Site Scanning Process: Technical Overview

Step-by-Step Technical Process

GSA Site Scanning Program

`r Sys.Date()`

Section 1

Overview

Introduction

This presentation describes the step-by-step technical process by which we 'scan' each Initial URL.

These steps come after the initial process of building the website index.

Initial Data Population

Before any scans take place, the process of ingesting the Initial URL list into the database populates the following fields:

- `Initial URL`
- `Initial Domain`
- `Initial Base Domain`
- `Initial Top Level Domain`
- `Agency`, `Bureau`, `Branch`
- `Data Source`
- `Public`, `Filtered`

Section 2

Scan Architecture

How Scans Are Executed

When scanning commences, this core file dictates which scans are run.

Key characteristics: - Scans run asynchronously (not necessarily in order) - Each scan operates separately - Scans don't communicate with each other

Current Scan Types

The current scans include:

- **primary** - Main URL analysis and data generation
- **dns** - DNS analysis using Node.js library
- **notFound** - 404 behavior testing
- **robotsTxt** - robots.txt file analysis
- **sitemapXml** - sitemap.xml file analysis
- **accessibility** - axe-core accessibility testing
- **performance** - Browser performance metrics
- **security** - Security analysis
- **www** - www subdomain testing

Section 3

Primary Scan Details

Primary Scan Overview

The primary scan uses Puppeteer to load an Initial URL in a headless Chrome/Chromium browser.

It runs multiple scan components asynchronously, found here.

URL Scan Component

urlScan analyzes the Initial URL and notes: - Whether it redirects - Final URL destination - Server status code and filetype - Base domain information - Whether Final URL is on same domain/website

Populates: `URL`, `Domain`, `Base Domain`, `Top Level Domain`, `Media Type`, `Live`, `Redirects`, `Status Code`

URL Scan Logic

Live status: Marked `TRUE` if final server status code is: - 200, 201, 202, 203, 204, 205, 206

Redirect status: Marked `TRUE` if there are one or more components in the redirect chain

CMS Detection

cmsScan looks for code snippets in page HTML and headers that indicate CMS usage.

- Uses patterns from Wappalyzer
- Checks x-server response header for `cloud.gov pages`
- Populates `Infrastructure - CMS Provider` field

Cookie Analysis

cookieScan uses Puppeteer's built-in functionality to: - Note domains of all cookies that load - Populates `Infrastructure - Cookie Domains` field

Digital Analytics Program (DAP)

dapScan captures outbound requests to detect: - DAP tag IDs ('G-CSLL4ZEK4L') - Google Analytics parameters - Self-hosted DAP snippets - URLs ending in `Universal-Federated-Analytics-Min.js`

Populates: `Infrastructure - DAP Detected`, `Infrastructure - DAP Parameters`

Login Detection

loginScan looks for code snippets indicating: - Presence of login forms - Use of specific login providers

Populates: `Infrastructure - Login Provider`, `Infrastructure - Login Detected`

Mobile Optimization

mobileScan looks for viewport meta tag presence: - Detects viewport meta tag code snippets - Populates `Mobile - Viewport Meta Tag Detected`

Required Links Detection

requiredLinksScan searches for required government links as specified on Digital.gov: - Analyzes hyperlinked text and URLs - Populates `Required Links - URL` and `Required Links - Text`

Search Functionality

searchScan detects: - Site search forms in HTML - Search.gov implementation - Populates `Infrastructure - Site Search Detected` and `Infrastructure - Search.gov Detected`

SEO Analysis

seoScan examines search engine optimization elements: - Title, description, og:title, og:description - Article published/modified times - Main element and canonical link presence

Populates multiple `SEO - *` fields

Third Party Services

thirdPartyScan captures outbound requests to: - Identify third-party service domains - Count unique third-party services - Populates `Infrastructure - Third Party Service Domains` and `Infrastructure - Third Party Service Count`

USWDS Detection

uswdsScan looks for US Web Design System elements: - Favicon, fonts (Merriweather, Public Sans, Source Sans) - CSS classes, inline CSS, semantic versions - Calculates likelihood of USWDS presence

Populates multiple `USWDS - *` fields

Section 4

Other Scan Types

DNS Scan

IPv6 Testing: Looks for AAAA record presence in DNS - Populates `DNS – IPv6` field

Hostname Analysis: Filters results containing specific strings to highlight common cloud services - Populates `DNS – Hostname` field

404 Testing

notFound scan appends a random string to the Target URL: - Tests how sites handle 404 errors - Populates `Target URL - 404 Test` field

Robots.txt Analysis

Appends `/robots.txt` to Target URL and analyzes: - Redirect behavior and final URL - Live status and server response - File size and media type - Crawl delay settings - Sitemap locations listed in robots.txt

Sitemap.xml Analysis

Appends `/sitemap.xml` to Target URL and analyzes: - Redirect behavior and final URL
- Live status and server response - File size and media type - Item count in sitemap - Count of PDF URLs listed

Section 5

Scan Status and Results

Scan Status Tracking

Each scan records completion status or failure reasons: - Timeout - DNS resolution error - Invalid SSL certificate - Connection refused/reset - Unknown error

Populates `Scan Status - *` fields for each scan type

Data Recording

`Scan Status - Date` field is populated when scan data is written to the database.

All scan results are stored and made available through the Site Scanning API and analysis tools.

Section 6

Technical Notes

Code Organization

In the scan folders: - `.ts` files are the scans/scan components - `.spec.ts` files are the test files

Performance Considerations

- Asynchronous execution improves performance
- Independent scan operation prevents cascading failures
- Headless browser approach enables comprehensive analysis

Quality Assurance

- Extensive testing framework
- Validation processes for scan results
- Continuous monitoring and improvement