

CMWAP COURSE

CMWAP Pentest Plan — Worked Example: Turnstile

> A full plan for engagement TURNSTILE-WEBAPP-01 inside an 8-hour working day — 270 testable minutes, 10 endpoints IN, everything else OUT or DEFERRED with a written reason.
> Companion to `[ANIM] 02-turnstile-wstg-plan - 017.html`, which shows this plan being built, reviewed and submitted.
> Source for `[EXAMPLE] Pentest Plan - Turnstile - 010.pdf`. The XSS Rat · CMWAP · Authorized testing only.

Pentest Plan — Turnstile Ticketing Systems Ltd

A black-box web-application test of the Turnstile customer portal, planned against the client's scope of work, keyed to OWASP WSTG v4.2, and structured to the PTES phases the SoW asks for.

Read this beside the template. Same sections, same clock. The difference from the RatCTF example is that this plan is written against a **client SoW you did not author** — so most of the work in sections 2 and 3 is reconciling what the client asked for with what fits in a working day, and writing down the difference rather than hiding it.

PTES mapping. The SoW asks for a PTES-aligned plan. The sections below carry the PTES phases: *pre-engagement* → the header table and §5; *intelligence gathering* → the recon block in §2 and the inventory in §3; *threat modelling* → §1 and §6 (thesis); *vulnerability analysis and exploitation* → the MUST/SHOULD blocks in §2 and §3; *post-exploitation and reporting* → the ring-fenced report block in §2 and §5.

Field	Value
Candidate / consultant	A. Candidate
Client	Turnstile Ticketing Systems Ltd
Engagement ID	TURNSTILE-WEBAPP-01
Target / engagement	The Turnstile customer portal — single host/URL issued at engagement start. Lab reference: `https://ratctf.com/challenges/turnstile-lab`
Engagement type	Authorized black-box web-application penetration test
Window opens (T+0:00)	Mon 09:00 CET
Plan due (T+0:45)	**Mon 09:45 CET** — submitted 09:41 to `security-approvals@turnstile-ticketing.example`
Testing may begin	On written sign-off only. No target contact before it.
Report + debrief due	Mon 17:00 CET
Authorization	SoW TURNSTILE-WEBAPP-01, §7. Authorization covers the single in-scope asset only and does not extend to any other system.
Plan version	v1.0 (supersedes v0.9 — see change log)

1 · Objective *(0:00 – 0:10)*

Determine whether a registered Turnstile customer can obtain tickets, ticket value, or venue access they have not paid for — by reaching another customer's tickets or orders, by transferring or re-using a ticket outside the intended flow, or by reaching the venue-access administration function — and state what that costs Turnstile in revenue, in customer data, and in gate integrity ahead of the launch.

One question, answerable in 270 minutes. Not "assess the security of the Turnstile portal", which is a multi-week engagement and would have produced a plan I could not deliver.

Why this question. The SoW names access controls, business logic and venue administration in scope and asks for real business impact. For a ticketing company the loss that matters is a ticket that works without being paid for, and a gate that opens for it.

2 · Time budget *(0:10 – 0:20)*

The SoW leaves the window open 24/7. That is an availability statement, not a workload. **Assumption A1:** this engagement is delivered as one professional 8-hour working day; the plan is budgeted and defended on that basis.

Fixed cost — comes off the top	Hours
Total window	8.00
This plan — write and send	−0.75
Await sign-off + setup, access, evidence admin	−0.50
Report — ring-fenced, never borrowed from	−1.75
Debrief video: record, review, upload	−0.50
ACTUAL TESTABLE TIME	**4.50 h = 270 min**

Wall clock	Block	Budget	What must be true when it ends
09:00–09:45	Plan — write and **send**	45 min	Submitted 09:41. Timer stops. No target contact.
09:45–10:15	Await sign-off, then set up	30 min	Written sign-off received; proxy, evidence tree and two customer accounts ready
10:15–11:15	Recon and mapping	60 min	Endpoint table confirmed against the real app; roles enumerated; §3 revised in the change log if wrong
11:15–13:15	IN · MUST — rows 1–5	120 min	The entitlement and gate questions answered either way
13:15–14:15	IN · SHOULD — rows 6–10	60 min	Supporting rows tested or explicitly parked with a reason
14:15–14:45	Verify and capture proof	30 min	Every finding re-run from its own written steps on a clean session
14:45–16:30	Report	105 min	Ring-fenced. Not testing time.
16:30–17:00	Debrief video (≤10 min run-time)	30 min	Recorded, reviewed and uploaded
Testing subtotal	recon + MUST + SHOULD + verify	**270 min**	= 4.50 h — equals the testable time above

3 · Endpoint inventory and disposition *(0:20 – 0:35)*

IN — authorized, relevant, affordable. **OUT** — not authorized by the SoW. Never tested. **DEFERRED** — authorized and relevant, does not fit 270 minutes.

This table is written black-box from the SoW before any access. Section 2 budgets 60 minutes of recon specifically to confirm it; where reality differs, the row changes and the change log records it.

#	Endpoint / function	Method	Role(s)	WSTG ID(s)	Disp.	Why this disposition	Budget
1	Ticket detail by identifier — `/tickets/{id}`, API equivalent	GET	customer ×2	WSTG-ATHZ-02, WSTG-ATHZ-04	**IN**	A ticket is the bearer instrument. Reading another customer's ticket is the shortest path to the loss named in the objective.	35 min
2	Order history and order detail — `/orders/{id}`	GET	customer ×2	WSTG-ATHZ-02	**IN**	Orders carry personal and payment-adjacent data. Same object-reference question, different data class, so it is cheap once row 1 is set up.	20 min
3	Ticket transfer / re-assignment	POST	customer ×2	WSTG-BUSL-01, WSTG-ATHZ-02	**IN**	Moves value between accounts. Tests whether transfer checks ownership, and whether a transferred ticket stays valid for the sender.	25 min
4	Checkout — price, quantity and discount handling	POST	customer	WSTG-BUSL-01, WSTG-BUSL-02	**IN**	The direct revenue path. Client-side price or quantity trust, negative quantities, coupon re-use.	25 min

#	Endpoint / function	Method	Role(s)	WSTG ID(s)	Disp.	Why this disposition	Budget
5	Venue-access administration function	GET/ POST	customer	WSTG-ATHZ-01, WSTG-ATHZ-02	**IN**	Named in the SoW as in scope. Vertical escalation into gate administration is the highest-impact outcome available in this application.	15 min
6	Session handling — cookie attributes, fixation, logout	—	customer	WSTG-SESS-01/02/03/06	**IN**	Every finding above depends on session integrity; cheap to test alongside rows 1–3.	15 min
7	Authentication — login, registration, password reset	POST	anon, customer	WSTG-ATHN-01/02/03	**IN**	Reset-token and user-enumeration weaknesses only. **No brute force** — see §5.	15 min
8	File handling — ticket/ badge download and any upload	GET/ POST	customer ×2	WSTG-ATHZ-01, WSTG-BUSL-08/09	**IN**	Named in the SoW as in scope. Ticket artefacts are downloadable objects; upload paths accept and later serve customer-supplied content.	15 min
9	Input surfaces — event search, name-	GET/ POST	customer	WSTG-INPV-01, WSTG-INPV-02	**IN**	Reflected and stored input reaching other users,	10 min

#	Endpoint / function	Method	Role(s)	WSTG ID(s)	Disp.	Why this disposition	Budget
	on-ticket, notes fields					including anything rendered on a printed or scanned ticket.	
10	Gate/scan validation endpoint — ticket presented at a venue	POST	customer	WSTG-BUSL-01	**IN**	Whether a single ticket can be validated twice, or after transfer. Small budget: confirm the behaviour, do not build a full harness.	5 min
11	Denial-of-service / resource-exhaustion testing	—	—	—	**OUT**	SoW §3 excludes it. Not tested, not measured, not "just once".	—
12	Password brute force / credential stuffing on live accounts	POST	—	WSTG-ATHN-03	**OUT**	SoW §3 excludes brute force that locks out or degrades live accounts. Lockout *policy* is observed passively only.	—
13	Network, infrastructure and hosting platform	—	—	—	**OUT**	SoW §3. No port scanning, no subdomain enumeration, no host discovery. The authorization covers one application on one host.	—

#	Endpoint / function	Method	Role(s)	WSTG ID(s)	Disp.	Why this disposition	Budget
14	Any host or asset other than the single in-scope target	—	—	—	**OUT**	SoW §3 and §7. Authorization does not extend to it.	—
15	Social engineering, phishing, physical access	—	—	—	**OUT**	SoW §3.	—
16	Cryptography and TLS configuration review	—	—	WSTG-CRYP-01/03	**DEFERRED**	Authorized and relevant, but low expected yield against the objective. Pull forward if a ticket or session token turns out to be self-signed data.	—
17	Client-side sweep — postMessage, DOM sinks, third-party JS	—	customer	WSTG-CLNT-01/11	**DEFERRED**	Does not fit 270 minutes. Pull forward if row 9 shows input reaching a client-side sink.	—
18	Full API fuzzing and schema abuse at scale	—	customer	WSTG-APIT-01	**DEFERRED**	Rows 1–3 cover the API surface that matters for the objective; a broad fuzz is a second day's work.	—

Budget check: MUST rows 1–5 = 120 min. SHOULD rows 6–10 = 60 min. Recon 60 + verify 30. Total 270 min, equal to the testable time in §2.

4 · Deferred register *(included above)*

Rows 16, 17 and 18 are deferred, not dropped. Each carries the condition that would pull it forward, and each will appear in the report's coverage statement so the client can see exactly what was and was not examined. Rows 11–15 are **out of scope** and are a different category entirely: they are not tested under any condition, and if one becomes relevant it is an escalation to the client, not a decision I take alone.

5 · Rules of engagement *(0:35 – 0:40)*

Condensed from SoW §4. The SoW is authoritative; this section does not override it.

Rule	This engagement
Authorization	SoW TURNSTILE-WEBAPP-01 §7. Single in-scope asset. **No testing before written sign-off of this plan.**
Data handling	No destruction, corruption or exfiltration beyond the minimum needed to demonstrate impact. One record is a demonstration; a hundred is an exfiltration.
Evidence	Exact request and response captured for every finding, with the role and session labelled. Stored encrypted, destroyed 30 days after report acceptance.
Critical SLA	Remote code execution, full account takeover or mass data exposure — stop, notify the client contact immediately, and flag it prominently in the report. Do not explore further.
Lockout guard	Authentication tests use my own accounts only, with at most two deliberate failures per account. If any account locks, stop that test, log it, and notify.
Rate	10 req/s ceiling, enforced in the tool. Nothing that could read as resource exhaustion.
Abort criteria	Service degradation · real customer data reached beyond the minimum · out-of-scope asset reached · any sign of pre-existing compromise. Stop, notify, write it down.
Attribution	Fixed source IP, user-agent `cmwap-pentest/1.0`, start and stop times logged, so my traffic is findable in their logs.
Comms	Written note at the end of the day whatever happened; immediate call for anything in the critical SLA.
AI use	An AI assistant reviewed this plan for feasibility and scope gaps. It did **not** write the plan and will not write the report. Every suggestion was accepted or rejected by me, in writing — see §6.

6 · Thesis, risks and change log *(0:40 – 0:45)*

Thesis

The finding is in authorization on ticket and order objects, and in the transfer flow. Turnstile is a ticketing system: the object identifiers are the money, three parties touch each ticket (buyer, recipient, gate), and the venue-access function is administered from the same portal as the customer experience. That combination is where I expect the loss.

Being wrong about this is normal and costs nothing. Not having a thesis means testing by list instead of by judgement.

Assumptions and risks

#	Assumption	If wrong
A1	The engagement is delivered as one 8-hour working day, despite the 24/7 window.	Re-budget and record it; the block structure scales, the dispositions do not.
A2	I can register two customer accounts myself.	Rows 1–3 and 8 lose their second session; I request a second account and log the delay.
A3	Only the customer role is guaranteed. Venue-staff and administrator roles may not be issued.	Row 5 becomes a black-box escalation attempt from the customer role only.
A4	The environment is not live production and contains seeded data.	Every data-touching test tightens to read-only demonstration; §5 abort criteria apply sooner.
A5	No live payment gateway. Checkout completes against a stub.	Row 4 drops real transaction tests and keeps client-side price and quantity handling only.

Change log

Time	Change	Cost	Decision	Approved by
09:32	v0.9 → v1.0 after AI feasibility and scope-gap review: added rows 5 and 8; fixed the budget arithmetic (8.25 h → 8.00 h); added the lockout guard to §5; recorded A3.	0 min	Accepted — all four were real gaps against the SoW.	self
09:34	AI suggested a checkout rate-limit stress test.	—	**Rejected** — resource exhaustion, SoW §3 OUT.	self
09:36	AI suggested subdomain enumeration and a port scan.	—	**Rejected** — network/ infrastructure, SoW §3 OUT; authorization covers one asset.	self
09:41	Plan v1.0 submitted to `security-approvals@turnstile-ticketing.example`.	—	Awaiting sign-off. No target contact.	—
10:12	Written sign-off received. Testing authorized to begin.	—	Window opens.	client

(Rows below this line are added during the engagement, never by editing the rows above.)

[EXAMPLE] · The XSS Rat · CMWAP Course · Authorized testing only. Turnstile Ticketing Systems Ltd is a fictional client used for training; the target is the RatCTF Turnstile lab.