스패로우

전주기적 통합 애플리케이션 보안 테스팅 기술

화이트 페이퍼

주식회사 스패로우

SPARROW

CONTENTS

1 통합 기술 개요

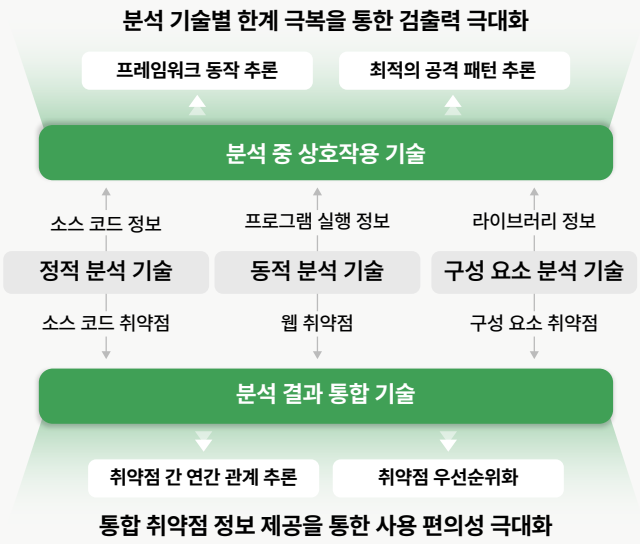
당사의 솔루션에는 복잡하고 다양한 기술이 적용됩니다. 그 중에서 핵심 기술의 종류는 크게 분석 기술과 통합 기술이라는 두 가지로 구분할 수 있습니다. 분석 기술은 솔루션의 코어라고 볼 수 있는 분석 엔진의 분석 수행 방법에 관련된 기술입니다. 여기에 포함된 기술의 종류는 정적 분석, 동적 분석, 구성 요소 분석으로 나누어집니다. 통합 기술은 이러한 분석의 결과를 활용하는 방법과 관련된 기술입니다. 여기에는 개별 분석 기술의 결과를 종합적으로 판단해서 결과를 표시하는 결과 통합 기술과 사용자의 소프트웨어 통합 및 배포(CI/CD) 파이프라인에 당사의 솔루션을 연결하여 활용하는 DevSecOps 기술이 포함됩니다.

통합 기술은 전주기적 통합 애플리케이션 보안 테스팅을 구현하기 위해 필수적인 기술입니다. 이 문서에서는 결과 통합 기술 및 DevSecOps 기술의 의미와 의의, 특징 및 세부 기술에 대한 설명을 간략히 서술하도록 합니다.

2 결과 통합 기술

당사의 정적 분석, 동적 분석, 구성 요소 분석과 같은 핵심 기술은 개발 단계에서 소스 코드 형태의 개발 중인 소프트웨어, 개발 완료 후 실제로 운영 가능한 형태의 애플리케이션, 다른 소프트웨어를 구성 요소로 포함하여 배포하려는 형태의 소프트웨어 패키지를 분석할 수 있습니다. 즉, 핵심 분석 기술은 애플리케이션의 형태에 따른 특징을 고려하여 최적화된 분석을 수행하게 됩니다. 정적 분석은 소스 코드만을 정보로 활용하여 소프트웨어의 실제 동작을 예측하는 데 최적화되어 있으며, 동적 분석은 실제 실행 중인 소프트웨어의 동작을 확인하고 분석하는 데 최적화되어 있습니다. 구성 요소 분석 기술은 소프트웨어에 포함된 다른 소프트웨어를 식별하는 데 최적화되어 있습니다. 이렇게 각각의 기술을 활용하는 것만으로도 소프트웨어 개발/운영 단계에서 다양한 유형의 애플리케이션에 존재하는 보안 취약점을 진단할 수 있기 때문에 전주기적 통합 애플리케이션 보안 테스팅에 가깝다고 할 수 있습니다.

당사는 여기에 만족하지 않고 테스팅 기술의 발전과 함께 각 기술에서 도출되는 서로 다른 유형의 보안 취약점 정보를 효과적으로 통합하여 분석하려고 노력하고 있습니다. 애플리케이션 보안 테스팅 분야에서도 이 문제를 중요한 과제로 여기고 있습니다. 이를 통해 개별 테스팅 방식이 지닌 고유한 특성과 한계를 상호 보완하여 더욱 정확하고 포괄적인 보안 취약점 진단을 실현할 수 있기 때문입니다. 결과 통합 기술은 정적 분석, 동적 분석, 구성 요소 분석이라는 기술을 모두 활용할 수 있다는 당사의 독보적인 기술적 위치에 기반하고 있습니다. 즉, 정적 분석, 동적 분석, 구성 요소 분석에 사용된 모든 세부 기술을 활용하여 애플리케이션을 다각도로 평가함으로써 사용자에게 개별 분석 기술의 특성을 살린 정보를 제공할 수 있습니다.



당사는 결과 통합 기술을 통해 사용자에게 종합적인 취약점 정보를 제공할 수 있습니다. 통합 정보를 통해 각각의 분석 기술이 지닌 장점을 효과적으로 결합하여 보안 취약점 검출 능력을 극대화할 수 있습니다. 이러한 통합적 접근 방식을 통해 단일 분석 기술로는 발견하기 어려운 복합적인 보안 취약점까지 식별할 수 있으며, 각 분석 결과의 연관성을 분석하여 오탐을 줄이고 정확도를 높일 수 있기 때문입니다. 특히 취약점이 발생한 정확한 위치와 영향을 받는 범위를 명확히 제시하고, 보안 취약점에 대해 발생 원인과 과정을 상세히 설명하여 개발자가 취약점의 본질을 이해할 수 있게 됩니다.

당사의 기술은 해당 취약점을 악용할 수 있는 잠재적 시나리오를 파악하고 있기 때문에 위험성을 정확하게 평가할 수 있습니다. 이렇게 평가된 정보를 일관된 형태로 제공하는 방법으로 사용자가 효율적으로 취약점을 관리하고 조치할 수 있도록 지원합니다. 취약점을 빠르게 조치할 수 있도록 업계 표준 보안 가이드라인과 모범 사례를 기반으로 한 구체적인 수정 방안을 제공하기도 합니다. 상세한 취약점 정보를 제공함으로써 보안 이슈를 빠르게 해결하고, 안전한 소프트웨어를 개발하도록 기여합니다.

2.1 상호작용 분석 기술

예를 들어, 웹 애플리케이션을 테스트하기 위해 정적 분석과 동적 분석을 모두 활용했다고 가정합시다. 연결되어 있는 특정 보안 취약점의 경우, 동적 분석의 결과는 웹 애플리케이션에서 보안 문제를 유발하는 입력 값을 보여주고, 정적 분석의 결과는 웹 애플리케이션을 구성하는 소스 코드에서 보안 문제를 유발시킬 수 있는 코드의 위치를 제공하게 됩니다. 그렇기 때문에 동적 분석의 결과를 통해 URL 및 파라미터 정보 등 해당 취약점을 실제로 확인할 수 있는 테스트 케이스를 생성할 수 있습니다. 그러나 테스트 케이스에 의해서 촉발된 보안 취약점을 수정하려는 경우 동적 분석의 결과로는 수정해야 하는 소스 코드의 위치를 파악하기 어렵습니다. 반면, 정적 분석의 결과, 즉, 취약점을 유발하는 소스 코드의 위치만으로는 취약점을 발생시키기 위해 실행할 수 있는 테스트 케이스를 확인하기가 어렵습니다.

상호작용 기술은 먼저 정적 분석을 통해 검출된 취약점과 동적 분석을 통해 검출된 취약점에 연관 관계가 있는지 확인합니다. 이를 위해 URL과 소스 코드 사이의 연관 정보, 정적 분석 결과의 소스 코드 정보, 동적 분석 결과의 URL 정보 등을 종합적으로 검토합니다. 두 개의 취약점이 연결되어 있다고 판단한 경우 개별 분석 기술로 검출된 결과를 통합하기 위해 정적 분석의 소스 코드와 동적 분석의 URL 사이에 맵핑 정보를 생성합니다. 맵핑을 토대로 각 분석 기술 간의 연관된 분석 정보를 추론함으로써 검출 결과를 통합합니다. 이러한 결과 통합 과정에서 개별 분석 기술의 기술적 특징 혹은 검출된 보안 취약점의 특징을 세부적으로 고려해야 합니다.

당사의 결과 통합 기술은 데이터를 연결하여 통합된 결과를 제공할 뿐만 아니라 통합된 검출 결과를 통해 해당 취약점의 결과를 먼저 검토하고 조치하도록 우선순위를 부여하려는 목적으로 개발되었습니다. 실제로 당사의 결과 통합 및 상호작용 기술은 보안 취약점 진단의 효율성과 정확성을 크게 향상시킵니다. 그렇기 때문에 사용자는 결과 통합 기술을 통해 식별한 취약점을 우선적으로 확인하게 됩니다. 그리고 해당 기술에서 제공하는 정보를 종합적으로 확인하여 신속하게 위험을 파악하고 대응 전략을 수립할 수 있습니다. 특히, 단일 테스팅 방식으로는 발견하기 어려운 복합적인 보안 취약점까지 식별할 수 있기 때문에 소프트웨어의 보안성을 한층 강화할 수 있다는 것이 강점입니다. 이를 통해 개발자와 보안 담당자는 더욱 안전하고 신뢰성 있는 소프트웨어를 효율적으로 개발하고 제공할 수 있습니다.

2.2 통합 위험지수 분석 기술

결과 통합 기술을 통해 정적 분석, 동적 분석 및 구성 요소 분석 기술로부터 검출된 취약점 정보에 대한 종합적 정보와 함께 해당 취약점이 얼마나 위험한지를 수치로 제공할 수 있습니다. 이 정보를 통합 위험지수라고 하며 취약점으로 인해 애플리케이션 전반에 발생할 수 있는 영향도를 표시합니다. 통합 위험지수는 각 분석 기술에 의해 검출된 취약점에 기본값을 부여하고 위험도별 가중치를 적용한 후 합산하여 측정하게 됩니다.

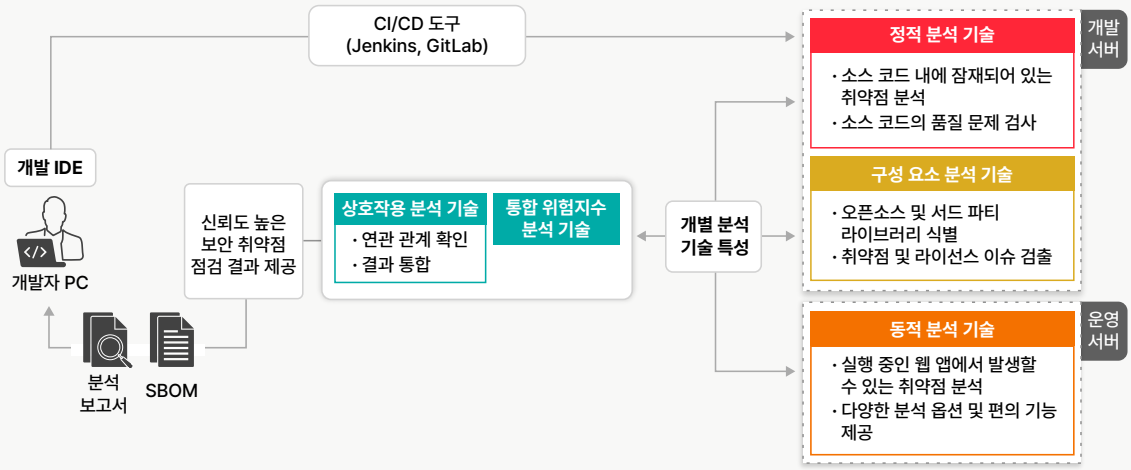
통합 위험지수 분석 기술을 통해 검출된 취약점 정보를 기반으로 애플리케이션의 전반적인 취약 수준을 파악할 수 있습니다. 이는 사용자로 하여금 애플리케이션 개발 생명 전주기에 걸쳐 다각도로 조치가 필요한 보안 취약점을 보다 빠르게 식별하고 대응할 수 있도록 합니다.

3 DevSecOps 기술

개발과 운영 프로세스를 통합하여 빠르게 서비스를 제공하려는 DevOps가 일반적인 소프트웨어 개발 및 배포 환경으로 인식되고 있습니다. DevSecOps는 DevOps가 지향할 다음 단계로서 개발 전과정에 보안을 통합하는 것을 목표로 단계별 보안 프로세스를 자동화할 것을 강조합니다.

소프트웨어 보안 테스트를 자동화하면 보안 프로세스가 CI/CD 파이프라인에 자연스럽게 통합어 코드가 변경될 때마다 보안 검증을 수행할 수 있습니다. 뿐만 아니라 자동화된 테스팅으로 인해 개발 초기 단계부터 보안 취약점을 발견하고 수정할 수 있습니다. 빠른 개발과 배포가 요구되는 DevSecOps에서 보안 테스팅 자동화는 개발 속도를 유지하면서도 효과적으로 소프트웨어 보안을 유지할 수 있게 하는 핵심입니다.

또한 보안 테스팅 자동화는 수동 테스트에 비해 인력과 시간을 크게 절약하고 반복적인 테스트를 효율적으로 수행할 수 있다는 장점이 있습니다. 특히 일관된 기준과 방법으로 보안 검증을 수행함으로써 테스트의 신뢰성을 높일 수 있습니다.



당사는 이러한 시장 요구에 부응하기 위해 보안 기술의 자동화에 최적화된 인터페이스를 제공합니다. 당사의 DevSecOps 기술은 주요 DevOps 도구와 분석 기술을 원활하게 통합하기 위한 다양한 연동 기능을 보유하고 있습니다. Jenkins, GitLab 등 널리 사용되는 CI/CD 플랫폼에 대한 연동 경험을 바탕으로 보안 테스팅 결과를 파이프라인에 자연스럽게 통합할 수 있는 기능을 제공합니다. 또한, Jira 등 이슈 트래킹 시스템과 연동함로써 발견된 취약점을 효율적으로 관리하고 추적할 수 있도록 지원합니다. 이러한 연동 기능과 경험이 당사의 핵심 기술에 적극적으로 반영되었기 때문에 사용자는 지나친 노력을 기울이지 않고도 효율적으로 DevSecOps 환경을 구축할 수 있게 됩니다.

4 맺음말

지금까지 결과 통합 기술 및 DevSecOps 기술의 의미와 의의, 특징 및 세부 동작에 대해 설명했습니다. 결과 통합 기술로 통합할 수 있는 데이터의 범위는 지속적인 연구 개발을 통해 확장되고 있으며 정확도 또한 향상되고 있습니다. 또한 DevSecOps 기술로 연동할 수 있는 플랫폼이나 연동에 적용된 사용자 인터페이스도 지속적인 연구 개발을 통해 개선되고 있습니다. 당사의 통합 기술 및 DevSecOps 기술을 사용하는 분석 솔루션을 직접 경험해보시고 당사에서 제공하는 기술 지원 및 컨설팅 서비스를 통해 소프트웨어 개발 및 운영에 활용해보시기 바랍니다.

추가적인 문의 사항은 스패로우 고객센터(https://cs.sparrow.im/ko/tickets)를 방문하세요.