REPORT

URL : <http://zero.webappsecurity.com/>

SOFTWARE USED: NetSparker

As per the results of scanned report,

* Critical issues- 0
* High-20
* Medium- 24
* Low- 93
* Best practice- 70
* Information- 26

Top 3 Vulnerabilities

* Expression Language Injection
* Insecure Transportation Security Protocol Supported
* Password transmitted over HTTP

As a reference from owasp, [**A1:2017-Injection**](https://owasp.org/www-project-top-ten/2017/A1_2017-Injection) is the top most security risk

Even Expression Language injection comes under this category .

**Details:**

The software constructs all or part of an expression language (EL) statement in a Java Server Page (JSP) using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended EL statement before it is executed.

**Common consequences:**

|  |  |  |
| --- | --- | --- |
| **Scope** | **Impact** | **Likelihood** |
| Confidentiality | **Technical Impact:***Read Application Data* |  |
| Integrity | **Technical Impact:***Execute Unauthorized Code or Commands* |  |

**Impacts:**

Almost any source of data can be an injection vector, environment variables, parameters, external and internal web services, and all types of users. Injection flaws occur when an attacker can send hostile data to an interpreter.

Injection can result in data loss, corruption, or disclosure to unauthorized parties, loss of accountability, or denial of access. Injection can sometimes lead to complete host takeover.  
The business impact depends on the needs of the application and data.

**How to Prevent:**

Preventing injection requires keeping data separate from commands and queries.  
\* The preferred option is to use a safe API, which avoids the use of the interpreter entirely or provides a parameterized interface, or migrate to use Object Relational Mapping Tools (ORMs).  
**Note**: Even when parameterized, stored procedures can still introduce SQL injection if PL/SQL or T-SQL concatenates queries and data, or executes hostile data with EXECUTE IMMEDIATE or exec().  
\* Use positive or “whitelist” server-side input validation. This is not a complete defense as many applications require special characters, such as text areas or APIs for mobile applications.  
\* For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter.  
**Note**: SQL structure such as table names, column names, and so on cannot be escaped, and thus user-supplied structure names are dangerous. This is a common issue in report-writing software.