BYLIGHT

Course and Lab Catalog

(Partial Listing, 01AUG23)

CompTIA Certifications

- □ **Cyber Aptitude Typology Indicator (CATI) (30 minutes)**
  CATI is designed to support cyber-interested individuals with actionable metrics that can be used to better match learning preferences, personality characteristics, and skillsets with cybersecurity work roles as defined by the National Initiative for Cybersecurity Education’s Cyber Workforce Framework (NICE CWF).

- □ **A+ Core 1 and Core 2 (80 hours)**
  The A+ Certification is an industry standard for computer technicians and is often the starting place for cybersecurity professionals. The lessons which compose this course are based on By Light's competency framework requiring users to perform tasks that demonstrate knowledge and skill for the A+ (core 1) certification exam. The A+ Certification is important because it is the starting point for multiple work roles such as Service Desk Analyst, Help Desk Technician, Technical Support Specialist, Field Service Technician, Associate Network Engineer, Data Support Technician, Desktop Support Administrator or End User Computing Technician.

- □ **Cyber Advanced Security Practioner (CASP) + (40 hours)**
  CompTIA CASP+ is an advanced certification that validates the skills necessary to be security architects and senior security engineers. It is the only hands-on, performance-based certification for advanced practitioners that covers both security architecture and engineering. This course can benefit you in two ways. If you intend to pass the CompTIA CASP+ (Exam CAS-004) certification examination, this course can be a significant part of your preparation, and today's job market demands individuals with demonstrable skills, the information and activities in this course can help you build your information security skill set so that you can confidently perform your duties as an advanced security practitioner.

- □ **Cloud + (40 hours)**
  CompTIA Cloud+ is a global certification (CV0-003) that validates the skills needed to deploy and automate secure cloud environments that support the high availability of business systems and data. Migrating to the cloud presents opportunities to deploy, optimize, and protect mission critical applications and data storage. CompTIA Cloud+ validates the technical skills needed to secure these valuable assets. The reality of operating multi-cloud environments poses new challenges. CompTIA Cloud+ is ideal for cloud engineers who need to have expertise across multiple products and systems. CompTIA Cloud+ is a focused certification approved for DoD 8570.01-M, offering an infrastructure option for individuals who need to certify in IAM Level I, CSSP Analyst and CSSP Infrastructure Support role.

- □ **Cyber Security Analyst (CySA) + (40+ hours)**
  The CySA+ Certification is an industry standard for cybersecurity professionals. The lessons which compose this course are based on By Light's competency framework requiring users to perform tasks that demonstrate knowledge and skill for the CySA+ certification exam. The CySA+ Certification is popular because its competency framework supports multiple work roles such as the Cyber Defense Analyst or Incident Responder and provides a cybersecurity foundation for government and commercial practitioners.

□ **Data +** **(40 hours)**

This course is designed to provide the knowledge and skills required to transform business requirements in support of data-driven decisions through mining and manipulating data, applying basic statistical methods, and analyzing complex datasets while adhering to governance and quality standards throughout the entire data life cycle. The CompTIA Data+ DA0-001 exam was released in February 2022.

□ **IT Fundamentals (ITF) +** **(40 hours)**

CompTIA's IT Fundamentals (ITF+) Certification focuses on the essential IT skills and knowledge needed to perform tasks commonly performed by advanced end-users and entry-level IT professionals alike. This course helps professionals to decide if a career in IT is right for them or to develop a broader understanding of IT. ITF+ is the only pre-career certification that helps students or career changers determine if they have a competency for information technology and if it is the right career path for them. CompTIA ITF+ also helps organizations prepare non-technical teams for digital transformation.

□ **IT Fundamentals (ITF) +** **(40 hours)**

CompTIA's IT Fundamentals (ITF+) Certification focuses on the essential IT skills and knowledge needed to perform tasks commonly performed by advanced end-users and entry-level IT professionals alike. This course helps professionals to decide if a career in IT is right for them or to develop a broader understanding of IT. ITF+ is the only pre-career certification that helps students or career changers determine if they have a competency for information technology and if it is the right career path for them. CompTIA ITF+ also helps organizations prepare non-technical teams for digital transformation.

□ **LINUX +** **(40 hours)**

CompTIA Linux+ (XK0-005) is an intermediate-level IT certification that provides IT professionals with knowledge of Linux. CompTIA Linux+ validates the competencies required of an early career supporting Linux systems. A strong knowledge of Linux is key for cybersecurity professionals involved in architecting a robust enterprise or individuals hardening systems to mitigate attacks.

□ **Network +** **(40 hours)**

This course is designed to provide foundational skills in Networking that can lead to a CompTIA Net+ (Exam N10-008) certification through the use of interactive instruction and immersion into a cyber range. Network+ validates the technical skills needed to securely establish, maintain and troubleshoot the essential networks that businesses rely on.

□ **Penetration Testing (PenTest) +** **(40 hours)**

The PenTest+ Certification is an industry standard for cybersecurity professionals. The lessons which compose this course are based on By Light's competency framework requiring users to perform tasks that demonstrate knowledge and skills for the PenTest+ certification exam. This course provides the principles of penetration testing and exposure to many of the tools used. A key aspect of the PenTest+ certification is the need to understand how to scope tests and analyze data which requires many hours of practice in the cyber range in order to retain the knowledge and apply it to basic cybersecurity scenarios.

□ **Project+** **(40 hours)**

Project+ is the Computing Technology Industry Association's (CompTIA) certification program planned to demonstrate validated learning and skills in the field of project management. The certification helps show employers that a potential employee is skilled, knowledgeable and capable of managing projects for their organization. CompTIA Project+ is best for professionals who need to manage small projects as part of their other job responsibilities. It's an excellent entry-level project management credential. Project+ covers essential project management concepts beyond the scope of just one methodology and framework. It confirms the capability to initiate, manage and maintain a project or business advantage. It is not just for IT professionals instead it is taken up by any individual who wishes to showcase their talent in the different companies including telecom businesses. The CompTIA Project+ exam covers the project management lifecycle. It also tests for the skills required to initiate, plan, execute, manage and review a project.

- □ **Security +** **(50 hours)**
  This course is a unique offering of the CompTIA Certification preparation using Infosec Learning labs to provide a deeper sustainable knowledge and skills base for a career in cyber. The lessons in this course are based on By Light's SEC+ competency framework requiring users to perform tasks that demonstrate knowledge and skills for the SEC+ Certification exam. The SEC+ Certification is one of the most popular because its competency framework supports multiple work roles and provides a cybersecurity foundation for government and commercial practitioners. This course provides the principles of cybersecurity, networking, threats and vulnerabilities. A key aspect of the SEC+ Certification is the need to understand how to implement various security concepts and technologies which requires many hours of practice in the cyber range to retain the knowledge and apply it to basic cybersecurity scenarios.

- □ **Server +** **(40 hours)**
  The CompTIA Server+ certification training course focuses on multi-vendor products and implementations, based on industry standards. The course covers system hardware, software, storage, best practices in an IT environment, disaster recovery and troubleshooting. CompTIA Server+ is a global certification that validates the hands-on skills of IT professionals who install, manage and troubleshoot servers in data centers as well as on-premise and hybrid environments. The exam covers essential hardware and software technologies of on-premise and hybrid server environments including high availability, cloud computing and scripting. The new exam includes performance-based questions that require the candidate to demonstrate multi-step knowledge to securely deploy, administer and troubleshoot servers.

DevSecOps

- □ **COD 107 – Secure Software Deployment** **(10 mins)**
  This course describes general principles to think about for improving your deployment process, best practices for logging and monitoring, as well as different ways to defend the operating system, web server and the database. After complicating this course you will understand attack surface reduction, compartmentalization, defense in depth, least privilege deployment, secure defaults and security incident response plans.

- □ **COD 108 – Software Operations and Maintenance** **(10 mins)**
  This course covers best practices for logging and monitoring, as well as security misconfiguration and mitigation techniques. After completing this course you will understand application security

patching processes, strategies for defense-in-depth, and mechanisms to ensure sufficient logging and monitoring.

□ CYB 310 – Using Cyber Supply Chain Risk Management (C-SCRM) to Mitigate Threats to IT/OT (NEW) **(40 mins)**

Using Cyber Supply Chain Risk Management (C-SCRM) to mitigate the risks associated with the distributed and interconnected nature of IT/OT product and service supply chains requires close coordination and information-sharing with reliable allies and constant monitoring for and evaluation of security risks and openings. Learners will gain a basic understanding of C-SCRM, including its central ideas, recommended procedures, and established norms. This course introduces how to create and execute effective C-SCRM strategies to safeguard their organizations' IT and OT systems against cyber risks originating in the supply chain via a mix of theoretical understanding and real-world experiences.

On successful completion of this course, learners should have the knowledge and skills required to:

- Detect supply chain threats and vulnerabilities
- Evaluation risk as part of supplier selection
- Examine third-party security, practices, and protocols
- Leverage supply chain security standards and frameworks
- Develop incident response and recovery
- Use C-SCRM to manage contracts
- Mitigate insider threats and monitor systems

□ CYB 311 – Threat Analysis with AI (NEW) **(20 mins)**

AI analyzes relationships between threats like malicious files, suspicious IP addresses or insiders in seconds or minutes. AI provides curated risk analysis, reducing the time security analysts take to make critical decisions and remediate threats. In this course, we discuss how AI is helping organizations protect themselves against cyber-attacks. This includes the fundamental components of AI, such as sandboxes and trained data, as well as the logic used in machine learning, neural networks, and deep learning.

On successful completion of this course, learners should have the knowledge and skills required to:

- Perform Threat Analysis with AI
- Understand AI logic and specific use cases of AI in the threat detection landscape
- Use AI for application development, malware analysis, and user behavioral analytics

□ DSO 201 – Fundamentals of Secure DevOps **(30 mins)**

Building a culture of collaboration between software development (Dev) and information-technology operations (Ops) is full of challenges and learning. The notion of DevOps requires a good understanding of complex technical problems and business needs at the same time. This course introduces learners to the philosophy and provides the fundamental knowledge needed to execute practices that shorten system development lifecycles and provide continuous delivery with high software quality.

After completing this course you will be able to:

- Understand the unique opportunities for including security in your DevOps pipeline
- Understand at which points in your development process—from architecture to deployment—you can improve security
- Automate security compliance and controls using a variety of open-source tools
- Identify opportunities for increased collaboration and better feedback loops

□ DSO 205 – Securing the COTS Supply Chain **(15 mins)**

The usage of Commercial-off-the-shelf software (COTS) by organizations while advantageous comes with its own set of challenges and complexities. Unfortunately, it is rare for acquisition approaches to account for complex software supply chains; this course provides learners with an understanding of how to apply DevSecOps best practices to reduce software supply chain risks. After completing this course you will be to:

- Employ acquisition strategies, contract tools, and procurement methods for the purchase of the software, COTS from suppliers
- Conduct a supplier review prior to entering into a contractual agreement to acquire the COTS
- Conduct an assessment of the COTS prior to selection, acceptance, or update
- Employ security safeguards to validate that the COTS received is genuine and has not been altered
- Establish and retains the unique identification of supply chain elements, processes, and actors for the COTS
- Establish a process to address weaknesses or deficiencies in supply chain elements identified during independent or organizational assessments of such elements

□ DSO 206 – Securing the Open Source Supply Chain **(15 mins)**

As modern software development evolves, organizations are finding themselves leveraging Open Source Software to reduce costs, simplify operations, accelerate innovation, and improve interoperability. Adoption is expected to continue, but distribution and licenses allow anyone to use, view, modify, and share source code, which introduces new security vulnerability risks into the supply chain. This course provides learners with an understanding of how to apply DevSecOps best practices to reduce software supply chain risks inherent with the use of open-source software.

After completing this course you will be to meet compliance requirements while developing a DevSecOps mindset, including:

- Expanding the development teams use of dependency checking tools, including integration into the build process
- Ensuring application security teams implement security tests to audit select open-source software and components
- Establishing a build process that ensures the installation of the latest open software releases for operations teams migrating server platforms to containers
- Understanding the importance of monitoring repo feeds and CVEs, especially critical libraries such as OpenSSL that could affect many different products

□ DSO 211 – Identifying Threats to Containers in a DevSecOps Framework **(20 mins)**

Widespread adoption of cloud computing and DevOps have led to containers becoming the most popular and efficient way to deploy applications. However, containerization presents enterprise security risks that question existing security policies and compliance frameworks. This course provides a necessary understanding of known attacks required to improve the security of container application deployments.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- The importance of Identifying threats to containers and data in the DevSecOps framework
- Why containers are particularly susceptible to image vulnerabilities, and how to mitigate the threat by rebuilding images as part of security updates.
- How to validate external images to prevent malware, unintended functionality, functional bugs, or components with known vulnerabilities into your environment

- Securely encrypting communication channels to avoid man-in-the-middle attacks designed to extract image contents, compromise credentials used to access registries or tamper with images being sent to orchestrators

□ DSO 212 – Fundamentals of Zero Trust Security **(15 mins)**

Zero-trust is a security concept that defines various practices and technologies that, when brought together, provide a multilayer security approach. Coverage in this course aligns with CISA Zero Trust Maturity Model. It will help you understand what zero-trust security is, why it is necessary, and which points an organization would consider when implementing a zero-trust architecture.
After you complete this course, you will have the knowledge needed to:

- Understand secure network strategy
- Identify strong versus weak security policy
- Identify common security-driven tools and methodologies.

□ DSO 253 – DevSecOps in the AWS Cloud **(20 mins)**

Using a cloud Platform solves issues with distributed complexity and provides DevOps automation with a standard and centralized platform for testing, deployment, and production creating a complementary relationship between the two. This course provides learners with an understanding of how to align and configure AWS services to NIST Cybersecurity Framework (CSF) core functions to achieve security in the cloud.
After completing this course you will be able to:

- Implement inventory and configuration controls and services, including AWS Config, AWS CloudFormation, and Amazon Inspector
- Ensure Infrastructure Security using Amazon VPC, AWS WAF, Customer-controlled encryption and automatic encryption of all traffic
- Mitigate DDoS threats with Autoscaling, Amazon CloudFront and Amazon Route 53
- Encrypt data using AWS Key Management Services (KMS), Server-side encryption (SSE), AWS CloudHSM; and leverage EBS, S3, Glacier, Oracle RDS, SQL Server RDS, and Redshift encryption features
- Meet Monitoring and Logging requirements using AWS CloudTrail and Amazon CloudWatch
- Use Identity and Access Controls to define, enforce, and manage user access policies with AWS Identity and Access Management (IAM), AWS Multi-Factor Authentication and AWS Directory Services
- Understand AWS policies for customer Penetration Testing

□ DSO 254 – DevSecOps in the Azure Cloud **(20 mins)**

Using a cloud Platform solves issues with distributed complexity and provides DevOps automation with a standard and centralized platform for testing, deployment, and production creating a complementary relationship between the two. Provides learners with an understanding of how to align and configure Azure services to NIST Cybersecurity Framework (CSF) core functions to achieve security in the cloud.
After completing this course you will be able to:

- Identify and manage the data, personnel, devices, systems, and facilities to meet the organization’s business objectives and risk strategy
- Protect assets and associated facilities by using Access Control, limiting access to authorized users, processes, or devices, and to authorized activities and transactions
- Protect data-at-rest and data-in-transit by leveraging security services such as Azure Storage Service Encryption, Azure Backup Data Encryption, Azure SQL Transparent Data Encryption BitLocker, Azure VPN Gateway

- Detect anomalous activity in a timely manner and understand the potential impact of events by using Azure Security Center, Advanced Threat Analytics, Design and Implementation for Active Directory (DIAD), SIEM integration and Cloud App Security
- Ensure response processes and procedures are executed and maintained to ensure timely response to detected cybersecurity events

□ DSO 256 – DevSecOps in the Google Cloud Platform **(20 mins)**

Using a cloud Platform solves issues with distributed complexity and provides DevOps automation with a standard and centralized platform for testing, deployment, and production creating a complementary relationship between the two. This course provides learners with an understanding of how to align and configure Google Cloud Services to meet the NIST Cybersecurity Framework (CSF) core functions to achieve security in the cloud.

Upon successful completion of this course, you will have the knowledge and skills to:

- Use Identity and Access Controls to define, enforce, and manage user access policies with Google Cloud Identity and Access Management (IAM)
- Develop strategies for integrating security into your DevOps pipeline
- Use different strategies for securing pipeline resources
- Understand different methods for protecting secrets used in deployment applications

□ DSO 301 – Orchestrating Secure System and Service Configuration **(20 mins)**

Building and maintaining quality software requires functional configuration management, but this is easier said than done in today’s day and age. This process involves automation, but minimizing errors while securely and systematically managing changes in systems is complicated. This course provides Systems Developers, Network Operations Specialists, System Administrators, and Systems Security Analysts with the necessary skills to consistently and securely manage environments.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- Identifying and mitigating gaps in your current orchestration security policies
- Ensuring the coordination and consistency of security policies across the enterprise
- The importance of maintaining immutability of live container instances, ensuring that changes occur in the source control and are only deployed via new versions of the resource
- Understanding the role of third-party tools such as Clair, Actuary, and Anchore in testing Infrastructure-as-Code (IAC) and Configuration-as-Code (CAC) platforms

□ DSO 302 – Automated Security Testing **(20 mins)**

Modern application development, increasing speed-to-market requirements, and assuring application security have made automated security testing a top priority for many organizations. Automating Security Testing can be difficult and daunting, but incorporating into workflows can provide consistency, expedience, and ensure software quality. This course teaches learners to integrate the built-in strengths of DevOps within the security Testing process while adhering to security testing needs.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- Understanding the importance of orchestrating secure system and service configuration
- Determining which types of automated tests should be performed at various stages of the software development lifecycle
- Creating policies that support simultaneous testing and building in keeping with DevSecOps secure software development

- Leveraging Information Security Continuous Monitoring (ISCM) tools to perform a broad range of tasks, including periodic security and vulnerability scans of all system components

□ DSO 303 – Automating Security Updates **(20 mins)**

Essential to keeping systems secure, reducing risk, introducing new or enhanced features, or improving compatibility, software updating can be challenging and resource-intensive. Automating this process eliminates routine tasks and frees up administrative time. This course introduces automation procedures for systems administration to effectively and efficiently manage IT software in adherence to functional and security requirements.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- Employ automated mechanisms to implement changes to the current system baseline and deploy the updated baseline across the installed base
- Review system changes to determine whether unauthorized changes have occurred
- Remove previous versions of software or firmware components after installing updated versions
- Ensure that security-relevant software or firmware updates are obtained from authorized sources with appropriate digital signatures

□ DSO 304 – Securing API Gateways in a DevSecOps Framework **(20 mins)**

APIs are a critical component of cloud computing, and modern development fueling the success of DevOps. This course enables learners to implement mechanisms to securely manage API requests through the use of API gateways in DevOps and serverless environments.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- Deployment of secure API gateways through the implementation of core features such as to request and response collapsing API Transformation, and Protocol Translation for microservice-based applications
- Implement secure Identity and Access Management (IAM) across all services
- Provide certificate management, secrets management, and encryption services
- Leverage APIs to gather, synthesize and alert on security-relevant events as part of a comprehensive cybersecurity risk management program

□ DSO 305 – Automating CI/CD Pipeline Compliance **(20 mins)**

The adoption of cloud infrastructure and DevOps requires consistent integration of security to achieve a reliable lifecycle of continuous deployment. Integrating compliance into the CI/CD Pipeline requires a coordinated effort by everyone involved in the development pipeline. This course enables learners to automate the implementation of security tasks across the CI/CD pipeline in adherence to compliance requirements.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- Automate scanning and reporting tasks to ensure privacy policies, applicable laws, regulations, and service-level agreements are reviewed and documented for compliance regulations
- Identify and document controls owned by outside parties
- Configure change monitors to identify changes to organizational systems and environments of operation that may affect security and privacy risk
- Verify that all control objectives are met, and all key controls are designed and operating effectively

□ DSO 306 – Implementing Infrastructure as Code **(20 mins)**

Used to automate infrastructure deployment processes, Implementing Infrastructure as Code comes with a unique set of challenges making it hard for organizations to maintain agility, control, and visibility. This course is designed to help developers leverage Infrastructure as Code to securely and effectively launch cloud environments.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- Using tools in the development stage to help convert requirements into source code
- Leveraging the security features available in most integrated development environments (IDEs) for multiple programming language support
- Identify and mitigate the most common IaC vulnerabilities, including Weak Authentication Tokens, Disclosure of Authentication Credentials, Excessive Privileges or Capabilities, Misconfigured Network Filtering, and Missing Encryption

□ DSO 307 – Secure Secrets Management **(20 mins)**

As the need to protect critical data increases, organizations must focus efforts on improving processes used to manage essential information. This course is designed to ensure software development teams employ appropriate techniques to manage identities, privileges, and secrets securely.

Upon successful completion of this course, learners will have the knowledge and skills required to meet compliance requirements while developing a DevSecOps mindset, including:

- Ensuring that approved cryptographic algorithms and methods are used for securing critical assets
- Aligning key-management processes and procedures with those recognized by industry-standards bodies
- Using Approved Random Number Generators| Providing strong entropy when Using Random Number Generator

Infrastructure Security

□ API 250 – Controlling Access to the Kubernetes API **(20 mins)**

At the core of Kubernetes' control plane is the API server and the HTTP API that it exposes. The Kubernetes API lets you query and manipulates the state of objects in Kubernetes. This course gives learners an understanding of the role secure access control plays in protecting the Kubernetes API. Controlling who has access and what actions they are allowed to perform must be the primary concern. Learners will understand how to control the Kubernetes platform and how to use API requests as the first line of defense against attackers.

On successful completion of this course, learners should have the knowledge and skills required to:

- Encrypt all traffic by default using Transport Layer Security (TLS)
- Leverage Kubernetes in-built API server authentication mechanisms for non-production or small clusters
- Implement role-based access control using API Authorization

□ COD 252 – Securing Google Platform Applications & Data **(25 mins)**

Google Cloud Platform adoption provides many organizations with the agility and scalability needed to transform their business but lack of awareness surrounding best security practices and control implementation increases the risk of a security breach. This course provides the knowledge and skills to implement and leverage GCP security features, manage secrets, and protect applications and data against common threats.

Topics Include:

- Google Cloud Platform security features
- Creating, managing, and protecting secrets
- Common security threats
- Google Cloud monitoring and auditing facilities.

□ DES 214 – Securing Infrastructure Architecture **(30 mins)**

This course is designed for Network Operations Specialists and aligns with the NICE requirements for the secure planning, implementation, and operation of network services and systems, including hardware and virtual environments.
Coverage includes:

- Security Principles
- Network Topologies
- Demilitarized Zones
- Routers; Switches; Bridges; and Firewalls
- Wireless Access Points
- Transmission Media
- Network Authentication
- Server Configuration

□ DES 215 – Defending Infrastructure **(30 mins)**

This course is designed for the System Administrator role and aligns with the NICE requirements for system administration on specialized cyber defense applications and systems (e.g; antivirus, audit, and remediation) or Virtual Private Network (VPN) devices, to include installation, configuration, maintenance, backup, and restoration.

□ DES 216 – Protecting Cloud Infrastructure **(40 mins)**

This course provides DevOps Engineers, IT Architects and Network Engineers responsible for the security of applications and data with the skills and knowledge required to protect their organization’s cloud infrastructure.
Topics Include:

- The role of Data Encryption
- Identity and Authentication
- Firewalls and Network Security (SDNs, VPNs, DMZs)
- Division of Duties
- Compliance requirements
- Securing VM and Containers

□ DES 217 – Securing Terraform Infrastructure and Resources **(20 mins)**

Terraform helps create a workflow and combine multiple automation tasks across a broad range of Infrastructure resources using configuration files, including IaaS, PaaS, SaaS, and hardware services. This course provides an understanding of how to securely use Terraform in infrastructure as code (IaC) deployments without disrupting automation and performance.
After completing this course, you will be able to:

- Identify and mitigate the most common IaC vulnerabilities
- Leverage the security features and capabilities of Terraform
- Maintain security of infrastructure definitions and deployment secrets

□ DES 218 – Protecting Microservices, Containers, and Orchestration **(30 mins)**

Using Microservices, organizations can isolate software functionality into multiple independent modules that are individually responsible for performing precisely defined, standalone tasks communicating with each other through simple, universally accessible application programming interfaces (APIs). Containers enable developers to simultaneously build and ship these microservices; integrate them with other systems and automatically orchestrate them using predefined rules and processes.

This course is designed to educate DevOps Engineers, IT Architects, and Network Engineers working in Linux or on the cloud to add value to the application lifecycle through proper orchestration and enable faster development and fault-prone provisioning and configurations.

Topics Include:

- Hardening the OS
- Vulnerability Scanning
- Docker, SELinux and AWS Microservices
- API Gateways
- Node monitoring with Prometheus
- Implementing OAuth
- Schedulers and Orchestrators
- Defining a Pod Security Policy
- Using Metadata Concealment

□ DES 219 – Securing Google’s Firebase Platform **(60 mins)**

Google Firebase offers an active backend as a service (BaaS) for building dynamic web and mobile applications, but it has disadvantages. This course gives learners a fundamental understanding of how Firebase Security Rules leverage extensible, flexible configuration languages to define what data your users can access for Realtime Database, Cloud Firestore, and Cloud Storage. Firebase Realtime Database Rules leverage JSON in rule definitions, while Cloud Firestore Security Rules and Firebase Security Rules for Cloud Storage leverage a unique language built to accommodate more complex rules-specific structures.

On successful completion of this course, learners should have the knowledge and skills required to:

- Understand Firebase security rules, security concepts, and setup
- Path match rules, Read and Write operations, Conditions and Functions
- Implement security for reading Documents
- Conduct role-based Authentication and Unit Testing
- Setup for data testing in Firestore
- Write tests for Firestore security

□ DES 261 – Securing Serverless Environments **(20 mins)**

Serverless computing has redefined how companies build, consume, and integrate cloud-native applications. This course introduces the best-practices developers, and cloud customers should follow when using a serverless architecture. Learners will develop an understanding of the fundamental technologies serverless architectures use and how they should be secured from a development perspective to protect against the most common threats to serverless environments today.

On successful completion of this course, learners should have the knowledge and skills required to:

- Identifying Key Components
- Secure the API Entry Point, Runtime Environment, and Data Storage
- Identify a reliable and secure cloud provider
- Account for cloud provider considerations

□ DES 313 – Hardening a Kubernetes Cluster **(20 mins)**

This course provides learners with an understanding of how to secure a Kubernetes ecosystem in accordance with compliance standards. The content and recommendations in this course align with CIS, NIST, NSA-CISA, PCI-DSS, and HIPAA data and privacy requirements.

After completing this course, you will understand:

- Logging and auditing strategies for secure administration
- Implementing AAA controls to monitor user and group permissions
- Configuring system maintenance configurations to harden user access and file paths
- Leveraging HIPAA resources via NIST
- Filesystem configuration methodologies
- Services that should be disabled
- Network configurations to harden containers

☐ **DES 314 – Hardening the Docker Engine** **(15 mins)**
Securing Docker depends mostly on your organization and its IT offerings to end users. To fully secure Docker, a multi-faceted approach should include Kernel Namespaces, Control Groups, Docker Daemon Attack Surface, Linux Kernel Capabilities, Docker Content Trust Signature Verification, and other Security Tools.

Upon completing this course, you should have the knowledge and skills to identify common Docker Engine vulnerabilities and automate software and application deployments.

☐ **ICS 310 – Protecting Information and System Integrity in Industrial Control System Environments** **(15 mins)**
Manufacturing organizations rely on industrial control systems (ICS) to monitor and control their machinery, production lines, and other physical processes that produce goods. Operational Technology (OT) encompasses a broad range of programmable systems or devices that interact with the physical environment (or manage devices that interact with the physical environment). These systems/devices detect or cause a direct change through the monitoring and/or control of devices, processes, and events. This course provides guidance on how to improve the security of Operational Technology (OT) systems while addressing their unique performance, reliability, and safety requirements.

On successful completion of this course, learners should have the knowledge and skills required to:

- Understand how to identify typical threats to organizational mission and business functions supported by Operational Technology (OT)
- Describe typical vulnerabilities in Operational Technology (OT)
- Provide recommended security safeguards and countermeasures to manage the associated risks

Learn Labs

☐ **LAB 101 – Identifying Broken Access Control Vulnerabilities** **(5 mins)**
This lab presents a challenge in the Shadow Bank cyber range that exploits a Broken Access Control vulnerability, caused in part by missing or broken input validation and a business logic flaw. According to OWASP.org “Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification or destruction of all data, or performing a business function outside of the limits of the user.

In this lab, you are an adversary acting outside of your intended permissions, attempting to sell the stock you don’t own.

☐ **LAB 102 – Identifying Broken Object-Level Authorization Vulnerabilities** **(5 mins)**
In this lab, while authenticated as an adversary, you will interrupt the purchase process to substitute the object ID of someone else’s credit card with that of your own. A proper

authorization check, if implemented, should prevent you from completing the purchase, as you should not be allowed to use credit cards that are not associated with your account.
This lab presents a challenge in the LetSee cyber range that exploits a Broken Object-Level Authorization vulnerability by allowing an adversary to charge a purchase to someone else's credit card. Adversaries can exploit failures in complex authorization mechanisms of API-based applications by manipulating parameters such as object IDs sent in requests.

- □ **LAB 103 – Identifying Broken User Authentication Vulnerabilities** **(7 mins)**
  Authentication is the process of attempting to verify identity; Problems with authentication can be introduced at many phases throughout the software development life cycle, so adversaries have a potentially broad attack surface to work with. One technique adversaries use and learners can perform as part of penetration testing is to interact with aspects of the authentication mechanism to find valid identifiers. Registration, or the process of creating new accounts, is part of authentication. This lab presents a challenge in the Gold Standard cyber range that reveals a Broken User Authentication vulnerability. The challenge is "Register as Loan Officer." Abusing the registration functionality allows an adversary to bypass filters or access controls in Gold Standard to gain access to a default higher-privilege account.

- □ **LAB 104 – Identifying Business Logic Flaw Vulnerabilities** **(7 mins)**
  This lab presents a challenge in the Account All cyber range that exploits a Business Logic Flaw vulnerability caused in part by improper input validation. Adversaries exploiting business logic flaws take advantage of the legitimate processes of an application, many times by interacting with the application in unexpected ways. Business rules or business logic implemented in the application should prevent users from doing harmful or nonsensical actions. However, flaws in the design of such logic can lead to adversaries circumventing these rules. In this lab, you are attempting to set a value for your W2 withholding that does not make sense. In the USA, this value is used in calculating the amount an employer withholds from an employee's pay over the course of the year for tax purposes.

- □ **LAB 105 – Identifying Credential Dumping: Vulnerability Identification** **(7 mins)**
  Use access techniques to exfiltrate an unprotected facility leftover by developers of a banking website from part of their testing suite to download credentials in the production site and exploit a credential dumping vulnerability.

- □ **LAB 106 – Identifying Cross-Site Scripting Vulnerabilities** **(7 mins)**
  This lab presents a challenge in the Account All cyber range that reveals the presence of a Cross-Site Scripting vulnerability, caused in part by improper input validation and filtering. Cross-site scripting vulnerabilities are web-based vulnerabilities that can be exploited whenever a web application embeds untrusted input data in site content or web responses without first validating the data or its encoding. In this lab, you are an adversary performing tests to determine whether Cross-Site Scripting vulnerabilities are present on the Timesheets page.

- □ **LAB 107 – Identifying Injection Vulnerabilities** **(7 mins)**
  This lab presents a challenge in the Account All cyber range that exploits an Injection vulnerability, caused in part by improper input validation and query handling. According to OWASP.org, "Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization." This lab's challenge is to force the Log In page to generate an unhandled exception. Solving this challenge will demonstrate the presence of a particular type of Injection

vulnerability. In this lab, you are an adversary acting outside of your intended permissions, attempting to input improper validation and potentially expose sensitive information.

□ **LAB 108 – Identifying Reverse Engineering Vulnerabilities** **(8 mins)**
This lab presents a Reverse Engineering challenge in the Runstoppable cyber range which simulates a mobile fitness application. In this lab, you are an adversary using Reverse Engineering to look for a Hardcoded Secret "READTHISFLAG1 " in the Runstoppable code. Competitive cyber range activities are sometimes known as "Capture the Flag" events. In this lab, the flag is the Hardcoded Secret that you will find by using Reverse Engineering, but in a real application, an adversary might discover other, more valuable secrets using this technique.

□ **LAB 109 – Identifying Security Misconfiguration Vulnerabilities** **(5 mins)**
Security Misconfiguration is not limited in scope to the application code itself. Improperly secured operating systems, web server applications, and databases all contribute to the overall attack surface. This lab presents a challenge in the InstaFriends cyber range that exploits an Integer Overflow vulnerability in its Messaging functionality, which in turn reveals a Security Misconfiguration vulnerability.

□ **LAB 110 – Identifying Sensitive Data Exposure Vulnerability Identification** **(7 mins)**
This lab presents a challenge in the ShadowBank cyber range that exploits a Sensitive Data Exposure vulnerability caused in part by a Hardcoded Secret and Missing or Weak Encryption. The discovered information leads to further Sensitive Data Exposure by exploiting a revealed Broken Access control vulnerability. To solve the challenge "Into the Shadows: Cryptanalysis" we are looking for a Hardcoded Secret encoded with Weak Encryption.

□ **LAB 111 – Identifying Server-Side Request Forgery** **(5 mins)**
This lab on Server-Side Request Forgery (SSRF) assesses the learner's understanding of how an existing SSRF vulnerability in a cloud application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities, leverage trust relationships among back-end systems protected by network topology but lacking more sophisticated access controls, and access resources not directly reachable by end-users.

□ **LAB 113 – Identifying Cryptographic Failures** **(5 mins)**
This lab challenges a learner to discover and exploit an existing cryptographic failure in the password hashing functionality of an online banking application. In this lab, you are an adversary leveraging tools to crack passwords and gain access to user accounts where you can perform all the actions the legitimate user can or move laterally in the system or application. In addition to exploring symptoms and causes under this category, participants will learn how to prevent and mitigate cryptographic failures.

□ **LAB 114 – Identifying Cookie Tampering** **(5 mins)**
This lab on Cookie Tampering assesses the learner's understanding of how an existing Cookie Tampering vulnerability in an online banking application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to bypass security policies and gain access or privileges.

□ **LAB 115 – Identifying Reflective XSS** **(5 mins)**
This lab on Reflective XSS assesses the learner's understanding of how an existing Reflective XSS vulnerability in an online e-commerce application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to execute arbitrary commands, such as JavaScript, and display arbitrary content in a victim's browser.

- □ **LAB 116 – Identifying Forceful Browsing** **(5 mins)**
  This lab on Forceful Browsing assesses the learner’s understanding of how an existing Forceful Browsing vulnerability in an online Human Resources (HR) back-office application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to bypass weak access controls and gain access to what should be restricted resources and higher privileged operations.

- □ **LAB 117 – Identifying Hidden Form Field** **(5 mins)**
  This lab on Hidden Form Fields assesses the learner’s understanding of how an existing vulnerability related to hidden form fields in an online banking application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to tamper with client-side data, in this case for monetary gain. Improper validation of hidden yet mutable field values potentially paves the way for other attacks such as Cross-Site Scripting, SQL Injection, or even gaining unauthorized access.

- □ **LAB 118 – Identifying Weak File Upload Validation** **(5 mins)**
  This lab on Weak File Upload Validation assesses the learner’s understanding of how an existing Weak File Upload Validation vulnerability in an online banking application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to upload malicious files to escalate privileges, execute arbitrary code, compromise the application, or compromise the host server.

- □ **LAB 119 – Identifying Persistent XSS** **(5 mins)**
  This lab on Persistent XSS assesses the learner’s understanding of how an existing Persistent XSS vulnerability in an online social media application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to leave malicious payloads that will continue to affect subsequent victims that visit the page.

- □ **LAB 120 – Identifying XML Injection** **(5 mins)**
  This lab on XML Injection assesses the learner’s understanding of how an existing XML Injection vulnerability in an online banking web application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to bypass authentication mechanisms and gain access to an application, sometimes with high-level privileges.

- □ **LAB 121 – Identifying Vulnerable and Outdate Components** **(5 mins)**
  This lab challenges a learner to discover and exploit the use of a vulnerable and outdated component in an online banking application that fails to properly validate the supply chain. In this lab, the outdated web framework used has a known vulnerability to Denial-of-Service attacks that can shut down the entire server. After completing this lab, the learner will understand how adversaries can exploit any known vulnerabilities in underlying components of your application and best practices to avoid and mitigate them.

- □ **LAB 122 – Identifying Insecure APIs** **(5 mins)**
  This lab challenges a learner to discover and exploit an existing API vulnerability to bypass authorization mechanisms and steal private files in a cloud application. In this lab, you are an adversary interacting with the application in a legitimate way to discover flaws in a REST API to bypass authorization mechanisms and steal private files that contain AWS Credentials. Participants will also learn best practices to prevent and mitigate broken object-level authorization vulnerabilities related to insecure APIs?

- □ **LAB 123 – Identifying Vertical Privilege Escalation** **(5 mins)**

  This lab challenges a learner to discover and exploit an existing credential management error in a cloud application to gain initial access and then escalate their privileges. In this lab, you are an adversary attempting to manually approve your own purchase using post parameter manipulation and vertical privilege escalation. Participants will also learn best practices to prevent and mitigate account hijacking and vertical privilege escalation exploitation.

- □ **LAB 124 – Identifying Horizontal Privilege Escalation** **(5 mins)**

  This lab on Horizontal Privilege Escalation assesses the learner's understanding of how existing Broken Object-level Authorization and Weak or Missing Cryptography vulnerabilities in an e-commerce application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to gain unauthorized access to objects belonging to other users with the same level of privilege in order to exfiltrate, tamper with, or destroy them.

- □ **LAB 125 – Identifying Buffer Overflow** **(5 mins)**

  This lab on Buffer Overflow assesses the learner's understanding of how an existing Buffer Overflow vulnerability in a cryptocurrency cyber range can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to cause the arbitrary execution of malicious code with the application's privileges, often without requiring any user interaction.

- □ **LAB 126 – Identifying Information Leakage** **(5 mins)**

  This lab on Information Leakage assesses the learner's understanding of how existing Insufficiently Protected Credentials and Insecure API vulnerabilities in a social media application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to gain unauthorized access to an area of the site which in turn exposes credentials, enabling adversaries to impersonate legitimate users.

- □ **LAB 127 – Identifying Security Logging and Monitoring Failures** **(5 mins)**

  This lab on Security Logging or Monitoring Failures assesses the learner's understanding of how an existing Insecure API vulnerability in an online e-commerce application can be discovered and exploited, revealing sensitive logging information. After completing this lab, the learner will understand how adversaries can probe insecure applications to exploit such vulnerabilities and gain insight into the inner workings of your application or data relationships.

- □ **LAB 128 – Identifying Unverified Password Change** **(5 mins)**

  This lab on Unverified Password Changes assesses the learner's understanding of how an existing Identification and Authentication Failure vulnerability in an online e-commerce application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can defeat weak cryptography and exploit broken password change mechanisms to take over other users' accounts.

- □ **LAB 129 – Identifying Error Message Containing Sensitive Information** **(5 mins)**

  This lab on Error Messages Containing Sensitive Information assesses the learner's understanding of how an existing SQL Injection vulnerability in an online cryptocurrency trading application can be discovered and exploited to trigger error messages from other insecurely configured layers. After completing this lab, the learner will understand how adversaries can probe insecure applications to exploit such vulnerabilities which expose the inner workings of your application or data relationships.

□ **LAB 130 – Identifying Generation of Predictable Numbers or Identifiers** **(5 mins)**

This lab on Generation of Predictable Numbers or Identifiers assesses the learner's understanding of how such an existing vulnerability in an online e-commerce application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can probe insecure applications to exploit such vulnerabilities and leverage this information to gain other users' credentials.

□ **LAB 131 – Identifying Improper Restriction of XML External Entity Reference** **(5 mins)**

This lab on Improper Restriction of XML External Entity References assesses the learner's understanding of how an existing Improper Restriction of XXE References vulnerability in a cloud-native marketing automation SaaS suite can be discovered and exploited.
After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to upload malformed XML documents. Such attacks may lead to the disclosure of confidential data, denial of service, server side request forgery, and other system impacts.

□ **LAB 132 – Identifying Exposed Services** **(5 mins)**

This lab on Exposed Services assesses the learner's understanding of how an existing security misconfiguration in a cloud-native marketing automation SaaS suite can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to grab server banners from HTTP response headers and leverage the information exposed to launch targeted attacks.

□ **LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables** **(5 mins)**

This lab on Exposure of Sensitive Information Through Environmental Variables assesses the learner's understanding of how such an existing vulnerability on a server hosting an ecommerce application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to steal secrets, gain unauthorized access, establish persistence, penetrate further into a system, and plan more damaging attacks.

□ **LAB 134 – Identifying Plaintext Storage of a Password** **(5 mins)**

This lab on Plaintext Storage of Passwords assesses the learner's understanding of how an existing Credentials Management Error in the database supporting an ecommerce application can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to steal secrets, gain unauthorized access, establish persistence, penetrate further into a system, and plan more damaging attacks.

□ **LAB 135 – Identifying URL Redirection to Untrusted Site** **(5 mins)**

This lab on URL Redirection to Untrusted Site (otherwise known as Open Redirect) assesses the learner's understanding of how an existing Open Redirect vulnerability in a cloud-native marketing automation SaaS suite can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to send users to a malicious site via a legitimate-looking URL to compromise their machine with malware or steal their credentials.

□ **LAB 136 – Identifying Improper Neutralization of Script in Attributes in a Web Page** **(5 mins)**

This lab on Improper Neutralization of Script in Attributes in a Web Page assesses the learner's understanding of how an existing persistent cross-site scripting vulnerability in the email templates of a cloud-native marketing automation SaaS suite can be discovered and exploited. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to leave malicious payloads that will continue to affect subsequent victims that use the template or receive emails generated by the template.

□ **LAB 137 – Identifying Improper Authorization (NEW)** **(5 mins)**
This lab on Improper Authorization assesses the learner's understanding of how such an existing vulnerability in a cloud-native marketing automation SaaS suite can be discovered and exploited. This vulnerability can lead to sensitive data exposure, arbitrary code execution, or other high-impact problems. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to access data or perform actions that otherwise they would not be able to if access control checks were properly implemented.

□ **LAB 138 – Identifying Authorization Bypass Through User-Controlled Key (NEW)** **(5 mins)**
This lab on Authorization Bypass Through User-Controlled Key assesses the learner's understanding of how an existing Insecure Direct Object Reference vulnerability in an e-commerce application can be discovered and exploited via parameter tampering.
After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities to bypass authorization and read or modify another user's data. If the user-controlled key identifies role or session instead of referencing a resource, the adversary may gain privileges or assume another user's identity.

□ **LAB 139 – Identifying Use of a Key Past its Expiration Date (NEW)** **(5 mins)**
This lab on Use of a Key Past its Expiration Date assesses the learner's understanding of how existing Credentials Management and Security Misconfiguration vulnerabilities in a cloud file storage application suite built on AWS can be discovered and exploited to gain unauthorized access to sensitive data. After completing this lab, the learner will understand how adversaries can exploit such vulnerabilities, in this case to steal code from a repository and modify it to use outdated credentials to access an otherwise private file.

Secure Design

□ **COD 104 – Designing Secure Software** **(15 mins)**
This course provides learners with the skill and knowledge required to perform threat modeling, and ensure that security principles are applied at each step of the design process.
Topics Include:

- Integrating attack surface reduction
- Secure defaults
- Least privilege
- Defense in depth
- Compartmentalization

□ **CYB 210 – Cybersecurity Incident Response** **(12 mins)**
This course provides learners with an understanding of the role Cybersecurity Incident Response plays within your organization's overall security plan. The content and the recommendations in the course align with CIS, NIST, NSA-CISA, PCI-DSS, and HIPAA guidelines.
After completing this course, you will understand:

- Parties who should be involved in establishing a Cybersecurity Incident Response plan
- Details about what should be included in the plan's development
- Execution of key portions of the plan after its development

□ CYB 211 – Identifying and Protecting Assets Against Ransomware **(12 mins)**

Ransomware is an evolving threat to the cyber and data security of many organizations. This course provides learners with an understanding of the role identifying and protecting assets plays in protecting against ransomware attacks. Learners will gain a better understanding of the attacker's mindset and the ransomware "Business Model."

On successful completion of this course; learners should have the knowledge and skills required to:

- Implement network segregation, anti-virus, and endpoint security
- Create a system shutdown policy, using Virtual Machines (VMs) for instant backup restore (snapshotting) and recovery of business processes and assets

□ CYB 212 – Fundamentals of Security Information & Event Management (SIEM) **(15 mins)**

Security Information & Event Management platforms have become a significant component in streamlining security workflows, but, as powerful as these platforms can be, they can be inherently challenging. This course provides learners with an understanding of the role of Security Information & Event Management (SIEM) in your organization's overall security plan.

On successful completion of this course, learners should have the knowledge and skills required to:

- Detect known and emerging threats
- Identify vulnerabilities
- Accelerate incident response
- Identify policy violations
- Provide system troubleshooting or forensic evidence in the event of a security breach

□ DES 101 – Fundamentals of Secure Architecture **(20 mins)**

In the past, software applications were created with little thought to the importance of security. Recently, businesses have become more rigorous about how they buy and deploy software as security is a large part of the total cost that risk software applications inherently carry. In this course, you examine the state of the industry from a security perspective, setting the foundation for secure software development.

Topics include:

- Application security architecture principles
- Lessons learned: security disasters in software design
- How to use confidentiality, integrity, and availability to drive better security design decisions

□ DES 151 – Fundamentals of the PCI Secure SLC Standard **(25 mins)**

The PCI Secure SLC Standard outlines security requirements and assessment procedures for software vendors to validate how they properly manage the security of payment software throughout the entire software lifecycle. This course provides baseline knowledge needed to implement security requirements and assessment procedures to validate proper management of the security of payment software throughout the entire software lifecycle.

After completing this course you will be able to:

- Identify and mitigate common threats and vulnerabilities defined in the PCI Secure SLC standard
- Build an environment for secure software development, change control, and management
- Improve communications for secure deployment, configuration and software updates.

□ DES 202 – Cryptographic Suite Services: Encoding, Encrypting & Hashing **(45 mins)**

This course presents an overview of the fundamental services provided by cryptographic suites, namely encoding, encrypting, and hashing.
Topics include:

- Encoding and decoding
- Encryption and decryption
- The difference between encoding and encryption
- The value and application of hashing
- Where, when, and how to use crypto

This course aligns with the National Initiative for Cybersecurity Education (NICE) requirement(s): K0018: Knowledge of encryption algorithms.

□ DES 203 – Cryptographic Components: Randomness, Algorithms, and Key Management **(15 mins)**

This course introduces three important elements of cryptographic systems: random number generation, algorithms and keys.
Topics include:

- The critical role of randomness in cryptography
- Common algorithms to perform cryptographic manipulation of information
- Types and roles of cryptographic keys
- The key management problem
- Common types of digital certificates and its creation process
- Components and roles of a public key infrastructure
- Weaknesses in the digital certificate trust mode
- Mechanisms to manage and distribute cryptographic keys

This course aligns with the National Initiative for Cybersecurity Education (NICE) requirement(s):

- K0018: Knowledge of encryption algorithms
- K0019: Knowledge of cryptography and cryptographic key management concepts

□ DES 204 – Role of Cryptography in Application Development **(15 mins)**

This course introduces cryptography and how cryptography can help secure software applications and data. It also provides an overview of common uses of cryptography.
Topics include:

- Identifying relevant cryptographic technologies
- Knowing common “data-in-motion” crypto options and the strengths/weaknesses of each
- Applying common “data-at-rest” crypto options and the strengths/weaknesses of each

□ DES 205 – Message Integrity Cryptographic Functions **(45 mins)**

This course explains how encrypting and signing a message works, how message authentication codes work, and why a digital signature is superior to a cryptographic hash for validating software integrity.
Topics include:

- Message integrity function is its value
- The difference between a message authentication code and a digital signature
- How a digital signature works
- Encrypting and signing messages
- Message authentication codes

- Digital signature vs. a cryptographic hash for validating software integrity

This course aligns with the National Initiative for Cybersecurity Education (NICE) requirement(s):

- K0018: Knowledge of encryption algorithms
- K0019: Knowledge of cryptography and cryptographic key management concepts

□ DES 206 – Meeting Cloud Governance and Compliance Requirements **(15 mins)**

The adoption of cloud services involves various roles making it difficult to govern the selection and brokering of cloud services while adhering to policies and procedures. This course is designed to ensure privacy and security teams may effectively and efficiently adopt cloud computing in support of strategic and business goals.

Upon successful completion of this course, learners will have the knowledge and skills required to meet privacy compliance requirements, including:

- Creating Policies, Procedures, Standards, and Controls that meet all regulatory and legal requirements, industry standards, and technical controls such as encryption.
- Establishing, deploy and assess a compliance baseline that determines targets
- Handling sensitive data, including how to identify and classify data, define data retention periods, and comply with data storage requirements.
- Prepare for compliance auditing and Reporting

□ DES 209 – Authentication and Lifecycle Management **(15 mins)**

The objectives of this course align with NIST Special Publication (SP) 800-63, Digital Identity Guidelines, and explains the fundamentals of authentication and how to maintain strong account access and authentication policies. After successfully completing this course, you will understand secure authentication methods, including the function of the access provisioning lifecycle, credential service providers, Public Key Infrastructure (PKI), and Federation basics.

□ DES 210 – Hardening Linux/Unix Systems **(30 mins)**

Hardening is a critical step in ensuring security and diligence as it reduces the chances of attack, but this requires the use of appropriate methodologies. In today's connected world securing an operating system has become increasingly sophisticated as computing ecosystems increase in complexity. This course provides learners with an understanding of best practices for hardening Linux and Unix systems.

After completing this course you will be able to:

- Upgrade your kernel
- Disable root cron jobs
- Enforce strict firewall rules
- Disable unnecessary services
- Check for backdoors and rootkits
- Check listening ports
- Monitor and manage logs using IDS

□ DES 212 – Architecture Risk Analysis & Remediation **(30 mins)**

This course defines concepts, methods, and techniques for analyzing the architecture and design of a software system for security flaws. Special attention is given to analysis of security issues in existing applications; however, the principles and techniques are applicable to systems under development. Techniques include accurately capturing application architecture, threat modeling with attack trees, attack pattern analysis, and enumeration of trust boundaries.

Topics include:

- How to assess design components for security flaws
- The use and value of threat modeling and attack surface analysis

- Techniques to remove architecture weak spots and avoid vulnerability propagation

□ DES 255 – Securing the IoT Update Process **(30 mins)**

Addressing updates across the Internet of Things (IoT) can be complicated due to the complex ecosystems of connected devices deployed across multiple environments. This course aims to educate learners to establish a secure, scalable update process for IoT devices.

After completing this course, you will be able to:

- Identify the risks of delivering IoT device updates
- Understand each phase in the IoT update process
- Determine considerations for the secure delivery of updates to the vehicle
- Securely design, develop, delivery, and install IoT update

□ DES 260 – Fundamentals of IoT Architecture & Design **(30 mins)**

This course focuses on topics related to architecting and designing a secure Internet of Things (IoT) system. Particular emphasis is placed on embedded IoT devices and their relationship with cloud services.

After completing this course, you will have a deep understanding of an IoT system, its components, and the security implications of various design choices.

Topics include:

- Elements to be reviewed and defined in the requirements phase
- Authorization considerations within the IoT device itself as well as connected components
- Designing a secure IoT architecture
- Authentication to validate the identity of users and devices
- Logical access controls to ensure users are granted appropriate levels of service
- Physical security concerns to protect access to IoT devices
- Monitor communications throughout the IoT system
- Secure communications between the various system components

□ DES 262 – Securing Enterprise Low-Code Applications Platforms **(20 mins)**

Low-code application platforms present new vulnerabilities that organizations are not prepared for as they introduce unintended threats and connections between core systems and third-party applications. This course is designed to create awareness around security and privacy risks related to Low-Code Application Platforms (LCAP) applications and provide learners with a fundamental understanding how to identify and mitigate the security risks associated with Low-Code Application Platforms (LCAP).

On successful completion of this course, learners should have the knowledge and skills required to:

- Avoid creating vulnerabilities that put communications and other systems at risk
- Ensure the privacy of customers
- Secure Node.js queries

□ DES 305 – Protecting Existing Blockchain Assets **(20 mins)**

Blockchain implementation poses a number of challenges from storage capacity and scalability to anonymity and data privacy thus making the protection of existing assets complex. This course provides learners with an understanding of how to secure existing Blockchain assets against security threats.

After completing this course you will be able to:

- Secure the cryptographic keys that allow access to the ledger
- Use hardware security modules (HSMs)
- Ensure the integrity of “Smart Contracts”
- Protect communications between nodes

□ DES 306 – Creating a Secure Blockchain Network **(20 mins)**

While Blockchain technology continues to emerge for its ability to improve data security, speed up transactions and save costs, it comes with its advantages it comes with a wide array of challenges. Properly securing a blockchain network begins with the implementation of strong authentication and cryptography key vaulting mechanisms. This course provides learners with an understanding of the essential requirements for creating a secure blockchain network.

After completing this course you will be able to:

- Identify operational, legal and compliance requirements
- Create a blockchain threat model
- Create blockchain trust policies, access controls, and smart contracts
- Manage identity, access, entitlements, certificates, and keys
- Monitor, report, and manage incidents

□ DES 311 – Creating Secure Application Architecture **(45 mins)**

Architecting secure solutions is paramount to ensure developers do not incorporate insecure components, which could introduce hundreds of individual security vulnerabilities in the as-built system. This course covers a set of key security principles to improve the security of application architecture and design.

Topics include:

- Applying defense to harden applications and make them more difficult for intruders to breach
- Reducing the amount of damage an attacker can accomplish
- Compartmentalizing to reduce the impact of exploits
- Using centralized input and data validation to protect applications from malicious input
- Reducing the risk in error code paths

□ DES 312 – Protecting Cardholder Data **(20 mins)**

While cardholder data consists of any personally identifiable information (PII) associated with a person who has a credit or debit card, the PCI Secure Standards Council (PCI SSC) has specific requirements to protect cardholder data at all times. Despite common misconceptions, this also includes account numbers, expiration date, and/or service code as cardholder data. This course is designed to provide Information Systems Security Developers with the knowledge needed to minimize the storage of cardholder data and take necessary precautions to protect it in adherence to the PCI Software Security Framework and NIST 800-53 Guidelines.

Upon successful completion of this course, learners will have the knowledge and skills required to meet privacy compliance requirements, including:

- Ensuring the software does not store sensitive authentication data after authorization, even if encrypted unless the software is intended only for use by issuers or organizations that support issuing services.
- Rendering the Primary Account Number (PAN) is unreadable anywhere it is stored.
- Guiding customers regarding the secure deletion of cardholder data after the expiration of the customer-defined retention period.

□ ENG 150 – Meeting Confidentiality, Integrity, and Availability **(30 mins)**

The CIA Triad – Confidentiality, Integrity, and Availability are the information security tenets used as a means for analyzing and improving the security of your application and its data. After completing this course, you will be able to understand and use confidentiality, integrity, and availability (CIA) as the three main tenets of information security.

□ ENG 151 – Fundamentals of Privacy Protection **(10 mins)**

Staying current on legislation and engaging the business on timely privacy compliance and practical solutions can be challenging. As the focus on compliance continues to increase, and the GRC landscape continues to evolve, compliance officers need to keep pace with emerging regulations. This course provides learners with a clear understanding of their role in meeting compliance requirements.

Upon successful completion of this course, learners will have the knowledge and skills required to meet privacy compliance requirements, including:

- Enable better privacy engineering practices that support privacy by design concepts
- Ensure collaboration on privacy protection objectives across your organization
- Apply the NIST Privacy Framework to meet region-specific regulations such as GDPR and CCPA
- Communicate privacy practices with individuals, business partners, assessors, and regulators

□ ENG 191 – Introduction to the Microsoft SDL **(25 mins)**

This course introduces the industry-leading Microsoft Security Development Lifecycle (SDL) Optimization Model and how to implement it.

Topics include:

- Capability areas of the Microsoft SDL Optimization Model
- Maturity levels and how to reach them
- Optimization techniques to reduce risk

□ ENG 192 – Implementing the Agile Microsoft SDL **(20 mins)**

The standard MS SDL process follows the traditional incremental waterfall model, while Agile methodologies are more iterative. This course focuses on the Agile variation of the SDL process and covers the following topics:

- How to map critical SDL security practices into every-sprint requirements, bucket or periodic requirements, and one-time requirements
- How to incorporate security education, tooling and automation, threat modeling, fuzz testing, handling bug-dense and at-risk code, exceptions, and the final security review into sprints

□ ENG 193 – Implementing the Microsoft SDL Optimization Model **(12 mins)**

This course describes the main phases of the Microsoft Security Development Lifecycle (SDL) process: Requirements, Design, Implementation, Verification, and Release, with a focus on security throughout. After completing this course, you will have a solid understanding of the SDL process and the recommended/required tasks for each phase.

□ ENG 194 – Implementing Microsoft SDL Line of Business **(20 mins)**

This course describes the Microsoft Security Development Lifecycle for Line of Business (SDL-LOB), which focuses on the development of internal or business-facing applications.

Topics include:

- The five primary phases of the SDL: Requirements, Design, Implementation, Verification, and Release
- LOB-specific tasks, requirements and deliverables for each phase of the SDL
- How to integrate security-improving tasks at each level of risk
- Necessary skills to be effective

□ ENG 195 – Implementing the Microsoft SDL Threat Modeling Tool **(20 mins)**

This course describes the features of the Microsoft SDL Threat Modeling tool, which complements the Microsoft SDL Threat Modeling process. While not required to perform threat modeling, using the tool facilitates the creation of threat models and helps enumerate threats using STRIDE.

Topics include:

- Creating accurate data flow diagrams (DFDs) in your threat model
- Identifying flaws in DFDs and analyzing it for potential threats
- Generating reports to export threats to issue tracking tools

□ ENG 205 – Fundamentals of Threat Modeling **(45 mins)**

This course describes how to take a question-driven approach to threat modeling to help identify security design problems early in development process. After completing this course, you will be able to create a threat model for your application scenario and use it to refine your application's design and improve communication within the team.

□ ENG 211 – How to Create Application Security Design Requirements **(15 mins)**

To preserve the confidentiality, integrity, and availability of application data, software applications must be engineered with security in mind. Without defined security requirements, design choices will be made without security guidance and security testing cannot be effective. This course provides technical and non-technical personnel with the knowledge to understand, create, and articulate security requirements as part of a software requirement document.

Topics include:

- Applying the application security maturity (ASM) model to the development process
- Key security engineering activities: gathering security objectives, applying security design guidelines, and creating threat models
- Identifying threats, attacks, vulnerabilities, and countermeasures
- How to conduct impactful security architecture and design reviews to identify potential security problems and minimize the application's attack surface.

□ ENG 212 – Implementing Secure Software Operations **(20 mins)**

All software activity involving critical assets must be tracked, and any methods that may expose sensitive data should also be tracked as defined by control objectives within the PCI Software Security Framework. Unfortunately, protecting the integrity of event datasets and analyzing records to detect attacks in real-time can be challenging. This course is designed to equip Information Systems Security Developers and Software Developers with the knowledge required to detect, respond to, and investigate attacks.

Upon successful completion of this course, learners will have the knowledge and skills required to meet the Secure Software Operations requirements described in PCI's Secure Software Requirements and Assessment Procedures, including:

- Ensuring that all access attempts and usage of critical assets are tracked and traceable to a unique individual
- Facilitating the retention of detailed activity records either within the software itself or by supporting integration with other solutions such as centralized log servers, cloud-based logging solutions, or a back-end monitoring solution
- Ensuring that the software possesses the basic functionality to differentiate between normal and anomalous user behavior: such changes in post-deployment configurations or obvious automated-attack behaviors

□ ENG 251 – Risk Management Foundations **(20 mins)**

Risk management should be a foundational tool used to facilitate thoughtful and purposeful defense strategies. In today's environment, the most significant threats to systems come from purposeful attacks that are often disciplined, well organized, and well-funded.

This course aims to educate IT architects, Analysts, and DevOps Engineers to understand their responsibilities when protecting organizational assets.
Topics Include:

- Key Risk Management Concepts
- Common management techniques and strategies
- various risk assessment methods and risk control strategies

□ ENG 311 – Attack Surface Analysis & Reduction **(25 mins)**

The attack surface of an application represents the number of entry points exposed to a potential attacker. The larger the attack surface, the larger the set of methods that can be used by an adversary breaking into software applications. Resultantly, minimizing it is a key exercise in risk reduction.
Topics covered:

- Understanding the goals and methodologies of attackers
- Identifying attack vectors that expose the application
- Defining and reducing an application's attack surface

□ ENG 312 – How to Perform a Security Code Review **(30 mins)**

Application developers have a variety of tools at their disposal to identify flaws in their software. However, many of them cannot be used until late in the development lifecycle: dynamic analysis tools require a staging site and sample data, and some static analysis tools require a compiled build. In contrast, manual code reviews can begin at any time leveraging secure coding knowledge. Because manual security code reviews can be laborious if done inefficiently, this course focuses on time saving but effective techniques.
Topics include:

- How to organize and approach code reviews
- Prioritizing code segments to be reviewed
- Maximizing security resources

□ ENG 320 – Using Software Composition Analysis (SCA) to Secure Open-Source Components (NEW) **(20 mins)**

Software Composition Analysis (SCA) provides visibility into the open-source components and libraries being incorporated into the software that development teams create. SCA can help manage security and license-related risks. This course provides learners with a fundamental understanding of how to use Software Composition Analysis (SCA) tools to securely integrate open-source software into new code.
On successful completion of this course, learners should have the knowledge and skills required to:

- Discuss the security risks associated with software vulnerabilities and license compliance
- Understand the SCA Architecture and how the technologies help to make dependency checks possible
- Use the Software Bill of Materials (SBOM) and Vulnerability Databases to fully perform software analysis
- Understand Development Workflow Integration and SCA Limitations
- Use SCA for Containerized Applications and Infrastructure as Code (IaC)

□ ENG 351 – Preparing the Risk Management Framework **(20 mins)**

Before any organization can adequately Implement the Risk Management Framework they must understand how to determine and apply appropriate security requirements. Preparation requires a disciplined and structured set of activities in order to execute the framework at appropriate risk

management levels. This course aims to provide Engineers, Software Architects, and Systems Analysts with context and priorities for managing security and privacy risk.
Topics Include:

- Identifying key Individuals and specification of roles and responsibilities in the risk management process
- Identifying risk tolerance and determining a particular strategy for risk management
- Conducting an organization-level risk assessment to ensure leadership is aligned
- Continuous monitoring to enable a rapid and effective response to changes in the risk landscape or changes in the effectiveness of controls

□ ENG 352 – Categorizing Systems and Information within the RMF **(10 mins)**

Security categorization provides a structured way to determine the criticality and sensitivity of the information being processed, stored, and transmitted by an information system. This course provides learners with an understanding of how to categorize the system and the information using the NIST SP 800-37 Rev. 2 Risk Management Framework.
After completing this course you will be able to:

- Identify all information types based on the system boundary
- Categorize information (processed, stored, or transmitted) by the potential adverse impact that information being compromised as regards confidentiality, integrity or availability
- Ensure the security categorizations are consistent with roles, operating environment, connectivity, and intended use

□ ENG 353 – Selecting, Implementing and Assessing Controls within the RMF **(20 mins)**

Selecting the appropriate set of security controls helps to achieve organizational operations and objectives. This course provides learners with an understanding of how to select, implement and assess security controls using the NIST SP 800-37 Rev. 2 Risk Management Framework.
After completing this course you will be able to:

- Select and document the controls necessary to protect the information system and organization commensurate with the risk to the organization
- Implement the controls in the security and privacy plans for the system and organization
- Document the specific details of the control implementation in a baseline configuration
- Assess the controls to determine if the controls are implemented correctly, operating as intended, and producing the desired outcomes with respect to satisfying the security and privacy requirements

□ ENG 354 – Authorizing and Monitoring System Controls within the RMF **(20 mins)**

Authorizing and monitoring security controls provides an understanding of security posture and provides an indication of whether or not cybersecurity controls are operating as intended. This course provides learners with an understanding of the Authorization and Monitoring steps of the NIST SP 800-37 Rev. 2 Risk Management Framework.
After completing this course you will be able to:

- Provide organizational accountability by requiring a senior management official to determine if the security and privacy risk to operations, assets, and individuals is acceptable

- Report authorization decisions, significant vulnerabilities, and risks to organizational officials | Monitoring the system and the associated controls on an ongoing basis
- Document changes to the system and environment of operation
- Conduct risk assessments and impact analyses | Reporting the security and privacy posture of the system

Secure Development

□ **API 210 – Mitigating APIs Lack of Resources & Rate Limiting (15 mins)**
By nature, APIs expose application logic and sensitive data such as Personally Identifiable Information (PII). In addition, new microservices architectures with individual application components have become de facto APIs that significantly expand the attack surface. This course focuses on strategies and solutions used to mitigate APIs' unique vulnerabilities and security risks.
On successful completion of this course, you should have the knowledge and skills required to:

- Define, identify, and create a plan for managing resources
- Understand uncontrolled resource consumption and the associated risks
- Identify the common causes that lead to a lack of resources
- Implement industry best practices to mitigate API's lack of resources and rate-limiting

□ **API 211 – Mitigating APIs Broken Object Level Authorization (15 mins)**
Aligned with OWASP API Security Top 10 and the NIST Cybersecurity Framework; this course is designed for NICE Workforce roles of Software Developer and Secure Software Assessor. Upon successful completion of this course, you should have the knowledge and skills required to identify and resolve object-level authorization issues; be aware of, and mitigate, the most common attack methods for APIs with broken level authorization; and employ industry best practices to prevent and mitigate broken object-level authorization vulnerabilities.
Upon successful completion of this course, you should have the knowledge and skills required to:

- Identify and resolve object-level authorization issues
- Be aware of, and mitigate, the most common attack methods for APIs with broken level authorization
- Employ industry best practices to prevent and mitigate broken object-level authorization vulnerabilities

□ **API 213 – Mitigating APIs Mass Assignment (15 mins)**
Mass Assignment occurs when adversaries exploit unexposed object properties or methods through API parameters. An API might be vulnerable to Mass Assignment if the application directly binds parameters to an internal object's properties without proper validation.
On successful completion of this course, learners should have the knowledge and skills required to:

- Avoid direct use automatic binding
- Define and enforce schemas for input payloads
- Validate and filter input before binding
- Use read-only properties where appropriate

□ **API 214 – Mitigating APIs Improper Asset Management (15 mins)**
In accordance with the OWASP API Security Top 10 2019 Report, APIs tend to expose more endpoints than traditional web applications, making proper and updated documentation highly

important. Proper hosts and deployed API versions inventory also play an important role in mitigating issues such as deprecated API versions and exposed debug endpoints.
On successful completion of this course, learners should have the knowledge and skills required to:

- Understand the importance of proper API asset management
- Gather and maintain an inventory of all vulnerable assets
- Determine API access requirements based on scope, API type, and user roles
- Understand the benefits of using an API gateway
- Create documentation for the information gathered about your organization's APIs

□ AWA 101 – Fundamentals of Application Security **(20 mins)**

This course introduces the fundamentals and primary drivers of application security, including: The CIA "triad"; the importance of meeting regulatory requirements; what motivates hackers, how to manage vulnerabilities, and the key elements of a responsible disclosure programs.
Upon successful completion of this course, you should have the knowledge and skills required to understand:

- Core concepts of application security risk management
- Why developing secure applications matters
- The importance of meeting regulatory compliance requirements
- Anatomy of an application attack and what motivates hackers.
- Common attack scenarios and how to manage vulnerabilities.
- Best practices for developing secure applications.

□ AWA 102 – Secure Software Concepts **(20 mins)**

This course provides a high-level overview of secure software concepts for web applications including application security and security best practices.
Upon successful completion of this course, you should have the knowledge and skills required to:

- Understand the root causes of application weaknesses
- Realize the importance of encrypting data
- Ensure input is properly validated
- Reduce your application's overall attack surface
- How to implement a security strategy based on your organization's risk

□ COD 102 – The Role of Software Security **(10 mins)**

This course introduces you to the overriding importance of software security for your organization, and the potential business consequences of developing and deploying insecure software.
Topics include:

- The difference between software security and network security
- Business imperatives for software security, including the high business costs of security breaches
- Compliance and legal implications of security breaches
- Customer expectations of security
- The increasing threat landscape

□ COD 103 – Creating Software Security Requirements **(10 mins)**

This course discusses the requirements phase of the software development lifecycle and provides software development teams with the knowledge and skill required to gather security requirements for the software that they are designing and implementing.
Topics Include:

- Identifying potential attacks and exploits
- Legal Security Requirements

- Business Requirements
- Customer Security Requirements

□ COD 105 – Secure Software Development **(20 mins)**

This course introduces you to secure development models, standards, and guidelines that provide you with a structure for reducing risk from application security vulnerabilities.
Topics Include:

- The role of Industry standards and security models like OWASP Top 10, CWE SANS Top 25, PA-DSS and many more
- The common criteria for Information Technology Security Education
- Formal methods applied to the analysis of software to ensure that it adheres to industry standards such as Code Analysis, Static Analysis, and Binary Analysis

□ COD 110 – Fundamentals of Secure Mobile Development **(45 mins)**

This course introduces developers to mobile environment threats and risks and presents secure programming principles to mitigate them.
Topics include:

- Common threats to mobile applications: client-side injection, sensitive data handling, network transition, application patching, web-based attacks, phishing, third-party code, location security and privacy and denial of service
- Defensive coding techniques: input validation, output encoding, least privilege, code signing, data protection at rest and in transit, avoiding client side validation, and using platform security capabilities as they apply in mobile environments
- Threat modeling of mobile applications

□ COD 141 – Fundamentals of Database Security **(30 mins)**

In practice, the database represents the goal of many attackers, as this is where the information of value is maintained. However, the functional requirements and security testing often focus on the interaction between a software user and the application, while the handling of data is assumed to be secure. This course describes how to apply authentication and access control to your database and provides an understanding of database privileges and limiting data access. Coverage also includes techniques for protecting the database and methods for securely concealing specific data while providing an introduction to cloud databases and database encryption.

□ COD 152 – Fundamentals of Secure Cloud Development **(20 mins)**

This course introduces developers to the common risks associated with Cloud applications and secure coding best practices to mitigate them.
Topics include:

- Security features of the different series models (IaaS, PaaS, and SaaS)
- How to identify common vulnerabilities and code defensively to avoid them
- Common threats to cloud applications: unauthorized account access, insecure APIs, shared technology, data leakage, and account hijacking
- Complying with regulatory requirements
- The unique security challenges of “Big Data”
- How to apply the Microsoft SDL to cloud applications

□ COD 160 – Fundamentals of Secure Embedded Software Development **(45 mins)**

Embedded devices tend to be linked to other devices via a wide array of technologies and often susceptible to targeted attacks. This course identifies security issues inherent to embedded devices and their deployment environments. You will also learn about the appropriate constraint of functionality from a security standpoint, and techniques to prevent common vulnerabilities.

Topics include:

- Techniques to identify system security and performance requirements
- Developing appropriate security architecture
- Selecting the correct mitigations
- How to develop policies that can ensure the secure operation of your system

□ COD 170 – Identifying Threats to Mainframe COBOL Applications & Data **(20 mins)**

This secure coding course covers the most common security issues that affect the confidentiality, integrity and availability of COBOL programs on mainframes. These include SQL Injection, Command Injection, Integer Overflow, Weak Cryptography, Unencrypted Communications and Race Conditions.

□ COD 201 – Secure C Encrypted Network Communications **(15 mins)**

This course explores secure communications using Transport Layer Security (TLS) and best practices for implementing these within C and C++ applications.
Topics include:

- Key principles of TLS
- Libraries and interfaces for implementing the TLS protocol
- TLS security considerations
- Alternatives to TLS

□ COD 202 – Secure C Runtime Protection **(15 mins)**

This secure coding course covers common run-time protection technologies that can be used to protect an application from attack.
Topics include:

- Run-time protection technologies and how to apply them to your applications
- Stack security cookies, Address Space Layout Randomization (ASLR), and No-eXecute (NX)
- Limitations of run-time protection technologies

□ COD 206 – Creating Secure C++ Code **(15 mins)**

This secure coding course highlights the most useful security features for avoiding memory corruption vulnerabilities in C++.
Additional topics include:

- Standard containers, bounds-checking functions, smart pointers, and standard concurrency features
- How to use object-oriented programming features to define and manipulate data in terms of objects, use range-based loops and native regular expressions

□ COD 207 – Communication Security in C++ **(15 mins)**

This secure coding course focuses on how to protect data in transit using encryption libraries and strong TLS ciphers in C++.
Topics include:

- Important issues about public key certificates including signing and verification
- Using well-trusted encryption libraries and strong TLS cipher suites to protect data in transit
- Protect and verify the integrity of public key certificates

□ COD 214 – Creating Secure GO Applications **(30 mins)**

As organizations continue to migrate to cloud infrastructures; development teams are finding themselves leveraging GO as a tool of choice. Lightweight and quick to compile due to generous libraries and abstractions that make it easier to program concurrent and distributed (read: cloud) applications it offers a slew of benefits from Static compilation with no dependencies, a strong standard library, a full development environment, and the ability to build for multiple architectures with no minimal hassle. This course will provide software developers and DevOps Engineers with working knowledge of fundamental concepts and advanced features of the GO programming language.

Topics Include:

- Identifying and preventing SQL injection attacks
- Understanding cross-site scripting
- Properly configuring browser cookies
- Understanding and preventing session hijacking attacks
- Knowing how to avoid cross-site request forgery vulnerabilities
- Understanding the difference between symmetric and asymmetric cryptography
- Implementing transport layer security
- Working with hashes and key derivation functions

*Indicates that the course is still in production and subject to change

□ COD 216 – Leveraging .NET Framework Code Access Security (CAS) **(60 mins)**

This course explores the foundation of .NET, the CLR's native security infrastructure (Code Access Security), and the ASP.NET security infrastructure.

Topics include:

- Differences between managed and unmanaged code
- Access control functions in Windows
- Code Access Security (CAS) functions in .NET
- Interactions between Windows access control and CAS
- Key aspects of ASP.NET security and understand the Level 2 Security Transparency Model

□ COD 217 – Mitigating .NET Security Threats **(45 mins)**

With a primary focus on .NET secure error handling and secure logging, this course describes secure coding techniques to avoid information disclosure and other vulnerabilities.

Topics include:

- Avoiding dangerous patterns when using CAS
- Avoiding common .NET security pitfalls
- Ensuring application fail safely

□ COD 219 – Creating Secure Code: SAP ABAP Foundations **(90 mins)**

This secure coding course presents best practices and techniques for secure SAP application development using Java and ABAP.

Topics include:

- Key application security principles, vulnerabilities and mitigations
- Validating input in SAP applications
- Protecting data using encryption
- Conducting security code analysis and code reviews

□ COD 241 – Creating Secure Oracle DB Applications **(45 mins)**

This secure coding course introduces database application developers to key industry best practices for data security.

Topics include:

- Secure query construction
- Secure communication and storage

- Creating safe stored procedures to prevent SQL Injection
- How to secure data at rest and data in transit using Oracle Database features

□ COD 242 – Creating Secure SQL Server & Azure SQL DB Applications **(40 mins)**

This secure coding course explores protecting sensitive data and ensuring the integrity of applications running on the Microsoft SQL Server Engine and Azure SQL Database.
Topics include:

- The security function of roles in controlling user and principal access to SQL Server securables
- Exercising fine-grained controls that adhere to the Principle of Least Privilege
- Leveraging the security features of Microsoft's Azure SQL Database to protect sensitive data and ensure the integrity of your applications

□ COD 246 – PCI DSS Requirement 3: Protecting Stored Cardholder Data **(20 mins)**

In this course, you will learn how to ensure compliance with PCI DSS Requirement 3 for protecting cardholder data.
On successful completion of this course, you should have the knowledge and skills required to:

- Identify information that qualifies as sensitive account data
- Minimize exposure of sensitive information
- Ensure the proper handling of sensitive data

□ COD 247 – PCI DSS Requirement 4: Encrypting Transmission of Cardholder Data **(15 mins)**

In this course, you will learn to ensure compliance with PCI DSS Requirement 4 for Encrypting Transmission of Cardholder Data. Coverage includes techniques for spotting missing encryption and using Transport Layer Security (TLS).
On successful completion of this course, you should have the knowledge and skills required to:

- Implement Strong Transport Cryptography; Understand Cipher Suites
- Select Strong Cipher Suites
- Properly use Valid TLS Certificates
- Maintain Certificate and Key Inventory, Wireless Technologies, and End User Messaging

□ COD 248 – PCI DSS Requirement 6: Develop and Maintain Secure Systems and Applications **(15 mins)**

In this course, you will learn to ensure compliance with PCI DSS Requirement 6 for Developing & Maintaining Secure Systems and Applications. Learners will understand the importance of following secure coding best practices, completing yearly developer training, and protecting sensitive data.
On successful completion of this course, you should have the knowledge and skills required to:

- Follow industry standards and best practices
- Consider information security issues throughout the entire software development lifecycle, from initial design to deployment
- Identify and address common software attacks and vulnerabilities
- Define and follow software engineering techniques

□ COD 249 – PCI DSS Requirement 11: Regularly Test Security Systems and Processes **(15 mins)**

In this course, you will learn to ensure compliance with PCI DSS Requirement 11 for Regularly Test Security Systems and Processes. Learners will understand the importance of following

industry-accepted approaches for application and network-layer penetration tests. They will recognize the importance of conducting vulnerability scans to identify and address threats and vulnerabilities as well as documenting the organizations approach for assessing and addressing risks from any exploitable vulnerabilities discovered.
On successful completion of this course, you should have the knowledge and skills required to meeting PCI's Secure Software Framework, including:

- Test Planning
- Vulnerability Scanning
- Penetration Testing
- Intrusion Detection
- Change Detection and Wireless Networks

□ COD 251 – Defending AJAX-Enabled Web Applications **(25 mins)**

This course introduces fundamentals of how to defend AJAX-enabled Web applications, including the difference between regular and AJAX-enabled web applications, AJAX security checks against challenges, and common attacks against AJAX-enabled applications.
Topics include:

- Architectural differences between regular web applications and AJAX-enabled applications
- Identifying threats to AJAX applications: cross-site scripting (XSS), cross-site request forgery (CSRF), and injection attacks
- Implementing countermeasures against attacks: protecting client resources, validating input, protecting web services requests, preventing request forgeries, and securing data access.

□ COD 253 – Creating Secure AWS Cloud Applications **(45 mins)**

This course examines the security vulnerabilities, threats, and mitigations for AWS cloud computing services and provides best practices for securing Web applications by leveraging AWS platform security features.
Topics include:

- AWS security features: Key Management Service (KMS), Hardware Security Module (HSM), Identity and Access Management (IAM), and CloudWatch
- How to leverage security features built into Common Amazon Cloud services such as Simple Storage Service (S3), Elastic Compute Cloud (Amazon EC2), Elastic Block Store (EBS), Amazon Glacier, Relational Database Service (RDS), DynamoDB, Elastic MapReduce (EMR), and Amazon Machine Images (AMI)

□ COD 254 – Creating Secure Azure Applications **(45 mins)**

This course examines key Azure security platforms and services that you can use to improve the security of your applications.
Topics include:

- Security vulnerabilities, threats, and mitigations for Azure cloud computing services
- How to identify common security threats to cloud-based applications
- Secure coding best practices to mitigate threats
- How to leverage built-in Azure features for an extra layer of defense

□ COD 255 – Creating Secure Code: Web API Foundations **(20 mins)**

This secure coding course introduces the fundamentals of secure web services development.
Topics include:

- Common web services threats that put your application at risk
- Impact of web services attacks
- Secure development best practices to protect web services

□ COD 256 – Creating Secure Code: Ruby on Rails Foundations **(90 mins)**

In this course, you will learn about best practices and techniques for secure application development with Ruby on Rails. After completing this course, you will be able to identify and mitigate injection vulnerabilities, such as SQL injection and cross-site scripting, build strong session management into your Rails applications, and prevent other common vulnerabilities, such as cross-site request forgery and direct object access.

Topics include:

- How to identify and mitigate injection vulnerabilities: SQL Injection (SQLi) and cross-site scripting (XSS)
- How to build strong session management into your rails applications
- Preventing common vulnerabilities such as cross-site request forgery (CSRF) and direct object access

□ COD 257 – Creating Secure Python Web Applications **(45 mins)**

In this course, you will learn about best practices and techniques for secure application development with Python. After completing this course, you will be able to understand various types of injection vulnerabilities, including SQL injection and cross-site scripting. You will also be able to understand how to build strong session management into your Python web applications and how to prevent common vulnerabilities, such as cross-site request forgery, direct object access, and others. Finally, you will be able to recognize file system threats to web applications, including vulnerabilities with path traversal, temporary files, and insecure client redirects.

Topics include:

- Types of Injection Vulnerabilities including SQL Injection (SQLi) and Cross-Site Scripting (XSS).
- File system threats to web applications including vulnerabilities with path traversal, temporary files, and insecure client redirects
- How to build strong session management into your python web applications
- Preventing common vulnerabilities such as cross-site request forgery (CSRF), direct object access, and others

□ COD 258 – Creating Secure PHP Web Applications **(30 mins)**

In this course, you will learn important concepts for secure PHP scripting. After completing this course, you will be able to use quotation marks correctly, discuss techniques for handling return codes and exceptions, canonicalize paths to identify the correct files, identify dangerous functions to avoid, apply techniques for preventing or mitigating different injection vulnerabilities, recognize that regular expressions must be handled carefully to avoid DoS attacks, and describe techniques to protect sensitive data in transit.

Topics covered:

- Key defensive coding principles such as proper session management, error handling, authentication, authorization, data storage, and use of encryption
- Avoiding and mitigating vulnerabilities such as SQL Injection (SQLi), Cross-Site Scripting (XSS), File Inclusion, Command Injection, Cross-Site Request Forgery (CSRF) and Null Byte attacks

□ COD 259 – Node.js Threats & Vulnerabilities **(30 mins)**

In this secure coding course, you will learn about system configuration, injection attacks, session management, package management, and the AngularJS framework, all within the context of Node.js security.

Topics include:

- Best practices for Node.js server and system configuration
- Types of injection attacks and mitigation techniques

- Proper settings for session cookie security
- Mitigating cross-site request forgery (CSRF) attacks
- Leveraging popular static analysis tools for Node.js
- Understand why templates and expressions are vulnerable to injection
- Methods, services, elements, and parameters that should not be used with untrusted data
- Best practices for loading templates

□ COD 261 – Threats to Scripts **(30 mins)**

In this secure coding course, you will learn about the impact of incorrect script development or lax security measures.
Topics include:

- Outcomes of vulnerable scripts
- Common scripting vulnerabilities such as SQL Injection (SQLi)
- Security issues related to permissions and privileges
- Impact of different types of resource

□ COD 262 – Fundamentals of Shell and Interpreted Language Security **(30 mins)**

In this secure coding course, you will learn about how shell scripting languages compare with more modern interpreted languages with respect to security features, and defensive coding techniques, and dealing with common differences between platforms that can alter script behavior.
Topics incude:

- Information security principles including least privilege and defense in depth
- The importance of data validation and how to validate using input, array indices, and environment variables
- Using file system operations safely to protect
- Preventing or mitigating cached secret disclosure
- The importance of up-to-date communication security techniques
- Operating system (OS) system portability issues

□ COD 263 – Secure Bash Scripting **(15 mins)**

In this secure coding course, you will learn about the importance of error and exception handling in shell scripts and interpreted languages such as Perl, Python, Bash and Ruby.
Topics covered:

- Techniques for handling errors and exceptions in shell scripts and interpreted languages
- Common syntax pitfalls and dangerous functions to avoid
- Techniques for preventing/mitigating different vulnerabilities including different types of injection

□ COD 264 – Secure Perl Scripting **(15 mins)**

Perceived as being difficult to fix in comparison to other programming languages Perl is commonly known as “the duct-tape of the Internet.” This general-purpose programming language is currently being used for a wide range of tasks as it takes the best features from other languages. In this course, you will learn about best practices for secure scripting in Perl, features of Perl’s taint mode, handling errors in Perl, protecting files, preventing format string and injection vulnerabilities, using regular expressions carefully, and protecting sensitive data in transit with Transport Layer Security (TLS).

□ COD 265 – Secure Python Scripting **(15 mins)**

In this secure coding course, you will learn important concepts for secure Python scripting including techniques for error and exception handling.
Topics Covered:

- Avoiding uncontrolled format string vulnerabilities
- Defending against Regular Expression Denial of Service (DoS) attacks
- Protecting sensitive data in transit
- Techniques for preventing/mitigating different vulnerabilities including different types of injection

□ COD 266 – Secure Ruby Scripting **(15 mins)**

In this secure coding course, you will learn important concepts for secure Ruby scripting, techniques for preventing/mitigating different vulnerabilities including different types of injection and protecting sensitive data in transit.
Topics covered:

- Validating command-line parameters
- Using quotation marks correctly
- Using unmask to set default file permissions
- Protecting files and canonicalizing paths
- Defending against Regular Expression Denial of Service (DoS) attacks

□ COD 267 – Securing Python Microservices **(30 mins)**

Microservices have become widely popular, replacing complicated XML-based schemas and service-oriented architectures (SOA) because of the ability to create separate, well-defined, individual components within a system. By leveraging python microservices, complex applications can be broken down into these components to ease further development and deployment. This course will provide cloud developers, python developers, and software architects with a working knowledge of possible attacks, how to secure interaction between services and an understanding of how to implement basic principles to ensure the security of python microservices.
Topics Include:

- Techniques for handling return codes and exceptions
- Canonicalizing paths to identify the correct files
- Identifying dangerous functions
- Applying techniques for mitigating injection vulnerabilities
- How to securely handle regular expressions
- How to protect sensitive data

□ COD 270 – Creating Secure COBOL & Mainframe Applications **(25 mins)**

This secure coding course covers countermeasures for security vulnerabilities on mainframe systems such as input validation, parameterized APIs, strong cryptography, and memory management issues.
Topics include:

- Identifying vulnerabilities and threats to mainframe applications and data
- Mitigating SQL injection threats using safe prepared statements and parameterized APIs
- Validating all input
- Using exec* functions instead of system functions to mitigate the risk of command injection
- Using key derivation functions to protect stored password
- Encrypting sensitive data at rest using AES-256
- Protecting sensitive data in transit with TLS
- Preventing deadlocks by using the ENQ and DEQ commands

- Avoiding manual memory management in order to prevent buffer overflow conditions

□ COD 283 – Java Cryptography **(45 mins)**

This secure coding course explores the key concepts of public key cryptography and teaches you how to use the Java keytool command-line utility for creating and managing keys and keystores.
Topics include:

- How public and private key pairs work together to encrypt and decrypt data for secure transfer and to create and verify digital signature
- Generating secure encryption keys and identifying related issues such as pseudo random number generators (PRNGs), key derivation functions, and initialization vectors
- Selecting an appropriate symmetric encryption algorithm, cipher mode, and authenticated encryption mode

□ COD 284 – Secure Java Coding **(30 mins)**

In this course, you will learn about secure Java coding practices, including techniques for avoiding Denial of Service (DoS) and regular expression DoS attacks, and guidelines for secure error handling and logging. You will also become familiar with the dangers of unreleased resources, null references, and XML external entity (XXE) attacks.
Topics include:

- Denial of Service and designing your application to handle or avoid such situations
- Guidelines for secure error handling and logging
- Identify the dangers of unreleased resources, null references, and XML external entity attacks

□ COD 285 – Developing Secure Angular Applications **(30 mins)**

Widely adopted amongst the software development community because of the versatility it provides, securing angular applications comes with a steep learning curve. While component-based architecture is one of the key benefits of using angular, managing components can be complicated. This course is designed to develop the skills required to design, build, and maintain secure Angular applications following software assurance best practices.
Upon successful completion of this course, learners will have the knowledge and skills required to meet Secure Angular.js compliance requirements, including:

- Securing AngularJS templates to help mitigate threats from expression Injection and dynamically loading templates from untrusted sources
- Ensuring that both the server and the client cooperate to eliminate these threats and potential security issues that need to be blocked
- Implementing Content Security Policies and secure routing

□ COD 286 – Creating Secure React User Interfaces **(10 mins)**

This JavaScript library has become a popular choice in the market because of its ability to help solve web development challenges. The framework makes it painless to create interactive user interfaces, design simple views, and reactively update to changes. This course is designed to develop the skills required to securely build user interfaces using multiple components and implement best practices to avoid common attacks.
Upon successful completion of this course, learners will have the knowledge and skills required to meet Secure React.js User Interfaces compliance requirements, including:

- Creating secure React components
- Avoiding vulnerable third-party React component libraries
- Preventing React component injection attacks
- Using and serializing JSON

□ COD 287 – Java Application Server Hardening (UPDATED) **(20 mins)**

This operations and maintenance course introduces NIST NICE roles Network Operations Specialist, Systems Security Analyst, and System Administrator to best practices for server hardening. Its objectives align with the OWASP Top 10 and Application Security Verification Standard (ASVS), and the NIST Cybersecurity Framework (CSF).

Upon successful completion of this course, you should have the knowledge and skills required to:

- Apply best practices for hardening Java application servers
- Reduce the attack surface
- Maintain currency of server software and dependencies protecting network connections

□ COD 288 – Java Public Key Cryptography (NEW) **(20 mins)**

Public key cryptography is a critical framework for secure communications and data transfer in Java applications. It operates on the principles of asymmetric cryptography, which involves producing a bound [pair of keys: one public and one private. The private key remains confidential, like a password, while the public key is made available to anyone, much like an email address. This dual-key system serves two purposes; the first is encryption and the second is for digital signatures. This course provides learners with the knowledge and skills to apply best practices for using Public Key Cryptography in Java.

On successful completion of this course, learners should have the knowledge and skills required to:

- Use Java Public Key Cryptography
- Secure Public Keys using Digital Signatures and CAs
- Limit the use of Self-Signed Certificates using Certificate Chains
- Store and manage keys securely using Keystores, signing Java Classes

□ COD 301 – Secure C Buffer Overflow Mitigations **(45 mins)**

This course focuses on C-language buffers. Upon completion of this course you will learn good memory management techniques and coding best practices to help you avoid buffer & integer overflows, format string vulnerabilities, and race conditions.

Topics include:

- Mitigating buffer overflows and race conditions
- Preventing memory management, format string, injection and integer overflow vulnerabilities
- Protecting data in memory

□ COD 302 – Secure C Memory Management **(20 mins)**

This secure coding course focuses on memory manipulation and allocation techniques for C-language software development.

Topics include:

- Key concepts of dynamic memory management
  Common mistakes that lead to Out-of-Range Memory Access
- Best practices to mitigate memory management vulnerabilities
- How to ensure that “freed” or “deleted” data in memory is no longer accessible

□ COD 303 – Common C Vulnerabilities & Attacks **(20 mins)**

In this secure coding course, you will review common C application vulnerabilities, how they manifest in code; as well as techniques and libraries that you can use to mitigate the risk of attack. After completing this course, you will be able to mitigate risk from the following vulnerabilities:

- Format string attacks
- Integer overflows
- Path Traversal issues

- Command injection
- SQL injection

□ COD 307 – Protecting Data in C++ **(25 mins)**

This secure coding course presents key concepts of public key cryptography, the risks of improper encryption, and defensive coding techniques to protect sensitive data.
Topics include:

- Generating strong encryption keys and identifying related issues such as pseudo random number generators (PRNGs), key derivation algorithms, and initialization vectors
- Selecting an appropriate symmetric encryption algorithm, cipher mode, and authenticated encryption mode
- Common libraries that support symmetric cryptography
- How public and private key pairs work together both to encrypt and decrypt data for secure transfer and to create and verify digital signatures
- Best practices to mitigate memory exposure vulnerabilities

□ COD 308 – Common ASP.NET MVC Vulnerabilities and Attacks **(45 mins)**

This course provides an overview of code security issues that affect ASP.NET MVC applications. You will also understand how other vulnerabilities can be mitigated with careful and complete input validation.
After completing this course, you will be able to understand model validation and its strengths and weaknesses, understand and prevent unique attacks, such as under-posting and over-posting, and implement protective measures against SQL injection, cross-site scripting, cross-site request forgery, and malicious URL redirects.

□ COD 309 – Securing ASP.NET MVC Applications **(30 mins)**

This course teaches the fundamentals of authentication and authorization in ASP.NET Web API, and the roles they play in the OWIN pipeline.
After completing this course, you will understand:

- Web API pipeline and where each component sits on that path
- Authentication and authorization filters and the role of each in your Web API application
- Different authentication options and how to implement them in your application
- The importance of secure communication and the use of Transport Layer Security (TLS) to create secure data exchange tunnels.

□ COD 315 – Preventing Vulnerabilities in iOS Code in Swift **(20 mins)**

In this secure coding course, you will learn how to code defensively to prevent iOS security vulnerabilities.
Topics Include:

- Mitigation approaches and Implementing Secure Coding Best Practices
- How to leverage iOS and Swift security services to mitigate threats
- Pros and Cons of Biometrics such as Touch ID and Face ID

□ COD 316 – Creating Secure iOS Code in Objective C **(30 mins)**

This secure coding course describes techniques for creating secure iOS applications.
Topics include:

- Common vulnerabilities such as exposure of authentication credentials, sensitive data, and other secrets; custom URL scheme abuse; and XML eXternal Entity (XXE) Injection

- Techniques for mitigating vulnerabilities including protecting data at rest with the Data Protection and Common Crypto APIs, mitigating sensitive data exposure in background snapshots, preventing custom URL scheme abuse, and mitigating XXE Injection

□ COD 317 – Protecting Data on iOS in Swift **(20 mins)**

In this secure coding course, you will learn how to code defensively to protect data on iOS
Topics Include:

- Protecting data in transit and at Rest
- App Transport Security (ATS and default settings
- Use of Valid Certificates and Certificate Pinning
- Using URL Loading System to establish Network Connections
- iOS Cryptography Framework and Data Protection Features

□ COD 318 – Protecting Data on Android in Java **(20 mins)**

In this secure coding course, you will learn how to protect data on Android applications using Java.
Topics include:

- Protecting Data in transit using Transport Layer Security (TLS)
- Protecting Data at rest using Symmetric Key Encryption
- Using Android KeyStor to Protect Data
- Dangers of External Storage on Android

□ COD 319 – Preventing Vulnerabilities in Android Code in Java **(20 mins)**

In this secure coding course, you will learn to meet Android security quality standards using Java.
Topics include:

- Restricting access to Interprocess Communications and shared data
- Applying the Principle of Least Privilege and deferring permissions
- Avoiding disclosure of sensitive data
- Reducing attack vectors for Cross-Site Scripting (XSS)
- Keeping all libraries and dependencies current

□ COD 321 – Protecting C# from Integer Overflows & Canonicalization **(30 mins)**

This secure coding course describes methods that will produce secure C# applications.
Topics include:

- Common security vulnerabilities such as Canonicalization Issues and Integer Overflows
- Unique features of C# and the .NET Framework that can be used to mitigate them
- Understand where and when canonicalization issues and integer overflows are likely to occur
- Avoiding common pitfalls

□ COD 322 – Protecting C# from SQL Injection **(8 mins)**

This secure coding course presents SQL Injection vulnerabilities and the features of the .NET Framework that can be used to mitigate them.
Topics include:

- Where and when SQL injection is likely to occur
- Avoiding common pitfalls when defending against SQL injection
- Defense-in-Depth Strategies and best practices for mitigating injection vulnerabilities

□ COD 323 – Using Encryption with C# **(20 mins)**

This secure coding course describes techniques to protect data both in transit and at rest in C# applications using strong cryptography.
Topics include:

- How to protect data using the Data Protection API (DPAPI)
- Avoiding common cryptographic pitfalls
- Protecting sensitive data in transit
- Alternatives to Transport Layer Security (TLS)

□ COD 324 – Protecting C# from XML Injection **(8 mins)**

This secure coding course presents XML Injection vulnerabilities and the features of the .NET Framework that can be used to mitigate them.
Topics include:

- Where and when XML injection is likely to occur
- Avoiding common pitfalls when defending against XML injection
- Defense-in-Depth Strategies and best practices for mitigating injection vulnerabilities

□ COD 352 – Creating Secure JavaScript and jQuery Code **(45 mins)**

In this secure coding course, you will learn about common client-side vulnerabilities and threats to jQuery applications, and techniques for mitigating them.
Additional topics include:

- How to implement new HTML5 security features to secure jQuery applications
- Best practices to secure local storage and implement Transport Layer Security

□ COD 361 – HTML5 Secure Threats **(15 mins)**

In this secure coding course, you will learn about security risks introduced by HTML5.
Additional topics include:

- Threats to HTML5 such as cross-site scripting (XSS), cross-site request forgery (CSRF), clickjacking, and threats to user privacy
- Secure coding techniques to mitigating HTML5 threats

□ COD 362 – HTML5 Built in Security Features **(20 mins)**

This secure coding course describes important HTML5 security features and how to leverage them to produce more robust applications.
Topics include:

- Implementing Same-Origin Policy, Content Security Policy, Cross-Origin Resource Sharing, and IFrame Sandboxing
- Understanding the limitations of Same-Origin Policy
- Employing best practices to avoid common attacks on HTML5 applications

□ COD 363 – Securing HTML5 Data **(20 mins)**

In this course, you will learn about new features that raise security issues in HTML5 forms, security issues surrounding local data storage, best practices for HTML5 connectivity with the WebSocket API and Server-ent Events, and best practices for the Web Workers, History, Geolocation, and Drag and Drop APIs.

□ COD 364 – Securing HTML5 Connectivity **(20 mins)**

In this course, you will learn about best practices for securing connections used by applications that leverage HTML5.

□ COD 366 – Creating Secure Kotlin Applications **(20 mins)**

As a prime option for building android applications because of its interoperability with java code, maintainability, reliability, and ability to boost team efficiency, Kotlin is being widely adopted but comes with its own set of challenges as does any technology. This course is designed to ensure learners avoid common mistakes and pitfalls as they leverage vital features and build secure mobile applications using this general-purpose programming language.

Upon successful completion of this course, learners will have the knowledge and skills required to meet privacy compliance requirements, including:

- Enforcing secure communication by safeguarding the data that you exchange between your app and other apps, or between your app and a website, thereby improving your app's stability and protecting the data that you send and receive.
- Using intents to defer permissions
- Storing all private user data within the device's internal storage (which is sandboxed per app)
- Ensuring the device deletes all files when the user uninstalls an app

□ COD 380 – Preventing SQL Injection in Java **(8 mins)**

This secure coding course describes ways to remediate and prevent SQL Injection (SQLi) vulnerabilities in your Java application.

Topics Include:

- Identifying data types that require encryption
- Best practices for encryption methods
- Avoiding common encryption errors

□ COD 381 – Preventing Path Traversal Attacks in Java **(8 mins)**

This secure coding course describes ways to mitigate security risks from Path Traversal Attacks in your Java application.

Topics Include:

- Identifying Path Traversal Attacks and understanding how they work
- Normalizing, canonicalizing, and validating file paths
- Implementing countermeasures to prevent Path Traversal Attacks

□ COD 382 – Protecting Data in Java **(30 mins)**

This course discusses protecting data at rest and in transit in Java applications. Several code examples are provided to illustrate key concepts. After completing this course, you will be able to protect data at rest appropriate cryptographic techniques and protect data in transit with appropriate cryptographic techniques.

□ COD 383 – Protecting Java Backend Services (UPDATED) **(30 mins)**

Backends are designed for applications that need faster performance, large amounts of addressable memory, and continuous or long-running background processes. The versatility of Java enables developers to design and deliver the right business solutions however their efficiency requires distinctive experiencer and great expertise. This course aims to provide software developers and DevOps Engineers with the next level understanding of best practices for developing back-end frameworks using Java while developing skills necessary to handle user input and build secure systems.

Upon successful completion of this course, you should have the knowledge and skills required to:

- Recognize what causes common vulnerabilities
- Apply techniques to prevent common vulnerabilities
- Explain and recognize the benefits of authorization in backend services
- Leverage technologies like OAuth2 JWT, JAAS API, and Spring Security
- Protect sensitive data in transit with TLS

□ COD 384 – Protecting Java from Information Disclosure **(8 mins)**

This secure coding course describes ways to identify and prevent Information disclosure in your Java application.
Topics Include:

- Identifying common Java information disclosure issues
- Protecting Java applications through improved error messaging
- Best practices for preventing information disclosure
- Audit error handling for information disclosure vulnerability

□ COD 385 – Preventing Race Conditions in Java Code **(8 mins)**

This secure coding course describes ways to identify and prevent race conditions in your Java application.
Topics Include:

- Common Java race condition issues
- Security risks introduced by race conditions
- Secure protection of temp files
- Best practices for preventing race condition issues

□ COD 386 – Preventing Integer Overflows in Java Code **(8 mins)**

This secure coding course describes ways to write code to identify and mitigate risks from integer overflows.
Topics Include:

- Common Integer Overflow security risks and prevention methods
- Precondition Testing, Upcasting, and BigInteger Objects
- Google Guava
- Common Integer Overflow Pitfalls

□ DES 207 – Mitigating OWASP API Security Top 10 **(15 mins)**

The OWASP API Security Top 10 defines the most critical Application Programming Interface (API) security risks and vulnerabilities, recognizing the changing security landscape as organizations embrace digital transformation. This API security course provides an understanding of API-specific issues that should be on an organization's security radar with a focus on strategies to mitigate these unique vulnerabilities and security risks of APIs based on the crucial role they play in application architecture.
Upon successful completion of this course, learners will understand:

- The purpose and scope of the OWASP API Security Top 10
- The unique risks and attack vectors when exposing APIs to the public
- Best practices for secure coding to mitigate common API vulnerabilities
- How to eliminate or mitigate these vulnerabilities

□ DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing **(15 mins)**

The CSA Top 11 Threats to Cloud Computing provides guidelines on what secure practices organizations should focus on when planning and establishing cloud environments. Naturally, as more and more organizations deploy cloud-based solutions, new security risks and challenges are introduced. This course provides an understanding of cloud-based security threats organizations should consider and focuses on unique mitigation strategies that should be explored when using a cloud-based infrastructure.
Upon successful completion of this course, learners will understand:

- The purpose and scope of the CSA Top 11 Threats to Cloud Computing
- The unique risks and attack vectors of cloud environments

- Best practices for secure coding to mitigate common cloud vulnerabilities
- How to eliminate or mitigate these vulnerabilities

□ DES 232 – Mitigating OWASP 2021 Injection **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A03:2021 Injection, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will understand how to:

- Keep data separate from commands and queries
- Implement multi-factor authentication
- Require weak-password checks
- Limit login attempts

□ DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A07:2021 Identification and Authentication Failures, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will have the knowledge and skills required to:

- Define and identify the common types of identification and authentication errors
- Mitigate brute force and password spraying attacks while implementing techniques to limit exposure to those attacks
- Enact strong password policies, using industry best practices, to achieve optimal password strength, proper storage, and secure processes

□ DES 234 – Mitigating OWASP 2021 Cryptographic Failures **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A02:2021 Cryptographic Failures, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will understand:

- The three main goals of cryptography
- How to secure data at rest and data in transit
- How to identify common cryptographic failures
- How to mitigate sensitive data exposure

□ DES 235 – Mitigating OWASP 2021 Insecure Design **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A04:2021 Insecure Design, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will understand:

- How the secure software development lifecycle improves software security
- How to use secure design patterns
- The importance of threat modeling
- How to prevent excessive resource consumption

□ DES 236 – Mitigating OWASP 2021 Broken Access Control **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A01:2021 Broken Access Control, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will understand:

- Implement access control policies
- Assess the effectiveness of current access controls
- Employ secure coding practices to ensure users cannot act outside intended permissions

□ DES 237 – Mitigating OWASP 2021 Security Misconfiguration **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A05:2021 Security Misconfiguration, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will have the knowledge and skills required to:

- Define and identify security misconfiguration vulnerabilities
- Mitigate misconfiguration vulnerabilities using attack surface and defense-in-depth strategies
- Secure operating systems, web servers, and databases against common misconfiguration attacks

□ DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF) **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A10:2021 Server-Side Request Forgery (SSRF), as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will have the knowledge and skills required to:

- Define and identify Server-Side Request Forgery (SSRF) vulnerabilities
- Recognize the conditions that lead to SSRF vulnerabilities in web applications
- Understand the common attacks and techniques used to exploit SSRF vulnerabilities

□ DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A08:2021 Software and Data Integrity Failures, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will have the knowledge and skills required to:

- Define and identify software and data integrity failures
- Understand the risks of serialization, deserialization, and mitigation techniques
- Protect stored data using message authentication codes and digital signatures
- Implement best practices to limit software and data integrity failures

□ DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A06:2021 Vulnerable and Outdated Components, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will understand how to:

- Monitor applications for out of date components
- Triage and apply updates for known vulnerabilities
- Apply secure coding practices over the lifetime of an application

□ DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures **(12 mins)**

In this course, you will learn how to mitigate the risks associated with A09:2021 Security Logging and Monitoring Failures, as defined by the Open Web Application Security Project (OWASP).
After completing this course, you will understand how to:

- Ensure all login, access failures, and input validation failures are logged
- Implement sufficient user context to identify suspicious behavior
- Allow sufficient time so malicious accounts can be tracked for forensic analysis
- Apply best practices for secure application logging

□ DES 271 – OWASP M1: Mitigating Improper Platform Usage **(12 mins)**

In this course, you will learn how to mitigate the risks associated with Improper Platform Usage which might include Android intents, platform permissions, misuse of TouchID, the keychain, or some other security control that is part of the mobile operating system.

After completing this course, you will be able to:

- Identify the most common security flaws in mobile apps related to improper platform usage
- Understand how an attacker might exploit such vulnerabilities in your software
- Eliminate or mitigate exposure to these common security threats

□ DES 272 – OWASP M2: Mitigating Insecure Data Storage **(12 mins)**

In this course, you will learn how to mitigate the risks associated with Insecure Data Storage which includes threat agents such as an adversary that has attained a lost/stolen mobile device; malware or another repackaged app acting on the adversary's behalf that executes on the mobile device.

After completing this course, you will be able to:

- Identify the most common security flaws in mobile apps related to insecure data storage
- Understand how an attacker might exploit such vulnerabilities in your software
- Eliminate or mitigate exposure to these common security threats

□ DES 273 – OWASP M3: Mitigating Insecure Communication **(12 mins)**

In this course, you will learn how to mitigate the risks associated with Insecure Communication which might include threat agents such as an adversary that shares local network (compromised or monitored Wi-Fi); carrier or network devices (routers, cell towers, proxy's, etc); or malware on your mobile device.

After completing this course, you will be able to:

- Identify the most common security flaws in mobile apps related to insecure communication
- Understand how an attacker might exploit such vulnerabilities in your software
- Eliminate or mitigate exposure to these common security threats

□ DES 274 – OWASP M4: Mitigating Insecure Authentication **(12 mins)**

In this course, you will learn how to mitigate the risks associated with Insecure Authentication which is typically exploited through automated attacks that use available or custom-built tools.

After completing this course, you will be able to:

- Identify the most common security flaws in mobile apps related to Insecure Authentication
- Understand how an attacker might exploit such vulnerabilities in your software
- Eliminate or mitigate exposure to these common security threats

□ DES 275 – OWASP M5: Mitigating Insufficient Cryptography **(12 mins)**

In this course, you will learn how to mitigate the risks associated with Insufficient Cryptography which includes threat agents such as anyone with physical access to data that has been encrypted improperly, or mobile malware acting on an adversary's behalf.

After completing this course, you will be able to:

- Identify the most common security flaws in mobile apps related to insufficient cryptography
- Understand how an attacker might exploit such vulnerabilities in your software
- Eliminate or mitigate exposure to these common security threats

□ DES 276 – OWASP M6: Mitigating Insecure Authorization **(12 mins)**

In this course, you will learn how to mitigate the risks associated with Insecure Authorization which allows an adversary to execute functionality they should not be entitled to using an authenticated but lower-privilege user of the mobile app.

After completing this course, you will be able to:

- Identify best practices for implementing secure authorization for Mobile Internet of Things
- How to mitigate the threat of Insecure Authorization
- Identify and mitigate Insecure Direct Object Reference (IDOR) vulnerabilities

□ DES 277 – OWASP M7: Mitigating Client Code Quality **(12 mins)**

In this course, you will learn how to mitigate the risks associated with poor code quality, including threat agents such as entities that can pass untrusted inputs to method calls made within mobile code.

After completing this course, you will be able to:

- Identify Uncontrolled Format String and Classic Buffer Overflow
- Recognize their potential impact
- Apply coding best practices to avoid them
- Find these weaknesses in your mobile application's source code
- Test your application to detect them

□ DES 278 – OWASP M8: Mitigating Code Tampering **(12 mins)**

In this course, you will learn how to mitigate the risks associated with code tampering. Typically, an attacker will exploit code modification via malicious forms of the apps hosted in third-party app stores. The attacker may also trick the user into installing the app via phishing attacks.

After completing this course, you will be able to:

- Identify code tampering vulnerabilities
- Defend against code tampering attacks

□ DES 279 – OWASP M9: Mitigating Reverse Engineering **(12 mins)**

In this course, you will learn how to mitigate risks associated with reverse engineering in which an attacker will typically download the targeted app from an app store and analyze it within their local environment using a suite of different tools.

After completing this course, you will be able to:

- Describe what kinds of knowledge reverse engineering may reveal to an attacker
- List mitigation techniques for reverse engineering

□ DES 280 – OWASP M10: Mitigating Extraneous Functionality **(12 mins)**

In this course, you will learn how to mitigate the risks associated with extraneous functionality. Typically, an attacker seeks to understand extraneous functionality within a mobile app in order to discover hidden functionality in backend systems. The attacker will typically exploit extraneous functionality directly from their own systems without any involvement by end-users.

After completing this course, you will be able to:

- Identify Extraneous Functionality
- Understand how an attacker might exploit this vulnerability in your software
- Mitigate exposure to this threat

□ DES 281 – OWASP IoT1: Mitigating Weak, Guessable or Hardcoded Passwords **(12 mins)**

In this course, you will learn how to mitigate the risks associated with the use of easily brute-forced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.

When you have completed this course, you will be able to:

- Identify best practices for implementing secure authentication for the Internet of Things
- Identify and mitigate password weaknesses in your applications

□ DES 282 – OWASP IoT2: Mitigating Insecure Network Services **(12 mins)**

In this course, you will learn how to mitigate the risks associated with unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control.

After you have completed this course, you will be able to:

- Identify best practices to protect network services on IoT devices, including:
  - Only open necessary ports
  - Do not overexpose ports
  - Block unusual traffic
  - Mitigate DoS vulnerabilities
  - Mitigate memory corruption vulnerabilities|Disable outdated protocols

□ DES 283 – OWASP IoT3: Mitigating Insecure Ecosystem Interfaces **(12 mins)**

In this course, you will learn how to mitigate the risks associated with insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.

After completing this course, you will be able to:

- Identify common threats to IoT web interfaces
- Apply best practices to mitigate these threats

□ DES 284 – OWASP IoT4: Mitigating Lack of Secure Update Mechanism **(12 mins)**

In this course, you will learn how to mitigate the risks associated with a lack of ability to securely update the device. This includes lack of firmware validation on a device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.

After you have completed this course, you will be able to:

- List the steps of a typical update process
- Describe how to protect update connections
- Explain how to protect the update server
- List the steps to securely sign and verify an update
- Evaluate whether Secure Boot is necessary for your device at this time
- Identify types of sensitive data that should not be included in updates
- Securely implement transport encryption for an Internet of Things (IoT) system

□ DES 285 – OWASP IoT5: Mitigating Use of Insecure or Outdated Components **(12 mins)**

In this course, you will learn how to mitigate the risks associated with the use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms and the use of third-party software or hardware components from a compromised supply chain. After you have completed this course, you will be able to identify and mitigate threats posed by insecure and outdated components.

□ DES 286 – OWASP IoT6: Mitigating Insufficient Privacy Protection **(12 mins)**

In this course, you will learn how to mitigate the risks associated with a user's personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.

After completing this course, you will learn to:

- Identify threats to personal information
- Identify ways to protect personal information

□ **DES 287 – OWASP IoT7: Mitigating Insecure Data Transfer and Storage (12 mins)**

In this course, you will learn how to mitigate the risks associated with a lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing.

After completing this course, you will be able to:

- Identify missing encryption
- Recognize the potential impact of this security defect
- Apply best practices to prevent insecure data transfer and storage

□ **DES 288 – OWASP IoT8: Mitigating Lack of Device Management (12 mins)**

In this course, you will learn how to mitigate the risks associated with a lack of ability to securely update the device. This includes lack of firmware validation on a device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanism.

After completing this course, you will be able to:

- Monitor and Track Assets
- Monitor, Handle and Retain Information
- Monitor and Control System and Network Access

□ **DES 289 – OWASP IoT9: Mitigating Insecure Default Settings (12 mins)**

In this course, you will learn how to mitigate the risks associated with devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations. After you have completed this course, you will be able to understand insecure default settings and their mitigation techniques.

□ **DES 290 – OWASP IoT10 Mitigating Lack of Physical Hardening (12 mins)**

In this course, you will learn how to mitigate the risks associated with a lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device.

After completing this course, you will be able to:

- Understand fail-safe defaults
- Use best practices for hardening

Security Principles

□ **ENG 110 – Essential Account Management Security (15 mins)**

This infrastructure security course provides essential guidance on implementing specific account management security controls at the hardware and software level to facilitate compliance with applicable regulatory requirements.

Topics include:

- How to define and control network access
- Creating a separation of duties policy
- Building and managing segregation of resources strategies
- Monitoring system access
- Using digital certificates for authentication

□ ENG 111 – Essential Session Management Security **(15 mins)**

This infrastructure security course provides guidance to system designers and developers on how to implement session management controls at the software level. These techniques enhance security of web applications and facilitates compliance with applicable regulatory requirements.
Topics include:

- Securing session identifiers
- Implementing Transport Layer Security (TLS) so sensitive data is always transmitted over secure channels
- Ensuring client browsers send cookies over HTTPS connections

□ ENG 112 – Essential Access Control for Mobile Devices **(15 mins)**

This infrastructure security course teaches designers and developers how to implement software-level access controls on mobile devices to mitigate threats, protect privacy, and comply with applicable regulatory requirements.
Topics include:

- Identifying threats to mobile devices
- The importance of protecting user privacy and confidentiality
- Methods for encrypting data at rest and data in transit
- Implementing application code signing to ensure software integrity

□ ENG 113 – Essential Secure Configuration Management **(15 mins)**

This infrastructure security course trains program managers, system designers, and developers on proper security practices for defining and implementing IT system configuration management.
Topics include:

- Key configuration practices and configuration change control
- Security impact analysis
- Access restrictions for change
- Principle of least functionality
- Information system component inventory

□ ENG 114 – Essential Risk Assessment **(15 mins)**

This infrastructure security course provides essential guidance on information system risk assessment techniques. Individuals responsible for information systems, IT security, risk management, or oversight responsibilities will find this course valuable. It teaches how to define and manage the purpose, scope, roles, and coordination among organizational entities to help ensure appropriate risk assessment and compliance with applicable regulatory requirements.
Topics include:

- Security categorization
- Risk assessment
- Vulnerability scanning
- The system development lifecycle
- Security engineering principles
- Developer security testing and evaluation
- Development process, standards, and tools
- Developer security architecture and design
- Component authenticity

□ ENG 115 – Essential System & Information Integrity **(15 mins)**

This infrastructure security course provides essential guidance to program managers, system designers and developers on how to identify systems affected by software flaws, assess potential vulnerabilities resulting from those flaws, and report this information to designated organizational personnel.
Topics include:

- Flaw remediation
- Malicious code protection
- Information system monitoring
- Software, firmware, and information integrity
- Information input validation
- Error handling
- Information handling and retention
- Information output filtering
- Memory protection

□ ENG 116 – Essential Security Planning Policy & Procedures **(15 mins)**

This infrastructure security course provides training to individuals with information security implementation and operational responsibilities for developing and disseminating an organization-wide security policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance mapping.
Topics include:

- Establishing rules of behavior
- Security concept of operations
- Personnel security policies and procedures
- Position risk designations
- Personnel screening
- Access agreements

□ ENG 117 – Essential Information Security Program Planning **(15 mins)**

This infrastructure security course provides essential guidance to individuals with information security implementation and operational responsibilities on how to build and communicate an information security program plan to facilitate compliance with applicable regulatory requirements.
Topics include:

- Identifying information security resources
- Performing an information system inventory
- Creating a critical infrastructure plan
- Risk management strategy
- Insider threat program
- Training and developing contacts with security groups and associations

□ ENG 118 – Essential Incident Response **(15 mins)**

This infrastructure security course teaches incident response policy development and the associated controls to help ensure appropriate communication and action throughout your organization.
Topics include:

- Incident response testing
- Incident handling
- Incident monitoring
- Incident reporting

□ ENG 119 – Essential Security Audit & Accountability **(15 mins)**

This infrastructure security course trains information system owners, system administrators, and information system security officers on how to build and communicate effective audit policies and controls.
Topics include:

- Documenting security audits

- Implementing audit controls
- Using audit tools
- Generating audit reports

□ ENG 120 – Essential Security Assessment & Authorization **(15 mins)**

This infrastructure security course provides guidance for developing and implementing personnel security policies and associated controls to help ensure appropriate screening, on-boarding, and off-boarding of staff.
Topics include:

- Position risk designation
- Personnel screening and termination
- Personnel transfer and access agreement

□ ENG 121 – Essential Identification & Authentication **(15 mins)**

This infrastructure security course teaches those responsible for information security how to develop identification and authentication policy and controls. The course spans personnel, devices, and information systems.
Topics include:

- Identification and authentication of users inside and outside your organization
- Device identification and authentication
- Identifier management
- Authenticator management and feedback
- Cryptographic module authentication
- Service identification and authentication
- Adaptive identification and authentication
- Processes for re-authentication

□ ENG 122 – Essential Physical & Environmental Protection **(15 mins)**

This infrastructure security course educates those responsible for developing physical and environmental protection policies how to create effective controls and comply with applicable regulatory requirements.
Topics include:

- Physical access authorizations and control
- Access control for transmission medium and output devices
- Monitoring physical access
- Information leakage, asset monitoring and tracking

□ ENG 123 – Essential Security Engineering Principles **(15 mins)**

This infrastructure security course provides direction to program managers, system designers, developers, information security engineers, and systems integrators responsible for new information systems development or systems undergoing major upgrades.
Topics include:

- System development life cycle
- Developer security testing and evaluation
- Development process, standards, and tools
- Developer security architecture
- Design and component authenticity

□ ENG 124 – Essential Application Protection **(15 mins)**

This infrastructure security course imparts guidance to system designers and developers on implementing specific security controls at the software level to protect applications and comply with applicable regulatory requirements.

Topics include:

- Implementing defense-in-depth
- Separation of system and user functionality
- Securing components
- Validating input
- Encoding output

□ ENG 125 – Essential Data Protection **(15 mins)**

This infrastructure security course delivers training to personnel in information systems, information security, systems design, software development, and IT operations on essential data security techniques. Focus is primarily on cryptographic controls at the information systems level and compliance with applicable regulatory requirements.

Topics include:

- Asymmetric key algorithms
- Using hash functions to protect data integrity
- Proper password storage for authentication purposes
- Encrypting file transfers and downloads
- Adding salt values before hashing
- Using Certificate Authorities

□ ENG 126 – Essential Security Maintenance Policies **(15 mins)**

This infrastructure security course offers guidance to individuals with information security implementation and operational responsibilities for developing system maintenance procedures and controls.

Topics include:

- Controlled maintenance
- Maintenance tools
- Non-local maintenance
- Timely maintenance

□ ENG 127 – Essential Media Protection **(15 mins)**

This infrastructure security course describes the development and dissemination of an organization-wide information media protection policy that addresses scope, roles, responsibilities, and coordination among organizational entities to facilitate compliance with applicable regulatory requirements.

Topics include:

- Best practices for media protection
- Controls for marking, storage, and transport of media
- Media sanitization and downgrading

Security Testing

□ ATK 201 – Using the MITRE ATT&CK Framework **(15 mins)**

The MiTRE ATT& CK Framework is a knowledge base of globally observed adversary tactics and techniques. This course provides an understanding of behaviors that may be used for developing threat models, mapping threats, classifying attacks, or training both red and blue teams.

Topics Include:

- The purpose of the ATT&CK Framework
  Structures, tactics, and techniques within the framework

- Using the ATT&CK Framework to detect and analyze threats
- Mitigation best-practices for preventing attacks

□ COD 106 – The Importance of Software Integration and Testing **(15 mins)**

This course introduces you to the Integration and Testing phases of the software development lifecycle, including the roles of Code Review, Fault Injection, Vulnerability Scanning, Penetration Testing, and Static Analysis.
Topics Include:

- Pros and cons of performing a code review
- Resources available when conducting a code review
- Identifying vulnerabilities that may have been missed by other secure testing techniques
- Utilizing fault injection
- Vulnerability Scanning and Penetration Testing
- Pros and cons of static analysis

□ CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP) **(20 mins)**

Proactive cyber threat hunting tactics have evolved to use new threat intelligence on previously collected data to identify and categorize potential threats in advance of attack. Learn to leverage NIST and MITRE ATT&CK security frameworks to protect your organization against cyber-attacks.
After completing this course, learners should have the knowledge and skills needed to understand:

- Basics of Cyber Threat Hunting & Threat Analytics
- Evolution of Cyber Threat Hunting as a Domain of Practice
- How Threat Hunting fits into the cyber risk management lifecycle
- Tactics, Techniques, Procedures, and the MITRE ATT&CK Framework
- Practical “Starting Points” for building a Cyber Threat Hunting Program

□ CYB 301 – Fundamentals of Ethical Hacking **(15 mins)**

As hackers continue to evolve their techniques organizations must train their employees to test their defenses through various penetration techniques. This course introduces common activities performed during the process of Ethical Hacking and provides a basic foundation of common attack techniques and examples of hacking tools.
Topics Include:

- Understanding authorization and scope that define ethical hacking
- Implementing the penetration testing process
- Fundamentals of attacker techniques and the ATT& CK framework
- An overview of hacking skills and tools knowledge domain

□ CYB 311 – Threat Analysis with AI (NEW) **(20 mins)**

AI analyzes relationships between threats like malicious files, suspicious IP addresses or insiders in seconds or minutes. AI provides curated risk analysis, reducing the time security analysts take to make critical decisions and remediate threats. In this course, we discuss how AI is helping organizations protect themselves against cyber-attacks. This includes the fundamental components of AI, such as sandboxes and trained data, as well as the logic used in machine learning, neural networks, and deep learning.
On successful completion of this course, learners should have the knowledge and skills required to:

- Perform Threat Analysis with AI
- Understand AI logic and specific use cases of AI in the threat detection landscape

- Use AI for application development, malware analysis, and user behavioral analytics

□ SDT 301 – Testing for Injection **(10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A03:2021 Injection, as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will understand how to:

- Keep data separate from commands and queries
- Implement multi-factor authentication
- Require weak password checks
- Limit login attempts

□ SDT 302 – Testing for Identification and Authentication Failures **(10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A07:2021 Identification and Authentication Failures, as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will understand how to:

- Use secure coding best practices to confirm user identity
- Implement strong authentication mechanisms
- Protect user sessions and session data

□ SDT 303 – Testing for Cryptographic Failures **(10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A02:2021 Cryptographic Failures, as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will understand how to:

- Identify common cryptographic failures
- Mitigate sensitive data exposure
- Define your organization's cryptography environment and testing methods
- Apply best practices and ensure compliance

□ SDT 304 – Testing for Insecure Design **(10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A04:2021 Insecure Design, as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will understand how to test for failures with common secure design issues, including:

- The secure software development lifecycle
- Using secure design patterns
- Performing threat modeling
- Preventing excessive resource consumption

□ SDT 305 – Testing for Broken Access Control **(10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A01:2021 Broken Access Control, as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will understand how to:

- Identify common access control flaws
- Mitigate access control failures
- Define mitigation measures to protect against broken access control
- Apply best practices and ensure compliance

□ **SDT 306 – Testing for Security Misconfiguration (10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A05:2021 Security Misconfiguration, as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will understand how to:

- Identify common vulnerabilities caused by security misconfiguration
- Develop and implement an effective testing strategy to evaluate attack surfaces
- Conduct manual and automated tests against insecure installation processes
- Identify targeted operating systems, web servers, and databases

□ **SDT 307 – Testing for Server-Side Request Forgery (SSRF) (10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A10:2021 Server-Side Request Forgery (SSRF), as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will have the knowledge and skills required to:

- Recognized user input potentially exploitable for executing SSRF attacks
- Exploit SSRF vulnerabilities, mapping normally unreachable networks
- Understand how to bypass detection and validation code
- Gain access to cloud metadata

□ **SDT 308 – Testing for Software and Data Integrity Failures (10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A08:2021 Software and Data Integrity Failures, as defined by the Open Web Application Security Project (OWASP).

After completing this course, you will have the knowledge and skills required to:

- Understand and Identify software and data integrity failures
- Know the risks of deserialization vulnerabilities and how to test for them
- Protect stored data using message authentication codes and digital signatures
- Implement best practices to limit software and data integrity failures

□ **SDT 309 – Testing for Vulnerable and Outdated Components (10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A06:2021 Vulnerable and Outdated Components, as defined by the Open Web Application Security Project (OWASP). After completing this course, you will understand how to test your application for flaws related to known insecure components and apply mitigation measures to protect against them.

□ **SDT 310 – Testing for Security Logging and Monitoring Failures (10 mins)**

This course explains how software developers and testers can determine if their web applications are vulnerable to A09:2021 Security Logging and Monitoring Failures, as defined by the Open Web Application Security Project (OWASP). After completing this course, you will understand how to test your application for insufficient logging and monitoring flaws and apply mitigation measures to protect against them.

□ **SDT 311 – Testing for Integer Overflow or Wraparound (15 mins)**

An integer overflow or wraparound may often be intended behavior; however, it can also introduce other weaknesses and security consequences. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-190 by the 2020 CWE Top 25.

Topics include:

- Recognizing the impact of this vulnerability
- Techniques for finding Integer Overflow issues through code review
- Application of secure coding best practices to prevent these attacks

- Testing to detect Integer Overflow or Wraparound

□ SDT 312 – Testing for (Path Traversal) Improper Limitation of a Pathname to a Restricted Directory **(15 mins)**

Many file operations are intended to take placed within a restricted directory, however, the software does not properly neutralize special elements within a pathname which results in various security consequences. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-22 by the 2020 CWE Top 25.
Topics include:

- Recognizing the impact of this vulnerability
- Techniques for finding path traversal issues through code review
- Application of secure coding best practices to prevent these attacks
- Testing to detect this security weakness

□ SDT 313 – Testing for (CSRF) Cross Site Request Forgery **(15 mins)**

Cross-Site Request Forgery (CSRF) occurs when a web application does not, or cannot, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-352 by the 2020 CWE Top 25.
Topics include:

- Recognizing the impact of this vulnerability
- Techniques for finding CSRF issues through code review
- Application of secure coding best practices to prevent these attacks
- Testing to detect this security weakness

□ SDT 314 – Testing for Unrestricted Upload of File with Dangerous Type **(15 mins)**

Unrestricted Upload of File with Dangerous Type vulnerabilities allows attackers to upload malicious code. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-434 by the 2021 CWE Top 25.
Topics include:

- Recognizing the impact of this vulnerability
- Techniques for finding Unrestricted Upload vulnerabilities in an application source code
- Application of secure coding best practices to prevent these attacks
- Testing to detect this security weakness

□ SDT 315 – Testing for Incorrect Permission Assignment for Critical Resource **(15 mins)**

The use of insecure settings for access permissions allows attackers to perform unauthorized access either to some part of the system or to an application-controlled resource. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-732 by the 2020 CWE Top 25.
Topics include:

- Recognizing the impact of this vulnerability
- Techniques for finding Incorrect Permission Assignment for critical resource in an application source code
- Application of secure coding best practices to prevent these attacks
- Testing to detect this security weakness

□ SDT 316 – Testing for Use of Hard-Coded Credentials **(15 mins)**

Applications that use authentication need a method for storing credentials that is secure because when a hacker recovers credentials, they can use them to authenticate with the application or to access external services. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-798 by the 2020 CWE Top 25.
Topics include:

- Recognizing the impact of this vulnerability
- Techniques for finding Hard-Coded credentials in source code
- Application of secure coding best practices to prevent these attacks
- Testing to detect this security weakness

□ SDT 317 – Testing for Improper Control of Generation of Code **(10 mins)**

When user input can influence dynamically generated code to influence program flow or execute arbitrary code the attack is often referred to as code injection. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-94 by the 2020 CWE Top 25.
Topics include:

- Recognizing the impact of this vulnerability
- Understanding various forms of this attack and their similarities
- Techniques for finding Hard-Coded credentials in source code
- Application of mitigation techniques for limiting the impact
- Leveraging various tools used to test for code injection vulnerabilities

□ SDT 318 – Testing for Insufficiently Protected Credentials **(10 mins)**

Much of the security we rely upon at some point comes down to the passwords we use to authenticate an application. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-522 by the 2020 CWE Top 25.
Topics include:

- Understanding the applicability and impact of this weakness in depth
- Using appropriate security mechanism to protect credentials
- Applying methods of prevention, testing, and mitigation to defend against Insufficiently Protected Credentials

□ SDT 319 – Testing for Out-of-bounds Read **(10 mins)**

Out-of-bounds Read is a security defect that can allow attackers to read sensitive information from other memory locations or cause a crash. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-125 by the 2020 CWE Top 25.
Topics include:

- Identifying Out-of-bounds Read errors
- Recognizing the impact of this vulnerability
- Application of secure coding best practices
- Testing to detect errors

□ SDT 320 – Testing for Out-of-bounds Write **(10 mins)**

Out-of-bounds Write can result in corruption of data, a crash, or code execution. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-787 by the 2020 CWE Top 25.
Topics include:

- Identifying Out-of-bounds Write errors
- Recognizing the impact of this vulnerability
- Application of secure coding best practices
- Testing to detect errors

□ SDT 321 – Testing for Uncontrolled Resource Consumption **(10 mins)**

Uncontrolled Resource consumption occurs when software does not properly control the allocation and maintenance of limited resources such as memory, file system storage, database connection pool entries, and CPU. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-400 by the 2020 CWE Top 25.
Topics include:

- Identifying Uncontrolled Resource Consumption
- Recognizing the impact of this vulnerability
- Application of secure coding best practices
- Testing to detect this vulnerability

□ SDT 322 – Testing for Improper Privilege Management **(10 mins)**

Improper Privilege Management occurs when software does not properly assign, modify, track, or check privileges for an actor, creating an unintended sphere of control for that actor. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-269 by the 2020 CWE Top 25.
Topics include:

- Identifying main threats that lead to abusing the privilege
- Recognizing the impact of this vulnerability
- Best practices for defending against unmanaged privileges
- Testing to detect Improper Privilege Management

□ SDT 323 – Testing for Improper Input Validation **(10 mins)**

Input validation is used to check potentially dangerous inputs but when software does not validate this input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-20 by the 2020 CWE Top 25.
Topics include:

- Identifying malicious input
- Recognizing the impact of this vulnerability
- Strategies for defending against Improper Input Validation
- Testing for Improper Input Validation weaknesses

□ SDT 324 – Testing for Improper Restriction of Operations within the Bounds of a Memory Buffer **(10 mins)**

Improper Restriction of Operations within the Bounds of a Memory Buffer allows attackers to execute arbitrary code, alter the intended control flow, read sensitive information, or cause a system to crash. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-119 by the 2020 CWE Top 25.
Topics include:

- Identifying Out of Range Memory Access errors
- Recognizing the impact of this vulnerability
- Applying preventative measures to avoid this weakness
- Common code mitigation strategies
- Using a multi-pronged approach to test for Improper Restriction of Operations with the Bounds of a Memory Buffer

□ SDT 325 – Testing for NULL Pointer Dereference **(10 mins)**

NULL pointer dereferences issues can occur through a number of flaws, including race conditions and simple programming omissions. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-476 by the 2020 CWE Top 25.
Topics include:

- Recognizing the impact of this vulnerability

- Defending Against NULL Pointer Dereference
- Best practices for preventing NULL Pointer Dereference
- Testing techniques for spotting NULL Pointer Dereference
- Mitigation strategies for this weakness

□ SDT 326 – Testing for Use After Free **(10 mins)**

The use of previously freed memory can have any number of adverse consequences, but these errors have two common and sometimes overlapping causes. This course introduces ways to identify and mitigate this security weakness, referenced as CWE-416 by the 2020 CWE Top 25. Topics include:

- Identification of Use After Free Errors
- Recognizing the impact of this vulnerability
- Defending against Use After Free weaknesses
- Methods of Prevention
- Testing techniques for spotting Use After Free
- Secure coding best practices for mitigating this vulnerability

□ TST 101 – Fundamentals of Security Testing **(20 mins)**

This course introduces security testing concepts and processes that will help testers/QA teams analyze an application from a security perspective to conduct more effective security testing. Topics include:

- Classes of security vulnerabilities and testing approaches that target them
- Manual and automated test techniques
- Identifying common security issues
- Threat modeling, approaches and how they apply to the design phase of the SDLC
- Vulnerability scanning, penetration testing, static analysis, and code review

□ TST 202 – Penetration Testing Fundamentals **(25 mins)**

Serving as a comprehensive way of testing for cybersecurity vulnerabilities Penetration Testing provides insight into a network, application, device, and/or physical security through the lens of an attacker to discover weaknesses and identify areas of improvement within your security posture. This course introduces concepts of penetration testing and provides an understanding of the stages of penetration testing as they relate to industry standards.
After completing this course, you will be able to:

- Conduct penetration testing according to an industry-standard methodology
- Identify the steps in a typical penetration testing process

□ TST 205 – Performing Vulnerability Scans **(45 mins)**

Performing vulnerability scans is a necessary first step to evaluating the security of an organization’s network and helping protect organizational data and assets; this includes assessing, mitigating, and reporting on any security vulnerabilities that exist in an organization’s systems and software.
Topic includes:

- Enumerating Platforms, Software Flaws, and improper configurations
- Formatting Checklists and test procedures
- Measuring vulnerability impact
- Analyzing vulnerability scan reports and results from security control assessments

□ TST 206 – ASVS Requirements for Developers **(20 mins)**

Ensuring developers understand application security needs can be overwhelming, but leveraging OWASP ASVS organizations can test and prove applications meet specific levels of security. This course is designed to equip Privacy and Cybersecurity Management with the knowledge required to provide development teams with a basis for testing web application technical security controls and a list of requirements for secure development in adherence to the Application Security Verification Standard (ASVS) 3.0 standard.

Upon successful completion of this course, learners will have the knowledge and skills required to meet ASVS compliance requirements, including:

- Using the ASVS to audit applications and to establish both internal and procurement metrics
- Understanding the role of ASVS Levels and Threat profiles
- Providing the necessary guidance and training to ensure your organization meets ASVS requirements.

□ TST 301 – Infrastructure Penetration Testing **(45 mins)**

Reliance on IT systems, regulatory compliance, and the evolving cyberthreat landscape are key indicators of the importance of Infrastructure penetration testing. Infrastructure Penetration tests can help inform cybersecurity strategies, validate existing security controls, and identify weaknesses in need of improvement. This course provides learners with the skills and knowledge necessary to perform penetration tests that simulate how attackers might attempt to compromise the organization's infrastructure.

After completing this course you will be able to:

- Perform pretest identification of potential vulnerabilities based on pretest analysis
- Leverage automated scanning tools
- Establish a baseline indication of the potential attack surface of the environment
- Interpret the results of automated tools to determine what additional testing is needed
- Perform host discovery, port scanning, and network segmentation checks
- Analyze the results of the penetration test are then compiled into a report

□ TST 302 – Application Penetration Testing **(45 mins)**

Applications store, process, and transmit data making them susceptible and vulnerable to hackers who can identify and exploit vulnerabilities. Penetration testing of these applications acts as a safeguard to reduce vulnerabilities and attack surface. This course provides learners with the skills and knowledge necessary to perform penetration tests that simulate how attackers might attempt to compromise the software applications.

After completing this course you will be able to:

- Conduct planning and reconnaissance
- Scan to understand how the target application will respond to various intrusion attempts
- Gain access using web application attacks, such as cross-site scripting, SQL injection, and backdoors, to uncover a target's vulnerabilities
- Maintain access to determine if the vulnerability can be used to achieve a persistent presence in the exploited system
- Analyze the results of the penetration test are then compiled into a report

□ TST 303 – Penetration Testing for Google Cloud Platform **(20 mins)**

Google Cloud Platform (GCP) offers many security features/services under a shared-responsibility model. Still, there are numerous ways an external attacker can gain access to your cloud environments, thus driving the need for in-depth assessments. This course covers the fundamentals of Penetration Testing within Google Cloud Platform for common GCP vulnerabilities and misconfigurations that can leave your cloud environments exposed.

After completing this course, you will be able to:

- Prepare, plan, and conduct penetration testing in accordance with industry-standard methodologies

- Identify general cloud attack vectors as well as recognize weaknesses specific to the Google Cloud Platform
- Use standard tools, techniques, and open-source software for testing Google Cloud Platform resources
- Create and implement a plan for securing the cloud-based on industry best practices

□ TST 304 – Penetration Testing for AWS Cloud **(20 mins)**

Amazon Web Services (AWS) offers a range of cloud hosting services, but AWS only permits security testing of user-operated services. Performing a penetration test in AWS requires adequate planning and expert knowledge of how AWS methodologies differ from traditional pen testing and what can be performed. This course covers the fundamentals of penetration testing within Amazon Web Services. It provides an understanding of how to evaluate AWS cloud services and the types of tools and tests permitted.

After completing this course, you will be able to:

- Prepare, plan, and conduct penetration testing in accordance with industry-standard methodologies
- Identify general cloud attack vectors as well as recognize weaknesses specific to the AWS cloud platform
- Use common tools, techniques, and open-source software for testing AWS resources
- Create and implement a plan for securing the cloud based on industry best practices

□ TST 305 – Penetration Testing for Azure Cloud **(20 mins)**

Conducting penetration testing of assets such as web applications, networks, and network devices in the Azure environment requires knowledge of Microsoft Azure methodologies and the common types of penetration tests allowed. This course covers the fundamentals of penetration testing within the Azure cloud while explaining how to evaluate Azure services and ensure your Azure cloud infrastructure is designed and configured according to best practices.

After completing this course, you will be able to:

- Prepare, plan, and conduct penetration testing in accordance with industry-standard methodologies
- Identify general cloud attack vectors as well as recognize weaknesses specific to the Azure cloud platform
- Use standard tools, techniques, and open-source software for testing Azure resources
- Create and implement a plan for securing the cloud-based on industry best practices

□ TST 351 – Penetration Testing for TLS Vulnerabilities **(12 mins)**

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. However, flaws in TLS protocol include weak cryptographic primitives, or specific implementation errors, cross-protocol vulnerabilities, or any combination of each. This course teaches how to identify vulnerabilities, detecting acceptance of unencrypted connections, and testing configurations.

After completing this course, you will be able to:

- Identify typical TLS misconfiguration vulnerabilities
- Detect network services accepting unencrypted connections
- Test web server TLS configuration

□ TST 352 – Penetration Testing for Injection Vulnerabilities **(12 mins)**

Stemming from improperly sanitized or completely unsensitized input injection flaws allow attackers to relay malicious code through an application to another system. This course teaches how to identify and test for these vulnerabilities within your code.

After completing this course, you will be able to:

- Identify common injection vulnerabilities
- Test for command injection vulnerabilities
- Detect code and XML injection vulnerabilities
- Exploit command and code injection vulnerabilities

□ TST 353 – Penetration Testing for SQL Injection **(12 mins)**

Used to attack data-driven applications in which malicious SQL statements are inserted into an entry field for execution SQL Injection allows attackers to conduct a number of malicious activities to data including but not limited to becoming administrators of the database server. This course teaches how to identify, test, and exploit these vulnerabilities.
After completing this course, you will be able to:

- Test for the presence of SQL Injection vulnerabilities
- Exploit SQL Injection vulnerabilities
- Identify the common tools and techniques used to exploit SQL Injection vulnerabilities

□ TST 354 – Penetration Testing for Memory Corruption Vulnerabilities **(12 mins)**

Occurring when the contents of a memory location are modified due to programmatic behavior that exceeds the intention of the original programmer or program/language constructs. This type of programming error can lead to a program crash or strange and bizarre program behavior. This course teaches how to identify, test, and exploit these vulnerabilities.
After completing this course, you will be able to:

- Identify common memory corruption vulnerabilities
- Test for buffer overflows + Exploit known memory corruption vulnerabilities
- Understand advanced techniques for finding memory corruption vulnerabilities

□ TST 355 – Penetration Testing for Authorization Vulnerabilities **(12 mins)**

Authorization is the process of enforcing policies; determining what types of qualities of activities, resources, or services a user is permitted. Authorization vulnerabilities include forceful browsing and privilege escalation. This course teaches how to identify, test, and exploit these vulnerabilities.
After completing this course, you will be able to:

- Identify common authorization vulnerabilities
- Test application access controls
- Exploit authorization vulnerabilities

□ TST 356 – Penetration Testing for Cross-Site Scripting (XSS) **(12 mins)**

Cross-site Scripting (XSS) is a client-side code injection attack where the attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application. This course teaches how to identify, test, and exploit these vulnerabilities.
After completing this course, you will be able to:

- Define the types of Cross-Site Scripting vulnerabilities
- Test applications for Cross-Site Scripting vulnerabilities
- Exploit Cross-Site Scripting vulnerabilities

□ TST 357 – Penetration Testing for Hardcoded Secrets **(12 mins)**

All modern applications rely on certain secrets to run from database connection strings to API keys or cryptographic keys. Keeping these secrets is critical to the security of the application as they typically create a significant hole that allows an attacker to bypass the authentication that has been configured by the software administrator. This course teaches how to identify and test for the use of hard-coded credentials.

After completing this course, you will be able to:

- Determine whether an application contains hard-coded authentication credentials
- Determine whether an application contains hard-coded cryptographic keys
- Find plain-text secrets in application binaries
- Find leaked secrets in code repositories
- Identify techniques for advanced testing of application code for the presence of hard-coded secrets

□ TST 358 – Penetration Testing Wireless Networks **(12 mins)**

Wireless networks have security issues that are vulnerable to various attacks. Organizations need to proactively search out any weakness in security if they are to avoid unauthorized access to network resources and data leakage. This course introduces tools and techniques while teaching how to Identify and test for common attacks.

After completing this course, you will be able to:

- Test for the presence of unauthorized wireless networks
- Identify common attacks on wireless networks
- Identify the common tools and techniques for testing wireless networks

□ TST 359 – Penetration Testing Network Infrastructure **(12 mins)**

Essential to every organization; Infrastructure penetration testing provides an opportunity to know about the current situation of a company and analyze existing potential breach points. The process includes all internal computer systems, associated external devices, internet networking, cloud, and virtualization testing. This course teaches how to perform Network Infrastructure penetration tests, perform necessary scans, and test controls.

After completing this course, you will be able to:

- Perform network-layer penetration tests
- Test network segmentation controls
- Perform a network scan to discover active devices
- Perform a port scan on a host to identify exposed network services

□ TST 360 – Penetration Testing for Authentication Vulnerabilities **(12 mins)**

Building authentication and session management schemes correctly is a difficult task often presenting flaws that may equally difficult to Identify. Common authentication attacks consist of brute force, insufficient authentication, and weak password recovery validation. These types of attacks target and attempt to exploit the authentication process a web site uses to verify the identity of a user, service, or application. This course teaches how to execute attacks, identify vulnerabilities, and verify controls.

After completing this course, you will be able to:

- Identify common authentication vulnerabilities
- Verify authentication controls
- Execute dictionary attacks

Skill Labs

□ LAB 211 – Defending Java Applications Against Credentials in Code Medium **(10 mins)**

The Defending Java Applications Against Credentials in Code Medium lab assesses the learner’s ability to fix code that contains unprotected credentials such as a password or cryptographic key.

After completing this lab, they will understand how to avoid exposing credentials in code medium.

- □ **LAB 212 – Defending Python Applications Against Credentials in Code Medium** **(10 mins)**
  The Defending Python Applications Against Credentials in Code Medium lab assesses the learner’s ability to fix code that contains unprotected credentials such as a password or cryptographic key. After completing this lab, they will understand how to avoid exposing credentials in code medium.

- □ **LAB 213 – Defending Node.js Applications Against Credentials in Code Medium** **(10 mins)**
  The Defending Node.js Applications Against Credentials in Code Medium lab assesses the learner’s ability to fix code that contains unprotected credentials such as a password or cryptographic key. After completing this lab, they will understand how to avoid exposing credentials in code medium.

- □ **LAB 214 – Defending C# Applications Against Credentials in Code Medium** **(10 mins)**
  The Defending C# Applications Against Credentials in Code Medium lab assesses the learner’s ability to fix code that contains unprotected credentials such as a password or cryptographic key. After completing this lab, they will understand how to avoid exposing credentials in code medium.

- □ **LAB 215 – Defending Java Applications Against Business Logic Error for Input Validation** **(10 mins)**
  The Defending Java Applications Against Business Logic Error for Input Validation lab assesses the learner’s ability to fix business logic errors that leave your application vulnerable to manipulation by attackers. After completing this lab, the learner will understand how to fix business logic code errors in Java Applications that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 216 – Defending Python Applications Against Business Logic Error for Input Validation** **(10 mins)**
  The Defending Python Applications Against Business Logic Error for Input Validation lab assesses the learner’s ability to fix business logic errors that leave your application vulnerable to manipulation by attackers. After completing this lab, the learner will understand how to fix business logic code errors in Python Applications that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 217 – Defending Node.js Applications Against Business Logic Error for Input Validation** **(10 mins)**
  The Defending Node.js Applications Against Business Logic Error for Input Validation lab assesses the learner’s ability to fix business logic errors that leave your application vulnerable to manipulation by attackers. After completing this lab, the learner will understand how to fix business logic code errors in Node.js Applications that may leave your application vulnerable to manipulation by attackers.

□ LAB 218 – Defending C# Applications Against Business Logic Error for Input Validation **(10 mins)**

The Defending C# Applications Against Business Logic Error for Input Validation lab assesses the learner's ability to fix business logic errors that leave your application vulnerable to manipulation by attackers. After completing this lab, the learner will understand how to fix business logic code errors in C# Applications that may leave your application vulnerable to manipulation by attackers.

□ LAB 220 – Defending Against Hard-Coded Secrets **(5 mins)**

Inclusion of Sensitive Information in source code comments is a type of vulnerability that allows malicious actors who are able to view the source code to recover that sensitive information, such as credentials or information about the infrastructure, and leverage it for attacks. This lab involves mitigating the issue in vulnerable code that contains authentication credentials. In this lab, the learner will use an IDE to fix a Hard-coded Secret vulnerability in the code of a static web page without making any unnecessary changes to the code or the system.

□ LAB 221 – Defending C# Applications Against SQL Injection **(10 mins)**

This lab simulates an Injection vulnerability found in the Gold Standard Cyber Range. The challenge includes a Web App developed in C# that fails to implement the security principle of "Establish Secure Defaults." Using Visual Studio Code, participants will analyze code to identify and mitigate instances of "Failure to validate input and improper use of user input in SQL statements". The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep C# applications safe
- Demonstrate the skills needed to discover and exploit SQL Injection attacks
- Fix a vulnerable SQL query in C#

□ LAB 222 – Defending Python Applications Against SQL Injection **(10 mins)**

This lab simulates an Injection vulnerability found in the Gold Standard Cyber Range. The challenge includes a Web App developed in Python that fails to implement the security principle of "Establish Secure Defaults." Using Visual Studio Code, participants will analyze code to identify and mitigate instances of "Failure to validate input and improper use of user input in SQL statements". The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep Python applications safe
- Demonstrate the skills needed to discover and exploit SQL Injection attacks
- Fix a vulnerable SQL query in Python

□ LAB 223 – Defending Node.js Applications Against SQL Injection **(10 mins)**

This lab simulates an Injection vulnerability found in the Gold Standard Cyber Range. The challenge includes a Web App developed in Node.js that fails to implement the security principle of "Establish Secure Defaults." Using Visual Studio Code, participants will analyze code to identify and mitigate instances of "Failure to validate input and improper use of user input in SQL statements". The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep Node.js applications safe
- Demonstrate the skills needed to discover and exploit SQL Injection attacks
- Fix a vulnerable SQL query in Node.js

□ LAB 224 – Defending Java Applications Against Forceful Browsing **(10 mins)**

The Defending Java Applications Against Forceful Browsing lab assesses the learner's ability to fix code that does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files. After completing this lab, the learner will understand how to fix Java Forceful Browsing that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 225 – Defending Python Applications Against Forceful Browsing** **(10 mins)**
  The Defending Python Applications Against Forceful Browsing lab assesses the learner's ability to fix code that does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files. After completing this lab, the learner will understand how to fix Python Forceful Browsing that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 226 – Defending Node.js Applications Against Forceful Browsing** **(10 mins)**
  The Defending Node.js Applications Against Forceful Browsing lab assesses the learner's ability to fix code that does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files. After completing this lab, the learner will understand how to fix Node.js Forceful Browsing that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 227 – Defending C# Applications Against Forceful Browsing** **(10 mins)**
  The Defending C# Applications Against Forceful Browsing lab assesses the learner's ability to fix code that does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files. After completing this lab, the learner will understand how to fix C# Forceful Browsing that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 228 – Defending Java Applications Against Weak AES ECB Mode Encryption** **(10 mins)**
  The Defending Against Weak Encryption Mode lab assesses the learner's understanding of using secure encryption modes. After completing this lab, the learner will understand how to use secure encryption modes.

- □ **LAB 229 – Defending Java Applications Against Weak PRNG** **(10 mins)**
  The Defending Against Weak PRNG lab assesses the learner's understanding of using cryptographically strong Pseudo-Random Number Generators. After completing this lab, the learner will understand how to use cryptographically strong Pseudo-Random Number Generators.

- □ **LAB 230 – Defending Java Applications Against XSS** **(15 mins)**
  This lab simulates a cross-site scripting vulnerability that can be found in an online banking application built using Java which fails to validate input and encode output. Using Visual Studio Code participants will determine whether the data is correctly encoded for the context in which it will appear in web application output. The objective of this lab is to find the cross-site scripting (XSS) vulnerability found in this Java web application and fix the issue.
  Upon completion of this lab participants will:
    - Apply strategic principles to keep web applications safe
    - Demonstrate the skills needed to discover and exploit cross-site scripting attacks
    - Fix a cross-site scripting vulnerability in Java

- □ **LAB 231 – Defending Python Applications Against XSS** **(15 mins)**
  This lab simulates a cross-site scripting vulnerability that can be found in an online banking application built using Python which fails to validate input and encode output Using Visual Studio Code participants will determine whether the data is correctly encoded for the context in which it will appear in web application output. The objective of this lab is to find the cross-site scripting (XSS) vulnerability found in this Python web application and fix the issue.
  Upon completion of this lab participants will:
    - Apply strategic principles to keep web applications safe
    - Demonstrate the skills needed to discover and exploit cross-site scripting attacks
    - Fix a cross-site scripting vulnerability in Python

□ **LAB 232 – Defending C# Applications Against XSS** **(15 mins)**
This lab simulates a cross-site scripting vulnerability that can be found in XYZ Range. Using Visual Studio Code participants will determine whether the data is correctly encoded for the context in which it will appear in web application output. The objective of this lab is to find the cross-site scripting (XSS) vulnerability found in this C# application and fix the issue.
Upon completion of this lab participants will:

- Apply strategic principles to keep web applications safe
- Demonstrate the skills needed to discover and exploit cross-site scripting attacks
- Fix a cross-site scripting vulnerability in C#

□ **LAB 233 – Defending Node.js Applications Against XSS** **(15 mins)**
This lab simulates a cross-site scripting vulnerability that can be found in an online banking application built using Node.js which fails to validate input and encode output. Using Visual Studio Code participants will determine whether the data is correctly encoded for the context in which it will appear in web application output. The objective of this lab is to find the cross-site scripting (XSS) vulnerability found in this TECHNOLOGY web application and fix the issue.
Upon completion of this lab participants will:

- Apply strategic principles to keep web applications safe
- Demonstrate the skills needed to discover and exploit cross-site scripting attacks
- Fix a cross-site scripting vulnerability in Node.js

□ **LAB 234 – Defending Java Applications Against Parameter Tampering** **(10 mins)**
The Defending Against Parameter Tampering lab assesses the learner’s understanding of user authorization to prevent Parameter Tampering vulnerabilities. After completing this lab, the learner will understand how to use authorization to prevent Parameter Tampering vulnerabilities.

□ **LAB 235 – Defending Java Applications Against Plaintext Password Storage** **(10 mins)**
The Defending Against Plaintext Password Storage lab assesses the learner’s understanding of protecting stored authentication credentials. After completing this lab, the learner will understand how to protect stored authentication credentials.

□ **LAB 236 – Defending Java Applications Against Sensitive Information in Error Messages** **(10 mins)**
The Defending Against Sensitive Information in Error Messages lab assesses the learner’s ability to prevent disclosing sensitive information in error messages. In this lab the learner will fix a vulnerability that discloses sensitive information in error messages. The lab features an authentication page that discloses whether a specific username is valid or not when invalid authentication credentials are provided, thus allowing valid username enumeration.

□ **LAB 237 – Defending Java Applications Against SQL Injection** **(20 mins)**
This lab simulates a SQL Injection vulnerability that can be found in Shadow Bank which fails to validate input and consists of improper use of user input in SQL statements. Using Visual Studio Code participants will determine if the generated SQL query can be exploited. The objective of this lab is to fix the SQL Injection vulnerability found in this Java application and fix the issue.
Upon completion of this lab participants will:

- Apply strategic principles to keep Java applications safe
- Demonstrate the skills needed to discover and exploit SQL Injection attacks
- Fix a vulnerable SQL query in Java

□ LAB 238 – Defending C# Applications Against Weak AES ECB Mode Encryption **(10 mins)**

The Defending Against Weak Encryption Mode lab assesses the learner's understanding of using secure encryption modes. After completing this lab, the learner will understand how to use secure encryption modes.

□ LAB 239 – Defending C# Applications Against Weak PRNG **(10 mins)**

The Defending Against Weak PRNG lab assesses the learner's understanding of using cryptographically strong Pseudo-Random Number Generators. After completing this lab, the learner will understand how to use cryptographically strong Pseudo-Random Number Generators.

□ LAB 240 – Defending Java Applications Against eXternal XML Entity (XXE) Vulnerabilities **(10 mins)**

This lab simulates a Weak File Upload Validation vulnerability found in the LetSee Cyber Range. The challenge includes a Web App developed in Java that fails to implement the security principle of "Validate all Untrusted Input Before Using." Using Visual Studio Code, participants will analyze code to identify and mitigate instances of "Failure to validate input in file upload". The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep Java applications safe
- Demonstrate the skills needed to discover XXE attacks
- Fix XXE vulnerabilities in Java

□ LAB 241 – Defending C# Applications Against eXternal XML Entity (XXE) Vulnerabilities **(10 mins)**

This lab simulates a Weak File Upload Validation vulnerability found in the LetSee Cyber Range. The challenge includes a Web App developed in C# that fails to implement the security principle of "Validate all Untrusted Input Before Using." Using Visual Studio Code, participants will analyze code to identify and mitigate instances of "Failure to validate input in file upload". The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep C# applications safe
- Demonstrate the skills needed to discover XXE attacks
- Fix XXE vulnerabilities in C#

□ LAB 242 – Defending Node.js Applications Against eXternal XML Entity (XXE) Vulnerabilities **(10 mins)**

This lab simulates a Weak File Upload Validation vulnerability found in the LetSee Cyber Range. The challenge includes a Web App developed in Node.js that fails to implement the security principle of "Validate all Untrusted Input Before Using." Using Visual Studio Code, participants will analyze code to identify and mitigate instances of "Failure to validate input in file upload". The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep Node.js applications safe
- Demonstrate the skills needed to discover XXE attacks
- Fix XXE vulnerabilities in Node.js

□ LAB 243 – Defending Python Applications Against eXternal XML Entity (XXE) Vulnerabilities **(10 mins)**

This lab simulates a Weak File Upload Validation vulnerability found in the LetSee Cyber Range. The challenge includes a Web App developed in Python that fails to implement the security

principle of “Validate all Untrusted Input Before Using.” Using Visual Studio Code, participants will analyze code to identify and mitigate instances of “Failure to validate input in file upload”. The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep Python applications safe
- Demonstrate the skills needed to discover XXE attacks
- Fix XXE vulnerabilities in Python

□ LAB 244 – Defending Java Applications Against Security Misconfiguration **(12 mins)**

This lab simulates a Security Misconfiguration vulnerability found in the DigiExchange Cyber Range. The challenge includes a Web App developed in Java that fails to implement the security principle of “Validate all Untrusted Input Before Using.” Using Visual Studio Code, participants will analyze code to identify and mitigate instances of “Failure to universally validate policy constraints”. The objective of this lab is to find the vulnerable code and fix the weakness.
Upon completion of this lab participants will:

- Apply strategic principles to keep Java applications safe from security misconfiguration
- Demonstrate the skills needed to discover security misconfiguration
- Fix security misconfiguration in Java

□ LAB 245 – Defending Node.js Applications Against Plaintext Password Storage **(10 mins)**

The Defending Against Plaintext Password Storage lab assesses the learner’s understanding of protecting stored authentication credentials. After completing this lab, the learner will understand how to protect stored authentication credentials.

□ LAB 246 – Defending Node.js Applications Against Weak AES ECB Mode Encryption **(10 mins)**

The Defending Against Weak Encryption Mode lab assesses the learner’s understanding of using secure encryption modes. After completing this lab, the learner will understand how to protect stored authentication credentials.

□ LAB 247 – Defending Node.js Applications Against Weak PRNG **(10 mins)**

The Defending Against Weak PRNG lab assesses the learner’s understanding of using cryptographically strong Pseudo-Random Number Generators. After completing this lab, the learner will understand how to use cryptographically strong Pseudo-Random Number Generators.

□ LAB 248 – Defending Node.js Applications Against Parameter Tampering **(10 mins)**

The Defending Against Parameter Tampering lab assesses the learner’s understanding of user authorization to prevent Parameter Tampering vulnerabilities. After completing this lab, the learner will understand how to use authorization to prevent Parameter Tampering vulnerabilities.

□ LAB 249 – Defending Python Applications Against Plaintext Password Storage **(10 mins)**

The Defending Against Plaintext Password Storage lab assesses the learner’s understanding of protecting stored authentication credentials. After completing this lab, the learner will understand how to protect stored authentication credentials.

□ **LAB 250 – Defending C# Applications Against Parameter Tampering (10 mins)**

The Defending Against Parameter Tampering lab assesses the learner's understanding of user authorization to prevent Parameter Tampering vulnerabilities. After completing this lab, the learner will understand how to use authorization to prevent Parameter Tampering vulnerabilities.

□ **LAB 251 – Defending C# Applications Against Plaintext Password Storage (10 mins)**

The Defending Against Plaintext Password Storage lab assesses the learner's understanding of protecting stored authentication credentials. After completing this lab, the learner will understand how to protect stored authentication credentials.

□ **LAB 252 – Defending Python Applications Against Weak AES ECB Mode Encryption (10 mins)**

The Defending Against Weak Encryption Mode lab assesses the learner's understanding of using secure encryption modes. After completing this lab, the learner will understand how to use secure encryption modes.

□ **LAB 253 – Defending Python Applications Against Weak PRNG (10 mins)**

The Defending Against Weak PRNG lab assesses the learner's understanding of using cryptographically strong Pseudo-Random Number Generators. After completing this lab, the learner will understand how to use cryptographically strong Pseudo-Random Number Generators.

□ **LAB 254 – Defending Python Applications Against Parameter Tampering (10 mins)**

The Defending Against Parameter Tampering lab assesses the learner's understanding of user authorization to prevent Parameter Tampering vulnerabilities. After completing this lab, the learner will understand how to use authorization to prevent Parameter Tampering vulnerabilities.

□ **LAB 260 – Defending C# Applications Against Sensitive Information in Error Messages (10 mins)**

The Defending Against Sensitive Information in Error Messages lab assesses the learner's ability to prevent disclosing sensitive information in error messages. In this lab the learner will fix a vulnerability that discloses sensitive information in error messages. The lab features an authentication page that discloses whether a specific username is valid or not when invalid authentication credentials are provided, thus allowing valid username enumeration.

□ **LAB 261 – Defending Python Applications Against Sensitive Information in Error Messages (10 mins)**

The Defending Against Sensitive Information in Error Messages lab assesses the learner's ability to prevent disclosing sensitive information in error messages. In this lab the learner will fix a vulnerability that discloses sensitive information in error messages. The lab features an authentication page that discloses whether a specific username is valid or not when invalid authentication credentials are provided, thus allowing valid username enumeration.

□ **LAB 262 – Defending Node.js Applications Against Sensitive Information in Error Messages (10 mins)**

The Defending Against Sensitive Information in Error Messages lab assesses the learner's ability to prevent disclosing sensitive information in error messages. In this lab the learner will fix a vulnerability that discloses sensitive information in error messages. The lab features an authentication page that discloses whether a specific username is valid or not when invalid authentication credentials are provided, thus allowing valid username enumeration.

- □ **LAB 263 – Defending Java Applications Against Sensitive Information in Log Files** **(10 mins)**
  The Sensitive Information in Log Files lab assesses the learner's ability to fix code in Java applications that places sensitive information in log files. After completing this lab, the learner will understand how to avoid disclosing sensitive information via application log files.

- □ **LAB 264 – Defending Python Applications Against Sensitive Information in Log Files** **(10 mins)**
  The Sensitive Information in Log Files lab assesses the learner's ability to fix code in Python applications that places sensitive information in log files. After completing this lab, the learner will understand how to avoid disclosing sensitive information via application log files.

- □ **LAB 265 – Defending Node.js Applications Against Sensitive Information in Log Files** **(10 mins)**
  The Sensitive Information in Log Files lab assesses the learner's ability to fix code in Node.js that places sensitive information in log files. After completing this lab, the learner will understand how to avoid disclosing sensitive information via application log files.

- □ **LAB 266 – Defending C# Applications Against Sensitive Information in Log Files** **(10 mins)**
  The Sensitive Information in Log Files lab assesses the learner's ability to fix code in C# applications that places sensitive information in log files. After completing this lab, the learner will understand how to avoid disclosing sensitive information via application log files.

- □ **LAB 267 – Defending Java Applications Against Deserialization of Untrusted Data** **(10 mins)**
  The Deserialization of Untrusted Data lab assesses the learner's ability to fix code in Java applications that allows attackers to execute arbitrary code by deserializing untrusted data using unsafe deserializers. After completing this lab, the learner will understand how to prevent and mitigate vulnerabilities associated with the use of unsafe deserializers.

- □ **LAB 268 – Defending Python Applications Against Deserialization of Untrusted Data** **(10 mins)**
  The Deserialization of Untrused Data lab assesses the learner's ability to fix code in Python applications that allows attackers to execute arbitrary code by deserializing untrusted data using unsafe deserializers. After completing this lab, the learner will understand how to prevent and mitigate vulnerabilities associated with the use of unsafe deserializers.

- □ **LAB 269 – Defending Node.js Applications Against Deserialization of Untrusted Data** **(10 mins)**
  The Deserialization of Untrused Data lab assesses the learner's ability to fix code in Node.js applications that allows attackers to execute arbitrary code by deserializing untrusted data using unsafe deserializers. After completing this lab, the learner will understand how to prevent and mitigate vulnerabilities associated with the use of unsafe deserializers.

- □ **LAB 270 – Defending C# Applications Against Deserialization of Untrusted Data** **(10 mins)**
  The Deserialization of Untrused Data lab assesses the learner's ability to fix code in C# applications that allows attackers to execute arbitrary code by deserializing untrusted data using unsafe deserializers. After completing this lab, the learner will understand how to prevent and mitigate vulnerabilities associated with the use of unsafe deserializers.

- □ **LAB 271 – Defending Java Applications Against SSRF (10 mins)**
  The Server-Side Request Forgery lab assesses the learner’s ability to fix code that allows attackers to exploit Java applications to send HTTP requests to arbitrary URLs. After completing this lab, the learner will understand how to prevent and mitigate Server-Side Request Forgery vulnerabilities.

- □ **LAB 272 – Defending Python Applications Against SSRF (10 mins)**
  The Server-Side Request Forgery lab assesses the learner’s ability to fix code that allows attackers to exploit Python applications to send HTTP requests to arbitrary URLs. After completing this lab, the learner will understand how to prevent and mitigate Server-Side Request Forgery vulnerabilities.

- □ **LAB 273 – Defending Node.js Applications Against SSRF (10 mins)**
  The Server-Side Request Forgery lab assesses the learner’s ability to fix code that allows attackers to exploit Node.js applications to send HTTP requests to arbitrary URLs. After completing this lab, the learner will understand how to prevent and mitigate Server-Side Request Forgery vulnerabilities.

- □ **AB 274 – Defending C# Applications Against SSRF (10 mins)**
  The Server-Side Request Forgery lab assesses the learner’s ability to fix code that allows attackers to exploit C# applications to send HTTP requests to arbitrary URLs. After completing this lab, the learner will understand how to prevent and mitigate Server-Side Request Forgery vulnerabilities.

- □ **LAB 275 – Defending Java Applications Against Command Injection (NEW) (10 mins)**
  The Defending Java Applications Against Command Injection lab assesses the learner’s ability to fix code that allows an attacker to execute arbitrary operating system (OS) commands on the server that is running the application. After completing this lab, they will understand how to defend Java applications against command injection vulnerabilities that may fully compromise the application and all its data.

- □ **LAB 276 – Defending Python Applications Against Command Injection (NEW) (10 mins)**
  The Defending Python Applications Against Command Injection lab assesses the learner’s ability to fix code that allows an attacker to execute arbitrary operating system (OS) commands on the server that is running the application. After completing this lab, they will understand how to defend Python applications against command injection vulnerabilities that may fully compromise the application and all its data.

- □ **LAB 277 – Defending Node.js Applications Against Command Injection (NEW) (10 mins)**
  The Defending Node.js Applications Against Command Injection lab assesses the learner’s ability to fix code that allows an attacker to execute arbitrary operating system (OS) commands on the server that is running the application. After completing this lab, they will understand how to defend Node.js applications against command injection vulnerabilities that may fully compromise the application and all its data.

- □ **LAB 278 – Defending C# Applications Against Command Injection (NEW) (10 mins)**
  The Defending C# Applications Against Command Injection lab assesses the learner’s ability to fix code that allows an attacker to execute arbitrary operating system (OS) commands on the

server that is running the application. After completing this lab, they will understand how to defend C# applications against command injection vulnerabilities that may fully compromise the application and all its data.

- □ **LAB 279 – Defending Java Applications Against Dangerous File Upload (NEW)** **(10 mins)**
  The Defending Java Applications Against Dangerous File Upload lab assesses the learner's ability to fix code that contains a File Upload vulnerability. After completing this lab, the learner will understand how to defend Java Applications against dangerous file uploads that allow attackers to place files onto a server and gain access forward using backdoor code.

- □ **LAB 280 – Defending Python Applications Against Dangerous File Upload (NEW)** **(10 mins)**
  The Defending Python Applications Against Dangerous File Upload lab assesses the learner's ability to fix code that contains a File Upload vulnerability. After completing this lab, the learner will understand how to defend Python Applications against dangerous file uploads that allow attackers to place files onto a server and gain access forward using backdoor code.

- □ **LAB 281 – Defending Node.js Applications Against Dangerous File Upload (NEW)** **(10 mins)**
  The Defending Node.js Applications Against Dangerous File Upload lab assesses the learner's ability to fix code that contains a File Upload vulnerability. After completing this lab, the learner will understand how to defend Node.js Applications against dangerous file uploads that allow attackers to place files onto a server and gain access forward using backdoor code.

- □ **LAB 282 – Defending C# Applications Against Dangerous File Upload (NEW)** **(10 mins)**
  The Defending C# Applications Against Dangerous File Upload lab assesses the learner's ability to fix code that contains a File Upload vulnerability. After completing this lab, the learner will understand how to defend C# Applications against dangerous file uploads that allow attackers to place files onto a server and gain access forward using backdoor code.

- □ **LAB 283 – Defending Java Applications Against RegEx DoS (NEW)** **(10 mins)**
  The Defending Java Applications Against RegEx DoS lab assesses the learner's ability to fix code allows attackers to capitalize on vulnerabilities RegEx engines face when matching regular expressions crashing the system or stopping the system from responding to user requests. After completing this lab, the learner will understand how to fix Java code that contains RegEx DoS vulnerability that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 284 – Defending Python Applications Against RegEx DoS (NEW)** **(10 mins)**
  The Defending Python Applications Against RegEx DoS lab assesses the learner's ability to fix code allows attackers to capitalize on vulnerabilities RegEx engines face when matching regular expressions crashing the system or stopping the system from responding to user requests. After completing this lab, the learner will understand how to fix Python code that contains RegEx DoS vulnerability that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 285 – Defending Node.js Applications Against RegEx DoS (NEW)** **(10 mins)**
  The Defending Node.js Applications Against RegEx DoS lab assesses the learner's ability to fix code allows attackers to capitalize on vulnerabilities RegEx engines face when matching regular expressions crashing the system or stopping the system from responding to user requests. After

completing this lab, the learner will understand how to fix Node.js code that contains RegEx DoS vulnerability that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 286 – Defending C# Applications Against RegEx DoS (NEW)** **(10 mins)**
  The Defending C# Applications Against RegEx DoS lab assesses the learner's ability to fix code allows attackers to capitalize on vulnerabilities RegEx engines face when matching regular expressions crashing the system or stopping the system from responding to user requests. After completing this lab, the learner will understand how to fix C# code that contains RegEx DoS vulnerability that may leave your application vulnerable to manipulation by attackers.

- □ **LAB 310 – ATT&CK: File and Directory Permissions Modification** **(12 mins)**
  The File and Directory Permissions Modification lab assesses the learner's ability to modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files. After completing this lab, the learner will understand how attackers may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files.

- □ **LAB 311 – ATT&CK: File and Directory Discovery** **(12 mins)**
  The File and Directory Discovery lab assesses the learner's ability to enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. After completing this lab, the learner will understand how attackers may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system.

- □ **LAB 312 – ATT&CK: Testing for Network Services Identification (NEW)** **(12 mins)**
  The File and Directory Permissions Modification lab assesses the learner's ability to identify network services on target hosts. After completing this lab the learner will understand how attackers may exploit Network Service Identification vulnerabilities to get a listing of services running on remote hosts and lock network infrastructure devices, including those that may be vulnerable to remote software exploitation.

- □ **LAB 313 – ATT&CK: Testing for Vulnerability Identification Using Vulnerability Databases (NEW)** **(12 mins)**
  The File and Directory Discovery lab assesses the learner's ability to identify vulnerabilities using public vulnerability databases. After completing this lab, the learner will understand how attackers use a listing of security software, configurations, defensive tools, and sensors that are installed on a system or in a cloud environment to shape follow-on behaviors, including whether or not they fully infect the target and/or attempt specific actions.

- □ **LAB 315 – ATT&CK: Updating Vulnerable Java Web Application Server Software** **(12 mins)**
  This lab simulates a Cross-Site Scripting (XSS) vulnerability found in the AccountAll Cyber Range. The challenge includes an HR Back Office System that fails to implement the security principle of "Establish Secure Defaults". Leveraging a virtual machine, participants will apply ATT&CK Mitigation "M1051 Update Software" to fix the vulnerable Java Web Application Server Software.

- □ **LAB 321 – ATT&CK: Password Cracking** **(5 mins)**
  The Brute Force: Password Cracking technique refers to attempting to recover authentication credentials from password hashes or other data sources that contain protected authentication credentials. This lab simulates a "Dictionary Attack", which uses a so-called dictionary of common passwords to determine whether any of these common passwords match the hashes

being cracked. The objective of this lab is to execute the Password Cracking ATT&CK technique. To complete the lab, you will need to answer a question about a secret that you discover by successfully executing the appropriate ATT&CK technique.

□ LAB 322 – ATT&CK: Exploiting Windows File Sharing Server with External Remote Services **(20 mins)**

The Exploitation of Remote Services technique refers to exploiting a vulnerability that is present in an online service provided by a target system. One of the defining characteristics of this technique is that the attacker has network access to the vulnerable service, either by the virtue of this service being exposed to a public network or because the attacker has already penetrated the network that the service is available on. The objective of this lab is to execute the Exploitation of Remote Services ATT&CK technique. To complete the lab, you will need to answer a question about a secret that you discover by successfully executing the appropriate ATT&CK technique.

□ LAB 323 – ATT&CK: Exploiting Vulnerable Java Web Application Server Software **(12 mins)**

This lab simulates a Lack of Resources & Rate Limiting vulnerability found in the LetSee Cyber Range. The challenge includes an Online Marketplace app that fails to implement the security principle of "Establish Secure Defaults". Within a virtual machine, participants will analyze code to identify and mitigate instances of "Failure to enforce strong password policy". The objective of this lab is to apply ATT&CK Techniques T1190 Exploit Public-Facing Application" and "T1133 External Remote Services".

□ LAB 324 – ATT&CK: Exploiting Java Web Application Server Misconfiguration **(12 mins)**

This lab simulates a Security Misconfiguration vulnerability found in the AccountAll Cyber Range. The challenge includes an HR Back Office System that fails to implement the security principle of "Establish Secure Defaults". Leveraging a virtual machine, participants will analyze code to identify and mitigate instances of "Misconfiguration of default credentials". The objective of this lab is to apply ATT&CK Techniques T1190 Exploit Public-Facing Application" and "T1133 External Remote Services".

□ LAB 330 – ATT&CK: Exploiting Java SQL Injection to Extract Password Hashes **(15 mins)**

Adversaries may "pass the hash" using stolen password hashes to move laterally within an environment, bypassing normal system access controls. Pass the hash (PtH) is a method of authenticating as a user without having access to the user's cleartext password. This method bypasses standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash. The objective of this lab is to execute a "pass the hash" attack using stolen password hashes.

□ LAB 331 – ATT&CK: Network Service Discovery **(12 mins)**

The Network Service Discovery lab assesses the learner's ability to identify the network services that are running on a server. In this lab the learner will enumerate the network services that are running on a target system. Attackers typically enumerate network services in order to discover vulnerable services that can be exploited.

□ LAB 332 – ATT&CK: Network Share Discovery **(12 mins)**

The Network Share Discovery lab assesses the learner's ability to identify the network shares on a file sharing server. In this lab the learner will enumerate the network shares on on a target system. Attackers typically enumerate shared filesystem resources in order to gain unauthorized access to data.

□ LAB 334 – ATT&CK: Create Account **(12 mins)**

The Create Account lab assesses the learner’s ability to create a user account on a compromised server. In this lab the learner will create a user account on a compromised system. Attackers usually create user accounts on compromised systems to maintain persistence and/or to gain additional privileges.

□ LAB 335 – ATT&CK: Unsecured Credentials **(12 mins)**

The Unsecured Credentials lab assesses the learner’s ability to recover unsecured credentials on a compromised system. In this lab the learner will recover unsecured credentials on a compromised system. Unsecured authentication credentials can be leveraged for additional access.

□ LAB 336 – ATT&CK: Data from Local System **(12 mins)**

The Data from Local System lab assesses the learner’s ability to recover valuable data from a compromised system. After completing this lab, the learner will understand how attackers recover valuable data from compromised systems.

□ LAB 337 – ATT&CK: Valid Accounts **(12 mins)**

The Valid Accounts lab assesses the learner’s ability to leverage compromised credentials. After completing this lab, the learner will understand how attackers leverage compromised credentials.

Learning Skill Labs

□ Information Security Fundamental Labs **(up to 90 mins per lab)**

1	Securing the pfSense Firewall
2	Implementing NAT and Allowing Remote Access
3	Implementing Common Protocols and Services
4	Examining Wireless Networks
5	Implementing Security Policies on Windows and Linux
6	Data Backups in Windows, BSD, and Linux
7	Incident Response Procedures, Forensics, and Forensic Analysis
8	Crafting and Deploying Malware Using a Remote Access Trojan (RAT)
9	Social Engineering Using SET
10	Breaking WEP and WPA and Decrypting the Traffic
11	Deep Dive in Packet Analysis - Using Wireshark and Network Miner
12	Remote and Local Exploitation
13	Patching, Securing Systems, and Configuring Anti-Virus
14	Using Active Directory in the Enterprise
15	Using Public Key Encryption to Secure Messages

□ A+ Core 2 Labs **(up to 90 mins per lab)**

1	Using Windows Features and Tools

2	Managing Files, Folders, and Disks in Windows
3	Configuring and Using Linux
4	Deploying a Windows Image
5	Configuring Windows Applications and Services
6	Using Windows Troubleshooting Tools
7	Configuring Network Connection Settings
8	Troubleshooting Network Connections
9	Configuring Shared Resources
10	Configuring Active Directory Accounts and Policies
11	Configuring Data Protection
12	Using Antivirus Software
13	Configuring Windows Backup
14	Using Windows PowerShell

□ Ethical Hacking and System Defense **(up to 90 mins per lab)**

1	Performing Reconnaissance from the WAN
2	Scanning the Network on the LAN
3	Enumerating Hosts using Wireshark, Windows, and Linux Commands
4	Remote and Local Exploitation
5	Crafting and Deploying Malware Using a Remote Access Trojan (RAT)
6	Capturing and Analyzing Network Traffic Using a Sniffer
7	Social Engineering Using SET
8	Performing a Denial of Service Attack from the WAN
9	Using Browser Exploitation to Take Over a Host’s Computer
10	Attacking Webservers from the WAN
11	Exploiting a Vulnerable Web Application
12	Performing SQL Injection to Manipulate Tables in a Database
13	Breaking WEP and WPA and Decrypting the Traffic
14	Attacking the Firewall and Stealing Data over an Encrypted Channel
15	Using Public Key Encryption to Secure Messages

□ Networking Fundamentals **(up to 90 mins per lab)**

1	Configuring Port Redirection
2	Implementing NAT and Allowing Remote Access

3	IPv4 vs IPv6 – Calculating, Configuring, and Testing
4	Network Management
5	Business Continuity - Disaster Recovery
6	Breaking WEP and WPA and Decrypting the Traffic
7	Closing Ports and Unnecessary Services
8	Implementing Security Policies on Windows and Linux
9	Network Security - Firewalls
10	Network Troubleshooting
11	TCP/IP Utilities
12	The OSI Model
13	TCP/IP Protocols – The Core Protocols
14	TCP/IP Protocols – Other Key Protocols
15	Types of Networks
16	Remote Access - RDP

□ Digital Forensics **(up to 90 mins per lab)**

1	Introduction to File Systems
2	Common Locations of Windows Artifacts
3	Hashing Data Sets
4	Drive Letter Assignments in Linux
5	The Imaging Process
6	Introduction to Single Purpose Forensic Tools
7	Introduction to Autopsy Forensic Browser
8	FAT File System
9	The NTFS File System
10	Browser Artifact Analysis
11	Communication Artifacts
12	User Profiles and the Windows Registry
13	Log Analysis
14	Memory Analysis
15	Forensic Case Capstone

□ Linux Server I: Linux Fundamentals **(up to 90 mins per lab)**

1	CentOS Server Linux Installation
2	Ubuntu Desktop Linux Installation 12.04

3	Installing Packages and Shared Libraries on CentOS and Ubuntu
4	Displaying Hardware
5	Adding a New Partition
6	Managing Filesystem Quotas
7	Booting and Restarting the System
8	Using the BASH Shell - 1
9	Using the BASH Shell - 2
10	Using the BASH Shell - 3
11	Using the BASH Shell - 4
12	Monitoring Processes
13	Working with Files
14	Managing Text Files - 1
15	Managing Text Files - 2
16	Managing Text Files - 3

Linux Server II: System Administration **(up to 90 mins per lab)**

1	Configuring X Windows in CentOS and Fedora Desktop
2	Accessibility Technologies
3	User and Group Accounts
4	System Administration Tasks - 1
5	System Administration Tasks - 2
6	System Administration Tasks - 3
7	crontab and at
8	Configuring Locale and Time Zone Settings
9	Working with Email - 1
10	Working with Email - 2
11	Basic Network Configuration
12	Basic Security Administration
13	Securing Data with Encryption on a Linux System
14	Host Security
15	BASH shell features
16	BASH Scripting
17	Working with a SQL Database

Scripting Fundamentals **(up to 90 mins per lab)**

1	Advanced Data Structure Usage
2	File I/O, String Parsing and Data Structures
3	Tuples(Arrays), Error handling and Secure Programming
4	Loops
5	Attacking and Defending Linux
6	Getting Started with Python on Ubuntu - Running from the Command Line
7	Introduction to Control Structures and Data Types
8	Getting Started with Python on Ubuntu - Writing Your First Program
9	Verifying a File Type with its Extension
10	Creating a Ping Scanner
11	Data Visualization
12	Pattern Matching
13	Extracting and Cleaning Data Using Python
14	Analysis with Kmeans
15	Inheritance

□ Network Security Fundamentals **(up to 90 mins per lab)**

1	Configuring a Windows based Firewall to Allow Incoming Traffic
2	Configuring a Linux based Firewall to Allow Incoming and Outgoing Traffic
3	Implementing Secure DHCP and DNS
4	Configuring a Linux based Firewall to Allow Outgoing Traffic
5	Configuring Access Control Lists on a Linux Based Firewall
6	Configuring a Virtual Private Network with PPTP
7	Configuring a Virtual Private Network with OpenVPN
8	Implementing RIP, RIPv2, and Securing RIP
9	Intrusion Detection using Snort
10	Writing Custom Rules
11	Host-Based Firewalls
12	Configuring RADIUS
13	Domain Security
14	Configuring a Site to Branch a Virtual Private Network
15	Closing Security Holes

□ Linux Based Security + **(up to 90 mins per lab)**

1	Configuring a VPN tunnel using the pfSense Firewall

2	Comparing and Contrasting using Clear Text Protocols
3	Linux Attack and Response
4	Log Analysis of Linux Systems with Grep and Gawk
5	Attacking and Defending Linux
6	Cracking Passwords on Linux Systems
7	Identifying & Analyzing Network Host Intrusion Detection System
8	Exploiting Shellshock
9	Vulnerability Scanning of a Linux Target
10	Encrypting Data using TrueCrypt and Attacking the TrueCypt password using TrueCrack
11	Injection Attacks using WebGoat
12	Permissions, Users, and Groups in Linux
13	Creating a Proxy Server and an SSL Certificate using the pfSense Firewall
14	Steganography

□ Pentesting and Understanding Vulnerabilities **(up to 90 mins per lab)**

1	Provisioning a Web Server
2	Exploring HTML
3	Provisioning a MySQL Database
4	Provisioning PHP
5	Dissecting the Login Process
6	SQL Injections (SQLi)
7	SQLi Vulnerability and Pentesting Steps
8	HTML Injections (HTMLi)
9	HTMLi Vulnerability and Mitigation
10	Reflected XSS
11	Reflected XSS Mitigation and URL Encoding
12	PHP Sessions and Cookies
13	Additional SCRIPT Elements
14	Session Stealing (Remote Reflected XSS)
15	Remote Reflected XSS Mitigation and URL Encoding
16	Vulnerable Forum
17	Pentesting the Forum
18	Session Stealing (Stored XSS)
19	Command Injection

20	Stateless Firewall
21	Abusing a Stateless Firewall
22	Stateful Firewall
23	Abusing a Stateful Firewall
24	IDS, SYSLOG, and NTP
25	Signature Detection and Alerting an Admin
26	IPS, SYSLOG, and NTP
27	Signature Detection and Remote Shells
28	Remote Shell: Embedding Client-side Code into a Package
29	Remote Shell Extracting Data
30	Incident Response

□ Hadoop Administration **(up to 90 mins per lab)**

1	Hadoop 1.2.1
2	Map Reduce
3	Hadoop 1.2.1 Cluster
4	Name Node Failover
5	Hadoop 2.7.3
6	Hadoop 2.7.3 Cluster

□ PC Maintenance and Repair **(up to 90 mins per lab)**

1	Examining PC Hardware
2	PC Operating Systems
3	Networking Essentials
4	Printers
5	Security Practices
6	Troubleshooting
7	Disk Maintenance and Data Recovery
8	Command Prompt Tools
9	Remote Access
10	Control Panel
11	Desktop Customization
12	Using Active Directory in the Enterprise ***
13	Data Backups in Windows, BSD, and Linux ***
14	Ubuntu Desktop Linux Installation ***

15	Domain Security ***

□ Introduction to Operating Systems **(up to 90 mins per lab)**

1	Introduction to Operating Systems
2	Computer Security Basics
3	Desktop Virtualization
4	Introduction to Windows 7
5	Introduction to Windows 8.1
6	Introduction to Windows 10
7	Supporting and Troubleshooting Windows
8	Linux on the Desktop
9	Connecting Desktops and Laptops to Networks
10	Mobile Operating Systems
11	File Management in the Cloud

□ Miscellaneous Labs **(up to 90 mins per lab)**

1	Ubuntu Desktop Linux Installation 18.04.1
2	Windows Kerberos Exploitation
3	Detecting Malware and Unauthorized Devices
4	Connect Devices to a Virtual Machine
5	Configure a Virtual Network
6	Configure and Secure Wireless Devices
7	Using Basic (Static) and Dynamic Virtual Disks and Disk Drives
8	Preserving the State of a Virtual Machine
9	Optimize the Performance of a Virtual Machine
10	Apply Drive Encryption to a Hard Drive

□ Cybersecurity Attack and Defend **(up to 90 mins per lab)**

1	Creating and Securing User Accounts
2	Network Exploitation
3	Finding Malicious Indicators
4	Static and Dynamic Malware Analysis
5	Local Operating System Exploitation
6	Investigating a Network Compromise
7	Log Analysis in Linux and Splunk

8	Network and System Monitoring
9	Hardening Windows
10	Hardening Linux
11	Windows Registry
12	Forensic Analysis of Windows Server
13	Forensic Analysis of a Windows 10 Client
14	Forensic Analysis of a Linux System
15	Using EFS
16	Using Disk Encryption
17	Using SSH and SCP
18	Using Hash Functions to Validate Data Integrity

□ Front End Web Development **(up to 90 mins per lab)**

1	Website Development Basics
2	HTML5 Basics I
3	HTML5 Basics II
4	HTML5 Basics III
5	CSS3 Basics I
6	CSS3 Basics II
7	Building a website
8	Introduction to JavaScript
9	JavaScript and HTML
10	Website Debugging

□ Security Analyst **(up to 90 mins per lab)**

1	Analyzing Output from Network Security Monitoring Tools
2	Analyzing Output from Security Appliance Logs
3	Analyzing Output from Endpoint Security Monitoring Tools
4	Analyzing Email Headers
5	Configuring SIEM Agents and Collectors
6	Analyzing, Filtering, and Searching Event Log and syslog Output
7	Collecting and Validating Digital Evidence
8	Analyzing Network-Related IoCs
9	Analyzing Host and Application IoCs
10	Observing IoCs during a Security Incident

11	Analyzing Output from Topology and Host Enumeration Tools
12	Testing Credential Security
13	Configuring Vulnerability Scanning and Analyzing Outputs
14	Assessing Vulnerability Scan Outputs
15	Performing Account and Permissions Audits
16	Configuring Network Segmentation and Security
17	Configuring and Analyzing Share Permissions
18	Assessing the Impact of Web Application Vulnerabilities
19	Optimize the Performance of a Virtual Machine
20	Analyzing Output from Cloud Infrastructure Assessment Tools

□ LogRhythm - Analyst Fundamentals **(up to 90 mins per lab)**

1	LogRhythm SIEM Familiarization
2	Analyst Use Case Walkthrough
3	Complete Use Case
4	Ransomware Injection
5	Botnet Detection
6	Reducing Downtime Caused by an Outage
7	Comply with Acceptable Use Policy

□ Cyber Challenge Range **(19 challenges as fast as the user can complete them)**

1	Challenge - Reconnaissance	The goal of the lab will be to find the outer firewall and perform a scan on it to find vulnerable services.
2	Challenge - Cracking the Perimeter	Cracking the Perimeter to find a way to get through the firewall to the DMZ behind it. Once there, find a way to pivot to the DEV network.
3	Challenge - Infiltration	The object of this next challenge is to gain access to the User network. You will have to retrace the steps from the previous labs to get into the DEV1 box where the challenge begins.
4	Challenge - Situational Awareness	Situational Awareness - The purpose of this lab will be to infiltrate the Admin network with the credentials discovered in the previous lab. To aid in this an implant was set up that will call out every 15 seconds to port 7979 on the attacking machine. Awareness of the Admin network segment will aid in choosing a target quickly.

5	Challenge - Carving Disk Images	Carving disk images - The goal of this lab will be to gain Local Administrative access to the Admin Workstation via information found in a backup image. An implant was set up on the backup server to call out to the attacking machine on port 4321 every 15 seconds. It will allow direct access to it and the admin network.
6	Challenge - Kerberoasting	Kerberoasting - The goal of this lab will be to create a golden ticket for the fakecorp.com domain administrator and to prove that it works.
7	Challenge - Locating the Crown Jewels	Locating the Crown Jewels - The goal of this lab is to locate this organizations 'Crown Jewels'. This is the accounting system that contains sensitive data.
8	Challenge - Exfiltrating Data	Exfiltrating data - The purpose of this lab is to pull the data for the bank accounts from the mainframe server.
9	Challenge - Covering your Tracks	Covering your tracks - During a pentest or red team exercise, it is often necessary to erase traces of your presence, to throw the opposition off. The easiest way is to erase the logs outright. Also, as a scenario wraps up you want to leave a way back in with the creation of new accounts on several systems and/or the domain.
10	Challenge - Maintaining Persistence	Maintaining persistence - The goal of the final lab is to leave a back door into the network. The choice of a print server or printer is a wise choice since they are often overlooked. We established a temporary backdoor that is installed on the multi-function to allow for ready access.
11	Challenge - Host Based Forensics	A Dell CPi notebook was seized on 20th September 2004 with the serial number: VLQLW. An external ‘home made’ antenna (802.11b) was also seized. It is suspected that the notebook was used for hacking purposes, however it cannot be directly linked to the suspect - Greg Schardt. He is alleged to haveit is parked his vehicle within range of Wireless Access Points (such as Starbucks and T-Mobile Hotspots) where he would then intercept internet traffic in an attempt to get credit card numbers, usernames & passwords.
12	Challenge - Mobile Forensics	Suspect Josh Hickman is suspected is stalking an individual during the month of

		February 2020. You are taken with examining the Pixel 3 image and seeing if photographs of scouted locations exist on the image and determining when and where these images were taken and verify the identity of the suspect and that there is reasonable suspicion that this individual was scouting the locations during this period. All times are in Pacific Standard Time (PST)
13	Challenge - Malware Analysis in Windows	The objective of this lab is to perform a quick analysis on some basic malware samples and determine program flow and the components, calls and libraries it accesses in its operation. This will allow the student to help determine the purpose and motivation behind these samples and locate indicators of compromise (IOCs) that can be used for later detections.
14	Challenge - Reverse Engineering in Linux	To reverse engineer and disassemble two Linux binaries. Having the skill to disassemble suspect binaries is an essential task in cyber security. This lab will challenge the students' ability to dissect a Linux binary using the built-in GNU Debugger (GDB).
15	Challenge - Binary Exploitation	You are contracted to work as a penetration tester and have been given have gained user access to a client system. Your objective is to compromise this system by exploitation access code left on the system. This access code will grant simulated access and contain a flag.
16	Challenge - Searching Through Evidence	A system was seized in a ransomware case. It is believed that the suspect has hidden bitcoin wallet addresses inside text files and has used encoding methods to hide these addresses. There is believed to be five accounts hidden in these files. Please search for these addresses. This data is needed to recover victim funds.
17	Challenge - File Recovery	An unknown file was found on a suspect's cell phone. As part of the forensics team, it is your responsibility to recover the file to a useful state. Use the tools available in the Flare VM to identify the file and then recover it.
18	Challenge - Recreating an Attack	As a forensic examiner, you have recreated an exploited box and want to determine

		what kind of exploit was used to compromise it.
19	Challenge - Debugging Existing Python Code	As a penetration tester, you may have to take someone’s proof of concept code and get it to run properly on your system. The goal of this challenge is to take existing code, debug it, and get it working.

AI/ML Starter Labs **(19 challenges as fast as the user can complete them)**

1	Introduction to ChatGPT and Generative AI
2	Use ChatGPT to Plan and Execute a Phishing Simulation for an Organization
3	Use ChatGPT for Social Media Threat Hunting
4	Use ChatGPT to create filters in Splunk
5	Use ChatGPT to detect security vulnerabilities in code
6	Use ChatGPT to write cybersecurity automation scripts
7	Use ChatGPT to write a security policy for an organization
8	Use ChatGPT to develop social engineering training