

Santos Gallegos - Curriculum Vitae



Hi there! I'm Santos Gallegos, I live in Ecuador. I'm passionate about web development, algorithms, Python, Rust, Neovim, cybersecurity, and free and open source software. I'm always eager to learn new things, when I'm not working I like to read, watch movies, draw and paint.

Overview

I'm currently working at Read the Docs as a backend developer and security engineer, in my free time I also work as a freelancer Rust developer, and pentester for several companies.

I have participated in several programming competitions, CTFs, and hackathons, I actively contribute to open source projects, and report security vulnerabilities.

I also help to organize events in my local Python community, Python Ecuador, and have given several talks about Python, open source, and cybersecurity.

You can find a more detailed description of my professional experience, and skills below.

Contact information

I prefer to be contacted by email, but you can also find me on LinkedIn.

- **Email:** stsewd@proton.me
- **LinkedIn:** <https://www.linkedin.com/in/stsewd/>

Languages

Languages I speak and prefer to communicate in:

- English (fluent)

- Spanish (native)

Table of contents

- Professional experience
- Competitions
- Talks and events
- Technical knowledge and skills
- Publicly disclosed security vulnerabilities
- Open source projects and contributions
- Find me on the Internet

Professional experience

- **Read the Docs, Inc:** Software developer and security engineer (2018 - present)

During my time at Read the Docs I've worked on several projects and touched many parts of the codebase. Mainly as a backend developer using Django, but also a bit of frontend, some of the things I've worked on are:

- Support for searching all documentation pages and per-user access using Elasticsearch and Django REST Framework
- Configuration file parsing and validation
- Documentation serving and caching using nginx, S3, and Cloudflare
- Building and designing APIs using Django REST Framework
- Optimizing and improving the performance of the application
- Developed and maintained Sphinx extensions
- Custom domains support using Cloudflare
- Custom authentication and authorization for documentation pages using tokens and passwords
- Webhooks and automation rules for build events
- User and organization security logs
- User invitations to organizations and projects
- Found and fixed several security vulnerabilities in the codebase
- I worked on consolidating our code base from private and public repositories into one

- **Freelancer Rust developer:** (2022 - present)

During recent years I've been learning Rust, and in

2022 I had the opportunity to work as a freelancer on a transport management system for a German company, using AWS services like DynamoDB, Lambdas, and S3.

- **Cooperativa CB:** Computer security audit (2022)
Penetration testing of their web application, mobile application, and core banking system.
- **Telconet S.A:** private bug bounty program (2022)
After participating in several CTFs, I was invited to participate in a private bug bounty program by Telconet, during the program I found and reported several security vulnerabilities.
- **Impodirect, CIA LTDA:** Computer security audit (2021 - 2022)
Penetration testing of their web application, and servers.
- **Cooperativa CB:** Computer security audit (2020)
Penetration testing of their web application, servers, and core banking system.

Competitions

From time to time I like to participate in programming competitions, hackathons, and CTFs. Here are some of the big ones I've participated in:

- **Devsu Code Jam:** 7th place, professionals category (2019)
Devsu Code Jam is a programming competition organized by Devsu, where students and professionals compete individually to solve a set of programming/algorithmic problems.
The contest had two rounds and two categories (students and professionals), the first round was online and the second round was in person. Out of 1760 participants from the online round, 100 were selected to participate in the in-person round (50 of each category). I got the 7th place in the professionals category.
- **IEEEExtreme Programming 11.0:** 1st place nationally (trivialbox team) (2017)

IEEEExtreme programming is a 24-hour global programming competition, where teams of up to 3 students from different universities compete to solve a set of programming/algorithmic problems. This was my third and last time (since I'm not in school anymore) participating in this competition, we got the 1st place out of all the teams in Ecuador.

- **IEEEExtreme Programming 10.0:** 1st place nationally (trivialbox team) (2016)

IEEEExtreme programming is a 24-hour global programming competition, where teams of up to 3 students from different universities compete to solve a set of programming/algorithmic problems. This was my second time participating in this competition, we got the 1st place out of all the teams in Ecuador.

- **Rally Latinoamericano de Innovación:** 1st place locally (Cuenca) and nationally (Ecuador) in the innovation category (Atuk Maskhay team) (2016)

Hackathon to promote open innovation and entrepreneurship in students, where teams of up to 10 students and mentors compete to present an innovative project in 28 hours. My team got the 1st place in the innovation category out of all the teams in Cuenca and Ecuador. We presented a project to build sustainable houses using recycled plastic bottles.

- **Hackaton UPS:** 2nd place (trivialbox team) (2016)

Hackathon organized in 2019 by Universidad Politécnica Salesiana, where teams of up to 4 students compete to present an innovative project in 24 hours. This was my first time participating in a hackathon, we got the 2nd place out of all the teams, we presented a project to help connect people that need help (like crossing the street) with people close to them that can help them.

- **IEEEExtreme Programming 9.0:** 3rd place nationally (EnigmaT team) (2015)

IEEEExtreme programming is a 24-hour global programming competition, where teams of up to 3 students from different universities compete to solve a set of programming/algorithmic problems. This was my first time participating with a team of friends, we got the 3rd place out of all the teams in Ecuador.

Talks and events

I help to organize several events in my local community:

- Python Ecuador meetups
- Hacktoberfest Cuenca
- Django Girls Cuenca

I've given several talks in local meetups and events (mostly in Spanish):

- Ciclo de vida de un proyecto FLOSS - FFLISoL Quito y Loja 2023
- Introducción al software libre y de código abierto - Hacktoberfest Cuenca 2022 and 2018
- Impulsa tu carrera con Open Source - JavaScript Ecuador meetup 2021
- Search the Docs - Elastic Search at Read the Docs - Elastic San Diego meetup 2021
- Escribiendo Documentación con Sphinx - FLISol Ecuador Online 2020
- Resolución de problemas y competencias de programación - Universidad de Cuenca 2019
- Open Source no es gratis. Porqué debería importarte - FLISol Cuenca 2019
- Viajando en el tiempo con Git - Python Ecuador meetup 2018
- Introducción a Neovim - Python Ecuador meetup 2018
- Comprensión de listas - Python Ecuador meetup 2017

Technical knowledge and skills

This is a non-exhaustive list of the technologies/tools/skills I actively use and have experience with:

- Programming languages
 - Python
 - Rust
 - C
 - JavaScript
 - Lua
 - Bash
- General tools
 - AWS
 - S3
 - Linux
 - Git
 - GitHub

- GitLab
- Vim/Neovim
- General skills
 - Free and Open Source Software
 - Open Source maintainer
 - Cybersecurity
 - Pentesting
 - Algorithms
 - Data structures
 - Problem solving
 - Command line
 - Bash scripting
 - Web development
- Web frameworks
 - Django
 - Django REST Framework
 - Celery
- Databases
 - PostgreSQL
 - SQLite
 - Redis
 - DynamoDB
 - Elasticsearch
- CI/CD
 - Docker
 - GitLab CI
 - GitHub Actions
 - CircleCI
- Testing
 - pytest
 - unittest
- Documentation
 - Sphinx
 - reStructuredText
 - Markdown
- Pentesting
 - OWASP ZAP

Publicly disclosed security vulnerabilities

Over the years, I have responsibly reported security vulnerabilities; some have been made public.

- **Untrusted search path on Windows systems leading to arbitrary code execution in GitPython: GHSA-wfm5-v35h-vwf4 (2023)**

This vulnerability could have allowed a malicious user to execute arbitrary code on Windows systems by tricking the user into running GitPython from an untrusted directory or repository.

- **Blind local file inclusion in GitPython: GHSA-cwvm-v4w8-q58c (2023)**

This vulnerability could have allowed a malicious user to make GitPython read arbitrary files from the filesystem, this can be used to check if a file exists or not, or cause a denial of service by reading a large file.

- **Arbitrary write to files from builder server in Read the Docs: GHSA-v7x4-rhpg-3p2r (2023)**

This vulnerability could have allowed a malicious user to write to any files that the application has write access to. The contents that can be written aren't fully controlled by the attacker, so the impact of this attack is limited.

- **Write access to projects via API V2 for any logged-in user in Read the Docs: GHSA-rqfv-8rrx-prmh (2023)**

This vulnerability could have allowed a malicious user to use the API v2 to create, update, and delete any projects on the community version of Read the Docs. On Read the Docs for Business the malicious user needs to belong to the organization of the target project in order to exploit this vulnerability (to any team of the organization with at least read only access).

- **CAS session hijacking in Read the Docs for Businesses: GHSA-4mgr-vrh5-hj8q (2023)**

This vulnerability could have allowed an attacker to hijack a specific type of user session, given a crafted URL. Exploiting this vulnerability would have required the user to follow a malicious link, allowing the attacker to steal the CAS ticket used to authenticate the user towards several read-only APIs.

- **Serving content from pull requests previews on main docs domains in Read the Docs: GHSA-h4cf-8gv8-4chf (2023)**

This vulnerability could have allowed a malicious user to serve arbitrary content under the main domain of documentation sites.

- **Cache poisoning: serving arbitrary content on documentation sites in Read the Docs GHSA-mp38-vprc-7hf5 (2023)**

This vulnerability could have allowed a malicious user to serve arbitrary content on documentation sites they don't own.

- **Arbitrary code execution when using treesitter with injections in Neovim: GHSA-6f9m-hj8h-xjgj (2023)**
- **Path traversal: access to files from any project in Read the Docs: GHSA-5w8m-r7jm-mhp9 (2023)**

This vulnerability could have allowed a malicious user to access documentation files from any project given its slug, regardless of user permissions. This was possible using a crafted URL that can bypass the path traversal protections.

- **Symlink following: arbitrary file access from builder server in Read the Docs: GHSA-hqwg-gjqw-h5wg (2023)**

This vulnerability could have allowed a malicious user to access any files that the application has read access to. Exploiting this vulnerability requires creating symlinks that pointed to files outside a project root.

- **Cache poisoning in Read the Docs: GHSA-7fcx-wwr3-99jv (2023)**

This vulnerability could have allowed a malicious user to get some private information about a user account like: username, name, and audit logs.

- **Arbitrary command execution in simple-git: CVE-2022-25860 (2022)**

- **Symlink following: arbitrary file access from builder server in Read the Docs: GHSA-368m-86q9-m99w (2022)**

This vulnerability could have allowed a malicious user to access any files that the application has read access to. Exploiting this vulnerability requires creating symlinks that pointed to files outside a project root.

- **Allow serving of arbitrary HTML files from the main domain in Read the Docs: GHSA-98pf-gfh3-x3mp (2022)**

This vulnerability could have allowed a malicious user to serve arbitrary HTML files from the main application domain (readthedocs.org/readthedocs.com).

- **CSRF from documentation domains in Read the Docs: GHSA-3v5m-qmm9-3c6c (2021)**

This could have allowed a malicious user to fetch internal and private information from a logged user in `readthedocs.org/readthedocs.com` by creating a malicious site hosted on `readthedocs.io/readthedocs-hosted.com` or from any custom domain registered in the platform.

- **Serving arbitrary files in domains of other projects in Read the Docs: <https://docs.readthedocs.io/en/stable/security.html#release-2-3-0> (2018)**

This vulnerability could have allowed a malicious user to serve arbitrary files in other projects domains by adding a project as translation of the target project.

Open source projects and contributions

A list of some of the open source projects I've created, help to maintain, or made substantial contributions to.

- **Read the Docs**

Read the Docs is a documentation hosting platform. I'm core developer and currently working for the company.

- **GitPython**

Python library used to interact with Git repositories. I have contributed bug fixes and helped to patch a security vulnerability.

See all contributions.

- **nvim-treesitter**

Integration of the tree-sitter parsing library for Neovim. I was an early contributor and core maintainer, helped to add support for new languages and improve features.

See all contributions.

- **tree-sitter-query**

Query grammar for tree-sitter, I have contributed adding new features and fixing bugs.

See all contributions

- **tree-sitter-python**

Python grammar for tree-sitter, I have contributed adding new features and fixing bugs.

See all contributions.

- **Python Ecuador**

Main website of the Python Ecuador community. I'm core maintainer and helped to build the website.

- **rstcheck**

Linter for reStructuredText. I have contributed adding new features and fixing bugs.

See all contributions

- **nox**

Flexible test automation for Python, alternative to tox. I was an early contributor and helped to add support for new features and fix bugs.

See all contributions.

- **fzf-checkout.vim**

Vim plugin to manage Git branches and tags with fzf. I'm the author and maintainer.

- **gx-extended.vim**

Vim plugin for extending the gx mapping for opening URLs, files, and more. I'm the author and maintainer.

- **sphinx.nvim**

Sphinx integration for Neovim. I'm the author and maintainer.

- **spotify.nvim**

Spotify integration for Neovim. I'm the author and maintainer.

- **treesitter-rst**

reStructuredText grammar for tree-sitter. I'm the author and maintainer.

- **treesitter-comment**

Tree-sitter grammar for comment annotations. I'm the author and maintainer.

- **ieee-pandoc-template**

Pandoc template for IEEE papers. I'm the author and maintainer.

Find me on the Internet

- **GitHub**: <https://github.com/stsewd>
- **GitLab**: <https://gitlab.com/stsewd>
- **Hackerrank**: <https://www.hackerrank.com/stsewd>
- **HackerEarth**: <https://www.hackerearth.com/@stsewd>
- **Stack Overflow**: <https://stackoverflow.com/users/5689214/>
- **LinkedIn**: <https://www.linkedin.com/in/stsewd/>
- **Personal blog**: <https://stsewd.dev>
- **Email**: stsewd@proton.me (I prefer to be contacted by email)