ABHISHEK VASISHT BHASKAR

Curriculum Vitae

CONTACT

Email: abhishekvasishtb@gmail.com
Address: Ithaca, NY
Homepage: abhishekvasishtb.github.io

EDUCATION

M.S – Computer engineering , Syracuse University, GPA : 3.83 June 2016
Thesis: *DroidUnpack: Automated code extraction from packed Android applications.*

B.E – Telecommunication Engineering, PESIT, Bangalore June 2014

PUBLICATIONS

- **Things You May Not Know About Android (Un)Packers: A Systematic Study based on Whole-System Emulation**
  Yue Duan, Mu Zhang, **Abhishek Vasisht Bhaskar**, Heng Yin, Xiaorui Pan, Tongxin Li, Xueqiang Wang, and Xiaofeng Wang in *NDSS 2018, San Diego, California, USA* (Acceptance Ratio: 15.4%).
- **Extract Me If You Can: Abusing PDF Parsers in Malware Detectors**
  Curtis Carmony, Mu Zhang, Xunchao Hu, **Abhishek Vasisht Bhaskar** and Heng Yin in *NDSS 2016, San Diego, California, USA* (Acceptance Ratio: 15.4%).
- **Binary Code Continent: Finer-Grained Control Flow Integrity for Stripped Binaries**
  Minghua Wang, Heng Yin, **Abhishek Vasisht Bhaskar**, Purui Su, and Dengguo Feng in *ACSAC 2015, Los Angeles, California, USA* (Acceptance Ratio: 24.4%)

EXPERIENCE

Software Engineer

July 2016 – present

GrammaTech, Inc.

- **API Anomaly Detection** (Ongoing) - Part of the team implementing a statistical/ML model based API usage anomaly detection using CodeSonar as part of a *DHS research contract*.
- **As part of the team adding Objective-C support to CodeSonar** - GrammaTech's Static Analysis Tool. This entailed integrating the *clang* compiler frontend to CodeSonar. My tasks included, but not limited to
  - Supplementing clang to generate GTIR (GrammaTech IR).
  - Writing small ObjC test programs iterating all language features.
  - *Design/Implementation* of type merging, data layout and field size/offset updating for all ObjC types in the CodeSonar backend.
  - Multiple changes to the generated IR for better results.
  - Various improvements to the CodeSonar core analysis to get better analysis results for ObjC.

Research Assistant

May 2015 – June 2016

SYCURELAB – Syracuse University

Advisor: Dr. Heng Yin

- **Principle Programmer for DECAF** - A dynamic program analysis tool.
  - Headed a project with Los Alamos National Laboratory to develop a software fault injection framework using plug-ins on DECAF.
  - Improved techniques for Virtual Machine Introspection – memory module/process discovery on both Linux and Windows hosts.
  - Combining SLEUTHKIT with DECAF to enable native function call tracing. User support.
- **Working on Droidscope –** A dynamic program analysis tool for Android. Updating to the latest Android Runtime(`libart`) and building an automated/generic application unpacker on top of it.
  - Studied AOSP internals and the Dalvik VM to develop a new VM introspection design on both native and Java semantic levels.
  - Built an unpacking framework, *DroidUnpack*, on top of this, which relied on intrinsic characteristics of the Android runtime using VM inspection to precisely recover hidden code and reveal packing behavior.
  - Ran DroidUnpack on applications packed with 6 known packers and results presented as part of master's thesis.
- (*Assistantship awarded on a competitive basis and included a complete tuition award*)

TECHNICAL SKILLS

- **Programming Languages**: C++, C, Python, Objective-C, x86 and ARM assembly, C# , Java, Linux Kernel.
- **Scripting**: Bash, Makefile.
- **Program Analysis**: Static Analysis (CodeSonar), DECAF, Droidscope, IDA/IDApython scripting.
- **Compiler instrumentation**: LLVM/clang Compiler toolchain.
- **Operating system internals**: The Linux kernel, Android internals, Objective-C runtime.

RESEARCH PROJECTS

- **Data Access Protection:**

  Source: `https://gitlab.com/TheLoneRanger14/mem_protect`

  Implemented a compiler instrumentation module (/LLVM pass) on the LLVM/clang tool-chain, with a run-time library to track reads/writes to sensitive memory, hence preventing malignant writes to them. Tool was tested on the Google Chromium project and other binaries with instrumentation of a few objects with no major overhead.