

Sami J. Lehtinen

sjl@iki.fi 
+358 40 8643001 
<http://iki.fi/~sjl/> 
@sjlehtin 
<https://github.com/sjlehtin> 
<https://www.linkedin.com/in/sjlehtin> 

Problem solver, systems designer, hacker, programmer, leader.

Since you are reading this, it is a safe assumption you or your customer has a problem that needs solving.

If the solution has already been attempted, and perhaps it is already in use, you can sleep easier by letting me take a look. I can audit your existing solution for security problems in the implementation, protocols used, or other practicalities in the implementation.

If you are just starting out, but do not have the expertise or the experience, I can help you design the solution.

Delivering software with consistent quality is hard and requires diligence and attention to detail every step of the way. I can help you build a software delivery process and system which will make sure a bug will only need to be fixed once in your product.

I am a jack-of-all-trades, capable of tackling pretty much any engineering problem in software development, with an emphasis in software quality and testing, secure coding, and security protocols. In my various roles during my long career with my past employer(s), I have done everything that is related to releasing software products, for example, requirements engineering and product ownership, customer communications, quality assurance, architecting, and implementation.

I have had the pleasure of working with many different kinds of systems, large and small, with completely different requirements as to availability, performance, features, and other aspects of the software.

I have lead teams both as a supervisor and as a hands-on architect. (Tectia Client/Server, CryptoAuditor)

As a technical lead or software architect, I was involved in the design of more projects I can now remember. In the project teams I directly reported to, I designed or reviewed most of the major components.

I have implemented both SSH and TLS. I've maintained multiple implementations of both protocols, sometimes over a decade, with very long continued customer relationships. (ssh2, sshg3, NIO TLS)

I've participated in the design of multiple protocols, security and application alike. The best documented of these is the SSH protocol suite, which I had a part in. My biggest contribution was creating a reference implementation of new protocol features, but I also contributed in the protocol design work itself.

I have built user interfaces both with native tools (WPF, Qt) and over the web (HTML5, CSS, Ajax). (Tectia configurator, CryptoAuditor, hobby projects)

I write code for both fun and profit. As a professional, it is my responsibility to learn new things and bring the useful learnings to the projects I'm involved in.

I am an M.Sc. from Aalto University, School of Science. I graduated in 2017. My thesis was about anomaly detection in interception proxies. I did my B.Sc. thesis about the edge-coloring problem.

Systems design On-the-spot problem analysis, rough design sketches, design validation, and rough estimates; these and more on the way to the coffee machine and back.

System implementation

Programming languages I'm most familiar with the usual suspects, Python, C/C++, C#, Javascript, with a smidgeon of Java on the side. I like to learn new languages and programming paradigms, and I know how to program, period.

Methodologies Test-driven development and friends are useful methodologies and invariably lead to better-quality code, lower defect rate, and better work-life balance.

Secure coding Secure coding practices, defensive coding, and input validation should be observed in any code facing the internet. Writing CVE entries about your own product is not fun, and even less fun is answering customer enquiries about them.

Embedded systems Experience mostly tangential during my career; although there have been projects where protocol implementations have been ported as SDKs for various embedded systems. Recently I have tinkered with Arduino in hobby projects.

Software configuration management

Version control Git, Mercurial, Subversion. I prefer Git, as although it can be a bit inconsistent in the command-line interface, it gets the job done efficiently and safely.

Continuous integration No team can say they are agile without having a CI system in place. I've administered BuildBot and Jenkins.

Review processes Using a pull request type process reduces code defects by making reviews a natural part of the software development.

Machine learning I'm very interested in working with machine learning. Current experience is mostly from my master's thesis and various school projects; I would really like to get hands-on with a real project.