Roland Bogosi | Curriculum Vitæ

Unit 7537, PO Box 6945, London – W1A 6US – UK

☎ +44 20 3411 1758 • ☎ +1 224 366 7654 • ✉ root@rolisoft.net
🌐 rolisoft.net • in RolandBogosi • Ⓢ RoliSoft • ◯ RoliSoft

"Knowing is not enough; we must apply. Willing is not enough; we must do." - Johann Wolfgang von Goethe

Education

Sapientia EMTE, Faculty of Technical and Human Sciences   **Târgu-Mureș**

B.Sc. Computer Engineering   *September 2012–July 2016*

Gave multiple presentations in various fields; attended the XVI. Scientific Student Conference with a project which won a special award; participated in a research group studying sentiment analysis as an extracurricular activity; doing thesis in the field of Information Security.

Experience

BrightAlien Ltd   **London (Remote)**

DevOps Engineer and Full-Stack Developer   *December 2013–December 2015*

Job responsibilities include the full-stack development of 9 websites plus an internal dashboard; the installation with continued proactive monitoring and administration of 4 Linux-based and 1 Windows-based production servers; miscellaneous DevOps responsibilities.

Lynx Solutions SRL   **Târgu-Mureș**

Android Developer Internship   *July 2014–August 2014*

Job responsibilities included the creation of a 30-page document describing the application to be developed and its specifications; the single-handed development of the documented Android-based application with UI accommodating both smartphones and tablets alike (through the use of fragments and multiple layouts with constraints); overseeing fellow interns through their journey of implementing the documented REST API.

Certifications

IPv6 Certified Sage, Hurricane Electric   *March 2011*

Publications

Sapiness Sentiment Analyser: Katalin Tünde Jánosi-Rancz, Zoltán Kátai and Roland Bogosi; *Acta Universitatis Sapientiae, Informatica,* vol. 7, no. 2, pp. 186–199, 2015.

Languages

English: Full professional proficiency

Hungarian, Romanian: Native or bilingual proficiency

Technical Skills

Languages: C/C++ *(8 years)*, C# *(10 years)*, Java *(9 years)*, Go, Python, PHP *(12 years)*, JavaScript, SQL

Libraries: STL, Boost, Qt, MFC, Swing, WinForms, WPF, C++/CLI, NUnit, jQuery, Bootstrap, Angular.js, PHPUnit...

Technologies: Android, Android NDK, Debian, CentOS, BSD, nginx, Apache, HHVM, exim, Postfix, Redis, memcached, ElasticSearch, Sphinx, MySQL/MariaDB, SQLite, Oracle, SOAP, JSON, Google ProtoBuf, XDebug, OpenVPN, L2TP/IPSec, Radius/LDAP, Tor/I2P, Meshnets, Google AppEngine, Amazon AWS/Linode/DigitalOcean, CloudFlare/Incapsula, NSIS, InnoSetup, WiX, MSTest, dotCover, dotMemory, dotTrace...

Fields: Penetration testing, Reverse engineering, System administration, Shell programming, Task automation, High availability (Load-balancing, TTFB optimization...), Software performance, Server/software security, Data encryption, Networking, Screen scraping, Testing, Search engine optimization (SEO), Continuous Integration

Personal Development

- **10 years** of experience in **.NET**/**C#**.
  - First exposure to object-oriented programming in **Visual Basic .NET**, then **C#**, via the .NET framework at the age of **13**.
  - Published first **desktop application**, which implemented an *original idea*, at the age of **13**, which utilized **databases**, **regular expressions**, **networking** and **screen-scraping**. Application was published to **Softpedia**, where it was accepted and reviewed by their staff, leading to a few hundred downloads during its lifetime.
  - Use of **neural networks**, in order to train a software to **recognize** numbers and letters that appear on a **webcam**-provided image in real-time, at the age of **16**.
  - Published first **open-source application** at the age of **16**, which then went **commercial** at the age of **18**, by integrating serial numbers generated and validated using **RSA public**-**private key encryption**, which were released by a **PHP script** called by either **PayPal Instant Payment Notification API** or the **Bitcoin API**.
  - Experience with **NSIS**, **InnoSetup**, and **WiX**, including scripting of custom methods.
  - Extensive knowledge of **Test**-**Driven Development** via leading **unit test libraries**, such as **NUnit** and **MSTest**.
  - Extensive experience with various development tools, used to:
    - profile and **analyze unit test coverage**, such as **dotCover**;
    - profile, detect and **mitigate memory leaks**, such as **ANTS Memory Profiler** and **dotMemory**;
    - profile, detect and **mitigate performance bottlenecks**, such as **ANTS Performance Profiler** and **dotTrace**.
  - Various short-term hobby projects along the time driven by the desire to learn and experiment in various fields, thus expanding my experience and increasing my knowledge of various algorithms, frameworks, and my programming ability as a whole.

- **8 years** of experience in **C/C++**.
  - First exposure to unmanaged programming at the age of **15**.
  - Extensive use of the UI libraries **Qt** and **MFC**.
  - Experience with **cross-platform** software development on **Windows**, **Linux**, **FreeBSD** and

Mac OS X platforms, including usage of OS-specific component implementations, such as differing network stacks on the lowest level.

- Usage of **qmake**, **CMake** (and its components **CPack** and **CTest**) to build, test and package software for distribution.
- Unit-testing with **Boost.Test** and **Google Test**, including mocking with **Google Mock**.
- Experiments with massively parallel systems, such as **OpenCL** and **nVidia**'s **CUDA**, which are **GPU-backed** languages. Frequent use of **OpenMP** and projects with **MPI**.

○ **12 years** of experience in **HTML**, **CSS**, **JavaScript**, **PHP** and **MySQL**.
- First exposure to the world of programming via PHP at the age of **11**.
- Published first dynamic webpage written from scratch at the age of **12**.
- Received first \$100 check from **Google AdSense** at the age of **14**.
- Developed first **autonomous web-services** that rely on user-contributed data at the age of **17**. Aforementioned web-service is a **recommendation system**, which provides **tailored recommendations** based on a user's preference list.
- Up-to-date knowledge of **HTML 5**, **CSS 3**, **PHP 5**, **JavaScript**, and **SQL** as of today.
- Extensive knowledge in the use and production of **REST APIs**, **OAuth**, **OpenID** and **SOAP**, also having experience in **data-interchange formats**, namely: **JSON**, **XML**, **Google Protocol Buffers**, **BSON**, and various others.
- Knowledge of various **SQL flavors**, in order of skill level: **MySQL**, **SQLite**, **Oracle**, **PostgreSQL** and **Microsoft SQL Server**.
- Extensive knowledge of **optimization and caching mechanisms**, including the optimization of **relational databases** (**query plans**, **indices**, **complexities**, . . .), **key-value stores** (such as **Redis** and **memcached**). **Multi-tiered caching systems**, utilizing the aforementioned technologies, with a fallback to file-based **LRU caches**. Brief exposure to **document stores**, namely **MongoDB** and **CouchDB**.
- Extensive exposure to **dedicated search servers**, namely **ElasticSearch** and **Sphinx**, including both their programmatic use and proper server configuration.
- Experience with **jQuery**, **semantic web** and **dynamic webpages**, utilizing **AJAX** and **JavaScript "MVW"** frameworks, namely **Angular.js**.
- Experience with **Bootstrap**, and **grid-based fluid** and **responsive web design**.
- Familiarity with **Model-View-Controller** (**MVC**) architectures.
- Extensive experience with **test-driven development** via **PHPUnit**, and the use of various development tools, such as **performance profilers** and **remote debuggers**, namely **XDebug**.
- Experience with both **horizontal** and **vertical scaling** of both **databases** and **codes**.
- Familiarity with the integration and implementation of various **payment gateway processors**, such as **PayPal IPN**, **Stripe**, and **Bitcoin RPC**.
- Intricate knowledge on the **security** front, and up-to-date on the **0-day scene**. While programming, I have a **security- and optimization-focused** mindset.

○ **8 years** of experience in **Unix-like systems** (**Linux**, **BSD**)
- Installed first **Linux distribution** at the age of **14**.
- Flashed first router with **OpenWRT** and **DD-WRT** at the age of **15**.
- Got the task of configuring and managing **unmanaged VPS** instances at the age of **15**.
- Installed **Linux from Scratch**, at the age of **16**.
- Obtained first certificate **IPv6 Certified Sage** from Hurricane Electric at the age of **18**.
- Extensive knowledge of the **Linux** ecosystem as of today. Personally owning and administering

multiple production and staging servers, while providing support for others on an incident-response basis.

- Early adopter on multiple fields, including **IPv6**, having been participated in the **private beta** of the IPv6 deployment of both **RDS&RCS** and **Dreamhost**. The configuration and use of **IPv6 networking** was the primary reason to stay with **OpenWRT**-flashed routers in the first place, as it was not yet available in consumer firmware due to the low technology adoption rate at the time.
- Tendency to **automate tedious processes**, by **scripting them** in the appropriate environment. (Such as **shell scripts** in **Unix** and **Unix-like systems**.)
- **Strong Bash scripting** experience, including **heavy terminal usage**, and heavy knowledge of the **BSD** and **GNU userland tools**.
- Ability to configure, administer, update and support **Debian** and **CentOS**-based systems for various purposes:
  · Web servers: **nginx**, **lighttpd**, **Apache**, with fast and secure configurations, including **caching** and **microcaching** techniques, on-the-fly optimization with **Google PageSpeed** modules; **Web Application Firewall** security modules, such as **mod_security** and **mod_evasive**; proper deployment of **TLS** according to best practices and high **Qualys** scores or compliance with **PCI DSS 3.1** or **NIST SP.800-52r1**; load-balanced environments, including various internal architectures and 3rd-party CDN integration (**CloudFlare**, **Incapsula**...)
  · Email servers: **exim**, **Postfix**, **Dovecot**, with proper deployment of current authentication methods for anti-spam such as **SPF** records, **DKIM** (DNS records and signing at the mailer daemon level), **DMARC** (DNS records and interpretation of the incoming reports) and **ADSP** records.
  · Database servers and various stores: **MySQL** (and forks, namely **MariaDB** and **PerconaDB**), **PostgreSQL**, **Redis**, **CouchDB**, **memcached**, etc.
  · Search servers: **ElasticSearch**, **Sphinx**, with optional content synchronization.
  · Proxies and VPN servers: **OpenVPN**, **L2TP/IPSec**, **PPTP** protocols, through various daemons which the selected distribution recommends, including multi-functional servers, such as **SoftEther**.
  · Authentication servers: **FreeRADIUS**, **OpenLDAP**
  · Type-1 hypervisors: **VMware ESXi**, **Microsoft Hyper-V**
  · Experience with **hidden services**, namely **Tor** and **I2P**, including configuring the **middleware** and securing the servers behind it not to leak personally-identifiable information.
  · Exposure to **meshnets**, such as the **Hyperboria network** on the **CJDNS**.
- Extensive experience with **cloud-service providers**, namely **Amazon AWS**, **Linode**, and **DigitalOcean**. Exposure to **Microsoft Azure** services.

o Extensive experience in **Penetration Testing** and **Reverse Engineering**
  - **Penetration Testing**
    · Up-to-date with **security bulletins**.
    · Knowledge of **Cross-Site Scripting**, **SQL Injection**, **Cross-Site Request Forgery** and similar techniques from a very young age, around **11** or so.
    · Familiarity with **Black**, **Grey** and **White Box Penetration Testing** methodologies.
    · Experience with vulnerability assessment tools **Nessus**, **Nexpose** and **OpenVAS**, including interpretation of results, vulnerability validation and elimination.
    · Extensive use of penetration testing frameworks and tools such as **Metasploit**, **nmap**, **tcpdump**, **OWASP ZAP**, **Burp Suite**, **sqlmap**, and many others.

- Familiarity with **Web Application Firewalls** and **Intrusion Detection Systems**, techniques to bypass them, and to strengthen them for different purposes.
- Notified university of being **vulnerable to OpenSSL's heartbleed bug**, approximately 5 hours after the public disclosure of the bug.
- **Reverse Engineering**
  - Experience with **disassembling**, **modifying** and **reassembling MSIL/CIL**, **Java bytecode** and **x86(-64) ASM**.
  - Familiarity with the tools and techniques used in **reverse engineering**, such as:
    - Debuggers: **WinDbg**, **OllyDbg**/**Immunity Debugger**.
    - Disassemblers/Decompilers: **IDA Pro**, **Reflector**, **JD**, amongst others.
    - Various file format/PE analysis tools, packer techniques and anti-virus evasion.
- **Screen-Scraping**
  - Strong knowledge of **regular expressions** and **XPath**.
  - Complicated setups involving scripts that circumvent anti-screen-scraping measures, even to the point of **OCR**-ing the Captcha, when it is weak enough, otherwise using the human-powered services available on the markets.
- **Search Engine Optimization**
  - **Google Panda**-tailored optimizations that have proven their legitimacy throughout the various websites I operate.
- **Version Control Systems**
  - Extensive use of **git** nowadays.
  - Previously used **SVN**.
  - Familiarity with other VCSs, such as **Mercurial** and **CVS**.
- **Continuous Integration**
  - Experience with CI through usage in open-source projects.
  - Familiarity with the set-up and usage of **Jenkins** and **Travis CI**.

○ Gained deeper insights into various fields pertaining to **Computer Science** and **Electrical Engineering**, of whose knowledge were previously vague and/or fragmented:
- Use of **Matlab** to solve problems, analyze data, draw in 2D and 3D space, and perform simulations.
- Knowledge of programming techniques, structures, algorithms and graph theory.
- Furthered knowledge in the field of relational databases, including relational algebra, database normalization, and so on.
- Intricate knowledge of the UNIX operating system, and general theories belonging to Operating Systems and computers in general.
- Introduction to the world of **Logical** and **Functional programming**, beginning from the implementation of various algorithms in a recursive way, until the implementation of more complex applications with **GUI**.
- **Analogue** and **Digital integrated circuits**, **Boole logic**, **Automata theory**, **finite-state automaton implementations** using analogue circuit elements, binary systems, decoders, (de-)multiplexers, mathematical operations, amplifiers, operational amplifiers, counters, mono-/bi-stables, registers, and memory types.
- **FPGA** programming, including implementation of state-machines using either or both **VHDL programming** and/or **schematic design** of circuit elements and gates.
- **Microcontrollers**, such as **PIC** and **Atmel** programmed in **ANSI C**. Intricate knowledge of the inner workings of a microprocessor thanks to **Microcontroller Design**, **Computer**

Architecture and **Assembly Language** classes.
- Built a fully working **CPU** in **FPGA** from scratch for a class assignment.
- Gained deeper insight into the automation of SCADA systems and networking in cars, including the **CAN** protocol and the **CAPL** programming language in a special course taught by Continental AG.
- Learned the advanced parts of **Artificial Intelligence** via the related courses, which were then later put in practice via several extracurricular projects.
- Furthered knowledge in the field of networking by embarking on a journey to study the intricacies of various **networking protocols** as an extracurricular activity, by writing a WiFi packet capture software that also analyzes the received data.

o Gave multiple presentations in classes as an extracurricular activity:
- **Facebook API and Query Language** on Database class, wherein I presented the practical uses of a database in real-life scenarios via Facebook's FQL interface.
- **The Deep Web** on Software Engineering class, where I talked about the general design of anonymous networks, including the networking and encryption parts. The presentation also included a general introduction to the deep web as we use it today on Tor, such as how anonymous payments are made, and then ended on a short live demo of browsing Tor.
- **Peer-to-Peer Systems** on Distributed Networks class, where I delved into the history of Peer-to-Peer networks, and their general structures, ending with a few case studies of important and popular P2P networks.

o When writing code, I focus on **simplicity**, **code-maintainability**, **optimization** and **security**.

Personal Skills

Communication skills gained through regular business and social interaction with clients and fellow developers. **Presentation skills** acquired by frequently giving presentations about various topics to varying audiences. **Organizational**/**Managerial skills** accumulated over time via superfluous scheduling and prioritizing to meet business and educational deadlines.

Interests

Self-teaching is an important part of my lifestyle. I try to keep up with the ever-evolving technologies of today. I constantly **try out new languages**, **new technologies** and **new practices**. I constantly go back and **re-do old projects**, with a twist, and I challenge myself to do it much better this time, by setting much higher goals. I read **research papers**, **0-day bulletins** and other sources of information that let me be ahead of the competition. I am not afraid to take initiative, and to color outside of the lines. I don't mind to get my hands dirty to try something out, let it be DoS-ing my own server in order to try out if an optimization technique or security practice did indeed work. I also watch **conference videos** (such as **DefCon**, **Black Hat**, etc) and **tech-talks** (such as **Microsoft's GoingNative**, and much more) in order to get an edge in their specific fields.

Personal Projects

RS TV Show Tracker

An **open-source** application which was born out of the need for the features it currently offers, as there were no alternative solutions for them at the time. To date, no software has so many features in this category, and as a result, under the four years it's been actively developed, its popularity has grown exponentially. Today, I single-handedly maintain it and push updates to hundreds of

thousands of users on a monthly basis.

The application was developed in **C#** with an interface in **WPF**, and was kept constantly up-to-date with the newer technologies that have been released during its development phase.

There are, as of writing this in May 2015, **135,700 active daily users** of the application, with the number of installations reaching into the **millions**. Development started in **February 2010**.

AlienSubtitles.com...

Full-stack development of the website and continued maintenance, including initial **devops responsibilities** (such as cloud server deployment on DigitalOcean) and continued **proactive monitoring** and administration of the web, database and load-balancer servers.

The website was developed in **PHP** (**HHVM**) backed by **MySQL** servers in **multi-master replication** configuration and **ElasticSearch** instances for the search feature.

At its peak, the site had **15 million unique users**, who were served by 4 geo-located and load-balanced front-end servers, attaining **100% uptime** and **under 10 millisecond response times** for all users throughout peak times. Development started in **December 2013**.

Extracurricular Projects

- **Streaming and Processing Sensor Data from Android Devices in Realtime** (*Object-Oriented Programming*)
  - This is a two-component application, which when used together can move the mouse on the computer of the user by moving a smartphone in the air.
  - AirMouse-Java-Server: Control the mouse of your PC or laptop by physically moving your phone in the air! Once an Android client connects to this Server, the streamed sensor data will be translated into up-down-left-right movements, which will be reflected upon the mouse on your computer.
  - AirMouse-Android-Client: An Android application, which streams accelerometer and/or gyroscope sensor data to a server application over the Internet or on the local network, where UDP broadcast-based service discovery is available for a faster server listing.
- **On-Demand Task Loader and Executor Unix Daemon** (*Advanced Programming Techniques*)
  - Project required the development of a UNIX daemon in C++ which would load and execute dynamically-linked libraries (.so) dropped into its monitored designated directory. It would also stop its execution and unload it if the .so file is removed, and more importantly re-initialize the library and restart the execution if the .so file is modified during its execution.
  - The library file had to be as independent as possible from the daemon application, which was successfully accomplished by giving the daemon application excellent customizability. The library file does not need to link with any component of the daemon.
  - The technologies used in the project included pthreads, inotify and signals.
- **Facebook FQL Query Tool and SQL Schema/Data Exporter** (*Databases*)
  - This is a two-part project, which first interfaced with Facebook's FQL API and provided the students with a familiar and easy-to-use interface to test their SQL skills on a real database, consisting of their Facebook data.
  - The second part of the project allows the users to export the data of their query, including an auto-generated schema, so they can import it into their RDBMS of choice and perform further analysis on the data.
- **Network Analysis and Protocol Dissection of Sniffed WiFi Traffic** (*Distributed Networks*)
  - This modular C++ project uses libpcap and the AirPcap Nx USB dongle, which is capable of sniffing the unencrypted traffic of a specified WiFi channel.
  - The raw bytestream captured WiFi traffic is then analyzed using the implemented dissectors, which are modules in the application capable of handling various protocols pertaining to various levels of the OSI layer.
  - The point of this project is educational, to gain insights into the inner workings of various protocols as

they are implemented and used on the wire. (Or, in our case, in the air.)

- o **An Untitled Minecraft-like 3D Multiplayer Game** (*Software Engineering*)
  - This project was a special group project, consisting of 5 students, who were required to do the planning, documentation and development of a 3D game (developed in C++ with OpenGL) that allowed players to play a Minecraft-like Creative game session over the internet.
  - My responsibilities were the planning, documentation and development of the networking part of the game, which, besides the protocol and the network API, also included the synchronization of player/map/game states and the insurance of a smooth gameplay. (For example, elimination of the rubber-banding effect to the best of my abilities.)
- o **Assembly Compiler for Course-made CPU on FPGA** (*Computer Architecture*)
  - As part of the Computer Architecture course, a fully working CPU was built from scratch on an FPGA. For testing, we were given .coe files in order to evaluate that our CPU is working properly, and writing an Assembler that would produce these was not part of the course.
  - This project assembles .coe files to be used with Xilinx ISE, out of .asm files which use a custom Intel-like syntax, with added MASM-like syntax sugar, such as #if, #while and #for macros. The assembler was written in C++.
- o **Market Prediction using Neural Networks** (*Artificial Intelligence*)
  - The scope of this classroom assignment was to analyze stock and currency data while examining the potential to accurately forecast indices by using neural networks and genetic algorithms.
  - The developed application can learn and predict the data itself or technical indicators used in trading with a very high accuracy, depending on the training set size and requested prediction count.