DDoS 대응을 위한 AWS 모범사례

2016 년 6 월

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

공지사항

이 문서는 정보제공을 목적으로 제공됩니다. 문서에 설명된 제품들과 방법들은 본 문서가 공개된 날짜를 기준으로 유효한 내용이며 추후 사전 공지 없이 변경될 수 있습니다. 이 문서에 나와있는 서비스나 설명 및 방법들에 대한 최종 판단은 고객 여러분의 개별적인 판단에 달려있으며, 이를 어떤 종류로든 보장하지는 않습니다. 이 문서에 대한 내용에 대해서 AWS 에서는 어떠한 보장도 제공하지 않습니다. AWS 와 고객 여러분 간의 법적인 책임은 AWS 와의 합의(Agreement)에 따르며 이 문서는 AWS 와 고객간의 맺는 합의(Agreement)에 해당하지 않으며 어떠한 영향도 주지 않습니다.

원본문서 : AWS Best Practices for DDoS Resilience
https://do.awsstatic.com/whitepapers/DDoS_White_Paper_June2015.pdf

한글번역 : 임기성 매니저 (gisunlim@amazon.com) – AWS Solutions Architect Team

목차

개요

본 문서는 AWS 를 사용하시는 고객 중 DDoS (Denial of Service)공격에 대비하여 자사 어플리케이션의 가용성을 높이고자 하는 분들을 위해 쓰여졌습니다. 본 문서에서는 DDoS 공격에 대한 개요, AWS 에서 제공되는 기능들, 완화 기법들, 그리고 어플리케이션의 가용성을 보호하는데 도움이 될 만한 DDoS 대응 레퍼런스 아키텍쳐를 제공합니다.

소개

본 문서는 네트워킹, 보안 및 AWS 에 대한 기본적인 지식과 경험이 있는 IT 실무자, 보안 담당자 들을 대상으로 쓰여졌습니다. 각각의 섹션에 나와있는 구성방법 등은 해당 페이지에 연결된 링크를 따라가면 어떻게 설정 하고 구성하는지에 대한 정보들을 쉽게 얻으실 수 있으며, 참고로 AWS 의 Re:Invent 세션 SEC307 – Building a DDoS-Resilient Architecture with AWS[1] 와 SEC306 – Defending Against DDoS Attacks[2] 를 통해 더 많은 정보들을 얻으실 수 있습니다

DDoS 공격

DDoS 공격의 목적은 최종 사용자(End User)가 여러분의 웹 사이트나 어플리케이션을 이용할 수 없도록 만드는 것입니다. 이것을 위해서, 공격자는 다양한 기술들을 사용하는데, 대표적으로 네트워크나 다른 자원들을 고갈시켜서, 사용자의 정당한 요청을 처리할 수 없게끔 만들어 버립니다. 가장 간단한 형태로써, DoS 공격은 단일 공격자가 하나의 호스트만을 가지고 아래 그림처럼 대상을 공격하는 것입니다.

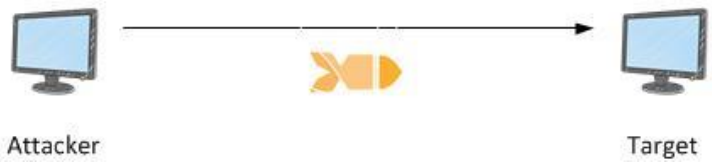

그림 1. 일반적인 DoS 공격 방식

DDoS 공격의 경우 공격자들은 여러 대의 호스트를 사용하는데, 이때 해당 호스트들은 공격자를 도와 같이 공격을 수행하는 호스트일 수도 있고, 혹은 특정 경로를 통해 악성코드에 감염되어 공격자에게 조종당하는 좀비 호스트 일수도

있습니다. 아래 그림은 일반적인 DDoS 공격을 도식화 한 것인데, 각각의 공범 혹은 좀비 호스트들이 목표 호스트를 상대로 많은 양의 패킷 전송 혹은 연결 요청을 보냄으로써 해당 호스트의 자원을 고갈 시키려는 시도를 합니다.

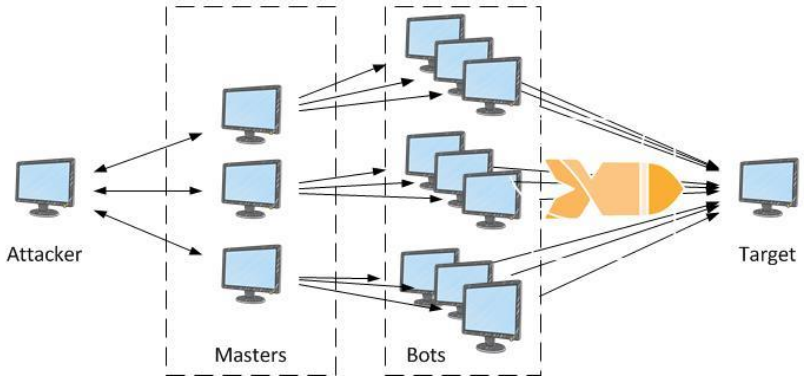

그림 2. 일반적인 DDoS 공격 방식

DDoS 공격은 아래 테이블 1 에 나와 있듯이, Open Systems Interconnection (OSI) 모델의 3,4,6,7 계층에서 가장 흔하게 이루어 집니다. 계층 3 과 4 의 공격은 OSI 모델의 네트웍과 전송 계층에 해당하며, 본 문서에서는 '인프라 계층 공격'이라고 부르겠습니다. 계층 6 과 7 공격은 OSI 모델의 표현과 응용 계층에 해당하며, 본 문서에서는 이를 '응용 계층 공격'으로 부르겠습니다.

#	계층	유닛	설명	대표적인 공격 벡터
7	응용(Application)	데이터	어플리케이션에 대한 네트웍 프로세스	HTTP floods, DNS query floods
6	표현(Presentation)	데이터	데이터 표현과 암호화	SSL abuse
5	*세션(Session)*	*데이터*	*호스트 간 통신*	*N/A*
4	전송(Transport)	세그먼트	종단 간 연결 및 신뢰성	SYN floods
3	네트웍(Network)	패킷	경로 결정과 논리적인 어드레싱	UDP reflection attacks
2	*데이터 링크(Data Link)*	*프레임*	*물리적인 어드레싱*	*N/A*
1	*물리(Physical)*	*비트*	*미디어, 시그널,바이너리 전송*	*N/A*

테이블 1: Open Systems Interconnection (OSI) 모델

각 계층에서 발생하는 공격의 유형이 상이하고 대응 방식도 완전히 다르기 때문에, 이와 같은 구분을 이해하는 것이 굉장히 중요합니다.

인프라 계층 공격

가장 흔한 DDoS 공격인, User Datagram Protocol (UDP) 증폭 공격과 Synchronize (SYN) flood 공격이 대표적인 인프라 계층 공격입니다. 공격자는 이와 같은 방식을 통해 네트웍이나 서버, 방화벽, IPS, 로드 밸런서 등의 시스템 용량을 넘어서는 많은 양의 트래픽을 발생시킵니다. 이같은 방식은 탐지 하기 쉬운 독특한 시그니쳐를 가지고 있습니다. 이러한 공격을 효과적으로 완화시키기 위해서는 공격자가 발생시킨 양을 넘어서는 충분한 네트웍이나 시스템 자원을 보유하고 있어야 합니다.

UDP 는 비 상태 유지(stateless) 프로토콜입니다. 따라서 공격자는 요청자의 정보를 임의로 변경하여 서버로부터 더 큰 응답이 돌아가도록 조작할 수 있습니다. 요청 사이즈 당 응답 사이즈의 비율인 증폭 팩터는 Domain Name System (DNS), Network Time Protocol (NTP), Simple Service Discovery Protocol (SSDP)와 같이, 공격에 사용된 프로토콜에 따라 달라집니다. 예를 들어, DNS 의 증폭 팩터는 28 에서 54 의 범위를 가집니다 — 아래 그림 3 에서와 같이 DNS 서버로 64 바이트의 요청 페이로드를 보내는 경우, 공격 대상은 3400 바이트의 원하지 않는 트래픽을 받게 됩니다.

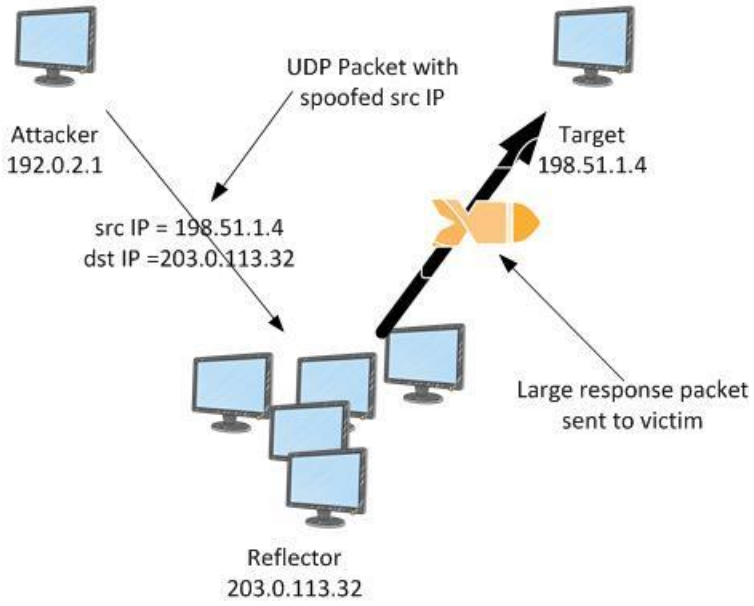

그림 3: UDP 증폭 공격

SYN flood 의 경우는, 'Half-Open' 상태로 연결을 유지하도록 하여 시스템의 가용한 리소스를 고갈시키는 방식으로써, 수 십 Gbps 의 규모로 발생할 수

있습니다. 아래 그림 4 와 같이, 웹 서버 등에 TCP 서비스를 연결할 때는 처음에 SYN 패킷을 보내게 됩니다. 해당 서버는 SYN-ACK 을 리턴하고, 클라이언트는 3-Way Handshake 를 완성하기 위해서 다시 ACK 를 리턴하게 됩니다.

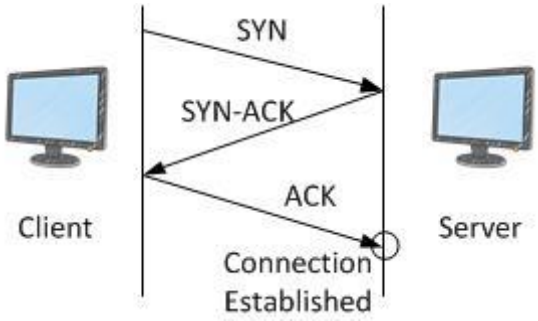

그림 4: SYN 3-웨이 핸드 쉐이크

SYN flood 공격에서, 마지막 ACK 를 리턴하지 않으면 해당 서버는 계속해서 응답을 기다리게 됩니다. 이를 통해 그 서버에 접근하려는 다른 사용자의 접속이 방해 받을 수 있습니다.

응용 계층 공격

비교적 덜 선택되어 지긴 하지만, 공격자는 7 계층 혹은 응용 계층에 대해 공격하기도 합니다. 이와 같은 공격을 보면, 서버가 가용하지 않게 만들기 위해 어플리케이션의 특정 기능을 계속해서 호출하는 방식을 취하기 때문에, 인프라 계층에 대한 공격과는 차이가 있습니다. 어떤 경우에는, 많은 네트웍 트래픽 볼륨을 생성하지 않는, 아주 적은 양의 요청을 통해서도 수행될 수 있으며, 이러한 공격은 탐지하거나 대응하기가 더욱 어렵게 됩니다. 이와 같은 응용 계층에 대한 공격의 예로는 HTTP flood, cache-busting attack, WordPress XML-RPC flood 같은 것들이 있습니다.

HTTP flood 의 경우, 공격자는 웹 어플리케이션 사용자의 실제 HTTP 요청을 보내게 됩니다. 일부 간단한 HTTP flood 의 경우, 특정 리소스를 노리고 공격하기도 하는 반면, 좀더 복잡한 공격은 사용자의 행동을 흉내내기도 합니다. 이 방식은 요청률 제한(request rate-limiting)과 같은 일반적인 완화 기법을 가지고 대응하는 것을 어렵게 합니다. Cache-busting 공격은 쿼리 스트링의 변경을 통해 CDN 캐쉬 Hit 를 방해하여, 원본 웹 서버의 부하를 발생시키게끔 하는 일종의 HTTP flood 입니다.

WordPress XML-RPC flood 의 경우, WordPress pingback flood 라고도 알려져 있으며, 대량의 HTTP 요청을 발생시키기 위해 WordPress 브랜드의 컨텐츠 관리 소프트웨어 상에서 호스팅 되는 웹사이트에 있는 XML-RPC API 기능을 이용하게 됩니다. 'pingback' 기능은 WordPress 사이트 A 에서, 다른 WordPress 사이트 B 로의 링크를 생성할 때, 이를 해당 사이트 B 로 통보하는 기능을 말합니다. 결과적으로, 사이트 B 입장에서는 링크를 확인하기 위해 사이트 A 에 접근하게 됩니다. 'pingback flood'공격에서, 공격자는 이와 같은 과정을 악용하여 사이트 B 가 사이트 A 를 공격하도록 할 수 있습니다. 이런 유형의 공격은 HTTP 요청 헤더의 'User-Agent' 부분에 'WordPress'라는 분명한 시그니쳐를 가지고 있게 됩니다.

응용 계층 공격은 또한 DNS 서비스를 대상으로 할 수 있습니다. 이런 공격의 가장 흔한 형태는 공격자가 다량의 정상적인(well-formed) DNS 쿼리를 통해 DNS 서버의 리소스를 고갈시키는 DNS query flood 입니다. 이와 같은 공격은 또한 DNS 서버의 로컬 캐쉬 Hit 를 회피하기 위해 서브 도메인 문자열을 랜덤하게 변경해주는 'cache-busting' 컴포넌트를 이용하기도 합니다.

SSL 상에서 서비스 되는 웹어플리케이션을 대상으로, 공격자는 SSL Negotiation 프로세스를 공격하기도 합니다. SSL 은 값비싼 계산이 필요하며, 공격자는 이를 노리고 의미 없는 데이터를 보냄으로써 서버의 가용성에 영향을 줄 수 있습니다. 이런 공격의 다른 변형으로는, SSL Handshake 를 시도하는 공격자가 계속해서 암호화 방식을 협상하는 방식도 있습니다. 이와 유사하게 공격자는 다수의 SSL 세션들을 생성하거나 삭제하는 과정에서 서버의 리소스를 고갈 시키는 방식을 선택하기도 합니다.

완화 기법들

AWS 인프라는 기본적으로 DDoS 에 대응할 수 있게 설계되어 있으며, 초과하는 트래픽을 자동으로 탐지하고 필터링할 수 있는 DDoS 완화 시스템들의 지원을 받습니다. 여러분들의 어플리케이션 가용성을 보호하는데 이러한 기능들을 잘 활용하는 것이 필요합니다.

AWS 의 가장 흔한 사용 사례 중 한가지는 인터넷 상에서 사용자에게 정적 혹은 동적 컨텐츠를 제공하는 웹 어플리케이션입니다. 아래 그림 5 를 보면, 흔히 적용되는 웹 어플리케이션의 DDoS 대응 레퍼런스 아키텍쳐가 있습니다.

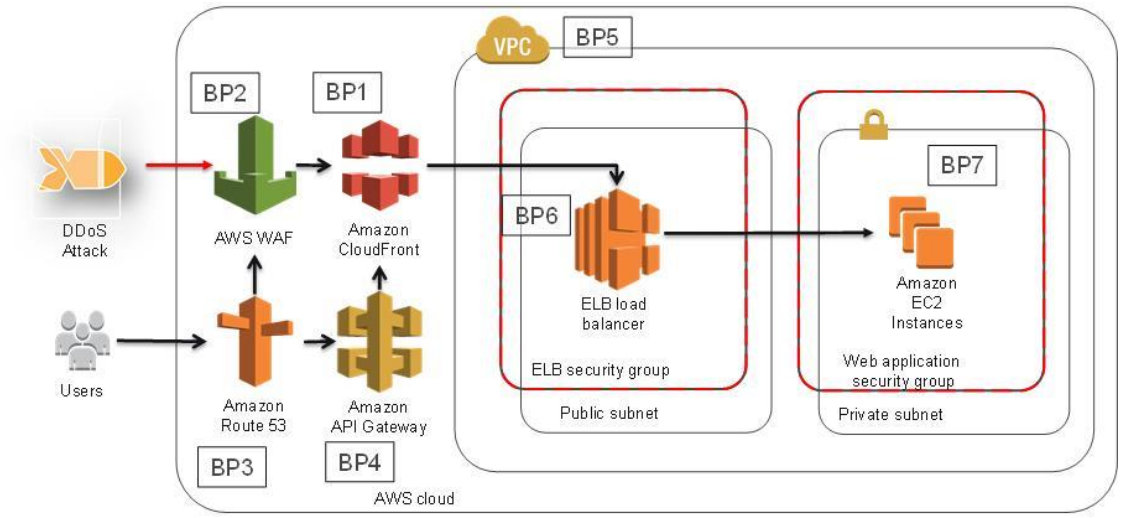

그림 5: DDoS-대응 레퍼런스 아키텍쳐

이 레퍼런스 아키텍쳐를 보면, 대상 웹 어플리케이션이 DDoS 공격에 대한 대응력을 높여주는 많은 AWS 서비스들을 이용하고 있음을 알 수 있습니다. 본 문서에서는 뒷 부분에서 개별로 설명하기 위해, 위 아키텍쳐 상의 각각의 모범 사례들에게 번호를 부여하였습니다. 예를 들어, Amazon CloudFront 가 제공하는 기능에 대한 설명은 'BP1'이라는 섹션에서 설명하게 됩니다. 이들 서비스들과 각 기능들에 대한 요약은 아래 테이블 2 에 있습니다.

	AWS 엣지 로케이션			**AWS 리전**		
	Amazon CloudFront 와 AWS WAF (BP1, BP2)	Amazon API Gateway (BP4)	Amazon Route 53 (BP3)	Elastic Load Balancing (BP6)	Amazon VPC (BP5)	Amazon EC2 with Auto Scaling (BP7)
3 계층 (예, UDP reflection) 공격 완화	✔	✔	✔	✔	✔	
4 계층 (예, SYN flood) 공격 완화	✔	✔	✔	✔		
6 계층 (예, SSL) 공격 완화	✔	✔	N/A	✔		
공격 지점 최소화	✔	✔	✔	✔	✔	
응용 계층의 트래픽을 흡수할 수 있는 확장	✔	✔	✔	✔		✔
7 계층 공격 완화	✔	✔	✔			
대규모 DDoS 공격과 과다 트래픽의 지리적인 고립과 분산 처리	✔	✔	✔			

테이블 2: 모범 사례 요약

ELB 와 EC2 와 같은 AWS 리전 내부의 서비스들은 해당 리전 내에서 예상치 못한 트래픽의 볼륨을 다룰 수 있게끔, 확장하는 방식으로 DDoS 대응력을 확보할 수 있습니다. Amazon CloudFront, AWS WAF, Amazon Route 53, Amazon API Gateway 와 같이 AWS 엣지 로케이션에서 제공되는 서비스들을 이용하면, 엣지 로케이션들의 글로벌 네트웍 커버리지를 이용하여, 여러분들의 어플리케이션이 보다 향상된 장애 대응력을 확보하고 대규모 트래픽 볼륨을 다룰 수 있도록, 규모를 확장시킬 수 있게 해 줍니다. 이어지는 섹션에서는 인프라와 응용 계층에 DDoS

공격에 대한 대응력을 제공하기 위해 이와 같은 서비스들을 이용하는 경우, 얻을 수 있는 이점에 대해 설명드립니다.

인프라 계층 방어(BP1, BP3, BP6, BP7)

전통적인 데이터 센터 환경에서, 여러분들은 용량을 과다하게 확보하거나 전문 DDoS 대응 장비를 적용하고, 혹은 DDoS 완화 서비스의 도움을 통해 트래픽을 정화하는 방식을 사용하여, 인프라 계층 DDoS 공격을 대응하고 있을 것입니다. AWS 상에서는 많은 자본 투자를 유발하거나 불필요한 복잡한 구성 없이, 여러분들의 어플리케이션이 대규모 트래픽 볼륨을 처리하고 쉽게 확장할 수 있는 옵션들을 확보할 수 있습니다. 대규모 DDoS 물량 공격을 완화시키기 위한 주요 고려사항은 가용한 전송 용량과 구간의 다변성을 확보하여, EC2 같은 AWS 리소스들을 공격 트래픽으로 부터 보호하는 것입니다.

인스턴스 사이즈 (BP7)

많은 AWS 고객들이 Amazon EC2 를 요건 변화에 맞추어 재빨리 확장하거나 축소시킬 수 있도록 사용하고 있습니다. 여러분들은 필요한 만큼 인스턴스들을 어플리케이션 환경에 추가하는 방식으로 수평적으로 확장하거나, 혹은 좀더 큰 인스턴스로 수직 확장시키는 방법을 선택할 수 있습니다. 어떤 인스턴스 유형들은 이를테면 10 기가비트 네트웍을 제공하거나 향상된 네트워킹(Enhanced Networking)기능을 제공하는데, 이러한 기능을 사용한다면, 보다 큰 트래픽 볼륨을 다룰 수 있게 됩니다.

10 기가 네트웍 인터페이스를 이용하면, 각 인스턴스가 좀더 많은 트래픽 볼륨을 처리할 수 있게 되며, EC2 인스턴스로 전송된 트래픽에 의한 인터페이스 정체 현상을 해결하는데 도움이 됩니다. 향상된 네트워킹(Enhanced Networking)을 지원하는 인스턴스들은 기존 보다 낮은 CPU 사용률을 가지고 더 높은 I/O 성능을 제공합니다. 이를 통해 해당 인스턴스는 보다 많은 네트웍 패킷 양을 처리할 수 있게 됩니다. AWS 는 여러분들에게 인바운드 데이터 전송 요금을 과금하지 않습니다.

10 기가 네트웍 인터페이스와 향상된 네트워킹(Enhanced Networking)을 지원하는 EC2 인스턴스에 관한 내용은 Amazon EC2 Instance Types[3]을 참조하시기 바랍니다. 향상된 네트워킹(Enhanced Networking)을 이용하는 방법은 Enabling Enhanced Networking on Linux Instances in a VPC[4]을 참조하시기 바랍니다.

리전의 선택 (BP7)

EC2 와 같은 많은 AWS 의 서비스들은 전세계 복수 개의 지역에서 사용할 수 있습니다. 이와 같이 지리적으로 분리된 영역을 AWS 리전 이라고 부릅니다. 어플리케이션을 설계할 때, 여러분들은 요건에 맞추어 하나 이상의 리전을 선택할 수 있으며, 이 경우 성능, 비용, 데이터 주권(Data Sovereignty) 같은 것들이 주요 고려 사항이 됩니다. AWS 는 최종 사용자까지의 최적화된 지연 시간과 성능을 제공하기 위해 각각의 리전 별로 적합한 인터넷 연결과 피어링 관계의 집합에 대한 접근을 제공합니다.

DDoS 대응의 관점에서 리전에 대한 선택은 매우 중요합니다. AWS 의 많은 리전들이 대규모 인터넷 거점 근처에 있습니다. 또한 많은 DDoS 공격들이 국제적인 규모로 발생하고 있기 때문에, 국제 회선 사업자나 높은 인지도를 가지고 있는 대규모 망 사업자와 가깝게 위치하는 것이 도움이 됩니다. 이것은 또한 최종 사용자가 대규모로 트래픽이 몰리는 어플리케이션에 접근할 경우에도 도움이 됩니다.

리전을 선택하는 것에 대한 자세한 내용은 Regions and Availability Zones[5] 을 참고하시기 바라며, 결정하실 때 도움이 될 만한 각각의 리전 별 특성에 대해서는 담당 영업에게 문의하시기 바랍니다.

부하 분산 (BP6)

대규모 DDoS 공격은 Amazon 의 단일 EC2 인스턴스의 범위를 상회할 수 있습니다. 이런 공격을 완화하기 위해, 여러분들은 과다한 트래픽 부하를 분산 처리하길 원할 수 있습니다. 여러분들은 Elastic Load Balancing (ELB)을 통해 많은 백 앤드 인스턴스들에게 부하를 배분함으로써 어플리케이션의 과부하에 대한 리스크를 경감시킬 수 있습니다. ELB 는 자동적으로 확장되어, 갑작스런 사용자 폭증이나 DDoS 공격과 같은 예측하지 못한 트래픽 양을 관리하는데 도움을 줄 수 있습니다.

ELB 는 오로지 정상적인 TCP 연결 만을 허용하게 됩니다. 이 말은 SYN flood 나 UDP reflection 공격과 같이, 흔히 DDoS 에서 사용되는 공격 유형들이 ELB 단에서 거부되고, 뒤에 있는 여러분의 어플리케이션 환경으로 전달되지 않는다는 것을 의미합니다. ELB 에서 이런 공격 유형을 탐지했을 때는 해당 트래픽을 흡수할 수 있게끔 자동으로 확장되며, 이 경우 여러분들은 별도의 비용을 부담하지않습니다.

ELB 를 이용하여 부하를 분산처리하고 Amazon EC2 인스턴스를 보호하기 위한 자세한 내용은 Getting Started with Elastic Load Balancing[6]를 참조하시기 바랍니다.

AWS 엣지 로케이션을 이용하여 확장성 구현하기 (BP1, BP3)

DDoS 공격을 흡수하고, 가용성에 대한 영향을 최소화하면서 최종 사용자에게 최적화된 지연 시간과 성능을 제공할 수 있는 중요한 방법은, 높은 수준으로 확장될 수 있고 다변화되어 있는 인터넷 연결을 제공하는 것입니다. AWS 엣지 로케이션은 Amazon CloudFront 나 Route 53 을 이용하여 웹 어플리케이션에 이러한 이점을 제공할 수 있는 별도의 네트웍 인프라를 제공합니다. 이런 서비스들을 이용한다면, 최종 사용자들과 가장 가까운 위치에서 여러분의 컨텐츠가 제공되거나, DNS 쿼리가 처리될 수 있게 됩니다.

엣지에서 웹 어플리케이션 서비스하기 (BP1)

Amazon CloudFront 는 정적, 동적, 스트리밍이나 대화형 컨텐츠를 포함한 여러분의 전체 웹 사이트를 서비스하는데 이용될 수 있는 컨텐츠 딜리버리 네트웍(CDN)입니다. 비록 해당 컨텐츠가 엣지 로케이션에 캐싱되지 못하는 형태라고 해도, 'TCP 연결 지속 기능(Persistent TCP connection)'과 'time-to-live (TTL)'변수를 이용하면, 컨텐츠의 제공 속도를 높이는데 도움이 될 수 있습니다. 이런 특성 때문에, 여러분의 웹사이트가 전혀 정적 컨텐츠가 없다고 해도, Amazon CloudFront 를 이용하여 웹 어플리케이션을 보호하는 것은 충분히 의미가 있습니다. Amazon CloudFront 는 흔히 DDoS 공격에 사용되는 유형인 SYN flood 나 UDP reflection 공격을 방어하기 위해, 오로지 정상적인 형태의 연결 만을 허용하며, 이를 이용하면 여러분의 원본(Origin) 서버로 이런 악성 공격들이 전달되는 것을 차단시켜 줄 수 있습니다. 또한 이런 DDoS 공격이 다른 지역에 영향을 주는 것을 방지하기 위해, 지리적으로 공격 원천과 가까운 지역으로만 제한시킬 수 있습니다. 이런 기능을 통해 대규모 DDoS 공격이 발생하는 동안에도, 최종 사용자에게 지속적으로 트래픽이 전송될 수 있는 능력을 획기적으로 향상 시킬 수 있게 됩니다. 여러분은 Amazon CloudFront 를 통해 AWS 환경이나 인터넷 상의 원본(Origin)을 보호할 수 있습니다.

Amazon CloudFront 를 이용하여 웹 어플리케이션의 성능을 최적화 하는 부분에 대한 자세한 설명은 Getting Started with CloudFront[7]을 참고하시기 바랍니다.

엣지에서 도메인 이름 변환하기 (BP3)

Amazon Route 53 은 여러분의 웹 어플리케이션으로 트래픽을 보낼 때 이용하는 높은 가용성과 확장성을 제공하는 DNS 서비스입니다. 이 서비스는 트래픽 플로우, 지연시간 기반 라우팅, 지리적 DNS 서비스, 상태 체크, 모니터링과 같은 많은 고급 기능들을 제공합니다. 이러한 기능들을 통해 지연시간, 상태 정보 혹은 다른 고려사항들을 최적화 하여 DNS 요청에 대해 응답할 수 있습니다. 여러분들은 웹 어플리케이션의 성능을 향상시키고 사이트 중단을 막는데 이러한 기능들을 활용할 수 있습니다.

Amazon Route 53 은 셔플 샤딩과 애니캐스트 스트라이핑 기능을 사용하여, Route 53 이 DDoS 공격을 받더라도, 사용자가 여러분의 어플리케이션에 접근할 수 있게 해줍니다. 셔플샤딩이란 위임 집합(delegation set) 내의 각 네임 서버들에 대해 엣지 로케이션들과 인터넷 경로들을 묶은 집합을 대응시켜주는 기능입니다. 이를 통해 고객들 간 중첩 부분을 최소화하고, 보다 향상된 장애 대응력을 제공할 수 있게 됩니다. 만약, 위임 집합 내의 한 네임 서버가 가용하지 않게 되면, 최종 사용자가 재 요청할 때는 다른 엣지 로케이션의 다른 네임 서버로부터 응답을 받게 됩니다. 애니캐스트 스트라이핑 기능은 각 DNS 쿼리가 가장 최적화된 엣지 로케이션으로 부터 서비스 받을 수 있도록 해줍니다. 이 기능은 DNS 지연시간을 줄이고 부하를 분산하는 효과를 주며, 결과적으로 최종 사용자 입장에서는 좀 더 빠른 응답을 받을 수 있게 됩니다. 부가적으로 Amazon Route 53 은 DNS 쿼리의 소스나 볼륨 내 변칙적인 부분을 탐지하고, 신뢰할 수 있다고 알려진 사용자로부터의 요청을 우선 처리할 수 있습니다.

만약 여러분이 Amazon Route 53 의 많은 호스트 존(hosted zone)을 가지고 있다면, 각 도메인 별로 정식 네임 서버들의 집합을 제공하는 재사용 가능한 위임 집합을 생성할 수 있으며, 이를 통해 호스트 존을 쉽게 관리할 수 있게 됩니다. AWS 입장에서도 DDoS 공격이 발생했을 때, 재 사용된 해당 위임 집합에 연계된 호스트 존들 전체에 대해 한번에 완화 조치를 취할 수 있게 됩니다.

Amazon Route 53 이 최종 사용자들을 여러분의 어플리케이션으로 어떻게 인도하는지에 대한 자세한 내용은 Getting Started with Amazon Route 53[8]를 참고하시기 바랍니다. 위임 집합의 재사용과 관련된 내용은 Actions on Reusable Delegation Sets[9] 를 참고하십시오.

응용 계층 방어(BP1, BP2, BP6)

지금까지 본 문서에서 다루었던 내용은 인프라 계층의 DDoS 공격으로 인한 가용성 문제를 효과적으로 완화하기 위한 여러가지 기법들을 다루었습니다. 여러분의 어플리케이션이 응용 계층의 공격을 막기 위해서는 요청 증가로 인한 확장을 탐지하고 악의적인 요청을 막을 수 있도록 아키텍쳐를 구성하는 것이 필요합니다. 네트웍 기반 DDoS 완화 시스템들이 일반적으로 복합적인 응용 계층 공격을 완화하는 데 비효율적이기 때문에 이 내용은 중요한 고려사항이 됩니다.

악의적인 웹 요청을 탐지해서 막기 (BP1, BP2)

웹 어플리케이션 방화벽(WAF)은 어플리케이션 상의 취약점을 통한 공격을 방어하는 용도로 사용되곤 합니다. 주로 SQL 인젝션이나 '사이트 간 요청 위조'(cross-site request forgery)를 막는 데 사용됩니다. 여러분들은 WAF 를 이용하여 웹 응용 계층에 대한 DDoS 공격을 완화시키는 데 이용할 수 있습니다.

여러분들은 AWS 상에서 Amazon CloudFront 와 AWS WAF 를 이용하여 이 같은 공격들로부터 어플리케이션을 방어할 수 있습니다. Amazon CloudFront 는 정적 컨텐츠를 캐싱하고 AWS 엣지 로케이션에서 제공하도록 하여, 원본(Origin)의 부하를 경감시키는데 도움을 줄 수 있습니다. 더불어서, Amazon CloudFront 는 slow-reading 이나 slow-writing 공격자(예, Slowloris)들로 부터의 연결을 자동으로 닫을 수 있습니다. Amazon CloudFront 지리적 위치 기반 제한 기능을 이용한다면, 여러분들의 컨텐츠에 접근하는 사용자들을 특정 지리적인 위치를 조건으로 제한시킬 수 있습니다. 이것을 통해, 여러분들은 최종 사용자로서 기대하지 않는 지리적인 위치로 부터의 공격을 차단할 수 있습니다.

HTTP flood 나 WordPress pingback flood 등, 나머지 유형의 공격들에 대응하기 위해 AWS WAF 를 이용하여 여러분들 만의 완화 기능을 구현할 수도 있습니다. 만일 차단하길 원하는 소스 IP 를 알고 있으면, WebACL 내부에 이와 관련된 차단 규칙을 생성할 수 있습니다. 해당 규칙에는 공격에 참여한 소스 IP 를 차단시키는 IP 주소 조건을 포함시키면 됩니다. 또한 URI, 쿼리 문자열, HTTP 메쏘드, 혹은 헤더 문자열 별로 차단하는 규칙들을 포함시킬 수도 있습니다. 이 경우 차단시킬 명확한 시그니쳐가 있는 경우 더욱 효과적입니다. 예를 들자면, WordPress pingback 공격 같은 경우 항상 User-Agent 헤더에 "WordPress"값을 가지고 있습니다.

DDoS 공격의 시그니쳐를 식별하거나 공격에 참여한 IP 주소를 정확히 파악하는 것은 부담스러운 작업이 될 수 있습니다. 때로는 웹서버 로그를 검색해서 관련

정보를 찾을 수도 있고, AWS WAF 콘솔에서 Amazon CloudFront 가 AWS WAF 로 전달 했던 샘플 요청들로부터 찾을 수 도 있습니다. 샘플 요청을 활용하면 응용 계층의 공격을 완화하기 위해 필요할 수 있는 규칙을 선별하는데 도움이 됩니다. 만일 랜덤 쿼리 문자열을 가진 요청들이 많다면, Amazon CloudFront 의 쿼리 문자열 포워딩 기능을 비 활성화 할 수도 있습니다. 이를 이용한다면, 여러분들의 원본(Origin)에 대한 cache-busting 공격을 완화하는데 도움이 될 수 있습니다.

일부 공격은 정상적인 최종 사용자의 트래픽처럼 보이게 끔 웹 트래픽을 위조하기도 합니다. 이런 유형의 공격을 완화하기 위해서는, AWS Lambda 를 통해 'Rate-based blacklisting' 기능을 구현할 수 있습니다. 'Rate-based blacklisting' 기능을 이용하면, 웹 어플리케이션이 얼마나 많은 요청을 처리할 것인지에 대한 임계치를 설정할 수 있습니다. 만약 봇이나 크롤러가 이 임계치를 넘었을 경우, AWS WAF 를 이용하여 자동으로 더이상 요청을 받지 않도록 차단시킬 수 있습니다.

Amazon CloudFront 배포에 대해 지리적으로 접근 제한을 거는 것과 관련된 자세한 내용은 Restricting the Geographic Distribution of Your Content[10]를 참고하시기 바랍니다.

AWS WAF 에 대해 더 자세한 내용은 Getting Started with AWS WAF[11] 과 Viewing a Sample of the Web Requests that CloudFront has Forwarded to AWS WAF[12]를 참고하시기 바랍니다.

AWS Lambda 를 이용한 Rate-based blacklisting 을 구성하는 방법은 How to Configure Rate-Based Blacklisting with AWS WAF and AWS Lambda[13]를 참조하시기 바랍니다.

공격을 흡수할 확장성 (BP6)

응용 계층 공격을 잘 다룰 수 있는 또 다른 방법은 확장성을 확보하는 것입니다. 웹 어플리케이션을 구성할 때 ELB 를 사용한다면, 갑자기 몰려든 트래픽이, 정상적인 사용자 폭증의 결과인지 응용 계층에 대한 DDoS 공격인지 간에, 몰려든 트래픽 폭증을 감당하기 위해, 자동으로 증설되었거나 사전에 준비되었던 다수의 Amazon EC2 인스턴스들에게 트래픽을 분배해 줄 수 있습니다. Amazon CloudWatch 경보 기능을 이용하면, 여러분이 정한 이벤트 상황에 맞추어 Amazon EC2 집단(fleet)의 규모를 자동으로 확장시킬 수 있는 Auto Scaling 기능을 활성화 할 수 있습니다. 이 기능을 통해 예상치 못한 규모의 요청을 다루어야 할 때, 어플리케이션의 가용성을 보호할 수 있게 됩니다. Amazon CloudFront 나 ELB 를 이용할 때, 배포지점이나

ELB 단에서 SSL negotiation 을 처리하도록 한다면, SSL 기반의 공격으로부터 여러분들의 인스턴스를 보호할 수 있게 됩니다.

Amazon CloudWatch 를 이용하여 Auto Scaling 기능을 구현하는 부분에 대한 좀더 자세한 내용은 Monitoring Your Auto Scaling Instances and Groups Using Amazon CloudWatch[14]를 참고하시기 바랍니다.

공격 지점 줄이기

AWS 상에서 아키텍쳐를 구성할 때, 유념해야 될 또 다른 중요한 고려사항은 공격자가 여러분들의 어플리케이션을 목표로 할 수 있는 가능성을 줄이기 위해 노력하는 것입니다. 예를 들어, 만약 어떤 리소스들은 전혀 최종 사용자들과 직접적인 상호 작용을 하지 않는다면, 해당 리소스들이 인터넷과의 직접적인 접근을 갖지 않도록 해야합니다. 이와 유사하게, 만일 최종 사용자나 외부 어플리케이션들과 여러분의 어플리케이션이 특정 포트 혹은 프로토콜 상에서는 상호 간 통신하지 않는다는 것을 파악했다면, 그 포트 또는 프로토콜을 통한 트래픽이 잘 차단되고 있는 지를 확인해야 할 것입니다. 이런 개념을 흔히 '공격 지점 줄이기'라고 말합니다. 이번 섹션에서는 공격 받을 수 있는 지점을 어떻게 줄일 수 있는지, 그리고 어플리케이션의 인터넷 접점을 어떻게 통제 할 수 있는지에 관한 모범 사례들을 다룰 것 입니다. 인터넷에 노출되지 않은 리소스들을 공격하는 것은 보다 어려우며, 이로 인해 공격자가 어플리케이션의 가용성을 훼손할 수 있는 옵션들이 제한될 수 있습니다.

AWS 리소스 감추기(BP1, BP4, BP5)

대부분의 어플리케이션에서는 AWS 리소스들이 인터넷 상에 완전히 노출될 필요가 없습니다. 예를 들어, ELB 뒷 단에 있는 Amazon EC2 인스턴스들은 인터넷 상에서 바로 접근할 필요가 없습니다. 이와 같은 시나리오에서는 ELB 만 Amazon EC2 인스턴스와 통신하고 최종 사용자가 ELB 상의 특정 TCP 포트들로 접근하도록 구성하기만 하면 됩니다. 이런 구성은 Amazon Virtual Private Cloud(VPC) 내부의 보안 그룹(Security Group)과 네트웍 접근제어 목록(NCAL)을 설정하면 됩니다. Amazon VPC 는 여러분들이 정의한 가상 네트웍상에서 AWS 리소스들을 기동할 수 있도록, AWS 클라우드 상의 논리적으로 분리된 영역을 제공합니다.

보안 그룹과 네트웍 ACL 은 VPC 내부의 AWS 리소스들에 대한 접근을 통제한다는 면에서 비슷합니다. 보안 그룹은 인스턴스 레벨에서 인바운드와 아웃바운드

트래픽을 제어할 수 있고, 네트웍 ACL 은 VPC 서브넷 레벨에서 유사한 기능을 제공합니다. 부가적으로, Amazon EC2 보안 그룹(SG) 규칙이나 네트웍 ACL 쪽으로의 인바운드 데이터 전송에 대해서는 과금되지 않습니다. 이것은 여러분들이 만든 보안 그룹 혹은 네트웍 ACL 에 의해 차단된 트래픽에 대한 요금이 과금되지 않음을 의미합니다.

보안 그룹 (BP5)

여러분들은 인스턴스를 시작할 때부터 보안 그룹을 설정할 수도 있고, 나중에 인스턴스에 보안 그룹을 연계할 수 있습니다. 인터넷으로부터 보안 그룹으로의 모든 트래픽들은 여러분들이 명시적으로 해당 트래픽을 허용해주지 않으면 기본적으로 차단됩니다. 예를 들어, 하나의 ELB 와 여러 대의 Amazon EC2 인스턴스들로 웹 어플리케이션을 구성했을 때, ELB 에 적용할 단일 보안 그룹('ELB 보안 그룹')을 적용할 지, 인스턴스 별로 여러 개의 서로 다른 보안 그룹('웹 어플리케이션 서버 보안 그룹')을 적용할 지는 결정해야 됩니다. 그런 다음, 인터넷에서 ELB 보안 그룹으로 흘러가는 것을 허용할 규칙을 만들고 ELB 보안 그룹에서 웹 어플리케이션 서버 보안 그룹으로 트래픽을 허용하는 규칙을 만들 수 있습니다. 결과적으로, 인터넷으로 부터의 트래픽은 여러분의 Amazon EC2 인스턴스와 직접 통신할 수 없게 되며, 이로 인해 공격자가 여러분의 어플리케이션에 대해 알아내는 것을 어렵게 만들 수 있습니다.

네트웍 접근제어 목록 (ACLs) (BP5)

네트웍 ACL 에서는 허용과 차단 규칙 둘다 설정할 수 있습니다. 이 기능은 여러분의 어플리케이션으로 특정 유형의 트래픽을 명시적으로 차단하려고 할 때, 유용합니다. 예를 들자면, 전체 서브넷에서 차단시켜야 되는 IP 주소 (CIDR 형식으로), 프로토콜과 목적지 포트 등을 설정할 수 있습니다. 만일 여러분의 어플리케이션이 TCP 만을 사용한다면, 모든 UDP 트래픽을 차단하도록 규칙을 설정하면 됩니다. 이 기능은 공격자의 소스 IP 나 다른 시그니쳐를 알고 있을 때, 공격을 완화하는 규칙을 설정할 수 있게 해 주기 때문에, DDoS 공격을 대응하는데 굉장히 유용합니다.

원본(Origin)을 보호하기 (BP1)

VPC 내부에 있는 원본(Origin)과 Amazon CloudFront 를 사용한다면, 반드시 AWS Lambda 함수를 이용하여 Amazon CloudFront 로부터 오는 트래픽만 허용하도록 구성된 보안 그룹의 소스 IP 조건을 자동으로 업데이트 하게 끔 해야 합니다. 이것을 통해 Amazon CloudFront 와 AWS WAF 가 놓치는 것이 없도록 하여, 원본(Origin)의 보안을 향상시킬 수 있게 됩니다.

자동으로 보안 그룹을 업데이트하도록 하여 여러분의 원본(Origin)을 방어하는 방법에 대해서는 How to Automatically Update Your Security Groups for Amazon CloudFront and AWS WAF by Using AWS Lambda[15]를 참고하시기 바랍니다.

여러분의 원본(Origin)으로 요청을 보내는 주체가 오로지 Amazon CloudFront 배포지점이라는 것을 확인하고자 한다면, Amazon CloudFront 가 원본(Origin)으로 요청을 포워딩할 때, 'Edge-to-Origin' 요청 헤더의 값을 추가하거나 덮어쓰는 방법을 이용할 수 있습니다. 또한 Amazon CloudFront 로부터 전송된 요청을 검증하는데 '*X-Shared-Secret' 헤더를 이용할 수도 있습니다.*

'*X-Shared-Secret* header'를 이용하여 원본(Origin)을 보호하는 방법에 대한 자세한 내용은 Forwarding Custom Headers to Your Origin[16]를 참고하시기 바랍니다.

API 엔드 포인트를 보호하기 (BP4)

통상적으로 하나의 API 를 공개적으로 오픈하게 되면 API 의 앞 단은 DDoS 공격의 목표가 될 수 있습니다. Amazon API Gateway 는 Amazon EC2, AWS Lambda 상에서 운영되는 어플리케이션이나 웹 어플리케이션에 대한 관문으로서 동작하는 하나의 API 를 만들 수 있는 완전 관리형 서비스입니다. Amazon API Gateway 를 이용하게 되면, API 의 앞 단에 별도의 서버를 구성할 필요가 없으며, 여러분의 어플리케이션 구성 요소들을 외부에서 잘 안보이도록 감출 수 있게 됩니다. 이를 통해 AWS 리소스들이 DDoS 공격의 대상이 될 수 있는 여지를 방지할 수 있습니다. Amazon API Gateway 는 Amazon CloudFront 와 연계되어 있으며, 여러분들의 서비스가 자체적으로 DDoS 대응력을 갖출 수 있게끔 해주는 이점을 줄 수 있습니다. REST API 에 있는 각 메쏘드 별로 'Burst rate limit' 설정을 하거나 표준을 설정해서 과다 트래픽으로부터 백 엔드 자원들을 보호할 수 있습니다.

Amazon API Gateway 를 이용하여 API 를 만드는 것에 대한 자세한 내용은 Getting Started with Amazon API Gateway[17]를 참고하시기 바랍니다.

운영 기법들

본 문서에 나와 있는 완화 기법들은 DDoS 공격에 대한 본질적인 대응을 위한 어플리케이션 아키텍쳐를 수립하는 내용들입니다. 많은 경우에, 여러분들의 어플리케이션에 대한 공격이 언제 발생 할 지를 알고 대응 조치를 준비하고 있는

것이 유용합니다. 또한 위협을 평가하고, 어플리케이션 아키텍쳐를 리뷰하기 위해 추가적인 리소스를 투입할 수도 있고 혹은 다른 지원을 요청하는 부분들이 필요해 질 수도 있습니다. 이번 섹션에서는 비 정상적인 행동들에 대한 가시성을 확보하고, 경보, 자동화, AWS 에 대한 추가 요청 등과 관련된 모범 사례들을 다루어 보겠습니다.

가시성

여러분의 어플리케이션의 정상 상태를 이해하는 것은 비 정상적인 상태에 빠졌을 때, 보다 빠른 대응을 취할 수 있도록 해줍니다. 주요 기준 항목이 예상 치에서 상당한 벗어났을 때, 어플리케이션의 가용성에 대한 공격 시도라고 간주 할 수 있는 지표가 됩니다. Amazon CloudWatch 를 통해, AWS 상에서 운영되고 있는 어플리케이션들을 모니터링 할 수 있습니다. CloudWatch 항목(Metric)들을 취합하고 추적하거나, 로그파일을 취합하고 모니터링 한다든지, 경보를 설정하고 AWS 리소스에 대한 변화가 생겼을 때 자동으로 대응하는 등의 일을 수행할 수 있게 해줍니다. DDoS 공격을 탐지하고 대응하는데 주로 활용되는 Amazon CloudWatch 항목에 대한 자세한 내용은 아래 테이블 3 을 참고하시기 바랍니다.

토픽	항목	설명
Auto Scaling	GroupMaxSize	Auto Scaling 그룹의 최대 규모
Amazon CloudFront	Requests	HTTP/S 요청의 개수
Amazon CloudFront	TotalErrorRate	모든 요청들 중 HTTP 상태 코드 4xx 혹은 5xx 인 경우의 비율
Amazon EC2	CPUUtilization	현재 사용중인 EC2 의 CPU 사용률
Amazon EC2	NetworkIn	인스턴스에 붙은 네트웍 인터페이스가 전달 받은 바이트 수
ELB	SurgeQueueLength	백엔드 인스턴스와 연결되기를 기다리고 있는, 로드 밸런서에 의해 큐에 들어간 요청의 개수.
ELB	UnHealthyHostCount	각 가용 영역 내 비정상 인스턴스의 개수
ELB	RequestCount	백 앤드 인스턴스로 잘 전달시킨 요청의 개수
ELB	Latency	요청이 로드밸런서를 떠난 이후 응답을 받을 때 까지 걸린 시간 (초 단위)

ELB	HTTPCode_ELB_4xx HTTPCode_ELB_5xx	로드밸런서가 발생시킨 HTTP 4xx 또는 5xx 에러 코드의 개수
ELB	BackendConnectionErrors	성공하지 못한 커넥션의 개수
ELB	SpilloverCount	큐가 꽉 차서 거부된 요청의 개수
Amazon Route 53	HealthCheckStatus	헬스 체크 엔드포인트의 상태

테이블 3: 추천할 만한 Amazon CloudWatch 항목(Metric)들

그림 5 에 나온 DDoS 대응 레퍼런스 아키텍쳐에 따른 어플리케이션이라면 공통 인프라 계층 공격들이 어플리케이션에 전송되기 전에 차단될 것입니다. 그렇기 때문에 이들 공격들은 Amazon CloudWatch 항목에 나오지 않습니다.

응용 계층 공격은 많은 항목(Metric)들의 수치를 향상시킬 것입니다. 예를 들어, HTTP flood 는 Amazon CloudFront, ELB, Amazon EC2 항목들 중에 요청과 CPU, 네트웍 사용률(Network Utilization) 항목을 올리게 됩니다. 만일 뒷 단 인스턴스들이 과다 요청들을 처리할 수 없게 되면, Amazon CloudFront 의 TotalErrorRate, ELB 상의 SurgeQueueLength, UnHealthyHostCount, Latency, BackendConnectionErrors, SpilloverCount, HTTPCode 같은 항목들의 수치가 올라가게 됩니다. 보통 이런 경우, 최종 사용자에게 정상적으로 서비스할 수 없을 정도의 수준까지 HTTP 요청의 규모가 줄어들게 됩니다. 여러분들은 어플리케이션의 뒷 단을 확장하거나 또는 본 문서의 초반에 설명했던 대로, AWS WAF 를 통해 과다 트래픽을 차단하는 방식으로 이와 같은 상황을 해결할 수 있습니다.

어플리케이션에 대한 DDoS 공격을 탐지하는 용도로 Amazon CloudWatch 를 사용하는 방법에 대한 자세한 내용은 Getting Started with Amazon CloudWatch[18]를 참고하시기 바랍니다.

어플리케이션에 대한 트래픽의 가시성을 확보하는 또다른 방법은 VPC Flow logs 를 사용하는 것입니다. 전통적인 네트웍 환경에서, 여러분들은 연결 문제나 보안 이슈를 해결하려고, 혹은 네트웍 접근제어 규칙이 예상대로 동작하는지를 확인하기 위해 네트웍 플로우 로그를 사용해 보셨을 겁니다. VPC Flow Logs 를 이용한다면, VPC 내의 네트웍 인터페이스들을 통해 주고받는 IP 트래픽에 대한 정보를 얻을 수 있습니다.

각 Flow logs 레코드는 소스 및 목적지 IP 주소, 소스 및 목적지 포트, 프로토콜, 수집 기간 동안 전송된 패킷의 수와 전송 바이트 수 정보를 포함합니다. 이 정보는

네트웍 트래픽의 이상 징후를 분석하고 특정 공격 방식을 파악하는데 도움을 줍니다. 예를 들어, 대부분의 UDP 반사(reflection)공격의 경우 특정 소스 포트를 가지고 있습니다(예, DNS 반사를 위한 53 번 소스 포트). 이것은 Flow logs 상에서 식별할 수 있는 명백한 시그니쳐가 될 수 있습니다. 이에 대한 대응으로 인스턴스 레벨에서 특정 소스 포트를 차단하거나, 필요하지 않는 프로토콜 전체를 네트웍 ACL 규칙으로 차단할 수 있습니다.

네트웍 이상 징후나 DDoS 공격 유형들을 파악하는데 VPC Flow Logs 를 사용하는 좀더 자세한 내용은 VPC Flow Logs[19] 와 VPC Flow Logs – Log and View Network Traffic Flows[20] 를 참고하시기 바랍니다.

지원 내역

실제 이벤트가 벌어지기 전에, DDoS 공격에 대한 계획을 세우는 것은 매우 중요합니다. 본 문서에서 말씀드리는 모범사례들은 선제적인 대책을 세우기 위한 목적이며, DDoS 공격을 받을 수 있는 어플리케이션을 오픈하기 전에 구현되어져야 합니다. 여러분들을 담당하는 영업팀에서는 여러분의 적용 케이스나 어플리케이션을 리뷰할 때 도움을 드릴 수 있고, 여러분의 특정 질문이나 요건에 대해서도 지원해 드릴 수 있습니다.

때로는 DDoS 공격을 받는 동안, 추가적인 지원을 위해 AWS 에 연락을 취하는 것이 도움이 될 수 있습니다. 여러분의 케이스에 대해 신속하게 회신을 받거나, 도와 줄 수 있는 전문가와 연결해 드릴 수 있습니다. 비지니스 지원을 받고 있는 고객들께는 24 x 7 기반으로 이메일, 채팅 혹은 전화 등을 통해 클라우드 기술지원 엔지니어의 지원을 받을 수 있습니다.

AWS 상에서 운영중인 미션 크리티컬 워크로드에 대해서는, 엔터프라이즈 지원을 고려하셔야 합니다. 엔터프라이즈 지원의 경우, 여러분들의 케이스는 가장 높은 우선 순위를 부여 받고, 선임 클라우드 기술지원 엔지니어에게 배정됩니다. 부가적으로, 엔터프라이즈 지원의 경우, 여러분들의 대리인이자 전담 기술 지원 담당인 Technical Account Manager (TAM)가 배정됩니다. 엔터프라이즈 지원은 또한 예정된 이벤트, 제품 출시 또는 마이그레이션 같은 작업을 하는 동안 실시간 운영 지원을 포함한 인프라 이벤트 관리 서비스를 제공합니다.

여러분의 독특한 요건에 맞는 지원 프로그램을 선택할 수 있는 방법에 대한 내용은 Compare AWS Support Plans[21]를 참고하시기 바랍니다.

결론

본 문서에 기술된 모범사례들을 이용한다면, 여러가지 형태의 인프라 혹은 응용 계층 DDoS 공격에 대하여, 여러분들의 어플리케이션의 가용성을 보호하기 위한 DDoS 대응 아키텍쳐를 수립할 수 있게 됩니다. 이들 모범 사례를 기준으로 어플리케이션 환경에 적용된 정도에 따라 여러분들이 완화 시킬 수 있는 DDoS 공격의 유형, 기법, 규모의 정도가 달라질 수 있습니다. AWS 는 여러분들이 DDoS 공격에 대항해서 어플리케이션 가용성을 잘 보호할 수 있게 이들 모범사례들을 이용할 것을 권장 드립니다.

기여한 분들

본 문서를 작성하는데 기여한 분들은 다음과 같습니다:

- Andrew Kiggins, AWS Solutions Architect

참고문헌

[1] https://www.youtube.com/watch?v=OT2y3DzMEmQ

[2] https://www.youtube.com/watch?v=Ys0gG1koqJA

[3] https://aws.amazon.com/ec2/instance-types/

[4] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html

[5] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

[6] http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elb-getting-started.html

[7] http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/GettingStarted.html

[8] http://docs.aws.amazon.com/Route 53/latest/DeveloperGuide/getting-started.html

[9] http://docs.aws.amazon.com/Route 53/latest/APIReference/actions-on-reusable-delegation-sets.html

[10] http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/georestrictions.html

[11] http://docs.aws.amazon.com/waf/latest/developerguide/getting-started.html

[12] http://docs.aws.amazon.com/waf/latest/developerguide/web-acl-testing.html#web-acl-testing-view-sample

[13] https://blogs.aws.amazon.com/security/post/Tx1ZTM4DT0HRH0K/How-to-Configure-Rate-Based-Blacklisting-with-AWS-WAF-and-AWS-Lambda

[14] http://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-monitoring.html

[15] https://blogs.aws.amazon.com/security/post/Tx1LPI2H6Q6S5KC/How-to-Automatically-Update-Your-Security-Groups-for-Amazon-CloudFront-and-AWS-W

[16] http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/forward-custom-headers.html

[17] https://aws.amazon.com/api-gateway/getting-started/

[18] http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/GettingStarted.html

[19] http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html

[20] https://aws.amazon.com/blogs/aws/vpc-flow-logs-log-and-view-network-traffic-flows/

[21] https://aws.amazon.com/premiumsupport/compare-plans/