

# **Detecting Security Design Patterns in Open-Source Projects**

Anna Rowena Waldron

CSS497: Capstone Report  
Spring 2022

Advisor: Professor Hazeline Asuncion  
Project Sponsor: Professor Brent Lagesse

## Table of Contents

|                                     |           |
|-------------------------------------|-----------|
| <b>Research Background</b>          | <b>2</b>  |
| <b>Purpose</b>                      | <b>2</b>  |
| <b>Process and Tasks</b>            | <b>3</b>  |
| <b>Results and Discoveries</b>      | <b>6</b>  |
| <b>Future Work and Improvements</b> | <b>12</b> |
| <b>Reflection</b>                   | <b>13</b> |

# Detecting SDP's in Open-source Projects

## Research Background

This research is to aid in developing a tool to help software engineers to quickly identify security design patterns. This tool would allow users to save time and resources, maintain security practices in legacy code, detect if secure code is reusable, and to mitigate common software vulnerabilities. It also assists in maintaining up-to-date security systems through easy SDP detection and security requirements comparisons. The research subject is improving the lightweight detection of security design patterns in source code. The purpose of the research is to expand testing to detect security design patterns in open source code.

The results of the tests conducted increase the capabilities of SDP detection by discovering limitations in PatternScout and Codeontology, improving understanding of the differences between SDP's and design patterns, testing the ease of use of the SDP detection algorithm, and providing information for future creation of .rq and .nt files to be done easier.

## Purpose

The research goal is to find SDP uml class diagrams online to begin testing on open source projects to test SDP detection. The current work is to start research on testing detecting security design patterns in source code. This testing aids in improving upon the algorithm for lightweight detection of SDP's by noting improvements and documenting difficulties, solutions and workarounds.

The purpose is to Begin preliminary testing of SDP's using different tools as part of a pattern detection algorithm. Tools to test include PatternScout and Codeontology. The focus will be on large open-source projects larger than 1,000 lines of code. These open-source projects are used to test the tools on different types of source code and coding styles and for testing the scalability of

the tools on large projects. Testing will be used in developing improvements in the security design pattern detection algorithm and discovering weaknesses and capabilities of the tools. Started testing to detect SDP's using UML class diagrams of the SDP's found published online.

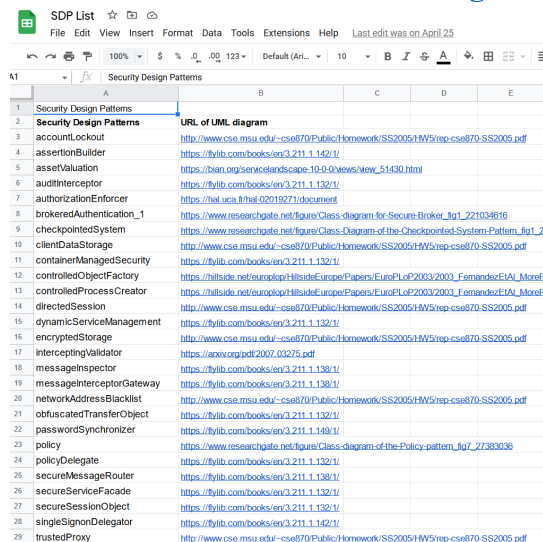
## Process and Tasks

The following steps are used to test for SDP's in open source projects. Some steps can be skipped if the .rq file is already created which, is the SPARQL query of the security design pattern without prior knowledge of what the class diagram is for the SDP.

### Tasks

- Found a total of 27 different security design patterns with published and available uml class diagrams online. Many of the diagrams came from university publications or research papers.
  - There seems to be no exact standard for uml class diagrams of security design patterns. This may be because using class diagrams of security design patterns is a new area so findings on the topic were extremely limited.
  - Based off the class diagrams I found online, I used VisualParadigm to create [UML class diagrams](#) for each of the SDP's. This involved translating what I saw onto VisualParadigm and making decisions on how to adjust things to fit the new format and adjust stereotypes, relations, methods, attributes, types, and visibility. Many of the class diagrams were very simple and many did not include attributes and methods. This made some SPARQL queries too broad.

### Online Evidence of SDP Class Diagrams



| SDP List  |                           |   |   |   |
|---|---------------------------|---|---|---|
| File Edit View Insert Format Data Tools Extensions Help Last edit was on April 25 |                           |   |   |   |
| Security Design Patterns  |                           |   |   |   |
| A   | B                         | C   | D | E |
| 1   | Security Design Patterns  | URL of UML diagram  |   |   |
| 2   | Security Design Patterns  | URL of UML diagram  |   |   |
| 3   | accountLogout             | <a href="http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf">http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf</a>                         |   |   |
| 4   | assertionBuilder          | <a href="https://hylib.com/books/en/3.211.1.142/1/">https://hylib.com/books/en/3.211.1.142/1/</a>   |   |   |
| 5   | assetValuation            | <a href="https://bian.org/services/landscape-10-0-0/Views/view_51430.html">https://bian.org/services/landscape-10-0-0/Views/view_51430.html</a>   |   |   |
| 6   | auditInterceptor          | <a href="https://hylib.com/books/en/3.211.1.132/1/">https://hylib.com/books/en/3.211.1.132/1/</a>   |   |   |
| 7   | authorizationEnforcer     | <a href="https://hal.uca.fr/hal-02019271/document">https://hal.uca.fr/hal-02019271/document</a>   |   |   |
| 8   | brokeredAuthentication_1  | <a href="https://www.researchgate.net/figure/Class-diagram-for-Secure-Broker_fig1_221034616">https://www.researchgate.net/figure/Class-diagram-for-Secure-Broker_fig1_221034616</a>                   |   |   |
| 9   | checkpointedSystem        | <a href="https://www.researchgate.net/figure/Class-Diagram-of-the-Checkpointed-System-Pattern_fig1_2">https://www.researchgate.net/figure/Class-Diagram-of-the-Checkpointed-System-Pattern_fig1_2</a> |   |   |
| 10  | clientDataStorage         | <a href="http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf">http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf</a>                         |   |   |
| 11  | containerManagedSecurity  | <a href="https://hylib.com/books/en/3.211.1.132/1/">https://hylib.com/books/en/3.211.1.132/1/</a>   |   |   |
| 12  | controlledObjectFactory   | <a href="https://halide.net/europlp/HalideEuropePapers/EuroPLP2003/2003_FernandezEIAI_MoreB">https://halide.net/europlp/HalideEuropePapers/EuroPLP2003/2003_FernandezEIAI_MoreB</a>                   |   |   |
| 13  | controlledProcessCreator  | <a href="https://halide.net/europlp/HalideEuropePapers/EuroPLP2003/2003_FernandezEIAI_MoreB">https://halide.net/europlp/HalideEuropePapers/EuroPLP2003/2003_FernandezEIAI_MoreB</a>                   |   |   |
| 14  | directedSession           | <a href="http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf">http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf</a>                         |   |   |
| 15  | dynamicServiceManagement  | <a href="https://hylib.com/books/en/3.211.1.132/1/">https://hylib.com/books/en/3.211.1.132/1/</a>   |   |   |
| 16  | encryptedStorage          | <a href="http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf">http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf</a>                         |   |   |
| 17  | interceptingValidator     | <a href="https://arxiv.org/pdf/2007.03275.pdf">https://arxiv.org/pdf/2007.03275.pdf</a>   |   |   |
| 18  | messageInspector          | <a href="https://hylib.com/books/en/3.211.1.139/1/">https://hylib.com/books/en/3.211.1.139/1/</a>   |   |   |
| 19  | messageInterceptorGateway | <a href="https://hylib.com/books/en/3.211.1.139/1/">https://hylib.com/books/en/3.211.1.139/1/</a>   |   |   |
| 20  | networkAddressBlacklist   | <a href="http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf">http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf</a>                         |   |   |
| 21  | obfuscatedTransferObject  | <a href="https://hylib.com/books/en/3.211.1.132/1/">https://hylib.com/books/en/3.211.1.132/1/</a>   |   |   |
| 22  | passwordSynchronizer      | <a href="https://hylib.com/books/en/3.211.1.149/1/">https://hylib.com/books/en/3.211.1.149/1/</a>   |   |   |
| 23  | policy                    | <a href="https://www.researchgate.net/figure/Class-diagram-of-the-Policy-pattern_fig7_27383036">https://www.researchgate.net/figure/Class-diagram-of-the-Policy-pattern_fig7_27383036</a>             |   |   |
| 24  | policyDelegate            | <a href="https://hylib.com/books/en/3.211.1.132/1/">https://hylib.com/books/en/3.211.1.132/1/</a>   |   |   |
| 25  | secureMessageRouter       | <a href="https://hylib.com/books/en/3.211.1.139/1/">https://hylib.com/books/en/3.211.1.139/1/</a>   |   |   |
| 26  | secureServiceFacade       | <a href="https://hylib.com/books/en/3.211.1.132/1/">https://hylib.com/books/en/3.211.1.132/1/</a>   |   |   |
| 27  | secureSessionObject       | <a href="https://hylib.com/books/en/3.211.1.132/1/">https://hylib.com/books/en/3.211.1.132/1/</a>   |   |   |
| 28  | singleSignonDelegator     | <a href="https://hylib.com/books/en/3.211.1.142/1/">https://hylib.com/books/en/3.211.1.142/1/</a>   |   |   |
| 29  | trustedProxy              | <a href="http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf">http://www.cse.msu.edu/~cse870/Public/Homework/SS2005/HW5/rep-cse870-SS2005.pdf</a>                         |   |   |

- Using VisualParadigm I created [.xmi files](#) of the class diagrams I created for each SDP in my list. I used IntelliJ to run PatternScout by putting the .xmi files into the test

directory in the uml folder in the project using file Explorer. Running PatternScout created .rq files for each.xml file, meaning a SPARQL query was created for each SDP. I had some issues with naming conventions with PatternScout. Naming for PatternScout included restraints such as class names must be in FirstSecond and attributes and methods as firstSecond, with no spaces or special characters or else no query will be created and there will be an error. Many SDP's included the usage relationship which is not currently supported. I worked on improving the .rq creation throughout the quarter.

- The following table is from the [paper](#) by Jeffy Jahfar Poozhithara, Hazeline U. Asuncion, Brent Lagesse titled Towards Lightweight Detection of Design Patterns in Source Code, In the International Conference on Software Engineering and Knowledge Engineering (SEKE), to appear July 2022. PatternScout has only the listed items supported currently.

| OO Entities  | Contain Relationships | Visibility/Property | Class Relationships   | Stereotypes | Interactions       |
|--------------|-----------------------|---------------------|-----------------------|-------------|--------------------|
| Classes      | hasMethod             | Public              | Association           | Constructor | Method Invocations |
| Methods      | hasType               | Private             | Generalization        | Override    |                    |
| Constructors | hasReturnType         | Protected           | Aggregation           |             |                    |
| Fields       | hasModifiers          | Static              | Composition           |             |                    |
| Method       | hasField              | Final               | Interface Realization |             |                    |
| Parameters   | hasParameter          | Synchronized        | Dependency            |             |                    |
| Interfaces   | hasConstructor        | Abstract            |                       |             |                    |

**Table 1: Supported UML Concepts by PatternScout**

- Since the class diagrams were created based on diagrams found online, it was difficult to tell visibility or stereotypes such as if something overrides or is abstract or if something is private or public.

## SPARQL Query of Checkpointed System SDP

```

checkpointedSystem2.xmi.rq - Notepad
File Edit Format View Help
PREFIX woc: <http://rdf.webofcode.org/woc/>

SELECT DISTINCT ?Client4 ?AbstractCheckPointStrategy7 ?ConcreteCheckPointStrategy111 ?ConcreteCheckPointStrategy215 ?performCheck6 ?performCheck9 ?performCheck13
WHERE {
  ?Client4 a woc:Class .
  ?Client4 woc:hasModifier woc:Public .
  ?AbstractCheckPointStrategy7 a woc:Class .
  ?AbstractCheckPointStrategy7 woc:hasModifier woc:Public .
  ?ConcreteCheckPointStrategy111 a woc:Class .
  ?ConcreteCheckPointStrategy111 woc:hasModifier woc:Public .
  ?ConcreteCheckPointStrategy215 a woc:Class .
  ?ConcreteCheckPointStrategy215 woc:hasModifier woc:Public .
  ?performCheck6 a woc:Method .
  ?performCheck6 woc:hasModifier woc:Public .
  ?performCheck9 a woc:Method .
  ?performCheck9 woc:hasModifier woc:Public .
  ?performCheck13 a woc:Method .
  ?performCheck13 woc:hasModifier woc:Public .
  ?ConcreteCheckPointStrategy111 woc:extends ?AbstractCheckPointStrategy7 .
  ?ConcreteCheckPointStrategy215 woc:extends ?AbstractCheckPointStrategy7 .
  ?AbstractCheckPointStrategy7 woc:hasMethod ?performCheck6 .
  ?performCheck6 woc:hasReturnType ?bool119 .
  ?ConcreteCheckPointStrategy111 woc:hasMethod ?performCheck9 .
  ?performCheck9 woc:hasReturnType ?bool119 .
  ?ConcreteCheckPointStrategy215 woc:hasMethod ?performCheck13 .
  ?performCheck13 woc:hasReturnType ?bool119 .
}

```

- The next step was to search for open-source projects with at least one of the 27 SDP's present within its source code. This took an extremely long time for many reasons. One of the things that was difficult was searching through github using keywords for each pattern. This required scrolling through pages of results and opening projects and searching through documentation, code comments, class names, and method names to decide if there was potential of a pattern.

- My research involved testing for large open source projects of at least 1,000 lines of code. Finding large projects that were open source with security patterns turned out to be easier than finding small projects using SDP's because usually larger projects tend to want security in their software and usually provide services that can be more at risk for attacks.
- The project with the fewest lines of code was openejb2, which was a part of a Geronimo patch or plug in with well over 15,000 lines of code with the largest around 10 million lines of code. (Guess based upon .nt file size).
- The large .nt files increased the difficulty to get the .rq files to run on the large files.
- Compiling projects was very difficult and I ended up with the list of created .nt files because I was successfully able to compile them and successfully run them through Codeontology to create .nt files.
- To compile projects I usually had to resolve bugs within the code which would prevent it from compiling. This would mean changing POM files of the project to include or exclude certain files, editing code to ignore or not rely upon outdated or no longer supported libraries, removing or editing classes which had dependency issues, testing different JDK versions for the version the code used, or if issues were too many I would move onto compiling a different project.

| ID  | name                                     | url                        | language | category    | description  | url   | url   | url   | url   |
|-----|--|----------------------------|----------|-------------|--|---|---|---|---|
|     |  |                            |          |             |  |   |   |   |   |
| 112 | unicaise                                 | Controlled Process Creator | java     | no          | a CASE tool that supports modeling effects of a software engineering project, such as components and tasks             | <a href="https://github.com/unicaise/1-unicaise">https://github.com/unicaise/1-unicaise</a>                       | <a href="https://github.com/unicaise/2-unicaise">https://github.com/unicaise/2-unicaise</a>                       | <a href="https://github.com/unicaise/3-unicaise">https://github.com/unicaise/3-unicaise</a>                       | <a href="https://github.com/unicaise/4-unicaise">https://github.com/unicaise/4-unicaise</a>                       |
| 113 | OMC                                      | Controlled Object Factory  | java     | yes         | Representation of the workings of airport systems  | <a href="https://github.com/omc-project/OMC">https://github.com/omc-project/OMC</a>                               | <a href="https://github.com/omc-project/OMC">https://github.com/omc-project/OMC</a>                               | <a href="https://github.com/omc-project/OMC">https://github.com/omc-project/OMC</a>                               | <a href="https://github.com/omc-project/OMC">https://github.com/omc-project/OMC</a>                               |
| 114 | OpenAM                                   | Account Lockout            | java     | no          | access management solution   | <a href="https://github.com/openam/openam">https://github.com/openam/openam</a>                                   | <a href="https://github.com/openam/openam">https://github.com/openam/openam</a>                                   | <a href="https://github.com/openam/openam">https://github.com/openam/openam</a>                                   | <a href="https://github.com/openam/openam">https://github.com/openam/openam</a>                                   |
| 115 | Isoc2002 API - Secureite Asset Valuation |                            | java     | needs check | Java API for ISO 2002 - Securities business domain   | <a href="https://sourceinc.com/Isoc2002">https://sourceinc.com/Isoc2002</a>                                       | <a href="https://sourceinc.com/Isoc2002">https://sourceinc.com/Isoc2002</a>                                       | <a href="https://sourceinc.com/Isoc2002">https://sourceinc.com/Isoc2002</a>                                       | <a href="https://sourceinc.com/Isoc2002">https://sourceinc.com/Isoc2002</a>                                       |
| 116 | SAW4JPanel                               | Asset Valuation            | java     | yes         | Business management operation tool   | <a href="https://github.com/saw4j/saw4j">https://github.com/saw4j/saw4j</a>                                       | <a href="https://github.com/saw4j/saw4j">https://github.com/saw4j/saw4j</a>                                       | <a href="https://github.com/saw4j/saw4j">https://github.com/saw4j/saw4j</a>                                       | <a href="https://github.com/saw4j/saw4j">https://github.com/saw4j/saw4j</a>                                       |
| 117 | Ares                                     | Asset Valuation            | java     | needs check | Area product is a decentralized cross-chain oracle service product that implements verification on the data chain      | <a href="https://github.com/ares-project/ares">https://github.com/ares-project/ares</a>                           | <a href="https://github.com/ares-project/ares">https://github.com/ares-project/ares</a>                           | <a href="https://github.com/ares-project/ares">https://github.com/ares-project/ares</a>                           | <a href="https://github.com/ares-project/ares">https://github.com/ares-project/ares</a>                           |
| 118 | CDAP                                     | Authorization Enforcer     | java     | yes         | Repository for CDAP, CDAP Security Extensions, and Eask Hydrator Plugins   | <a href="https://github.com/cdap-project/cdap">https://github.com/cdap-project/cdap</a>                           | <a href="https://github.com/cdap-project/cdap">https://github.com/cdap-project/cdap</a>                           | <a href="https://github.com/cdap-project/cdap">https://github.com/cdap-project/cdap</a>                           | <a href="https://github.com/cdap-project/cdap">https://github.com/cdap-project/cdap</a>                           |
| 119 | Cougar                                   | Interpreting Validator     | java     | no          | Cougar is a framework for making   | <a href="https://github.com/bettler/cougar">https://github.com/bettler/cougar</a>                                 | <a href="https://github.com/bettler/cougar">https://github.com/bettler/cougar</a>                                 | <a href="https://github.com/bettler/cougar">https://github.com/bettler/cougar</a>                                 | <a href="https://github.com/bettler/cougar">https://github.com/bettler/cougar</a>                                 |
| 120 | Genosimo Devtools                        | Container Managed Security | java     | no          | Mirror of Apache Genosimo Devtools   | <a href="https://github.com/apache/genosimo-devtools">https://github.com/apache/genosimo-devtools</a>             | <a href="https://github.com/apache/genosimo-devtools">https://github.com/apache/genosimo-devtools</a>             | <a href="https://github.com/apache/genosimo-devtools">https://github.com/apache/genosimo-devtools</a>             | <a href="https://github.com/apache/genosimo-devtools">https://github.com/apache/genosimo-devtools</a>             |
| 121 | Tiny Container                           | Container Managed Security | java     | yes         | Tiny Container is a light weight Java EE compliant, modular container for solutions where size and code reuse matters. | <a href="https://github.com/tinycontainer/tinycontainer">https://github.com/tinycontainer/tinycontainer</a>       | <a href="https://github.com/tinycontainer/tinycontainer">https://github.com/tinycontainer/tinycontainer</a>       | <a href="https://github.com/tinycontainer/tinycontainer">https://github.com/tinycontainer/tinycontainer</a>       | <a href="https://github.com/tinycontainer/tinycontainer">https://github.com/tinycontainer/tinycontainer</a>       |
| 122 | customized-ee                            | Container Managed Security | java     | no          | Genosimo Eclipse Plugin  | <a href="https://github.com/apache/genosimo-eclipse-plugin">https://github.com/apache/genosimo-eclipse-plugin</a> | <a href="https://github.com/apache/genosimo-eclipse-plugin">https://github.com/apache/genosimo-eclipse-plugin</a> | <a href="https://github.com/apache/genosimo-eclipse-plugin">https://github.com/apache/genosimo-eclipse-plugin</a> | <a href="https://github.com/apache/genosimo-eclipse-plugin">https://github.com/apache/genosimo-eclipse-plugin</a> |

[illegible]

could be because the queries need more details due to the complexity of SDP's. There also seemed to be an issue with running queries on large .nt files because the time it takes to parse such large files. Before fixing .rq files would get errors like files of results would be empty or contain blank tables.

## MachineTime-Out Caused by Large NT File

```
"(" ...
"{" ...

C:\Users\actsdev\parser>sparql --query "C:\Users\actsdev\uml-to-sparql-converter\test\output\controlledObjectFactory.xmi
.rq" --data "C:\Users\actsdev\Downloads\OMC.nt" > controlledObjectFactoryOMC.txt
```

## Results and Discoveries

The current result from running .rq files on .nt files is 0% due to the current limitations with large .nt files and the lack of information on SDP class diagrams. Currently the detection algorithm is limited to small projects. To support future research, the following list is a guide to testing the SDP detection algorithm. Furthering testing using this algorithm on smaller projects and smaller .nt files will produce more conclusive results.

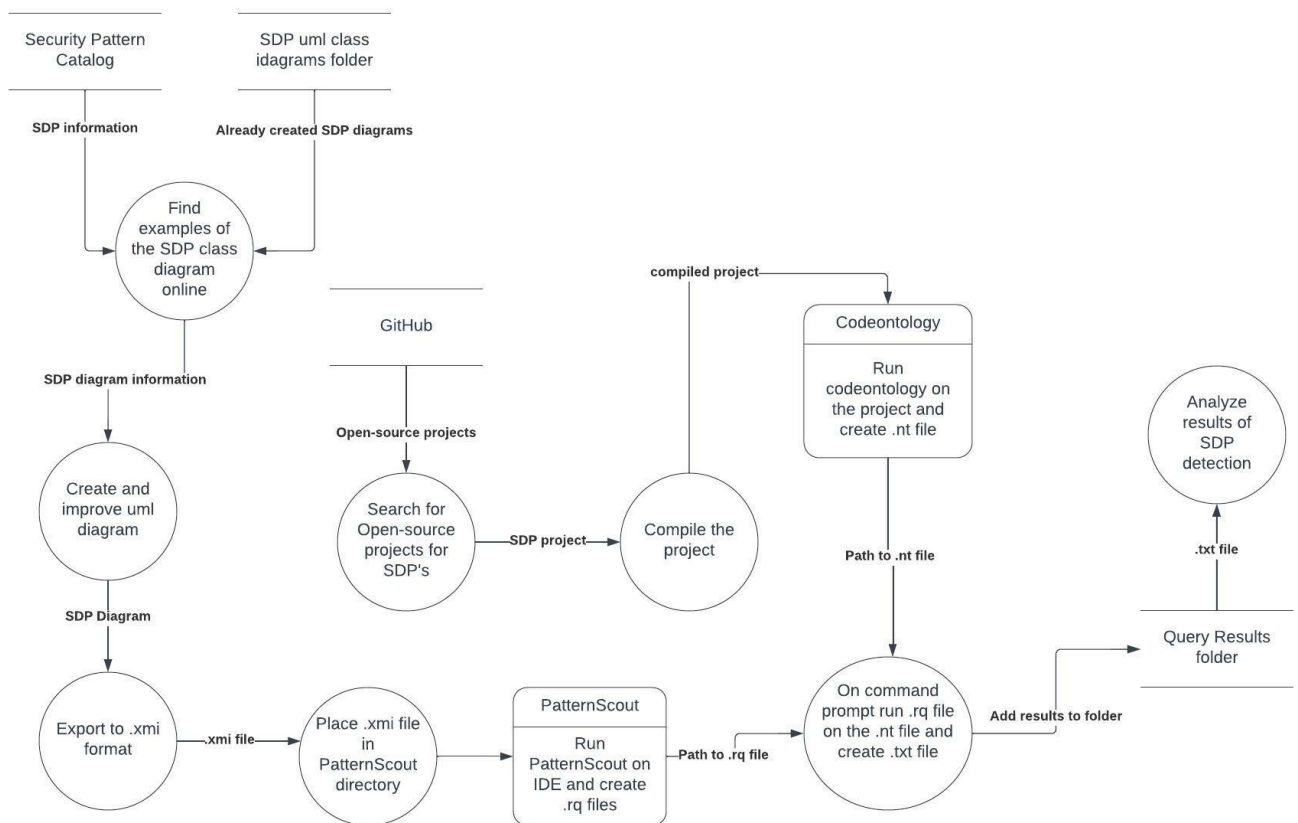
## Creating SDP Class Diagrams and Querying Open-source Projects

1. Find SDP class diagrams online by looking up the name of the security pattern and using known keywords of the SDP.
2. Create a class diagram in VisualParadigm of the SDP by using the diagram found online as a base and adding in missing elements. (relationships, connections, attributes, types, operations, visibility, stereotypes, ect)
  - a. Only use elements included in PatternScout based off the table.
  - b. Naming conventions for classes is FirstSecond, methods and attributes are firstSecond. Do not use spaces or special characters.
3. Create .xmi files from the class diagrams by exporting .xmi format in VisualParadigm. Simple to export to .xmi because of VisualParadigm included feature.
4. Run .xmi files through PatternScout for .rq files by going into the PatternScout directory in file explorer and going into the test folder and then placing the .xmi files in the uml folder. Run PatternScout using an IDE like intelliJ. The .rq SPARQL queries will be in the same test folder but under the output folder.
5. Find open source projects with manually detected SDP patterns following the guide by Kevin to manually detect security design patterns. Beware to find not too large files as this will cause problems later. Any .nt file that is more than 30,000 KB will be a problem.
6. Compile open-source projects. Found maven and ant projects have the best luck compiling. Check documentation for instructions to compile on the project, and fix bugs

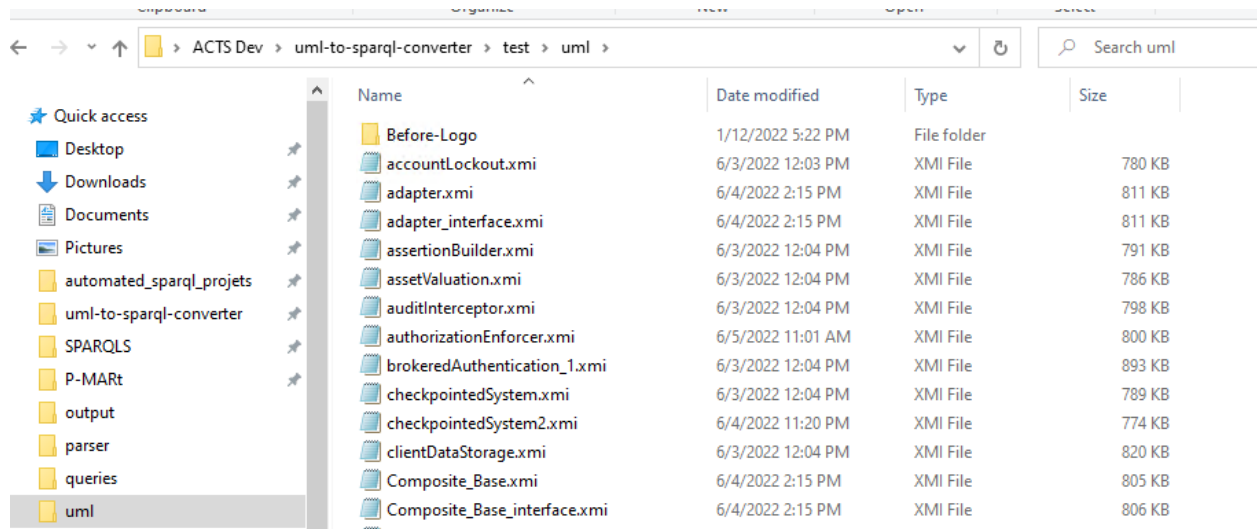
that prevent compilation. Maven projects have a pom file. Use OpenJDK 8 or 11 for most projects.

7. Use codeontology to create .nt files by going on the command line. More bugs may need to be fixed in the project before an .nt file can be created, even if the project compiles.
  - a. parser/**codeontology -i** <path to open source project> **-o** <name of project>.**nt -vf --dependencies**
  - b. parser/**codeontology --jar** <path to jar>.**jar -o** <name of project>.**nt -vf**
8. Run SPARQL queries on the .nt files on the command line. Issues with freezing if the .nt files are large.
  - a. **sparql --query** filepath of query **--data** file path of nt file > queryName.txt

## SDP Detection DFD

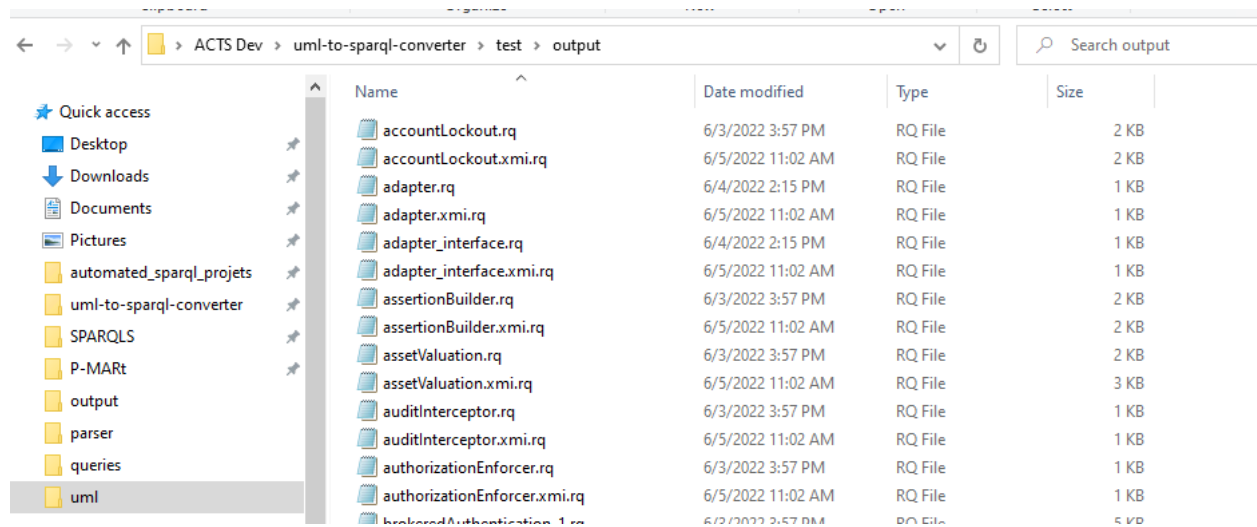


## PatternScout .xmi File Location



| Name                         | Date modified     | Type        | Size   |
|------------------------------|-------------------|-------------|--------|
| Before-Logo                  | 1/12/2022 5:22 PM | File folder |        |
| accountLockout.xmi           | 6/3/2022 12:03 PM | XMI File    | 780 KB |
| adapter.xmi                  | 6/4/2022 2:15 PM  | XMI File    | 811 KB |
| adapter_interface.xmi        | 6/4/2022 2:15 PM  | XMI File    | 811 KB |
| assertionBuilder.xmi         | 6/3/2022 12:04 PM | XMI File    | 791 KB |
| assetValuation.xmi           | 6/3/2022 12:04 PM | XMI File    | 786 KB |
| auditInterceptor.xmi         | 6/3/2022 12:04 PM | XMI File    | 798 KB |
| authorizationEnforcer.xmi    | 6/5/2022 11:01 AM | XMI File    | 800 KB |
| brokeredAuthentication_1.xmi | 6/3/2022 12:04 PM | XMI File    | 893 KB |
| checkpointedSystem.xmi       | 6/3/2022 12:04 PM | XMI File    | 789 KB |
| checkpointedSystem2.xmi      | 6/4/2022 11:20 PM | XMI File    | 774 KB |
| clientDataStorage.xmi        | 6/3/2022 12:04 PM | XMI File    | 820 KB |
| Composite_Base.xmi           | 6/4/2022 2:15 PM  | XMI File    | 805 KB |
| Composite_Base_interface.xmi | 6/4/2022 2:15 PM  | XMI File    | 806 KB |

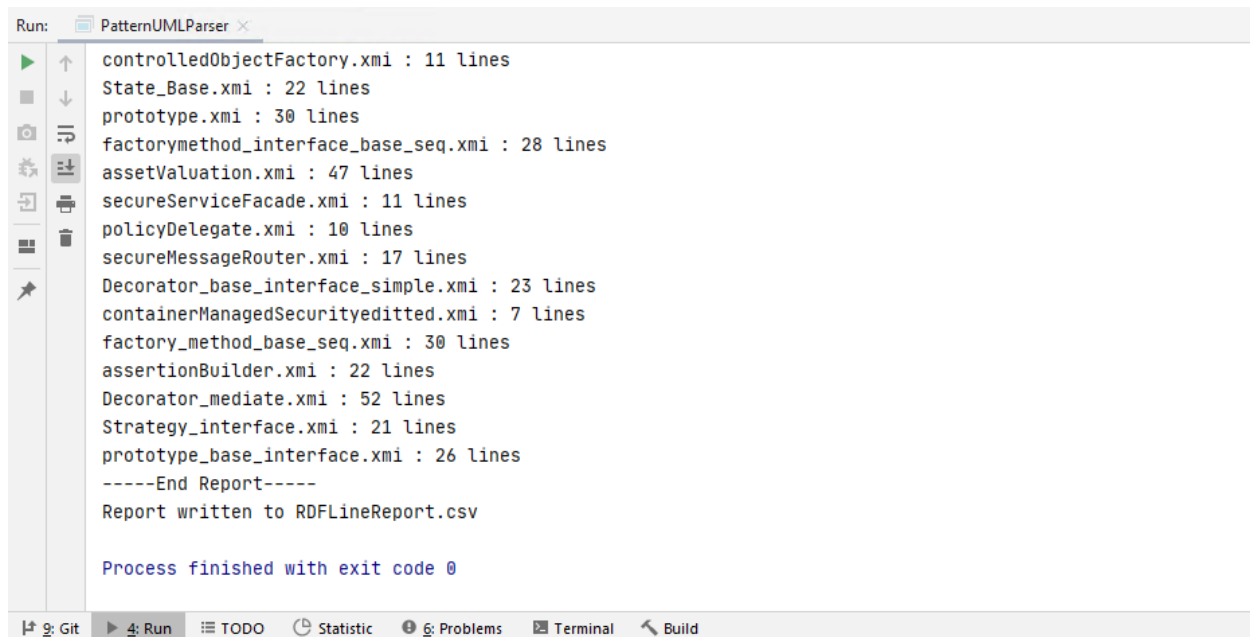
## PatternScout .rq File Output Location



| Name                         | Date modified     | Type    | Size |
|------------------------------|-------------------|---------|------|
| accountLockout.rq            | 6/3/2022 3:57 PM  | RQ File | 2 KB |
| accountLockout.xmi.rq        | 6/5/2022 11:02 AM | RQ File | 2 KB |
| adapter.rq                   | 6/4/2022 2:15 PM  | RQ File | 1 KB |
| adapter.xmi.rq               | 6/5/2022 11:02 AM | RQ File | 1 KB |
| adapter_interface.rq         | 6/4/2022 2:15 PM  | RQ File | 1 KB |
| adapter_interface.xmi.rq     | 6/5/2022 11:02 AM | RQ File | 1 KB |
| assertionBuilder.rq          | 6/3/2022 3:57 PM  | RQ File | 2 KB |
| assertionBuilder.xmi.rq      | 6/5/2022 11:02 AM | RQ File | 2 KB |
| assetValuation.rq            | 6/3/2022 3:57 PM  | RQ File | 2 KB |
| assetValuation.xmi.rq        | 6/5/2022 11:02 AM | RQ File | 3 KB |
| auditInterceptor.rq          | 6/3/2022 3:57 PM  | RQ File | 1 KB |
| auditInterceptor.xmi.rq      | 6/5/2022 11:02 AM | RQ File | 1 KB |
| authorizationEnforcer.rq     | 6/3/2022 3:57 PM  | RQ File | 1 KB |
| authorizationEnforcer.xmi.rq | 6/5/2022 11:02 AM | RQ File | 1 KB |
| brokeredAuthentication_1.rq  | 6/3/2022 3:57 PM  | RQ File | 5 KB |



## Example of Successful Execution of PatternScout



```
Run: PatternUMLParser x
controlledObjectFactory.xmi : 11 lines
State_Base.xmi : 22 lines
prototype.xmi : 30 lines
factorymethod_interface_base_seq.xmi : 28 lines
assetValuation.xmi : 47 lines
secureServiceFacade.xmi : 11 lines
policyDelegate.xmi : 10 lines
secureMessageRouter.xmi : 17 lines
Decorator_base_interface_simple.xmi : 23 lines
containerManagedSecurityeditted.xmi : 7 lines
factory_method_base_seq.xmi : 30 lines
assertionBuilder.xmi : 22 lines
Decorator_mediate.xmi : 52 lines
Strategy_interface.xmi : 21 lines
prototype_base_interface.xmi : 26 lines
-----End Report-----
Report written to RDFLineReport.csv

Process finished with exit code 0
```

## Discoveries

Several thoughts to help in future research:

SDP class diagram creation is a top priority to improve documentation and expand upon each of the 27 diagrams I created. These diagrams will assist future students in improving upon the class diagrams to improve the reliability of the lightweight detection of SDP's in source code algorithm. Large open-source projects create large .nt files which causes results to be inconclusive because the queries take too long to resolve before the machine shuts off.

- Discovered Codeontology has difficulties with certain projects even with successful compilation. Found compiled maven and ant projects worked best with codeontology.
- Most open-source projects require OpenJDK 8 or 11 in order to compile or work with Codeontology.
- Open-source code rarely includes sufficient documentation of items or instructions for use. Commenting within source-code is also rare, with many projects having no commenting at all.
- Many open-source projects were not able to be compiled due to many [reasons](#). Some reasons include outdated library use, no longer supported operations, bugs within the code, use of private libraries like ForgeRock, vague compilation errors, wrong JDK use, and so on.

## Challenges with .nt Files

- Compiling open source projects was very difficult and required learning the workings of each project I attempted to use.

- Tested compiling gradle projects, maven projects, and ant projects.
  - Gradle projects had many issues compiling and took great amounts of time to debug.
  - Maven projects had many issues with compiling but was able to narrow the types of JDK's most open source projects used with maven which was specifically OpenJDK 8 and OpenJDK 11. Oracle JDK's did not usually work.
  - These difficulties improved upon my skills like debugging on IntelliJ and searching online for different errors and potential solutions to make code compile.
- Many open-source projects are poorly documented. Many included brief statements of what the project was with no detail for installing, running, or compiling. Some projects included loose compiling instructions that still required further research.
- Many projects I was able to get to compile and build did not work with Codeontology. Errors with codeontology were hard to decipher and not very specific. The following examples are common Codeontology errors I came across. Sometimes the OpenJDK 8 or 11 would solve these errors but sometimes it would not.

### Example 1: Unresolved Codeontology Error

```
C:\Users\actsdev\parser>java -cp target/codeontology-1.0-SNAPSHOT-jar-with-dependencies.jar -Xms2G -Xmx4G org.codeontology.CodeOntology -i "C:\Users\actsdev\Downloads\agate-master" -o agate-master.nt -vf --dependencies
Running on C:\Users\actsdev\Downloads\agate-master
Module: C:\Users\actsdev\Downloads\agate-master\agate-web-model
Module: C:\Users\actsdev\Downloads\agate-master\agate-core
Module: C:\Users\actsdev\Downloads\agate-master\agate-rest
Module: C:\Users\actsdev\Downloads\agate-master\agate-webapp
Module: C:\Users\actsdev\Downloads\agate-master\agate-dist
Loading dependencies with Maven
Downloading dependencies...
Preparing module C:\Users\actsdev\Downloads\agate-master\agate-core
Preparing module C:\Users\actsdev\Downloads\agate-master\agate-webapp
Preparing module C:\Users\actsdev\Downloads\agate-master\agate-rest
Preparing module C:\Users\actsdev\Downloads\agate-master\agate-dist
Preparing module C:\Users\actsdev\Downloads\agate-master\agate-web-model
It was a good plan that went awry.
class jdk.internal.loader.ClassLoaders$AppClassLoader cannot be cast to class java.net.URLClassLoader (jdk.internal.loader.ClassLoaders$AppClassLoader and java.net.URLClassLoader are in module java.base of loader 'bootstrap')
[WARNING] Could not access file C:\Users\actsdev\Downloads\agate-master\agate-web-model\target\dependency\jackson-jaxrs-json-provider-2.8.11.jar
Analyzing file C:\Users\actsdev\Downloads\agate-master\agate-web-model\target\dependency\jackson-jaxrs-json-provider-2.8.11.jar
Running on C:\Users\actsdev\Downloads\agate-master\agate-web-model\target\dependency\jackson-jaxrs-json-provider-2.8.11.jar
Triples extracted successfully.
Jar files processed successfully in 19 ms.
```

- 
- **Example 2: Codeontology Error Resolved Using OpenJDK 8**

```
C:\Users\actsdev\parser>codeontology -i "C:\Users\actsdev\Downloads\vulnerable-sso-master" -o vulnerable-sso-master.nt -vf
C:\Users\actsdev\parser>set CODEONTOLOGY_LINE_ARGS=-i "C:\Users\actsdev\Downloads\vulnerable-sso-master" -o vulnerable-sso-master.nt -vf
C:\Users\actsdev\parser>java -cp target/codeontology-1.0-SNAPSHOT-jar-with-dependencies.jar -Xms2G -Xmx4G org.codeontology.CodeOntology -i "C:\Users\actsdev\Downloads\vulnerable-sso-master" -o vulnerable-sso-master.nt -vf
Running on C:\Users\actsdev\Downloads\vulnerable-sso-master
It was a good plan that went awry.
class jdk.internal.loader.ClassLoaders$AppClassLoader cannot be cast to class java.net.URLClassLoader (jdk.internal.loader.ClassLoaders$AppClassLoader and java.net.URLClassLoader are in module java.base of loader 'bootstrap')
C:\Users\actsdev\parser>
```

- **Example 3: Unresolved Codeontology Error**

```

Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-json
Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-zookeeper
Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-logfeeder-plugin-api
Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-logfeeder-container-registry
Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-local
Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-solr
Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-docs
Module: C:\Users\actsdev\Downloads\ambari\ambari-logsearch-web
Loading dependencies with Maven
Downloading dependencies...
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-zookeeper
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-logfeeder-plugin-api
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-logfeeder
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-logfeeder-container-registry
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-log4j2-appender
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-assembly
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-it
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-solr
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-docs
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-web
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-server
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-json
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-appender
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-local
Preparing module C:\Users\actsdev\Downloads\ambari\ambari-logsearch-config-api
It was a good plan that went awry.
class jdk.internal.loader.ClassLoaders$AppClassLoader cannot be cast to class java.net.URLClassLoader (jdk.internal.loader.ClassLoaders$Ap
pClassLoader and java.net.URLClassLoader are in module java.base of loader 'bootstrap')

```

- Discovered that sometimes using the right JDK fixed the issue, like using OpenJDK 8 or 11, but often the errors were more hard to decipher. When this happened I found it took less time to move onto trying to work on another project.
- Further processing of open source projects will require a solution to the issue of running .rq files on .nt files. This is a potential scaling issue that may have a solution in pre-processing .nt files to remove unnecessary RDF triples and focus on potentially more useful RDF triples.

### Open-source Projects with Manually Detected Security Design Patterns

| Project Name             | Description   | Manually Detected Pattern  | Codeontology Created .nt File Size | Lines of Code in .nt File (about 5 lines per 1 KB) |
|--------------------------|---|----------------------------|------------------------------------|--|
| Minecraft Autosave World | Minecraft Bukkit Plugin for creating Autosave in MC servers                                       | Checkpointed System        | 743,807 KB                         | 3,719,035  |
| LDAP SDK for Java        | Communication services with LDAP directory servers  | Policy Synchronizer        | 373,342 KB                         | 1,866,710  |
| openfire                 | RTC server used for instant messaging   | Directed Session           | 1,217,350 KB                       | 6,086,750  |
| Apache Ambari Log Search | Log aggregation, analysis, and visualization for Ambari managed (or any other) services           | Trusted Proxy              | 21,647 KB                          | 108,235  |
| Azure SDK for Java (RT)  | Repository for active development of the Azure SDK for Java                                       | Encrypted Storage          | 583,752 KB                         | 2,918,760  |
| openejb2                 | Geronimo patch or plug in   | Container Managed Security | 3,055 KB                           | 15,275   |
| Tiny Container           | light weight Java EE compliant, modular container for solutions where size and code reuse matters | Container Managed Security | 1,706,255 KB                       | 8,531,275  |

|        |  |                           |              |            |
|--------|--|---------------------------|--------------|------------|
| CDAP   | Repository for CDAP, CDAP Security Extensions, and Cask Hydrator Plugins | Authorization Enforcer    | 32,385 KB    | 161,925    |
| SSAWFA | Business management operation tool                                       | Asset Valuation           | 807,874 KB   | 4,039,370  |
| OMC    | Representation of airport systems  | Controlled Object Factory | 2,009,153 KB | 10,045,765 |

## Future Work and Improvements

The Ultimate difficulty came when trying to run SPARQL queries on the .nt files created by condeontology. From the errors created, it appears that running queries on .nt files created from large open source projects requires more resources than my private machine or a VM has. Scaling to potentially limit .nt files into essential parts only may help, or improving how queries are run on the .nt files may fix this issue.

There are also very few available publications on security design patterns and their use in diagrams, which made finding specifics on example class diagrams very difficult. Finding enough information to fill in the gaps in knowledge for how the diagrams should look was difficult. Some patterns had less information available than others. SDP's found online are often not descriptive or contain inaccuracies. The next steps to address this issue would be to dedicate research on creating detailed uml class diagrams for each security design pattern listed. The security design patterns I created based on the information available online will act as a starting point for future students. Variations will need to be created for many SDP's to adjust for the limitations set by class diagrams in general due to some concepts within SDP's being hard to depict through classes and relationships.

Future research needs to be done on scaling the process of running SPARQL queries on .nt files. Currently there is a major error with timing out the machine due to freezing from long execution times. Large open source projects are more difficult to check for SDP's since the queries take too long. This requires a scaling solution to accommodate for larger projects which would allow for the use of the algorithm for potentially more critical systems.

PatternScout is also a developing software tool with current limitations discovered in the process of testing with no current support for the usage relationship, realization relationship, dependency relationship and access relationship of class diagrams. PatternScout currently has limited support of relations and stereotypes and connection labeling. Currently this form of detection requires further research to improve the scalability of the detection algorithm. In the current state, the process is unreliable and inconclusive for larger .nt files and SDP class diagrams. SDP's involve concepts that are difficult to translate into a general class diagram as some security design

patterns involve concepts that do not translate to class diagrams and are therefore much more difficult to automatically detect. Further research into expanding the SDP class diagrams is needed to include more details about attributes, methods, visibility, connections, and so on.

## Reflection

SDP's are essential tools for cybersecurity environments. Having the ability to detect different types of security patterns that are already implemented in code such as legacy code, would make securing legacy code and comparing security requirements much easier and faster, using less resources. Conducting research for this task is important and I have learned new skills because of it. I was challenged with having problems that weren't documented online so I had to use reasoning skills and experimenting skills in order to complete tasks. Because the process is new I had to use trial and error to come up with ways to accomplish things to get tools to work. I became better at learning new languages as I had to learn SPARQL and I learned how to use technologies as a part of a larger algorithm. I had many challenges which I found myself to be extremely stuck on and required collaboration skills and learning to know how to ask the right questions. I learned how to document the process of testing and research. Because of the frustrations and setbacks I experienced I had to build efficient time management skills to complete tasks. The research and testing I did was to assist students with their research and continue upon the research conducted before me. The results from the research allows for students to save time and analyze the data.