CS 4800: Undergraduate Research

This course provides undergraduate students with opportunities to engage in research opportunities.

Faculty Advisor: Ren Quinn

If you are interested in taking this course, please discuss with a faculty advisor and fill out this form.

Past Projects

Towards Faster Speculative Execution Attacks

Damen Maughan – UCUR 2022, Poster

Spectre [1] is a CPU-level exploit that takes advantage of speculative execution to read secret data within the program space of a victim program. The process iterates through the victim program space, very slowly reading a single byte at a time. We propose a method for improving the speed by which Spectre can read secret data. Instead of reading an entire byte at a time, requiring 255 cache misses per byte, we split it up into 8 separate bits, requiring up to 8 cache misses per byte. By showing a faster method by which Spectre can read data, we expand the scope of possible ways to utilize this attack and thus reaffirm its seriousness.

Sound Wave Extraction from Background Accelerometer Readings

David Gary, Erick Gutierrez – UCUR 2022, Poster

Three-axis accelerometers, gyroscopes, and magnetometers are a necessary component for modern mobile devices, despite the fact that their data is not protected in the same way recordings from a microphone or camera would be. Studies have shown that this data can be exploited to reveal movement patterns and even matched to previously identified sound wave data for hotword detection. In this work, we attempt to expand what can be produced by using transform methods to extract full sound wave replications without previous identification or machine learning classification systems. Since there are few protections on accelerometer data in place, this study displays a severe vulnerability modern devices have to a new type of side-channel attack via the accelerometer.

ACCess Granted: Inferring Mobile Device Keystrokes Using Background Accelerometer Data.

Erick Gutierrez, David Gary – UCUR 2022, Poster

Accelerometers have become a common component in the mobile devices we use every day, and just like most data, many of the applications we use are able to access it. However, unlike the microphone and camera, there are no permission protocols protecting accelerometer data.. In this work, we collect accelerometer data on individual key presses when using a virtual on-screen keyboard. We then use that data to predict which key is pressed without acquiring the necessary security permissions to track keyboard inputs in the background. We explore how this information can be used to gather passwords and other sensitive information and emphasize the importance of required security permissions for accelerometer access.

A Web-Based Development Environment for Beginning Python Programmers

Duy Huynh, Ren Quinn – UT Research Symposium 2022, Slides

To abide by the philosophy of “active learning, active life”, the Department of Computing uses a tool called CodeGrinder to help students learn computer programming with interactive exercises. To facilitate the use of CodeGrinder by beginning programming students, we have integrated it with common development software called Integrated Development Environments (IDEs) that combine the necessary tools a programmer needs for developing software into a single interface. However, these software tools only work for specific platforms (e.g., Windows, MacOS) and are not compatible with commodity platforms that have limited functionality (e.g., ChromeBooks, tablets). Therefore, to further expand our reach, this project focuses on producing a web-based IDE that would provide students with a portable learning environment that would be accessible from any modern computer platform.

WebAssembly (Wasm) is an open toolkit that will allow us to overcome these accessibility limitations by simulating the platform-specific functionality required by traditional IDEs. In essence, it enables us to run the IDEs that students currently use but in a web browser with minimal modifications. However, Wasm lacks

support for the traditional graphical interface preferred by many beginning programmers. In this presentation, we discuss our short-term, practical approach to this problem: integrating a Wasm-based programming platform with a common web-based interface to create a traditional IDE experience for students to use on virtually any computing device. We also address our progress on a long-term fundamental solution to the challenges of providing native graphical interfaces in a web browser.