백서

SDV(Software-Defined Visibility): IT의 새로운 패러다임

사물인터넷(IoT)의 시대. 우리는 기기 대 기기의 의사소통이 필요한 새로운 세계에 살고 있다. 오늘날 디지털 비즈니스 인프라의 모든 요소는 그러한 의사소통을 촉진하기 위해 상호작용할 수 있어야 한다.

IT 관리자가 일상적인 비즈니스 활동에서 발생하는 사건에 보다 민첩하게 반응하고 대응할 수 있으려면 '자동화'를 한 단계 업그레이드시킬 수 있는 프레임워크가 필요하다. 비즈니스 인프라와 정보 시스템은 하루가 다르게 발전하고 있으며, 다양한 하위 시스템과 개별 구성요소들 간의 복잡한 상호작용을 수반하고 있다. 이러한 환경에서 민첩성은 이제 선택 사항이 아닌 첨단 IT의 토대라고 할 수 있을 것이다

자동화 프레임워크는 향상된 민첩성과 생산성은 물론이거니와, 그 이상의 다양한 가치를 제공한다. 특히 보안 위협에 노출될 가능성이 높은 상황에서, 업그레이드된 '자동화'는 보안 위협을 빠르고 지속적으로 탐지하고 대응하는 데 필수적인 요건이라 할 수 있다.

자동화와 빠른 보안 응답을 통해 민첩성의 요구를 충족시키는 솔루션은 일련의 개방형 RESTful API를 네트워크 가시성 인프라에 통합시킴으로써 보안 및 기타 네트워크 모니터링 어플라이언스가 직접 상호작용하도록 해준다. 관리자는 이러한 개방형 RESTful API를 활용함으로써 가시성의 기반을 강화·확장할 수 있는, 프로그래밍 가능한 프레임워크를 개발하고 사용할 수 있다. 관리자는 그들의 인프라에서 탐지된 위협 패턴에 맞설 동적 대응, 인라인 보안 툴을 위한 트래픽 모드 조정 및 추가적인 IT 운영·관리 역량 등 여러 기능을 자동화할 수 있을 것이다.

가시성 패브릭(Visibility Fabric)의 필요성

오늘날의 데이터 네트워크는 수많은 장치와 어플라이언스, 어플리케이션으로 북적이고 있다. 네트워크 복잡성은 하루가 다르게 깊어지고 있으며, 이러한 추세는 가상 인프라의 확산, 빅데이터 활용 및 관련 비즈니스의 증대, 사용자 장치의 폭발적 증가를 통해 더욱 가속화되고 있다. 보안 아키텍처만 보더라도 차세대 방화벽(Next-generation Firewall, NGFW), 침입 방지 시스템(Intrusion Protection System, IPS), 웹 어플리케이션 방화벽(Web Application Firewall, WAF), 침입 탐지 시스템, Intrusion Detection System, IDS), 보안 정보 및 이벤트 관리 시스템(Security Information and Event Management System, SIEM), 인라인 또는 아웃오브밴드 보안 어플라이언스 등 다양한 툴이 필요하다. 그리고 이 모든 시스템은 그 시스템들이 들여다보는 네트워크 트래픽 만큼의 효과가 있다. 네트워크 트래픽 가시성의 폭이 보안 아키텍처의 효과에 직접적인 영향을 미치기 때문이다.

광범위한 가시성을 위하여, 데이터센터와 물리, 가상 및 SDN/NFV 환경, 프라이빗/퍼블릭 클라우드에 존재하는 수많은 장치와 어플리케이션에서 네트워크 트래픽을 확보해야 한다. 그리고 이 고성능 노드의 분산 계층은 네트워크 트래픽 입력이나 네트워크 트래픽에서 추출한 플로우 레코드 등을 통해 필요한 운영 툴의 전체적인 모습을 제공하는 Visibility Fabric™을 형성한다. 사각지대를 없애고 전체 네트워크에 빠른 액세스를 제공할 수 있는 통일된 가시성 모델을 제공하는 것이다.

SDV(Software-Defined Visibility)의 중요성

SDV(Software-Defined Visibility)란 무엇인지 이해하기 위해 다음과 같은 질문부터 시작하기로 한다.

- 가시성 패브릭(Visibility Fabric)으로부터 트래픽을 수신하는 어플리케이션과 다른 운영 툴이 관리자의 개입을 기다리지 않고 이벤트에 능동적으로 대응할 수 있는 방법이 있다면 어떨까?
- 위협 패턴을 탐지했던 보안 어플리케이션이 해당 위협에 반응하고 대응하기 위해 트래픽을 자동으로 조정할 수 있는 능력이 있다면 어떨까?
- 자동화로 가시성을 구현하는 최선의 방법은 무엇일까?

CISO(Chief Information Security Officer) 또는 IT 운영의 이해관계자인 경우 처음 두 질문에 대한 답은 "그거 대단할 것 같군요!"일 것이다. 그러나 세 번째 질문에서는 잠깐 멈추게 될지 모른다.

Visibility Fabric의 광범위성은 IT 통합관리에 반드시 필요한 구조적 기반이지만, 현재 가장 두드러진 과제를 해결하려면 이것 하나만으로는 충분하지 않다. 새로운 사각지대가 등장함으로 인해, 이러한 사각지대를 탐지한 뒤 이를 제거하려면 가시성 인프라에 대한 동적 변경을 수행해야 하기 때문이다. 따라서 IT 관리자에게는 가시성 인프라가 네트워크 액세스를 약화시키는 이벤트·상황에 동적으로 대응할 수 있도록 자동화를 강화할 수 있는 프레임워크가 필요하다. 이러한 능력은 첨단 IT를 위해 반드시 필요한 빌딩 블록이다. 이를 구현하는 최고의 방법은 무엇일까?

강력하면서도 보편적인 접근법은 RESTful API(Application

Programming Interface)에 기반한 웹 서비스 프레임워크를 Visibility Fabric에 직접 통합하는 것이다. 이 접근법을 통해 네트워크의 모든 장치는 필요에 따라 Visibility Fabric과 직접 상호작용할 수 있다. 중앙집중식 정책 컨트롤러를 통해 노출된 API는 계획적인 방식으로 Visibility Fabric과 상호작용할 수 있는 능력을 외부 시스템에 제공하기 때문이다. 이러한 개방형 RESTful API는 광범위하고 능동적이면서도 매우 민첩한 가시성을 보장하기 위해 Visibility Fabric 자체의 프로그램 가능성도 지원한다. 이처럼 매우 계획적이며 자동화하기 쉬운 프레임워크를 SDV(Software-Defined Visibility)라고 부른다. SDV는 네트워크 보안과 IT 운영 관리의 새로운 패러다임이다

RESTful API를 이용한 SDV 구현

확장 가능한 웹 서비스는 오늘날 웹에서 일어나는 대다수 활동을 뒷받침하는 기초적인 아키텍처라고 할 수 있다. 이를 구현하는 가장 일반적인 방법 중 하나는 REST(Representational State Transfer)라는 프레임워크에서 찾을 수 있다. 클라이언트-서버 모델에 기반한 RESTful 아키텍처는 역량과 인기를 누릴 만한 수많은 특징을 가지고 있기 때문이다.

- **성능**—REST API의 성능은 네트워크 지연 같은 외부 요소가 아닌 시스템 구성요소 간 실제 상호작용에 의해 주로 제한된다.
- **확장성**—REST는 다수의 네트워크 구성요소와 그들 간의 상호작용을 수용하기 위해 특별히 설계되었다.
- **유연성**—REST 구성요소는 새로운 요구사항이 생겨남에 따라 쉽게 수정할 수 있다.
- **가시성**—구성요소 간 의사소통이 서비스 에이전트에게 보이도록 설계되었다.
- **이식성**—REST는 데이터를 포함한 프로그램 코드를 이동할 수 있기 때문에 이식성이 탁월하다.
- **신뢰성**—RESTful 구현은 개별 구성요소 수준에서 문제에 직면하는 경우에도 일반적으로 시스템 수준의 장애로부터 영향을 받지 않는다.

RESTful 시스템은 일반적으로 무상태(stateless), 캐시 가능(cacheable), 계층형(layered)의 표준 HTTP 메소드 (GET, POST, PUT, DELETE 등)를 포함해 유비쿼터스 HTTP 프로토콜을 사용해서 의사소통을 수행한다. API는 그림 1에 나온 것처럼 URI(Uniform Resource Identifier)라고 알려져 있는 가용 어플리케이션 리소스에서 작동한다.

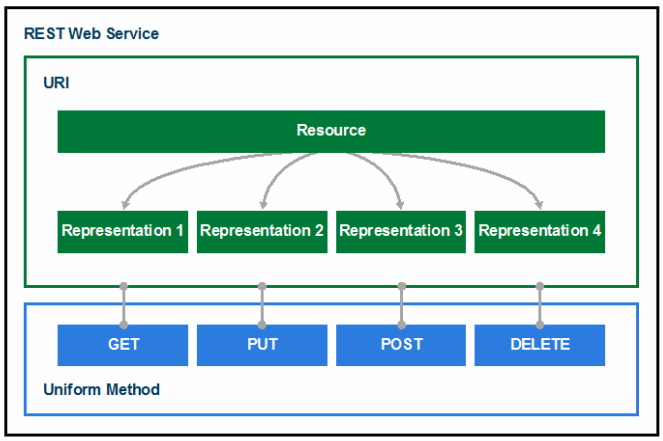

그림 1: 종합적인 RESTful 웹 서비스 아키텍처

Visibility Fabric 내에서의 API 기반 작업 흐름

일반적인 REST 작업 흐름은 간단하게 다음과 같이 서술할 수 있다.

- 클라이언트 어플리케이션이 서버에 리소스 생성, 판독, 업데이트 또는 삭제를 요청한다.
- 서버가 요청에 응답한다. 응답은 통상적으로 요청의 성공 또는 실패 여부를 나타내는 상태 코드를 포함하고 있다. 또한 응답은 상태 코드 외에 구조화된 형식의 데이터를 포함하고 있으며 어플리케이션은 상태 코드에 따라 추가 조치를 취하거나 데이터를 처리한다.

Visibility Fabric에서 "REST 서버" 기능은 중앙집중식 정책 컨트롤러에 통합된다.

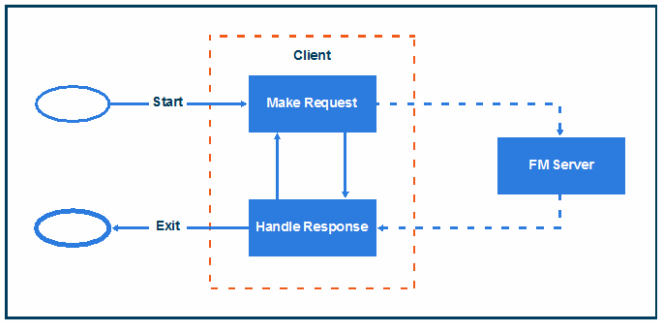

그림 2: 기본 REST 작업 흐름

RESTful API를 Visibility Fabric에 통합시키면 가장 뚜렷하게 새로 지원되는 작업 흐름에는 노드(장치) 관리, 포트 구성, 트래픽 맵 구성 등이 포함된다. 이처럼 RESTful API는 계획적인 방식으로 Visibility Fabric과 상호작용하는 개방형 인터페이스를 제공한다.

노드 관리. 노드(장치) 관리 작업 흐름의 일반적 기능은 장치를 Visibility Fabric에 추가하거나 제거하고, 노드 또는 장치와 관련한 정보를 발견해 확보하는 것이다.

포트 구성. 노드를 Visibility Fabric에 추가하고 난 다음 단계는 장치에 포트(인터페이스)를 구성하는 것이다. 적어도 어플리케이션은 API를 이용해서 포트를 생성, 표시, 업데이트, 삭제할 수 있다. Visibility Fabric의 세부 사항에 따라 어플리케이션은 포트를 논리적 번들로 묶을 수 있어, 여러 모니터링/관리 툴 간에 트래픽을 분산시키는 것도 가능하다.

트래픽 맵 구성. 트래픽 맵 생성은 Visibility Fabric이 지원하는 핵심 기능이다. 트래픽 맵은 소스로부터 관심 대상인 플로우를 추출해서 이를 원하는 대상(툴)에 안정적으로 전달하여 작동시키는, Visibility Fabric 내부 정책의 핵이다. Visibility Fabric을 올바르게 구현하고 통합했다면, REST API 세트는 클라이언트 어플리케이션이 필요에 따라 이러한 맵을 실시간으로 생성, 표시, 업데이트하도록 지원할 수 있어야 한다.

Visibility Fabric 내에서의 RESTful API 사용

이러한 작업 흐름은 노드(장치), 포트, 트래픽 맵 처리와 관련해 세 가지 주요 클래스를 제시한다. 위에서 언급했던 것처럼 구체적 이용 사례는 다음과 같이 세 가지 범주로 나뉜다.

- 노드 제거나 추가, 또는 노드 관련 정보 획득
- 포트 관련 정보 획득, 유형 수정 및 지원
- 트래픽 맵 생성 또는 수정

이런 능력들이 수많은 보안 및 IT 운영-관리 기능에 얼마나 유용한지는 쉽게 상상할 수 있다. 예를 들어, 기준에서 벗어난 트래픽 패턴을 탐지한 보안 장치는 Visibility Fabric이 검사나 차단 같은 후속 보안 운영을 수행하도록 구성할 수 있다.

표준 TCP 포트 번호(443)에서 통상적으로 예상되는 SSL 트래픽을 떠올려 보도록 하자. 많은 수의 SSL 상호작용이 비표준 포트에서 발생하는 경우, Visibility Fabric으로부터 SSL 트래픽을 수신하는 어플리케이션은 이를 수신자에게 전송하기 전에 패브릭이 SSL 트래픽을 복호화하도록 구성할 수 있다. 이는 수신 보안 어플리케이션이 복호화된 트래픽을 추가로 분석하고 검사할 수 있도록 지원한다.

두 번째 예는 원격 사이트의 모니터링이다. 통상적으로 원격 사이트(지사 사이트, 센서를 보관하고 있는 장소 등)는 방대한 WLAN 링크를 통해 본사/중심 위치에 연결되어 있다. 그런데 이들 사이트는 모니터링 및 보안 툴, 인력 등 자원이 부족한 경향이 있다. 그러한 원격 사이트를 모니터링하는 효과적인 방법은 원격 사이트에 물리적으로 위치해 있으면서 Visibility Fabric의 일부분인 가시성 노드의 미시험 플로우 레코드를 생성하는 것이다. 이상 활동이 탐지되면 원격 노드는 분석에 필요한 추가 정보를 얻기 위해 트리거링을 하고 전체 트래픽 흐름을 전송하도록 프로그래밍할 수 있다.

SDN(Software-Defined Networking) 맥락에서의 SDV

SDV(Software-Defined Visibility)는 가시성 인프라와 관련된 아키텍처이며, SDN(Software-Defined Networking)은 네트워크 인프라와 관련된 아키텍처이다. SDV는 광범위한 가시성 도달범위를 자동화 프레임워크와 결합해 준다.

SDN 인프라에서 네트워크 스위치와 라우터는 물리적 네트워크 또는 레이어 2~3 데이터 플레인을 구성한다. 가상 네트워크는 공통 인프라에서 멀티테넌시를 허용하기 위해 VXLAN, MPLS, NVGRE 등과 같은 캡슐화를 이용하는데, SDN 컨트롤러는 제어 및 관리 플레인을 지원하고, 이러한 플레인은 가상 및 물리 네트워크 제어를 제공한다.

SDV 인프라는 SDN과 비슷한 방식의 계층형 구조이지만, 지능형 가시성을 제공하도록 최적화되어 있다. Visibility Fabric 노드는 Visibility Fabric의 분산 요소를 형성하며 타사 네트워크 스위치에서 실행되는 물리(하드웨어) 노드, 가상(소프트웨어) 노드 또는 가시성 소프트웨어로 구현될 수 있다. 이들 패브릭 노드는 패브릭 서비스(Flow Mapping®,클러스터링 등)와 트래픽 인텔리전스(SSL 복호화, 어플리케이션 필터링, 중앙집중식 플로우 레코드 생성, 풍부한 메타데이터 등)를 제공하기 때문에 Visibility Fabric에 연결된 보안/운영 툴로 전송되는 원치 않는 트래픽의 양을 줄여준다. 중앙집중식 정책 컨트롤러는 물리, 소프트웨어 및 가상 노드를 포함한 전체 패브릭의 중앙집중식 관리를 제공한다. 타사 시스템은 패브릭에서 정보를 얻거나 통합 API를 통해 패브릭의 동작을 수정할 수 있다.

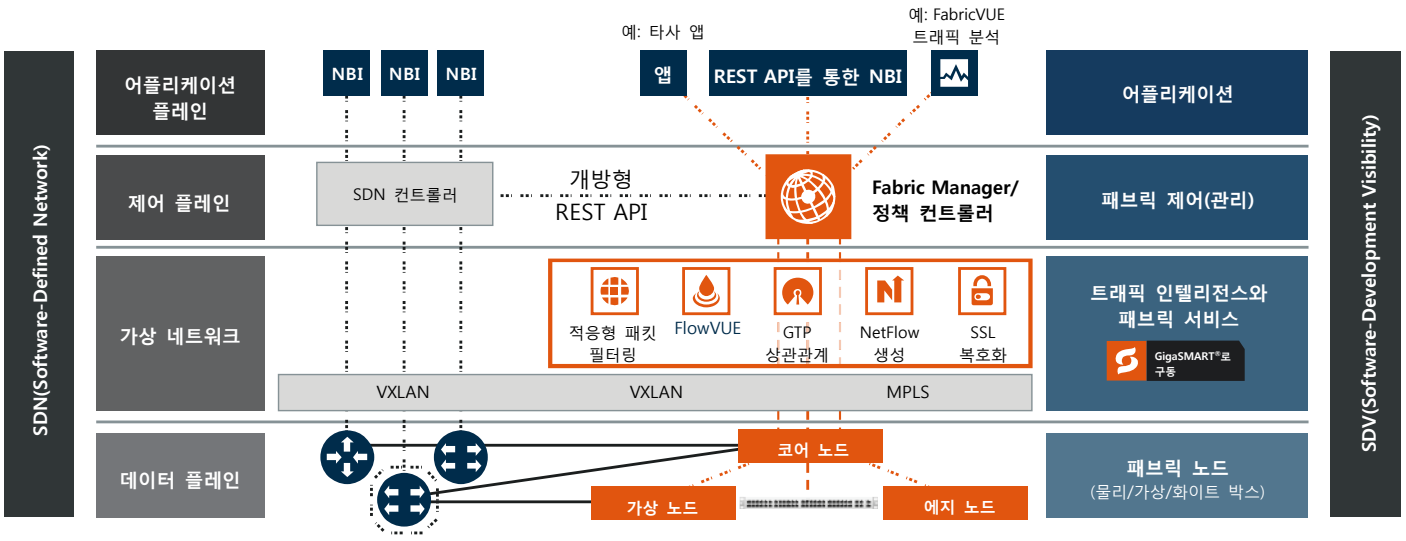

그림 3: SDN과 SDV의 관계

주요 SDV 이용 사례

RESTful API를 Visibility Fabric에 추가하면 Visibility Fabric은 네트워크 모니터링을 위해 정적인 계층에서 적극적이고 능동적이며 민첩한 계층으로 완전히 변화한다. 보안 위협, 노드·장치 상태의 변경, 트래픽 패턴의 변경(새로운 사각지대의 등장 포함) 등에 실시간으로 대응할 수 있다. RESTful API는 패브릭을 프로그래밍할 수 있으며, 노드와 장치, 어플라이언스 및 기타 모니터링/관리 툴에서 실행되는 클라이언트 어플리케이션을 지원한다.

아래에 설명되어 있는 이용 사례에서 RESTful API는 각각의 예가 포함된 그림에 묘사된 것처럼 정책 컨트롤러를 통해 Visibility Fabric과 직접 상호작용한다.

이용 사례 1: 위협 패턴 탐지 및 대응

이 활용 사례는 보안 어플라이언스의 위협 패턴 탐지가 계획적인 대응을 실시간으로 할 수 있는 능력과 결합해 장치와 프로세스를 어떻게 변모시키는가에 대한 예를 제공한다.

이 이용 사례에서는 통합 API가 수행하는 주요 운영에 다음이 포함될 수 있다.

1. 트래픽 패턴 불규칙성 또는 기타 의심스러운 거동을 평가하기 위한 플로우 맵 생성
2. 예상치 못한 서버에 표시되거나 특이한 볼륨으로 표시되는 SSL 암호화 트래픽의 복호화 및 스니핑
3. 선별적 필터링 기준 및 기타 기법에 기반한 주문형 패킷 캡처

이용 사례 2: 인라인 보안 툴을 위한 "모니터링 모드" 자동 조정

이 사례는 Visibility Fabric에 통합된 API가 어떤 방법으로 시스템 관리자의 시간과 노력을 크게 줄여주는지에 대한 예를 제공한다. 보안장비의 구성은 네트워크와 함께 인라인 배치되기 전에 미세하게 조정되어야 하는 경우가 많다. 미리 결정된 일련의 조건이 충족되면 관리자는 보안장비 배치를 아웃오브밴드 작동에서 네트워크 트래픽 흐름을 포함한 인라인으로 변경한다. Visibility Fabric 정책 관리에 대한 적절한 액세스 권한을 보유한 관리자는 이를 수동으로 수행하는 대신 Visibility Fabric API를 이용해 변경 일정을 자동으로 수립할 수 있다. 반대로, 인라인 툴은 유지보수를 오프라인으로 할 수 있으며, Visibility Fabric API를 이용한 비슷한 프로세스를 통해 작업을 자동화할 수도 있다.

이 사례를 위해 API를 사용해서 수행되는 별개의 작동은 다음과 같다.

1. 모니터링 모드를 대역 외에서 패브릭 내에 있는 인라인으로 변경
2. 그에 따라 Visibility Fabric의 플로우 맵 업데이트
3. 정의된 기준 세트에 기반한 트래픽 감소

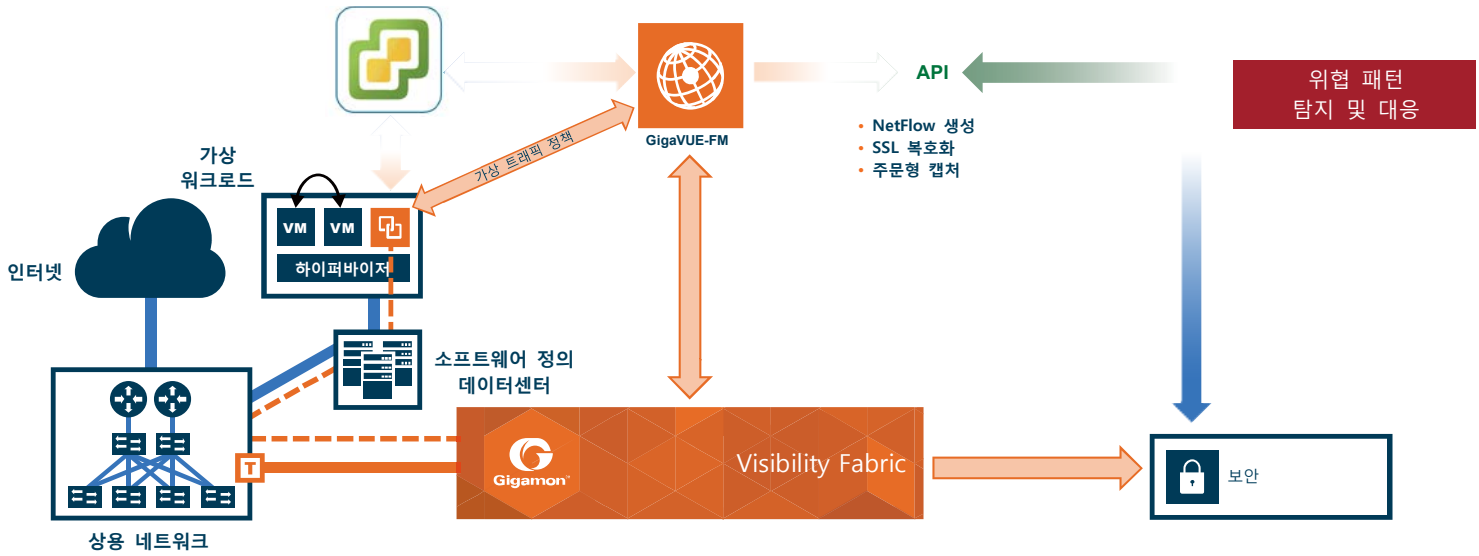

그림 4: 위협 탐지 및 대응

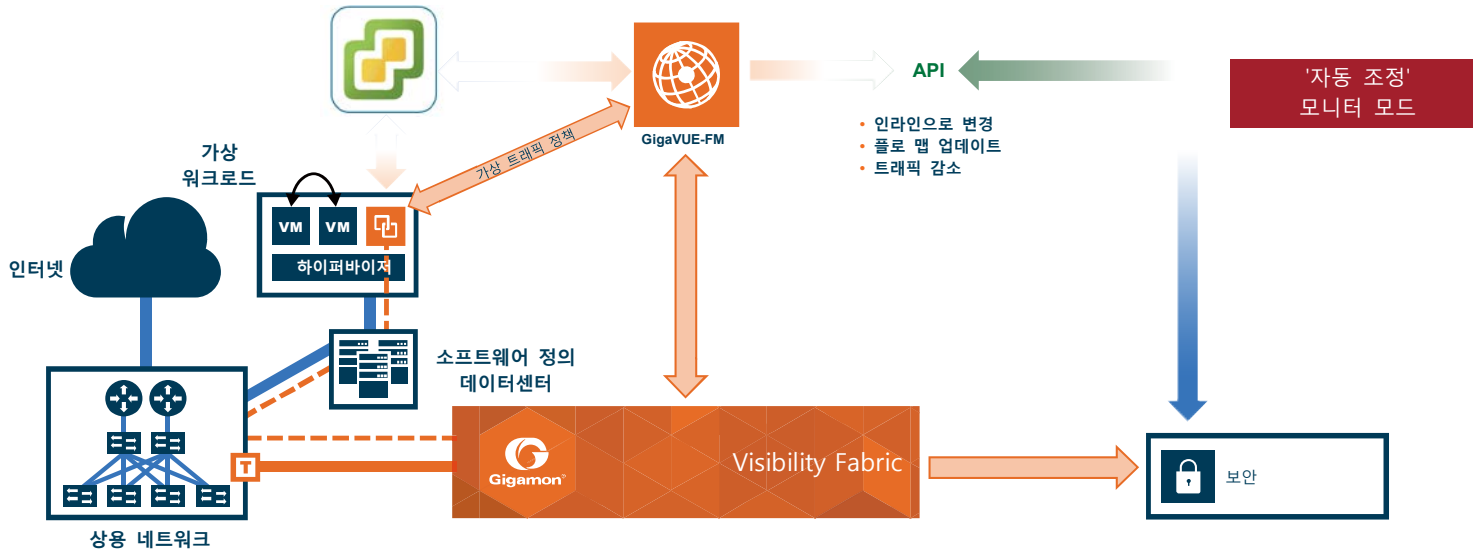

그림 5: 인라인 배치되어야 하는 보안 툴에 대한 자동 조정 모니터 모드

이용 사례 3: 프라이빗 클라우드와 가상 데이터센터 모니터링

이 이용 사례는 적극적이고 민첩한 Visibility Fabric을 통해 프로세스를 자동화함으로써 시간과 수고를 줄일 수 있는 방법의 또 다른 예이며, 패브릭을 가상화 환경으로 확장하는 작업의 중요성을 보여준다. 이 사례에서는 어플리케이션 성능과 네트워크 관리 모니터링 장치(또는 어플라이언스), 그리고 보안 기능을 수행하는 동등 장비에서 작동하는 클라이언트에 의해 API가 실행된다.

이 사례를 위해 API를 이용해서 수행되는 별개의 운영은 다음을 포함할 수 있다.

1. Visibility Fabric 자체(이 사례에서는 모든 클라우드 기반 가상 네트워크와 기기 활동을 모니터링할 소프트웨어)의 가상 요소 배포
2. 트래픽과 보안 정책 생성 및 배포

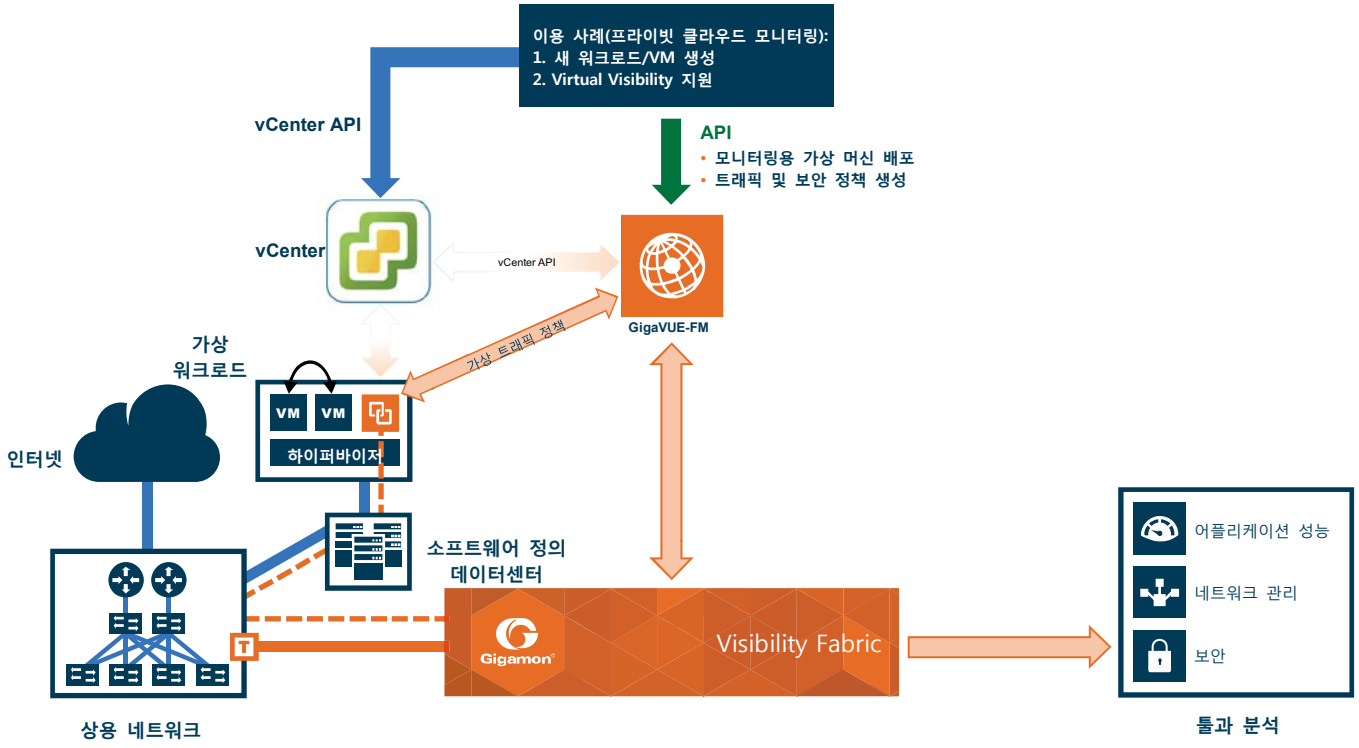

그림 6: 프라이빗 클라우드와 가상 데이터 센터 모니터링

이용 사례 4: IT 운영 관리를 위한 자동화 Visibility Fabric 사용

마지막 사례는 API를 통해 Visibility Fabric에서 지원되는 기능들이, 이전에는 대부분 수동으로 처리되었던 중대한 IT 운영관리 작업을 어떻게 자동화시키는지 그 방법을 보여준다. 그림 7에 나온 것처럼, 두 개의 개별적인 하위 사례, 즉 네트워크의 모든 장치에 대한 재고목록 작성과 적절한 프로비저닝을 통한 서비스 요청 티켓에 대한 대응을 살펴보자.

이러한 작업 수행에서 API가 수행하는 일부 개별 운영은 다음을 포함할 수 있다.

- 모든 네트워크 연결 항목에 대한 장치/노드 정보 얻기
- 필요에 따라 이들 장치에서 통계 자료 수집
- 프로비저닝 요청에 따라 필요한 포트 구성
- 새로 프로비저닝된 네트워크 요소를 통합하는 새로운 트래픽 맵 생성

인증 및 역할 기반 액세스 제어(RBAC)

API가 지원하는 자동화는 관리 및 모니터링 시스템을 위한 효율적인 수단을 제공하는 것과 함께, 인증된 사용자 및 인가된 사용자만 허용된 리소스를 실행할 수 있도록 하는 것이 매우 중요하다. Visibility Fabric은 로컬 데이터베이스 또는 일반적으로 사용되는 프로토콜(RADIUS, TACACS+, LDAP 등)을 통해 견고한 RBAC 및 인증 서비스를 제공할 수 있다. 또 관리자는 특정 인터페이스에 대한 관리 감독을 제한하기 위해 추가 정의할 수 있다. 이 밖에도 SDV 지원 RESTful API는 RBAC 감독 기준과 동일한 일련의 기준을 준수한다.

기가몬의 Unified Visibility Fabric: SDV 패러다임 활용

기가몬의 포트폴리오에서 GigaVUE-FM(Fabric Manager)은 Unified Visibility Fabric을 위해 정책 컨트롤러 역할을 수행한다. Fabric Manager는 견고하고도 완전한 RESTful API 세트를 통합하고 있다. 이들 API는 새로운 차원의 능력을 만들어내기 위해 Unified Visibility Fabric을 발전시키고, 보완하며, 변화하는 조건에 대응해 실시간으로 적극적·능동적으로 자동 대응할 수 있게 해준다. 이는 진정한 패러다임의 전환이며, Software-Defined Visibility라는 새로운 차원의 가시성을 구현하는 것이다.

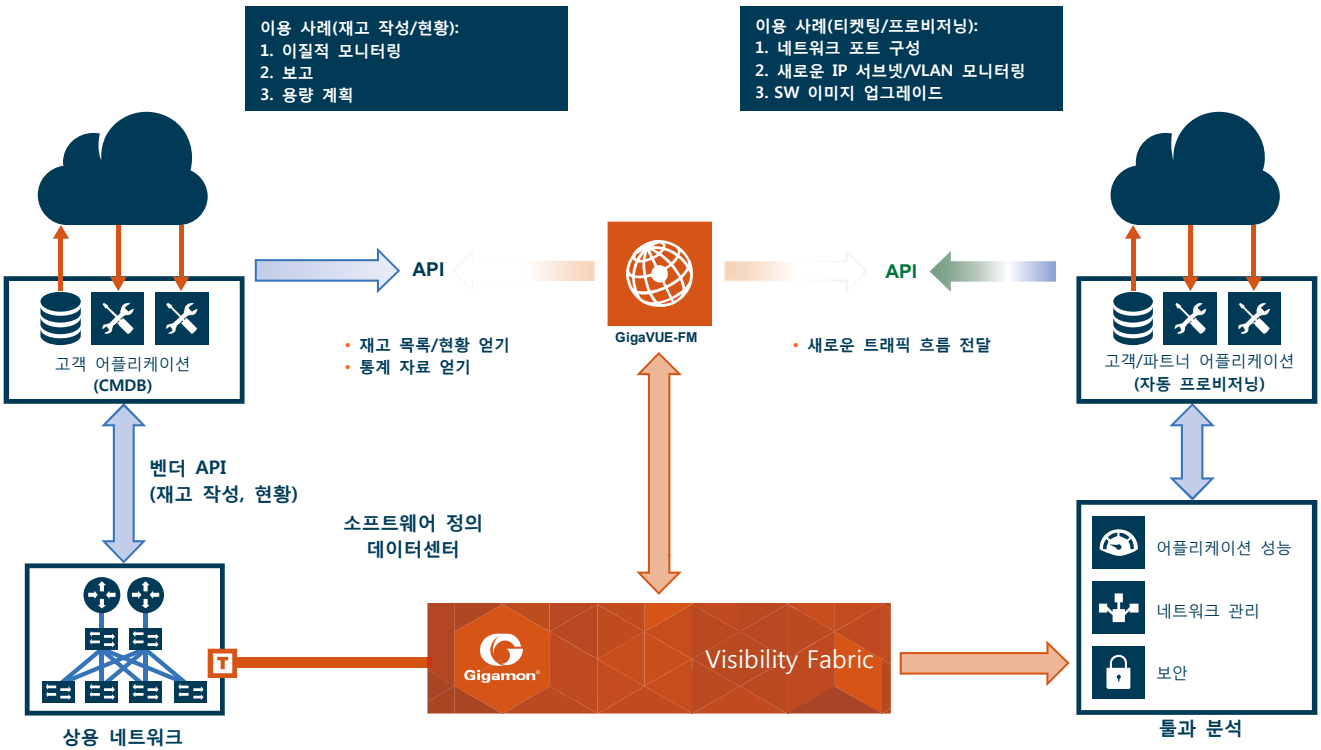

그림 7: API가 IT 운영 관리를 단순화하는 방법

API는 GigaVUE-FM에 통합되며 패브릭의 다른 주요 요소와 함께 완벽하게 작동한다. 여기에는 Visibility Fabric 노드를 구동하고 Flow Mapping, 클러스터링 및 인라인 바이패스 등의 패브릭 서비스를 제공하는 운영체제 소프트웨어인 GigaVUE-OS™, Visibility Fabric 노드에 내장된 고성능 컴퓨팅 플랫폼에서 구현되는 업계 최고의 트래픽 인텔리전스 어플리케이션 세트 GigaSMART®가 포함되어 있다. 트래픽 인텔리전스 어플리케이션의 기능에는 패킷 중복제거, SSL 복호화, 패킷 슬라이싱, 패킷 마스킹, 헤더 제거, 터널링, NetFlow 생성 및 적응형 패킷 필터링 등이 포함된다.

SDV 구현에 사용되는 RESTful API는 개발 파트너와 고객이 기존의 가시성 도구를 네트워크 조건 및 기타 인프라 변화에 맞춰 즉시 대응할 수 있는, 완전히 자동화된 시스템으로 개조할 수 있도록 개방형 인터페이스를 제공한다.

이러한 개방형 RESTful API를 사용하는 기가몬 Visibility App for Splunk는 IT 운영 관리(ITOM) 사용자를 위해 Visibility Fabric의 건강성과 분석 능력을 확장해 준다. 이 어플리케이션은 SecOps 팀과 NetOps 팀이 ITOM 영역 내에서 1급 문제해결을 트리거링할 수 있도록 상용 네트워크로부터 수집한 인텔리전스를 강화할 수 있다.

기가몬 Customer Portal은 사용자 커뮤니티가 RESTful API와 SDV를 채택하도록 돕기 위해 고객들이 아이디어와 이용 사례를 사용하고 교환할 수 있는 샘플 설명서와 스크립트를 위한 중심축 역할을 한다.

RESTfulAPI 출력 예

아래는 IT 운영자가 Visibility Fabric 재고 목록을 보려는 경우, API와 JSON(JavaScript Object Notation) 출력을 사용한 요청/응답 샘플이다.

```
Request:
GET https://<GigaVUE-FM IP address>/api/v1/nodes/flat

Response:
{
"clusters" : [ {
"family" : "H",
"clusterId" : "10.115.152.50",
"members" : [ {
"deviceId" : "10.115.152.50",
"deviceIp" : "10.115.152.50",
"dnsName" : "hc2-c04-29.gigamon.com",
"hostname" : "HC2-C04-29",
...
"clusterMode" : "Standalone",
"clusterId" : "10.115.152.50",
...
} ]
} ]
}
```

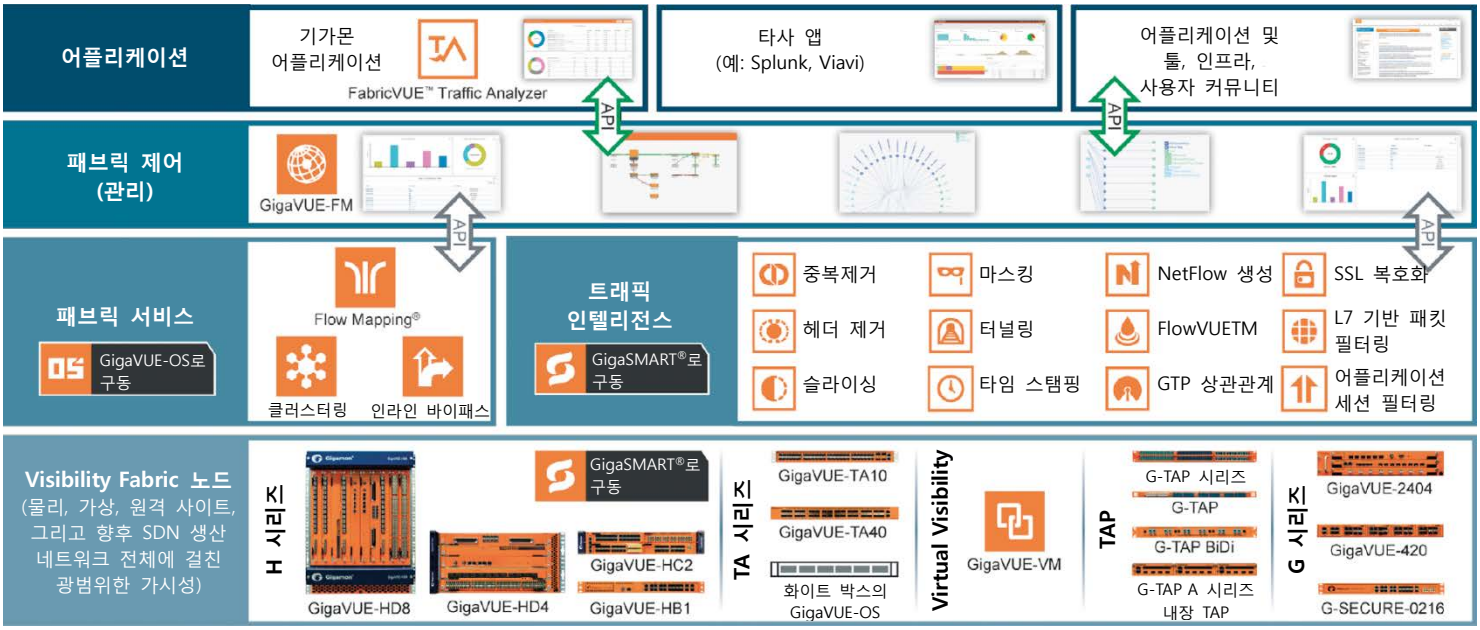

그림 8: 기가몬의 Unified Visibility Fabric

결론

이 백서는 기기 대 기기의 의사소통을 지원하는 데 있어 Visibility Fabric을 가장 효율적으로 개선하는 방법을 알고 싶어하는 IT 관리자들에게 해답을 제시한다. 또한 이벤트나 상황이 발생했을 때 그에 대응하여 패브릭의 동작을 계획적으로 변경하기 위해 모니터링, 관리 및 보안 어플라이언스에서 트리거링할 수 있는 접근 방법이 얼마나 혁신적인지도 보여준다. RESTful API를 통해 구현된 SDV(Software-Defined Visibility)의 새로운 패러다임은 이 요건을 충족시키기 위한 최적의 접근법을 제시할 것이다.

기가몬 소개

기가몬은 물리 및 가상 네트워크 트래픽에 Active Visibility를 제공하며, 보다 강력한 보안 기능과 탁월한 성능을 지원한다. 업계 최초의 보안 전달 플랫폼인 기가몬의 Visibility Fabric™및 GigaSECURE®는 엔터프라이즈, 정부 및 서비스제공업체 네트워크의 보안, 네트워크, 어플리케이션 성능 관리 솔루션이 보다 효율적이고 효과적으로 운영될 수 있도록 고급 인텔리전스를 제공한다. 추가 정보는 **http://www.gigamon.com/**에서 확인 가능하다.

KR 3167-03 12/15

Gigamon®

본사: 3300 Olcott Street, Santa Clara, CA 95054 USA | +1 (408) 831-4000 | www.gigamon.com
한국지사: 서울시 강남구 영동대로 511, 33층 (삼성동 트레이드타워) | T. 02 6007 2288 | F. 02 6007 2001