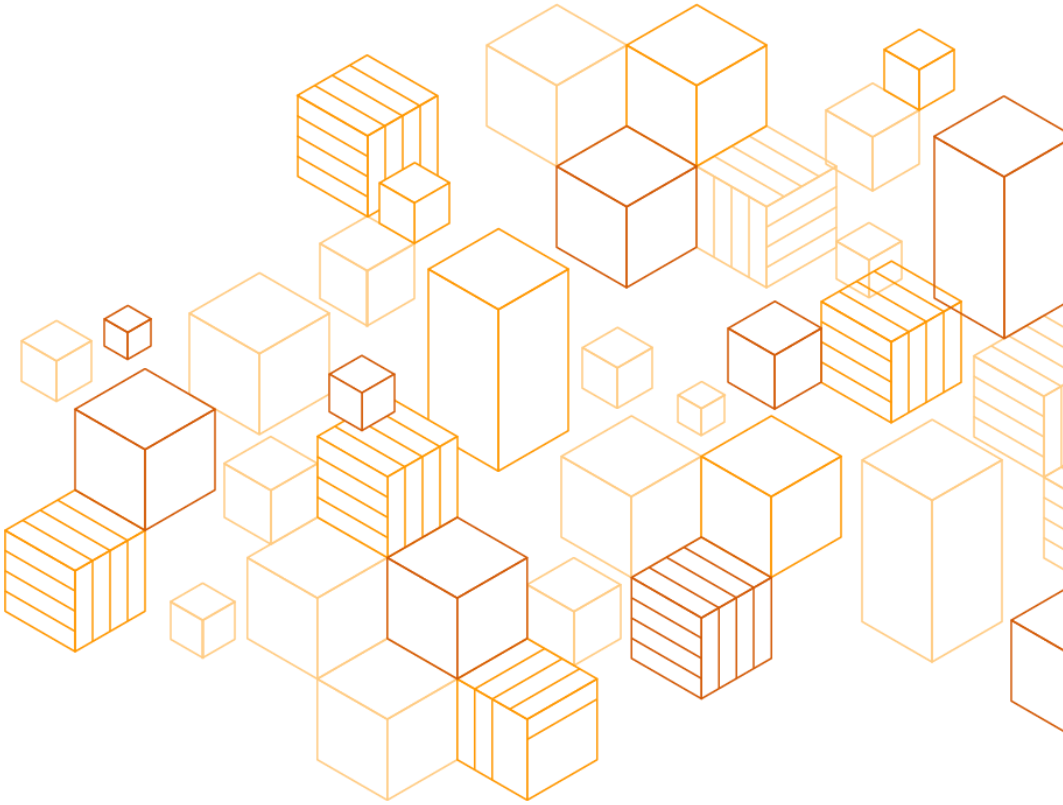
AWS WAF 구현 가이드라인

2020 년 10 월



본 문서 사용에 대한 공지

고객은 이 문서가 제공하는 정보에 대해 독립적으로 평가해야 할 책임이 있습니다. 본 문서는 정보 제공만을 목적으로 하며, 서술된 AWS 제품과 사례는 추후 예고 없이 변경될 수 있습니다. 또한 본 문서는 AWS 나 계열사, 공급업체, 인가업체의 보증이나 약정을 제공하지 않습니다. AWS 제품과 서비스는 명시적으로나 암묵적으로 보증이나 진술, 여하한 종류의 조건이 없이 "현 상태(As Is)"로 제공됩니다. 고객에 대한 AWS 의 책임사항와 법적 책무는 AWS 계약에 의해 관리되며, 이 문서는 AWS 와 고객간의 계약서에 속하지 않으며 계약서의 효력을 변경하지도 않습니다.

목차

이 문서에 관하여

AWS WAF 는 OSI 7 계층 중 애플리케이션 계층에서 다양한 공격으로부터 웹사이트와 웹 애플리케이션을 보호하는 웹 애플리케이션 방화벽(Web Application Firewall, WAF) 입니다. 이 백서는 기존 및 신규 웹 애플리케이션을 보호하기 위해 AWS WAF 를 구현할 때 필요한 최신 권장 사항을 설명합니다. 이 백서는 웹 애플리케이션의 보안 담당자에게 적합합니다.

개요

보안은 AWS 와 고객의 공동 책임이며, 각각의 책임 영역은 사용하는 AWS 서비스 등의 조건에 따라 달라집니다. 예를 들어, Amazon CloudFront, Amazon API Gateway, Application Load Balancer 등의 AWS 서비스를 사용하여 웹 애플리케이션을 구축한다면, OSI 모델의 7 계층에서 웹 애플리케이션을 보호하는 것은 고객의 책임입니다. AWS WAF 는 인터넷 등에서 유입되는 HTTP/HTTPS 트래픽을 모니터링하고 차단함으로써 고객의 웹 애플리케이션을 보호하는 도구입니다. 웹 애플리케이션 방화벽(Web Application Firewall, WAF)은 애플리케이션의 가용성에 영향을 미치거나 보안을 약화시키거나 리소스를 과다 사용하게 하는 등의 애플리케이션 계층의 일반적인 웹 공격으로부터 애플리케이션을 보호합니다. 이러한 공격으로는 OWASP 선정 10 대 보안 위협에 속하는 크로스 사이트 요청 위조, 크로스 사이트 스크립팅(XSS), 파일 인클루전, SQL 삽입 공격 등이 있습니다. 이 계층의 보안 도구와 다른 도구를 함께 사용함으로써 높은 보안성을 지닌 아키텍쳐를 구성할 수 있습니다.

AWS WAF 는 관리형 웹 애플리케이션 방화벽으로 Amazon VPC, AWS Shield Advanced 등 다양한 네트워크 및 보안 서비스와 결합하여 사용할 수 있습니다. AWS WAF 는 Amazon CloudFront, Amazon API Gateway, Application Load Balancer 에서 기본 사용되며 해당 서비스와 함께 배포됩니다. 이 서비스들은 TCP/TLS 연결을 종단하고 HTTP 요청을 받아서 처리한 후, AWS WAF 가 검사하고 필터링할 수 있도록 해당 요청을 전달합니다. 기존의 애플리케이션 기반 WAF 와는 다르게 AWS WAF 는 인프라를 설치하고 관리하거나, 용량 계획을 세울 필요가 없습니다. AWS WAF 는 관리형 규칙, 파트너 제공 규칙, 사용자 지정 규칙을 활용하여 애플리케이션 보호 기능을 구현하는 유연한 옵션을 제공합니다.

AWS WAF 는 애플리케이션의 수신 트래픽을 제어한다는 점을 기억하시기 바랍니다. 애플리케이션에서 외부로 나가는 송신 트래픽을 제어하려면 VPC 에 대한 보안 모범 사례를 참조하세요.

이 문서는 AWS WAF 를 이용하여 기존 및 신규 웹 애플리케이션을 보호하는 방안에 대해 다음과 같은 순서로 권고안을 제시합니다.

1. 위협에 대한 이해와 대처방안
2. AWS WAF 에 대한 요구사항

3. AWS WAF 구현하기
4. AWS WAF 를 프로덕션에 배포하기
5. 비용에 대한 검토

> **Note:** AWS WAF 는 WAFv2 와 WAF Classic 의 두 버전으로 제공됩니다. 최신 기능을 사용하려면 WAFv2 를 사용하세요. WAF Classic 은 더이상 업데이트 되지 않습니다. WAFv2 는 별도의 API 와 콘솔을 가지며 WAF Classic 에서는 제공되지 않는 기능을 지원합니다. 이 문서에서는 최신 WAFv2 를 기준으로 설명합니다.

위협에 대한 이해와 대처방안

AWS WAF 를 어떻게 구현할 것인지를 결정하기 전에 어떤 종류의 보안 위협이 웹 애플리케이션에 발생하는지, 그리고 AWS WAF 에서 제공하는 보호 옵션은 어떤 것이 있는지 이해할 필요가 있습니다. 다음과 같은 다양한 위협이 웹 애플리케이션에 발생하며, AWS WAF 를 통해 이 위협을 줄일 수 있습니다.

- DDoS 공격은 애플리케이션의 리소스를 고갈시켜 최종 사용자에게 서비스가 되지 않도록 합니다. 계층 7 에서의 DDoS 공격은 일반적으로 정상 형태의 HTTP 요청을 이용하여 애플리케이션 서버 및 다른 리소스의 고갈을 시도합니다.
- 웹 애플리케이션 공격은 애플리케이션 코드나 하위 소프트웨어의 취약점을 이용하여 웹 콘텐츠를 훔치거나 웹 서버의 제어권을 획득하거나 데이터베이스의 변조를 시도합니다. 이러한 공격은 의도적으로 변형된 HTTP 요청을 이용합니다.
- 인터넷 웹 사이트 트래픽의 많은 부분을 봇이 생성합니다. 검색엔진의 크롤러같은 좋은 봇들도 있지만, 악성 봇들은 애플리케이션의 취약점을 검색하거나, 콘텐츠를 스크래핑하거나, 백엔드 시스템에 악영향을 끼치거나 분석을 방해합니다.

AWS WAF 는 이러한 위협에 대해 보안성을 높입니다(*그림 1*).

그림 1 – 계층 7 에서의 위협

계층 7 에서의 DDoS 공격

HTTP Flood 공격에 대해서, AWS WAF 의 비율 제한 규칙을 사용하여 특정 IP 에서 애플리케이션에 대량의 요청을 발생시키는 클라이언트를 차단할 수 있습니다. 또한 AWS 관리형 규칙에 포함된 Amazon IP 평판 목록을 이용하거나 AWS Marketplace 에서 제공하는 AWS 파트너의 평판 목록을 이용하여 알려진 공격자 IP 를 차단할 수 있습니다. 보안성을 더욱 높이기 위해서는 AWS WAF Security Automation 솔루션을 이용하여 다음을 활성화하세요.

- 스캐너와 탐지기에 대한 보호- 애플리케이션 접근 로그를 읽어 과도한 오리진 에러를 유발하는 등의 비정상 트래픽을 찾아 차단합니다.
- 평판 목록에 의한 보호 - Spamhaus 의 DROP/EDROP 목록이나 TOR 출구 노드 목록, Proofpoint 의 Emerging Threats IP list 등 서드파티 목록을 이용하여 공격자 IP 를 차단합니다.

AWS WAF 에 추가로 OSI 계층 3, 4, 7 에서의 DDoS 공격에 대응하는 아키텍쳐를 위해서는 DDoS 복원력을 위한 AWS 모범사례를 참조하세요.

웹 애플리케이션 공격

AWS WAF 는 웹 애플리케이션 취약점 보호에 대해서 다음과 같은 옵션을 제공합니다.

AWS 관리형 규칙

AWS 관리형 규칙 그룹을 선택하거나 추가하여 다양한 위협에 대해서 애플리케이션을 보호할 수 있습니다. 관리형 규칙은 다음과 같습니다.

- OWASP 선정 10 대 보안 위협에 제시된 일반적인 위협과 보안 취약성 일부를 담당하는 기본 규칙 그룹
- OS 나 데이터베이스와 같은 애플리케이션 특성에 따라 추가적 보호를 제공하는 사용 사례에 특화된 규칙 그룹
- Amazon 위협 인텔리전스 팀이 제공하는 알려진 악성 IP 들로 구성된 IP 평판 목록

사용자 지정 규칙

AWS 관리형 규칙에 추가로 HTTP 의 헤더, 본문, 메서드, 쿼리 문자열, URI 등에 원치 않는 형태나 IP 주소 등을 차단하는 애플리케이션에 특화된 사용자 지정 규칙을 작성할 수 있습니다. 이러한 규칙은 AWS 관리형 규칙 그룹과 함께 사용되어 맞춤형 보호기능을 구현할 수 있습니다. AWS Management 콘솔에서 규칙 빌더를 이용하여 사용자 지정 규칙을 작성하거나, JSON 형태로 사용자 지정 규칙을 작성한 다음 AWS 명령줄 인터페이스(AWS CLI) 또는 AWS CloudFormation 같은 자동화 도구를 통해서 설정할 수 있습니다. 예를 들어, 사용자 지정 규칙을 이용하여 API URL 에 계획된 형태를 따르지 않는 호출을 차단할 수 있습니다. 사용자 지정 규칙에 쓸수 있는 논리적 규칙 문의 전체 목록을 보려면 규칙 문 목록을 참조하세요.

AWS Marketplace 규칙

보안 공급업체는 AWS WAF 에서 사용할 수 있도록 자체 제작한 규칙들을 AWS Marketplace 를 통해 공급합니다. 이 규칙들은 구독을 통해서 사용하실 수 있으며, AWS 관리형 규칙이나 사용자 지정 규칙과 함께 사용할 수 있습니다.

자동화된 완화 기법

봇의 행동에 기반한 방어는 AWS WAF 의 API 를 사용하여 구현할 수 있으며, 로그 분석이나 허니팟 URL 등을 통해 감지된 위협에 대해 자동으로 규칙을 업데이트하고 악성 IP 주소를 차단할 수 있습니다. 글로벌 서비스인 Amazon CloudFront 나 리전 서비스인 Application Load Balancer 혹은 Amazon API Gateway 모두 변경된 내용은 1 분내에 적용되므로, AWS WAF 웹 액세스 제어 목록(웹 ACL)의 기존 규칙을 갱신함으로써 위협에 빠르게 대처할 수 있습니다. AWS 는 다양한 보호 기법의 참고자료로 AWS WAF Security Automation 솔루션을 제공합니다.

악성 봇

악성 봇이 발생시키는 트래픽을 막기 위해서, 스캐너 형태의 봇에 대해서는 AWS 관리형 규칙에 포함된 IP 평판 목록 사용을 권장합니다. 또한 AWS WAF Security Automation 솔루션을 이용하여 허니팟을 구현하고 WAF 로그를 분석하여 행동기반의 봇 감지를 사용하실 수 있습니다. 크레덴셜 스터핑 등 애플리케이션 공격에 사용되는 탐지가 어려운 봇에 대해서는 봇 관리 솔루션을 아키텍쳐에 포함 시키기를 권장합니다. AWS Marketplace 에서 향상된 봇 완화 기능이 있는 서드파티 솔루션을 구할 수 있습니다. 몇몇 솔루션은 Lambda@Edge 를 이용한 Amazon CloudFront 와 통합을 지원하며 이를 통해 인라인 보호를 제공합니다.

요구사항

비즈니스 성공을 위한 AWS WAF 구현의 첫번째 단계로써 먼저 요구사항을 수집하고 정의해야 합니다. 이 섹션에서는 일반적인 WAF 요구사항들을 검토해봅니다.

보호

애플리케이션에 대해 어떤 종류의 위협이 존재하는지 확인되었다면, 성공적인 구현을 위한 기준 조건을 정의해야합니다. 성공적인 구현의 기준 조건은 서드파티나 자체 보안팀에서 행하는 침투 테스트를 통과하거나, 지정된 규정을 준수하거나, 일반적인 웹 취약점(OWASP 선정 10 대 보안 위협)에 대응할 수 있는 것입니다. 예를 들어, 애플리케이션이 제공하는 콘텐츠의 민감도에 따라 WAF 웹 액세스 제어 목록(웹 ACL)을 생성할 때 포지티브 보안이나 네거티브 보안 모델(API 호출을

허용할지 차단할지)을 선택할 수 있습니다. 만약 애플리케이션이 SQL 데이터베이스를 사용하지 않는다면, SQL 삽입 공격 감지 규칙을 추가하지 않음으로써 AWS WAF 웹 ACL 용량 단위(WCU)를 절약할 수 있습니다. 불필요한 규칙의 추가는 오탐을 증가시킬 수 있으므로 애플리케이션 요구사항에 특화된 WAF 규칙만 추가하기를 권장합니다. 오탐은 정상적인 요청이 WAF 에서 공격으로 오인되어 차단되는 경우를 말합니다.

이미 운영중인 애플리케이션의 경우는 애플리케이션의 사용 패턴을 이미 알고 있으므로 기존의 사건이나 관찰에서 확인된 악의적인 요청을 차단해야 할 것입니다. 따라서 이 경우에는 특정한 공격에 대한 보호 방법을 찾아야 합니다. 만약 이미 구현된 WAF 가 있다면, 기존 WAF 규칙에 의해 차단되는 요청 수에 따른 기준선이 있을 것입니다. 또한, 기존 규칙의 상세에 대해 알고 있으면 유사한 규칙을 AWS WAF 에 구현할 수 있습니다.

관리형 규칙과 사용자 지정 규칙

고객 조직의 자원과 보안 문화에 따라, 어떻게 AWS WAF 를 구현할지 결정해야 합니다. 고객은 작업이 필요없는 AWS 관리형 규칙 집합을 적용하거나, 직접 사용자 정의 규칙을 생성하거나 이 둘을 혼합해서 사용할 수 있습니다. 대다수의 애플리케이션에 대해서, AWS 관리 규칙의 기본적인 규칙 그룹과 Amazon IP 평판 목록으로 시작하여 애플리케이션의 프로파일에 맞는 규칙 그룹을 선택하시기를 권장합니다.

특정 워크로드에 대해서는 향상된 보호가 필요할 수 있습니다. 이런 경우 다른 규칙에 추가로 사용자 지정 규칙을 추가할 수 있습니다. 자체적인 규칙을 구현하고 유지하기 위해서는 고객의 보안 팀 또는 애플리케이션 팀이 WAF 규칙을 생성하고 관리하는 기술 역량을 지닐 수 있도록 해야 합니다. AWS Professional Services 나 AWS WAF 파트너는 이러한 워크로드를 지원하기 위해서 규칙을 생성하고 주기적인 리뷰를 수행하며 고객의 담당팀이 전문성을 갖추도록 교육을 제공할 수 있습니다.

거버넌스

WAF 구현을 조직내에서 어떻게 관리하고 모니터할지 정의하는 거버넌스 요구사항이 있을 수 있습니다. 어떠한 조직에서는 WAF 설정을 중앙화된 보안 팀에서 관리하며, 이 경우 보안 팀은 WAF

가 각 애플리케이션 팀에서 관리하는 자원에 제대로 설정되었는지 감사하고 설정을 강제할 수 있어야 합니다. 다른 조직에서는 WAF 설정과 배포를 애플리케이션 팀이 관리하고 보호 대상인 애플리케이션에 특화된 WAF 규칙을 배포할 수 있게 하기도 합니다.

AWS WAF 의 중앙화된 관리를 위해 AWS Firewall Manager 는 AWS Organization 내의 AWS 계정에 WAF 를 자동 배포하는 보안 정책 정의 기능을 제공합니다. AWS Firewall manager 는 자원들이 적합한 WAF 웹 ACL 에 연결되어있고 WAF 정책을 준수하는지 확인할 수 있는 가시성을 제공합니다. 이러한 경우에 대해서 다음의 거버넌스 예제를 참조하시기 바랍니다:

예제 1- AWS 관리형 방화벽 구현

이 예제에서는 애플리케이션 팀들이 자체적으로 WAF 설정을 운영하고 보안 팀이 감독하는 사례를 살펴봅니다.

- 보안 팀은 애플리케이션 팀이 따라야 할 모범 사례를 문서화해서 제공합니다.
- 보안 팀은 AWS Firewall Manager 의 WAF 정책을 이용해서, AWS 관리형 규칙의 기본적인 규칙 그룹을 바탕으로 하는 웹 ACL 을 각 팀의 AWS 계정에 배포합니다. 이때 비준수 리소스 자동 문제 해결을 선택하지 않습니다. 이 정책은 웹 ACL 의 복사본을 배포하지만 자동으로 애플리케이션 자원(CloudFront, Application Load Balancer, Amazon API Gateway)에 연결시키지는 않습니다. 이 방식은 애플리케이션 팀에게 WAF 를 통한 보호를 강제하지는 않지만, 보안팀에게 어떤 애플리케이션이 WAF 의 보호를 받고 있는지 가시성을 제공합니다.
- 애플리케이션 팀은 제공된 웹 ACL 을 그대로 사용하거나, 적용전에 수정해서 사용할 수 있습니다. 이러한 내용은 애플리케이션 팀의 보안 요구사항과 거버넌스 방침에 따라 결정됩니다.

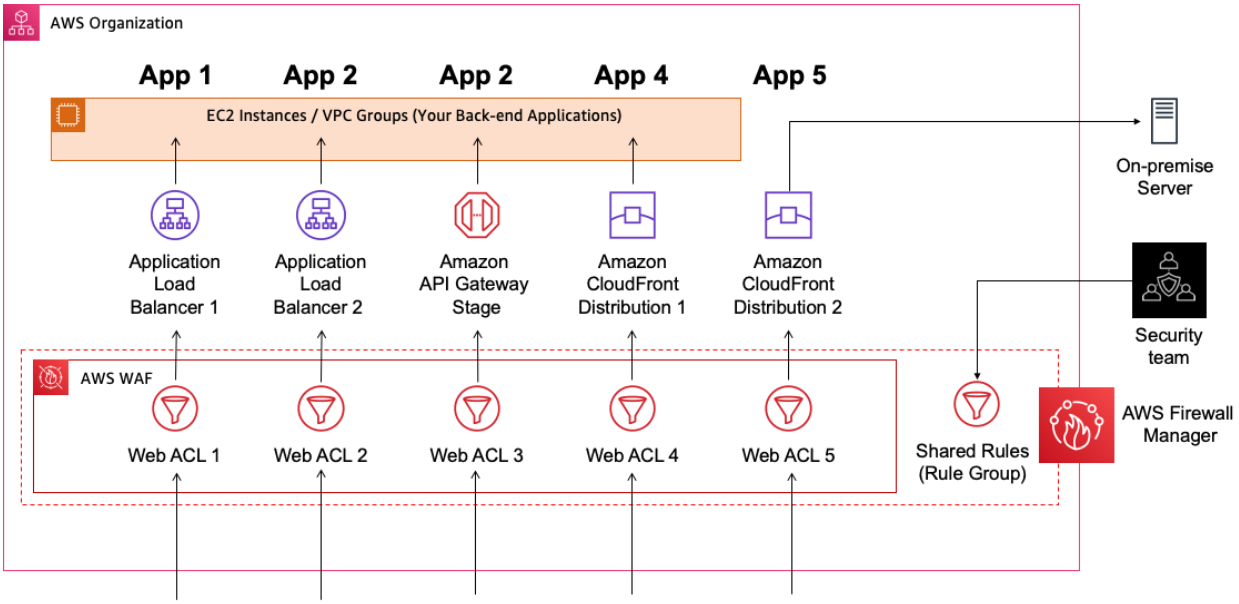


그림 2 - AWS Firewall manager 구현의 예제

예제 2- 2 개의 WAF 정책을 사용하는 AWS Firewall Manager 구현

이 예제에서는 중앙 보안 팀이 조직내의 애플리케이션별 WAF 규칙과 WAF 배포를 관리합니다.

- 중앙 보안 팀은 2 개의 자동 문제 해결이 활성화된 AWS Firewall Manager WAF 정책을 생성합니다.
- 첫번째 정책은 WordPress 애플리케이션으로 태그되어 있는 자원을 위해 WordPress 용 관리형 규칙을 사용합니다.(WordPress 는 예시용 애플리케이션으로 사용되었습니다.)
- 두번째 정책은 다른 HTTP(S) 애플리케이션을 위해 Amazon IP 평판 목록과 비율 기반 규칙을 사용합니다.
- 애플리케이션 팀은 WordPress 애플리케이션에 연관된 자원에 태그를 작성합니다.

- AWS Firewall Manager 는 각각의 AWS 계정내에 정책당 하나씩 2 개의 웹 ACL 을 만듭니다. AWS Firewall Manager 는 정책에서 설정된 대로 적합한 자원에 이 웹 ACL 을 자동으로 연결시킵니다. 만약 기존에 연결되어있던 웹 ACL 이 있으면 정책에 의한 웹 ACL 로 대체됩니다.
  Note: 이 백서의 출간시점에는 하나 이상의 정책이 한 자원에 적용이 될 경우 어떤 정책이 우선되도록 조정이 되지 않습니다. 그러므로 조직내에서 일관성을 유지하기 위해서 중복되는 정책을 피해 주시기 바랍니다.
- 보안 팀은 AWS Management 콘솔에 접속하여 AWS Firewall Manager 를 통해서 WAF 규정 준수 여부를 모니터 할 수 있습니다. AWS Firewall Manager 는 자원이 설정된 정책에 맞게 올바른 WAF 웹 ACL 에 연결되어있는지 확인 기능을 제공합니다. 또한 AWS Security Hub 과 AWS Firewall Manager 를 통합하여 WAF 규칙으로 보호받고 있지 않은 자원을 감지할 수 있습니다.

로그 기록

WAF 로그 기록은 규정 준수 및 감사를 위해 보안 팀에게 필요한 공통 요구 사항입니다. AWS WAF 는 Amazon Kinesis Data Firehose 를 통해서 준실시간 로그를 제공합니다. AWS WAF 에서 검사된 요청은 타임스탬프, 헤더 상세정보 등의 요청 정보와 일치하는 규칙의 작업정보를 포함하는 로그 기록을 남깁니다. 요청 본문은 현재 로그로 기록되지 않습니다. 로그는 디버그 용도로 활용하거나, 보안 정보 및 이벤트 관리(SIEM) 혹은 다른 로그 분석 도구와 통합하여 추가적인 포렌식에 활용될 수 있습니다. 기본적으로 웹 ACL 이 생성될 때는 로그 기록이 활성화 되어있지 않습니다. 자동화된 로그 활성화를 위해서는 AWS Config 을 이용하여 새로운 WAF 웹 ACL 이 생성될 때 마다 로그 설정을 하도록 할 수 있습니다.

그림 3 – 자동화된 로그 활성화

구현

시작점 고르기

요구사항을 확인했으면, 먼저 어떤 애플리케이션부터 시작할지 결정해야합니다. AWS WAF 를 처음 사용하신다면 가능한한 중요도가 낮은 애플리케이션부터 시작하기를 권유합니다. 이는 새로운 플랫폼에 익숙해질 기회를 제공하고 잘못된 설정으로 비즈니스에 영향을 미칠 위험도를 낮춰줍니다. 또한, 트래픽 패턴에 대해서 익히 알고 있는 애플리케이션 부터 시작하기를 권유합니다. 이는 WAF 배포시 영향에 대해서 빠르게 확인하고 적절히 조정을 할 수 있도록 해줍니다.

AWS WAF 통합 설계

애플리케이션의 요구사항에 따라 AWS WAF 를 어느 위치에 배포할 것인지 결정합니다. 이미 언급되었듯이, AWS WAF 는 Amazon CloudFront, Amazon API Gateway, Application Load Balancer 에 설정 가능합니다. 특별한 이유가 없다면 퍼블릭 대상의 웹 애플리케이션에 대해서는 최상의 보안성을 위해 AWS WAF 를 Amazon CloudFront 와 함께 배포하시기 바랍니다. 동적인 콘텐츠와 정적인 콘텐츠 모두에 CloudFront 를 사용하실 수 있습니다. 기본적으로 CloudFront 는 비 HTTP(S) 트래픽과 정상 형태가 아닌 HTTP 요청을 차단하며 네트워크 계층 3,4 에서의 공격을 1 초 이내에 완화하는 인라인 DDoS 보호 기능을 제공합니다. CloudFront 는 상태 유지 없는 SYN Flood 완화나, 대규모 볼륨 공격의 영향을 Amazon CloudFront 글로벌 엣지 네트워크에 분산 혹은 고립시킬 수

있는 자동화된 트래픽 관리 시스템(이는 Amazon Route53 과 같이 배포되었을 때 가장 효율적으로 동작합니다) 등의 향상된 DDoS 보호 기법을 사용합니다. 만약 애플리케이션이 AWS 외부에 호스팅 되어있다면, CloudFront 는 AWS 글로벌 네트워크를 이용하여 데이터 센터를 공격으로부터 보호합니다.

추가적인 요구사항을 가진 몇몇 애플리케이션에 대해서, 고객은 원본이나 로드밸런서에 사용되는 WAF 를 AWS WAF 와 함께 사용하는 계층화된 WAF 모델을 선택할 수 있습니다.

- 예를 들어 원본에서 응답된 결과를 검사하고 싶을 때에는, AWS WAF 를 CloudFront 와 함께 이용하여 들어오는 요청에 대해서 검사하고, 어플라이언스 기반의 WAF 를 통해 원본에 들어오는 요청과 나가는 응답을 검사할 수 있습니다. 몇몇 어플라이언스 기반의 WAF 는 적절한 권한이 있는 IAM 역할과 함께 설정한다면, 공격 트래픽을 검출했을 때 AWS WAF 용 규칙을 작성하고 AWS WAF 에 추가하는 기능을 가지고 있습니다.
- 또다른 예로는 여러 애플리케이션을 하나의 도메인으로 CloudFront 를 통해 제공하는 경우입니다. CloudFront 에 AWS WAF 를 적용하여 일반적인 IP 와 IP 를 통해 알아낸 지역 기반의 차단을 엣지에서 처리하고, 애플리케이션에 특화된 규칙을 각각의 Application Load Balancer 에 적용된 AWS WAF 에 배포할 수 있습니다.

스테이징 환경에서 검증하기

시작할 애플리케이션을 선택했으면 스테이징 환경의 설정을 권장합니다. 이러한 접근법으로 운영 트래픽에 좋지 않은 영향을 미치지 않으면서 AWS WAF 를 실험 해볼 수 있습니다.고객조직의 운영 방식에 따라 두 가지 방식으로 접근할 수 있습니다.

- AWS WAF 를 포함한 전체 애플리케이션 스택을 스테이징 환경으로 복제하기.
- 운영 환경에서 새로운 엔드포인트를 생성하기. 새로 만들어진 엔드포인트에 AWS WAF 를 배포하여 스테이징 환경을 만듭니다. 예를 들어, 새로운 CloudFront 배포본에 WAF 웹 ACL 을 연결하고 이미 존재하는 애플리케이션의 로드 밸런서를 원본으로 설정할 수 있습니다. **Note:** 만약 이미 CloudFront 를 사용하고 있다면 새로운 CloudFront 배포를 생성할 수 있지만, 기존 배포에서 사용하고 있는 동일한 도메인 이름을 사용할 수는 없습니다.

WAF 스테이징 환경을 설정하실 때 환경에 대한 접근 권한은 승인된 개발 팀에게만 허용하시기 바랍니다. 다음과 같은 방법들이 가능합니다.

- AWS WAF 를 이용해서 조직에서 사용되는 퍼블릭 IP 주소 대역밖에서의 요청을 차단합니다. 다만 이 방법은 인증기능이 없으며 개발자들이 원격으로 업무시에는 어려움이 있습니다(VPN 과 proxy 를 통해 회사 네트워크에서 접근해야합니다).
- 애플리케이션 내에 승인 메커니즘을 구현하고, CloudFront 의 캐시 동작 설정에서 해당 승인 헤더를 원본으로 전달하도록 합니다.
- 애플리케이션의 변경을 원치 않을 경우 엔드포인트로 승인기능을 넘길 수 있습니다. 예를 들어 CloudFront 를 사용한다면 접근 제어기능을 제공하는 Lambda@Edge 펑션을 사용하거나 CloudFront 에서 자체적으로 제공하는 서명된 쿠키를 사용할 수 있습니다.

스테이징에 배포하기

웹 ACL 을 배포할 때 다음과 같은 설정으로 시작하시기 바랍니다.

1. **정의된 요구사항에 따라 규칙을 추가합니다.**

   AWS WAF 에 익숙하지 않거나 특정한 요구사항이 없다면 AWS 관리형 규칙에서 제공되는 일반적인 웹 취약점에 대한 보호부터 시작하실 수 있습니다. AWS WAF 는 순서대로 규칙을 처리하며 일치된 규칙의 작업을 실행한 뒤에는 웹 ACL 처리를 멈추기 때문에 웹 ACL 내에서 규칙의 순서는 중요합니다.

   Note: 규칙을 차단 모드로 배포하면 스테이징 환경에서 규칙들이 어떻게 테스트 트래픽에 영향을 미치는지 확인할 수 있습니다. 하지만, 운영 환경에 적용하기 전에 운영자가 WAF 의 행동 방식에 익숙해지도록 스테이징에도 운영 환경 배포 절차에 따라 배포를 고려하시기 바랍니다. 대다수의 경우 새로운 규칙은 운영 환경에 개수 모드로 시작한 다음 차단 모드로 전환합니다. 이렇게 하면 잘못 설정된 WAF 규칙이 정상적인 트래픽을 차단 하여 애플리케이션의 가용성을 떨어트리는 상황을 피할 수 있습니다.

2. **HTTP flood 등 DDoS 공격에 대한 보호를 위해 비율 기반 규칙을 활성화 합니다.**

비율 기반 규칙은 5 분동안의 윈도우 내에 있었던 IP 주소당 요청의 횟수를 추적합니다. 이 윈도우는 30 초단위로 갱신되며 제한을 초과하면 전체 5 분이 되지 않더라도 즉시 조치를 취합니다. 비율 기반 규칙은 요청 횟수가 허용된 수치 이하로 내려올 때까지 해당 IP 주소의 요청을 계속 차단합니다.

Note: 비율 제한은 5 분내에 100 개의 요청까지 낮출 수 있으나 우선 2,000 개의 요청부터 시작하여 순차적으로 낮추시기 바랍니다.

스테이징 환경에 설정한 규칙이 충분히 준비되면 웹 ACL 의 규칙을 복사함으로써 운영 환경으로 복제할 수 있습니다. 웹 ACL 개요 페이지에서 모든 규칙을 포함하는 웹 ACL 설정을 JSON 파일로 다운로드가 가능합니다. JSON 파일을 다운로드 받은 후에는 수동으로 규칙을 복사해서 새 웹 ACL 을 만들거나, JSON 을 YAML 로 만들어서 CloudFormation 템플릿에 포함시킨 후 웹 ACL 을 배포할 수 있습니다.

모니터링과 가시성

구현된 WAF 의 운영을 위해서는 웹 ACL 이 어떤 요청을 차단하고 있는지에 대한 가시성을 확보해야 합니다. 이러한 가시성은 위협 인텔리전스의 확보, 규칙의 강화, 오탐의 트러블슈팅, 사고의 대응에 유효합니다. AWS WAF 는 여러 종류의 모니터링 옵션을 제공합니다.

Amazon CloudWatch 를 이용한 모니터링

AWS WAF 를 위한 대시보드를 설정하여 웹 ACL 의 규칙에 대한 작업 정보를 표시할 수 있습니다. 각각의 규칙에 대해서 CloudWatch 는 2 주일간 기록되는 *AllowedRequests, BlockedRequests, PassedRequests* 와 같은 메트릭을 준실시간으로 송출합니다. 다음의 이미지는 CloudWatch 로 쉽게 설정할 수 있는 웹 ACL 의 애플리케이션 보호에 대한 실시간 및 추적정보를 보여주는 예제입니다. 여기에 더하여, CloudWatch 메트릭에 알람을 설정하여 특정 WAF 규칙이 미리 지정된 임계점을 넘어 비정상적으로 자주 발생할 경우 알림을 받을 수 있습니다.

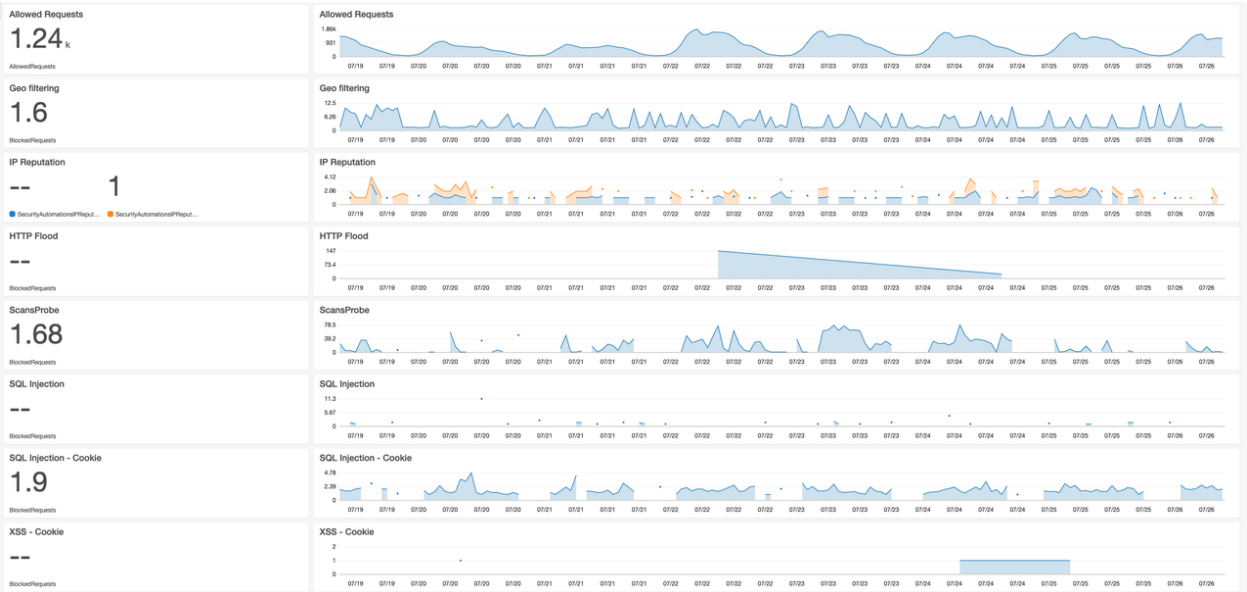


그림 4 – CloudWatch 를 이용한 보안 대시보드

그러나 CloudWatch 는 처리된 요청 자체에 대한 정보는 제공하지 않습니다. 만약 처리된 요청에 대해 상세한 정보가 필요하면 다음 두가지 방법이 있습니다.

- **WAF 콘솔에서 샘플링된 WAF 로그를 봅니다.** 각각의 샘플링된 요청에 대해서 IP 주소나 요청에 포함된 헤더 등의 상세 정보를 볼 수 있습니다. 이를 통해 스테이징 환경에서 오탐을 빠르게 디버깅할 수 있습니다. 샘플링된 요청은 사용자가 선택한 시간(최대 3 시간까지) 내에서 웹 ACL 이 처리한 요청중 임의로 5,000 개를 추출합니다.
- **상세한 정보를 위해 AWS WAF 로그를 활성화 하고 처리합니다.** 이 방법은 운영 환경에서 더 깊이있는 트러블슈팅을 하기에 적합합니다. 모든 요청에 대해서 AWS WAF 로그는 HTTP/S 헤더와 작동된 규칙 정보를 제공합니다. 더하여 AWS WAF 로그는 SQL 삽입 공격과 XSS 규칙에 일치된 정확한 패턴정보를 '*terminatingRuleMatchDetails*' 필드를 통해 제공합니다. AWS WAF 로그는 Amazon Kinesis Data Firehose 를 통해 인입되고 Amazon S3 을 포함한 다양한 목적지로 전송될 수 있습니다. 최선의 가시성과 트러블슈팅을 위해 모든 운영 워크로드에 이 방법을 이용하시기를 권장합니다.

AWS WAF 로그를 이용하여 준실시간으로 애플리케이션의 보안상태를 확인하며 필요에 따라 요청의 상세 정보를 보도록 하는 사용자 지정 대시보드를 구축 하는 경우가 종종 있습니다. AWS WAF 로그를 바탕으로 AWS 서비스나 서드파티 서비스를 이용하여 자체 대시보드를 구축할 수

있습니다. 만약 Splunk, Datadog, Sumo Logic 등의 서드파티 모니터링 서비스를 사용한다면 WAF 로그를 해당 서비스로 전송할 수 있습니다. 예를 들어 Sumo Logic 은 WAF 로그 분석용 대시보드를 만들수 있는 템플릿을 제공합니다.

테스트와 튜닝

처음 WAF 구현이 완료되면 일반적으로 잠재적인 오탐과 미탐을 완화시키는 튜닝 단계를 거치게 됩니다. 미탐은 공격이 WAF 에 의해 감지되지 않는 경우로서, 규칙의 강화를 필요로 합니다. 오탐은 정상적인 요청이 WAF 에 의해 공격으로 오인되어 차단되는 경우입니다.

미탐

보안 요구사항에 기반하여 **미탐**을 확인하기 위해서 침투 테스트 제공자나, 서드파티의 자동화된 취약점 스캐너, 오픈소스 웹 애플리케이션 보안 스캐너 등을 사용할 수 있습니다. 자동화된 취약점 스캐너는 알려진 취약점과 미리 정의된 공격 벡터에 대비한 아키텍쳐를 빠르게 점검할 수 있는 반면, 알려진 경우들만 커버할 수 있으며 실제로 존재하지 않는 이슈를 표시할 수도 있습니다. 그러므로 취약점 스캐너로 WAF 규칙들을 테스트하는 것이 애플리케이션의 완전한 보호를 보장하지는 않음을 유념해 주십시오. 하지만 온전성 검사를 위해 취약점 스캐너를 이용할 수 있습니다. 이에 더하여 정기적으로 스캐너를 갱신하고 주기적으로 실행하여 새로운 공격형태를 인식하고 필요할 때마다 WAF 규칙 갱신을 권장합니다.

오탐

오탐은 주로 품질 보증(Quality Assurance, QA) 팀에서 애플리케이션 코드나 WAF 설정의 변경후 테스트 과정에서 주로 확인됩니다. 대부분의 경우 이러한 테스트는 의심스러운 요청의 형태와 악의적인 공격의 차이를 판정할 수 있을 만큼 애플리케이션에 대한 깊이 있는 이해가 요구됩니다. 몇몇 경우에는 QA 테스트 커버리지가 부족하여 운영 환경에서만 오탐이 발견되기도 합니다. 이러한 경우를 확인하기 위해서 다음을 고려하시기 바랍니다.

- CloudWatch 에서 선택된 WAF 규칙에 알람을 설정하여 미리 정해진 임계점 이상으로 규칙이 트리거되면 알림을 받도록 합니다.

- 실 사용자가 예상치 못한 접근 실패를 보고할 수 있도록 애플리케이션의 사용자 경험을 갱신합니다. 예를 들어 WAF 가 CloudFront 와 같이 배포된 경우 403 오류 코드를 대체하는 사용자 지정 오류 페이지를 이용하여 사용자 친화적인 응답을 제공할 수 있습니다. 이 페이지에서 사용자에게 발생된 이슈 정보를 제공하도록 표시할 수 있습니다.
- WAF 로그를 활성화 합니다. 보안 팀과 애플리케이션 팀이 주기적으로 리뷰하고 차단된 트래픽의 기준선을 설정하여 위협의 패턴과 차단된 트래픽의 비정상 상태를 확인할 수 있게 합니다.

오탐을 감지하게 되면,오탐이 왜 발생했는지 확인해야합니다. AWS Management 콘솔에서 트리거된 규칙의 로그 샘플을 확인함으로써 빠르게 원인을 확인할 수 있습니다. 조금 더 상세한 방법은 AWS WAF 를 통해 생성된 로그를 확인하여 차단된 요청들을 URL, IP, 타임스탬프별로 필터링하는 것입니다. AWS WAF 를 CloudFront 와 함께 사용한다면, CloudFront 는 요청이 차단되었을 때 응답 헤더와 본체에 `Request ID` 를 포함하여 응답합니다. 이 `Request ID` 를 이용하여 WAF 로그에서 해당 기록을 찾을 수 있습니다.

그림 5 – CloudFront 가 표시하는 오류 메시지

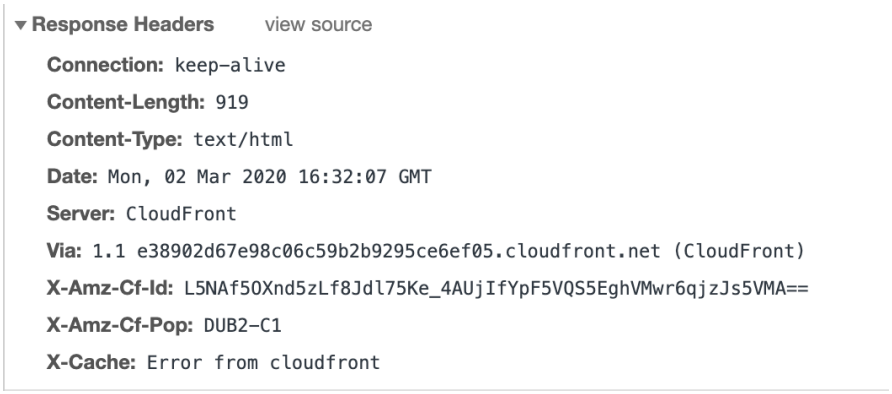


그림 6 – 오류 페이지 응답 헤더

로그 기록으로 작동된 규칙을 확인할 수 있습니다. 예를 들어 특정한 헤더가 길이 제한 규칙을 따르지 않았을 수 있습니다. 때때로 이런 정보가 이슈를 파악하기에 부족할 수도 있습니다. 예를 들어 SQL 삽입 공격 규칙이 요청의 쿠키에 의해 작동되었지만 정확히 쿠키의 어떤 부분이 원인인지 분명하지 않을 수 있습니다. 로그의 *terminatingRuleMatchDetails* 필드는 SQL 삽입 공격 규칙과 XSS 규칙에 대해서 상세한 일치 내용을 제공합니다.

오탐을 해결하기

이슈를 이해하고 나면 이를 해결할 수 있습니다. 가장 좋은 접근법은 애플리케이션 코드를 수정하여 공격처럼 보이는 요청을 발생시키지 않는 것이지만 이는 시간과 노력이 필요합니다. 빠른 조치로서 WAF 규칙에 예외를 생성할 수 있지만, 이는 애플리케이션이 잠재적인 공격에 노출될 수 있음을 인지 하셔야합니다.

예제 1 – AWS CLI 를 이용한 규칙 재정의

정상적인 URL 패턴 'xxxx'이 URI 경로명을 검사하는 XSS 규칙에 의해 차단되었다고 가정하면, 'xxxx' URL 패턴을 예외 처리하기 위해 별도의 규칙을 추가하므로써 차단을 해제할 수 있습니다:

```
BLOCK [XSS condition] AND NOT[String Match Condition on Path]
```

다음의 WAF 규칙 문은 이 예제를 보여줍니다. 이 방법은 SQL 삽입 공격, XSS 규칙뿐만 아니라 다른 모든 사용자 지정 규칙에 사용 가능합니다.

```
{
  "Name": "XSSprotection",
```

```
  "Priority": 0,
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "XSSprotection"
  },
  "Statement": {
    "AndStatement": {
      "Statements": [
        {
          "XssMatchStatement": {
            "FieldToMatch": {
              "UriPath": {}
            },
            "TextTransformations": [
              {
                "Type": "URL_DECODE",
                "Priority": 0
              }
            ]
          }
        },
        {
          "NotStatement": {
            "Statement": {
              "ByteMatchStatement": {
                "FieldToMatch": {
                  "UriPath": {}
                },
                "PositionalConstraint": "CONTAINS",
                "SearchString": "xxxx",
                "TextTransformations": [
                  {
                    "Type": "LOWERCASE",
                    "Priority": 0
                  }
                ]
              }
            }
          }
        }
      ]
    }
  }
}
```

예제 2 – 관리형 규칙을 이용한 규칙 재정의

AWS 관리형 규칙의 핵심 규칙 집합같은 관리형 규칙에 의해 차단되는 정상 URL 패턴 'xxxx'가 있다고 가정하면, 차단을 일으키는 규칙의 작업을 개수 모드로 재정의함으로써 오탐을 임시적으로 완화할 수 있습니다. 하지만 이 경우 애플리케이션은 해당 규칙에 의해 차단되는 공격에 노출 됩니다.

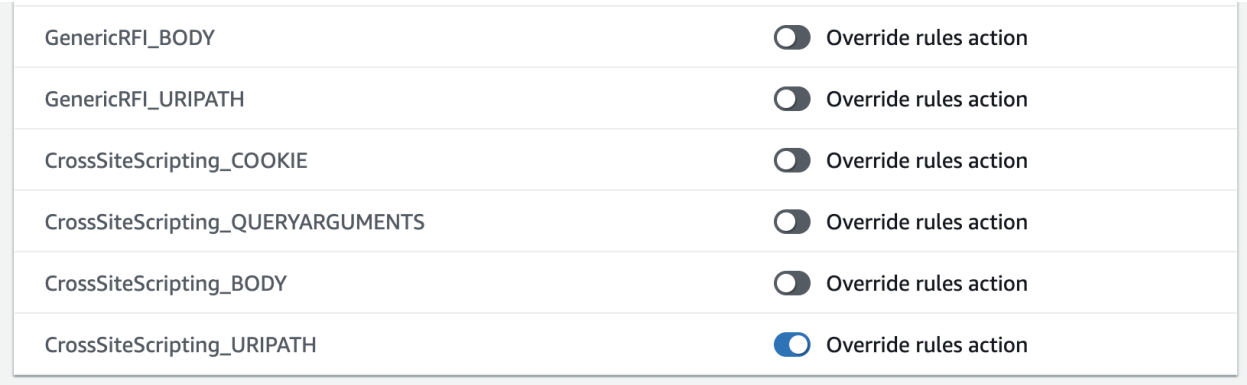


그림 7 – 관리형 규칙의 규칙 작업 재정의

다른 접근법은 관리형 규칙 이전에 오탐된 'xxxx' 패턴을 허용하는 규칙을 생성하여 이후의 규칙들을 실행하지 않도록 하는 방법입니다. 이 접근법은 오탐 허용처리 규칙이후의 규칙들에 의해 차단될 수 있는 공격에 애플리케이션을 노출시킵니다. 허용처리 규칙을 웹 ACL 순서의 가장 낮은 순위에 위치시켜 위험을 감소시킬수 있습니다.

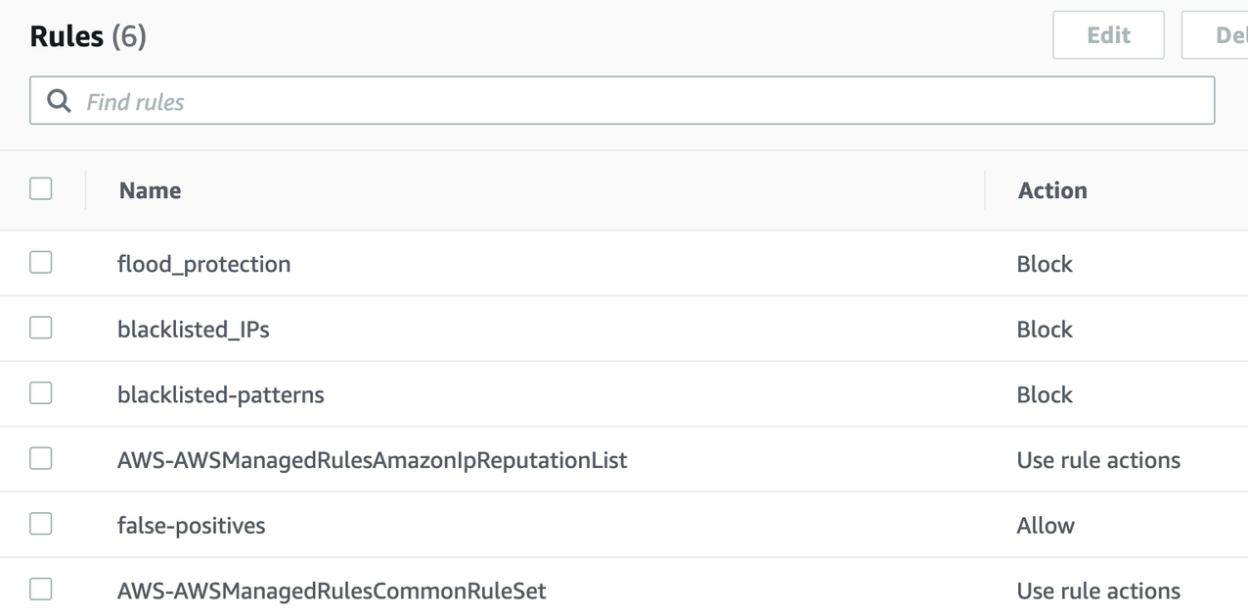


그림 8 – 예외 처리를 위해 규칙을 추가하기

WAF 규칙에 예외 처리를 하거나 애플리케이션 코드를 변경함으로써 오탐에 대한 처리를 하고 나면 cURL 이나 PostMan 같은 도구로 오탐을 일으켰던 요청을 재실행하여 예외처리 상태를 확인할 수 있습니다. 예외처리가 제대로 되었다면 요청은 차단되지 않을 것입니다.

운영 환경에 배포하기

운영 준비상태

스테이징 환경에서 WAF 구현을 테스트하고 검증한 후에는 언제 운영환경에 배포할지를 결정해야합니다. 사용자 트래픽이 가장 적을 것으로 생각되는 날짜와 시간을 선택하십시오. 배포 이전에 애플리케이션 팀과 보안 팀이 운영 준비상태를 검토하고 변경의 롤백 방안과 모든 메트릭과 알람이 제대로 설정되었는지 대시보드를 검토하십시오. 팀원들이 롤백이나 다른 완화조치를 어떻게 해야하는지 알 수 있도록 운영 매뉴얼을 작성할 수도 있습니다. 보안 위협 이벤트의 발생시, 팀은 설정을 어떻게 갱신하고 각기 다른 AWS 계정에 배포하는지, 문제를 어떻게 트러블슈팅하고 응답 해야하는지 알아야합니다. 운영 매뉴얼은 이벤트 발생시의 각 단계에 대해 서술합니다. 예를 들어 운영 매뉴얼은 다음과 같은 지시사항을 포함할 수 있습니다.

1. 어떻게 설정을 배포하는가.
2. 오탐을 어떻게 완화하는가.
3. 이슈를 어떻게 트러블슈팅하는가.
4. 애플리케이션 상태 점검을 위해 어떤 메트릭을 검토해야하는가. CloudFront 응답 종류(HTTP 200, 4xx, 5xx), 애플리케이션과 데이터베이스 서버의 CPU 및 메모리, WAF 메트릭 등.
5. 상세한 검사를 위해 로그 데이터에서 요청 데이터를 특정하는 주요 쿼리.
6. AWS WAF 를 설정하고 배포하는 보안 팀을 참여시키는 순서.
7. AWS 를 참여시키는 순서.

AWS Shield Advanced 사용자는 DDoS 공격을 받는 상황이 되면 AWS DDoS 응답 팀(DDoS response team, DRT)를 참여시킬 수 있습니다. DRT 는 의심스러운 활동의 분석과 이슈 완화를 지원할 수 있습니다. 완화 작업은 종종 고객의 계정내에 AWS WAF 규칙과 웹 ACL 생성 및 갱신을 포함합니다. DRT 는 고객을 위해서 고객의 AWS WAF 설정을 조사하고 AWS WAF 규칙이나 웹 ACL 을 대신 생성하거나 갱신할 수 있습니다. AWS Shield Advanced 설정 작업 중, DRT 가 이러한 작업을 할 수 있도록 접근 권한을 미리 승인하시기를 권장합니다. 접근 권한이 승인되어 있으면 실제 공격 상황에서 완화 작업의 지연을 피할 수 있습니다.

그리고 배포 이전에 네트워크 엔지니어, 보안 엔지니어, 애플리케이션 팀, AWS 어카운트 팀을 포함한 모든 관련자들과 사용할 공통 연락망을 설정하십시오. 이 연락망을 통해 변경 상황을 공유하십시오.

배포

AWS WAF 를 운영 환경 엔드포인트에 활성화할 준비가 되면, 스테이징 환경에서 발견되지 않았던 잠재적인 오탐을 발견하기 위해서 개수 모드를 사용할지 결정하십시오. 처음 WAF 규칙을 배포하는 경우라면 이 기법으로 정상적인 트래픽이 차단되는 상황을 피할 수 있습니다. 하지만 개수 모드로 규칙을 배포하면 이후 차단 모드로 바꾸기 전까지 애플리케이션은 공격에 취약하므로 주의를 요합니다. 오탐 발생 확률이 높지 않다고 판단되면 개수 모드는 필요 없을 수도 있습니다.

대시보드와 메트릭을 검토하여 규칙들이 일치되는 내용이 적절하면 개수 모드를 차단 모드로 변경할 수 있습니다.

만약 CDN 기반의 WAF 등 다른 WAF 를 사용하고 있다면, 기존 트래픽을 점진적으로 AWS WAF 로 이관하는게 좋습니다. 예를 들어 Amazon Route 53 의 가중치 라우팅 정책을 이용하여 트래픽을 WAF 가 활성화된 새로운 CloudFront 엔드포인트로 점진적으로 이관할 수 있습니다.

기능 평가를 위해 AWS WAF 와 다른 WAF 를 겹치는 구성은 규칙 작동 방법에 충돌이 생길 수 있으므로 권장하지 않습니다. 예를 들어 WAF 의 기능중 IP 주소를 차단하는 기능이 있습니다. 다수의 WAF 솔루션은 TCP 연결된 클라이언트의 IP 주소로 접근 IP 주소를 판단합니다. WAF 솔루션을 겹쳐 사용하면 클라이언트에 노출되지 않은 WAF 는 클라이언트 IP 주소에 대한 가시성을 잃어 효과적으로 작동할 수 없습니다. 몇몇의 경우에는 WAF 가 다른 WAF 의 정상적인 트래픽을 의도치 않게 차단할 수도 있습니다.

AWS Firewall Manager 를 사용하여 AWS WAF 를 배포하기로 결정했다면, 엔드포인트 하나에 배포하는 정책부터 시작하시기를 권고합니다. 엔드포인트 하나로 정책을 테스트하고 검증한 뒤 정책이 보호 대상 엔드포인트를 모두 포함하게 갱신할 수 있습니다. 유사한 엔드포인트(CloudFront 배포, Application Load Balancer, API Gateway)에 공통 태그를 달고 규칙 그룹을 해당 태그를 가진 모든 엔드포인트에 적용하는 Firewall Manager 정책을 생성할 수 있습니다. 새로운 운영 환경 엔드포인트 생성시 올바른 규칙 그룹이 연결되어야하므로, 정확한 태그를 하는 것도 중요합니다.

배포 후 단계

운영 배포후에는 정기적으로 애플리케이션을 검토하고 모니터하는 것이 중요합니다. 애플리케이션 팀과 보안 팀은 애플리케이션 트래픽 패턴의 기준을 숙지하기 위해 대시보드를 리뷰해야합니다. AWS WAF 로그를 Amazon ElasticSearch Service 나 Amazon Athena, 서드파티 SIEM 도구로 애플리케이션 트래픽 패턴과 행동의 변화를 분석하십시오. 이 정보를 기반으로 비정상 형태나 새로운 위협, 오탐에 대해 깊이 있게 분석할 수 있으며 적절한 방어를 위해서 WAF 규칙 설정작업을 반복할 수 있습니다. 운영 매뉴얼도 주기적으로 검토하고 훈련하며 갱신해야합니다. 관련 팀들이 보안 이벤트 응답에 익숙하도록 주기적으로 운영 매뉴얼에 따라 훈련하시기를 권유합니다. 예를 들어 모의 보안 상황을 통해 운영자가 필요 작업 순서를 숙지하도록 합니다.

최신의 위협과 취약점에 대비하기 위해 정기적인 침투테스트 실행을 고려하시기 바랍니다. 새롭게 발견된 위협에 대응하기 위해서 WAF 규칙을 최신으로 유지하는 것이 중요합니다. 이 작업 부담을 줄이기 위해서 사용자 지정 규칙 대신 관리형 규칙을 사용할 수 있습니다. 관리형 규칙은 WAF 공급업체(AWS 나 파트너)가 진화하는 위협 환경에 따라 갱신합니다. 하지만 애플리케이션에 특화된 사용자 지정 규칙을 애플리케이션의 변화에 따라 갱신하는 것도 중요합니다.

비용 고려사항

AWS WAF 는 웹 ACL 사용, 규칙 및 검사된 요청 수에 따라 요금이 부과되는 독립 실행형 요금을 제공합니다. 로깅 구성의 경우 Amazon Kinesis Data Firehose 사용량에 따라 요금이 청구됩니다. AWS Marketplace 에서 WAF 관리형 규칙을 사용하도록 선택한 경우 관리형 규칙을 구독하고 사용한 만큼만 요금을 지불할 수 있습니다.관리 규칙에는 시간 단위로 요금이 부과되므로 계약이나 구독 약정이 없습니다.

요청량이 많은 워크로드의 경우 요청당 요금을 줄이기 위해 AWS Shield Advanced 를 검토하시기 바랍니다. AWS WAF 를 AWS Shield Advanced 에 의해 보호되는 리소스와 함께 사용하는 경우 AWS WAF 및 AWS Firewall Manager 사용에 대한 추가 요금이 부과되지 않습니다. AWS Shield Advanced 와 관련된 요금만 지불하면 됩니다.이러한 접근 방식은 요청이 많은 워크로드에 대한 비용을 최적화하는 데 도움이 될 수 있습니다.요금에 대한 자세한 내용은 AWS Shield, AWS Firewall Manager, AWS WAF 요금 페이지를 참조하십시오.

결론

이 백서에서는 AWS WAF 가 해결할 수 있는 다양한 위협에 대해 설명하고, WAF 를 구현할 때 고려해야 할 다양한 요구 사항 및 업무 중단을 최소화하면서 AWS WAF 를 운영 환경에 배포하는 방법을 설명합니다. 웹 애플리케이션 보호를 담당하는 사람은 누구나 이 백서를 사용하여 AWS WAF 를 보호 도구의 일부로 활용하고 구현하는 방법을 더 잘 이해할 수 있습니다.

기여한 분들

본 문서를 작성하는데 기여한 분들은 다음과 같습니다.

- Achraf Souk, Principal Solutions Architect, Edge Services
- Tino Tran, Principal Solutions Architect, Edge Services
- Damindra Bandara, Senior Security Consultant, Professional Services

참고 문헌

추가 정보를 위해 다음을 참조하세요

- AWS WAF Documentation
- AWS Security Incident Response Guide
- AWS WAF Security Automations Solution
- AWS Best Practices for DDOS Resiliency

문서 개정 이력

일자	설명
2020 년 5 월	첫 번째 출판
2020 년 10 월	한글화