Pentest Scoping Checklist

Bring this to your scoping call. Count your attack surface, document your multipliers, and arrive with a number you can defend.

READ THE BLOG FIRST →

1 REST API Endpoints

Count your REST API endpoints

- [ ] Use the most accurate source available:
  - OpenAPI / Swagger specification
  - Route dump (rails routes, artisan route:list, manage.py show_urls)
  - Proxy capture of application traffic
  - Fallback: estimate features × 5

Path + HTTP method = 1 endpoint

- [ ] GET /users and POST /users are two endpoints, not one.

2 GraphQL Operations

Count your GraphQL operations

- [ ] Introspection query or schema file. Count every field under Query and every field under Mutation.

Report as: X queries + Y mutations = Z total

- [ ] One GraphQL URL ≠ one endpoint. Operation count is what drives scope.

3 Effort Multipliers — document these before your call

List every authentication method

- [ ] Password, SSO (Okta, Auth0), magic links, API keys, MFA — each has a distinct vulnerability profile requiring separate testing.

Count distinct user roles / permission levels

- [ ] Admin, standard, read-only, unauthenticated. Authorization must be verified per role on every sensitive endpoint.

List all third-party integrations

- [ ] Payment, CRM, analytics, communication tools — e.g. Stripe, Salesforce, HubSpot, Twilio.

List internal and cloud infrastructure dependencies

- [ ] Internal microservices, S3 buckets, and cloud storage — each is a data boundary that may be internal or external.

Flag unauthenticated endpoints separately

- [ ] Highest-priority attack surface. Testers focus here first.

4 Before the Scoping Call

Add REST endpoints + GraphQL operations for your base attack surface count

- [ ] This single number is the most important thing you bring to the scoping call.

Note which counting method you used and flag any uncertainties

- [ ] e.g. 'OpenAPI spec — may have dead routes' or 'Feature estimate × 5 — rough estimate, not verified against code.'

SCOPING METHOD REFERENCE

Method	Accuracy	Risk
Domains / Subdomains	Low	Misses API logic entirely
Features / User Stories	Medium	Inconsistent estimates
Pages / Screens	Low	Undercounts SPA backends
Endpoints / Operations ✓	**High**	**Most reliable — recommended**