Daniel Dressler
danieru.dressler@gmail.com

1x *Microsoft* Internship working on Windows

3x Google Summers of Code on Linux printing

Did you know I have programmed...

- A game for the original Gameboy in C?
- A FreeBSD rootkit for bonus marks?

- Firefox OS's Out-Of-Memory Killer logcat logging?
- A 3D turn-based-strategy game in javascript?

Linux's IPP over USB driver

Wrote Linux's driver for IPP USB class devices. Contributed patches to system-config-printer for autostart support. The driver, ippusbxd, proxies HTTP between TCP and USB Bulk. Completed in 2014. Shipping in Ubuntu 14.04. Written in **C**.

Developed with the support of Google Summer of Code from May 2014 to September 2014. Linux Foundation sponsored my trip to the August 2014 meeting of the Printer Working Group where I presented upon ippusbxd.

Microsoft Internship

From May 2013 to August 2013 I did an internship within the Windows Sustained Engineering divison, within the Printers and Peripherals team.

During this internship I learned the Windows Development process by guiding backported patches through the acceptance process into Windows 7 and Windows 8.

My primary concern during this internship was the development of the OCASE BCDE project. [B]ulk [C]rash [D]ump [E]xporter features a domain specific non-turing complete language for scripting interaction with the Windows Debugger (Windbg). This is used to generate reports which developers use to analyze specific crash buckets across thousands of crashes.

BCDE was written in **C++11** and leveraged internal Microsoft services for the selection and collection of crash dumps. It generated final results as Excel documents.

Before BCDE developers tested crash theories upon 5 to 10 crash dumps. They did so through time consuming manual operation of Windbg. Now with BCDE developers can test hypothosises against 1000+ crash dumps with a fraction of the effort.

A full time job offer was extended but I declined as it would have precluded coming to Japan.

FreeBSD Rootkit

The University of Calgary has a special course where students write exploits, virsuses, and in the final assignment lment an anti-virus program to combat everyone else's viruses.

The first several weeks of this course explore the legal situation. We read Canadian and American computer security laws to understand our obligations. The virus writing itself is restricted to a secure computer lab. No outside electronics are allowed in this room. An air-gap exists seperating the lab's computers preventing virus propogation. I mention this to impress upon you that I wrote the virus within a legal environment and in full compliance with the law.

At the begining of the course the professor announced an award for the most annoying viruses a group could produce. This award included a monetary prize split between the group members. The contest was to be decided by a vote among the students.

This prompted me to far outside the main assignment's requirements and develop a proper 1990's era rootkit. Early versions of this rootkit included self preservation features which were removed because I felt these were too cruel. The final did include methods for causing kernel panics should users attempt to unload the virus.

The rootkit's most interesting feature, and what won us the aforementioned prize, was how it attacked anti-virus programs. It used a psudeo-bloom filter hidden in an unused field in the FreeBSD's process struct. This bloom filter tracked which files a process had opened. A seperate stolen field tracked the total number of files opened. This was done to prevent the rootkit from detecting the terminal as an anti-virus.

Once a program opened N files the rootkit hooked the write system call and silenced any writes. This caused every anti-virus scanners to appear malfunctioning. The scanner would find the viruses in the first N files and go silent for other known infected files. The net result was many late nights for other groups. Some groups even gave up and accepted reduced marks in exchange for not handling our virus. My group won by a landslide vote.

Now again let me impress upon you that no copies of this rootkit ever existed outside of the secure lab. All copies were destroyed after final grading. I have never written viruses outside this course nor have I ever broken Canadian or American computer security laws.

Lockless Pool Allocator

I collaborated with my friend who is active working on Clang's C++ standard library. He approached me for help on implementing a memory management feature in an experimental C++14 standard. Specifically http://www.open-std.org/jtc1/sc22/WG21/docs/papers/2014/n3908.html Beyond prototyping the pool allocator my biggest contribution was extending my friend's implementation of the synchronized pool allocator to be lock-less.

Note: This set of patches has not been submitted for review but I include it to demonstrate I am familliar with the challenges of lock-less data structures.

Other projects

PPSSPP Keymapping (*C++*, 2013)
Restructured the PSP emulator PPSSPP's input path to support key mapping. Required extensive refactoring throughout execution path.

Contract Webdev (*HTML*, 2010-2013)
Developed 10+ websites ranging from marilynmonroe.com, necaonline.com, to eCommerce sites

Foomatic Database Engine (*Perl*, 2011)
Rewrote Linux's printer database generation into Perl, reducing code size by 8K lines. Added SQLite backend. Shipped in Ubuntu 12.04

ExtractZipFile (*C & Javascript*, 2013)
Wrote BlackberryOS10's HTML5 ZIP library. Provides Javascript with interfacing to native C for max performance.

Original Gameboy Game (*C*, 2014)
Wrote gameplay for a shmup running on the original Gameboy. Fought broken compilers to extract modern gameplay on a 1Mhz CPU.

LIBJTAPI (*C*, 2012)
Implemented Free Standards Group's abstract job ticket manipulation. Closely matches Printer Working Group's Semantic Model.

Degree

University of Calgary
Bachelor of Science in Computer Science
August 2010 - April 2014
Graduated 2014
GPA: 3.38

High School

Graduated April 2010

Address

Sakura House
Motohasunuma #210,
45-18 Hasunuma-cho,
Itabashi-ku, Tokyo 174-0052

Work Authorization

Working Holiday Visa
Valid until April 2015
Extendable to October 2015

Work History

GSoC Linux Foundation on ippusbxd (May to September 2014)

Microsoft Internship on BCDE (May to August 2013)

GSoC Linux Foundation on Libjtapi (May to September 2012)

GSoC Linux Foundation on Foomatic (May to September 2011)

Wal-Mart (2008-2010)

McDonalds as Cashier (2006-2008)

Github

https://github.com/daniel-dressler