BRIGHOUSE SECURITY LABS

SaaS / API Security Readiness Snapshot

Fictional public sample for startups, SaaS teams and agencies

FICTIONAL SAMPLE — NOT A CLIENT ASSESSMENT

All organisations, assets, domains, observations and evidence in this document are fabricated. The purpose is to demonstrate BrighouseSec report structure, risk language, prioritisation and remediation guidance for startups, SaaS teams and agencies.

SAMPLE TARGET	Example SaaS Ltd — demo-saas.example
DOCUMENT ID	BSL-SAMPLE-SAAS-002
REPORT TYPE	48-Hour External Security Snapshot / SaaS & API Readiness Review
PREPARED BY	Brighouse Security Labs / BrighouseSec
CLASSIFICATION	Public sample / fictional / non-confidential
CONTACT	sales@brighousesec.com

Realistic finding style. Clear remediation. Scope agreed first. No call required where suitable.

Important Sample Notice

This document is a fictional public sample. It must not be interpreted as a security assessment of any real organisation. The .example domains are placeholders reserved for documentation and demonstration material.

A real BrighouseSec engagement begins only after written scope, target confirmation, authorisation and payment confirmation. The report below demonstrates a professional structure for founder-facing SaaS security findings: scope, methodology, prioritised risk register, evidence-led observations and practical remediation guidance.

Executive Summary

This SaaS / API Security Readiness Snapshot demonstrates how BrighouseSec would communicate low-touch external findings to an early-stage SaaS founder, agency or product team. The fictional scenario includes a marketing site, application login surface, public API subdomain, status page and domain-level email security records.

The simulated review identified a moderate external security posture. No critical issues are presented in this sample. The most important themes are exposure management, email trust hardening, defensive HTTP headers, API error-handling consistency and login-flow security hygiene. These issues are typical of fast-moving product teams and can normally be remediated without a major architectural change.

OVERALL SAMPLE RISK	ACTION REQUIRED	CLIENT-FACING SUITABILITY
Moderate	**Yes**	**Good**
No critical items	Within 7 days	after remediation

Founder-Facing Priority Actions

PRIORITY	ACTION	REASON	SUGGESTED OWNER
1	Protect public staging surfaces	Reduces avoidable exposure before launch or handover	Product / Engineering
2	Move DMARC toward enforcement	Improves domain trust and reduces spoofing risk	Operations / IT
3	Add a baseline security header policy	Hardens browser-side behaviour for users	Engineering
4	Normalise API and password-reset responses	Reduces information leakage and enumeration risk	Engineering
5	Document a lightweight external exposure checklist	Prevents repeat issues during future releases	Founder / Product

Scope and Limitations

SCOPE-FIRST MODEL	For a real review, BrighouseSec confirms assets, testing restrictions, authorisation language and delivery expectations in writing before any work begins.
REVIEW STYLE	Low-touch external checks. No exploitation, no brute force, no destructive testing and no attempt to access customer data.
SAMPLE ASSUMPTION	A founder or web agency requested a readiness snapshot before launch, investor review, client handover or early customer onboarding.

Sample Assets in Scope

ASSET	PURPOSE	REVIEW TYPE
demo-saas.example	Marketing website and domain-level public footprint	Public surface and DNS review
app.demo-saas.example	Application login and public user flow	Unauthenticated observation only
api.demo-saas.example	Public API base URL	Safe unauthenticated behaviour review
status.demo-saas.example	Public status page	Metadata and exposure review
Mail/domain DNS records	SPF, DKIM, DMARC, MX and alignment checks	Passive DNS configuration review

Out of Scope for This Sample

- No authenticated application testing, credentialed access or password sharing.
- No source-code review, cloud-console review, internal-network testing or endpoint testing.
- No social engineering, phishing, denial-of-service testing or destructive scanning.
- No vulnerability chaining, exploit development or attempts to access data.
- No compliance certification, legal assurance or guarantee that all vulnerabilities would be found.

Methodology

PHASE	OBJECTIVE	EXAMPLE ACTIVITIES
1. Scope validation	Confirm assets, ownership and restrictions	Written scope, target list, testing boundaries, permission statement
2. External footprint review	Understand visible exposure without intrusive activity	Domain/subdomain review, public metadata, staging indicators
3. Domain trust checks	Review email and DNS trust posture	SPF, DKIM, DMARC, MX and domain alignment observations
4. Web hardening review	Identify visible browser-facing weaknesses	TLS posture, HTTP security headers, public endpoints, login exposure
5. API readiness review	Check safe unauthenticated API behaviour	Error handling, response consistency, broad CORS indicators, rate-limit signals
6. Prioritised reporting	Convert observations into an action plan	Risk rating, business impact, fix guidance, verification steps

Risk Rating Model

Findings are prioritised using a practical combination of technical severity, likelihood, public exposure, business relevance and remediation effort. CVSS-style scores are included where useful, but business priority remains the deciding factor for a founder-facing action plan.

RATING	DEFINITION	TYPICAL RESPONSE EXPECTATION
High	A weakness that could plausibly lead to unauthorised access, data exposure or serious business disruption.	Fix immediately or pause launch/handover until controlled.
Medium	A weakness that increases attack likelihood or reduces resilience, especially when combined with other issues.	Fix within 7-14 days or before launch/client handover.
Low	A hardening gap, metadata exposure or configuration weakness with limited direct impact.	Fix during the next planned engineering cycle.
Informational	Useful context or positive observation that does not represent a direct vulnerability.	Track as improvement guidance.

Risk Register

ID	FINDING	SEVERITY	ASSET	PRIORITY
F-01	Publicly discoverable staging login surface	**Medium**	staging.demo-saas.example	7-day fix
F-02	DMARC policy in monitoring mode only	**Medium**	demo-saas.example DNS/email posture	7-day fix
F-03	Incomplete browser security header baseline	**Medium**	app.demo-saas.example	7-day fix
F-04	Verbose API validation errors reveal implementation details	**Medium**	api.demo-saas.example	14-day fix
F-05	Password reset responses may allow account enumeration	**Medium**	app.demo-saas.example/ reset-password	14-day fix
F-06	Broad CORS policy observed on public API endpoint	**Medium**	api.demo-saas.example/v1/ public/config	14-day fix
F-07	Public status/config pages expose internal service labels	**Low**	status.demo-saas.example	Next cycle

Risk Distribution

HIGH	MEDIUM	LOW
0	**6**	**1**
none in sample	launch-readiness issues	hardening improvement

F-01 MEDIUM Publicly discoverable staging login surface

CVSS V3.1	ASSET	PRIORITY	REVIEW TYPE
6.5 / 10	**staging.demo-saas.example**	**7-day fix**	**Low-touch external**

OBSERVATION

The sample review identified a staging-style login surface reachable from the public internet. The page contained branding and release references suggesting it belonged to the same SaaS product family as the production application.

BUSINESS IMPACT

Public staging interfaces can expose unreleased features, configuration differences, test accounts, verbose errors or weaker access controls. Even where authentication is present, unnecessary public exposure gives attackers an additional place to inspect the product and test assumptions.

SAMPLE EVIDENCE

```
GET https://staging.demo-saas.example/login  ->  200 OK
title: Demo SaaS Staging Portal
x-environment: staging
```

RECOMMENDED REMEDIATION

- Restrict staging environments using network allowlisting, identity-aware access or VPN.
- Remove release identifiers, environment labels and debug banners from public pages.
- Add staging exposure to the release/handover checklist.

F-02 MEDIUM DMARC policy in monitoring mode only

CVSS V3.1	ASSET	PRIORITY	REVIEW TYPE
5.3 / 10	**demo-saas.example DNS/email posture**	**7-day fix**	**Low-touch external**

OBSERVATION

The domain publishes a DMARC record with p=none. This is monitoring-only mode and does not instruct receiving mail servers to quarantine or reject messages that fail alignment.

BUSINESS IMPACT

Monitoring-only DMARC is useful for visibility, but it does not provide enforcement. For a SaaS product that sends onboarding, invoices, password resets or customer notifications, weak domain trust increases spoofing and impersonation risk.

SAMPLE EVIDENCE

```
dig +short TXT _dmarc.demo-saas.example
"v=DMARC1; p=none; rua=mailto:dmarc-reports@demo-saas.example"
```

RECOMMENDED REMEDIATION

- Confirm all legitimate senders and align SPF/DKIM first.
- Move gradually from p=none to p=quarantine, then p=reject.
- Monitor aggregate reports during the transition.

F-03 MEDIUM Incomplete browser security header baseline

CVSS V3.1	ASSET	PRIORITY	REVIEW TYPE
5.4 / 10	**app.demo-saas.example**	**7-day fix**	**Low-touch external**

OBSERVATION

The application response did not include a complete security header baseline. The sample response was missing or weakening CSP, HSTS, X-Frame-Options, X-Content-Type-Options and Referrer-Policy controls.

BUSINESS IMPACT

These headers provide browser-side guardrails. Missing controls increase exposure to clickjacking, MIME sniffing, referrer leakage and content injection containment failures.

SAMPLE EVIDENCE

```
curl -sSI https://app.demo-saas.example/
Strict-Transport-Security: MISSING
Content-Security-Policy: MISSING
X-Content-Type-Options: MISSING
```

RECOMMENDED REMEDIATION

- Add a tested response header policy at the app or CDN layer.
- Use HSTS only after confirming all subdomains support HTTPS reliably.
- Deploy CSP in report-only mode first, then enforce once tested.

F-04 MEDIUM Verbose API validation errors reveal implementation details

CVSS V3.1	ASSET	PRIORITY	REVIEW TYPE
5.3 / 10	**api.demo-saas.example**	**14-day fix**	**Low-touch external**

OBSERVATION

Unauthenticated API requests returned verbose validation and implementation details. The sample response exposed field names, internal validation classes and stack-style messages.

BUSINESS IMPACT

Verbose errors can help attackers map expected inputs, internal logic and potential attack paths. This does not prove exploitability by itself, but it reduces the effort required to probe the API intelligently.

SAMPLE EVIDENCE

```
POST /v1/account/check
{"error":"ValidationError","field":"organisation_id","validator":"UUIDv4Required"}
```

RECOMMENDED REMEDIATION

- Return generic client-facing error messages for unauthenticated users.
- Log detailed diagnostic information server-side only.
- Create an API error taxonomy for production responses.

F-05 MEDIUM Password reset responses may allow account enumeration

CVSS V3.1	ASSET	PRIORITY	REVIEW TYPE
5.4 / 10	**app.demo-saas.example/reset-password**	**14-day fix**	**Low-touch external**

OBSERVATION

The password reset flow returned different responses for existing and non-existing accounts. In the sample scenario, timing and message content differed between the two cases.

BUSINESS IMPACT

Account enumeration allows attackers to confirm valid users before attempting credential stuffing, phishing or password reset abuse. This is especially relevant for SaaS products with high-value business users.

SAMPLE EVIDENCE

```
POST /reset-password email=valid@example.com -> "Reset email sent"
POST /reset-password email=unknown@example.com -> "No account found"
```

RECOMMENDED REMEDIATION

- Return a single generic response regardless of account existence.
- Rate-limit password reset attempts by IP, account and device fingerprint where appropriate.
- Monitor spikes in reset requests as a suspicious activity signal.

F-06 MEDIUM Broad CORS policy observed on public API endpoint

CVSS V3.1	ASSET	PRIORITY	REVIEW TYPE
5.0 / 10	**api.demo-saas.example/v1/public/config**	**14-day fix**	**Low-touch external**

OBSERVATION

A public API endpoint returned broad CORS headers in the sample response. The configuration suggested cross-origin access was allowed more widely than required.

BUSINESS IMPACT

Broad CORS is not automatically exploitable, but it can increase risk when combined with weak authentication, token handling issues or sensitive responses. CORS should be explicitly designed, not left as a permissive default.

SAMPLE EVIDENCE

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Allow-Headers: *
```

RECOMMENDED REMEDIATION

- Restrict CORS origins to known production frontends.
- Avoid wildcard origins for authenticated or sensitive endpoints.
- Document the intended CORS policy as part of API deployment review.

F-07 LOW Public status/config pages expose internal service labels

CVSS V3.1	ASSET	PRIORITY	REVIEW TYPE
3.7 / 10	**status.demo-saas.example**	**Next cycle**	**Low-touch external**

OBSERVATION

The public status/config page exposed internal service labels and environment-style names in the sample scenario. These labels did not include secrets, but they disclosed internal service structure.

BUSINESS IMPACT

Internal naming gives attackers context. It may reveal vendors, microservice boundaries or deployment patterns that make future probing more efficient. This is low risk in isolation, but easy to tidy up.

SAMPLE EVIDENCE

```
GET https://status.demo-saas.example/config.json
{"services":["auth-api-prod","billing-worker","internal-crm-sync"]}
```

RECOMMENDED REMEDIATION

- Replace internal labels with customer-facing service names.
- Review public JSON/config/status endpoints before launch.
- Keep operational detail in private observability tools rather than public pages.

48-Hour Remediation Plan

WHEN	ACTION	OWNER
First 24 hours	Protect staging access, add a basic security header policy, normalise password reset responses.	Engineering
Within 7 days	Move DMARC toward enforcement, restrict API error verbosity, document CORS intent.	Engineering / Operations
Within 14 days	Review all public status/config endpoints, add exposure checks to deployment and handover workflows.	Product / Engineering
Ongoing	Re-run the external snapshot after fixes and before major releases or client handovers.	Founder / Product

What a Client Receives

- A concise PDF report written for founders, product teams and technical owners.
- Prioritised findings with business impact and practical remediation guidance.
- Evidence-led observations that avoid fear-based language and avoid unsupported claims.
- A fixed written scope before testing begins and no destructive testing without explicit approval.
- Optional retest or deeper review if the snapshot identifies high-value follow-up work.

Positioning Note

This sample is designed to show how BrighouseSec communicates findings without exposing real client data. In a real engagement, target names, screenshots, logs and evidence would only be shared with the authorised recipient and would not be published without explicit written permission.

End of Sample Report

Brighouse Security Labs provides low-touch external security snapshots, SaaS/API readiness reviews and written remediation plans for startups, SaaS teams and agencies.

WEBSITE	brighousesec.com
EMAIL	sales@brighousesec.com
ENGAGEMENT MODEL	Fixed scope · written authorisation · no call required where suitable
SAMPLE STATUS	Fictional public sample — not client data