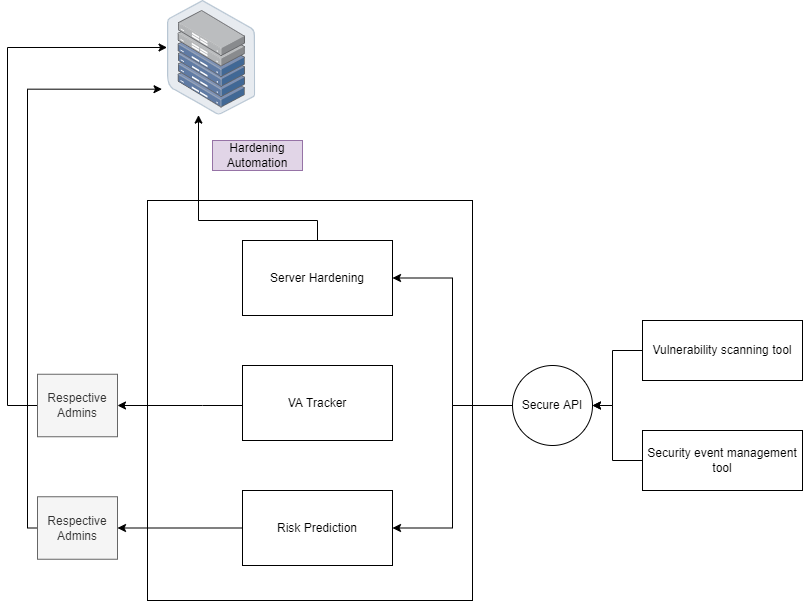
**Server Security Automation Solution**



**Sub component #1**

**Topic**

Server Vulnerability Scanning Automation

**Description**

Server vulnerability scanning automation is the process of using tools and techniques to automatically scan servers for vulnerabilities on a regular basis. This can help to identify and address vulnerabilities in a timely manner, improving the security of the servers and the overall network.

**Current Approaches**

Use a vulnerability scanning tool - There are many commercial and open-source tools available that can be used to scan servers for vulnerabilities. These tools typically operate by sending requests to the server and analyzing the responses to identify potential vulnerabilities.

Use a cloud-based vulnerability scanning service - Some cloud-based platforms offer vulnerability scanning as a service. These services can be used to automatically scan servers and identify vulnerabilities.

Use a security information and event management (SIEM) system - SIEM systems can be configured to automatically scan servers for vulnerabilities and generate alerts if any are found.

Use a configuration management tool - Configuration management tools, such as Ansible and Puppet, can be used to automate the process of deploying and configuring servers. These tools can be configured to run vulnerability scans as part of the deployment process.

**Drawbacks**

Cost: SIEM systems can be expensive to implement and maintain, particularly for organizations with large or complex networks.

Complexity: SIEM systems can be complex to set up and configure and may require specialized knowledge and expertise.

False positives - SIEM systems can generate a large number of alerts, many of which may be false positives that do not indicate a real security issue. This can lead to an excessive number of alerts that need to be reviewed and can make it more difficult to identify genuine security threats.

False negatives - SIEM systems can also sometimes fail to detect genuine security threats, either due to inadequate configuration or limitations in the system itself.

Maintenance and updates - SIEM systems require ongoing maintenance and updates to ensure that they continue to function correctly and remain effective at detecting security threats.

**Novelty**

New secure API to connect the system and vulnerability scanning tool. And implement a API monitoring tool to identify the anomalies of the API.

Use machine learning to improve alert accuracy - Some SIEM systems use machine learning algorithms to analyze security events and improve the accuracy of alerts. This can help to reduce the number of false positives and improve the system's overall effectiveness.

Automate patch management scripts – Using ML to identify what patches should be applied. These scripts automate the process of installing patches and updates to address known vulnerabilities.

**Sub component #2**

**Topic**

Server hardening automation according to CIS benchmark

**Description**

Server hardening is the process of securing a server by reducing its potential vulnerabilities and risks. Automating server hardening can help organizations deploy new servers more quickly and with a higher level of security. The Center for Internet Security (CIS) provides a set of scripts and tools for automating the implementation of their CIS Benchmarks. - CIS CAT Pro

**Current Approaches**

CIS CAT Pro

Configuration management tools (Ansible, puppet) - To automate the deployment and configuration of security measures on servers.

Security assessment tools (Nessus, Metasploit) - to scan servers for vulnerabilities

Containerization technologies (Docker) - to automate the deployment and management of secure, isolated environments for running applications.

**Drawbacks**

Complexity

High Expense

Resource consumption

***Ansible***

Complexity: Ansible can be complex to learn and use, especially for users who are new to automation and configuration management. It has a large number of modules and features that can be overwhelming for some users.

Dependencies: Ansible relies on a number of external dependencies, including Python and SSH, which can be a challenge to set up and maintain.

Performance: ansible can be slower than some other configuration management tools, especially for large infrastructures or when running complex playbooks.

Limited Windows support: ansible has limited support for managing Windows systems, which can be a limitation for organizations that rely heavily on Windows environments.

Poor error handling: ansible's error handling can be somewhat basic, which can make it difficult to troubleshoot problems when they occur.

***Nessus***

False positives: Nessus is prone to generating false positives, which can be time-consuming to investigate and can lead to a false sense of security.

Limited testing: Nessus is limited in the types of tests it can perform and may not be able to identify all types of vulnerabilities.

Complexity: Nessus can be complex to use and requires some technical knowledge to set up and configure.

Expense: Nessus is a commercial product and requires a subscription to use, which can be a cost issue for some organizations.

False negatives: Nessus may also miss some vulnerabilities, especially if they are not well-known or are not included in Nessus's database of known vulnerabilities.

***Docker***

Complexity: Docker can be complex to learn and use, especially for users who are new to containerization and container orchestration.

Resource consumption: Docker containers can consume a significant number of resources, especially when running multiple containers on the same host.

Security concerns: Docker containers can present security risks if they are not properly configured and managed. For example, containers can potentially access and modify the host system's resources, which can pose a security risk if not properly isolated.

Limited support for certain applications: Some applications may not be suitable for containerization due to their dependencies or resource requirements.

Integration with existing infrastructure: Docker may require significant changes to an organization's existing infrastructure and processes to fully integrate and take advantage of its benefits.

**Novelty**

Use AI to identify the nature of the server and the behaviour and customize the hardening.

Hardening automation and

**Sub component #3**

**Topic**

Server Risk Prediction automation

**Description**

Server risk calculation automation refers to the use of automated tools and processes to continuously assess the risks present on a server or servers. This can involve a range of activities, such as regularly scanning servers for vulnerabilities, analyzing security-related data to identify potential threats, and applying patches or other security measures to address identified risks.

**Current Approaches**

Use a machine learning model to analyze data about the server and its environment, such as its hardware and software configurations, the network it is connected to, and the security measures in place. The model could then use this data to predict the likelihood of a security incident occurring on the server, such as a caber-attack or data breach.

Another approach is to use a risk assessment framework, such as the National Institute of Standards and Technology (NIST) Cyber security Framework or the ISO/IEC 27001 standard, to assess the risks to the server. This can involve evaluating the server's vulnerabilities and the potential impact of a security incident, as well as the likelihood of such an incident occurring. The results of the assessment can be used to prioritize risk mitigation efforts and identify areas for improvement.

There are also tools and software platforms available that can automate the risk assessment process by collecting data about the server and its environment, analyzing it using a predefined risk assessment framework, and presenting the results in a user-friendly format. These tools can be helpful for organizations that need to conduct frequent risk assessments or have a large number of servers to manage.

**Drawbacks**

Limited flexibility: Risk frameworks are designed to be general-purpose tools that can be applied to a wide range of situations. However, this can also make them less effective in specific cases where the risks and vulnerabilities are unique or not well represented by the framework.

False positives: Automated systems can sometimes generate false positives, meaning that they identify potential risks that are not actually present. This can lead to unnecessary resources being spent on addressing these "risks," which can be frustrating and inefficient.

**Novelty**

Usable for different Frameworks.

Compliance recommendation system and followup system.

Machine learning algorithms: Machine learning algorithms can be used to analyze data and identify patterns or trends that may indicate risk. Improving the accuracy and efficiency of these algorithms can help improve the effectiveness of server risk prediction automation systems.

Integration with other systems: Server risk prediction automation systems should be integrated with other systems and tools, such as incident response and remediation systems, to enable a seamless and efficient response to identified risks.

User interface and usability: The user interface and usability of server risk prediction automation systems should be easy to use and understand so that non-technical users can effectively use the system to manage and mitigate risks.

**Sub component #4**

**Topic**

Server content inspection system

**Description**

Content inspection refers to the process of examining and analyzing the content of a message, file, or other type of data that is transmitted over a network or stored on a server. This can be done for a variety of purposes, such as to ensure compliance with legal or regulatory requirements, to protect against security threats, or to enforce acceptable use policies.

**Current Approaches**

Deep packet inspection (DPI): This involves examining the contents of network packets as they are transmitted across a network. DPI can be used to identify and block malicious traffic, enforce bandwidth limits, or monitor user activity.

Email inspection: This involves examining the contents of email messages to identify spam, malware, or other types of unwanted or potentially harmful content. Email inspection can also be used to enforce compliance with corporate policies or legal requirements.

File inspection: This involves examining the contents of files that are stored on a server or transmitted over a network. File inspection can be used to identify and block unauthorized or malicious files, such as viruses or malware.

**Drawbacks**

False positives: File inspection systems can sometimes flag benign files as malicious, which can result in legitimate content being blocked or quarantined. This can cause disruptions to business operations and can lead to frustration among users.

Limited coverage: File inspection systems can only detect known threats, so they may not be able to identify new or emerging threats. This can leave an organization vulnerable to attacks from unknown or zero-day threats.

**Novelty**

Auto identify the classification label of the file by its behaviour and content. And push the admins or users to classify it within a given time space.

**Sub component #5**

**Topic**

Admin Access Privilege management using multi-factor authentication including Face detection.

**Description**

Multi-factor authentication (MFA) is a security system that requires users to provide more than one method of authentication in order to access a system or service. This can include a combination of something the user knows (such as a password or personal identification number (PIN)), something the user has (such as a security token or key fob), or something the user is (such as a fingerprint or facial recognition).

**Current Approaches**

***Current approaches to MFA***

Security tokens: These are small physical devices that generate a one-time pass code that is required for authentication.

SMS authentication: This involves sending a one-time pass code via text message to the user's phone, which must be entered in order to authenticate.

Biometric authentication: This uses physical characteristics such as a fingerprint or facial recognition to authenticate the user.

Software authentication: This involves installing an authentication app on the user's device, which generates one-time pass codes or uses push notifications for authentication.

***Current approaches to face detection***

Viola-Jones algorithm: This is a machine learning-based approach that uses Haar cascades to detect faces in images. It works by dividing the image into small regions and evaluating the presence of certain features, such as edges and corners, in each region.

Deep learning-based approaches: These approaches use deep neural networks to detect faces in images. They typically involve training a model on a large data set of images and then using the trained model to detect faces in new images.

Feature-based methods: These approaches use algorithms to identify specific features of a face, such as the eyes, nose, and mouth. The features are used to locate and identify the face in the image.

Template matching: This approach involves creating a template of a face and then using it to search for the face in the image. The template is compared to different regions of the image, and the region with the highest match is identified as the face.

3D face detection: This approach uses 3D models of the face to detect and recognize faces in images and video. It can be more accurate than 2D approaches, but it requires specialized hardware and may be more computationally intensive.

**Drawbacks**

***Drawbacks of MFA***

False negatives: Biometric systems can sometimes fail to recognize the user's biometric data, resulting in a false negative. This can be frustrating for users and can lead to decreased confidence in the system.

False positives: In rare cases, biometric systems may mistakenly identify someone as the authorized user, resulting in a false positive. This can pose a security risk if an unauthorized user is granted access.

Privacy concerns: Some people may be concerned about the privacy implications of using biometric data for authentication. Biometric data is unique and cannot be easily changed, unlike a password, which can be reset if compromised.

Technical limitations: Biometric systems can be expensive to implement and maintain and may not be practical for all organizations. In addition, biometric systems can be affected by factors such as changes in the user's physical appearance, or the use of prosthetics or makeup.

Social acceptability: Some people may feel uncomfortable using biometric authentication due to cultural or personal reasons.

***Drawbacks of Face detection***

Accuracy: Face detection algorithms can sometimes have difficulty accurately detecting faces in images or video, particularly if the face is partially obscured or the lighting is poor.

Privacy concerns: Some people may be concerned about the privacy implications of using face detection, as it involves collecting and analyzing personal biometric data.

Bias: Face detection algorithms can sometimes exhibit bias, meaning that they are more accurate for certain groups of people (e.g., those with a certain skin tone or facial structure) and less accurate for others.

**Novelty**

***Improvements in the MFA approach***

Use a combination of authentication methods:

Using a combination of different types of authentication methods, such as something the user knows (e.g., a password), something the user has (e.g. a security token), and something the user is (e.g. a fingerprint), can increase the security of MFA.

Use time-based one-time pass codes: Time-based one-time pass codes (TOTPs) are pass codes that are generated by an authentication app and are only valid for a short period of time. This can help to prevent an attacker from using a stolen pass code to gain access to the system.

Use push notifications for authentication: Some MFA systems use push notifications to send a request for authentication to the user's device. The user can then confirm or deny the request, providing an additional layer of security.

***Improvements in Face detection***

Convolutional neural networks (CNNs): CNNs are a type of deep learning algorithm that is particularly effective at image classification and object detection tasks. They can be used to build a face detection model that is trained to recognize faces in images.

Viola-Jones algorithm: This is a machine learning-based approach that uses Haar cascades to detect faces in images. It works by dividing the image into small regions and evaluating the presence of certain features, such as edges and corners, in each region.

Deep learning-based approaches: Deep learning-based approaches to face detection can often be more accurate than traditional machine learning-based approaches, as they can learn to recognize complex patterns in the data.

3D face detection: 3D face detection algorithms can be more accurate than 2D approaches, as they can consider the three-dimensional structure of the face. However, this approach may be more computationally intensive and require specialized hardware.