정 보 보 호 학 회 지
제34권 제3호, 2024. 6

제로트러스트 도입을 위한 SDP 기술 동향

김 미 연*, 이 석 준**, 박 정 수***

요 약

제로트러스트는 전세계적으로 사이버 보안의 패러다임 변화를 가져오고 있다. 이러한 환경에서 세부화 된 워크로드 별로 안전하게 접속하기 위한 방법으로 SDP가 적용되거나 SDP 기술을 활용하는 방법이 소개되고 있다. 본 논문에서는 이러한 SDP의 최근 기술 동향을 분석하기 위하여, SDP v1, v2를 분석, 비교하였으며, SDP가 적용된 기술 논문들을 분석하였다. SDP는 '선 접속 후 인증'의 접근 방식을 가진 기존의 VPN과는 달리 '선 인증 후 접속'의 절차를 거쳐 네트워크에 접근하는 방식으로 경계망 위주의 보안을 제공했던 VPN과 NAC(Netwrok Access Control), 방화벽 등의 솔루션이 해결하지 못하는 보안적 한계를 완화한다. 이러한 SDP 기술을 바탕으로 추후 제로트러스트 환경에서 안전한 네트워크 접속 환경을 위한 방법들을 소개한다.

Ⅰ. 서 론

최근 전 세계적으로 클라우드 도입이 가속화되면서 기존의 고정 경계(Fixed Perimeter) 보안 솔루션은 점차 한계를 드러내고 있다[1]. 특히 모바일을 이용한 업무의 증가 및 클라우드 상용화 등으로 인해 기업의 리소스가 정해진 경계 내/외부에 관계없이 존재하게 되면서 동적으로 대응할 수 있는 보안 솔루션의 필요성이 증가하는 추세이다[1].

미국의 통신산업 관련 기업인 Verizon에서 발간한 2024 Data Breach Investigations Report에 따르면 APAC(아시아 태평양), NA (북미), EMEA (유럽, 중동, 아프리카) 지역에서 발생하는 사이버 사고의 85% 이상이 사회공학(Social Engineering), 시스템 침입(System Intrusion) 등의 공격으로 인한 것이었다[7]. 특히 EMEA 지역의 위협 행위자가 외부(External) 51%, 내부(internal) 49%인 점을 고려했을 때 내부자에 의한 정보 유출도 무시할 수 없는 실정이다[7]. 또한 사이버보안벤처(Cybersecurity Ventures), IBM 등의 단체에서는 사이버보안 사고로 인한 막대한 손실이 계속해서 증가할 것으로 전망하고 있다[8].

이러한 필요성에 의해 등장한 기술이 SDP(Software Defined Perimeter)이다. SDP는 클라우드 보안 협회인 CSA(Cloud Security Alliance)에서 추진하고 있는 보안 프레임워크로 보호가 필요한 민감한 자원은 사전에 인증되지 않은 유저나 장치에 어떠한 방식으로도 노출되지 않고 은닉된다[2][3]. 또한 보안 경계를 동적으로 설정하여 기존 보안 솔루션의 한계를 완화한다. 본 논문에서는 SDP를 이용한 연구를 요약 및 정리하여 SDP 프레임워크에 대한 현재 실정과 향후 연구할 필요성이 있는 주제 및 방향을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 SDP의 정의와 CSA(Cloud Security Alliance)에서 작성하여 배포하고 있는 SDP Specification을 버전별로 정리하고 3장에서 관련된 연구에 관해 기술한 후 4장 결론을 끝으로 마무리한다.

Ⅱ. Background

2.1. SDP(Software Defined Perimeter)

SDP는 미국 국방부(DoD, United States Department of Defense)의 산하 기관인 국방정보시스템국(DISA, Defense Information Systems Agency)의

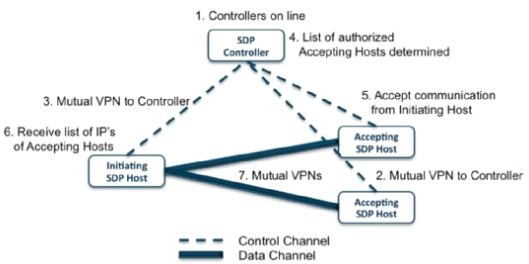
이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2024-00396797, 지능형 오픈랜(Open RAN) 보안 플랫폼 핵심기술 개발)
* 부산외국어대학교 스마트융합보안학과 (학부생, miyeon2002@naver.com)
** 가천대학교 컴퓨터공학부(스마트보안전공) (부교수, junny@gachon.ac.kr)
*** 강남대학교 ICT융합공학부 (조교수, jspark@kangnam.ac.kr)

GIG(Global Information Grid) Black Core 네트워크를 바탕으로 개발된 프레임워크이다[2]. SDP는 애플리케이션이나 인프라에 대한 접근 권한을 부여하기 전에 유저의 신원과 장치를 확인한 후 인증을 수행한다[2]. 인증되지 않은 유저에게는 리소스나 애플리케이션, 서비스와 같은 서비스를 숨겨 노출시키지 않는데 이러한 특성으로 인해 SDP를 '블랙 클라우드(Black Cloud)'라고 하기도 한다[3]. 최근 클라우드 및 모바일 기기의 사용자 수가 급증하게 되면서 SDP의 도입과 전환이 추진되고 있다.

SDP는 '선 접속 후 인증'의 접근 방식을 가진 기존의 VPN과는 달리 '선 인증 후 접속'의 절차를 거쳐 네트워크에 접근하는 방식으로 경계망 위주의 보안을 제공했던 VPN과 NAC(Netwrok Access Control), 방화벽 등의 솔루션이 해결하지 못하는 보안적 한계를 완화한다.

SDP의 주요 컴포넌트인 SDP 클라이언트, SDP 게이트웨이, SDP 컨트롤러로 구성되며 SPA(Single Packet Authorization)를 기반으로 권한이 없는 장치와 트래픽을 거부하여 접근제어를 수행한다[2]. SDP는 Micro-segmentation과 더불어 NIST에서 권장하는 ZTA(Zero Trust Architecture)의 핵심 구현 전략이기도 하다[4].

SDP Architecture는 [그림 1]과 같이 동작한다[10].



〔그림 1〕 SDP Workflow

2.2. SDP Specification_v1.0

2013년 CSA에서 발간한 문서 SDP_Specification_v1.0[10]는 SDP 호환 시스템의 기본적인 아키텍처부터 개념, 워크플로우, 프로토콜 등의 전반적인 내용을 상세하게 다룬다. SDP는 미국 국방부인 DoD(United States Department of Defense, DoD)가 개발하여 일부 연방 기관에서 사용하고 있던 기술을 CSA를 중심으로 표준화시켜 상용화한 것[1]으로 암호화 프로토콜에 대한 NIST의 지침을 준수하여 동작한다. SDP는 보안이 되지 않은 모든 네트워크로부터 격리되어 신뢰성을 가진 네트워크로 Air-gap 네트워크를 제공하여 네트워크 공격으로 인해 발생할 수 있는 피해를 최소화하도록 설계되었다.

SDP는 Control Plane과 Data Plane의 2개의 섹션으로 나뉘며, SDP에는 IH(Initiating Host), AH(Accepting Host), Controller로 3가지의 주요 컴포넌트가 존재한다. 경우에 따라 IH는 클라이언트, AH를 게이트웨이라고 지칭하는 문서도 있다[5]. SDP의 섹션 중 하나인 Control Plane은 IH(Initiating Host)와 AH(Accepting Host)가 컨트롤러와 통신하는 것을 일컫고, Data Plane은 IH와 AH가 상호 간에 통신하는 것이다. 또한 SDP 컴포넌트의 Controller는 SDP 프레임워크의 핵심 요소 중 하나로 IH와 백엔드 보안 컨트롤 사이에서 브로커 역할을 수행하여 주고 받는 모든 컨트롤 메시지를 담당한다. Controller는 IH가 AH에서 접근할 수 있는 권한을 부여받은 서비스를 결정하고 IH와 AH를 실시간으로 구성하여 상호 TLS 터널을 생성하는 데 기여한다. IH(Initiating Host)는 Controller와 통신하여 접속이 가능한 AH의 리스트를 요청한다. 이때 Controller는 하드웨어나 소프트웨어의 정보를 IH에 요청할 수 있다. 이전에 발급된 인증서를 통해 인증이 완료되면 IH를 권한이 있는 서버 또는 애플리케이션에 연결할 수 있는 상호 TLS 터널이 생성된다. IH는 서버나 애플리케이션에 인증이 완료되어야 접근할 수 있으므로 접근제어 기능을 향상시키는 데 도움이 될 수 있다. AH(Accepting Host)는 인증된 서비스나 애플리케이션을 수락하는 장치로 기본적으로는 Controller를 제외한 모든 호스트로부터의 통신을 거부하고 차단하는 역할을 수행한다.

SDP는 보안되지 않은 네트워크로부터 애플리케이션 소유자가 제공하는 서비스를 격리하기 위해 필요한 기능을 제공하기 위한 것이며, 장치와 유저를 확인한 후 애플리케이션 인프라에 대한 액세스를 제공하여 모든 유형의 서버를 네트워크를 기반으로 한 공격으로부터 보호할 수 있다. SDP로 보호되는 서버를 만들기 위해서는 VPN 게이트웨이로 보호되는 서버와는 또 다른 노력이 필요하며, SDP의 경우 Controller가 온라인 상태가 되면 사용자는 소프트웨어를 통해 보호된

서버를 구축하는 것이 가능하다. 특히 SDP는 사용자가 네트워크상에 조직, 구성원, 파일 등과 같은 데이터를 찾는 데 도움을 제공하는 프로토콜인 LDAP (Lightweight Directory Access Protocol) 연결을 통해 인증된 사용자와 그렇지 않은 사용자를 구별하는 것이 가능하다. VPN을 사용하는 것은 SDP를 사용하여 서버를 보호하는 것보다 비용이 많이 들고, SDP는 클라우드 환경에 배포할 수 있는 소프트웨어 구조로 설계되었다. 또한 SDP는 보안 및 원격 액세스를 동시에 사용하는 것이 가능하다는 점이 VPN과의 차이점이라고 할 수 있고, SDP는 대역폭(bandwidth) 서비스 거부 공격으로부터 서버를 보호하는 것이 가능하지만, VPN은 보호할 수 없다. 특히 AH는 보호 중인 애플리케이션 서버와는 다른 위치에 배포하는 것이 가능하기 때문에 권한이 부여된 사용자에게도 실제 위치를 숨길 수 있어 보안에 특화되어 있다는 장점이 존재한다.

SDP의 주요 프로토콜인 SPA(Single Packet Authorization)는 SDP의 핵심 중 하나인 인증을 위해 사용되는 프로토콜이다. SPA는 Controller나 게이트웨이 등과 같은 시스템 구성 요소가 네트워크에 접근하는 것을 허용하기 전에 시스템 장치나 사용자 신원을 검증하기 위해 사용된다[6]. SPA를 사용하면 서버는 클라이언트가 인증된 SPA를 제공할 때까지 클라이언트 연결에 응답하지 않는다. 또한 https 프로토콜을 사용하는 인터넷 서버는 DoS 공격에 매우 취약한데 SPA는 서버가 가짜 패킷을 처리할 때 리소스를 사용하는 것을 방지하기 위해 TLS handshake를 입력하기 전 TLS DoS 시도를 무시하여 DoS 공격을 완화할 수 있다[6]. 특히 다른 호스트에서 AH로 보내지는 첫 패킷은 SPA여야 하며, SDP는 단일 악성 패킷을 기반으로 하여 공격 여부를 결정하기 때문에 AH가 SPA 패킷이 아닌 다른 패킷을 수신하면 공격으로 간주한다. 모든 호스트끼리의 연결은 SDP의 인증된 구성원으로서 장치를 검증하기 위해 상호 인증을 지원하는 TLS(Transport Layer Security)/IKE(Internet Key Exchange)를 사용해야 한다. 이때 상호인증을 지원하지 않거나 약한 암호 집합은 차단된다. SDP에서는 IH, AH, Controller와 같은 컴포넌트 사이의 통신이 상호 검증된 TLS 암호화되며, 이를 통해 각 컴포넌트들은 서로를 검증하는 것이 가능해 시스템의 무결성이 보장된다[6]. 장치 유효성 검사(Device Validation)의 목적은 적절한 장치의 개인키 보유 여부와 장치에서 동작중인 소프트웨어를 신뢰할 수 있는지를 확인하기 위한 것이다. 장치 유효성 검사를 통해 자격 증명 도용과 사칭 등과 같은 공격을 완화할 수 있다.

SDP_Specification_1.0에서 제시하는 SDP 배포모델은 Client-to-gateway, Client-to-Server, Server-to-Server, Client-to-Server-to-Client로 5개가 있으며, 상황이나 환경을 고려해 문제해결에 가장 유리한 배포모델을 선택해서 사용해야 한다.

이 밖에도 SDP_Specification_1.0에서는 각종 프로토콜 및 로깅 등과 같이 SDP와 연관된 전반적인 내용을 상세하게 설명하고 있다.

2.3. SDP Specification_v2.0

2022년 10월에 발표된 SDP_Specification_2.0는 CISA가 SDP_Specification_1.0에서 언급한 내용들을 확장 및 수정하여 갱신한 문서이다. 2013년에 SDP_Specification_1.0이 발표된 이후, 코로나 19 팬데믹으로 인한 재택 근무 방식의 확대로 인해 제로 트러스트에 대한 관심과 기업의 도입이 증가함에 따라 SDP_Specification_2.0에서는 SDP 기반 솔루션에 상응하는 제로 트러스트 원칙도 함께 언급하고 있다. SDP Controller는 제로 트러스트의 정책 결정 포인트인 PDP(Policy Decision Point)의 역할을 수행하며, 모든 인증과 액세스 플로우를 관리한다. 즉, 액세스 정책을 정의하고 평가하기 위한 SDP 솔루션의 ‘두뇌’와 같은 역할을 한다.

SDP Controller는 기업의 인증 솔루션 등과 연계하여 인증 및 인가의 오케스트레이션을 담당하고, 정의된 액세스 정책에 의해 허가된 접속을 가시화하며, 제어 포인트의 역할도 수행한다. SDP Controller는 한 ID(user, group)가 조직의 서비스에 접근 가능한가에 대한 정보를 관리하며, 어떤 호스트가 상호 간에 통신을 할 수 있는지 결정한다. IH상의 사용자가 Controller에 접속하면, Controller는 사용자를 인증하여 ID나 디바이스의 속성을 포함하는 사용자 컨텍스트에 따라 사용자에게 허가된 서비스에 한해 액세스 권한을 부여한다. SDP Controller가 사용자를 인증하기 위해서는 내부 사용자 테이블을 사용하거나 3rd party의 IAM(Identity and Access Management)에 접속하거나 다중 요소 인증(Multi-Factor Authentication, MFA)을 적용할 수 있다. SDP Controller는 SPA 프로

토콜을 사용하는 격리 메커니즘에 의해 보호되고 승인되지 않은 사용자 및 디바이스로부터 은폐되어 접근할 수 없다. 해당 메커니즘은 Controller 앞에 SDP 게이트웨이를 설치하거나 Controller 내에 자체적으로 제공될 수 있다. 이처럼 SDP Controller는 NIST가 제로트러스트의 이념으로 주장하는 동적 제로 트러스트 정책 유형을 실현할 수 있게 하며, SDP 아키텍처에 따라서 클라우드나 온프레스에 배치할 수 있다.

SDP 클라이언트인 IH는 AH로 보호되는 기업의 리소스에 접근하는 프로세스를 시작하기 위해 SDP Controller와 통신한다. 따라서 Controller는 인증 단계에서 IH가 사용자 ID, 장치 상태와 같은 정보를 제공하도록 요구한다. Controller는 IH가 AH와의 보안 통신을 설정할 수 있는 메커니즘도 제공해야 한다. IH의 형태는 end-user의 컴퓨터 또는 웹에 설치된 클라이언트 응용 프로그램일 수 있으며, 이를 사용할 경우 디바이스의 상태를 확인하고 간소화된 인증 등의 많은 기능을 제공한다. 특히 IH의 핵심 역할은 SPA를 통해 접속을 시작하는 것으로 SPA 패킷은 브라우저 기반 SDP 클라이언트 즉, IH에 의해 구현될 수 있다.

AH는 제로 트러스트의 정책 시행 지점인 PEP(Policy Enforcement Point)로서 기능하고, SDP의 전용 소프트웨어 또는 하드웨어로 구현이 가능하다. AH는 SDP Controller의 지시에 따라 대상 서비스에 대한 네트워크 트래픽을 허가하거나 거부하며, ID가 연결할 수 있는 모든 리소스 및 서비스에 대한 접근을 제어한다. AH에 대한 모든 네트워크 액세스는 디폴트로 차단되며, 인증된 ID에 한해서만 접근할 수 있다. AH는 Controller에서 제어 정보를 수신하고 Controller의 지시를 받은 경우에만 IH로부터의 접속을 승인한다. AH는 SDP Controller로부터 수신한 정보를 통해 인가된 IH에만 보호된 서비스에 대한 액세스를 제공한다. AH는 IH로부터 트래픽 정보를 받아 보호된 백엔드 서비스로 중계하기 위한 교환기 역할 수행하고 백엔드의 응답을 다시 IH로 전달한다.

SDP_Specification_2.0에서 제시하는 SDP 배포 모델은 SDP_Specification_1.0에서 제시한 모델에 Gateway-to-Gateway 모델이 추가되었다. 해당 모델은 IoT 환경에서 적합한 것으로 이 모델의 게이트웨이는 경계 내에 은폐된다. 또한 SDP_Specification_2.0에서는 SDP_Specification_1.0과는 다르게 SPA 프로토콜에 대한 내용이 보다 상세하게 언급된다. IH-Controller, AH-Controller, IH-AH의 통신을 시작하며, SPA 패킷의 구현 방식에 TCP나 UDP 프로토콜을 사용해 시작한다. SPA 메시지 포맷 또한 보안 및 복원력 향상을 위해 SDP_Specification_1.0에서 언급되었던 것보다 상세하게 업데이트되었다. SDP_Specification_2.0에서는 SDP 핵심 메커니즘으로 SPA만을 제시하는 SDP_Specification_1.0과는 다르게 SPA가 SDP의 핵심 원칙을 달성하기 위한 유일한 메커니즘은 아니라고 언급한다.

Ⅲ. 관련 연구

Moubayed, et al[2].의 논문에서는 엔터프라이즈 내부 시나리오에 대한 가상화된 네트워크 테스트베드를 이용해 SDP Implementations 중 하나인 Client-to-gateway 아키텍처를 채택하여 SDP 프레임워크를 제안한다. 해당 연구를 위해 최신 네트워크 데이터 및 서비스 가용성을 위협하는 DDoS 공격과 데이터 프라이버시를 위협하는 포트 스캐닝 공격이 선택되었다. 이러한 두 가지는 SDP 프레임워크가 본질적으로 해결하고자 하는 네트워크상의 위협을 대표하기 때문이다. 본 논문에서는 제안된 프레임워크의 성능을 평가하기 위해 연결 설정 시간과 전체 네트워크 처리량을 성능 지표로 한다. 먼저 네트워크 처리량을 평가하기 위해 하나의 VM은 권한이 있는 클라이언트로 설정하고 또 다른 VM은 권한이 없는 클라이언트로 설정하여 DDoS 공격을 시뮬레이션했다. 평가 결과 SDP가 채택된 네트워크의 연결 설정 시간은 SPA 패킷을 복호화하는 과정과 검증 프로세스 및 클라이언트 간의 연결을 위한 방화벽 설정 단계가 필요하므로 초기 연결을 설정하는 데는 비교적 많은 시간이 소요되었다. 반면 SDP를 사용하지 않은 경우는 이러한 단계가 생략되므로 연결 프로세스는 훨씬 단축된다. 해당 논문의 실험에서는 TCP의 사용 가능한 대역폭을 2Gbps, UDP의 사용 가능 대역폭을 10Mbps로 유지하여 TCP와 UDP 트래픽을 모두 전송했다. 그 결과 TCP 사용 관점에서 SDP를 사용하지 않았을 때의 평균 데이터 처리량은 사용 가능 대역폭의 약 0.343%까지 떨어졌지만, SDP를 사용한 경우는 약 75.5%로 나타났다. 마찬가지로 UDP 관점에서도 SDP를 사용하지 않았을 때는 약 0.017%로 떨어졌지만 SDP를 사용한 경우는 약 88% 범위였다. 이는 SDP가 DDoS 공격으로부터 네트워크를 보호하는 데 있어서 상당히 효과적

이라는 것을 강조하는 객관적 지표가 된다. 또한 SDP가 없는 네트워크의 성능은 DDoS 공격의 영향을 받아 데이터 전송이 한 번에 이뤄지지 않았다. 즉, SDP 없이 TCP를 사용한 경우의 평균 데이터 처리량은 약 6.86Mb/s로 실제 사용 가능한 2Gb/s 대역폭에 비해 훨씬 떨어지는 수치였다. 그러나 SDP를 도입했을 때 데이터 전송은 완전히 성공했고 평균 대역폭은 1.51Gb/s에 달했다. UDP의 경우에도 SDP 없이는 1.7kb/s의 데이터 처리량을 기록한 반면 SDP를 도입했을 때는 8.8 Mbp/s의 데이터 처리량을 기록했다. 따라서 네트워크 처리량 측면에서 봤을 때 SDP를 채택하는 것은 초기 설정을 위해 비교적 많은 시간이 소요된다는 한계가 있지만 SDP를 채택하지 않았을 때보다 DDoS 공격에 효과적으로 대응할 수 있고 데이터 처리량을 높일 수 있다. 다음으로 시뮬레이션한 공격은 포트 스캐닝 공격이다. 이 공격은 도구를 통해 네트워크 호스트와 서비스를 검색하는 능동적인 공격이며, 사용할 수 있는 모든 포트를 검사하고 나열하는 것도 포함된다. 무료로 배포되는 포트 스캐닝 유틸리티 중 하나인 nmap을 사용해 공격을 실시한 결과 SDP가 없는 스캔 보고서에는 열려 있는 SSH의 수와 포트에서 실행되는 서비스에 대한 정보가 분명하게 표시되었지만 SDP가 있는 스캔 보고서에서는 열려 있는 포트 수와 실행되는 서비스에 대한 정보가 전혀 제공되지 않았다. 이 결과를 통해 SDP가 채택되면 애플리케이션 보안이 강화되고 권한이 없는 클라이언트의 접근이 불가능하다는 것을 확인할 수 있다. 이처럼 SDP는 DDoS와 포트 스캐닝 공격에 탄력적으로 대응할 수 있으며, 민감한 정보를 제공하지 않고도 네트워크 처리량을 평균적으로 유지하여 보안 프레임워크로써의 잠재력을 확고히 한다. 그러나 본 논문의 저자는 SDP가 충분히 주목받을 만한 유망한 프레임워크임에도 불구하고 아직 개척되지 않은 연구 분야가 훨씬 더 많다고 언급했다. 또한 NFV(Network Functions Virtualization), SDN(Software-Defined Networking) 등과 같이 미래 네트워크에서 중요한 역할을 수행할 것으로 예측되는 패러다임들과 통합시킬 수 있는 방안을 탐색하는 것이 필수적이라고 주장한다.

Tanimoto, Shigeaki, et al[9]. 의 논문에서는 SDP의 확장성, 신뢰성, 사용성 등의 한계를 극복하기 위해 구조가 다른 조직의 네트워크 더 쉽게 설치하고 관리할 수 있도록 PKI(Public Key Infrastructure) 신뢰 모델을 기반으로 하여 몇 가지 확장 기능을 갖춘 SDP 모델을 제안하고 있다. Single CA model, Hierarchical model, Web model, Mesh model, Bridge CA model의 5가지 PKI trust model 중 SDP의 확장성을 고려하여 3가지 모델(Hierarchical model, Mesh model, Bridge CA model)을 선택해 사용했으며, Hierarchical model와 Bridge CA model를 결합시켜 Hybird model을 개발하여 총 4가지 모델로 연구를 진행했다. 해당 논문에서는 4가지 모델에 대한 정성적 평가와 정량적 평가를 진행했는데 주관적 판단을 근거로 하는 정성적 평가에서는 확장성, 설치용이성, 비용, 관리용이성, 확장성 오버헤드를 평가 항목으로 정하고 문헌 고찰 및 저자 간의 논의를 통해 높음, 중간, 낮음의 3가지 수준으로 평가가 이루어졌다. 정성적 평가 결과 확장성, 설치용이성, 관리용이성이 높음으로 평가되고, 비용과 확장성 오버헤드가 낮음으로 평가된 Bridge model이 어떠한 조직에서도 합리적으로 적용될 수 있는 것으로 나타났다. 정량적 평가를 위해서는 각 모델의 sequence flow를 기반으로 제로트러스트 모델에서의 인증부터 최종적으로 호스트와 연결되는 데까지 필요한 평균 신호 수를 사용하였다. 평가 결과 가장 낮은 신호 수를 나타내는 모델은 Mesh model이었지만, 앞서 실시한 정성적 평가의 결과를 고려할 때 mesh model은 최적화 된 모델이라고 보기는 어렵다. 정량적 평가 결과의 차선책은 bridge model의 short path로 정성적/정량적 평가의 결과를 모두 참조했을 때 bridge model이 가장 적절한 모델이라고 언급한다. 즉, 제안된 4가지 모델 모두 실제 네트워크의 확장 가능한 설치와 관리를 지원하므로 엔터프라이즈의 특성이나 요구에 따라 보다 적합한 SDP 모델을 선택할 수 있다.

Omar et al[3].의 논문에서는 결과를 신뢰할 수 있도록 인증, 암호화, 세분화된 엑세스, 가시성, 네트워크 인프라 단순성, 공격 표면 최소화의 7가지 요인을 채택하여 NAC과 SDP의 비교를 수행한다. 시간이 지날수록 지능적이고 지속적인 사이버 공격이 광범위하게 발생됨에 따라 접근제어 방식의 중요성은 더욱 주목받고 있다[12].

전통적인 네트워크 접속 제어 방식인 NAC은 요청자(NAC Agent), 인증자(NAC 시행 지점), 인증 서버(NAC 서버)로 구성되어 있다. 인증자는 요청자(네트워크 리소스에 대한 접근을 요청하는 사용자 디바이스로 간주)의 요청을 수신하여 NAC 서버로 알려진 기존

의 인증 서버에 전달한다. NAC의 주요 기능은 인증과 권한 부여이지만, 요청자의 상태가 정상인지 확인하고 규정과 서비스에 대한 정책을 충족하지 않는 디바이스를 격리하는 등의 업무를 수행할 수 있다.

반면 SDP는 네트워크 접속 보안을 위한 접근 방식으로 사용자 인증이 완료된 경우에만 필요한 인프라나 리소스에 대한 접근이 가능하다. 이러한 방식의 달성은 SDP 컨트롤러, SPA 프로토콜, mTLS, 동적 방화벽과 같은 네 가지 요소에 의해 이루어진다. NAC와 SDP 모두 다중 인증(MFA, Multi-factor authentication)를 지원하지만 SDP의 SDP 컨트롤러는 인증의 수립을 담당하는 주축이 되며 리소스를 노출시키기 전에 사용자를 인증할 수 있기 때문에 상호 인증 측면에서 우수하다. 또한 SDP 시스템에서 데이터 트래픽과 같은 모든 리소스는 내/외부 사용자에 관계없이 전부 암호화된다. 데이터 암호화를 기준으로 봤을 때 NAC을 통한 접근제어 방식은 스위치, 방화벽과 같은 다른 장비와 통합되어야 실현이 가능하며, 이에 따른 관리와 비용은 네트워크 유지보수 부담을 증가시키게 된다.

기존의 접근제어 시스템은 접근제어 목록(ACL, Access Control List)을 이용해 데이터 흐름에 태그를 지정하여 접근을 제어하는 방식으로 발전했다. 그러나 SDP는 패킷 수준에서 세분화하여 접근을 제어하므로 측면 이동의 기능을 제한한다는 점에서 혁신적인 기술로 평가받는다. 또한 기존의 NAC은 IDS 또는 방화벽과 협력하면 내부 네트워크 모니터링이 가능했지만 클라우드 데이터 모니터링에 대해서는 여전히 한계가 존재했다. 그러나 SDP는 내부 네트워크나 클라우드에 상관없이 유연한 데이터 모니터링이 가능하다.

따라서 SDP는 인증되지 않은 사용자에게 중요한 자산을 숨기기 때문에 포트 스캐닝, DDoS, 중간자 공격(MITM, Man In The Middle)과 같은 위협을 대응하는 데 있어 우수하며, 데이터 암호화 및 상호 인증, 유지보수 비용 등의 측면에서 고려했을 때 SDP는 기존의 NAC을 이용한 접근제어 방식보다 유연하고 효율적인 솔루션임을 확인할 수 있다.

Ⅳ. 결 론

본 논문에서는 점차 증가하는 사이버보안 위협에 대응하기 위해 SDP 솔루션에 대해 다룬다. SDP는 기존의 NAC, 방화벽 등이 가진 고정 경계 방식의 한계를 완화하고 주요 자산에 대한 접근을 동적이고 유연하게 제어할 수 있기 때문에 그 중요성은 더욱 강조된다.

특히 SDP는 유지보수, 관리, 비용적 측면에서 매우 효율적인 솔루션이며, 허가되지 않은 사용자에게 중요한 자산을 절대로 노출시키지 않기 때문에 각종 사이버보안 위협으로 인해 발생하는 막대한 손실을 막을 수 있다.

또한 SDP의 유연한 특성으로 인해 상황과 환경에 적합한 SDP 모델을 채택하여 사용할 수 있고, 미래 네트워크 분야에서 다양한 업무를 수행할 패러다임과 결합시켜 그 효율을 더욱 극대화할 수 있다.

그러나 앞에서 언급했던 Moubayed, et al[2].의 논문의 내용처럼 SDP는 여러 측면에서 유망한 프레임워크로 주목받고 있고, 많은 기업에서 SDP의 도입을 고려하고 있는 상황이지만 SDP에 대한 연구와 도입 속도는 아직도 현저히 느린 실정이다.

따라서 SDP의 응용 및 연구를 가속화하여 기존의 접근제어 방식이 가지고 있던 한계를 완화해야 하고, 미래 네트워크의 접근제어가 보다 효율적이고 체계적으로 이루어질 수 있도록 논의되어야 하는 시점이다.

참 고 문 헌

[1] 박승규. “Cloud 및 IoT 시스템의 보안을 위한 소프트웨어 정의 경계기반의 접근제어시스템 개발.” 한국인터넷방송통신학회 논문지 21.2 (2021): 15-26.

[2] MOUBAYED, Abdallah; REFAEY, Ahmed; SHAMI, Abdallah. Software-defined perimeter (sdp): State of the art secure solution for modern networks. IEEE network, 2019, 33.5: 226-233.

[3] Omar, Rami Radwan, and Tawfig M. Abdelaziz. “A comparative study of network access control and software-defined perimeter.” Proceedings of the 6th International Conference on Engineering & MIS 2020. 2020.

[4] Syed, Naeem Firdous, et al. “Zero trust architecture (zta): A comprehensive survey.” IEEE Access 10 (2022): 57143-57179.

[5] 정진교, 이상구, and 김용민. “소프트웨어 정의 경계의 단일 패킷 인증 및 네트워크 접근통제 보안관리 개선.” 한국콘텐츠학회논문지 19.12 (2019):

407-415.

[6] Sintaro, Abel Tariku, and Yemi Emmanuel Komolafe. “SDP And VPN For Remote Access: A Comparative Study And Performance Evaluation.” (2021).

[7] S. Widup, A. Pinto, D. Hylender, G. Bassett, and P. Langlois, “DBIR 2024 Data Breach Investigation Report,” 2024.

[8] Hakim, Arif Rahman, et al. “A novel digital forensic framework for data breach investigation.” IEEE Access (2023).

[9] Tanimoto, Shigeaki, et al. “Suitable Scalability Management Model for Software-Defined Perimeter Based on Zero-Trust Model.” International Journal of Service and Knowledge Management 7.1 (2023).

[10] B.Bilger, A. Boehme, Z. Guterman, et al. “SDP Specification 1.0”, 2014.

[11] J.Garbis, J. Koilpillai, et al. “SDP Specification 2.0”, 2022.

[12] Lakbabi, Abdelmajid, Ghizlane Orhanou, and Said El Hajji. “Network Access Control Technology-Proposition to contain new security challenges.” arXiv preprint arXiv:1304.0807 (2013).

〈저자소개〉

김 미 연 (Miyeon Kim)

2021년 3월~현재 : 부산외국어대학교 스마트융합보안학과 학사과정

<관심분야> 제로 트러스트, 클라우드 보안, SDP

이 석 준 (Sokjoon Lee)

종신회원

1998년 2월 : 서울대학교 컴퓨터공학과 졸업

2000년 2월 : 서울대학교 컴퓨터공학과 석사

2019년 8월 : 한국과학기술원 전산학과 박사

2000년 2월~2022년 2월 : 한국전자통신연구원 정보보호연구본부 책임연구원

2022년 3월~현재 : 가천대학교 IT융합대학 컴퓨터공학부 스마트보안전공 교수

<관심분야> 암호 양자분석, 암호 엔지니어링, 제로트러스트, 위험관리

박 정 수 (Jungsoo Park)

종신회원

2013년 2월 : 숭실대학교 정보통신전자공학부 졸업

2015년 2월 : 숭실대학교 전자공학과 석사

2021년 8월 : 숭실대학교 융합소프트웨어학과 박사

2021년~2022년 : 부산외국어대학교 조교수

2024년~현재 : 강남대학교 교수

<관심분야> 모바일 보안, 클라우드 보안, AI 보안, 악성코드 분석, 제로트러스트