THE USA JOURNALS
THE AMERICAN JOURNAL OF ENGINEERING AND TECHNOLOGY (ISSN – 2689-0984)
VOLUME 06 ISSUE10

PUBLISHED DATE: - 19-10-2024
DOI: - https://doi.org/10.37547/tajet/Volume06Issue10-09 **PAGE NO.: - 75-85**

RESEARCH ARTICLE **Open Access**

APPLICATION SECURITY AND LEAST PRIVILEGE ACCESS IN MODERN DEVOPS

Ajay Chava

Abstract
In the context of modern DevOps, application security and the implementation of the principle of least privilege (PoLP) are becoming critical elements aimed at minimizing risks and improving the sustainability of IT systems. This article analyzes approaches to integrating security measures at all stages of the software development lifecycle, starting from the early phases, which reduces the likelihood of vulnerabilities. Special attention is paid to the principle of least privilege, which restricts access by users and system components to only the necessary rights, thereby increasing security and preventing unauthorized access. Strategies for minimizing permissions, ensuring infrastructure protection, and automating security checks in CI/CD pipelines are considered. The challenges associated with the implementation of these principles are also discussed, and ways to overcome them are proposed to improve the security and stability of software solutions.

Keywords DevOps, application security, principle of least privilege, PoLP, access control, CI/CD, DevSecOps, security automation, infrastructure as code.

INTRODUCTION

In the fast-paced environment of development and the continuous integration and delivery (CI/CD) cycle, the risks associated with vulnerabilities increase significantly. Consequently, ensuring application security has become a critical task that requires the implementation of effective protection strategies and methods. One of the most important principles in access management and IT infrastructure protection is the principle of least privilege (PoLP), which minimizes risks by granting users and systems only the permissions necessary to perform their tasks.

The relevance of this study is driven by the need to adapt security measures to the modern requirements of DevOps, where development and deployment processes are closely integrated.

The aim of this work is to examine existing approaches to application security within the DevOps framework, with a focus on the application of the principle of least privilege and the integration of security measures at all stages of development.

1. Implementing Security in the Early Stages of Development (Shift Left Security)

In the rapidly evolving world of technology and the increasing demand for innovative products and services, traditional methods of ensuring security and testing, typically viewed as final stages in the development lifecycle, are becoming outdated. In this context, the concept of "shifting left" emerges, representing a fundamental shift in the approach to software development. This new paradigm not only redefines the timing but also fundamentally changes our perception of security and testing throughout all stages of the development lifecycle.

In the classical software development lifecycle, security and testing were often pushed to the final phases, which frequently led to unforeseen difficulties. However, the current environment demands new practices, urging organizations to rethink their approaches and start integrating security measures and testing from the very first steps of project development [1].

Rather than addressing identified issues in the final stages of the process, it is more effective to detect and resolve these issues during the initial phase. "Shifting left" advocates for such a strategy, where security and testing become not merely control tools at the later stages but essential elements that accompany the project from the beginning.

This approach has enormous potential to transform the development process, and this material will explore its features, advantages, and impact on the software development industry in detail.

Specifically, the concept of Shift Left Security focuses on implementing security measures at all stages of development, starting from the earliest ones. In the past, security testing often occurred in the final stages, which frequently resulted in issues being discovered only after the main work on the software was completed. This approach was typically accompanied by delays or the release of a product without the necessary level of security [2].

The primary goal of this approach is to enable developers to identify and eliminate vulnerabilities and configuration errors in the early stages of development, thereby minimizing risks and preventing potential threats from reaching production environments.

The advantages of this approach are evident. First, addressing vulnerabilities before the software is deployed significantly reduces the costs associated with fixing errors and prevents vulnerabilities from entering public and production systems, which in turn lowers overall business risk. Second, integrating security into the early stages of development accelerates the time-to-market for the product, as security issues are resolved promptly, avoiding delays in the final stages of development [3].

Additionally, shifting security to the left contributes to the overall improvement of application security. Automating security and compliance checks in the early stages of development, using protective mechanisms, and equipping developers with the necessary tools for handling security helps create more secure code and minimize the risks of data breaches. This is particularly crucial for sectors where reputation and user trust play a key role, such as the financial and healthcare industries.

Regionally, in the first half of 2024, data sale and free distribution offers were most frequently observed from Asian countries, accounting for about a third of all listings (30%). This trend is also linked to increased cybercriminal activity in the region. According to a study of cyber threats in Asia from 2022–2023, nearly half of all successful attacks on organizations in the region resulted in data leaks.

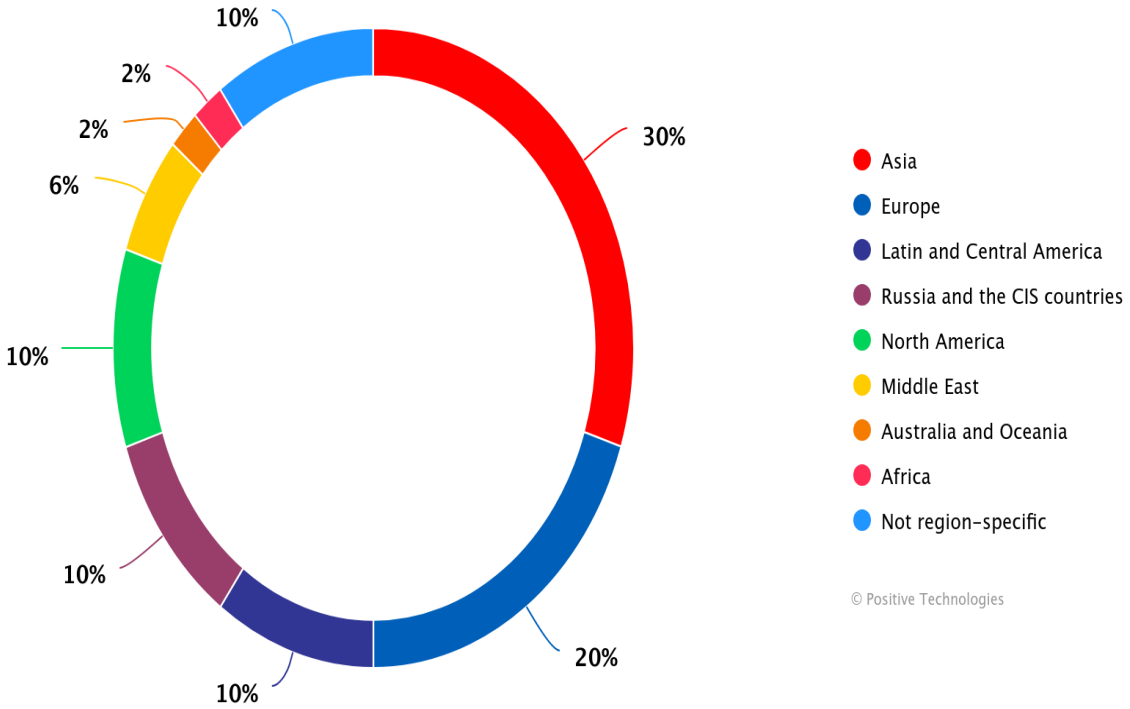

Fig.1. Data breach statistics by region [12].

From Figure 1, it is evident that in the first half of 2024, personal data generated the most interest in the darknet, accounting for over 83% of sale and distribution offers. It is important to note, however, that not all databases are put up for sale. In a significant number of cases, about 64%, the data is distributed for free, which is nearly double the share of commercial offers (33%). Russian companies were particularly vulnerable, with 88% of their data leaks involving free distribution. A high level of free databases was also characteristic of countries in Latin America, the United States, India, and Indonesia, where this share exceeds 70%. Meanwhile, databases from Chinese companies were more often sold, with their share reaching 60%. This phenomenon is explained by the fact that attackers are not always interested in selling the data. Often, they resort to ransom demands for non-disclosure, but not all victims agree to such terms. In the absence of demand for the data, hackers may release it publicly after some time [12].

The implementation of a left-shift security approach could help mitigate the risks of data breaches. The global GitLab DevSecOps report for 2023, which surveyed over 5,000 IT department heads, security leaders, and developers across various industries, confirmed that "DevSecOps teams increasingly see security as a shared responsibility." Meanwhile, approximately 83% of senior executives surveyed agreed that shifting security left is important for their organization, but 58% stated that this approach is burdensome for their developers. Experts warn that this experience, along with other challenges, may slow adoption and limit the value that the left-shift strategy can bring [13]. These challenges are reflected below in Figure 2.

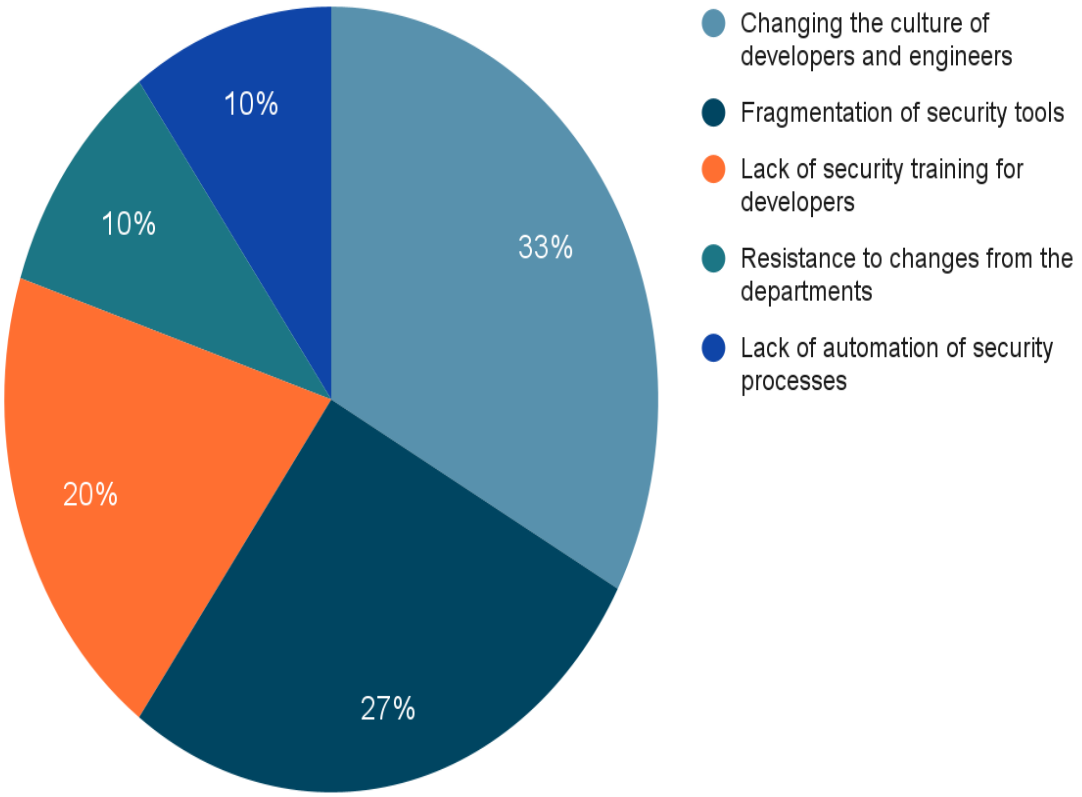

Fig.2. Difficulties in implementing the security shift approach to the left [13].

In such conditions, it is crucial to focus on developing a unified set of tools and methods that can automate security processes at all stages of development and reduce the burden on the teams responsible for security [4].

2. Principle of Least Privilege and Access Management

The Principle of Least Privilege (PoLP) is a fundamental approach in access management aimed at ensuring security and optimizing processes within IT systems. Essentially, this principle involves granting users only the levels of access and permissions necessary to perform their specific work tasks. This means that critical data and systems should be accessible only to those employees who genuinely need them to fulfill their job responsibilities [5]. Applying this approach requires considering multiple aspects and levels of access. However, achieving the isolation of shared resources often poses a significant challenge.

Access Management for Shared Resources: There are many cases where it is practical to use shared AWS resources. For example, using a single S3 bucket to store data for all users may be more efficient than creating a separate bucket for each user. This approach helps avoid exhausting the quota for buckets (currently limited to 100 buckets per account). Similarly, it might be convenient to access data from a single DynamoDB table rather than creating a new one for each user.

In such scenarios, developers are typically granted access to the entire bucket or database table. Access isolation is then implemented at the application level, ensuring that only the objects

related to a specific user are requested. However, this approach is not always optimal for adhering to the principle of least privilege, especially under stringent security requirements and regulatory compliance. There are more effective methods for fine-tuning access control. Let's explore one of these methods.

To implement granular access control, it is necessary to:

- Configure a Cognito Identity Pool to assign IAM roles to authenticated users.

- Set up the Identity Pool to add OpenID claims to user sessions using the "Attributes for Access Control" feature.

- Apply key tags in IAM policies to restrict access.

- Use identity credentials replacement with the Identity Pool to grant access to resources on behalf of the user.

Scenario Example

Let's consider an example to better understand the proposed approach. Imagine you are developing a multi-user application that allows users to store and retrieve data from an S3 bucket. The application also enables users to save configurations as key-value pairs in DynamoDB.

Amazon Cognito User Pool is used for authentication and authorization, linked with your API Gateway endpoints.

When a user sends a request to your API, they include an authorization header that the Cognito User Pool uses to verify the user's identity. Upon successful authentication, the API triggers a Lambda function, which allows the user to perform specific actions on the S3 bucket and DynamoDB table based on the payload provided. It's important to note that all users share the same bucket and table.

The challenge is to ensure that users can only access the data they have created or that is intended for them, rather than the entire resource.

Using amazon cognito identity pools. One of the key services that can help implement such access control is Amazon Cognito Identity Pools. This service allows you to link an identity pool with your identity provider (in this case, Amazon Cognito User Pool) and assign IAM roles to authenticated users.

Under this approach, a user is assigned an IAM role with policies that grant access to the necessary S3 and DynamoDB resources. After authenticating through Cognito, the user receives temporary AWS credentials, which allow them to perform actions on behalf of the Authenticated role. This way, access to protected resources is governed by strict rules and minimal privileges [6].

S3 Permissions Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
```

```
        "Resource": "arn:aws:s3:::BUCKET_NAME/*"
      }
  ]
}
```

DynamoDB Permissions Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:DeleteItem",
        "dynamodb:GetItem",
        "dynamodb:UpdateItem"
      ],
      "Resource":
"arn:aws:dynamodb:REGION:ACCOUNT_ID:table/TABLE_NAME"
    }
  ]
}
```

Now, let's move on to enhancing this solution to restrict access at the policy level. Assigning Tags to User Sessions Using Claims: With Amazon Cognito Identity Pool, you can create key-value attribute pairs using claims from your identity provider and tag the current user session with the appropriate tags. The example below demonstrates a default mapping where each user session is assigned a "username" tag, with "sub" representing the OpenId subject claim. This claim typically serves as a unique identifier that can be used to recognize individual users.

To properly implement this mapping, you need to modify the trust policy for the IAM role as shown below.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
        "Principal": {
          "Federated": "cognito-identity.amazonaws.com"
        },
        "Action": [
          "sts:AssumeRoleWithWebIdentity",
          "sts:TagSession"
        ],
        "Condition": {
          "StringEquals": {
            "cognito-identity.amazonaws.com:aud": "IDENTITY_POOL_ID"
          },
          "ForAnyValue:StringLike": {
            "cognito-identity.amazonaws.com:amr": "authenticated"
          }
        }
      }
  ]
}
```

This configuration enables the identification of individual user sessions created using temporary credentials and maintained by the authentication provider. It allows tracking each user individually, even when they use the same authorized role. This approach is particularly important for the subsequent steps, as we will need to modify existing permissions to restrict access only to objects and data associated with a specific user.

In DynamoDB, the concept of a partition key is used to uniquely identify records in a table. You can also create a composite primary key by combining the partition key with a sort key. For instance, if you need to store user configurations for different applications in a DynamoDB table, you can use a composite key where the username serves as the partition key, and the application name serves as the sort key. The structure of such a table might look as follows.

After setting the partition key to correspond to the username, you can modify the DynamoDB policy for the authenticated IAM role by adding a condition. This condition ensures that services using this role can only access records whose partition key matches the username of the current session [7].

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
```

```
        "Effect": "Allow",
        "Action": [
          "dynamodb:PutItem",
          "dynamodb:DeleteItem",
          "dynamodb:GetItem",
          "dynamodb:UpdateItem"
        ],
        "Resource":
"arn:aws:dynamodb:REGION:ACCOUNT_ID:table/TABLE_NAME",
        "Condition": {
          "ForAllValues:StringEquals": {
            "dynamodb:LeadingKeys": "${aws:PrincipalTag/username}"
          }
        }
      }
  ]
}
```

3. Ensuring Infrastructure Security and Automating Checks in CI/CD Pipelines

To ensure security and efficiency in CI/CD pipelines, DevOps teams must implement a comprehensive set of measures for access control and data protection. First and foremost, strict authentication and authorization mechanisms should be established to control which entities and users can interact with specific processes and tools. The principle of least privilege must be strictly adhered to, ensuring that access is granted only to those who genuinely need specific resources. Data protection plays a crucial role through the use of tokens, access keys, and passwords, minimizing the risk of malicious changes to the pipeline.

It is also critically important to secure the source code in version control repositories, such as Git, which are an integral part of CI/CD pipelines. These repositories contain not only application source code but also important manifests and intellectual property. To prevent potential attacks on the system, access to these repositories should be protected by multi-factor authentication. Additionally, developers should be regularly reminded of best practices for using Git and the proper configuration of files, such as `.gitignore`, to prevent accidental mistakes.

Ensuring the consistency of configurations across all environments involved in the pipeline, whether for development, testing, or production, is vital for successfully identifying issues during the testing phase. This consistency allows for more accurate detection and correction of vulnerabilities present in each environment. Such configuration parity can be achieved through the use of containerization technologies and a declarative approach to infrastructure management.

Moreover, rollback capabilities should be provided for deployments. If issues are detected after an update, the ability to quickly revert to a previous stable version allows for the elimination of

vulnerabilities before they are fully resolved. The most effective approach is to retain artifacts of previous versions until the successful deployment of the new one is confirmed.

Continuous scanning and monitoring for vulnerabilities should be integrated into every stage of the Software Development Life Cycle (SDLC). This allows for the timely detection and elimination of potential security threats at all levels—from source code to environment configurations.

Finally, it is necessary to regularly remove temporary and unnecessary resources used in CI/CD pipelines, such as containers and virtual machines. If these resources are not promptly cleaned up, they can become entry points for attackers, creating additional security risks. Maintaining order in pipeline resource management will help reduce the likelihood of vulnerabilities in the infrastructure [8].

First and foremost, it is essential to identify the key security requirements for your application and conduct a thorough assessment of potential threats. This will help determine which types of security testing should be included in your pipeline to minimize risks.

The next step is to select the appropriate security testing tools. These tools should be chosen based on the specific characteristics of your application and its security requirements. Among the most popular solutions are tools for static application security testing (SAST), dynamic application security testing (DAST), and software composition analysis (SCA). These tools will help identify vulnerabilities at various stages of development.

After selecting the tools, they need to be integrated into the CI/CD pipeline. The best approach is to implement them as automated checks, which can be done using plugins, scripts, or API interfaces provided by the selected tools.

Next, you should set up the automatic execution of security tests within the pipeline, establishing threshold values that determine the success or failure of the check. For example, the system can be configured to block further deployment of the application if critical vulnerabilities are detected.

Continuous monitoring of security test results is a crucial part of the process. It is important to respond promptly to identified issues, either by addressing them or by making appropriate changes to the pipeline configuration to prevent similar vulnerabilities from reoccurring.

Finally, it is important to implement best practices for security in the application development and deployment process. This includes using secure coding standards, design patterns, and configuration management, which will reduce the likelihood of new vulnerabilities emerging during the design and development stages of the application [9].

The importance of CI/CD pipeline security for your business cannot be underestimated. DevSecOps, or "Development, Security, and Operations," is a strategic approach to building and managing platforms that focuses on security at all stages of the IT lifecycle. Implementing DevSecOps helps protect sensitive data and reduces financial losses associated with cyber incidents. For example, in 2020, the average cost of data breaches was $3.86 million, and the expected cost of combating cybercrime by the end of 2021 could reach $6 trillion. Notably, 90% of web applications remain vulnerable to various types of attacks, with 68% at risk of data breaches. In the United States alone, over a thousand data breaches were reported in 2020, affecting more than 155 million people.

Given these statistics, ensuring security becomes a top priority for teams working with DevOps and Agile methodologies. In this context, the DevSecOps CI/CD pipeline can be seen as a logical extension of DevOps, incorporating security

measures at every stage of development [10, 11].

CONCLUSION

In conclusion, integrating security principles at all stages of development and enforcing strict access control based on the principle of least privilege are key factors in creating secure and threat-resistant applications within the DevOps context. These measures not only reduce the risks of unauthorized access but also contribute to faster product releases while maintaining a high level of protection. Implementing such approaches requires continuous knowledge updates and adaptation to new challenges, but the results justify the effort, ensuring robust security and compliance with modern security requirements.

REFERENCES

1. Dawoud A. et al. Better Left Shift Security! Framework for Secure Software Development //2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). – IEEE, 2024. – pp. 642-649.
2. Gonzalez D., Perez P. P., Mirakhorli M. Barriers to shift-left security: The unique pain points of writing automated tests involving security controls //Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). – 2021. – pp. 1-12.
3. Oka D. K., Fujikura T., Kurachi R. Shift left: Fuzzing earlier in the automotive software development lifecycle using hil systems //Proc. 16th ESCAR Europe. – 2018. – pp. 1-13.
4. Vaddadi S. A. et al. Shift left testing paradigm process implementation for quality of software based on fuzzy //Soft Computing. – 2023. – pp. 1-13.
5. Sanders M. W., Yue C. Mining least privilege attribute based access control policies //Proceedings of the 35th Annual Computer Security Applications Conference. – 2019. – pp. 404-416.
6. Billoir E. et al. Implementing the principle of least privilege using linux capabilities: Challenges and perspectives //2023 7th cyber security in networking conference (CSNet). – IEEE, 2023. – pp. 130-136.
7. Practicing the Principle of Least Privilege. [Electronic resource] Access mode: https://dev.to/kreuzwerker-/practicing-the-principle-of-least-privilege-1h04 (accessed 08/23/2024).
8. Singh A., Aggarwal A. Securing Microservice CICD Pipelines in Cloud Deployments through Infrastructure as Code Implementation Approach and Best Practices //Journal of Science & Technology. - 2022. – vol. 3. – No. 3. – pp. 51-65.
9. Deepak R. D. S., Swarnalatha P. Continuous Integration-Continuous Security-Continuous Deployment Pipeline Automation for Application Software (CI-CS-CD) //International Journal of Computer Science and Software Engineering. – 2019. – vol. 8. – No. 10. – pp. 247-253.
10. Mangla M. Securing CI/CD Pipeline: Automating the detection of misconfigurations and integrating security tools : dis. – Dublin, National College of Ireland, 2023.
11. Jammeh B. DevSecOps: Security Expertise a Key to Automated Testing in CI/CD Pipeline //Bournemouth University. – 2020.
12. Data leaks: current threats to companies in the first half of 2024. [Electronic resource] Access mode: https://www.ptsecurity.com/ww-en/analytics/data-leaks-current-threats-for-companies-in-H1-2024 / (accessed 08/23/2024).
13. Security from the beginning: the main

challenges in implementing left-wing approaches to cybersecurity. [Electronic resource] Access mode: https://www.csoonline.com/article/997815/secure-from-the-get-go-top-challenges-in-implementing-shift-left-cybersecurity-approaches.html (accessed 08/23/2024).