

Myślał kilka sekund

# Bezpieczeństwo i nagłówki bezpieczeństwa – inwood.pl

## 1 Informacje ogólne


**Narzędzie:** OWASP ZAP v2.16.1

**Data skanu:** 2025-05-02

**Zakres:** tylko domena <https://inwood.pl> (rekursywnie wszystkie podścieżki)

**Metoda:** Active Scan (domyślna polityka, siła ataku „Medium”)



## 2 Najważniejsze kategorie usterek

Priorytet	Usterka	Ilość	Rekomendacja
 <b>Krytyczne</b>	<b>Brak HSTS</b> – brak nagłówka <b>Strict-Transport-Security</b> , ryzyko downgrade HTTPS/MITM	13	<b>Włącz HSTS</b> na serwerze:

### # Apache

```
Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains; preload"
```

### # Nginx

```
add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains; preload" always;
...
|  Krytyczne | **Brak X-Content-Type-Options**<br>- możliwe
sniffowanie typu plików (drive-by download, XSS)
| 9      | **Dodaj**:
```Courier New
Header always set X-Content-Type-Options "nosniff"
...
|  **Wysoki**      | **Cache-Control**<br>- nieoptymalne reguły,
przestarzałe zasoby w cache
| 6      | Zweryfikuj `Cache-Control`/`Expires` dla assetów; np.
```Courier New
Cache-Control: public, max-age=31536000, immutable
...
|
```

```

| ● **Wysoki** | **CORS nieokreślony**<br>- brak ograniczeń
cross-origin
| 6 | Ogranicz CORS:
```Courier New
Header set Access-Control-Allow-Origin "https://inwood.pl"
Header set Access-Control-Allow-Methods "GET, OPTIONS"
```
|
| ● **Średni** | **Brak CSP**<br>- żadnej polityki
Content-Security-Policy, ryzyko inline-script/XSS
| 4 | Wprowadź CSP:
```Courier New
Header always set Content-Security-Policy "default-src 'self';
script-src 'self' https://cdn.inwood.pl; img-src 'self' data;";
```
|
| ● **Średni** | **Brak X-Frame-Options**<br>- podatność na
clickjacking
| 4 | Dodaj:
```Courier New
Header always set X-Frame-Options "DENY"
```
|
| ● **Średni** | **Komentarze w kodzie**<br>- mogą ujawniać
wewnętrzne informacje
| 6 | Usuń prywatne komentarze z HTML/JS przed publikacją.
|
| ● **Niski** | **Vulnerable JS Library**<br>- wykryto
Next.js v 14.0.4 z CVE-2024-56332
| 1 | **Zaktualizuj** bibliotekę do najnowszej wersji.
|

```

---

## ## ③ Krok po kroku - jak naprawić

```

1. **Włącz HSTS**
- **Apache** (w `httpd.conf` lub `.htaccess`):
  ```Courier New
  <IfModule mod_headers.c>
    Header always set Strict-Transport-Security
"max-age=31536000; includeSubDomains; preload"
  </IfModule>
  ```
- **Nginx** (w `nginx.conf`):
  ```Courier New
  add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains; preload" always;
  ```

```

```
2. **Dodaj X-Content-Type-Options**
```Courier New
Header always set X-Content-Type-Options "nosniff"
```

### 3. Optymalizuj cache

- o Dla wersjonowanych plików statycznych (\*.css, \*.js, obrazy):

```
Cache-Control: public, max-age=31536000, immutable
```

- o Dla dynamicznych stron (HTML):

```
Cache-Control: no-cache, no-store, must-revalidate
```

### 4. Zabezpiecz CORS

```
Header set Access-Control-Allow-Origin "https://inwood.pl"
```

```
Header set Access-Control-Allow-Methods "GET, OPTIONS"
```

### 5. Wdróż CSP

- Określ zaufane źródła:

```
Content-Security-Policy:
default-src 'self';
script-src 'self' https://cdn.inwood.pl;
img-src 'self' data:;
object-src 'none';
frame-ancestors 'none';
```

### 6. Brak clickjacking

```
Header always set X-Frame-Options "DENY"
```

### 7. Usuń prywatne komentarze

- Przeglądaj wszystkie pliki szablonów i skrypty.
- Usuń zakomentowane fragmenty zawierające dane wewnętrzne lub debug-printy.

## 8. Aktualizuj biblioteki

- Uruchom w katalogu projektu:

```
npm update           # JavaScript
composer update      # PHP/WordPress
```

- Sprawdź wersje i upewnij się, że Next.js > 14.0.4.

## 4 Narzędzia użyte

- **OWASP ZAP** – Active Scan
- *(wcześniej)* Screaming Frog, PageSpeed Insights, Security Headers, SSL Labs, UpGuard