Pentest a Payment Provider

- Team: sdmay21-06
- Email: sdmay21-06@iastate.edu
- Client: Dwolla
- Advisor: Benjamin Blakely

Introduction

Project Statement

- Research and develop an **exhaustive testing methodology**
- Perform a comprehensive penetration test for the **Payment Provider Dwolla**
- Document and **report discovered vulnerabilities**

Purpose

- Secure client systems and services
- Protect sensitive data and customer information
- Create trusted environment for money exchanges
- Follow OWASP, NIST, and PTES security standards for web and API applications

Project Discrepancies

Expectation

- Produce an end product for users
- Clearly defined specifications, requirements, and deliverables
- Provided templates and project requirements designed around a final product
- Unit testing, product development, prototyping, delivery, user manual
- System diagrams, maps, breakdowns

Reality

- Providing a service
- Scope and Rules of Engagement are the specifications and project requirements
- Providing a service does not fit the mold of the given templates
- There is no product development or testing, simply an attack framework and narrative
- System diagrams are just network infrastructure maps

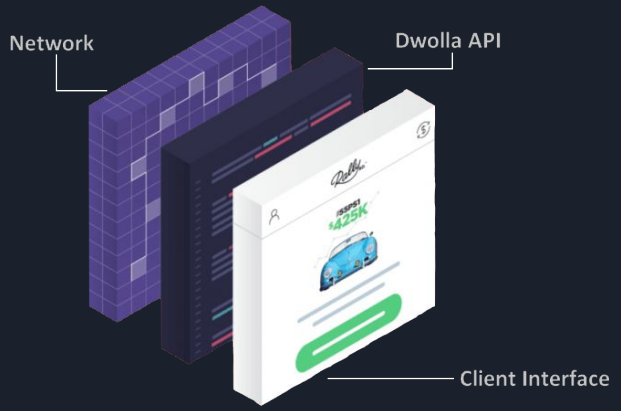
Compromises are made along the way to fit our approach with the expectations

Scope

Clearly defined area of operation for practical testing.

- Sandbox Environment
- Web Dashboard
- Public API
- Production Environment

Infrastructure



Rules of Engagement

Best Practices

Report critical breaches immediately
NIST, OWASP, PTES
Maintain ethical practices
Use tools and services properly
Assess intended and unintended uses

Scope Awareness

Remain within the scope
Avoid lasting damage
Report leaks to/from production environment

Reporting

Record each vulnerability
Assign risk severity
Document reproduction steps
Professional documentation

Resource Management

Avoid denial of service
Minimize crawlers and resource hogs
Abstain from damaging resources

Resource Requirements

Virtual Machines

Virtual Machines allow you to run an operating system in an app on your personal desktop

During testing we utilized Kali VMs to conduct our exercises

Pentest Tools

Pentest tools will help us scan, discover, and exploit vulnerabilities. Some of the tools we used are:

- OWASP-Zap
- Postman
- Intruder.io
- Burp Suite
- Firefox dev tools
- Google dorking

Dwolla Specifics

Dwolla sandbox account

Python SDK

API Interface

Other Requirements

NDA	It is a non-disclosure agreement with Dwolla to exercise reasonable care to prevent the disclosure of the Confidential Information to any third party and to only use this information for the educational purpose for which it is shared.
Risk	Address the associated risk to Dwolla systems and operations per unique vulnerability.
Standards	NIST acts as a framework for the attack narrative, methodology, and reporting OWASP Top-10 is a breakdown of the most common and severe vulnerabilities associated with either the Web App or API PTES is a comprehensive standard on the complete pentesting process
Validation	Validate all discovered vulnerabilities for accuracy and reproduction. Furthermore, perform any requested validation against potentially implemented patching.

Deliverables

Executive Summary

Serves to provide a high level overview of the penetration test

Information covered:

- Purpose of penetration test
- General findings
- Weakness source breakdown
- Remediation roadmap

Aimed to give non-technical staff at Dwolla insight regarding security concerns from a business-oriented standpoint

Technical Report

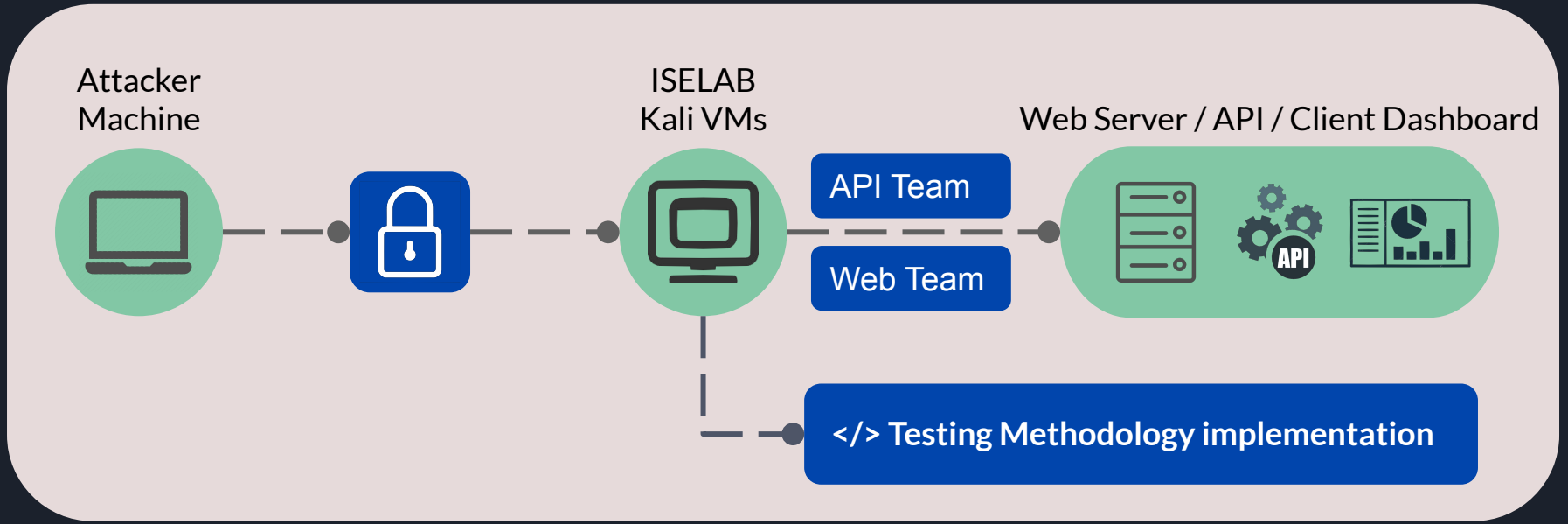
Serves to provide technical details from the penetration test

Information Covered:

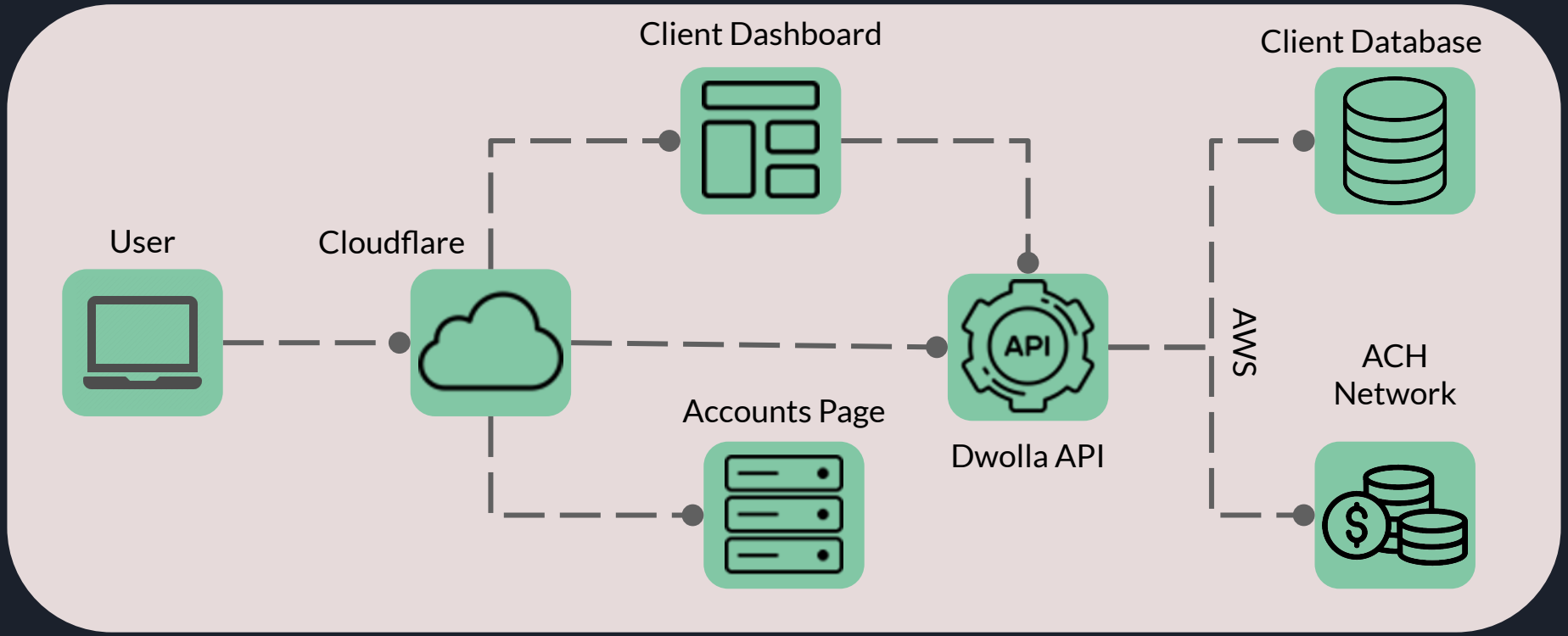
- Vulnerability risk assessment
- Exploitation validation
- Tools and techniques used
- Technical impact on systems

Provides Dwolla's technical staff in-depth technical details, making understanding and patching vulnerabilities easier.

Testing Process



Network Analysis



Methodology

Scope

The scope covers only the Sandbox environment. The methodology is developed around this topology.

Rules of Engagement

Our Rules of Engagement dictate how we conduct our tests. The methodology has been molded around these rules.

Vulnerabilities

Identifying potential vulnerabilities will be done using several methods.

- Targeted vulnerability testing will be done by cross referencing OWASP's top ten web application vulnerabilities and top ten API vulnerabilities.
- Conducted some lightweight scans with tools such as ZAP and Intruder.
- Dwolla SDK will allow us to test the Dwolla API by forming custom API requests and testing for expected output.

References: API, https://dashboard-sandbox.dwolla.com, https://accounts-sandbox.dwolla.com

Unit Testing

Adapted for a service rather than product

API Team
Common vulns: Broken authorization, data exposure, improper property assignment, function-level auth.
Tools: OWASP-ZAP, Postman, SDKs, Firefox dev tools

Web-App Team
Common vulns: Injection, XSS, misconfigurations, broken authorization, poor logic
Tools: OWASP-ZAP, Burp Suite, intruder.io

Production

Outside of scope

Assess for links between Sandbox and Production

Report leaks immediately

Challenges

Cloudflare	Cloudflare WAF was fronting the webapp which resulted in the automated blocking of much of our testing
ETG VM	Testing through the VM became redundant. There was a bottleneck for processing power and it was more realistic to test from our own machines
Risk assessment	Assigning meaningful severities to the vulnerabilities found and proper risk assessment was hard to quantify
Budget	Lack of access to paid tools, had to automate a bunch of little tests.

Risk Identification

Critical

Would completely undermine the integrity of either Dwolla or their clients on top of catastrophic financial or information loss

High

Would lead to a substantial financial or information loss for Dwolla

Moderate

Would lead to a minor financial or information loss for Dwolla

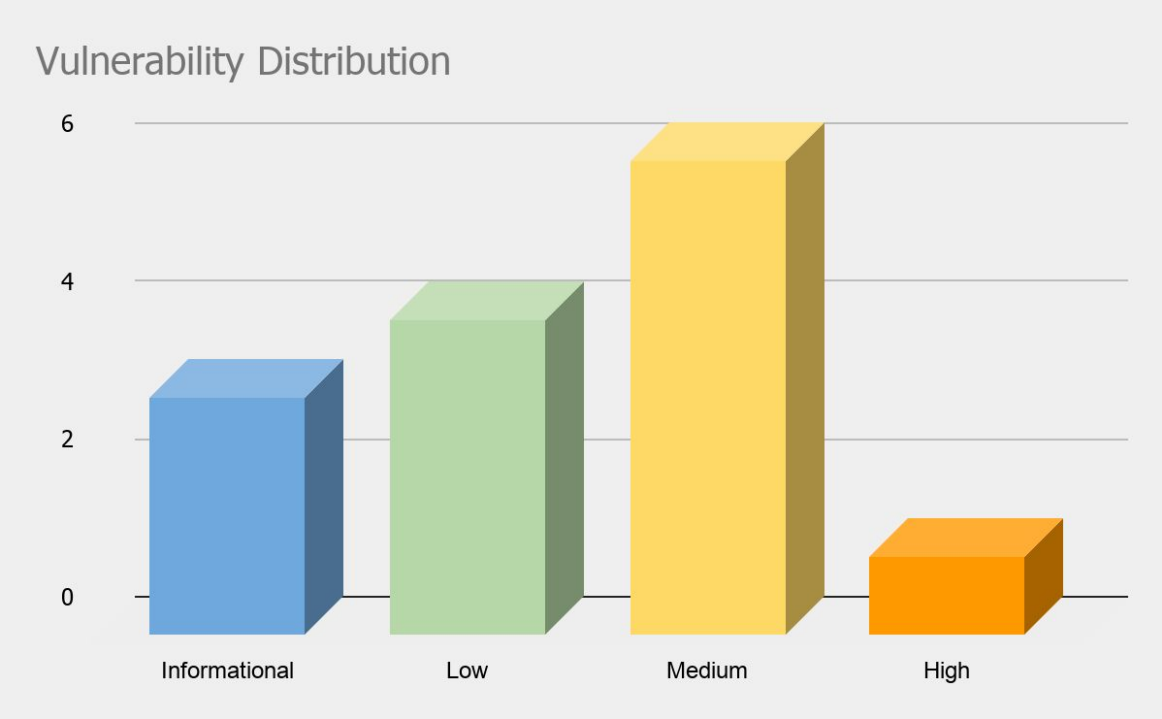
Low

Would have little to no financial or information impact on Dwolla

Informational

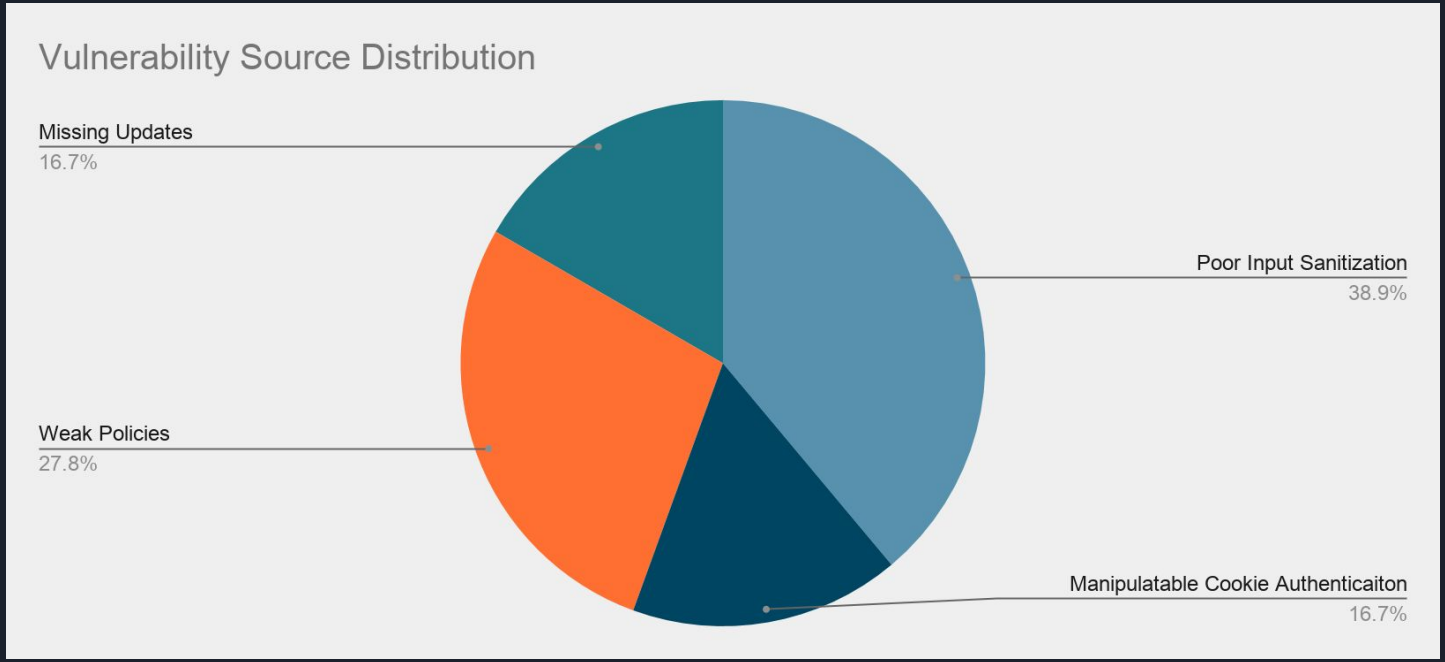
These represent non-trivial areas of improvement.

Testing Results



Distribution of vulnerabilities based on their rankings

Testing Results



Distribution of the root cause of the vulnerabilities found during testing

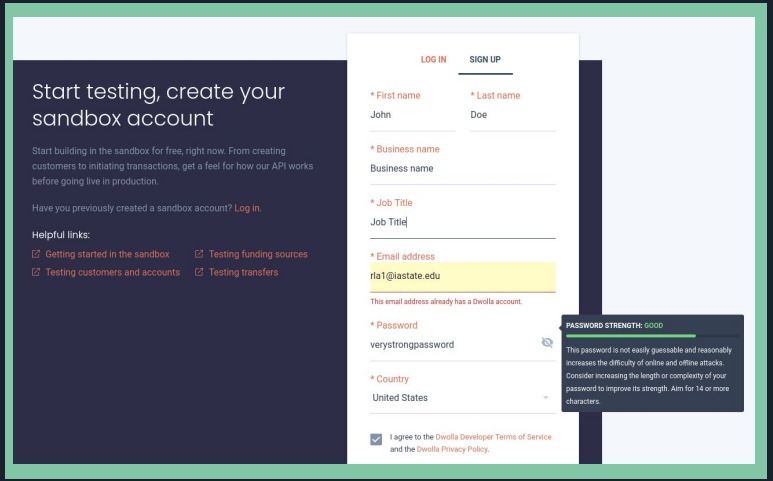
Vulnerability Analysis Example (Demo)

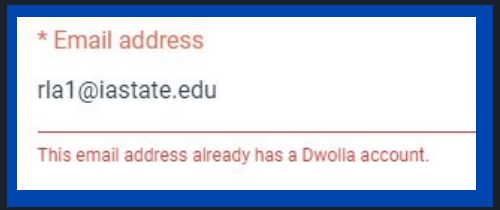
User Email Enumeration	Medium
Domains vulnerable: Sandbox accounts page	

Description: The email addresses of Dwolla's clients can be found out through the account creation process.

Confirmation: The screenshot listed below shows that when attempting to create an account for an existing user, the error "This email address already has a Dwolla account."

Remediation Advice: In order to resolve this issue, we recommend not erroring out when someone signs up with an existing email address, but instead responding with "If the account creation was successful, you will receive an email regarding account verification and activation". This message would apply to all account creation attempts.





Feedback

- Found relevant tools and vulnerabilities.
- Covered extensive attack surfaces across the complex application and API.
- Identified issues professional tests could not find.

DWOLLA

- Completed work in a punctual manner with professional quality.
- A pleasure to work with and I look forward to seeing where the team's careers take them

Knowledge Gained

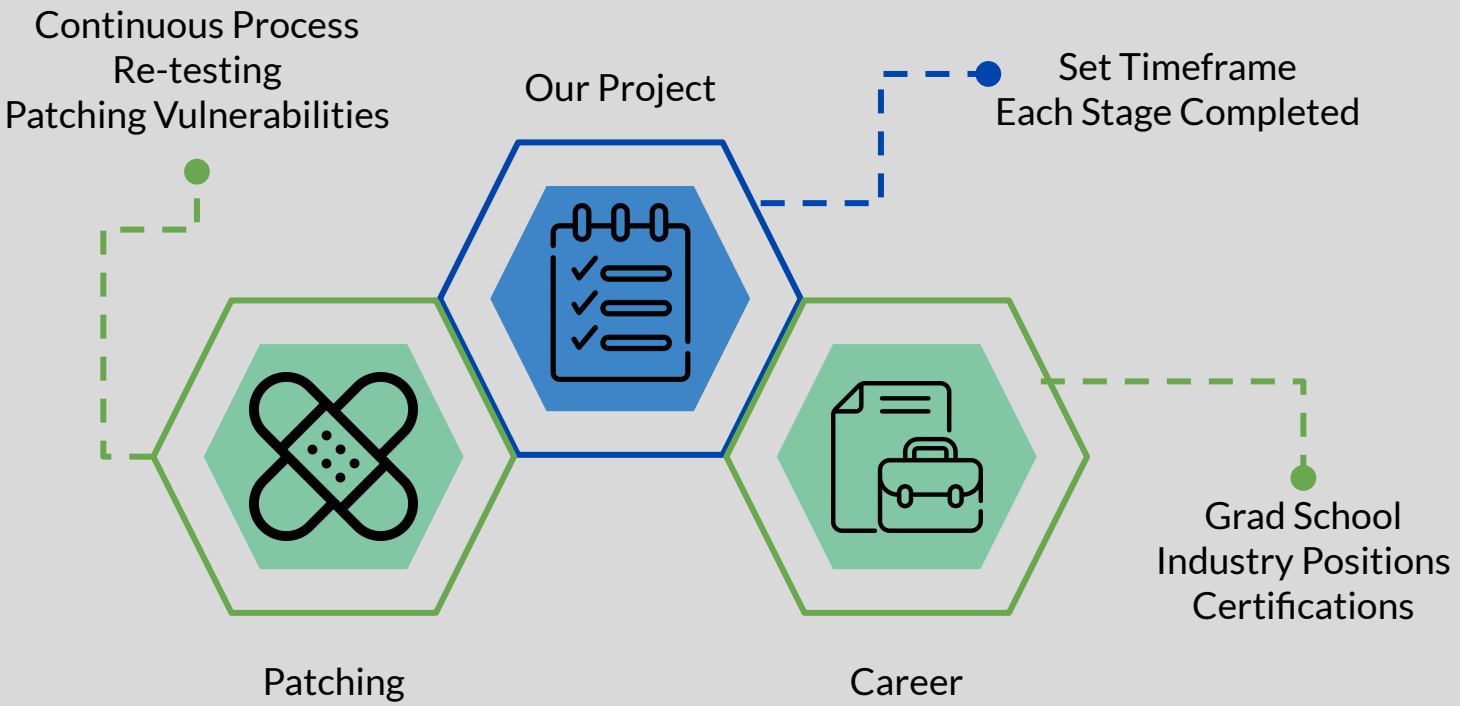
Technical Experience

- New experience with many popular tools for pentesting and vuln discovery
- Understanding of the intricacies and functionality of APIs with applications
- Working with Python to interface an API using scripts and automation
- Techniques and strategies in hunting and exploiting vulns in APIs and Web Applications

Professional Experience

- Professional reporting and communication skills
- Pivoting around unforeseen circumstances and adapting a new approach
- Problem-solving and brainstorming strategies to understand an obstacle and determine a solution
- Cooperation with external, professional client to meet a common objective

Future Work


Continuous Process
Re-testing
Patching Vulnerabilities
Our Project
Set Timeframe
Each Stage Completed
Grad School
Industry Positions
Certifications
Patching
Career

The Team

Chief Pentester

Max Solaro — Cybersecurity Engineering • 2021

Web - Team Lead

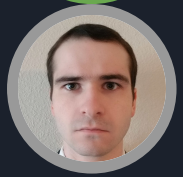
Ryan Anderson — Cybersecurity Engineering • 2021

API - Team Lead

Matthew Maiman — Cybersecurity Engineering • 2021

Facilitator & Scribe

Priyanka Kadaganchi — Computer Engineering • 2021



Editor

Nathan Key — Cybersecurity Engineering • 2021

Jacob Conn
CPR E • 2021

Web Testing Eng.

KayAnne Bryant
CPR E • 2021

API Testing Eng.

Thank you

Questions?