Guozhen She

GitHub: github.com/hazelnutsgz Email:gzshe15@fudan.edu.cn HomePage: https://sgzhazelnut.github.io/

EDUCATION

Fudan University, Shanghai, China

Bachelor of Computer Science expected in 07.2020

GPA (overall): **3.55/4.0**; Ranking: **21/117(17.9%)** GRE:331

Body:Computer Architecture(A)| Computer Network(A-)|Computer System(A)|Database Implementation(A-)|Distributed System(A)

Brain: Linear Algebra(A)|Mathematical Analysis(A)|Neural Network and Deep Learning(A-)

Mouth: C Programming(A)|C++ Programming(A-)|Web Development(A)|Digital Logic Design Experiments(A)

SKILLS

Programming Language: Python, Java, Golang, C, C++, JavaScript, MATLAB, Rust(newbie), Haskell(newbie)

Framework & Library: Tornado(Python),D3.js,Kernel Programming(C,Assembly,kgdb),Tensorflow,PyTorch, C++ STL

Tools: LaTeX, Vim, Git, Docker**,** GNU toolchain

Soft Skills: Networking,Open Source Engagement,Ask Smart Question

SELECTED PROJECTS

C++ Multiple Pattern Text Matching Tool (https://github.com/hazelnutsgz/NaiveACAutomation)

Interactive Visualization of Google Scholar Coauthorship(https://github.com/hazelnutsgz/NaiveScholarMap**)**

C++/C# UserAgent Parser(https://github.com/hazelnutsgz/UADetectorCPP**)**

Container Acceleration(https://github.com/hazelnutsgz/ContainerAcceleration)

ACADEMIC EXPERIENCES

University of Illinois Urbana-Champaign REU program | Research Intern

- **Seccomp Acceleration for Container**
  - The idea is utilizing the hash table to capture the historical intrusion data, to by pass the overhead of BPF filter. So implemented a pluggable kernel module which hook the functional pointer pointing to the bypass logic, the bypass logic contains a hierachical jhash table to cache the data and a predefined entry pool for serving the cache on demand. To enable the dynamic seccomp profile table, hacked the libseccomp to parsing the Docker seccomp profile when loading the policy, and patch the linux kernel to attach some hooks to support it. Finally, improvement on many common workloads is 5%~7%
- **Linux Security Module Optimization(ongoing)**
  - Located the bottleneck of selinux(compilation/loading), then read the source code of selinux module, reduced the overhead through bypassing the policydb serialization on virtual filesystem(selinuxfs).

Tsinghua University, Research Intern

- **Chat App Monitoring System**
  - Hacked WeChat communication protocol to intercept the message, built a daemon web service to monitor and store the intercepted messages(video, text, audio) by polling request.
  - Build a mobile app to help the community members recording their activities.

Fudan University, Mobile Systems And Networking Group Research Assistant April.2017-

- **Social Network Analysis (Google Scholar & LinkedIn)**
  - Crawled data from LinkedIn and Google Scholar.
  - Used deep learning techniques to auto detect the mis-configured profile.
- Qingyun Go
  - Built a **geo-based** social App. The communication between backend and frontend is hosted on HTTPS protocol based on RESTful API. The backend is FastCGI integrated with C++ code, while the frontend is a javascript runtime with asynchronous API. Then developed some streaming analytic tools to monitor the status of service and the behavior of users.

INDUSTRY EXPERIENCES

Microsoft Research Asia, System and Network Group | Research Intern Jan.2019-

- Real-time Bot Detection System for Azure Cloud Service
  - Implemented a Golang web service to **build the preprocessing pipeline** of daily network log data(8,000,000) from Bing, which parse the raw log concurrently leveraging **goroutines** into heterogeneous files hosted on a distributed file system. Implemented a tree-based mapping structure to support the range query for ip address in the internal service.
  - Based on that, provided an analytic service(React, D3.js) for visualizing and understanding data. For conservation of memory footprint on VM, the data was **fetched on demand** and cached as the data structures in memory, guaranteeing the data at **hotspot** would stay longer in memory to accelerate the analytic efficiency.
  - Implemented the algorithm to generate the behavior-based images for each session, then develop a **CNN-based** model to detect bot behavior by classifying the images generated, which reached **94.3%** accuracy on labeled Bing log data.
  - Submit patch to hyperscan, then built a useragent parsing library utilizing hyperscan for high performance regex matching..

Intel Asia-Pacific R&D, Open Source Technology Center | SDE Intern Aug.2018-Nov.2018

- **Contributed code** to StarlingX(OpenStack Foundation), assisted in deploying the StarlingX on bare-metal devices.
- Built a **rule-based** command line tool which migrates code from python2 to python3.
- Assisted the colleagues to setup the **compiling farm** based on K8S for building of StarlingX project.
- Implemented a static graph-based algorithm for **package dependency analysis** in the project.

PUBLICATION

LBSLab: A User Data Collection System in Mobile Environments

- Qingyuan Gong, Xinlei He, Qinge Xie, Shihan Lin, **Guozhen She**, Ruiyu Fang, Rui Han, Yang Chen, Yu Xiao, Xiaoming Fu, Xin Wang
- Proc. of Workshop MHC, **UbiComp** 2018