AKAMAI 백서

경계 중심의 보안 모델 뛰어넘기

실질적으로 리스크를 경감할 수 있는 종합 가이드

저자:

찰리 게로(Charlie Gero) – Akamai Technologies CTO
엔터프라이즈 및 고급 프로젝트 부문

Akamai

목차

네트워크 아키텍처의 간략한 역사

이제 인터넷의 기원이라 할 수 있는 ARPANET이 컴퓨터 4대를 연결해 구성된 게 거의 50년 전입니다. 기초적인 패킷 교환 방식 네트워크에서 전 세계에 위치한 자율 시스템들이 복잡하고 어지럽게 연결돼 필요한 데이터를 얻기 위해 연동되어 작동하는 시대로 진화함에 따라 이 기술을 활용하는 사이버 위협 역시 마찬가지로 진화했습니다. 지난 10년 동안 강력한 DDoS 공격, 수억 명에게 영향을 미치는 데이터 침해 사건, 민감한 정부 시스템에 대한 데이터 유출 행위, 랜섬웨어의 등장 등 다양한 일들을 겪었습니다.

하지만 이 기간 동안 벌어진 긍정적이거나 부정적인 변화들 중에서도 가장 일관적으로 일어난 일이 하나 있다면, 대부분의 기업들이 기본적인 허브 앤 스포크 네트워크 아키텍처를 활용해 왔다는 점이라 말할 수 있습니다.

과거에는 이 아키텍처가 적합했습니다. 오래전, 인터넷이 비즈니스와 코어 인프라로 북적이는 공간으로 발전하기 전까지는 기업이 워크로드를 저장하는 곳은 데이터 센터였습니다. 이러한 데이터 센터에는 작업을 행하는 데 필요한 중요 인프라와 애플리케이션이 저장됐습니다. 기업의 지사, 리테일 매장, 원격 위치 등이 온라인화함에 따라 중앙 집중식 애플리케이션에 접속해야 하는 상황이 되었습니다. 따라서 기업은 요구 사항을 반영하는 자사 네트워크를 구축했고, 모든 네트워킹이 핵심 데이터 센터를 백홀(backhaul)하는 방식을 사용했습니다. 즉, 데이터 센터는 모든 작업이 일어나는 중앙 위치였습니다.

시간이 지나면서 인터넷은 상업적으로 상당히 매력적인 공간으로 변모하기 시작했습니다. 1994년에 Netscape의 엔지니어들이 SSL을 개발하면서 온라인 상거래가 시작되었고, 직원들은 특정 기업용 서비스를 인터넷으로 제공받기를 요구했습니다. 이러한 요청을 서비스하는 복잡한 글로벌 네트워크를 구축하는 과정에 있던 기업과 통신사들은 자신들이 가장 잘 아는 일을 했습니다. 즉, 이러한 서비스를 내부 애플리케이션을 호스팅하는 데이터 센터에 함께 배포하고 인터넷 링크를 구매해 해당 데이터센터와 연결함으로써 경로를 구축했습니다. 의도치 않게 다음과 같은 두 가지 결과를 초래했습니다. 즉, 외부의 소비자들이 진입할 수 있게 된 반면, 여러 지사에 분산되어 있던 내부 직원들은 밖으로 나갈 수 있었습니다. 상당한 기간 동안 허브 앤 스포크는 네트워크 아키텍처의 정상에 서 있었습니다.

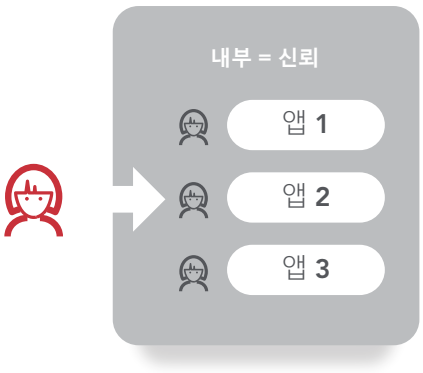
오랜 시간 동안 사이버 공격자들이 이 아키텍처를 악용함에 따라 데이터 센터 보안 스택이 완전히 새로운 산업으로서 탄생하게 되었습니다. 허브 앤 스포크 아키텍처는 인터넷 트래픽을 데이터 센터로 집중시키기 때문에, 고용량 라인을 수용하기 위해 강력한 대용량 박스가 개발되기 시작했습니다. 방화벽, 침입 탐지, 예방 등이 인바운드 트래픽에 사용되고 보안 웹 게이트웨이가 아웃바운드 방향에서 수용 가능한 사용 정책을 적용했습니다. 이렇게 중앙에 위치한 트래픽의 중심지에 보안 시스템을 배포하는 경향이 확산되면서, 네트워크 아키텍처로서 허브 앤 스포크의 확고한 위치는 더욱 견고해졌습니다. 그동안 성과 성벽(moat-and-castle) 방식의 보안 체계는 쓸만했고, 네트워크 경계 외부에 있는 사람들은 모두 위험하고 내부의 사람들은 모두 믿을만 하다는 네트워크 경계 개념이 당연한 것으로 받아들여졌습니다.

클라우드의 부상

ARPANET의 컴퓨터 4대가 온라인으로 연결되기 5년 전이었던 1964년에 가수 밥 딜런(Bob Dylan)은 '세상은 변해가네*(The Times They Are a-Changin)*'를 작곡했습니다. 지난 10년간 인터넷 세상을 완전히 지배한 클라우드를 표현하기에 너무나도 적절한 제목이 아닐까 합니다. SaaS가 등장하면서 기업들은 자사의 핵심 이니셔티브에 해당하지 않는 지원 시스템의 관리는 써드파티에 맡기는 것이 더 쉽고 안전하다는 사실을 깨닫기 시작했습니다. SaaS 서비스 업체가 클라우드에서, 자신들이 소유한 매니지드 서버에서 직접 CRM을 운영하면 훨씬 안전하고 성능도 뛰어나다면 기업이 내부에서 CRM을 직접 운영할 필요가 없습니다.

이메일, 액티브 디렉터리(AD), ID 등 너무 중요해서 마이그레이션이 불가능한 것으로 여겨졌던 요소들도 Office 365, G Suite 등이 등장함에 따라 글로벌 클라우드 인프라로 이동하고 있습니다.

또한, IaaS(Infrastructure as a Service)와 클라우드 컴퓨팅이 부상하면서 이러한 변화의 속도가 더욱 빨라졌습니다. Amazon AWS, Microsoft Azure, Google의 Compute 플랫폼은 SaaS를 통해 부수적인 지원을 오프로드하는 데 그치지 않고 기업이 물리적 인프라 자체를 사용량에 따라 지불하는 온디맨드 방식으로 전환하도록 유도합니다. 이에 따라 미션 크리티컬 핵심 비즈니스 애플리케이션 배포에 전례 없는 민첩성이 생겨났습니다.



한 마디로, 기업 애플리케이션이 이동한다는 뜻입니다. 하지만 움직이는 건 애플리케이션뿐이 아닙니다. 오늘날의 회사 직원들도 이동이 점점 많아지고 있습니다. 이제 많은 기업에서 직원들은 사무실 좌석에서 일하는 것만큼 커피숍, 자택, 공항 등에서 일하는 빈도가 잦아지고 있습니다.

결과적으로, 네트워크 경계란 더 이상 존재하지 않습니다. 최소한 인식할 수 있는 모습으로는 보이지 않습니다. 기업의 직원과 애플리케이션은 이전의 성과 성벽 방식이었다면 내부에 존재하는 만큼 외부에도 존재하는 경우가 많았을 것입니다. 또한, 정교한 위협과 멀웨어가 지속적으로 공격함에 따라, 기업은 의도치 않게 악의적 공격자가 보안 경계의 내부에서 기업의 가장 중요한 자산에 완전히 접근하도록 허용할 가능성도 상당히 높습니다.

20년 전까지만 해도 현대 세계에 가장 적합한 것으로 여겨졌던 보안 및 접속 방식을 지금 활용하는 것은 가장 바람직하지 못한 최악의 행동이라 해도 과언이 아닌 상황이 되었습니다. Forrester Research는 '*Future-Proof Your Digital Business With Zero Trust Security*'에서 다음과 같이 말합니다.

> **데이터 경제로 인해 오늘날의 경계 기반 보안은 쓸모가 없어졌습니다. 비즈니스가 정보와 인사이트를 복잡한 비즈니스 에코시스템 전체에서 수익화함에 따라 기업 경계에 대한 아이디어는 특이해질 수밖에 없고, 위험해지기도 합니다.**

이건 그저 이론이 아닙니다. 최근 5년간 일어났던 막대한 데이터 유출 사건에서도 명백하게 알 수 있었듯이, 데이터 유출 사건의 대부분은 네트워크 경계 내부에서 누군가가 신뢰 관계를 악용해 저지른 행위로 인한 것이었습니다.

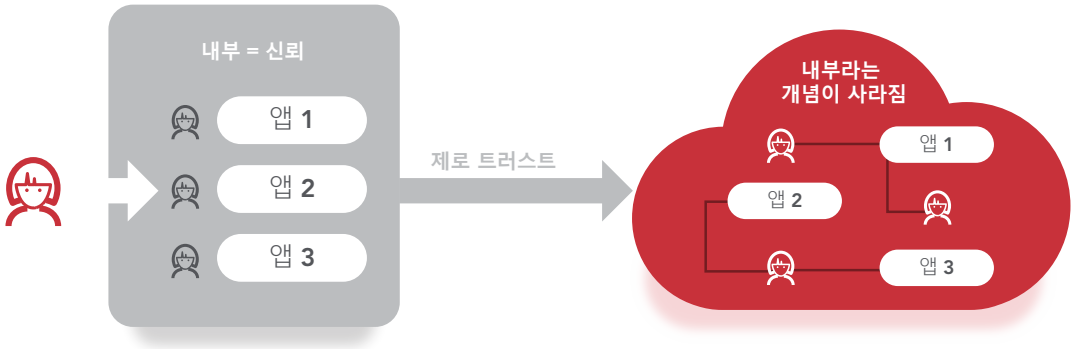
설상가상으로 이 문제는 더욱 커졌습니다. 네트워크 경계 내부에 배포하도록 설계된 애플리케이션의 보안 프로필이 최악인 경우가 상당히 많았습니다. 기업 시스템에 접근하는 사람은 오직 좋은 의도를 가진 인증된 직원들이라고 당연하게 생각하는 10년 전 개발자라면, 해커 중 상당수가 인터넷 기반 애플리케이션을 악용해 공격한다는 사실을 알고 있는 오늘날의 개발자만큼 방어적으로 시스템을 개발했을까요?

그럼 대체 어떻게 해야 할까요?

제로 트러스트

약 5년 전에 이 분야의 사고 리더이자 Forrester 애널리스트였던 존 킨더바그(John Kindervag)는 '제로 트러스트'라고 직접 이름을 붙인 솔루션을 제안했습니다. 이 솔루션의 기본 개념은 상당히 단순하면서 강력했습니다. 즉, 신뢰란 위치의 문제가 아니고, 기업 방화벽 내부에 있다는 이유 하나만으로 누군가를 믿지 말아야 하며, 대신 매우 비관적인 방향으로 발상을 전환해 모든 컴퓨터, 사용자, 서버는 신뢰할 수 있다고 입증되기 전까지는 신뢰하지 않아야 한다는 생각이었습니다.

제로 트러스트를 위한 입증 방식은 강력한 인증과 권한이고, 신뢰가 구축되기 전까지는 어떠한 데이터 전송도 행하지 않아야 합니다. 또한, 분석·필터링·로깅을 통해 행동이 적절한지를 확인하고 감염의 징후를 끊임없이 확인해야 합니다.

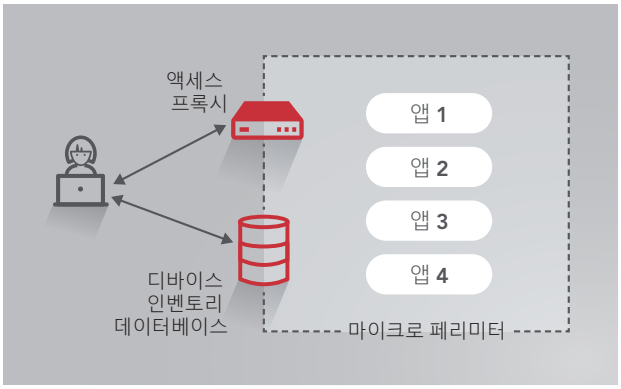


이렇게 보안 체계에 대한 근본적인 변화가 일어나면서 지난 10년 동안 등장했던 수많은 감염 사례를 극복할 수 있었습니다. 공격자는 성벽 내에 진입했다는 이유만으로 더 이상 기업 보안 경계의 취약점을 파고들어 민감한 데이터와 애플리케이션을 유출할 수 없게 되었습니다. 왜냐하면, 이제는 성벽 자체가 없기 때문입니다. 이제는 애플리케이션과 사용자만이 존재하며, 양쪽 모두 서로 인증하고 권한을 확인해야만 접속이 이루어질 수 있습니다.

그럼, 제로 트러스트는 어떻게 실현해야 할까요?

Google BeyondCorp™

몇 년 전에 Google은 제로 트러스트 접속에 관한 비전을 BeyondCorp라는 이름의 중요한 백서 여러 개를 통해 소개했습니다. BeyondCorp는 모든 애플리케이션의 제로 트러스트 관련 문제를 코드를 수정하지 않고 해결하고 애플리케이션이나 고객의 위치와 무관하게 간편한 방식으로 접속을 시도한다는 점에서 매력적입니다. 이해를 돕기 위해 다음과 같이 간략한 다이어그램을 준비했습니다.



이 다이어그램에 보이는 여러 항목은 다음과 같습니다.

- **사용자 및 노트북:** Google의 모델에서는 내부 애플리케이션에 접속해야 하는 사용자들은 인증서가 설치된 매니지드 노트북을 사용합니다. 또한, 이 노트북에서는 작은 소프트웨어 에이전트가 실행됩니다.
- **액세스 프록시:** 마이크로 페리미터 라인 위에 위치한 서버는 액세스 프록시라고 합니다. 이 서버는 마이크로 페리미터 내의 애플리케이션에 대한 보안 접속 정책을 실행합니다. 이 서버는 (서로 직접 통신하는 애플리케이션을 제외하면) 애플리케이션에 직접 도달할 수 있는 유일한 요소이며, 마이크로 페리미터의 비무장지대(DMZ)에 위치합니다.
- **디바이스 인벤토리 데이터베이스:** 다이어그램에 보이는 데이터베이스는 모든 Google 직원 노트북과 디바이스의 보안 체계에 대한 기록을 저장합니다. 이 데이터베이스에는 주기적으로 새로운 정보가 업데이트됩니다.
- **애플리케이션:** 직원들이 접속해야 할 필요가 있는 애플리케이션들입니다. Git, 이메일, 내부 기술 자료, 금융 애플리케이션, 데이터베이스 등이 이러한 애플리케이션에 속하며, 마이크로 페리미터 외부에서는 직접 접속할 수 없습니다.
- **마이크로 페리미터:** 마이크로 페리미터는 애플리케이션과 나머지 외부 사이를 가르는 일종의 경계망 역할을 합니다. 액세스 프록시는 이 사이를 연결하는 유일한 서버입니다. 또한, 이 마이크로 페리미터는 물리적인 데이터 센터에 위치할 필요가 없습니다. GCP나 AWS와 같은 클라우드 컴퓨팅 환경에서 마이크로 페리미터를 형성해도 마찬가지로 유효합니다.

BeyondCorp는 운영 면에서도 매우 간편하고 강력합니다. Google 직원의 컴퓨터에서 실행되는 소프트웨어 에이전트는 디바이스 인벤토리 데이터베이스에 접속해 디바이스의 현황을 평가합니다. 어떤 운영 체제가 설치되었는지, 어떤 패치가 있는지, 애플리케이션 상태, AV 등의 속성을 자세히 설명합니다.

디바이스 인벤토리 데이터베이스는 이 정보를 가지고 Trust Inferrer(위 그림에는 없음)라는 이름의 구성 요소의 지원을 받아 노트북의 평가 결과 또는 점수를 생성합니다. Google이 실제로 구현한 경우를 보면, 데이터는 사용자 디바이스의 소프트웨어 에이전트에 국한되지 않습니다. 데이터는 또한, 네트워크, 라우터, 방화벽 로그 등에서 실행되는 취약점 스캐너에서 올 수도 있습니다. 본질적으로, 디바이스 인벤토리 데이터베이스는 최대한 많은 정보를 지속적으로 수집해 디바이스 자체의 보안 체계를 아웃오브밴드로 판단합니다.

디바이스상이나 외부 상태에 변화가 일어나면 평가 결과나 점수에 영향을 미칩니다. 예를 들어, 보안 취약점이 특정 OS에서 발견될 경우, Google은 쉽고 간편하게 디바이스 인벤토리 데이터베이스가 해당 릴리스를 실행하는 모든 컴퓨터의 점수를 낮추도록 지시합니다.

디바이스 사용자가 애플리케이션에 접속을 시도하면 액세스 프록시로 연결됩니다. 액세스 프록시에는 각 애플리케이션에 대해 정의된 보안 접속 정책이 있으며, 이 정책은 리소스에 접근할 수 있는 사용자를 판단하고 사용 중인 디바이스에 요구되는 최소한의 보안 체계를 제시합니다.

실제로 BeyondCorp는 사용자 신원 및 디바이스의 상태를 애플리케이션 중심으로 함께 사용하기 때문에 매우 강력합니다. 예를 들어, Git에는 모든 패치를 적용한 최신 버전의 현대적인 OS를 실행하는 노트북을 사용하는 개발자 그룹의 사용자만이 접속할 수 있습니다. 회사 디렉터리에는 훨씬 적은 제한 정책이 정의되었을 가능성이 있으며, 이 경우 유효한 계정이 있고 Windows XP 이상의 운영 체제를 실행하는 컴퓨터가 있는 사람이라면 누구나 데이터에 접근할 수 있습니다. 재무 부서는 민감한 회계 정보에 접근할 수 있다고 말하는 것만으로는 부족합니다. 정책을 통해 사용하는 컴퓨터의 안전성 역시 보장해야 합니다.

액세스 프록시에 접속하면 사용자 인증이 시작됩니다. 이는 클라이언트 측 노트북의 인증서를 사용하는 상호 인증 세션을 통해 진행합니다. 성공적으로 인증되면 액세스 프록시는 접속하려는 사람과 사용하는 컴퓨터를 모두 알게 됩니다.

이 정보를 위에 정의된 정책에 대해 디바이스 인벤토리 데이터베이스와 연동해 실행하면 액세스 프록시가 애플리케이션에 대한 트래픽을 허용할지 또는 거부할지를 결정할 수 있는 상황이 됩니다.

BeyondCorp의 특성 중 마지막으로 주목할만한 면은 상당한 포부가 있다는 점입니다. 제로 트러스트는 기본적으로 전략에 속하며, BeyondCorp는 이를 실현하기 위한 하나의 수단일 뿐입니다. 현재 Google은 클라우드 배포 외부에서는 소프트웨어를 제공하지 않으며, Google과 동일한 수준의 제로 트러스트를 실현하는 방법에 관한 사고 리더십을 커뮤니티에 제공하는 이타적인 모습을 보여주고 있습니다.

Akamai 클라우드 보안 서비스

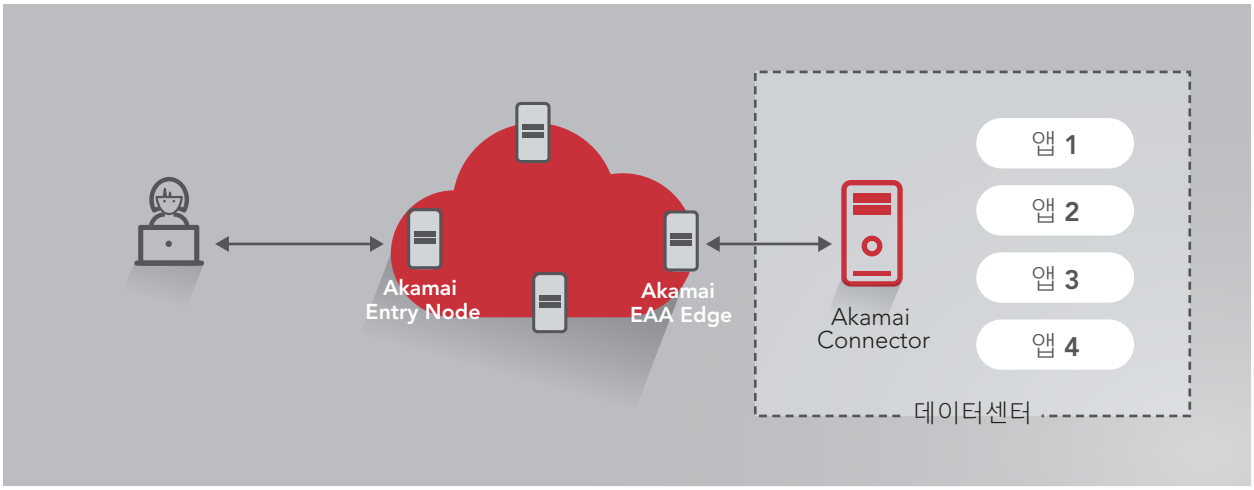
기본적인 BeyondCorp 모델에서 액세스 프록시는 DMZ 내부에 위치한 서버로서 마이크로 페리미터 내의 서비스에 대한 접속을 허용하거나 거부합니다. 이러한 서버가 처리해야 하는 작업량은 상당히 부담스러울 수밖에 없습니다.

최소한 액세스 프록시는 클라이언트 측 TLS 접속 종료, 인증 실행, 디바이스 인벤토리 데이터베이스 쿼리, 정책 적용, 마이크로 페리미터 내에서 서비스에 대한 연결 프록시 처리 등 다양한 작업을 책임지며 이 중 상당수는 암호화가 필요합니다. 이미 작업량이 상당히 부담스러운 상황에 WAF(Web Application Firewall) 및 행동 분석 등의 서비스를 추가하면 부담이 정말로 심각해질 수 있습니다.

게다가 이는 CPU에만 해당되는 문제입니다. 캐싱과 같은 특정 기능은 인터넷 업링크의 데이터 센터 측에서 작업 가능하긴 하지만, 요청하는 사용자와 가장 가깝게 위치한 업링크의 인터넷 측에서 작업하면 사실상 대역폭이 무한하므로 훨씬 장점이 많습니다.

표준 BeyondCorp 액세스 프록시 아키텍처는 보안 측면에서 비약적인 발전을 이루긴 했지만, 특성상 기업 DMZ 내에서 이를 완전히 실행하면 공격 트래픽을 흡수하고, 캐싱을 위해 대역폭을 무한으로 제공하고, 필요할 때 리소스를 자동 확장하는 등의 기능이 제약을 받을 수 있습니다.

Akamai는 클라우드 중심 기업으로서 기존 액세스 프록시에 약간의 아키텍처 수정을 설계했습니다. 이를 통해 액세스 프록시는 클라우드 내에 위치하고, 온디맨드로 확장 가능하며, CPU에 많은 부담을 주는 리소스를 기업 장비가 아닌 Akamai 네트워크에서 실행하고, 공격을 흡수하며, 사용자와 가장 가까운 위치에 캐시된 콘텐츠를 전송할 수 있습니다. Akamai에서는 이를 EAA(Enterprise Application Access)라 부르며, EAA의 구조는 다음과 같습니다.



이 아키텍처에서는 BeyondCorp 모델과 완전히 동일한 방식으로 애플리케이션을 마이크로 페리미터에 마이그레이션합니다. 하지만, DMZ에 액세스 프록시를 배치하는 대신, Akamai Connector라는 VM을 기업 마이크로 페리미터 내에서 실행합니다. Akamai Connector는 DMZ 내부에 위치할 필요가 없고, 위치해서도 안 됩니다. Akamai Connector의 주소는 프라이빗 IP 공간에 있어야 하고 인터넷에서 직접 접근하기가 불가능해야 합니다. 사실, 마이크로 페리미터 내에 배치할 다른 모든 애플리케이션과 완전히 동일한 것으로 보여야 합니다.

Akamai Connector가 실행되면, 즉시 Akamai 클라우드에 대한 암호화된 접속을 수립합니다. Akamai에 접속되면, Akamai 서버에서 설정을 다운로드하고 서비스 접속 준비를 마칩니다.

내부 애플리케이션 사용자가 마이크로 페리미터 내의 서비스에 접속을 시도하면, DNS CNAME을 통해 Akamai로 이동되며, Akamai Entry Node로 연결됩니다. Akamai Entry Node는 사용자와 가장 가까운 위치에 있는 노드에 있으며, Akamai는 WAF, 봇 탐지, 행동 분석, 캐싱 등의 작업을 적용할 수 있습니다. 이로써 업계 최고의 성능을 발휘하면서 동시에 잠재적인 공격자와 기업의 물리적 위치, 애플리케이션, 데이터 간의 거리를 최대한 멀리 유지할 수 있습니다.

사용자가 모든 확인 절차를 통과하면 Akamai의 고급 네트워크 오버레이를 통해 Akamai EAA Edge로 라우팅되며, 여기에서 일반적인 인증, MFA(멀티팩터 인증), SSO(Single Sign On), 디바이스 ID 기능 등을 수행합니다. 사용자와 컴퓨터가 인증되면, 클라이언트에서의 접속이 Akamai Connector에서의 아웃바운드 접속과 결합됩니다. 사용자 세션에서 오는 트래픽은 이렇게 결합된 프록시 연결을 통해 Akamai Connector로 이동한 다음 요청된 서비스 애플리케이션으로 접속됩니다. 이 시점에서 완전한 데이터 경로가 수립됩니다.

이러한 접속 방법에는 분명하고 현저한 장점이 있습니다. 성능과 보안에 가장 민감한 활동들은 사용자와 가장 가까운 위치의 네트워크 엣지에서 수행되는데, 이를 위해 Akamai는 전세계적으로 20만 대 이상의 서버를 구축했습니다. 마이크로 페리미터에 대한 민감한 유입 경로는 역방향 애플리케이션 터널을 통해 만들어지므로 경계의 IP를 파악할 수 없으며 대형 공격의 리스크가 줄어듭니다.

지금까지는 이 패러다임을 사용한다고 해서 지금까지 학습한 개념이 근본적으로 바뀌지 않았습니다. 효율성만 높아질 뿐입니다. 단계적 전환에 관해 논하면서 이러한 접근 방식을 더욱 빠르고 안전하다고 생각되는 방식으로 참조할 것이지만, 일반적인 개념은 가장 기본적인 BeyondCorp 액세스 프록시 접근 방식에도 적용됩니다.

애플리케이션 접속 vs. 네트워크 접속

이 백서를 읽다 보면 이러한 방식이 VPN과 같다는 생각이 드실 수도 있겠지만, 이는 상당히 큰 오해입니다. VPN은 네트워크 수준의 접속을 제공하며, 기술적인 토대로 인해 보안 및 성능이 관리 역량 및 간편함과 반비례하게 됩니다.

VPN 설정을 간편하게 하려면 다른 많은 기업들처럼 로그인한 사용자에게 전체 네트워크에 대한 IP 수준의 접속을 허용해야 합니다. 이것이 얼마나 위험한 일인지 이제 우리는 잘 알고 있습니다. 콜 센터 직원에게 소스 코드 리포지토리에 IP 수준의 접속을 허용해야 할 이유가 있을까요? 아니면 기업 요금 청구 시스템을 사용하는 계약직 직원이 신용카드 처리 터미널에 접속해야 할 이유가 있을까요? 접속 권한은 역할을 수행하는 데 필요한 만큼만 부여해야 합니다.

이를 바로잡으려면, VLAN을 통해 방화벽 뒤의 개별적인 세그먼트로 애플리케이션을 파티셔닝하기 시작하고, VPN 애그리게이터에서 개인 사용자나 그룹에 구식 IP 범위 기반 규칙을 실행해야 합니다. 하지만 이 작업은 상당히 취약하며 오류가 발생하기 쉽습니다. 관리자들은 최악의 상황에 연결이 끊어지는 현상을 자주 경험합니다. 어쩌면 누군가가 유지보수 작업 중에 컴퓨터를 새로운 랙으로 옮기거나 새로운 범위로 IP를 재설정해야 할 수도 있습니다. 갑자기 사용자의 접속이 끊기면서 지원 요청 전화가 북새통이 됩니다. 아니면 소프트웨어 업그레이드 중에 애플리케이션의 아키텍처 변경이 일어나면서 사용자가 워크플로우의 일환으로 다른 컴퓨터로 리디렉션되는데, 하필이면 방화벽 규칙이 업데이트되지 않아 일부 사용자나 그룹이 해당 컴퓨터에 접속할 수 없는 상황이 발생할 수 있습니다. 이러한 아키텍처는 애플리케이션 소유자, 네트워크 관리자, 보안 그룹 간에 매우 높은 수준의 커뮤니케이션이 이루어져야만 다운타임을 완전히 방지할 수 있습니다.

우리는 이러한 조율이 제대로 이루어지지 않을 경우 발생하는 사태를 오랜 기간 동안 많이 봤습니다. 관리자는 평소에는 모범 사례를 따르려 하지만, 절박한 상황에서는 문제가 많은 IP ANY/ANY ALLOW 규칙을 적용해 영향을 받는 내재한 문제를 진단하고 고칠 때까지 사용자가 모든 곳에 접속할 수 있도록 허용하기도 합니다. 하지만 현실적으로는 정석대로 과거의 취약점을 고칠 시간이 부족할 때가 많습니다. VPN 사용 시에는 제한 없는 수평적 접속의 보안상 단점을 극복하려면 상당한 복잡성과 운영상 오버헤드가 발생할 수밖에 없으며, 이러한 복잡성으로 인해 취약점이 생기고 임시 조치는 장기적으로 보안 체계에 악영향을 미칩니다.

성능에 대해서도 동일한 딜레마가 있습니다. 가장 단순한 형태의 VPN에서는 모든 트래픽이 데이터 센터로 직접 돌아갑니다. 여기에서 일어나는 헤어핀 효과(hairpin effect) 때문에 인터넷 자산과 SaaS에 대한 연결 속도가 극도로 느려질 수 있고, 데이터 센터 내에서 Facebook이나 YouTube 등 비즈니스와 무관한 트래픽에 대한 인터넷 업링크의 정체가 심해질 수 있습니다. 이미 오프 프레미스 상태인 사용자에게 일반적인 인터넷 접속 권한을 백홀(backhaul)해 줄 이유가 있을까요?

이 성능 부담을 극복하려면 관리자는 분할 터널을 배포해 VPN을 통과할 IP 범위와 인터넷으로 직접 송신될 IP 범위를 표시해야 합니다. 내부 경계가 단 한 개만 있을 때는 매우 효율적이고 간편합니다. 하지만 여러 데이터 센터와 가상 프라이빗 클라우드(VPC)를 IaaS 사업자에 추가하면 훨씬 복잡해지기 시작합니다. 관리자들은 VPN 애그리게이터를 모든 데이터 센터에 설치해야 할지와 멀티포인트 분할 터널을 효과적으로 관리하는 방법을 결정해야 합니다.

또한, 이 모든 솔루션들이 이론적으로 가능합니다. 이미 이들 중 몇 가지의 조합을 사용하고 계실지도 모르겠습니다. 문제는 이를 제대로 수행하고, 유지하고, 보안과 성능을 구현 수명 내내 계속 적절히 제공하기는 운영상 너무 복잡해 지속적으로 유지하기 어렵다는 점입니다. 많은 경우, 기업은 직원들이 애플리케이션에 접속할 수 있으니 모두 최적의 상태일 것이라고 자기 위안을 합니다. 위의 임시 조치 중 한 가지가 최악의 유출 사태를 일으키거나 가동 중단 또는 직원 생산성에 심각한 악영향을 미치는 성능 저하 현상을 유발하면 기업은 갑자기 예상치 못한 어려움에 빠지게 됩니다.

VPN에 가치가 없다는 의미는 아닙니다. 사실은 상당히 가치가 있으니까요. VPN은 여러 데이터 센터 인프라에 대한 사이트 간 접속에서 가장 빛을 발합니다. 단지 네트워크 수준의 접속은 사용자의 애플리케이션 접속에 사용하기에는 다소 적절한 패러다임이 아니라는 의미일 뿐입니다. 왜냐하면, 네트워크 수준의 접속은 간소함과 보안·성능 중 한쪽이 부자연스럽게 저하될 수밖에 없기 때문입니다.

BeyondCorp나 Akamai Enterprise Application Access와 같은 프록시 기반 접근 방식의 장점은 애플리케이션 수준 접속입니다. 애플리케이션 수준으로 접속할 수 있으면 성능과 보안을 복잡하지 않게 달성할 수 있습니다. 이는 설명하기도 정말 쉽기 때문에 말할 필요도 없이 명백한 사실입니다.

서로 인접한(예: 동일한 데이터 센터 또는 VPC에 호스팅된 경우) 모든 애플리케이션을 프라이빗 네트워크 IP 공간이나 제한적인 VLAN에 배치하고 해당 마이크로 페리미터의 DMZ에 접속을 제공하기만 하면 완료됩니다.

애플리케이션 소유자는 액세스 프록시에 보안 정책을 직접 설정해 접속할 수 있는 사람과 디바이스 특성을 정할 수 있습니다. 여기에서 중요한 점은 사용자의 위치를 가리지 않는다는 것입니다. 온프레미스와 오프프레미스 간의 구분이 없습니다. 사용자를 포함한 네트워크 경계가 없기 때문입니다. 커피숍에서 일하는 직원과 사무실의 직원이 완전히 동일하게 취급됩니다. 사용자가 인증되었는지와 사용하는 컴퓨터가 안전한지의 여부만이 중요합니다.

애플리케이션 수준 접속 기능 덕분에 배포와 사용이 쉬우면서도 성능을 이론상 동급 최고 수준으로 유지할 수 있습니다. 사용자는 호스팅되는 곳이나 사용하는 위치와 무관하게 인터넷 자체를 사용해 애플리케이션에 직접 접속할 수 있으며, 인터넷에서 경로에 없는 애그리게이터나 매개체를 거칠 필요 없이 패킷을 목적지로 바로 보낼 수 있습니다.

사실 애플리케이션 수준 접속을 사용하면 내부 네트워크가 단순한 게스트 Wi-Fi로 통합됩니다. 제로 트러스트가 정말로 효과를 발휘하려면, 내부 사용자와 외부 사용자를 완전히 동일하게 취급해야 합니다. 즉, 기본적으로 모두를 신뢰하지 말아야 합니다.

바람직한 최종 상태

애플리케이션 접속

최신 제로 트러스트 배포에서는 모든 애플리케이션이 위치와 목적에 따라 각자의 마이크로 페리미터로 분류되어야 합니다. 이러한 마이크로 페리미터는 프라이빗 IP 공간의 프라이빗 VLAN에 있어야 하며, 마주한 액세스 프록시를 제외하면 어떤 곳에서도 직접 접속이 불가능해야 합니다.

애플리케이션이 감염된 경우 수평 에스컬레이션 및 이동을 중단하기 위해 마이크로 페리미터 내에서 더욱 세부적으로 분류해야 할지는 각 기업의 판단에 맡깁니다. 이러한 행위는 이론적으로 가치가 있지만, 실제로는 제대로 분류하는 작업의 복잡성이 너무 심하기 때문에 보안이 강화된다고 해도 설득력이 부족합니다. 복잡한 애플리케이션이 경계 뒤에 가지고 있는 접점을 풀어내 본 사람들은 애플리케이션을 각각 분류하는 일이 얼마나 불안정한지 잘 알 것입니다. 하지만, 기업 환경이 이를 쉽고 유지 가능한 방식으로 허용한다면, 이를 도입하시기를 권장합니다.

모든 사용자는 온프레미스·오프프레미스와 무관하게 모든 애플리케이션을 Akamai Enterprise Application Access와 같이 표준 인증뿐이 아니라 MFA까지 수행하는 액세스 프록시를 통해 접속하도록 강제해야 합니다. 이에 더해, 애플리케이션 소유자가 디바이스 체계 자체에 정책을 적용하고 싶다면, 기업 리소스에 접근이 허용된 모든 디바이스의 보안 상태를 추적하는 강력한 디바이스 인벤토리 데이터베이스가 있어야 합니다. 컴퓨터에 인증서를 제공하고 보안상 필요할 때 인증서를 철회하려면 인증서 관리 및/또는 모바일 디바이스 관리(MDM)가 필요할 가능성이 높습니다.

또한, 저희는 제로 트러스트는 인증 및 권한 부여로 끝나지 않는다는 것을 굳게 믿습니다. 모던 스택은 또한, 액세스 프록시를 통해 지나가는 트래픽에 대해 어느 정도의 검사와 분석이 필요할 것입니다. 지속적이며 정교한 위협과 의도적으로 악의적인 행동을 하려는 사용자들을 막는 데 도움이 될 것입니다.

액세스 프록시 위에 레이어로 추가해야 할 중요한 보안 시스템은 WAF이며, 기업 내부 애플리케이션에 대해 사용자가 애플리케이션 수준 공격을 (의도적이든 의도적이지 않든) 실행하지 않도록 보장할 수 있습니다. 인간·봇 탐지와 같은 기타 고급 시스템은 API를 사용하지 않는 사이트에 활용해 멀웨어가 유효한 엔드포인트 뒤에서 정체를 숨기고 있지 않도록 지원합니다.

애플리케이션을 온라인화하고 액세스 프록시를 통해 접속 가능하도록 하면 DDoS 방지가 훨씬 더 중요해집니다. 마이크로 페리미터와 액세스 프록시에 대한 공격을 흡수할 수 있는 사업자와 협업해 트래픽이 부담스러울 때도 운영을 계속할 수 있어야 합니다.

마지막으로, 애플리케이션 성능을 업계 최고 수준으로 유지하고 사용자가 이러한 접속 방식의 변화를 받아들이는 데 그치지 않고 적극적으로 지지하게 하려면, 액세스 프록시에는 성능상 이득이 되는 네트워크를 사용해야 합니다. 특히, 기업은 CDN과 인터넷 라우팅 오버레이를 툴로 활용해 접속을 제공하는 데 그치지 않고 이전 방법보다 훨씬 뛰어난 성능도 제공해야 합니다.

인터넷 접속

Akamai Enterprise Application Access와 같은 솔루션을 악의적인 공격자로부터 기업 애플리케이션을 지키는 수단으로 생각하는 것은 바람직합니다. 그렇다면, 사용자가 멀웨어 감염을 당하는 바람에 악의적인 공격자가 되어버리는 일은 어떻게 막을 수 있을까요? 바로 이 부분에서 아웃바운드 트래픽에 대한 예방과 탐지가 중요해집니다.

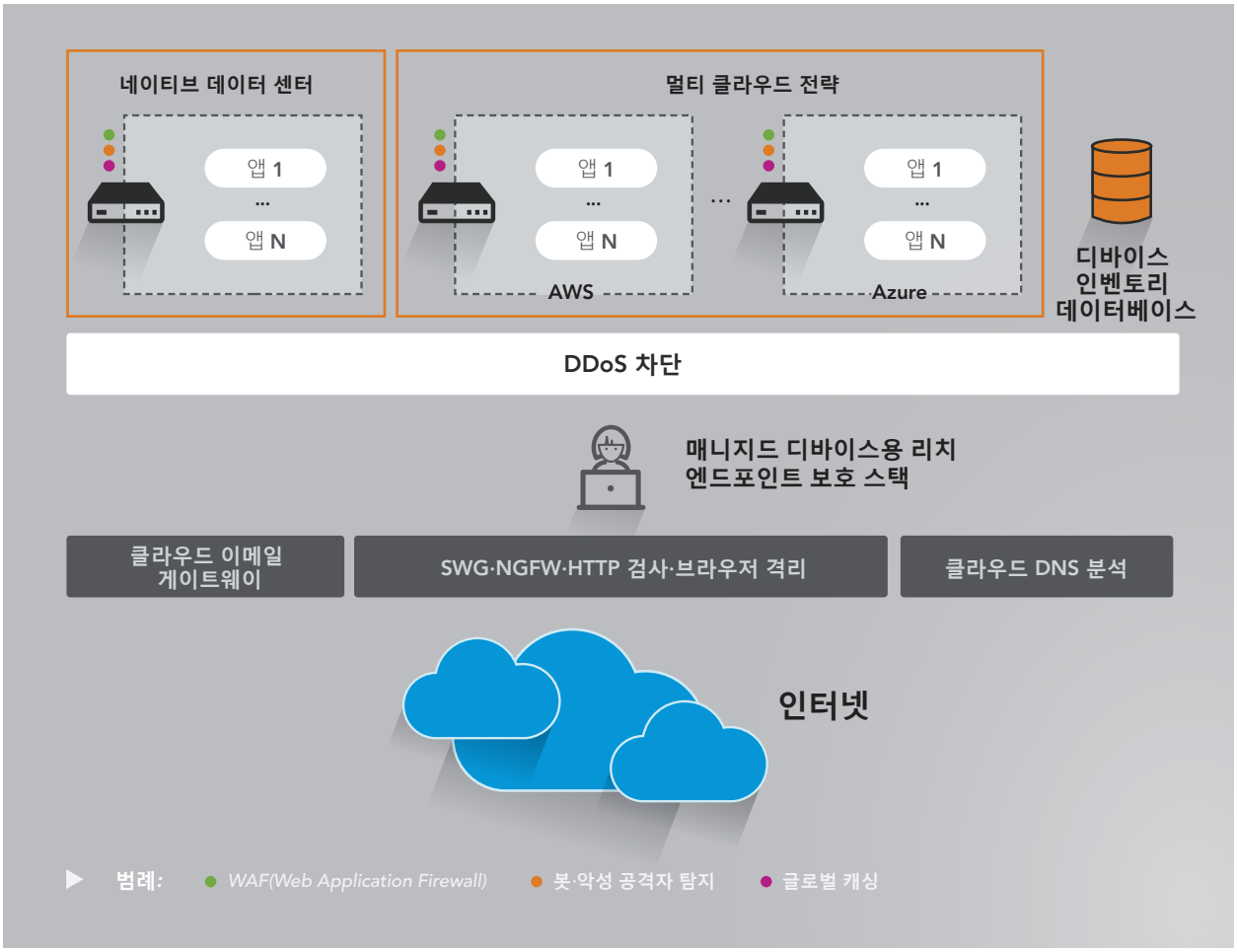
따라서 매니지드 디바이스에 대한 엔드포인트 보안은 미래에 더욱 중요해질 것이며, 제로데이 위협을 식별하는 능력에서 머신러닝 기반의 AV 솔루션이 기존 AV 스택을 앞지를 것으로 예상됩니다. 또한, 일부 기업은 애플리케이션 화이트리스트나 데스크톱 프로세스 격리 시스템은 예방보다 더욱 종합적으로 위협을 방지한다고 인지할 수도 있겠지만, 배포 유연성이 떨어져 실제로 활용하기에는 너무 어려울 가능성도 있습니다.

또한, 최종 사용자를 보호하기 위해 클라우드 기반 예방 스택을 배포해야 합니다. SWG나 NGFW와 같은 기존의 시스템들은 계속 중요하겠지만, 가장 리스크가 높은 사용 사례에 대처하는 포인트 솔루션의 지원을 받을 것입니다. 여기에는 브라우저 격리, 샌드박싱 및 웹 실행 파일 스캐닝 등의 작업과 피싱 및 멀웨어 첨부 파일을 탐지하는 이메일 게이트웨이가 포함됩니다.

예방도 중요하지만, 사용자에게 지속적으로 가해지는 위협에 관리자가 대응하는 데 가장 중요한 툴킷은 앞으로도 탐지 툴일 것이라 생각합니다. 이러한 툴킷 중 간편하게 배포할 수 있지만, 매우 효율적인 시스템으로는 DNS 방화벽이 있습니다. DNS는 정상적인 소프트웨어와 악의적 소프트웨어에서 모두 사용되지만, 감염을 탐지하는 훌륭한 신호기 역할을 하는 것은 물론, 이러한 솔루션을 쉽게 배포할 수 있기 때문에 더욱 가치가 높습니다.

아키텍처 시각화

제로 트러스트를 완전히 실현한 환경에서는 기업 사용자와 애플리케이션이 다음과 같은 모습일 것이라 예상합니다.



단계별 전환

지금까지 제로 트러스트를 상당히 포괄적으로 살펴보았으며, 완전한 제로 트러스트와 CARTA를 구현한 기업이 어떤 모습을 보여줘야 하는지도 알아봤습니다. 하지만, 보안을 쉽게 구현할 수 있어야 가장 확실히 구현할 수 있는 것처럼, 새로운 아키텍처의 배포 역시 간편하게 단계별로 전환할 수 있어야 합니다. 어느 정도의 규모가 있는 기업이라면 하루만에 이러한 비전을 따라 전체 네트워크를 전환하기는 어려울 것입니다. 제로 트러스트 자체는 전략이므로, 제로 트러스트의 배포도 마찬가지로 전략입니다.

완전한 제로 트러스트 아키텍처를 달성하기 위한 최고의 접근 방식은 관리 가능한 작은 배치로 Enterprise Application Access 모델로 전환을 시작하는 것입니다. 배치 크기는 전담하는 리소스 개수에 따라 달라지며, 기업에 따라 한 번에 하나의 배치밖에 전담하지 못할 수도 있습니다. 이와 무관하게, 전환되는 각 애플리케이션은 제로 트러스트를 완료하기 위한 과정으로써 여러 단계를 거칩니다. 각 단계에서 애플리케이션이 제대로 작동하는지 확인하고 인증이 예상대로 진행되는지 확인합니다.

2018년 초 현재, Enterprise Application Access는 클라이언트를 설치할 필요 없이 HTTP, VNC RDP, SSH 애플리케이션을 지원합니다. 클라이언트의 사용을 통한 보안 체계 평가와 추가적인 프로토콜은 2018년 내로 예정되어 있습니다. 이에 따라 Akamai는 현재 EAA 기능과 연동하기 위해 웹 기반 애플리케이션의 단계적 전환을 시차를 두고 먼저 진행하려 합니다.

프리스테이징 가정

기업 네트워크의 프리스테이징 네트워크에 대해서는 다음과 같은 가정을 합니다.

- 기업의 현재 경계에 있는 애플리케이션 및 사용자는 IP를 통해 서로 직접 연결될 수 있습니다
- Akamai EAA Connector를 현재 경계에 설치했습니다.
- 기업 애플리케이션을 최종적으로 배치하기 위해 방화벽으로 나뉜 VLAN을 생성했습니다.
- Akamai EAA Connector를 방화벽으로 나뉜 VLAN에 설치했습니다.
- 기업의 오프프레미스 엔터프라이즈 사용자들은 아직 전환되지 않은 애플리케이션에 연결하기 위해 단계적 전환 중에 VPN을 계속 설치할 것입니다.
- 매니지드 디바이스 인식이 필요한 경우, DMD와 같은 아웃오브밴드 방식을 통해 사용자의 컴퓨터에 인증서가 설치될 것입니다.

이러한 전제 조건이 만족되면, 각 애플리케이션이 전환을 완료하는 과정에서 거쳐야 할 단계를 논의할 수 있습니다. 이러한 과정들은 가장 높은 수준으로 세분화되면, 프로세스에 점점 적응함에 따라 세분화한 부분 중 일부를 결합하고 싶을 수도 있습니다.

사용자 그루핑

애플리케이션 스테이징 프로세스의 일부로서 이러한 기능을 선택된 사용자 그룹에 대해 단계적으로 전환하는 작업이 수반됩니다. 각 애플리케이션에 대해 3가지의 일반적인 사용자 그룹을 적용하기를 권장합니다.

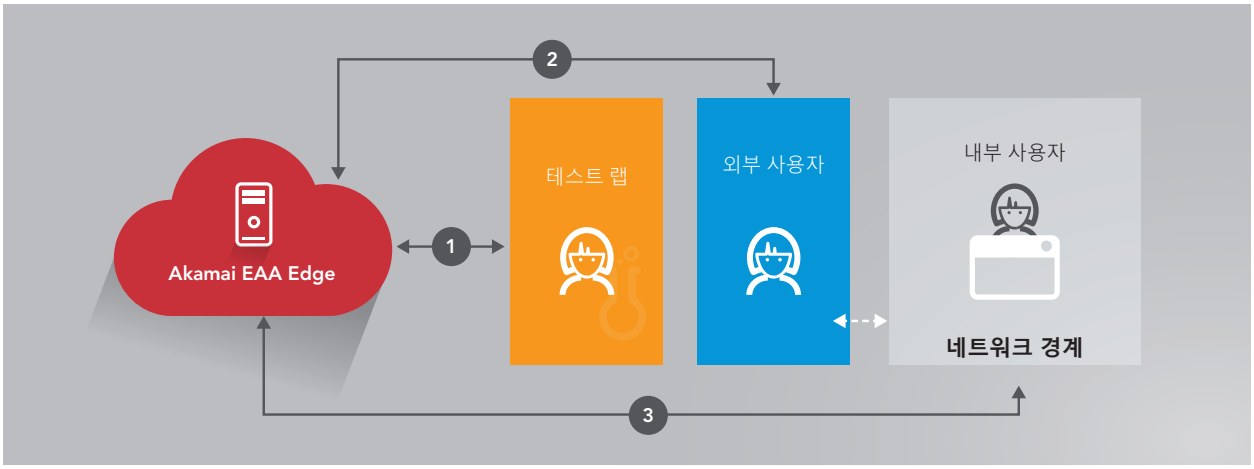
- **테스트 랩:** 이 사용자들은 애플리케이션이 Enterprise Application Access를 통해 처음으로 접속 가능해질 때 애플리케이션의 기능적 무결성을 검증할 책임이 있습니다. 애플리케이션의 운영적 의미 체계를 잘 알고 있어 표준 행동과 엣지 케이스를 모두 테스트할 수 있어야 합니다.

  추가적인 특정 보안 기능과 성능 기능이 레이어로 추가되면 이 사용자들에게는 또한 보안 기능이 예상대로 작동하고 성능이 제대로 강화되었는지 확인하기 위한 백그라운드가 생기게 됩니다.

  수많은 애플리케이션을 보유한 기업은 일반적인 성능 테스팅 및 직원의 많은 감독이 필요 없는 특정한 보안 검사 등, 특정 애플리케이션에 맞추지 않은 표준 검사를 주기적으로, 자동으로 행할 수 있게 지원하는 툴에 대한 투자를 고려할 것을 권장합니다. 이러한 툴로 수행해야 하는 특정한 보안 검사의 대상으로는 SQL 인젝션(SQLi), 크로스 사이트 스크립팅(XSS), 봇 탐지, 명령어 인젝션 공격 등이 있습니다.

- **외부 사용자:** Enterprise Application Access를 롤아웃(rollout)했으며 기업 네트워크 경계 외부의 애플리케이션에 접속할 모든 유효한 사용자로 구성된 운영 사용자의 첫 세트입니다. 이 그룹은 동적인 방식으로 작업합니다. 즉, 사용자가 오프프레미스로 이동할 때는 이 그룹에 동적으로 진입하게 되고, 다시 네트워크 경계로 이동할 때는 그룹을 동적으로 떠나게 됩니다.

- **내부 사용자:** Enterprise Application Access를 롤아웃(rollout)해 애플리케이션의 모든 사용자에 대한 마이그레이션이 완료된 운영 사용자의 마지막 세트입니다. 이 세트는 이 애플리케이션에 접속할 네트워크 경계 내의 모든 유효한 사용자로 구성됩니다. 위의 외부 사용자 그룹처럼, 이 그룹의 멤버십은 동적입니다. 사용자들은 내부 네트워크 경계에 진입하면서 자동으로 이 그룹에 들어오며, 나갈 때는 외부 사용자 그룹에 들어갑니다.



사용자 그루핑 방법론

단계적 전환의 예에서, Akamai의 EAA 제품을 액세스 프록시로 사용해 보겠습니다. Akamai는 DNS를 사용해 분산 네트워크로 애플리케이션을 온보딩합니다. 애플리케이션 호스트 이름은 우리 플랫폼으로 사용자를 라우팅하기 위해 Akamai로 CNAME 처리됩니다.

따라서, 호스트 이름을 조회한 결과로 Akamai에 대한 CNAME 체인이 반환되는 사용자들은 Akamai의 글로벌 네트워크로 온램프되며, 조회 결과로 본래의 내부 A 레코드가 반환되는 사용자들은 계속 오래된 경계 네트워크 방식을 사용해 애플리케이션에 접속합니다. 우리는 어떤 그룹이 CNAME 체인을 따르고 어떤 그룹이 내부 A 레코드를 수신하는지 제어함으로써 각 애플리케이션에 대해 각기 다른 사용자 그룹을 Enterprise Application Access로 단계적으로 전환하기 위한 방안으로써 이 아키텍처를 활용할 것입니다.

우리는 스플리트 호라이즌 DNS(split-horizon DNS)라고도 부르는 DNS 뷰를 사용해 이를 달성하고 위의 세 가지 사용자 세트를 정의할 것입니다. DNS 뷰는 완전히 같은 호스트 이름을 쿼리하는 두 명 이상의 사용자가 소스 IP에 따라 완전히 다른 답변을 받을 수도 있게 하는 방법입니다. 예를 들어, www.foo.bar라는 완전히 같은 호스트 이름을 쿼리했더라도, 중국의 사용자는 하나의 개별적인 IP를 받을 수 있고 영국의 사용자는 이와 완전히 다른 IP를 받을 수 있습니다.

우리는 소스 CIDR 블럭에 따라 각기 다른 사용자 그룹을 조직함으로써 이러한 방법을 활용합니다. 모든 테스트 랩 사용자는 하나의 CIDR 블럭에 있고, 내부 사용자는 기업의 전체 내부 네트워크를 정의하는 CIDR 블럭 내에 있으며, 외부 사용자는 앞의 두 세트와 일치하지 않는 모든 IP에서 소싱될 것입니다.

기업이 현대적인 환경에 배포하는 모든 공통 DNS 서버는 이 기능을 지원합니다. 예를 들어, 부록에 있는 오픈 소스 ISC BIND 패키지에 대한 샘플 설정의 경우에는 Enterprise Application Access를 활성화하려는 애플리케이션이 yourhost.com 도메인에 위치하게 됩니다. 이제 양한 사용자 그룹을 파티셔닝하는 방법을 이해했으니 실제 애플리케이션 롤아웃(rollout) 단계의 논의를 시작할 수 있습니다.

1. 애플리케이션 사전 체크 단계

이 단계에서는 배포한 액세스 프록시의 요구 사항을 애플리케이션이 충족하는지를 확인합니다. Akamai EAA의 경우, 애플리케이션이 지원되는 프로토콜인 HTTP, HTTPS, RDP 또는 SSH를 사용하는지 확인해야 합니다.

2. 액세스 프록시 준비 단계

다음으로는, 액세스 프록시가 애플리케이션과 특정한 보안 및 접속 권한을 인식하도록 설정합니다. Akamai EAA의 경우에는 이를 클라우드에서 설정하며, 설정은 모든 Akamai Connector와 전체 Akamai 네트워크로 즉시 전송돼 모든 Akamai Edge 컴퓨터가 사용자를 서비스할 수 있게 됩니다.

3. 테스트 랩 등록 단계

이제 액세스 프록시가 해당 애플리케이션을 인식하니 사용자 온보딩을 시작할 수 있습니다. Akamai EAA 액세스 프록시를 활용한다고 가정하는 이 단계에서는, DNS 뷰를 위에서 이야기한 바와 같이 수정해 테스트 랩 그룹에 속한 사용자들이 조회 시 특정한 애플리케이션 호스트 이름에 대해 Akamai로 CNAME 처리됩니다.

이 작업 다음으로는, 테스트 랩에 속한 사용자들은 해당 애플리케이션에 접속을 시도할 때마다 즉시 Akamai의 글로벌 네트워크로 이동합니다.

그동안 테스트 랩에 속한 사용자들은 인증이 제대로 작동하는지, 멀티팩터 인증이 적절히 설정되었는지, 이전에 온보딩되었던 애플리케이션 전체에서 SSO가 작동하는지 검증해야 합니다. 더 중요한 부분은, 애플리케이션이 기능적으로 올바른지에 대한 심층적인 테스트를 이 시점에서 행해야 합니다.

4. 보안 업그레이드 단계

테스트 랩에 속한 사용자들이 Enterprise Application Access를 사용해 안전하게 애플리케이션에 접속할 수 있으므로, 기존 경계 모델에서는 불가능했을 고급 보안 기능을 사용하는 것을 고려해야 합니다. 이 섹션에서는, 함께 잘 작동하며 Akamai EAA 액세스 프록시에 의해 활성화된 특정 Akamai 보안 제품을 참조할 것입니다. 하지만, 이러한 특정 제품을 사용하는지의 여부와 무관하게, 일반적인 권장 사항과 배포 단계는 유효해야 합니다.

WAF 기능을 얻기 위해 최소한 Akamai Kona Site Defender 제품을 활성화할 것을 권장합니다. 이를 사용하기 시작하면, 테스트 랩에 속한 사용자들은 애플리케이션에 대한 SQL 인젝션(SQLi), 크로스 사이트 스크립팅(XSS), 명령어 인젝션 공격 등이 WAF 플랫폼에 의해 거부되는지 검증해야 합니다.

Akamai EAA 액세스 프록시를 Akamai Ion과 연동해 활용하는 경우 쉽게 활성화할 수 있는 기능으로는 가벼운 클라이언트리스 보안 체계 평가(Lightweight Clientless Posture Assessment)가 있습니다. 애플리케이션 소유자들은 최소한의 설정을 사용해 애플리케이션에 대한 접속을 거부할 브라우저와 운영 체제를 정책으로 설정할 수 있습니다. 예를 들어, Akamai는 헤더 검사를 통해 보안 리스크가 있는 Firefox의 폐기된 버전을 필터링해서 제외할 수 있습니다.

이 단계에서는 애플리케이션 소유자가 간편하면서 매우 강력한 보안 강화 수단으로서 봇·사용자 탐지 기능을 사용할지 고려해야 합니다. 해당 애플리케이션이 API 서버에 없고 프로그래밍식으로 절대 평가되지 않아야 하는 경우에는 Akamai Bot Manager 제품을 통해 이 기능을 활성화하고, 실제 사람이 행하는 고급 분석을 통한 높은 수준의 명확성으로 판단할 수 없는 접속을 거부하도록 플랫폼에 지시할 수 있습니다. 유효한 사용자 세션을 가장하는 멀웨어와 기타 지속적이며 정교한 위협을 저지하기 위해선 많은 과정을 거쳐야 합니다.

또한, 이 단계에서는 해당 애플리케이션을 매니지드 디바이스에서만 사용하도록 제한해야 하는지 판단해야 합니다. 최종 디바이스에 인증서를 배포하면, 퍼블릭 인증서 기관 인증서를 Akamai EAA에 업로드해 관리되지 않는 모든 컴퓨터에서 오는 접속을 거부하게 할 수 있습니다.

마지막으로, 지역 및 IP 기반 제한도 활성화할 수 있습니다. 사용해야 하는 특정 CIDR 기반 화이트리스트나 블랙리스트를 알거나 접속을 거부해야 하는 지리적 지역을 알고 있다면, 이 시점에서 이 기능을 활성화해 테스트해볼 수 있습니다.

활성화하는 기능과 무관하게, 이 단계에서는 테스트 랩에서 보안 옵션이 작동함은 물론 애플리케이션의 올바른 작동을 방해하지 않는지를 확인해야 합니다.

5. 성능 업그레이드 단계

이제 애플리케이션이 온보딩되고 보안이 구현되었으니, 보안이 부족하지 않은지 확인할 차례입니다. 기존 경계 기반 접속 및 보안 기능에서는, 기업은 애플리케이션 서버와 기업의 지사 간에 연결된 링크의 안정성으로 인해 성능이 제한될 수 있습니다.

가끔 온프레미스 캐싱과 같은 기능은 이러한 문제를 줄이기 위해 배포할 수 있지만, 직원이 오프프레미스로 나가 출장을 가는 경우 상당히 부족해지게 되는데, 이는 캐싱이 사용자와 가까이 있을수록 더욱 우수한 성능 업그레이드를 끌어내기 때문입니다.

이 단계에서는 애플리케이션이 성능 업그레이드가 필요한지 평가하고, 업그레이드가 필요한 경우 사용 가능한 캐싱 및 기타 기술을 활용합니다. Akamai EAA 액세스 프록시를 사용 중이었다면, Akamai Ion 제품을 통해 Akamai의 고급 글로벌 CDN을 활성화하면 확실히 업그레이드가 될 것입니다. 전 세계에 배치된 25만 대에 달하는 Akamai 서버 전체에 걸쳐 캐싱이 활성화되어 업계 최고의 성능을 위치와 무관하게 모든 기업 사용자에게 제공할 수 있습니다. 또한, 이를 활성화하면 순방향 오류 교정(FEC), 라우팅 최적화, 패킷 복제 기능이 패킷 손실을 거의 없애며, 성능 기반 라우팅을 제공하는 Akamai의 고급 오버레이 네트워크에 액세스할 수 있습니다.

옵션으로서 이 단계는 외부 사용자 등록 단계가 실행될 때까지 지연시킬 수 있습니다. 외부 사용자는 네트워크 경계 외부에 있을 때 상대적인 성능 향상을 평가하는 데 가장 적합합니다.

어떤 방법을 사용하든, 성능 향상 효과를 정확히 측정하려면 이 단계와 애플리케이션 등록 전에 성능을 계측하는 것이 좋습니다. 브라우저 플러그인, 워터폴 그래프, 써드파티 성능 모니터링 서비스 등을 사용해 성능을 계측할 수 있습니다.

6. 외부 사용자 등록 단계

이 단계까지 왔다면, 애플리케이션이 온보딩되어 Enterprise Application Access를 통해 접속할 수 있고, 상당한 보안 업그레이드가 적용되었으며, 옵션으로 성능이 강화되었고, 내부 테스트 랩의 테스트를 거쳐 예상대로 작동한다는 신뢰가 생겼을 것입니다.

이 단계에서는 더 많은 사용자에게 애플리케이션을 롤아웃(rollout)해야 합니다. 이 시점에서는, 외부 사용자를 단계적으로 전환하는 것이 좋습니다. 외부 사용자를 위해 이러한 접속 방식을 사용해 최종적으로 VPN을 제거할 것이기 때문입니다. 또한, 외부 사용자는 성능 문제에 가장 많은 영향을 받으며, 이들은 가장 위험한 환경에 있기 때문에 위치에 따라 기업 애플리케이션과 데이터를 위험에 빠뜨릴 수도 있습니다. 외부 사용자를 위해 더 나은 성능·보안을 활용하면 Enterprise Application Access의 가장 뛰어난 장점이 즉시 나타납니다.

DNS 뷰 수정을 통해 이 단계에 들어가기 전에 외부 사용자에게 알림으로써 이들이 전환 과정을 놓치지 않도록 하는 것이 좋습니다. 이 시점까지 모든 것이 적절히 설정되었다면, 외부 사용자들은 성능 향상 효과를 제외하면 전환 과정을 거의 인식하지 못할 것입니다. 하지만, 사전 공지를 통해 기능이 올바르게 작동하는지 사용자들이 최대한 면밀하게 보도록 사용자들에게 알리면, 혹시 어떤 문제가 생길 경우 그 이유를 알고 적합한 관리자에게 연락할 준비가 될 것입니다.

7. 내부 사용자 등록 단계

이 단계까지 왔다면, 외부 사용자들이 등록된 지 어느 정도 시간이 지났을 것이고 운영상 버그가 모두 해결되었을 것입니다. 이 시점에서는 DNS 뷰 전용 설정에서 애플리케이션에 대한 참조를 모두 지우고 이를 일반 뷰에 CNAME 항목으로 추가해도 됩니다. 그럼 모든 사용자는 Enterprise Application Access 프록시를 사용해 기업 애플리케이션에 즉시 접속할 수 있습니다.

외부 사용자를 알리는 데 사용되는 모든 절차는 이 단계의 내부 사용자에도 적용해야 합니다. 이전 6개 단계를 통해, 모든 오류나 설정 오류를 파악하고 수정해야 합니다. 모두 올바르다고 가정하면, 이 시점에서는 Enterprise Application Access에 대한 애플리케이션 운영 전환이 성공적으로 이루어졌고, 모든 사용자가 더욱 쉽고 빠르며 안전하게 접속할 수 있어야 합니다.

8. VLAN 마이그레이션 단계

내부 사용자 등록 단계에서 적절한 시간이 지났다면, 마지막으로 방화벽으로 나뉜 VLAN으로 애플리케이션을 전환할 수 있습니다. 이 전환 작업을 하기 전까지는, 모든 유효한 사용자가 DNS를 사용한 Enterprise Application Access를 통해 전송되었다고 해도 IP 주소를 통해 애플리케이션 서버 자체에 직접 접속할 수 있으므로 기업 네트워크 경계 내부에 있는 멀웨어에 취약합니다.

이 최종 단계는 모든 직접 IP 접속을 제거하며, 애플리케이션과 액세스 프록시 자체를 제외한 모든 것 사이에 방화벽을 적용합니다.

이 작업이 완료되면, 애플리케이션은 공식적으로, 완전히 Enterprise Application Access로 전환됩니다.

포스트 스테이징 작업

모든 애플리케이션이 Enterprise Application Access로 전환되면, 사용자 시스템으로부터 VPN 클라이언트가 완전히 제거되었는지 조사를 시작할 수 있습니다.

또한, 이제 모든 애플리케이션 접속이 제로 트러스트 애플리케이션 접속 모델로 전환되었으므로 내부 사용자 네트워크를 게스트 Wi-Fi로 변환하기 위한 작업 수행을 고려해도 좋습니다.

마지막으로, 이 단계에서는 데이터 센터와 마이크로 페리미터에 대한 접속을 정교한 DDoS 공격으로부터 보호하는 일도 고려해야 합니다. Akamai Prolexic 제품은 이 부분을 지원할 수 있는 솔루션입니다.

요약

클라우드와 모바일이 급격히 발전하는 오늘날의 환경에서 기존의 허브 앤 스포크 네트워크 아키텍처는 보안에 사용되는 성과 성벽 보안 경계와 마찬가지로 성능과 보안을 효율적으로 제공할 수 없습니다. 이는 뒤처지면서 취약한 상태에 머물지 않기 위해 모든 기업들이 풀어야 할 숙제입니다. 오늘날 기업 정보 유출의 가장 중요한 원인은 기업이 더욱 안전한 엔터프라이즈 보안 아키텍처로 전환하지 못해서이며, 이는 더욱 악화될 것입니다. 한 마디로, 기업은 경계 뒤편에 있어도 안전하지 않다는 뜻이며, 그 이유는 경계라는 것 자체가 존재하지 않기 때문입니다.

그럼 제로 트러스트를 어떻게 달성해야 할까요?

Akamai의 클라우드 보안 서비스들을 조합하면 종합적이고 강력한 제로 트러스트 아키텍처를 구축할 수 있습니다. 클라우드 네이티브 환경에서 안전한 애플리케이션 접속을 제공할 뿐만 아니라 클라우드를 활용하므로 내부 기업 네트워크의 필요성이 거의 필요 없어지기 때문입니다.

Akamai의 고급 분산 액세스 프록시와 Enterprise Application Access를 글로벌 Akamai 플랫폼의 강력한 성능과 결합해 사용하면 마이그레이션 리스크 프로필을 거의 완전히 없애고 20년이라는 긴 시간 동안 입증된 Akamai 플랫폼의 성능과 보안 솔루션을 활용해 애플리케이션을 하나씩 단계적으로 전환함으로써 경계가 존재하지 않는 세상으로 아주 쉽게 전환할 수 있습니다.

이 분야가 발전을 거듭해도 Akamai는 항상 모든 단계에서 고객을 지원합니다. 네트워크를 관리하기 아주 쉽고 기업 애플리케이션과 데이터에 대한 접속을 제공할 뿐만 아니라 가장 높은 수준의 보안과 성능을 유지할 수 있는 아키텍처로 전환합니다.

부록

기업이 현대적인 환경에 배포하는 모든 공통 DNS 서버는 DNS 뷰를 지원합니다. 예를 들어, 다음과 같은 오픈 소스 ISC BIND 패키지에 대한 샘플 설정의 경우에는 Enterprise Application Access를 활성화하려는 애플리케이션이 yourhost.com 도메인에 위치하게 됩니다. 주 설정은 다음과 같습니다.

```
named.conf
acl testlab { 10.0.2.0/24; };
acl internal { 10.0.0.0/8; };

view “testlab” {
    match-clients { testlab; };
    recursion yes;
    zone “yourhost.com” {
        type master;
        file “yourhost.com.testlab.zone”;
    };
};

view “internal” {
    match-clients { internal; };
    recursion yes;
    zone “yourhost.com” {
        type master;
        file “yourhost.com.internal.zone”;
    };
};
```

```
view “external” {
    match-clients { any; };
    recursion no;
    zone “yourhost.com” {
        type master;
        file “yourhost.com.external.zone”;
    };
};
```

여기에서는 모든 뷰에서 공유되는 모든 항목을 위해 하나의 존 파일(zone file)을 사용합니다. 이는 속해 있는 사용자 그룹과 무관하게 항상 일관성을 유지하는 호스트와 DNS 항목입니다. 이는 조직별로 고유하지만, 다음과 유사한 모습을 보여줍니다.

```
common.zone
a$TTL 86400
@       IN      SOA     ns1.yourhost.com.    ns1.yourhost.com. (
                        2001062501 ; serial
                        21600      ; refresh after 6 hours
                        3600       ; retry after 1 hour
                        604800     ; expire after 1 week
                        86400 )    ; minimum TTL of 1 day

yourhost.com. IN      NS      ns1.yourhost.com.
yourhost.com. IN      NS      ns1.yourhost.com.

ns1.yourhost.com.     IN      A       10.0.1.1
ns2.yourhost.com.     IN      A       10.0.1.2

server1.yourhost.com.       IN      A       10.0.1.101
server2.yourhost.com.       IN      A       10.0.1.102
additional common entries…
```

Enterprise Application Access를 사용하는 애플리케이션에 접속하도록 전환된 각 사용자 그룹은 다음과 같은 특정한 존 파일(zone file)을 가지게 됩니다(다음 예에서는 전환된 애플리케이션이 app1.yourhost.com입니다).

```
$INCLUDE “common.zone”;
app1.yourhost.com.   IN      CNAME  app1.yourhost.com.sohacloud.akamai.com.
```

Enterprise Application Access를 통해 애플리케이션에 접속하도록 전환되지 않은 사용자 그룹에 대한 존 파일(zone file)은 다음과 같은 모습을 보여줍니다(기재된 IP는 예시일 뿐입니다).

```
$INCLUDE “common.zone”;
app1.yourhost.com.   IN     A      10.0.1.105
```

이 두 예시를 합쳐 좀 더 구체적인 예시를 보여드리겠습니다. 테스트 랩 및 외부 사용자 그룹이 Enterprise Application Access를 통해 내부 Jira 애플리케이션에 접속하도록 전환되었지만, 내부 사용자는 여전히 기존 경계 기반 접속을 사용하고 있는 경우에는 존 파일(zone file)이 다음과 같은 모습을 보여줍니다.

```
yourhost.com.testlab.zone
$INCLUDE “common.zone”;
app1.yourhost.com.   IN     CNAME  jira.yourhost.com.sohacloud.akamai.com.
yourhost.com.external.zone
$INCLUDE “common.zone”;
app1.yourhost.com.   IN     CNAME  jira.yourhost.com.sohacloud.akamai.com.
yourhost.com.internal.zone
$INCLUDE “common.zone”;
app1.yourhost.com.   IN     A      10.0.1.156
```