

Santos Gallegos

Software and Application Security Engineer



Hi there! I'm Santos Gallegos, I'm from Ecuador 🇪🇨. I'm passionate about web development, algorithms, Python, 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 application security engineer. I have participated in several programming competitions, CTFs, and hackathons, I actively contribute to open source projects, and report security vulnerabilities for fun and in bug bounty programs.

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

Below you can find a more detailed description of my professional experience, and skills.

The most up to date (and long) version of this document can be found at <https://stsewd.dev/cv/>.



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 am proficient 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 and application 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, and as a application security engineer. Some of the things I've worked on are:

- Improving the search experience with custom rankings, filtering syntax, and a new API 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
- Improved the redirects feature allowing wildcards, explicit ordering and more
- Consolidating our code base from private and public repositories into one
- Added support for SAML authentication
- Replacing the old GitHub OAuth app integration with a GitHub App
- Migrating the custom stripe integration to use dj-stripe

- File tree diff, to show the changes between documentation versions
- **Bug Bounty Hunter:** Find and report security vulnerabilities to several companies (2024 - present)

I participate in several bug bounty programs (mainly on [Hackerone](#)), where I've found and reported several security vulnerabilities to several companies.

- **CB Cooperativa:** Computer security audit (2023)

Penetration testing of their web application, mobile application, and core banking system.

- **Freelancer Rust developer:** (2022 - 2023)

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.

- **CB Cooperativa:** 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, I've found and reported several security vulnerabilities during this time.

- **Impodirect, CIA LTDA:** Computer security audit (2021 - 2022)

Penetration testing of their web application, and servers.

- **CB Cooperativa:** Computer security audit (2020)

Penetration testing of their web application, servers, and core banking system.

Competitions

Occasionally I like to participate in programming competitions, hackathons, and CTFs. Here are some of the main 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 participating in this competition (since I'm not in school anymore), 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 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, including:

- [Python Ecuador meetups](#)
- [Hacktoberfest Cuenca](#)
- [Django Girls Cuenca](#)

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

- [Content Security Policy \(CSP\): ¿El fin de XSS?](#) - pwn or die, Quito (2025)
- [Seguridad en aplicaciones Python de un developer a otro](#) - PyDay Piura, online (2024)
- [Documenta tu proyecto con herramientas open source](#) - FLISoL Quito, Quito (2024)
- [Reportando vulnerabilidades en proyectos Open Source](#) - Día de la Seguridad Informática, Cuenca (2023)
- [Ciclo de vida de un proyecto FLOSS](#) - FLISoL Quito and 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 or 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
 - Process automation
 - Web development
- **Web frameworks**
 - Django
 - Django REST Framework
 - Celery
- **Databases**
 - PostgreSQL
 - SQLite
 - Redis
 - DynamoDB
 - Elasticsearch
- **CI/CD**
 - Docker
 - GitLab CI
 - GitHub Actions
 - CircleCI
 - Codecov
- **Testing**
 - pytest

- unittest
- coverage
- **Documentation**
 - Sphinx
 - reStructuredText
 - Markdown
- **Cybersecurity**
 - OWASP ZAP
 - Bug bounty
 - Pentesting
 - Source code auditing
 - Security best practices
 - Secure coding
 - Web security
 - Vulnerability reporting and disclosure

Publicly disclosed security vulnerabilities

I have responsibly reported several security vulnerabilities; some of the public ones are listed below.

You can find the full list of vulnerabilities I've reported at <https://stsewd.dev/disclosed-vulnerabilities/>.

- **Write access to GitHub repositories using deploy key in Read the Docs for Business** - [GHSA-jqm9-f79c-8wx6](#) (2025)

This vulnerability could have allowed an attacker to gain write access to a GitHub repository if the repository was connected to a project in Read the Docs.

- **Email verification bypass in Read the Docs** - [GHSA-h73w-m588-h9r6](#) (2025)

This vulnerability could have allowed an attacker to mark any email address as verified.

- **XSS in Read the Docs** - [GHSA-gg7x-cc2v-mcp3](#) (2025)

This vulnerability could have allowed an attacker to execute arbitrary JavaScript code on the dashboard by performing an action that created a notification with a malicious payload.

- **XSS in Read the Docs for Business** - [GHSA-v4vm-q9fw-x237](#) (2024)

This vulnerability could have allowed an attacker to execute arbitrary JavaScript code on the dashboard of Read the Docs for Business when a user visited a page and clicked on a malicious link.

- **XSS in django-gravatar2** - [Fixed in 1.4.5](#) (2024)

This vulnerability could have allowed an attacker to execute arbitrary JavaScript code on any where the gravatar template tag was used with user input as alt text.

- **Stored XSS via integration of external issues in Sentry** - [CVE-2024-41656](#) (2024)

An unsanitized payload sent by an integration platform allows the storage of arbitrary HTML tags on the Sentry side. This payload could subsequently be rendered on the issues page. This vulnerability might lead to the execution of arbitrary scripts in the context of a user's session.

- **XSS in django-allauth** - [Fixed in 0.63.6](#) (2024)

This vulnerability could have allowed an attacker to execute arbitrary JavaScript code on any page that listed the available social integrations (like the login page) when a user visited a malicious URL.

- **XSS in django-impersonate** - Fixed in [1.9.3](#) and [1.9.4](#) (2024)

This vulnerability could have allowed an attacker to execute arbitrary JavaScript code on the listing page of the impersonate app when a user visited a malicious URL.

- **Login CSRF in the SAML implementation of django-allauth** - [Fixed in 0.63.3](#) (2024)

This vulnerability could have allowed an attacker to trick a user into logging in as the attacker in a SAML authentication flow.

- **Cross site scripting on beta dashboard in Read the Docs** - [GHSA-8v7c-r4x6-h796](#) (2024)

This vulnerability could have allowed an attacker to execute arbitrary JavaScript code on the beta dashboard of Read the Docs when a user visited a build page with the malicious payload.

- **Denial of service via regular expression in Django Wiki** - [CVE-2024-28865](#) (2024)

This vulnerability could have allowed an attacker to cause a denial of service condition by creating a malicious crafted article that would take a long time to process.

- **CAS session takeover in Read the Docs for Business** - [GHSA-pw32-ffxw-68rh](#) (2024)

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 have the pull requests previews feature enabled, and follow a malicious link, allowing the attacker to steal the CAS ticket used to authenticate the user towards several read-only APIs.

- **XSS in search integrations when including search results from malicious projects in Read the Docs** - [GHSA-qhgx-5j25-rv48](#) (2024)

This vulnerability could have allowed a malicious user to execute arbitrary JavaScript code on the search results page of any project that included search results from a malicious project. This vulnerability was present in three different search integrations provided by Read the Docs.

- **Creation of integrations for any project in Read the Docs** - [GHSA-45hq-g76r-46wv](#) (2023)

This vulnerability could allowed a malicious user to create integrations that don't require payload validation for any project. This integration could have been used to trigger builds to existing versions, create external versions, and update the identifier of the project's default version under some circumstances.

- **Arbitrary command execution on Windows in Vim** - [CVE-2023-4736](#) (2023)

This vulnerability could have allowed a malicious user to execute arbitrary code on Windows systems by tricking the user into opening a file with Vim from an untrusted directory.

- **Untrusted search path on Windows systems leading to arbitrary code execution in GitPython** - [CVE-2023-40590](#) (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** - [CVE-2023-41040](#) (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-xgj](#) (2023)

This vulnerability could have allowed an attacker to execute arbitrary code by tricking the user into opening a specially crafted file with Neovim.

- **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** - [2.3.0 release](#) (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** - core developer

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

- **GitPython** - contributor

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** - contributor and ex-core maintainer

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** - contributor

Query grammar for tree-sitter, I have contributed adding new features and fixing bugs. [See all contributions.](#)

- **tree-sitter-python** - contributor

Python grammar for tree-sitter, I have contributed adding new features and fixing bugs. [See all contributions.](#)

- **Python Ecuador** - author and core maintainer

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

- **rstcheck** - contributor

Linters for reStructuredText. I have contributed adding new features and fixing bugs. [See all contributions.](#)

- **nox** - contributor and ex-core maintainer

Flexible test automation for Python, alternative to tox. I was an early contributor and helped to add new features and fix bugs. [See all contributions.](#)

- **fzf-checkout.vim** - author

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

- **gx-extended.vim** - author

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

- **sphinx.nvim** - author

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

- **spotify.nvim** - author

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

- **tree-sitter-rst** - author

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

- **tree-sitter-comment** - author

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

- **ieee-pandoc-template** - author

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

Find me on the Internet

- **GitHub:** <https://github.com/stsewd>
- **Hackerone:** <https://hackerone.com/stsewd>
- **LinkedIn:** <https://www.linkedin.com/in/stsewd/>
- **Personal blog:** <https://stsewd.dev>
- **Email:** stsewd@proton.me (*I prefer to be contacted by email*)