Ben Murphy

benmmurphy@gmail.com [github.com/benmmurphy]

OBJECTIVE
- To utilize my computing knowledge and skills in an exciting and challenging environment.

EDUCATION
- **Queensland University Technology**, 2005 - 2006
  Honors, Information Technology
- **Queensland University Technology**, 2002 - 2004
  B.S., Information Technology
  Majoring in Software Development.

SKILLS
- Ruby, Erlang, C, Java, C#, Visual Basic, DTrace
- Rails, Git

WORK EXPERIENCE
- **Software Engineer**, Fonix Mobile November 2013 - Present
  - Created and maintained Fonix SMS Gateway written in Erlang that integrates with Mobile operators using a mix of SMPP and HTTP interfaces. The Gateway is used for both bulk SMS services and premium billing.
  - Worked on Fonix Ruby on Rails projects including ZenSend bulk SMS API, SMS competition platform and winner picking solution. This included work with ReactJS allowing non-technical users to create a tree diagram that controls how SMS messages are responded to. ReactJS was also used in creating a page similar to tweet deck with infinite scroll for displaying incoming SMS messages.
- **Software Engineer**, Mobile Interactive Group/Velti Sep 2008 - November 2013
  - Maintained MIG SMS Gateway written in Erlang.
  - Created and maintained MIG Mobile CMS platform written in Java using Spring and Hibernate.
- **Software Engineer**, Mincom February 2006 - June 2008

PROJECTS
- **Java** Between 2012 and 2014 I found 18 Java Sandbox escapes. Because of the way Oracle reports vulnerabilities it is difficult to attribute who was the first to report a vulnerability. I know that at least 2 of the vulnerabilities I reported collided with Adam Gowdiak because I reverse engineered them from public statements he made. I have written up three of the vulnerabilities on my blog. (ZDI-13-002, ZDI-13-040, ZDI-13-041, ZDI-13-042, ZDI-13-075, ZDI-13-079, ZDI-13-089, ZDI-13-132, ZDI-13-159, ZDI-13-160, ZDI-13-244, ZDI-13-245, ZDI-13-246, ZDI-13-247, ZDI-13-248, ZDI-14-103, ZDI-14-104, ZDI-14-105)
- **Joyent** Between 2015 and 2017 I found 3 SmartOS Zone escapes that used DTrace, 1 SmartOS Zone escape that was based on the previous BadIRet work and 5 SmartOS arbitrary kernel memory reads using DTrace. In addition I found 2 vulnerabilities in the Joyent SDC Docker API that allowed for Zone escapes. (ZDI-16-168, ZDI-16-169, ZDI-16-170, ZDI-16-465, ZDI-16-274, ZDI-16-464, ZDI-16-344, ZDI-16-498, ZDI-16-499, ZDI-16-500, ZDI-16-466, ZDI-16-453, /proc vulnerability)
- **Rails**, CVE-2012-2661 (Unsafe Query Generation), CVE-2013-0156 (Remote Code Execution)

SPEAKER
- **DTrace Conf 2016** DTrace Exploitation

HONOURS
- **Best Server Side Bug**, Pwnies, 2013
- **Pwn2Own Java**, 2013

`md5(resume.pdf) =` 7978EFA52DCB3C788A6C101012644EE9