EE/CPRE/SE 491 Bi-Weekly Report
sdmay21-06
Pentest a Payment Provider

Report Period: 2/8/21 - 2/22/21
Client & Advisor: Benjamin Blakely

Team Members & Roles:

Max Solaro - Chief Pentester
Matthew Maiman - Testing Engineer
Ryan Anderson - Lead Reporter
Nathan Key - Editor
Priyanka Kadaganchi - Facilitator & Scribe
Jacob Conn - Web Pentester
KayAnne Bryant - API Pentester

Weekly Summary:

This week we continued our investigation into Dwolla's Web dashboard application and API. Team members were divided into two separate groups one to test the web applications and the other to test the API. The Web application team utilized various automated scanning tools, including BurpSuite, Arachni, OWASP-ZAP, as well as manual testing, to conduct preliminary testing of the web application and to uncover any vulnerabilities or leads to vulnerabilities to explore further. So far, all vulnerabilities tested have come back negative, so the team plans to move forward with more manual testing for Dwolla's permissions settings as they relate to created users in the sandbox dashboard, and other high-impact areas of security. The API team got the Python SDK environment running on their devices. Started mapping the variables/elements and identifying their purpose. Researching and Starting to utilize BurpSuite, OWASP-ZAP, and NetSparker for API testing. Preliminary checks of the POST and GET functions to appear to be controlled which limits the data exposure. Also, tried testing injection methods like SQLi and it shows that the backend inputs properly but returns bad errors.

Past Week Accomplishments:

- Dwolla dashboard/accounts web app testing - Ryan, Nathan, Jacob
    - Ran multiple web application vulnerability testing tools against the Dwolla sandbox accounts and dashboard domain.
    - Performed preliminary manual testing of Dwolla's dashboard to exploit vulnerabilities with their authentication systems
    - Recorded vulnerabilities tested, a description of the test, and whether or not the vulnerability test was successful.

- Dwolla API testing- Max, Matt, Priyanka, KayAnne

- **Excessive Data Exposure-** Preliminary checks of the POST and GET functions to appear to be strict and tightly controlled, limiting data exposure to the necessities. Requires further testing.
- **Injection-** Early testing of various injection methods, particularly SQLi, shows that the Dwolla backend properly scrubs inputs and returns bad input errors if the input is not what is expected for a POST body. Requires further testing, but likely secure in this manner.

Pending Issues:

We have no pending issues at this time

Individual Contributions:

Name	**Contributions**	**Hours last 2 weeks**	**Cumulative Hours**
Max Solaro	Setup DwollaSDK and began running preliminary tests/writing scripts. Mapped out some API elements for a better understanding.	7	13
Matthew Maiman	Established proof of concept with the Python SDK and ran through various tests on Postman to further understand the API calls/requests. Furthermore, mapped out most of the API elements/variables for future reference.	6	11
Ryan Anderson	Performed automatic and manual testing of Dwolla's sandbox web dashboard and accounts page. Recorded vulnerabilities tested and their results.	14	20
Nathan Key	Aided with conducting preliminary scans of Dwolla's Sandbox web dashboard.	5	8
Priyanka Kadaganchi	Set Up of Dwolla SDK and started conducting some preliminary tests or writing scripts. Learning more about BurpSuite, NetSparker, OWASP-ZAP.	5	8
Jacob Conn	Performed automatic and manual testing of Dwolla's sandbox web dashboard and accounts page. Utilized OWASP-ZAP and W3af.	6	10

KayAnne Bryant	Setup DwollaSDK and have run some preliminary tests and writing testing scripts. Began setting up and understanding tools such as BurpSuite, NetSparker, and OWASP-ZAP.	6	11

Plans For Upcoming Week:

- Dwolla dashboard/accounts web app testing - Ryan, Nathan, Jacob
    - Enumerate out potential high-impact areas of the Dwolla dashboard that could not have been detected with scanners, and develop a testing methodology for those potential vulnerabilities.
    - Communicate with our advisor/client on areas of the Dwolla dashboard that could be a high impact area, and either develop additional/new tests for those specifically
    - Research emerging vulnerabilities that may affect the Dwolla dashboard and accounts page

- Dwolla API team: Max, Matt, Priyanka, KayAnne
    - Research more about and start working with OWASP Zap, NetSparker, and BurpSuite.
    - Work on Broken Object Level Authorization, Broken User Authorization, Broken Function Level Authorization, Mass Assignment, Security Misconfigurations, Improper Assets Management.

Advisor Meeting Summary:

During this week's meeting with our advisor, we discussed the various tests and automated scans we have completed on the Web dashboard as well as the intended area of focus for continuing manual testing. Members of the API team discussed their current findings, challenges, and next steps. Additionally, we briefly discussed the potential to move on to the OWASP Juice shop environment should our testing of Dwolla's Sandbox be unfruitful.