ZAP scanning rules are divided into two main parts: Passive Scan vs Active Scan.   
Passive: <https://www.zaproxy.org/docs/desktop/addons/passive-scan-rules/>   
Active: <https://www.zaproxy.org/docs/desktop/addons/active-scan-rules/>

Currently, all these passive and active scanning rules cover all the top ten vulnerabilities, in which the alerts that ZAP generates derived from these scanning rules. Hence, the idea is to make a template for every scanning rules, and the corresponding alerts will use that corresponding template. I use ZAP local API to get all these alerts using the link <http://localhost:8080/UI/> , in order to get the Active Scan (ascan) and Passive Scan (pscan) resulting point. I save all the results in GitHub in the /references folder with the naming as activescanrules\_json and passivescanrules\_json.

The current game plan is to scrape all the information details inside ZAP, like summary, solution (temp), other info, references, tags, etc. Then, based on the information provided, I can make the template guide. The current structure idea is as:

* Description
* Code Vulnerabilities (where the weakness occurs)
* Pseudocode
* Actual Sample Code
* Threats Avoided

The JSON output from above API scraping results in a total of 109 scanning rules, with 48 being active and 61 being passive. Considering that we are focusing onto OWASP top ten first, we can narrow down our scope and make the template for 26 of them first, being:

**## A01: Broken Access Control**

* **Path Traversal** (Active)
* **Remote File Inclusion** (Active)
* **Directory Browsing** (Passive)

**## A02: Cryptographic Failures**

* **Cookie Without Secure Flag** (Passive)
* **Strict-Transport-Security Header** (Passive)
* **Hash Disclosure** (Passive)
* **Heartbleed OpenSSL Vulnerability** (Active)

**## A03: Injection**

* **SQL Injection** (Active)
* **Cross Site Scripting (Reflected)** (Active)
* **Cross Site Scripting (Persistent)** (Active)
* **Remote OS Command Injection** (Active)
* **CRLF Injection** (Active)

**## A04: Insecure Design**

* **Absence of Anti-CSRF Tokens** (Passive)
* **GET for POST** (Active)

**## A05: Security Misconfiguration**

* **Content Security Policy (CSP) Header Not Set** (Passive)
* **Anti-clickjacking Header** (Passive)
* **X-Content-Type-Options Header Missing** (Passive)
* **.env Information Leak** (Active)
* **.htaccess Information Leak** (Active)
* **Source Code Disclosure - /WEB-INF Folder** (Active)

**## A06: Vulnerable and Outdated Components**

* **Vulnerable JS Library (Powered by Retire.js)** (Passive)
* **Log4Shell** (Active)
* **Spring4Shell** (Active)

**## A07: Identification and Authentication Failures**

* **Session ID in URL Rewrite** (Passive)
* **Weak Authentication Method** (Passive)

**## A08: Software and Data Integrity Failures**.

* **Insecure JSF ViewState** (Passive)