STANFORD SECURITY CLINIC

Lessons from Running a Product Security Clinic

Aditya Saligrama and Joey Holtzman

BSidesSF 2025 | April 27, 2025

Aditya Saligrama

Stanford ‘24 → Senior Software Engineer, Formal

Joey Holtzman

Stanford ‘27, Applied Cyber, OSCP

Prior art: cybersecurity clinics

- Many universities have started clinics over the last ~8 years – often as formal courses
- Focused on improving the security posture of organizations local to a university
  - Both private and public sector
- Usually corporate security focused

Public interest cybersecurity

Entrepreneurial environment

Stanford community spawns many startups

Young founders

Students often don’t take security courses before building

Stanford Security Clinic

Helping students build securely without a formal security education

Alas, we've done this before

- We have a history of **identifying security flaws** in student startups
- One even **threatened us with felony charges** in order to keep us quiet!
- Better to be **proactive** — let's have the startups come to us.

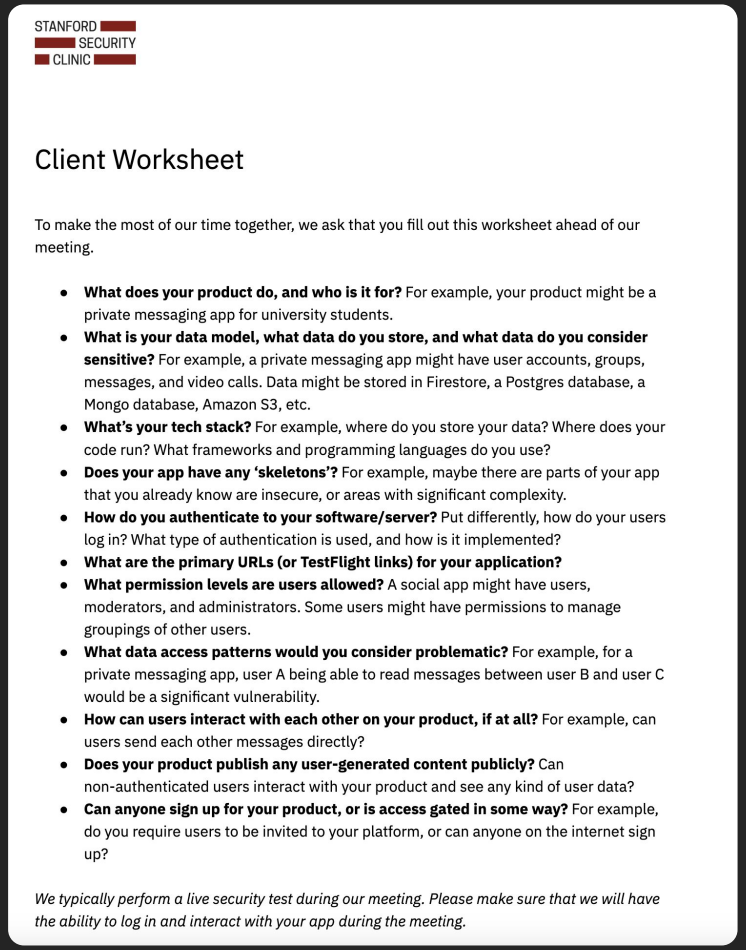
Common vulnerabilities we see

- BaaS misconfigurations: Firebase, Supabase
  - “client makes arbitrary queries” threat model
- API Authentication/Authorization confusion
- Unprotected GraphQL
- Other basic API security issues
  - IDOR, input sanitization errors

Structure of the SSC

A typical visit to the Clinic

1. Client fills out ‘intake worksheet’ + legal docs ahead of time
2. Application walkthrough
3. Threat modeling + infrastructure review
4. Data security consult
5. Active penetration test
6. Debrief + share a write up



STANFORD SECURITY CLINIC

Client Worksheet

To make the most of our time together, we ask that you fill out this worksheet ahead of our meeting.

- **What does your product do, and who is it for?** For example, your product might be a private messaging app for university students.
- **What is your data model, what data do you store, and what data do you consider sensitive?** For example, a private messaging app might have user accounts, groups, messages, and video calls. Data might be stored in Firestore, a Postgres database, a Mongo database, Amazon S3, etc.
- **What’s your tech stack?** For example, where do you store your data? Where does your code run? What frameworks and programming languages do you use?
- **Does your app have any ‘skeletons’?** For example, maybe there are parts of your app that you already know are insecure, or areas with significant complexity.
- **How do you authenticate to your software/server?** Put differently, how do your users log in? What type of authentication is used, and how is it implemented?
- **What are the primary URLs (or TestFlight links) for your application?**
- **What permission levels are users allowed?** A social app might have users, moderators, and administrators. Some users might have permissions to manage groupings of other users.
- **What data access patterns would you consider problematic?** For example, for a private messaging app, user A being able to read messages between user B and user C would be a significant vulnerability.
- **How can users interact with each other on your product, if at all?** For example, can users send each other messages directly?
- **Does your product publish any user-generated content publicly?** Can non-authenticated users interact with your product and see any kind of user data?
- **Can anyone sign up for your product, or is access gated in some way?** For example, do you require users to be invited to your platform, or can anyone on the internet sign up?

We typically perform a live security test during our meeting. Please make sure that we will have the ability to log in and interact with your app during the meeting.

Threat modeling & infrastructure review

- Discuss where clients may **most likely be targeted**
  - We focus the engagement specifically on these areas
- Review **cloud configurations** and **logging systems**
- Explore how clients can **minimize risk**

Data security consult

- Talk through **what** and **how** data is being stored
- Analyze what the communication between the users, backend, and database looks like
- Advise clients on how to **protect sensitive data**

Active security testing

- **Attempt to find vulnerabilities** in the client's product
- **Verify** that the product's current security does what the clients think
- **Recommend remediations** for all vulnerabilities we find

Success stories

The Clinic has been busy!

- The clinic has seen **many different startups** since it started.
- We identified **critical security flaws** in **most student startups** we met with.
- Three main kinds of startups we saw:
  - “Public interest” tech
  - B2B applied AI
  - Fintech, biotech

Beyond startups

- Recently, we've also done engagements with a Stanford internal development/infra team
  - Evaluated their websites' security
  - Discussed incident response plans
  - Recommended ways to mitigate against DDoS and script kiddie attacks

Some vulnerabilities we caught

- Client-side-only access control
  - Access admin dashboards without auth
- AWS API keys **exposed** in frontend
  - This was **actively** being exploited!
  - We led an incident response
- **RCE** through command injection
- **DoS** via an SSRF-able endpoint

Happy client after we identified a critical security vulnerability in his app!

Some vulnerabilities we caught

- Could have **wiped** one company with a single web request
- OpenAI API keys **exposed** in the frontend
- Accessing “VPN gated” admin panel by spoofing X-Forwarded-For IP header

Lessons learned

What worked well

- Clients fill out a worksheet and mutual NDA **before** engagements
  - We spend **less time learning** about their product and **more time testing** their security

- Clients create test accounts/data beforehand
  - Clients were generally good about making sure we tested against a staging environment rather than prod

Tips for working with early-stage startups

- Develop **automated security evaluation** for commonly seen software
  - Supabase, GraphQL, etc.
- Refine **screening methods** to accurately identify the **needs** of clients
  - Needs depend on the domain and maturity of their product
- Emphasize that **early stage startups are targets**!
  - e.g. startup AWS accounts are ripe cryptomining vehicles

STANFORD SECURITY CLINIC

Questions?

Aditya: aditya@saligrama.io
Joey: jholtz72@stanford.edu