

CARVER Live

A single-file, web-based risk and vulnerability assessment app that runs on GitHub Pages or locally in any modern browser. Built for security professionals who want fast, repeatable CARVER assessments, simple Likelihood and Impact mapping, live monitoring, and one-click reporting.

Table of contents

1. What this app does
2. Quick start
3. Deploy on GitHub Pages
4. Running locally
5. Data import from Excel
6. Live monitoring feed
7. Using the app step by step
8. Scoring and math
9. CARVER primer
10. Supplementing CARVER with the 5x5 matrix
11. Asset prioritization and reporting
12. Security and privacy notes
13. Troubleshooting
14. Roadmap

1) What this app does

- Interactive CARVER scoring with selectable scales: 1 to 5 or 1 to 10.
- Pa calculation and three prioritization modes: Pa, total CARVER, or L×I.
- Simple and explainable Likelihood and Impact mapping to a 5×5 matrix.
- Asset register for all DHS critical infrastructure sectors, with owners, status, DBT tags, notes, and watchers.
- Excel ingest using SheetJS to pull your existing worksheets straight into the app.
- One-click exports: JSON for pipelines, Markdown for briefs, compact PDF for quick distribution.
- Lightweight live monitoring that polls a JSON feed hosted in the same GitHub repo or anywhere reachable.
- A ready GitHub Actions workflow to refresh your data feed on a schedule.

2) Quick start

1. Save the provided `index.html` in a folder on your machine.
2. Double-click it to open in your default browser. No install required.
3. Click **Add Asset** or **Import Excel** to load your data.
4. Adjust CARVER sliders. The app calculates Pa and maps L and I automatically.
5. View the 5×5 matrix, sort the asset list by Pa, Total, or L×I.

6. Export results as JSON, Markdown, or PDF.

3) Deploy on GitHub Pages

1. Create a new GitHub repository. Name it as you wish.
2. Add `index.html` to the root of the repo. Commit and push.
3. Go to **Settings** → **Pages** and choose **Deploy from a branch** with the **main** branch and root directory.
4. Wait for Pages to publish, then open the site URL that GitHub provides.

Tip: Put your `datafeed.json` under `public/` or the repo root. The app accepts a full URL in Settings.

4) Running locally

- Double-click `index.html` or open it with any Chromium, Firefox, or Safari browser.
- The app stores your edits in the browser using `localStorage` so you can close and resume work later.

5) Data import from Excel

Click **Import Excel** and select a `.xlsx` file. Common headers are auto-mapped. Recognized names include:

- Asset name: `Name`, `Asset`, `Asset Name`
- Sector: `Sector`
- Type: `Type`
- Location: `Location`
- Status: `Status` (Open, Mitigating, Closed)
- Owner: `Owner`, `Assignee`
- Last assessed: `LastAssessed`, `Last`
- Notes: `Notes`
- CARVER factors: `C` or `Criticality`, `A` or `Accessibility`, `R` or `Recuperability` or `Recoverability`, `V` or `Vulnerability`, `E` or `Effect`, `Rz` or `Recognizability`
- Threat tags: `DBT`, `Threats`
- Watchers: `Watchers` (comma separated)

If your columns differ, import once, then adjust fields in the Asset Editor and export JSON for future reuse.

6) Live monitoring feed

The app can poll a JSON file at a fixed interval. Add a public JSON endpoint and paste its URL under **Monitoring** → **Data feed URL**.

Example `datafeed.json`:

```
{
  "updated": "2025-08-12T13:00:00Z",
  "signals": [
    { "type": "intel", "severity": 3, "summary":
      "New threat bulletin for energy substations in Region West" },
    { "type": "incident", "severity": 2, "summary":
      "Unauthorized access attempt detected at Water Treatment Plant 1" }
  ]
}
```

GitHub Actions template

Download the included YAML from the app or create `.github/workflows/update-datafeed.yml`:

```
name: Update CARVER Live feed
on:
  schedule:
    - cron: '*/*30 * * * *' # every 30 minutes
  workflow_dispatch: {}
permissions:
  contents: write
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Write datafeed.json
        run: |
          mkdir -p public
          echo '{"updated":"'$(date -u +%Y-%m-%dT%H:%M:%SZ)'" ,"notes":"replace
with your integration"}' > public/datafeed.json
      - name: Commit
        run: |
          git config user.name "github-actions"
          git config user.email "actions@users.noreply.github.com"
          git add public/datafeed.json
          git commit -m "chore: update datafeed" || echo "No changes"
          git push
```

Point the app to `https://<your-username>.github.io/<repo>/public/datafeed.json`.

7) Using the app step by step

1. **Create or import assets** and set Sector, Type, Location, Owner, Status, DBT tags, and Notes.
2. **Choose the CARVER scale** in Settings: 1 to 5 or 1 to 10.

3. **Score CARVER factors** using sliders for C, A, R, V, E, Rz.
4. The app computes **Pa** and maps **Likelihood** and **Impact** to 1 to 5.
5. Review the **5×5 matrix** to see concentration of risk by L and I.
6. Sort the asset list by **Pa**, **Total**, or **L×I** for prioritization.
7. Use **Export** buttons to generate JSON, Markdown, or PDF.
8. Optional: enable **Monitoring** by adding a feed URL. The snapshot shows inside the app.

8) Scoring and math

CARVER scale: selectable 1 to 5 or 1 to 10. Higher is worse for all factors under the user's standard.

- Pa (probability of attack proxy):

$$Pa = (C + A + R + V + E + Rz) / (6 \times scaleMax)$$

Result is 0 to 1. The UI shows Pa as a percent.

- Likelihood and Impact mapping (simple averages):

$$L_raw = average(A, V, Rz)$$

$$I_raw = average(C, E, R)$$

Normalize from the chosen CARVER scale to 1 to 5 for the matrix using linear mapping, then round to nearest integer.

- 5×5 matrix severity coloring: rising color intensity with higher L or I. The grid shows a count of assets per cell.
- Asset prioritization metric options:
 - **Pa**: normalized proxy for attack attractiveness.
 - **Total**: raw CARVER sum.
 - **L×I**: discrete 5×5 placement product.

9) CARVER primer

CARVER is a structured target analysis method that scores six factors.

- **C – Criticality**: how essential the asset is. Higher means a hit would cause larger mission, safety, or economic loss.
- **A – Accessibility**: ease of access for the adversary. Higher means easier to approach and engage without detection.
- **R – Recuperability**: how hard it is to restore the asset to required function. Under the user's standard, high scores mean low resilience and slow or difficult recovery. Low scores mean strong resilience and fast recovery.

- **V – Vulnerability**: how easy it is to damage or degrade the asset using adversary tools and TTPs.
- **E – Effect**: scale and scope of expected consequence if the asset is hit.
- **Rz – Recognizability**: how easy it is for the adversary to identify the target and its weak points.

Scales can be 1 to 5 or 1 to 10. Keep factor definitions stable inside your program so that re-assessments are comparable over time.

10) Supplementing CARVER with the 5×5 matrix

CARVER gives a granular, factor-level view that is excellent for asset triage and hardening plans. Many organizations also need a concise Likelihood and Impact view for executives and governance boards.

This app bridges both:

- Likelihood is computed from the access-centric set: A, V, Rz. This reflects opportunity and ease of exploitation.
- Impact is computed from consequence and recovery drag: C, E, R, where the R scoring direction follows the user's standard.
- Each average is normalized to a 1 to 5 scale and placed on the 5×5 grid. The grid communicates severity quickly and supports alignment with ISO 31000 style risk categories.

Practical benefits:

- Field teams can use CARVER sliders and keep their domain language.
- CSOs and boards get a simple heatmap that aligns to enterprise risk dashboards.
- You can export the same dataset to both CARVER-centric work plans and matrix-centric registers.

11) Asset prioritization and reporting

- Sort by **Pa** to focus on assets that are most attractive to attack.
- Sort by **Total** for a pure CARVER ranking when you want to compare only factor sums.
- Sort by **L×I** when stakeholders require matrix-aligned triage.

Reports:

- **Markdown**: paste directly into Substack or briefing notes.
- **PDF**: compact paginated summary of ranked assets.
- **JSON**: integrates with analytics pipelines and dashboards.

12) Security and privacy notes

- The app persists data in the browser via `localStorage`. Use exports for backups.
- When hosting on GitHub Pages, treat the site as public unless the repo is private.
- If your `datafeed.json` contains sensitive information, place the repo behind access controls or use a private endpoint.
- For highly sensitive programs, consider offline use only or a private Pages deployment.

13) Troubleshooting

- **Excel import fields not picked up:** check headers. See the list in section 5. You can also clean your sheet and re-import.
- **Matrix looks empty:** confirm you selected the intended CARVER scale and that A, V, Rz and C, E, R are not left at defaults.
- **Feed not updating:** verify the URL is reachable over HTTPS. Check the Actions logs for the scheduled workflow.
- **Mixed content in the browser:** if your Pages site is HTTPS but the feed is HTTP, the browser will block it. Serve the feed over HTTPS.

14) Roadmap

Planned enhancements that can be added without changing hosting model:

- Mitigation tracker per asset with target dates, cost estimates, and re-score deltas.
- CSV exports and a print-friendly risk matrix.
- Role views for assessor, supervisor, CSO with optional field locking.
- A guided DBT builder that pre-loads typical threats per sector and maps to CARVER prompts.

Credits: Designed for use by security assessors, site supervisors, and CSOs. Built for GitHub Pages or local browsers so teams can deploy in minutes.