APPS BY VIKAS

Meet Vikas



GenzLudo

Ludo with 3D characters and up to 8 players. A first-of-its-kind no-teaming-up feature keeps play fair. Try it - you'll love the difference.

Download on Google Play



Kachu

Too much scrolling straining your neck? Add your addictive apps and complete daily neck exercises to unlock them.

Download on Google Play

A REAL WEBSITE CASE STUDY

What 7,778 suspicious requests reveal

What bots looked for. What the logs actually show. What to check on your own website.

Vikas Yadav / vikasbuilds.com / November 2026

7,778

Recorded probe events

404-only logging category

1,025

Distinct requested paths

Not unique attackers

66

Days of daily totals

19 Sep - 23 Nov 2026

REQUEST

A scanner asks for a familiar file.

→

BOUNDARY

The app decides what can be served.

→

EVIDENCE

The log records one observed event.

Inside: volume chart, nine probe patterns, response-flow diagrams, a deployment worksheet and a copy-paste self-audit prompt.

This is a measured slice of traffic, not a certificate that a website is secure.

01 / READ THE EVIDENCE

What the numbers mean

Recorded events from one website. Keep each event type separate.

7,778

Suspicious-path probe events

All were logged after a 404 response.

809

Cross-origin rejection events

A separate event stream; not added to probes.

1,025

Distinct stored path strings

Paths can differ by case or encoding.

32

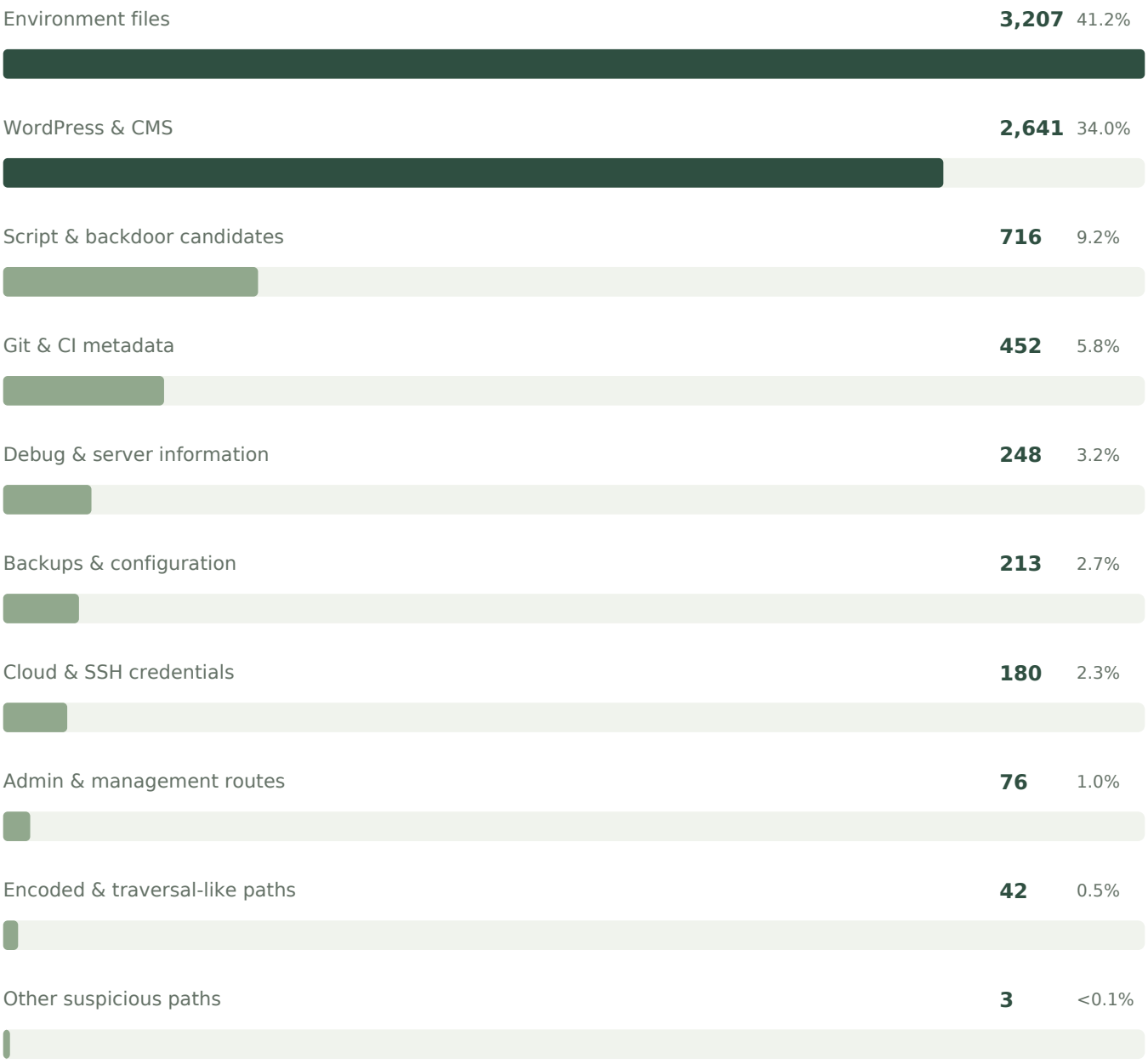
Admin requests without a session

May include expired sessions; not 32 attackers.

02 / TRAFFIC AT A GLANCE

Where the probes concentrated

One category per recorded path. Counts add up to 7,778.



75.2% of the recorded probes targeted environment files or WordPress/CMS paths.

Grouping is based on path names, not verified attacker intent. The nine pattern groups contain 7,775 events; three unmatched events remain in Other. Grouping rules are documented on page 12.

03 / PATTERNS 1 + 2

Secrets are the prize

Requests for a file do not prove the file exists or that a secret leaked.

01 Environment files

3,207 EVENTS

```
/.env /.env.production /api/.env
```

LOOKING FOR	Database passwords, service tokens and other application settings.
IN THIS DATA	These names matched the probe rule and returned 404.
YOUR CHECK	Keep secret-bearing files out of public output. Rotate a secret if it was exposed.

02 Cloud & SSH credentials

180 EVENTS

```
/.aws/credentials /.ssh/id_rsa
```

LOOKING FOR	Credentials that may unlock hosting accounts, servers or cloud services.
IN THIS DATA	The grouped requests returned 404; this is not an account-access audit.
YOUR CHECK	Use scoped credentials or platform bindings. Keep key files outside public assets.

VISUAL CHECK / TWO SEPARATE AREAS

PUBLIC OUTPUT

HTML, CSS, images and approved files

PRIVATE RUNTIME

Scoped secrets and application bindings

04 / PATTERNS 3 + 4

A filename is a guess

A scanner can try paths for software your website does not use.

03 WordPress & CMS

2,641 EVENTS

```
/wp-admin/install.php /xmlrpc.php
```

LOOKING FOR	CMS login, installation or plugin endpoints worth inspecting.
IN THIS DATA	The website is a custom app; these recorded requests returned 404.
YOUR CHECK	Inventory the software you actually run. Patch it and remove unused components.

04 Script & backdoor candidates

716 EVENTS

```
/test.php /i.php /images/images/cache.php
```

LOOKING FOR	Potential scripts, test pages or previously planted backdoors. Intent is inferred.
IN THIS DATA	The app recorded 404 responses for these candidate paths.
YOUR CHECK	Restrict executable code to your release. Uploaded content must not become code.

GUESS

A familiar CMS or script filename

→

ACTUAL STACK

Your own routes and deployed files

→

RESULT

Inspect the response; do not assume success.

Not running WordPress removes WordPress-specific exposure; it does not remove bugs in your own application.

05 / PATTERNS 5 + 6

Publish the product, not the repo

Old archives and source-control files can reveal more than a page should.

05 Backups & configuration

213 EVENTS

```
/backup.zip /dump.sql /config.json
```

LOOKING FOR Archives, database exports or configuration containing internal information.

IN THIS DATA The grouped requests returned 404. Public config is not automatically a secret.

YOUR CHECK Keep backups private. Inspect the final published files for unintended exports.

06 Git & CI metadata

452 EVENTS

```
/.git/config /.git/HEAD /.git-credentials
```

LOOKING FOR Repository details, source history or credentials that might be exposed.

IN THIS DATA These recorded paths returned 404. A public CI file alone is not proof of a leak.

YOUR CHECK Exclude .git and credential files from deploy output. Review history for leaked secrets.

RELEASE PIPELINE / KEEP THE BOUNDARY VISIBLE

SOURCE
Code, history and private configuration

→

BUILD
Produce only the required app output.

→

DEPLOY
Inspect the files that will become public.

06 / PATTERNS 7 + 8

Debug data and unusual paths

Different request shapes need different checks.

07 Debug & server information

248 EVENTS

```
/phpinfo.php /actuator/env /server-status
```

LOOKING FOR	Runtime settings, diagnostics, versions or memory-related information.
IN THIS DATA	The grouped requests returned 404; other diagnostic routes were not measured.
YOUR CHECK	Disable unnecessary diagnostics or protect them with access controls.

08 Encoded & traversal-like paths

42 EVENTS

```
/%2f.env /assets../.env
```

LOOKING FOR	Alternate path forms that may be interpreted differently by different layers.
IN THIS DATA	The recorded variants returned 404. Not every odd path is a working exploit.
YOUR CHECK	Check decoding, normalization and file boundaries in your actual server stack.

RAW PATH
Encoded or unusual characters

→

NORMALIZE
Check how each server layer parses it.

→

AUTHORIZE
Serve only approved routes and files.

Routing helps define a boundary. It is not proof that path-handling bugs are impossible. [7]

07 / PATTERN 9 + SEPARATE SIGNALS

Admin access is a different check

Do not confuse route discovery, sign-in attempts and session checks.

09

Admin & management routes

76 EVENTS

```
/admin /admin/login /admin.php
```

LOOKING FOR	Management pages, login screens or vulnerable management tools.
IN THIS DATA	These guessed paths returned 404. Separately, 32 requests lacked an admin session.
YOUR CHECK	Verify authentication and permissions on every admin route, not just the login page.

CROSS-ORIGIN EVENTS

809

The site recorded its origin-rejection condition. That alone does not identify a crawler or prove a malicious CSRF attempt.

ADMIN SESSION EVENTS

32

The current source redirects an unauthenticated admin request to sign-in. An expired owner session can also create this event.

Three separate protections

Authentication checks who signed in. Authorization checks what that person may do. CSRF protection helps stop another origin from triggering actions through a browser session.

Payment callbacks need their own verification. A zero-valued security metric cannot establish that no forged callback was accepted. Review the actual payment flow. [5, 6]

08 / TURN EVIDENCE INTO ACTION

Follow one request through the system

Each boundary should have a clear job and a way to verify it.

EDGE + ROUTE

WAF rules and request matching

→

APP CONTROLS

Authentication, input and permission checks

→

PRIVATE DATA

Scoped access to secrets and storage

Use this review sheet before a release. A tick means you have evidence, not just a setting that looks correct.

☐ **Secrets**	Public files contain no secret keys; exposed credentials are replaced.
☐ **Release output**	No .git directory, private backups, database dumps or key files.
☐ **Runtime**	Production mode; only necessary diagnostic endpoints are available.
☐ **Admin**	MFA where appropriate; rate limits and server-side permission checks.
☐ **Forms + APIs**	State-changing actions have appropriate CSRF and authorization checks.
☐ **Webhooks**	Verify provider authenticity and reconcile the intended payment state.
☐ **Uploads**	Validate files, restrict access and prevent execution as application code.
☐ **Dependencies**	Review relevant advisories and apply required updates.
☐ **Logging**	Record useful events; protect logs and understand collection limits.
☐ **Recovery**	Test backups, restore steps and how to rotate exposed secrets.

Evidence can be a deploy-file check, a code reference, a test result or a redacted log example. Avoid including live secrets in a report. [2, 3]

09 / COPY-PASTE SELF-AUDIT / PART 1

Inspect a small, authorized sample

Use this only for a website and repository you own or are authorized to assess.

```
You are reviewing my website for accidental public exposure.
I own this website and authorize this limited read-only review.
Website: <https://your-domain.example>
Repository: <local project folder>

SCOPE
- Review the code first. Choose only relevant paths below.
- No sign-in, password guessing, writes or exploit payloads.
- No form submission or payment/webhook requests.
- Do not execute downloaded code or inspect third-party hosts.
- Do not follow redirects off the authorized origin.

LIMITED HTTP CHECK
- GET only; one request at a time; at least 1 second apart.
- At most 20 requests total, including baseline requests.
- Use a 5-second timeout and inspect at most 16 KB per response.
- Stop on rate limiting, instability or possible secret exposure.
- Never print, save or transmit a discovered secret value.
- Record status, content type and a short redacted observation.

First request one normal page and one made-up missing path
as baselines. Then choose relevant examples:
/.env                 /.env.production
/api/.env             /.git/HEAD
/.git/config          /.git-credentials
/backup.zip           /dump.sql
/config.json          /.aws/credentials
/.ssh/id_rsa          /wp-login.php
/wp-admin/install.php /actuator/env
/phpinfo.php          /server-status
/admin                /admin/login

Do not judge by status alone. Compare the response with
baselines and inspect the content without exposing secrets.
A normal HTML fallback is not proof the whole app is safe.
If a result is unclear, mark it Needs review.

Continue with the code review and report on the next page.
```

The request budget includes the two baselines. This small sample checks exposure, not exploitability or overall security.

10 / COPY-PASTE SELF-AUDIT / PART 2

Review the code. Report the evidence.

Use narrow conclusions that match the checks actually completed.

```
CODE REVIEW
- Check secret handling and deploy exclusions. Review history
  for known secret exposure without reproducing secret values.
- Inspect final build output, static assets and file-serving rules.
- Check debug mode, error detail and diagnostic access controls.
- Inspect admin authentication, authorization and rate limits.
- Check CSRF handling for browser state-changing requests.
- Check webhook authenticity, replay/idempotency handling
  and server-side payment-state reconciliation.
- Review upload validation, access control and execution rules.
- Review relevant dependencies, security headers and logging.
- Identify missing evidence and tests; do not invent outcomes.

REPORT
Use these results:
1. Needs review / possible exposure
2. No exposure observed in this check
3. Not applicable, with a stack-specific reason
4. Not checked / insufficient evidence

Table: Check | Result | Evidence | Limitation | Next action
Prioritize issues by impact and confidence. Separate code
observations from live response observations. Recommend fixes;
do not apply or publish changes without authorization.
```

HOW TO INTERPRET A RESPONSE

404	No matching resource observed at that path. Coverage remains limited.
200 + site HTML	May be a fallback page. Compare with the missing-path baseline.
401 / 403	A control refused this request. It does not validate the entire system.
Content or errors	Review what was disclosed. Stop if it appears to contain a secret.

Keep the audit result and evidence with your release notes. Retest relevant checks after changes.

11 / METHODS + SOURCES

What this case study is based on

A reproducible snapshot with deliberate limits.

Dataset	66 IST dates: 19 September through 23 November 2026. Extracted on 24 November 2026 from the site's daily aggregates.
Reconciliation	The daily probe total and summed path counts both equal 7,778. There are 1,025 distinct stored path strings. This replaces the draft's unreconciled category breakdown.
Grouping	One category per path, using case-insensitive matching and one URL decode. Priority: encoded/traversal-like, cloud/SSH, env, CMS, Git/CI, debug, backup/config, admin, script, other.
Collection limits	The current code has a 100-row per-IP security limit and a daily security budget. Owner devices are excluded; writes can fail. Raw events are retained about 30 days; daily totals about 400.
Interpretation	Intent is inferred from filenames. Probe events are 404-selected. Source-country and network labels are not attacker identities. No compromise count is established by this snapshot.

SOURCES / CLICK TO OPEN

[1] First-party evidence	Daily security + probe-path aggregates; current logging and access-control source. Private aggregate snapshot reviewed for this PDF.
[2] OWASP / Logging	Event classification, confidence, verification and protection of log data.
[3] OWASP / Secrets Management	Managing secret material and its lifecycle.
[4] OWASP / Web Security Testing	Review old backups and unreferenced files for sensitive information.
[5] OWASP / CSRF Prevention	Controls for browser-originated state-changing requests.
[6] Review your provider docs	Webhook verification details depend on the payment provider and integration.
[7] OWASP / Path Traversal	Path encodings, file access and directory-boundary checks.