Michael Bailey

michaelbailey.co
linkedin.com/in/cybermichael
github.com/mike-bailey

Alexandria, Virginia
(703) 282 4029
mbaile18@gmu.edu
github.com/srct

Technical Experience

Web Developer (Primarily LAMP, jQuery) 2010-Present

Intern – The Crypsis Group (Tysons) Summer 2016 – **Present**

IT Consultant – NanoTech Computers (Alexandria) 2015-2016

Consultant – Métier Defense Solutions (Dulles) Summer 2014

- Mobile Device Security and basic Linux Administration

Mason Competitive Cyber – President 2017

Education and Certification

George Mason University Class of 2019

- Studied introductory Python and studying introductory Java
- Mason Competitive Cyber President
- Student Run Computing and Technology Systems Administrator
- Mentor for 3 Frost Middle School CyberPatriot Teams
- STARS Student/VSE STEM Outreach Member

Marshall Governor's STEM Academy 2013-2015

- Computer Systems A+, Network Administration

James Madison High School 2010-2015

Certified in…

- CompTIA A+ 2015-Present
- Microsoft Technology Associate 2010-Present

Hackathons and Cybersecurity Competitions

- Multiple Top 100 Placements in CTFs 2014-Present
- CapitalOne GMU Wargame Winning Team, Booz Allen CTF Winner 2017
- HoyaHacks Hackathon "Best Embedded Hack" 2016
- MakeCU Best Use of AWS, 2nd in MedTech Hack 2016

  National Finalist Captain (1 of 12 in Nation) in CyberPatriot VII 2015
- Finalist, HS and College Division Maryland Cyber Challenge 2013-2015
- 13th Place in CyberPatriot VI (Semifinalist) 2014
- State Champion Team, 2nd in Region – CyberPatriot & Governor's Cup 2014

Recent Projects

- Crypsis Slack Bots (all using Ruby)
  - AWS Bot – Manages AWS account. Allows group to list running EC2 instances, spin up instances, read through the SQS feed, and list and download objects via Slack.
  - Intel Bot – Offers GeoIP lookup (including any known threat intel on the IP), file hash lookup, CVE lookup, BTC exchange rate lookup, URL to screencap conversion, and more
- DFIR Automation (particular name under NDA)
  - Ruby web interface that generates executables to collect key DFIR artifacts from victim computers, sends them to AWS S3, where they're distributed to a managed Docker cluster to be processed using various Python and compiled tools into plaintext logs for Splunk ingestion, then reuploaded to S3. Also the AWS janitor, cleaning up data into the appropriate tiers and using billing as a metric
- GRR Docker Provisioning and Proxying
  - Integration into previous executable in which Google Rapid Response servers are provisioned per client and immediately made routable to one of three hosts, uses haproxy and Docker Remote API, enables the already existing executable to install the GRR agents onto the machines, forensic data backed up to AWS S3 automatically