https://doi.org/10.33778/kcsa.2023.23.2.027

랜섬웨어 공격탐지를 위한 신뢰성 있는 동적 허니팟 파일 생성 시스템 구현

국 경 완*, 류 연 승**, 신 삼 범***

요 약

최근 몇 년 동안 랜섬웨어 공격이 사회 공학, 스피어피싱, 심지어 기계 학습과 같은 전술을 사용하여 특정 개인이나 조직을 대상으로 하는 공격의 정교함과 더불어 더욱 조직화 되고 전문화되고 있으며 일부는 비즈니스 모델로 운영되고 있다. 이를 효과적으로 대응하기 위해 심각한 피해를 입히기 전에 공격을 감지하고 예방할 수 있는 다양한 연구와 솔루션들이 개발되어 운영되고 있다. 특히, 허니팟은 조기 경고 및 고급 보안 감시 도구 역할 뿐만 아니라, IT 시스템 및 네트워크에 대한 공격 위험을 최소화하는 데 사용할 수 있으나, 랜섬웨어가 미끼파일에 우선적으로 접근하지 않은 경우나, 완전히 우회한 경우에는 효과적인 랜섬웨어 대응이 제한되는 단점이 있다. 본 논문에서는 이러한 허니팟을 사용자 환경에 최적화하여 신뢰성 있는 실시간 동적 허니팟 파일을 생성, 공격자가 허니팟을 우회할 가능성을 최소화함으로써 공격자가 허니팟 파일이라는 것을 인지하지 못하도록 하여 탐지율을 높일 수 있도록 하였다. 이를 위해 동적 허니팟 생성을 위한 기본 데이터수집 모델 등 4개의 모델을 설계하고 (기본 데이터 수집 모델 / 사용자 정의 모델 / 표본 통계모델 / 경험치 축적 모델) 구현하여 유효성을 검증하였다.

Implementation of reliable dynamic honeypot file creation system for ransomware attack detection

Kyoung Wan Kug*, Yeon Seung Ryu**, Sam Beom Shin***

ABSTRACT

In recent years, ransomware attacks have become more organized and specialized, with the sophistication of attacks targeting specific individuals or organizations using tactics such as social engineering, spear phishing, and even machine learning, some operating as business models. . In order to effectively respond to this, various researches and solutions are being developed and operated to detect and prevent attacks before they cause serious damage. In particular, honeypots can be used to minimize the risk of attack on IT systems and networks, as well as act as an early warning and advanced security monitoring tool, but in cases where ransomware does not have priority access to the decoy file, or bypasses it completely. has a disadvantage that effective ransomware response is limited. In this paper, this honeypot is optimized for the user environment to create a reliable real-time dynamic honeypot file, minimizing the possibility of an attacker bypassing the honeypot, and increasing the detection rate by preventing the attacker from recognizing that it is a honeypot file. To this end, four models, including a basic data collection model for dynamic honeypot generation, were designed (basic data collection model / user-defined model / sample statistical model / experience accumulation model), and their validity was verified.

Key words : Ransomware, Ransom, Honeypot, Dynamic Honeypot file creation

접수일(2023년 5월 18일), 수정일(1차: 2023년 5월 22일),
게재확정일(2023년 6월 2일)

* 명지대학교 / 보안경영공학과(주저자)
** 명지대학교 / 보안경영공학과(공동저자)
*** 명지대학교 / 보안경영공학과(공동저자)

1. 서 론

랜섬웨어(Ransomware)는 컴퓨터 시스템을 감염시켜 사용자의 민감한 데이터를 암호화(Encryption)하고 데이터를 복호화(Decryption)하는 대가로 몸값(Ransome)을 요구하는 악성코드(Malware) 유형을 말한다. 최근 랜섬웨어의 특징은 더 정교화, 복잡화, 효율화되고 있다는 것이다. 또한, 공격자는 탐지를 피하기 위해 난독화(Obfuscation) 및 회피기술(Avoidance technique)을 사용하며 매일 새로운 랜섬웨어 변종을 생성하고 있다[1, 9].

여기에서 중요한 한가지 사실은 현재까지 확인된 랜섬웨어의 동작 패턴을 분석해 보면, 공격자가 침투에 성공한 이후, 가장 먼저 수행하는 동작은 바로 몸값을 요구하기 위해 암호화할 대상을 찾는다는 것이다. 공격자가 원하는 공격대상에 허니팟 기술을 사용하여 사전에 정보를 획득할 수 있다면 암호화 실행 이전에 사전 초치를 할 수 있어 랜섬웨어 공격을 효과적으로 막을 수 있다.

일반적으로 랜섬웨어 관련 허니팟(Honeypot) 기술은 해커의 침입이 예상되는 특정한 경로에 허니팟 미끼(Decoy) 파일을 배치하고, 랜섬웨어가 미끼파일에 접근하여 암호화, 또는 파일을 변경하려는 순간 랜섬웨어를 차단하는 방법이다. 이렇게 하면 암호화가 시작되기 전에 랜섬웨어 감염을 식별하고 근절할 수 있어 그 영향을 최소화 할 수 있다[2, 14].

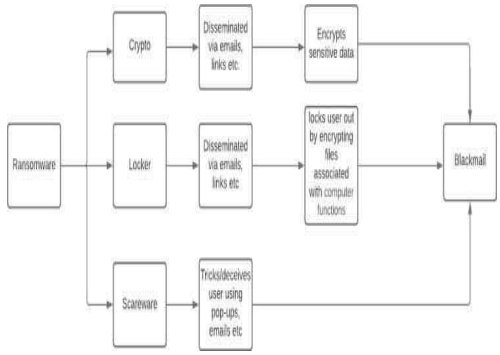
그러나, 이 기술의 단점은 랜섬웨어가 미끼파일에 우선적으로 접근하지 않은 경우나, 완전히 우회한 경우에는 효과적으로 랜섬웨어 대응에 제한되는 단점이 있다. 한편, 해커 입장에서 보면 해당 허니팟을 어떻게든 찾아내어 우회하고, 회피하려고 많은 노력을 투자할 것이며, 우리의 방어 수단보다 한발 앞서 나갈 수 있는 새로운 방법을 지속적으로 개발할 것이다. 이와같이 우회기술, 회피기술 또한 지속적으로 지능화 되고 있어 이에 능동적으로 대응할 수 있는 허니팟 기술이 필요한 시점이다.

본 논문에서는 이러한 허니팟을 사용자 환경에 최적화하여 신뢰성 있는 실시간 동적 허니팟 파일을 생성하여 공격자가 허니팟을 우회할 가능성을 최소화함으로써 공격자가 허니팟 파일이라는 것을 인지하지 못하도록 하여 탐지율을 높이는데 목적이 있다. 이를 위해 동적 허니팟 생성 시스템의 모델을 설계하고 구현하여 유효성을 검증하였다.

2. 이론적 배경

2.1 랜섬웨어 유형

다양한 유형의 랜섬웨어 공격은 간단하게 성가신 것부터 개인 및 기업 운영을 위협하는 것까지 다양하다. 모든 유형의 랜섬웨어에는 랜섬 요구(Ransom demand)라는 공통점이 있다. 랜섬웨어의 새로운 변종은 항상 개발 중이지만 주요 랜섬웨어 유형은 (그림 1)과 같다[3, 10, 15].



(그림 1) 랜섬웨어 유형

공격자들은 계속해서 새로운 유형의 랜섬웨어를 개발하고 있지만 결국, 랜섬웨어는 파일을 암호화하는 방법(Crypto ransomware)과 장치에서 사용자를 잠그는 방법(Locker ransomware)의 두 가지 범주로 분류할 수 있다.

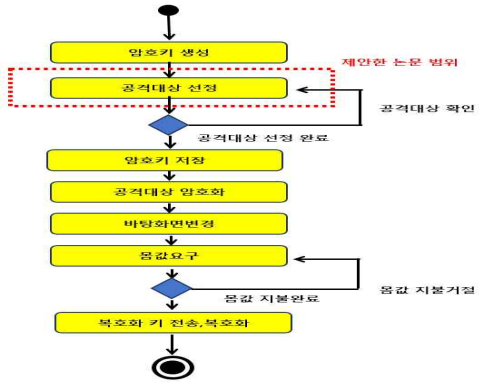
암호화 랜섬웨어 공격에서 공격자는 피해자의 파일을 암호화한 다음 피해자의 컴퓨터에 지불 세부 정보가 포함된 몸값 메모(Ransom note)를 남기는 특징을 가지고 있으며, 이러한 암호화 공격 방법은 랜섬웨어 변종의 90%를 차지하고 있다.

이에 반하여 락커 랜섬웨어나 화면 잠금 랜섬웨어는 공격자와 지속적으로 상호작용할 수 있도록 마우스와 키보드를 제한하면서 필수 기능을 차단해 피해자가 자신의 기기에 접근하는 것을 제한한다.

2.2 랜섬웨어 동작

랜섬웨어 감염의 일반적인 동작순서는 (그림 2)에서 보는 것과 같다. 최근 랜섬웨어는 기존에 사용하던 대칭키 암호화 알고리즘(Symmetric Key Encryption Algorithm)과 다르게 비대칭 암호화 알고리즘(Asymmetric Key Encryption Algorithm)을 사용하여 복호화를 어렵게 만들고 있다.

모든 랜섬웨어는 파일을 암호화하기 전에 반드시 공격대상 파일을 선정한다는 사실이다. 공격대상 선정시에는 일반적으로 파일크기, 시스템 파일 제외, 파일 생성일자, 확장자 등 공격대상 컴퓨터의 주요한 특징을 고려하여 선정한다.



(그림 2) 랜섬웨어 동작순서

위와 같이 랜섬웨어가 처음 공격대상 파일을 선정하는 초기 단계를 탐지할 수 있다면 방어자 입장에서는 그만큼 방어할 수 있는 시간을 확보하여 피해를 최소화할 수 있다. 즉, 랜섬웨어가 심각한 피해를 입히기 전에 감염을 방지하고 랜섬웨어를 중지시키는 것이 랜섬웨어 방어의 중요한 전략이 될 수 있다.

2.3 랜섬웨어 대응

2.3.1 허니팟(Honeypot)을 이용한 대응

컴퓨터 보안 측면에서 허니팟은 공격자를 함정으로 유인하는 것을 말한다. 그것은 미끼(Decoy)와 같은 공격을 유인하기 위한 희생적인 컴퓨터 시스템이라고 볼 수 있다. 통상 허니팟은 공격자의 목표를 모방하고 침입 시도를 사용하여 사이버 범죄자에 대한 정보와 활동 방식을 얻거나 다른 목표로부터 주의를 분산시킬 목적으로 사용된다[4].

이와 같은 이유로 다양한 유형의 허니팟을 사용하여 다양한 유형의 위협을 식별할 수 있는 장점이 있어 최근에 효과적인 사이버 대응 전략으

로 지속적으로 발전시키고 있다. 공격자 입장에서는 이를 허니팟으로 식별하는 경우 허니팟에 접근하지 않고 다른 시스템을 계속 공격할 수 있다는 것이 매력적일 수 있다. 반면, 방어자 입장에서 보면 이러한 허니팟을 인지하지 못하도록 하는게 매우 중요하다.

2.3.2 시그니처(Signature) 기반 대응

시그니처(서명 기반) 탐지는 시스템에서 맬웨어(Malware)의 존재를 식별하는 가장 간단한 방법이다. 맬웨어 서명에는 파일 해시, C&C 인프라의 도메인 이름 및 IP 주소, 맬웨어 샘플을 고유하게 식별할 수 있는 기타 지표와 같은 정보가 포함된다. 시그니처 기반 탐지 시스템은 이러한 시그니처 라이브러리를 저장하고 이를 시스템에 들어가거나 시스템에서 실행 중인 각 파일과 비교하여 맬웨어인지 확인한다[5, 11].

그러나, 패턴이 존재하는 랜섬웨어의 경우 대응 확률이 매우 높지만, 패턴에 반영되지 않은 랜섬웨어나 APT(지능형 지속 공격, Advanced Persistent Threat), 제로데이 공격(Zero-day attack)과 같은 알려지지 않은 신종 랜섬웨어에 대한 대응력은 낮다고 볼 수 있다[6].

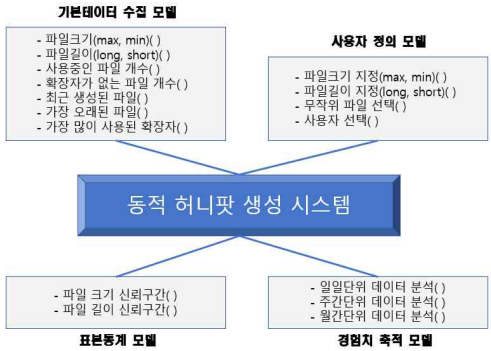
2.3.3 이상행위(Abnormal behavior) 탐지에 의한 대응

랜섬웨어를 탐지하는 표준 방법 중 하나는 비정상적인 활동에 대한 데이터 사용 패턴을 모니터링하는 것이다. 파일 이름이나 확장자 변경, 데이터 전송 또는 권한 업데이트와 같은 비정상적인 활동은 랜섬웨어 공격이 진행 중이라고 판단할 수 있다. 예를 들어, 짧은 기간 내에 이름이 바뀐 수백 개의 파일은 경고를 발생시켜야 한다. 다른 의심스러운 파일 동작으로는 원본 파일(암호화를 나타낼 수 있음)보다 엔트로피(Entropy)가 더 큰 파일의 새 복사본, 운영 체제에서 수행되는 비정상적인 암호화 작업, 의심스러운 파일 확장자 및 의심스러운 파일 리스트가 있다[7, 8].

3. 동적 허니팟 파일 생성 시스템 모델

3.1 모델개요

본 논문에서 설계한 신뢰성 있는 동적 허니팟 생성 시스템 전체 구조는 (그림 3)과 같이 기본 데이터 수집 모델, 사용자 정의 모델, 표본통계 모델, 경험치 축적 모델의 4개 부분으로 구성된다.



(그림 3) 동적 허니팟 생성 시스템 전체구조

3.2 기본데이터 모델

기본 데이터 수집 모델은 (그림 4)와 같이 허니팟이 위치하게 될 폴더의 기본적인 특성, 즉, 해당 폴더의 파일의 크기(max, min), 파일길이(long, short), 특정 파일의 사용 여부를 확인할 수 있는 사용 중인 파일 개수, 확장자가 없는 파일 개수, 가장 최근에 생성된 파일과 가장 오래전에 생성된 파일, 그리고 가장 많이 사용된 확장자를 수집한다.

(그림 4) 기본 데이터 수집 모델 절차도

3.3 표본통계 모델

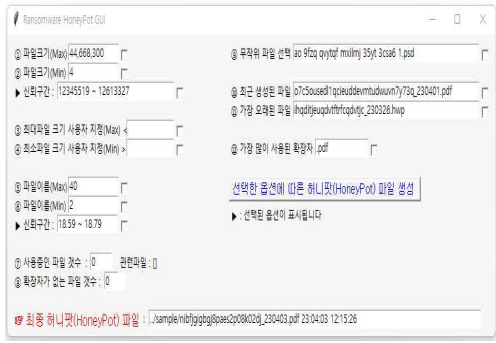
본 논문에서는 공격자가 허니팟이라는 것을 인식하지 못하도록 하여 기존 허니팟이 가지는 우회 가능성을 최소화하도록 하였다. 방어자 입장에서 보면 동적 허니팟 파일, 즉, 매력적인(attractive) 허니팟 파일을 만들어서 공격자가 허니팟을 공격 대상파일에 포함시키도록 하는 것이다. 이를 위해 사용자 컴퓨터의 환경을 분석하여 파일크기와 길이, 확장자 등과 같이 “대푯값(representative value)”을 산출하였다.

사용자는 이러한 대푯값을 활용하여 좀더 매력적인 동적 허니팟 파일을 생성할 수 있다. 이러한 대푯값에는 평균(mean), 중위수(median), 최빈수(mode) 등이 있다. 이 중에서도 평균은 그 집단에서 가장 신뢰할 만한 “대푯값”을 활용할 수 있다. 그러나 이러한 대푯값이나 평균값은 그 집단의 대푯값으로 단정 지울 수 없다. 왜냐하면 평균의 함정이 있기 때문이다.

이를 보완하기 위해 본 논문에서는 표본통계 방법 중 신뢰구간(Confidence Interval:CI)을 활용하는 방법을 제안하였다. 신뢰구간은 모수가 실제로 포함될 것으로 예측되는 범위로 하한(lower limit)과 상한(upper limit)값을 가진다.

3.4 사용자 정의모델

사용자 정의 모델은 허니팟이 위치한 폴더의 단편적인 기본 파일정보를 수집하는 것이 아니라 사용자가 사용자 환경에 최적화되도록 사용자가 정의할 수 있도록 하였다. 사용자 정의는 (그림 5)와 같이 사용자가 선택한 사항을 최우선으로 반영하여 생성되는 동적 허니팟으로 신뢰성을 향상시킬 수 있도록 하였다.



(그림 5) 사용자 정의 모델 GUI 화면

3.5 경험치 축적 모델

지금까지 설명한 기본데이터 수집 모델, 사용자 정의 모델, 표본통계 모델은 실시간 정보를 수집한다. 이러한 실시간 정보 수집은 사용자 컴퓨터 정보를 실시간 반영할 수 있는 장점이 있으나, 변동이 너무 크거나 이와 반대로 너무 변동사항이 없는 사용자 시스템일 경우에는 비 효율적일 수 있다. 이를 보완하기 위해 경험치 축적 모델은 사용자 환경에서 추출 가능한 정보를 추출한 뒤 허니팟 파일의 회피 가능 여부를 판단하기 위해 일정기간(30일) 축적하여 (그림 6)과 같이 변경될 수 있는 동적 데이터 정보를 분류하여 학습을 위한 데이터로 활용할 수 있도록 하였다.

날짜	파일크기 (min,max)	파일크기 신뢰구간	파일길이 (long,short)	파일길이 신뢰구간	사용중인 파일	사용중인 파일	확장자가 없는 파일	가장많이 사용된 확장자	파일생성 일자 (최근오래)
2023-03-26	0 ~ 17920477	849257 ~ 954803	4, 84	27 ~ 29	DailyHoneyPot.csv	aotto1.hwp	honeypot	.hwp	Basic.hwp New.doc

(그림 6) 경험치 축적 테이블 속성

4. 동적 허니팟 파일 생성 시스템 구현

4.1 실험목적

본 논문에서 제안한 신뢰성 있는 동적 허니팟 파일 생성 시스템의 실험 목적은 크게 2가지로 나눌 수 있다. 첫째, 공격자가 다양한 우회기술, 회피기술 등을 사용하여도 제안한 시스템이 유연하게 대처가 가능하다는 것을 실험으로 확인한다. 이를 위해 허니팟 탐지와 성능분석은 최근 쟁점이 되고 있는 11종의 랜섬웨어를 대상으로 실험하였다. 둘째, 제안한 시스템이 기존 허니팟 탐지 기법과 성능을 비교한다. 기존 허니팟 시스템은 파일의 크기 등 단편적인 정보를 가지고 허니팟을 생성하여 특정 폴더에 위치시켜서 활용했으나, 본 논문에서 제안한 동적 허니팟은 사용자 환경에 최적화된 허니팟을 제공함으로써 탐지율과 탐지시간을 측정하여 효율성을 검증하였다.

4.2 실험환경

실험하고자 하는 해당 랜섬웨어는 윈도용이므로 실험 대상 시스템은 <표 1>과 같이 윈도우-11 Enterprise 버전을 사용 하였으며, CPU는 i5-8500, 메모리는 16GB, 저장소는 512GB SSD를 사용하였다. 또한 동적 허니팟 생성을 위해 파이썬 언어와 Visual Studio Code(통합개발환경)를 활용하여 시스템을 구현하였다.

<표 1> 실험 환경

구 분	주요 사양
CPU	Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz
메모리 (RAM)	16GB
운영체제 (OS)	Window 11 Enterprise
개발환경	Visual Studio Code
개발언어 (Language)	Python 3.11

4.3 동적 허니팟 생성 알고리즘

본 논문에서 제안하는 동적 허니팟 생성 알고리즘은 공격자가 허니팟 이라는 것을 인지하지(눈치채지) 못하도록 하여 공격 대상파일 선정의 가능성을 높이고, 탐지 속도를 높이기 위하여 <표 2>와 같은 알고리즘을 제안하였다. 기존 허니팟 파일은 특정 파일이름이나 파일크기 등 단편적인 정보를 기준으로 허니팟을 생성하여 활용함으로써 공격자가 쉽게 허니팟 파일임을 감지하여 허니팟의 고유기능을 제대로 수행할 수 없는 문제점이 발생하였다.

<표 2> 동적 허니팟 생성 알고리즘 의사코드

```
PROCEDURE Dynamic_Honeypot_Pseudocode() :
    Get_info_CI = confidence_interval(file_size, file_length)
                  # 신뢰구간
    Get_info_FU = file_used() # 사용중인 파일
    Get_info_FN    = file_name(random(Count(Char)))
                  # 사용된 문자추출, 무작위로 조합
    Get_info_EXT = last_file_extension()
                  # 가장많이 사용한 확장자
    Get_info_DT = latest_file() # 최근에 생성된 날짜
    IF NOT(Get_info_FU) THEN
        honeypot_filename = Get_info_CI(File_length) +
        Get_info_FN + Get_info_EXT + Get_info_DT
    IF honeypot_filename.getsize() == Get_info_CI(file_size) THEN
        CREATE FILE(honeypot_filename)
END PRECEDURE
```

본 논문에서는 사용자 환경에 최적화되도록 파일크기와 파일길이에 대한 표본통계, 파일이름, 생성일자, 가장 많이 사용한 확장자, 현재 사용중인 파일 제외 등의 속성을 활용하여 최적의 동적 허니팟 파일을 생성한 후 실제 랜섬웨어를 실행하여 기존 허니팟이 가지는 단점을 보완토록 하였다.

4.4 실험에 사용된 파일과 랜섬웨어

실험에 사용될 파일을 생성하기 위해 확장자

는 이미지, 비디오, 문서, 사운드, 실행파일 및 프로그램, 게임파일, 압축파일, 데이터베이스 파일 등 다양한 형식으로 파일크기가 50 Mbyte 보다는 작도록 난수에 의해 500개의 실험파일을 생성하여 실제 폴더에 위치시켰다. 위와같이 다양한 파일 및 확장자, 파일크기 등을 갖는 파일을 실험 데이터로 활용하는 것은 최근 지능화되고 발전되고 있는 램섬웨어에 대해서도 공격 대상 파일를 선정하는 방법에 대하여 본 논문에서 제안한 시스템이 얼마나 잘 반영되는지를 확인하기 위해서이다.

본 논문에서 제안한 동적 허니팟 생성 시스템의 성능시험을 위해 11개 랜섬웨어를 선정하여 실험하였으며 주요 특징은 <표 3>과 같다[12, 13]. 선정기준은 전 세계적으로 매우 큰 피해를 일으켜 국내에서 최초로 대중에게 랜섬웨어를 알게 만든 멀웨어나, 다소 유명하지는 않지만 최근 피해 사례가 있는 랜섬웨어, 그리고 소스가 공개된 랜섬웨어, 파급 효과가 크고, 지금도 변형이 나오고 있는 랜섬웨어를 선정하였다.

<표 3> 실험에 사용된 랜섬웨어

관련 랜섬웨어	방 법	공격 선정대상 고려요소
SLoker, CryptoWire	파일 크기	공격 대상파일의 지정된 파일크기는 제외
SLoker, Ako, DeathRansom, MarkRansom, Cryptolocker, Locky, GonnaCry, HolyCrypt, WannaCry	확장자	랜섬웨어가 특정 확장자만을 선택
Ako, Kevin, CryptoWire	특정 폴더	윈도우 시스템 폴더, OS관련 파일은 제외

4.5 실험결과

실험은 정확도를 향상시키기 위하여 실험 준비과정에서 생성한 500개의 실험파일을 대상으로 하였다. 실험은 본 논문에서 제안한 기본데이터 수집 모델, 사용자 정의 모델, 표본통계 모델, 경험치 축적 모델의 주요 특성을 검증할 수 있도록 <표 4>과 같이 랜섬웨어별로 파일크기, 파일이름, 확장자, 무작위 파일, 표본통계, 경험치 축척 등에 관련된 실험을 진행하였다.

<표 4> 랜섬웨어 실행 의사코드

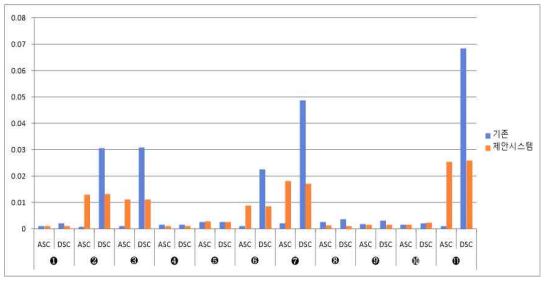
```
PRCEDURE Ransomware_detect_Pseudocode() :
    Honeypot = [ old_honepot_file, new_honepot_file ]
        # 기존방식 허니팟 파일과 개선된 허니팟 파일
    Ransom = [Ako, Kevin, WannaCry, HolyCrypt, GonnaCry,
              Locky, CryptoWire, Cryptolocker, MarkRansom,
              DeathRansom, SLoker]
        # 실행하고자 하는 랜섬웨어 파일
    FOR i IN Honeypot :
            # 기존 허니팟과 동적 허니팟을 인수로 받아 실행
        start_time ← NOW()
        skip_n, pos_n ← Ransom(Honeypot)
            # 회피건수, 탐색위치
        end_time ← NOW()
        elapsed = end_time - start_time
        IF i == Honeypot[0] :   # 기존과 제안 시스템 구분
            WRITE([Ransom(i) , detect_n, elapsed, '기존'])
        ELSE
            WRITE([Ransom(i) , detect_n, elapsed, '제안 시스템'])
END PRECEDURE
```

결과치 측정은 기존 방식의 허니팟과 본 논문에서 제안한 시스템에서 생성한 동적 허니팟의 결과를 비교하였다. 본 논문에서 생성한 동적 허니팟은 기존 방식의 단편적인 허니팟이 아닌 파일크기, 파일이름, 확장자 등을 고려하여 사용자 시스템에 최적화되도록 하여 공격자가 허니팟이라고 인지하지 못하도록 하였다.

(그림 7)과 (그림 8)은 파일이름을 기준으로 하여 허니팟으로 활용했을 때와 본 시스템에서 제안한 동적 허니팟 파일을 사용하여 탐지시간을 비교하여 실행한 결과를 보여주고 있다. 이 실행 결과에서 보는 것과 같이 사용자가 해당 시스템에 최적화된 특성을 반영한 동적 허니팟이 탐색시간이 빠르다는 것을 알 수 있었다. 그러나, 일부 랜섬웨어에서는 탐색 폴더 제외, 정렬, 회피 파일을 찾는 알고리즘이 적용되어 제안한 시스템보다 탐지시간이 더 빠른 경우도 발생하였는데, 이는 기본적으로 오름차순으로 대상을 선정하기 때문에 본 시스템에서 생성한 동적 허니팟 파일보다 먼저 대상으로 선택되기 때문으로 확인되었다.

구 분		기존	제안시스템
❶Ako	ASC	0.0012	0.00115
	DSC	0.00202	0.00114
❷Kevin	ASC	0.00092	0.01306
	DSC	0.03069	0.01321
❸WannaCry	ASC	0.00098	0.01114
	DSC	0.03083	0.01126
❹HolyCrypt	ASC	0.00165	0.00113
	DSC	0.00153	0.0011
❺GonnaCry	ASC	0.00255	0.00302
	DSC	0.00267	0.00273
❻Locky	ASC	0.00114	0.00886
	DSC	0.02259	0.00852
❼CryptoWire	ASC	0.00202	0.01827
	DSC	0.04881	0.01714
❽Cryptolocker	ASC	0.00259	0.00134
	DSC	0.00369	0.00118
❾MarkRansom	ASC	0.00185	0.00166
	DSC	0.00312	0.00163
❿DeathRansom	ASC	0.00162	0.00158
	DSC	0.00225	0.00231
⓫SLoker	ASC	0.00116	0.02555
	DSC	0.06856	0.02594

(그림 7) 파일이름 기준 탐지시간 측정



(그림 8) 파일이름에 따른 탐지시간 비교(예)

본 연구에서는 이와같은 실험방법을 활용하여 파일크기, 파일이름, 임의파일 선정, 다수 확장자, 파일 생성시간, 파일크기와 이름에 대한 신뢰구간, 사용자 정의, 경험치 누적 등의 방법을 통해 실험을 진행하였다.

본 연구를 통해 실험한 동적 허니팟 생성 시스템에 대한 실험결과를 <표 5>로 요약하였다. 시험 결과를 요약해 보면 탐지율은 82~100% 수준으로 대부분의 랜섬웨어를 탐지할 수 있는 것으로 확인되었다. 탐지시간은 기본방식 대비 전체적으로 향상된 것으로 확인하였다. 그러나 일부 랜섬웨어의 경우에는 탐지가 불가능하거나 탐지시간이 더 소요되는 경우도 발생하였다.

<표 5> 실험결과 요약

모델	실험대상	동적 허니팟 파일
기본데이터 수집모델	파일크기	·탐지율 : 82% ·탐지시간: 기본방식대비 전체적으로 향상 ·탐지제한: GonnaCry, DeathRansom
	파일이름	·탐지율 : 82% ·탐지시간: 오름차순시 향상 ·탐지제한: GonnaCry, DeathRansom
	임의파일 파일	·탐지율 : 91% ·탐지시간: 편차가 큼 ·탐지제한: GonnaCry
	다수 확장자	·탐지율 : 100% ·탐지시간: 편차가 큼 ·탐지제한: 없음, * 예외조건에 해당시 제한
	파일 생성시간	·탐지율 : 82% ·탐지시간: 편차가 큼 ·탐지제한: 없음 * 파일생성일자 영향 없음
사용자정의 모델	사용자 정의	·탐지율 : 100% ·탐지시간: 편차가 큼 ·탐지제한: 없음 * 사용자 시스템 반영 중요
표본통계 모델	표본통계	·탐지율 : 100% ·파일크기와 길이 산출 ·파일 길이 보다는 크기가 영향을 미침
경험치축적 모델	경험치 축적	·탐지율 : 100% ·일일단위보다 월간단위가 누적치가 탐색시간이 빠름

5. 결 론

본 연구에서는 사용자 환경에 최적화된 동적 허니팟 생성 시스템을 구현하여 랜섬웨어 공격자가 공격대상 파일을 선정시, 최단 기간내에 허니팟에 접근시켜 탐지시간을 최소화할 수 있도록 기본 데이터수집 모델 등 4개의 모델을 설계하고 시스템을 구현하여 실제 실험을 통해 제안한 방법이 우수하다는 것을 검증하였다.

첫째, 기본 데이터 수집 모델에서는 파일크기, 파일이름, 임의파일, 다수 확장자, 파일 생성일자 속성을 테스트하였다. 실험결과 탐지률은 82~100% 수준 이었으며, 기존 허니팟 파일을 활용시와 비교시 전체적으로 탐지시간이 짧았다. 그러나 임의파일, 다수 확장자, 파일 생성시간 등은 탐지시간 편차가 큰 것을 발견하였다. 또한 일부 속성은 영향이 미비한 것으로 확인되었다.

둘째, 사용자 정의 모델에서는 사용자가 정의한 항목에 우선권을 부여하여 동적 허니팟을 생성하여 실험한 결과, 탐지율은 100%였으며, 탐지시간은 편차가 큰 것으로 나왔다. 특히, 실험을 통하여 사용자 정의 모델에서는 사용자 시스템 환경을 반영한 속성을 정의하는 것이 무엇보다도 중요한 것으로 확인하였다.

셋째, 표본 통계모델에서는 신뢰구간 값을 활용하여 파일크기와 길이를 산출하여 동적 허니팟 파일을 생성 후 실험한 결과, 파일 길이 보다는 크기가 영향을 미치는 것으로 확인하였다.

넷째, 경험치 축적 모델에서는 1개월간 축적된 파일크기 데이터를 활용하여 실험을 진행하였으며, 일일단위보다 월간단위 누적자료가 탐색시간이 빠른것으로 확인하였다.

결론적으로 실험결과를 통해 파일크기, 파일이름, 확장자가 공격대상 선정시 중요한 요소로 작용되는 것을 알 수 있었다. 파일크기의 경우에는 특정범위에 있는 파일만 선택하고 있으며, 파일이름은 기본적으로 오름차순으로 대상을 선정하기 때문에 “0”, “a”, “가” 로 시작하는 파일명이 유리하다. 확장자는 대부분 문서작성 관련 파일이 우선순위가 높았으며, 시스템 운용 관련 파일은 우선순위가 낮음을 확인하였다. 또한 시험결과를 토대로 기존 허니팟과 비교하였을시 제안한 시스템은 일부 랜섬웨어를 제외하고는 대부분 공격 대상파일에 선정되었으며, 탐지시간도 비교적 제안한 시스템이 최단시간에 탐지되는 것을 확인할 수 있었다.

향후 연구는 더욱 다양하고 고도화된 랜섬웨어를 활용하여 본 논문에서 제안한 알고리즘을 고도화하는 것이다. 특히, 최근 각광을 받고 있는 딥러닝 기술을 이용하여 공격대상 선정 알고리즘을 학습을 통해 강화한다면, 향후 지능화· 고도화되는 랜섬웨어 공격대상 선정시 어떠한 공격기법에도 탄력적으로 대응할 수 있을 것이다.

또한, 향후 미래의 동적 허니팟 기술은 플러그 앤 플레이 기술로 발전해야 된다는 것이다. 플러그를 꼽기만 하면 바로 동작하듯이 허니팟이 배포할 허니팟 수, 배포 방법 및 환경과 조화를 이루는 모습을 자동으로 결정하고, 배포된 허니팟이 환경에 맞게 변경되고 적응되어야 한다는 것이다. 즉, 네트워크에 연결하기만 하면 환경을 학습하고 허니팟의 적절한 수와 구성을 배포하고 네트워크의 모든 변경 사항에 적응하는 솔루션인 어플라이언스(Appliance)로 발전해야 할 것이다.

참고문헌

[1] Moore, C., “Detecting ransomware with honeypot techniques”, Cybersecurity and Cyberforensics Conference (CCC) IEEE, pp. 77 - 81, 2016.

[2] Farouk, S.,, “Design and Implementation of a Real-Time Honeypot System for the Detection and Prevention of Systems Attacks”, Culminating Projects in Information Assurance, 2016.

[3] Andronio, N., Zanero S., Maggi F., “Dissecting and detecting mobile ransomware” Springer-Verlag, Berlin, Heidelberg Heldroid, pp. 382 - 404, 2015.

[4] Farouk, S., “Design and Implementation of a Real-Time Honeypot System for the Detection and Prevention of Systems Attacks”, Culminating Projects in Information Assurance, pp. 41 - 48, 2016.

[5] Devin, P., “How to Detect Ransomware Using 5 Techniques”, https://www.enterprisenetworkingplanet.com/security/ransomware-detection/, 2023.

[6] 박세균.,「Endpoint Level의 효과적인 APT 공격 대응 방안 연구」, 석사학위논문, 서울: 고려대학교 컴퓨터정보통신대학원, pp. 3 - 6, 2015.

[7] Devin, P., “How to Detect Ransomware Using 5 Techniques”, https://www.enterprisenetworkingplanet.com/security/ransomware-detection/, 2023.

[8] Bill, C., “Top 5 ransomware detection techniques: Pros and cons of each”, www.malwarebytes.com, 2022.

[9] Craig, B., Ashley, B., Toluwalope, D., Saqib, H., & Muhammad, K. K., “Ransomware: Recent advances, analysis, challenges and future research directions”, Elsevier Public Health Emergency Collection, 2021.

[10] Stockman, M., Rein, R., & Heile, A., An open-source honeynet system to study system banner message effects on hackers. Journal of Computing Sciences in Colleges, pp. 282-293., 2015.

[11] Drew, R. (2022), “Techniques for Ransomware Detection”, https://www.cioinsight.com/security/ransomware-detection(검색일: 2023.02.21.).

[12] Jonathan, E.(2023), “The Top Ransomware Trends to Watch Out for in 2023”, https://www.reliaquest.com/blog/ransomware-trends-2023/(검색일: 2023.03.21.).

[13] malwarebytes, “Ransomware”, https://www.malwarebytes.com/ransomware(검색일 : 2023.03.17.).

[14] 이후기, 성종혁, 김유천, 김종배, 김광용, "랜섬웨어 분석 및 탐지패턴 자동화 모델에 관한 연구", 한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 8, pp. 1581 - 1587, 2021.

[15] 이진우, 김용민, 이정환, 홍지만., "랜섬웨어 탐지를 위한 효율적인 미끼 파일 배치 방법", 스마트미디어 저널, Vol.8, No.1, pp. 28 - 33. 2019.

[저 자 소 개]

국 경 완(Kyoungwan Kug)
1992년 2월 금오공과대학 공학사
2002년 1월 국방대학교 공학석사
2021～현재 명지대학교 박사과정
2017～현재 국방통합데이터센터 경영혁신실장
email : kugstone@naver.com

류 연 승 (Yeonseung Ryu)
1990년 2월 서울대학교 공학사
1992년 2월 서울대학교 공학석사
1996년 8월 서울대학교 공학박사
2003～현재 명지대학교 주임교수
email : ysryu@mju.ac.kr

신 삼 범 (Sambeom Shin)
1996년 2월 동국대학교 경영정보학석사
2008년 2월 부경대학교 공학박사
2020년 9월 국방통합데이터센터장
2021년～현재 명지대학교 특임교수
email : ktma3419@mju.ac.kr