https://doi.org/10.33778/kcsa.2023.23.2.037

Markov Chain을 이용한 기만환경 침입 공격자의 기만 여부 예측 모델에 대한 연구★

유 선 모*, 위 성 모**, 한 종 화**, 김 용 현*, 조 정 식***

요 약

사이버 기만 기술은 공격자의 활동을 모니터링하고 새로운 유형의 공격을 탐지하는 데 중요한 역할을 한다. 그러나 기만 기술의 발전과 더불어 Anti-honeypot 기술 또한 발전하여 기만환경임을 알아챈 공격자가 기만환경에서의 활동을 중단하거나 역으로 기만환경을 이용하는 사례들도 존재하지만 현재 기만 기술은 이러한 상황을 식별하거나 대응하지 못하고 있다. 본 연구에서는 마코프 체인 분석 기법을 이용하여 기만환경에 침입한 공격자의 기만환경 식별 여부 예측 모델을 제안한다. 본 연구에서 제안하는 기만 여부 판단 모델은 확인한 바로는 공격자의 기만환경 식별 여부를 판단하기 위한 최초의 시도이며 기만환경을 식별한 공격자를 고려하지 않는 기존의 기만 기술 기반 공격자 분석에 대한 연구의 제한사항을 극복할 수 있을 것으로 예상한다. 본 연구에서 제안한 분류 모델은 기만환경임을 식별하고 활동하는 공격자 분류에 97.5%의 높은 정확도를 보였으며 공격자의 기만환경 식별 여부 예측을 통해 수많은 기만환경 침입 데이터 분석 연구에 정제된 데이터를 제공할 수 있을 것으로 기대된다.

A Study on the Model for Determining the Deceptive Status of Attackers using Markov Chain

Sunmo Yoo*, Sungmo Wi**, Jonghwa Han**, Yonghyoun Kim*, Jungsik Cho***

ABSTRACT

Cyber deception technology plays a crucial role in monitoring attacker activities and detecting new types of attacks. However, along with the advancements in deception technology, the development of Anti-honeypot technology has allowed attackers who recognize the deceptive environment to either cease their activities or exploit the environment in reverse. Currently, deception technology is unable to identify or respond to such situations. In this study, we propose a predictive model using Markov chain analysis to determine the identification of attackers who infiltrate deceptive environments. The proposed model for deception status determination is the first attempt of its kind and is expected to overcome the limitations of existing deception-based attacker analysis, which does not consider attackers who identify the deceptive environment. The classification model proposed in this study demonstrated a high accuracy rate of 97.5% in identifying and categorizing attackers operating in deceptive environments. By predicting the identification of an attacker's deceptive environment, it is anticipated that this model can provide refined data for numerous studies analyzing deceptive environment intrusions.

Key words : deception technology, probabilistic analysis, markov chain, deceptive status, anti-deception

접수일(2023년 06월 05일), 수정일(2023년 6월 12일), 게재확정일(2023년 06월 30일)

★ 본 연구는 2021년 국방과학연구소 주관 미래도전국방기술 연구개발사업(UD210030TD)의 지원을 받아 연구되었음.

* 주식회사 앰진/AIS 연구소

** 한국인터넷진흥원 탐지대응팀

*** 한국인터넷진흥원 탐지대응팀

1. 서 론

기만 기술은 더욱 정교하게 발전하는 공격에 대하기 위한 기술 중 하나로 공격 정보 수집에 상당히 중요한 역할을 하고 있다. 학계와 업계 모두 기만 기술을 이용하여 공격에 대해 능동적인 탐지와 방어뿐만 아니라 기만 환경을 통해 의미있는 공격 정보를 수집하기 위한 연구가 활발하다. 실제 서비스가 아닌 가짜 서비스로 운영되어 기만환경에 접근하는 대상은 모두 의도를 갖고 접근하기 때문에 정교하게 만들어진 기만 환경은 공격 정보를 수집하고 이를 기반으로 최근 공격 동향을 분석하거나 제로데이 공격을 찾아내기에 최적화된 환경으로 평가된다.

그러나 기만 기술의 발전과 동시에 기만 환경을 식별해내는 기술 또한 같이 발전하고 있어 공격자들은 기만 환경을 식별하여 공격 행위를 중단하거나, 고수준의 공격자들은 기만환경을 역 기만하여 잘못된 정보를 주입하거나 기만환경을 공격 자원으로 활용하기도 한다. 이러한 경우 수집하는 공격 정보가 오염되어 앞서 언급한 공격 유형 분류 및 예측 모델은 오히려 잘못된 결과를 가져올 수 있다.

본 연구에서는 기만환경에 접근하여 연속적인 공격을 수행하는 소스 IP에 대해 마코프 체인 분석을 이용하여 기만환경 식별 여부를 판단한다. 기만환경에서 수집한 공격 유형과 식별된 전체 공격에서 각 공격 유형이 발생하는 빈도를 공격 유형의 발생 확률로 정의한다. 또한 각각의 소스 IP 단위로 식별된 공격 이력을 확률과정으로 정의하고, 연속적으로 공격을 수행하는 기만당한 공격자와 기만환경임을 감지하여 활동하는 공격자를 분류한다.

본 연구를 통해 기여할 수 있는 바는 다음과 같다.

- 기만환경을 식별하여 정보를 오염시키거나 기만환경을 공격 자원으로 이용하는 공격자 식별
- 기만환경에서 수집된 정보를 기반으로 공격 유형을 분류하고 예측하는 연구 정확도 개선을 위해 정제된 데이터 제공
- 기만환경 식별 공격자에 대응하기 위한 별도의 기만 전술 전개

본 논문의 구성은 다음과 같다. 2장에서는 기만 기술을 이용하여 공격을 분류하고 예측하는 기존 연구의 특징 및 장단점에 대해 기술하고 설명한다. 3장에서는 제안하는 마코프 체인 분석 기반 공격자 기만환경 식별 여부 예측 모델의 세부 과정에 대해 설명한다. 4장에서는 제안하는 기법을 적용한 실증 결과를 제시한다. 5장에서는 제안 모델의 제한사항과 이어지는 연구에 대해 논의하고 마지막으로 5장에서는 결론으로 마친다.

2. 관련 연구

2.1 기만 기술

사이버 보안 분야에서 처음으로 기만 기술이라는 용어가 등장한 것은 1989년 Cliff Stoll의 저서 “The Cuckoo’s Egg”를 통해서이다 [1]. Cliff Stoll은 공격자에게 혼란을 야기하여 공격을 방해하거나 지연시킬 목적으로 가상의 사용자 계정으로 구성되고 가짜 파일들을 포함한 가상 시스템을 만드는 방법에 대해 설명했다. 2000년 이후, Lance Spitzner는 기만 기술에 대해 활발한 연구를 수행한 학자 중의 하나이다 [2]. 2001년 Lance Spitzner는 공격자 기만을 위한 가상의 환경을 Honeypot이라고 정의했다. 또한, Spitzner는 외부 위협에 대한 탐지와 정보 수집을 위해 사용되던 Honeypot 기술을 내부자에 의해 발생하는 위협 탐지 방법으로 제안하기도 했다. 이후 Spitzner는 파일, 데이터베이스 엔트리, 네트워크 포트 등 공격자 유인에 사용되는 모든 자원을 지칭하는 Honeytoken이라는 용어를 처음 제안하여 다양한 유형의 자원을 이용한 기만 기술 전개의 기반을 다졌다.

기만 기술은 상호작용 정도에 따라 Low-Interaction, High-Interaction 허니팟으로 구분된다. 간혹 그 중단 단계로 Medium-Interaction 허니팟을 주장하는 연구도 존재한다. Low-Interaction 허니팟은 정상 네트워크에 통신이 가능한 가짜 서버 또는 서비스로 설치되어 공격자가 접근할 수 있도록 프로토콜과 포트를 열어놓고, 허니팟에 접근하는 모든 트래픽을 침해 트래픽으로 간주하여 감지하고 알람하는 것을 목표로 한다. High-Interraction 허니팟은 SSH, Telnet, SMT

P 등의 실제 이용 가능한 서비스를 가짜로 구축하여 공격자와의 상호작용을 통해 공격자를 속이고 정보를 얻어낼 목적으로 설치된다. 2004년 개발된 허니팟 프레임워크 Honeyd는 Low-Interaction을 지원하는 대표적인 도구이며 2013년 개발된 Conpot은 Low/High-Interaction을 모두 지원하는 허니팟 프레임워크이다.

2.2 Anti-Deception 기술

사이버 기만 기술의 발전과 동시에 기만 환경을 감지하기 위한 기술도 함께 발전하였다. 공격자들은 조사 활동을 통해 기만 환경을 감지하여 공격 활동의 노출을 방지한다. 또한, 숙련된 공격자들은 기만 환경에 잘못된 정보를 노출하여 역으로 시스템에 혼란을 주거나 기만 자원을 공격 자원으로 활용하기도 한다. Low-Interaction 허니팟의 경우 단순 서비스 포트 오픈의 형태로 공격자가 잘못된 정보를 전달하거나 공격 자원으로 활용하기에 제한적이지만, High-Interaction 허니팟의 경우 실제와 같이 서비스 이용이 가능하기 때문에 정보를 오염시키거나 공격 자원으로의 활용이 가능하여 기만 환경임이 노출되었을 때 더 위험하다고 볼 수 있다. 학자들은 이러한 위험에 대응하기 위해 공격자들로부터 기만 환경이 식별되지 않도록 Anti-Honeypot 기술을 분석하고 탐지하는 연구를 수행하였다.

기만 환경 감지 기술은 크게 네트워크, 시스템, 애플리케이션 레이어에서 이루어진다. 네트워크 레이어에서의 기만 환경 감지 기법은 2006년 Xinwen Fu[3]과 2007년 Mukkamala[4]에 의해서 이루어졌다. Xinwen Fu는 허니팟과 허니팟 상단 게이트웨이 각각의 RTT 비교 분석을 통해 가상화 기반의 허니팟을 감지하였고 Mukkamala는 허니팟에서의 ICMP 응답속도를 이용하여 감지하는 기법을 연구하였다. 또한, TCP 헤더에 대한 SVM 알고리즘 적용을 통해 허니팟임을 예측하는 연구도 수행하였다. 2005년 T.Holz와 F.Raynal는 시스템 레이어에서의 허니팟 감지 기법에 대해 연구하였다. 리눅스 시스템에서 조회할 수 있는 시스템 아키텍쳐, CPU, Memory, Disk 등의 하드웨어 정보에 대한 핑거프린팅을 통해 가상 서버임을 식별하고 동작하고 있는 디버깅 도구를 탐지하여 기만 환경임을 감지하는 역를 수행하고 커널 파라미터의 수정을 통해 대응할 수 있는 방안을 제시하였다. 2004년 N.Krawetz는 최초로 애플리케이션 레이어에서의 조사를 통해 허니팟을 감지하는 사례와 대응 방안을 연구하였다. Send-Safe에서 운영하는 가짜 프록시 기반의 스팸메일 허니팟을 감지하는 허니팟 헌터들과 사용 기법에 대해 연구하고, 다시 허니팟 감지 기술을 방어하여 허니팟 헌터들을 기만하는 기술에 대해 연구하였다. 레이어별 Anti-Deception 기술은 [표 1]과 같이 정리할 수 있다.

<표 1> 레이어별 Anti-Honeypot 기술 연구

Layer	저자	연구주제
애플리케이션	N.Krawetz	스팸메일 허니팟 감지 대응
시스템	T.Holz, F.Raynal	시스템에서의 핑거프린팅 및 디버깅 도구 탐지 기반 허니팟 감지
네트워크	Xinwen Fu	RTT 측정 및 비교 기반 가상화 허니팟 탐지
	Mukkamala	ICMP, TCP 응답속도 측정 기반 가상화 허니팟 탐지

N.Krawetz의 연구와 같이 특정 애플리케이션 허니팟에 대한 기만 환경 감지 시도 탐지 및 우회 기법은 높은 정확도를 보일 수 있지만 모든 애플리케이션에 대해 하니팟 감지 우회 기술을 적용하는 것은 불가능하다. 또한, 네트워크 레이어에서의 허니팟 감지 시도 탐지에 대한 정확도는 아직 연구된 사례가 존재하지 않고 일시적인 대응 방안만을 제시하고 있다. 본 연구에서는 계속해서 변화하는 기만 환경에서 공격자의 기만 환경 감지 여부를 식별하기 위해 마코프 체인 기반의 예측 모델을 제안한다.

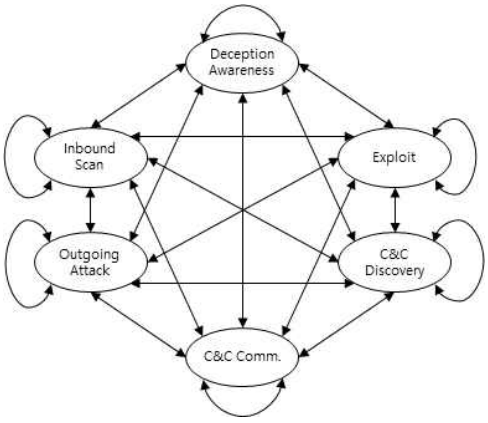
3. 제안 모델

이 섹션에서는 공격자의 기만 환경 감지 여부를 예측하는 모델을 제안한다.

3.1 상태 정의

Zainab.A.의 연구에서 공격자가 공격 목적 달성을

위해 수행하는 공격 순서를 정의했으며 본 연구에는 Zainab.A.가 정의한 공격 유형 및 순서를, 취약점을 공개하여 공격자를 유인하는 기만 환경에 맞게 변경하여 사용한다. 공격 상태는 기만 환경 정찰을 위한 스캔, 기만 환경 감지 시도, 취약점에 대한 익스플로잇, 익스플로잇에 의한 감염, C&C 통신 및 외부 자원에 대한 공격 및 정보 탈취로 분류하였으며 각 상태 간 관계는 그림 1과 같다. 공격 상태 간 순서에 대해 논의하자면 각 상태는 다른 모든 상태로 전이가 가능한 것으로 정의하였고 특히 공격자의 반복적인 시도 등으로 자체 전이 또한 가능한 것으로 정의하였다.



(그림 1] 공격 상태 도메인 및 전이 방향

그림 1의 포괄적인 모델을 적용하면 모델의 각 상태에 속하는 활동을 감지할 수 있다. 각 활동을 감지하기 위해 기만 환경에서 발생하는 모든 트래픽과 단말에서 발생하는 로그, 파일 시스템 변경 등을 모니터링하고 NIDS 및 HIDS를 통해 각 활동을 탐지한다. 본 연구에서는 Emerging Threats의 탐지룰 세트와 업계 표준 봇넷 탐지 IDS인 Suricata를 사용한다. 또한 엔드포인트에서는 EDR을 통해 단말에서 발생하는 악성 활동을 탐지한다.

<표 2> 각 공격 활동이 탐지되는 상태 및 설명

상태	설명
Inbound Scan	여러 호스트에서 스캔되는 동일한 포트(horizontal scan) 또는 한 호스트에서 스캔되는 여러 포트(vertical scan)로 특징지어지는 호스트의 모든 포트에서 들어오는 포트 스캔
Deception Awareness	기만 환경을 감지하기 위한 네트워크 레이어에서의 반응속도 측정 행위 및 시스템 레이어에서의 핑거프린팅
Exploit	피해자 시스템 또는 웹 브라우저에서 "백도어" 액세스 또는 원격 실행 권한을 얻기 위해 애플리케이션 또는 시스템 취약성을 대상으로 하는 모든 공격
C&C Discovery	HTTP를 통한 서버 목록에 대한 연결 시도 또는 가능한 도메인 목록에 대한 많은 수의 DNS 쿼리 등 Botnet의 C&C 서버에 대한 접속 시도
C&C Communication	발견된 자신의 C&C 서버와의 IRC, HTTP 또는 사용자 정의 프로토콜을 이용한 통신
Outgoing Attack	외부 자산에 대한 자가 전파, 포트 스캐닝 또는 대량 이메일 발송, 정보 도용, 피싱, DoS 또는 스팸과 같은 기타 공격

3.3 Markov Chain 모델

제한 분포의 유도를 위해 Markov 모델의 Balance Condition을 적용한다. 즉, 상태를 떠날 확률은 들어갈 확률과 같아야 하며 이를 통해 모델의 각 상태에 대한 Balance Equation을 수립할 수 있다. 일반적으로 n개의 상태 집합 x에서 들어오는 전이가 있고 m개의 상태 y집합으로 나가는 전이가 있는 각 상태 i는 다음과 같은 식 (1)로 표현할 수 있으며 P_i는 각 상태 i에 있을 확률, $t_{i,y}$는 상태 i에서 y로의 전이율을 나타낸다:

$$\sum_{x=1}^{n} P_x t_{x,i} = \sum_{j=1}^{m} P_i t_{i,j} \quad (1)$$

식 (1)을 기반으로 그림 1에 표시된 모델의 각 상태 $S_i(\forall i_{1..6})$에 대해 $i=1$ 인 경우 다음과 같이 균형 방정식을 정의할 수 있다.

$$P_2 t_{2,1} + P_3 t_{3,1} + P_4 t_{4,1} + P_5 t_{5,1} + P_6 t_{6,1} = P_1 t_{1,2} + P_1 t_{1,3} + P_1 t_{1,4} + P_1 t_{1,5} + P_1 t_{1,6} \quad (2)$$

각 상태 S_i에 대한 확률 P_i를 계산하려면 먼저 전환 확률 $t_{i,j}$가 필요하며, 따라서 먼저 각 셀 (i,j)가

상태 S_i와 S_j 사이의 전이율을 나타내는 데이터 세트에서 전이 확률 행렬 T를 생성한다. 전이 확률 행렬을 계산하는 일반적인 접근 방법은 일정 시간 동안 시스템을 관찰하여 상태의 시퀀스를 관찰하여 $T_{i,j}$에 대한 확률을 생성하는 것으로 본 연구에서는 다음과 같이 6×6 크기의 전이 확률 행렬 T를 구할 수 있다.

$$T=\begin{pmatrix} 0.682 & 0.030 & 0.033 & 0 & 0 & 0 \\ 0.035 & 0.426 & 0.527 & 0.012 & 0 & 0 \\ 0.035 & 0.426 & 0.527 & 0.012 & 0 & 0 \\ 0.001 & 0.001 & 0.926 & 0.072 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

각 행과 열이 나타내는 상태는 각각 Inbound Scan, Deception Awareness, Exploit, C&C Discovery, C&C Communication 및 Outgoing Attack이며 Inbound Scan 상태에서 각 상태에 대한 전이 확률은 표 4와 같이 확인할 수 있다.

<표 3> Inbound Scan 상태에서 각 상태로의 전이 확률

상태	전이 확률($P_{1,j}$)
Inbound Scan	0.682
Deception Awareness	0.030
Exploit	0.033
C&C Discovery	0
C&C Communication	0
Outgoing Attack	0

계산된 결과는 마코프 체인의 비환원성, 비주기성 및 가역성의 속성에 대해 고정 확률 분포가 Markov 고정 분포에 대한 조건도 충족하는지 경험적으로 검증할 필요가 있다.

4. 실증 평가

이 섹션에서는 제안된 모델의 정확도를 검증하고 성능을 조사하여 평가하는 것을 목표로 한다. 3장에서 제안한 모델을 실행하고 탐지 결과 정확도를 계산하여 탐지 성능을 검증한다.

4.1. Dataset

모델을 평가하기 위해 6개월 이상 기만환경을 운영하여 수집한 데이터를 활용한다. 검증에 사용된 데이터 수집 기간은 2022년 9월 1일부터 2023년 2월 28일까지이며 사용된 도착지 IP는 총 4,778개이며 이 때 접근한 출발지 IP의 수는 4,414,372개, Flow는 총 3,772,436,978개이다. 공격은 Emerging Threat에서 제공하는 탐지정책을 Suricata IDS 엔진과 EDR을 이용하여 식별하였고 식별되지 않은 공격은 미식별 공격으로 제외하였다. 데이터 검증을 위해서 전체 출발지 IP 중 3회 이상의 연속적인 공격이 탐지된 IP를 기준으로 0.1%에 해당하는 4,000개의 소스 IP를 추출하여 기만 환경 감지 여부를 라벨링하였다.

4.2. 모델 실행 결과

마코프 체인을 이용하여 공격자의 기만 환경 감지 여부를 예측할 수 있는 정확도를 조사한다. 기본 방법론은 다음과 같다. 예측을 위해, 우리는 먼저 학습을 통해 얻은 식 (3)의 전이 확률 행렬을 사용한다. 다음으로, 상태 집합 S와 현재 상태 S_i 및 전이 확률 집합 $T_{i,j}$가 주어지면 전이 확률 $P(S_{i,j})=\max(t_{i,j})$. 즉, 현재 상태에서 가장 확률이 높은 전환의 대상이 될 때마다 공격 상태를 예측하고, 모든 단계 i에서 이 예측을 반복한다.

4.2.1. 결과 평가

따라서 3.1장의 [그림 1]에 표시된 것처럼 원본 데이터 세트에서 전환 매트릭스를 구축할 때 C&C 통신 및 공격 상태에 대한 자체 전환 확률이 다른 상태로 이어지는 전환 확률을 압도하는 것을 확인할 수 있다. C&C Communication은 0.926의 자체 전이 확률을 가지며 Attack은 0.90의 자체 전이 확률을 갖는다. 이는 예측에 부정적인 영향을 미친다. 예를 들어 C&C 상태에서 가장 가능성이 높은 전환을 예측할 때 항상 자체 전환을 예측하게 된다. 실제로 공격을 예측할 수

있는 유일한 시간은 현재 상태가 공격 상태일 때이며, 공격 상태가 그대로 유지되는지 여부가 아니라 비공격 상태가 언제 공격으로 전환되는지 알아야 하므로 실제적인 목적으로는 유용하지 않다. 따라서 각 연속 경보 세트를 단일 경보로 간주하여 자체 전환을 모두 삭제하는 것이 실행 가능한 옵션인지 여부를 고려해야 한다.

공격 상태의 경우 자체 전환 여부와 기간이 중요하지 않다. 첫째, 공격 상태 내에서 후속 자체 전환이 아니라 다른 상태에서 공격 상태로의 첫 번째 진입을 예측하는 데 관심이 있다. 둘째, 우리는 지속 시간이 아니라 공격 이벤트의 발생을 예측하는 데에만 관심이 있다. 호스트가 공격에 참여하면 더 이상 공격 상태를 유지하는 시간이 중요하지 않다. 따라서 데이터에서 공격 상태의 자체 전환을 무시하도록 선택한다.

다음 질문은 다른 상태에서도 자체 전환을 무시할지 여부에 관한 것이다. 예를 들어 현재 상태가 C&C Communication 상태라면 다음 상태 변경이 공격 상태가 될 가능성이 있는지 예측할 수 있다. 그러나 공격 상태로 전환되기 전에 C&C 통신 상태가 얼마나 오래 자체 전환되는지에 대해 고려할 필요가 있으며 공격하기 전에 비공격 상태의 자체 전환에 특정 패턴이 있는 경우 이 값이 실제로 중요하다고 결론을 낼 수 있다. 예를 들어 C&C 통신 상태가 공격 상태로 전환되기 전에 x분 동안 대부분 자체 전환 상태로 남아 있음을 알 수 있다. 이 경우 C&C 통신 상태를 볼 때마다 x분 후에 공격이 발생할 가능성이 있다는 경고를 생성하는 것이 유용하다. 이러한 패턴이 존재하는지 확인하기 위해 데이터에서 각 상태의 자가 전이 기간에 대한 경험적 분석을 수행한다.

표 4 및 표 5는 각 상태의 자체 전환의 지속 시간(분 단위) 및 수(즉, 연속 경보 수)의 통계적 속성을 각각 나타낸다. 기간이나 전환 횟수에 예측 가능한 패턴이 없다는 것을 알수 있다. 표는 거의 모든 상태의 평균에 비해 기간과 자가 전이 횟수의 표준 편차가 매우 높고 최소값과 최대값의 차이도 일반적으로 매우 크다는 것을 보여준다. 공간 제약으로 인해 여기에 표시되지 않은 개별 호스트에 대해 유사한 분석을 수행하지만 단일 봇에 감염된 개별 호스트 내에서도 패턴을 예측할 수 없으며 기간 및 숫자 값이 널리 분산되어 있다.

<표 4> 각 상태에서 자체 변이 기간의 통계적 속성

State	Min.	Max.	Mode	Mean	Std.
Exploint	< 1	8	< 1	1.2	2.2
Binary Download	< 1	16	< 1	0.74	3.2
CNC Comm.	< 1	1449	< 1	1.9	45.3
Attack	< 1	1429	< 1	4.7	58.2

<표 5> 각 상태에서 자체 변이 횟수의 통계적 속성

State	Min.	Max.	Mode	Mean	Std.
Exploint	2	64	3	17	63
Binary Download	2	14	2	24	32
CNC Comm.	2	4928	2	63	87
Attack	2	21023	2	12	763

표 4는 각 상태의 최소 지속 시간이 1분 미만임을 보여주며 이는 모든 상태에 대한 모달 값이기도 하다. 이것은 실제 지속 시간이 높은 분산으로 인해 예측할 수 없다는 사실과 결합하여 다른 상태에 따른 공격을 예측할 때마다 항상 1분 이내에 발생할 가능성이 있다고 보수적으로 예측해야 한다는 결론을 내린다. 이후 실제 지속 시간은 무의미해진다. 이 접근 방식의 단점은 실제로 공격이 몇 분 안에 발생하지 않고 몇 시간 후에 발생하는 경우 의심되는 호스트를 오랫동안 모니터링하는 데 귀중한 리소스가 낭비되거나 해당 통신이 불필요하게 제한된다는 것이다. 그러나 장기간의 자체 전환이 일반적이지 않기 때문에 이것이 허용 가능하다고 생각한다. 데이터 세트에서 총 1,233건의 공격 상태 발생 중 지속 시간이 2분을 초과하는 경우는 27건에 불과하며 마찬가지로 C&C 통신 상태의 3659회 중 170회만 2분 이상 지속된다. 따라서 자체 전환이 원래 모델의 일부이지만 이제 데이터의 모든 자체 전환을 무시하기 한다.

학습 단계에서 모든 자체 전환 시퀀스를 단일 상태로 축소하는 데이터 세트에서 전환 매트릭스를 구축

한다. 상태에 대한 여러 연속 경고는 단일 경고로 간주된다. 테스트 단계에서 상태에서 예측할 때 상태 변경이 있을 때까지 동일한 상태에 대한 추가 경고를 무시하고 다른 예측을 한다.

이제 우리는 T라고 부르는 다음과 같은 전이 확률 행렬을 얻을 수 있다:

이전과 마찬가지로 각 행과 열이 나타내는 상태는 각각 Exploit, Binary Download, C&C Communication 및 Attack이다. 자체 전환이 제거되었으므로 대각선의 값은 이제 0과 같다. 그러나 이 모델은 여전히 3장에서 논의된 모든 Markov 속성을 만족함을 확인할 수 있다. 우리는 여기서 그 증거를 보여주지 않는다.

4.2.2. 결과: Markov Chain을 사용한 공격 예측

먼저 데이터 세트를 각 호스트에 대해 시간적으로 처음 1.5시간의 데이터를 교육으로, 나머지는 테스트 데이터로 나눈다. 데이터 지속 시간이 트레이스에 따라 상당히 다르므로 일부 트레이스는 2~3시간만 지속되므로 다소 짧은 훈련 간격을 선택한다. 따라서 더 긴 훈련 간격을 사용하면 모델을 테스트하는 데 사용할 수 있는 호스트의 추적이 더 적다. 그런 다음 섹션 3.3의 행렬 T와 유사하지만 더 적은 데이터를 기반으로 구축되었기 때문에 다른 값을 사용하여 훈련 데이터에서 확률 전이 행렬을 얻기 위해 Markov 훈련 프로세스를 실행한다. 테스트 단계는 다음과 같이 진행되었다. 테스트 데이터를 반복하여 이전에 볼 수 없었던 실시간 이벤트 스트림으로 처리한다. 각 상태를 관찰한 후 다음 상태가 공격 상태일 가능성이 있는지 예측한다. 앞에서 설명한 것처럼 공격 상태로의 전환이 현재 상태에서 가능한 모든 전환 중 가장 높은 확률을 갖는지 여부를 기반으로 이 결정을 내린다.

이 발생하지 않았다. 이 실험에서 우리는 98.3%의 전체 정확도를 달성했다.

5. 결 론

기존 연구의 초점은 EDR 기반의 VM 식별 API나 네트워크 응답속도 검사를 이용하여 기만 환경을 식별하는 Anti-Honeypot 행위를 탐지하는데 맞춰져 있었다. 특히 네트워크 응답속도 기반의 탐지의 경우 일반적인 네트워크 스캔과 다르지 않아 탐지 정확도가 떨어지는 문제로 활용이 어려웠다. 시스템에서의 탐지 또한 시스템 내에서 별도의 프로세스로 모니터링을 해야하며 이 또한 감지될 수 있다는 단점이 존재하고 있었다. 따라서 이번 연구를 통해 네트워크에서 침해 행위 탐지 결과에 대한 마코프 체인 기반의 학습을 통해 기만환경 감지 여부를 탐지하는 모델을 제안하였다.

마코프 체인 기반의 공격자의 기만 여부에 대한 판단 모델을 적용한 결과 Anti-Deception 시도를 통해 기만 환경임을 감지한 공격자를 탐지할 확률이 98.4%로 높은 탐지율을 보였으며 반대로 기만 당한 공격자를 탐지하는 결과 또한 97.8%의 높은 탐지율을 보였다.

따라서 본 연구에서는 기만환경에 접근하는 공격자의 기만 여부를 판단하는데 마코프 체인이 노은 정확도를 보임을 예측할 수 있었다. 본 연구에서 보완할 사항은 워낙 많은 기만환경 데이터를 활용한 탓에 0.5%에 해당하는 공격 IP에 대한 수기 검증을 수행하였고 더 많은 대상을 검증할 수 있는 검증 방안에 대한 개선이 필요하다.

<표 6> Markov Chain 기반의 Anti-Deception 탐지 결과

State	True Positives	False Positives	True Negatives	False Negatives	Accuracy
Anti-Deception	98.4%	1.6%	97.8%	2.2%	98.3%

실험의 정확도 분석은 표 6의 첫 번째 행에 나와 있다. 1.6%의 매우 낮은 오경보 비율로 매우 높은 백분율(98.3%)의 진정한 긍정 예측을 달성했으며 공격 예측을 한 시간의 1.6%만이 바로 다음 전환으로 공격

또한, 본 연구를 바탕으로 기만환경을 감지한 공격자가 식별되면 해당 공격자에게 어떤 기만 전술을 적용하여 혼란스럽게 할지, 또는 기만 자원의 공격 역이용을 방어할지 다음 전술에 대한 논의가 가능할 것으

로 보이며 군, 공공, 민간 등 다양한 사이버 방호 체계 운영에 전술 운용성을 유연하게 해줄 것으로 기대한다.

참고문헌

[1] Stoll, Cliff. 2005. The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage. Simon and Schuster.

[2] Spitzner, L. 2003. "Honeypots: Catching the Insider Threat." In 19th Annual Computer Security Applications Conference, 2003. Proceedings. 170 - 79. https://doi.org/10.1109/CSAC.2003.1254322.

[3] Pouget, F., Marc Dacier, and Hervé Debar. 2003. "White Paper: Honeypot, Honeynet, Honeytoken: Terminological Issues." Rapport Technique EURECOM 1275 (September).

[4] Cenys, Antanas, Darius Rainys, Lukas Radvilavicius, and Nikolaj Goranin. 2005. "Implementation of Honeytoken Module In DBMS Oracle 9ir2 Enterprise Edition for Internal Malicious Activity Detection." January.

[5] "HoneyGen: An Automated Honeytokens Generator." IEEE Conference Publication, IEEE Xplore. n.d. Accessed March 24, 2023. https://ieeexplore.ieee.org/abstract/document/5984063.

[6] Taofeek, Olayiwola Tokunbo, Moatsum Alawida, Abdulatif Alabdulatif, Abiodun Esther Omolara, and Oludare Isaac Abiodun. 2022. "A Cognitive Deception Model for Generating Fake Documents to Curb Data Exfiltration in Networks During Cyber-Attacks." IEEE Access 10: 41457 - 76. https://doi.org/10.1109/ACCESS.2022.3166628.

[7] Karuna, Prakruthi, Hemant Purohit, Sushil Jajodia, Rajesh Ganesan, and Ozlem Uzuner. 2021. "Fake Document Generation for Cyber Deception by Manipulating Text Comprehensibility." IEEE Systems Journal 15 (1): 835 - 45. https://doi.org/10.1109/JSYST.2020.2980177.

[8] Redwood, Owen, Joshua Lawrence, and Mike Burmester. 2015. "A Symbolic Honeynet Framework for SCADA System Threat Intelligence." In Critical Infrastructure Protection IX, edited by Mason Rice and Sujeet Shenoi, 103 - 18. IFIP Advances in Information and Communication Technology. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-26567-4_7.

[9] Banerjee, Mahesh, and Dr S D Samantaray. 2019. "Network Traffic Analysis Based IoT Botnet Detection Using Honeynet Data Applying Classification Techniques." 17 (8).

[10] Kumar, Sanjeev, B. Janet, and R. Eswari. 2019. "Multi Platform Honeypot for Generation of Cyber Threat Intelligence." In 2019 IEEE 9th International Conference on Advanced Computing (IACC), 25 - 29. https://doi.org/10.1109/IACC48062.2019.8971584.

[11] N. Krawetz, "Anti-honeypot technology," IEEE Security & Privacy, vol. 2, no. 1, pp. 76 - 79, Jan. 2004, doi: 10.1109/MSECP.2004.1264861.

[12] T. Holz and F. Raynal, "Detecting honeypots and other suspicious environments," in Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, Jun. 2005, pp. 29 - 36. doi: 10.1109/IAW.2005.1495930.

[13] X. Fu, W. Yu, D. Cheng, X. Tan, K. Streff, and S. Graham, "On Recognizing Virtual Honeypots and Countermeasures," in 2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, Sep. 2006, pp. 211 - 218. doi: 10.1109/DASC.2006.36.

[14] S. Mukkamala, K. Yendrapalli, R. Basnet, M. K. Shankarapani, and A. H. Sung, "Detection of Virtual Environments and Low Interaction Honeypots," in 2007 IEEE SMC Information Assurance and Security Workshop, Jun. 2007, pp. 92 - 98. doi: 10.1109/IAW.2007.381919.

[15] Sunmo. Yoo, Sungmo Wi, Jonghwa Han, Yonghyoun Kim, Jungsik Cho, "Anti-Deception Inference and Feature Extraction for Predicting Deception Awareness," in 2023 한국융합보안학회 하계학술대회.

[16] Inhwan Kim, Jiwon Kang, Hoonsang An and Byungkook Jeon, "A Study on Threat Detection Model using Cyber Strongholds," Journal of the Korea Convergence Society Vol. 22. No. 1, pp. 19-27, 2022. https://doi.org/10.33778/kcsa.2022.22.1.019.

[17] Jae-Hyun Choi, Hoo-Jin Lee, "A Study on the Real-time Cyber Attack Intrusion Detection Method",

Journal of the Korea Convergence Society Vol. 9. No. 7, pp. 55-62, 2018. https://doi.org/10.15207/JKCS.2018.9.7.055.

〔저자소개〕

조 정 식 (Jungsik Cho)
2007년 2월 중앙대학교 컴퓨터공학 석사
2011년 8월 중앙대학교 컴퓨터공학 박사
email : jungsik@kisa.or.kr

유 선 모 (Sunmo Yoo)
2012년 8월 한양대학교 수학과 학사
2019년 8월 호서대학교 정보보호학과 석사
email : seonmo87@gmail.com

위 성 모 (Sungmo Wi)
한국인터넷진흥원 침해대응팀
email : smwi0707@kisa.or.kr

한 종 화 (Jonghwa Han)
2022년 2월 국립목포대학교 정보보호학과 학사
email : jhhan@kisa.or.kr

김 용 현 (Yonghyoun Kim)
2007년 8월 동국대학교 컴퓨터멀티미디어학과 삭사
2011년 2월 성균관대학교 전자전기 컴퓨터공학과석사
email : yonghyoun@amgine.co.kr