Nikki Fayra

Email: nikki@fayra.com
Personal website with code samples: https://nikkifayra.com
LinkedIn: https://www.linkedin.com/in/nikkifayra

EDUCATION

Arizona State University August 2021

- Master of Computer Science *GPA: 3.96/4.00*
- Computer Security Concentration
- Topics studied: network security, binary security, reverse engineering, web application security, attacker tactics, operating system security, cryptography, artificial intelligence, machine learning, big data, data mining, SQL and NoSQL databases, distributed systems, operating systems

University of Illinois at Urbana-Champaign May 2016

- Bachelor of Science in Computer Science *GPA: 3.71/4.00*
- College of Engineering Dean's List
- Graduated with Honors

SKILLS / STRENGTHS

Technical Skills:

- **Languages**: C#, C, C++, JavaScript, TypeScript, Python, Java, Ruby, ASP.NET, JSX, HTML, CSS, SASS, SQL, NoSQL, PowerShell, Bash, InterSystems Caché, Scala, Haskell, PHP, Verilog, LaTeX, R, Wolfram Language, Visual Basic
- **Software and technologies:** .NET Framework, Node.js, React, Git, Subversion, jQuery, Bootstrap, Python Flask, Ruby Sinatra, SQL Server Reporting Services (SSRS), AWS (Amazon Web Services), Machine Learning, Microsoft Visual Studio, Jetbrains IDEs, Xcode, other IDEs, IIS, Docker, Jenkins, JIRA, Mathematica, Unity Game Engine, MIPS, Oculus Rift development, Android Development
- **UNIX/Linux & Windows Systems / Web Server Administration**

Professional strengths:

- **Self-motivated, fast learner, creative thinker, eagerness to learn new technologies, effective time manager**

EXPERIENCE

LPi Remote
Software Developer August 2021 – November 2022

- Full stack web development with Node.js, React, SQL, and AWS serverless Lambda functions invoked via a REST API. Programming languages used were TypeScript, JavaScript, and Python. Worked on a public-facing web application which contains a searchable directory of LPi's customers and sponsors, allows customers and sponsors to customize their pages, and lists sponsors for each customer.
- Developed an API to upload and retrieve customer photos along with a corresponding web interface. Photos are stored in AWS S3. Customer and sponsor page owners can upload photos to their pages via the web application. Photos are displayed on customer and sponsor pages.
- Developed functionality to synchronize documents in AWS CloudSearch with customer and sponsor data stored in a SQL database. When updates to the SQL database are made via an AWS SQS queue and several API routes, CloudSearch documents are synchronized with relevant updates. Developed an API to search for customers or sponsors via CloudSearch which is used by a search field on the web application. Search results are displayed with links to the customer and sponsor pages.
- Developed an API to set and retrieve weekly content displayed on customer pages. Developed an Admin UI to set the content which includes a searchable table to find content by date and slug. Content is parsed to render dynamic external links.
- APIs were integrated with the Auth0 identity platform for authentication.
- Developed functionality to receive mailing list subscriber information from Adobe Marketo's REST API.
- Updated UI elements throughout the web application to use Material UI library components for consistent styling.
- Web UIs were developed with mobile responsiveness.
- Configured cloud infrastructure in AWS.

Epic Systems Corporation Verona, WI
Software Developer July 2016 – April 2019

- Primarily worked on Epic's internal version control, build utilities, bug tracking, and project management software as part of large-scale web application used by all developers and quality assurance staff at Epic.
- Managed a team including two others for one year focusing on version control and build utilities for company-wide front-end, back-end, and database code. Was responsible for prioritizing tasks, reviewing project estimates, and evaluating my team members while maintaining a full load of my own development work. Supported 2000+ developers on all teams throughout the company with usage of our internal tools.
- Developed a C# ASP.NET web application to track the migration of Epic's entire codebase to a modern architecture. The application allowed developers to indicate which files in version control are migrated and obsolete during the code-merge process. The tool also allows reporting for management purposes and saves hundreds of hours of manual work each year tracking the progress.
- Developed build utilities for Epic's propitiatory C# ASP.NET web application framework. The suite includes a set of servers which automatically build the codebase when there are new revisions. The ability to retrieve the latest build was added to the process of source code checkout. The servers also make commits to source control to update transpiled source code (TypeScript compiled into JavaScript, and SASS compiled into CSS). A tool was also included to allow developers to manually build a minimal set of projects, automatically calculating and building the dependency tree. The utilities save thousands of hours per year.
- Maintained and added new features to Epic's continuous integration system which builds and tests new changes before they are committed to version control. Added an automated C# unit test runner which runs and reports results of unit tests before source code is committed. Assisted in the transition of the continuous integration system to a distributed server network.
- Developed a framework to track user activity and system performance across the majority of our internal version control and web applications. The data is used to help troubleshoot issues and to identify areas for improvement.
- Developed, maintained, and assisted in design/code review for additional version control features including transpiling TypeScript to InterSystems Caché (Epic's primary database language) during the merge process, and for other internal C# ASP.NET web applications including source code review, diffing, merge requests, bug/issue tracking, project management, internal/customer exams, time logging, and tracking for customer billing/trips.
- Developed support to use a customer-facing data analytics tool for internal use to generate reports on internal/customer exams and customer trips.

Wolfram Research Champaign, IL
Software Engineer Intern May 2015 – May 2016

- Worked on various projects involving audio processing and connected devices in the Wolfram Language and Mathematica in an agile development environment. Worked with C++ and Java.
- Worked on improvements to the Wolfram Language and Mathematica audio processing functionality. Developed a C++ audio streaming library used by the backend of the Wolfram Language "Audio" object. The library supports streaming audio from files, from the Internet, and from microphone input, using various audio codecs.
- Developed a C++ library allowing the Wolfram Language to edit the metadata of audio files, such as ID3v2 tags. The library is built upon the open-source TagLib library.
- Developed a Java library for the Philips Hue connected home lighting system to integrate with Mathematica using the Wolfram Language DeviceFramework.
- Developed a real-time audio visualizer for the Philips Hue system using my library. The visualization synchronizes the lights with audio in real time by analyzing a file during playback or by using microphone capture. This was publicly demoed at the Pygmalion Festival 2015.
- Developed a presentation with my team lead, "How to Connect Your Device to the Wolfram Language", illustrating the creation of C++ and Java libraries which interface with devices. Developed a few demos including a camera image capture library. The Wolfram Language DeviceFramework then interfaces with these libraries via WSTP/JLink.
- Worked on setting up cross-platform software build infrastructure for multiple components of the Wolfram Language and Mathematica. Additionally worked on cross-platform builds for third party software dependencies.

University of Illinois at Urbana-Champaign Urbana, IL
Course Assistant for Systems Programming June 2014 – May 2015

- Developed two digital programming exams testing students on C string manipulation, file I/O, system calls, POSIX threads, inter-process communication (signals, pipes), and networking.

- Assisted the students during weekly office hours and on an online discussion board.

PERSONAL PROJECTS

Senior thesis – The Design and Implementation of an Autonomous Surveillance and Security Drone

- For my senior thesis, I designed and implemented an autonomous unmanned aerial vehicle which can patrol an area and identify humans. The thesis is available here in PDF form: http://nikkifayra.com/thesis.pdf
- The drone includes a Raspberry Pi computer which controls it. The computer runs an on-board web server and a secure WiFi access-point running a controller web-application built with Python.
- The web application uses the Google Maps API and allows for the drone to be sent to a GPS way-point, specify the speed and altitude, and specify a circuit for the drone to patrol.
- The drone has a camera, and sends images to a computer connected to the drone's WiFi network. This computer continuously processes images with OpenCV in Python to detect humans. Humans are identified using a Histogram of Oriented Gradients method conjunction with a Linear Support Vector Machine. When a person is detected, the machine sends an email to notify the user.

Automated web application vulnerability tester

- Checks a list of IP addresses for vulnerabilities, using Nmap to scan for running services.
- Checks for Heartblead, Shellshock, XSS, and a return-to-libc buffer overflow via non-malicious exploits.
- Aggregates URLs on host systems and writes vulnerability data to a JSON file.
- Includes Metasploit auxiliary module support.

Real-time strategy game with procedurally generated planet terrain

- Features a procedural terrain generator written in C++ using the Diamond-Square algorithm.
- Program generates terrain for an Earth-sized planet, including ice caps.
- Displays a two-dimensional topographic map with the player's cities and units. Player and AI can manage their cities and units, and attack and conquer enemy cities, using path finding via the A* algorithm.

Subversion Portfolio website

- Developed a personal portfolio website written in Ruby with Bootstrap for CSS styling which includes a file browser for all files in the projects, capable of showing past revisions.
- Parses SVN log and list XML data to render the portfolio.
- Allows viewers to comment on each project, storing the comments in a MySQL database.

Teamspeak 3 chat/music bot

- Developed a chat bot in C for Teamspeak 3 voice communication software which communicates with the server via a telnet interface, reading in text commands from users to perform functions.
- Parses URLs containing audio to play back through the Teamspeak 3 server**.**

REFERENCES

- Available upon request