

Podatność CSRF/XSRF

Uwaga! Sprawozdanie zrealizować i oddać według zasad ustalonych w pliku
"BSK 2022 organizacja laboratorium.pdf"

Przypomnienie: zasady postępowania z migawkami zostały przedstawione w pliku
"BSK 2022 migawki.pdf"

Przed przystąpieniem do wykonania ćwiczenia zapoznać się z treścią całego pliku.

Przebieg realizacji ćwiczenia:

- 1) uruchomić maszynę wirtualną *Xubuntu*, wykonać migawkę przed zalogowaniem się, wykonać zrzut ekranu z stanowiskiem komputerowym i czasem wykonywania sprawozdania,
- 2) na maszynie *Xubuntu* uruchomić: *Visual Studio Code* i wprowadzić następujące zmiany w kodzie aplikacji *Cars*:

- w pliku `Program.cs` w linii 24 dodać fragment:

```
options.Cookie.SameSite = SameSiteMode.None;
```

- w pliku `Controllers/CarsController.cs` usunąć całkiem liniijkę 134 zawierającą:

```
[ValidateAntiForgeryToken]
```

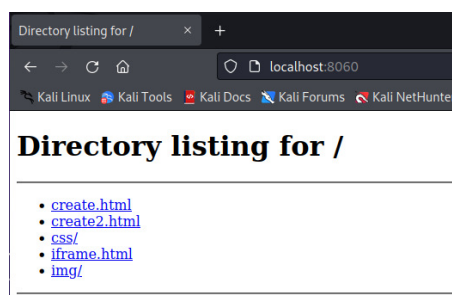
- 3) następnie uruchomić aplikację *Cars* (`dotnet watch run`),

- 4) uruchomić maszynę wirtualną *Kali Linux*, wykonać migawkę przed zalogowaniem się,

- 5) na pulpicie maszyny *Kali Linux* utworzyć folder `lab9`, do którego dostarczyć pliki pochodzące z archiwum `lab9.zip`,

- 6) przejść do emulatora terminala, przejść do folderu `lab9`, uruchomić *pythonowy http.server*:

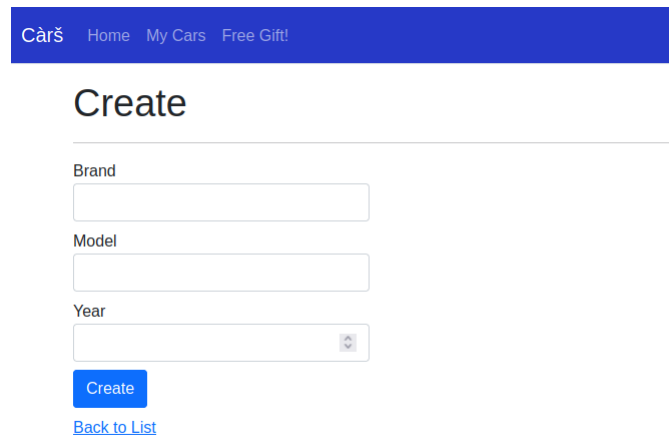
```
cd Desktop/lab9
python3 -m http.server 8060
```



Pozostawić serwer włączony do końca laboratorium.

7) w przeglądarce internetowej na Kali Linux wpisać adres <http://192.168.1.100:5227/> oraz zalogować się na użytkownika Marta (marta@email.com i 1234), otworzyć panel dla programistów (F12) i przejść do zakładki *Storage* → *Cookies* i sprawdzić szczegóły ciasteczka `.AspNetCore.Cookies`,

8) w nowej (drugiej) karcie przejść pod adres: <http://localhost:8060/create.html>,



Càrř Home My Cars Free Gift!

Create

Brand

Model

Year

Create

[Back to List](#)

- wpisać dane samochodu (wybrać rok od 2008 do 2015):



Renault
Laguna
20XX

- upewnić się że panel dla programistów (F12) jest widoczny, przejść do zakładki *Network*,
- na stronie w formularzu kliknąć na przycisk *Create*,
- w panelu kliknąć na wykonane żądanie *POST*,
- sprawdzić czy/jakie ciasteczka były dołączone do żądania (zakładka *Cookies*), oraz *payload* żądania (zakładka *Request* obok *Cookies*),
- podsumować sytuację (m.in. przebieg żądania, sprawdzić czy pojawił się nowy samochód),
- kliknąć 🗑,

9) mając cały czas otwarty panel F12, w drugiej karcie w przeglądarki cofnąć się o jedną stronę ← (powrócić pod adres <http://localhost:8060/create.html>) oraz:

- kliknąć migoczący link w *navbarze*: **Free Gift!**,
- w panelu kliknąć na wykonane żądanie *POST*,
- sprawdzić czy/jakie ciasteczka były dołączone do żądania (zakładka *Cookies*), oraz *payload* żądania (zakładka *Request* obok *Cookies*),
- podsumować sytuację (m.in. przebieg żądania, inicjator, sprawdzić czy pojawił się nowy samochód),
- kliknąć 🗑,

10) wylogować użytkownika Marta z aplikacji,

11) powrócić do VSCode i wprowadzić następujące zmiany w kodzie aplikacji *Cars*:

- w pliku `Controllers/CarsController.cs` przywrócić liniijkę 134 do postaci:

`[ValidateAntiForgeryToken]`

po czym zatrzymać aplikację Cars (Ctrl + C) i włączyć ponownie (dotnet watch run)*,

12) powrócić do przeglądarki na *Kali Linux*, zalogować się ponownie na użytkownika Marta (marta@email.com i 1234), sprawdzić dane ciasteczka `.AspNetCore.Cookies`, usunąć samochody marki Renault i Fiat (*Delete*),

13) ponowić wykonanie punktu 8),

14) w aplikacji Cars dodać dowolny nowy samochód za pomocą prawdziwego formularza *Create New* i porównać *payload* tego żądania *POST* z *paylodem* żądania *POST* z ponowienia w poprzednim punkcie,

15) ponowić wykonanie punktu 9),

16) wylogować użytkownika Marta z aplikacji,

17) powrócić do *VSCode* i wprowadzić następujące zmiany w kodzie aplikacji *Cars*:

- w pliku `Program.cs` w linijce 24 usunąć fragment:

```
options.Cookie.SameSite = SameSiteMode.None;
```

- w pliku `Controllers/CarsController.cs` usunąć całkiem liniijkę 134 zawierającą:

`[ValidateAntiForgeryToken]`

po czym zatrzymać aplikację Cars (Ctrl + C) i włączyć ponownie (dotnet watch run)*,

18) ponowić wykonanie punktu 7)

19) ponowić wykonanie punktu 8),

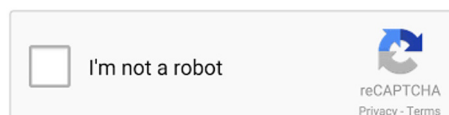
20) ponowić wykonanie punktu 9),

21) wylogować użytkownika Marta z aplikacji,

22) wykonać czynności związane z przywracaniem maszyn do stanu pierwotnego,

23) we wnioskach podsumować/wyjaśnić (o ile nie było wyjaśnione wcześniej):

- podatność *Cross-site request forgery (CSRF/XSRF)*,
- jakie działania, w tym szkody, mogą zostać spowodowane,
- podejścia stosowane w aplikacjach internetowych w celu zapobiegania tej podatności,
- działanie atrybutu `[ValidateAntiForgeryToken]` w aplikacji *ASP.NET (Core)*, (uwzględnić sytuację z aplikacji *Cars* o ile nie była wyjaśniona wcześniej),
- opisać zasadę działania, istotność flagi *SameSite*, jej wartości (uwzględnić sytuację z aplikacji *Cars* o ile nie była wyjaśniona wcześniej).



*) działania związane z możliwym problemem z *Hot Reload*.

Przypomnienie: po zakończeniu ćwiczenia i utrwaleniu wszystkich postępów w sprawozdaniu, wyłączyć maszyny, przywrócić do stanu sprzed migawek i usunąć migawki.

Na końcu nie można zapomnieć także o „odrzuconiu” stanu maszyny.

Zasady postępowania z migawkami zostały przedstawione w pliku "*BSK 2022 Migawki.pdf*"

Po umieszczeniu pliku ze sprawozdaniem w zakładce „*Zadania/Prace*” należy *przestać* rozwiązywanie. Dopiero przed wstawieniem następnego sprawozdania należy *cofnąć przestanie*, umieścić nowy plik i jeszcze raz *przestać*.

Wersja pliku: v1.0