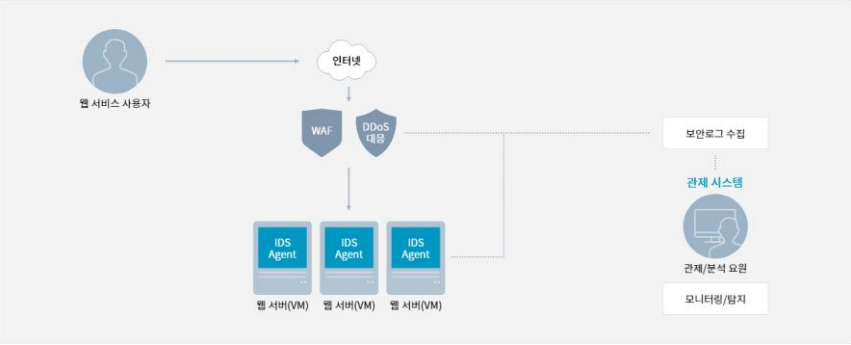
Collaborative Feature Maps of Networks and Hosts for AI-driven Intrusion Detection

230221
발표자 이연재

Introduction

침입 탐지 시스템(IDS) : 네트워크 또는 호스트에서 다양한 네트워크 위협 및 악의적인 동작으로부터 보호하는 중요한 보안 메커니즘 => 네트워크와 자산을 보호하는데 중요함



IDS (Security Cloud Platform Intrusion Detection System) | 클라우드 상품 | 삼성SDS (samsungsds.com)

- 홈페이지 및 애플리케이션 서버에 대한 침입을 실시간으로 탐지 및 대응하는 서비스
- 공격탐지를 위한 패턴 자동 업데이트 실시, 공격 유형, 타켓IP, 경보 목록을 대시보드 및 report 화면으로 제공하는 기능이 있다

NIDS : 네트워크의 특정 지점에서 여러 호스트를 대상으로 침입을 탐지하는 시스템. 스니핑(Sniffing) 을 사용하여 네트워크 패킷을 분석하고 침입을 탐지한다.

HIDS : 호스트 내부에 설치되며 내부 시스템의 여러가지 상태를 모니터링한다. 내부 시스템의 여러가지 상태를 모니터링하고 침입을 탐지하기 위해 syscall, log, 시스템 설정, 파일의 무결성을 분석한다.

CIDS(Combined Intrusion Detection System) : NIDS + HIDS

Introduction

< Contribution >

1. CIDS 분야에서 딥 러닝 모델을 사용하여 임의의 기능을 수동으로 마이닝하는 대신 호스트 기반 데이터에서 직접 기능을 추출하는 첫 번째 시도이다.

2. CIDS 데이터 세트를 만들기 위한 새로운 프레임워크가 제안된다.

3. SCVIC CIDS2021이라는 CIDS 데이터 세트는 잘 알려진 벤치마크 데이터 세트 CIC-IDS-2018 사용하여 생성된다.

4. 이 논문은 다양한 모양과 차원의 네트워크 흐름과 호스트 기반 기능을 포함할 수 있는 CIDS-Net을 제공한다.

5. C_Loss라는 새로운 손실 함수가 제안된다.

Related Work

네트워크 및 호스트 기반 데이터의 조합과 관련된 최근 연구 소개.

- KDD98/99[7] 및 NSL-KDD[8]: 네트워크 및 호스트 기반 기능을 모두 제공하지만 비현실적인 트래픽과 오래된 공격으로 비판을 받았다. 또한 메타 데이터에서 제한된 기능만 가져간다.
- CICFlowMeter : 네트워크 패킷에서 feature 을 추출하는 데 사용되지만 호스트 기반 데이터는 분석하지 않습니다.
- Vinayakumar et al.[12] : NIDS와 HIDS를 통합하기 위한 프레임워크를 제안하려고 시도하는 반면, 실험 파트는 NIDS와 HIDS 데이터 세트에서 DNN의 성능을 별도로 평가한다.

Related Work

네트워크 및 호스트 기반 데이터의 조합과 관련된 최근 연구

⇒ 네트워크 기반 데이터와 호스트 기반 데이터를 결합한 위의 연구는 네트워크 기반 기능을 수용하기 위해 호스트 기반 데이터를 표 형식으로 강제로 생성하여 많은 양의 귀중한 데이터 손실을 초래한 것으로 보인다.

⇒ 그럼에도 불구하고, 양 끝의 모델(NIDS 및 HIDS)은 크게 개선되었고, 데이터의 특성에 더 부합하기 위해 수많은 고급 방법/모델이 제안되었다.

예를 들어, Du et al. [14]는 LSTM(Long Short Term Memory) 네트워크를 사용하여 로그 항목을 시간 시퀀스로 표현하고 로그 파일에서 이상을 발견하는 DeepLog를 제공한다.

⇒ 이 연구는 다양한 형태와 모양의 데이터 샘플로 CIDS 데이터 세트를 생성하는 프레임워크와 침입 유형을 감지하기 위한 입력으로 사용할 수 있는 transformer-based CIDS-Net의 격차를 식별한다.

Method

CIDS Dataset Formation Framework

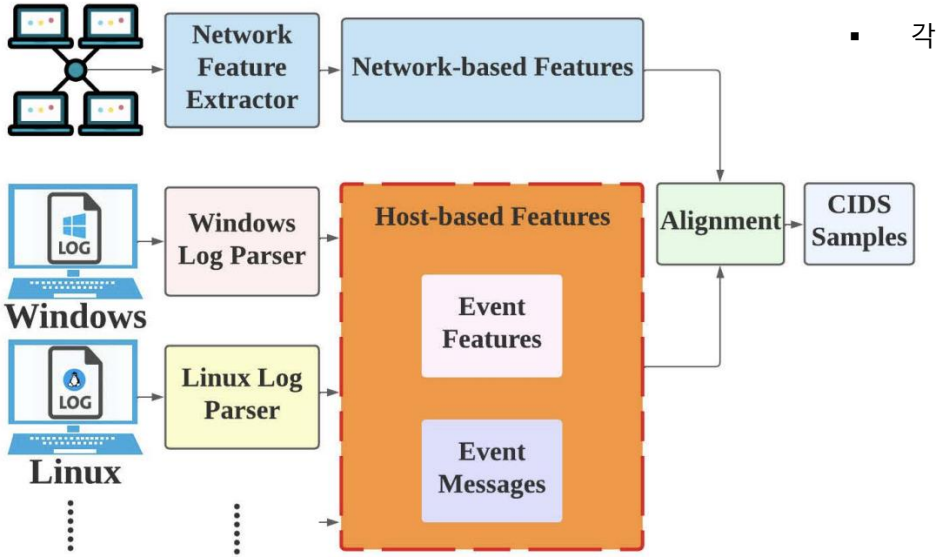


Fig. 1. CIDS Sample Generation Framework

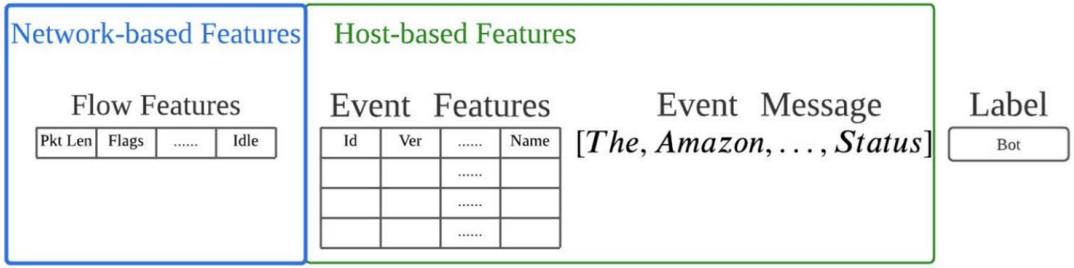
- 호스트 기반 기능과 네트워크 기반 기능은 별도의 타임스탬프와 기간 특성을 가지고 있음 → 정렬 알고리즘 제시
- 각 네트워크 기반 샘플은 여러 호스트 기반 샘플에 해당.
  - 네트워크 기반 인스턴스와 호스트 기반 인스턴스가 동일한 시간 프레임에 생성되도록 정렬

Input: $\boldsymbol{S}_n$; $\boldsymbol{S}_h$
Output: CIDS Samples

$sort(samples_h)$;
$\boldsymbol{S}_c \leftarrow \emptyset$;
foreach $s_n \in \boldsymbol{S}_n$ **do**
  $ip, d \leftarrow get\_ip\_timewindow(s_n)$;
  $\boldsymbol{E}_f, \boldsymbol{E}_m \leftarrow \emptyset, \emptyset$;
  foreach $s_h \in \boldsymbol{S}_h$ **do**
    if $ip \in s_h$ *and* $s_h.ts \in d$ **then**
      $e_f, e_m \leftarrow s_h$;
      $\boldsymbol{E}_f \leftarrow \begin{bmatrix} \boldsymbol{E}_f \\ e_f \end{bmatrix}$;
      $\boldsymbol{E}_m \leftarrow \boldsymbol{E}_m \frown e_m$;
    end
  end
  $\boldsymbol{S}_c \leftarrow \boldsymbol{S}_c \cup (s_n, \boldsymbol{E}_f, \boldsymbol{E}_m)$;
end

Method

CIDS sample



이벤트 피쳐 인스턴스는 벡터이며, 이벤트 피쳐의 출력은 행렬이고, event messages 는 직접 연결되어 더 긴 문자열을 만든다.

- **network feature** : 네트워크 흐름에 대한 통계 관측치 벡터(패킷 길이, TCP flags, 패킷 출발-도착 시간)

  ⇒ network-based features, events features, event messages, label 4가지 요소를 구성한다.

- **event feature** : R(n*m) 행렬 ⇒ n : window time 내에 발생하는 event 수, m : event feature 의 수

- **event messages** : 문자열 ⇒ NLP(Natural Language Processing) 기술이 활용됨.

CIDS-Net

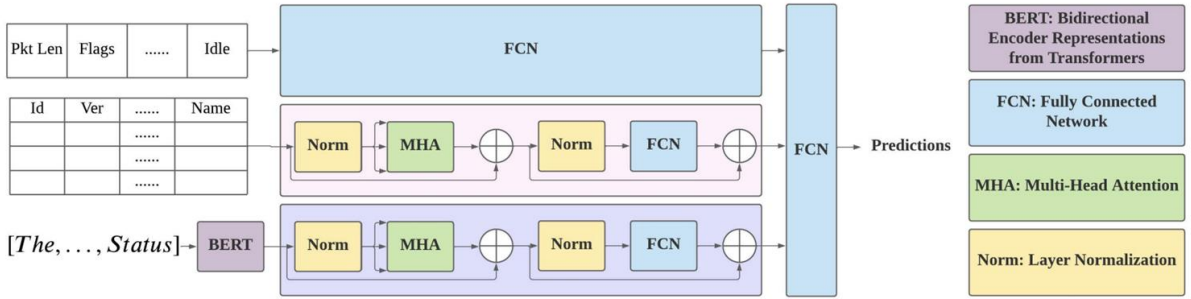


Fig. 3. The architecture of CIDS-Net

- **Network Feature Encoder** : network feature 는 표 형식의 데이터라 FCN 을 인코더로 사용(TabNet)
- **Event Feature Encoder** : 트랜스포머 인코더
- **Event Message Encoder** : BERT word 임베딩을 통해 벡터로 변환 → R(i*j) 행렬로 변환 (i : 문장의 토큰 수, j : 임베딩 벡터의 차원) , 트랜스포머 인코더
  - 큰 네트워크를 CIDS-Net에 연결하여 잠재적인 과적합을 방지하기 위해 미세 조정 없이 가중치를 고정하는 사전 훈련된 BERT 네트워크를 사용
- **aggregator(FCN)** : 인코더의 출력을 입력으로 받아 intrusion(침입) 을 예측하는 layer
  - 예측을 얻기 위해 인코딩된 hidden state 는 FCN layer 에 의해 aggregate 된다.

C_Loss

$$C\_Loss = \alpha * CE(y, \hat{y}_n) + (1 - \alpha) * CE(y, \hat{y})$$

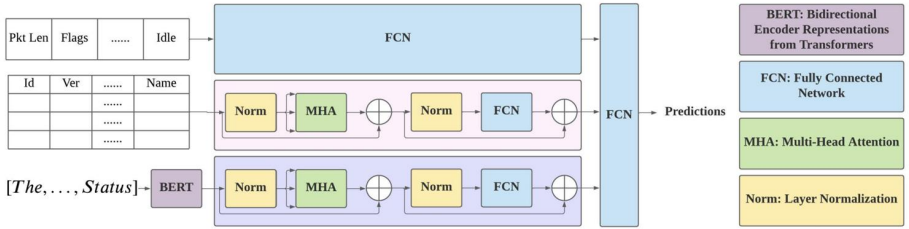


Fig. 3. The architecture of CIDS-Net

Combined Loss(C_loss)을 사용하여 네트워크 feature 인코더와 전체 CIDS-Net을 공동으로 훈련하여 둘 다 예측할 수 있도록 할 것을 제안한다. → 호스트 기반 데이터가 없을 때 유연성을 보장하고 더 복잡한 모델을 통합한다.

- CE : cross entropy loss
- y : sample 의 실제 레이블, y^ : aggregator 의 예측, y^n : network feature 인코더의 예측
- 알파 : 하이퍼파라미터

Cross Entropy Loss

$$C\_Loss = \alpha * CE(y, \hat{y}_n) + (1 - \alpha) * CE(y, \hat{y})$$

$$CE = -\sum_{c=1}^{C} L_c log P_c \qquad CE = -\frac{1}{n} \sum_{i=1}^{n} \sum_{c=1}^{C} L_c log P_c$$

C : 클래스의 총 개수
P : Softmax Function 에서 다룬 각각의 클래스에 속할 확률
L : 실제값의 One-hot Vector => 원-핫 인코딩을 통해 데이터가 벡터 형태로 변화된 형태

Python Pytorch 강좌 : 제 13강 - 다중 분류(Multiclass Classification) - YUN DAE HEE (076923.github.io)

Experiments

SCVIC-CIDS-2021 Dataset

CIC-IDS-2018 을 활용하여 2021 버전이 만들어졌다. network, host-based 와 현실적인 network setting 과 최근 공격 데이터가 필요하다.

- 2018 : network based feature 만 추출 (host 관련 데이터는 없음) , CDFF 에 입력되어져서 2021 을 만드는데 활용됨.
  - OS 로그는 Windows와 Linux 운영 체제에서 가져옵니다.
  - 'Get-WinEvent' cmdlet은 윈도우즈 OS 로그를 tabluar 형식으로 변환하고, 'ProviderName' 및 이벤트 메시지는 Linux OS 로그에서 추출됩니다.

- 2021 : PCAP format 의 network traffic , host-based data, labeling metadata 로 구성됨.
  - network feature 는 2018 만큼
  - system event entity 수가 28 ⇒ 모든 event feature 는 R(28 * 8) 크기로 zero padding 됨.

Experiments

SCVIC-CIDS-2021 Dataset

TABLE I
SCVIC-CIDS-2021 CLASS DISTRIBUTION

	Training Set	Test Set	Total
Benign	308375	152172	460547
Bot	60767	29693	90460
DDOS-HOIC	137147	67449	204596
DDOS-LOIC-HTTP	39019	19166	58185
DDOS-LOIC-UDP	760	342	1102
DoS-GoldenEye	2271	1163	3434
DoS-Hulk	13388	6553	19941
DoS-SlowHTTPTest	10579	5351	15930
DoS-Slowloris	1394	702	2096
FTP-BruteForce	32222	15918	48140
SSH-Bruteforce	11143	5418	16561
Sum	617065	303927	920992

Training, Test set sample 의 수

CIC-IDS-2018 에서 제공된 네트워크 흐름 기반 표 형식 데이터 세트에는 **IP 주소와 타임스탬프가 없기 때문에** 동일한 설정의 CICFlowMeter를 사용하여 재생성되고 IDS 2018에 나열된 attack traces 에 따라 레이블링된다.

Attacker	Victim	Attack Name	Date	Attack Start Time	Attack Finish Time
172.31.70.4 (Valid IP:18.221.219.4)	172.31.69.25 (Valid IP:18.217.21.148)	FTP-BruteForce	Wed-14-02-2018	10:32	12:09
172.31.70.6 (Valid IP:13.58.98.64)	18.217.21.148-172.31.69.25	SSH-Bruteforce	Wed-14-02-2018	14:01	15:31
172.31.70.46 (Valid IP:18.219.211.138)	18.217.21.148-172.31.69.25	DoS-GoldenEye	Thurs-15-02-2018	9:26	10:09

IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB

Experiments

CIDS-Net Results (Network-based feature 만 사용될 때)

	LR	AB	NB	TabNet	GB	DT	XGB	RF
Benign	0.780	0.891	0.698	0.999	0.999	0.999	0.999	1.000
Bot	0.009	0.000	1.000	1.000	1.000	1.000	1.000	1.000
DDOS-HOIC	0.547	0.788	0.998	1.000	1.000	1.000	1.000	1.000
DDOS-LOIC-HTTP	0.000	0.000	0.998	1.000	1.000	1.000	1.000	1.000
DDOS-LOIC-UDP	0.000	0.000	0.423	0.715	0.746	0.759	0.761	0.761
DoS-GoldenEye	0.000	0.000	0.828	0.995	0.996	0.999	1.000	1.000
DoS-Hulk	0.763	0.372	1.000	0.999	1.000	1.000	1.000	1.000
DoS-SlowHTTPTest	0.000	0.000	0.500	0.440	0.544	0.543	0.543	0.542
DoS-Slowloris	0.000	0.000	0.020	0.907	0.905	0.993	0.994	0.996
FTP-BruteForce	0.498	0.853	0.498	0.866	0.875	0.875	0.875	0.875
SSH-Bruteforce	0.000	0.000	0.999	1.000	1.000	1.000	1.000	1.000
macro avg	0.236	0.264	0.724	0.902	0.915	0.924	**0.925**	**0.925**
weighted avg	0.556	0.674	0.810	0.982	0.984	0.985	**0.985**	**0.985**

XGBoost의 출력은 CID 결과를 비교하는 Baseline

LR : logistic regression, AB : AdaBoost , NB:Naive Bayes, GB :gradient boost, DT : decision tree, RF : random forest

Experiments

CIDS-Net Results

TABLE III
CIDS-NET RESULTS UNDER SCVIC-CIDS-2021 USING VARIOUS NETWORK ENCODERS

	Baseline(XGB)	CIDS-Net using Different Network I			
		Identity	FCN	FCN (CLoss)	TabNet
Benign	0.9995	0.9995	0.9993	0.9963	0.9998
Bot	1.0000	0.9988	0.9978	0.9901	0.9994
DDOS-HOIC	1.0000	1.0000	0.9996	0.9998	1.0000
DDOS-LOIC-HTTP	0.9997	0.9997	0.9982	0.9989	0.9999
DDOS-LOIC-UDP	0.7609	0.9771	0.9884	0.9899	0.9927
DoS-GoldenEye	1.0000	0.9923	0.9914	0.9940	0.9996
DoS-Hulk	1.0000	0.9992	0.9998	0.9665	0.9988
DoS-SlowHTTPTest	0.5425	0.9228	0.9964	0.9233	0.9990
DoS-Slowloris	0.9936	0.9894	0.9929	0.9716	0.9993
FTP-BruteForce	0.8749	0.9721	0.9991	0.9713	0.9996
SSH-Bruteforce	0.9999	0.9998	0.9984	0.9994	0.9998
Macro Avg	0.9246	0.9864	0.9965	0.9819	0.9989
Weighted Avg	0.9848	0.9967	0.9990	0.9934	0.9998

CLoss를 사용하면 CIDS 성능이 약간 저하되지만 TabNet 을 네트워크 인코더로 사용하면 매크로 평균 F1 점수가 99.89%로 FCN을 네트워크 인코더로 사용하는 경우보다 약간(즉, 0.2%) 더 높다.

Conclusion

- 본 논문은 네트워크 및 호스트 기반 데이터를 통합하고 정렬하여 IDS 성능을 향상시킬 수 있는 새로운 결합 IDS(CIDS)를 제안
- CIDSNet에 대한 네트워크 인코더(즉, Identity, FCN 및 TabNet)와 손실 함수(즉, 교차 엔트로피 손실 및 CLoss)의 몇 가지 조합을 조사했다.
- FCN을 애그리게이터로 사용할 때, CIDS-Net은 네트워크 인코더로 TabNet을 사용할 때 최적의 성능을 발휘하여 99.89%의 매크로 F1 점수를 달성하는 것으로 나타났다.
- CIDS 결과는 호스트 기반 기능이 IDS 성능을 크게 향상시킬 수 있음을 보여주었다.
- 향후 연구는 머신 러닝 모델을 CIDS-Net에 통합하여 불균형 클래스의 성능을 향상시킬 수 있다.