UNIVERSIDAD NACIONAL DE LA MATANZA

Departamento de Ingeniería e Investigaciones Tecnológicas

Seguridad y Calidad en Aplicaciones Web

Material complementario

"Ejemplo de exposición y defensa de infecciona SQL implementado en JSF"

El siguiente es un bean para transporte de información utilizado en el ejemplo de injeccion, el mismo hereda de BaseBean que es una clase serializable.

```
@ManagedBean(name = "injection", eager = true)
@RequestScoped
public class InjectionBean extends BaseBean {
	public String buscar;
	public List<Producto> productos;

	public String getBuscar() {
		return buscar;
	}

	public void setBuscar(String buscar) {
		this.buscar = buscar;
	}

	public List<Producto> getProductos() {
		return productos;
	}

	public void setProductos(List<Producto> productos) {
		this.productos = productos;
	}
}
```

El siguiente código es un extracto del bean controlador utilizado para implementar la lógica del ejemplo, el mismo se encarga de obtener el bean de datos mediante la propiedad o atributo gestionado e implementar la acción de búsqueda del ejemplo (método submit).

```
@ManagedBean(name = "injectionController", eager = true)
@RequestScoped
public class InjectionController extends BaseController {
	private static final String PAGINA_DE_EJEMPLO = "/pages/tt-2013-
a1";

	@ManagedProperty(value="#{injection}")
	private InjectionBean injectionBean;

	public InjectionBean getInjectionBean() {
```

```
		return injectionBean;
	}

	public void setInjectionBean(InjectionBean injectionBean) {
		this.injectionBean = injectionBean;
	}

	public String submit(){
		Connection c=null;
		try {
			c = getDatabaseConnection();
			restartDatabase();

			List<Producto> tmp = new ArrayList<Producto>();
			ResultSet rs = null;
			/*
			 * Consulta a la base de datos. Ver los ejemplos
			 * en la siguiente seccion.
			 */
			while ( rs.next() ) {
				Producto p = new Producto();
				p.setId(rs.getInt(1));
				p.setNombre(rs.getString(2));
				p.setPrecio(rs.getDouble(3));
				tmp.add(p);
			}
			injectionBean.setProductos(tmp);
		} catch(Exception e){
			/*
			 *  Manejo de excepciones
			 */
		}
		return PAGINA_DE_EJEMPLO;
	}
```

Formas de consultar a la base de datos SQL desde el controlador.

1 – Sin proteccion, este metodo es totalmente INSEGURO y no debe ser utilizado.

```
String sql = "select * from PRODUCTO where ucase(nombre) like
UCASE('%" + injectionBean.getBuscar() + "%')";
rs = c.createStatement().executeQuery(sql);
```

2 – Utilizando parametrizacion por PreparedStatement

```
PreparedStatement pstmt = c.prepareStatement("select * from PRODUCTO
where ucase(nombre) like UCASE(?)");
pstmt.setString(1, '%' + injectionBean.getBuscar() + '%' );
rs = pstmt.executeQuery();
```

3 – Utilizando OWASP ESAPI

```
String esapiSql = "select * from PRODUCTO where ucase(nombre) like
UCASE('%" + ESAPI.encoder().encodeForSQL(new DB2Codec(),
injectionBean.getBuscar() + "%')");
rs = c.createStatement().executeQuery(esapiSql);
```

A continuación se muestra una plantilla xhtml para el ejemplo (En verde y rosa se observan las referencias a los beans JSF de la aplicación)

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      >

<h:head></h:head>
<h:body>
      <ui:composition template="/templates/topten.xhtml">
            <ui:define name="titulo">SCAW - A1-Inyección</ui:define>
            <ui:define name="proteccion">
                  <h:selectOneRadio value = "#{injection.proteccion}"
layout="pageDirection">
                     <f:selectItem itemValue = "none" itemLabel =
"Ninguna"/>
                     <f:selectItem itemValue = "param" itemLabel =
"Parametrizacion" />
                     <f:selectItem itemValue = "esapi" itemLabel =
"OWASP ESAPI" />
                  </h:selectOneRadio>
            </ui:define>
            <ui:define name="base">
                  <h3>Listado de productos</h3>
                  <label class="w3-left">Buscar por nombre de
producto:</label>
                  <h:inputText value="#{injection.buscar}"
styleClass="w3-input w3-border w3-round-large" label="-" title="="/>

                  <h:commandButton value="Buscar"
action="#{injectionController.submit}" styleClass="w3-btn w3-green w3-
right w3-round-large w3-margin"/>

                  <h:dataTable value="#{injection.productos}"
var="producto"
                     styleClass="w3-table-all w3-hoverable"
                     headerClass="w3-green"
                     >
                     <h:column>
                        <f:facet name="header">Codigo</f:facet>

                        <h:outputText value="#{producto.id}"/>
                     </h:column>

                     <h:column>
                        <f:facet name="header">Producto</f:facet>
                        <h:outputText value="#{producto.nombre}"/>
                     </h:column>

                     <h:column>
                        <f:facet name="header">Precio</f:facet>
                        <h:outputText value="#{producto.precio}">
```

```
                                        <f:convertNumber
currencySymbol="$" type="currency" />
                                </h:outputText>
                            </h:column>
                        </h:dataTable>
                </ui:define>
        </ui:composition>
</h:body>
</html>
```