During the class we have discussed the importance of architectural recovery techniques. Such techniques facilitate the process of understanding the underlying architectures of systems, which could guide development, maintenance, and changes in the system. In your third homework, you applied two recovery techniques to Tomcat and recovered Tomcat’s architectures in the form of clusters. However, you might have noticed that these recovery techniques are not very suitable to analyze the security of a system from an architecture perspective. The factors that contribute to this may be numerous: Some techniques, such as ACDC, rely on the structural breakdown of the architecture, and that does not reveal security architectural decisions which usually span more than one structural component (think authentication and authorization or input validation components); other techniques like ARC should be able to capture some security concerns, but they are focused on document-level concerns, whereas security decisions might be implemented at lower granularities of the implementation. It is, however, possible to implement changes that address these shortcomings in the existing recovery techniques to create their ‘security-aware’ versions. This is going to be your main task in the course project. For the project, you should pick at least one security-related architectural decision and extend one existing architecture recovery technique of your choice to properly account for that decision.

Tasks

1. Pick at least one security-related architecture decision. Analyze its implementation by looking into existing subject systems that have it as part of their architecture and implement it as well. The subject systems you choose to analyze are your choice. Understand the problem scope, i.e., how this decision can be recovered by an architecture recovery tool.

2. Pick an existing architecture recovery technique. Identify what prohibits it from recovering the security decision in #1, e.g., the technique clusters structurally, but the security decision spans structural components. NOTE: Although this is highly unlikely, it is possible that the architecture recovery technique you have chosen happens to recover some security decisions. In that case, you must either pick security decisions that are currently not supported, or a different recovery technique, or both.
3. Modify and extend the recovery technique to account for the security decision you picked in #1. Indicate clearly all your modifications from the original code base.

4. Provide a working prototype. We expect from the prototype to be able to:
   a. Identify the architectural security decision(s) being implemented in the subject system, e.g., by creating a cluster for it. You may select the subject systems on which you evaluate and demonstrate your tool.
      i. If your modification works for more than one architectural decision, then the prototype should be able to name the type of the decision as well.
   b. Identify all the system elements involved in that security decision. Note that depending on the architectural decision, an element could be a lower level of implementation than just a file.

You should treat the above as the minimal requirements. The projects will be graded on a curve and teams that achieve more will receive higher scores. Supporting the ability to identify more than one security decision, interesting visualizations of your results, extension and integration of multiple recovery techniques, etc. are all possible extensions to the above, “base” requirements.

You will be demoing your working prototypes after the submission deadline. You are expected to show why the recovery technique you selected was initially not able to recover the decision(s) from #1 and how modifications you have made to the technique help in recovering the particular decision.

**Selecting Groups**

The project’s scope is intended for groups of three students; groups of two students are allowed. You should spend the allotted time in class on Monday, 11/11 and Wednesday, 11/13 to reach out to other students and work on forming your groups. In exceptional situations, we may allow a single student to work on their own; please be sure to email us for permission ASAP and explain your reason. Each group should send us an email before noon on Friday, 11/15 with the names of the group’s members. Only one member per group should send the email.
Appendix

ACDC and ARC are well-known architecture recovery techniques. An implementation of them has been included in a larger tool, Architecture Recovery, Change, And Decay Evaluator (ARCADE). The tool has a detailed manual which you may consult.

1. [https://softarch.usc.edu/~lemduc/Recovered_files/ArchitectureEvolutionAnalysiswithARCADE.pdf](https://softarch.usc.edu/~lemduc/Recovered_files/ArchitectureEvolutionAnalysiswithARCADE.pdf)

The manual also consists of additional resources, such as the published papers that explain ACDC and ARC, which you may consult.