Implementing A Java-specific AppSec Program

How UNext Implemented A Niche Cybersecurity Program For A Tech Enterprise

unext

Overview

Securing and safeguarding a building is not just about deploying guards in entrance gates. It's also about incorporating CCTV cameras, smoke detectors, emergency exits, fire extinguishers and water hoses to ensure 360-degree safety and protection of people inside.

Cybersecurity demands a similar approach.

As one of the top **12 countries** that has been a consistent target for ransomware attacks, India detected over **200,000 such instances in 2023** alone. Attackers seamlessly detect vulnerabilities in file encryption or an application's loopholes and extort money or something more precious – data.

Forrester research also revealed that over **42% of the companies** that reported external attacks were able to pin-point **lack of application security** as the primary reason. Another GitLab survey shares that **developers fail to identify 75% of the vulnerabilities.**

Since it takes months to detect a loophole and develop a patch for it, stakeholders now prefer the incorporation of cybersecurity aspects in software development lifecycle.

One prominent market player intended to fast-track this approach and instill cybersecurity awareness and competencies in its developer workforce. To conceptualize, develop, and draft a solid learning intervention in cybersecurity, **it approached UNext.**

Let's find out how UNext understood the requirements and delivered an impeccable program for the partnering enterprise.

The Opportunity Statement

Reactive is the general modus operandi of most organizations when it comes to cybersecurity. It is only after an attack that vulnerabilities are detected, and reverse engineered for patches. Understanding the cost in terms of money, data, and reputation at stake, the partnering enterprise was keen on taking a proactive approach.

It wanted to address the issue at the grassroots level and secure applications it developed right from the coding phase. While simultaneously developing airtight applications, it aimed at fostering a culture of clean coding and awareness of its engineering workforce's culture.

The enterprise wanted us to implement a solid training program to upskill its IT teams in Cybersecurity with practical hands-on skills, to assess security vulnerabilities, and protect applications from various threats.

Core Learning Outcomes

- **Introduction To Cyber Threat and Attacks**
- **Preventing Cybersecurity Breaches**
- **Detecting Vulnerabilities & Potential Intrusions**
- **Respond & Recover From Cyber Attacks**

Learner Profile

Engineers with 2 to 10 years of work experience

Pedagogy

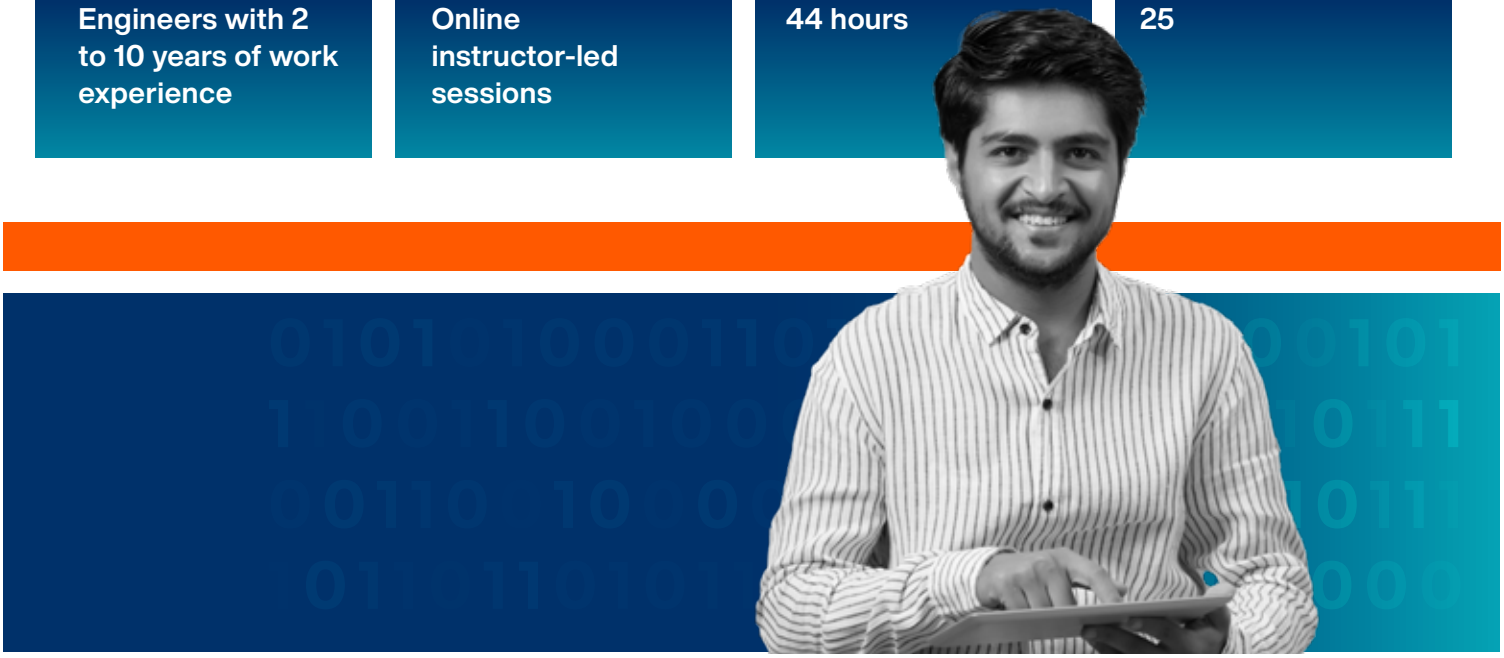
Online instructor-led sessions

Duration

44 hours

Total Number Of Learners

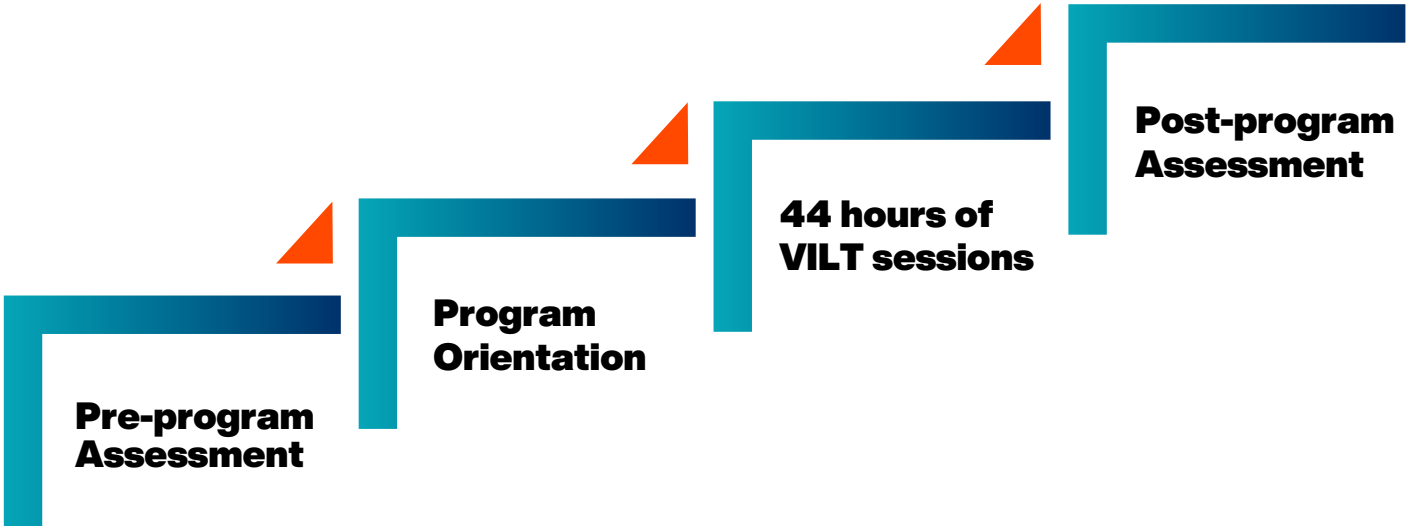
25



The Solution

UNext is prominent for its implementation of systematic workforce development programs. We step away from offering generic, one-size-fits-all training programs for cohorts, that generally dismiss the diverse sensibilities, academic backgrounds, and tech exposure every learner would come from in participating teams.

That's why we meticulously craft a 4-step program flow that features proven modules:



Pre-program Assessments

Due to the fact that learners were from diverse backgrounds and experiences, it was critical for us to get a clear idea of their current tech exposure and capabilities. That's why the preliminary step was to deploy am MCQ-based pre-program assessment to baseline participants' knowledge.

Program Orientation

From the results of the assessment, we understood the capability quotient of the cohort. This allowed us to map outcomes better and craft a curriculum from scratch that would minimize the learning curve for all participants.

VILT Sessions

We deployed a 44-hour VILT program for learners to ensure maximum knowledge transfer and learning impact.

Post-program Assessment

This is the standout milestone of our talent transformation programs, where we conduct stringent assessments to give you a clear visualization of which professionals are project-ready and which ones need additional handholding. For this, we rolled out MCQ based questions to gauge the depth of knowledge participants have acquired through this intervention.

Program Curriculum

Topic	Subtopic
Application Security Fundamentals	▪ Secure SDLC ▪ Introduction to Secure Software Development ▪ Secure Software Development Life Cycle processes ▪ Software Security Frameworks and Architecture ▪ SANS Cloud Security Architecture Principles ▪ Requirements ▪ System Quality Requirements Engineering Steps (SQUARE) ▪ Security Requirements ▪ Compliance Requirements ▪ Identify Risks and Prioritize Vulnerabilities ▪ Design ▪ Design and Architectural Principles for Secure Software Development ▪ Secure Design Patterns ▪ Patterns and Design Strategies for Secure Applications ▪ Threat Modelling ▪ Attack Surface Reduction ▪ Coding ▪ Secure Coding Principles ▪ Secure Code Review and Analysis ▪ Safe Programming Libraries ▪ Static Analysis ▪ Testing ▪ Security testing of Software ▪ Static Application Security Testing ▪ Dynamic Application Security Testing ▪ Review and Validation ▪ Deployment ▪ Secure Configurations ▪ Hardening ▪ Final Reviews

Topic	Subtopic
Application Information Security Controls	▪ Historical Vulnerability Trend Of Application
Application Security – Java Applications	**JAVA Application Vulnerabilities** ▪ Cross-Site Scripting (XSS) exploit and countermeasures ▪ Testing and prevention of Input Validation and XSS (Cross ▪ Site Scripting) ▪ Testing and prevention of Buffer Overflow and Command Injection ▪ Directory Traversal exploit and countermeasures ▪ HTTP Response Splitting exploit and countermeasures ▪ Parameter Manipulation exploit and countermeasures ▪ XML Injection attack ▪ SQL Injection attack ▪ Command Injection attack ▪ XPATH Injection attack ▪ Injection Attacks and their countermeasures
	Introduction to Java Security Platform) ▪ Overview of Java Security ▪ Overview of Java Security Platform and its components ▪ Java Security Sandbox ▪ Java Virtual Machine (JVM) and internal security model ▪ Java Security Manager and Policies ▪ Java Security Framework ▪ Java Authentication and Authorization Service (JAAS)
	Security in Java I/O Package ▪ Introduction to the java.io package ▪ Overview of Java File Input and Output ▪ File creation and Access Privileges to Files in Java ▪ Error handling guidelines for File-related errors ▪ Best Practices of File Input/Output usage ▪ Serialization Implementation guidelines ▪ Methods of Serialization ▪ Usage of Security Manager Class with check methods ▪ Java's Memory Management and Resource Leaks ▪ Java's built in Garbage Collection

Topic	Subtopic
Application Security – Java Applications	**Authentication and Authorization** ▪ Implementing Encryption and Certificates in Client Application ▪ Authentication: Weaknesses and Prevention ▪ Introduction to Authorization ▪ Session Management in Web Applications ▪ Java Authentication and Authorization (JAAS) ▪ JAAS Features ▪ JAAS Architecture ▪ JAAS's implementation of Pluggable Authentication Module (PAM) Framework ▪ JAAS Classes ▪ JAAS: Subject and Principals objects ▪ Authentication implementation in JAAS
	Java Cryptography ▪ Introduction to Cryptography ▪ Digital Signatures and Certificates in JCA ▪ Digital Signature Tool: DigiSigner ▪ Understanding Secure Socket Layer (SSL) connection in Java ▪ Understanding Keys and Certificates ▪ Key Management System in Java ▪ JAR Tools: Jarsigner, Jcryp Tool ▪ Java Cryptography Tools ▪ Best Practices for Java Cryptography
	▪ Java Concurrency and Session Management ▪ Overview of Java Concurrency/Multithreading ▪ Working with a Thread ▪ Use of ThreadPool instead of ThreadGroup ▪ Secured Practices for Handling Threads ▪ Overview of Session Management in Java ▪ Vulnerabilities of Session ▪ Types of Session Hijacking Attacks ▪ Checklist to Secure Credentials and Session Ids ▪ Guidelines for Secured Session Management

Topic	Subtopic
Application Security – Java Applications	**Java Error Handling and Logging** ▪ Overview of Exception and Error Handling ▪ Threats due to inappropriate Exception handling ▪ Best Practices for Handling Exceptions in Java ▪ Overview of Logging in Java ▪ Log4j and Java Logging API
	OWASP Top 10 ▪ Introduction to the OWASP Top 10 ▪ Broken Access Control ▪ Cryptographic Failures ▪ Injection ▪ Insecure Design ▪ Security Misconfiguration ▪ Vulnerable and Outdated Components ▪ Identification and Authentication Failures ▪ Software and Data Integrity Failures ▪ Security Logging and Monitoring Failures ▪ Server-Side Request Forgery (SSRF)
API Security	▪ Authentication and Authorization recommendations specific to rest-based applications ▪ SAML / JWT authentication best practices
DevSecOps Security	▪ DevOps Principles ▪ DevSecOps Principles

Key Program Highlights

- A handpicked cohort of 25 learners for equal-attention knowledge transfer
- Experiential modules comprising of real-world and industry-specific assignments, case studies, and capstone projects
- Access to sandbox labs tailored for each business unit-specific use cases for hands-on exposure
- Transparent reporting systems to track individual learner performance and programs across every single module and metric
- In-person classes and sessions from industry veterans and more

The Result

Thanks to the outcome-competency mapping protocol UNext deploys, we were able to precisely understand and deliver on requirements presented by the participating enterprise. The program seamlessly blended theoretical and practical aspects and fostered an experiential learning journey thanks to our sandbox environments as well.

The engineers now showcase substantial cybersecurity awareness when developing and coding, resulting in reduced vulnerabilities. This has also led to a significant decrease in development rework by eliminating back and forth between testing and patching.

Reduced software development cycles, improved security postures, and optimized trust in the market have been some of the key outcomes, thanks to the program.

If you intend to incorporate cybersecurity practices in your developer workforce or deploy a resilient tech workforce, reach out to us today. We can customize a special cybersecurity program based on your enterprise requirements.

About UNext

UNext (Part of Manipal Education and Medical Group) prioritizes 360-degree talent transformation through upskilling. We offer industry-relevant programs that help enterprises transform their talent through customized learning solutions across hierarchies. We have partnered with clients across IT-ITeS / BFSI / Automobiles / Healthcare / Manufacturing / Consulting / Retail / Pharma & more segments in training tech and non-tech audiences across various customized programs.

15+ years
of providing learning solutions

40,000+
Pre-joining programs

20,000+
Corporate bootcamps

6,000+
Role based programs

We have partnered with clients across IT-ITeS / BFSI / Automobiles / Healthcare / Manufacturing / Consulting & more segments in training tech and non-tech audiences in emerging technologies through various customized programs.

Transform
Your Workforce With Us

By tailoring programs for diverse domains and market segments across distinct functional roles, we offer the most practical and relevant workforce transformation programs in the market. Our program ecosystems are designed to seamlessly tackle massive volumes of simultaneous cohorts so you can precisely implement your workforce transformation goals. Reach out to us today.

Reach out to us at
+91 90198 87000

Write to us for more details
corporate.solutions@u-next.com

unext