Sample VAPT report: an illustrative example with invented data

This sample VAPT report walks through the structure of a BERRY9 report, from the executive summary to the finding pages and the remediation matrix. It is an illustrative sample built on an invented application with invented findings, so every name, address and result on this page is example data.

EXAMPLE TARGET	EXAMPLE FINDINGS	SCORING	DATA
Example Invoice App	**High, medium, low**	**CVSS 3.1 with vectors**	**Invented, illustrative**

Last reviewed November 4, 2026

- **Executive report**: summary, risk by severity and business impact.
- **Engineering report**: one page per finding, with the CVSS 3.1 vector and a proof of concept.
- **Remediation matrix**: owner, status and retest result for every finding.
- **How to read it**: severity, CVSS vectors and the matrix, explained.

01 / ILLUSTRATIVE SAMPLE

What this sample report shows

The sample follows one invented web application test from the summary to the last row of the matrix. The application, its users, its address and its three findings were made up for this page; only the structure is the one BERRY9 reports use.

Illustrative sample with invented example data. No client, no real system and no finding from any BERRY9 engagement appears on this page. The domain example.com is

reserved for documentation, so the addresses below cannot point to anyone's real system.

The engagement behind the sample

ILLUSTRATIVE SAMPLE: THE INVENTED ENGAGEMENT BEHIND THIS REPORT

Target

EXAMPLE ENTRY

Example Invoice App, an invented web application for sending and paying invoices

Address

EXAMPLE ENTRY

https://invoices.example.com

Test

EXAMPLE ENTRY

Web application penetration test

Tested as

EXAMPLE ENTRY

Signed out, and as billing users in two test organizations, A and B

Environment

EXAMPLE ENTRY

A staging copy holding test data only

Findings

EXAMPLE ENTRY

Three: one high, one medium and one low, numbered F-01 to F-03

The three parts of the report

Executive report

For management: the scope, the findings, their severity and their business meaning, in one short read.

Read the summary →

Engineering report

A page for each finding, written so a developer can reproduce it, understand the impact and apply the fix.

Open finding F-01 →

Remediation matrix

One row per finding with its fix, its owner and its status, kept up to date until the last item is closed.

See the matrix →

This sample is published as a web page, and a PDF copy is available: download the sample report (PDF).

02 / EXECUTIVE REPORT

Executive report: summary, risk and business impact

The executive report is written for founders, managers and auditors who will not read the finding pages. In this illustrative sample it gives the verdict, the count of findings by severity and what each finding means for the business, in plain language.

Executive summary, as the sample words it

We tested the Example Invoice App at invoices.example.com as a visitor who is not signed in and as billing users in two customer organizations. We confirmed three findings: one high, one medium and one low.

The high finding matters most. Any signed-in customer could open and edit invoices that belong to other customers by changing a number in the request. It should be fixed first, and the two other findings scheduled after it.

None of the findings needed special access or tools to find, which is why we recommend an automated check for each one in the build, so they cannot quietly return.

ILLUSTRATIVE SAMPLE: FINDINGS BY SEVERITY

SEVERITY	FINDINGS	IDS
CRITICAL	0	None
HIGH	1	F-01
MEDIUM	1	F-02
LOW	1	F-03

Business impact, in the sample's words

Customer data

Through F-01, every customer's invoices, amounts and billing contacts were open to every other customer: a breach of the data the product exists to hold.

Billing integrity

F-01 also allowed changes. An invoice altered by an outsider before payment is a direct route to disputes and lost revenue.

User sessions

F-02 lets a crafted link run script in a user's session, and F-03 lets someone on the user's network intercept a first visit over plain HTTP.

03 / SUMMARY OF FINDINGS

Summary of findings

The engineering report opens with every finding on one table, highest severity first, each linked to its own page. All three rows of this illustrative sample are invented.

ILLUSTRATIVE SAMPLE: SUMMARY OF FINDINGS

F-01 · Cross-organization access to invoices

BUSINESS IMPACT

Any customer reads and edits other customers' invoices

SEVERITY

HIGH

CVSS 3.1 8.1

RECOMMENDATION

Scope every invoice lookup to the caller's organization

F-02 · Outdated front-end library

BUSINESS IMPACT

A crafted link can run script in a signed-in user's session

SEVERITY

MEDIUM

CVSS 3.1 6.1

RECOMMENDATION

Upgrade to the fixed release and fail builds on known-vulnerable versions

F-03 · Missing HSTS header

BUSINESS IMPACT

A user's first visit over plain HTTP can be intercepted

SEVERITY

LOW

CVSS 3.1 3.1

RECOMMENDATION

Send Strict-Transport-Security on every HTTPS response

04 / FINDING F-01

Finding F-01: an authorization flaw (high)

Illustrative sample finding. A classic way for one customer's data to reach another is an API that checks who you are but not what you may open, and this invented example shows how such a finding is written up.

SAMPLE FINDING F-01 · EXAMPLE INVOICE APP

Invoices of another organization can be read and changed by editing the invoice ID

SEVERITY	CVSS 3.1
HIGH	Base score 8.1 · AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N

AFFECTED COMPONENT	TESTED AS
GET and PUT /api/v1/invoices/{invoiceId}	Billing user, organization A (test account)

PROOF OF CONCEPT

1. Sign in as the billing user of organization A.
2. Open invoice 1001, which belongs to organization A, and note its ID in the request.
3. Repeat the request with ID 2002, an invoice created for this test in organization B: the API returns it in full.
4. Send a PUT to the same address with a new due date: the change is saved to organization B's invoice.

```
PUT /api/v1/invoices/2002 HTTP/1.1
Host: invoices.example.com
Authorization: Bearer <org A token>

{"dueDate": "2026-12-31"}

HTTP/1.1 200 OK  (org B invoice saved)
```

RECOMMENDATION

Look up each invoice by its ID and the caller's organization together, so a record from another organization is never found, and answer 404 when it is not. Add an automated test that asks for another organization's invoice and expects a refusal.

ILLUSTRATIVE SAMPLE: THE REST OF FINDING F-01

CVSS 3.1 vector

EXAMPLE ENTRY

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N

Description

EXAMPLE ENTRY

The invoice API checks that the caller is signed in, but not that the invoice belongs to the caller's organization. A signed-in user who changes the ID in the address reaches the invoices of other organizations.

Impact

EXAMPLE ENTRY

Any customer of the app could read every other customer's invoices, including amounts and billing contacts, and change them before they are paid.

Owner

EXAMPLE ENTRY

Back-end team

Status in the remediation matrix

EXAMPLE ENTRY

Fixed

Retest result

EXAMPLE ENTRY

Passed: invoice 2002 now returns 404 to organization A

05 / FINDING F-02

Finding F-02: an outdated component (medium)

Illustrative sample finding. An outdated library is easy to miss in a large bundle and easy to fix once found; the finding page names the version, the advisory and the upgrade.

SAMPLE FINDING F-02 · EXAMPLE INVOICE APP

Outdated front-end library with a published cross-site scripting flaw

SEVERITY	CVSS 3.1
MEDIUM	Base score 6.1 · AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
AFFECTED COMPONENT	**TESTED AS**
/static/js/vendor-ui.js, invented library version 2.4.0	Signed out, then a signed-in billing user

PROOF OF CONCEPT

1. Read the library name and version from the header of the bundled file.
2. Compare that version with the public advisory, which names 2.6.0 as the first fixed release.
3. Open the reports page with a harmless test value in the title parameter: the browser runs it as script instead of showing it as text.

```
GET /reports?title=<test> HTTP/1.1
Host: invoices.example.com

HTTP/1.1 200 OK  (test ran as script)
```

RECOMMENDATION

Upgrade the library to 2.6.0 or later, and add a dependency check to the build that fails when a bundled version has a published advisory.

ILLUSTRATIVE SAMPLE: THE REST OF FINDING F-02

CVSS 3.1 vector

EXAMPLE ENTRY

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

Description

EXAMPLE ENTRY

The app bundles an invented front-end library at version 2.4.0. Its public advisory lists every release before 2.6.0 as open to cross-site scripting, and one page passes text from the address bar into the affected function.

Impact

EXAMPLE ENTRY

A crafted link opened by a signed-in user could run script in that user's session and act as them in the app. The user has to open the link, which keeps the score at medium.

Owner

EXAMPLE ENTRY

Front-end team

Status in the remediation matrix

EXAMPLE ENTRY

Fixed

Retest result

EXAMPLE ENTRY

Passed: the fixed release is bundled and the test value shows as text

06 / FINDING F-03

Finding F-03: a missing security header (low)

Illustrative sample finding. A low-severity finding still gets a full page, so the team that schedules it knows what it costs to leave open.

SAMPLE FINDING F-03 · EXAMPLE INVOICE APP

Strict-Transport-Security header is missing

SEVERITY	CVSS 3.1
LOW	Base score 3.1 · AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N
AFFECTED COMPONENT	TESTED AS
https://invoices.example.com, every response	Signed out

PROOF OF CONCEPT

1. Request the home page over HTTPS and read the response headers.
2. Confirm that no Strict-Transport-Security header is present.
3. In a fresh browser, type the address with http://: the first request leaves unencrypted before the redirect.

```
GET / HTTP/1.1
Host: invoices.example.com

HTTP/1.1 200 OK
Content-Type: text/html
(no Strict-Transport-Security header)
```

RECOMMENDATION

Send Strict-Transport-Security with a max-age of at least one year on every HTTPS response, and add includeSubDomains once every subdomain serves HTTPS.

ILLUSTRATIVE SAMPLE: THE REST OF FINDING F-03

CVSS 3.1 vector

EXAMPLE ENTRY

CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N

Description

EXAMPLE ENTRY

The app redirects plain HTTP to HTTPS but never tells browsers to use HTTPS only. A first visit, or a link written with http://, still starts unencrypted.

Impact

EXAMPLE ENTRY

Someone on the same network as a user, such as public Wi-Fi, could intercept or alter that first request. The attack needs that position and a user who starts on plain HTTP, so the score is low.

Owner

EXAMPLE ENTRY

Platform team

Status in the remediation matrix

EXAMPLE ENTRY

Open: planned for the next release

Retest result

EXAMPLE ENTRY

Not retested yet

07 / REMEDIATION MATRIX

Remediation matrix: status and retest result

The matrix turns the report into work your team can track. This illustrative sample shows it after a retest: two findings confirmed fixed, one still open.

The first five columns are filled in when the report is delivered. Status and retest result are the columns that keep changing: your team marks a finding fixed when the change ships, and the

retest result records whether testing confirmed it. A matrix that still shows open rows, like F-03 here, is the honest picture of what remains.

ILLUSTRATIVE SAMPLE: REMEDIATION MATRIX

F-01

FINDING

Cross-organization access to invoices

SEVERITY

HIGH

CVSS 3.1 8.1

FIX

Scope invoice lookups to the caller's organization; add a refusal test

OWNER

Back-end team

STATUS

Fixed

RETEST RESULT

Passed: invoice 2002 now returns 404 to organization A

F-02

FINDING

Outdated front-end library

SEVERITY

MEDIUM

CVSS 3.1 6.1

FIX

Upgrade the library; fail builds on advisories

OWNER

Front-end team

STATUS

Fixed

RETEST RESULT

Passed: the fixed release is bundled and the test value shows as text

F-03

FINDING

Missing HSTS header

SEVERITY

LOW

CVSS 3.1 3.1

FIX

Add the HSTS header to every HTTPS response

OWNER

Platform team

STATUS

Open: planned for the next release

RETEST RESULT

Not retested yet

How retesting is handled and what each scope includes are covered in the VAPT methodology and the VAPT cost guide.

---

08 / HOW TO READ IT

How to read a VAPT report

Three things carry most of the meaning in any report: the severity rating, the CVSS vector behind it and the remediation matrix. Here is how to read each, using the sample's invented findings.

Start with the rating, then read the vector. The rating (low, medium, high or critical) comes from the CVSS 3.1 base score, and the bands are set out on the VAPT methodology page. The vector string is the reasoning behind the number: eight short codes that anyone can check, and that a reviewer can challenge one by one if they see the flaw differently.

The table decodes F-01. Compare F-02: its vector says the user has to open a crafted link (UI:R) and that the script runs in the user's browser, beyond the vulnerable server (S:C), with limited effect on data (C:L, I:L). Those three codes are why it rates medium and not high. F-03 needs an attacker on the user's own network and a first visit over plain HTTP (AC:H, UI:R), so it rates low.

READING THE CVSS 3.1 VECTOR OF FINDING F-01, METRIC BY METRIC

AV: attack vector

VALUE IN F-01

N, network

WHAT IT SAYS

The flaw can be reached over the internet

AC: attack complexity

VALUE IN F-01

L, low

WHAT IT SAYS

No special conditions have to line up

PR: privileges required

VALUE IN F-01

L, low

WHAT IT SAYS

Any ordinary signed-in account is enough

UI: user interaction

VALUE IN F-01

N, none

WHAT IT SAYS

No other user has to click or open anything

S: scope

VALUE IN F-01

U, unchanged

WHAT IT SAYS

The damage stays inside the invoice app

C: confidentiality

VALUE IN F-01

H, high

WHAT IT SAYS

Other organizations' invoices can be read

I: integrity

VALUE IN F-01

H, high

WHAT IT SAYS

Those invoices can be changed

A: availability

VALUE IN F-01

N, none

WHAT IT SAYS

The service keeps running normally

Three habits for reading any report

- **Read the impact before the score.** FIRST's guidance says CVSS measures severity, not risk; the impact paragraph is where the report says what a finding means for your business.
- **Check that each finding can be reproduced.** A proof of concept with numbered steps and the request and response is what lets your developers confirm the problem before they change any code.
- **Use the matrix as the to-do list.** Work the open rows from the highest severity down, and treat a finding as closed only when its retest result says so.

09 / OTHER TESTS

How reports differ for other tests

Every BERRY9 test beyond the starter scope ends with the same three parts. Each kind of test then adds the material its readers need, taken from the deliverables of that service.

- **Network tests** add a topology diagram of the path the testers took, findings host by host, and MITRE ATT&CK mapping.
- **Active Directory tests** add a hygiene checklist for tier zero, meaning the accounts and settings with control over the domain.

- **Mobile app tests** cross-reference static and dynamic findings with OWASP MASVS and add a hardening guide for the build pipeline.
- **API tests** add an endpoint coverage map with the authorization tests run against it, and reproductions you can replay.
- **Source code reviews** point every finding to its file, line and commit, and add a list of patterns to search for in pull requests.
- **Web application tests** add coverage by category against the OWASP Top 10 and ASVS; this sample leaves that table out to stay short.

The specialist services use the same executive report, finding pages and remediation matrix; what each adds is set out on its own page:

Red team · Social engineering · Cloud · Wireless · Thick-client · IoT and firmware · DevSecOps

All services appear, grouped, on the offensive security services page. To discuss a report for your own systems, contact BERRY9.

RELATED PAGES

VAPT methodology: five phases from scope to report →

VAPT cost in India: a 2026 price guide →

Web application penetration testing →

Offensive security services and VAPT in India →

Contact BERRY9 →