대용량 과학데이터 전송환경에서의 접근제어정책 적용방안

권우창, 박병연*
한국과학기술정보연구원, *한국과학기술정보연구원
wckwon@kisti.re.kr, *bypark@kisti.re.kr

Applicability of access control policy in large-scale scientific data transmission environment

Kwon Woo Chang, Park Byung Yeon*
Korea Institute of Science and Technology Information

요 약

본 논문은 대용량 과학데이터 전송에 최적화된 ScienceDMZ 네트워크 모델을 기반으로 구성된 네트워크에서 보안성을 확보하기 위한 방안인 접근제어정책의 적용방안에 대해서 기술한다. 기본적으로 폐쇄망으로 구성된 ScienceDMZ 네트워크 구성에서 안전하게 대용량 데이터를 전송하고 네트워크를 구성하고 있는 자원 및 사용자에 대한 보안방안으로 ACL(Access Control List)을 적용하는 방안에 대해서 기술한다. 실제 KREONET 기반으로 구축된 ScienceDMZ 네트워크 구조에서 접근제어정책을 적용하는 방안에 대해서 기술하고 그 장점에 대해서 기술하였다.

Ⅰ. 서 론

최근 과학연구에 대한 추세는 초대형 실험장비의 방대한 데이터 분석을 통한 자연 현상을 기술하는 4세대 데이터 기반의 과학연구가 추세이다. 점차 많은 데이터 전송량을 요구하고 있지만 국내 연구기관 들의 경우 빅데이터 전송을 위한 전용의 고속의 네트워크가 구축된 사례가 흔치 않은 실정이다. 이러한 문제를 해결하기 위해 고안된 ScienceDMZ는 종단간 전송 효율성을 극대화하기 위해 네트워크, 데이터 전송노드 그리고 기관 내 네트워크 보안 정책, 성능 보장을 위한 네트워크 모니터링 기술 등 복합적인 구성요소들을 최적화하는 기술이다[1, 2]. 본 논문에서는 이러한 ScienceDMZ 기반으로 구축된 네트워크에서 보안성을 확보하기 위한 접근제어정책 적용에 대한 방안에 대해서 기술한다.

Ⅱ. 본론

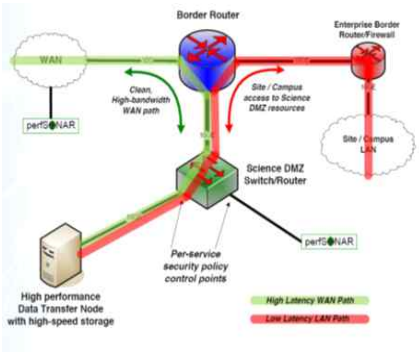

그림 1 ScienceDMZ 네트워크 구성(요약)

그림 1은 ScienceDMZ의 구성을 나타낸다. 논문에서는 ScienceDMZ의 네트워크 구조를 구성하기 위해서는 기존에 사용해오던 네트워크 구조에서 일반 인터넷 망과 연구를 위한 연구망 망을 분리하여 구성해야 한다. 이를 위해 일반 망에 존재하는 기존 보안장비를 구성했던 내부 네트워크에 설치되어 있던 데이터 전송 서버를 Border Router에 그림과 같이 새롭게 연결하여 DTN(Data Transfer Node) 서버를 구성하였을 때 보안장비들을 우회하여 성능이 향상되며, 일반 인터넷 망의 트래픽 혼재를 막을 수 있다.

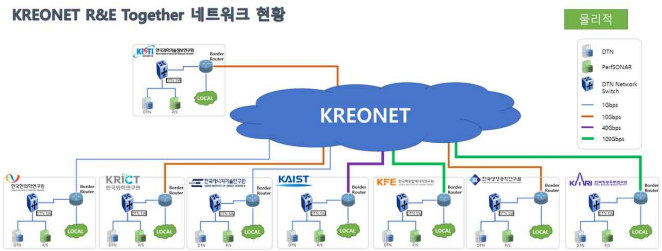

그림 2 ScienceDMZ 기반으로 구축된 네트워크 구성

그림 2는 실제 ScienceDMZ 기반으로 구축된 네트워크 구성을 나타낸다. KISTI를 중심으로 8개의 연구기관(7개 출연연, 1개 교육기관)에 구축된 ScienceDMZ 네트워크이다. 본 논문에서 제안하는 이런 네트워크 구성은 기본적으로 폐쇄망의 성격을 지니고 구성된 모든 기관은 성형(full-mesh) 형태로 구성되어 있으며, 상호간의 데이터 전송 및 자원교환이 이루어지는 구조이다.

ScienceDMZ로 구축된 네트워크의 특징은 과학 연구데이터의 전송구간에는 기존의 방화벽 및 보안을 위한 장비들이 존재하지 않는다. 그 이유는 구축된 네트워크의 대역폭을 최대한 사용하기 위해 망 전송성능을 저하하는 요소인 보안장비들을 우회하여 트래픽이 흐르도록 되어있다. 만약 이러한 네트워크에 접근제어정책이 수립되지 않는 상황에서는 border router에 직접 연결된 DTN(Data Transfer Node)에 외부에서 접속이 가능하여 연구데이터에 대한 유출 및 외부에서의 공격에 대한 대비를 하지 못하는 상황이 발생한다. 이러한 상황을 해결하려면 ScienceDMZ로 구축된 네트워크의 각 엔드포인트에 접근할 IP 및 서비스 port를 사전에 정의하여 각 엔드포인트 방화벽에 적용하여야 한다.

기본적인 접근제어정책은 인가된 서비스 및 사용자들만이 접속을 허용할

수 있도록 구성하는 것이 원칙이며, 그 외 모든 IP Port에 대해 접근을 허용치 않는다. 이러한 방법은 접근제어정책 방법 중 화이트리스트(White list) 방식으로서 접근제어할 노드 및 서비스가 늘어날수록 그 접근제어정책 운용에 대한 오버헤드가 커지는 단점이 있다. 예를 들어 하나의 노드(기관)에서 사용자 그룹이 추가되었다고 가정하면, 네트워크를 구성하는 모든 노드들의 방화벽에 있는 ACL에 해당 사용자그룹의 명세가 추가가 되어야 한다. 원활한 서비스를 위해서는 일괄적으로 모든 구성 노드의 ACL에 업데이트가 적용되어야 하는 단점이 존재한다.

그림 3 접근제어정책 적용에 대한 프로세스

그림 3은 접근제어정책이 전에 노드에 적용될 때 이루어지는 프로세스들에 대해 나타낸다. 모든 노드들에게 접근제어정책이 적용되려면 각 사이트에 있는 관리자들이 수작업으로 이를 업데이트해야 하는데 이는 접근제어정책에 대한 업데이트가 잦을수록, 전체를 구성하는 노드가 많을수록 오버헤드가 점차 늘어난다.

이러한 문제점을 해결하기 위해 본 논문에서는 중앙의 master 노드에서 모든 기관(DTN)에서 사용하는 서비스 port 및 사용자그룹 IP를 수집하여 일괄적으로 모든 기관(DTN)에 적용하는 방법을 적용하였다. master 노드는 모든 기관(DTN)에 접속하여 방화벽의 설정을 변경할 수 있도록 권한을 주고, 서비스 및 사용자 그룹의 변경이 있을 때 모든 기관에 일괄적으로 적용하여 서비스의 중단 및 사용자 그룹의 변경이 용이하도록 하였다.

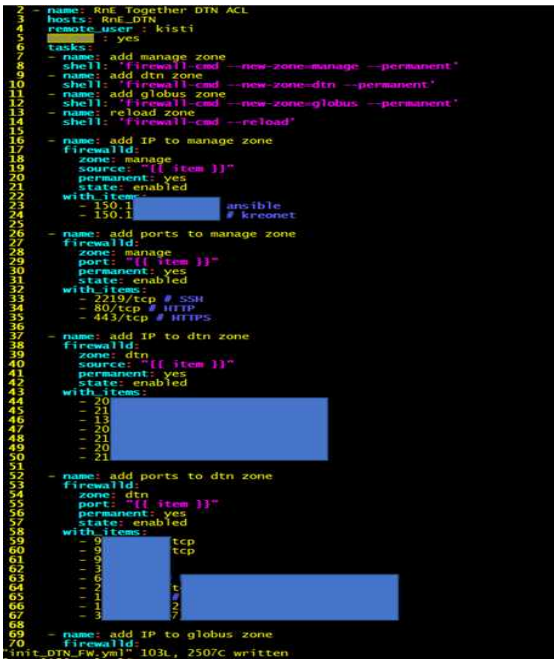

그림 4 Ansible Playbook을 통해 구현된 접근제어정책

본 논문에서는 제안하는 접근제어정책의 수립/배포/적용을 구현하기 위해 RedHat Ansible[3, 4]를 사용하여 구현하였다. Ansible은 IT 자동화 도구이다. 시스템을 구성하고 소프트웨어를 배포한다. 보안과 안정성에 중점을 두고 있으며 부품 이동의 최소화, (다른 전달 수단 및 pull 모드의 대안으로써) 전달을 위한 OpenSSH이 사용, 프로그램에 익숙하지 않은 사람까지도 이해하기 쉽도록 설계된 언어를 포함한다. Ansible은 셸과 같은 기존의 비정형 스크립트를 자동화 관점에서 기능을 모듈화(정형화)하여 쉽게 기능을 구현할 수 있도록 지원하는 언어로서, Python + YAML(YAML Ain't Markup Language)[5] 포맷 기반으로 만들어진 자동화 언어이다.

그림 4는 Ansible를 활용하여 접근제어정책을 구현하기 위해 작성된 playbook의 일부이다. 접근할 노드에서 수행할 task, 접근할 노드의 IP리스트, 제어할 서비스의 port에 대해서 나타내고 있다. 최상위 관리자는 접근제어정책을 Ansible을 통해서 사전에 수립하고 수립된 접근제어정책이 배포 및 적용이 필요할 때 일괄적으로 적용하여 접근제어정책을 운용하는데 있어서 발생하는 오버헤드를 노드의 수와 상관없이 일괄적으로 동기화하여 배포할 수 있었다.

Ⅲ. 결론

본 논문에서는 Ansible를 통해 ScienceDMZ 기반의 네트워크 구성에서 신규 접근 허용 IP 및 port에 대한 정책을 사전에 작성하여 쉽게 배포가 가능하였다. 가장 큰 장점은 중앙집중식으로 최상위 네트워크 관리자가 접근제어정책을 한 번에 적용 및 배포가 가능하다는 점과 이에 따른 모든 노드들에 대한 접근제어정책에 대한 동기화를 이룰 수 있었다. 기존 접근제어정책이 변경될 때마다 각 사이트의 관리자와 협업하여 해결해야 하는 부분에서 자동화가 이루어져 네트워크 운용에 있어 큰 성과가 있었다. 이는 접근제어정책 변경 이벤트가 발생할 때 수동적으로 대응하던 기존에서 항시 사용하지 않는 서비스 및 IP에 대해 상황별로 사전에 접근제어정책 수립 가능한 이점도 있었다.

ACKNOWLEDGMENT

본 논문은 2020년 한국과학기술정보연구원(KISTI) 주요사업 과제로 수행한 연구결과입니다.

참 고 문 헌

[1] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The ScienceDMZ: A network design pattern for data-intensive science," Scientific Programming, vol. 22, no. 2, pp. 173-185, 2014.

[2] ESNET, https://www.es.net/

[3] Red Hat ansible, https://www.redhat.com/

[4] Mohaan, Madhurranjan, and Ramesh Raithatha. Learning Ansible. Packt Publishing Ltd, 2014.

[5] Chatterjee, Rithik. "Red Hat Smart Management and Red Hat Insights." Red Hat and IT Security. Apress, Berkeley, CA, 2021. 149-175.