Inter-IIT Tech Meet 12.0
CERT-IN

Team - 41

December 15, 2023

Contents

Introduction

This research tackles three key concerns in the current environment of growing cyber dangers. First, we investigate methods to improve cybersecurity audit procedures, suggesting the creation of an application or tool to improve audit quality. Next, we discuss how to reduce vulnerabilities in products and apps proactively, with a focus on creating a security baseline. Finally, we look at a method for creating apps that are capable of successfully identifying, reporting, and addressing cyber threats. The study seeks to offer practical insights for strengthening corporate cybersecurity safeguards through these targeted actions.

1 Problem Statement 1

In our efforts to design an application to enhance the quality of audits, we designed a feature rich auditing solution which provides for all the necessities during a security audit.

AuditApp is designed with the objective to elevate the efficiency and caliber of audits, promoting seamless collaboration between auditors and auditees. Our program plans to streamline the adoption process by enabling it to integrate effortlessly with existing servers and Identity and Access Management (IAM) systems of the respective firms.

At its core, AuditApp serves as a dynamic interface that adapts itself to the unique business requirements of each firm, providing not just the tools but a comfortable environment for regulating security posture. Offering an intuitive bridge between auditors and firms, our software facilitates the hiring of auditors, equips them with essential auditing tools, and facilitates clear and efficient communication throughout the audit process.

Our application empowers auditors with a suite of features including customizable checklists for regulations followed, built-in communication tools, and interactive questionnaires. It enables automation of tedious tasks (such as writing a report including integration of all the evidence in a formal manner), allowing auditors to focus on their expertise rather than formalities. From generating comprehensive reports to visualizing data and performing intricate analytical operations, AuditApp aims to transform the audit process into a seamless, efficient, and insightful experience. A detailed overview of AuditApp's features is listed below, offering a glimpse into the wide array of functionalities.

1.1 Landing Page

A) **Accessible to:** Auditor, Auditee, and Regulator

B) **Functions:**

   i. A secure and seamless Single Sign-On (SSO) experience for users, integrating with diverse Identity and Access Management (IAM) systems, such as Active Directory or Okta allowing users to access the platform effortlessly without the need for redundant logins.

   ii. External Auditors are provided authenticated access during the audit period through the SSO mechanism, ensuring controlled login while aligning with stringent security protocols. This SSO integration not only enhances user convenience but also reinforces overall application security.

1.2 Audit Planning and Scheduling

A) **Accessible to:** Auditee (Upper Management)

B) **Functions:**

i. Enables the users to thoroughly plan both internal and external audits, introducing a dependency where external audits are dependent on the successful completion of internal audits. This ensures a systematic and well-structured audit framework.

ii. Offers timely notifications concerning new policy updates from regulations that they are already adhered to. These alerts inform the auditee to review the policy changes, enabling them to make informed decisions about whether adjustments are required and if a re-audit is necessary. This proactive approach facilitates swift responses and ensures ongoing compliance with evolving regulatory standards.

iii. Enables users to initiate external audits seamlessly with a single click, connecting to your chosen auditing institute without leaving the application. Also allows for easy access management for audit personnel to cloud servers in seconds with minimal steps.

iv. Presents the audit process visually, incorporating clear stages and progress indicators. Real-time updates on ongoing audits offer instant awareness of their progress, promoting transparency and efficiency in audit management.

v. Notifies if a certain percentage of employees still need to complete a required certification for cybersecurity awareness.

1.3 Audit Planning (Auditor Perspective)

A) **Accessible to:** Auditor

B) **Functions:**

i. Enables auditors with access to a planning interface, allowing them to set precise objectives, define scopes, and establish timelines for distinct topics within an audit. This feature ensures ease in managing multiple different factors of an audit.

ii. Offers a template featuring fundamental compliance guidelines that auditors can seamlessly edit and expand upon. This templated approach accelerates the planning phase while providing a solid foundation for compliance adherence.

iii. Enables auditors to assign tasks efficiently, distributing responsibilities among different team members.

1.4 Resources Page

A) **Accessible to:** Auditor and Auditee (Different versions)

B) **Functions:**

i. **Auditor Hub:** Provides a centralized resource combining extensive training materials for continual professional development with commonly used audit tools. From the latest best practices to vulnerability scanners and compliance assessment tools, this hub provides auditors with a one-stop solution, ensuring they stay informed and equipped for effective audits.

ii. **Auditee Hub:** Provides a dedicated section focused on cybersecurity training and awareness tailored for Auditees. This includes hands-on training modules, automated reminders for ongoing training, and a robust certification tracking system. The section aims to enlighten the employees with the knowledge and skills necessary to enhance cybersecurity awareness and compliance within the firm.

1.5 Communication Hub

A) **Accessible to:** Auditor, Auditee

B) **Functions:**

i. Contains tools to schedule secure online conferences among auditors and auditees.

ii. Contains modules which facilitate collaborative communication between auditors and auditees. This includes shared workspaces, discussion forums, and real-time messaging functionalities, promoting a cohesive and interactive environment for efficient collaboration throughout the auditing process.

1.6 Audit Data Collection

A) Accessible to: Auditor

B) Accessed By: Selecting a specific section in the Audit Planning page(Auditor Perspective)

C) Functions:

i. Auditors can actively engage in the audit process by posing detailed questions to the Auditee, regarding the functioning of various mechanisms within a company.

ii. These queries are seamlessly sent to the Auditee (viewed from a different perspective of the page), initiating a collaborative Q&A process. Auditees can respond with detailed answers and provide any necessary evidence to support their responses.

iii. The entire question-answer interactions is automatically recorded and stored in a structured manner, typically in a document or sheet format based on the auditor's preference and type of data collected.

iv. Auditors have the flexibility to categorize and save any provided evidence into specific sections or categories for organized documentation.

v. The results of past relevant tests, either carried out during the current or earlier audits, may also be effortlessly maintained within this section, providing a consolidated repository for full audit data.

1.7 Audit Data Analysis

A) **Accessible to:** Auditor, Auditee

B) **Accessed After:** Data Collection

C) **Functions:**

i. Enables auditors to thoroughly report instances of non-compliance or vulnerabilities identified during the data collection phase. This reporting feature serves as a critical component in the audit process, helps Auditee in the resolution and mitigation process for identified issues.

ii. Integrates a comprehensive Risk Matrix along with diverse risk metrics, providing auditors with a multifaceted approach to assess and communicate the severity of identified issues. This includes visualizations such as heatmaps, offering an intuitive representation of the risk landscape associated with each non-compliance or vulnerability. The incorporation of visual elements enhances the auditee's ability to make informed decisions effectively.

iii. Enables the auditor to present expert advice on effective mitigation strategies. This includes the option to incorporate additional information or evidence gathered during the audit process, ensuring a well-informed and actionable approach to issue resolution.This structured information can be effortlessly included in the report by selecting the evidence to be presented. The software will then organize the information in a coherent manner.

1.8 Audit Report

A) **Accessible to:** Auditor and Auditee

B) **Accessed After:** After all data Analytics are done

C) **Functions:**

i. A range of pre-designed templates is available to streamline the report creation process, ensuring consistency and professionalism. The auditor then utilizes computer-aided tools to meticulously prepare a comprehensive audit report. The software enhances the reporting process by recommending and facilitating the easy inclusion of evidence and scores in an intuitive manner.

ii. The completed audit report is seamlessly submitted and can be conveniently viewed from the auditee's perspective. Thus allowing auditees to access and review the finalized report efficiently.

iii. To ensure a mutual and precise understanding of the audit findings, an inbuilt dictionary is provided within the report. Auditees can effortlessly refer to this dictionary by hovering over specific terms, addressing potential discrepancies in interpreting sophisticated and precise auditor terminology. This promotes effective communication and collaboration between auditors and auditees.

1.9 Follow Up

A) **Accessible to:** Auditor and Auditee

B) **Accessed After:** After the report is submitted

C) **Functions:**

i. Upper Management takes charge by assigning identified vulnerabilities to respective departments, strategically prioritizing them based on their risk scores. This ensures a systematic and efficient approach to vulnerability resolution, addressing high-risk areas promptly and effectively.

ii. Departments, having received their assigned vulnerabilities, actively work on necessary changes and push updates. The structured process allows them to submit their changes for review, streamlining the remediation effort and ensuring accountability.

iii. When requested to review changes, Auditors meticulously check if the modifications complies with the regulations. If compliance is not met, they provide detailed feedback directly to the respective department. If the changes align with regulations, the compliance checklist is marked as done, signifying adherence to the required standards.

1.10 Audit Quality Assurance and Improvement

A) **Accessible to:** Auditor, Auditee, Regulators, or other concerned bodies

B) **Accessed After:** Initial Audit

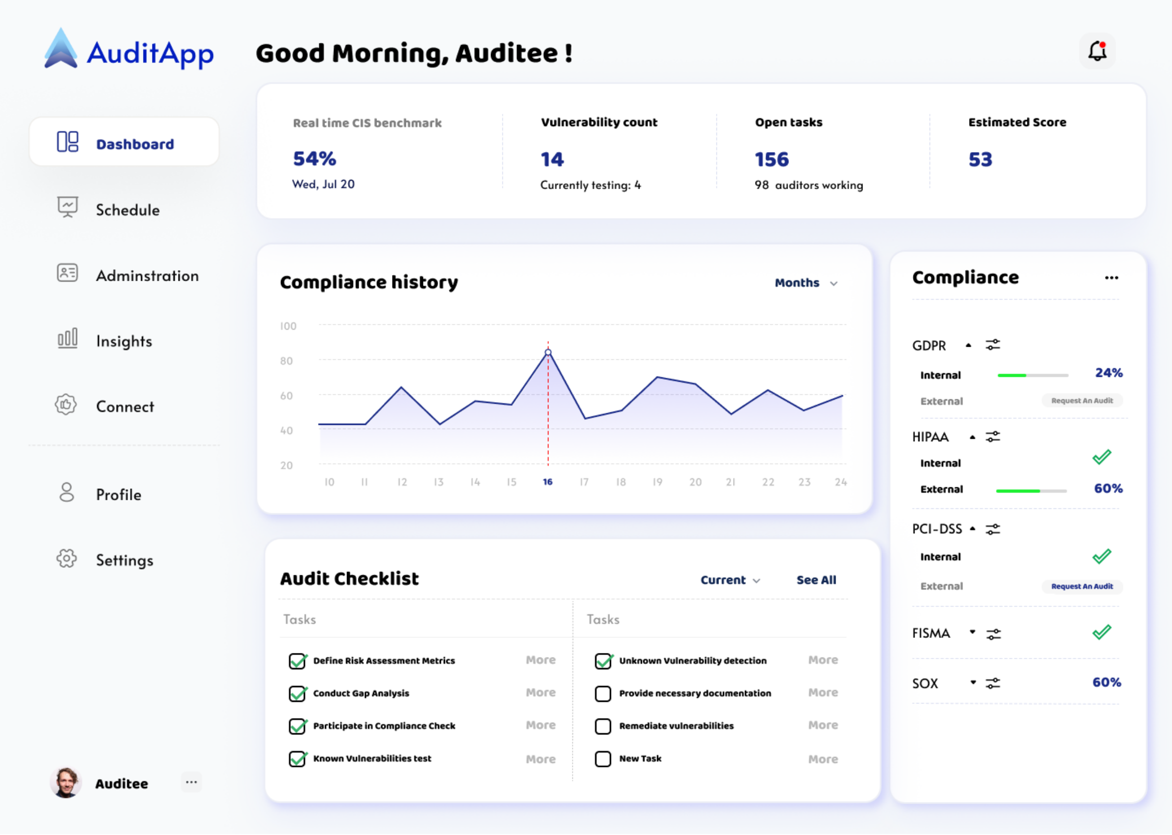
C) **Functions:**

i. Auditor provides a score ranging from 1-10 based on compliance laws, vulnerabilities, and severity.

ii. Upon meeting standards and compliance in the audit, a Certificate of Trust is issued, acknowledging the organization's commitment to robust security. It also visually reflects compliance with specific standards for informed decision-making by other bodies and regulators.

iii. All audit results are meticulously stored using version control mechanisms. This not only ensures a detailed historical record but also enhances transparency and trust in the auditing process. Stakeholders, including Auditors, Auditees, Regulators, and other concerned bodies, can refer to these archived results for a comprehensive understanding of the company's audit history, fostering a culture of continuous improvement and accountability.

With these aforementioned features, we want to make the tedious process of auditing more efficient and simple for auditors, auditees, and regulators. These were accomplished by developing innovative methods such as automation methods, enhanced data visualization, and the simplification of numerous tasks to just a few mouse clicks. Here are a few illustrations showcasing some of the mentioned features:



Good Morning, Auditee !

Real time CIS benchmark: 54%, Wed, Jul 20

Vulnerability count: 14, Currently testing: 4

Open tasks: 156, 98 auditors working

Estimated Score: 53

Compliance

GDPR: Internal 24%; External Request An Audit

HIPAA: Internal ✓; External 60%

PCI-DSS: Internal ✓; External Request An Audit

FISMA: ✓

SOX: 60%

Audit Checklist

Tasks

☑ Define Risk Assessment Metrics

☑ Conduct Gap Analysis

☑ Participate in Compliance Check

☑ Known Vulnerabilities test

Tasks

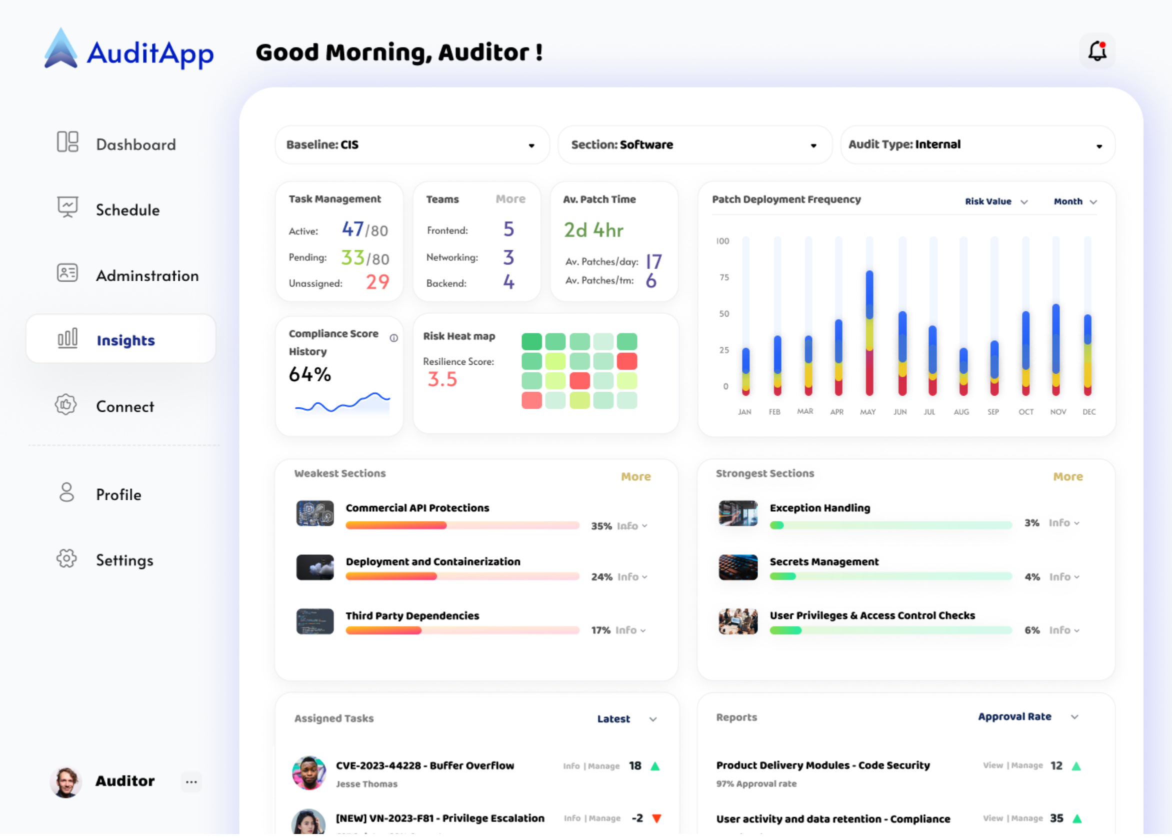
☑ Unknown Vulnerability detection

□ Provide necessary documentation

□ Remediate vulnerabilities

□ New Task

Figure 1: AuditApp Dashboard



AuditApp

Good Morning, Auditor !

Dashboard

Schedule

Adminstration

Insights

Connect

Profile

Settings

Auditor

Baseline: CIS

Section: Software

Audit Type: Internal

Task Management

Active: 47/80

Pending: 33/80

Unassigned: 29

Teams More

Frontend: 5

Networking: 3

Backend: 4

Av. Patch Time

2d 4hr

Av. Patches/day: 17

Av. Patches/tm: 6

Compliance Score History

64%

Risk Heat map

Resilience Score: 3.5

Patch Deployment Frequency

Weakest Sections More

Commercial API Protections 35% Info

Deployment and Containerization 24% Info

Third Party Dependencies 17% Info

Strongest Sections More

Exception Handling 3% Info

Secrets Management 4% Info

User Privileges & Access Control Checks 6% Info

Assigned Tasks Latest

CVE-2023-44228 - Buffer Overflow Info | Manage 18 ▲

Jesse Thomas

[NEW] VN-2023-F81 - Privilege Escalation Info | Manage -2 ▼

Reports Approval Rate

Product Delivery Modules - Code Security View | Manage 12 ▲

97% Approval rate

User activity and data retention - Compliance View | Manage 35 ▲

Figure 2: AuditApp Analytics

2 Problem Statement 2

In the dynamic world of cybersecurity, it's crucial to reinforce our software defenses against potential risks. This section focuses on a proactive strategy: creating security baselines for different parts of an application. The goal is straightforward—to prevent security vulnerabilities and reduce the chances of exploitation. By building strong security foundations, we aim to create software that can better withstand potential threats. This approach ensures that our defenses are ready and resilient, actively guarding against emerging cybersecurity issues.

2.1 Controls And Baselines

Control:

A control can be characterized as a high-level description delineating a feature or activity that requires attention. It transcends specificity regarding technology or implementation and serves as a comprehensive directive for addressing pertinent aspects.

Baseline:

A baseline constitutes the practical implementation of a control on individual services. This implementation provides a foundational framework for the enforcement and adherence to the specified control measures.

A security baseline is to defined spanning across various dimensions of the application/software developed. A good baseline should be free of conflicts, and provide separate detailed requirements for each section. Major sections where security guidelines and implementation requirements are to be defined include the following:

1. Network Security
2. Internally Developed Software Security
3. External Module Checks
4. Malware and Intrusion Defenses
5. Deployment Security
6. Access Controls
7. Data Transfer and Storage Security

Baselines should be consistent, and easily integrable into the development process of an application. Integrating baselines into the development sub-processes can be made easier by further prioritizing the baselines based on risk probabilities and relevance to the sub-processes. The following section delves into how this can be implemented efficiently.

2.2 Security Baseline for DevOps Lifecycle

The following security baseline provides a comprehensive framework for securing applications throughout the DevOps lifecycle. This adaptive guide focuses on preventing security vulnerabilities and minimizing the risk of exploitation. It is designed to be customizable based on organizational needs, technology stacks, and specific requirements.

A) **Planning Phase:**

   i. **Threat Intelligence Integration:**

      1) **Objective:** Integrate threat intelligence feeds to identify and prioritize potential threats during the planning phase.

      2) **Details:** Strengthen cybersecurity by utilizing centralized threat data management. Subscribe to diverse feeds, encompassing Open Source Threat Intelligence and commercial sources, to enrich threat data comprehensively. This strategic approach ensures a proactive and resilient cybersecurity stance, fortifying defenses against evolving threats.

   ii. **Compliance Requirements:**

      1) **Objective:** Identify and incorporate relevant regulatory and compliance requirements into the planning phase.

      2) **Details:** Efficiently manage compliance by automating checks. Regularly update compliance requirements to align with changes in regulations or industry standards. This automated approach ensures consistent adherence to compliance protocols, adapting swiftly to evolving regulatory landscapes.

B) **Coding Phase:**

   i. **Security Training:**

      1) **Objective:** Ensure developers undergo regular security training to enhance their awareness and skills.

2) **Details:** Facilitate skill development by offering access to online training platforms. Integrate code review tools to bolster secure coding practices. This comprehensive approach not only enhances knowledge acquisition through online platforms but also reinforces secure coding methodologies through robust code review. Additionally, provide specialized training modules addressing social engineering tactics, such as phishing awareness and simulated social engineering scenarios. Implement periodic simulated phishing exercises to enhance employee resilience and awareness against social engineering attacks.

ii. **Dependency Scanning:**

1) **Objective:** Use automated tools for dependency scanning to identify and address vulnerabilities in code dependencies.
2) **Details:** Identify vulnerabilities in third-party libraries by employing Dependency Scanning Tools. Implement Software Composition Analysis (SCA) Tools to assess and ensure the security of software components. This proactive strategy addresses potential risks associated with dependencies and reinforces a robust security posture.

iii. **Data Encryption Standards:**

1) **Objective:** Establish standards for encrypting sensitive data during the coding phase.
2) **Details:** Integrate encryption libraries to implement secure encryption algorithms. Ensure robust key storage and management using key management solutions. This comprehensive approach enhances data security by implementing strong encryption and establishing secure key management practices.

C) **Building and Compilation Phase:**

i. **Artifact Signing:**

1) **Objective:** Enforce the signing of build artifacts to ensure integrity and authenticity.
2) **Details:** Sign executable code with precision. Implement container image signing to sign container images systematically. This dual approach fortifies code integrity, assuring the authenticity of executable code and containerized applications in diverse environments.

ii. **Container Security:**

1) **Objective:** Implement container security practices during the building and compilation phase.
2) **Details:** Utilize container scanning tools to pinpoint vulnerabilities in container images. Implement container runtime security solutions to actively monitor and protect running containers. This dual strategy ensures a comprehensive approach to container security, addressing both static vulnerabilities in images and dynamic threats during runtime.

D) **Testing Phase:**

i. **Fuzz Testing:**

1) **Objective:** Integrate fuzz testing to identify potential security vulnerabilities by providing unexpected input.
2) **Details:** Implement fuzz testing frameworks to enhance software resilience through systematic testing. Utilize security testing platforms to meticulously identify vulnerabilities in application interfaces. This comprehensive testing approach strengthens the security posture, proactively addressing potential weaknesses in both application logic and external interfaces.

ii. **Security Champions Program:**

1) **Objective:** Establish a security champions program to advocate for secure coding practices.
2) **Details:** Establish collaboration platforms to facilitate communication and collaboration among security champions. Empower these champions with knowledge through training and awareness platforms. This dual-pronged approach not only enhances communication but also equips security champions with the necessary skills and awareness to contribute effectively to the overall security posture.

E) **Deployment Phase:**

i. **Configuration Management:**

1) **Objective:** Utilize configuration management tools for consistency in the deployment phase.
2) **Details:** Implement configuration management tools to automate configuration changes systematically. Leverage infrastructure as code tools to precisely define and provision infrastructure. This cohesive strategy ensures efficient and automated management of configurations and infrastructure, enhancing overall operational agility and consistency.

ii. **Secrets Management:**

1) **Objective:** Implement secure secrets management practices during deployment.
2) **Details:** Utilize secrets management solutions to securely store and manage sensitive information. Implement credential management tools to enforce privileged access management. This comprehensive approach ensures robust protection of secrets and privileged credentials, enhancing overall security posture and mitigating potential risks.

F) **Monitoring and Operation Phase:**

i. **User Behavior Analytics (UBA):**

1) **Objective:** Implement UBA tools to detect anomalous user behavior during the monitoring and operation phase.
2) **Details:** Implement UBA platforms for advanced threat detection through behavior analysis. Integrate security information and event management systems for centralized log management and comprehensive analysis. This synergistic deployment ensures a proactive stance against evolving threats and facilitates centralized monitoring and analysis of security events.

ii. **Vulnerability Remediation Process:**

1) **Objective:** Establish a process for addressing vulnerabilities during the monitoring and operation phase.
2) **Details:** Develop issue tracking and remediation platforms to systematically track and prioritize identified vulnerabilities. Implement automated remediation tools for swift and automated resolution of vulnerabilities. This integrated approach streamlines the vulnerability management process, ensuring efficient tracking, prioritization, and prompt remediation of security issues.

G) **End-of-Life Phase:**

i. **Data Erasure Standards:**

1) **Objective:** Define and adhere to industry-standard practices for secure data erasure during the end-of-life phase.
2) **Details:** Employ standard data destruction methods to guarantee secure and thorough data erasure. Implement secure file deletion tools for the permanent deletion of sensitive files. This meticulous approach ensures the irretrievable removal of data, contributing to robust data security practices.

ii. **Documentation Archiving:**

1) **Objective:** Archive security-relevant documentation for compliance purposes during the end-of-life phase.
2) **Details:** Utilize document management systems for structured and secure documentation. Implement version control systems to track changes made to security-related documents systematically. This combination ensures organized and secure documentation practices while allowing for transparent tracking of document revisions and updates.

H) **Continuous Improvement:**

i. **Threat Hunting:**

1) **Objective:** Introduce proactive threat hunting exercises for continuous improvement.
2) **Details:** Deploy threat hunting platforms for proactive threat detection through active hunting. Utilize network traffic analysis tools to monitor and analyze network behavior comprehensively. This dual strategy enhances the organization's ability to detect and respond to potential threats actively, ensuring a proactive stance against evolving cybersecurity risks.

ii. **Incident Response Drills:**

1) **Objective:** Conduct regular incident response drills for preparedness.
2) **Details:** Implement incident response platforms for streamlined incident handling processes. Utilize communication and collaboration tools to facilitate effective coordination during incident response. This integrated approach ensures efficient incident resolution and seamless collaboration among response teams, contributing to an agile and effective incident response strategy.

iii. **Security Automation:**

1) **Objective:** Explore automation for routine security tasks to enhance efficiency.
2) **Details:** Deploy security orchestration, automation, and response (SOAR) platforms for streamlined and automated incident response. Integrate CI/CD tools to automate security checks seamlessly within the development pipeline. This cohesive integration enhances overall incident response efficiency and automates critical security processes throughout the development lifecycle.

iv. **Regular Checks And Updates**

1) **Objective:** Identify anomalous behavior or outdated configurations for security systems.
2) **Details:** Conduct routine and comprehensive assessments of the attack surface at both individual and organizational levels. Perform regular audits on system logs to identify and address any potential malfunctions. Scrutinize configurations for outdated settings and ensure timely updates as needed. Additionally, review backup configurations whenever there are updates to the data storage policy.

In summary, our approach involves establishing a robust security baseline at every stage of the DevOps development process. This provides developers with key considerations to safeguard various aspects of an application. Emphasizing a proactive stance, we focus on preventing security vulnerabilities and minimizing the risk of exploitation through continuous and vigilant development.

3 Problem Statement 3

In today's world where cyber threats are increasing in frequency by the day, having swift detection and response techniques is key. This section outlines a strategic approach to application development, emphasizing the integration of advanced elements and techniques for detecting, reporting, and responding to cyber threats effectively. Through the strategic utilization of Security Information and Event Management (SIEM) systems, Extended Detection and Response (XDR), robust logging mechanisms, and other relevant approaches, our goal is to fortify applications against an array of potential attacks. Subsequent subsections will provide a detailed examination of key elements and a streamlined action plan for seamless implementation. This comprehensive strategy empowers organizations to enhance their security measures, ensuring the swift detection and response to any attempts of cyber threats.

3.1 Threat Modelling

- Threat modelling is a methodical and structured process aimed at identifying and assessing potential security threats and vulnerabilities within the context of a system, application, or process. This involves a comprehensive analysis of the system's components, a deep understanding of their interactions, and the proactive identification of potential points of compromise.
- Importance: Early identification of risks is crucial for proactively planning and implementing effective security measures. By systematically assessing potential threats, organizations can better anticipate and mitigate security risks before they can be exploited, enhancing the overall resilience of the system.

3.2 Following Secure Development Processes

To efficiently mitigate and detect potential cyber attacks, adopting practices that isolate vulnerabilities to specific modules is crucial. Here are some effective approaches:

3.2.1 Application And Network Segmentation

Application and network segmentation is a cybersecurity strategy aimed at organizing and isolating different components within a network to enhance security and minimize the impact of potential cyber threats. By dividing the network into segments or zones based on factors like sensitivity and criticality, organizations can create controlled environments that restrict lateral movement by attackers and limit the potential damage caused by security incidents.

- Assigning levels of importance and value to assets is an important first step for network segmentation.Labelling the data is also important to have segmentation in accordance to regulations.
- Network segmentation helps to improve network security by breaking the network into isolated segments.However there are legitimate dataflows which are to be permitted. Dataflows are for this purpose divided into North-bound,East-West,South-bound traffics.
- Certain assets within an organization's network are used for similar purposes and communicate regularly. Segmenting these systems off from one another does not make sense as a number of exceptions would be required to maintain normal functionality.
- Traffic between assets within a particular segment may be permitted to flow unrestricted. However, intersegment communications need to be monitored by the segment gateway and comply with access control policies. These policies should be defined based upon a principle of least privilege.

3.2.2 Proper Exception Handling

1. **Early Detection of Anomalies:** Integrated exception handling enables systematic error logging and monitoring, aiding in the detection of abnormal patterns indicative of cyber attacks.
2. **Incident Reporting and Documentation:** Exception handling, including incident reporting mechanisms and detailed logs, ensures efficient post-incident analysis, forensics, and compliance reporting during a cyber attack.
3. **Intrusion Detection System (IDS) Integration:** Integration of exception data with IDS enhances the ability to correlate and analyze potential security incidents, contributing to more effective detection and response.
4. **Automated Response Mechanisms:** Implementing automated responses triggered by specific exceptions can contain or mitigate ongoing cyber attacks, reducing manual response time.
5. **User Behavior Analysis:** Analyzing exceptions related to user interactions and access patterns helps detect unusual behavior, serving as an early indication of potential insider threats or compromised accounts.
6. **Legal and Compliance Obligations:** Proper exception handling ensures compliance with legal requirements for incident reporting, offering comprehensive logs as evidence for regulatory compliance and investigations.

3.2.3 Continuous Automated Analysis And Reviews

Integrate automated analysis tools into your development and deployment processes for early detection and resolution of vulnerabilities:

1. **SCA (Software Composition Analysis):** Identify vulnerabilities in third-party libraries by regularly scanning dependencies for known security issues. Early detection allows for timely updates or replacements.
2. **SAST (Static Application Security Testing):** Analyze source code for vulnerabilities without executing the program. Integrate SAST into the CI/CD pipeline for automated code reviews. Identify coding errors and misconfigurations early in development.
3. **DAST (Dynamic Application Security Testing):** Assess running applications through simulated real-world attacks. Automate DAST in testing to uncover vulnerabilities in deployed applications. Detect issues that may be missed during static analysis.
4. **CI/CD Integration:** Seamlessly integrate security checks into the CI/CD pipeline. Automate SCA, SAST, and DAST tools in the build and deployment process. Provide real-time feedback to developers, enabling prompt issue resolution.
5. **Automated Vulnerability Remediation:** Implement automated processes to fix obvious identified vulnerabilities. Integrate tools or scripts for automatic patching or updates. Expedite remediation, reducing exposure time to potential threats.

3.3 Integrating Proper Logging

1. **What to Log:** You should prioritize logging critical components of your network and business. This includes essential logs from your firewall, key servers such as the Active Directory server, and crucial application and database servers. Additionally, logs from your Intrusion Detection System (IDS), antivirus software, and web server should be closely monitored.

2. **Collecting Logs:** Logs are collected from various elements like workstations,server, network, devices and more.Every network has different systems and environments that generate various log formats, such as event logs, syslogs, and other application logs. Log collectors need to be flexible enough to accommodate all network devices and applications.

   - **Agent-based log collection:**Requires the deployment of an agent on the devices that generate logs. The agent not only collects and filters the logs, but it also parses and converts them into other formats before forwarding them to the log collection server.
   - **Agent-less log collection:**The log data generated by the devices is automatically sent to a SIEM server securely, eliminating the need for an additional agent to collect the logs, which reduces the load on the devices.

   In SIEM solutions, agentless log collection is the predominant method used to collect logs. In dynamic cloud environments, agentless auditing is critical to reduce costs, unlock visibility, and to accelerate the speed of deployment.

3. **Aggregating Logs:** Log aggregation is the mechanism for capturing, normalizing, and consolidating logs from different sources to a centralized platform for correlating and analyzing the data. This aggregated data then acts as a single source of truth for different use cases.

4. **Parsing:** Implement parsing to convert raw logs into structured data, facilitating easier analysis and interpretation.

5. **Normalization:** Normalization involves mapping known data attributes into a standardized template for easy comparison. Non-conforming data is typically excluded for simplicity, with discarded logs recommended for storage in a separate repository. However, discarding original data can be impractical for legal purposes. Therefore, most systems retain raw event logs for a user-specified period before archival. Some SIEM platforms may include links to original events in normalized logs, allowing for efficient 'drill-down' access to additional device-collected information.

6. **Enrichment:** Enhance logs by adding supplemental information (e.g., geo-location, transaction numbers) to facilitate analysis and reporting.Beyond adding external data analysing outcomes, like identity matching and behavioral detection, can augment records, providing more comprehensive insights. This introduces new dimensions to data utilization, with vendors likely introducing innovative ways to derive additional value from the extensive data they collect.

3.4 Detection of ongoing or probable attacks

This can be achieved by having proactive threat hunting. These hunters check for Indicators of Attacks (IOA), Indicators of Compromise (IOC) , using Vulnerability scans and using other threat intelligence to join the dots and identify if the system has already being compromised or when the potential compromise can occur.These hunters can be humans or can be automated to be done by machines.

Another approach is to use the SPoG (Single Pane Of Glass) approach of an XDR. An XDR uses a combination of data from SIEMs, EDR, NDR and other threat intelligence to give you a correlating result. The analysis is usually done by AI or a User Behavior Analytics Tool (UBA) and alerts can be sent out to all relevant entities.The workings of the tools mentioned earlier are defined in the below subsections.

3.4.1 Security Information And Event Management

SIEM analyzes volumes of logs from an organization's applications, devices, servers, and users which are pre-processed as defined in previous sections, in real-time so security teams can detect and block attacks. SIEMs use predetermined rules to help security teams define threats and generate alerts. Some examples of predefined rules one can use in SIEM:

Multiple Failed Login Attempts: Generate an alert if there are more than a specified number of consecutive failed login attempts within a defined time frame.

Unusual User Account Activity: Trigger an alert if a user account exhibits unusual behavior, such as logging in from multiple geographically distant locations within a short period.

Abnormal Pattern of File Access: Generate an alert if there is an unusual pattern of file access, such as a user accessing a large number of sensitive files in a short time.

3.4.2 Endpoint Detection And Response

EDRs record the activities and events taking place on endpoints and all workloads, providing security teams with the visibility they need to uncover incidents that would otherwise remain invisible.Consists an agent on each endpoint which conducts signature less detection and sends the behaviour of the device back to a server,where the processing occurs and any suspicious behaviour is made note of.

3.4.3 Network Detection Response

Network detection and response (NDR) is designed to detect cyber threats on corporate networks using artificial intelligence (AI), machine learning (ML), and data analytics. These tools build models of normal behavior by continuously analyzing network north/south traffic that crosses the enterprise perimeter as well as east/west lateral traffic, and then use these models to identify anomalous or suspicious traffic patterns.

3.5 Conducting Regular Security Checks

Regular security audits are critical for having a strong cybersecurity posture. These safeguards guarantee that any vulnerabilities are discovered as soon as feasible.These inspections also aid in spotting any detection software failures and gaps in our Incident Response Plans.The following aspects highlight the importance of continuous security practices:

1. **Code Review:** Conduct systematic reviews of source code to identify and rectify vulnerabilities. Proactive identification of security flaws before deployment minimizes the risk of exploitation. Regular code reviews foster a culture of secure coding practices among developers.

2. **Log Auditing:** Regularly audit system and application logs to detect unusual patterns or potential security incidents. Ensures adherence to security policies, regulatory requirements, and industry standards. Log audits contribute to forensic investigations and incident response effectiveness.

3. **Detection Software Evaluation:** Regularly assess and validate the proper functioning of detection software, such as SIEM (Security Information and Event Management). Ensure that the software effectively detects and responds to security incidents, and update configurations as needed.

4. **Incident Response Testing:** Regularly test the incident response plan through simulated cyberattack scenarios. Helps identify gaps in the response process, allowing for refinement and improvement. Ensures that the team is well-prepared to handle real-time incidents effectively.

5. **Security Training for Employees:** Regularly provide security awareness training to employees. Conduct simulated phishing exercises to educate and test employees' ability to recognize phishing attempts. Encourages a security-conscious culture, reducing the likelihood of human-related security incidents.

6. **Policy Review and Update:** Regularly review and update security policies to address emerging threats. Ensure security policies align with evolving compliance requirements and industry best practices. Communicate policy updates to all stakeholders to maintain awareness.

7. **Vulnerability Management:** Conduct regular vulnerability assessments and penetration testing. Promptly address and patch identified vulnerabilities to reduce the attack surface. Proactively manage risks associated with potential vulnerabilities in systems and applications.

8. **Access Controls and Privilege Reviews:** Regularly review and update user access controls and privileges. Ensure users have the minimum level of access necessary for their roles. Regular privilege reviews help detect unauthorized access and potential insider threats.

9. **Security Technology Updates:** Ensure that security technologies, including firewalls, antivirus, and intrusion detection systems, are regularly updated. Periodically review and optimize security technology configurations for effectiveness. Test the integration of various security technologies to ensure seamless operation.

In summary, we recommend prioritizing proactive security through the integration of robust logging, log analysis, event management, UBA, and other key elements.We also strongly advocate for proactive threat hunting using indicators to reduce Mean Time to Identify and Contain, minimizing potential losses for the firm.Continuous checks are crucial for detecting abnormal behaviors or security check malfunctions, providing insights for timely policy updates in line with changing times.

4 References

1. What is SIEM Integration? - Precisely
https://www.precisely.com/blog/big-data/what-is-siem-integration

2. Securing your applications with IDS/IPS - DevCentral
https://community.f5.com/t5/technical-articles/securing-your-applications-with-f5-big-ip-ids-ips/ta-p/288033

3. Certified Information Security Manager (CISM) Cert Prep (2022):3 Information Security Program
https://www.linkedin.com/learning/certified-information-security-manager-cism-cert-prep-2022-3-information-security-program/information-security-program

4. Security Control: Posture and vulnerability management - Microsoft
https://learn.microsoft.com/en-us/security/benchmark/azure/mcsb-posture-vulnerability- management

5. OWASP Proactive Controls - OWASP
https://owasp.org/www-project-proactive-controls/

6. The 18 CIS Critical Security Controls - Cisecurity
https://www.cisecurity.org/controls/cis-controls-list

7. How SIEM works and Architecture? - Relative Security
https://www.youtube.com/watch?v=V5XN2hd3BHMy

8. Application Security 101 - Snyk
https://www.youtube.com/watch?v=Dp019cWu1cg

9. XDR (Extended Detection & Response) Explained - IBM Technology
https://www.youtube.com/watch?v=Nwaigd9H60A

10. Cybersecurity Threat Hunting - IBM Technology
https://www.youtube.com/watch?v=VNp35Uw_bSM

11. Network Segmentation Security Best Practices - CheckPoint
https://www.checkpoint.com/cyber-hub/network-security/what-is-network-segmentation/network-segmentation-security-best-practices/