Ryan Kemper

Education

2014–2018 **Computer Science**, *UC Santa Barbara - College of Engineering*.
Pursuing BS

Experience

Fall–Winter 2018 **UCSB CS Capstone team member - 1st place**, *LogMeIn (sponsor)*, Goleta.
Developed proprietary software that uses machine learning to offer targeted feedback for public speaking:

- Implemented NLP techniques like TF-IDF to identify keywords
  - Used nltk brown corpus to extract english term frequency and inverse document frequency data
  - Wrote Python scripts to extract thesaurus data and corpus word frequencies;
- Maintained 3 backend modules (text/audio/video):
  - Wrote shell scripts using sed/awk to integrate backend and frontend repositories;
  - Specified and implemented backend API for use in frontend
- Used parallelization to speed up backend, reducing processing time 2-3x:
  - Multithreading to parallelize API calls to Google Cloud Vision and IBM Watson
  - Multiprocessing for cpu-bound tasks like OpenCV haarcascades, ffmpeg mp4 conversion
- Wrote decision tree logic to convert raw analysis results into plain english feedback for user

Jun–Aug 2015 **Junior Test Engineer**, *wiLAN*, Greater San Diego Area.

- Implemented and configured closed testbed consisting of multiple switches, routers, and Unix servers to simulate a modern telecommunications network;
- Designed and implemented validation process to verify key components of proprietary electronic QoE evaluation model:
  - Wrote Python scripts to parse log files, calculate key video metrics, and verify accuracy of internal models;
  - Discovered key error in Video Mean Opinion Score (VMOS) model resulting in significant discrepancies. Proposed and evaluated potential solutions, and updated VMOS model to restore consistent behavior, resulting in avoidance of critical error.

Jun–Aug 2011 **Software Engineer Intern**, *OnRamp Wireless*, San Diego.
Worked as part of a team designing sensor analysis software in Java and Python, and designed a JUnit test suite to verify network integrity.
Tested and deployed software across various *nix-based virtual machines (Ubuntu, Debian, etc).

2008–2014 **Programming / Web Security Instructor**, *Wintriss Technical Schools*, San Diego.
Taught computer programming (Java, Python) and basic web security to students from ages 9-18 for several years

Primary Languages

Python 3	**Advanced**	*Preferred language for scripting/general computing*
C	**Advanced**	*Preferred language for low-level systems*
C++	**Intermediate**	*Weaker proficiency than C due to massive language scope*
Java	**Intermediate+**	*First language ever used, but dislike verbosity*
Scala	**Beginner**	*Wrote simple interpreter, huge fan of pattern matching feature*
HTML/CSS	**Intermediate**	*Understand divs, spacing, styling, but have more backend exp*
Javascript	**Beginner**	*Basic commands, DOM, closures*

Core Skills

Git add, commit, push, feature branching, stashing, tagged commits, remotes, reflog, rebase

Scripting regexes, safe file handling, multiprocessing, JSON/csv/other common data output formats

Security SQLI, CSRF, XSS, privilege escalation, buffer overflows / shellcode injection via env vars

Cryptography secure hash functions (like the sha-2 family), salting, asymmetric key encryption (like gnupg)

***nix** Primary experience with debian and fedora, limited experience with qubes OS

Concurrency Synchronization primitives such as mutexes (locks), semaphores, wait/join, thread vs process

Important Coursework at UCSB

CS 160 Led a team of 3 to implement an LL1 parser generator ("compiler compiler"). Wrote python scripts to convert high-level representation into several hundred lines of c++ test cases. Final result was a C++ parser generator which when input production rules for target grammar in EBNF form, output an auto-generated LL1 parser in C++

CS 162 Exposure to functional and logical programming, tails-call recursion, currying, languages like Scala and Prolog

CS 170 Implemented basic OS kernel [supports 8 procs, most basic system calls (ioctl, open/close/read/write/sbrk/pipe/etc), piping/file descriptors, POSIX-compliant]

CS 177 Implemented "padding oracle" attack against CBC w/ PKCS #7, decrypting arbitrary ciphertext byte-by-byte. Wrote a password cracker (hash bruteforcer) in both C and Python, supports various bruteforce modes (a-z permutations, a-zA-Z permutations, alphanumeric) and dictionary/wordlist attacks (+ limited permutations on wordlist). Course concepts covered included AES, Diffie-Helman, RSA, buffer overflows, etc

CS 189AB See "UCSB CS Capstone" under Experience section [above]

Other interests

Financial Modelling Discounted cash flow models, can read balance sheets and 10-K's
Understand concepts like operating cash flow vs free cash flow, capex, inventory management, etc.

Spanish Fluent as a non-native speaker