Technical Support Intern Checkmarx (A Product-Based)

Internship Report

Submitted for the partial fulfillment of the degree of

Bachelor of Technology

In

Computer Science & Design

Submitted By

Anushree Sharma

0901CD211010

UNDER THE SUPERVISION AND GUIDANCE OF

Prof. Amit Kumar Manjhvar

Assistant Professor

Department of Computer Science & Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA
माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत
(Deemed to be University)
NAAC ACCREDITED WITH A++ GRADE

January-May 2025

DECLARATION BY THE CANDIDATE

I hereby declare that the work entitled **Internship at Checkmarx** is my work, conducted under the supervision of **Vered Litman Somekh, Director of Support EMEA APAC,** during the session January-May 2025. The report submitted by me is a record of bonafide work carried out by me.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Anushree Sharma

0901CD211010

Date: 21/05/2025
Place: Gwalior

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Guided By:

Prof. Amit Kumar Manjhvar
Assistant Professor
Department of Computer Science & Engineering
MITS, Gwalior

Departmental Project Coordinator

21/05/25

Dr. Kuldeep Narayan Tripathi
Computer Science and Engineering
MITS, Gwalior

21/05/25

Approved by HoD

Dr. Manish Dixit
Professor & HOD
Department of CSE
MITS, Gwalior

Dr. Manish Dixit
Computer Science and Engineering
MITS, Gwalior

PLAGIARISM CHECK CERTIFICATE

This is to certify that I, a student of B.Tech. in **Computer Science and Engineering,** have checked my complete report entitled **"Technical Support Intern at Checkmarx"** for similarity/plagiarism using the "Turnitin" software available in the institute.

This is to certify that the similarity in my report is found to be 25% which is within the specified limit (30%).

The full plagiarism report along with the summary is enclosed.

Anushree Sharma

0901CD211010

Checked & Approved By:

Prof.Mahesh Parmar
Assistant Professor
Computer Science and Engineering
MITS, Gwalior

ABSTRACT

This report presents an overview of the six-month internship undertaken by Me, Anushree Sharma at Checkmarx as a First Line Technical Support (FLS) Engineer Intern. Checkmarx is a global leader in application security, delivering solutions like SAST, DAST, SCA, and more to ensure secure development from code to cloud. The internship focused on application security, technical support, and real-world troubleshooting of security tools. The document outlines the learnings, outcomes, and contributions of the candidate in assisting global enterprise clients and working with security-focused development environments.

The core objective of this internship was to build a solid foundation in application security, understand real-world enterprise support workflows, and provide meaningful resolutions for customer issues. The internship enabled the student to work on tools like SAST, DAST, and SCA and understand AppSec integration in SDLC environments. With guidance from experienced mentors and self-learning initiatives, the candidate gained valuable insights into AppSec operations and support.

During the course of this internship, I was entrusted with critical responsibilities, including **is sue reproduction across client tenants, reverse engineering real-world scan failures, handling authentication and API errors, debugging scan failures in SAST, DAST, and SCA**, and contributing to overall **client integration troubleshooting in CI/CD environments**. These tasks not only strengthened my analytical and technical proble m-solving skills but also allowed me to gain firsthand experience with enterprise-

grade tools such as **Postman, Azure DevOps (ADO), Jenkins, GitHub Actions, CxFlow, ZAP, Kibana, SonarQube, and CLI- based operations**.

I would like to express my deepest gratitude to my **Industry Mentor, Mr. Reuben Mathias**, **FLS Team Lead at Checkmarx**, for his consistent mentorship and technical guidance throughout the internship. His expert knowledge, patience, and encouragement were instrumental in enhancing my understanding of secure development, client-

facing operations, and real-world application security strategies.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Prof. Amit Kumar Manjhwar, Assistance Professor, Department of Computer Science, Prof. Manish Dixit, Head of the Department, Department of Computer Science and Engineering, MITS Gwalior, for his invaluable guidance, encouragement, and support throughout the duration of this internship.

I also extend my heartfelt thanks to my mentors and the entire team at Checkmarx for providing me with the opportunity to learn and grow in a challenging, security-focused professional environment. Special thanks to the whole Support team at Checkmarx for their patience, support, and hands-on training throughout this journey.

This internship has significantly contributed to my academic and personal development, and I am truly grateful for the opportunity.

Anushree Sharma

0901CD211010

CONTENT

Table of Contents

CHAPTER 1: INTRODUCTION

In today's rapidly evolving digital landscape, the increasing frequency and sophistication of cyberattacks have made it imperative for organizations to prioritize security at every stage of their software development lifecycle (SDLC). Security can no longer be an afterthought; it must be seamlessly integrated from the earliest phases of application design through deployment and beyond. Recognizing this urgent need, Checkmarx, a global leader in application security solutions, provides a comprehensive suite of tools and technologies aimed at empowering organizations to detect, assess, and remediate security vulnerabilities across their applications.

As part of my internship at Checkmarx, I had the opportunity to work as a First Line Support Engineer at the L2 level, delivering timely and effective product-related and application security (AppSec) solutions to clients. My role involved direct engagement with clients to understand and resolve their technical challenges, ensuring optimal usage of Checkmarx's tools. Through this experience, I gained in-depth exposure to the company's flagship offerings, including Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA). These solutions play a critical role in helping organizations identify and address vulnerabilities in both the code they write and the third-party components they integrate.

This internship not only enhanced my technical proficiency in application security practices but also provided valuable insights into client-facing support operations, real-world security concerns, and the practical implementation of secure coding standards. It allowed me to bridge the gap between theoretical knowledge and industry practices, reinforcing the importance of proactive security measures in today's software-driven world.

CHAPTER 2: LITERATURE SURVEY

Application Security (AppSec) is a critical discipline within the broader field of cybersecurity, focusing on the identification, remediation, and prevention of vulnerabilities in software applications. As software systems become increasingly complex and integral to business operations, ensuring their security has become paramount to protect sensitive data, maintain business continuity, and safeguard organizational reputation.

AppSec encompasses a variety of methodologies and tools designed to proactively identify and address security weaknesses throughout the Software Development Lifecycle (SDLC). This chapter explores the core methodologies employed in AppSec, namely Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA).

Static Application Security Testing (SAST)

SAST is a white-box testing method that analyzes an application's source code, bytecode, or binary code for vulnerabilities without executing the program. By examining the application's internal structures, SAST identifies coding errors, insecure coding practices, and potential vulnerabilities at an early stage of development. This allows developers to remediate issues before the application moves into later stages of the SDLC, thereby reducing remediation costs and enhancing overall software quality. SAST is particularly effective in identifying issues such as SQL injection, cross-site scripting (XSS), and buffer overflows.

Dynamic Application Security Testing (DAST)

DAST, on the other hand, is a black-box testing approach that evaluates applications in their running state. Unlike SAST, DAST does not require access to the source code. Instead, it simulates external attacks on a live application to identify security flaws that could be exploited by malicious actors. DAST is instrumental in uncovering issues related to authentication, authorization, data leakage, and server configuration errors. It provides insights into how an application behaves under real-world attack conditions, helping organizations strengthen their defensive mechanisms against runtime vulnerabilities.

Software Composition Analysis (SCA)

Modern applications heavily rely on open-source components and third-party libraries, which can introduce hidden vulnerabilities if not properly managed. Software Composition Analysis (SCA) tools automate the identification of open-source components within an application and cross-reference them against known vulnerability databases. SCA helps organizations detect and remediate risks associated with outdated or vulnerable dependencies, ensuring compliance with licensing requirements and minimizing security risks associated with third-party code.

Integrated Security Solutions

Leading security platforms, such as those offered by Checkmarx, integrate SAST, DAST, and SCA into a unified framework, enabling organizations to adopt a holistic approach to application security. By combining multiple testing methodologies, these solutions provide comprehensive coverage across the entire application landscape, from proprietary code to third-party components, and from development environments to live deployments. This integrated approach ensures that security is embedded into the software development process, promoting a "shift-left" security mindset where vulnerabilities are addressed as early as possible.

In summary, the convergence of SAST, DAST, and SCA practices underlines the importance of adopting a multi-layered security strategy to effectively protect applications against evolving threats. Tools like those provided by Checkmarx empower organizations to not only detect and mitigate vulnerabilities but also to build security into the foundation of their software development practices.

CHAPTER 3: COMPANY PROFILE

Checkmarx is a global leader in **Application Security (AppSec)**, renowned for empowering organizations to develop and deploy secure software with confidence. Headquartered in **Israel**, with offices and operations across **North America, Europe, Asia-Pacific, and India**, Checkmarx has become the trusted partner for thousands of enterprises and government agencies aiming to fortify their software development lifecycle (SDLC) against modern security threats.

Since its founding, Checkmarx has focused on transforming how security is integrated into development pipelines by delivering innovative, developer-friendly security testing solutions. The company provides a **unified platform** for **Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Software Composition Analysis (SCA)**, **Container Security**, and **Infrastructure as Code (IaC) security**, enabling seamless integration into modern DevOps and CI/CD workflows.

The company's flagship product, **Checkmarx One**, is a cloud-native platform designed to provide **end-to-end visibility and control** over application security risks. It supports **API security testing**, supply chain risk management, and custom policy enforcement across large enterprise environments. By integrating early into the development process, Checkmarx helps organizations "shift left" and detect vulnerabilities before they reach production.

Checkmarx is trusted by over 1,800 customers worldwide, including **45 of the Fortune 100**. The organization has been consistently recognized in industry analyst reports, including **Gartner Magic Quadrant for Application Security Testing**, and has a strong emphasis on **developer enablement, risk management, and scalability**.

The company also provides comprehensive support services through its **First Line Support (FLS) teams**, which work closely with enterprise clients to resolve integration issues, troubleshoot security scan failures, and fine-tune configurations across SAST, DAST, and CI/CD implementations. With a diverse workforce of cybersecurity professionals and engineers, Checkmarx continues to lead the charge in making software security efficient, scalable, and developer-centric.

CHAPTER 4: ROLES AND RESPONSIBILITY

Key Responsibilities:

During the onboarding phase, I completed structured training on Checkmarx's platforms Salesforce for case management, Jira for issue tracking, and the Checkmarx One portal for vulnerability analysis alongside exposure to Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), API Security, Supply Chain Security, Software Composition Analysis (SCA), Malicious Package Protection, Container Security, and Infrastructure as Code (IaC) Security. This foundation enabled me to handle an increasing variety of support requests, underpinned by established service-level agreements and standardized workflows.

Throughout the internship, I observed and then independently managed a series of "shadowed" cases to solidify my understanding of core support processes. Subsequently, I addressed more challenging incidents across authentication, API integration, server-side troubleshooting, scanning configuration, and advanced debugging domains. This hands-on engagement not only enhanced my technical skill set but also cultivated essential soft skills effective communication, time management, and collaborative problem-solving that are vital in a global software- security environment.

1. Troubleshooting Customer Issues:

A major part of my role involved diagnosing and resolving customer-reported issues across Checkmarx's product portfolio, including Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) platforms.

- Performed detailed analysis of error logs, product configurations, and network environments to isolate root causes.
- Utilized internal tools and databases to identify known issues, patches, and recommended best practices.
- Provided customers with step-by-step troubleshooting guidance, configuration adjustments, and patch management instructions.

2. Case Management and Escalation:

I was responsible for managing escalated support tickets through the Checkmarx customer relationship management (CRM) system.

- Prioritized tickets based on severity (e.g., critical production outages, high-impact vulnerabilities) and customer SLAs (Service Level Agreements).
- Maintained clear, detailed, and professional communication with customers throughout the ticket lifecycle.
- Escalated unresolved or complex issues to L3 (Engineering) teams, ensuring all necessary diagnostic data, environment replication steps, and initial analyses were attached for faster resolution.

3. Environment Replication and Lab Testing:

To accurately replicate and troubleshoot customer-reported issues, I set up test environments in internal labs.

- Simulated client configurations using tools such as Docker, Kubernetes clusters, CI/CD pipelines, various IDEs, and Checkmarx-specific plugins (e.g., Checkmarx for Jenkins, Azure DevOps, GitHub integrations).
- Recreated various scanning scenarios, vulnerability detections, scan failures, and deployment errors.
- Documented reproduction steps and findings systematically to support Engineering teams and enhance internal knowledge bases.

4. Collaboration with Development and Engineering Teams:

A critical part of my role involved cross-functional collaboration with Checkmarx's Development, Quality Assurance (QA), and Product teams.

- Participated in regular case review meetings to discuss high-priority or recurring issues.
- Provided feedback to development teams based on real-world customer experiences and reported bugs.
- Acted as a liaison between customers and engineering, ensuring that critical

vulnerabilities or feature requests were communicated effectively.

5. Root Cause Analysis and Reporting:

I contributed to performing detailed Root Cause Analysis (RCA) for recurring or complex incidents.

- Analyzed underlying architectural flaws, misconfigurations, or overlooked use cases that led to issues.
- Helped prepare RCA reports to share insights with customers, internal stakeholders, and QA teams for process improvement.

6. Knowledge Base Contribution:

To improve support efficiency, I documented solutions, workarounds, and common troubleshooting techniques.

- Authored and updated internal Knowledge Base (KB) articles with troubleshooting guides, FAQs, and environment setup instructions.
- Helped new team members and customers by providing easily accessible technical documentation.

7. Customer-Focused Communication:

Throughout the internship, I maintained a strong focus on customer satisfaction.

- Delivered prompt, clear, and professional communication.
- Tailored technical explanations according to the client's level of expertise, ensuring both technical and non-technical stakeholders could understand proposed solutions.
- Followed up with clients to validate issue resolution and gather feedback on the support experience.

CHAPTER 5: CHECKMARX ONE – OUR PRODUCT

Checkmarx One is a cloud-native application security platform designed to consolidate multiple security testing modalities into a single, unified console. It is recognised as a Leader in Gartner's Magic Quadrant for Application Security Testing and is trusted by over 1,700 organisations worldwide . The platform delivers end-to-end vulnerability management from code commit through to cloud deployment by integrating seven core scanner modules under one interface .

Platform Interface and Administration

The browser-based console provides a centralised workspace where security teams can onboard applications, configure policies and review scan results without switching tools. Role-based access control, SAML and OpenID Connect authentication, and API-key management are all administered centrally, simplifying user governance and audit compliance.

Integrations and Developer Enablement

Checkmarx One integrates natively with leading CI/CD systems such as Jenkins, Bamboo, Azure DevOps, GitHub Actions and TeamCity via dedicated plugins and a command-line interface. A single pipeline step can invoke all scanners for a project, enforcing consistent, automated testing across modalities. First-party IDE plugins for Visual Studio, IntelliJ, VS Code and Eclipse allow developers to run scans and view findings directly in their development environment, embedding security early in the SDLC.

Automation and APIs

Beyond built-in CI/CD and IDE integrations, Checkmarx One offers REST APIs and webhook support for custom workflow automation. Security gates and automated remediation steps can be enforced within ticketing systems like Jira and ServiceNow, ensuring that vulnerabilities are tracked and managed as part of the wider DevOps process.

Analytics and Reporting

The AppSec Posture Management (ASPM) module consolidates scan histories, vulnerability trends and remediation metrics into interactive dashboards. Users can generate custom reports, schedule automated exports, produce executive summaries and generate SBOMs. Detailed audit logs and compliance records provide transparency and accountability for stakeholders at all levels.

CHAPTER 6: COMMON ISSUE TYPES AND SOLUTIONS

During the course of handling first-line support responsibilities, several recurring categories of technical issues emerge. These issues require a thorough understanding of system architecture, access protocols, integration methods, and software configurations. The following outlines the most common types of issues encountered, along with their typical nature, impact, and approaches used to investigate and address them.

1. Authentication Errors

Authentication-related issues are among the most frequently reported. These may involve login failures, access denials, token expiration, misconfigured identity providers, or LDAP/SAML- related configurations.

Typical Scenarios:

- Incorrect or expired login credentials.
- Misconfigured Single Sign-On (SSO) using LDAP or SAML.
- Token-based authentication failures due to expired or malformed tokens.
- OAuth 2.0 flow misalignment or configuration mismatches.

Resolution Approach:

- Verifying identity provider configurations and authentication logs.
- Reproducing the authentication issue in a test environment.
- Coordinating with internal IAM teams to validate token issuance or session policies.
- Providing corrective configuration steps or patch updates.

Tools Used:

- Postman (for token testing)
- Logs from Kibana
- Identity and Access Management platforms
- CLI (for token generation and validation.

2. API Issues

API-related problems often involve failed requests, unexpected status codes (such as 400, 401, 403, 500), incorrect payloads, or integration errors between systems.

Typical Scenarios:

- REST API call returning unexpected errors or no data.
- Integration with external services failing due to incorrect endpoint usage or timeout issues.
- CORS-related problems preventing browser-based API access.
- Version mismatches or deprecated endpoints being called.

Resolution Approach:

- Using Postman to replicate and debug failing API requests.
- Inspecting request headers, body, and response codes.
- Comparing API behavior across environments (e.g., Dev vs. Prod).
- Validating authentication headers and permissions associated with API keys or tokens.
- Reviewing API documentation and code integration (from GitHub or IntelliJ IDE).

Tools Used:

- Postman
- VS Code / IntelliJ
- API Gateway dashboards
- Swagger / OpenAPI Specs
- Logs from Kibana or Jenkins pipelines (for API errors during CI/CD)

3. Server and Backend Errors

These issues typically stem from infrastructure-related problems or failures in backend services such as databases, application servers, or internal services.

Typical Scenarios:

- Server returning 500 Internal Server Error or 503 Service Unavailable.
- Database connectivity failures due to misconfigured connection strings or expired credentials.
- Timeouts or memory overflows during processing.
- Configuration conflicts between services or misrouted requests.

Resolution Approach:

- Analyzing server logs using Kibana to detect crash reports or stack traces.
- Verifying service availability and uptime status through monitoring dashboards.
- Reproducing errors in a local environment using containerized setups (e.g., Docker).
- Collaborating with the DevOps team to trace deployment logs or rollback problematic builds.

Tools Used:

- Jenkins (for deployment pipelines)
- Kibana (for backend log monitoring)
- Docker (for reproducing server environments)
- CI/CD pipelines
- CLI (for accessing server logs and services)

4. Setup and Configuration Assistance

Users often need guidance on setting up tools, configuring integrations, or initializing scans and workflows, especially during onboarding or upgrades.

Typical Scenarios:

- Errors in configuring SAST, SCA, or DAST scans.
- Missing dependencies or incompatible plugin versions.
- Misconfigured CI/CD pipeline causing scan failures.
- Confusion in setting up GitHub Actions or integrating with Jenkins.

Resolution Approach:

- Providing step-by-step setup documentation or screen-sharing sessions.
- Validating YAML or JSON configuration files.
- Sharing sample configurations and command-line flags for correct usage.
- Checking compatibility of local tools and environments with platform requirements.

Tools Used:

- GitHub, GitHub Actions
- Jenkins and CI/CD pipelines
- VS Code / Visual Studio / IntelliJ
- Confluence (for setup guides)

CxFlow and CLI (for automated scan integr

CHAPTER 7: TECHNICAL SKILLS AND TOOLS USED

My internship as a First Line Support (FLS) Engineer Intern at Checkmarx provided me with the opportunity to work with a variety of technical tools and frameworks essential for application security, troubleshooting, and support management. Below is a detailed overview of the technical skills and tools I used throughout my internship:

1. Application Security Testing Tools (SAST & DAST):

- **Static Application Security Testing (SAST):**

I used SAST tools to analyze source code for vulnerabilities before execution. This involved identifying issues like SQL injection, Cross-Site Scripting (XSS), and hardcoded credentials during the development phase, enabling developers to remediate flaws early in the software lifecycle.

- **Dynamic Application Security Testing (DAST):**

DAST tools were employed to scan running applications for vulnerabilities. By simulating attacks on live environments, I identified issues such as broken authentication, sensitive data exposure, and insecure configurations.

- These tools were integrated into CI/CD pipelines to automate security scans, providing real-time feedback to development teams and promoting secure coding practices.

2. Case Management with Salesforce:

- I utilized **Salesforce** as the primary Customer Relationship Management (CRM) tool for handling support tickets.
- Tasks included:
  - Logging customer-reported issues with detailed descriptions and diagnostics.
  - Tracking ticket status and ensuring adherence to Service Level Agreements (SLAs).
  - Maintaining clear and consistent communication with customers through Salesforce's case management interface.
- The tool facilitated seamless collaboration with internal teams by allowing me to share updates, attach diagnostic files, and escalate unresolved issues to higher support tiers or development teams.

3. SQL for Database Queries and Troubleshooting:

- I frequently worked with **SQL** to address database-related queries and troubleshooting tasks.
  - Queried and analyzed customer databases to identify inconsistencies, configuration errors, or missing data that could cause application issues.
  - Assisted in optimizing queries to improve the performance of security scans and data retrieval processes.
  - Conducted basic data migration tasks and validated data integrity during issue resolution.

4. IIS and SSL Certificates for Secure Environment Configuration:

- **Internet Information Services (IIS):**
  - Configured IIS for hosting secure web applications and services. This included setting up bindings, managing application pools, and troubleshooting deployment issues.
- **SSL Certificates:**
  - Managed the installation and renewal of SSL/TLS certificates to ensure secure communication channels.
  - Assisted customers in troubleshooting certificate-related issues, such as expired or mismatched certificates, and configuring HTTPS protocols for their environments.

5. Jenkins and Azure DevOps for Build Process Management:

- **Jenkins:**
  - Integrated Checkmarx SAST and DAST tools into Jenkins pipelines for automated security testing during the build process.
  - Diagnosed and resolved pipeline failures related to security scan configurations and plugin issues.
- **Azure DevOps:**
  - Supported customers in setting up and managing pipelines within Azure DevOps for seamless security testing and deployment.
  - Collaborated with teams to troubleshoot integration issues, such as connectivity errors, credential misconfigurations, and API compatibility.

Other Tools and Technologies:

1. Log Analysis Tools:

- Analyzed log files to identify error patterns and debug application issues. Tools included built-in system logs, application-specific logs, and external logging frameworks.

2. Networking and System Diagnostics:

- Used networking utilities such as Ping, Telnet, and traceroute to troubleshoot connectivity issues between customer environments and Checkmarx servers.
- Monitored and resolved issues related to firewalls, proxy servers, and load balancers affecting application performance.

3. Version Control Systems:

- Worked with Git-based repositories (e.g., GitHub, GitLab) to support customers with their version control and integration processes.

Skills Developed:

- **Problem-Solving:** Analyzed complex technical challenges and proposed actionable solutions.
- **Communication:** Bridged the gap between technical teams and non-technical stakeholders by translating complex concepts into clear, understandable language.
- **Time Management:** Effectively prioritized tasks to meet SLA deadlines while managing multiple ongoing tickets.

CHAPTER 8: SHADOWING AND INDIVIDUAL WORK

Statement of Confidentiality

This document details the results of a simulated professional vulnerability assessment performed during an academic cybersecurity internship. The exercise was designed to mirror a real-world corporate engagement, encompassing network reconnaissance, web application testing, API security evaluations, and configuration audits across various infrastructure components. The primary objective was to identify and document insecure configurations, common web- based vulnerabilities, and misconfigurations that could be exploited by malicious actors.

All organization-specific details—including company names, system hostnames, internal IP addresses, and domain identifiers—have been fully anonymized or replaced with fictitious equivalents. Any references to cloud or on-premise platforms have been generalized to protect sensitive information. This anonymization ensures no real production environment or proprietary assets are exposed, while still preserving the fidelity of the techniques and findings for educational purposes.

The vulnerabilities and misconfigurations described herein are based on authentic methodologies and tools commonly used by security professionals, but they were applied to a purpose-built, non-production environment crafted solely for training. This report is intended strictly for academic review, skills demonstration, and the development of secure deployment practices. The contents should not be used for unauthorized or malicious activities. All material is the intellectual property of the internship program and must be handled in accordance with applicable academic and ethical guidelines.

Case 1: SAST Scan Failure Due to Deprecated Custom Query Reference

Domain: SAST Queries **Case Number**: 00239977 **Case Type:** Scan Failing **Priority Level:** Medium **Status:** Resolved **Background & Context**

SAST (Static Application Security Testing) scans rely on pre-built and custom queries to analyze source code for vulnerabilities. In enterprise environments, many organizations extend base query sets by developing custom queries tailored to their coding standards or business logic. However, when the core SAST query set is updated (e.g., during platform version upgrades), dependent custom queries may break if they rely on deprecated or renamed functions.

Customer Issue Summary:

The customer contacted support after their SAST scan began failing post-upgrade. The scan logs showed compilation errors, terminating the scan prematurely with status code 65. The error provided was as follows:

```
Error in queries compilation: error CS0103: The name 'Find_Inputs_NoWindowLocation' does not exist in the current context
```

This message indicated a missing reference used within a custom query, which pointed to a now non-existent function.

Technical Investigation:

- A log review confirmed that the issue stemmed from a query compilation failure.
- The query Find_Inputs_NoWindowLocation was found to be deprecated in Checkmarx version **9.7.2**.
- The customer's custom queries were not updated to reflect this change and still referenced the outdated identifier.
- This issue was further verified by comparing against the official 9.7.2 documentation, which listed the function as removed or replaced.

Action Taken:

1. **Root Cause Identification**:
   The failing reference was identified inside a custom query script written by the customer. It was established that the function had been removed and no longer existed in the SAST engine's standard query library.
2. Consulted Updated Query Library:
   Using the Checkmarx documentation for version 9.7.2 (internally referenced), it was found that the suitable alternative for the removed function
   was Find_LocationDocument_Desanitized_Inputs.
3. Provided Guidance:
   The customer was advised to:

- Replace all references
  to Find_Inputs_NoWindowLocation with Find_LocationDocument_Desanitized_Inputs.
- Audit the rest of their custom queries to ensure compatibility with version 9.7.2.
- Re-run the scan post-correction.

Outcome:

The issue was resolved successfully after the customer updated their custom query references. A subsequent scan was completed without any errors. This restored the SAST scan pipeline and ensured that vulnerability detection continued as expected.

Case 2: Tailored SQL Query Development for Retrieving Active Projects in Checkmarx

Case Number: 00240797 **Domain:** SAST Backend Query Customization **Case Type:** Functional Support – Data Retrieval Logic Optimization **Priority Level:** Medium **Status:** Resolved

Customer Issue Overview:

The customer requested assistance with constructing an SQL query to extract a distinct list of Project IDs and corresponding Project Names from the Checkmarx database. The objective was to acquire a dataset that excludes deleted or deprecated projects and accurately reflects only the projects currently being used or scanned in the system.

Step-by-Step Breakdown of Communication & Technical Handling:

1. Initial Request and Response

The customer opened the case with a requirement to fetch a list of all projects with their IDs and names. The following query was provided:

```
1. SELECT DISTINCT
2.  [ID],
3.  [Name]
4. FROM [CxDB].[dbo].[Projects];
```

1.Technical Note:

This query accesses the Projects table in the Checkmarx database and retrieves all records without filtering on status (e.g., deleted or archived). It meets the base requirement but lacks precision for active project filtering.

2. Feedback and Requirement Clarification

The customer responded that the result included deleted projects and clarified they were interested only in “currently being scanned” projects.

Follow-up Query Provided

```
1. SELECT DISTINCT
2.     p.[ID],
3.     p.[Name]
4. FROM [CxDB].[dbo].[Projects] p
5. JOIN [CxDB].[dbo].[ScanRequests] sr
6.     ON p.ID = sr.ProjectID
7. WHERE sr.CompletedOn IS NULL;
```

Technical Insight:

This query joins the Projects and ScanRequests tables to return only those projects where the scan request has not yet been marked as complete. It serves to identify active scan sessions in real time.

3. Customer Feedback and Empty Output Analysis

The customer reported that the above query yielded an empty table. Upon review, it was explained that this behavior is expected if no scans are actively running at the time of execution. To clarify the logic, a short video demo was provided to demonstrate query output during an active scan.

Resolution Support Provided:

This helped the customer understand the limitation of the query in context and ensured transparency on how data is structured within Checkmarx's scanning engine.

4. Final Requirement Clarification

After internal review, the customer refined their request, stating that their actual goal was not to find currently scanning projects but rather all *active* (non-deleted) projects in Checkmarx. This clarified the difference between "in-use" and "currently scanning."

Final Query Provided:

```
1. SELECT
2.     [ID] AS ProjectID,
3.     [Name] AS ProjectName
4. FROM
5.     [CxDB].[dbo].[Projects]
6. WHERE
7.     is_deprecated = 0;
```

Technical Justification:

The is_deprecated flag within the Projects table is used by Checkmarx to mark projects as deleted or archived. A value of 0 ensures only valid, usable project records are returned.

Outcome and Closure:

The final query precisely matched the user's requirement by delivering an accurate and up-to- date list of active projects. The customer confirmed that the shared SQL script was correct and effective. The case was marked as resolved after confirmation and appreciation from the customer.

Skills Demonstrated:

- Deep understanding of Checkmarx DB schema
- Query optimization and SQL filtering
- Effective interpretation of client needs through iterative communication
- Proactive handling using examples and demo support
- Clarification of data output logic with transparency

CHAPTER 9: DOMAIN WISE WORK

Case 1: Successful Promotion of Java 17 Support in Checkmarx Bamboo Plugin

Case Number: 00238709 **Domain:** CI/CD Integration and Plugin Compatibility **Case Type:** Feature Request and Platform Compatibility Issue

Priority Level: High

Status: Feature Accepted and Scheduled for Roadmap Delivery (Q2 2025)

Summary of Achievement

As part of my engagement with a high-impact customer issue at Align, I successfully initiated and drove a critical feature request for Java 17 support in the Checkmarx Bamboo Plugin, which is now officially accepted and scheduled for release in Q2 2025. This resolution was vital for maintaining compliance, operational continuity, and strategic tooling alignment for a major enterprise customer.

Problem Overview

The customer, Align, reported a major roadblock: their attempt to upgrade Bamboo to version 10.2.1 failed due to incompatibility between Checkmarx's Bamboo Plugin and Java 17, the runtime required by the updated Bamboo platform. Since the plugin lacked Java 17 support, the customer’s entire CI/CD-driven SAST pipeline was disrupted.

They emphasized that it was not feasible to shift automation tools, citing the volume of applications, repos, and branches involved. The situation was exacerbated by regulatory mandates (e.g., FDA and APAC release approvals) that required regular automated scans.

My Contribution

- **Technical Root Cause Identification**: Pinpointed the plugin's incompatibility with Java 17 as the core issue behind the failed Bamboo upgrade.
    - Proactive Feature Request Creation:
    - Raised internal request: **PLUGINS-273** (support for OpenJDK 17).
    - Filed a dedicated customer feature request: **SAST-I-1409**.
    - Business Impact Advocacy:
    - Worked with the customer to articulate the business-critical need for automation.
    - Submitted a compelling justification highlighting the risk of scan gaps, non-compliance, and team overload.
    - Customer Assurance & Escalation Strategy:
    - Positioned the issue for roadmap review.
    - Engaged customer's CSM to escalate through proper leadership channels.

Result

Feature Request Promoted to Roadmap

The engineering and product teams acknowledged the urgency and officially promoted the feature request under PLUGINS-342 to the product roadmap, targeting a Q2 2025 release.

This achievement directly resulted from my detailed case handling, impact assessment, and alignment with cross-functional stakeholders.

Strategic Importance

This feature not only restores CI/CD scanning functionality for Align but also ensures continued trust in Checkmarx as a scalable solution in Java 17+ environments. It serves as a foundational step for other enterprise users encountering similar platform shifts.

Key Takeaways

- **Effective Feature Delivery Begins with Impact Communication**: Articulating business risk is as critical as identifying technical gaps.
- **Customer-Centric Advocacy Drives Product Evolution**: Strategic listening, proper documentation, and escalation were instrumental in transforming a blocker into a roadmap commitment.
- **Ownership Matters**: By taking end-to-end ownership—from diagnosis to request submission to stakeholder follow-up—I was able to influence the product direction positively and tangibly.

Case 2: Intermittent DAST API Scan Failure – "Scan Results are Empty"

Case Number: 00240085 **Domain:** DAST Issues (Web/API/Configuration) **Case Status:**

Resolved **Priority:** Critical (POV – Proof of Value)

1. Context and Problem Statement

During an ongoing Proof-of-Value (POV) engagement with a financial services client (*ExampleBank Ltd.*), a critical issue was reported wherein DAST API scans were **failing intermittently** without providing direct or actionable error feedback on the UI. The scan targeting the following endpoint failed consistently:

```
1. https://uat-esbapp1.examplebank.com:7845/v1/FP_InterestCertificates
```

This endpoint failed during scheduled DAST scans executed via CheckmarxOne. Logs retrieved from the platform did not show an immediately obvious fault; however, the client emphasized this was a **critical blocker**, and an in-person escalation meeting was already scheduled with their stakeholders.

As a comparative reference, another scan from the same environment targeting a different endpoint completed successfully:

```
1. https://uat-esbapp1.examplebank.com:7862/bagic/ealthPremiumCalBagic
```

The client provided all necessary artifacts including:

- Postman Collection (used in the scan)
- YAML Configuration file
- DAST scan logs

2. Diagnostic Breakdown

2.1 Scan Metadata

Scan Type	DAST API Scan
Failing Scan	Scan ID: e2c6bcf2-a485-4336-ace3-c762ac607a70
Environment ID	edca4194-3e31-4d6e-8d73-e9e9c6b5862e
Timestamp of Failure	2025-04-18 10:26:39
Successful Scan	Scan ID: 2e8e4dff-f69b-4be7-9898-2bf3f4010749
Environment ID (Successful)	28b4abde-f1fe-4cff-9cc6-dc292dee48df

---

3. Investigation Steps and Findings

3.1 Log Inspection – Platform & Kibana Logs

Upon deep-diving into the logs (both system-generated and from the internal Kibana console), the following critical error message was identified:


```
{
  "level": "error",
  "componentName": "dast",
  "componentVersion": "0.1.860",
  "serviceName": "dast-worker",
  "serviceVersion": "1.0.242",
  "loggerName": "OneLog.Go",
  "correlationId": "1082ce3d-246a-4505-94c0-869c6b0c0abd",
  "scanId": "e2c6bcf2-a485-4336-ace3-c762ac607a70",
  "tenantId": "cde08d58-c44b-41f2-815b-293085aa41f0",
  "environmentId": "edca4194-3e31-4d6e-8d73-e9e9c6b5862e",
  "error": "rpc error: code = Unknown desc = scan results are empty",
  "scanErrorCode": "6010151",
  "timestamp": "2025-04-18T10:26:39.946Z",
  "msg": "Failed to send results"
}
```


This indicated that although the scan was initialized, the **DAST worker component** was unable to obtain any actionable data, returning a fatal RPC error with code 6010151.

3.2 YAML & Postman Collection Review

- The YAML configuration file was syntactically and semantically correct.
- The Postman collection was well-formed and mapped to the correct URL path and methods (i.e., GET/POST structure was valid).
- Environment IDs and other API tokens were properly aligned.

There were **no syntactic or structural issues** in the setup files provided.

4. Root Cause Hypothesis

Given that:

- The configuration was correct
- A nearly identical scan on a different API port was successful
- The error was scan results are empty

The likely root cause was **network reachability or authorization issue** between the CheckmarxOne DAST scanning infrastructure and the client's internal API.

Thus, the following possibilities were examined:

1. **Firewall Restrictions** or **API Gateway Filters** blocking traffic from Checkmarx DAST IPs.
2. The target API (7845) being **restricted internally**, not exposed publicly unlike the successful scan on port 7862.
3. **Load balancer or application logic** responding inconsistently or returning empty 200s.

5. Action Taken

5.1 Communication with the Client I
reached out to the client and asked:
"Is the API you're testing publicly accessible? Can CheckmarxOne servers reach this API endpoint externally, or is it internal-only?"

This was done to confirm whether **DAST scan infrastructure has appropriate access** to the API.

5.2 Suggested Remediation Steps

I provided the client with a robust workaround using **DAST Docker CLI**, which can be executed internally from behind their firewall:

```
1. docker run \
2.   -e CX_APIKEY=<your-api-key> \
3.   -v "<INPUT_DIR>:/path" \
4.   -v "<OUTPUT_DIR>:/output" \
5.   checkmarx/dast:latest api \
6.   --config=<CONFIGFILE>  \
7.   --base-url=<BASE_URL>  \
8.   --environment-id=<ENVIRONMENT_ID>  \
9.   --output=<OUTPUT_PATH>  \
10.   --postman=<POSTMAN_COLLECTION>
```

This approach allows the DAST scan to run **locally** within the client's infrastructure while still sending the results securely back to the Checkmarx platform.

6. Resolution

Upon follow-up, the client confirmed:

- DAST IP addresses were previously not whitelisted.
- After updating their firewall and reverse proxy configurations, the scan succeeded using the original YAML and Postman collection.

The RPC scan results are empty error **no longer occurred**, confirming that the issue was indeed due to **network-level access denial**, not a configuration or scan engine fault.

7. Final Outcome

- ✅ **Issue Resolved:** Whitelisting DAST IPs fixed the intermittent scan failure.
- ❌ No CLI scan was ultimately required.
- ✅ The client was unblocked before their scheduled POV meeting.

8. Lessons Learned & Impact

Learning Point	Description
Network Whitelisting is Crucial	All external DAST scans depend on proper API exposure or firewall exceptions.
Logs Are Critical	Despite the UI showing no obvious error, system logs (especially Kibana) revealed the hidden cause of failure.
Preparedness Pays	Offering the CLI workaround ensured the client had a fallback option, demonstrating proactive support.
Scan Parity Matters	Comparing successful and failing scans across different ports/environments is a powerful diagnostic tool.

CHAPTER 10: CHALLENGES AND SOLUTIONS

During my internship as a First Line Support (FLS) Engineer at Checkmarx, I encountered several challenges that tested my technical knowledge, problem-solving abilities, and communication skills. However, each challenge also served as an opportunity to learn, adapt, and grow professionally. Below are some of the key challenges I faced and the strategies I implemented to overcome them:

1. Troubleshooting Complex Customer Issues in Real-Time

Challenge:

- Many customer issues involved intricate configurations, multi-layered deployments, and integrated third-party systems, making it difficult to quickly diagnose the root cause without extensive investigation.
- Clients often required immediate solutions due to tight project deadlines or ongoing critical security scans.

Solution:

- To address this, I proactively **replicated customer environments** in a controlled **lab setup** within our internal systems.
- I simulated the same configurations, network setups, and workflows based on the information provided by customers.
- By recreating the issues independently, I was able to **experiment safely**, test potential solutions without impacting customer operations, and validate fixes before suggesting them.
- This hands-on troubleshooting approach significantly reduced the resolution time and increased the accuracy of solutions provided.

2. Dealing with Limited Information and Incomplete Logs

Challenge:

- Customers sometimes submitted incomplete details or logs, making it difficult to fully understand the scope of the issue or reproduce it accurately.

Solution:

- I developed a structured **information-gathering checklist** to ensure I collected all necessary technical details upfront, such as software versions, deployment methods, error logs, network topology, and recent changes.
- Clear, guided communication via Salesforce cases helped customers provide more targeted information.
- When logs were insufficient, I guided customers step-by-step to enable advanced debugging modes or collect system diagnostics using specialized tools provided by Checkmarx.

---

3. Managing Ticket Prioritization and Escalations

Challenge:

- Balancing multiple active tickets, each with varying levels of urgency, posed a significant challenge, especially when critical escalations were involved.

Solution:

- I adopted a **prioritization framework** based on customer impact, SLA commitments, and internal escalation guidelines.
- For high-priority cases, I collaborated closely with **internal engineering and product teams** to ensure faster escalations and proactive updates to the customers.
- Regular meetings with the support and engineering teams helped me stay updated on escalated cases and deliver timely, transparent communication to clients.

4. Understanding and Adapting to New Technologies

Challenge:

- Working with complex tools like SAST, DAST, Jenkins, Azure DevOps, and cloud-native architectures required continuous learning and adaptation.

Solution:

- I dedicated time outside regular working hours to **self-study official documentation**, participate in **internal technical training sessions**, and **shadow senior engineers** to accelerate my learning curve.
- Hands-on practice through lab environments helped solidify my understanding of new concepts, enabling me to support customers more effectively.

---

5. Cross-Team Collaboration and Communication

Challenge:

- Collaborating with cross-functional teams such as Product, QA, and Engineering, sometimes across different time zones, introduced communication and coordination challenges.

Solution:

- I practiced clear, concise, and professional communication through emails, Salesforce notes, and regular status meetings.
- Setting clear expectations on timelines, next steps, and ownership at every stage of a case helped streamline collaboration and avoided misunderstandings.

Skills Strengthened Through Overcoming Challenges:

- **Root Cause Analysis and Problem-Solving**
- **Technical Communication and Documentation**

- **Time and Task Management**
- **Adaptability and Quick Learning**
- **Customer-Centric Approach and Professionalism**

By facing and successfully overcoming these challenges, I enhanced not only my technical proficiency but also my ability to deliver high-quality customer support in a fast-paced, security-driven environment. These experiences have equipped me with valuable skills that will serve as a strong foundation for my future roles in the cybersecurity and application security domains.

CHAPTER 11: SOCIAL RELEVANCE

3.1 Empowering Secure Software Development:

- Application security plays a vital role in ensuring safe and resilient software
- Through this internship, contributed to the secure development lifecycle of global enterprises
- Enabled Fortune 100 companies to proactively mitigate security threats

3.2 Reducing the Attack Surface:

- Support in resolving security tool issues leads to faster deployment of patches and updates
- Helps reduce risk exposure and potential data breaches

3.3 Bridging Knowledge Gaps:

- As a student intern, brought academic knowledge of InfoSec and Cyber Security into real-world impact
- Shared findings with teams and contributed to the learning ecosystem within Checkmarx

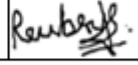
3.4 Career and Societal Preparedness:

- Strengthened foundation for a future career in Cybersecurity and AppSec
- Promoted the value of proactive security practices among peer networks

CHAPTER 12: MPRS & INTERNSHIP CERTIFICATE WITH STIPEND PROOF

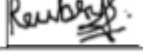
FORMAT

MONTHLY REPORT OF PROGRESS (MRP) FROM INDUSTRY MENTOR

Name of student	Anushree Sharma		Department	Technical Support	
Industry/Organization	Checkmarx		Date/Duration	06/01/2025 -07/02/2025	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge upgradation					✓
Performance/Quality of work					✓
Behaviour/Discipline/Team work					✓
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	I want to take a moment to commend Anushree for her outstanding learning and listening skills. She actively probes understanding the context and specifics. In situations where she faces challenges, she proactively consults with the team. I am confident that with continued effort, Anushree will become a valuable asset to any team.				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT** Excellent				
Name of Industry Mentor	Reuben Francis Mathias				
Signature of Industry Mentor					

Receiving Date		**Name of Faculty Mentor**		**Sign**	

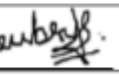
MONTHLY REPORT OF PROGRESS (MRP) FROM INDUSTRY MENTOR

Name of student	Anushree Sharma		Department	Techincal Support	
Industry/Organization	Checkmarx		Date/Duration	08/02/2025 -15/03/2025	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge upgradation					✓
Performance/Quality of work					✓
Behaviour/Discipline/Team work					✓
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	Anushree has demonstrated exceptional analytical and problem-solving skills in handling technical support cases. She approaches each issue with a structured mindset, quickly diagnosing problems and identifying effective solutions. Her ability to break down complex technical challenges, coupled with her proactive approach to seeking clarifications and collaborating with the team, ensures efficient resolution of cases. She consistently maintains a customer-focused approach, delivering clear and effective solutions. With her dedication and continuous learning, Anushree is well on her way to becoming a key asset in any Team.				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT** Excellent				
Name of Industry Mentor	Reuben Francis Mathias				
Signature of Industry Mentor					

Receiving Date	**16/02/2025**	**Name of Faculty Mentor**	**Dr. Amit Mangawar**	**Sign**	

FORMAT

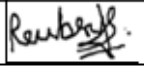
MONTHLY REPORT OF PROGRESS (MRP) FROM INDUSTRY MENTOR

Name of student	Anushree Sharma		Department	Techincal Support	
Industry/Organization	Checkmarx		Date/Duration	15/03/2025 -15/04/2025	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge upgradation					✓
Performance/Quality of work					✓
Behaviour/Discipline/Team work					✓
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	In just her third month of internship, Anushree has consistently outperformed expectations by closing more support cases than any other intern for two consecutive weeks. Her strong problem-solving skills are evident in how she tackles each technical challenge with precision and a structured approach. Anushree is especially proficient in working with APIs, CLI commands, and plugins—skills that have become her core strengths in resolving complex issues efficiently. She demonstrates a proactive attitude, often seeking clarity and collaborating effectively with the team to deliver timely and customer-focused solutions. With her dedication, quick learning ability, and technical acumen, Anushree is well on her way to becoming a vital asset to any team.				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT** Excellent				
Name of Industry Mentor	Reuben Francis Mathias				
Signature of Industry Mentor					

Receiving Date	**15/04/2025**	**Name of Faculty Mentor**	**Dr. Amit Mangawar**	**Sign**	

FORMAT

MONTHLY REPORT OF PROGRESS (MRP) FROM INDUSTRY MENTOR

Name of student	Anushree Sharma		Department	Technical Support	
Industry/Organization	Checkmarx		Date/Duration	16/04/2025 - 18/05/2025	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge upgradation					✓
Performance/Quality of work					✓
Behaviour/Discipline/Team work					✓
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	I'd like to take a moment to appreciate Anushree for her remarkable ability to learn and listen attentively. She consistently demonstrates a strong curiosity to understand the bigger picture and finer details of any task. Whenever she encounters difficulties, she doesn't hesitate to reach out and engage the team for support. With her proactive attitude and steady dedication, I'm confident that Anushree is on the path to becoming an indispensable contributor to any team she joins.				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT** Excellent				
Name of Industry Mentor	Reuben Francis Mathias				
Signature of Industry Mentor					

Receiving Date	**18/05/2025**	**Name of Faculty Mentor**	**Dr. Amit Manjvar**	**Sign**	

Checkmarx

12th December 2024

Anushree Sharma
Gwalior Madhya Pradesh

Dear Anushree,

Subject: Internship Letter ("Internship Letter")

Welcome to the Checkmarx Family!

Pursuant to your expression of interest and application to intern with us, we are pleased to offer you an internship opportunity with **Checkmarx India Technology Services Private Limited (**the "**Company**"). Your internship will be subject to terms and conditions as follows:

1. Stipend

During the Term of your internship, you will be entitled to a fixed monthly stipend of 25,000 INR (Twenty Five thousand only) ("**Stipend**"). This remuneration shall be subject to all the applicable tax and statutory deductions.

Apart from the Stipend, you shall not be eligible for any other performance or incentive bonus or any other employment benefits as are available to the other regular employees of the Company.

2. Joining Date, Designation and Term

Your Internship with the Company shall commence with effect from 6th January 2025 or such earlier/later date as agreed between you and the Company ("**Joining Date**") and will continue for a period of 6 month(s) thereafter ("**Term**") unless your internship is terminated for any reason whatsoever in accordance with the terms and conditions of this Internship Letter. If required, the Company may choose to extend this Term in writing, at its sole discretion by another period of 4 months and in all other cases the Term will be deemed to be expired at the end of the 6 month(s) period from the Joining Date.

Your designation will be **First Line Support Intern**. During the Term of your internship, you will report to Verad Litman and will have responsibilities and duties, as assigned by him/her.

Please note that this letter does not amount to an offer or commitment of employment. Upon completion of the Term, you will not be eligible to claim any employment or regularization with the Company.

3. Hours of Work

a) You will be required to work for nine (9) hours per day (*this is inclusive of the one (1) hour rest interval period*) from 9:00 am to 18:00 pm. Further, depending on project/ work contingencies, workload and business requirements, at any given time you may be required to work outside these stated hours, including weekends.

b) You may also be expected to travel to other locations and at times outside of your official working hours. You may at any time be called upon to perform other than your normal duties which in the opinion of

turnitin Page 1 of 55 - Cover Page

Anushree Sharma

Technical Support Intern at Checkmarx

My Files
My Files
Madhav Institute of Technology & Science

Document Details

Submission ID
trn:oid:::28506:96821327

Submission Date
May 20, 2025, 1:28 PM GMT+5:30

Download Date
May 20, 2025, 1:32 PM GMT+5:30

File Name
Internship Brief Report 0901CD211010 Anushree Sharma.docx

File Size
558.7 KB

50 Pages

7,112 Words

45,757 Characters

turnitin Page 1 of 55 - Cover Page


25% Overall Similarity
Filtered from the Report
Match Groups
Top Sources
Integrity Flags

Checkmarx

Date: 12 May 2025

Partial Completion Certificate

This is to certify that Ms. Anushree Sharma has partially completed the training and project as a part of the project in our company as mentioned below:

Project Tittle - Application Security Diagnostics and Tool Integration: Internship Experience with Checkmarx One

Date of Joining – Jan 6th, 2025

Date of partial completion – 20th May 2025

In partial fulfillment of VII semester project for B.Tech program of Madhav Institute of Technolody and Science, Gwalior.

DocuSigned by:
Shraddha Mishra
EBDBA513E66F45A...

Shraddha.Mishra
HRBP

Date - 5/16/2025
Place - Pune

CHAPTER 13: CONCLUSION

My internship at **Checkmarx** has been a transformative experience, offering me the opportunity to immerse myself deeply in the dynamic and critical field of **Application Security (AppSec)**. Through this role, I not only strengthened my theoretical understanding but also gained valuable **hands-on experience** in addressing real-world security challenges faced by global enterprises.

Working as a **First Line Support (FLS) Engineer Intern** exposed me to a wide range of technical scenarios involving **Static Application Security Testing (SAST)**, **Dynamic Application Security Testing (DAST)**, **Software Composition Analysis (SCA)**, and the security configurations of modern development environments. By managing customer cases, replicating complex environments, troubleshooting intricate issues, and collaborating across internal engineering teams, I honed my **technical expertise**, **problem-solving abilities**, and **communication skills**.

Additionally, working with state-of-the-art tools and platforms like **Salesforce**, **SQL**, **IIS**, **SSL certificates**, **Jenkins**, and **Azure DevOps** provided me with a comprehensive understanding of the technical ecosystems that support secure software development lifecycles (SDLC). I learned to approach security issues from a holistic perspective, considering not just code-level vulnerabilities but also infrastructure, integrations, and deployment pipelines.

The challenges I faced—ranging from incomplete logs to managing escalations under pressure—allowed me to develop a structured, analytical approach to problem-solving and emphasized the importance of adaptability, collaboration, and continuous learning in the fast-evolving cybersecurity landscape.

Beyond technical growth, my time at Checkmarx instilled in me a deeper appreciation for the broader mission of application security: **protecting critical systems, safeguarding user data, and fostering trust in digital experiences**. Being part of a company that plays a pivotal role in securing software across industries and geographies gave me a strong sense of purpose and motivation.

I am profoundly grateful for the mentorship, resources, and support provided by the Checkmarx team throughout my internship journey. This experience has significantly

contributed to my professional development and reinforced my aspiration to build a career in the field of **Cybersecurity and Application Security**. I look forward to applying the skills and knowledge I have acquired at Checkmarx to future roles and continuing to contribute to creating safer digital environments.

****