Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.6, Dec. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.6.1181

SDP 환경에서 SVDD 기반 이상행위 탐지 기술을 이용한 디바이스 유효성 검증 방안*

이 희 웅,[1†] 홍 도 원,[2‡] 남 기 효[3]
[1,2]공주대학교 (대학원생, 교수), [3]주식회사 유엠로직스 (부사장)

A Method of Device Validation Using SVDD-Based Anormaly Detection Technology in SDP Environment*

Heewoong Lee,[1†] Dowon Hong,[2‡] Kihyo Nam[3]
[1,2]Kongju National University (Graduate student, Professor),
[3]UMLogics Co., Ltd. (Vice President)

요 약

팬데믹 현상은 원격으로 문제를 해결할 수 있는 비대면 환경을 빠르게 발전시켰다. 하지만 급작스러운 비대면 환경으로 전환은 다양한 부분에서 새로운 보안 이슈들을 발생시켰다. 새로운 보안 이슈들 중 하나가 내부자에 의한 보안 위협이었고 이를 방어하기 위한 기술로 제로 트러스트 보안 모델이 다시 주목받게 되었다. SDP(Software Defined Perimeter) 기술은 다양한 보안 요소로 이루어져 있는데 이 중 디바이스 유효성 검증이라는 기술이 내부자의 사용 행위를 모니터링 하여 제로 트러스트 보안 모델을 실현할 수 있는 기술이다. 하지만 현재 SDP 명세서에는 디바이스 유효성 검증을 수행할 수 있는 기술이 제시되어 있지 않다. 따라서 본 논문에서는 SDP 환경에서 사용자 행위 모니터링을 통한 SVDD 기반 이상행위 탐지 기술을 이용해 디바이스 유효성 검증 기술을 제안하고 성능 평가를 진행하여 SDP 환경의 디바이스 유효성 검증 기술을 수행할 수 있는 방안을 제시한다.

ABSTRACT

The pandemic has rapidly developed a non-face-to-face environment. However, the sudden transition to a non-face-to-face environment has led to new security issues in various areas. One of the new security issues is the security threat of insiders, and the zero trust security model is drawing attention again as a technology to defend against it.. Software Defined Perimeter (SDP) technology consists of various security factors, of which device validation is a technology that can realize zerotrust by monitoring insider usage behavior. But the current SDP specification does not provide a technology that can perform device validation.. Therefore, this paper proposes a device validation technology using SVDD-based abnormal behavior detection technology through user behavior monitoring in an SDP environment and presents a way to perform the device validation technology in the SDP environment by conducting performance evaluation.

Keywords: ZeroTrust, Non-face-to-face Environment, Software Defined Perimeter, SVDD, Device Validtaion

Received(09. 27. 2021), Modified(11. 24. 2021), Accepted(11. 25. 2021)

* 본 연구는 2019년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2019R1A2C1003146).

† 주저자, leehw@smail.kongju.ac.kr
‡ 교신저자, dwhong@kongju.ac.kr(Corresponding author)

I. 서 론

코로나 바이러스로 인한 팬데믹 현상은 세계적으로 수많은 사상자를 발생시켰으며 인류에게 많은 변화를 초래했다. 크게 사회, 문화, 경제뿐만 개인의 일상생활까지 기존의 방식에서 벗어나야 했기 때문에 팬데믹을 극복하기 위한 다양한 방안들이 나타났다[1]. 특히 5G를 이용한 초연결시대를 살고 있는 우리는 PC, 모바일 그리고 IoT 기기를 이용하여 스마트 홈, 화상 회의, 원격 업무처리 등 일상생활에서도 개인과 개인이 직접 만나지 않고 비대면으로 다양한 업무를 처리할 수 있게 되었다. 하지만 대부분의 비대면 서비스는 사용자의 편의성만 극대화하고 보안 요구 사항은 만족하지 못한 채로 운영되고 있다. 따라서 비대면 시대의 신 보안 위협으로 사이버 보안, 가짜 뉴스, 융합 보안 등 다양한 유형의 위협들이 나타나게 되었고 편리함에 대한 부작용으로 새로운 유형의 보안 이슈가 지속적으로 생성되고 있다[2].

비대면 서비스가 활성화되기 전 기존의 보안 모델은 다양한 방어기제를 설치하여 외부의 침입자에 대해서만 경계하기 때문에 내부의 이용자들에 의한 행위는 제재하지 않았다. 하지만 비대면 서비스에서는 공격자들이 피싱 공격과 같은 전통적인 방법으로 내부 접속 권한을 획득하게 되면 인가된 사용자 인척 행동하며 내부정보 유출, 백도어 설치 등 소극적, 적극적 공격을 모두 실행할 수 있기 때문에 인가된 사용자에 대한 모니터링 기술의 필요성이 대두되었다[3]. 이에 대응하기 위해 부각된 기존 기술이 바로 제로 트러스트 보안 모델이다. 제로 트러스트 보안 모델은 '아무것도 신뢰할 수 없다'라는 무신뢰 원칙에서 시작하며 이용하고자 하는 서비스의 내부와 외부를 가리지 않고 정당한 인증 절차를 거쳐야 원하는 정보에 대한 접근 권한이 주어지는 모델이다[4]. 이러한 제로 트러스트 보안 모델을 실현할 수 있는 기술 중 하나로 CSA(Cloud Security Alliance)에서 개발한 SDP(Software Defined Perimeter)가 있다. SDP는 신원을 기반으로 리소스에 대해 접근제어를 수행하는 프레임워크로 네트워크 장치, 단말기, 사용자 정보를 체크하여 서비스 접속에 대한 인가 여부를 결정하게 된다. 이 SDP 기술에는 사용자 인증을 위한 SPA(Single Packet Authentication), 보안 통신을 위한 mTLS(mutual TLS), 접근제어를 위한 동적 방화벽, 다양한 서비스에 쉽게 적용할 수 있도록 하는 구현 편의성 등이 모두 제공되고 제로 트러스트를 실현할 수 있는 항목인 디바이스 유효성 검증도 포함되어 있지만 이 기술을 실현할 수 있는 기술이 명세화되어 있지 않다. 따라서 본 논문의 2장에서는 SDP 환경과 각 요소들에 대한 설명 및 SDP가 갖춰야 할 특징을 서술하고, 3장에서 디바이스 유효성 검증 기술의 필요성을 설명한다. 4장에서는 디바이스 유효성 검증 기술에 접목할 SVDD(Support Vector Data Description) 알고리즘에 대한 설명 후 5장에서 SVDD를 SDP에 적용하여 인가된 사용자와 비인가된 사용자를 구별할 수 있는 디바이스 유효성 검증 기술을 구현 및 평가하고 6장에서 결론을 짓는다.

II. SDP(Software Defined Perimeter)

2.1 SDP의 개념

SDP는 2007년 GIG(Global Information Grid) Black Core Network 이니셔티브에 따라 DISA(Defense Information System Agency)에서 수행한 작업에서 발전된 컴퓨터 보안 구조이다. CSA(Cloud Security Alliance)에서는 신원을 기반으로 접근제어를 수행하는 SDP 프레임워크를 개발하였다. SDP에서는 네트워크 장치, 단말기의 상태, 사용자의 ID를 체크하여 권한이 있는 사용자 및 디바이스에 대해서만 액세스 권한을 부여하고 인증받지 못한 단말기에 대해서는 그 어떠한 서비스 연결 정보도 얻지 못하게 된다. 따라서 인증되기 전에는 DNS 정보나 IP 주소를 알 수 없기 때문에 '블랙 클라우드'라고 불리며 공격자들도 쉽게 보안을 뚫을 수 없도록 구성되어 있다[5]. 이러한 SDP는 애플리케이션 소유자가 보안되지 않은 네트워크에서 서비스를 분리하기 위해 소프트웨어적인 경계를 배치할 수 있는 기능을 제공하는 것이 목표이다. SDP를 이루는 기술들은 완전히 새로운 것은 아니지만 기존 네트워크 구조의 이점을 유지하면서도 원격 접근 게이트웨이 어플라이언스가 필요하다는 단점을 없앴다. 또한 SDP는 보호된 서버에 접근하기 전에 서비스를 먼저 인증하고 인증받아야 하며 인증이 완료되면 시스템과 애플리케이션 간에 암호 통신이 실시간으로 생성되어 이용자가 안전하게 서비스를 이용할 수 있게 된다[6].

2.2 SDP의 구조[7]

기본적인 SDP 구조는 SDP Host와 SDP Controller의 두 가지 요소로 구성되어 있다. Fig.1.에서 볼 수 있듯이 SDP Host는 SDP Controller와 보안 채널을 이용해 연결을 시작하는 Initiating SDP Host와 연결을 수락하는 Accepting SDP Host 역할로 나누어져 수행한다.

Fig.1.에서 각각의 요소들에 대한 설명은 다음과 같다.

- SDP Controller : SDP Controller는 서로 통신할 수 있는 SDP 호스트를 결정하고 외부 인증 서비스에 호스트에 대한 정보를 전달할 수 있다.
- Initiating SDP Host(IH) : IH는 SDP 컨트롤러와 통신하여 연결할 수 있는 Accepting Host(AH) 목록을 요청하고 SDP 컨트롤러는 IH의 소프트웨어 또는 하드웨어와 같은 정보를 요청할 수 있다.
- Accepting SDP Host(AH) : AH는 기본적으로 SDP 컨트롤러를 제외한 모든 호스트 및 외부 네트워크와 연결이 거부되어 있고 컨트롤러에게 인가받은 IH만 연결을 수락한다.

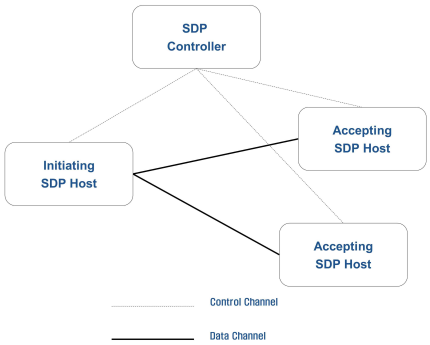

Fig. 1. Conceptual Diagram of SDP

2.3 SDP의 연결[7]

SDP의 각 요소들 간의 연결 순서를 도식화하면 Fig.2.와 같으며 자세한 내용은 다음과 같다.

- 하나 이상의 SDP 컨트롤러가 대기 상태로 존재하며 적절한 인증 및 인가를 위한 서비스와 연결되어 있다.

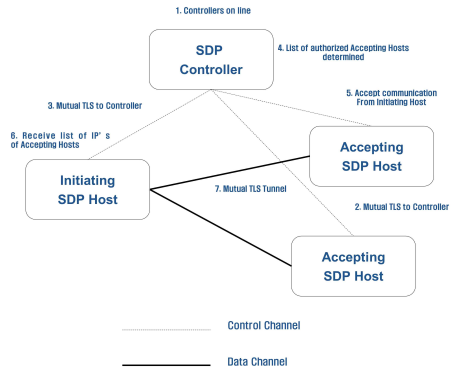

Fig. 2. Architecture of the SDP

- 하나 이상의 AH들도 SDP 컨트롤러와 Fig.3.과 같은 프로토콜을 이용하여 인증된 상태로 연결된다. 그리고 컨트롤러 이외의 다른 호스트들에 대한 연결 요청은 차단된 상태로 존재한다.
- 각각의 IH들도 SDP 컨트롤러와 연결하여 인증을 수행한다.
- IH와 인증한 뒤 SDP 컨트롤러는 IH가 통신할 수 있는 AH 목록을 설정한다.
- SDP 컨트롤러는 IH가 접근 가능한 모든 AH들에게 IH로부터 연결을 수락할 것과 암호화 통신에 필요한 모든 정책을 지시한다.
- SDP 컨트롤러는 IH에게 인증된 AH 목록과 암호화 통신에 필요한 모든 정책을 제공한다.(Fig.4.)
- IH는 승인된 각 AH에게 SDP 컨트롤러로부터 받은 정책으로 SPA(Single Packet

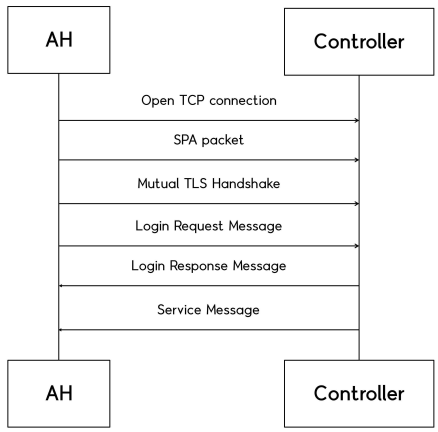

Fig. 3. AH connects to the Controller

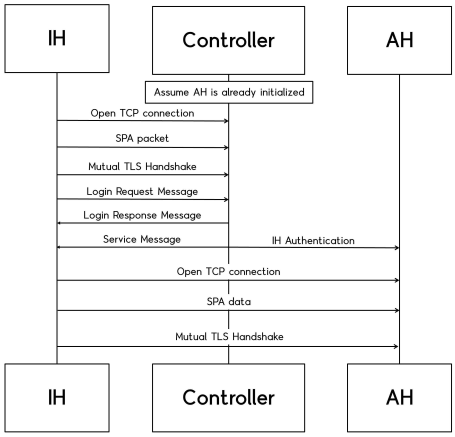

Fig. 4. IH connects to the Controller and an AH

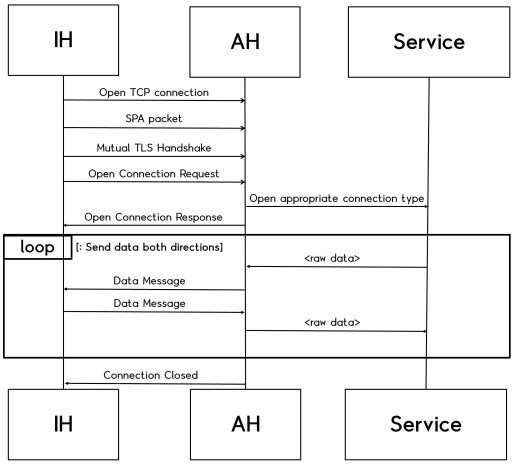

Fig. 5. IH Connects to an AH and then sends data to a Service

Authentication)를 생성하여 전달하고 mTLS로 연결하여 통신을 수행한다.(Fig.5.)

2.4 SDP의 구현방안[7]

SDP를 이용한 구현은 애플리케이션의 종류에 따라 다양한 방식이 존재하는데 자세한 내용은 다음과 같다.

- 클라이언트-게이트웨이 방식 : 하나 이상의 서버는 AH(Accepting Host) 뒤에 위치하여 보호되므로 AH는 클라이언트와 보호된 서버 사이에서 게이트웨이 역할을 할 수 있다. 이 구현 방식은 서버 스캐닝, OS 및 애플리케이션 취약성 공격, 중간자 공격 등 일반적인 단방향 공격을 방어하기 위해 네트워크 내부에 구현한다.
- 클라이언트-서버-클라이언트 방식 : 클라이언트-게이트웨이 방식과 유사하지만 클라이언트-서버 방식은 AH를 실행 중인 서버를 보호하는 것으로 보호하는 서버 수, 로드 밸런싱 방법 등에 따라 클라이언트-게이트웨이 방식과 클라이언트-서버 방식을 구분하여 선택할 수 있다.
- 서버-서버 방식 : 서버 간 통신에서 SDP를 구현하면 REST, SOAP, RPC나 인터넷을 통한 모든 종류의 API를 제공하는 것과 같은 다양한 서비스에 대한 부하를 줄일 수 있고 취약점 공격, DDoS, XSS, CSRF와 같은 다양한 공격을 완화할 수 있다.
- 클라이언트-서버 방식 : IP 전화, 채팅 및 비디오 회의와 같은 응용 프로그램에서 사용되며 SDP가 클라이언트와 서버의 IP 주소를 난독화하여 사용자와 서버를 숨겨 공격자로부터 보호할 수 있다.

III. 디바이스 유효성 검사의 필요성

3.1 SDP 구조의 보안 요소[8]

SDP 환경에서 디바이스 유효성 검사의 필요성에 대해 언급하기 전에 SDP 환경을 이루는 5가지의 독립된 보안 요소에 대해 먼저 설명하려고 한다.

- SPA(Single Packet Authentication) : SPA는 호스트를 인증하는데 가장 중요하게 쓰이는 보안 요소이다. SDP에서는 이 SPA를 이용하여 인가되지 않은 호스트의 트래픽을 차단한다. IH는 SDP 컨트롤러로 SPA를 암호화하여 전송하고 SDP 컨트롤러는 IH가 정당한 장치인지 확인 후 접근 권한을 부여한다. 또한 접근 권한을 획득한 뒤 SPA를 AH로도 전송하여 정당한 IH임을 확인한 뒤 트래픽을 허용한다.
- mTLS(Mutual Transport Layer Security) : 기존의 TLS는 인터넷을 통해 장치 인증과 기밀 통신을 가능하기 위해 클라이언트가 서버를 인증하는 단방향 인증이지만 SDP는 TLS의 모든 기능을 활용하여 상호 양방향 암호화 인증을 지원한다.
- 동적 방화벽 : 다양한 규칙을 가지는 기존 정적 방화벽과 달리 동적 방화벽을 이용하여 모든 트래

픽을 차단한다는 전제로 화이트리스트 차단 기법을 이용하고 있다. AH에서는 SDP 컨트롤러에 의해 인증된 권한이 있는 사용자만이 승인된 모든 응용 프로그램 및 서비스에 접근할 수 있도록 접근제어 규칙을 동적으로 추가하거나 서비스 이용이 끝나면 승인된 사용자의 접근 권한을 제거하며 수정할 수 있다.

- 응용 프로그램 바인딩 : SDP 환경에서는 SDP에 의해 생성된 TLS 터널을 강제로 사용하도록 하여 SDP 컨트롤러에 의해 승인된 응용 프로그램만 TLS 터널을 이용해 통신하고 승인되지 않은 응용 프로그램은 모두 차단된다.
- 디바이스 유효성 검사 : mTLS는 키가 만료되지 않았거나 취소되지 않았음을 증명하지만 사용된 암호화 키가 적절한 장치에 보관되어 있는지 확인할 수 없다. 따라서 장치가 인증된 사용자에게 속해 있고 신뢰할 수 있는 소프트웨어를 실행하고 있는지 확인할 수 있도록 디바이스 유효성 검사를 수행한다.

3.2 디바이스 유효성 검사 기술의 부재

SDP 환경에서 디바이스 유효성 검사는 사용자 키가 탈취되지 않았고 정당한 사용자에 의해 디바이스가 조작되고 있음을 알 수 있는 중요한 보안 요소이다. 하지만 CSA에서 2014년 발표한 SDP 명세서 버전 1.0에서는 디바이스 유효성 검사에 대해 향후 버전에서 해결되어야 한다고 명시하였고[7] 이후 발표한 2.0 버전에서도 디바이스 유효성 검사를 수행할 수 있는 기술을 표준화하여 명시하지 않았다. 따라서 정당한 사용자가 정당한 디바이스를 사용하고 있는지 판단할 수 있는 기술이 필요한데 이는 인가된 사용자가 평소에 행하는 디바이스 사용 행위 패턴을 학습하고 인가된 사용자와 비인가 된 사용자를 구별하여 비인가된 사용자로 판단될 경우 SDP 컨트롤러에 의해 서비스 접속을 제한하는 방식으로 시스템을 운영할 수 있다. 이때 비인가된 사용자를 탐지하는 방법은 기존에 알려진 이상행위 탐지 기능을 이용하여 해결할 수 있다.

IV. SVDD(Support Vector Data Description)

4.1 SVDD의 개념

비선형 SVM의 한 종류인 SVDD는 데이터 마이닝을 사용하는 방법으로 단일 클래스 데이터를 분류하는데 가장 일반적으로 사용된다. 단일 클래스 분류는 지정된 벡터 그룹에서 유효하지 않은 벡터를 식별하는 것으로 David에 의해 처음으로 제안되었다[9]. 이 방법은 비선형 변환을 통해 데이터를 더 높은 차원의 벡터 공간으로 매핑하고 매핑된 모든 데이터를 포함하는 최소 반지름으로 초구의 경계를 생성하여 동일한 특성의 데이터 클래스를 생성한다. 따라서 매핑된 데이터가 대부분 포함된 최적화된 초구를 생성할 수 있는 원점 a 와 반지름 R을 찾는 것이 중요하다. 이렇게 생성한 초구를 이용하여 초구 내부에 매핑된 데이터와 외부에 매핑된 데이터를 분류함으로써 데이터의 유효성을 결정할 수 있다 (Fig.6.)[10].

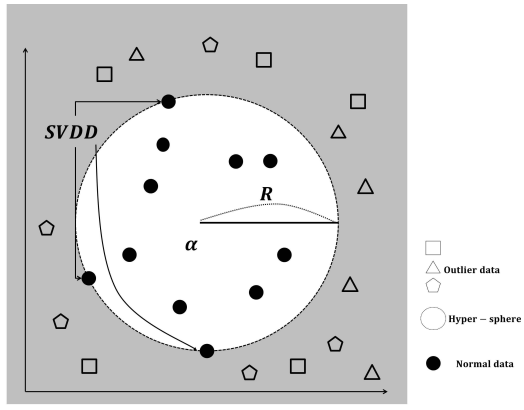

Fig. 6. Basic Concept of SVDD

4.2 SVDD 프로파일링

과거 전자 금융 사고 관련 논문에서[11] 스마트폰 사용자의 터치스크린에서 시작 좌표, 끝 좌표, 스크롤 속도를 측정하고 이를 데이터 마이닝 알고리즘에 적용하여 탐지 규칙을 만든 사례가 있다. 본 논문에서는 PC에서 추출할 수 있는 행위 정보로 사용자의 위치정보, 기기 정보, PC 입력 정보, 사용 시간 등을 이용하여(Table 1.) 인가된 사용자의 데이터 벡터를 생성하고 비인가된 사용자들의 데이터 벡터를

Table 1. User Device Behavior Information Profile

Type	Variable	Description
Device Information	Device Type	PC or Laptop
	OS	Operating System
	IP Address	Location information of the device
	MAC Address	Unique Information of the device
	Country Code	The region code where the user action occurred
User Action Information	Mouse Movement Speed	Mouse Mevement Speed
	Mouse Wasteful Movement	Mouse Wasteful Movement
	Mouse Direction	Mouse Cursor Direction
	Keyboard Typing Speed	Keyboard Typing Speed
	Keyboard Duration	Key Down/Up Speed
	Keyboard Interval	The Pressing Between Keys
	Backspace Count	The Proportion of Backspace Events in All Key events
	ID/PW	User Login Information
Agent Information	Version Information	Agent Version
	Execution Time	Agent Execution Time
	Termintaion Time	Agent Termintaion Time

생성하여 SVDD 학습을 통해 초구의 원점과 반지름을 생성하였다. 학습을 통해 생성된 초구는 사용자가 디바이스를 사용하면서 발생하는 데이터를 학습된 SVDD 초구에 매핑하게 되고 매핑된 데이터와 초구의 경계까지의 거리를 계산하여 초구의 내부에 위치하는지 외부에 위치하는지 판단함으로써 디바이스와 정당한 사용자에 대한 유효성을 결정할 수 있다.

4.3 SVDD 사용자 행위정보 선정

- Device Type : 사용자의 주 사용 디바이스 가 PC 인지 노트북인지 파악하기 위한 정보이다.
- OS : 사용자가 한 디바이스에서 다수의 OS를 사용하고 있는지 파악하기 위한 정보이다.
- IP Address : 동적 IP를 사용하거나 유/무선 네트워크 VPN 이용 여부를 확인하기 위한 정보이다.
- MAC Address : 사용자의 디바이스에 설치된 다수의 네트워크 장치를 파악하기 위한 정보이다.
- Country Code : VPN을 이용하여 해외를 경유할 수 있기 때문에 사용자의 위치정보를 파악한다.
- Mouse Movement Speed : 정지된 마우스 커서 위치에서 사용자가 클릭 이벤트를 발생시킬 때까지 소요된 시간과 좌표를 이용하여 속도를 측정한다.
- Mouse Wasteful Movement : 정지된 마우스 커서 위치에서 사용자가 클릭 이벤트를 발생한 위치까지의 최단거리와 실제 움직인 거리 간의 차이를 이용하여 낭비 동선을 측정한다.[12]
- Mouse Direction : 마우스 커서가 움직인 방향을 각 마우스 좌표들 간 내적의 합을 이용하여 측정한다.[12]
- Keyboard Typing Speed : 사용자가 키보드를 이용할 때 발생하는 분당 키보드 입력 횟수를 측정한다.
- Keyboard Duration : Key Down 이벤트와 Key Up 이벤트가 발생할 때 나타난 모든 시간차를 더한 뒤 키 Key Down/Up 이벤트 횟수로 나누어 측정한다.[13]
- Key Interval : 한 키의 Key Up 이벤트와 다른 키의 Key Down 이벤트가 발생할 때 나타난 모든 시간차를 더한 뒤 Key Up/Key Down 이벤트 횟수로 나누어 측정한다.[13]

- Backspace Count : 사용자별 오타율을 측정하기 위해 전체 키 이벤트에서 백스페이스키 이벤트가 차지하는 비율을 측정한다.
- ID/PW : 사용자 구분을 위한 ID/PW 정보이다.
- Version Information : 디바이스 유효성 검사를 수행하는 에이전트의 버전 정보이다.
- Excution Time : 사용자가 에이전트를 실행한 시간을 측정한다.
- Termintaion Time : 사용자가 에이전트를 종료한 시간을 측정한다.

4.4 SVDD를 이용한 디바이스 유효성 검사

SVDD를 이용한 디바이스 유효성 검사는 사용자 행위 패턴 벡터 생성, 사용자 행위 패턴을 이용한 학습, 디바이스 유효성 검사의 총 3단계로 이루어져 있으며 자세한 내용은 다음과 같다.

- 1단계 : 사용자의 일반적인 행위 패턴을 이용한 데이터 벡터 생성

표 1을 기반으로 n개의 원소를 포함하는 하나의 벡터 X를 생성한다.

$$X = \{x_1, x_2, x_3, \cdots, x_n\}$$

사용자 행위 정보를 K번 반복하고 정상적인 사용자 행위 패턴 X벡터를 포함하는 벡터 그룹 D를 생성한다.

$$D = \{X_1, X_2, X_3, \cdots, X_K\}$$

- 2단계 : SVDD 학습을 통해 정상적인 사용자 행위 패턴을 이용한 학습

1단계에서 벡터 D그룹으로 SVDD 알고리즘을 실행하여 최적화된 초구를 생성하여 생성된 원점 a와 반지름 R이 사용자의 정상적인 행위 패턴에 속하는 영역이 된다.

- 단계 : 디바이스 유효성 검사 수행

사용자가 디바이스를 사용하면서 일정 시간 동안 발생한 행위 정보를 평균 내어 벡터 Z 를 생성하고 생성한 벡터를 초구 평면에 매핑하여 벡터 Z가 매핑된 지점과 원점까지의 거리, 그리고 초구의 반지름을 비교하여 디바이스 유효성 검사를 수행하게 된다. 벡터 Z에서 매핑된 지점을 z라고 할 때

$$\| z - a \|^2 \leq R^2$$

인 경우 정상으로 판단, 반대인 경우는 이상으로 판단한다.

V. SDP 환경에서 SVDD 기반 디바이스 유효성 검사 성능 평가

5.1 평가 방법

본 논문에서 제안하는 디바이스 유효성 검사 성능을 평가하기 위한 방법은 SDP 환경에서 PC에 설치되는 IH(Initiating Host) 에이전트를 이용하여 사용자 행위 정보를 수집하고 인가된 사용자의 정상 행위와 비인가된 사용자의 행위 정보를 SVDD 알고리즘으로 학습하여 생성되는 원점과 반지름을 이용한 초구를 생성한다. 이후 에이전트를 이용하여 사용자의 행위 정보를 모니터링하고 정상적인 행위의 벡터와 비정상적인 행위를 벡터를 적용하여 디바이스의 유효성 검사를 수행한다. 성능 지표는 EER(Equal Error Rate)를 사용하였다. EER은 사용자 인증 알고리즘의 성능지표로써 인가된 사용자가 인증에 실패할 때 나타나는 오류율인 FRR(false Rejection Rate)와 비인가된 사용자가 인증에 성공할 때 나타나는 오류율인 FAR(False Acceptance Rate)가 최소가 되는 값을 의미한다. 일반적으로 FRR이 낮을수록 사용자의 편의성이 높아지고 FAR이 낮을수록 인증의 강도가 높아진다. 따라서 디바이스 유효성 검사를 수행할 때 SVDD 학습 모델의 패널티를 0.01부터 0.8까지 조절하며 평가하여 각 임계치별 FAR, FRR을 통해 EER을 결정하여 성능을 평가한다.

5.2 평가 환경

디바이스 유효성 검사를 위한 SDP 환경은 IH 역할을 수행하기 위해 에이전트가 설치된 사용자의 PC 그리고 컨트롤러 역할을 하는 서버와 AH 역할을 하며 사용자의 행위 정보를 수집할 수 있는 웹 서

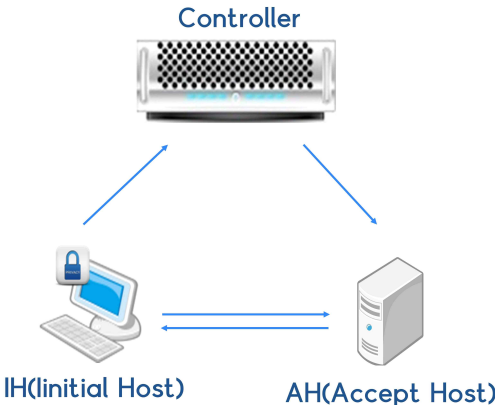

Fig. 7. Evaluation Environment

버로 구성하였다. 컨트롤러와 IH, 그리고 컨트롤러와 AH는 사전에 인증된 상태이며 IH는 접속 가능한 AH 목록을 알고 있다. 또한 인가된 사용자의 행위 정보를 프로파일링 하여 SVDD 모델이 생성된 상태이다.

5.3 평가 절차

- 1단계 : 인가된 사용자가 에이전트를 실행하여 아이디/패스워드를 입력하여 로그인을 한다.
- 2단계 : 사용자는 로그인된 에이전트를 이용하여 컨트롤러에게 AH로 연결을 요청한다.
- 3단계 : 컨트롤러는 IH의 SPA 패킷을 확인하여 정당한 IH 인지 확인한다.
- 4단계 : 컨트롤러는 AH에게 IH의 연결을 수락할 것을 명령하고 IH에게 서비스 이용 가능 메시지를 전달한다.
- 5단계 : AH는 IH의 정보를 확인하여 IH의 연결을 허용하고 암호 채널을 이용하여 통신한다.
- 6단계 : 사용자는 AH의 웹서버에 접속하여 마우스 이동, 마우스 클릭, 마우스 드래그, 긴 글 작성, 짧은 글 작성행위를 수행한다.
- 7단계 : 인가된 사용자가 6단계의 행위를 500회 반복하여 SVDD 모델 생성을 위한 학습 데이터를 생성한다.
- 8단계 : 인가된 사용자의 에이전트가 로그인된 상태에서 인가/비인가 된 사용자가 6단계의 행위를 수행한다.
- 9단계 : 모든 사용자의 행위 정보를 수집하여 데이터베이스에 저장한다.
- 10단계 : 사용자 인증을 위한 SVDD 학습 모델의 페널티를 변경해가며 FAR과 FRR을 목록화하고 최소의 EER을 구한다.

5.4 평가 수행

평가에는 1명의 인가된 사용자와 14명의 비인가된 사용자가 참여하였다. 총 15명의 사용자들이 순차적으로 인가된 사용자의 PC를 이용하여 마우스를 이용한 웹 페이지의 오브젝트 이동(드래그), 무작위 위치에서 나타나는 오브젝트 클릭 및 더블클릭, 웹 페이지에서 제공하는 긴 글 및 짧은 글 타자 치기의 행위를 수행하였다. 평가는 총 10회 반복하였으며 15명의 데이터가 10회 수집되어 총 150개의 데이터를 수집하여 분석을 진행했다. 디바이스 유효성 검증을 위한 SVDD 모델은 500개의 학습 데이터를 이용하였으며 이상 데이터의 페널티를 각각 0.01, 0.05, 0.08, 0.1, 0.3, 0.5, 0.8로 수정하면서 생성된 SVDD 모델별로 150개의 검증 데이터를 이용하여 FAR과 FRR을 측정하였다.

5.5 평가 결과

학습 데이터의 이상치에 페널티를 수정하며 총 8개의 SVDD 학습 모델을 생성하였고 인가된 사용자 10개 비인가된 사용자 140개로 총 150개의 검증 데이터를 이용하여 FAR과 FRR을 측정하였다. 그 결과 페널티 0.01인 경우에는 SVDD 모델의 반지름이 0.0865로 FAR은 5%로 최솟값을 보였으나 FRR이 50%로 최댓값이 나타났다. 이후 페널티를 증가시킨 모델에 검증 데이터를 입력한 결과 페널티

Table 2. FAR and FRR according to SVDD learning model error penalty.

Penalty	Radius	FAR	FRR
0.01	0.0865	5%	50%
0.03	0.1041	6%	40%
0.05	0.1066	7%	30%
0.08	0.1163	8%	20%
0.1	0.1156	8%	20%
0.3	0.1173	8%	20%
0.5	**0.1203**	**9%**	**10%**
0.8	0.1201	9%	10%

가 0.5일 때 생성한 SVDD 모델의 반지름은 0.1203으로 FAR은 최대 9%까지 증가하였고 FRR은 10%까지 줄어들어 모든 페널티 중 가장 유사한 것을 확인할 수 있었다.(Table 2.)

따라서 EER은 FAR과 FRR이 유사해지고 반지름이 최대가 되는 페널티 0.5 지점이 된다. Fig.8. 이 페널티 0.5로 생성한 SVDD 모델에 대해서 행위 정보를 꺾은선 그래프로 표시한 결과이다. 가운데 붉은 직선이 SVDD 모델에 의해 생성된 초구의 반지름이고 붉은 점은 인가된 사용자의 행위 정보가 초구의 중심에서 벗어난 거리를 표시한 데이터, 검은색 점이 비인가된 사용자의 행위 정보가 초구의 중심에서 벗어난 거리를 표시한 데이터이다. 그래프를 분석해 보면 붉게 표시된 SVDD 반지름보다 아래쪽에 위치한 붉은 점이 총 9개로 인가된 사용자의 10번의 행위 중 1번이 디바이스 유효성 검증에 실패한 것을 확인할 수 있다. 따라서 페널티 0.5일 때 인가된 사용자의 유효성 검증이 실패할 확률인 FRR은 10%이다. 마찬가지로 SVDD의 반지름보다 아래쪽에 위치한 검은 점이 13개로 비인가된 사용자의 140번의 행위 중 13번이 유효성 검증에 실패하여 비인가된 사용자가 디바이스 유효성 검증에 성공할 확률인 FAR은 9%이다.

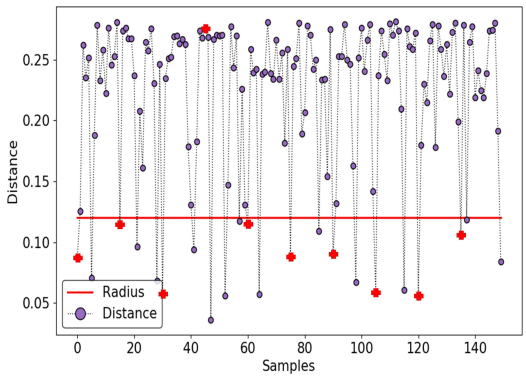

Fig. 8. Device Validation Evaluation Result

VI. 결 론

본 논문에서는 SDP 환경에서 디바이스 유효성 검사를 수행하기 위한 SDP의 개념 및 구조, 각 호스트들 간의 연결 절차와 구현 방안을 설명하였고 SDP 구조에 적용된 보안 요소들을 소개하였다. 그 중 디바이스 유효성 검사는 SDP 명세서에서도 적절한 기술을 제시하고 있지 않기 때문에 이를 해결하기 위해 SVDD를 이용한 디바이스 유효성 검사 방안을 제시하였다. IH(Initial Host) 에이전트에 이진 분류 기계학습 알고리즘인 SVDD를 적용하여 인가된 사용자의 기기 정보, PC 행위 정보, 에이전트 정보를 학습하여 모델을 생성하고 다수의 비인가된 사용자에게 학습된 디바이스를 사용하게 하여 성능을 평가하였다. 성능 평가는 사용자 인증 알고리즘의 성능 지표인 EER을 사용하였으며 EER을 측정하기 위해 비인가된 사용자가 디바이스 유효성 검증에 성공하는 FAR과 인가된 사용자가 디바이스 유효성 검증에 실패하는 FRR을 모두 측정하였다. FAR과 FRR 측정에 사용된 민감도는 SVDD 학습 시 사용되는 이상치에 대한 페널티를 조절하며 측정하였으며 그 결과 이상치 페널티 0.5일 때 SVDD의 반지름이 0.1203이고 이때 FAR과 FRR이 9%와 10%로 가장 유사하여 EER 지점으로 채택하였다. SDP 환경에서 디바이스 유효성 검사에 실패한 기기와 사용자는 1차적으로 ID/PW, FIDO, 핸드폰 등 추가 인증을 요청하여 사용자 재인증을 수행할 수 있고 지속적인 디바이스 유효성 검사에 실패하게 된다면 사용자의 서비스 제한, 사용자와 컨트롤러 간 재인증 등의 절차를 진행하여 인가된 사용자에게 제재를 가할 수 있다. 본 논문에서는 제한된 수의 행위 정보와 빠른 학습을 위해 SVDD를 적용한 디바이스 유효성 검증 기술을 제시하였지만 추후 연구로 현재의 행위 정보를 보다 세분화하여 사용자 행위 정보를 수집하고 연합 학습과 같은 향상된 인공지능 기술을 이용하여 디바이스 유효성 검사를 수행한다면 인가자에 의한 내부정보 유출, 백도어 프로그램 설치, 악성코드 유포 등 다양한 내부자 공격을 탐지 및 방어할 수 있고 비대면 환경의 새로운 애플리케이션에도 적용하여 다양한 서비스를 안전하게 이용할 수 있을 것이다.

References

[1] de Vet, J.M, et al. "Impacts of the COVID19 pandemic on EU industries" Publication for the committee on Industry, Research and Energy, Policy Department for Economic, Scientific and Quality of Life Policies, European Parliament, Luxembourg, Mar. 2021.

[2] Dong-Hyun Yu, Yong-Uk Kim, Young-Jae Ha and Yeon-Seung Ryu, "Consideration of New Convergence Security Threats and Countermeasures in the Zero-Contact Era" *Jonornal of The Korea Conergence Society, vol. 12, No.1*, pp. 1-9, Jan. 2012.

[3] Scoot Rose, Oliver Borchert, Stu Mitchel and Sean Connelly, "Zero Trust Architecture", NIST SP 800-207, Aug. 2020.

[4] John Kidervag, "No More Chewy Centers: Introducing The Zero Trust Model Of Information Security," FORRESTER, Sep. 2010.

[5] Wikipedia, "Software Defined Perimeter", https://en.wikipedia.org/wiki/Software_Defined_Perimeter, May 2021

[6] Jason Garbis and Juanita Koilpillai, "Software-Defined Perimeter ARCHITECTURE GUIDE," Cloud Security Alliance, Jul. 2019.

[7] BrentBilger, Alan Boehme, Bob Flores, Zvi Guterman, Mark Hoover, Michaela lorga, Junaid Islam, Marc Kolenko, Juanita Koilpilla, Gabor Lengyel, Gram Ludlow, Ted Schroeder and Jeff Schweitzer, "SDP_Specification 1.0," Cloud Security Alliance, Apr. 2014.

[8] Abdallah Moubayed, Ahmed Refaey and Abdallah Shami, "Software-Defined Perimeter(SDP): State of the Art Secure Solution for Modern Networks," IEEE Network, 33(5), pp. 226-233, Sep. 2019.

[9] T. David and D. Robert, "Support vector data description", Machine Learning, vol.54, nol.1, pp. 45-66, Jan. 2004.

[10] Mun-Kweon Jeong, Seong-Ho An and Kihyo Nam, "SVDD-Based Fianacial Fraud Detection MethodThrough Respective Learnings of Normal/Abnormal Behaviors," International Journal of Security and Its Applications, vol.10, No.3, pp. 429-436, Mar. 2016.

[11] J. H. Park, "Effective Normalization Method for Fraud Detection Using a Decision Tree", Jounal of the Korea Institute of Information Security & Cryptology, vol. 25, no. 1, pp. 133-146, Feb. 2015.

[12] Minsoo Park, Jumin Park, Hyungheon Kim and Yoojae Won, "User Identification Method Using Input Pattern Analysis", The Korean Society Of Computer And Information Conference, 25(1), pp. 213-216, Jan. 2017.

[13] Kyeong-Jin Sa, Jae-Yeon Woo and Heung-Youl Youm, "Behavior-based biometric authentication available for multi-factor authentication", KIISC review, 26(6), pp. 51-57, Dec. 2016

〈저자소개〉

이 희 웅 (Heewoong Lee) 학생회원
2013년 2월: 공주대학교 응용수학과 학사
2015년 2월: 공주대학교 수학과 이학석사
2021년 3월~현재: 공주대학교 응용수학과 박사과정
〈관심분야〉 네트워크 보안, 시스템 보안

홍 도 원 (Dowon Hong) 종신회원
1994년 2월: 고려대학교 수학과 학사
2000년 2월: 고려대학교 수학과 박사
2000년 4월~2012년 2월: 한국전자통신연구원 팀장, 책임연구원
2012년 3월~현재: 공주대학교 응용수학과 교수
〈관심분야〉 암호기술, 프라이버시 보호기술

남 기 효 (Kihyo Nam) 종신회원
1993년 2월: 고려대학교 산업공학 학사
1995년 2월: 고려대학교 산업공학 공학석사
1999년 2월: 고려대학교 산업공학 공학박사
2002년~2005년: ㈜프롬투정보통신 기술기획팀장
2005년~2010년: ㈜위너다임 이사
2010년~현재: ㈜유엠로직스 부사장
〈관심분야〉 암호알고리즘, PKI, 무선인터넷 보안 등