Day 3 — Web App & Content OSINT (step-by-step)

Goal: discover hidden files, unpublished documents, versioned pages, and sensitive content exposed on web servers (and historically) without exploiting anything.

Main tools: `gobuster` or `dirsearch` (directory brute force), browser Google Dorks, Burp Suite Community (or `curl`/`wget`), Wayback Machine (archive.org), `wget`/`curl` for safe download, SecLists wordlists.

Folder: `day3_webcontent/` with subfolders `scans/`, `evidence/`, `downloads/`

---

0) Setup

- Create folders:
    - `mkdir -p day3_webcontent/scans day3_webcontent/evidence day3_webcontent/downloads`
- Install wordlists (SecLists):
    - on Linux: `sudo apt install seclists` or `git clone https://github.com/danielmiessler/SecLists.git`
- Install tools:
    - `gobuster` (Go) or `dirsearch` (Python)
    - Burp Suite Community (Java) optional
    - `wget` / `curl` available everywhere

---

1) Collect robots.txt and sitemap.xml (first look)

These often point to interesting directories/files.

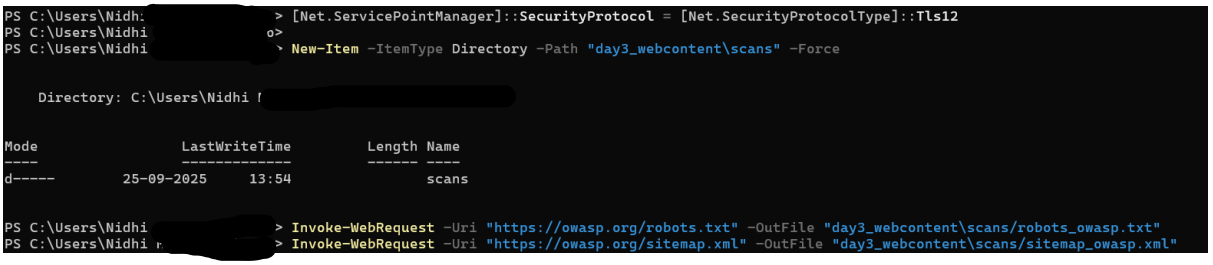
PowerShell:

```
# TLS 1.2 (if needed)
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12

# Ensure folder exists
New-Item -ItemType Directory -Path "day3_webcontent\scans" -Force

# Download robots.txt from a real site
Invoke-WebRequest -Uri "https://owasp.org/robots.txt" -OutFile
"day3_webcontent\scans/robots_owasp.txt"

# Download sitemap.xml from the same site (if available)
Invoke-WebRequest -Uri "https://owasp.org/sitemap.xml" -OutFile
"day3_webcontent\scans/sitemap_owasp.xml"
```

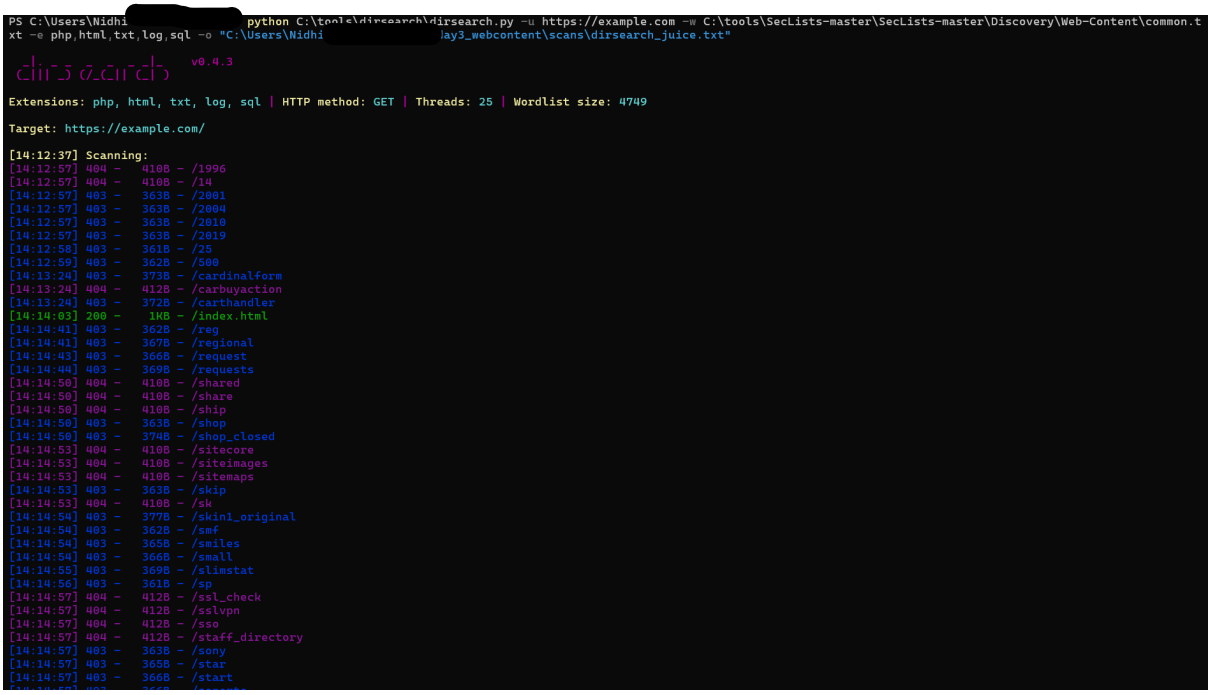


2) Directory brute-force (safe, non-destructive)

Important: Use only on domains you control. Keep rate low to avoid DoS.

Install: `pip install git+https://github.com/maurosoria/dirsearch.git`

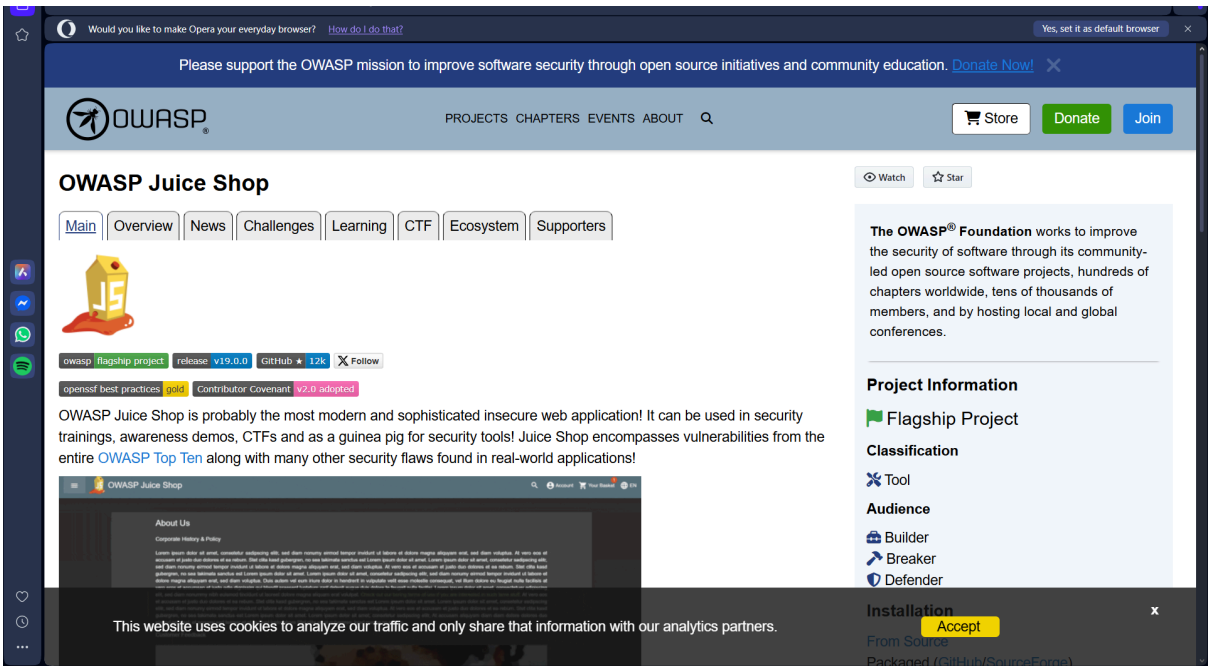
```
python C:\tools\dirsearch\dirsearch.py -u https://example.com -w
C:\tools\SecLists-master\SecLists-master\Discovery\Web-Content\commo
n.txt -e php,html,txt,log,sql -o "C:\Users\Nidhi
\day3_webcontent\scans\
.txt"
```



3) Crawl site to download visible content (non-recursive first)

Use `wget` carefully — recursive crawl can download a lot.

```
PS C:\Users\Nidhi [redacted] Invoke-WebRequest -Uri "https://owasp.org/www-project-juice-shop/" -OutFile "day3_webcontent/downloads/index.html"
```

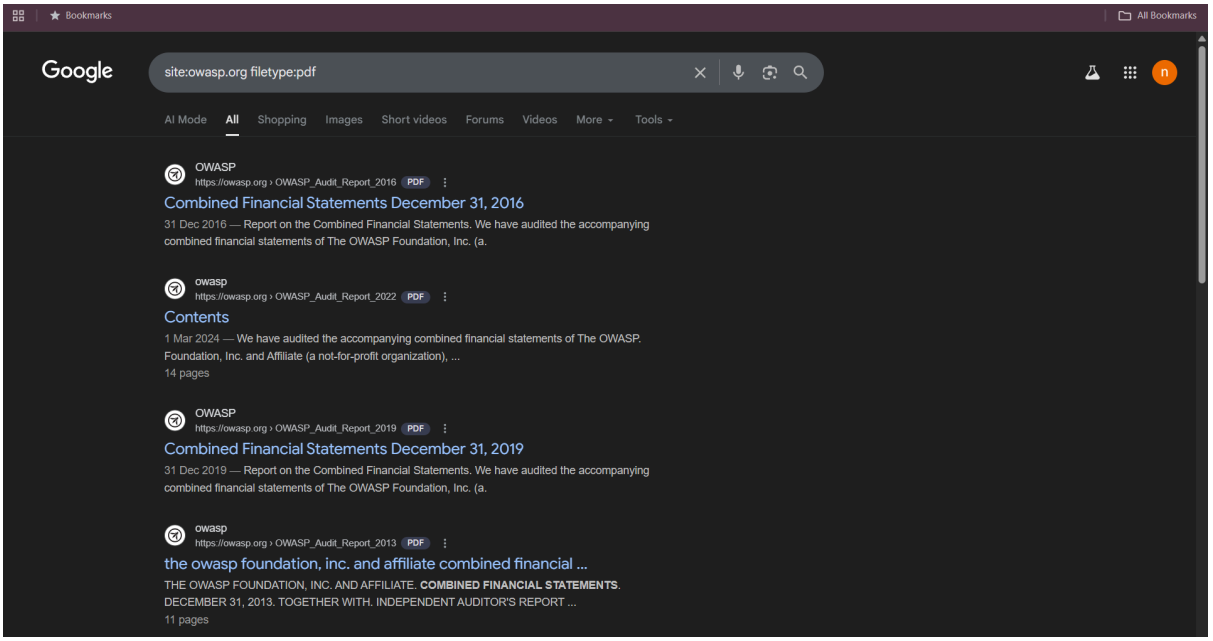


Record: saved documents (PDFs, DOCX), HTML pages, resource URLs. Put filenames into `day3_webcontent/scans/download_list.txt`.

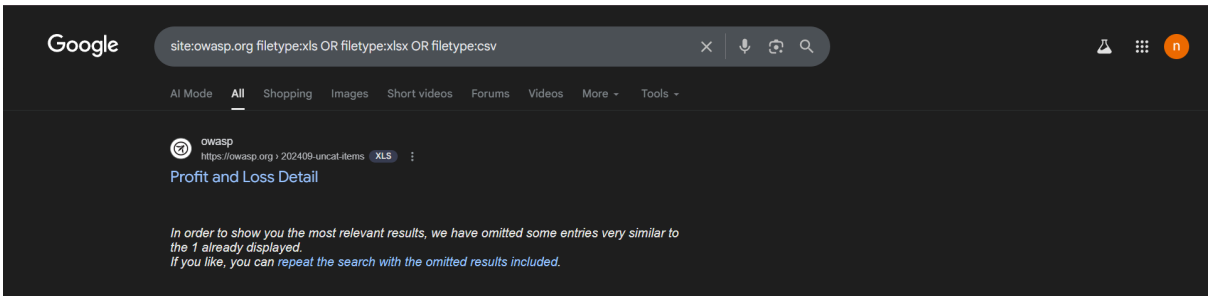
4) Google Dorking — targeted searches to find exposed docs

Use these dorks in Google (and Bing). Replace `https://owasp.org/www-project-juice-shop/` or try without domain to find loose exposures.

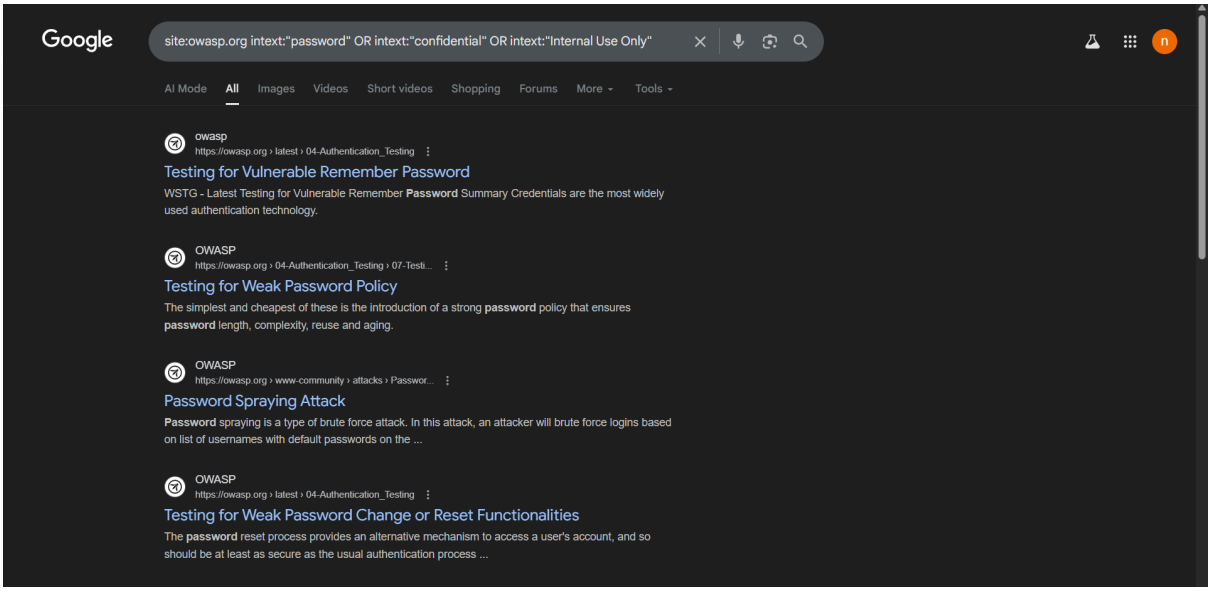
site:owasp.org filetype:pdf



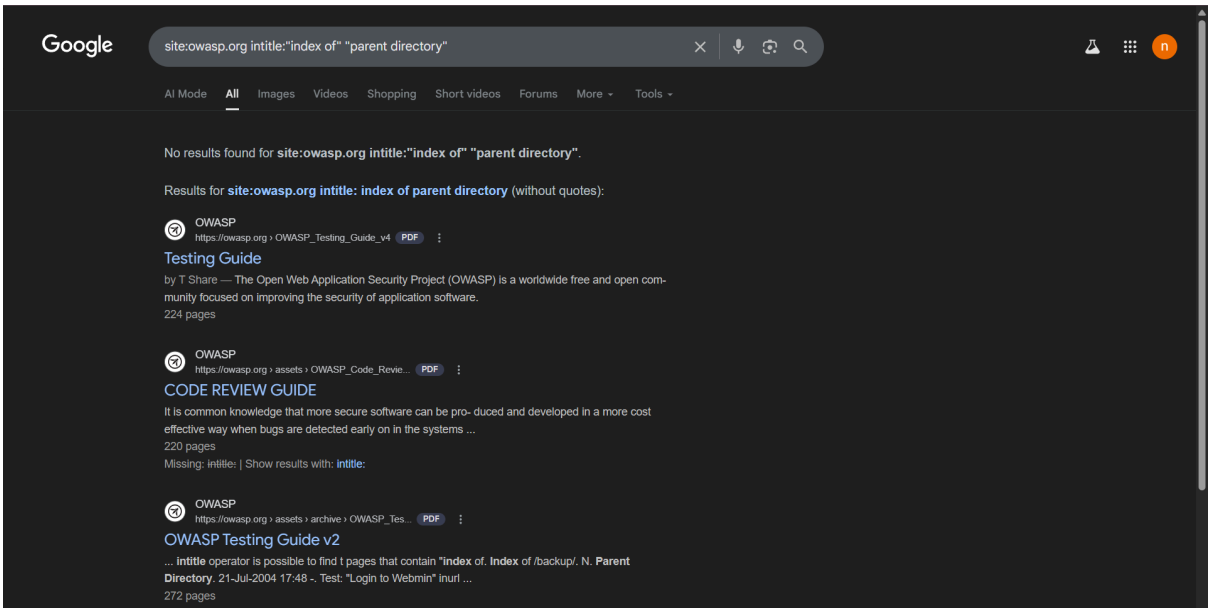
site:owasp.org filetype:xls OR filetype:xlsx OR filetype:csv



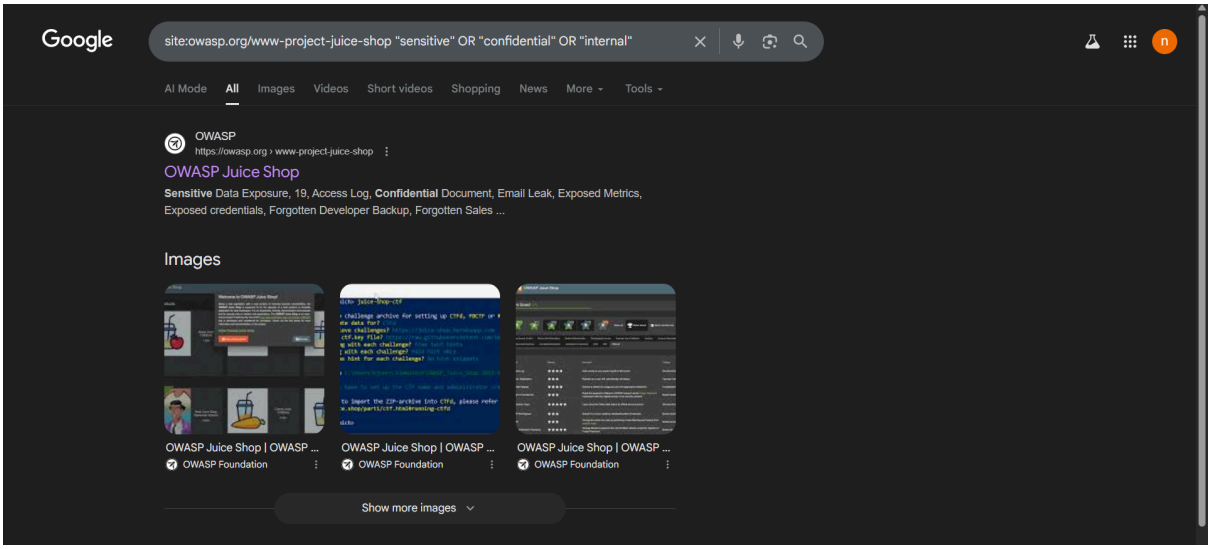
site:owasp.org intext:"password" OR intext:"confidential" OR intext:"Internal Use Only"



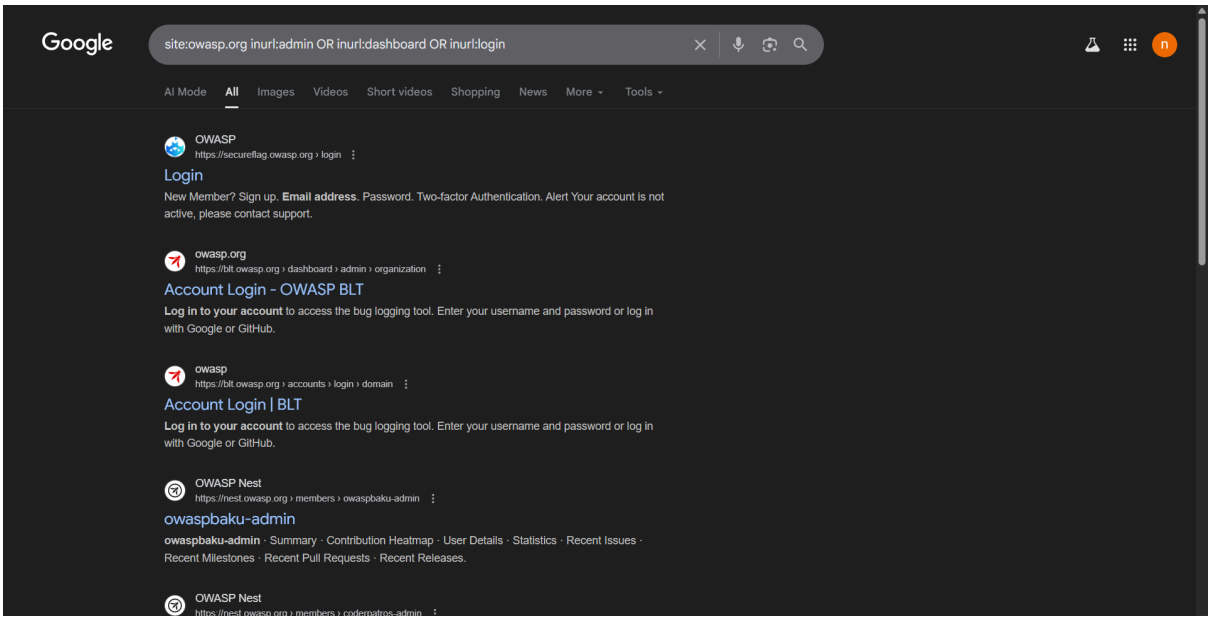
site:owasp.org intitle:"index of" "parent directory"



site:owasp.org/www-project-juice-shop "sensitive" OR "confidential" OR "internal"



site:owasp.org inurl:admin OR inurl:dashboard OR inurl:login

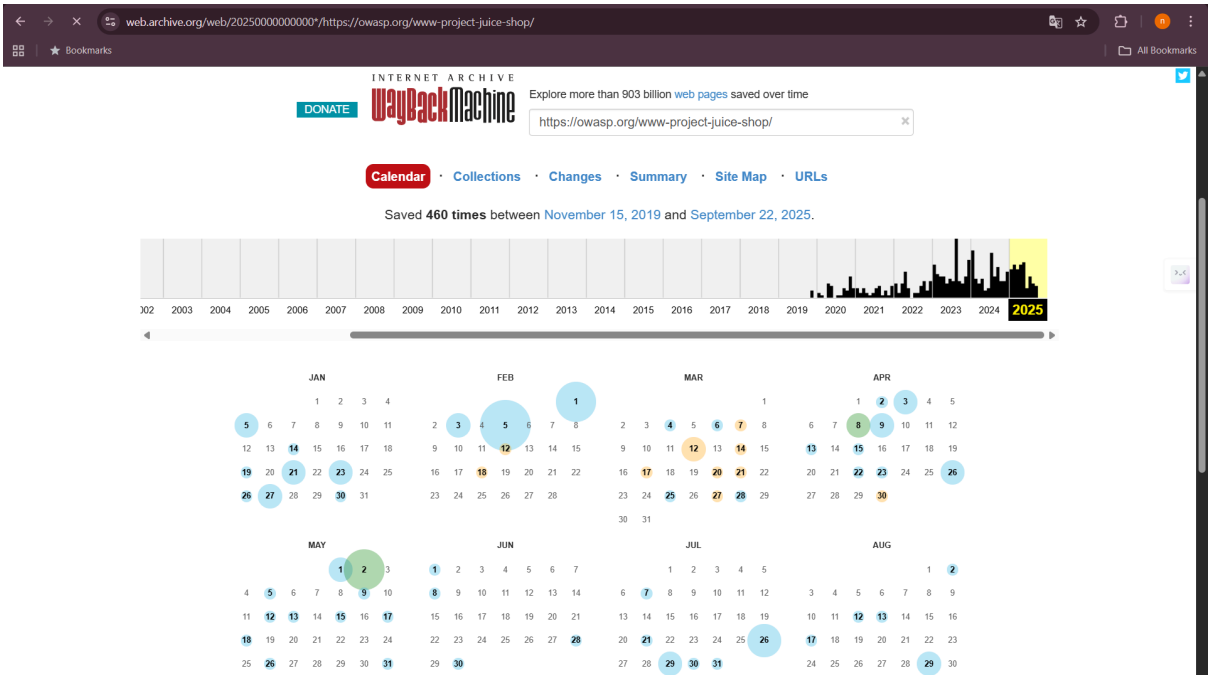


5) Wayback Machine (archive.org) — historical content

Use the Wayback web UI or the `waybackpy` / `waybackpack` tools.

Browser:

- Open https://web.archive.org
- Enter `https://owasp.org/www-project-juice-shop/` → browse snapshots → look for older pages, PDFs, comments, leaked configs.



6) Inspect downloaded documents for sensitive content

- For PDFs: open and search for keywords (`password`, `credential`, `confidential`, `AWS`, `API_KEY`).

```
Select-String -Path "C:\Users\Nidhi [redacted]
[redacted]\day3_webcontent\evidence\OWASP Juice Shop _ OWASP
Foundation.html" -Pattern
"password|apiKey|token|Authorization|mailto:|key="
-CaseSensitive:$false
```

```
PS C:\Users\Nidhi [redacted] > Select-String -Path "C:\Users\Nidhi [redacted]\day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html" -Pattern "password
|apiKey|token|Authorization|mailto:|key=" -CaseSensitive:$false

day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:832:<a href="mailto:bjoern.kimminich@owasp.org">Björn Kimminich</a> and is developed,
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1069:    <td><small>CAPTCHA Bypass, Extra Language, Multiple Likes, Reset Morty's Password</small></td>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1075:    <td><small>Bjoern's Favorite Pet, Change Bender's Password, GDPR Data Erasure, Login Bjoern,
Password Strength, Reset Bender's Password, Reset Bjoern's Password, Reset Jim's Password, Two Factor Authentication</small></td>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1123:    <td><small>Access Log, Confidential Document, Email Leak, Exposed Metrics, Exposed
credentials, Forgotten Developer Backup, Forgotten Sales Backup, GDPR Data Theft, Leaked Access Logs, Leaked Unsafe Product, Login Amy, Login MC SafeSearch, Meta Geo
Stalking, Misplaced Signature File, NFT Takeover, Reset Uvogin's Password, Retrieve Blueprint, Visual Geo Stalking</small></td>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1177:      Bully Chatbot, CAPTCHA Bypass, Extra Language, Login Support Team, Password Strength, Reset
Morty's Password
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1254:      Bjoern's Favorite Pet, Leaked Access Logs, Leaked Unsafe Product, Local File Read, Login
Amy, Login MC SafeSearch, Meta Geo Stalking, Reset Bender's Password, Reset Bjoern's Password, Reset Jim's Password, Reset Morty's Password, Reset Uvogin's Password,
Supply Chain Attack, Visual Geo Stalking, Vulnerable Library
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1287:      Bonus Payload, DOM XSS, Forged Feedback, Login Admin, Login Bender, Login Jim, Password
Strength, Privacy Policy, Reflected XSS, Score Board, View Basket
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1412:    <td style="min-width: 190px"><a
href="https://web.archive.org/web/20250502084243/https://demo.owasp-juice.shop/#/hacking-instructor?challenge=Password%20Strength" target="_blank">Password
Strength</a></td>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1487:    <td><small>Reset Morty's Password</small></td>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1493:    <td><small>Bjoern's Favorite Pet, Password Strength, Reset Bender's Password, Reset Bjoern's
Password, Reset Jim's Password</small></td>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1523:    <td><small>Access Log, Confidential Document, Exposed Metrics, NFT Takeover, Reset Uvogin's
Password</small></td>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1559:      <li><small><a
href="https://web.archive.org/web/20250502084243/https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html" target="_blank">Authorization Cheat
Sheet</a></small></li>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1599:      <li><small><a
href="https://web.archive.org/web/20250502084243/https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html" target="_blank">Forgot Password Cheat
Sheet</a></small></li>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:1611:      <li><small><a
href="https://web.archive.org/web/20250502084243/https://cheatsheetseries.owasp.org/cheatsheets/JSON_Web_Token_for_Java_Cheat_Sheet.html" target="_blank">JSON Web Token
for Java Cheat Sheet</a></small></li>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:2049:  <li><a href="mailto:bjoern.kimminich@owasp.org">Bjoern Kimminich</a></li>
day3_webcontent\evidence\OWASP Juice Shop _ OWASP Foundation.html:2050:  <li><a href="mailto:jannik.hollenbach@owasp.org">Jannik Hollenbach</a></li>
```