Vulnerability Report: Content Security Policy (CSP) Bypass

Content Security Policy (CSP) Bypass

Test Environment:

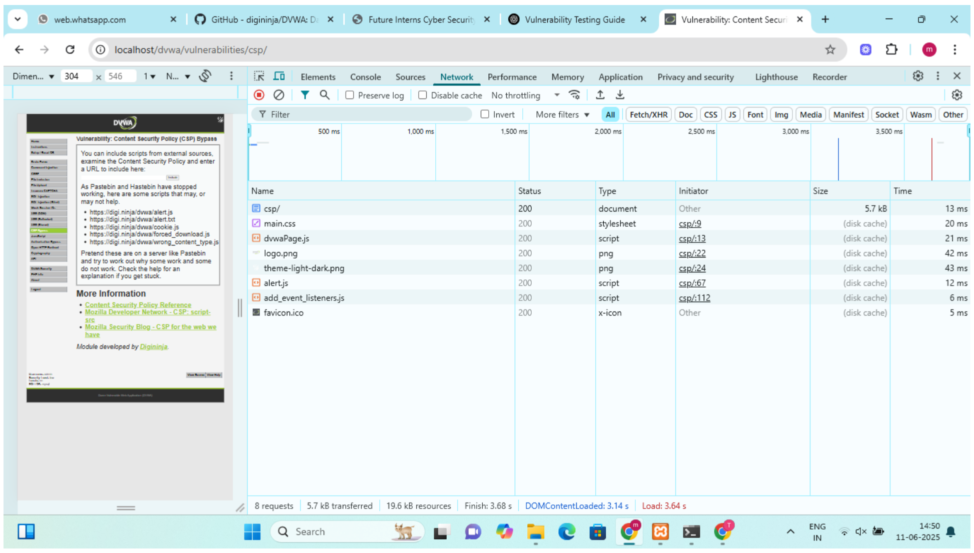
- **Platform**: DVWA (Damn Vulnerable Web Application)
- **Vulnerability Module**: `/dvwa/vulnerabilities/csp/`
- **Browser Tools Used**: Chrome Developer Tools – Elements, Network, and Console tabs

Test Steps Performed:

1. **Opened CSP Module in DVWA** at: `http://localhost/dvwa/vulnerabilities/csp/`
2. **Analyzed the HTML** using DevTools — confirmed script injection point is present.
3. **Inserted an external JavaScript URL** into the provided input field:
4. `https://digi.ninja/dvwa/alert.js`
5. **Observed Network tab** — the external script `alert.js` was:
   - Loaded successfully (HTTP 200)
   - Type: `script`
   - Source: External domain (`digi.ninja`)
   - No CSP restriction error was triggered.
6. **Confirmed execution** of the script — browser executed the script logic (`alert(1)`) without any intervention or warning.

Screenshot Evidence:

1. **Initial HTML Structure Inspection** (Elements tab):
   - Shows a clean `h1` tag with the title and the empty vulnerable code injection area.
2. **Network Activity Confirmation**:
   - `alert.js` was fetched successfully from `https://digi.ninja/`.



localhost/dvwa/vulnerabilities/csp/

Vulnerability: Content Security Policy (CSP) Bypass

You can include scripts from external sources, examine the Content Security Policy and enter a URL to include here:

As Pastebin and Hastebin have stopped working, here are some scripts that may, or may not help.

- https://digi.ninja/dvwa/alert.js
- https://digi.ninja/dvwa/alert.txt
- https://digi.ninja/dvwa/cookie.js
- https://digi.ninja/dvwa/forced_download.js
- https://digi.ninja/dvwa/wrong_content_type.js

Pretend these are on a server like Pastebin and try to work out why some work and some do not work. Check the help for an explanation if you get stuck.

More Information

- Content Security Policy Reference
- Mozilla Developer Network - CSP: script-src
- Mozilla Security Blog - CSP for the web we have

Module developed by Digininja.

Name	Status	Type	Initiator	Size	Time
csp/	200	document	Other	5.7 kB	13 ms
main.css	200	stylesheet	csp/:9	(disk cache)	20 ms
dvwaPage.js	200	script	csp/:13	(disk cache)	21 ms
logo.png	200	png	csp/:22	(disk cache)	42 ms
theme-light-dark.png	200	png	csp/:24	(disk cache)	43 ms
alert.js	200	script	csp/:67	(disk cache)	12 ms
add_event_listeners.js	200	script	csp/:112	(disk cache)	6 ms
favicon.ico	200	x-icon	Other	(disk cache)	5 ms

8 requests | 5.7 kB transferred | 19.6 kB resources | Finish: 3.68 s | DOMContentLoaded: 3.14 s | Load: 3.64 s

Result:

- **External script was fetched and executed.**
- No CSP error occurred, indicating:
- CSP is **either not implemented** or **improperly configured**.

Impact:

- **Security Risk**: Attackers can inject and execute arbitrary JavaScript from external sources.
- **Possible Exploits**:
  - Cookie theft
  - Session hijacking
  - Defacement
  - Redirection to malicious sites

Recommendations:

1. **Implement a strict CSP header**, e.g.:
2. `Content-Security-Policy: default-src 'self'; script-src 'self';`
3. **Avoid allowing script execution from untrusted domains**.
4. **Use nonce or hash-based CSP** for dynamic scripts.