DearleyPY- Augmentation of Python Decompilation
Jonson School Undergraduate Research Scholar Awards

David Wank, The University of Texas at Dallas

Background

- Python compiles to bytecode and is stored in a hierarchy
- Python bytecode may execute without the source code

Proposed Solution

DearleyPY augments existing decompilers, enabling end-users with little knowledge of decompilation details to produce patch rules in an end-to-end process

DearleyPY uses ML to automate significant portions of the analysis process, reducing workload on the end-user

Problem Statement

- Malicious actors may distribute malware as Python bytecode in a way that thwarts existing analysis tools which decompile code
- Can we improve the analysis tools to handle edge cases?

Results

- DearleyPY has a suite of built-in rules that can manipulate Python bytecode in various ways to resolve decompilation errors
- Users add patches similar to these to resolve errors they encounter

Conclusion

- We conducted the first-ever large-scale decompilation failure study, analyzing 1,635,411 cases in total and discovering seven major failure categories.
- We created DearleyPY, a suite of tooling for root cause analysis, binary patching, reduction of failures and more.