Project Bermuda
Software Requirements Specification

Emily Clauson, Stephen Swanson, Alex Iapara, Jake Khal

https://github.com/Fenix-A-I/Project-Bermuda

CS 422 Fall 2024 - Juan Flores

1. The Concept of Operations (ConOps)

The concept of operations (ConOps) “describes system characteristics for a proposed system from the users’ viewpoint.” (IEEE Std 1362-1998) The ConOps document communicates overall system characteristics to all stakeholders.

1.1. Current System or Situation

Capture the flag (CTF) events are commonly used in the cybersecurity world to learn about real life vulnerabilities. They provide users the chance to practice ethical hacking skills in controlled environments. “Hack the Box” and “VulnHub” are common platforms that are used for this. Although students and club members can access these external third party platforms, they typically require money in order to participate in the form of subscriptions. This not only restricts access to a lot of users who do not have financial resources, but causes students to be unmotivated to use these platforms. Currently, no system within the university offers a resource like this, creating a gap in cybersecurity skill development. Since external platforms are not integrated with the university it can also make it difficult to align challenges with curriculum, as well as assign challenges in classes or clubs. These issues make these platforms less accessible to students causing them to miss out on hands-on learning experiences that could be valuable for their education.

1.2. Justification for a New System

The University of Oregon including their cybersecurity club needs an educational platform such as this that is free and safe for students to use. In the past, the university had little cybersecurity resources in general to help students with their interests. Many chose majors that were related to the field, but did not directly teach students knowledge they needed to join the workforce for cybersecurity. With an extreme increase in technology in society, cybersecurity is becoming more important and prevalent. This is causing a higher demand for cybersecurity jobs. These increases were recognized by the university which caused them to create a new cybersecurity major as of 2024. Since this addition is new, there are still few resources to directly help students.

Another issue commonly faced by students is financial troubles. In 2022, it was recorded that over 45% of students were eligible for need based financial aid. With this, around 70%-80% of students work jobs while working towards their degree. Financial struggles that come with being a student are evident. External platforms often require subscriptions or other payments that make it less accessible to students. These shortcomings together show why a free platform should be made for the university.

1.3. Operational Features of the Proposed System

Bermuda is a gateway that will act as a foundation for a capture the flag experience. It will securely authenticate users, store credentials for them to have their own linux environment, and track progress in challenges. This addresses a gap in the University of Oregon's cybersecurity resources. Users will first see a welcome page that is designed to inform users of what the website is used for, and prompt them to log in with their university email. Their email will be authenticated by office o365, which will ensure security for the platform as well as grant free access to students. From here they will be redirected to a page that allows users to submit ssh keys or passwords. These credentials will be encrypted and stored in a database so that way they can be used for a virtual environment. From here users will be able to flip from the challenge system and terminal access. These will be individual containers that further ensure security. These containers will also be destroyed when users sign out to maintain an organized system. This foundation allows students full access to capture the flag challenges where they can learn from pen-testing and real-life vulnerabilities. The page will inform users about their progress on the challenges, as well as provide information for how to navigate the pages.

1.4. User Classes

Primary Users: University of Oregon students enrolled in cybersecurity programs, cybersecurity club, or any student at University of Oregon. They are familiar with basic networking concepts and security practices but may vary in experience with ethical hacking tools.

1.5. Modes of Operation

There is only one mode of operation for this system that allows users to sign in and store credentials as mentioned above. This mode allows the application to be used as intended. Maintenance and backend modes are not needed for the implementation of the project, but could be added for future updates.

1.6. Operational Scenarios (aka "Use Cases")

Use Case 1: Authentication on o365

Brief description: A student logs in using their Microsoft O365 credentials, which authenticates them as a University of Oregon student. Upon successful login, the system creates an account and stores their SSH key for future sessions.

Actors: A student

Preconditions:

1. The student has an active University of Oregon email address (@uoregon.edu).
2. The student has access to Microsoft O365 credentials and can log in.
3. The Bermuda platform is available and accessible via an internet connection.

Steps to Complete the Task:

1. The student navigates to the Bermuda login page.
2. The student selects the "Login with Microsoft O365" option.
3. The system redirects the student to the Microsoft O365 authentication page.
4. The student enters their university email and password, then submits the form.
5. Microsoft O365 verifies the student's credentials.
6. Upon successful authentication, the system redirects the student to their Bermuda dashboard.

Postconditions: The student is successfully logged into the Bermuda platform and can now access challenges and their personal profile.

Use Case 2: Upload SSH Key

Brief Description: This use case describes how a student uploads their SSH key to the Bermuda platform to enable secure access to their Linux environment.

Actors: A student.

Preconditions:

1. The student is logged into the Bermuda platform.
2. The student has an SSH public key available in a supported format.
3. The student has permission to access a personal Linux environment on the Bermuda platform.

Steps to Complete the Task:

1. The student navigates to the profile settings page on the Bermuda platform.
2. The student selects the “Upload SSH Key” option.
3. The system prompts the student to upload or paste their SSH public key.
4. The student provides the SSH key in the required format and submits it.
5. The system validates the format of the key.
6. The system stores the key securely in the database if it is valid.

Postconditions: The student’s SSH key is securely stored in the database and associated with their account, enabling SSH access to their Linux environment.

Use Case3: Track Progress on Challenges

Brief Description: This use case describes how the Bermuda platform updates and displays a student’s progress on completed challenges and lessons.

Actors: A student

Preconditions:

1. The student is logged into the Bermuda platform.
2. The student has completed at least one challenge or lesson on the platform.
3. The platform has the capability to log progress and display it within the user’s profile

Steps to Complete the Task:

1. The student completes a challenge or lesson on the Bermuda platform.
2. The system records the completion status of the challenge in the student’s profile.
3. The student navigates to their profile page to view their progress.
4. The profile page displays updated progress metrics, including completed challenges, points, or achievements.

Postconditions: The student’s profile reflects accurate and up-to-date progress, allowing them to view completed challenges and earned achievements.

2. Specific Requirements

2.1. External Interfaces (Inputs and Outputs)

2.1.1. Microsoft o365 Authentication (OAuth)

2.1.1.1. Name of item: Microsoft o365 Authentication

2.1.1.2. Description of purpose: Enables secure user authentication, allowing only authorized university-affiliated users to access the Bermuda platform.

2.1.1.3. Source of Input or Destination Output: Source of input is the Microsoft O365 authentication service; destination of output is the Bermuda platform, verifying the user's identity

2.1.1.4. Valid Ranges of Inputs and Outputs:

2.1.1.4.1. Input: Valid university email address and associated credentials.

2.1.1.4.2. Output: Authenticated session token and redirect to platform

2.1.1.5. Units of Measure: N/A

2.1.1.6. Data Formats: JSON Files

2.1.2. SSH Key Management Database

2.1.2.1. Name of item: SSH Key Management

2.1.2.2. Description of Purpose: Manages and stores SSH keys for each student to ensure secure, authenticated access to individual Linux environments.

2.1.2.3. Source of Input or Destination of Output: Input source is the user uploading their SSH public key, and the output destination is the secure database where keys are stored for retrieval.

2.1.2.4. Valid Ranges of Inputs and Outputs:

2.1.2.4.1. Input: User's SSH public key in OpenSSH or RSA format.

2.1.2.4.2. Output: Stored keys in the secure database and deployed to the user's individual environment.

2.1.2.5. Units of Measure: Number of SSH keys stored per user

2.1.2.6. Data formats: OpenSSH public key

2.1.3. User Progress Tracking

2.1.3.1. Name of Item: User Progress Tracking

2.1.3.2. Description of Purpose: Tracks user progress across challenges and lessons, updating their profiles and enabling personalized feedback and recommendations.

2.1.3.3. Source of Input or Destination of Output: Input source is the user's interaction with challenges; the output destination is the Bermuda user profile database, where progress data is recorded and stored.

2.1.3.4. Valid Ranges of Inputs and Outputs:

2.1.3.4.1. Input: Challenge completion status

2.1.3.4.2. Output: Updated user profile with completed challenges and earned points or achievements.

2.1.3.5. Units of Measure: Challenge completion status (binary: completed/not completed); points or experience points (XP) as integers.

2.1.3.6. Data Formats: JSON object with user ID, challenge ID, and completion status; optionally includes a timestamp.

2.2. Functions

Define the actions that must take place in the software to accept and process inputs and generate outputs (ISO/IEC/IEEE 29148:2011). These definitions must include:

1. Validity checks on the inputs.

1) **Email verification:** Ensure the email addresses are in the correct format and belong to the University of Oregon domain.
2) **SSH keys/Password Validation:** Ensure the SSH keys or passwords meet required security standards.
3) **Empty Field Check:** Ensure that no mandatory fields are left blank.

2. Sequence of operations in processing inputs.

1) **User login:**
    - Users are prompted to log in with their university email.
    - Email is authenticated using Office O365.
2) **Credentials Submission:**
    - Redirect the user to the credentials submission page.
    - Validate and Encrypt SSH keys or passwords.
    - Store the encrypted credentials in the database.
3) **Environmental Setup:**
    - Create a user-specific Linux environment (container).
    - Provide access to the terminal and challenge system.
4) **User Interaction:**
    - Track progress in challenges.
    - Update user progress and provide navigation information.

3. Responses to abnormal situations, including error handling and recovery.

1) **Authentication Failure:** Display an error message and prompt the user to re-enter their email.
2) **Invalid Credentials:** Inform the user of invalid SSH keys or passwords and ask for re-submission.
3) **System Errors:** Log the error details and display a user-friendly error message. Attempt to recover by retrying the operation if possible.
4) **Container Issues:** If a container fails, destroy it and create a new one. Notify the user of the issue.

4. Relationship of outputs to inputs, including

(a) input/output sequences

- Input: User submits their university email.
- Output: User is authenticated and logged in.
- Input: User submits SSH keys or passwords.
- Output: Credentials are stored, and a virtual environment is set up.

(b) formulas for input-output conversion

- Encryption: Use standard encryption algorithms to convert plaintext credentials to encrypted format for storage.

2.3. Usability Requirements

- **Effectiveness**: Users should be able to complete the login and setup process within 5 minutes with a success rate of 95%.
- **Efficiency**: Pages should load in less than 2 seconds on average, and interactions (e.g., submitting credentials) should complete within 1 second.
- **Satisfaction**: User satisfaction levels should be at least 4 out of 5 in usability surveys, with 80% of users providing positive feedback.

2.4. Performance Requirements

- **Static Numerical Requirements**:
  - Support up to 1000 concurrent users.
  - Handle up to 10,000 authentication requests per hour.

- **Dynamic Numerical Requirements**:
  - 95% of login attempts should be processed in less than 3 seconds.
  - 90% of SSH key/password submissions should be processed in less than 4 seconds.
  - Container setup and teardown should complete in less than 5 seconds for 95% of cases.

2.5. Software System Attributes

1) **Reliability**
   - Importance: Ensures the platform is always available and operational.
   - Steps: Implement redundancy, failover mechanisms, and continuous monitoring.
2) **Security**
   - **Importance:** Protects user data and maintains the integrity of the system.
   - **Steps:** Use encryption, regular security audits, and strong authentication methods.
3) **Privacy**
   - **Importance:** Ensures user data is protected and not exposed to unauthorized parties.
   - **Steps:** Implement data encryption, access controls, and anonymization where necessary.
4) **Maintainability**
   - **Importance:** Allows for easy updates and bug fixes.
   - **Steps:** Use modular design, write comprehensive documentation, and follow coding standards.
5) **Portability**
   - **Importance:** Enables the system to run on different platforms and environments.
   - **Steps:** Use platform-independent technologies and containerization.

3. References

Admissions, admissions.uoregon.edu/majors/cybersecurity. Accessed 5 Nov. 2024.

“Division of Student Life.” *Campus Employment | Division of Student Life*, studentlife.uoregon.edu/jobs. Accessed 5 Nov. 2024.

Saboorian, Jasmine. “UO Cybersecurity Major Expands.” *Daily Emerald*, dailyemerald.com/46256/news/uo-cybersecurity-major-expands/. Accessed 5 Nov. 2024.

“University of Oregon - Tuition and Financial Aid | US News Best Colleges.” *University of Oregon Tuition & Financial Aid*, Copyright 2024 © U.S. News & World Report L.P., 2024, www.usnews.com/best-colleges/university-of-oregon-3223/paying.

4. Acknowledgments

SRS template was provided by Juan Flores.