---------------------------------------------------------------------------------------

Title: Human-Centric Verification for Software Safety and Security
----------------------------------------------------------------------------------------

Abstract: Software is a critical part of our lives today. Verifying software to avoid violations of safety and security properties is a necessary task. It is also imperative to have an assurance that the verification process was correct. We propose a human-centric approach to software verification. This involves enabling human-machine collaboration to detect vulnerabilities and to prove the correctness of the verification.
We discuss two classes of vulnerabilities. The first class is Algorithmic Complexity (AC) vulnerabilities. AC vulnerabilities are a class of software security vulnerabilities that cause denial-of-service attacks. The description of AC vulnerabilities is not known apriori. The problem is equivalent to searching for a needle in the haystack when we don't know what the needle looks like. We present a novel approach to detect AC vulnerabilities in web applications. We present a case study audit from DARPA's Space/Time Analysis for Cybersecurity (STAC) program to illustrate our approach.
The second class of vulnerabilities is Memory Leaks. Although the description of the Memory Leak (ML) problem is known, a proof of the correctness of the verification is needed to establish trust in the results. We present an approach inspired by the works of Alan Perlis to compute evidence of the verification which can be scrutinized by a human to prove the correctness of the verification. We present a novel abstraction, Evidence Graph, that succinctly captures the verification evidence and show how to compute the evidence. We evaluate our approach against ML instances in the Linux kernel and report improvement over the state-of-the-art results. We also present two case studies to illustrate how the Evidence Graph can be used to prove the correctness of the verification.