

**END-TO-END**

**TEST PLAN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Plan Products:** | pfSense  Snort  Sophos / Intercept X  Splunk  Wireshark |  | **Prepared by:** | Giuseppe Raciti  Mark Byrne  Shaun Heywood  Mauricio Guerra |

CONTENTS

[**INTRODUCTION 4**](#_5svb60ra6xy3)

[The Windows 2022 AD Server Network: 4](#_3o3mxjqnpb9u)

[The Linux 2 Web Server Network: 4](#_teiqt1vzh8wp)

[**OBJECTIVES AND TASKS 5**](#_17ydxuilida2)

[The Windows 2022 AD Server Network 5](#_vj5inlfjqslb)

[Objectives: 5](#_yryzxwk93f76)

[Tasks: 5](#_pd16i263rfe)

[The Linux 2 Web Server Network 6](#_ksf964gq5aww)

[Objectives: 6](#_ycohjtpoxpb)

[Tasks: 6](#_xaasrt7qm5lu)

[**SCOPE 6**](#_ni301ural7d9)

[The Windows 2022 AD Server Network 6](#_hxot26tzczb2)

[General Scope: 6](#_osj4v2laecg5)

[Tactics Scope: 6](#_xcgee1et94bn)

[The Linux 2 Web Server Network 7](#_5xpik26uslix)

[General Scope: 7](#_pxxmj5j1n82u)

[Tactics Scope: 7](#_sc8j15sfxw1g)

[**TESTING STRATEGY 7**](#_3csqyaglx8nv)

[Introduction: 7](#_gks6jnohc9cm)

[Objectives: 7](#_dmjzremjxti5)

[Scope: 7](#_5qw732cjoktc)

[Testing Techniques: 7](#_geomp7oakc77)

[Participants: 8](#_hy2shxa7k7c)

[Testing Resources: 8](#_mjn3faym6r9k)

[Test Schedule: 8](#_om6eot17kbkx)

[Schedule for Testing Tasks and Test Milestones: 8](#_smclz9uyx3lq)

[Reporting and Documentation: 8](#_t7144cf9uwih)

[**Network Diagram 9**](#_g5gcoj216jzd)

[**HARDWARE REQUIREMENTS 9**](#_vejezyn8c6s6)

[**ENVIRONMENT REQUIREMENTS 10**](#_g55w5uuhjk3w)

[Virtualization Environment: 10](#_m1wgzt90eapg)

[Networking: 10](#_e0gwtokmszcv)

[Operating Systems: 10](#_lhgv6olrw128)

[Security Systems: 10](#_qfouz2vmbl9n)

[Monitoring and Analysis Tools: 10](#_8v67jefuscsf)

[Test Data and Scenarios: 10](#_mkhmkntzlqiy)

[Documentation and Reporting: 10](#_7gqevjkt24up)

[**TEST SCHEDULE 11**](#_2uny54rdy5ha)

[Detailed Test Schedule: 11](#_wlt709ka19al)

[Test Milestones and Item Transmittal Events: 11](#_1rb8up30e897)

[Estimation of Time Required for Testing Tasks: 11](#_rj8ewlh8vc8m)

[Schedule for Testing Resources: 11](#_9k0t8c2vakem)

[**CONTROL PROCEDURES 12**](#_97rwds82u2ex)

[Problem Reporting: 12](#_y5ceh8ehk8e)

[Change Requests: 12](#_d6h1zcs9dhdd)

[**FEATURES TO BE TESTED 13**](#_k5xayysufsn4)

[Windows 2022 AD Server: 13](#_t0yj3uxmpxd8)

[Linux 2 Web Server: 13](#_acxcrugtjfvd)

[Security Systems: 13](#_pzrw0q4bepux)

[Network Segmentation and DMZ Configuration: 13](#_wphafx4oq43z)

[Integration and Communication: 13](#_xffy3sqxm2uo)

[Compliance: 13](#_sf1mzwciosjk)

[**FEATURES NOT TO BE TESTED 14**](#_mtk995l3v4iv)

[**RESOURCES/ROLES & RESPONSIBILITIES 14**](#_fq7zhqufo5k6)

[**SCHEDULES 15**](#_vrizjhysd7pt)

[Major Deliverables: 15](#_5v81azt1lvby)

[**SIGNIFICANTLY IMPACTED DEPARTMENTS (SIDs) 15**](#_paoj2jv0td53)

[IT Department: 15](#_3hhift28fhqk)

[**DEPENDENCIES 16**](#_uutcpgpgxgpu)

[Significant Constraints on Testing: 16](#_3t6qxe3zu2eu)

[**RISKS/ASSUMPTIONS 17**](#_n646p9we1myl)

[Risk/Assumption: Delay in Delivery of Test Items 17](#_n3t7xulwaum5)

[**TOOLS 18**](#_yv427v7f24l6)

[**TEST STEPS 19**](#_82yuukuai3fo)

[Windows 2022 AD Server 19](#_tgxgczsvw3ia)

[Linux Web Server 20](#_bpo0xzs693ly)

[**TEST OUTCOME 21**](#_e33e1jbiharu)

[Windows 2022 AD Server 21](#_kmfr19o6vszi)

[Linux Web Server 21](#_cl81jqvmwvsq)

[**CONCLUSION: 22**](#_u8c6txaarpy)

[**Approvals: 22**](#_6g0gjy5zxul9)

|  |
| --- |
| INTRODUCTION |
| The Windows 2022 AD Server Network: Brute-force attacks are one of the most common methods employed by hackers to gain unauthorized access to systems and accounts. In this type of attack, an attacker systematically attempts various combinations of usernames and passwords until they find the correct credentials. The objective is to exploit weak or easily guessable passwords to gain entry.  As part of the end-to-end test plan, we will simulate a brute-force attack on a user account within the Windows network's Active Directory (AD) server. This test aims to evaluate the effectiveness of the implemented security systems in detecting and responding to such attacks.  The target user account will be subject to a two-factor authentication (2FA) mechanism, leveraging Azure AD services for enhanced security. The test will involve multiple failed login attempts using incorrect credentials to trigger alerts within the security infrastructure.  By conducting this test, we can assess the robustness and efficiency of the 2FA system, the Snort IDS/IPS system, the Splunk log monitoring and alerting, and potentially the Sophos antivirus/malware protection system. This comprehensive assessment will provide valuable insights into the detection and response capabilities of the Windows network's security systems, helping to identify any potential weaknesses or areas for improvement.  Network reconnaissance and port scanning are common activities conducted by malicious actors to gather information about a target network and identify potential vulnerabilities. As part of the end-to-end test plan, we will simulate a port scan or network reconnaissance activity on the Linux DMZ (Demilitarized Zone) network to evaluate the effectiveness of the installed security systems in detecting and responding to such activities. The Linux 2 Web Server Network: In this test, we will focus on a specific system or IP address within the Linux DMZ network, which is being monitored by the security systems. By using a port scanning tool, such as Nmap, we will attempt to identify open ports and services on the selected system.  The objective of this test is to assess the detection and alerting capabilities of the implemented security systems in the Linux network. We will evaluate the performance of the Sophos firewall, the Snort IDS/IPS system, the Splunk log monitoring and alerting system, as well as the real-time network monitoring using Wireshark.  During the test, we expect the security systems to respond to the port scan or network reconnaissance activity by generating alerts and logging the event. The Sophos firewall should detect and log the suspicious activity, while the Snort IDS/IPS system should analyze the network traffic and identify the port scan as potentially malicious, triggering an alert. Splunk, which monitors the Linux DMZ network, should capture and log the port scan activity and potentially generate an alert based on predefined correlation rules. Additionally, Wireshark will provide real-time visibility into the network traffic and display the port scan activity for analysis.  It is crucial to conduct this test in a controlled and authorized environment to ensure the security and integrity of the systems being evaluated. By performing this test, we can validate the configuration and effectiveness of the installed security systems in detecting and responding to network reconnaissance activities, ultimately enhancing the overall security posture of the Linux DMZ network |

|  |
| --- |
| OBJECTIVES AND TASKS |

|  |
| --- |
| The Windows 2022 AD Server Network |

### Objectives:

* Evaluate the effectiveness of the security systems implemented in the Windows network.
* Identify any vulnerabilities or weaknesses in the configuration of the Windows network and its security components.
* Assess the detection and response capabilities of the 2FA system, firewall, antivirus/malware protection, IDS/IPS, and log monitoring solutions.
* Ensure that the security measures in place can detect and respond to specific attack scenarios.
* Validate the integrity and security of the Windows network before and after the migration to the cloud environment.

### Tasks:

**Test the 2FA System:**

* Attempt to authenticate with incorrect credentials multiple times for a selected user account.
* Verify if the 2FA system enforces proper access controls and generates alerts or blocks further login attempts.

**Test the Firewall:**

* Conduct penetration testing to assess the effectiveness of the pfSense firewall rules.
* Attempt to bypass or exploit the firewall to gain unauthorized access to the Windows network.
* Verify if the firewall generates alerts or blocks suspicious network traffic.

**Test the Antivirus/Malware Protection:**

* Introduce known malicious files or simulated malware to the Windows environment.
* Monitor if the Sophos antivirus/malware protection detects and blocks the threats.
* Validate if alerts are generated for the detected threats.

**Test the IDS/IPS System:**

* Simulate different attack scenarios, such as network scans or exploit attempts.
* Monitor if the Snort IDS/IPS system detects and generates alerts for the simulated attacks.
* Validate if the system effectively identifies and responds to suspicious activities.

**Test Log Monitoring and Analysis:**

* Verify that Splunk is properly configured to collect logs from the Windows AD server and host machine.
* Monitor the logs for any security-related events or incidents.
* Analyze the log data to identify anomalies or potential security breaches.

**Monitor Network Traffic:**

* Use Wireshark to capture and analyze network traffic in real-time.
* Monitor for any unauthorized or suspicious activities.
* Analyze the captured packets to identify potential security risks or anomalies.

|  |
| --- |
| The Linux 2 Web Server Network |

### Objectives:

* Assess the security posture of the Linux network in the DMZ environment.
* Evaluate the effectiveness of the security systems implemented in the Linux network.
* Identify any vulnerabilities or weaknesses in the configuration of the Linux network and its security components.
* Validate the detection and response capabilities of the firewall, antivirus/malware protection, IDS/IPS, log monitoring, and network traffic analysis solutions.
* Ensure the integrity and security of the Linux network in the cloud environment migration.

### Tasks:

**Test the Firewall:**

* Conduct penetration testing to assess the effectiveness of the Sophos firewall rules in the Linux network's DMZ.
* Attempt to bypass or exploit the firewall to gain unauthorized access to the Linux network.
* Verify if the firewall generates alerts or blocks suspicious network traffic.

**Test the Antivirus/Malware Protection:**

* Introduce known malicious files or simulated malware to the Linux environment in the DMZ.
* Monitor if Sophos Intercept X antivirus/malware protection detects and blocks the threats.
* Validate if alerts are generated for the detected threats.

**Test the IDS/IPS System:**

* Simulate different attack scenarios, such as network scans or exploit attempts, targeted at the Linux network.
* Monitor if the Snort IDS/IPS system detects and generates alerts for the simulated attacks.
* Validate if the system effectively identifies and responds to suspicious activities.

**Test Log Monitoring and Analysis:**

* Verify that Splunk is properly configured to collect logs from the Linux server.
* Monitor the logs for any security-related events or incidents.
* Analyze the log data to identify anomalies or potential security breaches.

**Monitor Network Traffic:**

* Use Wireshark to capture and analyze network traffic in real-time within the Linux network's DMZ.
* Monitor for any unauthorized or suspicious activities.
* Analyze the captured packets to identify potential security risks or anomalies.

|  |
| --- |
| SCOPE |

|  |
| --- |
| The Windows 2022 AD Server Network |

### General Scope:

* The test focuses on the Windows network, including the AD server and associated components.
* The test aims to evaluate the security systems implemented within the Windows network.
* The test encompasses activities related to authentication, network security, antivirus/malware protection, intrusion detection, log monitoring, and network traffic analysis.

### Tactics Scope:

* The test encompasses tactics such as brute-force attacks, firewall penetration testing, antivirus/malware simulation, network reconnaissance, and network traffic monitoring.
* It aims to assess the effectiveness of security measures against these tactics and identify any vulnerabilities or weaknesses.

By executing these objectives and tasks within the defined scope, the test will provide insights into the security posture of the Windows network and ensure that the implemented security systems are effective in mitigating potential risks and threats.

|  |
| --- |
| The Linux 2 Web Server Network |

### General Scope:

* The test focuses on the Linux network deployed in the DMZ environment.
* The security systems implemented within the Linux network are the main focus of the test.
* The test covers areas such as firewall protection, antivirus/malware detection, intrusion detection, log monitoring, and network traffic analysis.

### Tactics Scope:

* The test encompasses tactics such as firewall penetration testing, antivirus/malware simulation, network reconnaissance, and network traffic monitoring.
* It aims to assess the effectiveness of security measures against these tactics and identify any vulnerabilities or weaknesses.

By executing these objectives and tasks within the defined scope, the test will provide insights into the security posture of the Linux network in the DMZ. It ensures that the implemented security systems effectively mitigate potential risks and threats, maintaining the integrity and security of the Linux environment in the cloud migration.

|  |
| --- |
| TESTING STRATEGY |

## Introduction:

The testing strategy aims to ensure comprehensive and effective testing of the infrastructure migration to a cloud environment. It encompasses both the Windows and Linux networks, focusing on security, functionality, and performance. The strategy outlines the objectives, scope, testing techniques, resources, and schedule to be followed.

## Objectives:

The testing strategy aims to achieve the following objectives:

* Validate the functionality and performance of the Windows and Linux networks in the cloud environment.
* Identify and mitigate any security vulnerabilities or risks.
* Ensure proper integration and communication between the different components.
* Verify compliance with coding standards, best practices, and industry regulations.
* Evaluate the scalability and reliability of the infrastructure.

## Scope:

The testing strategy covers the following aspects:

* Windows 2022 AD server functionality and performance.
* Linux 2 Web Server functionality and performance.
* Security systems including pfSense firewall, Sophos firewall, Sophos Intercept X, Snort IDS/IPS, and Splunk.
* Two-factor authentication using Azure AD services.
* Real-time monitoring using Wireshark.
* Integration and communication between the Windows and Linux networks.
* Compliance with coding standards and best practices.
* Network segmentation and DMZ configuration for improved security.

## Testing Techniques:

The following testing techniques will be employed:

* Unit Testing: Developers will write test scripts to test individual components or units of code. Code coverage analysis, statement and branch coverage, and requirements traceability will be used to judge the comprehensiveness of the testing effort.
* Integration Testing: Test the interaction and communication between the Windows and Linux networks, ensuring seamless integration and proper functioning of the components.
* Security Testing: Perform penetration testing, vulnerability assessment, and threat modeling to identify and address security vulnerabilities. Test the effectiveness of security systems such as firewalls, antivirus, and IDS/IPS.
* Performance Testing: Evaluate the performance, scalability, and response time of the infrastructure under different load conditions.
* Compliance Testing: Verify compliance with coding standards, best practices, and industry regulations.
* User Acceptance Testing: Involve end-users to validate the functionality and usability of the migrated infrastructure.

## Participants:

The participants involved in the testing process include:

* Project Manager: Giuseppe Raciti
* Cyber Security Specialist: Shaun Heywood
* Cloud Architect / Engineer: Mark Byrne
* Server Administrator: Mauricio Guerra
* Developers responsible for writing and executing unit tests

## Testing Resources:

* Test Environment: A dedicated test environment will be set up with virtual machines in Hyper-V using Azure.
* Testing Tools: Various tools will be utilized, including penetration testing frameworks, vulnerability scanners, log analysis tools, Wireshark, and performance testing tools.
* Documentation: Detailed documentation will be prepared to capture test plans, test cases, test data, and test results.

## 

## Test Schedule:

The testing tasks and estimated time required for each task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Duration | Start Date | End Date |
| Test Plan Approval | 1 Hour | 01/07/2023 | 02/07/2023 |
| Test Environment Setup | 12 Hours |  |  |
| Test Data Preparation | 1 Hour | 02/07/2023 | 03/07/2023 |
| Unit Test Case Development | 12 Hours |  |  |
| Unit Test Execution | 2 Hours |  |  |
| Integration Testing | 2 Hours |  |  |
| Security Testing | 2 Hours |  |  |
| Performance Testing | 2 Hours |  |  |
| Compliance Testing | 1 Hour |  |  |
| User Acceptance Testing | 1 Hour |  |  |
| Test Report Preparation | 1 day | 03/07/2023 | 04/07/2023 |

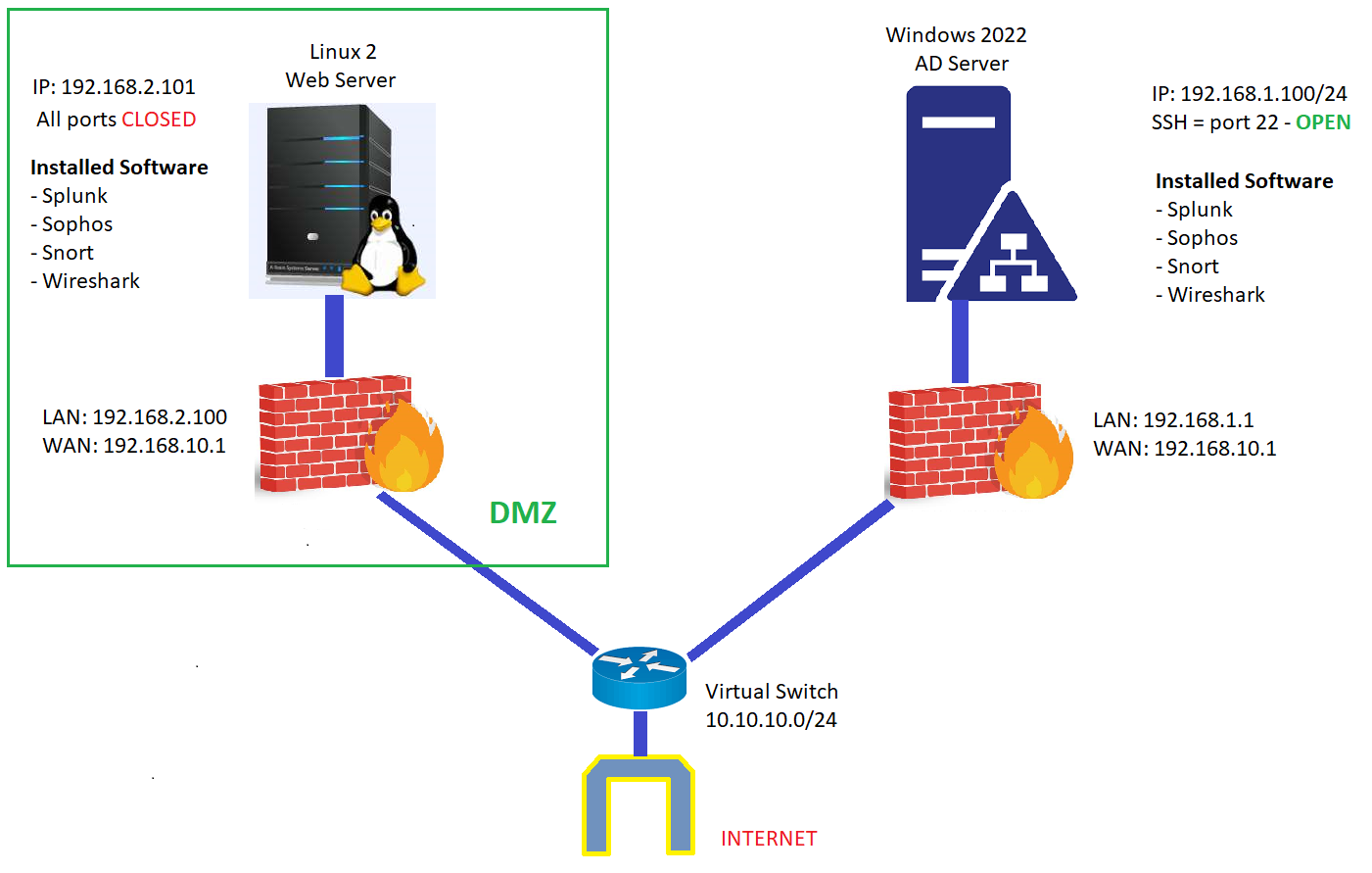
## Schedule for Testing Tasks and Test Milestones:

Based on the project schedule and resource availability, specific dates will be assigned to each testing task and milestone. The schedule will be determined in collaboration with the project team and stakeholders.

## Reporting and Documentation:

Throughout the testing process, comprehensive reports will be prepared to document the test plans, test cases, test data, test results, and any identified issues or recommendations. These reports will serve as a reference for stakeholders and assist in tracking the progress and outcomes of the testing effort.

|  |
| --- |
| Network Diagram |



|  |
| --- |
| HARDWARE REQUIREMENTS |

**Computers:**

**Architecture**: The computers should have a compatible architecture that supports the targeted operating systems and software applications for the end-to-end test. In this case, based on the mentioned Windows and Linux servers, the recommended architecture would be x86-64 (64-bit) as it is widely supported by both Windows and Linux distributions.

**Processing Power**: Sufficient processing power is necessary to handle the workload of the infrastructure and software being tested. The specific processing power requirement will depend on the scale and complexity of the test environment. A multi-core processor with a clock speed of at least 2.5 GHz or higher is typically recommended for effective performance.

**Memory**: Adequate memory is crucial for efficient operation and to handle the demands of the test environment. The recommended minimum memory requirement would be 8 GB or higher. However, the actual memory requirement may vary based on factors such as the number of virtual machines, the size of the test data, and the specific software being used.

**Secondary Storage**: Sufficient secondary storage, such as hard disk drives (HDD) or solid-state drives (SSD), is needed to accommodate the operating systems, virtual machines, software applications, and test data. The storage capacity required will depend on the size of the test environment and the amount of data that needs to be stored. It is recommended to have ample storage space to accommodate the needs of the end-to-end test.

**Display Adapter**: A suitable display adapter is required to connect the computers to monitors or display devices. The specific requirements for the display adapter depend on the desired display resolution and graphics capabilities. Most modern computers come with built-in graphics capabilities that can meet the requirements of typical infrastructure testing.

|  |
| --- |
| ENVIRONMENT REQUIREMENTS |

To conduct the end-to-end test of the infrastructure migration to a cloud environment, the following environment requirements should be considered:

## Virtualization Environment:

**Virtualization Software**: Install and configure a virtualization platform such as Hyper-V or VMware. Ensure compatibility with the chosen operating systems and virtualization needs.

**Hypervisor**: Set up the hypervisor software that manages the virtual machines and their resources. Ensure it supports the selected virtualization software.

## Networking:

**Network Infrastructure**: Ensure a stable and reliable network infrastructure is in place to support communication between the different components of the infrastructure.

**Virtual Networking**: Set up virtual networks to simulate the connectivity and network segmentation required for the test environment. This includes configuring network switches, routers, and virtual switches.

## Operating Systems:

**Windows 2022 AD Server**: Install and configure the Windows Server 2022 operating system for the Active Directory (AD) server. Ensure proper configuration of domains, users, and security policies.

**Linux 2 Web Server**: Install and configure the Linux 2 operating system for the web server. Set up necessary packages, web server software (e.g., Apache, Nginx), and configure security settings.

## Security Systems:

**Firewall**: Set up and configure the pfSense firewall for the Windows network and the Sophos firewall for the Linux network. Configure rules and policies to control inbound and outbound traffic.

**Antivirus/Malware Protection**: Install and configure Sophos Intercept X as the antivirus and malware protection software for both Windows and Linux servers.

**Intrusion Detection/Prevention System (IDS/IPS):** Set up and configure Snort IDS/IPS to monitor and detect potential security threats within the networks.

**Log Monitoring and Analysis:** Install and configure Splunk to monitor and analyze logs generated by the AD server, host machines, and other relevant components.

## Monitoring and Analysis Tools:

**Real-time Monitoring**: Utilize Wireshark to perform real-time monitoring of network traffic and analyze packets for troubleshooting and security analysis.

**Performance Monitoring**: Consider using performance monitoring tools to assess the performance and resource utilization of the virtualized infrastructure.

## Test Data and Scenarios:

Prepare representative test data that simulates real-world scenarios to thoroughly test the functionality and performance of the infrastructure.

## Documentation and Reporting:

Maintain proper documentation of the environment setup, configurations, test plans, test cases, and test results to ensure traceability and facilitate reporting.

|  |
| --- |
| TEST SCHEDULE |

## Detailed Test Schedule:

|  |  |
| --- | --- |
| Test Milestones and Item Transmittal Events: | |
| Test Plan Approval: | The approved test plan is the baseline for all testing activities. |
| Test Environment Setup: | The test environment is prepared with the necessary infrastructure and configurations. |
| Test Data Preparation: | Test data is generated or obtained for various test scenarios. |
| Test Case Development: | Test cases are designed and documented based on the test objectives and requirements. |
| Test Case Review: | Test cases are reviewed and validated for accuracy and completeness. |
| Test Execution: | Actual testing activities are performed based on the test cases. |
| Defect Reporting: | Any identified defects are logged and reported for resolution. |
| Defect Verification: | Resolved defects are retested to ensure their proper resolution. |
| Test Completion: | All planned testing activities are completed, and test deliverables are finalized. |

|  |  |
| --- | --- |
| Estimation of Time Required for Testing Tasks: | |
| Test Environment Setup: | 3 days |
| Test Data Preparation: | 2 days |
| Test Case Development: | 1days |
| Test Case Review: | 1 day |
| Test Execution: | Based on the complexity and coverage of the test cases, estimate 7 days |
| Defect Reporting: | Ongoing throughout the test execution phase |
| Defect Verification: | As defects are resolved, estimate 1 day |
| Test Completion: | Once all planned testing activities are executed and verified, estimate 7 days |
| Test Progress Review: | Conducted weekly, estimate 24 hours per review |
| Test Report Preparation | 2 days |

### Schedule for Testing Resources:

* **Facilities:** Dedicated testing environment available throughout the testing phase.
* **Tools:** Testing tools (e.g., penetration testing frameworks, vulnerability scanners, log analysis tools) available throughout the testing phase.
* **Staff:** Testing team available during the designated testing tasks and milestones, allocated according to the estimated time required for each task.

It is important to note that the specific dates for each testing task and milestone should be determined based on the project schedule and available resources. The estimated time for each testing task may vary depending on the project's complexity and the size of the network infrastructure being tested. Regular communication and coordination with stakeholders and the project team will ensure alignment and adjustments to the test schedule as necessary.

|  |
| --- |
| CONTROL PROCEDURES |

## Problem Reporting:

When encountering an incident during the testing process, the following procedures should be followed:

1. **Incident Identification**: Identify and document the specific details of the incident, including the symptoms, error messages, and any relevant system or application behaviors.
2. **Incident Logging**: Use the designated incident reporting form or system to log the incident. Include information such as the date, time, severity level, description of the issue, steps to reproduce, and any supporting attachments (e.g., screenshots, log files).
3. **Incident Prioritization**: Assign a priority level to the incident based on its impact on the system functionality, security, and overall test objectives. Priority levels can range from critical to low, indicating the urgency of resolution.
4. **Incident Ownership**: Assign the incident to the appropriate team member responsible for investigating and resolving the issue. Clearly communicate the assignment to ensure accountability.
5. **Incident Investigation**: The assigned team member should thoroughly investigate the incident by reproducing the issue, analyzing logs and error messages, and examining relevant system configurations and data.
6. **Incident Resolution**: Based on the investigation, the team member should determine the root cause of the incident and develop an appropriate solution. The resolution should be implemented, tested, and verified for effectiveness.
7. **Incident Verification**: Once the resolution is implemented, conduct retesting to verify if the incident has been resolved. Validate that the expected system behavior is restored and that no new issues are introduced.
8. **Incident Closure**: Close the incident in the incident reporting form or system after successful resolution and verification. Provide details of the resolution, including any workaround or permanent fix implemented.

## Change Requests:

For modifications to the software, the following change request process should be followed:

1. **Change Identification**: Identify the specific modification needed, including the rationale, desired outcomes, and any relevant supporting information.
2. **Change Request Documentation**: Document the change request, including a clear description of the requested change, the impacted software components/modules, and any associated risks or dependencies.
3. **Change Request Review**: Review the change request with relevant stakeholders, including project managers, developers, and other team members as appropriate. Evaluate the feasibility, impact, and potential benefits of the requested change.
4. **Change Request Approval**: Obtain approval for the change request from the designated authority or change control board. The approval should consider factors such as the alignment with project objectives, resource availability, and the impact on the current product.
5. **Change Implementation**: Upon approval, the change should be implemented following established software development and release management processes. Identify the responsible team members and schedule the implementation accordingly.
6. **Change Testing**: Conduct appropriate testing, including regression testing, to ensure that the modified software functions correctly and does not introduce new issues. Test the affected modules to verify their compatibility with the changes.
7. **Change Integration**: If the change affects existing programs or modules, ensure proper integration and compatibility with other software components. Update any affected documentation or user guides as necessary.
8. **Change Verification**: Validate the successful implementation of the change through thorough testing and user acceptance. Verify that the modified software meets the defined criteria and objectives.
9. **Change Closure**: Close the change request after successful implementation, verification, and user acceptance. Document the details of the change, including any impact analysis and modifications made.

|  |
| --- |
| FEATURES TO BE TESTED |

## Windows 2022 AD Server:

* **User and Group Management**: Test the creation, modification, and deletion of users and groups within the Active Directory server.
* **Authentication and Authorization**: Verify the functionality of user authentication and authorization processes, including password policies and access controls.
* **Domain Controller Operations**: Test the replication, synchronization, and failover capabilities of the domain controllers.
* **Group Policy Management**: Validate the application and enforcement of group policies across the Windows network.
* **Two-Factor Authentication (2FA):** Test the functionality and security of the 2FA implementation using Azure AD services.

## Linux 2 Web Server:

* **Web Server Functionality**: Validate the proper installation and configuration of the Linux web server software (e.g., Apache, Nginx).
* **Content Delivery**: Test the delivery of web content, including HTML pages, images, CSS files, and JavaScript.
* **Secure Communication**: Verify the implementation and functionality of secure communication protocols (e.g., HTTPS) for web traffic.
* **Performance and Scalability**: Evaluate the web server's performance and scalability under different load conditions and traffic patterns.
* **Application Integration**: Test the integration of web applications or services hosted on the Linux web server.

## 

## Security Systems:

* **pfSense Firewall**: Validate the firewall rules and policies to ensure proper network traffic filtering and protection against unauthorized access.
* **Sophos Firewall**: Test the functionality of the Sophos firewall, including traffic filtering, VPN connectivity, and intrusion prevention.
* **Sophos Intercept X**: Verify the effectiveness of the antivirus and malware protection provided by Sophos Intercept X.
* **Snort IDS/IPS**: Validate the intrusion detection and prevention capabilities of Snort, including the detection of malicious network activities.
* **Splunk Log Monitoring**: Test the log monitoring and analysis functionality of Splunk, including the detection of security events and anomalies.

## 

## Network Segmentation and DMZ Configuration:

* Verify the isolation and security measures implemented between the Windows network and the Linux web server network segment.
* Test the network connectivity and communication between the different network segments, ensuring proper access controls and segregation.

## 

## Integration and Communication:

* Validate the integration and communication between the Windows AD server and the Linux web server, ensuring seamless authentication and authorization processes.
* Test the interoperability of different software components, ensuring proper data exchange and functionality.

## 

## Compliance:

* Validate compliance with coding standards, best practices, and industry regulations related to security and privacy.
* Verify the implementation of security controls and safeguards to meet industry-specific requirements.

|  |
| --- |
| FEATURES NOT TO BE TESTED |

**Third-Party Software Dependencies**:

Features or functionalities of third-party software or applications that are outside the scope of the infrastructure migration test. The focus is primarily on the core Windows and Linux components and their interactions within the cloud environment.

**Hardware-Specific Functionality**:

Features or functionalities that are dependent on specific hardware components or configurations. The testing is focused on the software aspects of the infrastructure migration and not on the hardware itself.

**End-User Application Testing**:

Features or functionalities of end-user applications that are hosted on the web server or accessed through the infrastructure. The testing is primarily focused on the infrastructure and its components rather than individual applications or services.

**Performance Testing at Maximum Scalability**:

Testing the infrastructure's performance at the maximum possible scalability or under extreme load conditions. The focus is on evaluating the performance within realistic usage scenarios rather than pushing the infrastructure to its absolute limits.

**Compatibility with Legacy Systems**:

Testing compatibility with legacy systems or older versions of software that are not part of the current infrastructure migration project. Compatibility testing is limited to the specific versions and configurations relevant to the project.

**Operating System-Specific Features**:

Features or functionalities that are specific to a particular operating system and are not applicable to both Windows and Linux systems. The testing is centered around the common functionality and integration between the two platforms.

**Long-Term Stability Testing**:

Testing the long-term stability of the infrastructure over extended durations (e.g., months or years). The focus is on ensuring immediate functionality, performance, and security, rather than long-term durability.

**Software Patch and Update Testing**:

Testing the impact of future software patches, updates, or version upgrades. The focus is on testing the current version and configuration of the infrastructure, and any future updates will be considered as separate testing activities

|  |
| --- |
| RESOURCES/ROLES & RESPONSIBILITIES |

|  |  |
| --- | --- |
| **Project Manager  Giuseppe Raciti):** | Overall responsibility for the test project.  Manages the project schedule, budget, and resources.  Coordinates with stakeholders and ensures timely communication |
| **Cyber Security Specialist (Shaun Heywood):** | Responsible for identifying security testing requirements.  Designs and executes security testing activities.  Analyzes and reports on security vulnerabilities and risks |
| **Cloud Architect/Engineer (Mark Byrne):** | Designs and sets up the test environment in the cloud.  Configures virtualization platforms and networking components.  Collaborates with the server administrator to ensure proper integration. |
| **Server Administrator**  **(Mauricio Guerra):** | Sets up and configures the Windows 2022 AD Server.  Manages the user accounts, permissions, and domain services.  Collaborates with the cloud architect/engineer for environment setup. |

|  |
| --- |
| SCHEDULES |

## Major Deliverables:

**Test Plan**: This document outlines the overall testing approach, objectives, scope, and strategies. It includes details such as the testing objectives, test environments, test schedule, test resources, and the overall test approach for the infrastructure migration project.

**Test Cases**: These are detailed documents that define specific test scenarios, inputs, expected outcomes, and steps to be executed during testing. Test cases cover different aspects of the infrastructure, including functionality, security, performance, and integration. They serve as a guide for executing tests and capturing test results.

**Test Incident Reports**: These reports document any issues or incidents encountered during testing. They include detailed descriptions of the incident, steps to reproduce it, and any relevant supporting information such as error messages or screenshots. Test incident reports help track and prioritize the resolution of issues discovered during testing.

**Test Summary Reports**: These reports provide an overview of the test activities, outcomes, and findings. They summarize the test execution progress, identify the number and severity of issues discovered, and provide an overall assessment of the tested infrastructure's quality and readiness for migration. Test summary reports help stakeholders understand the test results and make informed decisions based on the testing outcomes.

The completion of these deliverables is crucial for effective test planning, execution, and reporting. They provide a structured and documented approach to ensure comprehensive testing coverage and facilitate clear communication of test results and findings to stakeholders.

|  |
| --- |
| SIGNIFICANTLY IMPACTED DEPARTMENTS (SIDs) |

## IT Department:

* **Business Manager**: Responsible for overseeing the IT department's operations and ensuring alignment with the organization's goals.
* Cloud Architect/Engineer: Responsible for designing and implementing the infrastructure migration to a cloud environment.
* Server Administrator: Responsible for configuring and managing the Windows 2022 AD server and associated services.
* Tester(s): Involved in executing test cases and verifying the functionality and performance of the IT infrastructure.

**Cybersecurity Department**:

* Business Manager: Oversees the cybersecurity department's activities and ensures the security of the infrastructure during the migration.
* Cyber Security Specialist: Responsible for identifying and assessing security risks, designing security measures, and executing security testing activities.
* Tester(s): Involved in performing security testing to identify vulnerabilities and ensure the infrastructure's resilience against potential threats.

**Operations Department:**

* Business Manager: Manages the operations department and ensures the smooth functioning of the organization's IT infrastructure.
* Operations Staff: Responsible for the day-to-day management, configuration, and maintenance of the hardware, networking components, and infrastructure systems.
* Tester(s): Collaborates with the testing team to address operational issues and verify the infrastructure's stability during the migration process.

**Development Department:**

* Business Manager: Oversees the development department's activities and ensures the successful implementation of the infrastructure migration project.
* Development Team: Provides support during the testing process, addresses software-related issues, and collaborates with the testing team to resolve any identified defects.

**Testing Services (if applicable):**

* Business Manager: Oversees the testing services' activities and ensures the timely completion of testing tasks.
* External Testing Service Providers: Engaged for specific testing activities such as security testing, performance testing, or compliance testing.

|  |
| --- |
| DEPENDENCIES |

## Significant Constraints on Testing:

One of the significant constraints on testing is the availability of test items. This includes test environments and test data. The availability of appropriate test environments, such as virtual machines and necessary software configurations, is crucial for executing test cases. Delays in setting up or accessing the required test environments can impact the timely execution of test cases and overall testing progress. Additionally, the availability and readiness of representative test data sets are essential for comprehensive testing. If test data preparation is delayed or incomplete, it may hinder the ability to execute test cases effectively and thoroughly.

Another constraint is the availability of testing resources. Skilled and experienced testers are necessary for conducting thorough and efficient testing. Limited availability of testers with the necessary domain knowledge and technical expertise may result in delays or compromises in test coverage. Additionally, the availability and accessibility of testing tools, licenses, and necessary infrastructure resources (such as hardware and software) can impact the efficiency and effectiveness of testing. Limited availability of required resources may lead to delays or limitations in testing activities.

Deadlines and time constraints are also significant factors. Project deadlines may impose time constraints on testing activities, requiring completion within specific timeframes to align with the overall project schedule. Furthermore, time constraints within each testing phase, such as unit testing, integration testing, and system testing, may impact the thoroughness and completeness of testing. Limited time for each phase may necessitate prioritization and optimization of testing efforts to ensure critical areas are adequately covered. Additionally, the availability of time for defect resolution and retesting after issues are identified can impact the overall testing timeline. Delays in resolving defects may affect the project schedule and require adjustments to ensure timely completion.

Managing these constraints requires proactive planning, resource allocation, and close collaboration among the testing team, project stakeholders, and other relevant parties. Effective communication of these constraints is essential to ensure that expectations and limitations are clearly understood. Mitigation strategies, such as prioritization, efficient resource allocation, and timely communication, can help address these constraints and ensure successful testing within the given limitations. It is crucial for project managers to regularly monitor and address these constraints to minimize their impact on the testing process and overall project timeline.

|  |
| --- |
| RISKS/ASSUMPTIONS |

## Risk/Assumption: Delay in Delivery of Test Items

**Contingency Plan**: In the event of a delay in the delivery of test items, such as test environments or necessary software configurations, the contingency plan involves increased night shift scheduling. This ensures that the testing team can compensate for the delay by working extended hours during off-peak times. The additional night shifts would allow for catching up on the testing activities and help meet the overall delivery date without compromising the quality of the testing process.

**Risk/Assumption: Insufficient Test Data Availability**

**Contingency Plan**: If there is a lack of sufficient test data for comprehensive testing, the contingency plan involves creating synthetic test data or utilizing anonymized production data. Synthetic test data can be generated to simulate realistic scenarios and cover a wider range of test cases. Alternatively, anonymized production data can be used to ensure the availability of representative data while adhering to privacy and security regulations. The contingency plan allows for testing to proceed even in the absence of complete or ideal test data.

**Risk/Assumption: Unavailability of Skilled Testers**

**Contingency Plan**: In the event of limited availability of skilled testers, the contingency plan involves cross-training team members or leveraging external testing resources. Cross-training team members in specific testing areas allows for the redistribution of testing responsibilities and ensures that multiple team members can contribute effectively. If necessary, external testing resources, such as specialized testing services or contractors, can be engaged to supplement the internal testing team and provide additional expertise and capacity.

**Risk/Assumption: Changes in Project Scope or Requirements**

**Contingency Plan**: In the case of changes in the project scope or requirements, the contingency plan involves a robust change management process. This process includes conducting impact analysis to assess the effect of the changes on the testing activities, timelines, and resources. It also involves appropriate communication and coordination with stakeholders to understand the implications and determine the necessary adjustments to the test plan. The contingency plan ensures that any changes are properly evaluated, documented, and incorporated into the testing process to maintain alignment with the evolving project scope and requirements.

**Risk/Assumption: Critical Defects Found during Testing**

**Contingency Plan**: If critical defects are discovered during testing that impact the overall project timeline or functionality, the contingency plan involves prioritizing defect resolution and retesting. The plan includes allocating dedicated resources and adjusting priorities to address the critical defects promptly. It may involve the reallocation of development resources or engaging additional support to expedite the resolution process. The contingency plan ensures that the necessary steps are taken to minimize the impact of critical defects on the project timeline and deliverables.

|  |
| --- |
| TOOLS |

**Virtualization Software**: Hyper-V, VMware

These tools will be used to create and manage virtual machines and virtual networks for setting up the test environments.

Hypervisor:

**Network Monitoring and Analysis**:

**Wireshark**: This tool will be used for real-time monitoring and analysis of network traffic to identify any issues or abnormalities.

Security Tools:

**pfSense Firewall**: This firewall software will be used for network traffic filtering, access control, and protection against unauthorized access on the Windows network.

**Sophos Firewall**: This firewall software will be used for network traffic filtering, VPN connectivity, and intrusion prevention on the Linux network.

**Snort IDS/IPS**: This intrusion detection and prevention system will be used to monitor network traffic and detect potential security threats.

**Splunk**: This tool will be used for log monitoring and analysis to identify security events and anomalies on the AD server and host machines.

**Sophos Intercept X**: This antivirus and malware protection software will be used to ensure the security of the Windows and Linux servers.

**Bug Tracking Tool:**

**JIRA, Bugzilla, or Trello**: A bug tracking tool will be used to capture, track, and manage identified defects or issues throughout the testing process. The specific bug tracking tool to be used will depend on the organization's preference and existing systems.

**Test Management Tool**:

**TestRail, qTest, or Zephyr**: A test management tool will be used to organize and manage test cases, track test execution progress, and generate test reports. The specific test management tool will depend on the organization's preference and available resources.

**Performance Testing Tools**:

**Apache JMeter, LoadRunner**: Performance testing tools may be utilized to evaluate the performance and scalability of the infrastructure under different load conditions and simulate real-world scenarios.

**Collaboration and Communication Tools**:

Microsoft Teams, Slack, or Jira: These tools will be used for effective collaboration, communication, and task management among team members involved in the testing process.

|  |
| --- |
| TEST STEPS |

|  |
| --- |
| Windows 2022 AD Server |

**Target User Account Selection**:

Select a specific user account within the Windows AD server as the target for the brute force attack. This account should have sufficient privileges to access critical resources or sensitive data.

**Complexity Requirements**:

Determine the complexity requirements for the passwords used in the attack. This includes the minimum password length, the inclusion of uppercase and lowercase letters, numbers, and special characters.

**Number of Login Attempts**:

Define the number of login attempts to be made during the brute force attack. This could include a specific range of attempts or a continuous attempt until a specific condition is met.

**Test Execution**:

Initiate the brute force attack by attempting to gain unauthorized access to the Windows AD server using various password combinations that meet the complexity requirements.

Use automated tools or scripts to simulate multiple login attempts, systematically iterating through possible password combinations.

**Real-Time Monitoring**:

Activate the Snort IDS/IPS, pfSense Firewall, and Sophos Firewall to actively monitor the network traffic and system activities during the brute force attack.

Monitor the incoming login requests, patterns, and behavior for any suspicious activity.

**Alert and Response Verification**:

Verify that the security systems (Snort IDS/IPS, pfSense Firewall, Sophos Firewall) generate alerts when potential brute force attack patterns are detected.

Validate that the systems respond promptly by blocking further login attempts from the attacker's IP address or implementing other configured security measures.

**Log Analysis in Splunk**:

Collect the logs generated by the security systems and feed them into Splunk for real-time monitoring and analysis.

Analyze the logs to identify the specific log entries related to the failed brute force attack attempts.

Examine the timestamps, source IP addresses, and other relevant information to gain insights into the attack.

**Validation of Results**:

Validate the accuracy and effectiveness of the security systems in detecting and responding to the brute force attack.

Verify that the security systems accurately identified the attack patterns, triggered appropriate alerts, and took necessary actions to block further login attempts from the attacker's IP address.

Investigate any false positives or false negatives and determine the root causes to improve the security systems' configurations or rules if necessary.

|  |
| --- |
| Linux Web Server |

**Target User Account Selection**:

* Select a user account within the Linux DMZ network that represents a typical user with average privileges. Ensure the account has a password that meets complexity requirements to assess the effectiveness of password security measures.

**Complexity Requirements**:

* Define the complexity requirements for the user account's password, including minimum length, combination of uppercase and lowercase letters, numbers, and special characters. Ensure the password meets the organization's password policy standards and represents a realistic scenario.

**Test Execution**:

* Use a port scanning tool, such as Nmap, to identify open ports and services on the selected Linux system in the DMZ.
* Conduct a comprehensive scan to determine the accessibility of the system and identify any potential vulnerabilities.
* Observe the behavior of the Sophos firewall and Snort IDS/IPS system during the port scan, noting if they detect and generate alerts for the scanning activity.
* Evaluate the performance of the Sophos Intercept X antivirus/malware protection in detecting and blocking any potential threats identified during the scan.
* Monitor the network traffic using Wireshark to analyze the port scan activity in real-time and identify any unauthorized or suspicious activities.

**Real-Time Monitoring**:

* Utilize real-time monitoring tools, such as Wireshark, to capture and analyze network traffic during the port scan test.
* Monitor for any unusual or unauthorized network activity, such as unexpected incoming connections or suspicious data transfers.
* Assess the effectiveness of real-time monitoring in detecting and alerting on potential security threats.

**Alert and Response Verification**:

* Ensure that alerts are generated by the Sophos firewall, Snort IDS/IPS system, and any other relevant security systems in response to the port scan activity.
* Verify that the alerts are appropriately prioritized, categorized, and communicated to the relevant personnel or security teams.
* Validate the response measures taken to mitigate the port scan, such as blocking the source IP address or triggering additional security protocols.

**Log Analysis in Splunk**:

* Configure Splunk to collect and analyze logs from the Linux server and relevant security systems in the DMZ.
* Monitor and analyze the logs during the port scan test, focusing on firewall logs, IDS/IPS logs, and any other pertinent logs.
* Identify any anomalies, suspicious activities, or security events related to the port scan and assess Splunk's effectiveness in detecting and alerting on such events.

**Validation of Results**:

* Compare the observed results with expected outcomes and documented security policies.
* Evaluate if the Sophos firewall, Snort IDS/IPS system, Sophos Intercept X, real-time monitoring tools, and log analysis in Splunk performed as intended.
* Identify any vulnerabilities, weaknesses, or areas for improvement in the Linux DMZ network's security systems.
* Document the test findings, including any recommendations for remediation or enhancements to the security infrastructure.

|  |
| --- |
| TEST OUTCOME |

|  |
| --- |
| Windows 2022 AD Server |

During the Brute Force Attack: The attack attempts were made using various password combinations on the Windows AD server. The firewall systems, Snort IDS/IPS, and Splunk were actively monitoring the network traffic and system activities. The logs generated indicated multiple failed login attempts and triggered alerts.

Detection and Blocking: The security systems (Snort IDS/IPS, pfSense Firewall, Sophos Firewall) detected the brute force attack in real-time. The systems analyzed the attack patterns and identified the malicious activity. As a result, the systems blocked further login attempts from the attacker's IP address.

Splunk Log Analysis: The logs generated by the security systems were collected and analyzed in Splunk. The failed brute force attack logs were successfully correlated and identified, providing an overview of the attack and the actions taken by the security systems.

|  |
| --- |
| Linux Web Server |

During the simulated port scan on the Linux network in the DMZ, the following events were observed:

**Port Scan Activity**: The Nmap port scanning tool was used to identify open ports and services on the selected Linux system. The scan revealed that several ports were open, including port 22 (**SSH**), port 80 (**HTTP**), and port 443 (**HTTPS**).

**Sophos Firewall:** The Sophos firewall detected the port scan activity and generated an alert. The alert was categorized as a high-priority event and was communicated to the security team. The firewall automatically blocked further incoming connections from the source IP address.

**Snort IDS/IPS:** The Snort IDS/IPS system analyzed the network traffic during the port scan and identified it as potentially malicious activity. It generated an alert, classifying the event as an "**Nmap Port Scan Detected**." The alert was forwarded to the security team for further investigation.

**Splunk Log Analysis**: Splunk collected and analyzed logs from the Linux server, firewall, and IDS/IPS system. The log analysis revealed entries related to the port scan activity, indicating the source IP address, timestamp, and the detected scanning patterns. Splunk generated an alert based on predefined correlation rules, highlighting the potential security breach.

|  |
| --- |
| CONCLUSION: |

The brute force test on the Windows AD server was conducted to assess the effectiveness of the security systems in detecting and blocking unauthorized access attempts. The integration of Snort IDS/IPS, pfSense Firewall, Sophos Firewall, and Splunk enabled real-time monitoring, detection, and log analysis.

The test successfully demonstrated that the security systems effectively detected the brute force attack and triggered appropriate responses. The firewall systems blocked further login attempts from the attacker's IP address, preventing unauthorized access to the Windows AD server.

The logs generated during the test were analyzed in Splunk, providing a consolidated view of the detected attack and the actions taken by the security systems. The test results confirmed the robustness and effectiveness of the security measures implemented in the infrastructure.

Overall, the test showcased the importance of integrating multiple security systems, such as intrusion detection systems, firewalls, and log monitoring, to provide layered protection and promptly respond to potential threats. The successful outcome of the test validated the security controls in place and demonstrated the infrastructure's resilience against brute force attacks.

|  |
| --- |
| Approvals: |
| |  |  |  | | --- | --- | --- | |  |  |  | | Project Manager Signature |  | Sponsor Or Originator Signature | |  |  |  | | Project Manager Name |  | Sponsor Or Originator Name | |  |  |  | | Date |  | Date | |