

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)
팀 명	멜리리를 찾아서
문서 제목	중간 보고서

Version	1.1
Date	2021-04-01

지도교수	윤 명근 교수
팀원	장 우혁(조장)
	김 민송

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "**O24Sec(Object-Oriented Clustering for Security Monitoring)**"를 수행하는 팀 "**멜러리를 찾아서**"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "**멜러리를 찾아서**"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	중간 보고서
원안작성자	김민송, 장우혁
수정작업자	김민송, 장우혁

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-03-05	전원	1.0	최초 작성	개요 초안 작성, 개발 목표 초안 작성 개발 배경 작성 산학협력 관련 작성
2020-04-03	전원	1.1	내용 추가	개발 사항 추가 진행 사항 추가 기대 효과 추가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

목 차

1	산학지정 과제	4
1.1	산학협력 회사(원스).....	4
1.2	산학 요구 사항.....	4
1.3	주제에 대한 사회적 연구.....	5
2	개발 목표 및 내용	6
2.1	제안 기술 배경.....	6
2.2	제안 기술 목표.....	6
2.3	제안 기술 개발 내용.....	7
2.3.1	보호대상 객체식별.....	7
2.3.1	암호-비암호 구별.....	10
	암호-비암호 구별[별첨].....	15
2.3.1	객체중심 오탐제거(진행 중).....	17
2.4	기대효과 및 활용방안.....	19
3	프로젝트 팀 구성 및 역할 분담	21
4	개발 일정 및 의사소통	22
4.1	개발 일정.....	22
4.2	프로젝트 간 의사소통.....	22
5	참고 문헌	23

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

1 산학지정 과제

기업이 필요로하는 차세대 보안관제 기술 중 일부를 졸업프로젝트 작품으로 개발.

1.1 산학협력 회사(원스)

원스는 2003년 코스닥 상장된 정보보호 전문 기업으로 1998년부터 네트워크 트래픽 분석 기술을 기반으로 2000년 한국 정보보호산업을 대표하는 브랜드인 '스나이퍼(SNIPER)'를 출시해 안정적인 사업궤도에 올라서있다.

원스는 네트워크 보안 분야에서 침입방지시스템(IPS), Ddos 공격대응솔루션, 지능형공격(APT) 대응 솔루션, 통합위협관리솔루션, 방화벽에 이르기까지 시장 이슈에 따른 핵심 솔루션에서 각각 우위를 기록하며 보안기술과 시장을 선도하고 있다. 2003년부터 2004년까지는 당시 네트워크 보안의 대표 솔루션인 침입 탐지 시스템(IDS)으로 보안시장에 성공적으로 안착했고 2005년과 2010년 개발, 공급된 침입방지시스템(IPS)과 DDoS차단 시스템까지 잇따라 국내 시장점유율 1위를 차지하며 현재까지 네트워크 정보보호 업계 선두를 달리고 있다.

1.2 산학 요구 사항

원스에서 제시한 과제 내용으로는 현재 원스에서 사용되는 보안 관제 제품인 SNIPER BD1에 대한 오탐 제거 기술 개발이다.

SNIPER BD1은 5G SIEM으로써 보안솔루션으로부터 다양한 정보를 수집하기 위한 기반기술과 대용량 배치처리 및 통계처리 분석, 위협 예측 등 고도화된 보안관제를 수행하기 위한 분석기술, 데이터 처리 분석에 따른 시각화 표현 기술을 적용한 응용/표현 기술로 구성되어 있다. 여기에서 현재 분석이나 예측 등, 보안 관제에서 인공지능을 이용해 서비스를 자동화하여 공격 의심 패킷에 대해 이벤트 알람을 발생시키고 있다.

그러나, 이러한 보안 관제에서 "오탐(False Positive)"가 현저하게 많아 이벤트를 분석하는 보안관제사들이 정확한 분석을 하는데 방해를 주고, 오탐을 분석하는데 걸리는 시간이 대부분 소요되어 정작 중요한 공격을 놓치고 있는 상황이다.

따라서 원스는 보안관제 제품의 정확도를 높여 오탐을 줄일 필요가 있었고, 이러한 문제를 해결하기 위해 이번 캡스톤 디자인에서 "멜러리를 위하여"과 원스는 협력하여 보안관제 제품의 정확도를 높이고자 프로젝트를 진행하였다.

1.3 주제에 대한 사회적 연구

1) 보안관제 정탐률 향상을 위한 인공지능 알고리즘 연구[1]

전기학회 논문지에서 발표된 보안관제 정탐률 향상을 위한 인공지능 알고리즘 연구에서는 보안 관제에 사용되는 인공지능에 대한 정탐률 향상을 위해 여러 알고리즘과 특징 선정을 통해 인공지능의 정탐률 향상을 보여줬지만 학습되는 데이터가 바뀔 경우 확연히 달라지는 것을 보여줌으로써 실제 환경에 인공지능에서는 각각의 환경에 적합한 데이터셋을 직접 선정 및 정제하여 사용하여 한다는 결론을 보여줬다.

<표 1 알고리즘 연구 전 정확도>

공격명	이벤트수	오탐수	오탐률
SQL 인젝션	152	131	86.18
Cross Site Scripting	375	0	0
File Upload	58	36	62.07
File Download	504	475	94.25
비인가 접근	1362	1300	95.45
계	2,451	1,942	79.23

<표 2 알고리즘 연구 후 정확도>

공격명	Accuracy	Recall	Precision
SQL Injection	90%	75%	89%
XSS	97%	100%	100%
FileUpload	91%	85%	78%
FileDownload	97%	84%	91%
비인가 접근	98%	89%	92%

<표 3 알고리즘 연구 후, 비 정제 과거 데이터 사용시 정확도>

공격명	Accuracy	Recall	Precision
SQL Injection	54%	59%	89%
XSS	77%	67%	72%
FileUpload	61%	75%	68%
FileDownload	67%	64%	61%
비인가 접근	79%	79%	72%

[그림 1] 알고리즘 연구를 통한 정확도 변화

2) 보안관제를 위한 AI 모델 기술 소개[2]

NetSec2020 세미나에서 과학기술 사이버 안전센터의 송중석 박사는 AI 모델의 정확도를 높일 때 모델자체의 기술을 바꾸지 않더라도 입력되는 데이터가 일관적이고 높은 정확도를 가지고 있다면 모델의 성능이 비약적으로 올라간다는 것을 데이터 전수 재 라벨링 작업을 통해 증명하였다.

모델	정확도	정밀도	재현율	F1-score
정합 전	98.02%	93.76%	59.6%	72.8%
정합 후	98.23%	96%	95.42%	94.3%

[그림 2] 데이터 정합 후 인공지능 모델 성능 향상(F1-score_[아래])

[F1-Score]

F1-Score 는 인공지능의 성능 평가로 쓰이는 오차 행렬이다.

정확도는 정확히 예측한 수를 전체 샘플 수로 나눈 값으로 인공지능이 잘 맞추는 가를 보여준다.

정밀도는 True 라고 예측된 것 중 진짜 True 의 비율로 False 인데 True 라고 말한 것이 적어야 높다.

재현율은 True 라 예측했는데 True 인 것과 False 라 예측했는데 False 인 즉, 진짜 맞춘 것들 중 True 의 비율을 나타낸다, 이는 양성 샘플이 얼마나 식별되었는지 보여주고 적중률이라고도 한다.

F1-score 는 정밀도와 재현율을 하나로 요약한 값으로 높을수록 인공지능의 성능이 좋다고 평가할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

2 개발 목표 및 내용

2.1 제안 기술 배경

NetSec2020 세미나에서 과학기술 사이버 안전센터의 송중석 박사는 AI 모델의 정확도를 높일 때 모델 자체의 기술을 바꾸지 않고 학습에 이용되는 데이터가 일관적이고 높은 정확도를 가지고 있다면 모델의 성능이 비약적으로 올라간다는 것을 발표했다.

보안관제 정탐률 향상을 위한 인공지능 알고리즘 연구에서는 모델의 정탐률에 대해 알고리즘의 차이도 중요하지만 모델이 동작하는 네트워크의 데이터 특성에 맞게 학습데이터를 선정하고 정확도 있는 데이터를 위해 정제하여 사용해야 한다고 발표했다.

이러한 연구에서 멜러리를 찾아서 팀은 모델 자체의 알고리즘 보다 학습되는 데이터의 신뢰도에 초점을 두기로 하였고 데이터 정제를 통해 머신 러닝의 학습을 더욱 효과적으로 하여 높은 정탐률을 얻어내기로 하였다.

멜러리를 찾아서 팀은 데이터 정제 방법을 4가지 네트워크 흐름에 대해(내부 서버 -> 외부 클라이언트, 내부 클라이언트 -> 외부 서버, 외부 서버 -> 내부 클라이언트, 외부 클라이언트 -> 내부 서버) 나누어서 볼 경우 각 객체에 공통적으로 보이는 특성이 있을 것이라 확신하여 기존의 학습용 데이터를 4가지 객체별로 나누고 각 객체별로 클러스터링을 하면 분석에 도움을 줄 수 있는 유의미한 정보가 나올 것이라 생각하였다.

2.2 제안 기술 목표

본 프로젝트는 기존에 보안장비에 들어가 있는 인공지능 모델에 학습 데이터로 사용되어지는 IPS 데이터셋에 대해 보호 자산 객체를 식별하여 각 객체에 따라 보안 취약점을 중심으로 분류 후 내용기반의 분석을 가능하도록 도와줄 수 있는 기술 개발을 목표로 한다.

■ 세부 목표

1. 보호대상 객체 식별
 - 서버/클라이언트 - 내부/외부로 분리하여 이벤트를 4 가지 객체로 분류
2. 암호-비 암호 구별
 - 3 번 목표에 사용할 수 있는 식별 가능한 정보를 가지고 있는 이벤트 구별
3. 객체 중심 오탐 제거
 - 1,2 번 목표에서 분류된 이벤트를 가지고 클러스터링 하여 각 클러스터링 별 특징을 확인 해 데이터 분석 조언

2.3 제안 기술 개발 내용

2.3.1 보호대상 객체식별

보호 자산 객체, 보안 취약점을 중심으로 분류하여 내용기반 분석을 진행하기 위해, 우리는 총 4가지의 객체로 이벤트를 분류했다.

객체	출발지	도착지	공격 시나리오
Case1	내부 클라이언트	외부 서버	<ul style="list-style-type: none"> - 내부 직원 PC가 감염되어 외부 C&C서버 접속 - 내부 정보 유출, DBD공격 등
Case2	외부 서버	내부 클라이언트	
Case3	외부 클라이언트	내부 서버	<ul style="list-style-type: none"> - 외부 해커가 내부 서버 공격 - sql injection, XSS, web shell, etc.
Case4	내부 서버	외부 클라이언트	

[그림3] 객체 구분과 객체별 특징

우리는 이벤트의 5-Tuple값만을 통해 분류하는 기술을 직접 구현했다. 그 이유는 다음과 같다.

1. 보안관제대상이 많아질 경우 각 대상 별 객체 분류 자동화
2. 내부 개발자가 새로운 서버를 도입하는 등의 업데이트가 반영되지 않음
3. 인공지능을 위해 주어지는 공유 데이터 등은 대체적으로 고객의 보안을 위해 비식별화 됨

따라서 5-Tuple만을 통해 객체를 식별 및 분류할 수 있는 기술을 개발한다면, IPS기기, 환경, 사용자 등에 영향을 받지 않고 객체 분류를 할 수 있어 범용성 높은 기술로 활용할 수 있다.

사용한 데이터로는 약 2000만건(1년)의 IPS Event 데이터를 사용했으며, 공통점 및 규칙을 찾기 위해 빅데이터 전처리 및 통계화 작업을 거쳐, 그 결과를 분석하고 실험했다.

또한 해당 과정에서의 예외케이스 분석 및 시행착오를 거쳤는데, 보안 데이터이므로 시행착오와 예외케이스 분석에 대한 자세한 기술은 담지 못했다.

추가적으로 데이터 셋의 지역별, 기간별 실험을 진행하여 강건성을 테스트했다.

이를 통해 단계별로 적용하는 기술을 개발했으며 그 기술들의 과정은 아래와 같다.

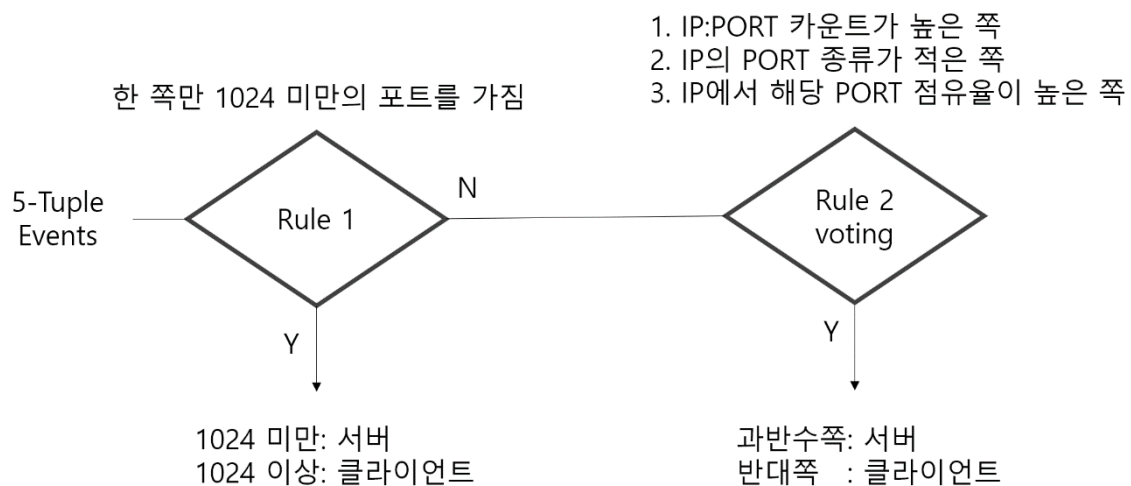
[1. 데이터 통계 추출]

두 분류 단계에 앞서, 5-Tuple값을 통계를 내기 위해, 데이터 통계 작업을 진행했다.
 서버-클라이언트 분류에서 사용할 각 IP별 Port 종류, Port 종류별 빈도를 추출하였고,
 내부-외부 분류에서 사용할 각 IP *B-Class대역의 통신한 상대 *B-Class의 종류를 추출하였다.

*B-Class: 32bit인 IP를 앞 16bit가 같은 IP들로 묶은 대역(A : 8bit, B : 16bit, C : 24bit)

이 외에도 필요한 정보를 추가하여 별도 파일로 추출해 통계를 살펴볼 수 있는 독립적 함수로 사용할 수도 있다.

[2. 서버 클라이언트 분류]



[그림4] 서버-클라이언트 분류 순서도

Rule1은 단일 이벤트만을 보고 분류를 하는 기준이다. 이벤트에서 한 쪽이 1024미만의 Well-Known Port를 사용하고, 반대쪽은 사용하지 않는다면, 1024미만 포트 쪽이 서버라고 결정하는 방법이다. 약 85%의 이벤트를 Rule1을 통해 분류할 수 있다.

Rule2는 앞서 추출했던 데이터 통계자료를 사용하는 3가지 규칙의 과반수로 결정한다.

1. 해당 IP의 해당 Port의 빈도가 더 높은 쪽을 서버라고 투표
2. 해당 IP의 Port종류가 적은 쪽을 서버라고 투표
3. 해당 Port가 해당 IP의 전체 빈도에서 차지하는 비율이 높은 쪽을 서버라고 투표

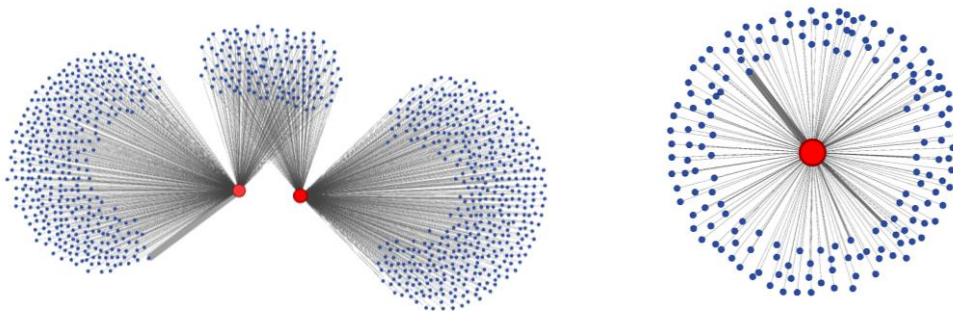
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

[3. 내부 외부 분류]

다음으로, 이벤트에서 보호대상을 식별하기 위해, 이벤트를 내부, 외부로 분류해야 한다.

내부를 식별하기 위해, 여러 접근방식을 시도했는데, 가장 효과적인 것은 IP를 대역으로 묶어서 통계를 살펴보는 것이었다.

C클래스 IP 대역의 종류, B클래스 IP 대역 분석등을 토대로 내부를 분류해낼 수 있는 기준을 찾아냈다.

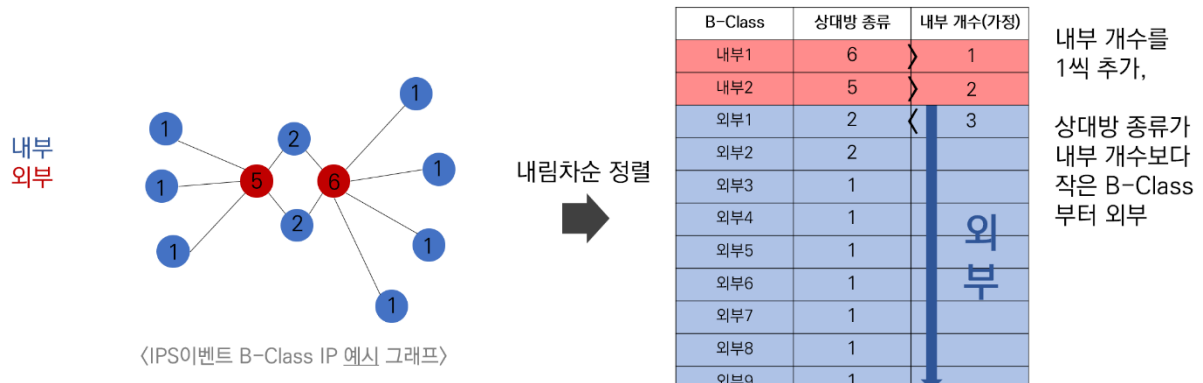


[그림5] 임의의 데이터 셋 B-class 대역 통신 그래프

B클래스 IP 대역으로 이벤트를 묶은 후, 통신한 상대방 B클래스의 종류가 많은 순으로 내림차순 정렬한다.

위 그래프를 참고하면, 1위부터 상위 B클래스들은 내부가 될 수밖에 없다. 붉은색 노드가 내부, 푸른색 노드가 외부인데, 당연하게도 내부 B클래스 대역이 가지는 엣지(통신)의 종류가 푸른색 노드보다 압도적으로 많음을 확인할 수 있다.

다음 문제는, 정렬된 상대 B클래스 IP대역 종류에서, 1위부터 연속되는 내부가 끝나는 지점 (임계점)을 찾아내는 것이었다.



[그림6] 통신한 상대 B-class IP 종류(Cardinality) 정렬 후 내부 임계점 찾기

임의의 예시로 내부가 2개 존재하는 B-Class 네트워크가 있다고 가정했을 때, 예시 그래프로 표현하여 확인하면, 외부는 상대방 종류를 내부 노드 개수 이하로 가짐을 알 수 있다.

이 사실에 기인하여 임계점을 반환하는 알고리즘을 작성하였다.

1. B-Class 대역 데이터를 통신한 상대방 B-Class종류를 내림차순으로 정렬한다.
2. 첫 번째 노드부터 탐색하며 내부의 개수를 한 개씩 늘린다.
3. 만약 내부의 개수 이하(현재 비교 값 미만)인 노드가 탐색이 된다면, 그 노드부터 모두 외부라고 판단한다.

이렇게 판단된 내부, 외부 B-Class 에 속하는 이벤트들을 내부, 외부로 분류하게 된다.



2.3.2 암호-비암호 구별

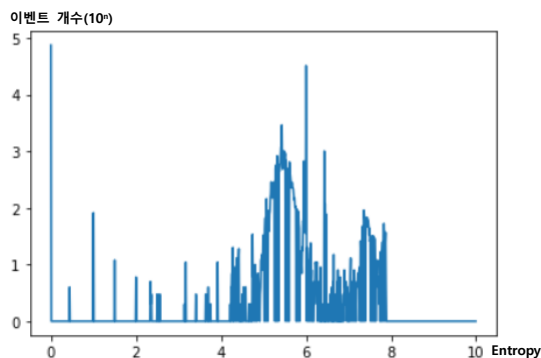
앞서 개발된 보호대상 객체 식별 기술을 통해 분리된 각각의 이벤트에서 이후 적용할 기술인 객체 내용 중심의 클러스터링 과정에서 유의미한 정보를 가지고 있는 암호화되지 않은 페이로드를 가진 이벤트와 페이로드를 통해 정보를 얻을 수 없는 암호화된 페이로드를 분리시켜 저장한다.

[여러 시행착오 및 구현 과정]

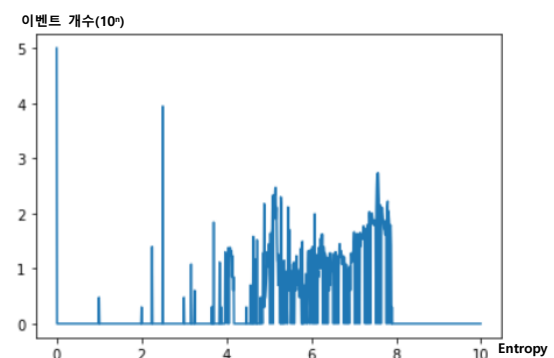
1. 페이로드를 디코딩하여 인간이 식별 가능한 영단어가 2 개 이상 있으면 암호화되지 않은 것으로 구분한다.

[한계] 데이터 개수가 너무 많아 소요되는 시간이 상당함 (영어 단어장, 스트링 매칭 시간) 영어 단어장에 흔히 단어로 취급하지 않는 매우 작은 단어들도 포함되어 있어, 암호문도 종종 평문으로 취급이 됨.

2. [배경 - 윤명근 교수님]엔트로피^[별첨 1] 를 통해 기준 점을 나누고 기준 점이 넘어가는 경우 암호화된 페이로드로 구분한다.



[그림 7] 443번포트에서 페이로드 엔트로피 분포



[그림 8] 80번포트에서 페이로드 엔트로피 분포

[한계 1] 모든 데이터에 통용할 만한 기준점이 없음, 일례로 길이가 짧은 암호화된 페이로드의 경우 Entropy 가 4 가 나오지만 완전히 암호화된 상태이고 길이가 긴 암호화가 안되어있는 페이로드의 경우 4 가 나오면 완전히 평문인 상태임

[한계 2] 그래프를 봤을 때 암호통신의 대표 프로토콜인 TLS(443 번포트)에서도 엔트로피 값이 연속적으로 넓게 분포되어 있어 임계치를 찾기 어렵고 평문 통신의 대표 프로토콜인 HTTP(80 번 포트)에서도 엔트로피 값이 연속적으로 분포되어 있고 높은 값이 있어 임계치를 정하기 어려움

[한계 3] 평문 중에 이미지나 동영상을 전송할 경우 해당 콘텐츠가 바이트로 보내지다 보니 바이트의 분포도가 높아져 엔트로피 값이 높게 측정돼 암호문으로 인식됨



3. 현재 엔트로피 측정은 8bit(1Byte) 기준으로 최대 256 가지의 경우를 측정하고 있으니 이를 반절로 줄여서 4bit 엔트로피로 측정을 한다면 길이가 짧은 페이로드에 대해서도 유의미한 엔트로피 값을 얻어 낼 수 있다.

[배경 - 윤명근 교수님] 짧은 길이 페이로드가 문제이니 짧은 길이 페이로드 부분을 커버할 수 있는 기술을 만들 필요가 있었다. 엔트로피 측정이 8비트로 진행되니 이를 4비트로 측정하게 된다면 16가지의 비트 종류에 대해서 엔트로피를 측정하게 되니 짧은 길이도 커버할 수 있을 것이다.

[한계] 엔트로피의 기본 원리가 네트워크 통신에서 DATA를 바이트화 시켰을 때 주로 사용되어지는 16진수 바이트가 한정되어 있기 때문에 효율성이 있었다, 하지만 짧은 길이에 대해서 판단을 하지 못해서 4bit 엔트로피로 측정을 해보았는데 이 경우에는 평문인 경우에도 엔트로피가 높게 나오는 경우가 많이 발견이 되었다. 왜 그런지 확인을 해보니 byte stream을 4bit 단위로 잘라 놓았을 때 알파벳이나 특수문자의 부분적 특성이 반영되지 않고 특히 알파벳의 경우 각자 byte로 바꿨을 때 1bit씩 차이가 나기 때문에 4bit로 엔트로피를 측정하게 되었을 경우 좋은 결과를 얻지 못했다.

4. 페이로드를 부분별로 나눠서 각 부분에 대해 엔트로피를 구해서 가장 낮은 것/각 부분의 엔트로피의 평균값/맨 처음 부분의 엔트로피를 대표 값으로 하여 대표 값이 특정 임계치 값을 넘어가면 암호화된 페이로드로 구분한다.

[배경 1] 일반적인 동영상이나 이미지 전송, 혹은 파일 업로드시 서버와 MIME TYPE 으로 통신을 하게 된다. 여기에서 MIME 선언 뒤부분인 콘텐츠 부분은 이미지를 BYTE 로 표현하기 때문에 높은 엔트로피가 나오지만 MIME 헤더부분은 일정한 규칙으로 전송을 하기 때문에 엔트로피가 상대적으로 낮다. 따라서 이 헤더부분의 엔트로피를 측정해 이 이벤트의 대표 값으로 사용한다면 이미지, 동영상 전송 이벤트도 일반적인 평문 이벤트로 분류할 수 있을 것 같다.

[배경 2 - 문성익 멘토님] 서버 통신 중 난수처럼 전송되는 것에는 이미지, 동영상, 압축 파일 등이 있는데 각 경우에 프로토콜을 통해 헤더를 첨부하여 보내게 되니 현재 실험방법이 괜찮을 것 같다. 보통의 헤더는 앞쪽에 있으니 앞에 부분의 길이를 어느정도 할당시켜 그 부분의 엔트로피를 측정해서 그 부분들끼리 비교해보는 것도 의미가 있을 것 같다.

[한계 1] 페이로드의 길이가 짧은 이벤트에 대해서는 절대크기, 상대크기 모두 애매해져서 정확한 비교를 하기 어려움.

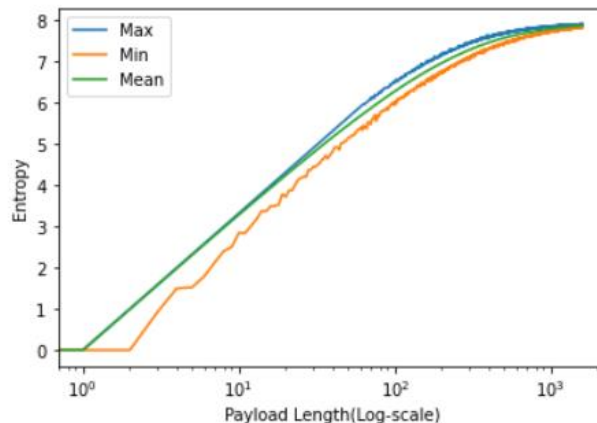


[한계 2] 2번 실험에서의 한계와 비슷하게 짧은 길이의 페이로드의 경우 암호화가 되었다
라도 긴 길이의 평균 엔트로피 값과 비슷해 특정 엔트로피 임계 값으로는 평균과 암호문
을 구분할 수 없음

[한계 3] 일부 암호화된 이벤트의 헤더 부분의 엔트로피도 낮게 측정될 때가 있음(TLS
HAND SHAKE)

5. 페이로드 길이별로 암호화된 페이로드를 직접 만들어봐서 각 길이에 대한 암호화된 엔트
로피 값의 평균을 정하고 길이별로 해당 평균값 이상으로 넘어가면 암호화된 것으로 처
리한다.

[배경] 길이 별로 엔트로피 값의 최대가 다르다면 각 길이마다의 엔트로피 값의 분포를
미리 구해두고 엔트로피의 길이를 기준으로 임계치를 비교해보면 좋을 것 같다 생각하여
실험을 진행했다. RANDOM 으로 BYTE 를 생성하는 라이브러리를 통해 길이 1 부터 1600
까지에 대해 각 길이당 100000 번씩 페이로드를 추출했고 이렇게 랜덤화 된 페이로드를
암호화된 페이로드라 가정하였을 때 그 페이로드로 길이마다의 평균값, 최소값, 최대값을
추출해 낼 수 있었다.



[그림 9] 암호화된 페이로드의 각 길이 별 엔트로피 대표값 분포

[한계] 최대값은 가장 큰 것이기에 기준이 될 수 없었고 최소값은 랜덤으로 추출된 바이트
이다 보니 우연의 일치로 한번 낮은 엔트로피가 출력될 경우 기준으로 하기에 신뢰도
가 떨어져 평균으로 사용하였다. 하지만 암호화 통신 중 TLS 통신에서 TLS 헤더로 인해
근소한 차이로 평균값보다 낮은 엔트로피 값을 가지는 이벤트들이 있었다.

6. 페이로드 길이별로 만들어진 엔트로피 측정값 테이블에서 각 페이로드 길이 별 평균값 - (최대값 - 평균값)을 임계치로 하여 이 이상인 엔트로피 값을 가지는 페이로드를 암호화된 페이로드로 처리한다.

[배경 - 윤명근 교수님] 각 길이별로 엔트로피 값의 기준을 만드는 방법은 좋은 시도 인 것 같기 때문에 각 길이에서 상대적으로 사용할 수 있는 기준을 찾아서 전체 데이터에 잘 맞는지 실험을 진행하기로 하였다. 이때 전체 데이터 개수가 상당히 많기 때문에 적절한 데이터들을 몇 개 추출해서 그 데이터에 비교하면서 임계치를 찾았다.

_id	len	entropy	label	mean - diff	mean - diff_label	payload
dlb20a2f-f5ac-4213-9b2e-9e2a7d4d04d1	51	4.181462597	0	5.299049688		0 b'x03x00x00x003
09cf5176-9c3b-404b-bddc-e40890e777e1	51	4.181462597	0	5.299049688		0 b'x03x00x00x003
aa77714d-3043-4fb2-acfe-eb3bb1ce06ee	51	4.181462597	0	5.299049688		0 b'x03x00x00x003
2fcbaffe-a19f-4882-a1d3-6fbb64eeef6	51	4.181462597	0	5.299049688		0 b'x03x00x00x003
ef534ca0-525a-4598-adc4-c45e8a427743	51	4.181462597	0	5.299049688		0 b'x03x00x00x003
d4bbe6f1-3e12-415c-9b54-ecbb64aa0a	7	2.521640636	0	2.763354922		0 b'x15x03x00x01x
5c63e8e8-67b8-47ef-8466-8f50f3b78f	7	2.521640636	0	2.763354922		0 b'x15x03x00x01x
b8eeeb4b-b6cf-4c83-9e8c-a1103294fe3d	35	4.900711588	1	4.872314652		1 b'x17x03x00x03x
5b93a527-3ccf-4fe4-a11b-1327bb619c99	35	5.014997303	1	4.872314652		1 b'x17x03x00x03x
123efca7-8f19-4ac2-a99d-a8869efad2db	325	4.181603477	0	7.196764803		0 b'x81Cx01x00x03
6a23420d-7a55-4183-aca2-9e17bc4c3de7	67	5.804749265	1	5.60918496		1 b'x17x03x00x03x
790772ea-9541-4d3a-b658-be2918e3a3de	66	5.760229157	1	5.592411928		1 b'x17x03x00x03x
a0806efe-ce41-46ae-9382-811553e48602	62	5.696131794	1	5.529971071		1 b'x17x03x00x03x
bcd32955-b4cd-4a5b-8974-04c85944bc3d	61	5.701229141	1	5.483434177		1 b'x17x03x00x03x
44bbf5fa-d166-44a2-a93a-cb1d72be8954	62	5.696131794	1	5.529971071		1 b'x17x03x00x03x
6faa1e5f-2953-439b-b315-9c69907ebe39	32	4.9375	1	4.762603821		1 b'x17x03x00x03x
d892a49b-8df7-4507-ae66-ae67d4785efa	325	4.214695439	0	7.196764803		0 b'x81Cx01x00x03
d02a918-8b9-4280-8e51-8ca509ff0095	109	6.405709325	1	6.176276933		1 b'x17x03x00x03x
58c10379-0596-459b-b91e-6595f6cca889	145	6.709118314	1	6.450534674		1 b'x17x03x00x03x
5444c09f-3c1d-4fe6-9330-a2b28001655	35	4.843568731	1	4.872314652		0 b'x17x03x00x03x
0a7d0277-1f24-43ee-9106-969ca0b1b2	35	4.900711588	1	4.872314652		1 b'x17x03x00x03x
f174a53d-2f9-43b6-976a-a9ddc995645	360	7.383837941	1	7.249580077		1 b'x03x00x0e9x3-
c880f46-e466-4986-81fa-4e4b869b626	325	4.199757737	0	7.196764803		0 b'x81Cx01x00x03
e9945604-7159-4cbc-aed7-4d827bc4fd7a	32	4.6875	1	4.762603821		0 b'x17x03x00x03x
cd9fa896-306c-4223-b94-1dad26b0708	35	4.843568731	1	4.872314652		0 b'x17x03x00x03x

[그림10] 465번포트에서 전체 패킷 기준 별 전수 조사

[한계 - 윤명근 교수님] 데이터 적으로 괜찮은 기준이 만들어 졌지만 수학적으로 설득력 이 없고, 우연히 찾아낸 임계 값이다.

7. 페이로드 길이별로 만들어진 엔트로피 측정값 테이블에서 각 페이로드 길이 별 평균값 - 6*표준편차를 임계치로 하여 이 이상인 엔트로피 값을 가지는 페이로드를 암호화된 페이 로드로 처리한다.

[배경 1 - 윤명근 교수님] 수학적으로 사람들을 설득시킬 때 범주형 데이터에 가장 많이 사용되는 것이 정규분포이기 때문에 현재 가지고 있는 데이터가 정규분포로 표현이 가능 한지 확인을 해 가능하다면 정규분포_[별첨 2]를 통해 임계 값을 만드는 것이 설득력 있을 것 같았다.

[배경 2] 현재 가지고 있는 데이터는 암호화되었는지 안되어 있는지 라벨도 안 되어 있는 상태이고 그에 대한 판단 조차도 실험자가 디코딩 해보며 확인하는 방법 밖에 없기 때문 에 정확도에 대한 측정이 어려웠고 정확한 라벨에 대한 기준이 없었다. 만들 수 있는 것 은 암호화되었을 때라고 가정할 수 있는 랜덤 바이트만 있기 때문에 랜덤 바이트를 통해

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

암호화된 페이로드의 기준을 만들어 내서 가지고 있는 데이터들을 해당 기준에 적용시켜 암호화된 것을 분류하고 나머지는 평문으로 분류하였다.

[한계] 전체적으로 데이터가 기준에 잘 맞지만 일부 TLS 통신에서 헤더^[별첨 3]로 인해 임계 값을 통한 검증이 안되는 경우가 있었다.

- 5 번에서 만들어진 조건에 화이트리스트와 블랙리스트를 적용하여 엔트로피만으로 처리하기 어려웠던 부분에 대해 직접 처리해준다.

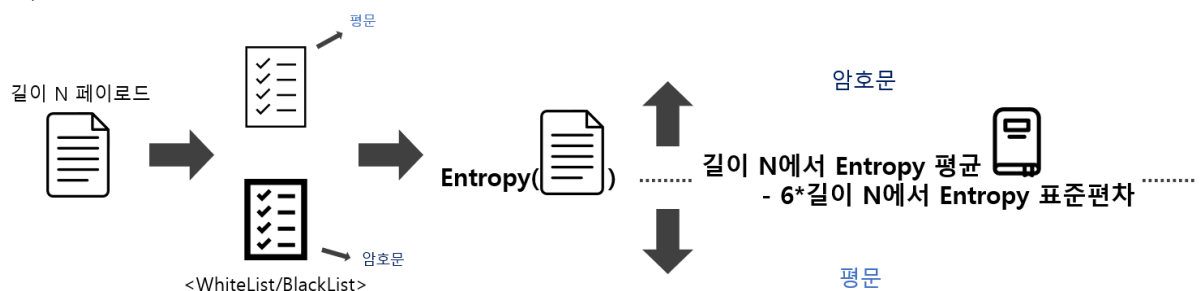
[배경] 지금까지의 임계 값을 통해 분리가 되지 않는 일부 이벤트 들이 있는데 이 이벤트 들은 공통적인 특징을 가지고 있다. TLS 헤더로 인해 불필요한 암호화 이벤트가 평문으로 취급되는 경우가 있고, 반면에 TLS 헤더 내에 난수 값을 전송하는 매개변수로 인해 필요한 이벤트가 암호문으로 취급되는 경우가 있다. 이 두경우 우리가 직접적으로 확인 가능한 특정 String^[별첨3]이 있기 때문에 해당 String을 통해 암호문으로 처리해야하는 경우에는 블랙리스트^[별첨4]를 통해 암호문으로 지정하고 평문으로 처리해야하는 경우에는 화이트리스트^[별첨4]를 통해 평문으로 지정한다.

[최종 결론]

각 페이로드의 길이에 대해 Entropy값이 정규분포 하기 때문에 그 길이에 해당하는 Entropy 평균 값에 6*표준편차^[별첨5] 값을 뺀 것보다 Entropy가 클 경우 암호화된 페이로드이다.

(암호화 된 페이로드 > 평균(엔트로피 길이) - 6*표준편차(엔트로피 길이))

단, 이벤트 페이로드에 대해 블랙리스트와 화이트리스트로 우선 처리한다.



[그림 11] 엔트로피를 통한 암호문/평문 구별

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

암호-비암호 구별[별첨]

[별첨1] Entropy

정보의 표본 공간이 N 개의 서로 다른 값들로 구성되어 있고 각 값의 비율(혹은 확률 $p_i, i=1,2,...,N$)을 알고 있을 때, 정보 엔트로피 S 는 다음과 같이 표현할 수 있다.

$$S = - \sum_{i=1}^N p_i \log p_i$$

즉, 1바이트는 256가지의 16진수로 표현이 되고 N-Byte의 페이로드가 있을 때 256가지의 16진수 중 특정 몇 가지만 계속 쓰일 경우 엔트로피가 낮고 256가지의 16진수가 골고루 사용되었을 때 엔트로피가 높게 측정이 된다.

일반적인 인터넷 통신에 사용되는 페이로드는 알파벳과 일부 특수문자가 주로 사용되어 지면서 통신을 하다 보니 엔트로피가 낮게 측정되고

암호화 통신의 경우에는 이런 일반적인 페이로드를 암호를 통해 완전히 난수화 시키기 때문에 여러 16진수가 쓰이게 되어 엔트로피가 높게 측정이 된다.

그러나, 1바이트는 256가지의 16진수로 표현이 되기 때문에 만약 페이로드의 크기가 256보다 작게 된다면 암호화 된 내용임에도 불구하고 16진수로 표현하였을 때 입력된 값이 몇 개 없기 때문에 특정 16진수만 사용 된 것처럼 계산돼 Entropy가 낮게 측정 될 수 있다.

[별첨 2] 정규분포

정규분포의 도수분포곡선은 평균 m 을 중심으로 좌우대칭인 종 모양을 이룬다. 이는 평균과 평균 근처에 많은 도수(확률)가 몰려 있고 평균에서 멀어질수록 도수도 급격히 적어지는 것을 의미한다. 정규분포의 도수분포곡선은 평균에서 좌우로 멀어질수록 x 축에 무한히 가까워지는 점근선이다. 또한 확률의 합은 1 이므로 도수분포곡선과 x 축 사이의 넓이는 1 이 된다.

정규분포를 결정하는 중요한 두 개의 값은 평균 m 과 표준편차 σ 이다.

[네이버 지식백과] [정규분포](#) [Normal Distribution, 正規分布] (두산백과)

여기에서 자연의 데이터 값을 정규화 시켰을 때 정규분포에 따른다면 정규분포의 신뢰 구간을 이용할 수 있다.

정규 분포의 신뢰구간은 평균에 표준편차를 더하고 뺀 값으로 데이터의 몇 퍼센트가 해당 구간에 있는지 알 수 있다. 보통 95%의 신뢰 구간은 평균 ± 1.96 *표준편차로 나타내고 99%의 신뢰 구간은 평균 ± 2.56 *표준편차로 나타낸다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

[별첨3] TLS 헤더

TLS는 인터넷 프로토콜에서 가장 기본적으로 사용되어지는 암호화 프로토콜이고 TLS 통신을 할 때에는 세션 처음에 TLS Header를 통해 어떤 자료가 통신 되어지는지 알리게 된다.

TLS Header에는 크게 4종류가 있다.

1. Handshake(Wx16): 종단 간에 보안 파라미터를 협상하기 위한 프로토콜
2. Change Cipher spec(Wx14): 협상된 보안 파라미터에 대해 적용/변경을 알리는 프로토콜
3. Alert(Wx15): SSL/TLS 통신 과정에서 발생하는 오류를 통보하기 위한 프로토콜
4. Application data(Wx17): 협상된 보안 파라미터로 직접 데이터를 통신하는 프로토콜

뒤에 괄호는 해당 프로토콜임을 알리는 시그니처 바이트이다.

이중 HandShake 프로토콜이나 Change Cipher Spec 프로토콜 같은 경우에는 공격자가 보안 파라미터를 조작해 공격 될 수 있는 여지가 있기 때문에(ex. Heartbleed Attack, Drown Attack, Poodle Attack..) 뒤에서 클러스터링을 진행할 때 필요한 정보이다. 반면에 Application Data 프로토콜이나 Alert 프로토콜 같은 경우에는 헤더 부분만 식별 가능하고 뒷부분은 모두 암호화된 정보로 지나 다니고 헤더 부분에도 공격으로 쓰인 요소가 없기 때문에 암호화된 정보라고 처리할 필요가 있었다.

[별첨4] 화이트리스트/블랙리스트

화이트리스트는 항상 허락해주는 무한 긍정의 리스트로 본 프로젝트에서는 화이트리스트에 있는 스트링으로 시작하는 페이로드의 경우에는 암호화되지 않은 페이로드로 처리를 하였다.


블랙리스트는 항상 차단해야 하는 무한 부정의 리스트로 본 프로젝트에서는 블랙리스트에 있는 스트링으로 시작하는 페이로드의 경우에는 암호화된 페이로드로 처리를 하였다.

화이트리스트[Wx140301, Wx140302 ...] -0302는 TLS Version

블랙리스트[Wx150301, Wx150302 ...]

[별첨5] 6-시그마

6 시그마는 모토로라가 등록한 상표이다. 시그마(σ)는 원래 정규분포에서 표준편차를 나타내며 6 표준편차인 100만 개 중 3.4개의 불량률(Defects per million opportunities, DPMO)을 추구한다는 의미에서 나온 말이다. 실제로 ± 6 시그마 수준은 100만 개중 2개의 불량(0.002% 불량률)로서, 6 시그마는 불량 제로를 추구하는 말이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

본 프로젝트에서는 2번 주제 이후 이뤄지는 클러스터링에 대해서 암호화된 페이로드로 인한 노이즈를 제거하기 위해 암호화 기준을 넓혀 노이즈가 될 만한 것을 모두 제거해야 한다는 의미로 6 시그마를 기준으로 두었다.

2.3.3 객체중심 오탐제거(진행 중)

앞서 보호대상 객체 식별 기술과 암호-비암호 판별 기술을 통해 원본 데이터에서 각 객체별로 비암호화 된 데이터들로 모을 수 있었다.

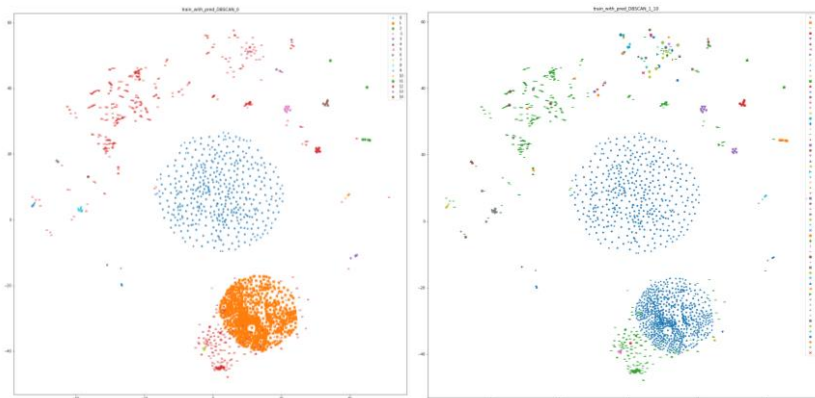
이제 정제된 데이터로 그룹화 시켜 각 그룹에서의 특징을 살펴보고 라벨이 잘못 붙어 있는 경우, 즉 한 그룹내에서 소수의 이벤트만 라벨이 다를 경우 그 소수의 이벤트를 중점적으로 사람을 확인 해 볼 수 있고, 한 그룹내에 모두 같은 라벨을 가지고 있고 공통적 특징으로 공격에 대한 시그니처를 보인다면 사람은 그 그룹에 대해 판단을 추가로 하지 않아도 된다.

이를 통해 사람은 라벨 재정합과정의 시간을 월등히 단축시킬 수 있고 컴퓨터를 이용한 더 상세한 정합을 통해 기존 목표였던 머신 러닝의 정확도도 올릴 수 있다.

이 목적을 위해서 현재 멜러리를 찾아서 팀은 페이로드를 청킹해서 각 페이로드들을 Stream Vector화 시켜 유사도를 기반으로 클러스터링 작업을 진행하고 있다.

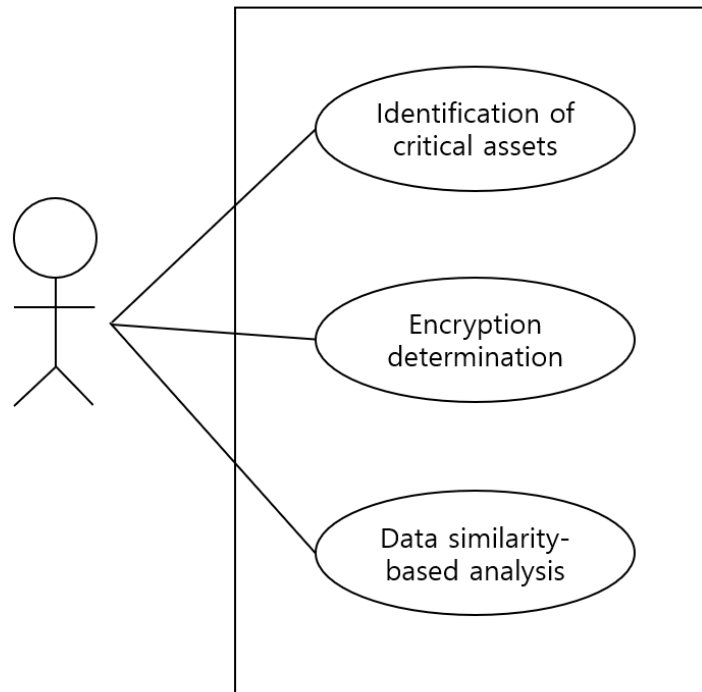
클러스터링에 관련해서는 실리콘밸리에서 현업에 종사하고 계신 문성익 멘토님과 많은 의견을 주고 받으면서 진행을 했다. 먼저 페이로드를 청킹할 때 현업에서는 n-gram을 많이 사용하였고, 클러스터링 과정에서 알고리즘에 따른 결과차이가 크지 않으니 전처리에 무게를 더 두라는 의견을 주셔서 먼저 청킹하는 전처리 과정에 대해 좀 더 찾아보았다.

현재 진행 중인 사항으로는 AE Chunking[3]을 통해 네트워크 페이로드에서 청킹된 벡터가 잘 나오는 윈도우 사이즈를 구하고 있고, 이에 대해 간단히 대표 클러스터링 알고리즘을 통해 결과 추이를 보고 있다.



[그림12] 알고리즘, NLP기법을 여러가지 시도하면서 실험 중

2.4 기대효과 및 활용방안



1) Identification of critical assets

- 보안 중요도가 높은 회사 내부 자산을 분리시켜 줄 수 있다.
- 회사 내부와 회사 외부에 대해 분리시켜 데이터를 다룰 수 있다.

2) Encryption determination

- 암호화된 페이로드를 분리시킬 수 있다.
- 분석에 무의미한 정보를 제외시킬 수 있다.

3) Data similarity-based analysis

- 유사한 이벤트끼리 군집화 시킬 수 있다.
- 데이터 분석 시 유사한 것끼리 묶어 일괄적 처리가 가능해진다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

+ 추가 적인 기대 효과

SNIPER BD1의 학습데이터 재 라벨링 뿐만 아니라 SNIPER BD1에서 출력하는 이벤트들에 대한 분석에도 적용하여 분석 소요 시간을 줄일 수 있다.

기존에 너무 많은 알람으로 인해 소요되는 시간을 O24SEC을 통해 알람을 군집화 시켜 만일 한 군집에서 공통적으로 나오는 단어가 공격 시그니처라면 해당 군집은 모두 공격으로 판단가능 하기 때문에 기존에 매우 많이 소요되던 시간을 줄일 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

3 프로젝트 팀 구성 및 역할 분담

프로젝트에 참여하는 멤버의 역할을 구체적으로 명시한다.


이름	역할
장우혁	<ul style="list-style-type: none"> - 팀장 - (정) 보호대상 객체식별 기술 개발 - (부) 암호-비암호 구별 기술 개발 - (공동) 객체 별 유사도 실험, 클러스터링 기법 실험
김민송	<ul style="list-style-type: none"> - 발표 - (정) 암호-비암호 구별 기술 개발 - (부) 보호대상 객체식별 기술 개발 - (공동) 객체 별 유사도 실험, 클러스터링 기법 실험

 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>캡스톤 디자인 I</p> </div>	보고서		
	프로젝트 명	O24Sec(Object-Oriented Clustering for Security Monitoring)	
	팀 명	멜러리를 찾아서	
	Confidential Restricted	Version 1.1	2021/04/03

4 개발 일정 및 의사소통

4.1 개발 일정

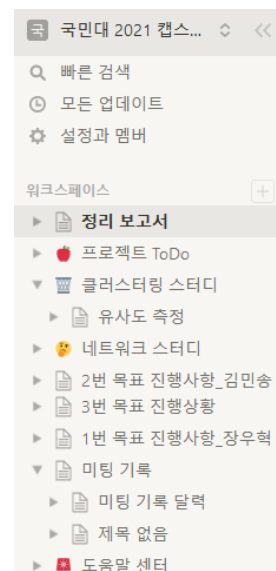


완료: 

4.2 프로젝트 간 의사소통

1. 팀원 간의 의사소통

현재 개발 상황이나 앞으로 스터디 해야할 부분,
공통적으로 관리 해야하는 미팅 기록이나 과제의
경우 노션(Notion)을 이용해 실시간으로 정보 주고받음



2. 산학 담당 교수님과의 의사소통
주 2 회 이상의 미팅을 통해 현재까지 진행상황과 앞으로 예정된 계획 주기적인 피드백
3. Ceeya 멘토 – 문성익 멘토님
보안 분야가 아니시기 때문에 보안 쪽 멘토링 보다는 클러스터링 위주로 질의 응답을 받아 과거 진행하셨던 프로젝트와 연결시켜 유사도 기반의 클러스터링 관련 아이디어 피드백
4. 산학 담당 원스 최병환 팀장님

안녕하세요

공유 감사합니다.

어제 논의한 회의록 공유 드립니다.

그리고 공유해주신 IP 분석 결과 다음과 같습니다.

=> 정상
한국과학기술정보연구원 (KREONet)

=> 위협 IP 확정
ISP ICME Limited
Usage Type Data Center/Web Hosting/Transit
Country Sweden
City Stockholm, Stockholms lan

데이터의 특이점이나 프로젝트 진척도를 주기적으로 공유하면서 회사와의 의견 일치 및, 방향성에 대한 점검 진행

5 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	논문	보안관제 정탐률 향상을 위한 인공지능 알고리즘 연구	대한전기학회, 전기학회논문 69(7)	2020.7	최승환, 장민해, 김명수	
2	발표	보안관제를 위한 AI 모델 기술 소개	NetSec-KR 2020, 과학기술사이버안전센터	2020	송중석	
3	논문	AE:An Asymmetric Extremum Content Defined Chunking Algorithm for Fast and Bandwidth-Efficient Data Deduplication	IEEE Conference on Computer Communications(INFOCOM)	2015	Yucheng Zhang 외 7 명	