Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.5, Oct. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.5.1141

N-gram을 활용한 DGA-DNS 유사도 분석 및 APT 공격 탐지

김 동 현,† 김 강 석‡
아주대학교

DGA-DNS Similarity Analysis and APT Attack Detection Using N-gram

Donghyeon Kim,† Kangseok Kim‡
Ajou University

요 약

APT(Advanced Persistent Threat) 공격에서 감염 호스트와 C&C(Command and Control) 서버 간 통신은 공격 대상의 내부로 침입하기 위한 핵심단계이다. 공격자는 C&C 서버를 통해 다수의 감염 호스트를 제어하고, 침입 및 공격 행위를 지시하는데, 이 단계에서 C&C 서버가 노출되면 공격은 실패할 수 있다. 따라서 최근의 경우 DGA(Domain Generation Algorithm)를 통해 C&C 서버의 DNS를 짧은 시간 간격으로 교체하여 탐지를 어렵게 하고 있다. 특히 하루에도 500만개 이상 새로 등록되는 DNS 전부를 검증하고 탐지하는 것은 매우 어렵다. 이러한 문제점을 해결하기 위해 본 논문에서는 정상 DNS와 DGA를 통해 생성된 DNS(DGA-DNS)의 형태적 유사도(similarity) 분석을 이용한 DGA-DNS 탐지와 이를 통해 APT 공격 징후로 판단하는 모델을 제시하고 유효성을 검증한다.

ABSTRACT

In an APT attack, the communication stage between infected hosts and C&C(Command and Control) server is the key stage for intrusion into the attack target. Attackers can control multiple infected hosts by the C&C Server and direct intrusion and exploitation. If the C&C Server is exposed at this stage, the attack will fail. Therefore, in recent years, the Domain Generation Algorithm (DGA) has replaced DNS in C&C Server with a short time interval for making detection difficult. In particular, it is very difficult to verify and detect all the newly registered DNS more than 5 million times a day. To solve these problems, this paper proposes a model to judge DGA-DNS detection by the morphological similarity analysis of normal DNS and DGA-DNS, and to determine the sign of APT attack through it, then we verify its validity.

Keywords: Advanced Persistent Threat, Intrusion Detection, Domain Generation Algorithm, N-gram, Data Analysis

I. 서 론

최근 사이버 위협은 특정 목표를 대상으로 한 공격이 대표적이며 침입한 시스템에서 오랫동안 발견되지 않고 은폐한 채 정보를 수집 또는 악용하는 것이 특징이다. 대표적인 예로 2016년 5월, 인터넷 쇼핑몰 인터파크가 1,030만 고객정보를 탈취당한 사고가 발생했다[1]. 이는 이메일을 통한 특정 목표 공격을 시작으로 네트워크 내부의 관련자 간 지속적인 감염을 통해 대규모 개인정보를 탈취당한 사례이다. 또 다른 예로, 2011년 ESTsoft의 업데이트 서버가 해킹을 당했고 해당 기업 소프트웨어를 사용하던 SK컴즈가

Received(06. 25. 2018), Modified(09. 19. 2018), Accepted(10. 04. 2018)
† 주저자, kzeroz@ajou.ac.kr
‡ 교신저자, kangskim@ajou.ac.kr(Corresponding author)

3,100만 건에 달하는 개인정보를 탈취 당했다. 이는 업데이트 서버의 취약점을 특정 목표로 공격하여 서버에 접속하는 사용자들의 PC를 지속적으로 감염시켜 대규모 피해를 입힌 사례이다[1]. 이처럼 지속적으로 은밀히 공격하며 정보를 탈취하는 방법을 지능형 지속 위협(APT, Advanced Persistent Threat)이라 한다. APT 공격은 다양한 공격기법을 활용하여 기존의 보안 솔루션을 우회하고 장기간에 걸쳐 취약점을 찾아 지속적으로 공격하므로 탐지 및 대응이 매우 어렵다. 특히 공격임에도 불구하고 마치 네트워크 정상행위로 가장함으로써 피해 발생 후에 식별되는 경우가 대부분이다. 그러므로 특이점을 찾기 힘든 APT 공격을 탐지하기 위해서는 공격을 구성하는 각 단계의 특징을 고려한 맞춤형 탐지 방법을 설계하는 것이 필요하다. APT 공격 단계 구분을 명확히 정한 바는 없지만 본 논문에서는 트렌드마이크로[2]에서 제시한 6단계로 나눠 다룬다. 공격 탐지를 위한 주요 단계로 2, 3, 4단계를 주목할 필요성이 있다.

- 1 단계 (Intelligence Gathering): 공격 목표의 IT 환경과 조직 구조 등 전략적 정보 수집
- 2 단계 (Point of Entry): 이메일, 인스턴트 메시징, 소셜 네트워킹 또는 악성코드 개발을 통해 네트워크에 진입
- 3 단계 (Command & Control): 공격자가 확보한 호스트와 C&C 서버 간 지속적 통신
- 4 단계 (Lateral Movement): 공격목표 네트워크 내부에서 중요 정보를 보유한 호스트 탐색
- 5 단계 (Asset / Data Discovery): 분리 추출할 가치 있는 데이터 식별
- 6 단계 (Data Exfiltration): 공격자가 통제하는 위치로 데이터 전송

즉, 〔2단계〕 최초 악성코드 침투 여부 탐지, 〔3단계〕 지속적 감염전파로 확보한 호스트 및 C&C 서버 간 통신 트래픽 탐지[3], 〔4단계〕 확산단계에서 공격탐지가 필요하다. 가장 최근의 공격사례 중 2017년 씨클리너 해킹 공격[4]과 구글 서비스를 통한 해킹 공격을 보면 모두 〔3단계〕에 해당하는 C&C 서버 통신단계를 적극적으로 활용했다. 〔3단계〕는 C&C 서버가 감염 호스트와 통신하는 단계로써 실제적인 명령을 주고받는 공격의 핵심단계이다. 그러므로 공격자는 APT 공격의 지속성을 유지하기 위해 이 단계가 차단되지 않도록 많은 노력을 기울인다. 특히 차단 회피를 위해 공격자는 주로 C&C 서버의 존재를 은폐시키는 방법을 사용하는데 IP 또는 DNS를 직접적으로 숨기거나 수시로 변화시키기도 한다. 이를 탐지하기 위해 기존에는 공격으로 의심되는 IP(또는 DNS)를 블랙리스트로 차단하고 정상 IP(또는 DNS)는 승인 후에 화이트리스트로 관리하는 방식을 주로 사용하였다. 이에 대응하여 공격자들은 Fast Flux를 통해 IP(또는 DNS)를 은닉하거나, DGA(Domain Generation Algorithm)등을 활용해 수시로 C&C 서버의 DNS를 바꿈으로써 공격 근원지를 은폐한다. 때문에 단순한 블랙리스트 관리만으로는 탐지하기가 점점 더 어려워지고 있다.

이처럼 〔3단계〕의 C&C 서버 통신단계는 APT 공격에서 핵심 단계이며 공격자는 이 단계가 봉쇄되지 않도록 지속적으로 은폐를 시도한다. 특히 C&C 서버 은폐를 위한 시도가 반드시 일어나게 되므로, 악성코드 분석을 통해 해당 시도를 탐지할 수 있다면 APT 공격 징후로 판단할 수 있을 것이다. 따라서 본 논문은 〔3단계〕 C&C 서버 통신단계에서 반드시 발생하는 C&C 서버 은폐시도 탐지를 통해 APT 공격 징후를 탐지하는 것을 목표로 한다. 다양한 은폐시도 방법 중 DNS를 통한 은폐시도를 중심으로 연구를 진행하였다. APT 공격에 사용된 악성코드들을 대상으로 Reverse Engineering에 의해 밝혀진 DGA를 통해 DGA-DNS 리스트를 생성하고 이를 정상 DNS 리스트와 유사도를 분석한다. 이 결과를 분석하여 공격에 사용가능한 DNS인지 판별하는 모델을 설계하였다. 유사도 분석에는 텍스트마이닝에 사용하는 알고리즘인 n-gram을 활용하였으며, 최종적으로 DGA-DNS 판별 임계점을 제시하였다.

본 논문의 2장에서는 DGA-DNS를 탐지하기 위한 다양한 연구를 살펴보고, 3장에서 n-gram을 활용한 DGA-DNS 유사도 분석 모델을 제시한다. 4장에서는 실험을 통해 모델에 대한 유효성을 검증하고, 마지막 5장에서 결론과 향후 적용방안에 대해 논의한다.

II. 관련 연구

세계적인 DNS 보안 기업 Nominum의 DNS 보안 트랜드 분석 보고서[5]에 따르면 현재까지 악성코드에 의한 공격은 90% 이상 DNS를 활용하고 있으며 특히 DGA가 생성한 DNS를 통해 새로운 호스트의 감염/업데이트 및 C&C 서버 통신에 사

용한다고 보고하고 있다. 공격자가 DGA를 사용하는 목적은 C&C 서버를 은닉하기 위함이다. APT 공격에 있어서 감염 호스트와 C&C 서버 간 통신단계는 해당 공격의 성공여부를 좌우하는 핵심단계이다. 만약 이 단계에서 C&C 서버가 노출되어 차단된다면 모든 단계의 노력은 수포로 돌아간다. 그러므로 공격자는 C&C 서버를 은폐하기 위해 많은 노력을 해왔다. 최초에는 C&C 서버의 IP를 하드코딩 하는 방식으로 숨기는 방식을 적용하였지만 고정된 IP를 사용하는 점에서 쉽게 차단되었다. 이러한 약점을 보완하기 위해 IP/DNS Fast Flux 방식을 적용하여 매 10초마다 DNS와 IP의 맵핑을 변경하였다. 하지만 [6,7]과 같이 Fast Flux 탐지 관련 연구들이 많이 진행되면서 이상(anomaly) DNS에 대한 블랙리스트 관리 등 접근통제 정책을 마련할 수 있게 되었다. 이에 따라 공격자들은 DGA를 개발하여 C&C 서버와의 DNS 맵핑을 짧은 시간 간격으로 변경함으로써 기존의 블랙리스트 정책을 무력화시켰고 최근까지도 C&C 서버 차단을 매우 어렵게 만들었다.

기존의 DGA-DNS 공격 탐지에는 네트워크 트래픽에서 발생하는 이상 쿼리 분석[8], DNS Profile 검증을 통한 화이트/블랙리스트 접근통제[5], DNS Cache 서버의 빅데이터를 분석하는 방법[5] 등이 활용되었다. 하지만 하루에도 500만개 이상의 DNS가 새로 등록된다는 점에서 트래픽 분석만으로는 적절한 탐지가 어려워지고 있다. 따라서 트래픽 분석 형태가 아닌 DNS 형태를 분석하여 이상 DNS임을 판별하는 연구들이 많이 다뤄지게 되었다. Raghuram 등의 연구[9]에서는 기존 화이트리스트 DNS로부터 사람이 발음할 수 있는 언어 성격을 지니고 있는지 특징적 분석을 통해 DGA-DNS를 판별하였다. Wei-Wei 등의 연구[10]에서는 어휘적 특징인 형태소(morpheme)를 기준으로 DGA-DNS를 판별하였다. Crawford 등의 연구[11]에서는 통계적인 관점에서 정상 DNS의 문자열 길이를 분석하고, 이를 벗어나는 이상 DNS를 구별하여 DGA-DNS를 판별하였다. Marchal 등의 연구[12]에서는 의미론(semantics)과 자연어 처리 분석을 통해 트래픽 상 수집되는 DNS 중 DGA-DNS를 판별하였다.

이와 같이 트래픽 분석이 아닌 DNS 형태를 분석하여 DGA-DNS를 판별하는 다양한 연구가 시도되었다. 다만 언어학적 특징을 통해 분석하는 것은 각 언어별 특이점을 모두 반영하기에 한계가 있으며, 특히 공격자가 언어별 사전형태의 데이터를 구축하여 그럴 듯한 단어로 DNS를 구성할 경우 무력화 될 가능성이 있다. 예를 들어 "whatyouwant.co.kr"이라는 가상의 DGA-DNS가 있다고 할 때, 언어학적 관점에서 발음이 가능하고, 문법에 이상이 없으며, 사전적 단어를 이용함으로써 그 의미가 분명히 전달되므로 정상 DNS로 판별할 가능성이 크다. 하지만 형태적 관점에서는 정상 운용중인 DNS와의 유사도 분석을 실시 할 경우 예시의 DNS와 유사한 기존 데이터가 없으므로 낮은 유사율로 인해 비정상 DNS로 판별할 것이다. 이와 같은 한계점으로 인해 본 논문에서는 언어적 측면을 제외하고 형태적 측면을 중심으로 연구를 진행하였다. 텍스트 분석 알고리즘(n-gram)을 활용하여 화이트 리스트 DNS 형태를 기준으로 유사도 분석을 통해 DGA-DNS를 판별하기 위한 모델을 제안한다.

III. 유사도 분석을 통한 DGA-DNS 탐지

3.1 기존 n-gram 분석방법

n-gram[13]은 자연어로 표현된 텍스트의 특징을 추출하여 단순한 기호의 나열로 다룰 수 있도록 한다. 즉, 기호열의 특징을 조사하기 위해 동일한 부분의 기호열이 반복되는지 확인하는데 이때 n개씩 잘라낸 기호열 중 같은 기호열이 발견되면 카운트하여 분석한다. Table 1의 경우 n-gram (n=5)을 작성한 예시이며 아래와 같이 대상 문장을 첫 부분부터 5개의 문자로 추출하여 각 단어의 출현 빈도를 조사하는 방식이다.

위와 같이 n-gram 분석방법[14,15,16]은 기호열에서 추출 가능한 n개의 연속된 시퀀스 집합을 얻고, 집합 내 구성요소 간의 유사성을 비교하는 것이다. 어떤 기호열 P의 기호열을 구성하는 전체 시퀀스를

Table 1. Example of n-gram creation (n=5)

Sentence : Minpyo was beginning to
Minpy
inpyo
npyo
pyo w
yo wa
o was

$(sp_1, sp_2, \cdots, sp_n)$ 이라고 할 때, $1 \le k \le n$ 을 만족하는 k에 대해 이 기호열로 부터 추출한 k-gram 집합인 $kgramS(P)$는 아래와 같이 정의된다[14].

$$kgramS(P) = \{(sp_i, sp_{i+1}, \cdots, sp_{i+k-1}) | 1 \le i \le n-k+1\}$$

기호열의 총 시퀀스 길이가 n이므로, 중복되는 k-gram이 나타나지 않을 경우 k-gram 집합은 최대 n-k+1개의 k-gram에 포함될 수 있다. n-gram 방법은 두 기호열을 비교하기 위해 각각의 기호열로 부터 추출한 n-gram 집합의 원소가 얼마나 유사한지를 평가한다. 동일한 n-gram이 많다는 것은 길이가 n인 동일한 기호열 패턴을 많이 공유하고 있다는 의미이며 이를 통해 두 기호열 간 유사성을 예측할 수 있다. 두 기호열 P, Q가 있을 때, 각각의 n-gram 집합을 ngramS(P), ngramS(Q)라 할 때, 두 기호열간 유사도 Similarity(P,Q)는 아래와 같이 정의된다[14].

$$Similarity(P, Q) = \frac{|ngramS(P) \cap ngramS(Q)|}{\min(|ngramS(P)|, |ngramS(Q)|)}$$

여기서 |S|는 집합 S에 포함된 원소의 개수를 의미하며, 두 기호열의 유사성을 분석하기 위해 전체 n-gram의 개수 중에서 공유하고 있는 n-gram의 개수의 비율을 측정하는 방법을 이용한다.

3.2 제안된 n-gram 분석방법

기존의 유사도 Similarity(P,Q)의 경우, 두 집합 중 최소 원소수를 기준으로 분모를 정하는데 이를 DGA-DNS 일치율(concordance rate)에 적용할 경우 문제가 발생할 수 있다. 그 이유는 DGA-DNS가 정상 DNS를 포함하고, 그 뒤에 임의의 기호열을 추가할 경우 DGA-DNS이지만 매우 높은 일치율로 판별될 수 있기 때문이다. 즉, Table 2에서 보는 바와 같이 DGA가 기존의 정상 DNS를 공격에 활용할 경우 기존 식으로는 일치율이 실제보다 크게 증가하는 문제점이 있다.

Table 2. Comparison of normal DNS and DGA-DNS

Normal DNS	DGA-DNS	Min Match Rate (%)		
		Exist	Modify	Differ
google .com	googleqrlads.com	55.5	33.3	-22.2
ahnlab.com	ahnlab.com.tr	100	75.0	-25.0

이를 해결하기 위해 본 논문에서는 두 집합 중 최소 원소수(Min)가 아닌, 최대 원소수(Max)를 분모로 사용하여 정상 DNS를 공격에 활용한 경우에도 탐지할 수 있도록 개선하였다. 아래는 개선한 유사도 식을 나타낸다.

$$Similarity(P, Q) = \frac{|ngramS(P) \cap ngramS(Q)|}{\max(|ngramS(P)|, |ngramS(Q)|)}$$

3.3 n-gram을 적용한 DGA-DNS 탐지 모델 설계

n-gram 알고리즘에 적용할 기준(정상) DNS 데이터셋(dataset)은 Amazon에서 제공하는 Alexa Top 1 Million Site[17]의 데이터를 활용하였다. Alexa Top 1M은 전 세계 웹 사이트에서 발생하는 네트워크 트래픽을 비교하여 1위부터 1,000,000위까지 순위를 낸 리스트로써 해당 사이트의 규모를 비교할 수 있는 데이터이다. 즉, 신뢰성이 보장되고, 정상적으로 운영되고 있는 사이트의 DNS 리스트인 Alexa Top 1M을 화이트 리스트로 활용하여, 이를 기준으로 DGA에 의해 생성된 DNS인지 판별하도록 설계하였다. 탐지모델은 정상 DNS를 기준으로 DGA-DNS(DGA에 의해 생성된 DNS)를 비교하여 유사도를 측정하였다. 이를 위해 먼저 Alexa Top 1M을 입력한 n-gram 알고리즘을 적용하여 분절된 DNS 기호열을 생성하였다. 분절된 DNS 기호열은 n-gram 알고리즘이 적용된 DGA-DNS 기호열을 비교하는 대조군(기준) 데이터로 사용하였다. Table 3은 n=4 일 때, Dictionary 구성의 예시를 나타낸다.

Dictionary가 완성되면, DGA 알고리즘에 의해 생성된 DNS를 n-gram 알고리즘에 적용하고, 그 결과를 기준 Dictionary와 비교하여 유사도를 계산한다. 참고로, 분석에 사용되는 DGA-DNS 데이터셋은 Reverse Engineering에 의해 밝혀진 악성코드의 DGA Algorithm[18]을 통해 얻을 수 있으며 유

Table 3. Example of n-gram dictionary (n=4)

(Site) : amazon.com (Dictionary) : amaz mazo azon zon. on.c n.co .com

사도 계산방식은 3.2에서 제시한 Similarity(P,Q) 식을 이용하여 구할 수 있다.

위 실험에서 주목할 점은, 1단계 실험에서 n-gram 알고리즘을 적용하는 과정에서 n 값에 따라 유사도 판별 결과가 크게 달라질 수 있기 때문에 어떤 n 값을 사용할지를 결정해야한다. 그러므로 DGA-DNS 판별에 가장 적절한 n 값을 찾는 과정이 실험에서 수반되었다. 단계별 실험 목표는 아래와 같으며 전 단계에서 사용되는 n-gram 알고리즘은 Table 4와 같다.

1 단계: Alexa Top 1M (정상 DNS List)을 n-gram 알고리즘에 적용하여 탐지모델에서 사용 할 DNS Dictionary를 구성한다.

2 단계: Locky의 DGA로 부터 생성된 DGA-DNS를 n-gram 알고리즘에 적용하고, 1 단계에서 구축된 Dictionary에 적용하여 유사도를 분석한다. 이를 통해 탐지모델의 성능을 점검한다.

3 단계: Locky 외 다른 악성코드 14개의 DGA에 적용하여 탐지모델의 유효성을 확인한다.

Table 4. DNS similarity analysis code

```
Similarity (P, Q)
{
   index i, j;
   count cnt = 0;
   concordance result = 0;

   for (i = 0 to |ngramS(P)|)
      for (j = 0 to |ngramS(Q)|)
         if (compare(P(i), Q(j)) == SIMILAR)
           cnt = cnt + 1;

   result = cnt / max(|kgramS(P)|, |kgramS(Q)|);

   return result;
}
```

IV. 실 험

4.1 실험 및 환경

실험은 3단계로 구성되어지며, 1단계에서 실험을 위한 모델을 구축하고, 2단계에서 정상 DNS와 DGA-DNS 사이에 유사도를 정상 판별하는지 검증을 실시하였으며 3단계에서 14개 Malware로 부터 얻은 DGA-DNS를 탐지 모델에 적용하여 탐지 능력을 확인하였다. 실험 환경은 Table 5와 같다. C++를 이용해 n-gram 알고리즘을 구현하였고 Python으로 구현된 Malware의 DGA를 통해 DGA-DNS를 추출하였다. Table 6은 실험에 사용된 DGA-DNS의 패턴과 특징을 나타낸다.

Table 5. Experimental environment

System	Specification
CPU	Intel Core i7 2.60GHz
RAM	8 GB
HDD	NVMe. SSD 128 GB
Compiler	Visual Studio 2017 (C++)
	Pycharm (Python 3.6)
Data	Amazon ALEXA Top 1M (Authorized DNS List)

Table 6. DNS patterns generated from DGA

DGA	Pattern / Features
banjori	earnestnessbiophysicalohax.com
	random characters+existing DNS
corebot, symmi	3japih7hsvufc27f7ds25.ddns.net
	random DNS generation of top-level / second-level domain (.ddns.net)
locky, necurs, proslikefan	ktcjrnbeuurxhvclr.org
	random DNS generation using year/month/day
nymaim	puocjiffoxu.info
	random DNS generation using MD5-based pseudo-random number
pykspa	qkltecuiwcymao.net
	random DNS generation with 5,000 pre-generated host sets and Unix timestamp
qadars	0aw6kugqw642.org
	random DNS generation after generating pseudo random number with linear joint generator
qakbot, ranbyus, tinba	hznlhsxxqtcakxgmngn.biz
	random DNS generation using a user-defined random number generator
vawtrak	lemonwuhal.com
	random DNS generation of top-level domain (com)

4.2 (1단계) 최적 n 선택 실험

1단계 실험모델 구축을 위해 Alexa Top 1M 1,000,000개 데이터를 n=4 조건으로 n-gram 알고리즘에 입력한 결과, 총 기호열(Dictionary) 12,063,010개를 얻었으며, 결과를 얻는데 까지 실험 환경 기준 약 11초 정도 소요되었다. n에 따라 생성 기호열 수는 약 ±100만개정도 차이가 있었으며, n이 증가할수록 기호열 수는 감소한다. 구축한 기호열 리스트는 DGA-DNS 판별에 반복적으로 사용하기 위해 텍스트 파일로 추출하였다. Table 7은 n에 따른 Dictionary 구축 소요 시간 및 생성된 기호열 수를 나타낸다.

최초 유사도 실험을 위해 실험군으로 랜섬웨어(Locky)의 DGA-DNS를 사용하고 대조군으로 위의 Dictionary(Alexa Top 1M)를 사용하였다. 유사도 분석은 3.2절의 수정 유사도 분석 식을 기반으로 실험하였고, n에 따라 유사도 분석에 따른 DGA 일치율에 차이가 발생하였다. 이는 신뢰도에 영향을 미칠 수 있는 사항이므로, 적절한 판별을 위한 n의 범위 설정이 필요하다.

Table 8은 n에 따른 표본데이터(DGA-DNS) 1천개 당 최대, 최소-일치율 결과를 나타낸다. 단, 표 해석에 있어서 주의할 점은 「일치율 ≠ 탐지율」이란 점이다. 일치율은 대조군(정상 DNS)와 실험군(DGA-DNS) 간 얼마나 유사한지 나타내는 수치이다. 그러므로 일치율이 크다는 것은 실험군이 정상 DNS일 가능성이 높다는 의미이다. 반면, 탐지율의 경우 실험군의 DGA-DNS일 가능성을 나타내는 것으로써 이 수치가 높다는 것은 실험군이 DGA가 생성한 DNS일 가능성이 높다는 의미이다. 결론적으로 (일치율(%) ∝ 1 / DGA-DNS 탐지율(%))의 관계로 설명되며 일치율이 높을 때 탐지율은 낮고, 탐지율이 높으면 일치율은 낮게 나타난다. Table 8의 결과를 보면 n이 증가할수록 최대-일치율 및 최소-일치율이 모두 감소하는 것을 볼 수 있다. 그 이유는 n이 커질수록 n-gram이 생성하는 기호열의 길이가 증가하는데 그만큼 판별해야 할 경우의 수가 증가하므로 확률적으로 일치율은 감소하게 된다. 이때 주목할 점은 n=2, n=3 일 때 최소-일치율이다. 최소-일치율의 경우 그 값이 높을수록 불리한데, 그 이유는 최소-일치율과 n의 관계 때문이다. 최소-일치율은 n이 작아질수록 그 값이 커지는데 n이 작아지는 것은 n-gram이 만들어내는 기호열이 더욱 짧아짐을 의미한다. 기호열이 짧아질수록 비교하는 범위가 좁아지지만 비교할 경우의 수는 늘어나므로 유사한 기호열을 찾을 가능성도 높아진다. 하지만 짧은 분절의 기호열을 판별에 사용할수록 사람이 인식하기 어려운 형태의 기호열을 일치율에 반영시킬 가능성이 커진다. 특히 DGA는 Seed값을 통해 공격자가 의도한 랜덤 함수를 실행시켜 DNS를 생성하므로 일반적인 단어 형태가 아닌 무작위 기호열 형태로 많이 존재한다. 그러므로 짧은 분절(n=2, 3인 경우)로 구성된 기호열을 통한 유사도 판별은 정상 DNS와의 일치도 분석에 있어서 신뢰도를 저하시킬 수 있다.

Table 9는 실제 실험에서 발생한 최소-일치율에 해당하는 DGA-DNS 리스트이며, 이를 통해 무작위 기호열이 일치율에 반영되고 있음을 확인할 수 있

Table 7. N-gram generation symbol strings

N-gram	Time (sec)	Number of Symbol Strings
2	13	14,063,010
3	12	13,063,010
4	11	12,063,010
5	12	11,063,010
6	12	10,063,069

Table 8. Experimental group (DGA-DNS) concordance rate according to N

N-gram	Concordance Rate (%)			False Positive	*Reliability (%)
	Max	Min	Average		
n = 2	58.33	10.53	29.07	0	100%
n = 3	54.55	5.56	22.12	0	100%
n = 4	50.00	0.00	15.82	0	100%
n = 5	44.44	0.00	8.65	0	100%
n = 6	37.50	0.00	3.11	0	100%

* 신뢰도(%) = $\frac{(True\ Positive - False\ Positive)}{True\ Positive} \times 100$

Table 9. Comparison of normal DNS and DGA-DNS according to N

N-gram	Concordance Rate (%) Min	Noraml DNS Symbol Strings	DGA-DNS Symbol Strings
n = 2	10.53	a**pk**home.net ap**kh**ome.net	e**pk**huhrvkruvttaqb.su ep**kh**uhrvkruvttaqb.su
n = 3	5.56	a**pkh**ome.net	e**pkh**uhrvkruvttaqb.su
		dailypakistan.co**m.p**k	yjqdvfsjgkiolvmk**m.p**w
		my**uhc**.com	uq**uhc**wamdrfditwwb.su

다. Fig. 1은 n에 따른 일치율 누적분포곡선이며 n=2, n=3의 시작점이 각각 10.53(%), 5.56(%)인 것은 동일한 원인에 따른 것이다. 결론적으로 n-gram을 이용한 DGA-DNS 탐지는 일치율은 낮으면서, 탐지율은 높은 n을 선택하는 것이 가장 유리하다. 이를 위해 본 논문에서 제시하는 조건은 2가지이다. 첫 번째는 위에서 밝힌 바와 같이 짧은 분절에 대한 의미 없는 유사도 판별을 하지 않도록 최소-일치율이 0.00(%)가 되는 n을 선택하는 것이다. 이는 Table 8을 통해 확인할 수 있는 바와 같이 $n \geq 4$일 때에 해당한다. 두 번째로 다양한 DGA-DNS에 대한 판별능력을 고려해야한다. Fig. 1은 n에 따라 DGA-DNS와 일치하는 기호열 수의 누적분포를 나타낸 그래프이다. 그래프를 보면 n=5와 n=6의 경우, 일치율 0.00(%)에 각각 336개, 735개로 일치하는 기호열이 밀집하는 경향이 있다. 이를 통해 n이 증가할수록 다양한 형태의 DGA-DNS에 대한 적응도가 떨어지고, 한쪽으로 치우치는 특징이 관찰된다. 그러므로 위 2가지 조건을 고려할 때, 실험을 통해 n=4일 때 탐지모델에 적용하는 것이 가장 바람직하다.

4.3 (2단계) 최적 n을 적용한 DGA 탐지 실험

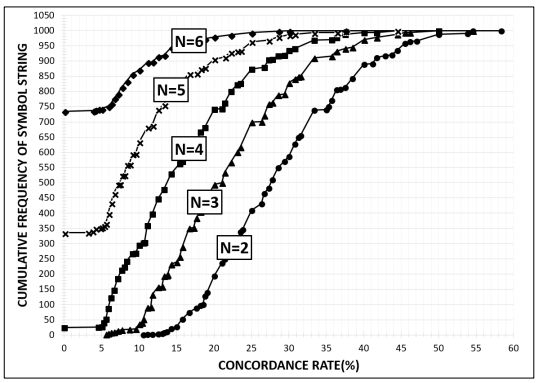

Fig. 1. Concordance rate cumulative distribution curve according to N

2단계 실험을 통해 정상 DNS와 DGA가 생성한 DNS를 모두 포함하는 데이터 내에서 DGA-DNS를 얼마나 잘 판별할 수 있는지 확인하였다. 특히, 일치율(%)에 따른 DGA-DNS 판별을 위한 임계점을 결정하기 위해 표본 데이터 Locky(DGA-DNS)를 1000개에서 10,000개로 증가 시켰고, 평균적으로 15.90(%) 이하의 일치율을 보이는 것을 확인하였다. 다만 DGA-DNS 판별은 최대-일치율이 해당 DGA를 판별할 수 있는 임계점이 되므로, 평균값 자체는 큰 의미가 없다. 결론적으로 임계점은 Locky DGA의 최대-일치율 66.67%로 결정하였다. Table 10은 2단계 실험 결과이며 Fig. 2의 그래프는

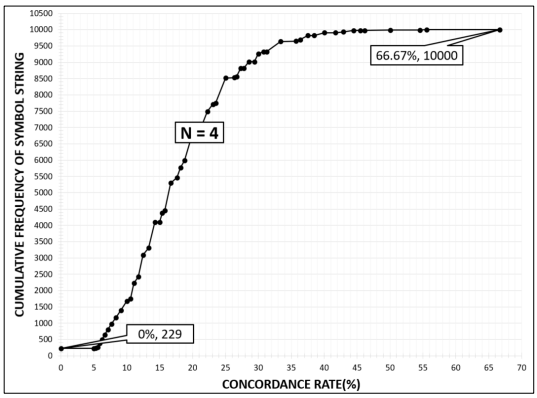

Fig. 2. Concordance rate cumulative distribution curve in case of n=4 and 10,000 data

Table 10. Experimental group (Locky DGA-DNS) concordance rate (n=4)

Division		Data
N-gram		4
Concordance Rate (%)	Average	15.90
	Max	66.67
	Min	0.00
Standard Deviation (σ)		8.73
Time (min)		24.04

Table 11. Results of applying 14 DGA algorithms

Malware DGA Algorithm	Concordance Rate (%)			Standard Deviation (σ)	True Positive (%)	False Positive (%)	*Θ〈 Concordance (%)		Time (min)
	Avr	Max	Min				Count	Avr	
banjori	20.72	57.14	18.18	4.36	100	0.00	-	-	25:25
corebot	16.87	40.00	6.90	8.81	100	0.00	-	-	24:31
★ locky	15.90	66.67	0.00	8.37	100	0.00	-	-	24:04
necurs	13.47	57.14	0.00	7.86	100	0.00	-	-	23:32
nymaim	31.30	85.71	8.33	10.10	**99.9964**	**0.0036**	**36**	**77.38**	25:10
proslikefan	16.22	65.00	0.00	10.52	100	0.00	-	-	21:09
pykspa	26.53	85.71	7.69	10.75	**99.9994**	**0.0006**	**6**	**73.38**	22:55
qadars	16.85	46.15	7.69	5.30	100	0.00	-	-	23:30
qakbot	17.37	55.56	3.85	7.82	100	0.00	-	-	24:17
ranbyus	12.28	62.5	0.00	6.48	100	0.00	-	-	24:45
simda shiz	44.14	87.5	25.00	8.05	**99.9965**	**0.0035**	**35**	**81.25**	20:34
symmi	36.70	64.29	28.57	5.93	100	0.00	-	-	25:06
tinba	19.93	81.82	0.00	7.09	**99.9999**	**0.0001**	**1**	**81.82**	23:20
vawtrak	39.85	87.5	16.67	10.26	**99.9902**	**0.0098**	**98**	**77.17**	24:46
Result (Average)	24.50	68.94	8.78	7.95	99.9965	0.0035	35.2	78.02	23:24

10,000개 DGA-DNS 입력에 따른 유사도 분석 분포도를 나타낸다.

4.4 (3단계) DGA 탐지 모델의 일반화 실험

3단계 실험은 1단계에서 설계한 모델 및 2단계에서 설정한 임계점을 적용하여 Locky 포함 14개 악성코드의 DGA로부터 생성한 DGA-DNS를 실험군으로 사용하였다. 14개의 DGA로부터 각각 10,000개의 DGA-DNS를 추출하였고, 결과적으로 총 140,000개의 데이터를 실험군으로 사용하였다. 실험 조건은 n=4, 임계점(66.67%)일 때 DGA 판별로 설정하였다. 실험 결과에 따른, True Positive는 99.9965(%), False Positive는 0.0035(%)로, 신뢰도 99.9(%)를 보장함을 확인하였다. Fig. 3의 그래프는 14개 DGA-DNS 입력에 따른 유사도 분석 분포도이며, Table 11은 3단계 실험 결과를 나타낸다. 실험결과 14개의 DGA 중 5개의 DGA에서 False Positive가 관찰되었는데 각각의 데이터(10,000개)에서 차지하는 비율은 0.01% 이하였다.

그러나 vawtrak DGA의 경우 특이하게 비교적 높은 False Positive가 관찰되었다. vawtrak의 경우 임계점(Θ) 초과(Θ 〈 일치율)에 해당하는 DGA-DNS가 총 10,000개 중 98개(Count)이며 이들의 평균 일치율(Avr)은 77.17(%)이다. 이와 같이 임계점을 초과한 DGA-DNS는 격리를 통해 분석한 후 Θ 값에 반영시키는 과정이 필요하다. 즉, 격리를 통해 정상 DNS가 아님이 판명되면 Table 11의 「Θ 〈 일치율」에 해당하는 기호열들의 평균 일치율(78.02%)을 새로운 Θ 값으로 반영함으로써 주어진 악성코드의 DGA-DNS를 거의 모두 판별할 수 있으며 이는 Fig. 3의 그래프 결과를 통해서도 확인 가능하다. Fig. 3은 14개의 DGA로 부터 얻은 DGA-DNS 140,000개의 유사도를 분석한 결과이며 각 일치율 별 DNS의 분포를 나타낸다. 가장 많은 수의 DGA-DNS가 해당하는 일치율은 37.5(%)이며 그 수는 8,779개이다. 또한, 새로운 Θ 값인 일치율 78.02(%)에 포함되는 DNS는 3개이며 임계점에 가까울수록 DNS의 분포 수는 크게 감소한다.

V. 결 론

지금까지 실험을 통해 n-gram을 활용한 DGA-DNS 탐지를 확인해 보았다. 실험 결과를 통해 실제 악성코드의 DGA로부터 생성된 DNS를 판별할 수 있음을 확인하였고, 이를 통해 공격자 의도에 의한 C&C 서버 은폐 시도를 탐지할 수 있었다. 따라서 APT 공격 징후 탐지에 사용할 수 있는 유효한 지표임을 확인하였다. 특히, APT 공격이 정상 네트워크 행위로 가장하는 등 그 방식이 매우 은밀하여 공격 징후 탐지가 어렵기 때문에 공격 단계 중 반드

* Θ (Threshold) 〈 Concordance Rate(%) : Locky의 최대 일치율(Θ, 66.67%)을 초과한 기호열 수

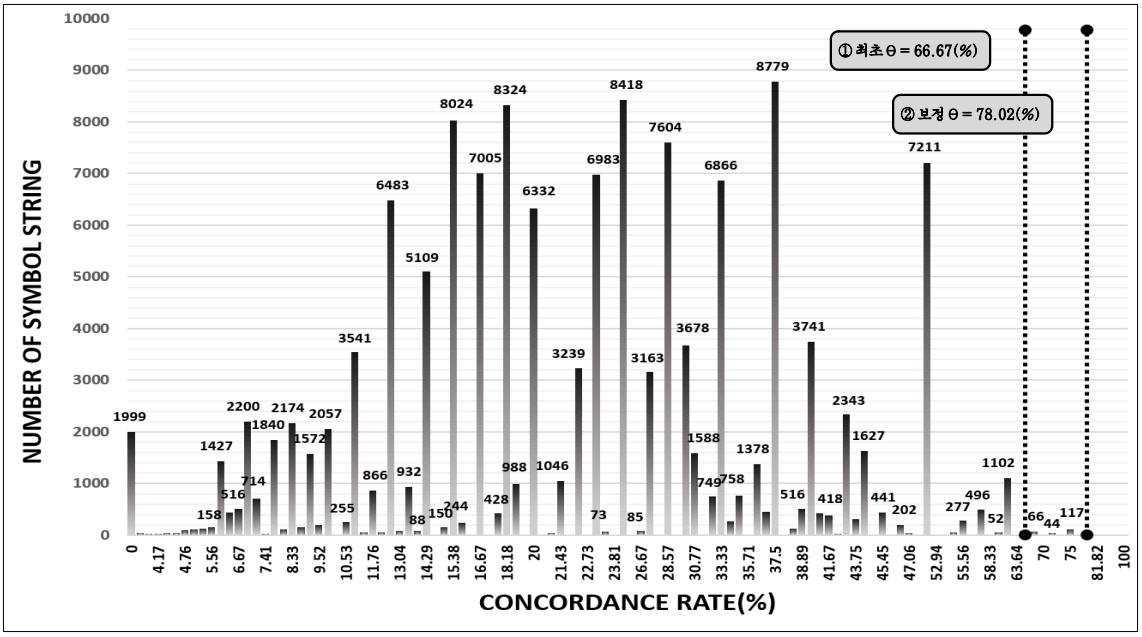

Fig. 3. Results of applying 14 DGA algorithms: concordance rate and number of symbol strings

시 발생하는 C&C 서버와 감염 호스트 간 통신 단계의 특징을 탐지에 활용했다는 점에서 의미가 있다.

본 논문에서 제안한 모델은 DNS의 형태적 관점에서 기존 정상 DNS와의 유사도 분석을 통해 공격자가 생성한 DNS 여부를 판별하는 것이 핵심이다. 이는 기존의 탐지 방법(트래픽 이상 쿼리 분석, DNS Profile 검증, DNS Cache 분석 등) 대비 매우 많은 수의 DNS를 직관적으로 빠르게 판별할 수 있으며 특히, 언어학적 관점에서 독립적이므로 그 형태와 상관없이 새로 생성된 DNS에 대해 유사도 판별을 수행할 수 있다는 장점이 있다.

그러나 본 논문이 제시하는 모델을 실시간에 적용하기 위해서는 다음과 같은 사항을 향후 연구에서 추가해야 할 것으로 판단된다. 첫 번째는 임계점을 동적으로 반영시키는 것이 필요하다. 14개 악성코드 DGA 외 계속해서 만들어지는 신규 DGA 알고리즘을 반영하기 위해 주기적으로 임계점을 계산하고 업데이트해야 한다.

만약 공격자가 DGA 알고리즘을 단순히 랜덤함수에 의해 기호열을 생성하는 것이 아닌, 그럴 듯한 단어를 혼용하는 등의 방법을 사용할 경우 일치율이 증가하여 False Positive가 다수 판별될 가능성이 있다. 두 번째는 주기적인 화이트 리스트 추가가 필요하다. 본 논문의 모델은 ALEXA의 100만개 DNS를 사용하였지만, 새로운 DNS가 출현했을 시 해당 DNS가 정상적인 DNS인지 공신력 있는 데이터베이스를 통해 화이트 리스트에 반영함으로써 지속적으로 DGA-DNS를 판별할 Dictionary를 업데이트 시켜야 한다. 이를 통해 제시된 모델의 탐지 유효율을 지속적으로 확보 할 수 있을 것으로 기대한다.

References

[1] Sul-Hwa Im, Jong-Soo Kim, Jun-Keun Yang and Chae-Ho Lim, "APT status and new malicious code countermeasures," Review of KISSC(Korea Institute of Information Security and Cryptology), 24(2), pp. 63-72, Apr. 2014.

[2] S. Hsieh, "Building threat intelligence to detect APTs in lateral movement," Trend Micro, July, 2013. https://blog.trendmicro.com/trendlabs-security-intelligence/building-threat-intelligence-to-detect-apts-in-lateral-movement/

[3] Dae-Sung Moon, Han-Sung Lee and

Ik-Kyun Kim, "Host based feature description method for detecting APT attack," Journal of The Korea Institute of Information Security and Cryptology, 24(5), pp. 839-850, Oct. 2014.

[4] P. Rascagneres, "CCleanup: A vast number of machines at risk," Cisco Talos Report, Sept. 2017. https://www.cecyf.fr/wp-content/uploads/2018/01/2018-RASCAGNERES-CCleaner.pdf

[5] Jun-Woo Park, "Security trend analysis with DNS," Information Sharing Cyber Infringement Accident Seminar in Korea Internet and Security Agency, Sept. 2017. https://www.boho.or.kr/data/reportView.do?bulletin_writing_sequence=26711

[6] D. Truong and G. Cheng, "Detecting domain-flux botnet based on DNS traffic features in managed network," Security and Communication Networks, vol.9, no.14, pp.2338-2347, May 2016.

[7] R. Sharifnya and M. Abadi, "DFBotKiller: domain-flux botnet detection based on the history of group activities and failures in DNS traffic," Digital Investigation, vol.12, pp.15-26, Mar. 2015.

[8] Sun-Hee Lim, Jong-Hyun Kim and Byung-Gil Lee, "Detecting cyber threats domains based on DNS traffic," Journal of Korea Information and Communications Society, 37(11), pp.1082-1089, Nov. 2012.

[9] J. Raghuram, D.J. Miller, and G. Kesidis, "Unsupervised, low latency anomaly detection of algorithmically generated domain names by generative probabilistic modeling," Journal of Advanced Research, vol.5, no.4, pp.423-433, July 2014.

[10] Z. Wei-wei, G. Jian, and L. Qian, "Detecting machine generated domain names based on morpheme features," Proceedings of the 1st International Workshop on Cloud Computing and Information Security (CCIS 2013), pp.408-411, Oct. 2013.

[11] H. Crawford and J. Aycock, "Kwyjibo: automatic domain name generation," Software: Practice and Experience, vol.38, no.14, pp.1561 - 1567, Apr. 2008.

[12] S. Marchal, J. François, R. State, and T. Engel, "Semantic based DNS forensics," IEEE International Workshop on Information Forensics and Security (WIFS 2012), pp.91 - 96, Dec. 2012. DOI: 10.1109/WIFS.2012.6412631

[13] H. Wallach, "Topic modeling: beyond bag-of-words," Proceedings of the 23rd International Conference on Machine Learning (ICML 2006), pp.977-984, June 2006. DOI: 10.1145/1143844.1143967

[14] Hyun-il Lim, "Comparing binary programs using approximate matching of k-grams," Journal of KIISE: Computing Practices and Letters, 18(4), pp.288-299, Apr. 2012.

[15] Hee-Jun Kwon, Sun-Woo Kim and Eul-Gyu Im, "An Malware classification system using multi n-gram," Journal of Security Engineering, 9(6), pp.531-542, Dec. 2012.

[16] Myung-Gwon Hwang, Dong-Jin Choi, Hyo-Gap Lee, Chang Choi, Byeong-Kyu Ko and Pan-Koo Kim, "Domain n-gram construction and its application," Proceedings of the Korea Information Science Society Conference, 37(2C), pp.47-51, Nov. 2010.

[17] Amazon Alexa Top Sites, https://aws.amazon.com/ko/alexa-top-sites/

[18] A. Sood and S. Zeadally, "A taxonomy of domain-generation algorithms,"

IEEE Security & Privacy Magazine,
vol.14, no.4, pp.46-53, Aug. 2016.

〈저자소개〉

김 동 현 (Donghyeon Kim) 정회원
2014년 2월: 경북대학교 생명공학과 졸업
2018년 8월: 아주대학교 정보통신대학원 석사
2018년 9월~현재: 아주대학교 컴퓨터공학과 박사과정
2015년 11월~현재: 국방부 전산군무주무관
〈관심분야〉 정보보호, 빅데이터 보안, 융합보안

김 강 석 (Kangseok Kim) 정회원
2007년: 인디애나대학교 컴퓨터공학(박사)
2010년~2016년: 아주대학교 대학원 지식정보공학과 연구교수
2016년~현재: 아주대학교 사이버보안학과 부교수
〈관심분야〉 클라우드 컴퓨팅, 유비쿼터스 컴퓨팅, 모바일 보안, 빅데이터 보안분석