Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.5, Oct. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.5.721

명령 제어 프레임워크 (Command and Control Framework) 도구 추적 방안에 대한 연구*

권 혁 주,[1†] 곽 진[2‡]
[1,2]아주대학교 (대학원생, 교수)

A Study on Tracking Method for Command and Control Framework Tools*

Hyeok-Ju Gwon,[1†] Jin Kwak[2‡]
[1,2]Ajou University (Graduate student, Professor)

요 약

명령 제어 프레임워크 (Command and Control Framework)는 모의 침투 및 교육용으로 개발되었으나 사이버 범죄 그룹과 같은 위협 행위자들이 악용하고 있다. 잠재적인 위협을 식별하고 선제 대응을 하는 사이버 위협 헌팅 관점에 따라 명령 제어 프레임워크가 작동하고 있는 서버를 식별하고 사전 차단하면 위험 요소 관리에 기여를 할 수 있다. 따라서, 이 논문에서는 명령 제어 프레임워크를 사전 추적할 수 있는 방법론을 제안한다. 방법론은 명령 제어 프레임워크 관련 서버 목록 수집, 단계적 전달 모사, 봇넷 설정 추출, 인증서 수집 및 특징 추출 4단계로 구성된다. 또한, 상용 명령 제어 프레임워크인 코발트 스트라이크에 제안한 방법론을 적용하여 실험을 수행한다. 실험에서 수집된 비콘, 인증서에 대한 분석 내용을 공유함으로써 명령 제어 프레임워크로부터 발생할 수 있는 사이버 위협 대응 기반을 마련하고자 한다.

ABSTRACT

The Command and Control Framework was developed for penetration testing and education purposes, but threat actors such as cybercrime groups are abusing it. From a cyber threat hunting perspective, identifying Command and Control Framework servers as well as proactive responding such as blocking the server can contribute to risk management. Therefore, this paper proposes a methodology for tracking the Command and Control Framework in advance. The methodology consists of four steps: collecting a list of Command and Control Framework-related server, emulating staged delivery, extracting botnet configurations, and collecting certificates that feature is going to be extracted. Additionally, experiments are conducted by applying the proposed methodology to Cobalt Strike, a commercial Command and Control Framework. Collected beacons and certificate from the experiments are shared to establish a cyber threat response basis that could be caused from the Command and Control Framework.

Keywords: Command and Control Framework, Cyber Threat Hunting, Shodan, Cobalt Strike

I. 서 론

모의 침투는 조직이 사용하는 네트워크 인프라를 의도적으로 공격하고 내부에 침투하여 잠재적인 보안 취약점 발견 및 피해 영향도를 측정하는 일련의 과정으로 보안성 평가의 일종이다.

Received(05. 02. 2023), Modified(07. 28. 2023), Accepted(08. 21. 2023)
* 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-01806), 스마트공장 보안 내재화 및 보안관리 기술 개발)
† 주저자, win32virus@ajou.ac.kr
‡ 교신저자, security@ajou.ac.kr(Corresponding author)

초기 침투를 위해, 네트워크상 공개되어 있고 해당 조직이 사용하는 소프트웨어의 취약점을 점검하거나 임직원 대상으로 피싱 이메일을 전송하는 식으로 진행된다. 이때, 모의 침투를 수행하는 인원을 통상적으로 레드 팀이라고 부른다[1].

초기 침투에 성공하면 레드 팀은 자체 개발했거나 오픈소스 또는 상용 명령 제어 프레임워크를 사용하여 네트워크 내 전파를 수행한다.

명령 제어 프레임워크 (Command and Control Framework) 는 실제 악성코드의 기능이 구현되어 공격자가 원격에서 감염된 기기를 제어할 수 있도록 하는 소프트웨어로 명령을 내리는 서버와 감염 수행 및 서버의 명령을 수신해 실행하는 봇넷으로 구성된다. 이 도구는 교육 목적과 모의 침투를 위해 개발되었다[2]. 오픈소스 명령 제어 프레임워크는 소스코드가 공개되어 있어 누구나 사용할 수 있고 상용 명령 제어 프레임워크는 크랙되어 다크웹 포럼과 같은 커뮤니티에서 공유되고 있다. 사이버 범죄 그룹인 Conti 랜섬웨어 그룹과 DarkSide 랜섬웨어 그룹이 상용 명령 제어 프레임워크인 코발트 스트라이크를 사용한다고도 알려졌다[3][12]. 즉, 사이버 위협 행위자 (Threat Actor)들이 이러한 명령 제어 프레임워크 도구를 악용하는 것은 흔한 양상으로 볼 수 있다. 시중에 알려져 있는 명령 제어 프레임워크의 종류는 다양하나 탐지 건수에 따라 위협 행위자들이 어떠한 도구를 선호하는지 대략적으로 파악 가능하다.

레코디드 퓨처가 2022년 발간한 2022 Adversary Infrastructure Report에 따르면 2020년부터 2022년 까지 탐지된 명령 제어 프레임워크는 코발트 스트라이크, 메타스플로잇, Covenant, Koadic, Mythic, PoshC2, Empire, Pupy, Sliver과 같은 도구들이 상위 10위를 차지했다. 특히, 2012년에 출시된 상용 명령 제어 프레임워크인 코발트 스트라이크는 2020년에 1441건, 2021년 3691건, 2022년 7480건으로 3년간 1위를 차지하였다. 2위를 차지한 메타스플로잇이 2020년에 1381건, 2021년 1441건, 2022년 1495건으로 탐지 증가 폭 또한 다른 도구보다 월등히 높기 때문에 위협 행위자들의 코발트 스트라이크에 대한 선호도가 높다고 볼 수 있다[6].

잠재적인 위협에 대한 사전 탐지 및 대응을 하는 사이버 위협 헌팅 (Cyber Threat Hunting) [5] 관점에서는 이러한 명령 제어 프레임워크가 작동하고 있는 서버들을 사전에 추적하여 목록을 수집하고 위협이 발생하기 전에 차단하는 식으로 조치가 가능할 것이다. 관련되어 기 진행된 연구들은 특정한 명령 제어 프레임워크에서 발생되는 네트워크 트래픽 탐지 중심적이거나 C2 서버 호스트에 대한 핑거프린팅 방법에 대한 연구가 주를 이뤘다[2][4][19].

명령 제어 프레임워크는 크게 제어 서버, 봇넷, 핸들러로 이뤄져 있는데, 네트워크 트래픽 탐지 관점에서의 연구는 제어 서버와 핸들러를 담당한다고 볼 수 있으나 명령 제어 프레임워크의 실제적인 악성코드 파일 부분을 담당하는 봇넷에서 설정을 추출하는 부분을 다루는 내용의 연구를 찾아보기 힘든 실정이다. 대부분의 악성코드는 기능 수행을 위해 설정을 어떠한 형태로든 내부에 보관하고 있기 때문에 이를 추출한다면 추가적인 정보를 얻을 수 있다.

따라서, 이 논문에서는 기존의 네트워크 트래픽 관점에 봇넷 확보 및 설정 추출 과정까지 포함시킨 명령 제어 프레임워크 추적 방법론을 제안한다. 제안하는 방법론은 명령 제어 프레임워크 관련 서버 목록 수집, 단계적 전달 모사, 봇넷 설정 추출, 인증서 수집 및 특징 추출의 4단계로 구성된다.

서버 목록 수집 단계에서는 특정한 스캐닝 방법을 통해 추적하고자 하는 명령 제어 프레임워크 서버 목록을 수집한다. 스캐닝 방법은 외부 스캐너를 이용하거나 직접 스캐너를 개발하는 방법을 사용할 수 있다. 세부적인 스캔 방법에 따라 수집된 서버 목록이 수집하고자 하는 명령 제어 프레임워크 서버에 해당하지 않게 되는 오탐 문제가 발생할 수 있다.

이에 대한 검증 차원 및 악성코드 확보차원으로 단계적 전달 모사를 수행한다. 단계적 전달은 수집된 서버로부터 명령 제어 프레임워크의 악성코드 부분인 봇넷을 다운로드하는 단계이다. 명령 제어 프레임워크는 대부분 단계적 전달 기능을 구현하고 있기에 이 과정으로 자연스럽게 수집된 서버에 대한 검증 작업도 수행할 수 있게 된다.

봇넷 설정 추출 단계에서는 확보한 봇넷에서 추가적인 제어 서버 주소나 악성코드 식별 값 등에 해당하는 설정을 추출하여 추가 정보를 확보하는 단계이다. 인증서 수집 및 특징 추출 단계는 기존의 단계적 전달을 거쳐 검증이 완료된 서버가 TLS 통신 서버일 경우에 인증서 관련 정보를 수집하는 단계이다.

제안하는 방법론이 효용성이 있는지에 대한 평가를 위하여 위협 행위자들에게 선호도가 제일 높다고 알려진 코발트 스트라이크를 대상으로 제안하는 방법론을 적용한 실험을 수행한다. 더불어 코발트 스트라

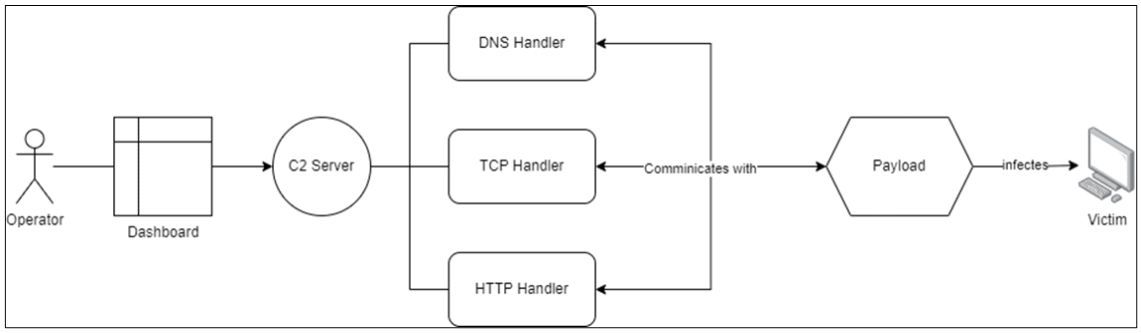

Fig. 1. Command and Control Framework Components

이크 추적 실험으로부터 획득한 분석 내용을 공유하여 악용된 명령 제어 프레임워크로부터 발생할 수 있는 사이버 위협에 대한 대응 기반을 마련하고자 한다.

2장에서는 이 연구와 유사한 관련 연구들을 살펴보고 3장에서는 이 연구에서 다루는 코발트 스트라이크를 비롯해 관련되어 필요한 배경 사항을 설명한다. 4장에서는 명령 제어 프레임워크 추적 방법론에 대하여 상세한 설명을 하고 이를 코발트 스트라이크 도구에 적용하여 수행한 실험 결과를 서술한다. 5장에서는 결론 및 향후 연구 방향으로 마무리를 맺는다.

II. 관련 연구

2.1 오픈 소스 명령 제어 프레임워크에 관한 연구

오픈 소스 기반의 명령 제어 프레임워크에 대한 연구가 진행된 바 있다[2]. Metasploit, Empire, Pupy 3개의 오픈 소스 명령 제어 프레임워크 도구를 대상으로 수행하였으며 기업 네트워크에서 확보한 정상 트래픽과 실험 환경에서 발생시킨 명령 제어 프레임워크들의 트래픽으로 연구를 진행하였다.

Passive Detection Model 과 Targeted Scanning 관점에서 관측되는 특징을 추출하고 이를 기계 학습을 통해 임의의 트래픽에서 명령 제어 프레임워크 도구가 발생시키는 이상 트래픽을 탐지할 수 있도록 하였다. 이를 통해 약 98.5%의 탐지율과 0.01% 의 오탐율을 보인다고 언급했다.

Passive Detection Model 은 네트워크 트래픽 상에서 보이는 지표들을 관찰했다면 Targeted Scanning 은 명령 서버임이 확인된 서버들에 한해 임의로 구성된 패킷을 전송하여 서버에서 응답한 내용을 특징으로 구성하였다. 가령, Empire 는 RC4 알고리즘을 이용한 암호화 통신을 사용하는데, Bit Flipping을 통해 서버에서 잘못된 처리를 하도록 유도하고 이에 따른 결과를 수집하여 연구를 진행했다.

해당 연구는 실제 공격자들이 다수 사용하는 명령 제어 프레임워크에 대한 연구가 학술적으로 연구되었다는 점과 네트워크 트래픽 상에서 보이는 시그니처들로만 특징을 구성한 것이 아니라 임의 구성된 패킷의 서버 응답 값도 포함 시켜 그 신뢰도를 높였다는 점으로 의미 있는 연구라고 평가할 수 있다.

2.2 코발트 스트라이크 탐지 연구

본 연구에서 실험 대상으로 선정한 상용 명령 제어 프레임워크인 코발트 스트라이크 탐지 연구가 수행되었다[4].

실험 환경에서 코발트 스트라이크 4.0 버전의 트래픽을 발생시키고 발생한 트래픽을 네트워크 트래픽 모니터링 시스템인 NetFlow을 이용한 탐지 알고리즘을 고안하여 탐지하였다. 기계 학습을 수행하지 않고도 99.996%의 탐지율을 보였다고 연구에서 전했다. 해당 연구는 상용 명령 제어 프레임워크인 코발트 스트라이크에 관해 학술적으로 연구했다는 점에 의의가 있다.

앞서 언급한 두 연구의 공통점은 명령 제어 프레임워크 서버를 실험 환경에서 구축하여 진행되었고 이상 트래픽 탐지에 초점을 맞췄다는 것이다.

실험 환경과 실제 공격자들이 사용하는 환경과의 괴리가 있을 가능성이 있기에 연구 결과에 대한 실용성에 대해서는 조심스럽게 접근할 필요가 있다. 또한, 네트워크 탐지 중점적이기에 악성코드 분석으로 얻을 수 있는 특징들에 대한 접근이 제한적이라는 점도 있다.

2.3 C2 서버에 대한 스캐닝 및 호스트 핑거프린팅

Random Forest, Host Distance, JARM hash, Self-Signed TLS 인증서, TLS 인증서 공개키와 같은 지표로 핑거프린팅을 수행한 연구가 진행되었다. 대상 악성코드는 코발트 스트라이크, Dridex, TrickBot, Ave Maria, Mirai, Remcos 이다.

해당 연구는 C2 서버 탐지에 중점을 맞춘 연구로 Random Forest를 통해 Dridex, Trickbot 서에 대해 높은 탐지율을 보였고 Host Distance를 통해 Trickbot, Mirai 서버에 대해 높은 탐지율을 보였다. 그러나 코발트 스트라이크 서버 탐지에는 Random Forest 가 약 50%, Host Distance 는 대략 70% 의 탐지율을 보였다. 연구에 따르면 몇몇 코발트 스트라이크 서버가 CloudFlare 와 같은 CDN을 이용하여 도메인 프론팅을 사용하기 때문에 탐지에 어려움이 있다고 언급하였다.

해당 연구는 네트워크 트래픽 관점 중점이기에 이 연구에서 다루고자 하는 JARM hash, TLS 인증서 관련 내용을 일부 다루고는 있으나 실제 다운로드된 봇넷으로부터 설정을 추출하는 관점이 포함되어 있지 않다.

따라서 본 연구에서는 이 접근들과는 반대로 네트워크 스캔을 통해 실제 열려진 명령 제어 프레임워크 서버를 찾고 악성코드 확보를 통해 네트워크 탐지 뿐 아니라 다운로드된 악성코드로부터 추가 정보를 획득하는 방향까지 연계할 수 있는 방법론을 제시하고자 한다.

2.4 코발트 스트라이크 비콘 팀서버 식별

팔로알토 네트웍스의 Unit42 팀이 작성한 분석 글에서도 본 연구에서 다룰 대부분의 코발트 스트라이크 관련 배경 지식을 설명하고 있다. 코발트 스트라이크의 단계적 전달과 http-post-uri 에 대해 다루고 있으며 JARM을 이용한 ITW (In-the-wild) 에서의 코발트 스트라이크 팀 서버를 발견하는 방법에 대한 내용도 등장한다[21].

그러나, 글에서는 실제 코발트 스트라이크를 사용하는 공격자들이 어떻게 서버 세팅을 하고 있는지와 같은 관측 내용이 포함되어 있지 않다. 진행할 연구에서는 Unit42 의 분석 글에서 다루고 있지 않은 코발트 스트라이크 팀 서버의 인증서 관련 분석 내용도 포함하며 실제 공격자들이 코발트 스트라이크를 사용할 때 어떤 설정을 하고 사용하는지에 대한 관측 내용도 다룬다.

III. 배 경

3.1 명령 제어 프레임워크

명령 제어 프레임워크는 모의 침투 및 레드팀을 위한 도구로 레드팀 도구, 오펜시브 도구 등으로도 불린다. 상용 명령 제어 프레임워크와 오픈소스 명령 제어 프레임워크가 존재하며 상용 명령 제어 프레임워크의 경우, 코발트 스트라이크가 대표적이고 오픈소스의 경우 Covenant, Koadic, Metasploit, Empire, Pupy, Mythic, PoshC2, Sliver 등이 존재한다[2][6].

명령 제어 프레임워크의 공통 요소는 제어 서버와 봇넷, 핸들러로 이루어져 있다. 제어 서버는 봇넷에게 명령을 내리는 역할을 하며 모의 침투를 진행하는 사용자에게 봇넷 감염 상황과 같은 전체적인 사항들을 별도의 클라이언트나 대시보드, 패널 등으로 제공한다.

봇넷은 실제적으로 감염을 시키는 역할을 하며 자격증명 탈취, 화면 캡처, 파일 탈취 등 악성 행위들이 구현되어 있다. 봇넷은 코발트 스트라이크의 경우 비콘이라고 부르며 Covenant 는 Grunt, Mythic 은 Agent 라고 칭하는 등 명령 제어 프레임워크마다 봇넷을 칭하는 이름들이 다르다.

핸들러는 제어 서버에서 등록되어 봇넷이 통신하는 구성요소로 리스너(Listener) 라고도 불린다. 이 요소의 역할은 HTTP, DNS, SMB 등의 다양한 통신 프로토콜을 지원하는 것이다. Fig 1.은 위에서 설명된 명령 제어 프레임워크의 전체적인 구조를 나타낸다.

3.1.1 단계적 전달

단계적 전달 (Staged Delivery) 은 봇넷을 한 번에 전달하는 것이 아닌, 단계별로 전달하는 기능을 말한다. 이는 네트워크 탐지를 우회하고 봇넷의 기능을 모듈화 하여 성능적으로도 이점이 있다. 명령 제어 프레임워크의 구현마다 이 기능을 칭하는 이름은 각기 다르다. 코발트 스트라이크에서는 해당 기능을 Stager 라고 부르고 한 번에 봇넷을 전달하는 기능

을 Stageless 라고 부른다. Fig 2. 는 이러한 단계적 전달을 설명한 그림이다.

명령 제어 프레임워크마다 상세 구현이 다르지만 일반적으로는 로더가 핸들러에 다운로드 요청을 하고 메인 봇넷을 다운로드 받아 이를 실행하는 구조로 구현되어 있다. Fig 3.에서 이러한 구조를 그림으로 설명한다. 이 논문에서는 로더가 핸들러 서버에 메인 봇넷을 다운로드하는 동작을 분석하고 이를 모사하여 명령 제어 프레임워크의 메인 봇넷을 확보할 예정이다.

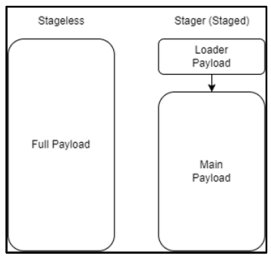

Fig. 2. Staged Delivery

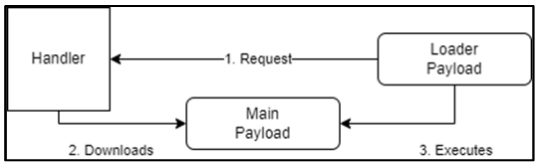

Fig. 3. Request to Handler and Downloads Main Payload

3.1.2 TLS 지원

명령 제어 프레임워크에서 HTTPS 핸들러를 등록하는 경우 봇넷과 핸들러 통신에 TLS 통신을 사용한다. 이 연구에서 TLS 통신에서 사용되는 인증서로부터 추출할 수 있는 값들을 명령 제어 프레임워크 추적에 활용할 것이기에 관련 내용이 설명될 필요가 있다.

TLS 통신은 초기에 TLS Handshake 과정을 수행한다. 이는 클라이언트와 서버가 암호화 통신 알고리즘, 인증 방법 등을 협상하는 과정이다. 이 과정에서 클라이언트와 서버가 서로의 인증서를 주고받으며 RFC 8446에서 해당 과정에 대해 상세히 설명하고 있다[9]. 인증서를 사용하기 위해 X.509 규격을 따른다. X.509 는 공개 키 기반의 ITU-T 표준으로 인증서에서 확인할 수 있는 필드들을 정의하고 있다. 인증서에서 확인할 수 있는 필드 중 Optional 이 아닌 필드를 Table 1에서 확인할 수 있다. 이는 X.509 v3 인증서를 소개하고 있는 RFC 5280에서 확인할 수 있다[10]. 인증서 필드 중 발행자 (issuer) 와 소유자 (subject) 는 DN (Distinguished Name) 형식으로 표현하여 이름, 조직, 국가, 주소 등을 식별할 수 있게 한다[11]. Table 2에서 DN 필드를 확인할 수 있다.

Table 1. X.509 Certificates Fields

Field	Description
version	Version of Certificate
serialNumber	Certificate Serial Number
signature	Algorithm Identifier
issuer	Name of Issuer
validity	Validity Duration
subject	Name of Subject
subjectPublicKey Info	Public Key information of Subject
signatureAlgorht im	Signature Algorithm of Certificate
signatureValue	Signature of Certificate

Table 2. Distinguished Name Fields

String	Attribute Type
CN	commonName
L	localityName
ST	stateOrProvinceName
O	organizationName
OU	organizationUnitName
C	countyName
STREET	streetAddress
DC	domainComponent
UID	userID

3.2 인증서 체인

인증서 체인은 RFC 5280에서 정의된 인증서 경로 (Certification Path) 개념을 구체화한 것이다[10]. 최상위 인증 기관 (Root CA)에서 발급한 인증서를 비롯해 최종 인증서까지 나열된 인증서를 의미하며 신뢰 체인 (Chain of Trust)을 보장한다. 인증서에는 상위 인증 기관의 서명과 공개키가 포함되어 있기에 이러한 연속된 신뢰가 가능하다.

대표적으로 최상위 인증서, 중간 인증서, 최종 인증서로 3계층으로 이뤄진 구조를 주로 사용한다. 최상위 인증서는 최상위 인증 기관에서 스스로 서명(Self-Signed)하여 그 자체로 신뢰가 보장된 인증서를 의미한다.

중간 인증서 (Intermediate Certificate)는 중간 인증 기관이 최상위 인증 기관에 발급 요청하여 발급한 인증서를 말한다. 발급 요청 시 인증서의 내용이 최상위 기관에 전달된다. 최상위 인증 기관은 전달된 인증서 내용과 이 내용의 해시값을 개인 키로 서명한다. 발급된 중간 인증서는 최상위 기관의 서명을 받는다. 최상위 인증 기관과 중간 인증 기관까지의 신뢰는 사실상 불변하다.

최종 인증서를 발급하려면 중간 인증 기관에 인증서 발급 요청을 해야 한다. 중간 인증 기관 또한 최상위 인증 기관이 수행한 것처럼 인증서 발급 절차를 따른다. 최종 인증서는 중간 인증 기관의 서명을 포함한 채로 발급된다. 인증서 체인 구성을 위해 인증서의 서명 목록에 상위 기관들의 서명들이 포함되어 인증서가 발급된다.

보통은 최상위 기관으로부터 인증서 체인을 통한 신뢰가 구축되면 이러한 인증서는 믿을만하다고 여겨지나 무료로 TLS 인증서를 발급하는 비영리 기관인 Let's Encrypt 에 의해 공격자들 또한 이를 사용할 수 있는 상황이다. 따라서, 실험에서 수집할 인증서 중 Let's Encrypt 가 발급한 인증서에 대한 정보도 명령 제어 프레임워크 추적에 활용할 수 있는 척도라고 볼 수 있다.

3.3 코발트 스트라이크

본 논문에서는 상용 명령 제어 프레임워크 중 가장 대표적인 코발트 스트라이크 (Cobalt Strike) 도구를 대상으로 제안한 방법론에 대해 실험을 수행하였다.

코발트 스트라이크는 명령 서버인 팀 서버 (Team Server) 와 봇넷에 해당하는 비콘 (Beacon) 으로 이뤄져 있다[4]. Java 로 개발되어 2012년에 등장 후 현재까지 서비스되고 있는 코발트 스트라이크는 최근 3년간 발견된 명령 제어 프레임워크 서버 중 가장 많은 수를 기록하고 있다[6]. 사이버 범죄 그룹 중 명성을 떨친 Conti 랜섬웨어 그룹과 DarkSide 랜섬웨어 그룹이 코발트 스트라이크를 사용했다고 알려져 공격자들에게 각광 받는 도구로 볼 수 있다[3][12].

3.3.1 팀 서버

팀 서버는 코발트 스트라이크의 명령 서버이다. 비콘이 통신하는 리스너를 등록할 수 있으며 명령을 하달해 비콘이 이를 수행할 수 있도록 한다. 코발트 스트라이크의 사용자는 별도 클라이언트를 사용하여 팀 서버에 접속할 수 있으며 기본적으로는 50050 포트를 사용한다.

리스너의 지원 프로토콜은 HTTP/HTTPS, DNS, SMB, Raw TCP가 있다. 리스너를 팀 서버와 동일한 주소로 설정하여 비콘과 팀서버가 직접적으로 연결되도록 할 수 있다. 또한, 외부 명령 제어 (External C2)를 통해 비콘과 팀 서버가 직접적인 연결이 아닌 한 단계의 프록시 서버를 두는 형태로 간접적으로 통신하도록 설정할 수 있다. 이 연구에서는 HTTPS 리스너를 팀 서버와 동일한 주소로 설정한 경우에 대하여 추적을 수행하였다.

3.3.2 비콘

비콘은 코발트 스트라이크의 봇넷이다. 팀 서버에 등록된 리스너와 통신을 하게끔 되어 있으며 직간접적인 연결을 통해 최종적으로 팀 서버와 통신을 수행하는 구조이다.

비콘은 Stager 와 완전한 기능을 갖춘 백도어 두 개의 형태로 나뉜다. 3.1.1.에서 설명된 단계적 전달 기능에서 Stager 비콘을 사용한다. Stager 비콘은 쉘코드 형태의 경량 비콘이다. Stager는 감염 환경에 대한 기본적인 검사를 한 뒤, 리스너에 완전한 기능을 갖춘 백도어 비콘을 다운로드 요청하고 이를 실행시킨다.

3.3.3 Malleable C2 Profile

코발트 스트라이크 사용자는 Malleable C2 Profile 이라고 불리는 기능을 이용해 비콘이 수행할 통신과 비콘 설치 과정에 있어 세부적인 사항을 설정할 수 있다. 이는 코발트 스트라이크가 HTTP, DNS 등 다양한 통신 프로토콜을 지원할 수 있도록 한다. 또한, HTTP 프로토콜 상 포함되는 User-Agent 와 같은 헤더들을 포함하여 통신 주기, 인증서 관련 설정을 함으로써 APT 및 사이버

범죄 그룹에서 사용한 악성코드의 통신을 모사하거나 역으로 정상 트래픽을 모사할 수 있도록 한다.

Malleble C2 Profile 기능 사용을 위해 스크립트를 작성해야 한다. 설정 가능한 옵션은 글로벌 옵션과 로컬 옵션으로 나뉜다. 글로벌 옵션은 전체적인 비콘 설정을 변경하며 로컬 옵션은 통신 중 특정한 트랜잭션 하나에 대한 설정을 변경한다[13].

글로벌 옵션은 통신 주기, 서버 주소, 기본 User-Agent, Staging 여부 등을 포함하며 로컬 옵션은 한 트랜잭션과 관련된 설정과 감염 방법에 대한 설정이 가능하다.

글로벌 옵션 중, host_stage 라는 값이 있다. 이 값은 HTTP, HTTPS, DNS 와 같은 프로토콜에 의한 비콘의 단계적 전달 기능 활성화 여부를 말하며 기본적으로 true 로 설정되어 있다[13]. 이러한 특성 덕에, 코발트 스트라이크에서 단계적 전달로 비콘을 다운로드 받는 조건을 만족한다면 명령 서버로부터 비콘을 확보하는 것이 가능하다.

비콘과 명령 서버 통신 시, TLS 통신과 별개로 RSA와 AES를 이용한 암호화 통신이 수행된다. 비콘은 감염 시스템에 대한 정보와 AES 키를 포함한 메타데이터를 비콘 내부에 포함된 RSA 공개키로 암호화하여 서버로 전달한다. 서버는 개인키를 가지고 있기에 비콘으로부터 메타데이터 수신 후 암호화된 메타데이터를 복호화하여 AES Key를 획득함으로써 키 교환이 완료된다. 이에 관한 내용은 3.2.4. 비콘 설정 추출에서도 언급된다[14].

3.3.4 비콘 설정 추출

코발트 스트라이크는 앞서 설명된 Malleable C2 Profile 이라는 기능을 지원하고 있기에 비콘에서 이와 관련된 설정을 내부적으로 포함하고 있다[4]. 연구자들은 이러한 코발트 스트라이크 비콘으로부터 설정 값을 추출하는 비콘 설정 추출기를 개발하여 비콘 분석에 활용한다[8]. 본 연구에서는 SentinelOne에서 개발한 코발트 스트라이크 비콘 설정 추출기를 활용했으며 이 도구로 확인할 수 있는 옵션 중 일부를 Table 3에서 확인할 수 있다.

Table 3. Cobalt Strike Configuration

Option	Description
Beacon Type	Type of Beacon Communication
Port	Port of C&C Server
Sleep Time	Sleep Duration (ms)
C2 Server	Address of C&C Server
User Agent	Default User Agent while HTTP protocol is used
http-post Uri	URI of HTTP POST communication
Malleable C2 Instructions	Information about configured Malleable C2 Profile
http-get metadata	HTTP Request specification while http-get is used
http-post metadata	HTTP Request specification while http-post is used
watermark	Cobalt Strike license serial number
Public Key	RSA Public Key for AES Key Exchange

Beacon Type 은 비콘이 통신할 프로토콜의 종류를 의미하며 Port 와 C2 Server 는 비콘이 통신할 리스너의 주소 및 포트를 말한다. Sleep Time 은 마이크로초 단위의 통신 주기이다. User Agent 는 HTTP 프로토콜 사용 시 기본적으로 설정되는 User Agent 헤더 값을 뜻하며 http-post Uri 는 POST 전송에서 사용할 기본 URI가 설정된다. Malleable C2 Instructions 은 C2 Profile 에 설정된 행위들이 설명되어 있다.

http-get metadata 에서는 비콘이 명령 서버에 전송할 메타데이터의 인코딩 방법 등이 기술되며 http-post metadata 에는 비콘 당 할당된 세션 ID를 전달할 방법에 대해 적힌다[14].

상용 명령 제어 프레임워크인 코발트 스트라이크의 유효한 라이센스를 구분하기 위해 비콘 설정에 watermark가 포함되어 있으며 AES 키교환을 위한 RSA 공개키가 포함되어 있다.

RSA 개인 키와 관련된 파일은 서버에서 .cobaltstrike.beacon_keys 라는 이름으로 저장된다. 해당 파일이 존재하지 않다면 새로 해당 파일을 생성한다. 이 파일은 직렬화된 Java 클래스 파일로 팀 서버 프로그램에서 동적으로 로딩되어 사용되며 생성된 RSA 개인 키 내용도 포함하고 있다.

해당 파일은 고의로 삭제하지 않으면 계속 사용되

고 비콘과 서버 간 키교환 및 암호화 통신에 관여하기 때문에 비콘에서 추출된 공개키는 코발트 스트라이크 사용자들에 대한 추적 요소로도 활용할 수 있을 가능성이 있다.

3.4 쇼단 (Shodan)

네트워크에 연결된 장치들의 검색 엔진인 쇼단(Shodan)을 사용하면 악성코드의 명령 제어 서버 목록을 수집하는 것이 가능하다[7].

DarkComet RAT 생태계 연구에서도 관련된 서버 수집 출처 중 하나로 쇼단을 선택하여 사용했다[16]. 따라서, 쇼단으로 악성코드 관련 서버를 추적하는 것은 연구자들 사이에서 자연스러운 접근 방법이라고 볼 수 있다.

특히, 코발트 스트라이크의 경우, 쇼단 자체적으로 해당 서버가 코발트 스트라이크인지 구분해주는 기능을 지원하기 때문에 서버 목록 수집에 더욱 용이하다. 이 연구에서는 쇼단을 이용해 코발트 스트라이크 서버의 IP 목록을 수집할 예정이다.

3.4.1 JARM

JARM은 Salesforce 사에서 개발한 TLS 서버 핑거프린트 도구이다[15]. 3.1.2에서 설명되었듯이 TLS 통신을 위해서는 TLS Handshake 과정이 필요하다. Handshake 과정에서 클라이언트는 TLS ClientHello를 서버에 전송한다. ClientHello 에는 운영체제 버전, TLS 버전, 암호화 방법 등이 포함되어 전달되며 이를 수신한 서버는 ClientHello에서 전달된 내용을 토대로 지원 가능한 암호화 방법, 버전 등을 ServerHello 에 실어서 클라이언트로 전송한다[9]. JARM은 이러한 TLS 통신의 특성을 이용하여 JARM 해시를 생성한다. 생성 과정은 다음과 같다.

버전, 암호화 방법, 확장 (Extension) 이 각기 다르게 구성된 ClientHello 패킷을 10번 전송하고 서버의 응답을 받는다. 서버는 클라이언트가 전송한 값에 따라 다른 고유한 응답을 한다. 서버의 응답에 따른 해시 결과는 3 바이트로 총 30바이트의 결과가 도출된다. 이후 TLS Extension 에 대한 SHA256 해싱 결과인 32바이트가 뒤에 붙여져 62바이트에 해당하는 JARM 해시가 생성된다.

JARM을 이용하면 TLS 통신을 지원하는 명령 제어 프레임워크 서버에 대한 추적을 수행할 수 있으며 쇼단에서도 JARM에 기반한 검색을 지원하고 있다.

IV. 방법론 제안

4.1 명령 제어 프레임워크 관련 서버 목록 수집

네트워크 스캔을 통해 명령 제어 프레임워크와 관련된 서버 목록을 수집한다. 이를 위해서는 네트워크 스캔을 지원하는 쇼단이나 센시스(Censys) 와 같은 서비스를 이용하거나 별도의 네트워크 스캐너를 설계하여 사용한다. 3.1.2 절에서 언급되었듯이 명령 제어 프레임워크가 TLS 통신을 지원하기 때문에 3.3.1 절에서 언급한 JARM에 의한 스캔 또한 가

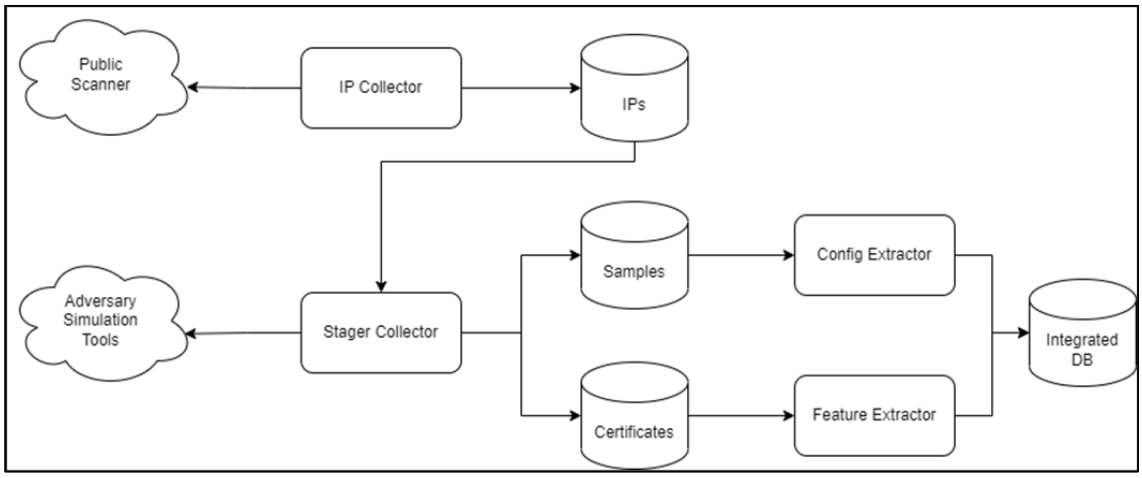

Fig. 4. Proposed System

능하다. TLS를 지원하지 않는다면 각 명령 제어 프레임워크 서버 고유의 탐지 가능한 특징을 사전 분석하고 이를 네트워크 스캐너에 반영하는 식으로 경우를 세분화 할 수 있다.

5.1. 실험 환경 구성에서는 쇼단을 이용하여 코발트 스트라이크 서버 목록을 수집하는 방법 및 관련된 쇼단 쿼리에 대한 내용을 설명한다.

4.2 단계적 전달 모사

3.1.1.에서 언급되었듯 명령 제어 프레임워크는 네트워크 탐지 우회 목적으로 단계적 전달을 지원한다. 명령 제어 프레임워크 서버에 단계적 전달 요청을 하는 것처럼 조건을 맞춘 패킷을 구성하여 전송한다면 서버는 추가 봇넷을 제공할 것이다. 명령 제어 프레임워크의 구현에 따라 이 단계의 난이도가 책정된다. 코발트 스트라이크의 경우에는 조건에 맞춘 HTTP 요청으로 이 과정이 가능하나, 다른 명령 제어 프레임워크는 해당 단계를 위한 추가 분석이 필요하다. 5.1. 실험 환경에서 코발트 스트라이크에서의 단계적 전달 모사를 설명한다.

명령 제어 프레임워크들은 설정에 따라 단계적 전달을 수행하도록 구현되어 있다. 코발트 스트라이크와 Metasploit 은 HTTP 핸들러에 전달된 URI이 4글자 이하이며 checksum8 계산 결과가 92일 경우에 추가 봇넷을 제공한다. Sliver 는 요청 URI가 .woff 로 끝나면 추가 봇넷을 제공한다. Koadic의 경우 무작위 5글자의 URI 가 설정되며 이 URI는 드로퍼의 설정에서 추출할 수 있고 Covenant, Empire 는 별도의 드로퍼에서 URI를 추출할 수 있게끔 되어 있다.

제안할 방법론에서 수행할 단계적 전달 모사는 코발트 스트라이크나 Metasploit, Sliver처럼 핸들러 서버에 정해진 규격의 HTTP 요청을 하여 추가 봇넷 파일을 확보하는 과정이라고 볼 수 있다.

4.3 봇넷 설정 추출

단계적 전달 모사를 통해 봇넷을 확보하면 봇넷 설정을 추출하여 서버 주소와 같은 추적에 유용한 정보 획득이 가능하다. 봇넷들마다 설정이 포함되어 있는 방법이 다르고 추출기가 개발되었는지에 따라 이 단계의 난이도가 다를 것이다.

코발트 스트라이크의 경우 오픈소스로 공개된 설정 추출기를 어렵지 않게 찾을 수 있다. 공개된 설정 추출기가 없는 명령 제어 프레임워크의 봇넷은 바이너리 분석을 통해 설정이 어떻게 저장되어 있는지 분석하고 설정 추출 방법을 고안하여 별도 추출기를 개발해야 한다.

4.4 인증서 수집 및 특징 추출

단계적 전달을 통해 봇넷을 제공한 서버가 HTTPS 등 TLS를 지원하는 서버일 경우 인증서를 수집한다. 3.1.2 절에서 설명되었듯, TLS Handshake 과정에서 서버와 클라이언트가 인증서를 주고받는다. 인증서에는 발행자와 소유자를 포함하여 서명, 서명 알고리즘, 공개키 등이 있으므로 이러한 특징들을 추출한다면 AV/EDR 등 보안 장비의 탐지 시그니처로 활용 가능하다. 또한 발행자와 소유자의 경우 HTTPS를 지원하는 도메인이 포함되는 경우도 있기에 추가 추적 요소로도 활용할 수 있다.

Table 4. Shdoan Queries

Idx	Shodan Query	Searched IP	Downloaded Beacon	Query Description
1	ssl.cert.serial:146473198	298	221	Search by Serial Number of Certificate
2	ssl.jarm:07d14d16d21d21d00042d41d00041de5fb3038104f457d92ba02e9311512c2	496	5	Search by Cobalt Strike related JARM hash
3	ssl.jarm:07d14d16d21d21d07c42d41d00041d24a458a375eef0c576d23a7bab9a9fb1+port:443	5547	10	Search by Cobalt Strike related JARM hash
4	product:'Cobalt Strike Beacon'	2489	730	Search by shodan feature to scan Cobalt Strike

3.2. 절에서 언급했듯, 인증서의 인증서 체인인증서는 X.509 포맷을 따르고 있기에 이와 관련하여 openssl 과 같은 암호화 지원 도구를 사용한다면 인증서에서 특징을 추출할 수 있다. 5.2. 분석 결과에서 수집된 인증서에 대한 내용이 등장한다. 제안한 방법론의 구조가 도식화된 그림은 Fig 4에서 확인할 수 있다.

V. 실험 및 분석

5.1 실험 수행

4절에서 제안한 방법론에 따라 상용 명령 제어 프레임워크인 코발트 스트라이크를 사전 추적할 수 있는 실험을 수행한다.

5.1.1 서버 목록 수집

방법론 중 첫 번째 단계인 명령 제어 프레임워크 관련 서버 목록 수집 수행을 위해 검색 엔진인 쇼단을 사용하였다. 코발트 스트라이크 서버 기본 인증서의 시리얼과 관련되어 공개된 JARM hash를 대상으로 검색을 수행하였다[17]. 또한 쇼단에서 제공하는 코발트 스트라이크에 서버 스캔 기능을 검색에 활용하였다. Table 4는 코발트 스트라이크 검색을 위해 사용한 쇼단 쿼리와 검색 결과를 나타낸다.

Table 4 의 Idx 1 에 해당하는 쿼리는 기본 인증서 시리얼에 해당하는 쿼리로 298개의 검색 결과에서 221개의 코발트 스트라이크 비콘을 다운로드 받을 수 있었다. Idx 2와 3은 공개된 JARM hash를 이용한 검색 쿼리이며 각각 496개, 5547개의 IP가 검색되었으나 실질적으로 유효한 비콘 다운로드는 5건, 10건으로 개수가 상당히 적은 것을 확인할 수 있다.

Idx 4는 쇼단에서 자체적으로 지원하는 코발트 스트라이크 전용 검색 쿼리이며 2489 건의 검색 결과 중 730건이 유효한 비콘 다운로드로 이어졌다. 또한, Idx 1,2,3에서 다운로드 되었던 221건, 5건, 10건의 비콘들이 Idx 4의 결과인 730건에 포함되었음을 확인되었다. 즉, Idx 1,2,3 의 조건에 맞지 않는 코발트 스트라이크 서버들이 상당수 존재한다는 것이며 특히, 공개된 JARM hash 로 검색하는 방법이 그리 효과적이지 않다는 것을 보여준다.

5.1.2 코발트 스트라이크 단계적 전달 모사

단계적 전달 모사를 통해서 쇼단 검색 결과로 획득한 IP 목록에 대하여 비콘 획득을 시도하였다. 코발트 스트라이크는 단계적 전달 모사 옵션이 기본적으로 활성화되어있다. 코발트 스트라이크의 단계적 전달 내부 구현상, 요청하는 URI가 4byte 이하이고 checksum8 결과가 92일 경우에 비콘이 제공된다. 따라서 이를 만족하는 aaa9를 요청하여 966개의 비콘 다운로드가 이뤄졌고 중복을 제외하여 730개의 비콘을 획득하였다.

IP 수집 시점과 직접 접속을 통한 비콘 확보 시점에 차이가 있어 검색된 IP 임에도 비콘 확보 시점에 접속이 불가한 경우가 일부 발생하였다.

쇼단의 경우 코발트 스트라이크 비콘에 대한 정보도 제공하고 있으나 실제 비콘 바이너리는 제공하고 있지 않다. 따라서, 접속 불가한 서버를 차치하고 비콘 확보 당시 접속 가능한 서버들에 초점을 맞춰 실험을 수행하였다.

5.1.3 코발트 스트라이크 설정 추출기

확보된 730개의 비콘을 대상으로 공개된 코발트 스트라이크 설정 추출기를 사용하여 추출을 진행하여 Table 3에 해당하는 설정 값들을 추출할 수 있었다[8].

630개 비콘에서 설정 추출이 가능하였고 설정 추출이 실패한 100개에 대한 분석은 해당 연구에서는 진행되지 않으나 차후 연구에서 분석될 가능성이 있다.

5.1.4 코발트 스트라이크 서버 인증서 수집

확보된 730개의 비콘 중, HTTPS 서버로부터 다운로드 된 비콘들에 대해서 인증서 수집이 수행됐다. 총 332개의 인증서가 수집되었으며 수집에 openssl 라이브러리가 사용되었다. 수집 항목으로는 소유자와 발행자, 인증서 체인, 유효 여부, 사용 알고리즘, 유효기간이 포함되었다. 같은 IP에서 여러 포트로 HTTPS 서버를 호스팅 하는 경우 등이 확인되었기에 한 IP에서 2개의 포트로 HTTPS를 서비스하는 경우 2개로 측정하는 식으로 계산하였다.

5.2 실험 결과 분석

실험 결과 Table 4의 1,2,3번 쿼리를 통한 IP 검색 결과가 4번 결과에 포함됨을 확인했다. 즉, 실질적으로 2489건의 IP를 대상으로 분석이 진행되었다. 이 중, HTTPS 서버는 1118건으로 확인되었으며 HTTP 서버는 1378건으로 확인되었다. 또한, 다운로드된 비콘은 중복 제거 730건이 확보되었고 이 중 설정 추출에 성공한 630건에 대해 분석하였다. HTTPS 서버 관련 JARM hash 와 서버 인증서 관련으로 분석했고 비콘 관점에서는 Table 3에서 설명되는 항목 중에 Beacon Type, http-post-uri, watermark, public key 총 4건에 대한 통계를 도출하였다.

5.2.1 추가적인 JARM hash

쇼단 검색에 사용된 JARM hash 2종을 제외하고 다른 JARM hash 가 사용되는 것이 파악되었다. Table 5. 에서는 HTTPS 서버 1118건에 대하여 상위 10개의 JARM hash를 나타내었다.

쇼단 검색을 통한 IP 수집 시점과 IP 접속을 통한 JARM hash 수집 시점에 차이가 있어 쇼단으로 검색된 IP이더라도 접속이 불가한 경우가 다수 발생하였다. 실제 접속이 가능했던 시점에서의 JARM 통계를 도출하기 위해 쇼단에서 제공하는 스캔 결과를 이용하였다.

Table 4. 의 2번 쿼리에 사용한 JARM 인 07d14d16d21d21d00042d41d00041de5fb3038104f457d92ba02e9311512c2 는 상위 10개에 들지 않았다. 이는 다운로드된 비콘 수가 5 건이기에 10위 밖에 집계되었기 때문이다. 3번 쿼리에 해당하는 07d14d16d21d21d07c42d41d00041d24a458a375eef0c576d23a7bab9a9fb1 는 30건을 차지하는 것을 확인할 수 있다. 위 JARM hash 의 검색 결과에서 비콘 다운로드 수는 10건에 해당하나 이는 열려져 있는 포트를 443으로 고정하여 검색했기 때문이다. 해당 JARM hash 는 전체에서 6위에 해당하나 1118건 중 30건인 상당히 적은 비율을 차지한다는 것을 알 수 있다.

이 결과로, 코발트 스트라이크에 한해서는 공개된 JARM Hash를 사용하여 관련 서버를 추적하는 것이 효용성이 상대적으로 떨어진다고 해석할 수 있을 것이다.

Table 5. Top 10 of Cobalt Strike Server JARM Hash

JARM Hash	Count
2ad2ad16d2ad2ad00042d42d00042ddb04deffa1705e2edc44cae1ed24a4da	517
07d14d16d21d21d00042d41d00041d47e4e0ae17960b2a5b4fd6107fbb0926	201
07d14d16d21d21d00042d43d00041de5fb3038104f457d92ba02e9311512c2	145
07d14d16d21d21d07c07d14d07d21d9b2f5869a6985368a9dec764186a9175	68
2ad2ad0002ad2ad0002ad2ad2ad2ade1a3c0d7ca6ad8388057924be83dfc6a	39
07d14d16d21d21d07c42d41d00041d24a458a375eef0c576d23a7bab9a9fb1	30
28d28d28d00028d00042d42d00042de0d120da90d5910c5de2af43b74a6bd8	14
07d13d15d21d21d07c07d13d07d21dd7fc4c7c6ef19b77a4ca0787979cdc13	13
07d14d16d21d21d07c42d41d00041d58c7162162b6a603d3d90a2b76865b53	10
1dd28d28d00028d00042d41d00041df1e57cd0b3bf64d18696fb4fce056610	9

또한, 다수에 해당하는 517건의 JARM hash (2ad2ad16d2ad2ad00042d42d00042ddb04deffa1705e2edc44cae1ed24a4da) 를 서버 목록 수집 과정에서 이용하는 식으로 실험 재구성을 할 수 있으며 추가적인 코발트 스트라이크 서버 추적에 활용할 수 있을 것으로 판단된다.

5.2.2 인증서 통계

332개의 HTTPS 서버에서 수집한 인증서의 유효 여부 상태는 자가 서명 (self-signed), 발행자 정보 찾을 수 없음 (unable to get local issuer

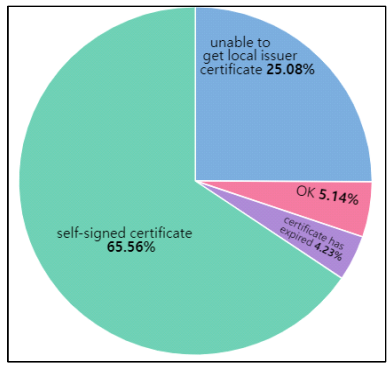

Fig. 5. Statistics of Certificate Validity

certificate), 유효 기간 만료 (certificate has expired), 유효(OK) 로 구분한다. 각 유효 여부 상태에 대한 통계는 Fig 5.에서 나타내고 있다. 자가 서명은 217건 (65.56%) 으로 가장 많은 비율을 차지하고 있으며 발행자 정보 찾을 수 없음이 83건 (25.08%), 유효 상태가 17건(5.14%), 유효 기간 만료는 14건(4.23%)이 확인되었다.

5.2.3 자가 서명 인증서

자가 서명된 인증서는 발행자와 소유자가 같은 경우를 말하며 최상위 인증 기관에서 발급하는 경우를 제외하고 일반적으로 신뢰받지 않는다. 자가 서명된 인증서의 발행자, 소유자 정보를 상위 10개만 추려서 Table 6.에 정리하였다.

Table 6. Top 10 of Self-Signed Certificate

DN of Issuer & Subject	Count
i:C = , ST = , L = , O = , OU = , CN =	134
i:C = US, ST = , L = , O = jQuery, OU = Certificate Authority, CN = jquery.com	13
i:C = KZ, ST = KZ, L = , O = NN Fern Sub, OU = NN Fern, CN = foren.zik	8
i:C = CN, ST = Cyberspace, L = Somewhere, O = Software, OU = baidu.com, CN = CN	3
i:C = US, ST = WA, L = Redmond, O = Microsoft Corporation, OU = Microsoft Corporation, CN = *.microsoft.com	3
i:C = US, ST = WA, L = Redmond, O = Microsoft Corporation, OU = Microsoft IT, CN = www.windowsupdate.com	3
i:C = CN, ST = , L = , O = Juntan JiTuan, OU = Juntan JiTuan, CN = Juntan	2
i:C = CN, ST = \E5\8C\97\E4\BA\AC, L = \E5\8C\97\E4\BA\AC, O = \E8\BF\90\E7\BB\B4\E9\83\A8, OU = \E5\8C\97\E4\BA\AC\E5\A5\87\E5\AE\89\E4\BF\A1\E7\A7\91\E6\8A\80\E6\9C\89\E9\99\90\E5\85\AC\E5\8F\B8, CN = www.qianxin.com	2
i:C = CN, ST = zhejiang, L = zhejiang, O = zjzwfw.gov.cn, OU = zjzwfw.gov.cn, CN = China	2
i:C = Earth, ST = Cyberspace, L = Somewhere, O = cobaltstrike, OU = AdvancedPenTesting, CN = Major Cobalt Strike	2

주목 할 만 한 점은 해당 표의 10번째 항목에서 코발트 스트라이크 명칭을 그대로 사용하며 인증서를 발급했다는 것이다. 건수로는 2건이 확인되었으나 코발트 스트라이크가 본래 모의 침투를 위해 개발된 도구이기에 해당되는 서버가 이러한 레드팀 활동을 위한 서버라는 것을 명시하려는 의도로 추정된다.

5.2.4 Let's Encrypt 인증서

유효 상태는 인증서 체인에 의해 일반적으로 신뢰받는 최상위 인증 기관과 중간 인증 기관이 최종 인증서를 발급하고 서명되었으며 유효기간이 지나지 않은 상태이다.

유효 상태의 인증서들은 비영리 재단 Let's Encrypt에서 발급 및 서명한 인증서들로 확인되었다. Let's Encrypt 재단은 HTTPS 공급을 위해 무료로 TLS 인증서를 발급하기 때문에 유효 상태의 인증서가 수집될 수 있었던 것으로 보인다. Table 7.에서 Let's Encrypt에서 서명된 인증서에 해당하는 도메인을 나열하였다.

Table 7. Verified by Let's Encrypt Domain of Cobalt Strike

Domain	Count
repository-tools.com	3
brosift.com	1
dbx.formsift.io	1
ersanca.com	1
ggstech.xyz	1
integrated-security.net	1
lkjsdlkjjlkfdafdalkjfdslkjs.northeurope.cloudapp.azure.com	1
mailapps-development-ms.com	1
pfizer.eastus.cloudapp.azure.com	1
pull.jira-software.com	1
raycestyle.com	1
spotifyus.shop	1
straxotechnology.com	1
svchosexec.com	1
www.fzupdate.com	1

이상 행위 탐지에 있어 인증서가 유효하면 검사를 생략하는 정책을 펼칠 때가 있다. 해당 도메인의 소유자가 침투 테스트를 하는 보안 조직인지 사이버 범죄자인지는 알기 어려우나 공격자들이 탐지 우회를 위해 Let's Encrypt 와 같은 유효한 인증서를 발급하여 활용한다는 것을 관측할 수 있다.

5.2.5 비콘 타입

추출된 코발트 스트라이크 비콘의 Beacon Type은 HTTP 가 362건으로 57.46%를 차지하고 HTTPS가 268건으로 42.54%를 차지했다. Beacon Type 차트를 Fig 6.에서 나타낸다. 설정 추출이 되지 않은 100건 중에서 64건이 HTTPS에 해당 되었기에 총 332개로 SSL 인증서 수집 건과 일치한다.

이 통계를 보면 공격자들이 코발트 스트라이크를 사용하는데에 HTTPS 보다는 HTTP를 더 선호하는 경향이 있으며 이는 HTTPS 서버 설정에 인증서가 필요하고 세팅에 더 시간이 들기 때문에 상대적으로 간단한 HTTP 서버 사용을 선호하는 것으로 보인다.

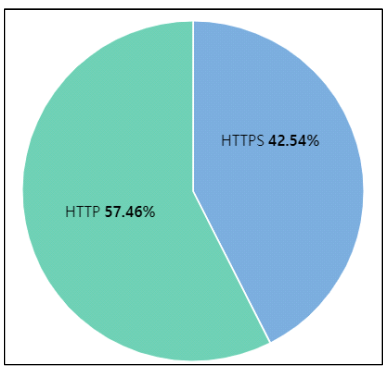

Fig. 6. Statistics of Beacon Type

5.2.6 http-post-uri

명령 서버와 POST 통신 시 사용하는 URI 설정인 http-post-uri 설정에 대해서 상위 10종을 Table 8. 에 기재했다. /submit.php 가 357개로 제일 많은 수를 기록했다.

/submit.php 는 코발트 스트라이크의 Malleable C2 profile 의 기본 설정으로 공격자들이 코발트 스트라이크를 사용할 때 주로 기본 값을 유지한 채 사용한다는 것을 관측할 수 있다[20].

Table 8. Top 10 of http-post-uri

http-post-uri	Count
/submit.php	357
/jquery-3.3.2.min.js	43
/index/rss.php	23
/admin/user	13
/api/postit	12
/api/y	7
/email/	7
/owa/qs3XxFyeDCOvFfLvl8k-fR2nT3nsFnYJ	7
/IMXo	6
/N4215/adj/amzn.us.sr.aps	6

이러한 지표로 네트워크 상에서 코발트 스트라이크 비콘 탐지에 활용할 수 있고 기계 학습으로까지 연구를 연장시킬 수 있을 의미 있는 데이터라고 볼 수 있다.

5.2.7 워터마크와 공개키

상용 명령 제어 프레임워크인 코발트 스트라이크의 구매 라이센스를 나타내는 워터마크에 대해서도 상위 10종을 도출하여 Table 9. 에 나열했다. 워터마크는 십진수로 표현되며 1,234,567,890과 0이 각각 114건, 73건으로 큰 비중을 차지함을 확인할 수 있다. 같은 코발트 스트라이크 소프트웨어는 같은 워터마크를 가지기 때문에 워터마크 추적을 통해 위

Table 9. Top 10 of watermark

Watermark	Count
1,234,567,890	114
0	73
391,144,938	69
305,419,896	60
100,000	59
426,352,781	48
12,345	42
1,580,103,824	26
674,054,486	18
1,359,593,325	13

협 행위자들 간 연관성을 파악하는 데에도 활용할 수 있다.

3.3.3에서 설명되었듯, 비콘과 명령 서버 통신 간, TLS 통신과는 별개로 RSA 와 AES 알고리즘을 이용한 암호화 통신이 수행된다. 이를 지원하기 위해 비콘에서는 RSA 공개키가 포함되어 있다. 공개키도 워터마크와 마찬가지로 같은 코발트 스트라이크 소프트웨어를 사용한다면 키 쌍을 임의로 재설정하지 않는 한 같은 키를 가진다. 따라서, 비콘에서 추출된 공개키 추적을 통해 코발트 스트라이크 비콘을 사용하는 공격자들의 연관성 파악에 도움이 될 것이다. 이에 대한 내용은 바로 다음 절인 5.3. 절에서 상세히 설명한다.

Table 10. Top 10 of Public Key (MD5)

MD5 of Public Key	Count
f8cb37f8c8daf1d31cbd0572766c95a9	42
c0bf697e08008f2b6fce7cc8f76e8550	28
1a5779a38fe8b146455e5bf476e39812	19
8b02c223ce146b158dd3b65a679da7c6	7
887b8c9e3ea19c2b91db143db064ee1c	6
ae92f1c136e00a08f17254f875a2ab38	6
476a3d578e65d28645a85577324cb479	5
f34636101878a4e11841d6f8f3e21878	5
0ec1db04a860520adfe4e6e2989dfc0c	4
1bf15ba2909f7c9aabe266ae92988e35	4

5.3 분석 데이터 활용 사례

제안한 방법론을 코발트 스트라이크에 적용한 결과로 봇넷에서 추출한 설정으로부터 각 코발트 스트라이크 비콘들의 공개키와 워터마크를 획득할 수 있었다. 이 2개 지표는 코발트 스트라이크를 사용하는 그룹들을 추적하는 데에 활용될 수 있다.

대표적인 예시로 TrendMicro 의 Earth Longzhi 캠페인 분석을 들 수 있다. TrendMicro 는 APT41 의 서브그룹인 Earth Longzhi 관련 캠페인에 대해서 분석하였다.

분석된 Earth Longzhi 관련 캠페인에서 사용된 코발트 스트라이크의 공개키와 워터마크가 또 다른 APT41 의 서브그룹인 Earth Baku 의 것과 일치하다는 점을 Earth Longzhi 가 APT41 의 서브그룹인 근거로 삼았다. 본 연구에서 수행한 방법론으로 수집한 코발트 스트라이크 비콘에서 426,352,781 이라는 워터마크가 추출되었다. 이는 Earth Longzhi 캠페인 분석 글에서도 언급된다[18].

이처럼 코발트 스트라이크의 봇넷인 비콘을 확보하고 비콘에서 추출된 설정으로 위협 행위자 그룹 추적에 사용할 수 있듯이 다른 명령 제어 프레임워크 유사한 방법으로 활용될 수 있을 것으로 보인다.

VI. 결 론

명령 제어 프레임워크는 모의 침투 및 교육을 위해 개발되었으나 강력한 기능 때문에 위협 행위자들에게도 악용된다. 본 연구에서는 악용된 명령 제어 프레임워크 추적을 위해 서버 목록 수집, 단계적 전달 모사, 봇넷 설정 추출, 인증서 수집까지 4개의 절차로 구성된 방법론을 제안하였다.

기존 연구들은 네트워크 관점에서 C2 서버를 핑거프린팅하거나 트래픽 탐지에 중점을 맞췄다. 그렇기에 이 연구에서는 명령 제어 프레임워크의 구성 요소인 봇넷을 다루기 위해 봇넷을 직접적으로 내려 받는 단계적 전달 모사 과정을 포함시켰다. 확보한 봇넷에서 설정을 추출함으로써 추가적인 데이터 확보를 할 수 있도록 방법론을 구성 하였고 제안한 방법론을 공격자들이 가장 선호하는 상용 명령 제어 프레임워크인 코발트 스트라이크에 적용하여 실험을 수행하였다.

실험 결과, 서버 목록 수집 단계에서 쇼단을 이용하였고 2489개의 IP를 수집했으며 730개의 비콘을 확보했다. 이 중, 630개 비콘에서 설정을 추출에 성공했고 비콘을 확보한 HTTPS 서버에서 인증서를 수집하여 332개의 인증서를 수집할 수 있었다.

수집된 인증서의 서명 상태를 분석하여 자가 서명된 서명과 유효한 서명을 확인했으며 유효한 서명은 Let's Encrypt를 통해 발급되었음을 파악했고 이에 따른 공격자들의 전략을 관찰할 수 있는 계기가 되었다.

추출된 비콘 설정 중, POST 통신에 사용되는 URI 와 유효한 라이센스를 나타내는 워터마크, 명령 서버와 암호화 통신 시 사용하는 공개키에 대한 통계를 도출함으로써 코발트 스트라이크 탐지 및 추적에 대한 정보를 제공했고 명령 제어 프레임워크 추적에 대한 기반을 마련하였다. 워터마크와 공개키는 기존에 코발트 스트라이크를 사용했던 위협 행위자 그룹을 추적에 대표적으로 사용되는 지표이기에 이를 활용한다면 그룹 추적에 충분히 기여를 할 수 있을 것으로 보인다.

차후 확보된 정보를 토대로 기계 학습에 연계하거나 비콘 설정 추출에 실패했던 경우를 조사하여 실험에 사용되었던 설정 추출기 개선, 쇼단 뿐 아닌 다른 스캐너를 사용하거나 스캔 방법을 달리하는 식으로 추가 연구 방향을 잡을 수 있다.

또한, 본 연구에서 제안한 4단계로 이뤄진 방법론은 비단 코발트 스트라이크 뿐 아니라 단계적 전달을 지원하는 다른 명령 제어 프레임워크에도 적용할 수 있을 것으로 예상되며 명령 제어 프레임워크 추적에 있어 방향 기반을 설정하는 데에 도움이 될 수 있을 것으로 보인다.

References

[1] A. B. Ajmal, M. A. Shah, C. Maple, M. N. Asghar and S. U. Islam, "Offensive Security: Towards Proactive Threat Hunting via Adversary Emulation," IEEE Access , vol. 9, pp. 126023-126033, 2021.

[2] J. Piet, B. Anderson and D. McGrew, "An In-Depth Study of Open-Source Command and Control Frameworks," 2018 13th International Conference on Malicious and Unwanted Software (MALWARE), pp. 1-8, 2018.

[3] TJ OConnor, "HELO DarkSide: Breaking Free From Katas and Embracing the Adversarial Mindset in Cybersecurity Education", Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1 (SIGCSE 2022), vol. 1. Association for Computing Machinery, pp. 710–716, 2022.

[4] Ejik, Vincent van der, Coen Schuijt and Ralph Koning. "Detecting Cobalt Strike beacons in NetFlow data." Master Thesis, University of Amstredam, 2020.

[5] F. Aldauiji, O. Batarfi and M. Bayousef, "Utilizing Cyber Threat Hunting Techniques to Find Ransomware Attacks: A Survey of the State of the Art," IEEE Access, vol. 10, pp. 61695-61706, 2022.

[6] Recorded Future, "2022 Adversary Infrastructure Report", https://go.recordedfuture.com/hubfs/reports/cta-2022-1215.pdf, accessed: Jan. 2023.

[7] M. Bada and I. Pete, "An exploration of the cybercrime ecosystem around Shodan," 2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), pp. 1-8, 2020.

[8] SenitnelOne, "CobaltStrikeParser", https://github.com/Sentinel-One/CobaltStrikeParser, accessed: Jan. 2023.

[9] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, Aug. 2018.

[10] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[11] Wahl, M., Kille, S., and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC 2253, Dec. 1997.

[12] U.S Department of Defense, "Conti Ransomware", https://media.defense.gov/2021/Sep/22/2002859507/-1/-1/0/CSA_CONTI_RANSOMWARE_20210922.PDF, accessed: Jan. 2023.

[13] HelpSystems, "Profile Language", https://hstechdocs.helpsystems.com/manuals/cobaltstrike/current/userguide/content/topics/malleable-c2_profile-language.htm, accessed: Feb. 2023.

[14] HelpSystems, "A Beacon HTTP Transaction Walk-Through", https://hstechdocs.helpsystems.com/manuals/cobaltstrike/current/userguide/content/topics/malleable-c2_beacon-http-transaction

-walkthru.htm, accessed: Mar. 2023.
〔15〕 Salesforce, "JARM", https://github.com/salesforce/jarm, accessed: Jan. 2023.
〔16〕 Brown Farinholt, Mohammad Rezaeirad, Damon McCoy, and Kirill Levchenko, "Dark Matter: Uncovering the DarkComet RAT Ecosystem." Proceedings of The Web Conference 2020 (WWW '20) Association for Computing Machinery, pp. 2109-2120, 2020.
〔17〕 Raphael Mudge, "A Red Teamer Plays with JARM", https://www.cobaltstrike.com/blog/a-red-teamer-plays-with-jarm/, accessed: Feb. 2023.
〔18〕 TrendMicro, "Hack the Real Box: APT 41's New Subgroup Earth Longzhi", https://www.trendmicro.com/en_gb/research/22/k/hack-the-real-box-apt41-new-subgroup-earth-longzhi.html, accessed: Mar. 2023.
〔19〕 Y. Nakamura and B. Åström, "Scanning and Host Fingerprinting Methods for Command and Control Server Detection," Dissertation, 2021.
〔20〕 Unit42 of PaloaltoNetworks, "Cobalt Strike Analysis and Tutorial: How Malleable C2 Profiles Make Cobalt Strike Difficult to Detect", https://unit42.paloaltonetworks.com/cobalt-strike-malleable-c2-profile/, accessed: Mar. 2023.
〔21〕 Unit42 of PaloaltoNetworks, "Cobalt Strike Analysis and Tutorial: Identifying Beacon Team Servers in the Wild", https://unit42.paloaltonetworks.com/cobalt-strike-team-server/, accessed: Jul. 2023.

〈저자소개〉

권 혁 주 (Hyeok-Ju Gwon) 정회원
2019년 8월: 아주대 사이버보안학과 학사
2021년 9월~현재: 아주대학교 사이버보안학과 석사과정
〈관심분야〉 악성코드, 해킹, 시스템보안

곽 진 (Jin Kwak) 종신회원
2000년 8월: 성균관대학교 학사
2003년 2월: 성균관대학교 석사
2006년 2월: 성균관대학교 박사
2006년 4월~2006년 11월: 일본 큐슈대학교 방문연구원
2006년 8월~2006년 11월: 일본 큐슈시스템정보기술연구소 특별연구원
2006년 11월~2007년 2월: 정보통신부 정보보호기획단 개인정보보호팀 통신사무관
2007년 3월~2015년 2월: 순천향대학교 정보보호학과 교수
2008년 1월~현재: 한국정보보호학회 상임이사
2011년 1월~현재: 한국정보처리학회 이사
2015년 3월~현재: 아주대 사이버보안학과 교수
〈관심분야〉 컨텐츠 보안, 암호프로토콜, 응용시스템보안, 클라우드 컴퓨팅 보안, 개인정보보호, 정보보호제품평가