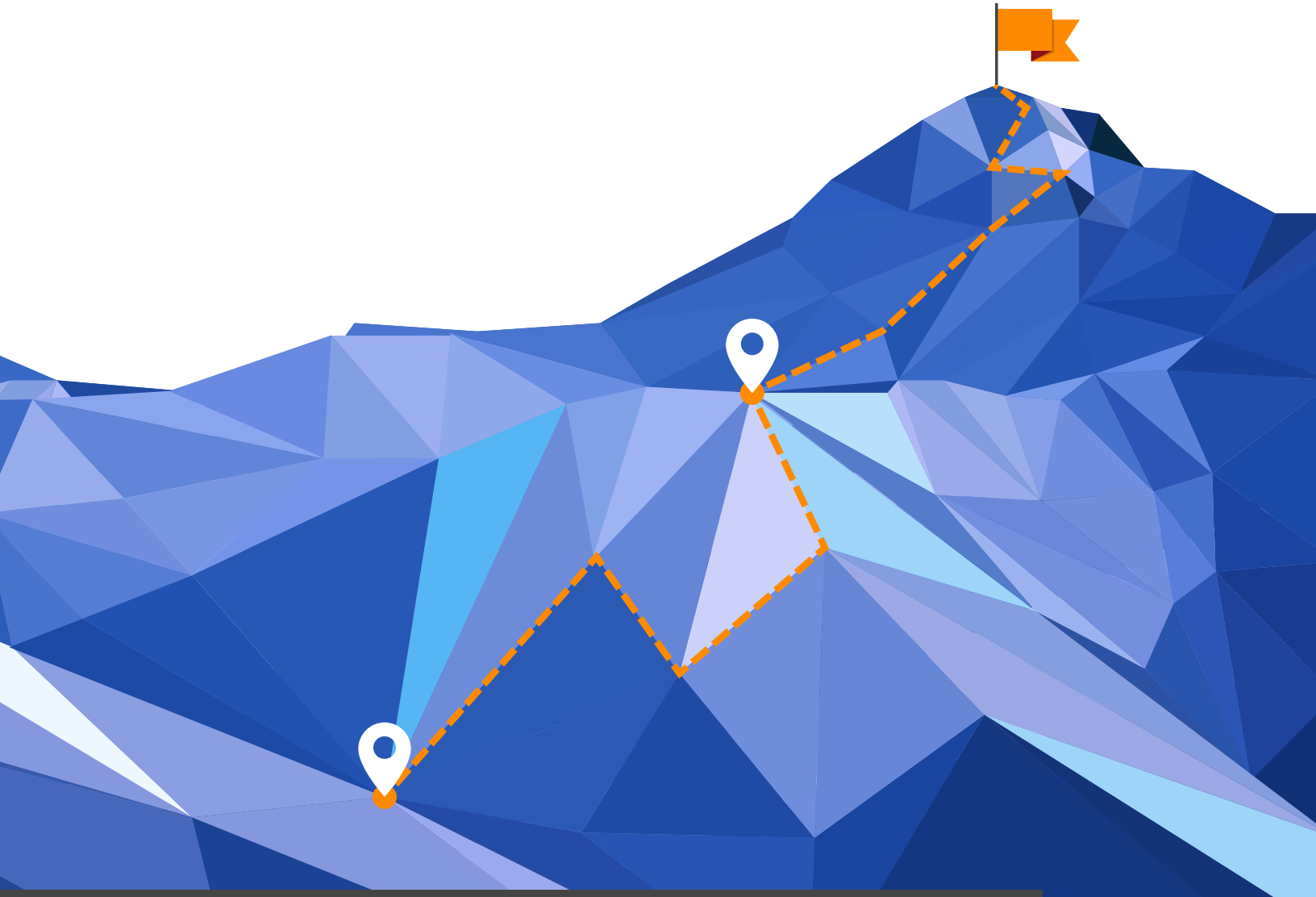

LEARNING PATHS

Software Security Role-Based Curriculum

cmd+ctrl

Contents

CSC 101 – Secure Developer – Core

The Secure Developer – Core Learning Path introduces application security's fundamental and primary drivers. The curriculum provides individuals with an understanding of the importance of secure software development while preparing them to perform at the organizational level. Learners will gain in-depth knowledge of security principles, attacks, tools, and processes to develop secure software. By introducing the OWASP Top 10, learners are prepared to identify the most critical web application security risks, appropriately address those vulnerabilities, and prevent software flaws that enable cyberattacks.

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

- Define the value of having secure applications
- Integrate secure software development practices into all phases of the software development lifecycle
- Explain the anatomy of an application attack
- Apply best practices to protect all components of the software
- Identify and mitigate the most common application security risks
- Implement a security strategy based on your organization's risk
- Produce well-secured software

NOTE: This Learning Path is considered principal to all Elite Secure Developer Learning Paths. Learn and Skill labs are elective training modules that help transform concepts into tangible skills through hands-on, realistic examples of real-world threat scenarios.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

15 Courses

10 Labs

6 Hours

7 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- AWA 106 – Building Secure Software: Challenges in Application Security
- AWA 107 – Building Secure Software: Foundations & Best Practices
- AWA 108 – Building Secure Software: A Guide to Software Integration, Testing, and Deployment
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Mitigating Insecure Deserialization
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components

- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- LAB 113 – Identifying Cryptographic Failures
- LAB 115 – Identifying Reflective XSS
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 121 – Identifying Vulnerable and Outdated Components
- LAB 123 – Identifying Vertical Privilege Escalation
- LAB 127 – Identifying Security Logging and Monitoring Failures
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 137 – Identifying Improper Authorization

CSC 201 – Secure Developer – Advanced

The Secure Developer – Advanced Learning Path explores different models, standards, frameworks, and security concepts that you can use to understand security issues and improve the security posture of your applications. The curriculum provides individuals with an understanding of how to ensure security is part of software design. Learners will gain in-depth knowledge of security practices that must be considered within every phase of the development lifecycle to help secure software applications and data. By introducing the DevSecOps philosophies, learners are prepared to focus on time saving but effective techniques that maximize security resources all while shortening system development lifecycles and providing continuous delivery of high-quality software.

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

- Use NIST and MITRE ATT&CK security frameworks to identify and categorize potential threats
- Identify and apply relevant cryptographic technologies to secure applications and data
- Apply techniques to remove architecture weak spots and avoid vulnerability propagation
- Implement a zero-trust architecture
- Create a threat model for application scenarios
- Manage identities, privileges, and secrets securely
- Understand, create, and articulate security requirements as part of a software requirement document
- Use NIST and MITRE ATT&CK security frameworks to identify, categorize, and respond to potential threats
- Determine which types of automated tests should be performed at various stages of the SDLC

NOTE: This Learning Path is considered principal to all Elite Secure Developer Learning Paths. Learn and Skill labs are elective training modules that help transform concepts into tangible skills through hands-on, realistic examples of real-world threat scenarios.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

18 Courses

3 Labs

9 Hours

10 CPE Credits

Advanced

- API 251 – Implementing Web Application and API Protection (WAAP)
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- DES 204 – The Role of Cryptography in Application Development
- DES 212 – Architecture Risk Analysis and Remediation
- DES 250 – Secure Software Acceptance and Deployment
- DES 311 – Creating Secure Application Architecture
- DES 361 – Mitigating LCNC (Low-Code/No-Code) Account Impersonation
- DES 362 – Mitigating LCNC (Low-Code/No-Code)) Authorization Misuse

- DES 364 – Mitigating LCNC (Low-Code/No-Code) Authentication and Secure Communication Failures
- DSO 212 – Fundamentals of Zero Trust Security
- DSO 302 – Automated Security Testing
- DSO 307 – Secure Secrets Management
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- ENG 212 – Implementing Secure Software Operations
- ENG 312 – How to Perform a Security Code Review
- ENG 320 – Using the Software Composition Analysis (SCA) to Secure Open Source Components
- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 301 – 322 – Elite Secure Developer

The Elite Secure Developer Learning Paths include a variety of security courses for those who design, develop, and host applications across various platforms using different programming languages. The learning paths are designed to provide a working knowledge for developing solid and secure applications as well as recognizing and remediating common software security vulnerabilities.

The Elite Secure Developer learning paths cover key application security concepts, including:

- Best practices for designing, developing, and testing applications using common standards and frameworks
- Unique programming language constructs
- Identifying common application threats and risks
- Creating threat models
- Platform configuration
- Core implementation practices
- Identifying and mitigating vulnerabilities
- Platform-specific secure coding best practices and defensive coding practices
- Protecting data using secure coding best practices
- Cloud computing characteristics, service and deployment models, and regulatory requirements

NOTE: Secure Developer – Core and Advanced Learning paths are considered principal to all Elite Secure Developer Learning Paths. All Learn and Skill labs are elective training modules that help transform concepts into tangible skills through hands-on, realistic examples of real-world threat scenarios.

(See next page for Course Matrix)

LEARNING PATH DETAILS

	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
COURSES	15	10	11	12	11	18	6	4	10	5	14	12	13	13	9	15	13	11	5	8	8	10	23
LABS	21	46	76	12	0	7	25	22	24	53	23	3	21	31	0	41	0	0	4	0	0	20	30
DURATION (Hours)	8	10	17	9	4	10	8	5	9	11	9	6	9	11	4	14	5	4	3	4	4	7	14
CPE CREDITS	13	15	21	9	6	11	9	6	9	13	10	7	11	13	5	16	6	5	3	4	5	8	17

COURSE TITLE

AWA 106. Building Secure Software: Challenges in Application Security																	✓						
AWA 107. Building Secure Software: Foundations & Best Practices																	✓						
AWA 108. Building Secure Software: A Guide to Software Integration, Testing, and Deployment																	✓						
API 210. Mitigating APIs Lack of Resources & Rate Limiting	✓																						✓
API 211. Mitigating APIs Broken Object Level Authorization	✓																						✓
API 213. Mitigating APIs Mass Assignment	✓																						✓
API 214. Mitigating APIs Improper Asset Management	✓																						✓

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
API 250. Controlling Access to the Kubernetes API																							✓
API 251. Implementing Web Application and API Protection (WAAP)					✓																		
API 351. Securing Kubernetes in the Build and Release Stage					✓																		
COD 110. Fundamentals Secure Mobile Development				✓		✓									✓					✓	✓		✓
COD 141. Fundamentals of Database Security																		✓					
COD 152. Fundamentals of Secure Cloud Development					✓																		
COD 160. Fundamentals of Secure Embedded Software Development						✓																	
COD 201. Secure C Encrypted Network Communications						✓													✓				
COD 206. Creating Secure C++ Code						✓						✓											
COD 207. Communication Security in C++						✓						✓											
COD 214. Creating Secure GO Applications		✓																					✓
COD 215. Mitigating .NET Application Vulnerabilities (NEW)									✓								✓					✓	

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
COD 216. Leveraging NET Framework Code Access Security (CAS)									✓														
COD 219. Creating Secure Code SAP ABAP Foundations							✓																
COD 241. Creating Secure Oracle Database Applications	✓													✓				✓					✓
COD 242. Creating Secure SQL Server and Azure SQL Database Applications																	✓						
COD 245. Securing NoSQL Cloud Databases	✓				✓																		
COD 246. PCI DSS Requirement 3: Protecting Stored Cardholder Data																		✓					
COD 247. PCI DSS Requirement 3: Encrypting Transmission of Cardholder Data																		✓					
COD 248. PCI DSS Requirement 6: Develop & Maintain Secure Systems & Applications																		✓					
COD 249. PCI DSS Requirement 11: Regularly Test Security Systems and Processes																		✓					
COD 251. Defending AJAX-Enabled Web Applications	✓										✓		✓	✓		✓		✓					✓
COD 254. Creating Secure Azure Applications																	✓						

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
COD 255. Creating Secure Code Web API Foundations	✓										✓	✓	✓	✓		✓						✓	✓
COD 256. Creating Secure Code Ruby on Rails Foundations	✓										✓												✓
COD 257. Creating Secure Python Web Applications								✓															✓
COD 258. Creating Secure PHP Web Applications			✓							✓			✓	✓									
COD 259. Node.js Threats and Vulnerabilities							✓			✓	✓		✓	✓		✓							
COD 261. Threats to Scripts				✓		✓							✓		✓				✓	✓	✓		
COD 262. Fundamentals of Shell and Interpreted Language Security												✓	✓										
COD 265. Secure Python Scripting								✓															✓
COD 267. Securing Python Microservices					✓			✓															✓
COD 268. Mitigating TypeScript Application Vulnerabilities		✓	✓	✓						✓				✓									
COD 284. Secure Java Coding							✓																
COD 285. Developing Secure Angular Applications		✓	✓													✓							✓
COD 286. Creating Secure React User Interfaces		✓																					✓

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
COD 287. Java Application Server Hardening	✓	✓	✓				✓									✓							✓
COD 288. Java Public Key Cryptogrpahy	✓	✓	✓				✓									✓							✓
COD 289. Securing Java Spring APIs							✓																
COD 301. Secure C Buffer Overflow Mitigations						✓													✓				
COD 303. Common C Vulnerabilities and Attacks						✓													✓				
COD 304. Principles of C++ Memory Safety						✓						✓											
COD 305. C++ Secure Memory Management						✓						✓											
COD 306. C++ Memory Safety: Debugging Tools and Techniques						✓						✓											
COD 307. Protecting Data in C++						✓						✓											
COD 308. Common ASP.NET Vulnerabilities and Attacks									✓	✓						✓						✓	
COD 309. Securing ASP.NET MVC Applications (UPDATED)									✓	✓						✓	✓					✓	
COD 310. Securing ASP.NET Core Applications			✓						✓													✓	

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
COD 315. Preventing Vulnerabilities in iOS Code in Swift				✓											✓					✓			
COD 316. Creating Secure iOS Code in Objective C				✓											✓					✓			
COD 317. Protecting Data on iOS in Swift				✓											✓					✓			
COD 318. Protecting Data on Android in Java (UPDATED)				✓																	✓		
COD 319. Preventing Vulnerabilities in Android Code in Java (UPDATED)				✓																	✓		
COD 321. Protecting C# from Integer Overflows and Canonicalization Issues									✓													✓	
COD 322. Protecting C# from SQL Injection									✓													✓	
COD 323. Using Encryption with C#									✓													✓	
COD 324. Protecting C# from XML Injection									✓													✓	
COD 325. Protecting Data in C# for .NET Core			✓						✓													✓	
COD 352. Creating Secure JavaScript and jQuery Code	✓	✓	✓								✓					✓							✓
COD 361. HTML5 Security Threats											✓		✓	✓		✓							

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
COD 362. HTML5 Built-In Security Features											✓		✓	✓		✓							
COD 363. Securing HTML5 Data											✓		✓	✓		✓							
COD 364. Securing HTML5 Connectivity											✓		✓	✓		✓							
COD 383. Protecting Java Backend Services	✓	✓	✓													✓							✓
CYB 213. Generative AI Privacy & Cybersecurity Risk (NEW)		✓																					
DES 151. Fundamentals of the PCI Secure SLC Standard																		✓					
DES 203. Cryptographic Components: Randomness, Algorithms, & Key Management												✓											
DES 206. Meeting Cloud Governance and Compliance Requirements					✓																		
DES 207. Mitigating OWASP API Security Top 10	✓										✓	✓	✓	✓				✓	✓				✓
DES 209. Authentication and Lifecycle Management																		✓					
DES 215. Defending Infrastructure					✓																		
DES 216. Protecting Cloud Infrastructure					✓																		

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
DES 218. Protecting Microservices, Containers, and Orchestration					✓																		
DES 219. Securing Google's Firebase Platform												✓	✓	✓									
DES 255. Securing the IoT Update Process				✓		✓									✓					✓	✓		
DES 260. Fundamentals of IoT Architecture and Design				✓		✓									✓					✓	✓		
DES 261. Securing Serverless Environments						✓																	
DES 270. Mitigating OWASP Mobile Top 10 Risks				✓											✓						✓		
DES 311. Creating Secure Application Architecture			✓																				
DES 312. Protecting Cardholder Data																		✓					
DES 313. Hardening a Kubernetes Cluster					✓	✓																	
DES 314. Hardening the Docker Engine					✓	✓																	
DSO 304. Securing API Gateways in a DevSecOps Framework	✓	✓									✓		✓	✓		✓							✓
DSO 306. Implementing Infrastructure as Code								✓			✓												✓

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
DSO 307. Secure Secrets Management			✓																				
ENG 112. Essential Access Controls for Mobile Devices				✓											✓					✓	✓		
ENG 191. Introduction to the Microsoft SDL																	✓						
ENG 192. Implementing the MS SDL Optimization Model																	✓						
ENG 193. Implementing the Agile MS SDL																	✓						
ENG 194. Implementing MS SDL Line of Business																	✓						
ENG 195. Implementing the MS SDL Threat Modeling Tool																	✓						
ICS 310. Protecting Information and System Integrity in Industrial Control System Environments						✓																	
SDT 301. Testing for Injection											✓												
SDT 303. Testing for Cryptographic Failures											✓												

COURSE TITLE

LABS

	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 122. Identifying Insecure APIs	✓	✓																					✓
LAB 132. Identifying Exposed Services	✓	✓																					✓
LAB 201. Defending Java Applications Against Canonicalization			✓				✓									✓							
LAB 202. Defending Python Applications Against Canonicalization			✓					✓		✓													
LAB 203. Defending C# Applications Against Canonicalization			✓						✓													✓	
LAB 204. Defending Node.js Applications Against Canonicalization		✓	✓							✓	✓		✓	✓		✓							
LAB 205. Defending Java Applications Against XPath Injection			✓				✓									✓							
LAB 206. Defending Python Applications Against XPath Injection			✓					✓		✓													
LAB 207. Defending Node.js Applications Against XPath Injection		✓	✓							✓	✓		✓	✓		✓							
LAB 208. Defending C# Applications Against XPath Injection			✓						✓													✓	
LAB 211. Defending Java Applications Against Credentials in Code Medium							✓																

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 212. Defending Python Applications Against Credentials in Code Medium								✓															
LAB 213. Defending Node.js Applications Against Credentials in Code Medium		✓								✓													✓
LAB 214. Defending C# Applications Against Credentials in Code Medium									✓														
LAB 215. Defending Java Applications Against Business Logic Error for Input Validation							✓																
LAB 216. Defending Python Applications Against Business Logic Error for Input Validation								✓															
LAB 217. Defending Node.js Applications Against Business Logic Error for Input Validation		✓								✓													✓
LAB 218. Defending C# Applications Against Business Logic Error for Input Validation									✓														
LAB 220. Defending Against Hard-Coded Secrets (HTML5)			✓																				
LAB 221. Defending C# Against SQL Injection									✓													✓	

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 222. Defending Python Against SQL Injection								✓		✓													
LAB 223. Defending Node.js Against SQL Injection		✓	✓							✓	✓		✓	✓		✓							✓
LAB 224. Defending Java Applications Against Forceful Browsing							✓																
LAB 225. Defending Python Applications Against Forceful Browsing								✓															
LAB 226. Defending Node.js Applications Against Forceful Browsing		✓								✓													✓
LAB 227. Defending C# Applications Against Forceful Browsing									✓														
LAB 228. Defending Java Applications Against Weak AES ECB Mode Encryption			✓				✓									✓							
LAB 229. Defending Java Applications Against Weak PRNG			✓				✓									✓							
LAB 230. Defending Java Against Cross-Site Scripting (XSS)			✓				✓									✓							
LAB 231. Defending Python Against Cross-Site Scripting (XSS)								✓		✓													
LAB 232. Defending C# Against Cross-Site Scripting (XSS)									✓													✓	

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 233. Defending Node.js Against Cross-Site Scripting (XSS)		✓	✓							✓	✓		✓	✓		✓							✓
LAB 234. Defending Java Applications Against Parameter Tampering			✓				✓									✓							
LAB 235. Defending Java Applications Against Plaintext Password Storage			✓				✓									✓							
LAB 236. Defending Java Applications Against Sensitive Information in Error Messages			✓				✓									✓							
LAB 237. Defending Java Against SQL Injection			✓				✓									✓							
LAB 238. Defending C# Applications Against Weak AES ECB Mode Encryption									✓													✓	
LAB 239. Defending C# Applications Against Weak PRNG									✓													✓	
LAB 240. Defending Java Against ExternalXML Entity Vulnerabilities			✓				✓									✓							
LAB 241. Defending C# Against ExternalXML Entity Vulnerabilities									✓													✓	
LAB 242. Defending Node.js Against ExternalXML Entity Vulnerabilities		✓	✓							✓	✓		✓	✓		✓							✓
LAB 243. Defending Python Against ExternalXML Entity Vulnerabilities								✓		✓													

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 244. Defending Java Against Security Misconfiguration			✓				✓									✓							
LAB 245. Defending Node.js Applications Against Plaintext Password Storage		✓	✓							✓	✓		✓	✓		✓							✓
LAB 246. Defending Node.js Applications Against Weak AES ECB Mode Encryption		✓	✓							✓	✓		✓	✓		✓							✓
LAB 247. Defending Node.js Applications Against Weak PRNG		✓	✓							✓	✓		✓	✓		✓							✓
LAB 248. Defending Node.js Applications Against Parameter Tampering		✓	✓							✓	✓		✓	✓		✓							✓
LAB 249. Defending Python Applications Against Plaintext Password Storage								✓		✓													
LAB 250. Defending C# Applications Against Parameter Tampering									✓													✓	
LAB 251. Defending C# Applications Against Plaintext Password Storage									✓													✓	
LAB 252. Defending Python Applications Against Weak AES ECB Mode Encryption								✓		✓													
LAB 253. Defending Python Applications Against Weak PRNG								✓		✓													

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 254. Defending Python Applications Against Parameter Tampering								✓		✓													
LAB 260. Defending C# Applications Against Sensitive Information in Error Messages									✓													✓	
LAB 261. Defending Python Applications Against Sensitive Information in Error Messages								✓		✓													
LAB 262. Defending Node.js Applications Against Sensitive Information in Error Messages		✓	✓							✓	✓		✓	✓		✓							✓
LAB 263. Defending Java Applications Against Sensitive Information in Log Files			✓				✓									✓							
LAB 264. Defending Python Applications Against Sensitive Information in Log Files								✓		✓													
LAB 265. Defending Node.js Applications Against Sensitive Information in Log Files		✓	✓							✓	✓		✓	✓		✓							✓
LAB 266. Defending C# Applications Against Sensitive Information in Log Files									✓													✓	

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 267. Defending Java Applications Against Deserialization of Untrusted Data			✓				✓									✓							
LAB 268. Defending Python Applications Against Deserialization of Untrusted Data								✓		✓													
LAB 269. Defending Node.js Applications Against Deserialization of Untrusted Data		✓	✓							✓	✓		✓	✓		✓							✓
LAB 270. Defending C# Applications Against Deserialization of Untrusted Data									✓													✓	
LAB 271. Defending Java Applications Against SSRF			✓				✓									✓							
LAB 272. Defending Python Applications Against SSRF								✓		✓													
LAB 273. Defending Node.js Applications Against SSRF		✓	✓							✓	✓		✓	✓		✓							✓
LAB 274. Defending C# Applications Against SSRF									✓														
LAB 275. Defending Java Applications Against Command Injection			✓				✓									✓							
LAB 276. Defending Python Applications Against Command Injection								✓		✓													

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 277. Defending Node.js Applications Against Command Injection		✓	✓							✓	✓		✓	✓		✓							✓
LAB 278. Defending C# Applications Against Command Injection									✓													✓	
LAB 279. Defending Java Applications Against Dangerous File Upload			✓				✓									✓							
LAB 280. Defending Python Applications Against Dangerous File Upload								✓		✓													
LAB 281. Defending Node.js Against Dangerous File Upload		✓	✓							✓	✓		✓	✓		✓							✓
LAB 282. Defending C# Applications Against Dangerous File Upload									✓													✓	
LAB 283. Defending Java Applications Against RegEx DoS			✓				✓									✓							
LAB 284. Defending Python Applications Against RegEx DoS								✓		✓													
LAB 285. Defending Node.js Applications Against RegEx DoS		✓	✓							✓	✓		✓	✓		✓							✓
LAB 286. Defending C# Applications Against RegEx DoS									✓													✓	
LAB 287. Defending Java Applications Against Null Pointer Dereference			✓				✓									✓							

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 288. Defending C# Applications Against Null Pointer Dereference									✓													✓	
LAB 289. Defending Java Applications Against Path Traversal			✓				✓									✓							
LAB 290. Defending Python Applications Against Path Traversal								✓		✓													
LAB 291. Defending Node.js Applications Against Path Traversal		✓	✓							✓	✓		✓	✓		✓							✓
LAB 292. Defending C# Applications Against Path Traversal									✓													✓	
LAB 293. Defending Java Applications Against Integer Overflow			✓				✓									✓							
LAB 294. Defending C# Applications Against Integer Overflow									✓													✓	
LAB 301. Defending Java Applications Against Open Redirect			✓				✓				✓		✓			✓							
LAB 302. Defending Python Applications Against Open Redirect			✓					✓		✓	✓												
LAB 303. Defending C# Applications Against Open Redirect			✓						✓													✓	
LAB 304. Defending Node.js Applications Against Open Redirect		✓	✓							✓	✓		✓	✓		✓							
LAB 305. Defending Java Applications Against Weak Password Reset			✓				✓				✓		✓			✓							

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 306. Defending Python Applications Against Weak Password Reset			✓					✓		✓	✓												
LAB 307. Defending C# Applications Against Weak Password Reset			✓						✓													✓	
LAB 308. Defending Node.js Applications Against Weak Password Reset		✓	✓							✓	✓		✓	✓		✓							✓
LAB 309. Defending TypeScript Applications Against Unrestricted Upload of File with Dangerous Type		✓	✓	✓						✓				✓									
LAB 314. Defending TypeScript Applications Against SSRF		✓	✓	✓						✓				✓									
LAB 316. Defending TypeScript Applications Against Hard-coded Credentials		✓	✓	✓						✓				✓									
LAB 320. Defending TypeScript Applications Against Code Injection		✓	✓	✓						✓				✓									
LAB 325. Defending TypeScript Applications Against CSRF		✓	✓	✓						✓				✓									
LAB 326. Defending TypeScript Applications Against Path Traversal		✓	✓	✓						✓				✓									
LAB 327. Defending C Applications Against Path Traversal						✓													✓				

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 328. Defending C++ Applications Against Path Traversal						✓						✓											
LAB 329. Defending Go Applications Against SSRF	✓		✓																				✓
LAB 333. Defending Go Applications Against Hard-coded credentials.	✓		✓																				✓
LAB 338. Defending Go Applications Against CSRF	✓		✓																				✓
LAB 339. Defending Go Applications Against Path Traversal	✓		✓																				✓
LAB 340. Defending C Applications Against Use After Free						✓													✓				
LAB 341. Defending C ++ Applications Against Use After Free						✓						✓											
LAB 342. Defending TypeScript Applications Against Command Injection		✓	✓	✓						✓				✓									
LAB 343. Defending GO Applications Against Command Injection	✓		✓																				✓
LAB 344. Defending TypeScript Applications Against Incorrect Authorization.		✓	✓	✓						✓				✓									
LAB 345. Defending GO Applications Against Incorrect Authorization.	✓		✓																				✓

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 346. Defending TypeScript Applications Against Deserialization of Untrusted Data.		✓	✓	✓						✓				✓									
LAB 347. Defending C Applications Against Null Pointer Dereference.						✓													✓				
LAB 348. Defending C++ Applications Against Null Pointer Dereference						✓						✓											
LAB 349. Defending TypeScript Applications Against SQL Injection		✓	✓	✓						✓				✓									
LAB 350. Defending Go Applications Against SQL Injection	✓		✓																				✓
LAB 351. Defending TypeScript Applications Against Cross-Site Scripting		✓	✓	✓						✓				✓									
LAB 352. Defending Go Applications Against Cross-Site Scripting	✓		✓																				✓
LAB 353. Defending TypeScript Applications Against Improper Authentication		✓	✓	✓						✓				✓									
LAB 354. Defending Go Applications Against Improper Authentication	✓		✓																				✓
LAB 355. Defending C Applications Against Stack-based Buffer Overflow						✓													✓				

COURSE TITLE	Back-End	Front-End	Web	Mobile	Cloud	IoT & Embedded	Java	Python	C#	Node.js	Ruby on Rails	C++	PHP	JavaScript	iOS	HTML5	Microsoft SDL	PCI	C	Swift	Android	.NET	Golang
LAB 356. Defending Python APIs from Broken Object Level Authorization	✓	✓	✓																				
LAB 357. Defending Python APIs from Broken Authentication	✓	✓																					
LAB 358. Defending Python APIs from Broken Object Property Level Authorization	✓	✓																					
LAB 359. Defending Python APIs from Unrestricted Resource Consumption	✓	✓																					
LAB 360. Defending Python APIs from Broken Function Level Authorization	✓	✓	✓																				
LAB 361. Defending Python APIs from Unrestricted Access to Sensitive Business Flows	✓	✓																					
LAB 362. Defending Python APIs from Server Side Request Forgery	✓	✓	✓																				
LAB 363. Defending Python APIs from Security Misconfiguration	✓	✓	✓																				
LAB 364. Defending Python APIs from Improper Inventory Management	✓	✓																					
LAB 365. Defending Python APIs from Unsafe Consumption of APIs	✓	✓	✓																				

CSC 323 – Secure DevOps Practitioner

The Secure DevOps Practitioner Learning Path includes a variety of security courses designed for those who work closely with Software Engineers to help them deploy and operate various systems. The curriculum provides teams with a solid foundation of security features necessary to automate and streamline operations and processes while keeping security top of mind. Learners will apply best practices to develop new features and write scripts across various technologies.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

55 Courses

23 Labs

25 Hours

30 CPE Credits

Core

- AWA 106 – Building Secure Software: Challenges in Application Security
- AWA 107 – Building Secure Software: Foundations & Best Practices
- AWA 108 – Building Secure Software: A Guide to Software Integration, Testing, and Deployment
- DES 101 – Fundamentals of Secure Architecture
- DES 151 – Fundamentals of the PCI Secure SLC Standard
- ENG 123 – Essential Security Engineering Principles
- ENG 124 – Essential Application Protection
- ENG 125 – Essential Data Protection
- TST 101 – Fundamentals of Security Testing

Advanced

- API 210 – Mitigating APIs Lack of Resources & Rate Limiting
- API 211 – Mitigating APIs Broken Object Level Authorization
- API 213 – Mitigating APIs Mass Assignment
- API 214 – Mitigating APIs Improper Asset Management
- API 250 – Controlling Access to the Kubernetes API
- COD 252 – Securing Google Platform Applications & Data
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 209 – Authentication and Lifecycle Management
- DES 214 – Securing Infrastructure Architecture
- DES 215 – Defending Infrastructure
- DES 216 – Protecting Cloud Infrastructure
- DES 218 – Protecting Microservices, Containers, and Orchestration
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 261 – Securing Serverless Environments

- DSO 201 – Fundamentals of Secure DevOps
- DSO 211 – Identifying Threats to Containers in a DevSecOps Framework
- DSO 212 – Fundamentals of Zero Trust Security
- DSO 253 – DevSecOps in the AWS Cloud
- DSO 254 – DevSecOps in the Azure Cloud
- DSO 256 – DevSecOps in the Google Cloud Platform
- ENG 205 – Fundamentals of Threat Modeling
- ENG 251 – Risk Management Foundations
- TST 202 – Penetration Testing Fundamentals
- TST 205 – Performing Vulnerability Scans
- TST 206 – ASVS Requirements for Developers
- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization
- LAB 329 – Defending Go Applications Against SSRF
- LAB 333 – Defending Go Applications Against Hard-coded credentials
- LAB 338 – Defending Go Applications Against CSRF
- LAB 339 – Defending Go Applications Against Path Traversal
- LAB 343 – Defending Go Applications Against Command Injection
- LAB 345 – Defending Go Applications Against Incorrect Authorization
- LAB 350 – Defending Go Applications Against SQL Injection
- LAB 352 – Defending Go Applications Against Cross-Site Scripting
- LAB 354 – Defending Go Applications Against Improper Authentication
- LAB 357 – Defending Python APIs from Broken Authentication
- LAB 359 – Defending Python APIs from Unrestricted Resource Consumption
- LAB 363 – Defending Python APIs from Security Misconfiguration
- LAB 364 – Defending Python APIs from Improper Inventory Management

Elite

- API 351 – Securing Kubernetes in the Build and Release Stage
- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- CYB 311 – Threat Analysis with AI
- DES 313 – Hardening a Kubernetes Cluster
- DES 314 – Hardening the Docker Engine
- DES 361 – Mitigating LCNC (Low-Code/No-Code) Account Impersonation
- DES 362 – Mitigating LCNC (Low-Code/No-Code)) Authorization Misuse
- DES 364 – Mitigating LCNC (Low-Code/No-Code) Authentication and Secure Communication Failures
- DSO 301 – Orchestrating Secure System and Service Configuration
- DSO 302 – Automated Security Testing
- DSO 303 – Automating Security Updates

- DSO 304 – Securing API Gateways in a DevSecOps Framework
- DSO 305 – Automating CI/CD Pipeline Compliance
- DSO 306 – Implementing Infrastructure as Code
- DSO 307 – Secure Secrets Management
- ENG 312 – How to Perform a Security Code Review
- ENG 351 – Preparing the Risk Management Framework

CSC 324 – Ethical Hacker

The Ethical Hacker Learning Path includes a variety of security courses geared towards individuals responsible for assessing systems and networks within the network environment and identifying where those systems/networks deviate from acceptable configurations, enclave policy, or local policy. The curriculum provides a solid foundation of the skills needed to measure the effectiveness of defense-in-depth architecture against known vulnerabilities and verify and improve the security of a company's computer systems.

After completing this learning path learners will have the knowledge and skills necessary to:

- Analyze cyber defense policies and configurations
- Evaluate compliance with regulations and organizational directives
- Conduct and/or support authorized penetration testing on enterprise network assets
- Deploy cyber defense audit toolkit to support cyber defense audit missions
- Maintain knowledge of applicable cyber defense policies, regulations, and compliance documents specifically related to cyber defense auditing
- Prepare audit reports that identify technical and procedural findings and provide recommended remediation strategies/solutions
- Conduct required reviews as appropriate within an environment
- Perform evaluation of technology and of people and operations risk
- Perform vulnerability assessments of relevant technology focus areas
- Make recommendations regarding the selection of cost-effective security controls to mitigate risk

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

45 Courses

59 Labs

27 Hours

32 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- COD 141 – Fundamentals of Database Security
- DES 101 – Fundamentals of Secure Architecture
- ENG 110 – Essential Account Management Security
- ENG 114 – Essential Risk Assessment
- ENG 118 – Essential Incident Response
- ENG 120 – Essential Security Assessment & Authorization
- ENG 123 – Essential Security Engineering Principles
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design

- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Mitigating Insecure Deserialization
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- TST 101 – Fundamentals of Security Testing
- LAB 113 – Identifying Cryptographic Failures
- LAB 115 – Identifying Reflective XSS
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 121 – Identifying Vulnerable and Outdated Components
- LAB 127 – Identifying Security Logging and Monitoring Failures
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 137 – Identifying Improper Authorization

Advanced

- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- DES 203 – Cryptographic Components: Randomness, Algorithms, and Key Management
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 210 – Hardening Linux/Unix Systems
- DES 212 – Architecture Risk Analysis & Remediation
- DES 214 – Securing Infrastructure Architecture
- DES 215 – Defending Infrastructure
- DES 216 – Protecting Cloud Infrastructure
- DES 217 – Securing Terraform Infrastructure Resources
- DES 218 – Protecting Microservices, Containers, and Orchestration
- DES 272 – OWASP M2: Mitigating Insecure Data Storage
- DES 282 – OWASP IoT2: Mitigating Insecure Network Services
- DES 288 – OWASP IoT8: Mitigating Lack of Device Management
- DES 289 – OWASP IoT9: Mitigating Insecure Default Settings
- DSO 205 – Securing the COTS Supply Chain
- DSO 206 – Securing the Open-Source Supply Chain
- DSO 211 – Identifying Threats to Containers in a DevSecOps Framework
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- ENG 251 – Risk Management Foundations
- TST 202 – Penetration Testing Fundamentals
- TST 205 – Performing Vulnerability Scans
- LAB 111 – Identifying Server-Side Request Forgery
- LAB 114 – Identifying Cookie Tampering
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 118 – Identifying Weak File Upload Validation
- LAB 122 – Identifying Insecure APIs

- LAB 123 – Identifying Vertical Privilege Escalation
- LAB 124 – Identifying Horizontal Privilege Escalation
- LAB 125 – Identifying Buffer Overflow
- LAB 126 – Identifying Information Leakage
- LAB 128 – Identifying Unverified Password Change
- LAB 130 – Identifying Generation of Predictable Numbers or Identifiers
- LAB 131 – Identifying Improper Restriction of XML External Entity Reference
- LAB 132 – Identifying Exposed Services
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 135 – Identifying URL Redirection to Untrusted Site
- LAB 136 – Identifying Improper Neutralization of Script in Attributes in a Web Page
- LAB 138 – Authorization Bypass Through User-Controlled Key
- LAB 139 – Use of a Key Past its Expiration Date
- LAB 358 – Defending Python APIs from Broken Object Property Level Authorization
- LAB 640 – ATT&CK: Search Victim-Owned Websites
- LAB 641 – ATT&CK: Password Policy Discovery
- LAB 642 – ATT&CK: Permission Groups Discover

Elite

- CYB 301 – Fundamentals of Ethical Hacking
- ENG 311 – Attack Surface Analysis & Reduction
- LAB 321 – ATT&CK: Password Cracking
- LAB 322 – ATT&CK Exploiting Windows File Sharing Server with Eternal Romance Remote Services
- LAB 323 – ATT&CK: Exploiting Vulnerable Java Web Application Server Software
- LAB 324 – ATT&CK: Exploiting Java Web Application Server Misconfiguration
- LAB 612 – ATT&CK: Network Service Identification
- LAB 613 – ATT&CK: Vulnerability Identification Using Vulnerability Databases
- LAB 615 – ATT&CK: Updating Vulnerable Java Web Application Server Software
- LAB 616 – ATT&CK: Host Vulnerability Scanning
- LAB 617 – ATT&CK: Testing for Plaintext Secrets in Files
- LAB 618 – ATT&CK: Log Analysis
- LAB 619 – ATT&CK: Exfiltration Over C2 Channel
- LAB 620 – ATT&CK: Exploitation of Remote Services (Advanced)
- LAB 625 – ATT&CK: Exploit Public-Facing Application (Advanced)
- LAB 626 – Using and Exploit Framework for SQL Injection
- LAB 627 – Using and Exploit Framework for Port Scanning
- LAB 628 – Using and Exploit Framework for SMB version Scanning
- LAB 629 – Using and Exploit Framework for SNMP Scanning
- LAB 630 – ATT&CK: Exploiting Java SQL Injection to Extract Password Hashes
- LAB 631 – ATT&CK: Network Service Discovery
- LAB 632 – ATT&CK: Network Share Discovery
- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 634 – ATT&CK: Create Account
- LAB 635 – ATT&CK: Unsecured Credentials
- LAB 636 – ATT&CK: Data from Local System
- LAB 637 – ATT&CK: Valid Accounts
- LAB 638 – Using Mimikatz
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 325 – Secure Network Engineer

The Secure Network Engineer Learning Path credential is designed to provide the security skills and knowledge required to protect infrastructure and the sensitive data it handles. The curriculum provides participants with the ability to demonstrate the competency to apply best practices for managing systems and services across all environments.

After completing this learning path learners will have the knowledge and skill to:

- Apply best practices for managing systems and services across all environments
- Improve the stability, security, efficiency, and scalability of environments
- Understand how to create and modify scripts to perform tasks

NOTE: Learn and Skill labs are elective training modules that help transform concepts into tangible skills through hands-on, realistic examples of real-world threat scenarios.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

13
Courses

17
Labs

10
Hours

11
CPE Credits

Core

- DES 101 – Fundamentals of Secure Architecture
- ENG 110 – Essential Account Management Security
- ENG 114 – Essential Risk Assessment
- ENG 115 – Essential System and Information Integrity
- ENG 119 – Essential Security Audit and Accountability
- ENG 121 – Essential Identification and Authentication

Advanced

- API 251 – Implementing Web Application and API Protection (WAAP)
- COD 261 – Threats to Scripts
- COD 262 – Fundamentals of Shell and Interpreted Language Security
- DES 210 – Hardening Linux
- DES 214 – Securing Infrastructure Architecture

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management (C-SCRM) to Mitigate Threats to IT/OT
- CYB 311 – Threat Analysis with AI
- LAB 610 – ATT&CK: File and Directory Permissions Modification
- LAB 611 – ATT&CK: File and Directory Discovery
- LAB 615 – ATT&CK: Updating Vulnerable Java Web Application Server Software
- LAB 621 – ATT&CK: Password Cracking
- LAB 622 – ATT&CK: Exploiting Windows File Sharing Server with External Remote Services

- LAB 623 – ATT&CK: Exploiting Vulnerable Java Web Application Server Software
- LAB 624 – ATT&CK: Exploiting Java Web Application Server Misconfiguration
- LAB 630 – ATT&CK: Exploiting Java SQL Injection to Extract Password Hashes
- LAB 631 – ATT&CK: Network Service Discovery
- LAB 632 – ATT&CK: Network Share Discovery
- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 634 – ATT&CK: Create Account
- LAB 635 – ATT&CK: Unsecured Credentials
- LAB 636 – ATT&CK Data from Local System
- LAB 637 – ATT&CK Valid Accounts
- LAB 638 – Using Mimikatz
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 326 – Secure Automation Engineer

The Secure Automation Engineer Learning Path includes a variety of security courses designed for those who design, program, simulate and test automated machinery and processes to complete exact tasks. The curriculum covers essential goals and controls needed to create secure software, managing risk in the software development lifecycle, cryptography, handling input and output, and the OWASP Top Ten.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

32 Courses

0 Labs

8 Hours

10 CPE Credits

Core

- ENG 110 – Essential Account Management Security
- ENG 113 – Essential Secure Configuration Management
- ENG 114 – Essential Risk Assessment
- ENG 119 – Essential Security Audit & Accountability
- ENG 120 – Essential Security Assessment & Authorization
- ENG 123 – Essential Security Engineering Principles
- ENG 124 – Essential Application Protection
- ENG 125 – Essential Data Protection

Advanced

- DES 209 – Authentication and Lifecycle Management
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DSO 211 – Identifying Threats to Containers and Data in a DevSecOps Framework
- ENG 251 – Risk Management Foundations
- ICS 210 – ICS/SCADA Security Essentials

Elite

- DSO 302 – Automated Security Testing
- DSO 303 – Automating Security Updates
- DSO 306 – Implementing Infrastructure as Code
- ENG 351 – Preparing the Risk Management Framework

- ICS 310 – Protecting Information and System Integrity in Industrial Control System Environments
- SDT 301 – Testing for Injection
- SDT 302 – Testing for Identification Failures
- SDT 314 – Testing for Unrestricted Upload of File with Dangerous Type
- SDT 315 – Testing for Incorrect Permission Assignment for Critical Resource
- SDT 316 – Testing for Use of Hard-Coded Credentials

CSC 327 – Embedded Test Engineer

The Embedded Test Engineer Learning Path includes a variety of security courses designed for those responsible for verifying and assuring the application security of embedded systems. The curriculum provides learners with a solid understanding of applied testing techniques and a well-rounded base of knowledge to perform their tasks. Learners will explore security best practices for conducting penetration tests and vulnerability assessment activities on embedded systems.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

57 Courses

0 Labs

16 Hours

19 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 101 – Fundamentals of Secure Architecture
- ENG 114 – Essential Risk Assessment
- ENG 123 – Essential Security Engineering Principles
- TST 101 – Fundamentals of Security Testing

Advanced

- ATK 201 – Using the MITRE ATT&CK Framework
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- DES 212 – Architecture Risk Analysis and Remediation
- DES 255 – Securing the IoT Update Process
- DES 260 – Fundamentals of IoT Architecture and Design
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- ICS 210 – ICS/SCADA Security Essentials
- TST 202 – Penetration Testing Fundamentals

Elite

- CYB 301 – Fundamentals of Ethical Hacking
- DSO 302 – Automated Security Testing
- ENG 312 – How to Perform a Security Code Review
- ICS 310 – Protecting Information and System Integrity in Industrial Control System Environments
- SDT 301 – Testing for Injection
- SDT 302 – Testing for Identification and Authentication Failures
- SDT 304 – Testing for Insecure Design
- SDT 303 – Testing for Cryptographic Failures
- SDT 305 – Testing for Broken Access Control
- SDT 306 – Testing for Security Misconfiguration
- SDT 307 – Testing for Server-Side Request Forgery (SSRF)

- SDT 308 – Testing for Software and Data Integrity Failures
- SDT 309 – Testing for Vulnerable and Outdated Components
- SDT 310 – Testing for Security Logging and Monitoring Failures
- SDT 311 – Testing for Integer Overflow or Wraparound
- SDT 312 – Testing for (Path Traversal) Improper Limitation of a Pathname to a Restricted Directory
- SDT 313 – Testing for (CSRF) Cross Site Request Forgery
- SDT 314 – Testing for Unrestricted Upload of File with Dangerous Type
- SDT 315 – Testing for Incorrect Permission Assignment for Critical Resource
- SDT 316 – Testing for Use of Hard-Coded Credentials
- SDT 317 – Testing for Improper Control of Generation of Code
- SDT 318 – Testing for Insufficiently Protected Credentials
- SDT 319 – Testing for Out-of-bounds Read
- SDT 320 – Testing for Out-of-bounds Write
- SDT 321 – Testing for Uncontrolled Resource Consumption
- SDT 322 – Testing for Improper Privilege Management
- SDT 323 – Testing for Improper Input Validation
- SDT 324 – Testing for Improper Restriction of Operations within the Bounds of a Memory Buffer
- SDT 325 – Testing for NULL Pointer Dereference
- SDT 326 – Testing for Use After Free
- TST 301 – Infrastructure Penetration Testing
- TST 302 – Application Penetration Testing
- TST 351 – Penetration Testing for TLS Vulnerabilities
- TST 352 – Penetration Testing for Injection Vulnerabilities
- TST 353 – Penetration Testing for SQL Injection
- TST 354 – Penetration Testing for Memory Corruption Vulnerabilities
- TST 355 – Penetration Testing for Authorization Vulnerabilities
- TST 356 – Penetration Testing for Cross-Site Scripting (XSS)
- TST 357 – Penetration Testing for Hardcoded Secrets
- TST 358 – Penetration Testing Wireless Networks
- TST 359 – Penetration Testing Network Infrastructure
- TST 360 – Penetration Testing for Authentication Vulnerabilities

CSC 328 – QA Test Engineer

The Secure Learning Path includes a variety of security courses designed for those responsible for assessing and testing the quality of specifications and technical design. The curriculum provides learners with understanding of how to perform hands-on testing for the most common software vulnerabilities. Learners will gain the knowledge and skill to review and understand systems requirements and design, review test strategy and design, and identify mitigations for defects identified during testing.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

81 Courses

51 Labs

34 Hours

41 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 101 – Fundamentals of Secure Architecture
- ENG 114 – Essential Risk Assessment
- ENG 123 – Essential Security Engineering Principles
- TST 101 – Fundamentals of Security Testing

Advanced

- API 210 – Mitigating APIs Lack of Resources & Rate Limiting
- API 211 – Mitigating APIs Broken Object Level Authorization
- API 213 – Mitigating APIs Mass Assignment
- API 214 – Mitigating APIs Improper Asset Management
- ATK 201 – Using the MITRE ATT&CK Framework
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- DES 202 – Cryptographic Suite Services: Encoding, Encrypting & Hashing
- DES 203 – Cryptographic Components: Randomness, Algorithms, and Key Management
- DES 204 – Role of Cryptography in Application Development
- DES 205 – Message Integrity Cryptographic Functions
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 209 – Authentication and Lifecycle Management
- DES 212 – Architecture Risk Analysis and Remediation
- DES 214 – Securing Infrastructure Architecture
- DES 215 – Defending Infrastructure
- DES 216 – Protecting Cloud Infrastructure
- DES 218 – Protecting Microservices, Containers, and Orchestration
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures

- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DSO 212 – Fundamentals of Zero Trust Security
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- TST 202 – Penetration Testing Fundamentals
- TST 205 – Performing Vulnerability Scans
- LAB 111 – Identifying Server-Side Request Forgery
- LAB 113 – Identifying Cryptographic Failures
- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 118 – Identifying Weak File Upload Validation
- LAB 119 – Identifying Persistent Cross-Site Scripting (XSS)
- LAB 120 – Identifying XML Injection
- LAB 121 – Identifying Vulnerable and Outdated Components
- LAB 122 – Identifying Insecure APIs
- LAB 123 – Identifying Vertical Privilege Escalation
- LAB 124 – Identifying Horizontal Privilege Escalation
- LAB 125 – Identifying Buffer Overflow
- LAB 126 – Identifying Information Leakage
- LAB 127 – Identifying Security Logging and Monitoring Failures
- LAB 128 – Identifying Unverified Password Change
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 130 – Identifying Generation of Predictable Numbers or Identifiers
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization

Elite

- CYB 301 – Fundamentals of Ethical Hacking
- DES 311 – Creating Secure Application Architecture
- DSO 302 – Automated Security Testing
- ENG 312 – How to Perform a Security Code Review
- SDT 301 – Testing for Injection
- SDT 302 – Testing for Identification and Authentication Failures
- SDT 303 – Testing for Cryptographic Failures
- SDT 304 – Testing for Insecure Design
- SDT 305 – Testing for Broken Access Control
- SDT 306 – Testing for Security Misconfiguration
- SDT 307 – Testing for Server-Side Request Forgery (SSRF)

- SDT 308 – Testing for Software and Data Integrity Failures
- SDT 309 – Testing for Vulnerable and Outdated Components
- SDT 310 – Testing for Security Logging and Monitoring Failures
- SDT 311 – Testing for Integer Overflow or Wraparound
- SDT 312 – Testing for (Path Traversal) Improper Limitation of a Pathname to a Restricted Directory
- SDT 313 – Testing for (CSRF) Cross Site Request Forgery
- SDT 314 – Testing for Unrestricted Upload of File with Dangerous Type
- SDT 315 – Testing for Incorrect Permission Assignment for Critical Resource
- SDT 316 – Testing for Use of Hard-Coded Credentials
- SDT 317 – Testing for Improper Control of Generation of Code
- SDT 318 – Testing for Insufficiently Protected Credentials
- SDT 319 – Testing for Out-of-bounds Read
- SDT 320 – Testing for Out-of-bounds Write
- SDT 321 – Testing for Uncontrolled Resource Consumption
- SDT 322 – Testing for Improper Privilege Management
- SDT 323 – Testing for Improper Input Validation
- SDT 324 – Testing for Improper Restriction of Operations within the Bounds of a Memory Buffer
- SDT 325 – Testing for NULL Pointer Dereference
- SDT 326 – Testing for Use After Free
- TST 351 – Penetration Testing for TLS Vulnerabilities
- TST 352 – Penetration Testing for Injection Vulnerabilities
- TST 353 – Penetration Testing for SQL Injection
- TST 354 – Penetration Testing for Memory Corruption Vulnerabilities
- TST 355 – Penetration Testing for Authorization Vulnerabilities
- TST 356 – Penetration Testing for Cross-Site Scripting (XSS)
- TST 357 – Penetration Testing for Hardcoded Secrets
- TST 358 – Penetration Testing Wireless Networks
- TST 359 – Penetration Testing Network Infrastructure
- TST 360 – Penetration Testing for Authentication Vulnerabilities
- LAB 615 – ATT&CK: Updating Vulnerable Java Web Application Server Software
- LAB 616 – ATT&CK: Host Vulnerability Scanning
- LAB 617 – ATT&CK: Testing for Plaintext Secrets in Files
- LAB 618 – ATT&CK: Log Analysis
- LAB 619 – ATT&CK: Exfiltration Over C2 Channel
- LAB 620 – ATT&CK: Exploitation of Remote Services (Advanced)
- LAB 621 – ATT&CK: Password Cracking
- LAB 622 – ATT&CK: Exploiting Windows File Sharing Server with Eternal Romance Remote Services
- LAB 623 – ATT&CK: Exploiting Vulnerable Java Web Application Server Software
- LAB 624 – ATT&CK: Exploiting Java Web Application Server Misconfiguration
- LAB 625 – ATT&CK: Exploit Public-Facing Application (Advanced)
- LAB 626 – Using and Exploit Framework for SQL Injection
- LAB 627 – Using and Exploit Framework for Port Scanning
- LAB 628 – Using and Exploit Framework for SMB version Scanning
- LAB 629 – Using and Exploit Framework for SNMP Scanning
- LAB 630 – ATT&CK: Exploiting Java SQL Injection to Extract Password Hashes
- LAB 631 – ATT&CK: Network Service Discovery
- LAB 632 – ATT&CK: Network Share Discovery

- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 634 – ATT&CK: Create Account
- LAB 635 – ATT&CK: Unsecured Credentials
- LAB 636 – ATT&CK: Data from Local System
- LAB 637 – ATT&CK: Valid Accounts
- LAB 638 – Using Mimikatz
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 329 – Secure IT Architect

The Secure IT Architect Learning Path includes a variety of security courses designed for those responsible for designing and maintaining computer networks. The curriculum covers best practices for secure software design, creating integrated architecture across business and technology, and protecting data and resources from disclosure, modification, and deletion.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

46 Courses

19 Labs

24 Hours

28 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 101 – Fundamentals of Secure Architecture

Advanced

- API 250 – Controlling Access to the Kubernetes API
- API 251 – Implementing Web Application and API Protection (WAAP)
- API 351 – Securing Kubernetes in the Build and Release Stages
- COD 252 – Securing Google Platform Applications & Data
- DES 202 – Cryptographic Suite Services: Encoding, Encrypting & Hashing
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 207 – Mitigating OWASP API Security Top 10
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 209 – Authentication and Lifecycle Management
- DES 210 – Hardening Linux/Unix Systems
- DES 212 – Architecture Risk Analysis and Remediation
- DES 214 – Securing Infrastructure Architecture
- DES 215 – Defending Infrastructure
- DES 216 – Protecting Cloud Infrastructure
- DES 217 – Securing Terraform Infrastructure and Resources
- DES 218 – Protecting Microservices, Containers, and Orchestration
- DES 255 – Securing the IoT Update Process
- DES 260 – Fundamentals of IoT Architecture and Design
- DES 261 – Securing Serverless Environments
- DSO 211 – Identifying Threats to Containers and Data in a DevSecOps Framework
- DSO 212 – Fundamentals of Zero Trust Security
- DSO 256 – DevSecOps in the Google Cloud Platform
- ENG 211 – How to Create Application Security Design Requirements
- ENG 251 – Risk Management Foundations

- LAB 363 – Defending Python APIs from Security Misconfiguration
- LAB 642 – ATT&CK: Permission Groups Discovery

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- CYB 311 – Threat Analysis with AI
- DES 313 – Hardening a Kubernetes Cluster
- DES 314 – Hardening the Docker Engine
- DES 361 – Mitigating LCNC (Low-Code/No-Code) Account Impersonation
- DES 362 – Mitigating LCNC (Low-Code/No-Code)) Authorization Misuse
- DES 364 – Mitigating LCNC (Low-Code/No-Code) Authentication and Secure Communication Failures
- DSO 301 – Orchestrating Secure System and Service Configuration
- DSO 304 – Securing API Gateways in a DevSecOps Framework
- DSO 305 – Automating CI/CD Pipeline Compliance
- DSO 306 – Implementing Infrastructure as Code
- ENG 311 – Attack Surface Analysis and Reduction
- ENG 351 – Preparing the Risk Management Framework
- ENG 352 – Categorizing Systems and Information within the RMF
- ENG 353 – Selecting, Implementing and Assessing Controls within the RMF
- ENG 354 – Authorizing and Monitoring System Controls within the RMF
- TST 303 – Penetration Testing for Google Cloud Platform
- TST 304 – Penetration Testing for AWS Cloud
- TST 305 – Penetration Testing for Azure Cloud
- LAB 612 – ATT&CK: Network Service Identification
- LAB 613 – ATT&CK: Vulnerability Identification Using Vulnerability Databases
- LAB 615 – ATT&CK: Updating Vulnerable Java Web Application Server Software
- LAB 616 – ATT&CK: Host Vulnerability Scanning
- LAB 617 – ATT&CK: Testing for Plaintext Secrets in Files
- LAB 618 – ATT&CK: Log Analysis
- LAB 619 – ATT&CK: Exfiltration Over C2 Channel
- LAB 620 – ATT&CK: Exploitation of Remote Services (Advanced)
- LAB 622 – ATT&CK: Exploiting Windows File Sharing Server with Eternal Romance Remote Services
- LAB 623 – ATT&CK: Exploiting Vulnerable Java Web Application Server Software
- LAB 624 – ATT&CK: Exploiting Java Web Application Server Misconfiguration
- LAB 625 – ATT&CK: Exploit Public-Facing Application (Advanced)
- LAB 626 – Using an Exploit Framework for SQL Injection
- LAB 627 – Using an Exploit Framework for Port Scanning
- LAB 628 – Using an Exploit Framework for SMB Version Scanning
- LAB 629 – Using an Exploit Framework for SNMP Scanning
- LAB 638 – Using Mimikatz

CSC 330 – Secure Embedded Architect

The Secure Embedded Architect Learning Path includes a variety of security courses designed for those responsible for designing and implementing software of embedded devices and systems and provides insight into the unique resource requirements of embedded environments and best practices for designing secure software for them.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

11 Courses

0 Labs

5 Hours

6 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- DES 101 – Fundamentals of Secure Architecture

Advanced

- DES 202 – Cryptographic Suite Services: Encoding, Encrypting & Hashing
- DES 212 – Architecture Risk Analysis and Remediation
- DES 255 – Securing the IoT Update Process
- DES 260 – Fundamentals of IoT Architecture and Design
- ICS 210 – ICS/SCADA Security Essentials

Elite

- DES 311 – Creating Secure Application Architecture
- ENG 311 – Attack Surface Analysis & Reduction
- ENG 312 – How to Perform a Security Code Review
- ICS 310 – Protecting Information and System Integrity in Industrial Control System Environments

CSC 331 – Secure Software Architect

The Secure Software Architect Learning Path includes a variety of security courses designed for those making design choices, coordinating, and overseeing technical standards and includes software coding standards, tools, and platforms. The curriculum provides learners with the ability to apply secure software architecture best practices to early phase SDLC activities. Learners will gain the knowledge and skill to implement defensive coding techniques and avoid systemic issues found in insecure software.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

73 Courses

23 Labs

32 Hours

38 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- AWA 106 – Building Secure Software: Challenges in Application Security
- AWA 107 – Building Secure Software: Foundations & Best Practices
- AWA 108 – Building Secure Software: A Guide to Software Integration, Testing, and Deployment
- COD 141 – Fundamentals of Database Security
- DES 101 – Fundamentals of Secure Architecture
- DES 151 – Fundamentals of the PCI Secure SLC Standard

Advanced

- API 210 – Mitigating APIs Lack of Resources & Rate Limiting
- API 211 – Mitigating APIs Broken Object Level Authorization
- API 213 – Mitigating APIs Mass Assignment
- API 214 – Mitigating APIs Improper Asset Management
- API 251 – Implementing Web Application and API Protection (WAAP)
- COD 252 – Securing Google Platform Applications & Data
- COD 261 – Threats to Scripts
- COD 267 – Securing Python Microservices
- COD 289 – Securing Java Applications with Spring Security
- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- DES 202 – Cryptographic Suite Services: Encoding, Encrypting & Hashing
- DES 203 – Cryptographic Components: Randomness, Algorithms, and Key Management
- DES 204 – Role of Cryptography in Application Development
- DES 205 – Message Integrity Cryptographic Functions
- DES 207 – Mitigating OWASP API Security Top 10
- DES 209 – Authentication and Lifecycle Management
- DES 212 – Architecture Risk Analysis and Remediation
- DES 214 – Securing Infrastructure Architecture

- DES 215 – Defending Infrastructure
- DES 216 – Protecting Cloud Infrastructure
- DES 217 – Securing Terraform Infrastructure and Resources
- DES 218 – Protecting Microservices, Containers, and Orchestration
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DES 250 – Secure Software Acceptance and Deployment
- DES 255 – Securing the IoT Update Process
- DES 260 – Fundamentals of IoT Architecture and Design
- DSO 201 – Fundamentals of Secure DevOps
- DSO 211 – Identifying Threats to Containers and Data in a DevSecOps Framework
- DSO 212 – Fundamentals of Zero Trust Security
- DSO 256 – DevSecOps in the Google Cloud Platform
- ENG 211 – How to Create Application Security Design Requirements
- ENG 251 – Risk Management Foundations
- TST 206 – ASVS Requirements for Developers
- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization
- LAB 220 – Defending Against Hard-Coded Secrets
- LAB 357 – Defending Python APIs from Broken Authentication

Elite

- COD 287 – Java Application Server Hardening
- COD 383 – Protecting Java Backend Services
- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- DES 311 – Creating Secure Application Architecture
- DSO 301 – Orchestrating Secure System and Service Configuration
- DSO 302 – Automated Security Testing
- DSO 304 – Securing API Gateways in a DevSecOps Framework
- DSO 305 – Automating CI/CD Pipeline Compliance
- ENG 311 – Attack Surface Analysis & Reduction

- ENG 312 – How to Perform a Security Code Review
- ENG 320 – Using Software Composition Analysis (SCA) to Secure Open-Source Components
- ENG 351 – Preparing the Risk Management Framework
- ENG 352 – Categorizing Systems and Information within the RMF
- ENG 353 – Selecting, Implementing and Assessing Controls within the RMF
- ENG 354 – Authorizing and Monitoring System Controls within the RMF
- SDT 302 – Testing for Identification and Authentication Failures
- SDT 314 – Testing for Unrestricted Upload of File with Dangerous Type
- SDT 315 – Testing for Incorrect Permission Assignment for Critical Resource
- SDT 316 – Testing for Use of Hard-Coded Credentials
- TST 303 – Penetration Testing for Google Cloud Platform
- TST 304 – Penetration Testing for AWS Cloud
- TST 305 – Penetration Testing for Azure Cloud
- LAB 615 – ATT&CK: Updating Vulnerable Java Web Application Server Software
- LAB 621 – ATT&CK: Password Cracking
- LAB 623 – ATT&CK: Exploiting Vulnerable Java Web Application Server Software
- LAB 624 – ATT&CK: Exploiting Java Web Application Server Misconfiguration
- LAB 630 – ATT&CK: Exploiting Java SQL Injection to Extract Password Hashes
- LAB 631 – ATT&CK: Network Service Discovery
- LAB 632 – ATT&CK: Network Share Discovery
- LAB 634 – ATT&CK: Create Account
- LAB 635 – ATT&CK: Unsecured Credentials
- LAB 636 – ATT&CK: Data from Local System
- LAB 637 – ATT&CK: Valid Accounts

CSC 332 – Secure Business Analyst

The Secure Business Analyst Learning Path includes a variety of security courses designed for those responsible for defining, analyzing, and documenting requirements in the software development lifecycle.

After completing this learning path learners will have the knowledge and skill to:

- Adhere to system and information security policies
- Meet compliance mandates for relevant government and industry standards
- Understand access control, configuration management, risk assessment, auditing, and authentication

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

17 Courses

1 Labs

6 Hours

7 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 101 – Fundamentals of Secure Architecture
- DES 151 – Fundamentals of the PCI Secure SLC Standard
- ENG 114 – Essential Risk Assessment
- ENG 116 – Essential Security Planning Policy & Procedures
- ENG 117 – Essential Information Security Program Planning

Advanced

- DSO 201 – Fundamentals of Secure DevOps
- ENG 211 – How to Create Application Security Design Requirements
- ENG 251 – Risk Management Foundations
- TST 202 – Penetration Testing Fundamentals
- TST 206 – ASVS Requirements for Developers
- LAB 361 – Defending Python APIs from Unrestricted Access to Sensitive Business Flows

Elite

- DSO 302 – Automated Security Testing
- ENG 351 – Preparing the Risk Management Framework
- ENG 352 – Categorizing Systems and Information within the RMF
- ENG 353 – Selecting, Implementing and Assessing Controls within the RMF
- ENG 354 – Authorizing and Monitoring System Controls within the RMF

CSC 333 – Secure Systems Analyst

The Secure Systems Analyst Learning Path includes a variety of security courses designed for those who specialize in the implementation of computer system requirements. The curriculum provides the fundamental knowledge required to secure networks and systems including:

- Taking a holistic approach to network and system security
- Defining and analyzing system problems
- Designing and testing standards and solutions
- Controls, monitoring access, operational procedures, auditing, and logging

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

62 Courses

38 Labs

27 Hours

33 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- ENG 110 – Essential Account Management Security
- ENG 111 – Essential Session Management Security
- ENG 112 – Essential Access Control for Mobile Devices
- ENG 113 – Essential Secure Configuration Management
- ENG 114 – Essential Risk Assessment
- ENG 115 – Essential System & Information Integrity
- ENG 116 – Essential Security Planning Policy & Procedures
- ENG 117 – Essential Information Security Program Planning
- ENG 118 – Essential Incident Response
- ENG 119 – Essential Security Audit & Accountability
- ENG 120 – Essential Security Assessment & Authorization
- ENG 121 – Essential Identification & Authentication
- ENG 122 – Essential Physical & Environmental Protection
- ENG 123 – Essential Security Engineering Principles
- ENG 124 – Essential Application Protection
- ENG 125 – Essential Data Protection
- ENG 126 – Essential Security Maintenance Policies
- ENG 127 – Essential Media Protection

Advanced

- API 210 – Mitigating APIs Lack of Resources & Rate Limiting
- API 211 – Mitigating APIs Broken Object Level Authorization
- API 213 – Mitigating APIs Mass Assignment

- API 214 – Mitigating APIs Improper Asset Management
- COD 268 – Mitigating TypeScript Application Vulnerabilities
- CYB 210 – Cybersecurity Incident Response
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- DES 210 – Hardening Linux/Unix Systems
- DES 217 – Securing Terraform Infrastructure and Resources
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DSO 212 – Fundamentals of Zero Trust Security
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- ENG 212 – Implementing Secure Software Operations
- ENG 251 – Risk Management Foundations
- TST 206 – ASVS Requirements for Developers
- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 118 – Identifying Weak File Upload Validation
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 123 – Identifying Vertical Privilege Escalation
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization
- LAB 220 – Defending Against Hard-Coded Secrets
- LAB 357 - Defending Python APIs from Broken Authentication
- LAB 362 – Defending Python APIs from Server Side Request Forgery
- LAB 364 – Defending Python APIs from Improper Inventory Management

Elite

- COD 310 – Securing ASP.NET Core Applications
- COD 325 – Protecting Data in C# for .NET Core
- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- CYB 311 – Threat Analysis with AI
- DSO 301 – Orchestrating Secure System and Service Configuration
- DSO 302 – Automated Security Testing
- DSO 304 – Securing API Gateways in a DevSecOps Framework

- DSO 305 – Automating CI/CD Pipeline Compliance
- ENG 320 – Using Software Composition Analysis (SCA) to Secure Open-Source Components
- ENG 351 – Preparing the Risk Management Framework
- ENG 352 – Categorizing Systems and Information within the RMF
- ENG 353 – Selecting, Implementing and Assessing Controls within the RMF
- ENG 354 – Authorizing and Monitoring System Controls within the RMF
- ICS 310 – Protecting Information and System Integrity in Industrial Control System Environments
- TST 303 – Penetration Testing for Google Cloud Platform
- TST 304 – Penetration Testing for AWS Cloud
- TST 305 – Penetration Testing for Azure Cloud
- LAB 309 – Defending TypeScript Applications Against Unrestricted Upload of File
- LAB 314 – Defending TypeScript Applications Against SSRF
- LAB 316 – Defending TypeScript Applications Against Hard-coded Credentials
- LAB 320 – Defending TypeScript Applications Against Code Injection
- LAB 325 – Defending TypeScript Applications Against CSRF
- LAB 326 – Defending TypeScript Applications Against Path Traversal
- LAB 342 – Defending TypeScript Applications Against Command Injection
- LAB 344 – Defending TypeScript Applications Against Incorrect Authorization
- LAB 346 – Defending TypeScript Applications Against Deserialization of Untrusted Data
- LAB 349 – Defending TypeScript Applications Against SQL Injection
- LAB 351 – Defending TypeScript Applications Against Cross-Site Scripting
- LAB 353 – Defending TypeScript Applications Against Incorrect Authorization
- LAB 616 – ATT&CK: Host Vulnerability Scanning
- LAB 617 – ATT&CK: Testing for Plaintext Secrets in Files
- LAB 618 – ATT&CK: Log Analysis
- LAB 619 – ATT&CK: Exfiltration Over C2 Channel
- LAB 620 – ATT&CK: Exploitation of Remote Services (Advanced)
- LAB 625 – ATT&CK: Exploit Public-Facing Application (Advanced)
- LAB 626 – Using an Exploit Framework for SQL Injection
- LAB 627 – Using an Exploit Framework for Port Scanning
- LAB 628 – Using an Exploit Framework for SMB Version Scanning
- LAB 629 – Using an Exploit Framework for SNMP Scanning

CSC 334 – Secure Systems Administrator

The Secure Systems Administrator Learning Path includes a variety of security courses designed for those responsible for preventing and mitigating security breaches that may arise within computer systems. The curriculum provides a holistic approach to network and system security with an exploration of controls, monitoring access, operational procedure, and formal auditing and logging.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

69 Courses

4 Labs

26 Hours

31 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- COD 141 – Fundamentals of Database Security
- DES 151 – Fundamentals of the PCI Secure SLC Standard
- ENG 110 – Essential Account Management Security
- ENG 111 – Essential Session Management Security
- ENG 113 – Essential Secure Configuration Management
- ENG 118 – Essential Incident Response
- ENG 119 – Essential Security Audit & Accountability
- ENG 121 – Essential Identification & Authentication
- ENG 122 – Essential Physical & Environmental Protection
- ENG 123 – Essential Security Engineering Principles
- ENG 125 – Essential Data Protection
- ENG 127 – Essential Media Protection
- ENG 150 – Meeting Confidentiality, Integrity, and Availability Requirements
- ENG 151 – Fundamentals of Privacy Protection

Advanced

- API 210 – Mitigating APIs Lack of Resources & Rate Limiting
- API 211 – Mitigating APIs Broken Object Level Authorization
- API 213 – Mitigating APIs Mass Assignment
- API 214 – Mitigating APIs Improper Asset Management
- COD 219 – Creating Secure Code SAP ABAP Foundations
- COD 252 – Securing Google Platform Applications & Data
- COD 261 – Threats to Scripts
- COD 262 – Fundamentals of Shell and Interpreted Language Security
- COD 263 – Secure Bash Scripting
- COD 264 – Secure Perl Scripting
- COD 265 – Secure Python Scripting
- COD 266 – Secure Ruby Scripting

- CYB 210 – Cybersecurity Incident Response
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 209 – Authentication and Lifecycle Management
- DES 210 – Hardening Linux/Unix Systems
- DES 214 – Securing Infrastructure Architecture
- DES 215 – Defending Infrastructure
- DES 216 – Protecting Cloud Infrastructure
- DES 217 – Securing Terraform Infrastructure and Resources
- DES 218 – Protecting Microservices, Containers, and Orchestration
- DES 219 – Securing Google's Firebase Platform
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DES 262 – Securing Enterprise Low-Code Application Platforms
- DSO 201 – Fundamentals of Secure DevOps
- DSO 211 – Identifying Threats to Containers and Data in a DevSecOps Framework
- DSO 212 – Fundamentals of Zero Trust Security
- DSO 256 – DevSecOps in the Google Cloud Platform
- ENG 205 – Fundamentals of Threat Modeling
- LAB 642 – ATT&CK: Permission Groups Discovery

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- DES 361 – Mitigating LCNC (Low-Code/No-Code) Account Impersonation
- DES 362 – Mitigating LCNC (Low-Code/No-Code)) Authorization Misuse
- DES 364 – Mitigating LCNC (Low-Code/No-Code) Authentication and Secure Communication Failures
- DSO 301 – Orchestrating Secure System and Service Configuration
- DSO 303 – Automating Security Updates
- DSO 304 – Securing API Gateways in a DevSecOps Framework
- DSO 305 – Automating CI/CD Pipeline Compliance
- ENG 320 - Using Software Composition Analysis (SCA) to Secure Open-Source Components
- TST 303 – Penetration Testing for Google Cloud Platform
- TST 304 – Penetration Testing for AWS Cloud
- TST 305 – Penetration Testing for Azure Cloud
- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 638 – Using Mimikatz
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 335 – Secure Database Administrator

The Secure Database Administrator Learning Path includes a variety of security courses designed for those responsible for capacity planning, installation, configuration, database design, migration, performance monitoring, security, troubleshooting, as well as back-end data recovery. The curriculum builds fundamental knowledge of secure database development including:

- Common database attacks
- Platform-specific threats
- Database secure coding best practices

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

37 Courses

0 Labs

14 Hours

17 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 101 – Fundamentals of Secure Architecture
- COD 141 – Fundamentals of Database Security

Advanced

- COD 241 – Creating Secure Code - Oracle Database Applications
- COD 242 – Creating Secure SQL Server and Azure SQL Database Applications
- COD 245 – Securing NoSQL Cloud Databases
- COD 261 – Threats to Scripts
- COD 262 – Fundamentals of Shell and Interpreted Language Security
- DES 202 – Cryptographic Suite Services: Encoding, Encrypting & Hashing
- DES 203 – Cryptographic Components: Randomness, Algorithms, and Key Management
- DES 204 – Role of Cryptography in Application Development
- DES 205 – Message Integrity Cryptographic Functions
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 212 – Architecture Risk Analysis and Remediation
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components

- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DSO 212 – Fundamentals of Zero Trust Security
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements

Elite

- COD 352 – Creating Secure jQuery Code
- COD 383 – Protecting Java Backend Services
- DES 311 – Creating Secure Application Architecture
- ENG 311 – Attack Surface Analysis and Reduction
- ENG 312 – How to Perform a Security Code Review
- SDT 302 – Testing for Identification and Authentication Failures
- SDT 314 – Testing for Unrestricted Upload of File with Dangerous Type
- SDT 315 – Testing for Incorrect Permission Assignment for Critical Resource
- SDT 316 – Testing for Use of Hard-Coded Credentials

CSC 336 – Secure Linux Administrator

The Secure Linux Administrator Learning Path dives into operating system configuration and administration of virtual servers. Learners will develop the working knowledge needed to support development, testing, and systems integration. Additionally, the curriculum will provide learners with a solid understanding of secure development best practices.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

16 Courses

0 Labs

6 Hours

7 CPE Credits

Core

- ENG 110 – Essential Account Management Security
- ENG 114 – Essential Risk Assessment
- ENG 115 – Essential System & Information Integrity
- ENG 119 – Essential Security Audit & Accountability
- ENG 121 – Essential Identification & Authentication
- ENG 150 – Meeting Confidentiality, Integrity, and Availability Requirements

Advanced

- COD 261 – Threats to Scripts
- COD 262 – Fundamentals of Shell and Interpreted Language Security
- COD 263 – Secure Bash Scripting
- COD 264 – Secure Perl Scripting
- COD 265 – Secure Python Scripting
- COD 266 – Secure Ruby Scripting
- DES 214 – Securing Infrastructure Architecture
- DES 215 – Defending Infrastructure
- DES 260 – Fundamentals of IoT Architecture and Design
- ENG 205 – Fundamentals of Threat Modeling

CSC 337 – Secure Product Owner

The Secure Product Owner Learning Path includes a variety of security courses designed for those responsible for setting, prioritizing, and evaluating the work generated by a software Scrum team to ensure impeccable features and functionality of the product. The curriculum introduces application security fundamentals including the essentials goals and controls needed to create secure software and manage risk in the software development lifecycle

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

36 Courses

1 Labs

11 Hours

14 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 151 – Fundamentals of the PCI Secure SLC Standard
- ENG 124 – Essential Application Protection
- ENG 125 – Essential Data Protection
- ENG 150 – Meeting Confidentiality, Integrity, and Availability Requirements
- ENG 151 – Fundamentals of Privacy Protection
- ENG 191 – Introduction to the Microsoft SDL
- ENG 192 – Implementing the Agile Microsoft SDL
- ENG 193 – Implementing the Microsoft SDL Optimization Model
- ENG 194 – Implementing Microsoft SDL Line of Business
- ENG 195 – Implementing the Microsoft SDL Threat Modeling Tool
- TST 101 – Fundamentals of Security Testing

Advanced

- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DES 212 – Architecture Risk Analysis and Remediation
- DES 260 – Fundamentals of IoT Architecture and Design

- DSO 201 – Fundamentals of Secure DevOps
- ENG 211 – How to Create Application Security Design Requirements
- ENG 251 – Risk Management Foundations
- TST 202 – Penetration Testing Fundamentals
- TST 206 – ASVS Requirements for Developers
- LAB 364 – Defending Python APIs from Improper Inventory Management

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management (C-SCRM) to Mitigate Threat to IT/OT
- DSO 302 – Automated Security Testing
- ENG 311 – Attack Surface Analysis and Reduction
- ENG 351 – Preparing the Risk Management Framework

CSC 338 – Secure Project Manager

The Secure Project Manager Learning Path includes a variety of security courses that introduces project managers to the essentials of access control, configuration management, risk assessment, auditing, and authentication. The curriculum provides the knowledge and skills necessary to ensure adherence to your organization's system and information security policies as well as relevant governmental and industry standards.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

38 Courses

0 Labs

14 Hours

17 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- AWA 106 – Building Secure Software: Challenges in Application Security
- AWA 107 – Building Secure Software: Foundations & Best Practices
- AWA 108 – Building Secure Software: A Guide to Software Integration, Testing, and Deployment
- COD 141 – Fundamentals of Database Security*
- COD 152 – Fundamentals of Secure Cloud Development*
- DES 101 – Fundamentals of Secure Architecture
- DES 151 – Fundamentals of the PCI Secure SLC Standard
- ENG 123 – Essential Security Engineering Principles
- ENG 124 – Essential Applications Protection
- ENG 125 – Essential Data Protection
- ENG 150 – Meeting Confidentiality, Integrity, and Availability Requirements
- ENG 151 – Fundamentals of Privacy Protection

Advanced

- COD 252 – Securing Google Platform Applications & Data
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- DES 204 – The Role of Cryptography in Application Development
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 212 – Architecture Risk Analysis and Remediation
- DES 214 – Securing Infrastructure Architecture
- DES 215 – Defending Infrastructure
- DES 216 – Protecting Cloud Infrastructure
- DES 218 – Protecting Microservices, Containers, and Orchestration
- DSO 201 – Fundamentals of Secure DevOps
- DSO 206 – Securing the Open-Source Software Supply Chain
- DSO 211 – Identifying Threats to Containers and Data in a DevSecOps Framework
- DSO 212 – Fundamentals of Zero Trust Security

- DSO 256 – DevSecOps in the Google Cloud Platform
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- ENG 251 – Risk Management Foundations
- TST 206 – ASVS Requirements for Developers

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management (C-SCRM) to Mitigate Threat to IT/OT
- DSO 301 – Orchestrating Secure System and Service Configuration
- DSO 302 – Automated Security Testing
- DSO 305 – Automating CI/CD Pipeline Compliance
- ENG 312 – How to Perform a Security Code Review
- ENG 351 – Preparing the Risk Management Framework

CSC 339 – Cyber Security Professional

The Elite Cyber Security Professional Learning Path includes a variety of security courses designed for those tasked with everything from the technical aspects of security, security policy, and everything in between.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

30 Courses

13 Labs

11 Hours

14 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- ENG 117 – Essential Information Security Program Planning
- ENG 118 – Essential Incident Response
- ENG 124 – Essential Application Protection
- ENG 151 – Fundamentals of Privacy Protection
- TST 101 – Fundamentals of Software Security Testing

Advanced

- API 250 – Controlling Access to the Kubernetes API
- CYB 210 – Cybersecurity Incident Response
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 212 – Fundamentals of Security Information & Event Management (SIEM)
- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 207 – Mitigating OWASP API Security Top 10
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 217 – Securing Terraform Infrastructure and Resources
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DSO 212 – Fundamentals of Zero Trust Security
- TST 202 – Penetration Testing Fundamentals
- TST 206 – ASVS Requirements for Developers
- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection

- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management (C-SCRM) to Mitigate Threat to IT/OT
- DES 361 – Mitigating LCNC (Low-Code/No-Code) Account Impersonation
- DES 362 – Mitigating LCNC (Low-Code/No-Code)) Authorization Misuse
- DES 364 – Mitigating LCNC (Low-Code/No-Code) Authentication and Secure Communication Failures
- ENG 320 – Using Software Composition Analysis (SCA) to Secure Open-Source Components
- TST 303 – Penetration Testing for Google Cloud Platform
- TST 304 – Penetration Testing for AWS Cloud
- TST 305 – Penetration Testing for Azure Cloud
- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 638 – Using Mimikatz
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 340 – Secure Operations/IT Manager

The Elite Operations/IT Manager Learning Path includes a variety of security courses designed for those responsible for managing operations and sharing responsibility for project success and managing day-to-day IT processes. The curriculum covers secure IT Operations concepts including:

- Essential goals and controls needed for secure software development
- Managing risk associated with the software development lifecycle
- Developing, implementing, and ensuring compliance with operational application security policies and procedures

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

30 Courses

27 Labs

15 Hours

18 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- ENG 117 – Essential Information Security Program Planning
- ENG 118 – Essential Incident Response
- ENG 124 – Essential Application Protection
- ENG 151 – Fundamentals of Privacy Protection
- TST 101 – Fundamentals of Software Security Testing

Advanced

- API 250 – Controlling Access to the Kubernetes API
- CYB 210 – Cybersecurity Incident Response
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 212 – Fundamentals of Security Information & Event Management (SIEM)
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 207 – Mitigating OWASP API Security Top 10
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 217 – Securing Terraform Infrastructure and Resources
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 250 – Secure Software Acceptance and Deployment
- DES 261 – Securing Serverless Environments
- DES 262 – Securing Enterprise Low-Code Application Platforms
- DSO 212 – Fundamentals of Zero Trust Security
- TST 202 – Penetration Testing Fundamentals
- TST 206 – ASVS Requirements for Developers

- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 118 – Identifying Weak File Upload Validation
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 123 – Identifying Vertical Privilege Escalation
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization

Elite

- DES 361 – Mitigating LCNC (Low-Code/No-Code) Account Impersonation
- DES 362 – Mitigating LCNC (Low-Code/No-Code)) Authorization Misuse
- DES 364 – Mitigating LCNC (Low-Code/No-Code) Authentication and Secure Communication Failures
- TST 303 – Penetration Testing for Google Cloud Platform
- TST 304 – Penetration Testing for AWS Cloud
- TST 305 – Penetration Testing for Azure Cloud
- LAB 612 – ATT&CK: Network Service Identification
- LAB 613 – ATT&CK: Vulnerability Identification Using Vulnerability Databases
- LAB 616 – ATT&CK: Host Vulnerability Scanning
- LAB 617 – ATT&CK: Testing for Plaintext Secrets in Files
- LAB 618 – ATT&CK: Log Analysis
- LAB 619 – ATT&CK: Exfiltration Over C2 Channel
- LAB 620 – ATT&CK: Exploitation of Remote Services (Advanced)
- LAB 625 – ATT&CK: Exploit Public-Facing Application (Advanced)
- LAB 626 – Using and Exploit Framework for SQL Injection
- LAB 627 – Using and Exploit Framework for Port Scanning
- LAB 628 – Using and Exploit Framework for SMB version Scanning
- LAB 629 – Using and Exploit Framework for SNMP Scanning
- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 638 – Using Mimikatz
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 341 – Application Security Champion

The Elite Application Security Champion Learning Path includes a variety of security courses designed for those chartered with driving a culture of "Security Built-in" to the software development lifecycle. The curriculum explains application security concepts such as privacy, secure development and architecture, security testing, threat modeling, cryptography, and cyber threat analysis and remediation.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

35 Courses

13 Labs

15 Hours

18 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- AWA 106 – Building Secure Software: Challenges in Application Security
- AWA 107 – Building Secure Software: Foundations & Best Practices
- AWA 108 – Building Secure Software: A Guide to Software Integration, Testing, and Deployment
- ENG 124 – Essential Application Protection
- ENG 125 – Essential Data Protection
- ENG 150 – Meeting Confidentiality, Integrity, and Availability Requirements
- ENG 151 – Fundamentals of Privacy Protection
- TST 101 – Fundamentals of Security Testing

Advanced

- API 251 – Implementing Web Application and API Protection (WAAP)
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- DES 204 – The Role of Cryptography in Application Development
- DES 207 – Mitigating OWASP API Security Top 10
- DES 212 – Architecture Risk Analysis and Remediation
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DSO 212 – Fundamentals of Zero Trust Security
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements

- TST 202 – Penetration Testing Fundamentals
- TST 205 – Performing Vulnerability Scans
- TST 206 – ASVS Requirements for Developers
- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- DSO 302 – Automated Security Testing
- ENG 311 – Attack Surface Analysis and Reduction
- ENG 312 – How to Perform a Security Code Review
- LAB 633 – Using an Exploit Framework for Web Application Scanning
- LAB 638 – Using Mimikatz
- LAB 639 – Using an Exploit Framework via Command Line Interface

CSC 342 – Information Security Specialist

The Elite Information Security Specialist Learning Path includes a variety of courses designed for those responsible for protecting systems, defining access privileges, control structures, and resources. The curriculum helps build the skills required to identify, protect, detect, and recover from risks, vulnerabilities, and threats to the security of information and/or data.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

82 Courses

36 Labs

35 Hours

43 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- COD 141 – Fundamentals of Database Security
- DES 151 – Fundamentals of the PCI Secure SLC Standard
- ENG 110 – Essential Account Management Security
- ENG 111 – Essential Session Management Security
- ENG 112 – Essential Access Control for Mobile Devices
- ENG 113 – Essential Secure Configuration Management
- ENG 114 – Essential Risk Assessment
- ENG 115 – Essential System & Information Integrity
- ENG 116 – Essential Security Planning Policy & Procedures
- ENG 117 – Essential Information Security Program Planning
- ENG 118 – Essential Incident Response
- ENG 119 – Essential Security Audit & Accountability
- ENG 120 – Essential Security Assessment & Authorization
- ENG 121 – Essential Identification & Authentication
- ENG 122 – Essential Physical & Environmental Protection
- ENG 123 – Essential Security Engineering Principles
- ENG 124 – Essential Application Protection
- ENG 125 – Essential Data Protection
- ENG 126 – Essential Security Maintenance Policies
- ENG 127 – Essential Media Protection
- ENG 151 – Fundamentals of Privacy Protection
- TST 101 – Fundamentals of Security Testing

Advanced

- API 210 – Mitigating APIs Lack of Resources & Rate Limiting
- API 211 – Mitigating APIs Broken Object Level Authorization
- API 213 – Mitigating APIs Mass Assignment
- API 214 – Mitigating APIs Improper Asset Management

- COD 241 – Creating Secure Code Oracle Foundations
- COD 242 – Creating Secure SQL Server & Azure SQL Database Applications
- COD 246 – PCI DSS 3: Protecting Stored Cardholder Data
- COD 247 – PCI DSS 4: Encrypting Transmission of Cardholder Data
- COD 248 – PCI DSS 6: Develop and Maintain Secure Systems and Applications
- COD 249 – PCI DSS 11: Regularly Test Security Systems and Processes
- COD 256 – Creating Secure Code Ruby on Rails Foundations
- COD 261 – Threats to Scripts
- COD 287 – Java Application Server Hardening
- COD 288 – Java Public Key Cryptography
- CYB 210 – Cybersecurity Incident Response
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 212 – Fundamentals of Security Information & Event Management (SIEM)
- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- CYB 250 – Cyber Threat Hunting: Tactics, Techniques, and Procedures (TTP)
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 207 – Mitigating OWASP API Security Top 10
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 212 – Architecture Risk Analysis and Remediation
- DES 217 – Securing Terraform Infrastructure and Resources
- DES 219 – Securing Google's Firebase Platform
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 250 – Secure Software Acceptance and Deployment
- DES 261 – Securing Serverless Environments
- DES 262 – Securing Enterprise Low-Code Application Platforms
- DES 272 – OWASP M2: Mitigating Insecure Data Storage
- DES 273 – OWASP M3: Mitigating Insecure Communication
- DES 274 – OWASP M4: Mitigating Insecure Authentication
- DES 275 – OWASP M5: Mitigating Insufficient Cryptography
- DES 276 – OWASP M6: Mitigating Insecure Authorization
- DES 277 – OWASP M7: Mitigating Client Code Quality
- DES 278 – OWASP M8: Mitigating Code Tampering
- DES 279 – OWASP M9: Mitigating Reverse Engineering
- DES 280 – OWASP M10: Mitigating Extraneous Functionality
- DSO 212 – Fundamentals of Zero Trust Security
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- ENG 212 – Implementing Secure Software Operations
- TST 206 – ASVS Requirements for Developers
- LAB 114 – Identifying Cookie Tampering
- LAB 115 – Identifying Reflective XSS
- LAB 116 – Identifying Forceful Browsing
- LAB 117 – Identifying Hidden Form Field
- LAB 118 – Identifying Weak File Upload Validation

- LAB 119 – Identifying Persistent XSS
- LAB 120 – Identifying XML Injection
- LAB 123 – Identifying Vertical Privilege Escalation
- LAB 129 – Identifying Error Message Containing Sensitive Information
- LAB 133 – Identifying Exposure of Sensitive Information Through Environmental Variables
- LAB 134 – Identifying Plaintext Storage of a Password
- LAB 137 – Identifying Improper Authorization

Elite

- API 351 – Securing Kubernetes in the Build and Release Stages
- COD 383 – Protecting Java Backend Services
- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- CYB 311 – Threat Analysis with AI
- DES 313 – Hardening a Kubernetes Cluster
- DES 314 – Hardening the Docker Engine
- ENG 311 – Attack Surface Analysis and Reduction
- ENG 312 – How to Perform a Security Code Review
- ENG 320 – Using Software Composition Analysis (SCA) to Secure Open-Source Components
- TST 303 – Penetration Testing for Google Cloud Platform
- TST 304 – Penetration Testing for AWS Cloud
- TST 305 – Penetration Testing for Azure Cloud
- LAB 612 – ATT&CK: Network Service Identification
- LAB 613 – ATT&CK: Vulnerability Identification Using Vulnerability Databases
- LAB 615 – ATT&CK: Updating Vulnerable Java Web Application Server Software
- LAB 616 – ATT&CK: Host Vulnerability Scanning
- LAB 617 – ATT&CK: Testing for Plaintext Secrets in Files
- LAB 618 – ATT&CK: Log Analysis
- LAB 619 – ATT&CK Exfiltration Over C2 Channel
- LAB 620 – ATT&CK: Exploitation of Remote Services (Advanced)
- LAB 621 – ATT&CK: Password Cracking
- LAB 622 – ATT&CK: Exploiting Windows File Sharing Server Eternal Romance Remote Services
- LAB 623 – ATT&CK: Exploiting Vulnerable Java Web Application Server Software
- LAB 624 – ATT&CK: Exploiting Java Web Application Server Misconfiguration
- LAB 625 – ATT&CK: Exploit Public-Facing Application (Advanced)
- LAB 626 – Using and Exploit Framework for SQL Injection
- LAB 627 – Using and Exploit Framework for Port Scanning
- LAB 628 – Using and Exploit Framework for SMB version Scanning
- LAB 629 – Using and Exploit Framework for SNMP Scanning
- LAB 630 – ATT&CK: Exploiting Java SQL Injection to Extract Password Hashes
- LAB 631 – ATT&CK: Network Service Discovery
- LAB 632 – ATT&CK: Network Share Discovery
- LAB 634 – ATT&CK: Create Account
- LAB 635 – ATT&CK: Unsecured Credentials
- LAB 636 – ATT&CK: Data from Local System
- LAB 637 – ATT&CK: Valid Accounts

CSC 343 – Secure Systems Leadership

The Secure Systems Leadership Learning Path includes a variety of security courses designed for those responsible for computers and their complex operating systems. The curriculum explores application security best practices necessary to ensure strategies and plans support business needs and align with departmental and organizational objectives and goals. Learners will gain comprehensive application security knowledge and skills necessary for leading application development and design projects.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

32 Courses

0 Labs

11 Hours

14 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 151 – Fundamentals of the PCI Secure SLC Standard

Advanced

- API 251 – Implementing Web Application and API Protection (WAAP)
- COD 287 – Java Application Server Hardening
- COD 288 – Java Public Key Cryptography
- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- DES 206 – Meeting Cloud Governance and Compliance Requirements
- DES 219 – Securing Google's Firebase Platform
- DES 232 – Mitigating OWASP 2021 Injection
- DES 233 – Mitigating OWASP 2021 Identification and Authentication Failures
- DES 234 – Mitigating OWASP 2021 Cryptographic Failures
- DES 235 – Mitigating OWASP 2021 Insecure Design
- DES 236 – Mitigating OWASP 2021 Broken Access Control
- DES 237 – Mitigating OWASP 2021 Security Misconfiguration
- DES 238 – Mitigating OWASP 2021 Server-Side Request Forgery (SSRF)
- DES 239 – Mitigating OWASP 2021 Software and Data Integrity Failures
- DES 240 – Mitigating OWASP 2021 Vulnerable and Outdated Components
- DES 241 – Mitigating OWASP 2021 Security Logging and Monitoring Failures
- DES 262 – Securing Enterprise Low-Code Application Platforms
- DSO 201 – Fundamentals of Secure DevOps
- DSO 212 – Fundamentals of Zero Trust Security
- TST 206 – ASVS Requirements for Developers

Elite

- COD 383 – Protecting Java Backend Services
- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- DES 311 – Creating Secure Application Architecture
- DSO 301 – Orchestrating Secure System and Service Configuration
- DSO 302 – Automated Security Testing
- DSO 303 – Automating Security Updates
- DSO 305 – Automating CI/CD Pipeline Compliance
- ENG 320 – Using Software Composition Analysis (SCA) to Secure Open-Source Components

CSC 344 – Secure Development Manager

The Secure Development Manager Learning Path includes a variety of security courses designed for those responsible for planning, preparing, and ensuring that projects are completed. The curriculum introduces application security best practices required to adhere to system and information security policies and compliance. Learners can apply these best practices to the requirements, design, and implementation phases of the software development lifecycle.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

30 Courses

0 Labs

11 Hours

13 CPE Credits

Core

- AWA 101 – Fundamentals of Application Security
- AWA 102 – Secure Software Concepts
- DES 101 – Fundamentals of Secure Architecture
- DES 151 – Fundamentals of the PCI Secure SLC Standard
- ENG 110 – Essential Account Management Security
- ENG 114 – Essential Risk Assessment
- ENG 117 – Essential Information Security Program Planning
- ENG 151 – Fundamentals of Privacy Protection
- ENG 191 – Introduction to the Microsoft SDL
- ENG 192 – Implementing the Agile Microsoft SDL
- ENG 193 – Implementing the Microsoft SDL Optimization Model
- ENG 194 – Implementing Microsoft SDL Line of Business
- ENG 195 – Implementing the Microsoft SDL Threat Modeling Tool

Advanced

- CYB 211 – Identifying and Protecting Assets Against Ransomware
- CYB 213 – Generative AI Privacy & Cybersecurity Risk
- DES 208 – Defending Against the CSA Top 11 Threats to Cloud Computing
- DES 250 – Secure Software Acceptance and Deployment
- DES 255 – Securing the IoT Update Process
- DES 260 – Fundamentals of IoT Architecture and Design
- DSO 201 – Fundamentals of Secure DevOps
- ENG 205 – Fundamentals of Threat Modeling
- ENG 211 – How to Create Application Security Design Requirements
- TST 206 – ASVS Requirements for Developers

Elite

- CYB 310 – Using Cyber Supply Chain Risk Management(C-SCRM) to Mitigate Threats to IT/OT
- DES 361 – Mitigating LCNC (Low-Code/No-Code) Account Impersonation

- DES 362 – Mitigating LCNC (Low-Code/No-Code)) Authorization Misuse
- DES 364 – Mitigating LCNC (Low-Code/No-Code) Authentication and Secure Communication Failures
- DSO 302 – Automated Security Testing
- DSO 305 – Automating CI/CD Pipeline Compliance
- ENG 320 – Using Software Composition Analysis (SCA) to Secure Open Source Components

CSC 345 - Go Developer Elite

The Secure Go Application Developer Elite Learning Path designed to equip learners with the knowledge and skills necessary to develop secure software applications across various programming environments. This learning path blends theoretical concepts with practical hands-on labs, ensuring learners gain a deep understanding of security best practices, potential vulnerabilities, and effective mitigation strategies.

This path covers a broad spectrum of topics, including secure coding practices for Go, Angular, React, Java, JavaScript, jQuery, Python, Ruby on Rails, and Oracle DB Applications. It places a strong emphasis on securing APIs, web applications, backend services, and microservices against a wide array of security threats. Additionally, the learning path dives into specific security concerns related to the Kubernetes API, AJAX-enabled web applications, and Python scripting, among others.

The labs section is designed to provide hands-on experience in identifying and defending against vulnerabilities in Node.js and Go applications. These labs tackle common security issues like SQL Injection, XSS, SSRF, command injection, and more, offering learners the opportunity to apply knowledge gained in a practical setting.

By the end of this journey, learners will have an in-depth understanding of how to develop secure applications and protect against common and advanced security threats.

Each learning path may consist of course content that is not covered as part of certification exams. These courses are considered elective training and suggested based on our alignment with the National Initiative for Cybersecurity Education **(NICE) Cybersecurity Workforce Framework. To understand how courses map to this framework, please contact us.*

Primary Training Details

20	30	14	17
Courses	Labs	Hours	CPE Credits

Objectives

Upon successful completion of this learning path, learners will:

- Understand the Fundamentals of Application Security
- Gain a solid foundation in the principles of secure application development across different programming languages and frameworks.
- Learn to identify and defend against common security threats such as SQL Injection, XSS, CSRF, and more, with a particular focus on mitigating OWASP Top 10 API security risks.
- Acquire specialized knowledge in writing secure code in Go, along with other backend programming languages
- Implement secure coding practices such as secure by design, incorporating best practices in access control, encryption, and error handling to prevent security breaches.
- Gain practical experience in identifying application vulnerabilities and implementing effective security measures to mitigate them.
- Learn to integrate security practices within the development and operations processes, ensuring continuous delivery of secure applications.
- Equip yourself with the skills and confidence to tackle real–world security challenges

Core

- LAB 122 – Identifying Insecure APIs
- LAB 132 – Identifying Exposed Services

Advanced

- API 210 – Mitigating APIs Lack of Resources & Rate Limiting
- API 211 – Mitigating APIs Broken Object Level Authorization
- API 213 – Mitigating APIs Mass Assignment
- API 214 – Mitigating APIs Improper Asset Management
- COD 214 – Creating Secure GO Applications
- COD 241 – Creating Secure Oracle Database Applications
- COD 251 – Defending AJAX–enabled Web Applications
- COD 255 – Creating Secure Code–Web API Foundations
- COD 256 – Creating Secure Code Ruby on Rails Foundations
- COD 257 – Creating Secure Python Web Applications
- COD 265 – Secure Python Scripting
- COD 267 – Securing Python Microservices
- COD 285 – Developing Secure Angular Applications
- COD 286 – Creating Secure React User Interfaces
- COD 288 – Java Public Key Cryptography
- DES 207 – Mitigating OWASP API Security Top 10
- LAB 213 – Defending Node.js Applications Against Credentials in Code Medium
- LAB 217 – Defending Node.js Applications Against Business Logic Error for Input
- LAB 223 – Defending Node.js Applications Against SQL Injection
- LAB 226 – Defending Node.js Applications Against Forceful Browsing
- LAB 233 – Defending Node.js Applications Against XSS
- LAB 242 – Defending Node.js Applications Against eXternal XML Entity (XXE) Vulne
- LAB 245 – Defending Node.js Applications Against Plaintext Password Storage
- LAB 246 – Defending Node.js Applications Against Weak AES ECB Mode Encryption
- LAB 247 – Defending Node.js Applications Against Weak PRNG
- LAB 248 – Defending Node.js Applications Against Parameter Tampering
- LAB 262 – Defending Node.js Applications Against Sensitive Information in Error
- LAB 265 – Defending Node.js Applications Against Sensitive Information in Log Fi
- LAB 269 – Defending Node.js Applications Against Deserialization of Untrusted Da
- LAB 273 – Defending Node.js Applications Against SSRF
- LAB 277 – Defending Node.js Applications Against Command Injection
- LAB 281 – Defending Node.js Applications Against Dangerous File Upload
- LAB 285 – Defending Node.js Applications Against RegEx DoS
- LAB 291 – Defending Node.js Applications Against Path Traversal
- LAB 329 – Defending Go Applications Against SSRF
- LAB 333 – Defending Go Applications Against Hard–coded Credentials
- LAB 338 – Defending Go Applications Against CSRF
- LAB 339 – Defending Go Applications Against Path Traversal
- LAB 343 – Defending Go Applications Against Command Injection
- LAB 345 – Defending Go Applications Against Incorrect Authorization

Elite

- API 351 – Securing Kubernetes in the Build and Release Stages
- COD 352 – Creating Secure JavaScript and jQuery Code
- DSO 304 – Securing API Gateways in a DevSecOps Framework
- DSO 306 – Implementing Infrastructure as Code

- LAB 308 – Defending Node.js Applications Against Weak Password Reset
- LAB 350 – Defending Go Applications Against SQL Injection
- LAB 352 – Defending Go Applications Against Cross-Site Scripting
- LAB 354 – Defending Go Applications Against Improper Authentication