**Final Group Project 550**

**Due 4/28/25**

**Elizabeth Heavirland, Ashmin Bhandari, Yash Patel, Ruchi Bhusal**

Here are some project ideas that could be beneficial for learners:

1. Crack Simple Software: Start by reverse-engineering simple, open-source Python applications to understand their flow and logic. It can teach you the foundation of reading code and identifying key functions.
2. Develop a Fuzzer: Build a tool in Python that tests other programs for vulnerabilities by inputting large amounts of random data or "fuzzing." It can give you insights into how software responds to unexpected or malicious input.
3. Disassemble Python Bytecode: Use the dis module in Python to disassemble and inspect the bytecode of Python programs. It can deepen your understanding of how Python code is executed.
   1. We did a disassembler in class on one of the other projects that we could expand on if we would like
4. Analyze Malware Samples: With appropriate precautions, analyze benign malware samples to understand their structure and behavior. Note that this should only be done in a controlled, secure environment to avoid potential harm.
   1. We could use the already existing work on malp1 and malp2?
5. Reverse Engineer Network Protocols: Create your own client to interact with open-source servers by reverse-engineering their network protocols. It teaches you about network communication and protocol analysis.
6. Build a Debugger: Construct a simple debugger with Python that can attach to running processes, halt execution, and inspect memory and registers. It's a challenging project that gives you a peek into the world of software at run time.
   1. I just personally think this would be interesting to do
7. Automate Reverse Engineering Tasks: Write Python scripts to automate common reverse engineering tasks, such as extracting strings from binaries, scanning for known vulnerabilities, or automating static code analysis.

I think we can format this paper like the following

1. Introduction
   1. Purpose of paper
   2. Objective of project
   3. Any background or history on subject
2. Code/tools
   1. Describe code/malware/pseudocode used in testing project
   2. Describe any and all tools used in testing project
   3. Describe known alternative tools
3. Testing
   1. Test the tool
   2. Describe the process of the tests including all steps taken and why
4. findings/results
   1. Show its results
   2. Describe if they are aligned with what we thought would happen why/why not
      1. Or describe how the results achieve the purpose of the paper
5. Challenges faced/future work
   1. Describe what issues we ran into
   2. Describe how this tool could be used going forward
   3. Describe any future works (if applicable) that would benifit this tool
6. Conclusion
   1. Summarize purpose of paper and findings of the tool
   2. Describe key takeaways and what was learned and how it could apply to other software reverse engineering efforts.
7. References
   1. If applicable add any references

Also, please check this link:

[Reverse Engineering - Software Engineering - GeeksforGeeks](https://www.geeksforgeeks.org/software-engineering-reverse-engineering/)

Be advised to email me if you have suggested your idea. Alternatively, you can come up with your own project.

**Note: All students must participate in the presentation.**

**Each group needs to submit a presentation for your project. The presentation should contain the following points: (I think the following should be in a slides format)**

1. **Project title**
2. **Introduction**
3. **Project parts [input [i.e. data], processing (Technique or your proposed method], output [screenshots for the obtaining results]**
4. **Project input, process, output by listing all the used TOOLS**
5. **Live demo for the execution/ implementation, record video previously if the demo requires longer time than 15 minutes.**