**EQUIPMENT TEST PLAN**

**Wireshark - Linux 2 Web Server**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Title: | **AWS CyberShift Initiative** |  | Date Prepared: | 23rd of June, 2023 |

|  |
| --- |
| **Overall project scope and objectives** |
| The AWS CyberShift Initiative project will secure OzCazual's cloud infrastructure and enable a safe and secure migration from their existing local server to Amazon AWS.  The primary goal is to address the sudden 200% increase in online sales and staff, create a scalable infrastructure that can meet future business demands, and ensure the confidentiality, integrity, and availability of the systems and customer data.  The project will Implementing various security controls, and upgrade the systems and tools currently used at OzCazual |
| **Test objectives and success criteria** |
| The objective of this unit test plan is to verify the successful installation of Wireshark on a Linux Web Server. The test plan will cover the key steps involved in the installation process and ensure that the installation is completed without errors or issues. |
| **Test resources required (people, hardware, software, test tools)** |
| The test environment should consist of the following:   * Linux web server * Test server with appropriate hardware specifications * Test user account with administrative privileges  Table - Software Version to Be Tested  |  |  | | --- | --- | | **Wireshark Version** | **Description** | | **Version 3.6.2** | Running on Linux 5.15.0-30 generic |  Table - People, Roles, and Time Allocation  |  |  |  | | --- | --- | --- | | **Role** | **Name** | **Resource Allocation** | | *Program Manager* | *Giuseppe Raciti* | *As required* | | *Test Lead* | *Mark Byrne* | *100%* | |
| **Test schedule** |
| *The below test schedule outlines the key milestones to successfully complete the Wireshark unit test:*  **Table 4-5. Test Schedule**   |  |  |  | | --- | --- | --- | | **Date** | **Milestones** | **Resource Allocation** | | *26/6/2023* | *Test Plan Start* | *High-level test case review with customer and account team* | | *26/6/2023* | *Test Plan—Review & Approval* | *Test Plan document review with customer and account team* | | *27/6/2023* | *Test Start* | *Dependent on test entrance criteria documented in EC* | | *27/6/2023* | *Test Complete* | *Completion of all test cases* | | *27/6/2023* | *Test Result Report Complete* | *Final test results report complete* | |
| **Format for Written Test Case** |
| *The below test scenario is a unit test to confirm the successful installation and operation of Wireshark on a Linux Web Server*   |  |  | | --- | --- | | **Test ID:** | Test 002 | | **Node List:** | N/A | | **Test Description:** |  | | **Test Phase:** | Baseline test | | **Test Steps:** | Pre-installation Checks:   * Verify that the Linux web server meets the minimum system requirements for Wireshark installation (e.g., supported Linux distribution, processor, memory, disk space). * Ensure that the test user account has administrative privileges and necessary permissions for installation.   Downloading the Installation Package:   * Download the appropriate Wireshark installation package for the Linux distribution being used. * Confirm that the package is downloaded successfully and is accessible.   Installation Process:   * Execute the installation package using appropriate commands (e.g., dpkg, rpm) based on the Linux distribution. * Monitor the installation process for any errors or warnings. * Verify that the installation completes without any issues.   Post-installation Verification:   * Check if the Wireshark executable is installed in the correct location. * Ensure that the necessary dependencies or libraries are installed and functioning properly. * Confirm that Wireshark can be launched from the command line or desktop environment. * Execute a basic Wireshark command (e.g., "wireshark -v") to verify the version information and check for any errors.   Testing Functionality:   * Capture Test: Perform a simple packet capture using Wireshark on a network interface and verify that packets are being captured correctly. * Protocol Analysis: Select a captured packet and analyze it to ensure that Wireshark can dissect and display the protocol information accurately. * Filter Functionality: Apply different filters (e.g., IP address, protocol) to the captured packets and verify that the filtering is working as expected. * Exporting and Importing: Export captured packets to a file, import the file back into Wireshark, and ensure that the imported data matches the original capture. | | **Expected Results:** | * Confirm that the installation process completes without any errors or warnings. * Confirm that the application does not crash or produce any error messages. * Validate that Wireshark accurately interprets and decodes network protocols. * Verify that the exported file is generated successfully and contains the captured data. * Verify that appropriate error messages are displayed, guiding the user on how to resolve the issue. | | **Observed Results:** | * Confirm that the installation process completes without any errors or warnings.      * Confirm that the application does not crash or produce any error messages.        * Validate that Wireshark accurately interprets and decodes network protocols.      * Verify that the exported file is generated successfully and contains the captured data.      * Verify that appropriate error messages are displayed, guiding the user on how to resolve the issue.   N/A | | **Pass/Fail:** | **Pass** | |