SSTI (Server-Side Template Injection) Prevention

https://icarocomix.github.io/appuntidiprogrammazione

SSTI (Server-Side Template Injection) Prevention

ANALISI TECNICA

PROBLEMA

Esecuzione di codice remoto (RCE) tramite la manipolazione dei parametri che determinano quale file di template deve essere renderizzato.

PERCHÉ

Sanitizzazione rigorosa dei path. Ho scelto di mappare gli input utente su una Enum predefinita prima di restituire il nome della stringa al DispatcherServlet.

ESEMPIO IMPLEMENTATIVO - PART 1

```
// CODICE VULNERABILE: concatenazione diretta di input utente nel
     nome della view @GetMapping("/page") public String
    VULNERABLE_render(@RequestParam String lang, Model model) {
    return "pages/" + lang + "/index"; // VULNERABILE: lang =
    "../../../etc/passwd" o payload SSTI } /* SOLUZIONE 1: Allow-
    list tramite Enum. Solo le viste esplicitamente definite
    nell'Enum possono essere caricate. Qualsiasi input non
    riconosciuto viene rifiutato con un errore controllato. */
public enum AllowedPage { HOME("pages/home/index"),
    ABOUT("pages/about/index"), CONTACT("pages/contact/index"),
    IT("pages/it/index"), EN("pages/en/index"),
    DE("pages/de/index"); private final String templatePath;
    AllowedPage(String templatePath) { this.templatePath =
    templatePath; } public String getTemplatePath() { return
    templatePath; } public static Optional
<AllowedPage>
    fromCode(String code) { return Arrays.stream(values())
        .filter(p -> p.name().equalsIgnoreCase(code))
        .findFirst(); } } @GetMapping("/page") public String
        safeRender(@RequestParam String code, Model model) {
        // Mapping sicuro: solo valori nell'Enum passano al
            motore Thymeleaf AllowedPage page =
            AllowedPage.fromCode(code) .orElseThrow(() -> new
            ResponseStatusException(HttpStatus.BAD_REQUEST,
            "Pagina non riconosciuta: " + code));
            model.addAttribute("pageCode", page.name()); return
            page.getTemplatePath(); // Percorso hardcoded
            nell'Enum: zero input utente nel path } /* SOLUZIONE
            2: Allow-list tramite Map statica per logiche più
            dinamiche. */
```

ESEMPIO IMPLEMENTATIVO - PART 2

```
@Controller public class ThemeController {
    // Mappa statica: i percorsi sono hardcoded, l'utente
         sceglie solo la chiave private static final
        Map<String, String> ALLOWED_THEMES = Map.of(
        "light", "themes/light/index", "dark",
        "themes/dark/index", "high-contrast",
        "themes/high-contrast/index" );
        @GetMapping("/theme") public String
        renderTheme(@RequestParam String theme, Model
        model) { String templatePath =
        ALLOWED_THEMES.get(theme); if (templatePath ==
        null) { log.warn("Tentativo di accesso a tema non
         autorizzato: '{}' da IP: {}", theme,
        request.getRemoteAddr()); return "error/400"; //
        Pagina di errore predefinita } return
        templatePath; } } /* SOLUZIONE 3: Per
        l'internazionalizzazione, uso
        LocaleChangeInterceptor di Spring invece di
        costruire percorsi manualmente. */
    @Configuration public class WebMvcConfig implements
        WebMvcConfigurer { @Bean public
        LocaleChangeInterceptor localeChangeInterceptor()
         { LocaleChangeInterceptor interceptor = new
        LocaleChangeInterceptor();
        interceptor.setParamName("lang");
    // Spring gestisce il cambio locale in modo sicuro
        return interceptor; } @Override public void
        addInterceptors(InterceptorRegistry registry) { r
        egistry.addInterceptor(localeChangeInterceptor())
        ; } } /* Con LocaleChangeInterceptor, il
```

ESEMPIO IMPLEMENTATIVO - PART 3

```
    controller non tocca mai il parametro lang: */
@GetMapping("/page") public String safePage(Model
    model) {
    // Thymeleaf usa il Locale corrente per
        #messages: zero manipolazione del path return
         "pages/index"; // Percorso fisso: Spring
        risolve l'i18n dai messaggi }
```

CONSOLE DI DEBUG

```
> Stato post: [utile]

> Azione: salva_post_ora()


> Requisito:
  Click sull'icona Segnalibro
```