The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 16, No. 6, pp.33-42, Dec. 31, 2016. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2016.16.6.33
JIIBC 2016-6-5

머신러닝 알고리즘이 적용된 가상화 내부 환경의 보안 인증벡터 생성에 대한 연구

A Study on Security Authentication Vector Generation of Virtualized Internal Environment using Machine Learning Algorithm

최도현*, 박중오**

Choi Do-Hyeon*, Jung Oh Park**

요 약 최근 인공지능 분야는 구글, 아마존, 마이크로 소프트 등 선진 기업을 중심으로 머신러닝에 대한 투자와 연구 경쟁이 가속화 되고 있다. 가상화 기술은 가상화 보안 구조에 대한 보안 취약점 문제가 지속적으로 이슈화 되었다. 또한 내부 데이터 보안이 플랫폼 제공자의 가상화 보안 기술에 의존적인 경우가 대부분이다. 이는 기존 소프트웨어, 하드웨어 보안 기술은 가상화 영역 접근이 어렵고 보안 기능 수행에 데이터 분석 및 처리 효율성이 낮기 때문이다. 본 논문은 사용자 중요 정보를 기계학습 알고리즘을 적용하고, 학습 가능한 보안 인증벡터 생성하여 이를 가상화 내부 영역에서 보안 검증을 수행할 수 있는 방법을 제안한다. 성능분석 결과 인증벡터의 가상화 환경의 내부 전송 효율성, 연산 방법의 높은 효율성과 주요 생성 파라미터에 대한 안전성을 입증하였다.

Abstract Recently, the investment and study competition regarding machine running is accelerating mainly with Google, Amazon, Microsoft and other leading companies in the field of artificial intelligence. The security weakness of virtualization technology security structure have been a serious issue continuously. Also, in most cases, the internal data security depend on the virtualization security technology of platform provider. This is because the existing software, hardware security technology is hard to access to the field of virtualization and the efficiency of data analysis and processing in security function is relatively low. This thesis have applied user significant information to machine learning algorithm, created security authentication vector able to learn to provide with a method which the security authentication can be conducted in the field of virtualization. As the result of performance analysis, the interior transmission efficiency of authentication vector in virtualization environment, high efficiency of operation method, and safety regarding the major formation parameter were demonstrated.

Key Words : Virtualization, Machine Learning, Hypervisor, Cloud Platform, Authentication Vector

Ⅰ. 서 론

최근 무인 드론(Drone)을 이용한 서비스[1][2], 무인 자동차의 상용화 예정 소식[3], 인공지능 알파고(Alpha Go)의 바둑 대국[4] 등 인공지능 분야의 가능성은 기존 침체되었던 인공지능 분야를 다시 활성화 시키는 계기가 되었다. 현재 클라우드(Cloud) 시장은 Google, Amazon, Microsoft, IBM 등을 중심으로 연 평균 54% 성장률[5] 증

*정회원, 숭실대학교 컴퓨터공학
**정회원, 성결대학교 파이데이아학부 (교신저자)
접수일자 : 2016년 8월 24일, 수정완료 : 2016년 10월 24일
게재확정일자 : 2016년 12월 9일

Received: 24 August, 2016 / Revised: 24 October, 2016 / Accepted: 9 December, 2016
**Corresponding Author: jopark02@sungkyul.ac.kr
Paideia College Dept, of Paideia Sungkyul University, Korea

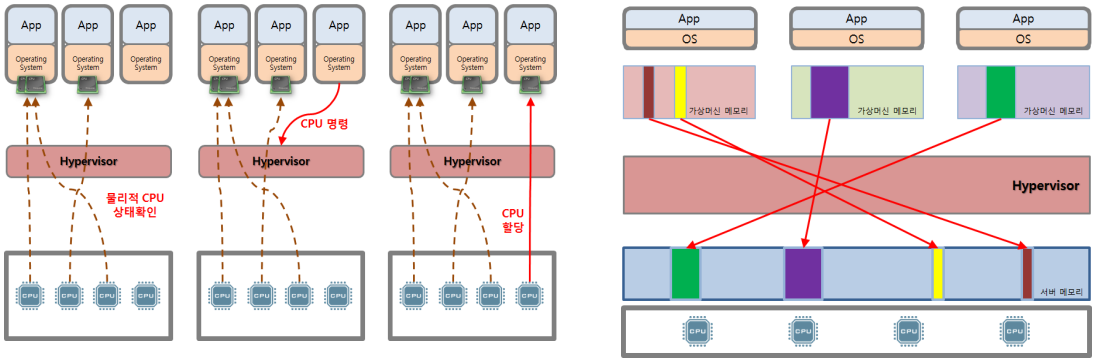

그림 1. CPU와 메모리 가상화(예)
Fig. 1. CPU and Memory Virtualization(Example)

가율을 기록하였고, 주요 선진 기업들은 자사 클라우드 플랫폼을 기반으로 적극적인 투자와 전문 인력 고용, 차세대 인공지능 서비스 연구 및 개발 등에 나서고 있다. 클라우드 서비스는 가상화(Virtualization)라는 개념으로 기존 운영체제나 사용자 프로그램을 웹브라우저 상에서 실행할 수 있는 형태로 변화시켰다[6]. 이에 따라 기존 주요 어플리케이션 및 중요 데이터가 가상화된 자원으로 관리될 것으로 예상된다. 기존 알려진 가상화 기술에는 Xen, RHEV, VMM, VMware 등 지속적으로 취약점[7-11]이 발견되고 있기 때문에 가상화 보안 기술은 안정적인 사용자 서비스를 위한 필수 고려 사항이 되었다.

본 논문은 가상화 환경의 내부 데이터 보호를 위해 학습 가능한 인증벡터의 생성 및 활용 방법에 대해 제안한다. 인증벡터에 삽입되는 인증정보는 가상화 환경의 연산에 적절한 기계학습 알고리즘 적용하였다. 인증 패턴은 학습되어 누적하고 신뢰도 예측을 통해 비정상 사용자를 탐지하는데 활용하였다. 또한 보안성, 효율성, 실용성을 고려하여 SSO, SSL, PKI 등 기존 사용자 인증 기술에서 암·복호화 기능을 모듈 형태로 적용할 수 있도록 구현하기 쉬운 구조로 설계되었다.

2장 관련연구는 기존 가상화 보안 기술과 머신러닝 기법의 취약점과 문제점을 분석한다. 3장은 가상화 보안을 위한 인증벡터의 생성 및 검증 방법에 대하여 설명한다. 4장 인증벡터의 효율성 및 안전성을 분석한다. 5장 결론으로 마친다.

II. 관련연구

본 절에서는 하이퍼바이저(Hypervisor) 보안 기술, 머신러닝(Machine Learning) 기술의 문제점을 설명한다.

1. 하이퍼바이저 보안 기술과 취약점 분석

최근 PC, 모바일 등 다양한 환경이 인터넷과 결합되면서 저장 데이터 용량의 급격한 증가와 함께 컴퓨팅 패러다임이 클라우드 환경으로 전환되었다[12]. 또한 대부분 서버 플랫폼은 데이터 관리 및 통합의 비용적인 문제를 해결하기 위해 가상화 기술을 도입하고 있는 추세이다[13].

그림 1와 같이 가상화 환경에서 운영되는 모든 데이터는 하이퍼바이저를 통해 전달됨을 확인할 수 있다. 하이퍼바이저는 그 형태에 따라 Type1, Type2로 구분된다. Type1은 하이퍼바이저 위에서 가상머신을 실행하고, Type2는 기존 전통적인 호스트 운영체제 내부에 VM을 운영하는 형태이다. Type2의 경우 호스트 운영체제를 통해 데이터를 통신하기 때문에 속도가 느린 단점이 있다. 개인 사용자 가상 프로그램으로 VMware(Works tation, Fusion), Microsoft(Virtual PC), Oracle(Virtual Box), QEMU 등이 있다. 현재 클라우드 플랫폼들은 ESX, Xen, and Hyper-V 등 효율성이 높은 Type1을 많이 활용하고 있다.

성능 효율성으로는 실시간 데이터 처리에 적합한 Type1 형태가 보안 기능 수행에 효율적인 것으로 분석된다. 표 1은 하이퍼바이저 보안 기술[14]에 대한 분석결과를 나타낸다. 분석 결과 각 가상화 보안 기술이 장·단점

이 존재하지만 독립된 관리모듈, 성능 효율성이 높은 내부정보 분석 기반 침입탐지가 적합한 것으로 분석된다. 그러나 관리 효율이 낮고 하이퍼바이저 내부에서 전달되는 정보가 유효한 경우에만 정상 동작하는 단점이 존재한다. 처리해야하는 중요 데이터 항목과 범위가 명확하게 정의되어야 한다. 공통적인 문제는 접근권한 분리가 독립적이지 않고 모두 플랫폼 서버의 보안구조에 의존적인 것으로 나타났다.

표 1. 가상화 보안 기술의 취약성 및 문제점
Table 1. Vulnerabilities and Problems of Virtualization Security Technology

Entry	Virtual Machine Introspection(VMI)	Agentless-based
Privilege Separation	플랫폼 제공자에 보안 기술에 의존적	플랫폼 제공자에 보안 기술에 의존적
Typical Vulnerabilities	관리모듈 해킹에 대한 해결책이 없음	관리모듈 해킹에 대한 해결책이 없음
Other Problems	수집정보가 유효한 경우에만 정상 동작	데이터 접근성(컨텍스트 데이터 분석)이 어려움

2. 보안 기능 적용을 위한 머신러닝 기법의 문제점 분석

머신러닝 기법의 보안 요구사항, 본 논문에서 사용된Naive Bayes, K-nearest neighbor, Apriori에 대하여 분석한다[15-18]. 표 2는 각 알고리즘의 특성과 용도를 나타낸다.

표 2. 머신러닝 알고리즘의 특징과 활용
Table 2. Features and Utilization of Machine Learning Algorithms

Algorithm	Description
Naive Bayes	특정분류 항목이 있는 데이터 집합에서 높은 확률을 가지는 분류항목을 선택, 마케팅을 위한 감성분석, 메일 필터링과 같은 문서 분류에서 활용
K-nearest neighbor	데이터 항목에 대한 범주를 모르는 경우 데이터를 분류, 좌표를 이용하여 속성 간에 거리를 구함, 비디오에서 글자나 얼굴 인식, 유전자 데이터 식별 등 패턴인식 분야에서 활용
Apriori	데이터 그룹간의 패턴을 명시하는 연관규칙을 찾아내는 방법, 학습되지 않은 데이터를 발견하는데 활용

주요 요구사항은 데이터 처리를 위한 머신러닝 알고리즘의 최적화, 보안기능 수행을 위한 명확한 데이터 범위정의와 추출로 분석되었다. 표 3은 데이터 접근 제어 관점에서의 머신러닝 보안 요구사항을 나타낸다.

표 3. 머신러닝의 보안 요구사항
Table 3. Security Requirement of Machine Learning

Entry	Requirement
Data Access	수집 데이터 범위가 명확한 범위로 정의되어야 함
Machine Learning Permissions	머신러닝 모듈 플랫폼 제공자로부터 분리되어야 함
Machine Learning data Extraction Range	변수가 많이 존재하면 연산 성능, 예측 결과의 신뢰도가 떨어지기 때문에 데이터의 1차 가공이 요구됨
Saving Data	학습데이터(입력)와 학습패턴(추출)이 독립적으로 저장 관리되어야 함
Machine Learning	모든 알고리즘이 연산의 최적화가 요구됨

표 4는 보안 분야 적용을 위한 머신러닝 요구사항(상세)을 나타낸다. 상세 요구사항은 공통적으로 데이터 1차 가공, 거리 값 추출, 데이터 그룹 범위 최소화 등 알고리즘 특성에 따라 최적화가 요구되는 것으로 분석되었다.

표 4. 머신러닝 알고리즘 요구사항(상세)
Table 4. Requirement of Machine Learning (Detail)

Algorithm	Requirement
Naive Bayes	알고리즘 특성상 1차 가공된 선행데이터가 요구됨으로 입력 데이터에 대한 키워드가 정의되어 있어야 함
K-nearest neighbor	거리 값에 대한 예측치는 객관성이 떨어져 예측 가능한 최근접 거리만을 추출
Apriori	연관규칙 추출 복잡도를 낮추기 위해 최소한의 데이터 그룹 범위에 대한 관계분석이 요구됨

Ⅲ. 머신러닝 기법에 최적화된 보안 인증벡터의 생성 및 검증

본 논문에서 제안하는 인증벡터는 사용자가 서버로 전달하는 다중 인증정보를 암호화하고 인증벡터에 삽입한다. 암·복호화에 앞서 인증벡터 내부 좌표의 거리 값(K-nearest neighbor)와 구조를 검증(Naive Bayes)을 선

행으로 수행한다. 인증벡터에서 추출된 데이터의 좌표 값과 인증정보는 인증패턴으로 추출된다.

1. 인증벡터의 생성

초기 인증벡터는 정방행렬 구조로 60 × 60(3600)크기의 행렬을 생성하고, 각 좌표(x, y)에 랜덤 정수를 삽입하고 셔플링(Shuffling)을 적용한다. 현재 시간, 분, 초의 타임스탬프(TimeStamp)를 초(Second)단위로 변환하고 인증벡터의 초기 좌표로 사용한다. 예로 5시 12분 30초인 경우 18750초를 x=18, y=750로 분리하여 초기 좌표로 사용한다. 그림 2는 초기 인증벡터의 좌표와 정방행렬(Squre Matrix)을 선택방법을 나타낸다. 초기 좌표를 기준으로 랜덤 크기(최대 60x60 크기)의 정방행렬을 선택하고 가운데 좌표를 선택한다. 선택된 가운데 좌표는 이후 실제 데이터 삽입 위치를 찾는 기준이 된다. 행렬 B의 경우 초기 좌표와 암호화 데이터 삽입 위치는 0, 3이고 정방행렬의 5 × 5의 가운데 좌표는 5, 3이다.

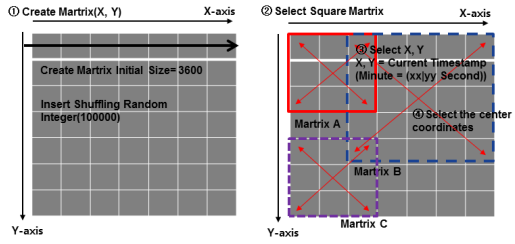

그림 2. 인증벡터의 생성과 좌표선택
Fig. 2. Generation of Authentication Vector and Coordinates Selection

24시를 기준으로 생성될 수 있는 인증벡터의 최대 크기 좌표는 86400초로 x=86, y=400이다. 전체행렬은 타임스탬프 값에 따라 최소 3600에서 최대 34400의 크기로 확장될 수 있다.

2. 데이터 암호화 및 삽입

표 5은 암호화 하는 사용자 인증정보 유형의 예를 나타낸다. 사용자가 전송한 인증정보는 아이디, 패스워드, 아이피, 운영체제 정보, 웹 브라우저 정보 등 사용자의 인증수행 중에 전송되는 중요 정보들을 의미한다. 표 5는 사용자 인증정보 유형의 예를 나타낸다.

표 5. 사용자 인증정보 유형(예)
Table 5. Type of User Authentication Information (Example)

Type	Description
ID	-
PASSWORD	-
Uniq_NUM	요청 고유 번호, 타임스탬프로부터 추출된 좌표 값 사용
SEQ_NUM	순서번호, 초기 : 0000
IP	-
OS_INFO	정의됨(윈도우, 맥, 리눅스 등)
Web_Browser	정의됨(익스, 크롬, 사파리 등)
TimeStamp	년, 월, 시간, 분, 초
Option 1	임시 필드 1
Option N	임시 필드 N

그림 3은 각 정방행렬에 초기좌표에 암호화된 인증정보를 삽입하는 과정을 나타낸다. 본 논문에서 삽입되는 메시지 암호화는 NIST 표준 AES-256을 적용하였다.

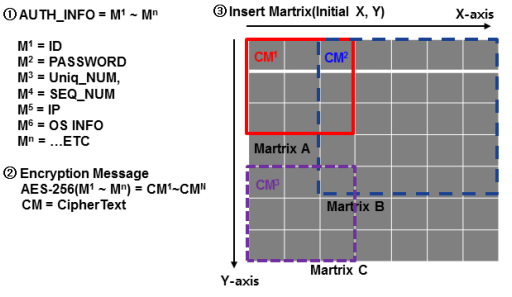

그림 3. 암호화된 인증정보의 삽입 과정
Fig. 3. Encrypted Authentication Information Insertion Process

표 6은 인증벡터 검증을 위해 추출된 인증벡터 테이블을 나타낸다. 삽입 좌표가 중복될 경우 정방행렬 선택 과정을 재수행해야 한다. 이는 인증벡터의 중복 데이터 쓰기를 방지하기 위해서이다. 추가적으로 현재 선택된 정방행렬 내에 다른 좌표가 삽입되었는지 확인한다.

표 6. 인증벡터 테이블(예)
Table 6. Authentication Table(Example)

Authentication Vector	Pattern Number	Center x, y	Request Number	Insert
A	10	1,1	1	1
B	28	1,6	3	2
C	11	5,3	2	1

인증벡터 테이블에는 패턴 번호, 가운데 좌표, 요청 번호를 저장하여 관리한다. 패턴 번호와 요청번호의 순서는 현재 삽입된 데이터의 인증정보 유형을 파악하는데 사용된다. 인증정보 추출 이후 정렬을 수행한다. 겹침 부분(Insert)은 정방행렬 영역이 겹치는 개수를 저장해 둔다. 현재 인증벡터에서 추출된 사용자 인증패턴은 10, 28, 11이다. 요청번호로 정렬 했을 때 사용자 인증패턴은 10, 11, 28 이다.

3. 인증벡터의 구조 검증

인증벡터는 최대 34400 크기 행렬까지 생성될 수 있다. 암호화된 데이터 추출을 위해서 전체행렬을 탐색할 경우 연산이 비효율적이다. 그림 4 인증벡터에서 패턴번호가 나타날 확률 P1과 패턴번호가 정방행렬 겹쳐질 조건부 확률 P2를 계산하는 방법은 수식 (1)와 같다.

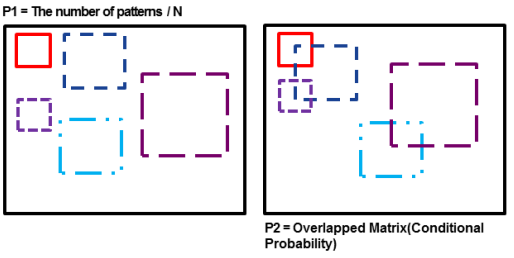

그림 4. 조건부 확률 P1과 P2
Fig. 4. Conditional Probability P1 and P2

$$P1(A) = \frac{count(Pattern\ Number)}{N}$$
$$P2(A|B) = \frac{P(B|A)P1(A)}{P(B)} = \frac{P(A \cap B)}{P(B)} \quad (1)$$

Naive Bayes의 조건부 확률을 이용하여 P1과 P2의 확률을 검사한다. P1은 10, 10(100) 크기인 행렬에 4개의 패턴번호가 삽입되어 있는 경우 4 / 100 = 0.04%이다. P2는 패턴 테이블의 겹침(Insert)을 변수 I 라고 정의했을 때 ((I / 4) * 0.04) / (3 / 100) = 0.3%(소수점 1자리)이다. 이는 선택된 4개 행렬 안에 1개의 좌표가 중복 포함되는 확률은 약 30% 라는 것을 추측할 수 있다. 데이터 추출 이전에 정상적인 인증벡터 구조인지 확인할 수 있다.

4. 다중 인증벡터 거리 값 검증

일반적으로 사용자는 개인 PC, 노트북, 스마트폰 등 다양한 디바이스를 통해 로그인을 수행하기 때문에 각 다른 사용자 인증패턴을 생성할 수 있다. 인증정보는 아이디나 패스워드와 같은 동일한 정보, 아이피, 운영체제, 웹 브라우저 등 변경될 수 있는 정보, 순서번호나 요청번호와 같은 반드시 변경되는 정보들로 구성된다. 이는 동일한 디바이스를 통해 인증벡터를 재생성 하더라도 각 다른 인증벡터를 생성하게 된다. 그림 5 다중 인증벡터간의 좌표 계산 방법은 수식 (2)와 같다.

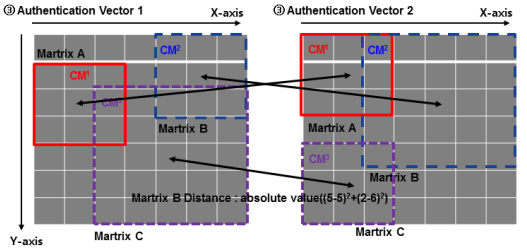

그림 5. 인증벡터 사이의 좌표거리 계산
Fig. 5. Coordinates Distance Calculation between Authentication Vector

$$dist(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} \quad (2)$$

K-nearest neighbor의 유클리드 거리를 이용하여 인증벡터 사이의 좌표 거리를 계산한다. 위 인증벡터 1, 2에서 행렬 B의 거리는 4이다. 타임스탬프를 이용한 좌표를 사용하기 때문에 행렬의 크기와 좌표간의 거리가 지속적으로 증가하게 된다. 표 7은 좌표 거리 계산에 대한 요구사항 나타낸다.

표 7. 좌표거리 계산의 요구사항
Table 7. Requirement of Coordinates Distance Calculation

	Description
Authenticati on number	인증과정을 2회 이상 수행(최소 2개)한 경우 거리를 계산할 수 있다.
Size	초기 생성된 인증벡터 전체 크기와 같거나 커야 한다.
Time	타임스탬프로 인해 생성되는 모든 정방행렬 좌표 거리는 반드시 증가해야 한다.
Distance	좌표 거리의 차이는 24시간 - 현재 시각의 좌표 거리 보다 작아야 한다.
Distance Reset	최근 인증을 수행한 인증벡터가 24시간(하루) 이상경과 되었을 경우 거리 값을 초기화 한다.

패턴 10, 11, 28에서 각 행렬 좌표의 거리는 3, 4, 5이

다. 이후 새로 생성된 인증벡터 쌍의 거리 비교를 지속적으로 비교한다. 새로 생성된 인증벡터 쌍(최소 2개)은 이전 인증벡터 쌍의 좌표거리에서 타임스탬프만큼 이동한 거리에서 벗어날 경우 세션을 종료하게 된다.

5. 암호화된 인증정보 추출

인증벡터 구조 검사, 거리 값 검증을 정상적으로 수행한 이후 삽입된 데이터를 추출한다. 인증벡터에 저장되는 실제 정보는 패턴 번호 10, 11, 28과 암호화된 인증정보만을 저장하고 있다. 삽입된 좌표를 찾기 위해서는 해당 패턴번호의 정방행렬 가운데 좌표, 정방행렬 크기, 요청 번호가 요구된다. 그림 6은 암호화된 정보의 좌표 추출 방법을 나타낸다.

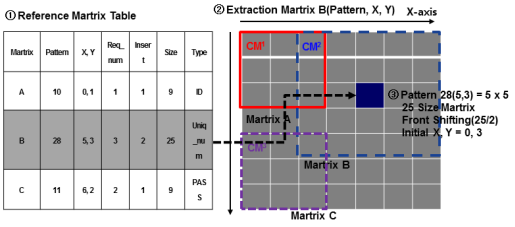

그림 6. 암호화된 정보의 좌표 추출
Fig. 6. Extracting Coordinates of Encrypted Authentication Information

행렬 B의 경우 패턴 번호 28에 대한 정방행렬 전체 크기 25와 가운데 좌표 5, 3을 변수로 3 × 3의 행렬의 초기 좌표 0, 3에서 암호화된 데이터를 추출할 수 있다.

6. 사용자 인증패턴의 활용

표 8은 인증벡터 내 인증정보의 실제 예를 나타낸다. 201.111.111.xxx의 IP 대역을 사용하는 영문 윈도우 7과 우분투 데스크톱에서 익스플로러와 구글 크롬 웹브라우저를 이용하여 로그인을 수행한 예이다. 인증벡터에서 추출되는 인증정보는 사용자의 로그인, 세션이 생성된 이후 특정 애플리케이션 사용 정보 등 다양한 사용자 행동 패턴을 지속적으로 저장한다. Apriori를 활용하여 특정 인증정보가 포함된 비율인 지지도(SP)를 계산하는 방법은 수식 (3)과 같다. N은 전체 인증정보 항목 개수, x는 인증정보 항목을 의미한다.

표 8. 사용자 인증 정보(예)
Table 8. User Authentication Information(Example)

Authentication Vector 1	Authentication Vector 2
Guest1	Guest1
Password	Password
11496	11496
0000	0001
201.111.111.111	201.111.111.112
Windows 7	Ubuntu 14.04
Internet Explorer 10	Google Chrome
20151031113686919 9	20151031113687823

$$SP = \frac{count(x)}{N} \quad (3)$$

표 9는 다중 인증벡터 예에서 나타날 수 있는 인증 정보 그룹의 예를 나타낸다. 각 인증벡터에서 공통적으로 발생한 인증정보 그룹은 Guest, Password, 201.111.111.112, Ubuntu 14.04, Google Chrome)과 (Guest, Password, 201.111.111.110, Windows 7, Internet Explorer 10) 2가지 유형으로 나타난다.

표 9. 사용자 인증 정보 그룹(예)
Table 9. User Authentication Information Group(Example)

Vector	Authentication Information Group
Vector 1	Guest, Password, 0000, 201.111.111.112, Ubuntu 14.04, Google Chrome, 20151031113687823
Vector 2	Guest, Password, 0001, 201.111.111.112, Ubuntu 14.04, Google Chrome, 20151031113687855
Vector 3	Guest, Password, 0002, 201.111.111.110, Windows 7, Google Chrome, 20151031113687860
Vector 4	Guest, Password, 0003, 201.111.111.110, Windows 7, Internet Explorer 10, 20151031113687865
Vector N	Guest, Password, 0004, 201.111.111.112, Ubuntu 14.04, Google Chrome, 20151031113687870

표 10은 각 정보 유형 별 지지도(SP)를 나타낸다.

표 10. 인증정보 유형 별 지지도(SP)
Table 10. Type of Authentication Information(SP)

Type	Information	SP(X)
ID	Guest	6/6 = 1.00
PASSWORD	Password	6/6 = 1.00
Uniq_NUM	11496	6/6 = 1.00
SEQ_NUM	-	모두 다름
IP	201.111.111.110	3/6 = 0.50
	201.111;.111.112	
OS_INFO	Windows 7	3/6 = 0.50
	Ubuntu 14.04	
Web Browser	Internet Explorer 10	3/6 = 0.50
	Google Chrome	
TimeStamp	-	모두 다름

지지도와 신뢰도는 인증정보 데이터 패턴에 따라 지속적으로 변경될 수 있다. 지지도(SP)를 기반으로 측정된 신뢰도(CON)를 계산하는 방법은 수식 4와 같다.

$$CON(X \rightarrow Y) = \frac{SP(X, Y)}{SP(X)} \quad (4)$$

Apriori는 모든 데이터의 가능한 조합을 예상할 경우 성능이 비효율적이지만 연산 항목을 명확하게 정의한 경우 높은 성능 효율성을 제공한다. 표 11은 사용자 인증정보 그룹의 신뢰도(CON)을 나타낸다.

ID <-> PASSWORD는 신뢰도 1로써 반드시 존재해야 한다. ID <-> IP의 경우 특정 사용자가 .112 IP를 사용했을 때 신뢰도가 0.60으로 다른 IP를 사용할 수 있다는 것을 나타내며, 반대로 해석하면 .112의 IP는 신뢰도가 1.00으로 반드시 현재 사용자가 이 주소를 사용해야 한다는 것을 의미한다.

표 11. 인증정보 유형 별 신뢰도
Table 11. Type of Authentication Information(CON)

연관 규칙(X->Y)	SP(X, Y)	CON = SP(X, Y) / SP(X)
ID -> PASSWORD	1.00	1.00 / 1.00 = 1.00
PASSWORD -> ID	1.00	1.00 / 1.00 = 1.00
ID -> IP(.112)	0.60	0.60 / 1.00 = 0.60
IP(.112) -> ID	0.60	0.60 / 0.50 = 1.20
ID -> Windows	0.40	0.40 / 0.50 = 0.80
ID -> Chrome	0.60	0.60 / 0.50 = 1.20
Windows 7 -> Chrome	0.20	0.20 / 0.50 = 0.40

운영체제의 경우 Windows 7이 0.39 신뢰도, Windows 7에서 크롬 웹 브라우저 신뢰도가 0.32로 사용비율이 둘 다 낮다는 것을 의미한다. 연관규칙(X->Y)에 따라 누적된 지지도와 신뢰도는 비정상적인 사용자의 탐지를 자동화 하는데 데이터로 활용 될 수 있다.

Ⅳ. 효율성 및 안전성 분석

그림 7, 표 12는 성능 분석 환경과 구성요소 및 설정을 나타낸다. 인증정보 전송을 위해 아이디, 패스워드 기반 사용자 로그인을 수행하였다. 각 구성요소의 인증벡터 생성 및 검증 파라미터는 중요한 연산 파라미터가 될 수 있기 때문에 반드시 분리되어야 한다. 서비스 제공자는 사용자 등록 및 인증과 머신러닝 기능을 수행하여 인증정보 테이블을 저장하고, 플랫폼 제공자는 인증벡터 만을 저장하는 구조로 설계되었다.

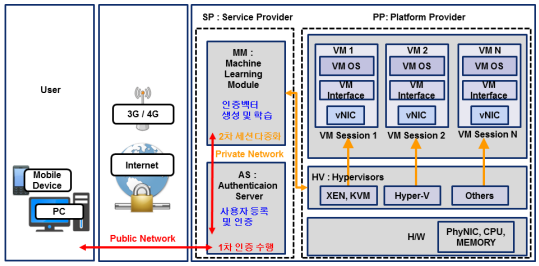

그림 7. 성능 분석 환경
Fig. 7. Performance Analysis of Environments

표 12. 구성요소와 프로그램 설정
Table 12. Components and Program Settings

	Operating System	Application
User	Windows 7 64bit Intel® Core2Duo 2.66 Ghz 6G Memory	Google Chrome, Internet Explorer 11
SP(AS)	Linux Kernel 2.6.32(Ubuntu 14.04 64bit) Intel® Core2Duo 3.16 Ghz 12G Memory	Apache Tomcat 7.0, Jmeter
SP(MM)		Login and Machine Learning(Servlet)
PP(HV)	Linux Kernel 3.16(Ubuntu 14.04 64bit) Intel® Core2Quad 2.66 Ghz 8G Memory	Apache Spark 1.5.1 OpenStack(juno) - KVM

1. 인증벡터 효율성 분석

인증벡터의 생성 및 검증에 비용이 각 플랫폼에 데이터 처리 효율에 끼치는 효율성을 분석한다. 표 13, 표 14, 그림 8, 그림 9는 인증벡터 생성 및 추출 과정의 지연시간과 실시간 데이터 처리량 비교를 나타낸다.

표 13. 인증벡터의 생성 및 검증시간
Table 13. Generating Authentication Vector and Verification Time

Round	Generation	Encrypt and Insert	Decrypt and Extract	Total Time (avg)
1	0.202	0.011	0.134	0.347
10	0.274	0.011	0.126	0.411
100	0.313	0.012	0.139	0.464
Not applied Time(avg)	0.115(TCP) + 0.436(SSL Handshake)			0.551

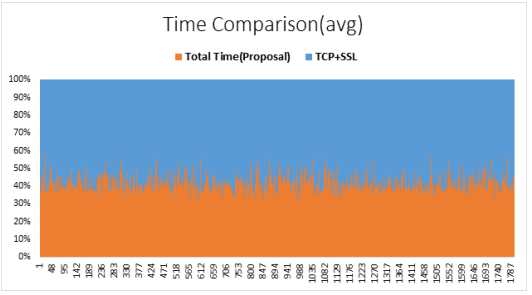

그림 8. 인증벡터의 생성 및 검증시간(차트)
Fig. 8. Generating Authentication Vector and Verification Time(Chart)

SSL 인증과정을 포함하는 기본 통신 시간은 0.551(초) 100% 기준으로 정의되었다. 지연시간 분석 결과 인증벡터 전체 연산시간 평균 비율은 1회(63%), 10회(75%), 100회(85%)로 비교적 지연시간이 효율적인 것을 확인하였다.

표 14. 인증벡터의 실시간 데이터 처리량
Table 14. Real-time Data Throughput of Authentication Vector

	Minimum	Maximum	Average
Throughput (1)	29.0Mb	47.0Mb	37.4Mb
Throughput (10)	33.1Mb	54.7Mb	43.2Mb
Throughput (100)	41.1Mb	56.2Mb	48.5Mb
Not applied Throughput	29.8Mb	43.5Mb	38.3Mb

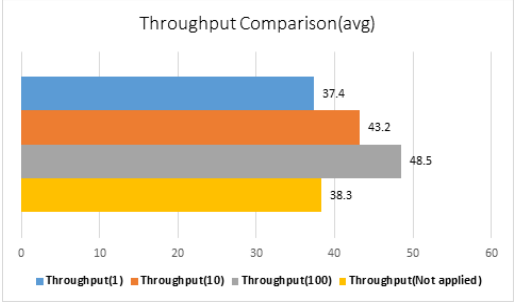

그림 9. 인증벡터의 실시간 데이터 처리량(차트)
Fig. 9. Real-time Data Throughput of Authentication Vector(Chart)

인증벡터의 개수가 증가할수록 지연시간이 다소 증가함을 확인할 수 있다. 인증벡터의 좌표 선정에 타임스탬프를 좌표 값 사용했기 때문에 전체 행렬크기 증가와 함께 연산비용이 증가한 것으로 분석된다.

인증벡터 생성과정을 포함하지 않는 실시간 처리 전송률은 평균 38.3Mb로 측정되었다. 실제 한 사용자가 1세션에 100회 이상의 다중 세션을 생성하기 위해서는 수십 번의 인증과정을 동시에 수행해야 한다. 일반적으로 2, 3개의 디바이스를 이용하여 동시 로그인을 수행했다고 가정했을 때 10회 정도의 인증벡터 생성이 적절한 것으로 분석된다. 1회 37.4Mb(0.9Mb 감소)로 처리율에 영향을 끼치지 않았으며, 10회 43.2Mb(4.9Mb 증가)에서 다소 증가한 것으로 나타났다. 100회의 경우 48.5Mb (10.2Mb 증가)로 다중 세션을 유지하는데 처리해야 하는 데이터 용량이 인증벡터 개수에 따라 증가한다는 것을 알 수 있다.

2. 인증벡터 안전성 분석

패턴 테이블 정보 유지에 대한 정책은 보안 어플리케이션 설계 단계에서 고려되어야 할 부분이며, 실제 구현되었을 경우 사용자 보안 정책 설정이나 서비스 제공자의 보안 정책에 따라 변경 되어야 한다. 표 15, 16은 인증벡터 테이블과 인증벡터의 안전성 분석을 나타낸다.

표 15. 인증벡터 테이블의 안전성
Table 15. Security of the Authentication Vector Table

	Description
Pattern Number	현재 세션의 패턴테이블에 일치되는 인증벡터가 요구되기 때문에 패턴번호만으로는 의미가 없다.
Center x, y	실제 삽입되어 있는 데이터의 위치를 찾기 위해서는 가운데 좌표로부터 정방행렬 사이즈를 알아야하기 때문에 의미가 없다.
Request Number	패턴 번호를 정렬하는 용도로 사용되기 때문에 요청번호 만으로는 의미가 없다.
Insert	겹침 부분에 대한 확인 값으로 신뢰도 측정을 위해서는 반드시 인증벡터로부터 지지도를 추출해야하기 때문에 의미가 없다.

저장되어야 하는 주요 중요 정보는 인증벡터 테이블과 생성된 인증벡터이다. 보안 세션이 성립된 경우 서비스 제공자(SP(AS))는 인증벡터 테이블을 저장하고 서비스 플랫폼(PP(HV))은 해당 인증벡터를 저장한다.

각 파라미터 분석 결과 서비스 제공자가 패턴 테이블에서 유지하고 있는 정보는 반드시 플랫폼 제공자의 인증벡터를 전송받은 후 암호화된 데이터 탐색을 위한 정보들로 구성되어 있음을 확인할 수 있다.

표 16. 인증벡터의 안전성 분석
Table 16. Security of Authentication Vectors

	Description
Pattern Number	인증벡터 생성 시 삽입된 무작위의 정수 열을 패턴번호로 사용하기 때문에 패턴번호만으로는 의미가 없다. 반드시 패턴 테이블 정보가 요구된다.
Center x, y	정방행렬 사이즈만으로는 삽입된 초기 좌표를 탐색할 수 없다. 모든 좌표를 탐색 한다고 해도 패턴 테이블에 일치되는 인증벡터를 알아낼 방법이 없다.
Encrypted Data	인증벡터와 일치되는 인증벡터 테이블을 보유했다고 가정했을 때 현재 세션에서 공유된 비밀 키 탐색 비용이 추가적으로 요구된다.
Authenticati on Vector Distance	좌표 거리검사는 인증패턴 생성할 때 마다 새로운 검사를 수행한다. 이는 지속적으로 새로운 거리 값을 추출하기 때문에 이전 누적된 거리 값은 의미가 없다.

실제 사용자의 중요 정보는 인증벡터에 삽입되어 있다. 이는 인증패턴과 암호화된 인증정보의 추출을 위해서는 현재 세션이 유지되는 시간 내에 검증과정을 모두 우회하고 서비스 제공자, 플랫폼 제공자의 서버를 모두 해킹 가능해야 한다. 인증벡터만 보유하고 있는 경우 내부 데이터 탐색과 암호화된 정보의 키 탐색, 누적된 신뢰도에 대한 비정상 사용자 필터링 등 공격에 대한 비용이 높다. 또한 인증벡터의 패턴거리 검증(거리가 지속적으로 변화)으로 인해 재전송 공격에도 안전하다고 할 수 있다.

V. 결 론

본 논문은 현재 인공지능 분야에서 이슈화 되고 있는 머신러닝 기법을 사용하여 학습 가능한 보안 인증벡터에 대해 제안하였다. 가상화 환경의 중요 데이터 보안과 기술적인 문제점들을 분석하고 이에 대한 해결책을 제시하였다. 가상화 영역에서 전달되는 중요정보를 인증벡터로 생성하고 구조 검증, 좌표 거리 검증 등 기존 인증과는 다른 보안 기능을 수행할 수 있음을 확인하였다. 또한 정의된 인증정보 항목을 이용하여 인증벡터의 생성과 데이터를 추출 할 수 있도록 범위를 축소하여 실시간 처리에 적합하도록 설계되었다. 또한 패턴 테이블과 인증벡터 정보를 독립적으로 분리하여 안전성을 입증하였다.

최근 인공지능 분야의 발전은 새로운 하이퍼바이저 구조의 차세대 서버 플랫폼, 딥 러닝을 위한 발전된 기계학습 알고리즘 등 관련 기술들에 대한 발전을 앞당기고 있다. 그러나 새로운 가상화 기술의 등장은 새로운 보안 취약성들을 낳기 때문에, 가상화 보안 기술은 인공지능 분야의 발전과 함께 다양한 연구 및 개발이 진행되어야 할 것이다.

References

[1] Lee Jeong, Yoon Hyukjin, “2015 Begins Popularized of Drones”, Eugene Ivestment & Securities, Monthy Global IT No. 13, 2015.
[2] Ryu Hanseok, “Platform as a Drones and Implications”, Digieco Report, Issue & Trend, 2015.
[3] Kim Iljoong, Eli Hong, “Content Industry Trend of USA”, Weekly in Depth Issues, Vol. 6, No. 16, 2016.
[4] Kim Seokwon, An Sungwon, Chu Hyungseok, “AI AlphaGo, The Artificial Intelligence of Google’s Baduk Win the Champions Humans”, SPRI Issue Report, Vol. 1, 2016.
[5] Synergy Research Group, “The Big Four Cloud Providers are Leaving the Rest of the Market Behind”, 2015.
[6] Kwon Aera, “Change and Countermeasures in the IT Ecosystem, According to a Spreading Cloud Services”, Korea IT Service Industry Association, Industry Issue pp. 80-103, 2010.
[7] CVE-2012-0217, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2012-0217
[8] CVE-2014-0160, http://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-0160
[9] CVE-2015-3245, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3245
[10] CVE-2015-0012, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-0012
[11] CVE-2015-0235, http://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-0235
[12] Kim julie, Bahn hyokyung, “An Efficient Log Data Management Architecture for Big DataProcessing

in Cloud Computing Environments", The journal of the Institute of Internet Broadcasting and Communication(JIIBC), Vol. 13, No. 2, pp. 1-7, 2012.

[13] Jung Hyeonjun, "Trends in Virtualization Technologies and Key Issues(I)", Koerea Information Society Development Insitute, Vol. 25, No. 3, pp. 63-92, 2013.

[14] Sin Youngsang, "Hypervisor-based Security Technology Trends in Cloud Environments", Korea Information Security Agency, Internet & Security Focus, pp. 55-75, 2014.

[15] Kotsiantis, Sotiris, Zaharakis, Pintelas, "Supervised machine learning: A review of classification techniques", Emerging Artificial Applications in Computer Engineering, IOS Press, pp. 3-24, 2007.

[16] Lewis, David, "Naive (Bayes) at forty: The independence assumption in information retrieval", Machine Learning: ECML-98, pp. 4-15, 2005.

[17] Weinberger, Saul, "Distance metric learning for large margin nearest neighbor classification", The Journal of Machine Learning Research, pp. 207-244, 2009.

[18] Jovanoski, Viktor, Nada Lavrac, "Classification rule learning with APRIORI-C", Springer Berlin Heidelberg, pp. 44-51, 2002.

저자 소개

최 도 현(정회원)

- 2008년 2월 : 동서울대학교 컴퓨터소프트웨어학과 졸업
- 2010년 8월 : 숭실대학교 컴퓨터학과 석사
- 2016년 3월 : 숭실대학교 컴퓨터학과 박사

<주관심분야 : Mobile, Network Security, PKI, Virtualization>

- E-mail : cdhgod0@ssu.ac.kr

박 중 오(정회원)

- 2000년 7월 : 성결대학교 컴퓨터공학과 졸업
- 2003년 3월 : 명지대학교 전자계산교육 석사
- 2011년 8월 : 숭실대학교 컴퓨터공학 박사
- 2013년 3월 ~ 2016년 2월 : 동양미래대학교 조교수
- 2016년 3월 ~ : 성결대학교 조교수

<주관심분야 : PKI, Network security, 암호학>

- E-mail : jopark02@sungkyul.ac.kr