Model Curriculum

QP Name: Application Security Analyst

QP Code: SSC/Q0903

QP Version: 4.0

NSQF Level: 5

Model Curriculum Version: 4.0

IT-ITeS Sector Skill Council || IT-ITeS Sector Skill Council, NASSCOM, Plot No - 7, 8, 9 & 10, 3rd Floor, Sector 126, Noida
Uttar Pradesh – 201303

Table of Contents

Training Parameters 4

Program Overview 6

Training Outcomes 6

Compulsory Modules 6

Module Details 9

Module 1: Application Security and their Fundamental Concepts 9

Module 2: Application Vulnerabilities 11

Module 3: Application Security Testing and Documentation 14

Module 4: Web Server, Application Security & Active Directory Hardening 16

Module 5: Fundamentals of Cybersecurity 18

Module 6: API and Application Security Practices 20

Module 7: Application Security and Vulnerability Management 22

Module 8: IT Infrastructure Security Management 24

Module 9: PowerShell for System Administration and Log Management 26

Module 10: Application Security Assessment and Threat Management 28

Module 11: Risk Assessment and Incident Management 30

Module 12: Ticket Management and Resolution 32

Module 13: Security Compliance and Monitoring 33

Module 14: Introduction to Employability Skills 35

Module 15: Constitutional values - Citizenship 35

Module 16: Becoming a Professional in the 21st Century 35

Module 17: Basic English Skills 36

Module 18: Career Development and Goal Setting 36

Module 19: Communication skills 36

Module 20: Diversity and Inclusion 37

Module 21: Financial and Digital Literacy 37

Module 22: Essential Digital Skills 37

Module 23: Entrepreneurship 38

Module 24: Customer Service 38

Module 25: Getting Ready for Apprenticeship and Jobs 38

Annexure 39

Trainer Requirements 39

Assessor Requirements 39
Assessment Strategy 40
Recommended Supplemental Readings 43
References 44
Glossary 44
Acronyms and Abbreviations 45

Training Parameters

Sector	IT-ITeS
Sub-Sector	Future Skills
Occupation	Information and Cyber Security
Country	India
NSQF Level	5
Aligned to NCO/ISCO/ISIC Code	NCO-2015/ NIL
Minimum Educational Qualification and Experience	*Relevant Experience in job roles related in IT/Computer Science/Cybersecurity. The relevant experience would include work, internship, and apprenticeship after completing relevant educational qualifications. ** UG or diploma with courses related to Engg./ Science Completed 2nd year of 3-year/ 4-year UG** OR Completed 3-Year Diploma** after 10th with 1.5 year of relevant experience* OR Previous Relevant qualification of NSQF level 4 with 3 years of relevant experience*
Pre-Requisite License or Training (suggested but not mandatory)	NA
Minimum Job Entry Age	18
Last Reviewed On	18/02/2025
Next Review Date	18/02/2028
NSQC Approval Date	18/02/2025
QP Version	4.0
Model Curriculum Creation Date	18/02/2025
Model Curriculum Valid Up to Date	18/02/2028
Model Curriculum Version	4.0

Minimum Duration of the Course	510 hours
Maximum Duration of the Course	510 hours

Program Overview

This section summarizes the end objectives of the program along with its duration.

Training Outcomes

At the end of the program, the learner should have acquired the listed knowledge and skills.

- Define key cybersecurity terms such as information security, cyber security, threat, vulnerability, and risk, demonstrating comprehension of foundational concepts in the field.
- Identify and explain the interrelationship between confidentiality, integrity, and availability (CIA triad), applying these principles to secure information systems effectively.
- List common types of cyber threats, including malware, phishing, DDoS attacks, and ransomware, and assess their potential impact and likelihood of occurrence in practical situations.
- Identify fundamental security protocols (e.g., SSL/TLS, HTTPS, encryption methods) and describe their roles in safeguarding data across applications and networks.
- Explain the importance of adhering to security standards such as ISO 27001, NIST Cybersecurity Framework, and GDPR in maintaining security compliance, alongside implementing rate limiting and authentication mechanisms in API gateways.
- Identify security misconfigurations in sample application environments and suggest corrective actions to mitigate potential vulnerabilities.
- Use PowerShell scripts to enhance security features in application environments, demonstrating practical scripting skills.
- Describe the role of firewalls in protecting applications and databases, assessing various application and database defense mechanisms.
- Explain how to utilize threat intelligence for threat identification and evaluate emerging threats and their implications for cybersecurity.
- Discuss the importance of logging incidents in ticketing systems and describe the processes for classifying service requests according to organizational policies.

Compulsory Modules

The table lists the modules and their duration corresponding to the Compulsory NOS of the QP.

NOS and Module Details	Theory Duration	Practical Duration	On-the-Job Training Duration (Mandatory)	On-the-Job Training Duration (Recommended)	Total Duration
SSC/N0958- Access API and application for security NOS Version No. 1 NSQF Level 5	**50:00**	**70:00**	**30:00**	**00:00**	**150:00**
Module 1: Application Security and their Fundamental Concepts	10:00	20:00	10:00	00:00	40:00
Module 2: Application Vulnerabilities	30:00	30:00	10:00	00:00	70:00

Module 3: Application Security Testing and Documentation	10:00	20:00	10:00	00:00	40:00
SSC/N0959– Manage application security, hardening and vulnerability NOS Version No. 1 NSQF Level 5	**50:00**	**70:00**	**30:00**	**00:00**	**150:00**
Module 4: Web Server, Application Security & Active Directory Hardening	10:00	20:00	05:00	00:00	35:00
Module 5: Fundamentals of Cybersecurity	10:00	10:00	05:00	00:00	25:00
Module 6: API and Application Security Practices	10:00	20:00	05:00	00:00	35:00
Module 7: Application Security and Vulnerability Management	10:00	10:00	05:00	00:00	25:00
Module 8: IT Infrastructure Security Management	10:00	10:00	10:00	00:00	30:00
SSC/N0960– Oversee the Cloud security of deployed applications and solutions to detect potential breaches and compromises NOS Version No. 1 NSQF Level 5	**50:00**	**70:00**	**30:00**	**00:00**	**150:00**
Module 9: PowerShell for System Administration and Log Management	10:00	10:00	05:00	00:00	25:00
Module 10: Application Security Assessment and Threat Management	15:00	30:00	05:00	00:00	50:00
Module 11: Risk Assessment and Incident Management	10:00	10:00	05:00	00:00	25:00
Module 12: Ticket Management and Resolution	05:00	10:00	05:00	00:00	20:00
Module 13: Security Compliance and Monitoring	10:00	10:00	10:00	00:00	30:00
Employability Skill 60 Hours DGT/VSQ/N0102 NOS Version No. 1 NSQF Level 4	**24:00**	**36:00**	**0:00**	**0:00**	**60:00**
Module 14: Introduction to Employability Skills	00:30	01:00	00:00	00:00	01:30
Module 15: Constitutional values – Citizenship	00:30	01:00	00:00	00:00	01:30
Module 16: Becoming a Professional in the 21st Century	01:00	01:30	00:00	00:00	02:30
Module 17: Basic English Skills	04:00	06:00	00:00	00:00	10:00
Module 18: Career Development & Goal Setting	01:00	01:00	00:00	00:00	02:00
Module 19: Communication Skills	02:00	03:00	00:00	00:00	05:00

Module 20: Diversity & Inclusion	01:00	01:30	00:00	00:00	02:30
Module 21: Financial and Legal Literacy	02:00	03:00	00:00	00:00	05:00
Module 22: Essential Digital Skills	04:00	06:00	00:00	00:00	10:00
Module 23: Entrepreneurship	03:00	04:00	00:00	00:00	07:00
Module 24: Customer Service	02:00	03:00	00:00	00:00	05:00
Module 25: Getting ready for apprenticeship & Jobs	03:00	05:00	00:00	00:00	08:00
Total Duration	**174:00**	**246:00**	**90:00**	**00:00**	**510:00**

Module Details

Module 1: Application Security and their Fundamental Concepts

Mapped to SSC/N0958 (Version 1)

Terminal Outcomes:

- Evaluate the effectiveness of various cybersecurity tools by comparing their strengths and weaknesses in real-world applications.
- Analyze given scenarios to identify potential cybersecurity threats and vulnerabilities, prioritizing them based on severity and likelihood of exploitation.
- Demonstrate the implementation of basic security controls, such as configuring firewalls and enabling multi-factor authentication, to enhance network security.

Duration: *10:00*	**Duration**: *20:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Define key terms such as information security, cyber security, threat, vulnerability, and risk. • Explain the difference between information security and cyber security, highlighting their scope and areas of application. • Identify and explain the relationship between confidentiality, integrity, and availability (CIA triad) in securing information systems. • Evaluate real-world scenarios where breaches of confidentiality, integrity, or availability have occurred, and assess the impact. • List common types of cyber threats, including malware, phishing, DDoS attacks, and ransomware. • Assess the severity of different cyber threats based on their potential impact and likelihood of occurrence. • Identify fundamental security protocols (e.g., SSL/TLS, HTTPS, encryption methods) used in safeguarding data.	• Analyze given scenarios or case studies and identify potential cybersecurity threats and vulnerabilities. • Perform a basic vulnerability assessment on a sample system to identify weaknesses in the security configuration. • Evaluate the risk posed by each identified vulnerability, prioritizing them based on severity and likelihood of exploitation. • Demonstrate the process of implementing basic security controls such as setting up firewalls, configuring antivirus software, and enabling multi-factor authentication (MFA) on a classroom network. • Review and modify security settings of a provided system, analyzing how each control improves security posture. • Demonstrate how to respond to common cyber threats by following provided protocols (e.g., handling a phishing email, securing a system from malware).

• Describe the importance of following security standards such as ISO 27001, NIST Cybersecurity Framework, and GDPR in maintaining security compliance. • Identify common cybersecurity tools such as firewalls, intrusion detection systems (IDS), antivirus software, and encryption methods. • Analyze how structured vs. unstructured data and large datasets influence application decision-making. • Prioritize specific data sources based on their importance to application functionality. • List common frameworks and programming languages (e.g., React, Angular, Spring). • Compare features of different frameworks to identify the most appropriate category for an application. • Evaluate how a programming language or framework affects application performance and usability. • Identify essential API security features like authentication and data validation. • Compare API authentication methods (e.g., OAuth, JWT) for preventing unauthorized access. • Assess how rate limiting and data validation impact API performance and security.	• Collect sample datasets from an application and assess their completeness and relevance. • Compare different data sources and recommend additional data collection methods if needed. • Classify sample applications based on their programming languages and frameworks.
Classroom Aids:	
Whiteboard and markers Chart paper and sketch pens LCD Projector and Laptop for presentations	
Tools, Equipment and Other Requirements	
Labs equipped with the following: • PCs/Laptops	

- Internet with Wi-Fi (Min. 2 Mbps dedicated)
- Samples of the templates and checklists used in organizations
- Static Application Security Testing (SAST) Tools: SonarQube
- Dynamic Application Security Testing (DAST) Tools: OWASP ZAP
- Software Composition Analysis (SCA) Tools: OWASP Dependency-Check
- Vulnerability Management Platforms: OpenVAS
- Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR
- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 2: Application Vulnerabilities

Mapped to SSC/N0958 (Version 1)

Terminal Outcomes:

- Analyze the impact of API vulnerabilities on application security by evaluating case studies and identifying potential risks.
- Implement rate limiting and authentication mechanisms in an API gateway setup to enhance security measures against unauthorized access.
- Evaluate the effectiveness of vulnerability scanning tools like Checkmarx or Snyk by interpreting scan results and proposing mitigation strategies for identified vulnerabilities.

Duration: *30:00*	**Duration**: *30:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Identify common vulnerabilities in API operations related to authentication, authorization, and data integrity. • Analyse the impact of vulnerabilities in API operations on overall application security. • Evaluate the effectiveness of targeted assessments in ensuring API security. • Explain the purpose of API gateway security measures like rate limiting and authentication. • Assess how these security measures improve API performance and security. • Explain the significance of security misconfigurations and supply chain vulnerabilities. • Compare best practices for resolving misconfigurations and addressing vulnerabilities. • List the key dependencies between application patching and IT infrastructure needs. • Analyze the interdependencies of patching and infrastructure for security management. • Describe the factors influencing vulnerability prioritization, such as business impact and exploitability.	• Perform targeted API assessments to identify vulnerabilities related to authentication and authorization in a provided API. • Implement rate limiting and authentication mechanisms in an API gateway setup. • Identify security misconfigurations in sample application environments and suggest corrective actions. • Utilize scanning tools like Checkmarx or Snyk to identify vulnerabilities in a sample application. • Interpret the results from vulnerability scanning tools and suggest mitigation steps. • Implement security measures in a Docker or Kubernetes environment to address vulnerabilities. • Review logs and metrics to evaluate the effectiveness of security measures applied to containerized setups. • Use PowerShell scripts to enhance security features in a given application environment. • Conduct threat modeling for a sample application and identify

• Justify the prioritization of specific vulnerabilities over others based on potential risk. • Explain the nature and characteristics of different types of vulnerabilities in information systems. • Analyze various attack vectors (e.g., social engineering, zero-day exploits) and the vulnerabilities they exploit. • Define the role of IaC (e.g., Terraform, CloudFormation) in securing cloud environments. • Compare different IaC templates in terms of security best practices and compliance requirements. • Identify advanced vulnerability scanning tools like Checkmarx, Veracode, and Snyk. • Analyze scan results to detect vulnerabilities and assess the risk posed to applications. • Describe common security vulnerabilities in containerized environments like Docker and Kubernetes. • Evaluate the effectiveness of security measures implemented in containerized environments. • Explain the importance of identifying root causes and architectural composition in vulnerability management. • Evaluate the probability of exposure and effectiveness of fixes in reducing vulnerabilities. • List key steps in the threat modeling process. • Analyze potential threats during software design to uncover vulnerabilities.	potential security weaknesses in the architecture.
Classroom Aids:	
Whiteboard and markers LCD Projector and Laptop for presentations	

Tools, Equipment and Other Requirements
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated) • Samples of the templates and checklists used in organizations • Static Application Security Testing (SAST) Tools: SonarQube • Dynamic Application Security Testing (DAST) Tools: OWASP ZAP • Software Composition Analysis (SCA) Tools: OWASP Dependency-Check • Vulnerability Management Platforms: OpenVAS • Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR • Programming languages like PHP, Java, Python, or Go etc. • Operating Systems: Linux, Windows.

Module 3: Application Security Testing and Documentation

Mapped to SSC/N0958 (Version 1)

Terminal Outcomes:

- Analyze the key components of an application tracking system to demonstrate its importance in capturing essential information for stakeholder decision-making.
- Design a comprehensive penetration testing plan that outlines the scope, methodologies, and tools for effectively assessing target systems.
- Evaluate the effectiveness of different testing methods, including black box, white box, and manual testing techniques, in identifying security vulnerabilities within applications.
- Integrate results from static, dynamic, and interactive security testing to create a holistic approach that enhances overall application security measures.

Duration: *10:00*	**Duration:** *20:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Explain the key components of an application tracking system and its importance in capturing essential information. • Analyze the parameters involved in planning a penetration test, including scope, methodologies, and tools. • Describe various testing methods such as black box, white box, and manual testing techniques. • Compare malware sandboxing techniques and their effectiveness in isolating potential threats. • Evaluate the effectiveness of automated scanning tools and manual testing methods in penetration testing. • Explain the role of documenting security requirements across the application lifecycle for clients and stakeholder. • Assess the importance of audit trails and secure data storage in maintaining the integrity of security assessments. • Analyze the benefits of integrating results from static, dynamic, and interactive security testing for improved application security.	• Develop and configure an application tracking system to capture and record essential application details. • Design a penetration testing plan with specified parameters, including target systems, testing scope, and timelines. • Perform hands-on testing of applications using a variety of methods, including automated scanning and manual testing techniques. • Applying: Utilize sandboxing techniques in the classroom to analyze and isolate sample malware in a secure environment. • Create documentation that records security requirements, vulnerabilities, and mitigation steps throughout the testing process. • Record all testing activities and findings to create a comprehensive audit trail. • Implement secure storage practices for storing vulnerability data and testing results. • Integrate static, dynamic, and interactive application security testing results in an automated fashion.
Classroom Aids:	
Whiteboard and markers	

Chart paper and sketch pens LCD Projector and Laptop for presentations
Tools, Equipment and Other Requirements
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated) • Samples of the templates and checklists used in organizations • Static Application Security Testing (SAST) Tools: SonarQube • Dynamic Application Security Testing (DAST) Tools: OWASP ZAP • Software Composition Analysis (SCA) Tools: OWASP Dependency-Check • Vulnerability Management Platforms: OpenVAS • Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR • Programming languages like PHP, Java, Python, or Go etc. • Operating Systems: Linux, Windows.

Module 4: Web Server, Application Security & Active Directory Hardening

Mapped to SSC/N0959 (Version 1)

Terminal Outcomes:

- Identify and categorize key components of web servers, web applications, and administrative interfaces through hands-on exercises.
- Analyze the importance of patching web servers, applications, and databases by evaluating their compliance with STIGs and industry best practices.
- Implement hardening measures on domain controllers to strengthen security and prevent vulnerabilities in a controlled lab setting.

Duration: *10:00*	**Duration:** *20:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Identify key components of web servers, web applications, and administrative interfaces. • Explain the importance of patching web servers, applications, and databases to comply with STIGs and industry best practices. • Analyze the risk of unauthorized systems or functionalities in the network infrastructure. • Evaluate the role of Active Directory in managing user authentication and system security. • Describe how Kerberos operates and explain potential attacks like Kerberos ticket forging. • Justify the necessity of hardening domain controllers to prevent vulnerabilities.	• Locate and secure the administrative interfaces of web servers and web applications in a classroom environment. • Assess web servers, web applications, and databases in sample systems for patch compliance and adherence to STIGs. • Remove unauthorized systems and unnecessary functionalities from a list of applications in a practical exercise. • Demonstrate how Active Directory operates by setting up and configuring sample AD instances. • Detect and address Kerberos ticket forging attacks using classroom lab setups. • Implement hardening measures to strengthen domain controllers in a controlled lab environment.
Classroom Aids:	
Whiteboard and markers Chart paper and sketch pens LCD Projector and Laptop for presentations	
Tools, Equipment and Other Requirements	
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated)	

- Samples of the templates and checklists used in organizations
- Static Application Security Testing (SAST) Tools: SonarQube
- Dynamic Application Security Testing (DAST) Tools: OWASP ZAP
- Software Composition Analysis (SCA) Tools: OWASP Dependency-Check
- Vulnerability Management Platforms: OpenVAS
- Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR
- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 5: Fundamentals of Cybersecurity

Mapped to SSC/N0959 (Version 1)

Terminal Outcomes:

- Analyze web attack logs to identify specific signs of compromise and document findings in a structured report.
- Evaluate the security features of major cloud platforms (AWS, Azure, GCP) and compare their effectiveness in mitigating vulnerabilities.

Duration: *10:00*	**Duration:** *10:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Analyze web attack logs to identify signs of compromise. • Evaluate common indicators of web attacks and their implications for security. • Describe the role of firewalls in protecting applications and databases. • Assess various application and database defense mechanisms. • Compare security features of major cloud platforms (AWS, Azure, GCP). • Evaluate how cloud security features protect internal servers. • Assess cloud infrastructures for potential vulnerabilities. • Justify compliance with security best practices in cloud environments. • Explain how to utilize threat intelligence for threat identification. • Evaluate emerging threats and their implications for cybersecurity. • Analyze current threat data to customize security assessments. • Evaluate the importance of tailoring assessments to evolving threats. • Identify the latest threat indicators and their relevance to security. • Analyze the implications of staying informed about emerging threats. • Evaluate IoT devices for potential security vulnerabilities. • Assess the impact of security flaws like lack of encryption on IoT applications.	• Conduct a practical review of sample logs to identify signs of compromise. • Implement firewall rules in a simulated environment to protect applications. • Perform a comparative analysis of security features across cloud platforms in a lab setting. • Conduct a vulnerability assessment of a mock cloud infrastructure and report findings. • Participate in a drill to identify and respond to simulated emerging threats using threat intelligence. • Develop and present customized security assessments based on recent threat data. • Assess real IoT devices in a lab for security vulnerabilities and propose mitigation strategies. • Establish and test safeguards for secure communications between devices in a classroom exercise. • Evaluate sample AI/ML models for vulnerabilities and biases in a structured lab activity. • Use CSPM tools in a hands-on lab to evaluate and enhance the security posture of cloud environments.

• Describe safeguards for protecting communications between devices. • Assess the effectiveness of various communication security measures. • Analyze the security of AI/ML models for vulnerabilities and biases. • Evaluate potential adversarial attacks on AI/ML systems. • Explain the importance of compliance with cloud security best practices. • Assess how to enhance security posture using CSPM tools.	

Classroom Aids:

Whiteboard and markers
Chart paper and sketch pens
LCD Projector and Laptop for presentations

Tools, Equipment and Other Requirements

Labs equipped with the following:

- PCs/Laptops
- Internet with Wi-Fi (Min. 2 Mbps dedicated)
- Samples of the templates and checklists used in organizations
- Static Application Security Testing (SAST) Tools: SonarQube
- Dynamic Application Security Testing (DAST) Tools: OWASP ZAP
- Software Composition Analysis (SCA) Tools: OWASP Dependency-Check
- Vulnerability Management Platforms: OpenVAS
- Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR
- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 6: API and Application Security Practices

Mapped to SSC/N0959 (Version 1)

Terminal Outcomes:

- Evaluate the effectiveness of fuzz testing in identifying API vulnerabilities through practical application and analysis of results.
- Analyze the potential security weaknesses in APIs by interpreting the findings from fuzz testing activities.
- Assess the importance of ongoing monitoring for security breaches in API management by examining real-world case studies.
- Implement automated security testing within CI/CD pipelines during classroom exercises to enhance software security and evaluate its effectiveness.
- Collaborate with peers to integrate security practices into a mock Software Development Life Cycle (SDLC) project.

Duration: *10:00*	**Duration**: *20:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Explain the principles of fuzz testing and its role in identifying API vulnerabilities. • Analyze the results of fuzz testing to identify potential security weaknesses in APIs. • Assess the importance of ongoing monitoring for security breaches in API management. • Describe the significance of integrating security practices throughout the Software Development Life Cycle (SDLC). • Analyze how security testing can be automated within CI/CD pipelines to enhance software security. • Evaluate the effectiveness of collaboration between development, operations, and security teams in the SDLC.	• Perform fuzz testing on sample APIs to identify vulnerabilities and report findings. • Evaluating: Monitor the API for security breaches and evaluate the effectiveness of the monitoring process. • Collaborate in small groups to integrate security practices into a mock SDLC project. • Automate security testing in a CI/CD pipeline during a classroom exercise and assess its effectiveness. • Review a sample application for vulnerabilities and propose patches or updates based on findings. • Conduct a peer review of vulnerability assessments and patches applied in sample projects. • Establish a security baseline for a classroom environment, including malware protection and updates. • Review the security baseline implementation and discuss improvements based on real-world scenarios.

Classroom Aids:
Whiteboard and markers LCD Projector and Laptop for presentations
Tools, Equipment and Other Requirements
Labs equipped with the following: ● PCs/Laptops ● Internet with Wi-Fi (Min. 2 Mbps dedicated) ● Samples of the templates and checklists used in organizations ● Static Application Security Testing (SAST) Tools: SonarQube ● Dynamic Application Security Testing (DAST) Tools: OWASP ZAP ● Software Composition Analysis (SCA) Tools: OWASP Dependency-Check ● Vulnerability Management Platforms: OpenVAS ● Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR ● Programming languages like PHP, Java, Python, or Go etc. ● Operating Systems: Linux, Windows.

Module 7: Application Security and Vulnerability Management

Mapped to SSC/N0959 (Version 1)

Terminal Outcomes:

- Evaluate the effectiveness of collaboration between security teams and application developers in identifying and addressing security vulnerabilities in applications.
- Analyze common types of application vulnerabilities and their potential impacts on organizational security and operations.
- Assess various mitigation strategies for application security vulnerabilities and their effectiveness in reducing risks.
- Describe the process of timely patching and its significance in maintaining application security.
- Implement hardening techniques on sample applications, documenting changes and testing security improvements against established baseline metrics.

Duration: *10:00*	**Duration:** *10:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Explain the role of collaboration between security teams and application developers in identifying security vulnerabilities. • Analyze the common types of application vulnerabilities and their potential impact on the organization. • Assess the effectiveness of various mitigation strategies for security vulnerabilities in applications. • Describe the importance of timely patching in maintaining application security.	• Explain a sample IT infrastructure process to incorporate patch management requirements. • Present the redesigned process to peers and justify the changes made based on patch management principles. • Implement hardening techniques on sample applications and document the changes made. • Test the hardened applications to evaluate their security improvements compared to baseline metrics. • Use a security tool to generate outcomes and record the results systematically. • Review the documentation created for completeness and clarity, providing feedback for improvement.
Classroom Aids:	
Whiteboard and markers LCD Projector and Laptop for presentations	
Tools, Equipment and Other Requirements	

Labs equipped with the following:

- PCs/Laptops
- Internet with Wi-Fi (Min. 2 Mbps dedicated)
- Samples of the templates and checklists used in organizations
- Static Application Security Testing (SAST) Tools: SonarQube
- Dynamic Application Security Testing (DAST) Tools: OWASP ZAP
- Software Composition Analysis (SCA) Tools: OWASP Dependency-Check
- Vulnerability Management Platforms: OpenVAS
- Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR
- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 8: IT Infrastructure Security Management

Mapped to SSC/N0959 (Version 1)

Terminal Outcomes:

- Evaluate the effectiveness of existing IT infrastructure processes for patch management by identifying specific areas for redesign.
- Justify the alignment of redesigned IT infrastructure processes with patch management requirements.
- Apply application hardening techniques on sample applications and document the changes to improve overall security posture.
- Analyze the documentation quality for security tool outcomes, providing feedback for clarity and usability in reporting security improvements.

Duration: *10:00*	**Duration**: *10:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Explain the principles of patch management and its importance in IT infrastructure. • Analyze existing IT infrastructure processes to identify areas needing redesign for effective patch management. • Justify the alignment of redesigned processes with patch management requirements. • List industry best practices for hardening applications to improve security. • Describe the role of application hardening in mitigating security risks. • Assess the effectiveness of different application hardening techniques based on case studies. • Identify key metrics and outcomes generated by security tools and solutions. • Analyze the documentation processes to ensure comprehensive records of tool outcomes. • Critique the documentation quality for usability and clarity in reporting security outcomes.	• Redesign a sample IT infrastructure process to incorporate patch management requirements. • Present the redesigned process to peers and justify the changes made based on patch management principles. • Implement hardening techniques on sample applications and document the changes made. • Test the hardened applications to evaluate their security improvements compared to baseline metrics. • Use a security tool to generate outcomes and record the results systematically. • Review the documentation created for completeness and clarity, providing feedback for improvement.
Classroom Aids:	

Whiteboard and markers LCD Projector and Laptop for presentations
Tools, Equipment and Other Requirements
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated) • Samples of the templates and checklists used in organizations • Static Application Security Testing (SAST) Tools: SonarQube • Dynamic Application Security Testing (DAST) Tools: OWASP ZAP • Software Composition Analysis (SCA) Tools: OWASP Dependency-Check • Vulnerability Management Platforms: OpenVAS • Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR • Programming languages like PHP, Java, Python, or Go etc. • Operating Systems: Linux, Windows.

Module 9: PowerShell for System Administration and Log Management

Mapped to SSC/N0960 (Version 1)

Terminal Outcomes:

- Analyze the roles of authorized personnel in confirming the monitoring scope to ensure comprehensive security oversight.
- Develop and execute a security script using PowerShell to automate a routine administrative task, demonstrating effective scripting skills.
- Evaluate the effectiveness of various tools for log capture and management, identifying strengths and weaknesses in real-world scenarios.
- Demonstrate the process of documenting log management procedures, applying established standards to format captured logs accurately.

Duration: *10:00*	**Duration:** *10:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Analyze the roles of authorized personnel in confirming monitoring scope. • Evaluate the implications of improperly defined monitoring scopes. • Recall basic PowerShell syntax and commands used for system administration. • Explain the purpose of PowerShell in automating administrative tasks. • Analyze the impact of effective command execution on system performance. • Describe the process of developing security scripts using PowerShell. • Analyze how automation improves efficiency in system administration. • Evaluate the effectiveness of different security scripts in task automation. • Define key operational processes involved in effective log management. • Explain the significance of log management in maintaining system security. • Analyze the consequences of poor log management practices. • Identify key events and activity logs necessary for monitoring systems.	• Apply techniques to confirm the scope of application assets with authorized personnel. • Conduct discussions to clarify monitoring requirements with relevant stakeholders. • Execute basic PowerShell commands in a classroom environment to automate simple tasks. • Demonstrate the use of PowerShell syntax through hands-on exercises. • Develop and execute a security script using PowerShell for a specific routine task. • Perform troubleshooting of security scripts in real-time during training sessions. • Create a step-by-step log management process for classroom demonstration. • Conduct a practical exercise on documenting log management procedures. • Use appropriate tools to identify and capture key events and activity logs during a hands-on session. • Format captured logs according to established standards and present findings to the class.

• Explain the standards and formats used for capturing logs. • Analyze the effectiveness of different tools for log capture and management.	

Classroom Aids:

Whiteboard and markers
LCD Projector and Laptop for presentations

Tools, Equipment and Other Requirements

Labs equipped with the following:

- PCs/Laptops
- Internet with Wi-Fi (Min. 2 Mbps dedicated)
- Samples of the templates and checklists used in organizations
- Static Application Security Testing (SAST) Tools: SonarQube
- Dynamic Application Security Testing (DAST) Tools: OWASP ZAP
- Software Composition Analysis (SCA) Tools: OWASP Dependency-Check
- Vulnerability Management Platforms: OpenVAS
- Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR
- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 10: Application Security Assessment and Threat Management

Mapped to SSC/N0960 (Version 1)

Terminal Outcomes:

- Assess security vulnerabilities in applications by identifying and evaluating potential risks and threat vectors, such as hacking attempts and malware.
- Explain the importance of implementing security controls in software development and how they align with industry security standards.
- Conduct comprehensive security assessments on sample applications to identify vulnerabilities and potential risks effectively.
- Create secure sandbox environments to isolate and test various types of threats encountered in cybersecurity scenarios.

Duration: *15:00*	**Duration:** *30:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Assess security vulnerabilities in applications by identifying potential risks and understanding threat vectors such as hacking attempts and malware. • Explain the importance of implementing security controls in software development and how they align with industry security standards. • Describe techniques for real-time monitoring and threat analysis to effectively respond to security breaches.	• Conduct comprehensive security assessments on sample applications to identify vulnerabilities and potential risks. • Implement security controls within a software development environment and evaluate adherence to industry standards. • Perform real-time monitoring and threat analysis on simulated data to detect and respond to security incidents. • Create secure sandbox environments to isolate and test different types of threats encountered in cybersecurity scenarios.
Classroom Aids:	
Whiteboard and markers LCD Projector and Laptop for presentations	
Tools, Equipment and Other Requirements	
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated) • Samples of the templates and checklists used in organizations • Static Application Security Testing (SAST) Tools: SonarQube	

- Dynamic Application Security Testing (DAST) Tools: OWASP ZAP
- Software Composition Analysis (SCA) Tools: OWASP Dependency-Check
- Vulnerability Management Platforms: OpenVAS
- Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR
- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 11: Risk Assessment and Incident Management

Mapped to SSC/N0960 (Version 1)

Terminal Outcomes:

- Analyze the likelihood and potential consequences of recognized risks to classify their urgency in real-case scenarios.
- Demonstrate the necessary steps to assess and address recognized risks within organizational frameworks through hands-on exercises.
- Evaluate incidents by logging them into a ticketing system during a mock analysis of suspicious findings.
- Classify service requests according to organizational policies and guidelines through practical group activities and case studies.

Duration: *10:00*	**Duration:** *10:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Analyze the likelihood and potential consequences of recognized risks to classify their urgency. • Explain the necessary steps to assess and address recognized risks within organizational frameworks. • Discuss the importance of logging incidents in ticketing systems for suspicious findings. • Describe the processes for classifying service requests according to organizational policies. • Evaluate how to allocate tickets to appropriate individuals based on risk type and organizational procedures. • Assess the guidelines for arranging service requests in accordance with organizational standards.	• Classify risks based on urgency by assessing likelihood and potential impact using real-case scenarios. • Identify and outline the steps taken to address recognized risks through hands-on exercises. • Log incidents into a ticketing system during a mock analysis of suspicious findings. • Practice classifying service requests using case studies in alignment with organizational processes. • Allocate tickets to individuals based on provided risk scenarios, adhering to organizational procedures. • Arrange service requests according to organizational guidelines in a practical group activity.
Classroom Aids:	
Whiteboard and markers LCD Projector and Laptop for presentations	
Tools, Equipment and Other Requirements	
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated)	

- Samples of the templates and checklists used in organizations
- Static Application Security Testing (SAST) Tools: SonarQube
- Dynamic Application Security Testing (DAST) Tools: OWASP ZAP
- Software Composition Analysis (SCA) Tools: OWASP Dependency-Check
- Vulnerability Management Platforms: OpenVAS
- Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR
- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 12: Ticket Management and Resolution

Mapped to SSC/N0960 (Version 1)

Terminal Outcomes:

- Coordinate effectively with team members to ensure timely resolution of tickets, demonstrating communication and collaboration skills in a classroom setting.
- Document the outcomes of monitoring and resolving tickets using standardized forms, ensuring accuracy and adherence to documentation protocols.

Duration: *05:00*	**Duration:** *10:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Explain the importance of coordinating with personnel to address tickets within specified timelines. • Describe the process of seeking assistance from specialists for issues outside one's expertise. • Identify the standardized documentation protocols for monitoring and resolving tickets.	• Coordinate with team members in a classroom setting to ensure timely actions on raised tickets. • Role-play scenarios to seek guidance from specialists for complex issues during ticket resolution. • Complete standardized forms to document outcomes of monitoring, ticket creation, and resolution activities.
Classroom Aids:	
Whiteboard and markers LCD Projector and Laptop for presentations	
Tools, Equipment and Other Requirements	
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated) • Samples of the templates and checklists used in organizations • Static Application Security Testing (SAST) Tools: SonarQube • Dynamic Application Security Testing (DAST) Tools: OWASP ZAP • Software Composition Analysis (SCA) Tools: OWASP Dependency-Check • Vulnerability Management Platforms: OpenVAS • Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR • Programming languages like PHP, Java, Python, or Go etc. • Operating Systems: Linux, Windows.	

Module 13: Security Compliance and Monitoring

Mapped to SSC/N0960 (Version 1)

Terminal Outcomes:

- Analyze and evaluate the adherence to laws, regulations, policies, and guidelines in cybersecurity through the examination of relevant case studies.
- Utilize external data sources to track security issues and compile a comprehensive report on potential risks affecting organizational security.

Duration: *10:00*	**Duration:** *10:00*
Theory – Key Learning Outcomes	**Practical – Key Learning Outcomes**
• Explain the importance of adhering to laws, regulations, policies, and guidelines in cybersecurity. • Identify key external data sources for tracking security issues and their relevance to organizational security. • Describe the role of telemetry monitoring in detecting security platform issues.	• Demonstrate adherence to applicable laws and policies by reviewing case studies in a classroom setting. • Utilize external data sources to track security issues and compile a report on potential risks. • Conduct telemetry monitoring exercises to identify and analyze issues within a simulated security platform.
Classroom Aids:	
Whiteboard and markers LCD Projector and Laptop for presentations	
Tools, Equipment and Other Requirements	
Labs equipped with the following: • PCs/Laptops • Internet with Wi-Fi (Min. 2 Mbps dedicated) • Samples of the templates and checklists used in organizations • Static Application Security Testing (SAST) Tools: SonarQube • Dynamic Application Security Testing (DAST) Tools: OWASP ZAP • Software Composition Analysis (SCA) Tools: OWASP Dependency-Check • Vulnerability Management Platforms: OpenVAS • Security Orchestration, Automation, and Response (SOAR) Platforms: Cortex XSOAR	

- Programming languages like PHP, Java, Python, or Go etc.
- Operating Systems: Linux, Windows.

Module 14: Introduction to Employability Skills

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Discuss the Employability Skills required for jobs in various industries
- List different learning and employability related GOI and private portals and their usage

Duration:1.5 Hours (0.5 Theory + 1 Practical)

Module 15: Constitutional values - Citizenship

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Explain the constitutional values, including civic rights and duties, citizenship, responsibility towards society and personal values and ethics such as honesty, integrity, caring and respecting others that are required to become a responsible citizen
- Show how to practice different environmentally sustainable practices

Duration:1.5 Hours (0.5 Theory + 1 Practical)

Module 16: Becoming a Professional in the 21st Century

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Discuss importance of relevant 21st century skills.
- Exhibit 21st century skills like Self-Awareness, Behaviour Skills, time management, critical and adaptive thinking, problem-solving, creative thinking, social and cultural awareness, emotional awareness, learning to learn etc. in personal or professional life.
- Describe the benefits of continuous learning

Duration:2.5 Hours (1 Theory + 1.5 Practical)

Module 17: Basic English Skills

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Show how to use basic English sentences for everyday conversation in different contexts, in person and over the telephone
- Read and interpret text written in basic English
- Write a short note/paragraph / letter/e -mail using basic English

Duration: 10 Hours (4 Theory + 6 Practical)

Module 18: Career Development and Goal Setting

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Create a career development plan with well-defined short- and long-term goals

Duration: 2 Hours (1 Theory + 1 Practical)

Module 19: Communication skills

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Demonstrate how to communicate effectively using verbal and nonverbal communication etiquette.
- Explain the importance of active listening for effective communication
- Discuss the significance of working collaboratively with others in a team

Duration: 5 Hours (2 Theory + 3 Practical)

Module 20: Diversity and Inclusion

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Demonstrate how to behave, communicate, and conduct oneself appropriately with all genders and PwD
- Discuss the significance of escalating sexual harassment issues as per POSH

Duration: 2.5 Hours (1 Theory+ 1.5 Practical)

Module 21: Financial and Digital Literacy

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Outline the importance of selecting the right financial institution, product, and service
- Demonstrate how to carry out offline and online financial transactions, safely and securely

Duration: 5 Hours (2 Theory+ 3 Practical)

Module 22: Essential Digital Skills

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Describe the role of digital technology in today's life
- Demonstrate how to operate digital devices and use the associated applications and features, safely and securely
- Discuss the significance of displaying responsible online behaviour while browsing, using various social media platforms, e-mails, etc., safely and securely
- Create sample word documents, excel sheets and presentations using basic features
- utilize virtual collaboration tools to work effectively

Duration: 10 Hours (4 Theory+ 6 Practical)

Module 23: Entrepreneurship

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Explain the types of entrepreneurship and enterprises
- Discuss how to identify opportunities for potential business, sources of funding and associated financial and legal risks with its mitigation plan
- Describe the 4Ps of Marketing-Product, Price, Place and Promotion and apply them as per requirement
- Create a sample business plan, for the selected business opportunity

Duration: 7 Hours (3 Theory+ 4 Practical)

Module 24: Customer Service

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Describe the significance of analysing different types and needs of customers
- Explain the significance of identifying customer needs and responding to them in a professional manner.
- Discuss the significance of maintaining hygiene and dressing appropriately

Duration: 5 Hours (2 Theory+ 3 Practical)

Module 25: Getting Ready for Apprenticeship and Jobs

Mapped to NOS DGT/VSQ/N0102 (Version No. 1)

Key Learning Outcomes:

- Create a professional Curriculum Vitae (CV)
- Use various offline and online job search sources such as employment exchanges, recruitment agencies, and job portals respectively
- Discuss the significance of maintaining hygiene and confidence during an interview
- Perform a mock interview
- List the steps for searching and registering for apprenticeship opportunities

Duration: 8 Hours (3 Theory+ 5 Practical)

Annexure

Trainer Requirements

1.	**Trainer's Qualification and experience in the relevant sector (in years)** *(as per NCVET guidelines)*	**Educational Qualification:** Graduate in any discipline, preferably Engineering/Science/Computer Science/Electronics and Engineering /Information Technology. **Industry & Training Experience:** 2 years of industry experience in the field of cyber security. **Certification: "Trainer"** mapped to the Qualification Pack **"MEP/Q2601"** Minimum accepted score is 80% aggregate.
2.	**Master Trainer's Qualification and experience in the relevant sector (in years)** *(as per NCVET guidelines)*	**Educational Qualification:** Graduate in any discipline, preferably Engineering/Science/Computer Science/Electronics and Engineering /Information Technology. **Industry & Training Experience:** 4 years of industry experience in field of cyber security. **Certification: "Master Trainer"** mapped to the Qualification Pack **"MEP/Q2602"** Minimum accepted score is 90% aggregate
3.	**Tools and Equipment Required for the Training**	☒Yes ☐No *(If "Yes", details to be provided in Annexure)*
4.	**In Case of Revised Qualification, details of Any Upskilling Required for Trainer**	NA

Assessor Requirements

1.	**Assessor's Qualification and experience in relevant sector (in years)** *(as per NCVET guidelines)*	**Educational Qualification:** Graduate in any discipline, preferably Engineering/Science/Computer Science/Electronics and Engineering /Information Technology. **Industry & Training Experience:** 2 years of industry experience in the field of cyber security. **Certification: "Assessor"** mapped to the Qualification Pack **"MEP/Q2701"** Minimum accepted score is 80% aggregate.

2.	**Proctor's Qualification and experience in relevant sector (in years)** *(as per NCVET guidelines), (wherever applicable)*	**Educational Qualification:** Graduate in any discipline, preferably Engineering/Science/Computer Science/Electronics and Engineering /Information Technology. **Industry & Training Experience:** 2 years of industry experience in the field of cyber security. **Certification: "Proctor"** mapped to the Qualification Pack **"MEP/Q2701"** Minimum accepted score is 80% aggregate.
3.	**Lead Assessor's/Proctor's Qualification and experience in relevant sector (in years)** *(as per NCVET guidelines)*	**Educational Qualification:** Graduate in any discipline, preferably Engineering/Science/Computer Science/Electronics and Engineering /Information Technology. **Industry & Training Experience:** 4 of industry experience in the field of cyber security. **Certification: "Lead Assessor"** mapped to the Qualification Pack **"MEP/Q2702"** Minimum accepted score is 90% aggregate.
4.	**Assessment Mode** *(Specify the assessment mode)*	Online or Offline
5.	**Tools and Equipment Required for Assessment**	☒ Same as for training ☐ Yes ☐ No *(details to be provided in Annexure-if it is different for Assessment)*

Assessment Strategy

Assessment Process Overview

Batch Creation & Assessment Request:

Training Providers (TP) or Training Centers (TC), including any other authorized partner of Ministry/ Department create batches / push batches on the SIDH portal. Assessment requests are submitted through the SIDH portal or via email or other media as authorized from time to time. For NON-SIDH schemes, assessment requests are received electronically or through respective State Skill Mission portals. TP/TC initiates the assessment request through the InSDMS portal and processes the payment (where applicable).

Batch Alignment & Confirmation:

Upon payment confirmation, batches are assigned to the Assessment Agency based on factors like:

- Assessment readiness
- Availability of certified assessors for the specific job role

- Assessment capping to an assessment agency as prescribed from time to time for an AB An email communication / prescribed mode communication is sent to TP/TC for confirmation of the assessment date, with IT-ITeS SSC in the loop. Once confirmation is received, the Assessment Agency designates a TOA-certified assessor to conduct or facilitate the assessment.
- Batches are only formed when the Qualification is active.

Candidate Verification & Assessment Execution:

Candidate details are verified and documented at the beginning of the assessment by a certified assessor. A Quality Assurance (QA) mechanism is enforced, requiring an undertaking from the TC. Regular feedback is collected from TP/TC to ensure continuous improvement.

Evidence Collection & Validation:

Proctors or assessors capture date/time-stamped and geo-tagged photographs of the assessment location during the process. Attendance is also ensured offline. A PC-wise result analysis is conducted to refine assessment standards.

Monitoring & Compliance:

Batch monitoring follows established protocols, ensuring adherence to assessment guidelines. Sample based surprise visits are conducted at TC locations during both training and assessments to verify compliance. This structured approach ensures transparency, quality control, and validation throughout the assessment process.

Testing Environment:

- Check the Assessment location, date and time
- If the batch size is more than 30, then there should be 2 Assessors.
- Check that the allotted time to the candidates to complete Theory & Practical Assessment is correct.

Assessment Quality Assurance levels/Framework:

IT-ITeS SSC nasscom is responsible for the development and periodic review of the question bank developed for a specific job role. We publish an openly accessible sample /model question paper on our website for all stakeholders. The quality of the Question Bank created by the assessment designer is validated by a Subject matter experts on the following parameters:

- Appropriateness of the Question Bank in terms of facts, data and information.
- Checks for grammar, spellings, scripting and formatting.
- The information provided should be specific enough to remove any ambiguity in answers/solutions to the question.
- Relevance – Assessing the topic well w.r.t. the job role.
- Check if the difficulty level of each question is as per the matrix.
- Check if the images used in the question are clear and relevant.
- All variables, symbols and abbreviations used must be declared.
- The correct answer option should be unique, and the options should not be overlapping

Recommended Supplemental Readings

The learning modules covered in the Model Curriculum for Analyst- Application Security are designed to meet the expected outcomes as per the QP. While the modules aligned to NOS are focused on technical/ behavioral competencies, bridge modules cover the prerequisite/ preparatory topics that are indispensable to complete the course. However, to provide additional QP specific knowledge to the learners, the following supplemental readings on related topics are recommended. These readings will equip the learners with an understanding of advanced or ancillary concepts to take up more complex tasks as listed in the QP.

QP	Recommended Supplemental Reading
SSC/Q0903: Analyst Application Security	1. Cryptography 2. Data Privacy Safeguards 3. Cloud Workload Protection 4. SaaS and API Security

References

Glossary

Term	Description
Key Learning Outcome	Key learning outcome is the statement of what a learner needs to know, understand and be able to do in order to achieve the terminal outcomes. A set of key learning outcomes will make up the training outcomes. Training outcome is specified in terms of knowledge, understanding (theory) and skills (practical application).
Training Outcome	Training outcome is a statement of what a learner will know, understand and be able to do **upon the completion of the training**.
Terminal Outcome	Terminal outcome is a statement of what a learner will know, understand and be able to do **upon the completion of a module.** A set of terminal outcomes help to achieve the training outcome.
National Occupational Standard	National Occupational Standard specify the standard of performance an individual must achieve when carrying out a function in the workplace
Performance Criteria	Performance Criteria indicates what specific characteristics an individual should be able to demonstrate in order to achieve the learning outcomes
Persons with Disability	Persons with Disability are those who have long-term physical, mental, intellectual or sensory impairments which in interaction with various barriers may hinder their full and effective participation in society on an equal basis with others.

Acronyms and Abbreviations

Term	Description
QP	Qualification Pack
NSQF	National Skills Qualification Framework
NSQC	National Skills Qualification Committee
NOS	National Occupational Standards
SSC	Skill Sectors Councils
NASSCOM	National Association of Software and Service Companies
PwD	Persons with Disability
NCO	National Classification of Occupations
ISCO	International Standard Classification of Occupations
ISIC	International Standard Industrial Classification
ISO	International Organization for Standardization
SLA	Service Level Agreement
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
OSI	Open Systems Interconnection
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TCP	Transmission Control Protocol
FTP	File Transfer Protocol
SSH	Secure Shell
SFTP	SSH File Transfer Protocol
SNMP	Simple Network Management Protocol
VPN	Virtual Private Network
RDP	Remote Desktop Protocol
HTTPS	Hypertext Transfer Protocol Secure
2FA	Two-Factor Authentication
RDBMS	Relational Database Management System
SDLC	Software Development Lifecycle
SIEM	Security Information and Event Management
OWASP	Open Web Application Security Project
OSSIM	Open-source Security Information and Event Management System
CRM	Customer Relationship Management
PC	Performance Criteria