UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Corso di Sicurezza nelle applicazioni 2019-2020

Professore Donato Malerba

Dott. Paolo Mignone

Documentazione del progetto d'esame di Gianluca Scatigna, MAT. 718633

Analisi statica del codice

1. Cross-site scripting

/servlets/CharacterEncodingFilter.java	setta il CharacterEncoding in “UTF-8” per request e response
/servlets/WhitelistingXssFilter.java	matching della regex `[A-Za-z0-9 ]+` per pameter e attribute
/utils/Sanity.java	Regex mail Regex password

- XSS mitigation per le .jsp
  - encoding in “UTF-8”:

```
<%@ page language="java" contentType="text/html; charset=utf-8"
   pageEncoding="utf-8"%>

<meta charset="utf-8">
```

  - Uso di JSTL per l’html escaping
  - Uso di org.owasp.encoder per l’html escaping (supporta più formati)

2. Malicious file upload mitigation:

/utils/Sanity.java	`Void getFileMetadata()`	File parsing dei metadata con Apache Tika
	`boolean checkFileExtension()`	Check dell’estensione richiesta con Tika
	`boolean fileContainsXss()`	Lettura del file alla ricerca della parola “script”
	@MultipartConfig	Dimensione massima del file fino 1Mb: @MultipartConfig(maxFileSize = 1000000, maxRequestSize = 1000000)

3. Directory traversal mitigation:

/WEB-INF/welcome.jsp	L'id univoco, "`projectId`", del file esposto nella select della .jsp per la visualizzazione, è generato randomicamente da 30 bytes e convertito in esadecimale risultando una stringa di 60 chars. Security by oscurity ovvero impedisce all'attaccante di predire l'id univoco del file e conoscere quanti file sono realmente presenti su DB.
	Il parametro `projectId` è analizzato dal @WebFilter WhitelistingXssFilter.java
/servlets/ProjectServlet.java	La richiesta di visualizzazione del file avviene in POST

4. SQL injection mitigation:

• /servlets/WhitelistingXssFilter.java • /servlets/CharacterEncodingFilter.java • /utils/Sanity.java	Input analizzato dai @WebFilter e/o regex
/dao/JdbcFacadeImpl.java	Uso di prepared statement con variable binding

5. Authentication

Cookie based

/dao/CookieUtils.java	Genarazione di una coppia di stringhe random selector-validator
	Entrambi i valori della coppia sono in plaintext in cookie distinti
	I valori sono memorizzati nel database per realizzare il remember me

Il record della tabella user_token che contiene i dati dei cookie è così formato:

ID	Plaintext selector	SHA256(validator)	Timestamp della scadenza del cookie	user_id_foreign_key

L'autenticazione avviene matchando la coppia di valori dei cookie con il record e controllando che il timestamp di scadenza dei cookie non sia stato superato.

La coppia di valori è formata da stringhe random abbastanza lunghe e con una scadenza breve (1 giorno) tali da non poter essere forgiate.

La gestione dei cookie scaduti memorizzati su database è più semplice poiché sincronizzando il timestamp di scadenza tra l'oggetto Cookie e il database non rende più necessario rimuovere i record dei cookie scaduti perché non saranno considerati validi dalla query.

L'hashing del validator sul database impedisce che l'esfiltrazione dei record dei cookie di autenticazione a causa di un data breach possa essere un problema per la sicurezza.

Il rilascio di un nuovo cookie validator e sincronizzazione della sua scadenza su database avviene:

- Se è il cookie è scaduto
- Ad ogni nuova sessione se l'autenticazione è basata su cookie

```
Parametri cookie

cookie.setMaxAge(60*60*24); //24H
cookie.setHttpOnly(true); //Non può essere usato da script JS
cookie.setPath("/");
```

Email & password based

/utils/Account.java	Entrambi i valori sono validati da espressioni regolari
	La password memorizzata su database è il digest di SHA256(PlaintextPassword, Salt)
	Il salt è memorizzato in plaintext su un database differente
	I valori sono manipolati all'interno di byte array e char array

Componente @WebFilter /servlets/AuthFilter.java

Tutte le risorse che richiedono l'autenticazione sono poste nella path /private/*
Il filtro intercetta la request e provvede all'autenticazione mediante cookie o redirect sulla login page

Logout

/servlets/LogoutServlet.java	Invalida la sessione
	Clear dei cookie

Session timeout

web.xml

```
<session-config>
            <session-timeout>10</session-timeout>
</session-config>
```