Secret Finding in Web Apps

Eyad El-Taher

February 1, 2026

1 Introduction

Finding misconfigured JavaScript (JS) files in web applications is an important security testing methodology. This vulnerability discovery consists of two main parts: first, finding all JS files of a target website, and second, extracting sensitive information from those JS files.

2 Step 1: Finding JS Files

To find all JavaScript files of a target website, we use Katana, a web crawling framework. The following command enumerates JS files from a list of domains:

```
katana -list {domains.txt} -d 5 -jc | grep ".js$" | uniq | sort |
    tee js_files.txt
```

Listing 1: Command to find JS files using Katana

2.1 Command Explanation

- `katana -list domains.txt`: Use Katana with a list of target domains
- `-d 5`: Set crawling depth to 5
- `-jc`: Enable JavaScript file crawling
- `grep ".js$"`: Filter for files ending with .js extension
- `uniq`: Remove duplicate entries
- `sort`: Sort the output alphabetically
- `| tee js_files.txt`: Shows output on screen AND saves to file
- The saved file `js_files.txt` will contain all discovered JS file URLs

Note: Replace {`domains.txt`} with your actual file containing subdomains of the target website.

3 Step 2: Extracting Sensitive Information

After obtaining the list of JS files, we use **SecretFinder** to extract sensitive information from these files.

3.1 Running SecretFinder

Execute the following command to scan all JS files:

```
cat js_files.txt | while read url; do
    SecretFinder -i $url -o cli;
done > secrets_found.txt
```

Listing 2: Command to extract secrets from JS files

3.2 Command Explanation

- `cat jsfilesgottenfromkatana.txt`: Read the file containing JS URLs
- `while read url; do ... done`: Process each URL line by line
- `python3 SecretFinder/SecretFinder.py`: Run the SecretFinder tool
- `-i $url`: Input URL to scan
- `-o cli`: Output results to command line interface

Note: Replace {`jsfilesgottenfromkatana.txt`} with the actual filename containing your discovered JS files.