BRONAT

SECURITY ASSESSMENT

Web Application Penetration Test

Meridian Workspace

Web application & API | Grey-box assessment

October 21, 2026



SYNTHETIC SAMPLE

Fictional client, systems, identities and evidence.

REPORT NAVIGATION

Contents

From business risk to technical evidence.

1. Executive summary 3

2. Dashboard 4

3. Project summary & scope 5

4. Methodology & approach 7

5. Findings overview 9

6. Detailed findings 10

- 6.1 Cross-tenant invoice disclosure 10
- 6.2 Member-to-administrator privilege escalation 12
- 6.3 Access tokens remain valid after logout 14
- 6.4 Account enumeration through password recovery 16
- 6.5 Invoice PDFs remain cached on a shared workstation 18
- 6.6 Content Security Policy is report-only 20

7. Field definitions 22

8. Contact information 23

BUSINESS PERSPECTIVE

1. Executive summary

Synopsis

Meridian Workspace engaged BROONAT to assess the security of its web application and supporting API. The assessment focused on access controls, account security and the handling of customer information across a multi-tenant environment.

Testing was conducted from 14 to 18 September 2026 using a grey-box approach in staging. A targeted retest on 28 September evaluated the agreed corrections. This report presents the initial observations and subsequent verification outcomes.

Scope

- **Web application and API:** authenticated and unauthenticated workflows, including tenant and role boundaries.
- **Account and session security:** sign-in, recovery, session termination and token handling.
- **Customer information:** invoice access and delivery, browser-side storage and relevant security controls.

Limitations

Testing was limited to the staging environment and available workflows. Production-only approval flows and outbound integrations could not be tested. Production infrastructure, payment processing and third-party systems were outside scope. Conclusions apply to the tested releases and workflows.

Positive Observations

The tested administrator sign-in flow required multi-factor authentication, and password-reset tokens expired after use. Sampled upload controls rejected executable extensions. These measures provided useful protection within the workflows examined.

Key Findings

The main weaknesses concerned access control and session lifecycle management. In the initial assessment, an ordinary member could access another tenant's billing information and obtain elevated permissions within their own workspace. Additional observations concerned account recovery, local storage of sensitive documents and browser policy enforcement.

The retest confirmed corrections to both high-severity findings and the account-recovery issue. Session revocation remained partially effective; local document caching and browser policy enforcement required further action. The detailed findings describe the supporting evidence and recommended corrections.

ASSESSMENT AT A GLANCE

2. Dashboard

6
validated findings

3
verified fixed

Control effectiveness

INITIAL ASSESSMENT
Ineffective

AFTER TARGETED RETEST
Partially effective

Initial testing demonstrated bypasses of tenant isolation and role enforcement. The retest verified those corrections, but incomplete token revocation and remaining browser-side gaps prevented an Effective assessment for the tested scope.

Severity breakdown

Critical	0
High	2
Medium	2
Low	1
Informational	1

Category breakdown

Access control	2
Session management	1
Authentication	1
Data protection	1
Hardening	1

Each finding is counted once, using its original severity and one primary category.

Verification outcomes



Verified fixed	3	Partially fixed	1
Not fixed	1	Not retested	1

Targeted retest: September 28, 2026. Finding-level outcomes and recommendations appear in section 6.

Control effectiveness is an assessor judgement limited to the tested scope and releases; it is not an average of CVSS scores.

ENGAGEMENT CONTEXT

3. Project summary & scope

Meridian Workspace is a fictional multi-tenant SaaS application. The engagement examined the web interface, REST API and invoice-delivery flow using two seeded tenants and five access contexts. The same asset set was used for the targeted retest.

Testing	September 14, 2026 - September 18, 2026
Effort	8 person-days + 1 retest day
Initial release	`2026.09.14`
Retest release	`2026.09.28`
Prepared / reviewed	Daniel Reed / Sofia Bennett

Project timeline

Date	Milestone
Sep 11, 2026	Scope and rules agreed
Sep 14, 2026	Testing begins
Sep 15, 2026	Interim update: initial observations shared
Sep 17, 2026	Interim update: progress and outstanding checks shared
Sep 18, 2026	Initial testing completed
Sep 21, 2026	Initial findings delivered
Sep 28, 2026	Targeted retest and cleanup
Sep 29, 2026	Final technical handover

In-scope assets

Asset	Target and boundary
Web application	`app.meridian.example` Login, users, projects and billing
REST API	`api.meridian.example/v1` 42 documented routes; two tenants
Invoice delivery	`files.meridian.example` Invoice downloads initiated by the app

All .example targets are reserved specimen identifiers and were not contacted.

SCOPE & CONDITIONS

Access, boundaries and limitations

Accounts and access contexts

Account / alias	Role	Purpose
`usr-a-member`	Member / tenant A	Least-privileged access and escalation
`usr-a-admin`	Administrator / tenant A	Positive controls and privileged flows
`usr-b-member`	Member / tenant B	Cross-tenant negative controls
`usr-b-admin`	Administrator / tenant B	Seeded invoice ownership baseline
`Anonymous`	Unauthenticated	Public routes and recovery

Rules of engagement

The scenario permitted manual validation with seeded data. Validation stopped once sufficient evidence had been collected. Automated requests were capped at two per second. Emma Clarke could pause testing through the dedicated DEMO-ENGAGEMENT channel. No incident or scope change occurred.

Out of scope

Cloud control plane, operating systems, mobile applications, employee endpoints, live payment processing and third-party systems were outside scope. Production access, bulk extraction, denial-of-service, social engineering, credential spraying and persistence were excluded.

Assessment limitations

This is a time-bounded assessment of one staging release and a targeted retest of a later release. Production-only approval workflows and outbound integrations were unavailable for testing. New deployments, alternative paths and untested integrations may introduce other risks.

All clients, identities, findings, screenshots and retest records in this demonstration are fictional. They provide no assurance about a real organisation. The reserved .example addresses were not contacted.

HOW THE ASSESSMENT WAS PERFORMED

4. Methodology & approach

The assessment combines application mapping, role and tenant boundary analysis, manual validation and limited automation. Testing was guided by OWASP WSTG v4.2, with API risk labels drawn from API Security Top 10 2023. The following overview describes the assessment areas and practical constraints.

01 **Prepare**

Agree scope, access, test accounts, seeded data and stop conditions.

02 **Map**

Enumerate the documented interface, API routes and business workflows.

03 **Test unauthenticated access**

Review public routes, recovery flows and exposed information.

04 **Test authenticated boundaries**

Compare roles and tenants; exercise authorisation, token lifecycle and business logic.

05 **Validate and report**

Record minimal evidence, explain impact, check ratings and review recommendations.

06 **Retest**

Verify the original case and positive controls on the specified remedial build.

Evidence and tooling

The simulated workflow uses an intercepting HTTP proxy, browser developer tools and scripted requests. Screenshots are illustrative mock-ups. No tools were executed against client systems, so actual tool versions and original traffic captures are not available. In a real engagement, evidence records would include timestamps, build information and tool versions.

ASSESSMENT APPROACH

Assessment areas and limitations

The areas below describe the work performed and the constraints encountered within the agreed scope.

Area	Testing performed	Limitations
Authentication	Reviewed sign-in, account recovery and password-reset token handling.	Available staging workflows; no credential spraying.
Authorisation	Exercised tenant isolation, role boundaries and ownership checks for customer information.	Seeded tenants, accounts and roles within the agreed scope.
Session lifecycle	Validated logout behaviour and token revocation across invoice and export routes.	Token types and routes available in the tested releases.
Input and files	Examined representative input handling and file-upload controls.	Sampled inputs and upload paths; no exhaustive payload coverage.
Business flows	Reviewed accessible invoice and workspace-administration workflows.	Production-only approval flows were unavailable.
Browser protections	Reviewed invoice caching and browser security policy headers.	Tested browser environment and in-scope document flows.
Integrations	Dynamic testing was not performed.	Outbound test connector unavailable in staging.

> **Reading this overview**
> Listing an area does not imply exhaustive testing or that its controls passed. Findings and conclusions are supported by the evidence described in the report.

Method references

- FIRST CVSS v4.0 specification
- OWASP Web Security Testing Guide v4.2
- OWASP API Security Top 10 2023
- OWASP Application Security Verification Standard

ASVS is a supporting reference; no full ASVS level or complete WSTG coverage is claimed. CVSS is owned by FIRST and used by permission.

PRIORITISED RESULTS

5. Findings overview

Findings are grouped by original severity and link to their supporting evidence. Next steps identify the work still required after retest.

Section	Finding / next step	Severity CVSS-B	Verification
6.1	**Cross-tenant invoice disclosure** Access control	**High** 7.1	Verified fixed
6.2	**Member-to-administrator privilege escalation** Access control	**High** 8.6	Verified fixed
6.3	**Access tokens remain valid after logout** Session management **Next step:** Complete token revocation on the export route.	**Medium** 5.3	Partially fixed
6.4	**Account enumeration through password recovery** Authentication	**Medium** 6.9	Verified fixed
6.5	**Invoice PDFs remain cached on a shared workstation** Data protection **Next step:** Prevent local caching of sensitive invoice PDFs.	**Low** 2.0	Not fixed
6.6	**Content Security Policy is report-only** Hardening **Next step:** Validate and enable an enforcing CSP.	**Informational** N/A	Not retested

Reading the ratings

CVSS 4.0 Base scores describe technical severity. Difficulty describes the conditions needed to exploit a finding; it is separate from impact. Definitions appear in section 7.

> **Boundary of the conclusions**
> Evidence supports the behaviours described in each finding. Untested consequences are not counted as demonstrated impact. Remediation does not change the original severity.

TECHNICAL ASSESSMENT

6. Detailed findings

HIGH SEVERITY FINDINGS

6.1 Cross-tenant invoice disclosure

Severity	CVSS 4.0 Base	Difficulty
High	7.1	Low

Target

`GET https://api.meridian.example/v1/invoices/{id}`

Classification: CWE-639; API1:2023

CVSS 4.0: `CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N`

Vulnerability details

The invoice lookup authenticated the caller but did not enforce ownership by the caller’s tenant. Changing the invoice ID returned a seeded invoice from a different tenant. Three controlled cross-tenant reads succeeded; bulk extraction was not attempted.

Business impact

The returned records contained seeded customer names, billing addresses, tax references and invoice totals. This demonstrates disclosure of confidential billing data across tenants. Modification, payment-card exposure and total database extraction were not demonstrated.

> **Exploit scenario**
>
> An ordinary member of tenant A obtains a valid invoice identifier belonging to tenant B. The member submits it to the invoice API using their own token. Because the lookup omits a tenant-ownership check, the response contains the other tenant's billing record. The demonstration used three seeded invoices.

Rating rationale

Network access with a low-privilege account and no user interaction. Confidentiality is High because the boundary failure exposes complete sensitive billing records. No write or availability effect was established.

Evidence and validation

1. Using usr-b-admin, confirm that INV-B-1042 belongs to tenant B.
2. Using usr-a-member, request that same identifier without changing the tenant context.
3. Compare the returned invoice with the owner baseline; repeat for two other seeded invoices.

6.1 Cross-tenant invoice disclosure SYNTHETIC EXAMPLE

HTTP inspector / Invoice lookup

REQUEST · tenant A / member

```
GET /v1/invoices/INV-B-1042 HTTP/2
Host: api.meridian.example
Authorization: Bearer [REDACTED_A_MEMBER]

Authenticated tenant: A
Requested invoice owner: tenant B
```

RESPONSE · foreign invoice returned

```
HTTP/2 200 OK
Content-Type: application/json
{
"invoice": "INV-B-1042",
"tenant": "B",
"amount": 1250.00, "currency": "EUR"
}
```

15 Sep 2026 10:14 UTC · initial observation

Fictional data · no live-system capture

A tenant A token returns a tenant B invoice.

Recommendations

- Derive the tenant scope from the authenticated identity and apply it to every invoice lookup, download and export.
- Deny access by default and use consistent, non-disclosing error responses.
- Add two-tenant positive and negative regression cases. Random identifiers do not replace authorisation checks.

Verification criteria

Cross-tenant reads, downloads and exports return a non-disclosing denial. Owner requests still succeed. Verify both tenant directions.

RETEST · SEPTEMBER 28, 2026

Verified fixed

Verified fixed on 2026.09.28. Six cross-tenant read/download/export requests returned non-disclosing 404 responses. The two owner controls still returned 200. No equivalent bypass was identified in these checked paths.

Additional resources

- OWASP API1:2023 - Broken Object Level Authorization
- OWASP Authorization Cheat Sheet

HIGH SEVERITY · ACCESS CONTROL

6.2 Member-to-administrator privilege escalation

Severity	CVSS 4.0 Base	Difficulty
High	8.6	Low

Target

`PATCH https://api.meridian.example/v1/members/{id}`

Classification: CWE-915; API3:2023

CVSS 4.0: `CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N`

Vulnerability details

The member-profile update route accepted a role property supplied by the caller. A member assigned the administrator role to their own identity and gained access to tenant administration functions. The same field could be submitted directly even though the web UI did not expose it.

Business impact

The account obtained the tenant administrator’s user list and could modify a seeded workspace setting. This could enable unauthorised user-management and workspace-data changes. Cross-tenant administrator access was not demonstrated.

Exploit scenario

A member of tenant A sends a profile update containing an administrator role. The API writes the role without checking whether the caller may grant it. The same account then accesses tenant administration and changes a seeded workspace setting. The boundary crossed is within that tenant.

Rating rationale

Low privileges and no user interaction are required. The resulting administrator access allows broad disclosure and modification within the same tenant, so VC and VI are High. Availability impact was not established.

Evidence and validation

1. Record usr-a-member as role=member and confirm that GET /v1/admin/users returns
2. Submit an update to that member record with the role set to admin.
3. Request the administration route again and verify a reversible change to a seeded workspace setting.

6.2 Member-to-administrator privilege escalation SYNTHETIC EXAMPLE

HTTP inspector / Role update

REQUEST · member self-service

```
PATCH /v1/members/usr-a-member HTTP/2
Host: api.meridian.example
Authorization: Bearer [REDACTED_A_MEMBER]
Content-Type: application/json

{"role":"admin"}
```

RESPONSE + FOLLOW-UP · same account

```
HTTP/2 200 OK
{"id":"usr-a-member","role":"admin"}

GET /v1/admin/users
Before role update: 403 Forbidden
After role update: 200 OK
Scope: tenant A only
```

15 Sep 2026 13:32 UTC · initial observation

Fictional data · no live-system capture

A member gains access to the administration route.

Relevant HTTP exchange

```
PATCH /v1/members/usr-a-member HTTP/2
Host: api.meridian.example
Authorization: Bearer [REDACTED_MEMBER]
Content-Type: application/json

{"display_name":"Sample Member","role":"admin"}

HTTP/2 200 OK
```

Recommendations

- Allow-list fields that members may update on their own profile.
- Move role changes to a dedicated administrator operation with server-side permission and tenant checks.
- Reject privilege fields on self-service routes and record authorised role changes in the audit log.

Verification criteria

Self-service requests cannot change roles. An authorised administrator can change a role through the dedicated operation, within their own tenant.

RETEST · SEPTEMBER 28, 2026

Verified fixed

Verified fixed on 2026.09.28. Role changes submitted through self-service routes returned 403 and left the stored role unchanged. An authorised administrator could still change the role through the dedicated route.

Additional resources

- OWASP Mass Assignment Cheat Sheet

MEDIUM SEVERITY FINDINGS

6.3 Access tokens remain valid after logout

Severity	CVSS 4.0 Base	Difficulty
Medium	5.3	Low

Target

`https://api.meridian.example/v1/invoices/{id} and /v1/invoices/export`

Classification: CWE-613; API2:2023

CVSS 4.0: `CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

Vulnerability details

Logout cleared the browser session but did not revoke the account's read-only personal API token. A token captured before logout continued to retrieve invoice data for its own tenant. The initial validity window was 60 minutes.

Business impact

A copied token could retain read access after the legitimate user logs out. The scenario does not demonstrate token theft or an attacker's ability to mint tokens; it demonstrates the failure to invalidate an already issued token.

Exploit scenario

A person already holding the member's read-only API token continues using it after the member logs out. Before remediation, invoice reads remained available until the 60-minute expiry. At retest, the remaining export route still accepted the token for up to five minutes. Token theft is a prerequisite, not a demonstrated finding.

Rating rationale

Token possession is treated as low privileges. The demonstrated token is read-only and exposes a limited billing subset, so VC is Low; no integrity or availability effect was shown.

Evidence and validation

1. Issue a read-only personal API token for usr-a-member and record a successful invoice request.
2. Log out and confirm the browser session no longer accesses the UI.
3. Repeat the captured token request; verify acceptance at 1, 15 and 59 minutes, and expiry after 60 minutes.

6.3 Access tokens remain valid after logout SYNTHETIC EXAMPLE

HTTP inspector / Session revocation

LOGOUT · browser session ended	REPLAY · previous read-only token
`POST /v1/auth/logout HTTP/2` `Host: api.meridian.example`	`GET /v1/invoices/INV-A-1008 HTTP/2` `Authorization: Bearer [REDACTED_PRE_LOGOUT]`

`HTTP/2 204 No Content`	`+01 min 200 OK · invoice returned`
	`+15 min 200 OK · invoice returned`
	`+59 min 200 OK · invoice returned`

`Browser UI: signed out` `Previous API token retained for validation`	`+60 min 401 · token expired`

16 Sep 2026 09:41 UTC · initial observation — Fictional data · no live-system capture

The previous API token still works after logout.

Recommendations

- Apply central revocation checks to every endpoint accepting this token class, including exports.
- Document the lifecycle of personal API tokens and revoke them on the security-state changes promised to users.
- Keep short expiries as a supporting control; verify revocation across every issuer and route.

Verification criteria

After logout, the previous token fails on invoice read, PDF download and export routes. A newly issued authorised token still works. Test both tenants and all read-only token issuers.

> RETEST · SEPTEMBER 28, 2026
>
> ### Partially fixed
>
> Partially fixed. Standard invoice reads rejected the logged-out token, but GET /v1/invoices/export still accepted it. The token lifetime was reduced to five minutes. Central revocation coverage is due 2 October; the original Medium rating is retained.

Additional resources

- OWASP Session Management Cheat Sheet

MEDIUM SEVERITY · AUTHENTICATION

6.4 Account enumeration through password recovery

Severity	CVSS 4.0 Base	Difficulty
Medium	6.9	Low

Target

`POST https://api.meridian.example/v1/auth/recovery`

Classification: CWE-204; API2:2023

CVSS 4.0: `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

Vulnerability details

The unauthenticated password-recovery route returned different messages for registered and unregistered addresses. This disclosed whether an address belonged to a user. Ten seeded registered identities and ten unregistered aliases were tested at the agreed request rate.

Business impact

An unauthenticated caller could identify registered accounts and refine phishing or credential-abuse targeting. Password disclosure, recovery-token prediction and account takeover were not demonstrated.

Exploit scenario

An unauthenticated caller submits candidate email addresses to password recovery. The API gives different replies for known and unknown users. The caller can infer account membership and use that information to refine targeting; account takeover was not demonstrated.

Rating rationale

Network access, no privileges and no user interaction are required. Only account membership is disclosed, so VC is Low.

Evidence and validation

1. Submit the seeded registered alias member-a@meridian.example.
2. Submit the unregistered alias nobody-01@meridian.example with the same request structure.
3. Repeat across the 20 agreed aliases and compare the response body and status.

6.4 Account enumeration through password recovery SYNTHETIC EXAMPLE

HTTP inspector / Response comparison

REGISTERED ADDRESS · sample 1 / 10	UNREGISTERED ADDRESS · sample 1 / 10

`POST /v1/auth/recovery HTTP/2`
`Content-Type: application/json`
 	
`{"email":"member-a@meridian.example"}`
	`POST /v1/auth/recovery HTTP/2`
	`Content-Type: application/json`
	
	`{"email":"nobody-01@meridian.example"}`

`HTTP/2 202 Accepted` `{"message":"Recovery email sent"}`	`HTTP/2 404 Not Found` `{"error":"Account not found"}`

16 Sep 2026 14:05 UTC · initial observation — Fictional data · no live-system capture

Registered and unknown accounts return different responses.

Recommendations

- Return the same status, message and externally observable flow for known and unknown accounts.
- Keep request-rate controls and recovery-abuse monitoring.
- Review timing separately; identical wording alone does not remove every observable difference.

Verification criteria

Registered and unregistered aliases return the same status and body. Review timing and request-rate behaviour with an agreed validation sample.

RETEST · SEPTEMBER 28, 2026

Verified fixed

Verified fixed for the tested response channel. All 20 aliases returned the same 202 status and generic body. No reliable timing split appeared in the small validation sample; this was not a statistical proof against every timing side channel.

Additional resources

- OWASP Forgot Password Cheat Sheet

LOW SEVERITY FINDINGS

6.5 Invoice PDFs remain cached on a shared workstation

Severity	CVSS 4.0 Base	Difficulty
Low	2.0	Low

Target

`https://app.meridian.example/invoices/{id}.pdf`

Classification: CWE-525

CVSS 4.0: `CVSS:4.0/AV:L/AC:L/AT:P/PR:L/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

Vulnerability details

Invoice PDF responses permitted private browser caching. After a seeded invoice had been viewed and the user had logged out, the same local browser profile could reopen the PDF while offline. The issue affects previously viewed documents, not arbitrary invoices.

Business impact

A person with local access to the same browser profile could view previously opened billing information after logout. Remote access, retrieval from another profile and operating-system protection bypass were not demonstrated.

Exploit scenario

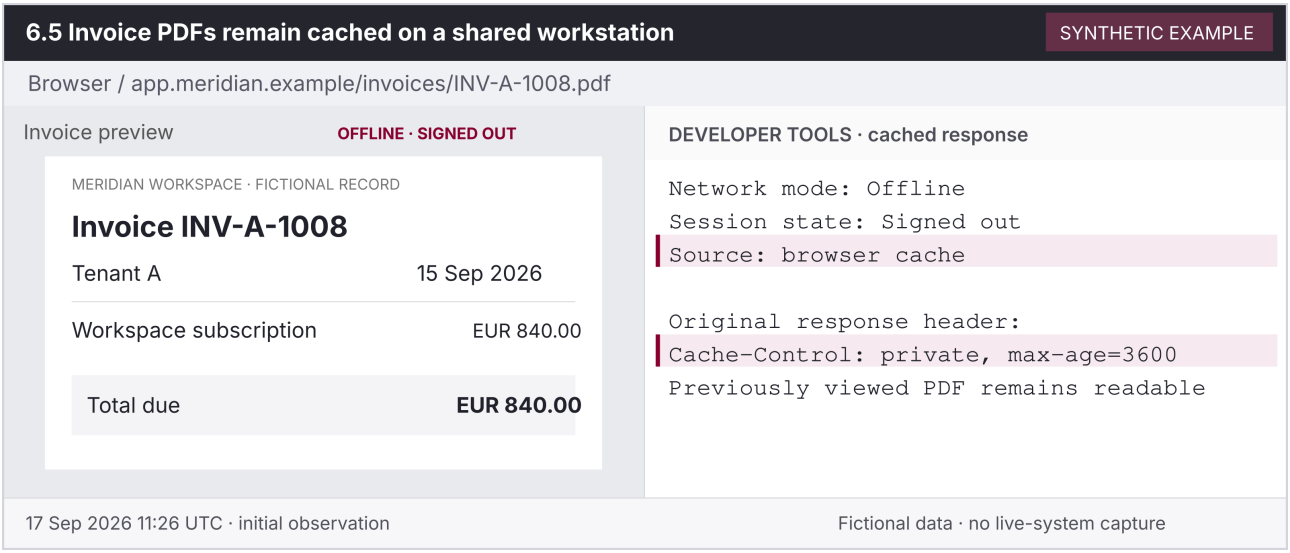
A user views an invoice and logs out on a shared workstation. Someone with access to the same browser profile later opens that invoice from browser history while offline. The cached PDF remains readable. The scenario requires a previously viewed invoice and local access.

Rating rationale

AV:L and PR:L reflect local profile access. AT:P captures the prerequisite that the sensitive document is already cached. Confidentiality impact is limited to the cached content; integrity and availability are unaffected.

Evidence and validation

1. Open seeded invoice INV-A-1008 in the browser and confirm the PDF response is cacheable.
2. Log out, close the invoice tab and disconnect the browser from the network.
3. Reopen the invoice URL from history in the same profile and confirm that the seeded PDF remains readable.



A cached invoice remains readable offline after logout.

Recommendations

- Send Cache-Control: no-store on sensitive invoice responses.
- Review service-worker and application-managed caches; clear relevant entries at logout.
- Verify history and offline behaviour in a fresh browser profile without breaking authorised downloads.

Verification criteria

A fresh browser profile does not retain sensitive PDFs in application-managed caches. After logout and disconnection, the tested app/history path cannot reopen them.

RETEST · SEPTEMBER 28, 2026

Not fixed

Not fixed. The retest reproduced offline access on 2026.09.28. The issue remains assigned to Frontend Engineering for 9 October. Shared-device use is discouraged as a temporary operational measure; this is not recorded as a technical fix.

Additional resources

- OWASP Session Management - Web Content Caching

INFORMATIONAL SEVERITY FINDINGS

6.6 Content Security Policy is report-only

Severity	CVSS 4.0 Base	Difficulty
Informational	N/A	Not applicable

Target

`https://app.meridian.example/login and /workspace`

Classification: CWE-693; hardening observation

CVSS 4.0: Not applicable

Vulnerability details

The application delivered Content-Security-Policy-Report-Only without an enforcing Content-Security-Policy header. Violation reports were collected, but disallowed resources were not blocked by the browser.

Business impact

The application lacks the additional resource restriction that an enforcing policy could provide if an injection weakness is introduced. This observation does not establish script execution, data access or account compromise.

Exploit scenario

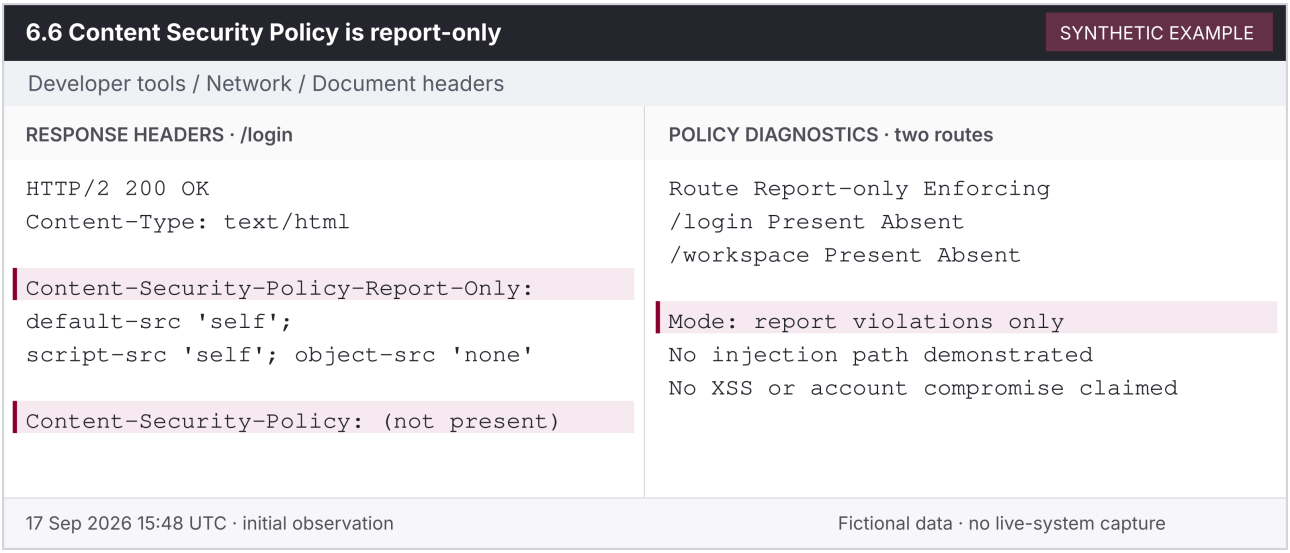
No independent exploit was demonstrated. A future injection weakness could have greater impact because the report-only policy records violations without blocking resources. This is a hardening observation, not a confirmed cross-site scripting vulnerability.

Rating rationale

No exploitable injection path was demonstrated and no independent impact was established. This is an Informational hardening observation with no CVSS score, not a claim of confirmed XSS.

Evidence and validation

1. Inspect the document response on the login and workspace pages.
2. Confirm the presence of the report-only header and the absence of an enforcing policy.
3. Use the browser’s policy diagnostics to confirm reporting mode; do not infer an injection vulnerability.



CSP reporting is enabled without an enforcing policy.

Recommendations

- Use report telemetry to build a restrictive policy, favouring nonces or hashes where required.
- Test payment and support integrations before enabling enforcement.
- Retain input handling and output encoding. CSP provides an additional layer of protection.

Verification criteria

An enforcing policy blocks a controlled violation. Required payment and support integrations pass the agreed regression tests.

> RETEST · SEPTEMBER 28, 2026
>
> ### Not retested
>
> Not retested. No enforcing-policy build was available on 28 September. Platform Engineering scheduled compatibility testing for 16 October. The lack of retest is not counted as a verified fix.

Additional resources

- OWASP Content Security Policy Cheat Sheet

RATING METHODOLOGY

7. Field definitions

Technical severity uses CVSS 4.0 Base scores. Threat and Environmental metrics are not supplied in this sample. Informational observations are a separate reporting category without a numerical score.

Severity	CVSS-B	Reading the result
Critical	9.0–10.0	Highest technical severity; prioritise urgent review and containment.
High	7.0–8.9	Material compromise potential in the stated attack conditions.
Medium	4.0–6.9	Meaningful exposure requiring planned correction.
Low	0.1–3.9	Limited technical impact or constrained attack conditions.
Informational	N/A	Hardening observation without independently demonstrated impact.

Difficulty

Low | Medium | High | Not applicable

Low: straightforward once prerequisites are met. **Medium:** additional coordination, timing or preparation. **High:** specialised knowledge or restrictive conditions. **Not applicable:** no exploit scenario is asserted. Green means harder to exploit; severity describes impact separately. Difficulty does not replace CVSS Attack Complexity.

Control effectiveness

Effective	Tested key controls performed their intended function.
Partially effective	Controls worked in part, with material gaps remaining.
Ineffective	Key protective controls were demonstrably bypassed.
Not assessed	Testing was insufficient to support a conclusion.

A manual, evidence-based judgement for the tested scope and releases. It is not a CVSS average or an assurance rating for the entire organisation.

Verification status

Verified fixed
The original case and relevant positive controls passed on the listed build.

Partially fixed
A related path or part of the issue remains.

Not fixed
The issue reproduced on the retest build.

Not retested
No verification conclusion was reached.

Risk acceptance is a separate business decision. A verified fix does not establish that the application is free of vulnerabilities.

QUESTIONS AND FOLLOW-UP

8. Contact information

Technical clarity.
Practical next steps.

BROONAT

security@broonat.com
broonat.com

BROONAT

Daniel Reed
Lead Penetration Tester
daniel.reed@broonat.example

Sofia Bennett
Technical Reviewer
sofia.bennett@broonat.example

Meridian Workspace

Mark Ellis
CTO / Engagement Sponsor
mark.ellis@meridian.example

Emma Clarke
Security Engineer / Technical Contact
emma.clarke@meridian.example

About this demonstration

Named engagement contacts are fictional and use reserved example-domain addresses. The BROONAT contact details above are provided for enquiries. Assessment results and screenshots are fictional.