# Danny Kim

## Personal Data


Address: [redacted]
Phone: [redacted]
email: [redacted]


## Education

| | |
|---|---|
| *Current* | Ph.D. in Computer Engineering, **University of Maryland**<br>Ph.D. Focus: Cybersecurity and Program Analysis<br>Thesis: "Analyzing Program-level Features to Detect Malware"<br>Advisor: Prof. Rajeev Barua<br>Gpa: 3.63/4.00 |
| May 2014 | Bachelor of Science Electrical Engineering, **University of Maryland**<br>Gpa: 3.76/4.00 |
| Spring 2013 | Exchange Semester at **Denmark Technical Institute**, Lyngby, Denmark |

## Publications and Presentations

| | |
|---|---|
| July 2017 | DIMVA Conference 2017<br>*DynODet: Detecting Dynamic Obfuscation in Malware* |
| June 2017 | Annual Laboratory of Telecommunication Sciences Presentation<br>*Detecting Advanced Obfuscation in Malware* |
| Fall 2015 | Leidos Research Presentation<br>*Detecting Advanced Malware* |
| June 2016 | Annual Laboratory of Telecommunication Sciences Presentation<br>*Analyzing the Prevelance of Obfuscation in Malware* |

## Scholarships and Awards

| | |
|---|---|
| Present | ARCS Scholarship Recipient |
| May 2014 | University Honors Graduate |
| Sep 2013 | AFCEA Behesda Scholarhsip Recipient |
| Sep 2013 | Jeong H. Kim Scholarship Recipeient |
| Sep 2013 | A. James Clark Scholarship Recipient |
| Sep 2012 | Dean's Scholarship Recipient |

## Programming Knowledge

| | |
|---|---|
| Proficient: | C, C++, Python, Bash, x86 Assembly |
| Familiar: | Java, Matlab |

# Research Experience

## 6.1 Summary

My thesis focuses on discovering fundamental behavioral differences between malware and benign software. My work primarily involves dynamic analysis, which is when the program is executed in a protected environment. I use a dynamic binary instrumentation tool to discover behavior that most current dynamic tools cannot. Most dynamic tools today analyze malware using OS-level behavior, but with a DBI tool, I can analyze instruction-level behavior. Instruction-level behavior is more specific than OS-level indicators, which often results in a higher accuracy of detection. I initially focused on a class of behavior called obfuscation, in which malware tries to hide from detection tools. I found quantifiable differences between obfuscation occur in malware and benign software, allowing me to build a tool to detect the difference. I have expanded on this work and am working on a general malware detection tool using a combination of dynamic analysis and machine learning.

## 6.2 Specifics

- Researching an innovative method using static and dynamic program analysis to aid malicious software detection.
- Developing behavioral analysis tools with binary rewriters such as DynamoRio and Pin in C++ to obtain instruction-level behavior into a program's execution.
- Enhancing existing dynamic malware detection schemes that rely solely on OS-level features by augmenting analysis with program-level behavior.
- Analyzing the differences in obfuscation and binary protection schemes present in malware versus benign software.
- Implementing just-in-time disassembly to get a limited, but accurate projection of a program's execution.
- Creating dynamic signatures that can detect malicious behavior without any prior knowledge.
- Using Cuckoo Sandbox as the dynamic analysis scheduler and virtual-machine manager.
- Building a machine-learning malware detection tool with SciKit and TensorFlow that is able to correctly detect more than 95% of all programs as benign or malicious.
- Investigating methods of obtaining a unique dataset, such as using the import address table hash, for reproducible testing for dynamic malware analysis with machine learning.
- Using advanced debugging tools such as IdaPro and OllyDbg to obtain instruction-level intuition on a program's behavior.
- Rebuilding control-flow graphs of obfuscated malware to ensure control-flow integrity.

## Engineering Experience

Jun 2016 – Aug 2016
*Windows Authentication Development Engineering Intern*
**Microsoft**, Redmond, WA
- Implemented a new security feature that allowed protected Windows processes to communicate with an unprotected process without compromising security.
- Developed new methods of authentication in collaboration with a multi-disciplinary team in order to ensure quality and customer satisfaction.
- Shipped a new security feature in the latest release of Windows.
- Maintained an aggressive timeline to meet the real-time needs of customers.

Jun 2014 – Aug 2014
*Computer Engineering Intern*
**Key Technologies**, Baltimore, MD
- Developed and tested firmware written in C for a handheld medical device measuring blood glucose levels.
- Proposed the use of a bluetooth chip as method of communication between the device and its controller.
- Collaborated with multiple disciplines on the design of a device to ensure the device met all the customer's standards.

Jan 2014 – May 2014
*Computer Engineer*
**Image Engineering**, Baltimore, MD
- Improved speed of existing laser software controller by rewriting the firmware in ARM assembly.
- Created a seamless transition between the old and new system in order to ensure minimal transitional downtime.
- Incorporated a modular design scheme in the firmware to improve later firmware upgrades.

Sep 2013 – Jan 2014
*System Engineer Project*
**University of Maryland**, College Park, MD
- Used the STM32-M4 Discovery Board to design a multi-meter in a combination of C and assembly.
- Integrated knowledge of Bus structures, memory, I/O interfacing and data structures to complete project.
- Created a reference manual that documented the different aspects of the project and the design considerations.

Jun 2013 – Jan 2014 | *Consulting Engineering Intern*
**iVeia**, Annapolis, MD
- Built a python-based GUI that installed platform specific debian packages on customers' computers.
- Implemented new internal tools that integrated Bash and Python scripts that automated the generation of release notes and sped up product build time.
- Analyzed filesystem performance with IO stress tests under real-world conditions.

Jun 2013 – Jan 2014 | *System Engineering Project*
**University of Maryland**, College Park, MD
- Built a Bluetooth-enabled LCD driven by a microcontroller to display real-time statistics from a car.
- Used existing OBD2 communication library functions to obtain information from the car using the microcontroller.
- Designed a custom piece of hardware to communicate between the controller and the LCD.

## Leadership Experience

Fall 2016 – Fall 2014 | *Graduate Teaching Assistant*
**University of Maryland**, College Park, MD
- Led two weekly discussion classes of 20 students each for a computer organization course
- Reinforced topics taught in lecture by giving examples and responding to students' feedback.
- Met with the Professor and other GTAs for quality checks and status reports.

Spring 2016 – Fall 2015 | *ECEGSA Vice President*
**University of Maryland**, College Park, MD
- Created and planned events for the ECE department graduate students to facilitate both personal and academic growth and encourage multi-disciplinary collaboration.

May 2014 – Jan 2012 | *Resident Assistant*
**University of Maryland**, College Park, MD
- Served as a resource to residents and students in addition to promoting student development and interaction in residence halls by facilitating community-building events and activities for approximately 50 students.
- Enforced behavioral expectations and responded promptly to individual crises and discipline matters.