스패로우
동적 분석 기술

화이트 페이퍼

주식회사 스패로우

SPARROW

CONTENTS

1 핵심 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.

동적 분석 기술은 당사의 전주기적 통합 애플리케이션 보안 테스팅 솔루션에서 제공하는 분석 기술인 정적 분석, 동적 분석, 구성 요소 분석 중 하나입니다. 세 가지 기술은 분석에 사용하는 목적물(분석 대상)의 차이를 기준으로 분류되며, 각 기술을 통해 분석을 수행하는 절차 혹은 방법(분석 방법), 분석 후 표시되는 내용(분석 결과)에도 고유한 특징이 있습니다. 이 문서에서는 동적 분석 기술의 분석 대상에 따른 분석 방법 및 분석 결과에 대한 설명을 간략히 서술하도록 합니다.

2 동적 분석 기술

웹 애플리케이션 및 API로 제공되는 서비스가 급속히 증가함에 따라 운영 중인 서비스에 대한 보안도 중요해지고 있습니다. 이러한 요구에 대응할 수 있는 테스트 방법이 동적 분석입니다.

동적 분석의 주요 기법으로는 행위 탐지, 샌드박스, 모의 해킹 등이 있습니다. 행위 탐지는 분석 대상인 프로그램을 실행할 때 발생하는 취약점 및 이상 현상을 탐지하는 방법입니다. 샌드박스는 가상 환경에서 분석 대상 프로그램을 직접 실행하고 그 때 발생하는 이상 현상을 분석합니다. 모의 해킹은 취약점을 진단하기 위해 실제 화이트 해커가 직접 프로그램을 공격하는 방식을 의미합니다.

당사의 동적 분석(DAST, Dynamic Application Security Testing) 기술은 모의 해킹을 자동화한 프로그램으로 실행 중인 애플리케이션에 임의의 요청을 전송하는 방식으로 보안 취약점을 탐지합니다. 앞서 설명한 정적 분석은 애플리케이션을 실행하여 분석하지 않기 때문에 잠재적으로 발생할 수 있는 문제를 유추합니다. 반면, 동적 분석 기술은 실제 공격자와 유사한 방법으로 테스트를 수행하기 때문에 런타임 환경에서 발생할 수 있는 취약점을 식별하기에 유용합니다.

동적 분석 기술로 웹 애플리케이션을 분석하는 방법을 예로 들면 다음과 같습니다. 먼저 취약점을 발생시킬 수 있는 HTTP 요청을 구성하고, 해당 요청을 분석 대상에 해당하는 서버에 송신합니다. 그리고 서버로부터 수신한 HTTP 응답을 분석합니다. 그 결과 특정 요청으로 인해 의도하지 않은 동작이 발생하거나 시스템에 보안상 문제가 발생하는 경우 해당 서버에 취약점이 존재한다는 것을 파악할 수 있습니다.

즉, 동적 분석은 분석 대상 프로그램과 분리된 환경에서 분석을 수행하여 분석 대상에 실제로 존재하는 취약점을 즉시 발견할 수 있습니다. 또한, 분석 대상의 소스 코드를 식별하지 않고도 대상 프로그램을 분석할 수 있다는 장점이 있습니다. 하지만, 정적 분석과 달리 취약점이 발생하는 코드의 위치를 지정하기 어렵고 분석 보고서를 해석할 때 최소한의 보안 지식을 요구하기 때문에 진입 장벽이 될 수 있습니다. 마지막으로 테스트에 많은 시간이 소요될 수 있다는 단점이 있습니다.

동적 분석의 한계를 보완하기 위해 당사는 여러 가지 동적 분석 기술을 제공하고 있습니다. 먼저 런타임 분석(Runtime Analysis)을 개발하고 동적 분석에 적용하여 트루스캔(TrueScan)이라는 분석 기술을 지원하고 있습니다. 트루스캔을 통해 웹 애플리케이션의 응답에서 발견한 취약점이 웹 앱 내부의 어떤 지점에서 발생했는지를 확인할 수 있습니다.

이행 분석 기술은 기존에 검출한 취약점 중 일부를 다시 분석하여 해당 취약점이 해결되었는지를 확인할 수 있는 부분 분석 방법입니다. 이러한 기술을 통해 웹 애플리케이션 전체를 분석하는데 많은 시간이 소요되는 동적 분석 테스트의 단점을 보완하고 빠르고 지속적으로 취약점을 관리할 수 있게 됩니다.

또한, 당사의 동적 분석 기술은 기본적으로 헤드리스 브라우저 기반 수집 기술을 갖추고 있습니다. 헤드리스 브라우저는 사용자 인터페이스(UI) 없이 백그라운드에서 실행되기 때문에 화면에 표시되지 않는 데이터 및 퍼포먼스까지 측정하기에 적합합니다. 또한, 내장된 JavaScript 엔진을 통해 렌더링된 결과에서 동적 요소에 대한 정보를 수집하거나 클릭 등 동작을 자동으로 수행하여 사용자 인터랙션으로 생성되는 정보도 가져올 수 있습니다.

다양한 웹 환경을 지원하기 위해 필요한 경우 분석 옵션을 설정하여 특정 웹 환경에 최적화된 분석을 수행하거나 사용자가 원하는 범위를 분석할 수 있도록 구현했습니다. 분석 옵션은 종류가 다양하고 개별 옵션의 기능이 다르기 때문에 이후에 자세히 설명하도록 하겠습니다.

분석 과정에서 특정 동작을 수행하거나 값을 입력하려는 경우 이벤트 클립보드를 사용할 수 있습니다. 이벤트 클립보드는 사용자가 수행한 동작을 기록하고 재생할 수 있는 프로그램입니다. 해당 프로그램을 실행하여 로그인과 같은 인증 과정을 파일로 저장하고 분석에 활용할 수 있습니다.

종합적인 관점에서 동적 분석 기술의 단점은 정적 분석 기술과 대비되는 양상을 보입니다. 따라서 동적 분석과 정적 분석을 모두 적용함으로써 단일 기술만으로 해결하기 어려운 문제들을 보다 효과적으로 해결할 수 있습니다. 당사는 두 기술을 모두 보유함으로써 분석 기술의 장점을 극대화하는 제품을 개발해왔습니다.

2.1 트루스캔 (TrueScan)

트루스캔은 분석 대상 웹 애플리케이션의 서버로 전송한 공격 요청으로 인해 내부에서 발생한 함수 호출 정보를 기록하고 해당 정보를 기반으로 취약점이 발생했는지를 식별하는 분석 방법입니다. 일반적인 동적 분석 기술은 웹 애플리케이션의 응답을 분석하여 취약점을 식별하기 때문에 응답에 드러나지 않은 결과를 확인하기 어렵습니다. 이에 비해 런타임 분석 기술을 활용함으로써 심도 있고 정확한 결과를 얻을 수 있습니다.

이 기술을 설명하기 위해서는 먼저 런타임 분석(Runtime Analysis)을 알아야 합니다. 런타임 분석이란, 실행 중인 프로그램으로부터 해당 프로그램의 동작에 관한 정보를 수집하고 분석하는 기술입니다. 이 기술을 통해 분석 대상 프로그램의 내부 정보(실행 옵션, 설정, 시스템 정보 등), 분석 대상 프로그램이 동작하는 도중 호출한 함수 정보(함수의 이름, 인수, 반환값, 콜스택 등), 분석 대상 프로그램이 사용하는 자원 정보(CPU, 메모리, 네트워크 트래픽 등), 분석 대상 프로그램의 성능 정보(특정 동작의 소요 시간 등)를 얻게 됩니다.

런타임 분석 기술을 구현할 때는 동적 계측(Dynamic Instrumentation) 기법을 활용합니다. 동적 계측은 실행 중인 프로그램을 중단하거나 해당 프로그램의 코드를 수정하는 등의 동작 없이 프로그램 내부를 모니터링하는 기술입니다. 이 기법을 통해 실행 중인 프로그램 내부로 계측에 사용할 코드를 삽입하고, 기존 프로그램에 존재하지 않은 기능을 추가할 수 있습니다. 또한, 특정 기능의 동작을 수행할 것인지를 제어하고, 내부에서 발생하는 함수 혹은 데이터의 발생 및 흐름을 수집하며, 프로그램의 성능을 측정하거나 기록할 수도 있게 됩니다. 이러한 동적 계측의 일부 기술을 활용한 런타임 분석 기술을 통해 얻은 정보를 동적 분석에서 수집한 정보와 비교하여 분석하는 기술이 트루스캔입니다.

트루스캔은 동적 분석 기술이 태생적으로 가지고 있는 한계를 개선할 수 있는 중요한 기술입니다. 앞서 언급한 것처럼 동적 분석은 수신한 응답에서 확인할 수 없는 취약점을 식별할 수 없기 때문입니다. 예를 들어, 동적 분석 도구로 삽입(Injeciton) 공격을 수행했을 때 해당 공격이 성공했는지 점검하는 방법은 굉장히 제한적입니다. 공격의 성공 여부가 웹 페이지까지 전달되지 않을 수도 있고, 결과가 도착하더라도 도구에서 감지하기 어려운 경우도 있습니다. 반면, 트루스캔 기능을 활용할 경우 함수 호출 정보를 통해 공격이 프로그램에 접근했는지를 알 수 있습니다. 따라서 보다 정확하게 공격이 성공했는지를 판단하게 됩니다.

당사의 동적 분석 기술에서는 동적 분석을 수행할 때 런타임 분석 기술을 추가적으로 적용하도록 설정할 수 있습니다. 그러면 동적 분석으로 파악한 데이터와 함께 런타임 분석 정보를 활용한 취약점을 검출할 수 있습니다. 당사는 런타임 분석에서 확인한 정보를 활용하여 취약점을 검출하는 검출 규칙을 트루스캔 기반 검출 규칙으로 분류하고 있습니다. 트루스캔 기반 검출 규칙은 애플리케이션 내부에서 측정되는 함수 호출 정보를 바탕으로 실제 공격이 이루어지는지를 판단합니다.

2.2 이행 분석 기능

동적 분석 기술의 경우 실제 동작하는 웹 애플리케이션에 다양한 패턴으로 이루어진 대량의 요청을 전송하는 모의 공격을 수행하기 때문에 한 번 분석을 완료하기까지 상당한 시간이 걸릴 수 있습니다. 따라서 식별된 취약점 이슈 중 일부를 해결했을 때 문제를 적절하게 수정했는지 확인하기 위해 웹 사이트 전체를 다시 분석하기에는 부담이 따릅니다.

당사의 이행 분석은 이런 상황에서 취약점을 간단하게 확인할 수 있도록 설계된 기능입니다. 사용자가 확인할 취약점을 선택하여 이행 분석을 실행하면, 동적 분석 엔진은 이전에 취약점을 검출했을 때 수행한 분석 과정을 그대로 서버에 다시 전송합니다. 그리고 서버로부터 확인된 동작을 분석하여 이전에 검출된 문제가 애플리케이션에 여전히 존재하는지를 확인합니다.

이행 분석은 동적 분석을 통해 애플리케이션 전체를 다시 분석하지 않고도 조치된 취약점이 해결되었는지를 빠르게 확인할 수 있습니다. 다만, 빠른 확인을 목적으로 사용하는 기술이기 때문에 특정 패턴의 특정한 결과만을 분석할 수 있다는점에 유의해야 합니다.

2.3 헤드리스 브라우저 및 이벤트 기반 수집 기술

일반적인 브라우저가 사용자 인터페이스(UI)를 통해 웹 페이지를 보여주는 것과 대조적으로 헤드리스 브라우저는 UI를 렌더링하지 않고 백그라운드에서 웹 페이지를 로드하고 상호작용할 수 있는 브라우저 환경을 제공하게 됩니다. 이를 통해 실제 사용자가 웹 애플리케이션과 상호작용하는 것과 유사한 방식으로 이벤트를 발생시키고 이벤트에 따라 동적으로 생성되는 콘텐츠와 자원을 실시간으로 수집할 수 있습니다.

좀 더 구체적으로 설명하면 HTML 소스 코드로 콘텐츠를 분석하는 전통적인 정적 분석 방식으로 감지하기 어려운 동적 콘텐츠를 자동으로 수집하고 분석할 수 있습니다. 예를 들어, AJAX 요청을 전송하고 수신하거나, JavaScript로 생성되는 동적 요소를 수집하고, 비동기적으로 데이터를 로드하는 동작을 수행하게 됩니다.

그리고 실제 사용자처럼 웹 페이지와 상호작용하며 필요한 데이터를 수집할 수 있습니다. 예를 들어, 사용자가 버튼을 클릭하거나 스크롤한 후에 로드되는 콘텐츠, 값을 입력한 다음에 표시되는 자동 완성이나 실시간 검색 결과, 마우스 호버 및 탭 전환 등으로 활성화되는 비활성된 상태의 요소 등 특정 시점에서만 나타나는 동적 데이터가 여기에 해당합니다.

당사의 동적 분석 기술은 이렇게 이벤트 기반의 수집 방법을 사용합니다. 이벤트 기반으로 데이터를 수집하므로 DOM 변화를 감지하여 언제 생성될지 모르는 데이터를 즉각적으로 포착하고 필요한 때에 적절하게 정보를 수집할 수 있습니다. 결과적으로 당사가 개발한 헤드리스 브라우저를 사용하는 이벤트 기반 수집 방법은 현대적인 웹 애플리케이션의 동적 특성을 정확하게 분석하고, 일반적인 HTTP 요청으로는 접근할 수 없는 데이터를 비롯한 다양한 요소의 보안성을 평가하는 데 필수적인 기술입니다.

2.4 분석 옵션

당사의 동적 분석에 포함된 분석 옵션에서는 동적 분석 기술이 분석할 분석의 범위 및 분석 수준을 설정할 수 있습니다. 예를 들어, 분석 대상 웹 애플리케이션의 URL에서 상위 경로를 수집하거나, 서브 도메인을 수집하도록 설정합니다. 특정 이벤트가 페이지에 포함된 경우 해당하는 요소 및 하위 요소의 이벤트를 수행하거나 제외하도록 설정할 수도 있습니다. 대규모 분석을 수행하는데 많은 시간이 걸리기 때문에 필요한 경우 URL을 수집하거나 분석하기 위해 소요되는 시간의 최대값을 설정할 수 있습니다. 분석 대상인 웹 애플리케이션이 모바일 화면을 지원하는 경우 분석에서 사용하는 내부 브라우저를 모바일 화면 형태로 설정하도록 할 수 있습니다. URL 수집 깊이 혹은 DOM 수집 깊이를 지정할 수도 있습니다.

이외에도 주석 분석 여부, 대기 시간, 페이지 로드 정도 등을 설정할 수 있는 약 30개의 분석 옵션이 지원됩니다.

2.5 이벤트 클립보드 및 이슈 재현 기능

이벤트 클립보드는 당사에서 개발한 프로그램으로 브라우저에서 수행한 이벤트를 기록 및 저장하고 재현할 수 있는 도구입니다. 이벤트 클립보드는 웹 브라우저에서 직접 사용할 수 있는 확장 프로그램 형태로 제공됩니다. 사용자가 이벤트 클립보드를 활성화하면 이벤트 클립보드가 웹 브라우저를 통해 동작 중인 웹 애플리케이션에 접근합니다. 그리고 사용자가 웹 애플리케이션에서 수행하는 이벤트를 보드에 기록하거나, 사전에 기록된 이벤트가 있는 경우 해당 웹 애플리케이션에서 이벤트를 순차적으로 수행합니다.

당사의 동적 분석으로 탐지된 웹 취약점 중 이벤트 기반으로 식별된 취약점은 해당 취약점이 발생하기까지 거쳐온 과정을 이벤트로 저장합니다. 따라서 이벤트 클립보드를 사용하면 웹 취약점과 관련하여 어떠한 형태의 공격이 이루어졌고 공격의 결과가 어떤 형태로 웹 브라우저에 표시되는지를 직접 확인할 수 있습니다. 발견된 취약점을 다시 보여준다는 의미로 이러한 기능을 이슈 재현이라고 부릅니다.

또한, 앞서 설명한 분석 옵션 중 이벤트 클립보드로 기록한 동작을 사용할 수 있는 설정이 있습니다. 여기에서 사용자가 수행한 이벤트를 파일로 저장하여 분석 범위에 입력함으로써 웹 취약점 분석 작업의 분석 범위를 확장할 수 있습니다. 분석 대상인 웹 애플리케이션에 특정한 사용자 ID와 비밀번호를 입력하여 로그인해야 하는 경우, 먼저 이벤트 클립보드에서 해당 동작(이벤트)을 파일로 저장합니다. 그런 다음, 분석 옵션의 로그인 기록 파일 혹은 로그아웃 기록 파일에 해당 파일을 업로드하면 동적 분석 엔진이 파일에 저장된 정보를 사용하여 분석을 진행하게 됩니다.

3 맺음말

동적 분석 기술의 경우 실제 동작하는 웹 애플리케이션에 다양한 패턴으로 이루어진 대량의 요청을 전송하는 모의 공격을 수행하기 때문에 한 번 분석을 완료하기까지 상당한 시간이 걸릴 수 있습니다. 따라서 식별된 취약점 이슈 중 일부를 해결했을 때 문제를 적절하게 수정했는지 확인하기 위해 웹 사이트 전체를 다시 분석하기에는 부담이 따릅니다.

당사의 이행 분석은 이런 상황에서 취약점을 간단하게 확인할 수 있도록 설계된 기능입니다. 사용자가 확인할 취약점을 선택하여 이행 분석을 실행하면, 동적 분석 엔진은 이전에 취약점을 검출했을 때 수행한 분석 과정을 그대로 서버에 다시 전송합니다. 그리고 서버로부터 확인된 동작을 분석하여 이전에 검출된 문제가 애플리케이션에 여전히 존재하는지를 확인합니다.

이행 분석은 동적 분석을 통해 애플리케이션 전체를 다시 분석하지 않고도 조치된 취약점이 해결되었는지를 빠르게 확인할 수 있습니다. 다만, 빠른 확인을 목적으로 사용하는 기술이기 때문에 특정 패턴의 특정한 결과만을 분석할 수 있다는점에 유의해야 합니다.