

# FARRAH QASEM

## SR. SOFTWARE ENGINEER / APPLICATION SECURITY ENGINEER



farrah.qasem@outlook.com



(202) 810-3141



Bethesda, MD 20814

### EXPERTISE

#### Methodology:

Agile (Sprint, Scrum), Waterfall

#### Frameworks:

Spring MVC, Spring Security,  
AngularJS,  
Hibernate, JUnit, Mockito,  
Marklogic Client REST API  
ElasticSeach Client REST API

#### Programming Languages:

Java 6/7/8, Unix Shell Scripts, SQL,  
JavaScript, TypeScript, Python

#### Servers, Containers and Tools:

Apache Tomcat,  
Oracle, PostgreSQL, MySQL,  
Git, Jenkins, Jira, Nexus,  
Eclipse, Maven,  
Fortify SSC, SonarQube,  
Jenkins Dependency Checker,  
Splunk Logs, Confluence Plugins

### EDUCATION

M.Sc in Advanced Software  
Engineering,  
King's College,  
University of London, UK, Sep 2004.

### CERTIFICATIONS

CompTia Security Plus, 2021  
AWS Solutions Architect, 2017.  
Certified Scrum Master, 2015.  
Sun Microsystems Certified  
Java Web Component  
Developer.  
Sun Microsystems Certified  
Java Programmer.

### PROFILE

Seasoned development professional with over 10 years of experience in the Government, Financial and Private sectors. Federal experience includes developing, maintaining, and reviewing code in accordance with NIST SP 800-53.

### PROJECT HIGHLIGHTS

**Working as a contractor for Department of Defense, Fort Belvoir, VA | JAN 2020 – present**

#### *Senior Software Engineer / Application Security Engineer*

Currently working as a Software Developer and Security Professional on DoD projects (Application Security, Identity and Access Management and O&M Teams) with a focus on secure coding and secure development lifecycle in accordance with NIST SP 800-53.

**Teksystems / Raytheon, Silver Spring, MD | NOV 2018 – Dec 2019**

#### *Senior Software Engineer*

Worked as a Java and Python developer on NOAA's AWIPS project.

[AWIPS](#) is a weather display and analysis package developed by the National Weather Service for operational forecasting. The python-awips package provides a data access framework for requesting meteorological and geographic datasets from an [EDEX](#) server.

**Aceinfo Solutions / NOAA, Silver Spring, MD | AUG 2014 – OCT 2018**

#### *Senior Software Engineer*

At National Oceanic and Atmospheric Administration (NOAA), worked as back-end Java Developer with OSGi bundles running on Apache Karaf. It used JAVA 7 and 8, Apache Camel frameworks, Apache ActiveMQ, Hibernate, and PostgreSQL. Deployed new code and bug fixes in AWS environment for development purposes.

**Starpoint Solutions / Federal Reserve Bank of NY | JAN 2013 – JUL 2014**

#### *Senior Software Engineer*

Worked with the Development Services team, which provided central support for development infrastructure comprising of both custom and 3rd Party components including Subversion, Maven, Hudson, and Nexus.

**Agreeya Solutions / Verizon Wireless, Warren, New Jersey | JUN 2012 – DEC 2012**

#### *Software Engineer*

Worked on Verizon's complex network system ACSS, which communicates with front end and middleware software via web services.



## PROFESSIONAL EXPERIENCE:

**Project: Application Security, Identity and Access Management and O&M Projects - Department of Defense, Fort Belvoir, VA**  
**Position: Senior Software Engineer / Application Security Engineer** **January 2020 - present**

Currently working as a Software Developer and a Security Professional contractor with DoD (Application Security, Identity and Access Management and O&M Teams) with a focus on secure coding and secure development lifecycle. As part of the Application Security Team, I worked on maintaining the security aspect of the DTIC projects. This included periodical vulnerability assessment of the project code base using Fortify SSC scans and Jenkins Dependency Checker scans, design and implement remediation involving the development teams and configure and maintain Jenkins jobs for these assessments. The Jenkins Dependency Checker plugin uses CVE from NIST NVD and OWASP.

### **Responsibilities:**

- Worked on the Application Security, Identity and Access Management and O&M Teams.
- Implemented security controls for federal IT systems in accordance with NIST SP 800-53.
- NIST SP 800-53 outlines 20 different control families. I've worked in several areas including Access Control (AC), Configuration Management (CM), Systems and Services Acquisition (SA).
- For **Systems and Services Acquisition (SA)**, I implemented Secure Coding. I used DISA STIGS in FORTIFY Software Security Center (Fortify SSC is a **Static Code Analysis Tool**) to check code for security vulnerabilities. DISA STIG refers to an organization (DISA — Defense Information Systems Agency) that provides technical guides (STIG — Security Technical Implementation Guide). DISA is part of the Department of Defense (DoD). It's a combat support agency that provides IT and communication support to all institutes and individuals working for the DoD. DISA oversees the IT and technological aspects of organizing, delivering, and managing defense-related information. This includes STIG guidelines. These guides outline how an organization should handle and manage security software and systems. Fortify SSC which is a centralized server to track Software vulnerabilities uses this STIG to check the vulnerabilities against the code. As a developer I fixed these vulnerabilities in the code following STIG guidelines and recommendations provided by Fortify SSC. It then went through a Security Review Process where a Security Reviewer had to sign off on the fixed code before it could be deployed on Stage and Production server.
- For **Access Control**, I worked on fixing the code in accordance with Access Control rules defined by the organization. For example, who can have access to the classified information on the MarkLogic Server. The application code had to interact with the request and response from the ML server. I also configured developer access to Git repositories on Gitlab. In addition, worked on the Identity and Access Management team which has a suite of applications created and maintained by DoD. AIM-Siteminder project was a custom authentication scheme implementation using Siteminder authentication API.
- For **Configuration Management**, I worked on Jenkins Job Configuration for building and deploying application code on AWS Development environment. I also configured Fortify SSC scan jobs and Dependency Checker plugins on Jenkins and Confluence plugin and log configuration on Confluence 7.13.
- For **Log Monitoring and Analysis**, Used Splunk logs to track issues on Development, Stage and Production environment.
- Did peer **Code Review**, code analysis for bug fixing, testing on AWS Development and Stage environment.
- Used Jira to track issues, Microsoft Teams and Cisco VPN for remote work.

**Environment:** Java 1.8, Spring 5.2.8, Spring-Security 5.3.3, Elasticsearch-rest-client 6.1.2, MarkLogic Client API 3.0.5, AngularJS 1.12.5, Tapestry 5.5. **Tools used for secure coding and code quality:** Fortify Software Security Center, Jenkins Dependency Checker, SonarQube for Code Quality, Splunk Log Monitoring, Eclipse, Gitlab, Git, Maven.

**Company: Teksystems (Client: Raytheon)**

**Project: AWIPS, NOAA, Silver Spring, MD**

**Position: Senior Software Engineer**

**November 2018 – December 2019**

Worked as a Java and Python developer on National Weather Service - NWS's AWIPS project. The Advanced Weather Interactive Processing System (AWIPS) is a technologically advanced information processing, display, and telecommunications system that is the cornerstone of the National Weather Service modernization and restructuring. AWIPS is an interactive computer system that integrates all meteorological and hydrological data with satellite and radar imagery. This helps the forecaster prepare and issue more accurate and timely forecasts and warnings. Forecasters at more than 130 NWS offices across the nation use the capabilities of AWIPS to make increasingly accurate weather predictions and to dispense rapid, highly reliable warnings and advisories.

Hierarchical Data Format (v.5) - HDF5 is the primary data storage format used by AWIPS for processed grids, satellite and radar imagery and other products. HDF5 supports multiple types of data within a single file. For example, a single HDF5 file of radar data may contain multiple volume scans of base reflectivity and base velocity as well as derived products such as composite reflectivity. The file may also contain data from multiple radars.

PyPIES, Python Process Isolated Enhanced Storage, (httpd-pypies) was created for AWIPS to isolate the management of HDF5 Processed Data Storage from the EDEX processes. PyPIES manages access, i.e., reads and writes, of data in the HDF5 files. In a sense, PyPIES provides functionality similar to a DBMS (i.e PostgreSQL for metadata); all data being written to an HDF5 file is sent to PyPIES, and requests for data stored in HDF5 are processed by PyPIES. PyPIES is implemented in two parts: 1. The PyPIES manager is a Python application that runs as part of an Apache HTTP server and handles requests to store and retrieve data. 2. The PyPIES logger is a Python process that coordinates logging.

#### **Responsibilities:**

- Use CentOS as the development environment in local machines
- Use Python, Java7/8, Spring and Hibernate frameworks and PostgreSQL.
- Work on bugs/issues as assigned for the NOAA AWIPS system.
- Coordinate with remotely located team-members.
- Use eclipse IDE, Git for version control and code review process.

**Environment:** Java 1.8, Python 2.7, Apache Camel 2.14.4, Spring Batch 3.0, Hibernate 4.3.6, PostgreSQL 9.4.3, Eclipse Equinox, Ant, Jenkins.

**Company: Aceinfo Solutions**

**Project: NLETS, NOAA, Silver Spring, MD**

**Position: Senior Software Engineer**

**August 2014 – October 2018**

NOAA's National Law Enforcement and Tele Communications System (NLETS) is a central data dissemination system. It reads data from different channels, processes it and transports it to different destinations (SOAP server, FTPS server, etc). As a Java developer, I worked mainly on the backend OSGi/karaf bundles, using Apache Camel, activeMQ, Spring and Hibernate. OSGi (Open Service Gateway Initiative) is a Java framework for developing and deploying modular software programs and libraries. This project used 2-to-3-week sprints for development and testing.

#### **Responsibilities:**

- Use Fedora VM as the development environment in local machines
- Use Java7/8, Apache Camel, Spring and Hibernate frameworks.
- Create OSGi bundles based on functional requirement. Ingesting, parsing and transporting logic in java classes were grouped together to form osi bundles. The bundles contained mainly java code, configuration files and xml files called camel routes or blueprints which controlled each bundle's code execution sequence.
- Implement load-balancers with Apache ActiveMQ for parallel systems running on multiple servers
- Write Junits. Use mocking frameworks (jmockit) to mock objects for junits.
- Write Exception handling code in java classes and in Camel routes.
- Use java regular expression api for parsing logic
- Write java code for transforming messages suitable for SOAP and FTP
- Write shell scripts for tasks such as automatic deployment of code in Karaf container

- Write cron jobs in Apache Camel for automatic archiving of files based on its expected expiration date
- Use configuration files for database, soap and ftps connectivity and AtiveMQ queue names
- Use Soap Server simulator and VSFTPD as FTP server for local testing
- Test the code following DT&E. Add new tests (modify old tests) in DT&E.
- Use eclipse IDE, Git/gerrit for version control and code review process.
- Deploy new code and bug fixes in AWS environment for development purposes
- Build EC2 instances, configure load balancers, create separate database instances and set up users, assigning roles and groups using Identity Access Management (IAM).

**Environment:** Java 1.7 / 1.8, Apache Camel 2.14.4, Apache ActiveMQ 5.10.2, Hibernate 4.3.6, Maven, PostgreSQL 9.4.3, Struts, Jenkins, Eclipse, Git.

**Company:** Starpoint Solutions

**Project:** Development Services, Federal Reserve Bank, New York, NY

**Position:** Senior Software Engineer

**January 2013 – July 2014**

Worked with the Development Services team. As part of Common Services, the Development Services (DS) team provides central support for development infrastructure which comprises of both custom and 3rd Party components including Subversion, Maven, Hudson, and Nexus. The team works closely with all members of the development team, especially developers, to ensure that developers are getting the greatest value out of the tools (e.g. Eclipse) provided to them.

## **Responsibilities:**

### *Infrastructure Tool Development*

- Created custom plugins for Hudson using Java
- Created tools in Java used by developers, e.g. changing passwords for maven, checking checksums of downloaded software, checking the names of Wiki pages for invalid characters etc.
- Wrote custom unix shell scripts to set the environment for Hudson, e.g. to use a new version of Java – Tomcat's catalina.sh calls setenv.sh which was modified to use a new version of Java
- Wrote custom installation unix scripts which automated the Hudson installation process. Hudson install script - runs the install commands based on some conditions, e.g., before starting Hudson check if it's already running, if it's running shut it down etc.
- Customized Eclipse to include features such as PMD and Findbugs to detect bugs in the code developed by programmers
- Configured Maven POMs with PMD and Findbugs plugins to detect bugs in the code during Maven builds
- Created XML files for holding PMD and Findbugs rules; configured these files in maven parent POM. All other team's POMs extended from this POM.

### *Development Services*

- Configured updated versions of Nexus, Hudson, Maven, JDK on Solaris Server as Admin
- Used Unix BASH commands to install software on Solaris
- Created and maintained Wiki instructional pages
- Supported developers having issues with Eclipse, Maven and SVN.
- Worked with Maven clean, install, deploy, release:prepare and release:perform
- Configured jobs/projects in Hudson as Admin
- Setup group ids, privileges in Nexus for several development teams to have access to certain Nexus repositories and artifacts
- Supported developers having issues with new installed versions of JDK

**Environment:** Java 1.6; Mavan 2.2, 3.0; Hudson 2.2.0, 3.0.1; Nexus 1.9.2.4, 2.3.0 - 04; Eclipse Indigo-3.7, Juno-4.2, Kepler-4.3; Solaris 10; HP QC; spring-core-3.2.6, Oracle 11g, ojdbc6-11.2.0.4

**Company: Agreeya Solutions**

**Project: Customer Care System, Verizon Wireless, Warren, NJ**

**Position: Software Engineer**

**Jun 2012 – Dec 2012**

In Verizon's network systems, Automatic Customer Support System (ACSS) ESB is a software component, which represents the piece of software that lives between the business applications and enables communication among them, and tries to replace all direct contact with the applications on the bus by XML HTTP web services, so that all communication takes place via the ESB. Since the front end of ACSS accesses middleware by web services, ACSS ESB-MW can provide plenty of complex services for the front end and shift a lot of business logic load from front end to middleware.

#### **Responsibilities:**

- Created Web Services in Java
- Worked with XSLT for creating XSL files for request and response
- Created XML documents for registering services and task/job orchestration
- Integrated applications and orchestrated services in a service-oriented architecture (SOA).
- Created Request and Response XML schemas
- Created handler and utility classes for middleware.
- Used WorkManager API to schedule work/task
- Wrote SQL and PL/SQL Stored Procedures for DAO classes
- Used WebSphere Application Server v8.0
- Used IBM Rational Software Architect as IDE
- Used Accurev as the Source Control System
- Used Altova XMLSpy for XML schema generation

**Environment:** Java 1.6, Log4j 1.2.14, JAX-WS 2.0, SOAP 1.1, WSDL 1.1, Websphere v8.0, IBM RSA 8.0, Accurev

**Company: Accenture**

**Project: Admin center, JP Morgan Chase, Jersey City, NJ**

**Position: Java Developer**

**Feb 2010 – Jun 2012**

This was a new project without a single line of pre-existing code. JPMC's Corporate Internet Group created a client administration tool (Admin Center) for their Treasury Services clients. Administration Center became the integrated hub for clients and employees to view and manage administrative activities, and an "operating system" that provisions client, product, user information and entitlements to applications on any channel globally.

#### **Responsibilities:**

- Utilized Spring MVC framework; Hibernate for persistence layer.
- Used J2EE design patterns like Data Transfer Object, Data Access Object, Façade, Decorator and composite.
- Wrote Junit tests for Action classes, Helper classes and Dao classes.
- Used Mockito Framework to mock objects for tests classes.
- Used Spring Webflow, JSF, Richfaces for presentation layer.
- Created XHTML pages using JSF and Richfaces tags.
- Wrote Javascripts, JQuery functions for UI.
- Added CSS style classes to xhtml pages when needed.
- For non-UI use cases, worked with Drools WorkItemHandler Java classes, and created workflows.
- Created DAO classes for the data access layer.
- Used the Object Model to write Hibernate HQL queries and DAO finder methods for data access;
- Wrote Hibernate Transformers and used the Criteria feature of Hibernate.
- Used TOMCAT as application server.

- Used Clearcase and Subversion for version control.

**Environment:** Eclipse 3.6, Java 1.6, Tomcat v6, SQL Developer 3.0, Subversion 1.6.5, Maven 2.1.0, Soap-UI 4.0, log4j-1.2.16, jsf-api-1.2-14, jsf-facelets-1.1.14, servlet-api-2.5, spring-core-3.0.2, spring-security-3.0.2, spring-webflow-2.1.0, spring-faces-2.1.0, richfaces-3.3.3, jta-1.1, jms-1.1, Websphere MQ v7.0, ojdbc6-11.2.0, hibernate-core-3.3.2, hibernate-annotations-3.4.0, hibernate-validator-4.0.2, javax-validation-api-1.0, javax-persistence-api-1.0, quartz-scheduler-2.1.1, mule-core-3.1.2, drools-5.1.1.7, ehcache-core-2.3.2, aspect-weaver-1.6.10, junit-4.8.1, mockito-1.8.4, powermock-mockito-1.3.8