Project Plan
Secure Application Layer API Proxy
The Capstone Experience

Team Symantec
Lauren Allswede
Steven Kneiser
Jacob Carl
TJ Kelly
Yili Luo

Department of Computer Science and Engineering
Michigan State University
Fall 2017
Functional Specifications

• Problem
  ▪ Developers want to use a more modern web protocol when integrating VIP with their applications

• Solution
  ▪ Build a secure proxy to wrap the existing interface and provide a more modern web transfer protocol.

• Translation between protocols

• Secure authentication
  ▪ Client and Proxy
  ▪ Proxy and VIP

• Robust Testing Plan
Design Specifications

• No front-end design
  ▪ Used by developers
  ▪ API calls

• Visualization using a basic web application
  ▪ Show what logging in with VIP looks like
  ▪ Walk through steps happening in the background
Screen Mockup: Initial Login page
Screen Mockup: 2-Factor Submission
Screen Mockup: Converting SOAP to REST

Returning SOAP Response

Returning REST Response
Technical Specifications

• Translating between REST and SOAP
  ▪ Mapping REST requests to SOAP requests
  ▪ Converting JSON to XML
  ▪ Converting XML to JSON

• Authentication
  ▪ JSON Web Tokens (JWT)
  ▪ VIP Certificates
  ▪ HTTPS/SSL

• Testing
  ▪ Exhaustive integration tests
  ▪ 100% code coverage with unit tests
System Architecture

Diagram:
- Client
  - REST API Request with access token
  - Front End
  - REST API Response
- Proxy
  - REST
  - Back End
- VIP
  - SOAP
  - Back End
- Proxy Certificate
- SOAP API Request with client ID
- SOAP API Response
System Components

- Software Platforms / Technologies
  - Development
    - ASP .NET Core (C#)
    - VIP .NET SDK
    - NUnit testing framework
    - Gitlab
    - SoapUI
  - Authentication
    - JHASH
    - JWT
    - VIP Access Manager
  - Protocols
    - SOAP
      - XSD
      - WSDL
      - XML
    - REST
      - OpenSSLv3
      - JSON
Testing Plan

• Two separate integration tests per API endpoint
  ▪ Pass
  ▪ Expected Fail
• Unit tests
  ▪ Verify XML and JSON conversion
  ▪ Ensure feature-parity with SOAP API
• Performance under load
Risks

• Load Testing Ability
  - **Difficulty:** Medium
  - **Description:** Our ability to load test our proxy is limited. We need to be able to simulate several hundred to thousand requests per second to properly test the scalability of our API.
  - **Mitigation:** Talking with client about multiple mock accounts.

• Client Side Authentication
  - **Difficulty:** Medium
  - **Description:** We are required to implement client side authentication without an existing or building an authentication server. This blocks us from using technologies such as OAuth2.
  - **Mitigation:** Work with client and research to figure out a client-proxy authentication method that does not rely on an authentication server.

• VIP API documentation
  - **Difficulty:** Easy
  - **Description:** The documentation provided is cumbersome and does not thoroughly explain how to communicate with the SOAP API.
  - **Mitigation:** Searching for more API related documentation and working with client to identify and clarify misunderstandings.
Questions?