# Peer Review of the Presentation: Dynamic Code Analysis

## Introduction

This is a peer review of the presentation titled *Dynamic Code Analysis*. It highlights the main ideas discussion topics and provides a critique of the slides and forum engagement.

## Main Ideas from the Presentation

The presentation focuses on dynamic security testing and its role in application security. It defines Dynamic Code Analysis also known as Dynamic Application Security Testing (DAST). This method tests a running application to find security vulnerabilities unlike Static Application Security Testing (SAST) which reviews code without running it. The slides show the importance of DAST in real-world environments simulating attacks which find runtime vulnerabilities.

Key points include:

1. **Dynamic vs. Static Analysis**: Explaining the difference between analyzing code while it runs (DAST) versus analyzing code without running it (SAST).
2. **Benefits of DAST**: Showing how DAST provides better visibility finds runtime vulnerabilities and uncovers configuration and dependency issues.
3. **DAST Tools and Automation**: Listing tools like Valgrind and Invicti Security support automated scanning in CI/CD pipelines. It also explains how DAST integrates with DevSecOps for continuous security checks.
4. **Debunking DAST Myths**: Clearing up misunderstandings such as DAST being suitable only for simple websites when it can handle complex applications.
5. **Best Practices and Challenges**: Offering tips for using DAST tools effectively and addressing common challenges like the need for manual analysis and infrastructure.

The presentation ends with a Q&A section whcih encourages further exploration of DAST's role the vulnerabilities it addresses and how it complements SAST.

## Discussion Forum Topics

In the discussion forum the presenter covered topics to deepen the audience's understanding. Key topics discussed include:

- The differences between DAST and SAST and how they are used in the Software Development Lifecycle (SDLC).
- Challenges of integrating DAST into existing workflows including managing infrastructure and configuring tools.
- Advantages of black-box testing where testers examine the application without knowing the code simulating real external threats.
- How DAST can be automated in CI/CD pipelines for ongoing security assessments.

# Critique of the Presentation Slides

## Improvement Over Previous Versions (30 Points)

The slides are comprehensive covering both basic and advanced topics related to dynamic code analysis. They show improvement by including DevSecOps integration and automation.

## Flow and Clarity (10/10)

The presentation flows logically from basic concepts to advanced applications and tools. The information is clear and easy to understand.

## Coverage of Topic (10/10)

The slides thoroughly cover the essentials of DAST including methods tool integration and real-world applications.

# Critique of the Discussion Forum Activity (10 Points)

## Depth of Questions (10/10)

The questions in the forum are relevant and address the main challenges and practical aspects of DAST. They encourage meaningful discussions about security practices and testing methodologies.

## Coverage of Key Issues (10/10)

The questions cover all major issues surrounding DAST including automation's role and its limitations allowing participants to consider DAST's scalability and efficiency.

# Critique of the Discussion Forum Interaction (10 Points)

## Engagement and Further Discussion (10/10)

The presenter is engaged in the forum as shown by the detailed Q&A section. Effective responses to these questions foster further discussion and address common issues in DAST implementation.

# Conclusion

In conclusion the presentation on Dynamic Code Analysis is excellent covering both foundational knowledge and practical applications relevant to today's security landscape. The use of visual aids and interactive forum questions enhances understanding.