c26f0015af-01
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
```

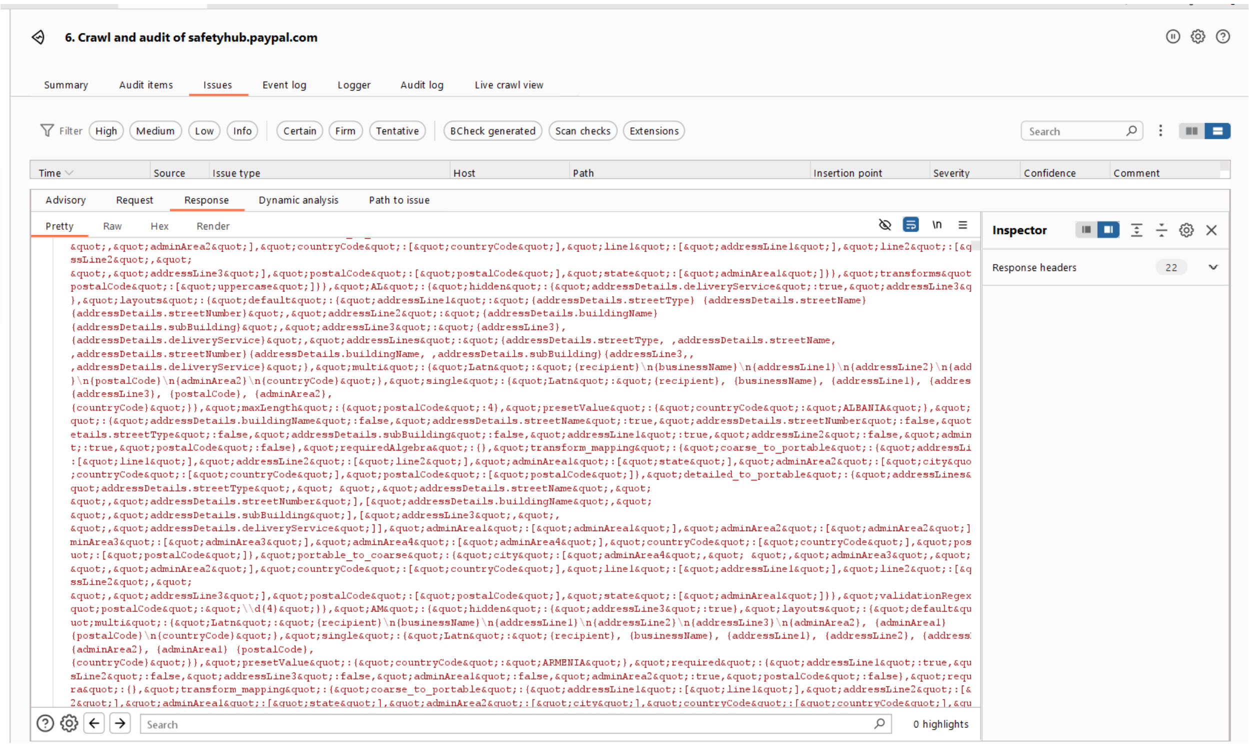


6. Crawl and audit of safetyhub.paypal.com

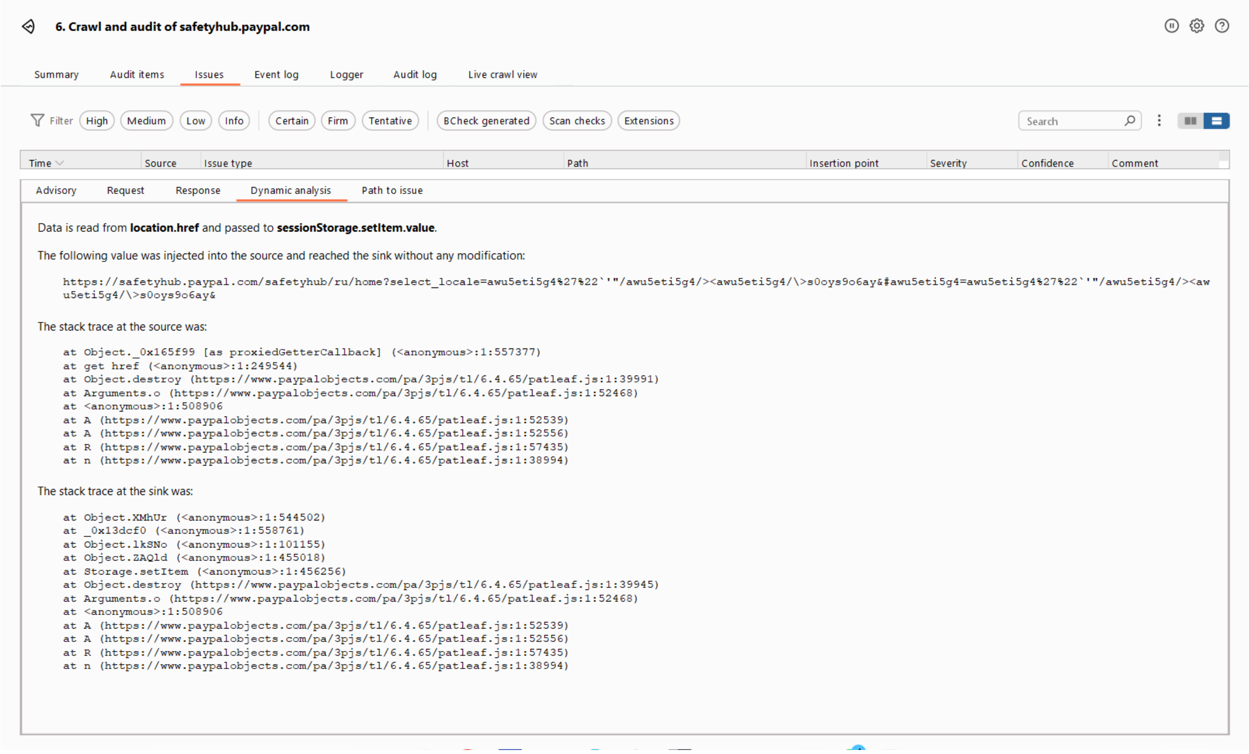
Time	Source	Issue type	Host	Path	Insertion point	Severity	Confidence	Comment

```
Strict-Transport-Security: max-age=63072000

<html>
  <head>
    <title>
      SAFETY HUB
    </title>
    <link rel="icon" href="/safetyhub/favicon.ico"/>
    <script defer="defer" src="/safetyhub/js/runtime.0c89d1a1b5e04950891a.js">
    </script>
    <script defer="defer" src="/safetyhub/js/vendors.dfc86d2b471fa1f2c0b6.js">
    </script>
    <script defer="defer" src="/safetyhub/js/main.aef027f1bc4b903212f2.js">
    </script>
    <link href="/safetyhub/vendors.css" rel="stylesheet"/>
    <link href="/safetyhub/main.css" rel="stylesheet"/>
  </head>
  <body data-token="vP6nafa15neKryoxFbSljK2oHva0743TwKQZs=" data-init="
  {&quot;landingPage&quot;:true,&quot;flow&quot;:{&quot;name&quot;:&quot;GUEST&quot;,&quot;step&quot;:&quot;LANDING_PAGE&quot;}}" debugId="4ce71f490c
  data-server="
  {&quot;cdnHost&quot;:&quot;www.paypalobjects.com&quot;,&quot;locality&quot;:{&quot;timezone&quot;:&quot;Europe/Moscow&quot;,&quot;locale&quot;:&quo
  uot;,&quot;region&quot;:&quot;RU&quot;,&quot;directionality&quot;:&quot;ltr&quot;},&quot;safetyhubCountries&quot;:[{&quot;name&quot;:&quot;Соединен
  Штаты&quot;,&quot;code&quot;:&quot;us&quot;,&quot;locale&quot;:&quot;en-US&quot;},{&quot;name&quot;:&quot;Великобритания&quot;,&quot;code&quot;:&qu
  t;,&quot;locale&quot;:&quot;en-GB&quot;},{&quot;name&quot;:&quot;Австралия&quot;,&quot;code&quot;:&quot;au&quot;,&quot;locale&quot;:&quot;en-AU&quo
  t;name&quot;:&quot;Франция&quot;,&quot;code&quot;:&quot;fr&quot;,&quot;locale&quot;:&quot;fr-FR&quot;},{&quot;name&quot;:&quot;Германия&quot;,&quot
  t;:&quot;de&quot;,&quot;locale&quot;:&quot;de-DE&quot;},{&quot;name&quot;:&quot;Россия&quot;,&quot;code&quot;:&quot;ru&quot;,&quot;locale&quot;:&qu
  quot;},{&quot;name&quot;:&quot;Италия&quot;,&quot;code&quot;:&quot;it&quot;,&quot;locale&quot;:&quot;it-IT&quot;},{&quot;name&quot;:&quot;Испания&q
  t;code&quot;:&quot;es&quot;,&quot;locale&quot;:&quot;es-ES&quot;},{&quot;name&quot;:&quot;Другие
  страны&quot;,&quot;code&quot;:&quot;row&quot;,&quot;locale&quot;:&quot;en-GB&quot;}],&quot;isROW&quot;:false,&quot;userSelectedLocale&quot;:true,&q
  Ready&quot;:{&quot;supplemental&quot;:{&quot;likelySubtags&quot;:{&quot;ru&quot;:&quot;ru-Cyrl-RU&quot;,&quot;und&quot;:&quot;en-Latn-US&quot;},&qu
  leAddressData&quot;:{&quot;AD&quot;:{&quot;hidden&quot;:{&quot;addressDetails.deliveryService&quot;:true,&quot;addressLine3&quot;:true},&quot;layou
  {&quot;default&quot;:{&quot;addressLine1&quot;:&quot;{addressDetails.subBuilding}
  {addressDetails.buildingName}&quot;,&quot;addressLine2&quot;:&quot;{addressDetails.streetType} {addressDetails.streetName}
  {addressDetails.streetNumber}&quot;,&quot;addressLine3&quot;:&quot;{addressLine3},
  {addressDetails.deliveryService}&quot;,&quot;addressLines&quot;:&quot;{addressDetails.subBuilding, ,addressDetails.buildingName}{addressDetails.str
```



```
&quot;,&quot;adminArea2&quot;],&quot;countryCode&quot;:[&quot;countryCode&quot;],&quot;line1&quot;:[&quot;addressLine1&quot;],&quot;line2&quot;:[&q
ssLine2&quot;,&quot;
&quot;,&quot;addressLine3&quot;],&quot;postalCode&quot;:[&quot;postalCode&quot;],&quot;state&quot;:[&quot;adminArea1&quot;]}},&quot;transforms&quot
postalCode&quot;:[&quot;uppercase&quot;]}},&quot;AL&quot;:{&quot;hidden&quot;:{&quot;addressDetails.deliveryService&quot;:true,&quot;addressLine3&q
},&quot;layouts&quot;:{&quot;default&quot;:{&quot;addressLine1&quot;:&quot;{addressDetails.streetType} {addressDetails.streetName}
{addressDetails.streetNumber}&quot;,&quot;addressLine2&quot;:&quot;{addressDetails.buildingName}
{addressDetails.subBuilding}&quot;,&quot;addressLine3&quot;:&quot;{addressLine3},
{addressDetails.deliveryService}&quot;,&quot;addressLines&quot;:&quot;{addressDetails.streetType, ,addressDetails.streetName,
,addressDetails.streetNumber}{addressDetails.buildingName, ,addressDetails.subBuilding}{addressLine3,,
,addressDetails.deliveryService}&quot;},&quot;multi&quot;:{&quot;Latn&quot;:&quot;{recipient}\n{businessName}\n{addressLine1}\n{addressLine2}\n{add
}\n{postalCode}\n{adminArea2}\n{countryCode}&quot;},&quot;single&quot;:{&quot;Latn&quot;:&quot;{recipient}, {businessName}, {addressLine1}, {addres
{addressLine3}, {postalCode}, {adminArea2},
{countryCode}&quot;}},&quot;maxLength&quot;:{&quot;postalCode&quot;:4},&quot;presetValue&quot;:{&quot;countryCode&quot;:&quot;ALBANIA&quot;},&quot;
quot;:{&quot;addressDetails.buildingName&quot;:false,&quot;addressDetails.streetName&quot;:true,&quot;addressDetails.streetNumber&quot;:false,&quot
etails.streetType&quot;:false,&quot;addressDetails.subBuilding&quot;:false,&quot;addressLine1&quot;:true,&quot;addressLine2&quot;:false,&quot;admin
t;:true,&quot;postalCode&quot;:false},&quot;requiredAlgebra&quot;:{},&quot;transform_mapping&quot;:{&quot;coarse_to_portable&quot;:{&quot;addressLi
:[&quot;line1&quot;],&quot;addressLine2&quot;:[&quot;line2&quot;],&quot;adminArea1&quot;:[&quot;state&quot;],&quot;adminArea2&quot;:[&quot;city&quo
;countryCode&quot;:[&quot;countryCode&quot;],&quot;postalCode&quot;:[&quot;postalCode&quot;]},&quot;detailed_to_portable&quot;:{&quot;addressLines&
quot;addressDetails.streetType&quot;,&quot; &quot;,&quot;addressDetails.streetName&quot;,&quot;
&quot;,&quot;addressDetails.streetNumber&quot;],[&quot;addressDetails.buildingName&quot;,&quot;
&quot;,&quot;addressDetails.subBuilding&quot;],[&quot;addressLine3&quot;,&quot;,
&quot;,&quot;addressDetails.deliveryService&quot;]],&quot;adminArea1&quot;:[&quot;adminArea1&quot;],&quot;adminArea2&quot;:[&quot;adminArea2&quot;]
minArea3&quot;:[&quot;adminArea3&quot;],&quot;adminArea4&quot;:[&quot;adminArea4&quot;],&quot;countryCode&quot;:[&quot;countryCode&quot;],&quot;pos
uot;:[&quot;postalCode&quot;]},&quot;portable_to_coarse&quot;:{&quot;city&quot;:[&quot;adminArea4&quot;,&quot; &quot;,&quot;adminArea3&quot;,&quot;
&quot;,&quot;adminArea2&quot;],&quot;countryCode&quot;:[&quot;countryCode&quot;],&quot;line1&quot;:[&quot;addressLine1&quot;],&quot;line2&quot;:[&q
ssLine2&quot;,&quot;
&quot;,&quot;addressLine3&quot;],&quot;postalCode&quot;:[&quot;postalCode&quot;],&quot;state&quot;:[&quot;adminArea1&quot;]}},&quot;validationRegex
quot;postalCode&quot;:&quot;\\d{4}&quot;}},&quot;AM&quot;:{&quot;hidden&quot;:{&quot;addressLine3&quot;:true},&quot;layouts&quot;:{&quot;default&qu
uot;multi&quot;:{&quot;Latn&quot;:&quot;{recipient}\n{businessName}\n{addressLine1}\n{addressLine2}\n{addressLine3}\n{adminArea2}, {adminArea1}
{postalCode}\n{countryCode}&quot;},&quot;single&quot;:{&quot;Latn&quot;:&quot;{recipient}, {businessName}, {addressLine1}, {addressLine2}, {address
{adminArea2}, {adminArea1} {postalCode},
{countryCode}&quot;}},&quot;presetValue&quot;:{&quot;countryCode&quot;:&quot;ARMENIA&quot;},&quot;required&quot;:{&quot;addressLine1&quot;:true,&qu
sLine2&quot;:false,&quot;addressLine3&quot;:false,&quot;adminArea1&quot;:false,&quot;adminArea2&quot;:true,&quot;postalCode&quot;:false},&quot;requ
ra&quot;:{},&quot;transform_mapping&quot;:{&quot;coarse_to_portable&quot;:{&quot;addressLine1&quot;:[&quot;line1&quot;],&quot;addressLine2&quot;:[&
2&quot;],&quot;adminArea1&quot;:[&quot;state&quot;],&quot;adminArea2&quot;:[&quot;city&quot;],&quot;countryCode&quot;:[&quot;countryCode&quot;],&qu
```



Data is read from **location.href** and passed to **sessionStorage.setItem.value**.

The following value was injected into the source and reached the sink without any modification:

```
https://safetyhub.paypal.com/safetyhub/ru/home?select_locale=awu5eti5g4%27%22`'"/awu5eti5g4/><awu5eti5g4/\>s0oys9o6ay&#awu5eti5g4=awu5eti5g4%27%22`'"/awu5eti5g4/><aw
u5eti5g4/\>s0oys9o6ay&
```

The stack trace at the source was:

```
at Object._0x165f99 [as proxiedGetterCallback] (<anonymous>:1:557377)
at get href (<anonymous>:1:249544)
at Object.destroy (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:39991)
at Arguments.o (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:52468)
at <anonymous>:1:508906
at A (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:52539)
at A (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:52556)
at R (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:57435)
at n (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:38994)
```

The stack trace at the sink was:

```
at Object.XMhUr (<anonymous>:1:544502)
at _0x13dcf0 (<anonymous>:1:558761)
at Object.lkSNo (<anonymous>:1:101155)
at Object.ZAQld (<anonymous>:1:455018)
at Storage.setItem (<anonymous>:1:456256)
at Object.destroy (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:39945)
at Arguments.o (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:52468)
at <anonymous>:1:508906
at A (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:52539)
at A (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:52556)
at R (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:57435)
at n (https://www.paypalobjects.com/pa/3pjs/tl/6.4.65/patleaf.js:1:38994)
```

2.7 Proposed mitigation or fix

- Input Validation.
- Avoid Storing Sensitive Data.
- Use Secure coding Practices.
- Session Management and Authentication.
- Regular Security Audits.
- Error Handling and logging.
- User Education.